



REST API Reference

NorthStar Controller 6.1 (April 21, 2023)



REST API Reference

NorthStar Controller 6.1 (2023-04-21)

Copyright © 2010-2016 Juniper Networks All rights reserved.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Table of Contents

Working with Pathfinder RESTful APIs	viii
1. REST basics	viii
1.1. How you're already using REST	viii
1.2. Patterns for Pathfinder RESTful Web services	ix
1. Paragon Authentication	1
1.1. IAM	1
1.2. Paragon IAM Authentication API	1
1.2.1. Get ID Token	2
2. Paragon PathFinder RESTCONF API	3
2.1. RESTCONF Endpoint	4
2.1.1. URIs	4
2.2. RESTCONF HTTP Operations	4
2.3. RESTCONF Query Data	5
2.4. Authentication	5
2.5. ietf-network YANG Model	5
2.5.1. Get container ietf-network:networks	7
2.5.2. Get container ietf-network:networks	242
2.6. ietf-te YANG Model	242
2.6.1. Get container ietf-te:te	243
2.6.2. Get container ietf-te:te	247
2.6.3. Insert in container named-admin-groups	248
2.6.4. replace container named-admin-groups	249
2.6.5. modify container named-admin-groups	250
2.6.6. delete container named-admin-groups	251
2.7. ietf-yang-library YANG Model	251
2.7.1. Get container ietf-yang-library:modules-state	252
2.8. Supported YANG operations	254
2.8.1. execute RPC ietf-te-operations:tunnel-atomic-create	256
2.8.2. execute RPC ietf-te-operations:tunnel-atomic-delete	258
2.9. Network Slicing	258
2.9.1. Get slice id information via topology-group	260
2.9.2. Adding slice ID to Node	262
2.9.3. Delete slice ID from Node	263
2.9.4. Bulk adding slice ID to Nodes	264
2.9.5. Bulk deleting slice ID from Nodes	269
2.9.6. Adding slice ID to Link	274
2.9.7. Delete slice ID from Link	276
2.9.8. Adding LSP with slice	278
3. Paragon PathFinder API	281
3.1. Notification API	300
3.2. Topology	302
3.2.1. Get List of Topologies	304
3.2.2. Get Topological Elements	305
3.2.3. Delete the Topology	319
3.2.4. Start a SSE Stream	320
3.2.5.	321
3.2.6. Get PCEP Protocol Adapter Status	322
3.2.7. Get BGP-LS Protocol Adapter Status	323

3.2.8. Get Path Computation Server Status	324
3.2.9. Get Transport Topology Acquisition Daemon Status	325
3.2.10. Get Analytics Data Collection Component Status	326
3.2.11. Get Analytics Data Collection Database Status	327
3.3. Node	327
3.3.1. Get Nodes	329
3.3.2. Create a Node	331
3.3.3. Search Nodes	334
3.3.4. Start a SSE Stream	337
3.3.5. Get a Single Node	338
3.3.6. Update Node Properties	339
3.3.7. Patch Node	340
3.3.8. Delete a Node	341
3.3.9. Get the Node Event History	342
3.3.10. Create a List of Nodes	345
3.3.11. Update a List of Nodes	346
3.3.12. Update a List of Nodes using PATCH	347
3.3.13. Delete a List of Nodes	348
3.4. Links	348
3.4.1. Get Links	350
3.4.2. Create a Link	353
3.4.3. Search Links	358
3.4.4. Start a SSE Stream	362
3.4.5. Get a Single Link	363
3.4.6. Update Link	365
3.4.7. Patch Link	367
3.4.8. Delete a Link	369
3.4.9. Get Links utilization only	370
3.4.10. Get the Link Event History	372
3.5. TE-LSPs	374
3.5.1. Get TE-LSPs	376
3.5.2. Create a TE-LSP	389
3.5.3. Search LSPs	404
3.5.4. Start a SSE Stream	416
3.5.5. Get a Single TE-LSP	417
3.5.6. Update a TE-LSP	419
3.5.7. Patch a TE-LSP	421
3.5.8. Delete a TE-LSP	422
3.5.9. Create a List of TE-LSPs	423
3.5.10. Update a List of TE-LSPs	425
3.5.11. Update a List of TE-LSPs using PATCH	426
3.5.12. Delete a List of TE-LSPs	427
3.5.13. Get the LSP Event History	428
3.6. Demands	431
3.6.1. Gets all Demands	433
3.6.2. Create a Demand	436
3.6.3. Start a SSE Stream	438
3.6.4. Get a Specific Demand	439
3.6.5. Update a Specific Demand	441
3.6.6. Update a Specific Demand using PATCH	443
3.6.7. Delete a Specific Demand	445

3.6.8. Create a set of Demands	446
3.6.9. Update a set of Demands	449
3.6.10. Patch a set of Demands	452
3.6.11. Delete a set of Demands	454
3.7. Task scheduler	454
3.7.1. Gets all Tasks	456
3.7.2. Get tasks based on filter tasktype	458
3.7.3. Update a task	463
3.7.4. Remove and add a new Task	464
3.7.5. Delete tasks	465
3.7.6. Gets task execution status history in HTML table format for a given taskId	466
3.7.7. Gets task execution status history for a given taskId	470
3.8. Configuration	471
3.8.1. Get Paragon Pathfinder component configurations	473
3.8.2. Updates all component configurations	478
3.8.3. Patch some component configurations	482
3.8.4. DELETE configuration	484
3.9. Device Profiles	484
3.9.1. Get all Profiles	486
3.9.2. Create New Profiles	487
3.9.3. Update Profiles	488
3.9.4. Delete Profiles	489
3.9.5. Create a New Collect	490
3.9.6. Get the Status of a Collection Job	491
3.10. TE-Containers	491
3.10.1. getAllTEContainers	493
3.10.2. createTEContainer	494
3.10.3. Start a SSE Stream	495
3.10.4. getATEContainer	496
3.10.5. updateContainer	497
3.10.6. patchContainer	498
3.10.7. deleteContainer	499
3.10.8. createContainerBulk	500
3.10.9. updateContainerBulk	501
3.10.10. patchContainerBulk	502
3.10.11. deleteContainerBulk	503
3.11. Facilities	503
3.11.1. getAllFacilities	505
3.11.2. createAFacility	506
3.11.3. Start a SSE Stream	507
3.11.4. getAFacility	508
3.11.5. updateFacility	509
3.11.6. deleteFacility	510
3.12. P2MP	510
3.12.1. Get P2MP groups	512
3.12.2. Create a P2MP group	514
3.12.3. Start a SSE Stream	522
3.12.4. Get a Specific P2MP Group	523
3.12.5. Update a Specific P2MP Group	526
3.12.6. Update a Specific P2MP Group using a PATCH document	528

3.12.7. Delete a P2MP Group	534
3.12.8. Delete list of P2MP Groups	535
3.12.9. Create a P2MP Leaf	536
3.12.10. Get a Specific P2MP Leaf	537
3.12.11. Delete a P2MP Leaf	540
3.12.12. Delete list of P2MP leaves	541
3.13. VRF	541
3.13.1. Get the nodes with VRF	542
3.13.2. List the VRF on a given node	544
3.14. BGP Routes	545
3.14.1. getRoutesSummary	546
3.14.2. getNodeRoutes	548
3.15. IPE Policies	549
3.15.1. getIPEPolicies	551
3.15.2. createIPEPolicies	553
3.15.3. updateIPEPolicies	554
3.15.4. deleteIPEPolicies	555
3.15.5. getIPEPolicy	556
3.16. Analytics API	556
3.17. Transport Topology acquisition	568
3.17.1. Get Transport Controller List	570
3.17.2. Create Transport Controller Configuration	571
3.17.3. Get Transport Controller Configuration	572
3.17.4. Update Transport Controller Configuration	573
3.17.5. Delete Transport Controller Configuration	574
3.18. Transport Topology devices	574
3.18.1. Get Transport Controller Device Groups	576
3.18.2. Create A Transport Controller Device Group	577
3.18.3. Delete a Transport Controller Device Group	578
3.18.4. Get Transport Controller Device Groups	579
3.18.5. Create New Devices in a Group	580
3.18.6. Update Devices in the Group	581
3.18.7. Delete Devices in the Group	582
3.19. Path Computation API	582
3.19.1. computePath	583
3.20. Maintenance API	584
3.20.1. Get Maintenance	585
3.20.2. Create a new maintenance	587
3.20.3. Start a SSE Stream	589
3.20.4. Get a Specific Maintenance	590
3.20.5. Update a Specific Maintenance	591
3.20.6. Delete a Specific Maintenance	593
3.21. RPCs	593
3.22. Path Optimization RPC	593
3.22.1. global_Optimize.post	594
3.22.2. global_Optimize.get	595
3.22.3. targeted_Optimize.post	596
3.23. Routing and Failure Simulation RPC	596
3.23.1. Failure Simulation optimization RPC	598
3.23.2. Get Simulation optimization RPC list	600
3.23.3. Get a specific simulation	602

3.23.4. Get a specific report of one simulation	604
3.24. P2MP Tree design RPC	605
3.24.1. P2MP Diverse Tree Design optimization RPC	607
3.24.2. Get Simulation optimization RPC list	615
3.24.3. Get a Specific Simulation optimization RPC Result	616
3.24.4. P2MP Diverse Tree Design optimization with flow mapping	624
3.25. Notification examples	625
3.25.1. Node events	625
3.25.2. Link events	626
3.25.3. TE-LSPs events	627
3.25.4. P2MP Groups	628
3.25.5. Node events	629
3.25.6. Topology events	629
3.25.7. High Availability events	630

Working with Pathfinder RESTful APIs

Use the Juniper Pathfinder APIs to monitor and control your TE-LSPs.

The current API definitions can be found on the [Pathfinder API v2](#) page.

You can use the APIs and extensions after you authenticate through the authentication API.

The Pathfinder RESTful APIs are designed to enable access over HTTP to most of the same data and analytics that are available to you from both the Pathfinder GUI and the Pathfinder CLI.

1. REST basics

Many Pathfinder users have little or no prior experience developing programs or scripts that make use of RESTful APIs. For this reason, we are providing a brief introduction designed to help you with some basics about REST and what it means to access data and functionality using common conventions associated with REST. In addition, we are providing a few simple code examples that illustrate common uses of specific APIs.

REST (REpresentational State Transfer) is a set of useful conventions and principals about transfer of information over the World Wide Web.

1.1. How you're already using REST

Many Web services are now using the principals of REST in their design.

When you type a URL into your browser, like `http://example.net`, your browser software creates an HTTP header that identifies:

- a desired action: GET ("get me this resource").
- a target machine (`www.domain-name.com`).

When you update your credit card information to an online account, your browser software creates an HTTP header with a payload that contains your form data;

The HTTP header identifies:

- a desired action: POST ("here's some update info").
- a target machine (`www.domain-name.com`).

The payload contains:

- form data to POST to a resource on the target machine.

1.1.1. Common HTTP verbs

The HTTP verbs (also known as methods) that are commonly used in RESTful requests are POST, GET, PUT, and DELETE. These are known as CRUD, for create, read, update, and

delete operations. Additional HTTP methods exist, but are used less frequently, including PATCH, OPTIONS and HEAD, among others, but these are not used in the Pathfinder APIs

Below is a table summarizing some helpful information about common HTTP methods:

Table 1. CRUD

HTTP Verb	CRUD	Entire Collection (e.g. /customers)	Specific Item (e.g. /customers/{id})
POST	Create	201 (Created), 'Location' header with link to /customers/{id} containing new ID.	404 (Not Found), 409 (Conflict) if resource already exists..
GET	Read	200 (OK), list of customers. Use pagination, sorting and filtering to navigate big lists.	200 (OK), single customer. 404 (Not Found), if ID not found or invalid.
PUT	Update/Replace	404 (Not Found), unless you want to update/replace every resource in the entire collection.	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
DELETE	Delete	404 (Not Found), unless you want to delete the whole collection—not often desirable.	200 (OK). 404 (Not Found), if ID not found or invalid.

1.2. Patterns for Pathfinder RESTful Web services

RESTful Web services are commonly used for many familiar transactions over the Web, and their use is expanding.

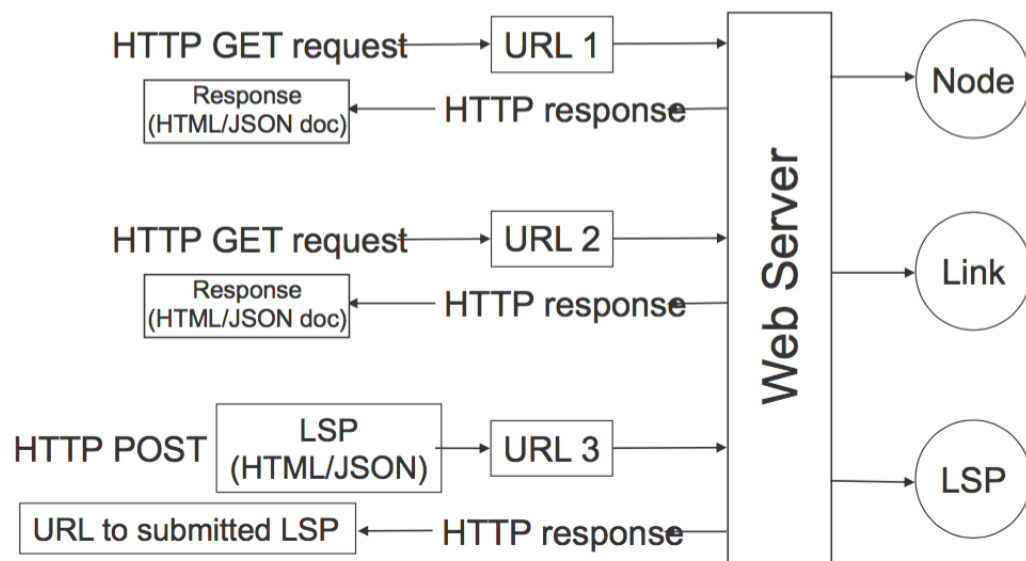
RESTful requests to Pathfinder will often follow patterns similar to the following request, which retrieves a full topology:

https://paragon.example.net/traffic-engineering/api/topology/
v2/1

The data in a response message body can range from a single entry to thousands of lines, depending on the request. The following example illustrates the kind of data you might see returned in response to a topology request:

Figure 1. Example topology data

```
[{"nodeType": "link", "topologyIndex": 0, "name": "0840.0000.0105.02", "nodeIndex": 6, "AutonomousSystem": {"asNumber": 84}}, {"topObjectType": "node", "topologyIndex": 0, "name": "0840.0000.0105.03", "nodeIndex": 7, "AutonomousSystem": {"asNumber": 84}}, {"topObjectType": "node", "topologyIndex": 0, "name": "0840.0000.0106.02", "nodeIndex": 9, "AutonomousSystem": {"asNumber": 84}}, {"topObjectType": "node", "topologyIndex": 0, "name": "84.0.0.101", "nodeIndex": 1, "AutonomousSystem": {"asNumber": 84}}, {"topObjectType": "node", "topologyIndex": 0, "name": "84.0.0.102", "nodeIndex": 2, "AutonomousSystem": {"asNumber": 84}}, {"topObjectType": "node", "topologyIndex": 0, "name": "84.0.0.103", "nodeIndex": 3, "AutonomousSystem": {"asNumber": 84}}, {"topObjectType": "node", "topologyIndex": 0, "name": "84.0.0.104", "nodeIndex": 4, "AutonomousSystem": {"asNumber": 84}}, {"topObjectType": "node", "topologyIndex": 0, "name": "84.0.0.105", "nodeIndex": 5, "AutonomousSystem": {"asNumber": 84}}, {"topObjectType": "node", "topologyIndex": 0, "name": "84.0.0.106", "nodeIndex": 8, "AutonomousSystem": {"asNumber": 84}}, {"topObjectType": "node", "topologyIndex": 0, "name": "84.0.0.107", "nodeIndex": 10, "AutonomousSystem": {"asNumber": 84}}, {"links": [{"topObjectType": "link", "topologyIndex": 0, "name": "L111.1.84.1_111.1.84.2", "linkIndex": 12, "endZ": true}, {"topObjectType": "interface", "ipv4Address": {"topObjectType": "ipv4", "address": "111.1.84.2"}, "protocols": ["RSVP"], "bandwidth": "10G"}, {"bandwidth": "10G", "EMetric": "10", "TEcolor": "0", "srIs": "J", "unreservedBw": [1000000000, 1000000000, 1000000000, 1000000000, 1000000000, 1000000000, 1000000000, 1000000000], "node": {"topObjectType": "node", "name": "84.0.0.105"}]}, {"topObjectType": "link", "topologyIndex": 0, "name": "L84.103.107.1_84.103.107.2", "linkIndex": 4, "endA": true}, {"topObjectType": "interface", "ipv4Address": {"topObjectType": "ipv4", "address": "84.103.107.1"}, "protocols": ["RSVP"], "bandwidth": "40G"}, {"bandwidth": "40G", "EMetric": "10", "TEcolor": "0", "srIs": "J", "unreservedBw": [1000000000, 1000000000, 1000000000, 1000000000, 1000000000, 1000000000, 1000000000, 1000000000], "node": {"topObjectType": "node", "name": "84.0.0.105"}]}]
```

Figure 2. Common Pathfinder RESTful requests/response patterns

1. Paragon Authentication

To access the Pathfinder API from some environments, your application must use IAM authentication.

1.1. IAM

The Paragon Identity and Access Management (IAM) service provides authentication and authorization for Pathfinder.

1.2. Paragon IAM Authentication API

The Paragon IAM Authentication API is available at the configured Paragon address under "iam/authenticate" (e.g. <https://example.paragon.net/iam/authenticate>). To access the Pathfinder API, a client can use the Authentication API to retrieve an access token for a given user. To retrieve the token the user must authenticate by passing in account credentials, which will return an `id_token`, a list of scopes and other metadata. Subsequent requests to the Pathfinder API should add the `id_token` to the request header under the `x-iam-token` key.

Method	URI	Description
POST	/iam/authenticate	Get an <code>id_token</code> and list of scopes for a user

1.2.1. Get ID Token

Method	URI	Description
POST	/iam/authenticate	Get an id_token and list of scopes for a user

Normal response codes: 200

1.2.1.1. Request

Example 1.1. Get ID Token: JSON request

```
{
  "user": {
    "domain": "spadmin",
    "name": "admin"
  },
  "password": "<password>",
  "methods": [
    "PASSWORD"
  ]
}
```

1.2.1.2. Response

Example 1.2. Get ID Token: JSON response

```
{
  "status": "AUTHENTICATED",
  "user_id": "09b56e78-ed53-41d5-ab08-09fec3bfcf9f",
  "id_token": "<...token...>",
  "token": {
    "user_id": "09b56e78-ed53-41d5-ab08-09fec3bfcf9f",
    "user_name": "admin",
    "iat": "1621628806",
    "exp": "1621632406",
    "unique_identifier": "72f07a88-656d-4840-9826-f6cffb7d9364-1621628806"
  },
  "scopes": [
    {
      "id": "43085f38-eff5-4252-aaf2-2f6f3adb375a",
      "name": "default",
      "display_name": "default",
      "type": "SP"
    }
  ],
  "access_token": "<...access token...>",
  "identity_provider_type": "Local"
}
```

2. Paragon PathFinder RESTCONF API

The RESTCONF API introduced in NorthStar 5.1 is a partially RFC8040-compliant endpoint that exposes a set of YANG Model using a REST-Like interface. Pathfinder supports only JSON content type in this release. The RESTCONF API enables querying of topology, and management of TE-LSP. The Topology is read-only on the RESTCONF API. Tunnel RPCs operation are used to create and delete TE-LSPs.

Method	URI	Description
RESTCONF Endpoint		
RESTCONF HTTP Operations		
RESTCONF Query Data		
Authentication		
ietf-network YANG Model		
GET	/data/ietf-network:networks	GET container ietf-network:networks.
GET	/operational/ietf-network:networks	GET container ietf-network:networks.
ietf-te YANG Model		
GET	/data/ietf-te:te	GET container ietf-te:te.
GET	/operational/ietf-te:te	GET container ietf-te:te.
POST	/data/ietf-te:te/globals/named-admin-groups	Create new entry in container named-admin-groups.
PUT	/data/ietf-te:te/globals/named-admin-groups	Replace the container named-admin-groups or child.
PATCH	/data/ietf-te:te/globals/named-admin-groups	Replace the container named-admin-groups or child.
DELETE	/data/ietf-te:te/globals/named-admin-groups	delete the container named-admin-groups or child.
ietf-yang-library YANG Model		
GET	/data/ietf-yang-library:modules-state	GET container ietf-yang-library:modules-state.
Supported YANG operations		
POST	/operations/ietf-te-operations:tunnel-atomic-create	RPC ietf-te-operations:tunnel-atomic-create
POST	/operations/ietf-te-operations:tunnel-atomic-delete	RPC ietf-te-operations:tunnel-atomic-delete
Network Slicing		
GET	/data/juniper-pathfinder-topology-group:topology-groups	Get slice id information via topology-group.
PUT	/traffic-engineering/api/topology/v2/{topology_id}/nodes/{nodeIndex}	Request Payload with sliceId.
PUT	/traffic-engineering/api/topology/v2/{topology_id}/nodes/{nodeIndex}	Request Payload without sliceId.
PATCH	/traffic-engineering/api/topology/v2/{topology_id}/nodes/bulk	Adding slice id to nodes using patch with op = add
PATCH	/traffic-engineering/api/topology/v2/{topology_id}/nodes/bulk	Deleting slice id from nodes using patch with op = remove
PUT	/traffic-engineering/api/topology/v2/{topology_id}/links/{linkIndex}	Request Payload with sliceId.

Method	URI	Description
PUT	/traffic-engineering/api/topology/v2/{topology_id}/links/{linkIndex}	Request Payload without sliced.
POST	/traffic-engineering/api/topology/v2/{topology_id}/te-lsps	Slice Include All, Slice Include Any, or Slice Include Exclude.

2.1. RESTCONF Endpoint

The RESTCONF endpoint is exposed over SSL on the same server as the other Pathfinder REST API. The default port is 443. The Base path for RESTCONF is `/restconf/`. Pathfinder provides the following subroutes:

- data
- operational
- operations

the data and operational in the context of the supported YANG module are equivalent.

2.1.1. URIs

An advantage of RESTful APIs is that the resources can be accessed using well-defined URLs. The models exposed using RESTCONF follow a hierarchical model described using YANG. The resource location is fully described by the YANG model and RFC8040 Section 3.5.3. Pathfinder supports this resource URL to access any part of the model served.

2.2. RESTCONF HTTP Operations

The Standard RESTCONF operations are defined in [RFC8040 Section 4](#). Pathfinder offers the following support

Table 2.1. Pathfinder RESTCONF HTTP operations support

Verb	Supported	Description
OPTIONS	Yes	Determine which methods are supported by the server.
GET	Yes	Retrieve data and metadata about a resource.
HEAD	Yes	The same as GET, but only the response headers are returned.
POST	On specific model	Create a resource or invoke an RPC operation.
PUT	On specific model	Create or replace a resource.
PATCH	On specific model	Create or update (but not delete) various resources. Both plain patch (RFC 8040) is supported.
DELETE	On specific model	Sent by a client to delete a target resource.

2.3. RESTCONF Query Data

RESTCONF defines query parameters in [RFC8040 Section 4.8](#).

Table 2.2. Pathfinder RESTCONF query parameters support

Query parameter	Supported	method	Possible Values	Description
content	No	GET, HEAD	all, config, nonconfig	Whether config and/or operational data is returned – default is all if content is not provided in the request
depth	No	GET, HEAD	Any positive integer value	Number of subtrees of configuration, relative to the resource, which are returned
fields	No	GET, HEAD	A semicolon-separated list of path string values	Request a subset of the content exposed by the resource or its children (e.g., id,other-elem/name to select only the ID and name of the other-element child)
filter	No	GET, HEAD	XPath 1.0 expression to be evaluated in the root context.	Boolean notification filter for event stream resources.
insert	No (not applicable)	POST, PUT	first, last, before, after	Insertion mode for "ordered-by user" data resources.
point	No (not applicable)	POST, PUT	Path to the resource being used as insertion point.	Insertion mode for "ordered-by user" data resources.
start-time	No	GET, HEAD	Date and time following ietf-yang-type date-and-time type.	Replay buffer start time for event stream resources.
stop-time	No	GET, HEAD	Date and time following ietf-yang-type date-and-time type	Replay buffer stop time for event stream resources.
with-defaults	No	GET, HEAD	report-all, trim, explicit, report-all-tagged	Allows the content of a resource to be included or excluded depending on whether the value is equal to the schema default

2.4. Authentication

An access token from the Paragon IAM authentication API should be used to access the RESTCONF APIs endpoints.

2.5. ietf-network YANG Model

Access the Pathfinder data model represented as ietf-network and extensions.

The following document version are supported as read-only models.

Table 2.3. Pathfinder ietf-network and extensions

Model	Document	Revision
ietf-network	RFC8345	2018-02-26
ietf-network-topology	RFC8345	2018-02-26
ietf-te-topology	RFC8795	2020-08-06
ietf-l3-te-topology	draft-ietf-teas-yang-l3-te-topo-09	2020-05-03
ietf-te-topology-packet	draft-ietf-teas-yang-l3-te-topo-09	2020-07-03
ietf-sr-mpls-topology	draft-ietf-teas-yang-l3-te-topo-09	2020-11-01
ietf-l3-unicast-topology	RFC8346	2018-02-26

Method	URI	Description
GET	/data/ietf-network:networks	GET container ietf-network:networks.
GET	/operational/ietf-network:networks	GET container ietf-network:networks.

2.5.1. Get container ietf-network:networks

Method	URI	Description
GET	/data/ietf-network:networks	GET container ietf-network:networks.

Normal response codes: 200

2.5.1.1. Request

This operation does not accept a request body.

2.5.1.2. Response

Example 2.1. Get container ietf-network:networks : application/yang-data+jsonresponse

Response example:

```
{
  "ietf-network:networks": {
    "network": [
      {
        "ietf-network-topology:link": [
          {
            "link-id": "NorthStar-Live:link:L0110.1200.0010_11.11.
0.105_0110.1200.0014_11.11.0.106:fwd",
            "source": {
              "source-node": "NorthStar-Live:node:0110.1200.
0010",
              "source-tp": "NorthStar-Live:node:0110.1200.
0010:11.11.0.105"
            },
            "destination": {
              "dest-node": "NorthStar-Live:node:0110.1200.0014",
              "dest-tp": "NorthStar-Live:node:0110.1200.0014:11.
11.0.106"
            },
            "ietf-te-topology:te": {
              "oper-status": "up",
              "te-link-attributes": {
                "access-type": "point-to-point",
                "admin-status": "up",
                "link-protection-type": "unprotected",
                "max-link-bandwidth": {
                  "te-bandwidth": {
                    "generic": "100000000"
                  }
                },
                "unreserved-bandwidth": [
                  {
                    "priority": 0,
                    "te-bandwidth": {
                      "generic": "100000000"
                    }
                  },
                  {
                    "priority": 1,
```

```

        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 2,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 3,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 4,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 5,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 6,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 7,
        "te-bandwidth": {
            "generic": "10000000"
        }
    }
],
"te-default-metric": 70,
"te-srlgs": [],
"te-igp-metric": 20,
"administrative-group": "01",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",
"one-way-max-delay-normality": "normal",
"one-way-delay-variation-normality": "normal",
"one-way-packet-loss-normality": "normal",
"two-way-min-delay": 0,
"two-way-min-delay-normality": "normal",

```

```

        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    },
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric2": 20,
        "ietf-sr-mpls-topology:sr-mpls": {
            "sid": 30004,
            "is-persistent": false,
            "is-part-of-set": false,
            "is-local": false,
            "value-type": "index",
            "address-family": "ipv4",
            "is-backup": false,
            "advertise-protection": "single",
            "is-on-lan": false
        }
    },
    },
    {
        "link-id": "NorthStar-Live:link:L0110.1200.0010_11.11.
0.105_0110.1200.0014_11.11.0.106:rvs",
        "source": {
            "source-node": "NorthStar-Live:node:0110.1200.
0014",
            "source-tp": "NorthStar-Live:node:0110.1200.
0014:11.11.0.106"
        },
        "destination": {
            "dest-node": "NorthStar-Live:node:0110.1200.0010",
            "dest-tp": "NorthStar-Live:node:0110.1200.0010:11.
11.0.105"
        },
        "ietf-te-topology:te": {
            "oper-status": "up",
            "te-link-attributes": {
                "access-type": "point-to-point",
                "admin-status": "up",
                "link-protection-type": "unprotected",
                "max-link-bandwidth": {
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                "unreserved-bandwidth": [
                    {
                        "priority": 0,
                        "te-bandwidth": {
                            "generic": "10000000"
                        }
                    }
                ]
            }
        },
    },
    {

```

```

        "priority": 1,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 2,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 3,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 4,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 5,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 6,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 7,
        "te-bandwidth": {
            "generic": "10000000"
        }
    }
],
"te-default-metric": 70,
"te-srlgs": [],
"te-igp-metric": 20,
"administrative-group": "01",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",
"one-way-max-delay-normality": "normal",
"one-way-delay-variation-normality": "normal",
"one-way-packet-loss-normality": "normal",
"two-way-min-delay": 0,

```

```

        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30004,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0010_11.11.
0.113_0110.1200.0014_11.11.0.114:fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0010",
        "source-tp": "NorthStar-Live:node:0110.1200.
0010:11.11.0.113"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0014",
        "dest-tp": "NorthStar-Live:node:0110.1200.0014:11.
11.0.114"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                }
            ]
        }
    }
},

```

```

        {
            "priority": 1,
            "te-bandwidth": {
                "generic": "10000000"
            }
        },
        {
            "priority": 2,
            "te-bandwidth": {
                "generic": "10000000"
            }
        },
        {
            "priority": 3,
            "te-bandwidth": {
                "generic": "10000000"
            }
        },
        {
            "priority": 4,
            "te-bandwidth": {
                "generic": "10000000"
            }
        },
        {
            "priority": 5,
            "te-bandwidth": {
                "generic": "10000000"
            }
        },
        {
            "priority": 6,
            "te-bandwidth": {
                "generic": "10000000"
            }
        },
        {
            "priority": 7,
            "te-bandwidth": {
                "generic": "10000000"
            }
        }
    ],
    "te-default-metric": 70,
    "te-srlgs": [],
    "te-igp-metric": 20,
    "administrative-group": "01",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",

```

```

        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    },
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric2": 20,
        "ietf-sr-mpls-topology:sr-mpls": {
            "sid": 30004,
            "is-persistent": false,
            "is-part-of-set": false,
            "is-local": false,
            "value-type": "index",
            "address-family": "ipv4",
            "is-backup": false,
            "advertise-protection": "single",
            "is-on-lan": false
        }
    },
    {
        {
            "link-id": "NorthStar-Live:link:L0110.1200.0010_11.11.
0.113_0110.1200.0014_11.11.0.114:rvs",
            "source": {
                "source-node": "NorthStar-Live:node:0110.1200.
0014",
                "source-tp": "NorthStar-Live:node:0110.1200.
0014:11.11.0.114"
            },
            "destination": {
                "dest-node": "NorthStar-Live:node:0110.1200.0010",
                "dest-tp": "NorthStar-Live:node:0110.1200.0010:11.
11.0.113"
            },
            "ietf-te-topology:te": {
                "oper-status": "up",
                "te-link-attributes": {
                    "access-type": "point-to-point",
                    "admin-status": "up",
                    "link-protection-type": "unprotected",
                    "max-link-bandwidth": {
                        "te-bandwidth": {
                            "generic": "10000000"
                        }
                    }
                },
                "unreserved-bandwidth": [
                    {
                        "priority": 0,
                        "te-bandwidth": {
                            "generic": "10000000"
                        }
                    }
                ]
            }
        }
    }
}

```

```

    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 70,
  "te-srlgs": [],
  "te-igp-metric": 20,
  "administrative-group": "01",
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
"normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
"normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
"normal",
  "one-way-min-delay-normality": "normal",
  "one-way-max-delay-normality": "normal",
  "one-way-delay-variation-normality": "normal",

```



```

        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30004,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0009_11.11.
0.65_0110.1200.0010_11.11.0.66: fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0010",
        "source-tp": "NorthStar-Live:node:0110.1200.
0010:11.11.0.66"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0009",
        "dest-tp": "NorthStar-Live:node:0110.1200.0009:11.
11.0.65"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            }
        },
        "unreserved-bandwidth": [
            {
                "priority": 0,
                "te-bandwidth": {
                    "generic": "10000000"
                }
            }
        ]
    }
}

```

```

    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 10,
  "te-igp-metric": 10,
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
"normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
"normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
"normal",
  "one-way-min-delay-normality": "normal",
  "one-way-max-delay-normality": "normal",
  "one-way-delay-variation-normality": "normal",
  "one-way-packet-loss-normality": "normal",

```

```

        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 20,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0009_11.11.
0.65_0110.1200.0010_11.11.0.66:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0009",
        "source-tp": "NorthStar-Live:node:0110.1200.
0009:11.11.0.65"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0010",
        "dest-tp": "NorthStar-Live:node:0110.1200.0010:11.
11.0.66"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                }
            ]
        }
    }
}

```

```

    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 10,
  "te-igp-metric": 10,
  "administrative-group": "00",
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
"normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
"normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
"normal",
  "one-way-min-delay-normality": "normal",
  "one-way-max-delay-normality": "normal",
  "one-way-delay-variation-normality": "normal",
  "one-way-packet-loss-normality": "normal",

```

```

        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 16,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0006_11.11.
0.73_0110.1200.0010_11.11.0.74: fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0010",
        "source-tp": "NorthStar-Live:node:0110.1200.
0010:11.11.0.74"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0006",
        "dest-tp": "NorthStar-Live:node:0110.1200.0006:11.
11.0.73"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            }
        },
        "unreserved-bandwidth": [
            {
                "priority": 0,
                "te-bandwidth": {
                    "generic": "10000000"
                }
            }
        ]
    }
}

```

```

    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 57,
  "te-srlgs": [],
  "te-igp-metric": 20,
  "administrative-group": "01",
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
"normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
"normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
"normal",
  "one-way-min-delay-normality": "normal",
  "one-way-max-delay-normality": "normal",
  "one-way-delay-variation-normality": "normal",

```

```

        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30003,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0006_11.11.
0.73_0110.1200.0010_11.11.0.74:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0006",
        "source-tp": "NorthStar-Live:node:0110.1200.
0006:11.11.0.73"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0010",
        "dest-tp": "NorthStar-Live:node:0110.1200.0010:11.
11.0.74"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            }
        },
        "unreserved-bandwidth": [
            {
                "priority": 0,
                "te-bandwidth": {
                    "generic": "10000000"
                }
            }
        ]
    }
}

```

```

    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 57,
  "te-srlgs": [],
  "te-igp-metric": 20,
  "administrative-group": "01",
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
    "normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
    "normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
    "normal",
  "one-way-min-delay-normality": "normal",
  "one-way-max-delay-normality": "normal",

```



```

        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    },
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric2": 20,
        "ietf-sr-mpls-topology:sr-mpls": {
            "sid": 30003,
            "is-persistent": false,
            "is-part-of-set": false,
            "is-local": false,
            "value-type": "index",
            "address-family": "ipv4",
            "is-backup": false,
            "advertise-protection": "single",
            "is-on-lan": false
        }
    },
    {
        "link-id": "NorthStar-Live:link:L0110.1200.0006_11.11.
0.77_0110.1200.0010_11.11.0.78: fwd",
        "source": {
            "source-node": "NorthStar-Live:node:0110.1200.
0010",
            "source-tp": "NorthStar-Live:node:0110.1200.
0010:11.11.0.78"
        },
        "destination": {
            "dest-node": "NorthStar-Live:node:0110.1200.0006",
            "dest-tp": "NorthStar-Live:node:0110.1200.0006:11.
11.0.77"
        },
        "ietf-te-topology:te": {
            "oper-status": "up",
            "te-link-attributes": {
                "access-type": "point-to-point",
                "admin-status": "up",
                "link-protection-type": "unprotected",
                "max-link-bandwidth": {
                    "te-bandwidth": {
                        "generic": "100000000"
                    }
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {

```

```

        "generic": "10000000"
      }
    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 57,
  "te-srlgs": [],
  "te-igp-metric": 20,
  "administrative-group": "01",
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
"normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
"normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
"normal",
  "one-way-min-delay-normality": "normal",

```

```

        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30003,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0006_11.11.
0.77_0110.1200.0010_11.11.0.78:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0006",
        "source-tp": "NorthStar-Live:node:0110.1200.
0006:11.11.0.77"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0010",
        "dest-tp": "NorthStar-Live:node:0110.1200.0010:11.
11.0.78"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,

```

```

        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 1,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 2,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 3,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 4,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 5,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 6,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 7,
        "te-bandwidth": {
            "generic": "10000000"
        }
    }
],
"te-default-metric": 57,
"te-srlgs": [],
"te-igp-metric": 20,
"administrative-group": "01",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",

```

```

        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30003,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0011_11.11.
0.81_0110.1200.0010_11.11.0.82: fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0010",
        "source-tp": "NorthStar-Live:node:0110.1200.
0010:11.11.0.82"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0011",
        "dest-tp": "NorthStar-Live:node:0110.1200.0011:11.
11.0.81"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            }
        },
        "unreserved-bandwidth": [
            {

```

```

        "priority": 0,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 1,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 2,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 3,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 4,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 5,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 6,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 7,
        "te-bandwidth": {
            "generic": "10000000"
        }
    }
],
"te-default-metric": 10,
"te-igp-metric": 10,
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",

```

```

        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 17,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0011_11.11.
0.81_0110.1200.0010_11.11.0.82:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0011",
        "source-tp": "NorthStar-Live:node:0110.1200.
0011:11.11.0.81"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0010",
        "dest-tp": "NorthStar-Live:node:0110.1200.0010:11.
11.0.82"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,

```

```

        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 1,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 2,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 3,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 4,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 5,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 6,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 7,
        "te-bandwidth": {
          "generic": "10000000"
        }
      }
    ],
    "te-default-metric": 10,
    "te-igp-metric": 10,
    "administrative-group": "00",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",

```



```

        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 16,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0011_11.11.
0.85_0110.1200.0012_11.11.0.86: fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0011",
        "source-tp": "NorthStar-Live:node:0110.1200.
0011:11.11.0.85"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0012",
        "dest-tp": "NorthStar-Live:node:0110.1200.0012:11.
11.0.86"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,

```

```
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 1,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 2,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 3,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 4,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 5,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 6,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 7,
        "te-bandwidth": {
          "generic": "10000000"
        }
      }
    ],
    "te-default-metric": 10,
    "te-igp-metric": 10,
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
```

```

        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    },
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric2": 10,
        "ietf-sr-mpls-topology:sr-mpls": {
            "sid": 17,
            "is-persistent": false,
            "is-part-of-set": false,
            "is-local": false,
            "value-type": "index",
            "address-family": "ipv4",
            "is-backup": false,
            "advertise-protection": "single",
            "is-on-lan": false
        }
    },
    },
    {
        "link-id": "NorthStar-Live:link:L0110.1200.0011_11.11.
0.85_0110.1200.0012_11.11.0.86:rvs",
        "source": {
            "source-node": "NorthStar-Live:node:0110.1200.
0012",
            "source-tp": "NorthStar-Live:node:0110.1200.
0012:11.11.0.86"
        },
        "destination": {
            "dest-node": "NorthStar-Live:node:0110.1200.0011",
            "dest-tp": "NorthStar-Live:node:0110.1200.0011:11.
11.0.85"
        },
        "ietf-te-topology:te": {
            "oper-status": "up",
            "te-link-attributes": {
                "access-type": "point-to-point",
                "admin-status": "up",
                "link-protection-type": "unprotected",
                "max-link-bandwidth": {
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {

```

```

        "generic": "10000000"
      }
    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 10,
  "te-igp-metric": 10,
  "administrative-group": "00",
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
"normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
"normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
"normal",
  "one-way-min-delay-normality": "normal",
  "one-way-max-delay-normality": "normal",

```

```

        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    },
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric2": 10,
        "ietf-sr-mpls-topology:sr-mpls": {
            "sid": 20,
            "is-persistent": false,
            "is-part-of-set": false,
            "is-local": false,
            "value-type": "index",
            "address-family": "ipv4",
            "is-backup": false,
            "advertise-protection": "single",
            "is-on-lan": false
        }
    },
    },
    {
        "link-id": "NorthStar-Live:link:L0110.1200.0012_11.11.
0.129_0110.1200.0016_11.11.0.130: fwd",
        "source": {
            "source-node": "NorthStar-Live:node:0110.1200.
0012",
            "source-tp": "NorthStar-Live:node:0110.1200.
0012:11.11.0.129"
        },
        "destination": {
            "dest-node": "NorthStar-Live:node:0110.1200.0016",
            "dest-tp": "NorthStar-Live:node:0110.1200.0016:11.
11.0.130"
        },
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            }
        },
        "unreserved-bandwidth": [
            {
                "priority": 0,
                "te-bandwidth": {

```

```

        "generic": "10000000"
      }
    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 70,
  "te-srlgs": [],
  "te-igp-metric": 20,
  "administrative-group": "02",
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
"normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
"normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
"normal",
  "one-way-min-delay-normality": "normal",

```

```

        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30004,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0012_11.11.
0.129_0110.1200.0016_11.11.0.130:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0016",
        "source-tp": "NorthStar-Live:node:0110.1200.
0016:11.11.0.130"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0012",
        "dest-tp": "NorthStar-Live:node:0110.1200.0012:11.
11.0.129"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,

```

```

        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 1,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 2,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 3,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 4,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 5,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 6,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 7,
        "te-bandwidth": {
            "generic": "10000000"
        }
    }
],
"te-default-metric": 70,
"te-srlgs": [],
"te-igp-metric": 20,
"administrative-group": "02",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",

```



```

        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30004,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0012_11.11.
0.137_0110.1200.0016_11.11.0.138:fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0012",
        "source-tp": "NorthStar-Live:node:0110.1200.
0012:11.11.0.137"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0016",
        "dest-tp": "NorthStar-Live:node:0110.1200.0016:11.
11.0.138"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            "unreserved-bandwidth": [
                {

```

```

        "priority": 0,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 1,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 2,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 3,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 4,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 5,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 6,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 7,
        "te-bandwidth": {
            "generic": "10000000"
        }
    }
],
"te-default-metric": 70,
"te-srlgs": [],
"te-igp-metric": 20,
"administrative-group": "02",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",

```

```

        "normal",
        "one-way-utilized-bandwidth-normality":
            "one-way-min-delay-normality": "normal",
            "one-way-max-delay-normality": "normal",
            "one-way-delay-variation-normality": "normal",
            "one-way-packet-loss-normality": "normal",
            "two-way-min-delay": 0,
            "two-way-min-delay-normality": "normal",
            "two-way-max-delay": 0,
            "two-way-max-delay-normality": "normal",
            "two-way-delay-variation": 0,
            "two-way-delay-variation-normality": "normal",
            "two-way-packet-loss": 0,
            "two-way-packet-loss-normality": "normal",
            "one-way-delay-offset": 0,
            "measure-interval": 30,
            "advertisement-interval": 0,
            "suppression-interval": 120
        }
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric2": 20,
        "ietf-sr-mpls-topology:sr-mpls": {
            "sid": 30004,
            "is-persistent": false,
            "is-part-of-set": false,
            "is-local": false,
            "value-type": "index",
            "address-family": "ipv4",
            "is-backup": false,
            "advertise-protection": "single",
            "is-on-lan": false
        }
    },
    {
        "link-id": "NorthStar-Live:link:L0110.1200.0012_11.11.
0.137_0110.1200.0016_11.11.0.138:rvs",
        "source": {
            "source-node": "NorthStar-Live:node:0110.1200.
0016",
            "source-tp": "NorthStar-Live:node:0110.1200.
0016:11.11.0.138"
        },
        "destination": {
            "dest-node": "NorthStar-Live:node:0110.1200.0012",
            "dest-tp": "NorthStar-Live:node:0110.1200.0012:11.
11.0.137"
        },
        "ietf-te-topology:te": {
            "oper-status": "up",
            "te-link-attributes": {
                "access-type": "point-to-point",
                "admin-status": "up",
                "link-protection-type": "unprotected",
                "max-link-bandwidth": {
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                }
            }
        }
    },

```

```
        "unreserved-bandwidth": [
          {
            "priority": 0,
            "te-bandwidth": {
              "generic": "10000000"
            }
          },
          {
            "priority": 1,
            "te-bandwidth": {
              "generic": "10000000"
            }
          },
          {
            "priority": 2,
            "te-bandwidth": {
              "generic": "10000000"
            }
          },
          {
            "priority": 3,
            "te-bandwidth": {
              "generic": "10000000"
            }
          },
          {
            "priority": 4,
            "te-bandwidth": {
              "generic": "10000000"
            }
          },
          {
            "priority": 5,
            "te-bandwidth": {
              "generic": "10000000"
            }
          },
          {
            "priority": 6,
            "te-bandwidth": {
              "generic": "10000000"
            }
          },
          {
            "priority": 7,
            "te-bandwidth": {
              "generic": "10000000"
            }
          }
        ],
        "te-default-metric": 70,
        "te-srlgs": [],
        "te-igp-metric": 20,
        "administrative-group": "02",
        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
```

```

        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    },
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric2": 20,
        "ietf-sr-mpls-topology:sr-mpls": {
            "sid": 30004,
            "is-persistent": false,
            "is-part-of-set": false,
            "is-local": false,
            "value-type": "index",
            "address-family": "ipv4",
            "is-backup": false,
            "advertise-protection": "single",
            "is-on-lan": false
        }
    },
    {
        "link-id": "NorthStar-Live:link:L0110.1200.0009_11.11.
0.69_0110.1200.0012_11.11.0.70: fwd",
        "source": {
            "source-node": "NorthStar-Live:node:0110.1200.
0012",
            "source-tp": "NorthStar-Live:node:0110.1200.
0012:11.11.0.70"
        },
        "destination": {
            "dest-node": "NorthStar-Live:node:0110.1200.0009",
            "dest-tp": "NorthStar-Live:node:0110.1200.0009:11.
11.0.69"
        },
        "ietf-te-topology:te": {
            "oper-status": "up",
            "te-link-attributes": {
                "access-type": "point-to-point",
                "admin-status": "up",
                "link-protection-type": "unprotected",
                "max-link-bandwidth": {
                    "te-bandwidth": {
                        "generic": "100000000"
                    }
                }
            }
        }
    }

```

```

    },
    "unreserved-bandwidth": [
      {
        "priority": 0,
        "te-bandwidth": {
          "generic": "100000000"
        }
      },
      {
        "priority": 1,
        "te-bandwidth": {
          "generic": "100000000"
        }
      },
      {
        "priority": 2,
        "te-bandwidth": {
          "generic": "100000000"
        }
      },
      {
        "priority": 3,
        "te-bandwidth": {
          "generic": "100000000"
        }
      },
      {
        "priority": 4,
        "te-bandwidth": {
          "generic": "100000000"
        }
      },
      {
        "priority": 5,
        "te-bandwidth": {
          "generic": "100000000"
        }
      },
      {
        "priority": 6,
        "te-bandwidth": {
          "generic": "100000000"
        }
      },
      {
        "priority": 7,
        "te-bandwidth": {
          "generic": "100000000"
        }
      }
    ],
    "te-default-metric": 10,
    "te-igp-metric": 10,
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",

```

```

        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 21,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0009_11.11.
0.69_0110.1200.0012_11.11.0.70:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0009",
        "source-tp": "NorthStar-Live:node:0110.1200.
0009:11.11.0.69"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0012",
        "dest-tp": "NorthStar-Live:node:0110.1200.0012:11.
11.0.70"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            }
        }
    },

```

```

        "unreserved-bandwidth": [
            {
                "priority": 0,
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            {
                "priority": 1,
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            {
                "priority": 2,
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            {
                "priority": 3,
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            {
                "priority": 4,
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            {
                "priority": 5,
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            {
                "priority": 6,
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            {
                "priority": 7,
                "te-bandwidth": {
                    "generic": "10000000"
                }
            }
        ],
        "te-default-metric": 10,
        "te-igp-metric": 10,
        "administrative-group": "00",
        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",

```



```

        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 17,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0008_11.11.
0.89_0110.1200.0012_11.11.0.90: fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0012",
        "source-tp": "NorthStar-Live:node:0110.1200.
0012:11.11.0.90"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0008",
        "dest-tp": "NorthStar-Live:node:0110.1200.0008:11.
11.0.89"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            }
        }
    },

```

```
    "unreserved-bandwidth": [  
      {  
        "priority": 0,  
        "te-bandwidth": {  
          "generic": "10000000"  
        }  
      },  
      {  
        "priority": 1,  
        "te-bandwidth": {  
          "generic": "10000000"  
        }  
      },  
      {  
        "priority": 2,  
        "te-bandwidth": {  
          "generic": "10000000"  
        }  
      },  
      {  
        "priority": 3,  
        "te-bandwidth": {  
          "generic": "10000000"  
        }  
      },  
      {  
        "priority": 4,  
        "te-bandwidth": {  
          "generic": "10000000"  
        }  
      },  
      {  
        "priority": 5,  
        "te-bandwidth": {  
          "generic": "10000000"  
        }  
      },  
      {  
        "priority": 6,  
        "te-bandwidth": {  
          "generic": "10000000"  
        }  
      },  
      {  
        "priority": 7,  
        "te-bandwidth": {  
          "generic": "10000000"  
        }  
      }  
    ],  
    "te-default-metric": 57,  
    "te-srlgs": [],  
    "te-igp-metric": 20,  
    "administrative-group": "02",  
    "one-way-residual-bandwidth": "0x0p0",  
    "one-way-residual-bandwidth-normality":  
"normal",  
    "one-way-available-bandwidth": "0x0p0",  
    "one-way-available-bandwidth-normality":  
"normal",
```

```

"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
        "normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
  },
  },
  "ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
      "sid": 30003,
      "is-persistent": false,
      "is-part-of-set": false,
      "is-local": false,
      "value-type": "index",
      "address-family": "ipv4",
      "is-backup": false,
      "advertise-protection": "single",
      "is-on-lan": false
    }
  },
  {
    "link-id": "NorthStar-Live:link:L0110.1200.0008_11.11.
0.89_0110.1200.0012_11.11.0.90:rvs",
    "source": {
      "source-node": "NorthStar-Live:node:0110.1200.
0008",
      "source-tp": "NorthStar-Live:node:0110.1200.
0008:11.11.0.89"
    },
    "destination": {
      "dest-node": "NorthStar-Live:node:0110.1200.0012",
      "dest-tp": "NorthStar-Live:node:0110.1200.0012:11.
11.0.90"
    },
    "ietf-te-topology:te": {
      "oper-status": "up",
      "te-link-attributes": {
        "access-type": "point-to-point",
        "admin-status": "up",
        "link-protection-type": "unprotected",
        "max-link-bandwidth": {
          "te-bandwidth": {
            "generic": "100000000"
          }
        }
      }
    }
  }
}

```

```
    },
    "unreserved-bandwidth": [
      {
        "priority": 0,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 1,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 2,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 3,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 4,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 5,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 6,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 7,
        "te-bandwidth": {
          "generic": "10000000"
        }
      }
    ],
    "te-default-metric": 57,
    "te-srlgs": [],
    "te-igp-metric": 20,
    "administrative-group": "02",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
```

```

        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30003,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0008_11.11.
0.93_0110.1200.0012_11.11.0.94: fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0012",
        "source-tp": "NorthStar-Live:node:0110.1200.
0012:11.11.0.94"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0008",
        "dest-tp": "NorthStar-Live:node:0110.1200.0008:11.
11.0.93"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {

```

```
        "generic": "10000000"
      }
    },
    "unreserved-bandwidth": [
      {
        "priority": 0,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 1,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 2,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 3,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 4,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 5,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 6,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 7,
        "te-bandwidth": {
          "generic": "10000000"
        }
      }
    ],
    "te-default-metric": 57,
    "te-srlgs": [],
    "te-igp-metric": 20,
    "administrative-group": "02",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
```

```

        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30003,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0008_11.11.
0.93_0110.1200.0012_11.11.0.94:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0008",
        "source-tp": "NorthStar-Live:node:0110.1200.
0008:11.11.0.93"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0012",
        "dest-tp": "NorthStar-Live:node:0110.1200.0012:11.
11.0.94"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {

```

```
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      "unreserved-bandwidth": [
        {
          "priority": 0,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 1,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 2,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 3,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 4,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 5,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 6,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 7,
          "te-bandwidth": {
            "generic": "10000000"
          }
        }
      ],
      "te-default-metric": 57,
      "te-srlgs": [],
      "te-igp-metric": 20,
      "administrative-group": "02",
      "one-way-residual-bandwidth": "0x0p0",
```



```

        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30003,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0013_11.11.
0.101_0110.1200.0016_11.11.0.102:fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0013",
        "source-tp": "NorthStar-Live:node:0110.1200.
0013:11.11.0.101"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0016",
        "dest-tp": "NorthStar-Live:node:0110.1200.0016:11.
11.0.102"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 10,
"te-igp-metric": 10,
"one-way-residual-bandwidth": "0x0p0",
```

```

        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 16,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0013_11.11.
0.101_0110.1200.0016_11.11.0.102:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0016",
        "source-tp": "NorthStar-Live:node:0110.1200.
0016:11.11.0.102"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0013",
        "dest-tp": "NorthStar-Live:node:0110.1200.0013:11.
11.0.101"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 10,
"te-igp-metric": 10,
"administrative-group": "00",
"one-way-residual-bandwidth": "0x0p0",
```

```

        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 17,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0013_11.11.
0.97_0110.1200.0014_11.11.0.98: fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0013",
        "source-tp": "NorthStar-Live:node:0110.1200.
0013:11.11.0.97"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0014",
        "dest-tp": "NorthStar-Live:node:0110.1200.0014:11.
11.0.98"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 10,
"te-igp-metric": 10,
"one-way-residual-bandwidth": "0x0p0",
```

```

        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 17,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0013_11.11.
0.97_0110.1200.0014_11.11.0.98:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0014",
        "source-tp": "NorthStar-Live:node:0110.1200.
0014:11.11.0.98"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0013",
        "dest-tp": "NorthStar-Live:node:0110.1200.0013:11.
11.0.97"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 10,
"te-igp-metric": 10,
"administrative-group": "00",
"one-way-residual-bandwidth": "0x0p0",
```



```

        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 21,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0001_11.11.
0.1_0110.1200.0002_11.11.0.2: fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0001",
        "source-tp": "NorthStar-Live:node:0110.1200.
0001:11.11.0.1"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0002",
        "dest-tp": "NorthStar-Live:node:0110.1200.0002:11.
11.0.2"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 10,
"te-igp-metric": 10,
"one-way-residual-bandwidth": "0x0p0",
```

```

        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 16,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0001_11.11.
0.1_0110.1200.0002_11.11.0.2:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0002",
        "source-tp": "NorthStar-Live:node:0110.1200.
0002:11.11.0.2"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0001",
        "dest-tp": "NorthStar-Live:node:0110.1200.0001:11.
11.0.1"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 10,
"te-igp-metric": 10,
"administrative-group": "00",
"one-way-residual-bandwidth": "0x0p0",
```

```

        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 16,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0001_11.11.
0.5_0110.1200.0004_11.11.0.6: fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0001",
        "source-tp": "NorthStar-Live:node:0110.1200.
0001:11.11.0.5"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0004",
        "dest-tp": "NorthStar-Live:node:0110.1200.0004:11.
11.0.6"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 10,
"te-igp-metric": 10,
"one-way-residual-bandwidth": "0x0p0",
```

```

        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 17,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0001_11.11.
0.5_0110.1200.0004_11.11.0.6:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0004",
        "source-tp": "NorthStar-Live:node:0110.1200.
0004:11.11.0.6"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0001",
        "dest-tp": "NorthStar-Live:node:0110.1200.0001:11.
11.0.5"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 10,
"te-igp-metric": 10,
"administrative-group": "00",
"one-way-residual-bandwidth": "0x0p0",
```



```

        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 21,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0006_11.11.
0.109_0110.1200.0014_11.11.0.110:fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0014",
        "source-tp": "NorthStar-Live:node:0110.1200.
0014:11.11.0.110"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0006",
        "dest-tp": "NorthStar-Live:node:0110.1200.0006:11.
11.0.109"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 30,
"te-igp-metric": 20,
"administrative-group": "01",
"one-way-residual-bandwidth": "0x0p0",
```

```

        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30005,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0006_11.11.
0.109_0110.1200.0014_11.11.0.110:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0006",
        "source-tp": "NorthStar-Live:node:0110.1200.
0006:11.11.0.109"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0014",
        "dest-tp": "NorthStar-Live:node:0110.1200.0014:11.
11.0.110"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 30,
"te-igp-metric": 20,
"administrative-group": "01",
"one-way-residual-bandwidth": "0x0p0",
```

```

        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30005,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0006_11.11.
0.117_0110.1200.0014_11.11.0.118:fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0014",
        "source-tp": "NorthStar-Live:node:0110.1200.
0014:11.11.0.118"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0006",
        "dest-tp": "NorthStar-Live:node:0110.1200.0006:11.
11.0.117"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 30,
"te-igp-metric": 20,
"administrative-group": "01",
"one-way-residual-bandwidth": "0x0p0",
```

```

        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30005,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0006_11.11.
0.117_0110.1200.0014_11.11.0.118:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0006",
        "source-tp": "NorthStar-Live:node:0110.1200.
0006:11.11.0.117"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0014",
        "dest-tp": "NorthStar-Live:node:0110.1200.0014:11.
11.0.118"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 30,
"te-igp-metric": 20,
"administrative-group": "01",
"one-way-residual-bandwidth": "0x0p0",
```



```

        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30005,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0015_11.11.
0.121_0110.1200.0014_11.11.0.122:fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0014",
        "source-tp": "NorthStar-Live:node:0110.1200.
0014:11.11.0.122"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0015",
        "dest-tp": "NorthStar-Live:node:0110.1200.0015:11.
11.0.121"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 10,
"te-igp-metric": 10,
"one-way-residual-bandwidth": "0x0p0",
```

```

        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 18,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0015_11.11.
0.121_0110.1200.0014_11.11.0.122:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0015",
        "source-tp": "NorthStar-Live:node:0110.1200.
0015:11.11.0.121"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0014",
        "dest-tp": "NorthStar-Live:node:0110.1200.0014:11.
11.0.122"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 10,
"te-igp-metric": 10,
"administrative-group": "00",
"one-way-residual-bandwidth": "0x0p0",
```

```

        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 17,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0014_11.11.
0.13_0110.1200.0002_11.11.0.14: fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0014",
        "source-tp": "NorthStar-Live:node:0110.1200.
0014:11.11.0.13"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0002",
        "dest-tp": "NorthStar-Live:node:0110.1200.0002:11.
11.0.14"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 30,
"te-srlgs": [],
"te-igp-metric": 20,
"administrative-group": "01",
```

```

    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
  }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
  "metric2": 20,
  "ietf-sr-mpls-topology:sr-mpls": {
    "sid": 30001,
    "is-persistent": false,
    "is-part-of-set": false,
    "is-local": false,
    "value-type": "index",
    "address-family": "ipv4",
    "is-backup": false,
    "advertise-protection": "single",
    "is-on-lan": false
  }
},
{
  "link-id": "NorthStar-Live:link:L0110.1200.0014_11.11.
0.13_0110.1200.0002_11.11.0.14:rvs",
  "source": {
    "source-node": "NorthStar-Live:node:0110.1200.
0002",
    "source-tp": "NorthStar-Live:node:0110.1200.
0002:11.11.0.14"
  },
  "destination": {
    "dest-node": "NorthStar-Live:node:0110.1200.0014",
    "dest-tp": "NorthStar-Live:node:0110.1200.0014:11.
11.0.13"
  },
  "ietf-te-topology:te": {
    "oper-status": "up",
    "te-link-attributes": {
      "access-type": "point-to-point",

```

```
"admin-status": "up",
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000"
    }
  }
],
"te-default-metric": 30,
"te-srlgs": [],
"te-igp-metric": 20,
```



```

        "administrative-group": "01",
        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30001,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0002_11.11.
0.10_0110.1200.0014_11.11.0.9: fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0014",
        "source-tp": "NorthStar-Live:node:0110.1200.
0014:11.11.0.9"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0002",
        "dest-tp": "NorthStar-Live:node:0110.1200.0002:11.
11.0.10"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {

```

```
"access-type": "point-to-point",
"admin-status": "up",
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "100000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "100000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "100000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "100000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "100000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "100000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "100000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "100000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "100000000"
    }
  }
],
"te-default-metric": 30,
"te-srlgs": [],
```

```

        "te-igp-metric": 20,
        "administrative-group": "01",
        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30001,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0002_11.11.
0.10_0110.1200.0014_11.11.0.9:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0002",
        "source-tp": "NorthStar-Live:node:0110.1200.
0002:11.11.0.10"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0014",
        "dest-tp": "NorthStar-Live:node:0110.1200.0014:11.
11.0.9"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",

```

```
"te-link-attributes": {
  "access-type": "point-to-point",
  "admin-status": "up",
  "link-protection-type": "unprotected",
  "max-link-bandwidth": {
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  "unreserved-bandwidth": [
    {
      "priority": 0,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 30,
```

```

        "te-srlgs": [],
        "te-igp-metric": 20,
        "administrative-group": "01",
        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    },
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric2": 20,
        "ietf-sr-mpls-topology:sr-mpls": {
            "sid": 30001,
            "is-persistent": false,
            "is-part-of-set": false,
            "is-local": false,
            "value-type": "index",
            "address-family": "ipv4",
            "is-backup": false,
            "advertise-protection": "single",
            "is-on-lan": false
        }
    },
    {
        "link-id": "NorthStar-Live:link:L0110.1200.0015_11.11.
0.125_0110.1200.0016_11.11.0.126:fwd",
        "source": {
            "source-node": "NorthStar-Live:node:0110.1200.
0015",
            "source-tp": "NorthStar-Live:node:0110.1200.
0015:11.11.0.125"
        },
        "destination": {
            "dest-node": "NorthStar-Live:node:0110.1200.0016",
            "dest-tp": "NorthStar-Live:node:0110.1200.0016:11.
11.0.126"
        },
        "ietf-te-topology:te": {

```

```
"oper-status": "up",
"te-link-attributes": {
  "access-type": "point-to-point",
  "admin-status": "up",
  "link-protection-type": "unprotected",
  "max-link-bandwidth": {
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  "unreserved-bandwidth": [
    {
      "priority": 0,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ]
},
```

```

        "te-default-metric": 10,
        "te-igp-metric": 10,
        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 16,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0015_11.11.
0.125_0110.1200.0016_11.11.0.126:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0016",
        "source-tp": "NorthStar-Live:node:0110.1200.
0016:11.11.0.126"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0015",
        "dest-tp": "NorthStar-Live:node:0110.1200.0015:11.
11.0.125"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",

```

```
"te-link-attributes": {
  "access-type": "point-to-point",
  "admin-status": "up",
  "link-protection-type": "unprotected",
  "max-link-bandwidth": {
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  "unreserved-bandwidth": [
    {
      "priority": 0,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 10,
```



```

        "te-igp-metric": 10,
        "administrative-group": "00",
        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 16,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0008_11.11.
0.133_0110.1200.0016_11.11.0.134:fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0016",
        "source-tp": "NorthStar-Live:node:0110.1200.
0016:11.11.0.134"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0008",
        "dest-tp": "NorthStar-Live:node:0110.1200.0008:11.
11.0.133"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",

```

```
"te-link-attributes": {
  "access-type": "point-to-point",
  "admin-status": "up",
  "link-protection-type": "unprotected",
  "max-link-bandwidth": {
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  "unreserved-bandwidth": [
    {
      "priority": 0,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 25,
```

```

        "te-igp-metric": 20,
        "administrative-group": "02",
        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30005,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0008_11.11.
0.133_0110.1200.0016_11.11.0.134:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0008",
        "source-tp": "NorthStar-Live:node:0110.1200.
0008:11.11.0.133"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0016",
        "dest-tp": "NorthStar-Live:node:0110.1200.0016:11.
11.0.134"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",

```

```
"te-link-attributes": {
  "access-type": "point-to-point",
  "admin-status": "up",
  "link-protection-type": "unprotected",
  "max-link-bandwidth": {
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  "unreserved-bandwidth": [
    {
      "priority": 0,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 25,
```

```

        "te-igp-metric": 20,
        "administrative-group": "02",
        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30005,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0008_11.11.
0.141_0110.1200.0016_11.11.0.142:fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0016",
        "source-tp": "NorthStar-Live:node:0110.1200.
0016:11.11.0.142"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0008",
        "dest-tp": "NorthStar-Live:node:0110.1200.0008:11.
11.0.141"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",

```

```
"te-link-attributes": {
  "access-type": "point-to-point",
  "admin-status": "up",
  "link-protection-type": "unprotected",
  "max-link-bandwidth": {
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  "unreserved-bandwidth": [
    {
      "priority": 0,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 25,
```

```

        "te-igp-metric": 20,
        "administrative-group": "02",
        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30005,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0008_11.11.
0.141_0110.1200.0016_11.11.0.142:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0008",
        "source-tp": "NorthStar-Live:node:0110.1200.
0008:11.11.0.141"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0016",
        "dest-tp": "NorthStar-Live:node:0110.1200.0016:11.
11.0.142"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",

```

```
"te-link-attributes": {
  "access-type": "point-to-point",
  "admin-status": "up",
  "link-protection-type": "unprotected",
  "max-link-bandwidth": {
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  "unreserved-bandwidth": [
    {
      "priority": 0,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 25,
```



```

        "te-igp-metric": 20,
        "administrative-group": "02",
        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30005,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0016_11.11.
0.25_0110.1200.0004_11.11.0.26: fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0016",
        "source-tp": "NorthStar-Live:node:0110.1200.
0016:11.11.0.25"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0004",
        "dest-tp": "NorthStar-Live:node:0110.1200.0004:11.
11.0.26"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",

```

```
"te-link-attributes": {
  "access-type": "point-to-point",
  "admin-status": "up",
  "link-protection-type": "unprotected",
  "max-link-bandwidth": {
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  "unreserved-bandwidth": [
    {
      "priority": 0,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 30,
```

```

        "te-srlgs": [],
        "te-igp-metric": 20,
        "administrative-group": "02",
        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    },
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric2": 20,
        "ietf-sr-mpls-topology:sr-mpls": {
            "sid": 30001,
            "is-persistent": false,
            "is-part-of-set": false,
            "is-local": false,
            "value-type": "index",
            "address-family": "ipv4",
            "is-backup": false,
            "advertise-protection": "single",
            "is-on-lan": false
        }
    },
    {
        "link-id": "NorthStar-Live:link:L0110.1200.0016_11.11.
0.25_0110.1200.0004_11.11.0.26:rvs",
        "source": {
            "source-node": "NorthStar-Live:node:0110.1200.
0004",
            "source-tp": "NorthStar-Live:node:0110.1200.
0004:11.11.0.26"
        },
        "destination": {
            "dest-node": "NorthStar-Live:node:0110.1200.0016",
            "dest-tp": "NorthStar-Live:node:0110.1200.0016:11.
11.0.25"
        },
        "ietf-te-topology:te": {

```

```
"oper-status": "up",
"te-link-attributes": {
  "access-type": "point-to-point",
  "admin-status": "up",
  "link-protection-type": "unprotected",
  "max-link-bandwidth": {
    "te-bandwidth": {
      "generic": "10000000"
    }
  },
  "unreserved-bandwidth": [
    {
      "priority": 0,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ]
},
```

```

        "te-default-metric": 30,
        "te-srlgs": [],
        "te-igp-metric": 20,
        "administrative-group": "02",
        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    },
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric2": 20,
        "ietf-sr-mpls-topology:sr-mpls": {
            "sid": 30001,
            "is-persistent": false,
            "is-part-of-set": false,
            "is-local": false,
            "value-type": "index",
            "address-family": "ipv4",
            "is-backup": false,
            "advertise-protection": "single",
            "is-on-lan": false
        }
    },
    {
        "link-id": "NorthStar-Live:link:L0110.1200.0016_11.11.
0.29_0110.1200.0004_11.11.0.30: fwd",
        "source": {
            "source-node": "NorthStar-Live:node:0110.1200.
0016",
            "source-tp": "NorthStar-Live:node:0110.1200.
0016:11.11.0.29"
        },
        "destination": {
            "dest-node": "NorthStar-Live:node:0110.1200.0004",
            "dest-tp": "NorthStar-Live:node:0110.1200.0004:11.
11.0.30"
        },
    },

```

```
"ietf-te-topology:te": {
  "oper-status": "up",
  "te-link-attributes": {
    "access-type": "point-to-point",
    "admin-status": "up",
    "link-protection-type": "unprotected",
    "max-link-bandwidth": {
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    "unreserved-bandwidth": [
      {
        "priority": 0,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 1,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 2,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 3,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 4,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 5,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 6,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 7,
        "te-bandwidth": {
          "generic": "10000000"
        }
      }
    ]
  }
}
```

```

],
"te-default-metric": 30,
"te-srlgs": [],
"te-igp-metric": 20,
"administrative-group": "02",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",
"one-way-max-delay-normality": "normal",
"one-way-delay-variation-normality": "normal",
"one-way-packet-loss-normality": "normal",
"two-way-min-delay": 0,
"two-way-min-delay-normality": "normal",
"two-way-max-delay": 0,
"two-way-max-delay-normality": "normal",
"two-way-delay-variation": 0,
"two-way-delay-variation-normality": "normal",
"two-way-packet-loss": 0,
"two-way-packet-loss-normality": "normal",
"one-way-delay-offset": 0,
"measure-interval": 30,
"advertisement-interval": 0,
"suppression-interval": 120
    }
  },
  "ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
      "sid": 30001,
      "is-persistent": false,
      "is-part-of-set": false,
      "is-local": false,
      "value-type": "index",
      "address-family": "ipv4",
      "is-backup": false,
      "advertise-protection": "single",
      "is-on-lan": false
    }
  },
  {
    "link-id": "NorthStar-Live:link:L0110.1200.0016_11.11.
0.29_0110.1200.0004_11.11.0.30:rvs",
    "source": {
      "source-node": "NorthStar-Live:node:0110.1200.
0004",
      "source-tp": "NorthStar-Live:node:0110.1200.
0004:11.11.0.30"
    },
    "destination": {
      "dest-node": "NorthStar-Live:node:0110.1200.0016",
      "dest-tp": "NorthStar-Live:node:0110.1200.0016:11.
11.0.29"
    }
  }
}

```

```
},
"ietf-te-topology:te": {
  "oper-status": "up",
  "te-link-attributes": {
    "access-type": "point-to-point",
    "admin-status": "up",
    "link-protection-type": "unprotected",
    "max-link-bandwidth": {
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  },
  "unreserved-bandwidth": [
    {
      "priority": 0,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ]
}
```



```

    }
    ],
    "te-default-metric": 30,
    "te-srlgs": [],
    "te-igp-metric": 20,
    "administrative-group": "02",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
  }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
  "metric2": 20,
  "ietf-sr-mpls-topology:sr-mpls": {
    "sid": 30001,
    "is-persistent": false,
    "is-part-of-set": false,
    "is-local": false,
    "value-type": "index",
    "address-family": "ipv4",
    "is-backup": false,
    "advertise-protection": "single",
    "is-on-lan": false
  }
}
},
{
  "link-id": "NorthStar-Live:link:L0110.1200.0003_11.11.
0.17_0110.1200.0002_11.11.0.18:fwd",
  "source": {
    "source-node": "NorthStar-Live:node:0110.1200.
0002",
    "source-tp": "NorthStar-Live:node:0110.1200.
0002:11.11.0.18"
  },
  "destination": {
    "dest-node": "NorthStar-Live:node:0110.1200.0003",

```

```
11.0.17"      "dest-tp": "NorthStar-Live:node:0110.1200.0003:11.
               },
               "ietf-te-topology:te": {
                 "oper-status": "up",
                 "te-link-attributes": {
                   "access-type": "point-to-point",
                   "admin-status": "up",
                   "link-protection-type": "unprotected",
                   "max-link-bandwidth": {
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                 },
                 "unreserved-bandwidth": [
                   {
                     "priority": 0,
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                   {
                     "priority": 1,
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                   {
                     "priority": 2,
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                   {
                     "priority": 3,
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                   {
                     "priority": 4,
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                   {
                     "priority": 5,
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                   {
                     "priority": 6,
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                   {
                     "priority": 7,
                     "te-bandwidth": {
```

```

        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 10,
  "te-igp-metric": 10,
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
"normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
"normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
"normal",
  "one-way-min-delay-normality": "normal",
  "one-way-max-delay-normality": "normal",
  "one-way-delay-variation-normality": "normal",
  "one-way-packet-loss-normality": "normal",
  "two-way-min-delay": 0,
  "two-way-min-delay-normality": "normal",
  "two-way-max-delay": 0,
  "two-way-max-delay-normality": "normal",
  "two-way-delay-variation": 0,
  "two-way-delay-variation-normality": "normal",
  "two-way-packet-loss": 0,
  "two-way-packet-loss-normality": "normal",
  "one-way-delay-offset": 0,
  "measure-interval": 30,
  "advertisement-interval": 0,
  "suppression-interval": 120
    }
  },
  "ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
      "sid": 21,
      "is-persistent": false,
      "is-part-of-set": false,
      "is-local": false,
      "value-type": "index",
      "address-family": "ipv4",
      "is-backup": false,
      "advertise-protection": "single",
      "is-on-lan": false
    }
  },
  {
    "link-id": "NorthStar-Live:link:L0110.1200.0003_11.11.
0.17_0110.1200.0002_11.11.0.18:rvs",
    "source": {
      "source-node": "NorthStar-Live:node:0110.1200.
0003",
      "source-tp": "NorthStar-Live:node:0110.1200.
0003:11.11.0.17"
    },
    "destination": {
      "dest-node": "NorthStar-Live:node:0110.1200.0002",

```

```
11.0.18"      "dest-tp": "NorthStar-Live:node:0110.1200.0002:11.
               },
               "ietf-te-topology:te": {
                 "oper-status": "up",
                 "te-link-attributes": {
                   "access-type": "point-to-point",
                   "admin-status": "up",
                   "link-protection-type": "unprotected",
                   "max-link-bandwidth": {
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                 },
                 "unreserved-bandwidth": [
                   {
                     "priority": 0,
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                   {
                     "priority": 1,
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                   {
                     "priority": 2,
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                   {
                     "priority": 3,
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                   {
                     "priority": 4,
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                   {
                     "priority": 5,
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                   {
                     "priority": 6,
                     "te-bandwidth": {
                       "generic": "100000000"
                     }
                   },
                   {
                     "priority": 7,
                     "te-bandwidth": {
```

```

        "generic": "10000000"
      }
    },
    ],
    "te-default-metric": 10,
    "te-igp-metric": 10,
    "administrative-group": "00",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
  }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
  "metric2": 10,
  "ietf-sr-mpls-topology:sr-mpls": {
    "sid": 16,
    "is-persistent": false,
    "is-part-of-set": false,
    "is-local": false,
    "value-type": "index",
    "address-family": "ipv4",
    "is-backup": false,
    "advertise-protection": "single",
    "is-on-lan": false
  }
}
},
{
  "link-id": "NorthStar-Live:link:L0110.1200.0002_11.11.
0.41_0110.1200.0006_11.11.0.42: fwd",
  "source": {
    "source-node": "NorthStar-Live:node:0110.1200.
0002",
    "source-tp": "NorthStar-Live:node:0110.1200.
0002:11.11.0.41"
  },
  "destination": {
    "dest-node": "NorthStar-Live:node:0110.1200.0006",

```

```
11.0.42"      "dest-tp": "NorthStar-Live:node:0110.1200.0006:11.
               },
               "ietf-te-topology:te": {
                 "oper-status": "up",
                 "te-link-attributes": {
                   "access-type": "point-to-point",
                   "admin-status": "up",
                   "link-protection-type": "unprotected",
                   "max-link-bandwidth": {
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                 },
                 "unreserved-bandwidth": [
                   {
                     "priority": 0,
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                   {
                     "priority": 1,
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                   {
                     "priority": 2,
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                   {
                     "priority": 3,
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                   {
                     "priority": 4,
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                   {
                     "priority": 5,
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                   {
                     "priority": 6,
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                   {
                     "priority": 7,
                     "te-bandwidth": {
```

```

        "generic": "10000000"
      }
    },
    ],
    "te-default-metric": 40,
    "te-srlgs": [],
    "te-igp-metric": 20,
    "administrative-group": "01",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
  }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
  "metric2": 20,
  "ietf-sr-mpls-topology:sr-mpls": {
    "sid": 30002,
    "is-persistent": false,
    "is-part-of-set": false,
    "is-local": false,
    "value-type": "index",
    "address-family": "ipv4",
    "is-backup": false,
    "advertise-protection": "single",
    "is-on-lan": false
  }
},
{
  "link-id": "NorthStar-Live:link:L0110.1200.0002_11.11.
0.41_0110.1200.0006_11.11.0.42:rvs",
  "source": {
    "source-node": "NorthStar-Live:node:0110.1200.
0006",
    "source-tp": "NorthStar-Live:node:0110.1200.
0006:11.11.0.42"
  },
  "destination": {

```

```
11.0.41"      "dest-node": "NorthStar-Live:node:0110.1200.0002",
               "dest-tp": "NorthStar-Live:node:0110.1200.0002:11.
               },
               "ietf-te-topology:te": {
                 "oper-status": "up",
                 "te-link-attributes": {
                   "access-type": "point-to-point",
                   "admin-status": "up",
                   "link-protection-type": "unprotected",
                   "max-link-bandwidth": {
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                 },
                 "unreserved-bandwidth": [
                   {
                     "priority": 0,
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                   {
                     "priority": 1,
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                   {
                     "priority": 2,
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                   {
                     "priority": 3,
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                   {
                     "priority": 4,
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                   {
                     "priority": 5,
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                   {
                     "priority": 6,
                     "te-bandwidth": {
                       "generic": "10000000"
                     }
                   },
                   {
                     "priority": 7,
```



```

        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    "te-default-metric": 40,
    "te-srlgs": [],
    "te-igp-metric": 20,
    "administrative-group": "01",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
    },
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric2": 20,
        "ietf-sr-mpls-topology:sr-mpls": {
            "sid": 30002,
            "is-persistent": false,
            "is-part-of-set": false,
            "is-local": false,
            "value-type": "index",
            "address-family": "ipv4",
            "is-backup": false,
            "advertise-protection": "single",
            "is-on-lan": false
        }
    },
    {
        "link-id": "NorthStar-Live:link:L0110.1200.0002_11.11.
0.45_0110.1200.0006_11.11.0.46: fwd",
        "source": {
            "source-node": "NorthStar-Live:node:0110.1200.
0002",
            "source-tp": "NorthStar-Live:node:0110.1200.
0002:11.11.0.45"
        },

```

```
11.0.46"
    "destination": {
      "dest-node": "NorthStar-Live:node:0110.1200.0006",
      "dest-tp": "NorthStar-Live:node:0110.1200.0006:11.0.46"
    },
    "ietf-te-topology:te": {
      "oper-status": "up",
      "te-link-attributes": {
        "access-type": "point-to-point",
        "admin-status": "up",
        "link-protection-type": "unprotected",
        "max-link-bandwidth": {
          "te-bandwidth": {
            "generic": "10000000"
          }
        }
      },
      "unreserved-bandwidth": [
        {
          "priority": 0,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 1,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 2,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 3,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 4,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 5,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 6,
          "te-bandwidth": {
            "generic": "10000000"
          }
        }
      ]
    }
  }
}
```

```

        "priority": 7,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    "te-default-metric": 40,
    "te-srlgs": [],
    "te-igp-metric": 20,
    "administrative-group": "01",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
    },
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric2": 20,
        "ietf-sr-mpls-topology:sr-mpls": {
            "sid": 30002,
            "is-persistent": false,
            "is-part-of-set": false,
            "is-local": false,
            "value-type": "index",
            "address-family": "ipv4",
            "is-backup": false,
            "advertise-protection": "single",
            "is-on-lan": false
        }
    },
    {
        "link-id": "NorthStar-Live:link:L0110.1200.0002_11.11.
0.45_0110.1200.0006_11.11.0.46:rvs",
        "source": {
            "source-node": "NorthStar-Live:node:0110.1200.
0006",
            "source-tp": "NorthStar-Live:node:0110.1200.
0006:11.11.0.46"
        }
    }
}

```

```
11.0.45"
    },
    "destination": {
      "dest-node": "NorthStar-Live:node:0110.1200.0002",
      "dest-tp": "NorthStar-Live:node:0110.1200.0002:11.0.45"
    },
    "ietf-te-topology:te": {
      "oper-status": "up",
      "te-link-attributes": {
        "access-type": "point-to-point",
        "admin-status": "up",
        "link-protection-type": "unprotected",
        "max-link-bandwidth": {
          "te-bandwidth": {
            "generic": "100000000"
          }
        }
      },
      "unreserved-bandwidth": [
        {
          "priority": 0,
          "te-bandwidth": {
            "generic": "100000000"
          }
        },
        {
          "priority": 1,
          "te-bandwidth": {
            "generic": "100000000"
          }
        },
        {
          "priority": 2,
          "te-bandwidth": {
            "generic": "100000000"
          }
        },
        {
          "priority": 3,
          "te-bandwidth": {
            "generic": "100000000"
          }
        },
        {
          "priority": 4,
          "te-bandwidth": {
            "generic": "100000000"
          }
        },
        {
          "priority": 5,
          "te-bandwidth": {
            "generic": "100000000"
          }
        },
        {
          "priority": 6,
          "te-bandwidth": {
            "generic": "100000000"
          }
        }
      ]
    }
  },
}
```

```

        {
            "priority": 7,
            "te-bandwidth": {
                "generic": "10000000"
            }
        }
    ],
    "te-default-metric": 40,
    "te-srlgs": [],
    "te-igp-metric": 20,
    "administrative-group": "01",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30002,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0003_11.11.
0.21_0110.1200.0004_11.11.0.22: fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0003",

```

```

"source-tp": "NorthStar-Live:node:0110.1200.
0003:11.11.0.21"
    },
    "destination": {
      "dest-node": "NorthStar-Live:node:0110.1200.0004",
      "dest-tp": "NorthStar-Live:node:0110.1200.0004:11.
11.0.22"
    },
    "ietf-te-topology:te": {
      "oper-status": "up",
      "te-link-attributes": {
        "access-type": "point-to-point",
        "admin-status": "up",
        "link-protection-type": "unprotected",
        "max-link-bandwidth": {
          "te-bandwidth": {
            "generic": "10000000"
          }
        }
      },
      "unreserved-bandwidth": [
        {
          "priority": 0,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 1,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 2,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 3,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 4,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 5,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 6,
          "te-bandwidth": {
            "generic": "10000000"
          }
        }
      ]
    }
  }
}

```

```

    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "100000000"
    }
  }
],
"te-default-metric": 10,
"te-igp-metric": 10,
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",
"one-way-max-delay-normality": "normal",
"one-way-delay-variation-normality": "normal",
"one-way-packet-loss-normality": "normal",
"two-way-min-delay": 0,
"two-way-min-delay-normality": "normal",
"two-way-max-delay": 0,
"two-way-max-delay-normality": "normal",
"two-way-delay-variation": 0,
"two-way-delay-variation-normality": "normal",
"two-way-packet-loss": 0,
"two-way-packet-loss-normality": "normal",
"one-way-delay-offset": 0,
"measure-interval": 30,
"advertisement-interval": 0,
"suppression-interval": 120
}
},
"ietf-l3-unicast-topology:l3-link-attributes": {
  "metric2": 10,
  "ietf-sr-mpls-topology:sr-mpls": {
    "sid": 17,
    "is-persistent": false,
    "is-part-of-set": false,
    "is-local": false,
    "value-type": "index",
    "address-family": "ipv4",
    "is-backup": false,
    "advertise-protection": "single",
    "is-on-lan": false
  }
}
},
{
  "link-id": "NorthStar-Live:link:L0110.1200.0003_11.11.
0.21_0110.1200.0004_11.11.0.22:rvs",
  "source": {
    "source-node": "NorthStar-Live:node:0110.1200.
0004",

```

```

"source-tp": "NorthStar-Live:node:0110.1200.
0004:11.11.0.22"
    },
    "destination": {
      "dest-node": "NorthStar-Live:node:0110.1200.0003",
      "dest-tp": "NorthStar-Live:node:0110.1200.0003:11.
11.0.21"
    },
    "ietf-te-topology:te": {
      "oper-status": "up",
      "te-link-attributes": {
        "access-type": "point-to-point",
        "admin-status": "up",
        "link-protection-type": "unprotected",
        "max-link-bandwidth": {
          "te-bandwidth": {
            "generic": "10000000"
          }
        }
      },
      "unreserved-bandwidth": [
        {
          "priority": 0,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 1,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 2,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 3,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 4,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 5,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 6,
          "te-bandwidth": {
            "generic": "10000000"
          }
        }
      ]
    }
  }
}

```



```

    }
    },
    {
        "priority": 7,
        "te-bandwidth": {
            "generic": "100000000"
        }
    }
],
"te-default-metric": 10,
"te-igp-metric": 10,
"administrative-group": "00",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",
"one-way-max-delay-normality": "normal",
"one-way-delay-variation-normality": "normal",
"one-way-packet-loss-normality": "normal",
"two-way-min-delay": 0,
"two-way-min-delay-normality": "normal",
"two-way-max-delay": 0,
"two-way-max-delay-normality": "normal",
"two-way-delay-variation": 0,
"two-way-delay-variation-normality": "normal",
"two-way-packet-loss": 0,
"two-way-packet-loss-normality": "normal",
"one-way-delay-offset": 0,
"measure-interval": 30,
"advertisement-interval": 0,
"suppression-interval": 120
}
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 18,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0004_11.11.
0.57_0110.1200.0008_11.11.0.58: fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0004",

```

```

"source-tp": "NorthStar-Live:node:0110.1200.
0004:11.11.0.57"
    },
    "destination": {
      "dest-node": "NorthStar-Live:node:0110.1200.0008",
      "dest-tp": "NorthStar-Live:node:0110.1200.0008:11.
11.0.58"
    },
    "ietf-te-topology:te": {
      "oper-status": "up",
      "te-link-attributes": {
        "access-type": "point-to-point",
        "admin-status": "up",
        "link-protection-type": "unprotected",
        "max-link-bandwidth": {
          "te-bandwidth": {
            "generic": "10000000"
          }
        }
      },
      "unreserved-bandwidth": [
        {
          "priority": 0,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 1,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 2,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 3,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 4,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 5,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 6,
          "te-bandwidth": {
            "generic": "10000000"
          }
        }
      ]
    }
  }
}

```

```

    }
    },
    {
        "priority": 7,
        "te-bandwidth": {
            "generic": "100000000"
        }
    }
],
"te-default-metric": 40,
"te-srlgs": [],
"te-igp-metric": 20,
"administrative-group": "02",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",
"one-way-max-delay-normality": "normal",
"one-way-delay-variation-normality": "normal",
"one-way-packet-loss-normality": "normal",
"two-way-min-delay": 0,
"two-way-min-delay-normality": "normal",
"two-way-max-delay": 0,
"two-way-max-delay-normality": "normal",
"two-way-delay-variation": 0,
"two-way-delay-variation-normality": "normal",
"two-way-packet-loss": 0,
"two-way-packet-loss-normality": "normal",
"one-way-delay-offset": 0,
"measure-interval": 30,
"advertisement-interval": 0,
"suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30002,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0004_11.11.
0.57_0110.1200.0008_11.11.0.58:rvs",
    "source": {

```

```

"source-node": "NorthStar-Live:node:0110.1200.
0008",
"source-tp": "NorthStar-Live:node:0110.1200.
0008:11.11.0.58"
},
"destination": {
  "dest-node": "NorthStar-Live:node:0110.1200.0004",
  "dest-tp": "NorthStar-Live:node:0110.1200.0004:11.
11.0.57"
},
"ietf-te-topology:te": {
  "oper-status": "up",
  "te-link-attributes": {
    "access-type": "point-to-point",
    "admin-status": "up",
    "link-protection-type": "unprotected",
    "max-link-bandwidth": {
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    "unreserved-bandwidth": [
      {
        "priority": 0,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 1,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 2,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 3,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 4,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 5,
        "te-bandwidth": {
          "generic": "10000000"
        }
      },
      {
        "priority": 6,

```

```

        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 7,
        "te-bandwidth": {
            "generic": "10000000"
        }
    }
],
"te-default-metric": 40,
"te-srlgs": [],
"te-igp-metric": 20,
"administrative-group": "02",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",
"one-way-max-delay-normality": "normal",
"one-way-delay-variation-normality": "normal",
"one-way-packet-loss-normality": "normal",
"two-way-min-delay": 0,
"two-way-min-delay-normality": "normal",
"two-way-max-delay": 0,
"two-way-max-delay-normality": "normal",
"two-way-delay-variation": 0,
"two-way-delay-variation-normality": "normal",
"two-way-packet-loss": 0,
"two-way-packet-loss-normality": "normal",
"one-way-delay-offset": 0,
"measure-interval": 30,
"advertisement-interval": 0,
"suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30002,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{
    "link-id": "NorthStar-Live:link:L0110.1200.0004_11.11.
0.61_0110.1200.0008_11.11.0.62: fwd",

```

```

    "source": {
      "source-node": "NorthStar-Live:node:0110.1200.
0004",
      "source-tp": "NorthStar-Live:node:0110.1200.
0004:11.11.0.61"
    },
    "destination": {
      "dest-node": "NorthStar-Live:node:0110.1200.0008",
      "dest-tp": "NorthStar-Live:node:0110.1200.0008:11.
11.0.62"
    },
    "ietf-te-topology:te": {
      "oper-status": "up",
      "te-link-attributes": {
        "access-type": "point-to-point",
        "admin-status": "up",
        "link-protection-type": "unprotected",
        "max-link-bandwidth": {
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
      },
      "unreserved-bandwidth": [
        {
          "priority": 0,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 1,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 2,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 3,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 4,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 5,
          "te-bandwidth": {
            "generic": "10000000"
          }
        }
      ]
    }
  },
  {
    "source": {
      "source-node": "NorthStar-Live:node:0110.1200.
0004",
      "source-tp": "NorthStar-Live:node:0110.1200.
0004:11.11.0.61"
    },
    "destination": {
      "dest-node": "NorthStar-Live:node:0110.1200.0008",
      "dest-tp": "NorthStar-Live:node:0110.1200.0008:11.
11.0.62"
    },
    "ietf-te-topology:te": {
      "oper-status": "up",
      "te-link-attributes": {
        "access-type": "point-to-point",
        "admin-status": "up",
        "link-protection-type": "unprotected",
        "max-link-bandwidth": {
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
      },
      "unreserved-bandwidth": [
        {
          "priority": 0,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 1,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 2,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 3,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 4,
          "te-bandwidth": {
            "generic": "10000000"
          }
        },
        {
          "priority": 5,
          "te-bandwidth": {
            "generic": "10000000"
          }
        }
      ]
    }
  }
]

```

```

        "priority": 6,
        "te-bandwidth": {
            "generic": "10000000"
        }
    },
    {
        "priority": 7,
        "te-bandwidth": {
            "generic": "10000000"
        }
    }
],
"te-default-metric": 40,
"te-srlgs": [],
"te-igp-metric": 20,
"administrative-group": "02",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",
"one-way-max-delay-normality": "normal",
"one-way-delay-variation-normality": "normal",
"one-way-packet-loss-normality": "normal",
"two-way-min-delay": 0,
"two-way-min-delay-normality": "normal",
"two-way-max-delay": 0,
"two-way-max-delay-normality": "normal",
"two-way-delay-variation": 0,
"two-way-delay-variation-normality": "normal",
"two-way-packet-loss": 0,
"two-way-packet-loss-normality": "normal",
"one-way-delay-offset": 0,
"measure-interval": 30,
"advertisement-interval": 0,
"suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30002,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{

```

```

    "link-id": "NorthStar-Live:link:L0110.1200.0004_11.11.
0.61_0110.1200.0008_11.11.0.62:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0008",
        "source-tp": "NorthStar-Live:node:0110.1200.
0008:11.11.0.62"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0004",
        "dest-tp": "NorthStar-Live:node:0110.1200.0004:11.
11.0.61"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 1,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 2,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 3,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 4,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 5,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                }
            ]
        }
    }
}

```



```

        },
        {
            "priority": 6,
            "te-bandwidth": {
                "generic": "100000000"
            }
        },
        {
            "priority": 7,
            "te-bandwidth": {
                "generic": "100000000"
            }
        }
    ],
    "te-default-metric": 40,
    "te-srlgs": [],
    "te-igp-metric": 20,
    "administrative-group": "02",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 20,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 30002,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},

```

```

    {
      "link-id": "NorthStar-Live:link:L0110.1200.0005_11.11.
0.33_0110.1200.0006_11.11.0.34:fwd",
      "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0005",
        "source-tp": "NorthStar-Live:node:0110.1200.
0005:11.11.0.33"
      },
      "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0006",
        "dest-tp": "NorthStar-Live:node:0110.1200.0006:11.
11.0.34"
      },
      "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
          "access-type": "point-to-point",
          "admin-status": "up",
          "link-protection-type": "unprotected",
          "max-link-bandwidth": {
            "te-bandwidth": {
              "generic": "10000000"
            }
          },
          "unreserved-bandwidth": [
            {
              "priority": 0,
              "te-bandwidth": {
                "generic": "10000000"
              }
            },
            {
              "priority": 1,
              "te-bandwidth": {
                "generic": "10000000"
              }
            },
            {
              "priority": 2,
              "te-bandwidth": {
                "generic": "10000000"
              }
            },
            {
              "priority": 3,
              "te-bandwidth": {
                "generic": "10000000"
              }
            },
            {
              "priority": 4,
              "te-bandwidth": {
                "generic": "10000000"
              }
            },
            {
              "priority": 5,
              "te-bandwidth": {
                "generic": "10000000"
              }
            }
          ]
        }
      }
    }

```

```

    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "100000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "100000000"
    }
  }
],
"te-default-metric": 10,
"te-igp-metric": 10,
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",
"one-way-max-delay-normality": "normal",
"one-way-delay-variation-normality": "normal",
"one-way-packet-loss-normality": "normal",
"two-way-min-delay": 0,
"two-way-min-delay-normality": "normal",
"two-way-max-delay": 0,
"two-way-max-delay-normality": "normal",
"two-way-delay-variation": 0,
"two-way-delay-variation-normality": "normal",
"two-way-packet-loss": 0,
"two-way-packet-loss-normality": "normal",
"one-way-delay-offset": 0,
"measure-interval": 30,
"advertisement-interval": 0,
"suppression-interval": 120
}
},
"ietf-l3-unicast-topology:l3-link-attributes": {
  "metric2": 10,
  "ietf-sr-mpls-topology:sr-mpls": {
    "sid": 16,
    "is-persistent": false,
    "is-part-of-set": false,
    "is-local": false,
    "value-type": "index",
    "address-family": "ipv4",
    "is-backup": false,
    "advertise-protection": "single",
    "is-on-lan": false
  }
}
},
{

```

```

    "link-id": "NorthStar-Live:link:L0110.1200.0005_11.11.
0.33_0110.1200.0006_11.11.0.34:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0006",
        "source-tp": "NorthStar-Live:node:0110.1200.
0006:11.11.0.34"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0005",
        "dest-tp": "NorthStar-Live:node:0110.1200.0005:11.
11.0.33"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 1,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 2,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 3,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 4,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 5,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                }
            ]
        }
    }
}

```

```

        },
        {
            "priority": 6,
            "te-bandwidth": {
                "generic": "100000000"
            }
        },
        {
            "priority": 7,
            "te-bandwidth": {
                "generic": "100000000"
            }
        }
    ],
    "te-default-metric": 10,
    "te-igp-metric": 10,
    "administrative-group": "00",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 19,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{

```

```

    "link-id": "NorthStar-Live:link:L0110.1200.0005_11.11.
0.37_0110.1200.0008_11.11.0.38:fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0005",
        "source-tp": "NorthStar-Live:node:0110.1200.
0005:11.11.0.37"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0008",
        "dest-tp": "NorthStar-Live:node:0110.1200.0008:11.
11.0.38"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 1,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 2,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 3,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 4,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 5,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                }
            ]
        }
    }
}

```

```

    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 10,
  "te-igp-metric": 10,
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
"normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
"normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
"normal",
  "one-way-min-delay-normality": "normal",
  "one-way-max-delay-normality": "normal",
  "one-way-delay-variation-normality": "normal",
  "one-way-packet-loss-normality": "normal",
  "two-way-min-delay": 0,
  "two-way-min-delay-normality": "normal",
  "two-way-max-delay": 0,
  "two-way-max-delay-normality": "normal",
  "two-way-delay-variation": 0,
  "two-way-delay-variation-normality": "normal",
  "two-way-packet-loss": 0,
  "two-way-packet-loss-normality": "normal",
  "one-way-delay-offset": 0,
  "measure-interval": 30,
  "advertisement-interval": 0,
  "suppression-interval": 120
},
"ietf-l3-unicast-topology:l3-link-attributes": {
  "metric2": 10,
  "ietf-sr-mpls-topology:sr-mpls": {
    "sid": 17,
    "is-persistent": false,
    "is-part-of-set": false,
    "is-local": false,
    "value-type": "index",
    "address-family": "ipv4",
    "is-backup": false,
    "advertise-protection": "single",
    "is-on-lan": false
  }
}
},
{

```

```

    "link-id": "NorthStar-Live:link:L0110.1200.0005_11.11.
0.37_0110.1200.0008_11.11.0.38:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0008",
        "source-tp": "NorthStar-Live:node:0110.1200.
0008:11.11.0.38"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0005",
        "dest-tp": "NorthStar-Live:node:0110.1200.0005:11.
11.0.37"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 1,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 2,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 3,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 4,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 5,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                }
            ]
        }
    }
}

```



```

        },
        {
            "priority": 6,
            "te-bandwidth": {
                "generic": "100000000"
            }
        },
        {
            "priority": 7,
            "te-bandwidth": {
                "generic": "100000000"
            }
        }
    ],
    "te-default-metric": 10,
    "te-igp-metric": 10,
    "administrative-group": "00",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 16,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{

```

```

    "link-id": "NorthStar-Live:link:L0110.1200.0007_11.11.
0.49_0110.1200.0006_11.11.0.50:fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0006",
        "source-tp": "NorthStar-Live:node:0110.1200.
0006:11.11.0.50"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0007",
        "dest-tp": "NorthStar-Live:node:0110.1200.0007:11.
11.0.49"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 1,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 2,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 3,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 4,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 5,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                }
            ]
        }
    }
}

```

```

    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 10,
  "te-igp-metric": 10,
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
"normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
"normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
"normal",
  "one-way-min-delay-normality": "normal",
  "one-way-max-delay-normality": "normal",
  "one-way-delay-variation-normality": "normal",
  "one-way-packet-loss-normality": "normal",
  "two-way-min-delay": 0,
  "two-way-min-delay-normality": "normal",
  "two-way-max-delay": 0,
  "two-way-max-delay-normality": "normal",
  "two-way-delay-variation": 0,
  "two-way-delay-variation-normality": "normal",
  "two-way-packet-loss": 0,
  "two-way-packet-loss-normality": "normal",
  "one-way-delay-offset": 0,
  "measure-interval": 30,
  "advertisement-interval": 0,
  "suppression-interval": 120
},
"ietf-l3-unicast-topology:l3-link-attributes": {
  "metric2": 10,
  "ietf-sr-mpls-topology:sr-mpls": {
    "sid": 23,
    "is-persistent": false,
    "is-part-of-set": false,
    "is-local": false,
    "value-type": "index",
    "address-family": "ipv4",
    "is-backup": false,
    "advertise-protection": "single",
    "is-on-lan": false
  }
}
},
{

```

```

    "link-id": "NorthStar-Live:link:L0110.1200.0007_11.11.
0.49_0110.1200.0006_11.11.0.50:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0007",
        "source-tp": "NorthStar-Live:node:0110.1200.
0007:11.11.0.49"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0006",
        "dest-tp": "NorthStar-Live:node:0110.1200.0006:11.
11.0.50"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 1,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 2,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 3,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 4,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 5,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                }
            ]
        }
    }
}

```

```

        },
        {
            "priority": 6,
            "te-bandwidth": {
                "generic": "100000000"
            }
        },
        {
            "priority": 7,
            "te-bandwidth": {
                "generic": "100000000"
            }
        }
    ],
    "te-default-metric": 10,
    "te-igp-metric": 10,
    "administrative-group": "00",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 17,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{

```

```

    "link-id": "NorthStar-Live:link:L0110.1200.0007_11.11.
0.53_0110.1200.0008_11.11.0.54:fwd",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0007",
        "source-tp": "NorthStar-Live:node:0110.1200.
0007:11.11.0.53"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0008",
        "dest-tp": "NorthStar-Live:node:0110.1200.0008:11.
11.0.54"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 1,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 2,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 3,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 4,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 5,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                }
            ]
        }
    }
}

```

```

    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000"
      }
    }
  ],
  "te-default-metric": 10,
  "te-igp-metric": 10,
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
"normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
"normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
"normal",
  "one-way-min-delay-normality": "normal",
  "one-way-max-delay-normality": "normal",
  "one-way-delay-variation-normality": "normal",
  "one-way-packet-loss-normality": "normal",
  "two-way-min-delay": 0,
  "two-way-min-delay-normality": "normal",
  "two-way-max-delay": 0,
  "two-way-max-delay-normality": "normal",
  "two-way-delay-variation": 0,
  "two-way-delay-variation-normality": "normal",
  "two-way-packet-loss": 0,
  "two-way-packet-loss-normality": "normal",
  "one-way-delay-offset": 0,
  "measure-interval": 30,
  "advertisement-interval": 0,
  "suppression-interval": 120
},
"ietf-l3-unicast-topology:l3-link-attributes": {
  "metric2": 10,
  "ietf-sr-mpls-topology:sr-mpls": {
    "sid": 16,
    "is-persistent": false,
    "is-part-of-set": false,
    "is-local": false,
    "value-type": "index",
    "address-family": "ipv4",
    "is-backup": false,
    "advertise-protection": "single",
    "is-on-lan": false
  }
}
},
{

```

```

    "link-id": "NorthStar-Live:link:L0110.1200.0007_11.11.
0.53_0110.1200.0008_11.11.0.54:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:0110.1200.
0008",
        "source-tp": "NorthStar-Live:node:0110.1200.
0008:11.11.0.54"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:0110.1200.0007",
        "dest-tp": "NorthStar-Live:node:0110.1200.0007:11.
11.0.53"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 1,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 2,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 3,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 4,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                },
                {
                    "priority": 5,
                    "te-bandwidth": {
                        "generic": "10000000"
                    }
                }
            ]
        }
    }
}

```



```

        },
        {
            "priority": 6,
            "te-bandwidth": {
                "generic": "100000000"
            }
        },
        {
            "priority": 7,
            "te-bandwidth": {
                "generic": "100000000"
            }
        }
    ],
    "te-default-metric": 10,
    "te-igp-metric": 10,
    "administrative-group": "00",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
    }
},
"ietf-l3-unicast-topology:l3-link-attributes": {
    "metric2": 10,
    "ietf-sr-mpls-topology:sr-mpls": {
        "sid": 19,
        "is-persistent": false,
        "is-part-of-set": false,
        "is-local": false,
        "value-type": "index",
        "address-family": "ipv4",
        "is-backup": false,
        "advertise-protection": "single",
        "is-on-lan": false
    }
}
},
{

```

```

0.0.2_10.0.0.2: fwd",
    "link-id": "NorthStar-Live:link:L1.1.1.1_10.0.0.1_127.
1"
    "source": {
        "source-node": "NorthStar-Live:node:1.1.1.1",
        "source-tp": "NorthStar-Live:node:1.1.1.1:10.0.0.
2"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:127.0.0.2",
        "dest-tp": "NorthStar-Live:node:127.0.0.2:10.0.0.
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "10000000000"
                    }
                },
                {
                    "priority": 1,
                    "te-bandwidth": {
                        "generic": "10000000000"
                    }
                },
                {
                    "priority": 2,
                    "te-bandwidth": {
                        "generic": "10000000000"
                    }
                },
                {
                    "priority": 3,
                    "te-bandwidth": {
                        "generic": "10000000000"
                    }
                },
                {
                    "priority": 4,
                    "te-bandwidth": {
                        "generic": "10000000000"
                    }
                },
                {
                    "priority": 5,
                    "te-bandwidth": {
                        "generic": "10000000000"
                    }
                }
            ]
        }
    },
    "unreserved-bandwidth": [
        {
            "priority": 0,
            "te-bandwidth": {
                "generic": "10000000000"
            }
        },
        {
            "priority": 1,
            "te-bandwidth": {
                "generic": "10000000000"
            }
        },
        {
            "priority": 2,
            "te-bandwidth": {
                "generic": "10000000000"
            }
        },
        {
            "priority": 3,
            "te-bandwidth": {
                "generic": "10000000000"
            }
        },
        {
            "priority": 4,
            "te-bandwidth": {
                "generic": "10000000000"
            }
        },
        {
            "priority": 5,
            "te-bandwidth": {
                "generic": "10000000000"
            }
        }
    ]
}

```

```

        {
            "priority": 6,
            "te-bandwidth": {
                "generic": "10000000000"
            }
        },
        {
            "priority": 7,
            "te-bandwidth": {
                "generic": "10000000000"
            }
        }
    ],
    "te-default-metric": 10,
    "te-srlgs": [],
    "te-igp-metric": 10,
    "administrative-group": "01",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
    }
},
{
    "link-id": "NorthStar-Live:link:L1.1.1.1_10.0.0.1_127.
0.0.2_10.0.0.2:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:127.0.0.2",
        "source-tp": "NorthStar-Live:node:127.0.0.2:10.0.
0.2"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:1.1.1.1",
        "dest-tp": "NorthStar-Live:node:1.1.1.1:10.0.0.1"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {

```

```
"access-type": "point-to-point",
"admin-status": "up",
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "10000000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "10000000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "10000000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "10000000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "10000000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "10000000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "10000000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "10000000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "10000000000"
    }
  }
],
"te-default-metric": 10,
"te-srlgs": [],
```

```

        "te-igp-metric": 10,
        "administrative-group": "01",
        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
{
    "link-id": "NorthStar-Live:link:L1.1.1.1_10.0.0.5_127.
0.0.3_10.0.0.6:fwd",
    "source": {
        "source-node": "NorthStar-Live:node:1.1.1.1",
        "source-tp": "NorthStar-Live:node:1.1.1.1:10.0.0.
5"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:127.0.0.3",
        "dest-tp": "NorthStar-Live:node:127.0.0.3:10.0.0.
6"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "200000000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "200000000000"
                    }
                }
            ]
        }
    }
}

```

```

    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    }
  ],
  "te-default-metric": 20,
  "te-srlgs": [],
  "te-igp-metric": 20,
  "administrative-group": "01",
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
    "normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
    "normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
    "normal",
  "one-way-min-delay-normality": "normal",
  "one-way-max-delay-normality": "normal",
  "one-way-delay-variation-normality": "normal",

```

```

        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
      }
    },
    {
      "link-id": "NorthStar-Live:link:L1.1.1.1_10.0.0.5_127.
0.0.3_10.0.0.6:rvs",
      "source": {
        "source-node": "NorthStar-Live:node:127.0.0.3",
        "source-tp": "NorthStar-Live:node:127.0.0.3:10.0.
0.6"
      },
      "destination": {
        "dest-node": "NorthStar-Live:node:1.1.1.1",
        "dest-tp": "NorthStar-Live:node:1.1.1.1:10.0.0.5"
      },
      "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
          "access-type": "point-to-point",
          "admin-status": "up",
          "link-protection-type": "unprotected",
          "max-link-bandwidth": {
            "te-bandwidth": {
              "generic": "20000000000"
            }
          },
          "unreserved-bandwidth": [
            {
              "priority": 0,
              "te-bandwidth": {
                "generic": "20000000000"
              }
            },
            {
              "priority": 1,
              "te-bandwidth": {
                "generic": "20000000000"
              }
            },
            {
              "priority": 2,
              "te-bandwidth": {
                "generic": "20000000000"
              }
            },
            {
              "priority": 3,

```

```

        "te-bandwidth": {
            "generic": "20000000000"
        }
    },
    {
        "priority": 4,
        "te-bandwidth": {
            "generic": "20000000000"
        }
    },
    {
        "priority": 5,
        "te-bandwidth": {
            "generic": "20000000000"
        }
    },
    {
        "priority": 6,
        "te-bandwidth": {
            "generic": "20000000000"
        }
    },
    {
        "priority": 7,
        "te-bandwidth": {
            "generic": "20000000000"
        }
    }
],
"te-default-metric": 20,
"te-srlgs": [],
"te-igp-metric": 20,
"administrative-group": "01",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",
"one-way-max-delay-normality": "normal",
"one-way-delay-variation-normality": "normal",
"one-way-packet-loss-normality": "normal",
"two-way-min-delay": 0,
"two-way-min-delay-normality": "normal",
"two-way-max-delay": 0,
"two-way-max-delay-normality": "normal",
"two-way-delay-variation": 0,
"two-way-delay-variation-normality": "normal",
"two-way-packet-loss": 0,
"two-way-packet-loss-normality": "normal",
"one-way-delay-offset": 0,
"measure-interval": 30,
"advertisement-interval": 0,
"suppression-interval": 120
    }
}

```



```

    },
    {
      "link-id": "NorthStar-Live:link:L127.0.0.1_10.0.0.
13_127.0.0.3_10.0.0.14:fwd",
      "source": {
        "source-node": "NorthStar-Live:node:127.0.0.1",
        "source-tp": "NorthStar-Live:node:127.0.0.1:10.0.
0.13"
      },
      "destination": {
        "dest-node": "NorthStar-Live:node:127.0.0.3",
        "dest-tp": "NorthStar-Live:node:127.0.0.3:10.0.0.
14"
      },
      "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
          "access-type": "point-to-point",
          "admin-status": "up",
          "link-protection-type": "unprotected",
          "max-link-bandwidth": {
            "te-bandwidth": {
              "generic": "20000000000"
            }
          },
          "unreserved-bandwidth": [
            {
              "priority": 0,
              "te-bandwidth": {
                "generic": "20000000000"
              }
            },
            {
              "priority": 1,
              "te-bandwidth": {
                "generic": "20000000000"
              }
            },
            {
              "priority": 2,
              "te-bandwidth": {
                "generic": "20000000000"
              }
            },
            {
              "priority": 3,
              "te-bandwidth": {
                "generic": "20000000000"
              }
            },
            {
              "priority": 4,
              "te-bandwidth": {
                "generic": "20000000000"
              }
            },
            {
              "priority": 5,
              "te-bandwidth": {
                "generic": "20000000000"
              }
            }
          ]
        }
      }
    }
  ]
}

```

```

    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    }
  ],
  "te-default-metric": 20,
  "te-srlgs": [],
  "te-igp-metric": 20,
  "administrative-group": "01",
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
    "normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
    "normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
    "normal",
  "one-way-min-delay-normality": "normal",
  "one-way-max-delay-normality": "normal",
  "one-way-delay-variation-normality": "normal",
  "one-way-packet-loss-normality": "normal",
  "two-way-min-delay": 0,
  "two-way-min-delay-normality": "normal",
  "two-way-max-delay": 0,
  "two-way-max-delay-normality": "normal",
  "two-way-delay-variation": 0,
  "two-way-delay-variation-normality": "normal",
  "two-way-packet-loss": 0,
  "two-way-packet-loss-normality": "normal",
  "one-way-delay-offset": 0,
  "measure-interval": 30,
  "advertisement-interval": 0,
  "suppression-interval": 120
}
},
{
  "link-id": "NorthStar-Live:link:L127.0.0.1_10.0.0.
13_127.0.0.3_10.0.0.14:rvs",
  "source": {
    "source-node": "NorthStar-Live:node:127.0.0.3",
    "source-tp": "NorthStar-Live:node:127.0.0.3:10.0.
0.14"
  },
  "destination": {
    "dest-node": "NorthStar-Live:node:127.0.0.1",
    "dest-tp": "NorthStar-Live:node:127.0.0.1:10.0.0.
13"
  },
},

```

```
"ietf-te-topology:te": {
  "oper-status": "up",
  "te-link-attributes": {
    "access-type": "point-to-point",
    "admin-status": "up",
    "link-protection-type": "unprotected",
    "max-link-bandwidth": {
      "te-bandwidth": {
        "generic": "20000000000"
      }
    },
    "unreserved-bandwidth": [
      {
        "priority": 0,
        "te-bandwidth": {
          "generic": "20000000000"
        }
      },
      {
        "priority": 1,
        "te-bandwidth": {
          "generic": "20000000000"
        }
      },
      {
        "priority": 2,
        "te-bandwidth": {
          "generic": "20000000000"
        }
      },
      {
        "priority": 3,
        "te-bandwidth": {
          "generic": "20000000000"
        }
      },
      {
        "priority": 4,
        "te-bandwidth": {
          "generic": "20000000000"
        }
      },
      {
        "priority": 5,
        "te-bandwidth": {
          "generic": "20000000000"
        }
      },
      {
        "priority": 6,
        "te-bandwidth": {
          "generic": "20000000000"
        }
      },
      {
        "priority": 7,
        "te-bandwidth": {
          "generic": "20000000000"
        }
      }
    ]
  }
}
```

```

],
"te-default-metric": 20,
"te-srlgs": [],
"te-igp-metric": 20,
"administrative-group": "01",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",
"one-way-max-delay-normality": "normal",
"one-way-delay-variation-normality": "normal",
"one-way-packet-loss-normality": "normal",
"two-way-min-delay": 0,
"two-way-min-delay-normality": "normal",
"two-way-max-delay": 0,
"two-way-max-delay-normality": "normal",
"two-way-delay-variation": 0,
"two-way-delay-variation-normality": "normal",
"two-way-packet-loss": 0,
"two-way-packet-loss-normality": "normal",
"one-way-delay-offset": 0,
"measure-interval": 30,
"advertisement-interval": 0,
"suppression-interval": 120
    }
  },
  {
    "link-id": "NorthStar-Live:link:L127.0.0.2_10.0.0.
10_127.0.0.1_10.0.0.9:fwd",
    "source": {
      "source-node": "NorthStar-Live:node:127.0.0.1",
      "source-tp": "NorthStar-Live:node:127.0.0.1:10.0.
0.9"
    },
    "destination": {
      "dest-node": "NorthStar-Live:node:127.0.0.2",
      "dest-tp": "NorthStar-Live:node:127.0.0.2:10.0.0.
10"
    },
    "ietf-te-topology:te": {
      "oper-status": "up",
      "te-link-attributes": {
        "access-type": "point-to-point",
        "admin-status": "up",
        "link-protection-type": "unprotected",
        "max-link-bandwidth": {
          "te-bandwidth": {
            "generic": "10000000000"
          }
        },
        "unreserved-bandwidth": [
          {
            "priority": 0,

```

```

        "te-bandwidth": {
            "generic": "10000000000"
        }
    },
    {
        "priority": 1,
        "te-bandwidth": {
            "generic": "10000000000"
        }
    },
    {
        "priority": 2,
        "te-bandwidth": {
            "generic": "10000000000"
        }
    },
    {
        "priority": 3,
        "te-bandwidth": {
            "generic": "10000000000"
        }
    },
    {
        "priority": 4,
        "te-bandwidth": {
            "generic": "10000000000"
        }
    },
    {
        "priority": 5,
        "te-bandwidth": {
            "generic": "10000000000"
        }
    },
    {
        "priority": 6,
        "te-bandwidth": {
            "generic": "10000000000"
        }
    },
    {
        "priority": 7,
        "te-bandwidth": {
            "generic": "10000000000"
        }
    }
],
"te-default-metric": 10,
"te-srlgs": [],
"te-igp-metric": 10,
"administrative-group": "01",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",

```

```

        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
}
},
{
    "link-id": "NorthStar-Live:link:L127.0.0.2_10.0.0.
10_127.0.0.1_10.0.0.9:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:127.0.0.2",
        "source-tp": "NorthStar-Live:node:127.0.0.2:10.0.
0.10"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:127.0.0.1",
        "dest-tp": "NorthStar-Live:node:127.0.0.1:10.0.0.
9"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "10000000000"
                    }
                },
                {
                    "priority": 1,
                    "te-bandwidth": {
                        "generic": "10000000000"
                    }
                },
                {
                    "priority": 2,
                    "te-bandwidth": {
                        "generic": "10000000000"
                    }
                }
            ]
        }
    }
}

```

```

    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "100000000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "100000000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "100000000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "100000000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "100000000000"
    }
  }
],
"te-default-metric": 10,
"te-srlgs": [],
"te-igp-metric": 10,
"administrative-group": "01",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",
"one-way-max-delay-normality": "normal",
"one-way-delay-variation-normality": "normal",
"one-way-packet-loss-normality": "normal",
"two-way-min-delay": 0,
"two-way-min-delay-normality": "normal",
"two-way-max-delay": 0,
"two-way-max-delay-normality": "normal",
"two-way-delay-variation": 0,
"two-way-delay-variation-normality": "normal",
"two-way-packet-loss": 0,
"two-way-packet-loss-normality": "normal",
"one-way-delay-offset": 0,
"measure-interval": 30,

```

```

        "advertisement-interval": 0,
        "suppression-interval": 120
    }
}
},
{
    "link-id": "NorthStar-Live:link:L127.0.0.2_10.0.0.
17_127.0.0.3_10.0.0.18:fwd",
    "source": {
        "source-node": "NorthStar-Live:node:127.0.0.2",
        "source-tp": "NorthStar-Live:node:127.0.0.2:10.0.
0.17"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:127.0.0.3",
        "dest-tp": "NorthStar-Live:node:127.0.0.3:10.0.0.
18"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "10000000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "10000000000"
                    }
                },
                {
                    "priority": 1,
                    "te-bandwidth": {
                        "generic": "10000000000"
                    }
                },
                {
                    "priority": 2,
                    "te-bandwidth": {
                        "generic": "10000000000"
                    }
                },
                {
                    "priority": 3,
                    "te-bandwidth": {
                        "generic": "10000000000"
                    }
                },
                {
                    "priority": 4,
                    "te-bandwidth": {
                        "generic": "10000000000"
                    }
                }
            ]
        }
    }
},

```



```

        {
            "priority": 5,
            "te-bandwidth": {
                "generic": "10000000000"
            }
        },
        {
            "priority": 6,
            "te-bandwidth": {
                "generic": "10000000000"
            }
        },
        {
            "priority": 7,
            "te-bandwidth": {
                "generic": "10000000000"
            }
        }
    ],
    "te-default-metric": 10,
    "te-srlgs": [],
    "te-igp-metric": 10,
    "administrative-group": "01",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
    }
},
{
    "link-id": "NorthStar-Live:link:L127.0.0.2_10.0.0.
17_127.0.0.3_10.0.0.18:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:127.0.0.3",
        "source-tp": "NorthStar-Live:node:127.0.0.3:10.0.
0.18"
    },
    "destination": {

```

```
17"      "dest-node": "NorthStar-Live:node:127.0.0.2",
        "dest-tp": "NorthStar-Live:node:127.0.0.2:10.0.0.
    },
    "ietf-te-topology:te": {
      "oper-status": "up",
      "te-link-attributes": {
        "access-type": "point-to-point",
        "admin-status": "up",
        "link-protection-type": "unprotected",
        "max-link-bandwidth": {
          "te-bandwidth": {
            "generic": "10000000000"
          }
        },
        "unreserved-bandwidth": [
          {
            "priority": 0,
            "te-bandwidth": {
              "generic": "10000000000"
            }
          },
          {
            "priority": 1,
            "te-bandwidth": {
              "generic": "10000000000"
            }
          },
          {
            "priority": 2,
            "te-bandwidth": {
              "generic": "10000000000"
            }
          },
          {
            "priority": 3,
            "te-bandwidth": {
              "generic": "10000000000"
            }
          },
          {
            "priority": 4,
            "te-bandwidth": {
              "generic": "10000000000"
            }
          },
          {
            "priority": 5,
            "te-bandwidth": {
              "generic": "10000000000"
            }
          },
          {
            "priority": 6,
            "te-bandwidth": {
              "generic": "10000000000"
            }
          },
          {
            "priority": 7,
```

```

        "te-bandwidth": {
            "generic": "10000000000"
        }
    },
    "te-default-metric": 10,
    "te-srlgs": [],
    "te-igp-metric": 10,
    "administrative-group": "01",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
    "one-way-available-bandwidth-normality":
"normal",
    "one-way-utilized-bandwidth": "0x0p0",
    "one-way-utilized-bandwidth-normality":
"normal",
    "one-way-min-delay-normality": "normal",
    "one-way-max-delay-normality": "normal",
    "one-way-delay-variation-normality": "normal",
    "one-way-packet-loss-normality": "normal",
    "two-way-min-delay": 0,
    "two-way-min-delay-normality": "normal",
    "two-way-max-delay": 0,
    "two-way-max-delay-normality": "normal",
    "two-way-delay-variation": 0,
    "two-way-delay-variation-normality": "normal",
    "two-way-packet-loss": 0,
    "two-way-packet-loss-normality": "normal",
    "one-way-delay-offset": 0,
    "measure-interval": 30,
    "advertisement-interval": 0,
    "suppression-interval": 120
    }
},
{
    "link-id": "NorthStar-Live:link:L127.0.0.2_10.0.0.
21_127.0.0.4_10.0.0.22:fwd",
    "source": {
        "source-node": "NorthStar-Live:node:127.0.0.2",
        "source-tp": "NorthStar-Live:node:127.0.0.2:10.0.
0.21"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:127.0.0.4",
        "dest-tp": "NorthStar-Live:node:127.0.0.4:10.0.0.
22"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "20000000000"
                }
            }
        }
    }
}

```

```
    },
    "unreserved-bandwidth": [
      {
        "priority": 0,
        "te-bandwidth": {
          "generic": "200000000000"
        }
      },
      {
        "priority": 1,
        "te-bandwidth": {
          "generic": "200000000000"
        }
      },
      {
        "priority": 2,
        "te-bandwidth": {
          "generic": "200000000000"
        }
      },
      {
        "priority": 3,
        "te-bandwidth": {
          "generic": "200000000000"
        }
      },
      {
        "priority": 4,
        "te-bandwidth": {
          "generic": "200000000000"
        }
      },
      {
        "priority": 5,
        "te-bandwidth": {
          "generic": "200000000000"
        }
      },
      {
        "priority": 6,
        "te-bandwidth": {
          "generic": "200000000000"
        }
      },
      {
        "priority": 7,
        "te-bandwidth": {
          "generic": "200000000000"
        }
      }
    ],
    "te-default-metric": 20,
    "te-srlgs": [],
    "te-igp-metric": 20,
    "administrative-group": "01",
    "one-way-residual-bandwidth": "0x0p0",
    "one-way-residual-bandwidth-normality":
"normal",
    "one-way-available-bandwidth": "0x0p0",
```

```

        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
{
    "link-id": "NorthStar-Live:link:L127.0.0.2_10.0.0.
21_127.0.0.4_10.0.0.22:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:127.0.0.4",
        "source-tp": "NorthStar-Live:node:127.0.0.4:10.0.
0.22"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:127.0.0.2",
        "dest-tp": "NorthStar-Live:node:127.0.0.2:10.0.0.
21"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "20000000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "20000000000"
                    }
                },
                {
                    "priority": 1,
                    "te-bandwidth": {
                        "generic": "20000000000"
                    }
                }
            ]
        }
    }
}

```

```

    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    }
  ],
  "te-default-metric": 20,
  "te-srlgs": [],
  "te-igp-metric": 20,
  "administrative-group": "01",
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
"normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
"normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
"normal",
  "one-way-min-delay-normality": "normal",
  "one-way-max-delay-normality": "normal",
  "one-way-delay-variation-normality": "normal",
  "one-way-packet-loss-normality": "normal",
  "two-way-min-delay": 0,
  "two-way-min-delay-normality": "normal",
  "two-way-max-delay": 0,
  "two-way-max-delay-normality": "normal",
  "two-way-delay-variation": 0,

```

```

        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
      }
    },
    {
      "link-id": "NorthStar-Live:link:L127.0.0.3_10.0.0.
25_127.0.0.4_10.0.0.26:fwd",
      "source": {
        "source-node": "NorthStar-Live:node:127.0.0.3",
        "source-tp": "NorthStar-Live:node:127.0.0.3:10.0.
0.25"
      },
      "destination": {
        "dest-node": "NorthStar-Live:node:127.0.0.4",
        "dest-tp": "NorthStar-Live:node:127.0.0.4:10.0.0.
26"
      },
      "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
          "access-type": "point-to-point",
          "admin-status": "up",
          "link-protection-type": "unprotected",
          "max-link-bandwidth": {
            "te-bandwidth": {
              "generic": "10000000000"
            }
          },
          "unreserved-bandwidth": [
            {
              "priority": 0,
              "te-bandwidth": {
                "generic": "10000000000"
              }
            },
            {
              "priority": 1,
              "te-bandwidth": {
                "generic": "10000000000"
              }
            },
            {
              "priority": 2,
              "te-bandwidth": {
                "generic": "10000000000"
              }
            },
            {
              "priority": 3,
              "te-bandwidth": {
                "generic": "10000000000"
              }
            }
          ]
        }
      }
    }
  ]
}

```

```

        "priority": 4,
        "te-bandwidth": {
            "generic": "10000000000"
        }
    },
    {
        "priority": 5,
        "te-bandwidth": {
            "generic": "10000000000"
        }
    },
    {
        "priority": 6,
        "te-bandwidth": {
            "generic": "10000000000"
        }
    },
    {
        "priority": 7,
        "te-bandwidth": {
            "generic": "10000000000"
        }
    }
],
"te-default-metric": 10,
"te-srlgs": [],
"te-igp-metric": 10,
"administrative-group": "01",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",
"one-way-max-delay-normality": "normal",
"one-way-delay-variation-normality": "normal",
"one-way-packet-loss-normality": "normal",
"two-way-min-delay": 0,
"two-way-min-delay-normality": "normal",
"two-way-max-delay": 0,
"two-way-max-delay-normality": "normal",
"two-way-delay-variation": 0,
"two-way-delay-variation-normality": "normal",
"two-way-packet-loss": 0,
"two-way-packet-loss-normality": "normal",
"one-way-delay-offset": 0,
"measure-interval": 30,
"advertisement-interval": 0,
"suppression-interval": 120
    }
},
{
    "link-id": "NorthStar-Live:link:L127.0.0.3_10.0.0.
25_127.0.0.4_10.0.0.26:rvs",
    "source": {

```



```

    "source-node": "NorthStar-Live:node:127.0.0.4",
    "source-tp": "NorthStar-Live:node:127.0.0.4:10.0.
0.26"
  },
  "destination": {
    "dest-node": "NorthStar-Live:node:127.0.0.3",
    "dest-tp": "NorthStar-Live:node:127.0.0.3:10.0.0.
25"
  },
  "ietf-te-topology:te": {
    "oper-status": "up",
    "te-link-attributes": {
      "access-type": "point-to-point",
      "admin-status": "up",
      "link-protection-type": "unprotected",
      "max-link-bandwidth": {
        "te-bandwidth": {
          "generic": "10000000000"
        }
      },
      "unreserved-bandwidth": [
        {
          "priority": 0,
          "te-bandwidth": {
            "generic": "10000000000"
          }
        },
        {
          "priority": 1,
          "te-bandwidth": {
            "generic": "10000000000"
          }
        },
        {
          "priority": 2,
          "te-bandwidth": {
            "generic": "10000000000"
          }
        },
        {
          "priority": 3,
          "te-bandwidth": {
            "generic": "10000000000"
          }
        },
        {
          "priority": 4,
          "te-bandwidth": {
            "generic": "10000000000"
          }
        },
        {
          "priority": 5,
          "te-bandwidth": {
            "generic": "10000000000"
          }
        },
        {
          "priority": 6,
          "te-bandwidth": {

```

```

        "generic": "10000000000"
    }
},
{
    "priority": 7,
    "te-bandwidth": {
        "generic": "10000000000"
    }
}
],
"te-default-metric": 10,
"te-srlgs": [],
"te-igp-metric": 10,
"administrative-group": "01",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",
"one-way-max-delay-normality": "normal",
"one-way-delay-variation-normality": "normal",
"one-way-packet-loss-normality": "normal",
"two-way-min-delay": 0,
"two-way-min-delay-normality": "normal",
"two-way-max-delay": 0,
"two-way-max-delay-normality": "normal",
"two-way-delay-variation": 0,
"two-way-delay-variation-normality": "normal",
"two-way-packet-loss": 0,
"two-way-packet-loss-normality": "normal",
"one-way-delay-offset": 0,
"measure-interval": 30,
"advertisement-interval": 0,
"suppression-interval": 120
    }
}
},
{
    "link-id": "NorthStar-Live:link:L127.0.0.3_10.0.0.
29_127.0.0.5_10.0.0.30:fwd",
    "source": {
        "source-node": "NorthStar-Live:node:127.0.0.3",
        "source-tp": "NorthStar-Live:node:127.0.0.3:10.0.
0.29"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:127.0.0.5",
        "dest-tp": "NorthStar-Live:node:127.0.0.5:10.0.0.
30"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",

```

```
"link-protection-type": "unprotected",
"max-link-bandwidth": {
  "te-bandwidth": {
    "generic": "20000000000"
  }
},
"unreserved-bandwidth": [
  {
    "priority": 0,
    "te-bandwidth": {
      "generic": "20000000000"
    }
  },
  {
    "priority": 1,
    "te-bandwidth": {
      "generic": "20000000000"
    }
  },
  {
    "priority": 2,
    "te-bandwidth": {
      "generic": "20000000000"
    }
  },
  {
    "priority": 3,
    "te-bandwidth": {
      "generic": "20000000000"
    }
  },
  {
    "priority": 4,
    "te-bandwidth": {
      "generic": "20000000000"
    }
  },
  {
    "priority": 5,
    "te-bandwidth": {
      "generic": "20000000000"
    }
  },
  {
    "priority": 6,
    "te-bandwidth": {
      "generic": "20000000000"
    }
  },
  {
    "priority": 7,
    "te-bandwidth": {
      "generic": "20000000000"
    }
  }
],
"te-default-metric": 20,
"te-srlgs": [],
"te-igp-metric": 20,
"administrative-group": "01",
```

```

        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
{
    "link-id": "NorthStar-Live:link:L127.0.0.3_10.0.0.
29_127.0.0.5_10.0.0.30:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:127.0.0.5",
        "source-tp": "NorthStar-Live:node:127.0.0.5:10.0.
0.30"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:127.0.0.3",
        "dest-tp": "NorthStar-Live:node:127.0.0.3:10.0.0.
29"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "20000000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {
                        "generic": "20000000000"
                    }
                }
            ]
        }
    }
}

```

```

        "priority": 1,
        "te-bandwidth": {
            "generic": "200000000000"
        }
    },
    {
        "priority": 2,
        "te-bandwidth": {
            "generic": "200000000000"
        }
    },
    {
        "priority": 3,
        "te-bandwidth": {
            "generic": "200000000000"
        }
    },
    {
        "priority": 4,
        "te-bandwidth": {
            "generic": "200000000000"
        }
    },
    {
        "priority": 5,
        "te-bandwidth": {
            "generic": "200000000000"
        }
    },
    {
        "priority": 6,
        "te-bandwidth": {
            "generic": "200000000000"
        }
    },
    {
        "priority": 7,
        "te-bandwidth": {
            "generic": "200000000000"
        }
    }
],
"te-default-metric": 20,
"te-srlgs": [],
"te-igp-metric": 20,
"administrative-group": "01",
"one-way-residual-bandwidth": "0x0p0",
"one-way-residual-bandwidth-normality":
"normal",
"one-way-available-bandwidth": "0x0p0",
"one-way-available-bandwidth-normality":
"normal",
"one-way-utilized-bandwidth": "0x0p0",
"one-way-utilized-bandwidth-normality":
"normal",
"one-way-min-delay-normality": "normal",
"one-way-max-delay-normality": "normal",
"one-way-delay-variation-normality": "normal",
"one-way-packet-loss-normality": "normal",
"two-way-min-delay": 0,

```

```

        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
      }
    },
    {
      "link-id": "NorthStar-Live:link:L127.0.0.4_10.0.0.
33_127.0.0.5_10.0.0.34:fwd",
      "source": {
        "source-node": "NorthStar-Live:node:127.0.0.4",
        "source-tp": "NorthStar-Live:node:127.0.0.4:10.0.
0.33"
      },
      "destination": {
        "dest-node": "NorthStar-Live:node:127.0.0.5",
        "dest-tp": "NorthStar-Live:node:127.0.0.5:10.0.0.
34"
      },
      "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
          "access-type": "point-to-point",
          "admin-status": "up",
          "link-protection-type": "unprotected",
          "max-link-bandwidth": {
            "te-bandwidth": {
              "generic": "10000000000"
            }
          },
          "unreserved-bandwidth": [
            {
              "priority": 0,
              "te-bandwidth": {
                "generic": "10000000000"
              }
            },
            {
              "priority": 1,
              "te-bandwidth": {
                "generic": "10000000000"
              }
            },
            {
              "priority": 2,
              "te-bandwidth": {
                "generic": "10000000000"
              }
            },
            {
              "priority": 3,
              "te-bandwidth": {

```

```

        "generic": "10000000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "10000000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "10000000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "10000000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "10000000000"
      }
    }
  ],
  "te-default-metric": 10,
  "te-srlgs": [],
  "te-igp-metric": 10,
  "administrative-group": "01",
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
"normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
"normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
"normal",
  "one-way-min-delay-normality": "normal",
  "one-way-max-delay-normality": "normal",
  "one-way-delay-variation-normality": "normal",
  "one-way-packet-loss-normality": "normal",
  "two-way-min-delay": 0,
  "two-way-min-delay-normality": "normal",
  "two-way-max-delay": 0,
  "two-way-max-delay-normality": "normal",
  "two-way-delay-variation": 0,
  "two-way-delay-variation-normality": "normal",
  "two-way-packet-loss": 0,
  "two-way-packet-loss-normality": "normal",
  "one-way-delay-offset": 0,
  "measure-interval": 30,
  "advertisement-interval": 0,
  "suppression-interval": 120
}
},

```

```

    {
      "link-id": "NorthStar-Live:link:L127.0.0.4_10.0.0.
33_127.0.0.5_10.0.0.34:rvs",
      "source": {
        "source-node": "NorthStar-Live:node:127.0.0.5",
        "source-tp": "NorthStar-Live:node:127.0.0.5:10.0.
0.34"
      },
      "destination": {
        "dest-node": "NorthStar-Live:node:127.0.0.4",
        "dest-tp": "NorthStar-Live:node:127.0.0.4:10.0.0.
33"
      },
      "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
          "access-type": "point-to-point",
          "admin-status": "up",
          "link-protection-type": "unprotected",
          "max-link-bandwidth": {
            "te-bandwidth": {
              "generic": "10000000000"
            }
          },
          "unreserved-bandwidth": [
            {
              "priority": 0,
              "te-bandwidth": {
                "generic": "10000000000"
              }
            },
            {
              "priority": 1,
              "te-bandwidth": {
                "generic": "10000000000"
              }
            },
            {
              "priority": 2,
              "te-bandwidth": {
                "generic": "10000000000"
              }
            },
            {
              "priority": 3,
              "te-bandwidth": {
                "generic": "10000000000"
              }
            },
            {
              "priority": 4,
              "te-bandwidth": {
                "generic": "10000000000"
              }
            },
            {
              "priority": 5,
              "te-bandwidth": {
                "generic": "10000000000"
              }
            }
          ]
        }
      }
    }

```



```

    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "100000000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "100000000000"
      }
    }
  ],
  "te-default-metric": 10,
  "te-srlgs": [],
  "te-igp-metric": 10,
  "administrative-group": "01",
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
"normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
"normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
"normal",
  "one-way-min-delay-normality": "normal",
  "one-way-max-delay-normality": "normal",
  "one-way-delay-variation-normality": "normal",
  "one-way-packet-loss-normality": "normal",
  "two-way-min-delay": 0,
  "two-way-min-delay-normality": "normal",
  "two-way-max-delay": 0,
  "two-way-max-delay-normality": "normal",
  "two-way-delay-variation": 0,
  "two-way-delay-variation-normality": "normal",
  "two-way-packet-loss": 0,
  "two-way-packet-loss-normality": "normal",
  "one-way-delay-offset": 0,
  "measure-interval": 30,
  "advertisement-interval": 0,
  "suppression-interval": 120
}
},
{
  "link-id": "NorthStar-Live:link:L127.0.0.4_10.0.0.
37_127.0.0.6_10.0.0.38:fwd",
  "source": {
    "source-node": "NorthStar-Live:node:127.0.0.4",
    "source-tp": "NorthStar-Live:node:127.0.0.4:10.0.
0.37"
  },
  "destination": {
    "dest-node": "NorthStar-Live:node:127.0.0.6",
    "dest-tp": "NorthStar-Live:node:127.0.0.6:10.0.0.
38"
  },
  "ietf-te-topology:te": {

```

```
"oper-status": "up",
"te-link-attributes": {
  "access-type": "point-to-point",
  "admin-status": "up",
  "link-protection-type": "unprotected",
  "max-link-bandwidth": {
    "te-bandwidth": {
      "generic": "20000000000"
    }
  },
  "unreserved-bandwidth": [
    {
      "priority": 0,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "200000000000"
      }
    }
  ]
},
```

```

        "te-default-metric": 20,
        "te-srlgs": [],
        "te-igp-metric": 20,
        "administrative-group": "01",
        "one-way-residual-bandwidth": "0x0p0",
        "one-way-residual-bandwidth-normality":
"normal",
        "one-way-available-bandwidth": "0x0p0",
        "one-way-available-bandwidth-normality":
"normal",
        "one-way-utilized-bandwidth": "0x0p0",
        "one-way-utilized-bandwidth-normality":
"normal",
        "one-way-min-delay-normality": "normal",
        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
    }
},
{
    "link-id": "NorthStar-Live:link:L127.0.0.4_10.0.0.
37_127.0.0.6_10.0.0.38:rvs",
    "source": {
        "source-node": "NorthStar-Live:node:127.0.0.6",
        "source-tp": "NorthStar-Live:node:127.0.0.6:10.0.
0.38"
    },
    "destination": {
        "dest-node": "NorthStar-Live:node:127.0.0.4",
        "dest-tp": "NorthStar-Live:node:127.0.0.4:10.0.0.
37"
    },
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "admin-status": "up",
            "link-protection-type": "unprotected",
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "200000000000"
                }
            },
            "unreserved-bandwidth": [
                {
                    "priority": 0,
                    "te-bandwidth": {

```

```

        "generic": "20000000000"
      }
    },
    {
      "priority": 1,
      "te-bandwidth": {
        "generic": "20000000000"
      }
    },
    {
      "priority": 2,
      "te-bandwidth": {
        "generic": "20000000000"
      }
    },
    {
      "priority": 3,
      "te-bandwidth": {
        "generic": "20000000000"
      }
    },
    {
      "priority": 4,
      "te-bandwidth": {
        "generic": "20000000000"
      }
    },
    {
      "priority": 5,
      "te-bandwidth": {
        "generic": "20000000000"
      }
    },
    {
      "priority": 6,
      "te-bandwidth": {
        "generic": "20000000000"
      }
    },
    {
      "priority": 7,
      "te-bandwidth": {
        "generic": "20000000000"
      }
    }
  ],
  "te-default-metric": 20,
  "te-srlgs": [],
  "te-igp-metric": 20,
  "administrative-group": "01",
  "one-way-residual-bandwidth": "0x0p0",
  "one-way-residual-bandwidth-normality":
"normal",
  "one-way-available-bandwidth": "0x0p0",
  "one-way-available-bandwidth-normality":
"normal",
  "one-way-utilized-bandwidth": "0x0p0",
  "one-way-utilized-bandwidth-normality":
"normal",
  "one-way-min-delay-normality": "normal",

```

```

        "one-way-max-delay-normality": "normal",
        "one-way-delay-variation-normality": "normal",
        "one-way-packet-loss-normality": "normal",
        "two-way-min-delay": 0,
        "two-way-min-delay-normality": "normal",
        "two-way-max-delay": 0,
        "two-way-max-delay-normality": "normal",
        "two-way-delay-variation": 0,
        "two-way-delay-variation-normality": "normal",
        "two-way-packet-loss": 0,
        "two-way-packet-loss-normality": "normal",
        "one-way-delay-offset": 0,
        "measure-interval": 30,
        "advertisement-interval": 0,
        "suppression-interval": 120
      }
    }
  },
  "topology-id": "NorthStar-Live",
  "network-id": "NorthStar-Live",
  "network-types": {
    "ietf-te-topology:te-topology": {
      "ietf-te-topology-packet:packet": {}
    },
    "ietf-l3-unicast-topology:l3-unicast-topology": {
      "ietf-l3-te-topology:l3-te": {},
      "ietf-sr-mpls-topology:sr-mpls": {}
    }
  },
  "node": [
    {
      "node-id": "NorthStar-Live:node:0110.1200.0010",
      "ietf-network-topology:termination-point": [
        {
          "ietf-te-topology:te-tp-id": "11.11.0.105",
          "ietf-te-topology:te": {
            "interface-switching-capability": [
              {
                "encoding": "ietf-te-types:lsp-
encoding-packet",
                "switching-capability": "ietf-te-
types:switching-psc1",
                "max-lsp-bandwidth": [
                  {
                    "priority": 0,
                    "te-bandwidth": {
                      "generic": "100000000"
                    }
                  }
                ]
              }
            ]
          }
        }
      ],
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "11.11.0.105"
        ]
      }
    }
  ],

```

```

    "tp-id": "NorthStar-Live:node:0110.1200.
0010:11.11.0.105"
    },
    {
      "ietf-te-topology:te-tp-id": "11.11.0.113",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic": "10000000"
                }
              }
            ]
          }
        ]
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "11.11.0.113"
        ]
      },
      "tp-id": "NorthStar-Live:node:0110.1200.
0010:11.11.0.113"
    },
    {
      "ietf-te-topology:te-tp-id": "11.11.0.66",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic": "10000000"
                }
              }
            ]
          }
        ]
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "11.11.0.66"
        ]
      },
      "tp-id": "NorthStar-Live:node:0110.1200.
0010:11.11.0.66"

```

```

    },
    {
      "ietf-te-topology:te-tp-id": "11.11.0.74",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic": "10000000"
                }
              }
            ]
          }
        ]
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "11.11.0.74"
        ]
      },
      "tp-id": "NorthStar-Live:node:0110.1200.
0010:11.11.0.74"
    },
    {
      "ietf-te-topology:te-tp-id": "11.11.0.78",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic": "10000000"
                }
              }
            ]
          }
        ]
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "11.11.0.78"
        ]
      },
      "tp-id": "NorthStar-Live:node:0110.1200.
0010:11.11.0.78"
    },
  },
  {

```

```

        "ietf-te-topology:te-tp-id": "11.11.0.82",
        "ietf-te-topology:te": {
            "interface-switching-capability": [
                {
                    "encoding": "ietf-te-types:lsp-
encoding-packet",
                    "switching-capability": "ietf-te-
types:switching-psc1",
                    "max-lsp-bandwidth": [
                        {
                            "priority": 0,
                            "te-bandwidth": {
                                "generic": "10000000"
                            }
                        }
                    ]
                }
            ]
        },
        "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
            "ip-address": [
                "11.11.0.82"
            ]
        },
        "tp-id": "NorthStar-Live:node:0110.1200.
0010:11.11.0.82"
    },
    "ietf-te-topology:te-node-id": "11.12.0.10",
    "ietf-te-topology:te": {
        "te-node-attributes": {
            "name": "use1-sv3-spine1-sv3"
        }
    },
    "ietf-l3-unicast-topology:l3-node-attributes": {
        "name": "use1-sv3-spine1-sv3",
        "router-id": [
            "11.12.0.10"
        ],
        "ietf-sr-mpls-topology:sr-mpls": {
            "": [
                {
                    "lower-bound": 8000,
                    "upper-bound": 9000
                }
            ]
        }
    }
},
{
    "node-id": "NorthStar-Live:node:0110.1200.0011",
    "ietf-network-topology:termination-point": [
        {
            "ietf-te-topology:te-tp-id": "11.11.0.81",
            "ietf-te-topology:te": {
                "interface-switching-capability": [
                    {
                        "encoding": "ietf-te-types:lsp-
encoding-packet",

```



```

types:switching-psc1",
                                "switching-capability": "ietf-te-
                                "max-lsp-bandwidth": [
                                    {
                                        "priority": 0,
                                        "te-bandwidth": {
                                            "generic": "10000000"
                                        }
                                    }
                                ]
                            },
                            "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                                "ip-address": [
                                    "11.11.0.81"
                                ]
                            },
                            "tp-id": "NorthStar-Live:node:0110.1200.
0011:11.11.0.81"
                        },
                        {
                            "ietf-te-topology:te-tp-id": "11.11.0.85",
                            "ietf-te-topology:te": {
                                "interface-switching-capability": [
                                    {
                                        "encoding": "ietf-te-types:lsp-
encoding-packet",
                                        "switching-capability": "ietf-te-
types:switching-psc1",
                                        "max-lsp-bandwidth": [
                                            {
                                                "priority": 0,
                                                "te-bandwidth": {
                                                    "generic": "10000000"
                                                }
                                            }
                                        ]
                                    }
                                ]
                            },
                            "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                                "ip-address": [
                                    "11.11.0.85"
                                ]
                            },
                            "tp-id": "NorthStar-Live:node:0110.1200.
0011:11.11.0.85"
                        }
                    ],
                    "ietf-te-topology:te-node-id": "11.12.0.11",
                    "ietf-te-topology:te": {
                        "te-node-attributes": {
                            "name": "use2-sv3"
                        }
                    },
                    "ietf-l3-unicast-topology:l3-node-attributes": {
                        "name": "use2-sv3",

```

```

        "router-id": [
            "11.12.0.11"
        ],
        "ietf-sr-mpls-topology:sr-mpls": {
            "": [
                {
                    "lower-bound": 8000,
                    "upper-bound": 9000
                }
            ],
            "msd": 5
        }
    },
    {
        "node-id": "NorthStar-Live:node:0110.1200.0012",
        "ietf-network-topology:termination-point": [
            {
                "ietf-te-topology:te-tp-id": "11.11.0.86",
                "ietf-te-topology:te": {
                    "interface-switching-capability": [
                        {
                            "encoding": "ietf-te-types:lsp-
encoding-packet",
                            "switching-capability": "ietf-te-
types:switching-psc1",
                            "max-lsp-bandwidth": [
                                {
                                    "priority": 0,
                                    "te-bandwidth": {
                                        "generic": "100000000"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                    "ip-address": [
                        "11.11.0.86"
                    ]
                },
                "tp-id": "NorthStar-Live:node:0110.1200.
0012:11.11.0.86"
            },
            {
                "ietf-te-topology:te-tp-id": "11.11.0.129",
                "ietf-te-topology:te": {
                    "interface-switching-capability": [
                        {
                            "encoding": "ietf-te-types:lsp-
encoding-packet",
                            "switching-capability": "ietf-te-
types:switching-psc1",
                            "max-lsp-bandwidth": [
                                {
                                    "priority": 0,
                                    "te-bandwidth": {
                                        "generic": "100000000"
                                    }
                                }
                            ]
                        }
                    ]
                }
            }
        ]
    }
]

```

```

    }
  }
]
},
"ietf-l3-unicast-topology:l3-termination-
  "ip-address": [
    "11.11.0.129"
  ],
  "tp-id": "NorthStar-Live:node:0110.1200.
0012:11.11.0.129"
},
{
  "ietf-te-topology:te-tp-id": "11.11.0.137",
  "ietf-te-topology:te": {
    "interface-switching-capability": [
      {
        "encoding": "ietf-te-types:lsp-
encoding-packet",
        "switching-capability": "ietf-te-
types:switching-psc1",
        "max-lsp-bandwidth": [
          {
            "priority": 0,
            "te-bandwidth": {
              "generic": "10000000"
            }
          }
        ]
      }
    ]
  },
  "ietf-l3-unicast-topology:l3-termination-
    "ip-address": [
      "11.11.0.137"
    ],
    "tp-id": "NorthStar-Live:node:0110.1200.
0012:11.11.0.137"
  },
  {
    "ietf-te-topology:te-tp-id": "11.11.0.70",
    "ietf-te-topology:te": {
      "interface-switching-capability": [
        {
          "encoding": "ietf-te-types:lsp-
encoding-packet",
          "switching-capability": "ietf-te-
types:switching-psc1",
          "max-lsp-bandwidth": [
            {
              "priority": 0,
              "te-bandwidth": {
                "generic": "10000000"
              }
            }
          ]
        }
      ]
    }
  }
}

```

```

    ]
  }
]
},
"ietf-l3-unicast-topology:l3-termination-
  "ip-address": [
    "11.11.0.70"
  ]
},
"tp-id": "NorthStar-Live:node:0110.1200.
0012:11.11.0.70"
},
{
  "ietf-te-topology:te-tp-id": "11.11.0.90",
  "ietf-te-topology:te": {
    "interface-switching-capability": [
      {
        "encoding": "ietf-te-types:lsp-
encoding-packet",
        "switching-capability": "ietf-te-
types:switching-psc1",
        "max-lsp-bandwidth": [
          {
            "priority": 0,
            "te-bandwidth": {
              "generic": "10000000"
            }
          }
        ]
      }
    ]
  },
  "ietf-l3-unicast-topology:l3-termination-
    "ip-address": [
      "11.11.0.90"
    ]
  },
  "tp-id": "NorthStar-Live:node:0110.1200.
0012:11.11.0.90"
},
{
  "ietf-te-topology:te-tp-id": "11.11.0.94",
  "ietf-te-topology:te": {
    "interface-switching-capability": [
      {
        "encoding": "ietf-te-types:lsp-
encoding-packet",
        "switching-capability": "ietf-te-
types:switching-psc1",
        "max-lsp-bandwidth": [
          {
            "priority": 0,
            "te-bandwidth": {
              "generic": "10000000"
            }
          }
        ]
      }
    ]
  }
}
}

```

```

    ],
    },
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
            "11.11.0.94"
        ]
    },
    "tp-id": "NorthStar-Live:node:0110.1200.
0012:11.11.0.94"
    }
],
"ietf-te-topology:te-node-id": "11.12.0.12",
"ietf-te-topology:te": {
    "te-node-attributes": {
        "name": "use2-sv3-spine2-sv3"
    }
},
"ietf-l3-unicast-topology:l3-node-attributes": {
    "name": "use2-sv3-spine2-sv3",
    "router-id": [
        "11.12.0.12"
    ],
    "ietf-sr-mpls-topology:sr-mpls": {
        "": [
            {
                "lower-bound": 8000,
                "upper-bound": 9000
            }
        ]
    }
}
},
{
    "node-id": "NorthStar-Live:node:0110.1200.0013",
    "ietf-network-topology:termination-point": [
        {
            "ietf-te-topology:te-tp-id": "11.11.0.101",
            "ietf-te-topology:te": {
                "interface-switching-capability": [
                    {
                        "encoding": "ietf-te-types:lsp-
encoding-packet",
                        "switching-capability": "ietf-te-
types:switching-psc1",
                        "max-lsp-bandwidth": [
                            {
                                "priority": 0,
                                "te-bandwidth": {
                                    "generic": "100000000"
                                }
                            }
                        ]
                    }
                ]
            }
        }
    ],
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
            "11.11.0.101"
        ]
    }
}
]
},

```

```

    ],
    "tp-id": "NorthStar-Live:node:0110.1200.
0013:11.11.0.101"
  },
  {
    "ietf-te-topology:te-tp-id": "11.11.0.97",
    "ietf-te-topology:te": {
      "interface-switching-capability": [
        {
          "encoding": "ietf-te-types:lsp-
encoding-packet",
          "switching-capability": "ietf-te-
types:switching-psc1",
          "max-lsp-bandwidth": [
            {
              "priority": 0,
              "te-bandwidth": {
                "generic": "10000000"
              }
            }
          ]
        }
      ]
    },
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
      "ip-address": [
        "11.11.0.97"
      ],
      "tp-id": "NorthStar-Live:node:0110.1200.
0013:11.11.0.97"
    }
  ],
  "ietf-te-topology:te-node-id": "11.12.0.13",
  "ietf-te-topology:te": {
    "te-node-attributes": {
      "name": "use1-sv4"
    }
  },
  "ietf-l3-unicast-topology:l3-node-attributes": {
    "name": "use1-sv4",
    "router-id": [
      "11.12.0.13"
    ],
    "ietf-sr-mpls-topology:sr-mpls": {
      "": [
        {
          "lower-bound": 8000,
          "upper-bound": 9000
        }
      ],
      "msd": 5
    }
  }
},
{
  "node-id": "NorthStar-Live:node:0110.1200.0001",
  "ietf-network-topology:termination-point": [

```

```

    {
      "ietf-te-topology:te-tp-id": "11.11.0.1",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic": "10000000"
                }
              }
            ]
          }
        ],
        "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
          "ip-address": [
            "11.11.0.1"
          ],
          "tp-id": "NorthStar-Live:node:0110.1200.
0001:11.11.0.1"
        },
        {
          "ietf-te-topology:te-tp-id": "11.11.0.5",
          "ietf-te-topology:te": {
            "interface-switching-capability": [
              {
                "encoding": "ietf-te-types:lsp-
encoding-packet",
                "switching-capability": "ietf-te-
types:switching-psc1",
                "max-lsp-bandwidth": [
                  {
                    "priority": 0,
                    "te-bandwidth": {
                      "generic": "10000000"
                    }
                  }
                ]
              }
            ],
            "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
              "ip-address": [
                "11.11.0.5"
              ],
              "tp-id": "NorthStar-Live:node:0110.1200.
0001:11.11.0.5"
            },
            }
          ],
          "ietf-te-topology:te-node-id": "11.12.0.1",

```

```

        "ietf-te-topology:te": {
            "te-node-attributes": {
                "name": "use1-sv1"
            }
        },
        "ietf-l3-unicast-topology:l3-node-attributes": {
            "name": "use1-sv1",
            "router-id": [
                "11.12.0.1"
            ],
            "ietf-sr-mpls-topology:sr-mpls": {
                "": [
                    {
                        "lower-bound": 8000,
                        "upper-bound": 9000
                    }
                ],
                "msd": 5
            }
        }
    },
    {
        "node-id": "NorthStar-Live:node:0110.1200.0014",
        "ietf-network-topology:termination-point": [
            {
                "ietf-te-topology:te-tp-id": "11.11.0.106",
                "ietf-te-topology:te": {
                    "interface-switching-capability": [
                        {
                            "encoding": "ietf-te-types:lsp-
encoding-packet",
                            "switching-capability": "ietf-te-
types:switching-psc1",
                            "max-lsp-bandwidth": [
                                {
                                    "priority": 0,
                                    "te-bandwidth": {
                                        "generic": "10000000"
                                    }
                                }
                            ]
                        }
                    ]
                }
            },
            "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                "ip-address": [
                    "11.11.0.106"
                ],
                "tp-id": "NorthStar-Live:node:0110.1200.
0014:11.11.0.106"
            },
            {
                "ietf-te-topology:te-tp-id": "11.11.0.114",
                "ietf-te-topology:te": {
                    "interface-switching-capability": [
                        {
                            "encoding": "ietf-te-types:lsp-
encoding-packet",

```



```

types:switching-psc1",
                                "switching-capability": "ietf-te-
                                "max-lsp-bandwidth": [
                                    {
                                        "priority": 0,
                                        "te-bandwidth": {
                                            "generic": "10000000"
                                        }
                                    }
                                ]
                            },
                            "ietf-l3-unicast-topology:l3-termination-
                                "ip-address": [
                                    "11.11.0.114"
                                ]
                            },
                            "tp-id": "NorthStar-Live:node:0110.1200.
0014:11.11.0.114"
                        },
                        {
                            "ietf-te-topology:te-tp-id": "11.11.0.98",
                            "ietf-te-topology:te": {
                                "interface-switching-capability": [
                                    {
                                        "encoding": "ietf-te-types:lsp-
encoding-packet",
                                        "switching-capability": "ietf-te-
types:switching-psc1",
                                        "max-lsp-bandwidth": [
                                            {
                                                "priority": 0,
                                                "te-bandwidth": {
                                                    "generic": "10000000"
                                                }
                                            }
                                        ]
                                    }
                                ]
                            },
                            "ietf-l3-unicast-topology:l3-termination-
                                "ip-address": [
                                    "11.11.0.98"
                                ]
                            },
                            "tp-id": "NorthStar-Live:node:0110.1200.
0014:11.11.0.98"
                        },
                        {
                            "ietf-te-topology:te-tp-id": "11.11.0.110",
                            "ietf-te-topology:te": {
                                "interface-switching-capability": [
                                    {
                                        "encoding": "ietf-te-types:lsp-
encoding-packet",
                                        "switching-capability": "ietf-te-
types:switching-psc1",

```

```

                                "max-lsp-bandwidth": [
                                    {
                                        "priority": 0,
                                        "te-bandwidth": {
                                            "generic": "100000000"
                                        }
                                    }
                                ]
                            }
                        ]
                    },
                    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                        "ip-address": [
                            "11.11.0.110"
                        ]
                    },
                    "tp-id": "NorthStar-Live:node:0110.1200.
0014:11.11.0.110"
                },
                {
                    "ietf-te-topology:te-tp-id": "11.11.0.118",
                    "ietf-te-topology:te": {
                        "interface-switching-capability": [
                            {
                                "encoding": "ietf-te-types:lsp-
encoding-packet",
                                "switching-capability": "ietf-te-
types:switching-psc1",
                                "max-lsp-bandwidth": [
                                    {
                                        "priority": 0,
                                        "te-bandwidth": {
                                            "generic": "100000000"
                                        }
                                    }
                                ]
                            }
                        ]
                    },
                    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                        "ip-address": [
                            "11.11.0.118"
                        ]
                    },
                    "tp-id": "NorthStar-Live:node:0110.1200.
0014:11.11.0.118"
                },
                {
                    "ietf-te-topology:te-tp-id": "11.11.0.122",
                    "ietf-te-topology:te": {
                        "interface-switching-capability": [
                            {
                                "encoding": "ietf-te-types:lsp-
encoding-packet",
                                "switching-capability": "ietf-te-
types:switching-psc1",
                                "max-lsp-bandwidth": [
                                    {

```

```

        "priority": 0,
        "te-bandwidth": {
            "generic": "100000000"
        }
    }
}
],
},
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
            "11.11.0.122"
        ]
    },
    "tp-id": "NorthStar-Live:node:0110.1200.
0014:11.11.0.122"
},
{
    "ietf-te-topology:te-tp-id": "11.11.0.13",
    "ietf-te-topology:te": {
        "interface-switching-capability": [
            {
                "encoding": "ietf-te-types:lsp-
encoding-packet",
                "switching-capability": "ietf-te-
types:switching-psc1",
                "max-lsp-bandwidth": [
                    {
                        "priority": 0,
                        "te-bandwidth": {
                            "generic": "100000000"
                        }
                    }
                ]
            }
        ]
    },
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
            "11.11.0.13"
        ]
    },
    "tp-id": "NorthStar-Live:node:0110.1200.
0014:11.11.0.13"
},
{
    "ietf-te-topology:te-tp-id": "11.11.0.9",
    "ietf-te-topology:te": {
        "interface-switching-capability": [
            {
                "encoding": "ietf-te-types:lsp-
encoding-packet",
                "switching-capability": "ietf-te-
types:switching-psc1",
                "max-lsp-bandwidth": [
                    {
                        "priority": 0,
                        "te-bandwidth": {

```

```

    "generic": "10000000"
  }
}
]
},
"ietf-l3-unicast-topology:l3-termination-
point-attributes": {
  "ip-address": [
    "11.11.0.9"
  ],
  "tp-id": "NorthStar-Live:node:0110.1200.
0014:11.11.0.9"
},
],
"ietf-te-topology:te-node-id": "11.12.0.14",
"ietf-te-topology:te": {
  "te-node-attributes": {
    "name": "use1-sv4-spine1-sv4"
  }
},
"ietf-l3-unicast-topology:l3-node-attributes": {
  "name": "use1-sv4-spine1-sv4",
  "router-id": [
    "11.12.0.14"
  ],
  "ietf-sr-mpls-topology:sr-mpls": {
    "": [
      {
        "lower-bound": 8000,
        "upper-bound": 9000
      }
    ]
  }
},
{
  "node-id": "NorthStar-Live:node:0110.1200.0015",
  "ietf-network-topology:termination-point": [
    {
      "ietf-te-topology:te-tp-id": "11.11.0.121",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic": "10000000"
                }
              }
            ]
          }
        ]
      }
    ]
  }
}
]

```

```

    },
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
            "11.11.0.121"
        ]
    },
    "tp-id": "NorthStar-Live:node:0110.1200.
0015:11.11.0.121"
},
{
    "ietf-te-topology:te-tp-id": "11.11.0.125",
    "ietf-te-topology:te": {
        "interface-switching-capability": [
            {
                "encoding": "ietf-te-types:lsp-
encoding-packet",
                "switching-capability": "ietf-te-
types:switching-psc1",
                "max-lsp-bandwidth": [
                    {
                        "priority": 0,
                        "te-bandwidth": {
                            "generic": "10000000"
                        }
                    }
                ]
            }
        ]
    },
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
            "11.11.0.125"
        ]
    },
    "tp-id": "NorthStar-Live:node:0110.1200.
0015:11.11.0.125"
},
    ],
    "ietf-te-topology:te-node-id": "11.12.0.15",
    "ietf-te-topology:te": {
        "te-node-attributes": {
            "name": "use2-sv4"
        }
    },
    "ietf-l3-unicast-topology:l3-node-attributes": {
        "name": "use2-sv4",
        "router-id": [
            "11.12.0.15"
        ],
        "ietf-sr-mpls-topology:sr-mpls": {
            "": [
                {
                    "lower-bound": 8000,
                    "upper-bound": 9000
                }
            ],
            "msd": 5
        }
    }
}

```

```

    }
  },
  {
    "node-id": "NorthStar-Live:node:0110.1200.0016",
    "ietf-network-topology:termination-point": [
      {
        "ietf-te-topology:te-tp-id": "11.11.0.130",
        "ietf-te-topology:te": {
          "interface-switching-capability": [
            {
              "encoding": "ietf-te-types:lsp-
encoding-packet",
              "switching-capability": "ietf-te-
types:switching-psc1",
              "max-lsp-bandwidth": [
                {
                  "priority": 0,
                  "te-bandwidth": {
                    "generic": "100000000"
                  }
                }
              ]
            }
          ]
        },
        "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
          "ip-address": [
            "11.11.0.130"
          ],
          "tp-id": "NorthStar-Live:node:0110.1200.
0016:11.11.0.130"
        }
      },
      {
        "ietf-te-topology:te-tp-id": "11.11.0.138",
        "ietf-te-topology:te": {
          "interface-switching-capability": [
            {
              "encoding": "ietf-te-types:lsp-
encoding-packet",
              "switching-capability": "ietf-te-
types:switching-psc1",
              "max-lsp-bandwidth": [
                {
                  "priority": 0,
                  "te-bandwidth": {
                    "generic": "100000000"
                  }
                }
              ]
            }
          ]
        },
        "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
          "ip-address": [
            "11.11.0.138"
          ],
        },
      }
    ]
  }
]

```

```

    "tp-id": "NorthStar-Live:node:0110.1200.
0016:11.11.0.138"
    },
    {
      "ietf-te-topology:te-tp-id": "11.11.0.102",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic": "10000000"
                }
              }
            ]
          }
        ]
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "11.11.0.102"
        ]
      },
      "tp-id": "NorthStar-Live:node:0110.1200.
0016:11.11.0.102"
    },
    {
      "ietf-te-topology:te-tp-id": "11.11.0.126",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic": "10000000"
                }
              }
            ]
          }
        ]
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "11.11.0.126"
        ]
      },
      "tp-id": "NorthStar-Live:node:0110.1200.
0016:11.11.0.126"

```

```

    },
    {
      "ietf-te-topology:te-tp-id": "11.11.0.134",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic": "100000000"
                }
              }
            ]
          }
        ]
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "11.11.0.134"
        ]
      },
      "tp-id": "NorthStar-Live:node:0110.1200.
0016:11.11.0.134"
    },
    {
      "ietf-te-topology:te-tp-id": "11.11.0.142",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic": "100000000"
                }
              }
            ]
          }
        ]
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "11.11.0.142"
        ]
      },
      "tp-id": "NorthStar-Live:node:0110.1200.
0016:11.11.0.142"
    },
  },
  {

```



```

        "ietf-te-topology:te-tp-id": "11.11.0.25",
        "ietf-te-topology:te": {
            "interface-switching-capability": [
                {
                    "encoding": "ietf-te-types:lsp-
encoding-packet",
                    "switching-capability": "ietf-te-
types:switching-psc1",
                    "max-lsp-bandwidth": [
                        {
                            "priority": 0,
                            "te-bandwidth": {
                                "generic": "10000000"
                            }
                        }
                    ]
                }
            ]
        },
        "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
            "ip-address": [
                "11.11.0.25"
            ]
        },
        "tp-id": "NorthStar-Live:node:0110.1200.
0016:11.11.0.25"
    },
    {
        "ietf-te-topology:te-tp-id": "11.11.0.29",
        "ietf-te-topology:te": {
            "interface-switching-capability": [
                {
                    "encoding": "ietf-te-types:lsp-
encoding-packet",
                    "switching-capability": "ietf-te-
types:switching-psc1",
                    "max-lsp-bandwidth": [
                        {
                            "priority": 0,
                            "te-bandwidth": {
                                "generic": "10000000"
                            }
                        }
                    ]
                }
            ]
        },
        "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
            "ip-address": [
                "11.11.0.29"
            ]
        },
        "tp-id": "NorthStar-Live:node:0110.1200.
0016:11.11.0.29"
    }
],
"ietf-te-topology:te-node-id": "11.12.0.16",
"ietf-te-topology:te": {

```

```

        "te-node-attributes": {
            "name": "use2-sv4-spine2-sv4"
        },
        "ietf-l3-unicast-topology:l3-node-attributes": {
            "name": "use2-sv4-spine2-sv4",
            "router-id": [
                "11.12.0.16"
            ],
            "ietf-sr-mpls-topology:sr-mpls": {
                "": [
                    {
                        "lower-bound": 8000,
                        "upper-bound": 9000
                    }
                ]
            }
        },
        {
            "node-id": "NorthStar-Live:node:0110.1200.0002",
            "ietf-network-topology:termination-point": [
                {
                    "ietf-te-topology:te-tp-id": "11.11.0.2",
                    "ietf-te-topology:te": {
                        "interface-switching-capability": [
                            {
                                "encoding": "ietf-te-types:lsp-
encoding-packet",
                                "switching-capability": "ietf-te-
types:switching-psc1",
                                "max-lsp-bandwidth": [
                                    {
                                        "priority": 0,
                                        "te-bandwidth": {
                                            "generic": "10000000"
                                        }
                                    }
                                ]
                            }
                        ]
                    },
                    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                        "ip-address": [
                            "11.11.0.2"
                        ],
                        "tp-id": "NorthStar-Live:node:0110.1200.
0002:11.11.0.2"
                    },
                    {
                        "ietf-te-topology:te-tp-id": "11.11.0.14",
                        "ietf-te-topology:te": {
                            "interface-switching-capability": [
                                {
                                    "encoding": "ietf-te-types:lsp-
encoding-packet",
                                    "switching-capability": "ietf-te-
types:switching-psc1",

```

```

                                "max-lsp-bandwidth": [
                                    {
                                        "priority": 0,
                                        "te-bandwidth": {
                                            "generic": "100000000"
                                        }
                                    }
                                ]
                            }
                        ]
                    },
                    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                        "ip-address": [
                            "11.11.0.14"
                        ]
                    },
                    "tp-id": "NorthStar-Live:node:0110.1200.
0002:11.11.0.14"
                },
                {
                    "ietf-te-topology:te-tp-id": "11.11.0.10",
                    "ietf-te-topology:te": {
                        "interface-switching-capability": [
                            {
                                "encoding": "ietf-te-types:lsp-
encoding-packet",
                                "switching-capability": "ietf-te-
types:switching-psc1",
                                "max-lsp-bandwidth": [
                                    {
                                        "priority": 0,
                                        "te-bandwidth": {
                                            "generic": "100000000"
                                        }
                                    }
                                ]
                            }
                        ]
                    },
                    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                        "ip-address": [
                            "11.11.0.10"
                        ]
                    },
                    "tp-id": "NorthStar-Live:node:0110.1200.
0002:11.11.0.10"
                },
                {
                    "ietf-te-topology:te-tp-id": "11.11.0.18",
                    "ietf-te-topology:te": {
                        "interface-switching-capability": [
                            {
                                "encoding": "ietf-te-types:lsp-
encoding-packet",
                                "switching-capability": "ietf-te-
types:switching-psc1",
                                "max-lsp-bandwidth": [
                                    {

```

```

        "priority": 0,
        "te-bandwidth": {
            "generic": "100000000"
        }
    }
}
]
},
"ietf-l3-unicast-topology:l3-termination-
point-attributes": {
    "ip-address": [
        "11.11.0.18"
    ]
},
"tp-id": "NorthStar-Live:node:0110.1200.
0002:11.11.0.18"
},
{
    "ietf-te-topology:te-tp-id": "11.11.0.41",
    "ietf-te-topology:te": {
        "interface-switching-capability": [
            {
                "encoding": "ietf-te-types:lsp-
encoding-packet",
                "switching-capability": "ietf-te-
types:switching-psc1",
                "max-lsp-bandwidth": [
                    {
                        "priority": 0,
                        "te-bandwidth": {
                            "generic": "100000000"
                        }
                    }
                ]
            }
        ]
    },
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
            "11.11.0.41"
        ]
    },
    "tp-id": "NorthStar-Live:node:0110.1200.
0002:11.11.0.41"
},
{
    "ietf-te-topology:te-tp-id": "11.11.0.45",
    "ietf-te-topology:te": {
        "interface-switching-capability": [
            {
                "encoding": "ietf-te-types:lsp-
encoding-packet",
                "switching-capability": "ietf-te-
types:switching-psc1",
                "max-lsp-bandwidth": [
                    {
                        "priority": 0,
                        "te-bandwidth": {

```

```

    "generic": "10000000"
  }
}
]
}
],
},
"ietf-l3-unicast-topology:l3-termination-
point-attributes": {
  "ip-address": [
    "11.11.0.45"
  ],
  "tp-id": "NorthStar-Live:node:0110.1200.
0002:11.11.0.45"
},
],
"ietf-te-topology:te-node-id": "11.12.0.2",
"ietf-te-topology:te": {
  "te-node-attributes": {
    "name": "use1-sv1-spine1-sv1"
  }
},
"ietf-l3-unicast-topology:l3-node-attributes": {
  "name": "use1-sv1-spine1-sv1",
  "router-id": [
    "11.12.0.2"
  ],
  "ietf-sr-mpls-topology:sr-mpls": {
    "": [
      {
        "lower-bound": 8000,
        "upper-bound": 9000
      }
    ]
  }
},
{
  "node-id": "NorthStar-Live:node:0110.1200.0003",
  "ietf-network-topology:termination-point": [
    {
      "ietf-te-topology:te-tp-id": "11.11.0.17",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic": "10000000"
                }
              }
            ]
          }
        ]
      }
    ]
  }
}
]

```

```

    },
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
            "11.11.0.17"
        ]
    },
    "tp-id": "NorthStar-Live:node:0110.1200.
0003:11.11.0.17"
},
{
    "ietf-te-topology:te-tp-id": "11.11.0.21",
    "ietf-te-topology:te": {
        "interface-switching-capability": [
            {
                "encoding": "ietf-te-types:lsp-
encoding-packet",
                "switching-capability": "ietf-te-
types:switching-psc1",
                "max-lsp-bandwidth": [
                    {
                        "priority": 0,
                        "te-bandwidth": {
                            "generic": "10000000"
                        }
                    }
                ]
            }
        ]
    },
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
            "11.11.0.21"
        ]
    },
    "tp-id": "NorthStar-Live:node:0110.1200.
0003:11.11.0.21"
    }
],
"ietf-te-topology:te-node-id": "11.12.0.3",
"ietf-te-topology:te": {
    "te-node-attributes": {
        "name": "use2-sv1"
    }
},
"ietf-l3-unicast-topology:l3-node-attributes": {
    "name": "use2-sv1",
    "router-id": [
        "11.12.0.3"
    ],
    "ietf-sr-mpls-topology:sr-mpls": {
        "": [
            {
                "lower-bound": 8000,
                "upper-bound": 9000
            }
        ],
        "msd": 5
    }
}

```

```

    },
    {
      "node-id": "NorthStar-Live:node:0110.1200.0004",
      "ietf-network-topology:termination-point": [
        {
          "ietf-te-topology:te-tp-id": "11.11.0.6",
          "ietf-te-topology:te": {
            "interface-switching-capability": [
              {
                "encoding": "ietf-te-types:lsp-
encoding-packet",
                "switching-capability": "ietf-te-
types:switching-psc1",
                "max-lsp-bandwidth": [
                  {
                    "priority": 0,
                    "te-bandwidth": {
                      "generic": "10000000"
                    }
                  }
                ]
              }
            ]
          },
          "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
            "ip-address": [
              "11.11.0.6"
            ],
            "tp-id": "NorthStar-Live:node:0110.1200.
0004:11.11.0.6"
          }
        },
        {
          "ietf-te-topology:te-tp-id": "11.11.0.26",
          "ietf-te-topology:te": {
            "interface-switching-capability": [
              {
                "encoding": "ietf-te-types:lsp-
encoding-packet",
                "switching-capability": "ietf-te-
types:switching-psc1",
                "max-lsp-bandwidth": [
                  {
                    "priority": 0,
                    "te-bandwidth": {
                      "generic": "10000000"
                    }
                  }
                ]
              }
            ]
          },
          "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
            "ip-address": [
              "11.11.0.26"
            ],

```

```

"tp-id": "NorthStar-Live:node:0110.1200.
0004:11.11.0.26"
    },
    {
      "ietf-te-topology:te-tp-id": "11.11.0.30",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic": "10000000"
                }
              }
            ]
          }
        ],
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "11.11.0.30"
        ],
      },
      "tp-id": "NorthStar-Live:node:0110.1200.
0004:11.11.0.30"
    },
    {
      "ietf-te-topology:te-tp-id": "11.11.0.22",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic": "10000000"
                }
              }
            ]
          }
        ],
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "11.11.0.22"
        ],
      },
      "tp-id": "NorthStar-Live:node:0110.1200.
0004:11.11.0.22"
    }
  ]
}

```



```

    },
    {
      "ietf-te-topology:te-tp-id": "11.11.0.57",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic": "10000000"
                }
              }
            ]
          }
        ]
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "11.11.0.57"
        ]
      },
      "tp-id": "NorthStar-Live:node:0110.1200.
0004:11.11.0.57"
    },
    {
      "ietf-te-topology:te-tp-id": "11.11.0.61",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic": "10000000"
                }
              }
            ]
          }
        ]
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "11.11.0.61"
        ]
      },
      "tp-id": "NorthStar-Live:node:0110.1200.
0004:11.11.0.61"
    }
  ],

```

```

        "ietf-te-topology:te-node-id": "11.12.0.4",
        "ietf-te-topology:te": {
            "te-node-attributes": {
                "name": "use2-sv1-spine2-sv1"
            }
        },
        "ietf-l3-unicast-topology:l3-node-attributes": {
            "name": "use2-sv1-spine2-sv1",
            "router-id": [
                "11.12.0.4"
            ],
            "ietf-sr-mpls-topology:sr-mpls": {
                "": [
                    {
                        "lower-bound": 8000,
                        "upper-bound": 9000
                    }
                ]
            }
        }
    },
    {
        "node-id": "NorthStar-Live:node:0110.1200.0005",
        "ietf-network-topology:termination-point": [
            {
                "ietf-te-topology:te-tp-id": "11.11.0.33",
                "ietf-te-topology:te": {
                    "interface-switching-capability": [
                        {
                            "encoding": "ietf-te-types:lsp-
encoding-packet",
                            "switching-capability": "ietf-te-
types:switching-psc1",
                            "max-lsp-bandwidth": [
                                {
                                    "priority": 0,
                                    "te-bandwidth": {
                                        "generic": "10000000"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                    "ip-address": [
                        "11.11.0.33"
                    ],
                    "tp-id": "NorthStar-Live:node:0110.1200.
0005:11.11.0.33"
                },
                {
                    "ietf-te-topology:te-tp-id": "11.11.0.37",
                    "ietf-te-topology:te": {
                        "interface-switching-capability": [
                            {
                                "encoding": "ietf-te-types:lsp-
encoding-packet",

```

```

types:switching-psc1",
                                "switching-capability": "ietf-te-
                                "max-lsp-bandwidth": [
                                    {
                                        "priority": 0,
                                        "te-bandwidth": {
                                            "generic": "100000000"
                                        }
                                    }
                                ]
                            },
                            "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                                "ip-address": [
                                    "11.11.0.37"
                                ]
                            },
                            "tp-id": "NorthStar-Live:node:0110.1200.
0005:11.11.0.37"
                        },
                    ],
                    "ietf-te-topology:te-node-id": "11.12.0.5",
                    "ietf-te-topology:te": {
                        "te-node-attributes": {
                            "name": "use1-sv2"
                        }
                    },
                    "ietf-l3-unicast-topology:l3-node-attributes": {
                        "name": "use1-sv2",
                        "router-id": [
                            "11.12.0.5"
                        ],
                        "ietf-sr-mpls-topology:sr-mpls": {
                            "": [
                                {
                                    "lower-bound": 8000,
                                    "upper-bound": 9000
                                }
                            ],
                            "msd": 5
                        }
                    }
                },
                {
                    "node-id": "NorthStar-Live:node:0110.1200.0006",
                    "ietf-network-topology:termination-point": [
                        {
                            "ietf-te-topology:te-tp-id": "11.11.0.73",
                            "ietf-te-topology:te": {
                                "interface-switching-capability": [
                                    {
                                        "encoding": "ietf-te-types:lsp-
encoding-packet",
                                        "switching-capability": "ietf-te-
types:switching-psc1",
                                        "max-lsp-bandwidth": [
                                            {
                                                "priority": 0,

```

```

        "te-bandwidth": {
            "generic": "100000000"
        }
    }
}
    ],
    },
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
            "11.11.0.73"
        ],
        },
        "tp-id": "NorthStar-Live:node:0110.1200.
0006:11.11.0.73"
    },
    {
        "ietf-te-topology:te-tp-id": "11.11.0.77",
        "ietf-te-topology:te": {
            "interface-switching-capability": [
                {
                    "encoding": "ietf-te-types:lsp-
encoding-packet",
                    "switching-capability": "ietf-te-
types:switching-psc1",
                    "max-lsp-bandwidth": [
                        {
                            "priority": 0,
                            "te-bandwidth": {
                                "generic": "100000000"
                            }
                        }
                    ]
                }
            ]
        },
        "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
            "ip-address": [
                "11.11.0.77"
            ],
            },
            "tp-id": "NorthStar-Live:node:0110.1200.
0006:11.11.0.77"
        },
        {
            "ietf-te-topology:te-tp-id": "11.11.0.109",
            "ietf-te-topology:te": {
                "interface-switching-capability": [
                    {
                        "encoding": "ietf-te-types:lsp-
encoding-packet",
                        "switching-capability": "ietf-te-
types:switching-psc1",
                        "max-lsp-bandwidth": [
                            {
                                "priority": 0,
                                "te-bandwidth": {
                                    "generic": "100000000"
                                }
                            }
                        ]
                    }
                ]
            }
        }
    ]
}

```

```

    }
  }
]
},
"ietf-l3-unicast-topology:l3-termination-
  "ip-address": [
    "11.11.0.109"
  ],
  "tp-id": "NorthStar-Live:node:0110.1200.
0006:11.11.0.109"
},
{
  "ietf-te-topology:te-tp-id": "11.11.0.117",
  "ietf-te-topology:te": {
    "interface-switching-capability": [
      {
        "encoding": "ietf-te-types:lsp-
encoding-packet",
        "switching-capability": "ietf-te-
types:switching-psc1",
        "max-lsp-bandwidth": [
          {
            "priority": 0,
            "te-bandwidth": {
              "generic": "100000000"
            }
          }
        ]
      }
    ]
  },
  "ietf-l3-unicast-topology:l3-termination-
    "ip-address": [
      "11.11.0.117"
    ],
    "tp-id": "NorthStar-Live:node:0110.1200.
0006:11.11.0.117"
  },
  {
    "ietf-te-topology:te-tp-id": "11.11.0.42",
    "ietf-te-topology:te": {
      "interface-switching-capability": [
        {
          "encoding": "ietf-te-types:lsp-
encoding-packet",
          "switching-capability": "ietf-te-
types:switching-psc1",
          "max-lsp-bandwidth": [
            {
              "priority": 0,
              "te-bandwidth": {
                "generic": "100000000"
              }
            }
          ]
        }
      ]
    }
  }
}

```

```

    ]
  }
]
},
"ietf-l3-unicast-topology:l3-termination-
  "ip-address": [
    "11.11.0.42"
  ]
},
"tp-id": "NorthStar-Live:node:0110.1200.
0006:11.11.0.42"
},
{
  "ietf-te-topology:te-tp-id": "11.11.0.46",
  "ietf-te-topology:te": {
    "interface-switching-capability": [
      {
        "encoding": "ietf-te-types:lsp-
encoding-packet",
        "switching-capability": "ietf-te-
types:switching-psc1",
        "max-lsp-bandwidth": [
          {
            "priority": 0,
            "te-bandwidth": {
              "generic": "10000000"
            }
          }
        ]
      }
    ]
  }
},
"ietf-l3-unicast-topology:l3-termination-
  "ip-address": [
    "11.11.0.46"
  ]
},
"tp-id": "NorthStar-Live:node:0110.1200.
0006:11.11.0.46"
},
{
  "ietf-te-topology:te-tp-id": "11.11.0.34",
  "ietf-te-topology:te": {
    "interface-switching-capability": [
      {
        "encoding": "ietf-te-types:lsp-
encoding-packet",
        "switching-capability": "ietf-te-
types:switching-psc1",
        "max-lsp-bandwidth": [
          {
            "priority": 0,
            "te-bandwidth": {
              "generic": "10000000"
            }
          }
        ]
      }
    ]
  }
}
}

```

```

    ],
    },
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
            "11.11.0.34"
        ]
    },
    "tp-id": "NorthStar-Live:node:0110.1200.
0006:11.11.0.34"
    },
    {
        "ietf-te-topology:te-tp-id": "11.11.0.50",
        "ietf-te-topology:te": {
            "interface-switching-capability": [
                {
                    "encoding": "ietf-te-types:lsp-
encoding-packet",
                    "switching-capability": "ietf-te-
types:switching-psc1",
                    "max-lsp-bandwidth": [
                        {
                            "priority": 0,
                            "te-bandwidth": {
                                "generic": "10000000"
                            }
                        }
                    ]
                }
            ]
        }
    },
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
            "11.11.0.50"
        ]
    },
    "tp-id": "NorthStar-Live:node:0110.1200.
0006:11.11.0.50"
    },
    ],
    "ietf-te-topology:te-node-id": "11.12.0.6",
    "ietf-te-topology:te": {
        "te-node-attributes": {
            "name": "use1-sv2-spine1-sv2"
        }
    },
    "ietf-l3-unicast-topology:l3-node-attributes": {
        "name": "use1-sv2-spine1-sv2",
        "router-id": [
            "11.12.0.6"
        ],
        "ietf-sr-mpls-topology:sr-mpls": {
            "": [
                {
                    "lower-bound": 8000,
                    "upper-bound": 9000
                }
            ]
        }
    }
}

```

```

    }
  },
  {
    "node-id": "NorthStar-Live:node:0110.1200.0007",
    "ietf-network-topology:termination-point": [
      {
        "ietf-te-topology:te-tp-id": "11.11.0.49",
        "ietf-te-topology:te": {
          "interface-switching-capability": [
            {
              "encoding": "ietf-te-types:lsp-
encoding-packet",
              "switching-capability": "ietf-te-
types:switching-psc1",
              "max-lsp-bandwidth": [
                {
                  "priority": 0,
                  "te-bandwidth": {
                    "generic": "100000000"
                  }
                }
              ]
            }
          ]
        },
        "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
          "ip-address": [
            "11.11.0.49"
          ],
          "tp-id": "NorthStar-Live:node:0110.1200.
0007:11.11.0.49"
        }
      },
      {
        "ietf-te-topology:te-tp-id": "11.11.0.53",
        "ietf-te-topology:te": {
          "interface-switching-capability": [
            {
              "encoding": "ietf-te-types:lsp-
encoding-packet",
              "switching-capability": "ietf-te-
types:switching-psc1",
              "max-lsp-bandwidth": [
                {
                  "priority": 0,
                  "te-bandwidth": {
                    "generic": "100000000"
                  }
                }
              ]
            }
          ]
        },
        "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
          "ip-address": [
            "11.11.0.53"
          ],
          "tp-id": "NorthStar-Live:node:0110.1200.
0007:11.11.0.53"
        }
      }
    ]
  }
]
}

```



```

                                "tp-id": "NorthStar-Live:node:0110.1200.
0007:11.11.0.53"
                                }
                                ],
                                "ietf-te-topology:te-node-id": "11.12.0.7",
                                "ietf-te-topology:te": {
                                    "te-node-attributes": {
                                        "name": "use2-sv2"
                                    }
                                },
                                "ietf-l3-unicast-topology:l3-node-attributes": {
                                    "name": "use2-sv2",
                                    "router-id": [
                                        "11.12.0.7"
                                    ],
                                    "ietf-sr-mpls-topology:sr-mpls": {
                                        "": [
                                            {
                                                "lower-bound": 8000,
                                                "upper-bound": 9000
                                            }
                                        ],
                                        "msd": 5
                                    }
                                }
                                },
                                {
                                    "node-id": "NorthStar-Live:node:0110.1200.0008",
                                    "ietf-network-topology:termination-point": [
                                        {
                                            "ietf-te-topology:te-tp-id": "11.11.0.89",
                                            "ietf-te-topology:te": {
                                                "interface-switching-capability": [
                                                    {
                                                        "encoding": "ietf-te-types:lsp-
encoding-packet",
                                                        "switching-capability": "ietf-te-
types:switching-psc1",
                                                        "max-lsp-bandwidth": [
                                                            {
                                                                "priority": 0,
                                                                "te-bandwidth": {
                                                                    "generic": "10000000"
                                                                }
                                                            }
                                                        ]
                                                    }
                                                ]
                                            }
                                        ],
                                        "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                                            "ip-address": [
                                                "11.11.0.89"
                                            ]
                                        },
                                        "tp-id": "NorthStar-Live:node:0110.1200.
0008:11.11.0.89"
                                    },
                                    {
                                        "ietf-te-topology:te-tp-id": "11.11.0.93",

```

```

    "ietf-te-topology:te": {
      "interface-switching-capability": [
        {
          "encoding": "ietf-te-types:lsp-
encoding-packet",
          "switching-capability": "ietf-te-
types:switching-psc1",
          "max-lsp-bandwidth": [
            {
              "priority": 0,
              "te-bandwidth": {
                "generic": "10000000"
              }
            }
          ]
        }
      ]
    },
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
      "ip-address": [
        "11.11.0.93"
      ]
    },
    "tp-id": "NorthStar-Live:node:0110.1200.
0008:11.11.0.93"
  },
  {
    "ietf-te-topology:te-tp-id": "11.11.0.133",
    "ietf-te-topology:te": {
      "interface-switching-capability": [
        {
          "encoding": "ietf-te-types:lsp-
encoding-packet",
          "switching-capability": "ietf-te-
types:switching-psc1",
          "max-lsp-bandwidth": [
            {
              "priority": 0,
              "te-bandwidth": {
                "generic": "10000000"
              }
            }
          ]
        }
      ]
    },
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
      "ip-address": [
        "11.11.0.133"
      ]
    },
    "tp-id": "NorthStar-Live:node:0110.1200.
0008:11.11.0.133"
  },
  {
    "ietf-te-topology:te-tp-id": "11.11.0.141",
    "ietf-te-topology:te": {
      "interface-switching-capability": [

```

```

    {
      "encoding": "ietf-te-types:lsp-
encoding-packet",
      "switching-capability": "ietf-te-
types:switching-psc1",
      "max-lsp-bandwidth": [
        {
          "priority": 0,
          "te-bandwidth": {
            "generic": "100000000"
          }
        }
      ]
    }
  ],
  },
  "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
    "ip-address": [
      "11.11.0.141"
    ]
  },
  "tp-id": "NorthStar-Live:node:0110.1200.
0008:11.11.0.141"
},
{
  "ietf-te-topology:te-tp-id": "11.11.0.58",
  "ietf-te-topology:te": {
    "interface-switching-capability": [
      {
        "encoding": "ietf-te-types:lsp-
encoding-packet",
        "switching-capability": "ietf-te-
types:switching-psc1",
        "max-lsp-bandwidth": [
          {
            "priority": 0,
            "te-bandwidth": {
              "generic": "100000000"
            }
          }
        ]
      }
    ]
  },
  "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
    "ip-address": [
      "11.11.0.58"
    ]
  },
  "tp-id": "NorthStar-Live:node:0110.1200.
0008:11.11.0.58"
},
{
  "ietf-te-topology:te-tp-id": "11.11.0.62",
  "ietf-te-topology:te": {
    "interface-switching-capability": [
      {

```

```

                                "encoding": "ietf-te-types:lsp-
encoding-packet",
                                "switching-capability": "ietf-te-
types:switching-psc1",
                                "max-lsp-bandwidth": [
                                    {
                                        "priority": 0,
                                        "te-bandwidth": {
                                            "generic": "10000000"
                                        }
                                    }
                                ]
                            }
                        ],
                    },
                    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                        "ip-address": [
                            "11.11.0.62"
                        ]
                    },
                    "tp-id": "NorthStar-Live:node:0110.1200.
0008:11.11.0.62"
                },
                {
                    "ietf-te-topology:te-tp-id": "11.11.0.38",
                    "ietf-te-topology:te": {
                        "interface-switching-capability": [
                            {
                                "encoding": "ietf-te-types:lsp-
encoding-packet",
                                "switching-capability": "ietf-te-
types:switching-psc1",
                                "max-lsp-bandwidth": [
                                    {
                                        "priority": 0,
                                        "te-bandwidth": {
                                            "generic": "10000000"
                                        }
                                    }
                                ]
                            }
                        ]
                    },
                    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                        "ip-address": [
                            "11.11.0.38"
                        ]
                    },
                    "tp-id": "NorthStar-Live:node:0110.1200.
0008:11.11.0.38"
                },
                {
                    "ietf-te-topology:te-tp-id": "11.11.0.54",
                    "ietf-te-topology:te": {
                        "interface-switching-capability": [
                            {
                                "encoding": "ietf-te-types:lsp-
encoding-packet",

```

```

types:switching-psc1",
                                "switching-capability": "ietf-te-
                                "max-lsp-bandwidth": [
                                    {
                                        "priority": 0,
                                        "te-bandwidth": {
                                            "generic": "10000000"
                                        }
                                    }
                                ]
                            },
                            "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                                "ip-address": [
                                    "11.11.0.54"
                                ]
                            },
                            "tp-id": "NorthStar-Live:node:0110.1200.
0008:11.11.0.54"
                        },
                    ],
                    "ietf-te-topology:te-node-id": "11.12.0.8",
                    "ietf-te-topology:te": {
                        "te-node-attributes": {
                            "name": "use2-sv2-spine2-sv2"
                        }
                    },
                    "ietf-l3-unicast-topology:l3-node-attributes": {
                        "name": "use2-sv2-spine2-sv2",
                        "router-id": [
                            "11.12.0.8"
                        ],
                        "ietf-sr-mpls-topology:sr-mpls": {
                            "": [
                                {
                                    "lower-bound": 8000,
                                    "upper-bound": 9000
                                }
                            ]
                        }
                    }
                },
                {
                    "node-id": "NorthStar-Live:node:0110.1200.0009",
                    "ietf-network-topology:termination-point": [
                        {
                            "ietf-te-topology:te-tp-id": "11.11.0.65",
                            "ietf-te-topology:te": {
                                "interface-switching-capability": [
                                    {
                                        "encoding": "ietf-te-types:lsp-
encoding-packet",
                                        "switching-capability": "ietf-te-
types:switching-psc1",
                                        "max-lsp-bandwidth": [
                                            {
                                                "priority": 0,
                                                "te-bandwidth": {

```

```

                                "generic": "10000000"
                                }
                                }
                                ]
                                }
                                ],
                                "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                                "ip-address": [
                                    "11.11.0.65"
                                ]
                                },
                                "tp-id": "NorthStar-Live:node:0110.1200.
0009:11.11.0.65"
                                },
                                {
                                    "ietf-te-topology:te-tp-id": "11.11.0.69",
                                    "ietf-te-topology:te": {
                                        "interface-switching-capability": [
                                            {
                                                "encoding": "ietf-te-types:lsp-
encoding-packet",
                                                "switching-capability": "ietf-te-
types:switching-psc1",
                                                "max-lsp-bandwidth": [
                                                    {
                                                        "priority": 0,
                                                        "te-bandwidth": {
                                                            "generic": "10000000"
                                                        }
                                                    }
                                                ]
                                            }
                                        ]
                                    }
                                },
                                "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                                "ip-address": [
                                    "11.11.0.69"
                                ]
                                },
                                "tp-id": "NorthStar-Live:node:0110.1200.
0009:11.11.0.69"
                                }
                                ],
                                "ietf-te-topology:te-node-id": "11.12.0.9",
                                "ietf-te-topology:te": {
                                    "te-node-attributes": {
                                        "name": "use1-sv3"
                                    }
                                },
                                "ietf-l3-unicast-topology:l3-node-attributes": {
                                    "name": "use1-sv3",
                                    "router-id": [
                                        "11.12.0.9"
                                    ],
                                    "ietf-sr-mpls-topology:sr-mpls": {
                                        "": [
                                            {

```

```

        "lower-bound": 8000,
        "upper-bound": 9000
      }
    ],
    "msd": 5
  }
},
{
  "node-id": "NorthStar-Live:node:1.1.1.1",
  "ietf-network-topology:termination-point": [
    {
      "ietf-te-topology:te-tp-id": "10.0.0.1",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic":
"100000000000"
                }
              }
            ]
          }
        ]
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "10.0.0.1"
        ]
      },
      "tp-id": "NorthStar-Live:node:1.1.1.1:10.0.0.
1"
    },
    {
      "ietf-te-topology:te-tp-id": "10.0.0.5",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic":
"200000000000"
                }
              }
            ]
          }
        ]
      }
    }
  ]
}

```

```

    ],
    },
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
            "10.0.0.5"
        ]
    },
    "tp-id": "NorthStar-Live:node:1.1.1.1:10.0.0.
5"
    }
],
"ietf-te-topology:te-node-id": "1.1.1.1",
"ietf-te-topology:te": {
    "te-node-attributes": {
        "name": "h-65535-01010101"
    }
},
"ietf-l3-unicast-topology:l3-node-attributes": {
    "name": "h-65535-01010101",
    "router-id": [
        "1.1.1.1"
    ]
}
},
{
    "node-id": "NorthStar-Live:node:127.0.0.1",
    "ietf-network-topology:termination-point": [
        {
            "ietf-te-topology:te-tp-id": "10.0.0.13",
            "ietf-te-topology:te": {
                "interface-switching-capability": [
                    {
                        "encoding": "ietf-te-types:lsp-
encoding-packet",
                        "switching-capability": "ietf-te-
types:switching-psc1",
                        "max-lsp-bandwidth": [
                            {
                                "priority": 0,
                                "te-bandwidth": {
                                    "generic":
"200000000000"
                                }
                            }
                        ]
                    }
                ]
            }
        },
        {
            "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                "ip-address": [
                    "10.0.0.13"
                ]
            },
            "tp-id": "NorthStar-Live:node:127.0.0.1:10.0.
0.13"
        },
        {
            "ietf-te-topology:te-tp-id": "10.0.0.9",

```



```

        "ietf-te-topology:te": {
          "interface-switching-capability": [
            {
              "encoding": "ietf-te-types:lsp-
encoding-packet",
              "switching-capability": "ietf-te-
types:switching-psc1",
              "max-lsp-bandwidth": [
                {
                  "priority": 0,
                  "te-bandwidth": {
                    "generic":
"100000000000"
                  }
                }
              ]
            }
          ],
          "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
            "ip-address": [
              "10.0.0.9"
            ],
            "tp-id": "NorthStar-Live:node:127.0.0.1:10.0.
0.9"
          }
        },
        "ietf-te-topology:te-node-id": "127.0.0.1",
        "ietf-te-topology:te": {
          "te-node-attributes": {
            "name": "h-0-7f000001"
          }
        },
        "ietf-l3-unicast-topology:l3-node-attributes": {
          "name": "h-0-7f000001",
          "router-id": [
            "127.0.0.1"
          ]
        }
      },
      {
        "node-id": "NorthStar-Live:node:127.0.0.2",
        "ietf-network-topology:termination-point": [
          {
            "ietf-te-topology:te-tp-id": "10.0.0.2",
            "ietf-te-topology:te": {
              "interface-switching-capability": [
                {
                  "encoding": "ietf-te-types:lsp-
encoding-packet",
                  "switching-capability": "ietf-te-
types:switching-psc1",
                  "max-lsp-bandwidth": [
                    {
                      "priority": 0,
                      "te-bandwidth": {
                        "generic":
"100000000000"
                      }
                    }
                  ]
                }
              ]
            }
          ]
        }
      }
    ]
  }
}

```

```

    },
    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
            "10.0.0.2"
        ],
        "tp-id": "NorthStar-Live:node:127.0.0.2:10.0.
0.2"
    },
    {
        "ietf-te-topology:te-tp-id": "10.0.0.10",
        "ietf-te-topology:te": {
            "interface-switching-capability": [
                {
                    "encoding": "ietf-te-types:lsp-
encoding-packet",
                    "switching-capability": "ietf-te-
types:switching-psc1",
                    "max-lsp-bandwidth": [
                        {
                            "priority": 0,
                            "te-bandwidth": {
                                "generic":
"100000000000"
                            }
                        }
                    ]
                }
            ]
        },
        "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
            "ip-address": [
                "10.0.0.10"
            ],
            "tp-id": "NorthStar-Live:node:127.0.0.2:10.0.
0.10"
        },
        {
            "ietf-te-topology:te-tp-id": "10.0.0.17",
            "ietf-te-topology:te": {
                "interface-switching-capability": [
                    {
                        "encoding": "ietf-te-types:lsp-
encoding-packet",
                        "switching-capability": "ietf-te-
types:switching-psc1",
                        "max-lsp-bandwidth": [
                            {
                                "priority": 0,
                                "te-bandwidth": {
                                    "generic":
"100000000000"
                                }
                            }
                        ]
                    }
                ]
            }
        }
    ]
}

```

```

    }
  }
]
},
"ietf-l3-unicast-topology:l3-termination-
point-attributes": {
  "ip-address": [
    "10.0.0.17"
  ],
  "tp-id": "NorthStar-Live:node:127.0.0.2:10.0.
0.17"
},
{
  "ietf-te-topology:te-tp-id": "10.0.0.21",
  "ietf-te-topology:te": {
    "interface-switching-capability": [
      {
        "encoding": "ietf-te-types:lsp-
encoding-packet",
        "switching-capability": "ietf-te-
types:switching-psc1",
        "max-lsp-bandwidth": [
          {
            "priority": 0,
            "te-bandwidth": {
              "generic":
"200000000000"
            }
          }
        ]
      }
    ]
  },
  "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
    "ip-address": [
      "10.0.0.21"
    ],
    "tp-id": "NorthStar-Live:node:127.0.0.2:10.0.
0.21"
  },
  "ietf-te-topology:te-node-id": "127.0.0.2",
  "ietf-te-topology:te": {
    "te-node-attributes": {
      "name": "h-0-7f000002"
    }
  },
  "ietf-l3-unicast-topology:l3-node-attributes": {
    "name": "h-0-7f000002",
    "router-id": [
      "127.0.0.2"
    ]
  }
},
{

```

```

        "node-id": "NorthStar-Live:node:127.0.0.3",
        "ietf-network-topology:termination-point": [
            {
                "ietf-te-topology:te-tp-id": "10.0.0.6",
                "ietf-te-topology:te": {
                    "interface-switching-capability": [
                        {
                            "encoding": "ietf-te-types:lsp-
encoding-packet",
                            "switching-capability": "ietf-te-
types:switching-psc1",
                            "max-lsp-bandwidth": [
                                {
                                    "priority": 0,
                                    "te-bandwidth": {
                                        "generic":
                                        "200000000000"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                    "ip-address": [
                        "10.0.0.6"
                    ],
                    "tp-id": "NorthStar-Live:node:127.0.0.3:10.0.
0.6"
                },
                {
                    "ietf-te-topology:te-tp-id": "10.0.0.14",
                    "ietf-te-topology:te": {
                        "interface-switching-capability": [
                            {
                                "encoding": "ietf-te-types:lsp-
encoding-packet",
                                "switching-capability": "ietf-te-
types:switching-psc1",
                                "max-lsp-bandwidth": [
                                    {
                                        "priority": 0,
                                        "te-bandwidth": {
                                            "generic":
                                            "200000000000"
                                        }
                                    }
                                ]
                            }
                        ]
                    },
                    "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                        "ip-address": [
                            "10.0.0.14"
                        ],
                    },

```

```

0.14"
    },
    {
      "ietf-te-topology:te-tp-id": "10.0.0.18",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic":
                    "100000000000"
                }
              }
            ]
          }
        ]
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "10.0.0.18"
        ]
      },
      "tp-id": "NorthStar-Live:node:127.0.0.3:10.0.
0.18"
    },
    {
      "ietf-te-topology:te-tp-id": "10.0.0.25",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic":
                    "100000000000"
                }
              }
            ]
          }
        ]
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "10.0.0.25"
        ]
      },
    },
  ],
}

```

```

    "tp-id": "NorthStar-Live:node:127.0.0.3:10.0.
0.25"
    },
    {
      "ietf-te-topology:te-tp-id": "10.0.0.29",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [
              {
                "priority": 0,
                "te-bandwidth": {
                  "generic":
                    "200000000000"
                }
              }
            ]
          }
        ]
      },
      "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
        "ip-address": [
          "10.0.0.29"
        ]
      },
      "tp-id": "NorthStar-Live:node:127.0.0.3:10.0.
0.29"
    }
  ],
  "ietf-te-topology:te-node-id": "127.0.0.3",
  "ietf-te-topology:te": {
    "te-node-attributes": {
      "name": "h-0-7f000003"
    }
  },
  "ietf-l3-unicast-topology:l3-node-attributes": {
    "name": "h-0-7f000003",
    "router-id": [
      "127.0.0.3"
    ]
  }
},
{
  "node-id": "NorthStar-Live:node:127.0.0.4",
  "ietf-network-topology:termination-point": [
    {
      "ietf-te-topology:te-tp-id": "10.0.0.22",
      "ietf-te-topology:te": {
        "interface-switching-capability": [
          {
            "encoding": "ietf-te-types:lsp-
encoding-packet",
            "switching-capability": "ietf-te-
types:switching-psc1",
            "max-lsp-bandwidth": [

```

```

    {
      "priority": 0,
      "te-bandwidth": {
        "generic":
"200000000000"
      }
    }
  ]
}
],
{
  "ietf-l3-unicast-topology:l3-termination-
    "ip-address": [
      "10.0.0.22"
    ]
  },
  "tp-id": "NorthStar-Live:node:127.0.0.4:10.0.
0.22"
},
{
  "ietf-te-topology:te-tp-id": "10.0.0.26",
  "ietf-te-topology:te": {
    "interface-switching-capability": [
      {
        "encoding": "ietf-te-types:lsp-
encoding-packet",
        "switching-capability": "ietf-te-
types:switching-psc1",
        "max-lsp-bandwidth": [
          {
            "priority": 0,
            "te-bandwidth": {
              "generic":
"100000000000"
            }
          }
        ]
      }
    ]
  },
  "ietf-l3-unicast-topology:l3-termination-
    "ip-address": [
      "10.0.0.26"
    ]
  },
  "tp-id": "NorthStar-Live:node:127.0.0.4:10.0.
0.26"
},
{
  "ietf-te-topology:te-tp-id": "10.0.0.33",
  "ietf-te-topology:te": {
    "interface-switching-capability": [
      {
        "encoding": "ietf-te-types:lsp-
encoding-packet",
        "switching-capability": "ietf-te-
types:switching-psc1",
        "max-lsp-bandwidth": [

```

```

    {
        "priority": 0,
        "te-bandwidth": {
            "generic":
"100000000000"
        }
    }
]
}
],
},
"ietf-l3-unicast-topology:l3-termination-
    "ip-address": [
        "10.0.0.33"
    ]
},
"tp-id": "NorthStar-Live:node:127.0.0.4:10.0.
0.33"
},
{
    "ietf-te-topology:te-tp-id": "10.0.0.37",
    "ietf-te-topology:te": {
        "interface-switching-capability": [
            {
                "encoding": "ietf-te-types:lsp-
encoding-packet",
                "switching-capability": "ietf-te-
types:switching-psc1",
                "max-lsp-bandwidth": [
                    {
                        "priority": 0,
                        "te-bandwidth": {
                            "generic":
"200000000000"
                        }
                    }
                ]
            }
        ]
    },
    "ietf-l3-unicast-topology:l3-termination-
        "ip-address": [
            "10.0.0.37"
        ]
    },
    "tp-id": "NorthStar-Live:node:127.0.0.4:10.0.
0.37"
    }
],
"ietf-te-topology:te-node-id": "127.0.0.4",
"ietf-te-topology:te": {
    "te-node-attributes": {
        "name": "h-0-7f000004"
    }
},
"ietf-l3-unicast-topology:l3-node-attributes": {
    "name": "h-0-7f000004",
    "router-id": [

```



```

        "127.0.0.4"
      ]
    },
    {
      "node-id": "NorthStar-Live:node:127.0.0.5",
      "ietf-network-topology:termination-point": [
        {
          "ietf-te-topology:te-tp-id": "10.0.0.30",
          "ietf-te-topology:te": {
            "interface-switching-capability": [
              {
                "encoding": "ietf-te-types:lsp-
encoding-packet",
                "switching-capability": "ietf-te-
types:switching-psc1",
                "max-lsp-bandwidth": [
                  {
                    "priority": 0,
                    "te-bandwidth": {
                      "generic":
"200000000000"
                    }
                  }
                ]
              }
            ]
          },
          "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
            "ip-address": [
              "10.0.0.30"
            ],
            "tp-id": "NorthStar-Live:node:127.0.0.5:10.0.
0.30"
          },
          {
            "ietf-te-topology:te-tp-id": "10.0.0.34",
            "ietf-te-topology:te": {
              "interface-switching-capability": [
                {
                  "encoding": "ietf-te-types:lsp-
encoding-packet",
                  "switching-capability": "ietf-te-
types:switching-psc1",
                  "max-lsp-bandwidth": [
                    {
                      "priority": 0,
                      "te-bandwidth": {
                        "generic":
"100000000000"
                      }
                    }
                  ]
                }
              ]
            },
            "ietf-l3-unicast-topology:l3-termination-
point-attributes": {

```

```

        "ip-address": [
            "10.0.0.34"
        ],
        "tp-id": "NorthStar-Live:node:127.0.0.5:10.0.
0.34"
    },
    },
    "ietf-te-topology:te-node-id": "127.0.0.5",
    "ietf-te-topology:te": {
        "te-node-attributes": {
            "name": "127.0.0.5"
        }
    },
    "ietf-l3-unicast-topology:l3-node-attributes": {
        "router-id": [
            "127.0.0.5"
        ]
    }
},
{
    "node-id": "NorthStar-Live:node:127.0.0.6",
    "ietf-network-topology:termination-point": [
        {
            "ietf-te-topology:te-tp-id": "10.0.0.38",
            "ietf-te-topology:te": {
                "interface-switching-capability": [
                    {
                        "encoding": "ietf-te-types:lsp-
encoding-packet",
                        "switching-capability": "ietf-te-
types:switching-psc1",
                        "max-lsp-bandwidth": [
                            {
                                "priority": 0,
                                "te-bandwidth": {
                                    "generic":
"2000000000000"
                                }
                            }
                        ]
                    }
                ]
            },
            "ietf-l3-unicast-topology:l3-termination-
point-attributes": {
                "ip-address": [
                    "10.0.0.38"
                ],
                "tp-id": "NorthStar-Live:node:127.0.0.6:10.0.
0.38"
            }
        },
        "ietf-te-topology:te-node-id": "127.0.0.6",
        "ietf-te-topology:te": {
            "te-node-attributes": {
                "name": "127.0.0.6"
            }
        }
    },

```

```
        "ietf-l3-unicast-topology:l3-node-attributes": {  
            "router-id": [  
                "127.0.0.6"  
            ]  
        }  
    },  
    ],  
    "provider-id": 0,  
    "client-id": 0  
}  
]  
}
```

2.5.2. Get container ietf-network:networks

Method	URI	Description
GET	/operational/ietf-network:networks	GET container ietf-network:networks.

Return the data for ietf-network:networks defined in revision 2017-01-03 Serves as top-level container for a list of networks.

Normal response codes: 200

2.5.2.1. Request

This operation does not accept a request body.

2.5.2.2. Response

See the /data example

2.6. ietf-te YANG Model

Access the Pathfinder data model represented as ietf-te and extensions.

The following document version are supported as read-only models.

Table 2.4. Pathfinder ietf-te and extensions

Model	Document	Revision
ietf-te	draft-ietf-teas-yang-te-25	2020-07-12
ietf-segment-routing-common	draft-ietf-spring-sr-yang-15	2020-01-09
ietf-te-mpls	draft-ietf-teas-yang-te-mpls-03	2020-03-09

Read/write access is provided for the named admin group container, in addition to the GET method the POST/PUT/PATCH/DELETE are supported on the container and its childs.

Method	URI	Description
GET	/data/ietf-te:te	GET container ietf-te:te.
GET	/operational/ietf-te:te	GET container ietf-te:te.
POST	/data/ietf-te:te/globals/named-admin-groups	Create new enty in container named-admin-groups.
PUT	/data/ietf-te:te/globals/named-admin-groups	Replace the container named-admin-groups or child .
PATCH	/data/ietf-te:te/globals/named-admin-groups	Replace the container named-admin-groups or child.
DELETE	/data/ietf-te:te/globals/named-admin-groups	delete the container named-admin-groups or child .

2.6.1. Get container ietf-te:te

Method	URI	Description
GET	/data/ietf-te:te	GET container ietf-te:te.

Return the data for ietfTE defined in draft-ietf-teas-yang-te-25 revision 2020-07-12 TE global container.

Normal response codes: 200

2.6.1.1. Request

This operation does not accept a request body.

2.6.1.2. Response

Example 2.2. Get container ietf-te:te : application/yang-data+jsonresponse

Response example:

```
{
  "ietf-te:te": {
    "globals": {
      "named-admin-groups": {
        "named-admin-group": [
          {
            "name": "Red",
            "bit-position": 11
          },
          {
            "name": "Blue",
            "bit-position": 2
          },
          {
            "name": "yellow",
            "bit-position": 3
          }
        ]
      }
    },
    "tunnels": {
      "tunnel": [
        {
          "operational-state": "ietf-te-type:tunnel-state-up",
          "identifier": 63,
          "name": "ExampleTunnel 11.12.0.5",
          "te-topology-identifier": {
            "topology-id": "NorthStar-Live",
            "provider-id": 0,
            "client-id": 0
          },
          "source": "11.12.0.5",
          "destination": "11.12.0.9",
          "setup-priority": 7,
          "hold-priority": 7,
          "primary-paths": {
            "primary-path": [
```

```

{
  "name": "ExampleTunnel",
  "setup-priority": 7,
  "hold-priority": 7,
  "computed-paths-properties": {
    "computed-path-properties": [
      {
        "k-index": 1,
        "path-properties": {
          "te-bandwidth": {
            "generic": "0"
          },
          "path-route-objects": {
            "path-route-object": [
              {
                "index": 1,
                "numbered-link-
hop": {
                  "link-tp-id":
                    "11.11.0.38",
                  "hop-type":
                    "strict",
                  "direction":
                    "outgoing"
                }
              },
              {
                "index": 2,
                "numbered-link-
hop": {
                  "link-tp-id":
                    "11.11.0.90",
                  "hop-type":
                    "strict",
                  "direction":
                    "outgoing"
                }
              },
              {
                "index": 3,
                "numbered-link-
hop": {
                  "link-tp-id":
                    "11.11.0.69",
                  "hop-type":
                    "strict",
                  "direction":
                    "outgoing"
                }
              }
            ]
          }
        }
      }
    ]
  },
  "lsp": {
    "lsp": [
      {
        "tunnel-name": "ExampleTunnel",

```

```

        "node": "11.12.0.5",
        "lsp-id": 1,
        "path-properties": {
            "te-bandwidth": {
                "generic": "0"
            },
            "path-route-objects": {
                "path-route-object": [
                    {
                        "index": 1,
                        "numbered-link-
hop": {
                            "link-tp-id":
                                "11.11.0.38",
                                "hop-type":
                                    "strict",
                                    "direction":
                                        "outgoing"
                                }
                    },
                    {
                        "index": 2,
                        "numbered-link-
hop": {
                            "link-tp-id":
                                "11.11.0.90",
                                "hop-type":
                                    "strict",
                                    "direction":
                                        "outgoing"
                                }
                    },
                    {
                        "index": 3,
                        "numbered-link-
hop": {
                            "link-tp-id":
                                "11.11.0.69",
                                "hop-type":
                                    "strict",
                                    "direction":
                                        "outgoing"
                                }
                    }
                ]
            }
        },
        "path-computation-method": "te-types:path-
locally-computed",
        "use-path-computation": true,
        "path-scope": "te-types:path-scope-end-to-
end",
        "preference": 1,
        "k-requested-paths": 1,
        "link-protection": "te-types:link-protection-
unprotected",
        "signaling-type": "te-types:path-setup-rsvp"

```

```

    },
    "description": "None",
    "encoding": "te-types:lsp-encoding-packet",
    "switching-type": "te-types:switching-psc1",
    "admin-state": "te-types:tunnel-admin-state-up",
    "bidirectional": false,
    "link-protection": "te-types:link-protection-unprotected",
    "signaling-type": "te-types:path-setup-rsvp"
  }
]
},
{lsp: {
  "lsp": [
    {
      "tunnel-name": "ExampleTunnel",
      "source": "11.12.0.5",
      "node": "11.12.0.5",
      "destination": "11.12.0.9",
      "lsp-id": 1,
      "extended-tunnel-id": "11.12.0.5",
      "lsp-protection-role": "working",
      "operational-state": "ietf-te-types:state-up",
      "origin-type": "ingress",
      "signaling-type": "ietf-te-types:path-setup-rsvp",
      "lsp-record-route-information": {
        "lsp-record-route-information": [
          {
            "index": 1,
            "numbered-link-hop": {
              "link-tp-id": "11.11.0.38"
            }
          },
          {
            "index": 2,
            "numbered-link-hop": {
              "link-tp-id": "11.11.0.90"
            }
          },
          {
            "index": 3,
            "numbered-link-hop": {
              "link-tp-id": "11.11.0.69"
            }
          }
        ]
      }
    }
  ],
  "lsp-resource-status": "primary",
  "lockout-of-normal": false,
  "freeze": false,
  "lsp-protection-state": "te-types:normal",
  "protection-group-ingress-node-id": "0.0.0.0",
  "protection-group-egress-node-id": "0.0.0.0"
}
}
}

```


2.6.2. Get container ietf-te:te

Method	URI	Description
GET	/operational/ietf-te:te	GET container ietf-te:te.

Normal response codes: 200

2.6.2.1. Request

This operation does not accept a request body.

2.6.2.2. Response

See the /data example

2.6.3. Insert in container named-admin-groups

Method	URI	Description
POST	/data/ietf-te:te/globals/named-admin-groups	Create new entry in container named-admin-groups.

Normal response codes: 201

2.6.3.1. Request

Example 2.3. Insert in container named-admin-groups: application/yang-data+jsonrequest

```
{"named-admin-group":[{"name":"green","bit-position":1}]}
```

2.6.3.2. Response

On success empty response (Following RFC8040 Section-4.4)

2.6.4. replace container named-admin-groups

Method	URI	Description
PUT	/data/ietf-te:te/globals/named-admin-groups	Replace the container named-admin-groups or child .

Normal response codes: 200

2.6.4.1. Request

Example 2.4. replace container named-admin-groups: application/yang-data+jsonrequest

```
{"named-admin-group":[{"name":"Red","bit-position":1}, {"name":"Blue","bit-position":2}, {"name":"yellow","bit-position":3}]}
```

2.6.4.2. Response

On success empty response (Following RFC8040 Section-4.5)

2.6.5. modify container named-admin-groups

Method	URI	Description
PATCH	/data/ietf-te:te/globals/named-admin-groups	Replace the container named-admin-groups or child.

Normal response codes: 200

2.6.5.1. Request

Example 2.5. modify container named-admin-groups: application/yang-data+jsonrequest

PATCH merge the content of the body with the target resource, for instance to change the bit-position of the Red named admin group a PATH request can be made to /named-admin-group=Red with the following payload

```
{"named-admin-group":[{"bit-position":11}]}
```

2.6.5.2. Response

On success empty response (Following RFC8040 Section-4.6)

2.6.6. delete container named-admin-groups

Method	URI	Description
DELETE	/data/ietf-te:te/globals/named-admin-groups	delete the container named-admin-groups or child .

Normal response codes: 200

2.6.6.1. Request

Example 2.6. delete container named-admin-groups: application/yang-data+jsonrequest

2.6.6.2. Response

On success empty response (Following RFC8040 Section-4.7)

2.7. ietf-yang-library YANG Model

The list of supported YANG models and their version is accessed through the following endpoint.

Deviation: Pathfinder does not support the features parameter.

Method	URI	Description
GET	/data/ietf-yang-library:modules-state	GET container ietf-yang-library:modules-state.

2.7.1. Get container ietf-yang-library:modules-state

Method	URI	Description
GET	/data/ietf-yang-library:modules-state	GET container ietf-yang-library:modules-state.

Return the data for ietf-yang-library_modules-state defined in revision 2016-06-21 Contains YANG module monitoring information. The feature and deviation are not supported by NorthStar

Normal response codes: 200

2.7.1.1. Request

This operation does not accept a request body.

2.7.1.2. Response

Example 2.7. Get container ietf-yang-library:modules-state : application/yang-data+jsonresponse

Response example:

```
{
  "ietf-yang-library:modules-state": {
    "module-set-id": "6035e37f290102c5ae5191356c13b24a",
    "module": [
      {
        "name": "ietf-inet-types",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-inet-types",
        "revision": "2013-07-15",
        "feature": [],
        "conformance-type": "implement"
      },
      {
        "name": "ietf-yang-types",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-types",
        "revision": "2013-07-15",
        "feature": [],
        "conformance-type": "implement"
      },
      {
        "name": "ietf-routing-types",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-routing-types",
        "revision": "2017-12-04",
        "feature": [],
        "conformance-type": "implement"
      },
      {
        "name": "ietf-network",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-network",
        "revision": "2018-02-26",
        "feature": [],
        "conformance-type": "implement"
      },
      {
        "name": "ietf-network-topology",
```

```

    "namespace": "urn:ietf:params:xml:ns:yang:ietf-network-
topology",
    "revision": "2018-02-26",
    "feature": [],
    "conformance-type": "implement"
  },
  {
    "name": "ietf-te-types",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-te-types",
    "revision": "2020-06-10",
    "feature": [
      "p2mp-te",
      "frr-te",
      "extended-admin-groups",
      "named-path-affinities",
      "named-extended-admin-groups",
      "named-srlg-groups",
      "named-path-constraints",
      "path-optimization-metric",
      "path-optimization-objective-function"
    ],
    "conformance-type": "implement"
  },
  {
    "name": "ietf-te-topology",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-te-topology",
    "revision": "2020-08-06",
    "feature": [
      "nsrlg",
      "te-topology-hierarchy",
      "template"
    ],
    "conformance-type": "implement"
  },
  {
    "name": "ietf-l3-unicast-topology",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-l3-unicast-
topology",
    "revision": "2018-02-26",
    "feature": [],
    "conformance-type": "implement"
  },
  {
    "name": "ietf-l3-te-topology",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-l3-te-
topology",
    "revision": "2020-05-03",
    "feature": [],
    "conformance-type": "implement"
  },
  {
    "name": "ietf-segment-routing-common",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-segment-
routing-common",
    "revision": "2020-01-09",
    "feature": [
      "sid-last-hop-behavior"
    ],
    "conformance-type": "implement"
  },
  },

```

```

    {
      "name": "ietf-sr-mpls-topology",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-sr-mpls-
topology",
      "revision": "2020-11-01",
      "feature": [
        "msd"
      ],
      "conformance-type": "implement"
    },
    {
      "name": "ietf-te-packet-types",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-te-packet-
types",
      "revision": "2020-06-10",
      "feature": [],
      "conformance-type": "implement"
    },
    {
      "name": "ietf-te-topology-packet",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-te-topology-
packet",
      "revision": "2020-07-03",
      "feature": [
        "te-performance-metric"
      ],
      "conformance-type": "implement"
    },
    {
      "name": "ietf-te",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-te",
      "revision": "2020-07-12",
      "feature": [],
      "conformance-type": "implement"
    },
    {
      "name": "ietf-te-mpls",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-te-mpls",
      "revision": "2020-03-09",
      "feature": [],
      "conformance-type": "implement"
    },
    {
      "name": "ietf-yang-library",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-library",
      "revision": "2016-06-21",
      "feature": [],
      "conformance-type": "implement"
    }
  ]
}

```

2.8. Supported YANG operations

Pathfinder supports the following RESTCONF RPCs

Method	URI	Description
POST	/operations/ietf-te-operations:tunnel-atomic-create	RPC ietf-te-operations:tunnel-atomic-create
POST	/operations/ietf-te-operations:tunnel-atomic-delete	RPC ietf-te-operations:tunnel-atomic-delete

2.8.1. execute RPC ietf-te-operations:tunnel-atomic-create

Method	URI	Description
POST	/operations/ietf-te-operations:tunnel-atomic-create	RPC ietf-te-operations:tunnel-atomic-create

Execute the RPC calls ietf-te-operations:tunnel-atomic-create defined in ietf-te-operations. Atomic request for tunnel operation in case of cooperative operation. Requested tunnel is created in the datastore only if feasible; in other terms resources are available at the time of the RPC. The creation is done according to requested attributes: so if compute-only flag is present, no resources are allocated in the network. The tunnels, created using this RPC, are not ephemeral. In case of no path found or of failure no tunnel must be created both in the datastore and in the network: no resource must remain allocated. The RPC MUST return only when all the operations are concluded, also in the network if requested.

Normal response codes: 200

2.8.1.1. Request

Example 2.8. execute RPC ietf-te-operations:tunnel-atomic-create: application/yang-data+jsonrequest

NorthStar has additional restrictions, described by the following schema:

```
{
  "id": "ietf-te-operations-deviation.json",
  "title": "NS restrictions",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "$ref": "ietf-te-operations-deviation.json#/definitions/tunnelConfig",
  "definitions": {
    "tunnelConfig": {
      "type": "object",
      "required": ["name", "source", "destination", "identifier"],
      "additionalProperties": true,
      "properties": {
        "identifier": {
          "type": "integer"
        },
        "source": {
          "type": "string"
        },
        "destination": {
          "type": "string"
        },
        "type": {
          "type": "string",
          "enum": ["ietf-te-types:tunnel-p2p"]
        },
        "admin-status": {
          "type": "string",
          "enum": ["ietf-te-types:state-up"]
        },
        "lsp-protection-type": {
          "type": "string",
          "enum": ["lsp-prot-unprotected", "lsp-prot-reroute-extra", "lsp-prot-reroute", "ietf-te-types:lsp-prot-unprotected", "ietf-te-types:lsp-prot-reroute-extra", "ietf-te-types:lsp-prot-reroute"]
        }
      }
    }
  }
}
```

Request example:

```
{
  "input": {
    "tunnel-request": [
      {
        "id": "11",
        "config": {
          "identifier": 15,
          "name": "tunnel-type=p2p&pce=174.59.0.12&tid=tunnel:15&id=15",
          "source": "11.12.0.1",
          "destination": "11.12.0.9",
          "lsp-priority-setup": 3,
          "lsp-priority-hold": 0,
          "te-bandwidth": {
            "psc": "SZiWgA=="
          }
        }
      }
    ]
  }
}
```

```

    },
    "primary-paths": [
      {
        "config": {
          "name": "path=1&pce=174.59.0.12&tid=tunnel:15&id=15",
          "preference": 1,
          "dynamic": [
            null
          ],
          "path-constraints": {
            "cost-limit": 200000000,
            "metric-type": "ietf-te-types:path-metric-igp"
          }
        }
      }
    ]
  }
}

```

2.8.1.2. Response

Example 2.9. `execute RPC ietf-te-operations:tunnel-atomic-create: application/yang-data+jsonresponse`

Response example:

```

{
  "output": {
    "response": [
      {
        "id": "11",
        "source": "11.12.0.1",
        "destination": "11.12.0.9",
        "tunnel-name": "tunnel-type=p2p&pce=174.59.0.12&tid=tunnel:15&id=15 11.12.0.1",
        "tunnel-identifier": 52,
        "primary-paths": [
          {
            "state": {
              "name": "path=1&pce=174.59.0.12&tid=tunnel:15&id=15",
              "path-constraints": {
                "metric-type": "ietf-te-types:path-metric-igp",
                "cost-limit": 200000000
              },
              "dynamic": [
                null
              ]
            }
          }
        ]
      }
    ]
  }
}

```

2.8.2. execute RPC ietf-te-operations:tunnel-atomic-delete

Method	URI	Description
POST	/operations/ietf-te-operations:tunnel-atomic-delete	RPC ietf-te-operations:tunnel-atomic-delete

Execute the RPC calls ietf-te-operations:tunnel-atomic-delete defined in ietf-te-operations. The RPC is an Atomic request for tunnel operation in case of cooperative operation. Requested tunnel is removed in the datastore only if feasible; The RPC MUST return only when all the operations are concluded, also in the network if requested.

Normal response codes: 200

2.8.2.1. Request

NorthStar has additional restrictions, described by the following schema:

```
{
  "id": "ietf-te-operations-deviation.json",
  "title": "NS restrictions",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "$ref": "ietf-te-operations-deviation.json#/definitions/tunnelConfig",
  "definitions": {
    "tunnelConfig": {
      "type": "object",
      "required": ["name", "source", "destination", "identifier"],
      "additionalProperties": true,
      "properties": {
        "identifier": {
          "type": "integer"
        },
        "source": {
          "type": "string"
        },
        "destination": {
          "type": "string"
        },
        "type": {
          "type": "string",
          "enum": ["ietf-te-types:tunnel-p2p"]
        },
        "admin-status": {
          "type": "string",
          "enum": ["ietf-te-types:state-up"]
        },
        "lsp-protection-type": {
          "type": "string",
          "enum": ["lsp-prot-unprotected", "lsp-prot-reroute-extra", "lsp-prot-reroute", "ietf-te-types:lsp-prot-unprotected", "ietf-te-types:lsp-prot-reroute-extra", "ietf-te-types:lsp-prot-reroute"]
        }
      }
    }
  }
}
```

2.8.2.2. Response

2.9. Network Slicing

The first endpoint follows Pathfinder RESTCONF API, while the following ones starting with /traffic-engineering/api/ belong to [Pathfinder REST API](#). We put them together here because they are both parts of Network Slicing feature.

Method	URI	Description
GET	/data/juniper-pathfinder-topology-group:topology-groups	Get slice id information via topology-group.
PUT	/traffic-engineering/api/topology/v2/{topology_id}/nodes/{nodeIndex}	Request Payload with sliceId.
PUT	/traffic-engineering/api/topology/v2/{topology_id}/nodes/{nodeIndex}	Request Payload without sliceId.
PATCH	/traffic-engineering/api/topology/v2/{topology_id}/nodes/bulk	Adding slice id to nodes using patch with op = add
PATCH	/traffic-engineering/api/topology/v2/{topology_id}/nodes/bulk	Deleting slice id from nodes using patch with op = remove

Method	URI	Description
PUT	/traffic-engineering/api/topology/v2/{topology_id}/links/{linkIndex}	Request Payload with sliceId.
PUT	/traffic-engineering/api/topology/v2/{topology_id}/links/{linkIndex}	Request Payload without sliceId.
POST	/traffic-engineering/api/topology/v2/{topology_id}/te-lsps	Slice Include All, Slice Include Any, or Slice Include Exclude.

2.9.1. Get slice id information via topology-group

Method	URI	Description
GET	/data/juniper-pathfinder-topology-group:topology-groups	Get slice id information via topology-group.

Normal response codes: 200

2.9.1.1. Request

This operation does not accept a request body.

2.9.1.2. Response

Example 2.10. Get slice id information via topology-group: application/yang-data+jsonresponse

```
{
  "juniper-pathfinder-topology-group:topology-groups": [
    {
      "link-count": "4",
      "links": [
        "12",
        "14",
        "11",
        "7"
      ],
      "node-count": "4",
      "nodes": [
        "5",
        "9",
        "10",
        "1"
      ],
      "tg-id": 100
    },
    {
      "link-count": "3",
      "links": [
        "2",
        "7",
        "4"
      ],
      "node-count": "3",
      "nodes": [
        "5",
        "3",
        "1"
      ],
      "tg-id": 101
    },
    {
      "link-count": "3",
      "links": [
        "4",
        "7",

```

```
        "2"
      ],
      "node-count": "3",
      "nodes": [
        "3",
        "1",
        "5"
      ],
      "tg-id": 102
    },
    {
      "link-count": "7",
      "links": [
        "7",
        "15",
        "16",
        "4",
        "11",
        "14",
        "12"
      ],
      "node-count": "6",
      "nodes": [
        "3",
        "11",
        "1",
        "5",
        "10",
        "9"
      ],
      "tg-id": 103
    }
  ]
}
```

2.9.2. Adding slice ID to Node

Method	URI	Description
PUT	/traffic-engineering/api/topology/v2/{topology_id}/nodes/{nodeIndex}	Request Payload with sliceId.

2.9.2.1. Request

This table shows the URI parameters for the adding slice id to node request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{nodeIndex}	Int	The unique nodeIndex.

Example 2.11. Adding slice ID to Node: JSON request

```
{
  "topoObjectType": "node",
  "topologyIndex": 1,
  "nodeIndex": {nodeIndex},
  "layer": "IP",
  "live": true,
  "name": "11.0.0.102",
  "hostName": "vmx102",
  "protocols": {
    "management": {
      "vendor": "JUNIPER"
    },
    "PCEP": {
      "extensions": {
        "lsp-association-protection": true
      }
    },
    "SR": {
      "capabilities": {
        "unrestrictedFirstHop": false
      }
    }
  },
  "extraIpAddresses": [],
  "slices": [
    {
      "sliceId": 78
    }
  ]
}
```


2.9.3. Delete slice ID from Node

Method	URI	Description
PUT	/traffic-engineering/api/topology/v2/{topology_id}/nodes/{nodeIndex}	Request Payload without sliceId.

2.9.3.1. Request

This table shows the URI parameters for the delete slice id from node request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{nodeIndex}	Int	The unique nodeIndex.

Example 2.12. Delete slice ID from Node: JSON request

```
{
  "topoObjectType": "node",
  "topologyIndex": 1,
  "nodeIndex": {nodeIndex},
  "layer": "IP",
  "live": true,
  "name": "11.0.0.102",
  "hostName": "vmx102",
  "protocols": {
    "management": {
      "vendor": "JUNIPER"
    },
    "PCEP": {
      "extensions": {
        "lsp-association-protection": true
      }
    },
    "SR": {
      "capabilities": {
        "unrestrictedFirstHop": false
      }
    }
  },
  "extraIpAddresses": []
}
```

2.9.4. Bulk adding slice ID to Nodes

Method	URI	Description
PATCH	/traffic-engineering/api/topology/v2/{topology_id}/nodes/bulk	Adding slice id to nodes using patch with op = add

Normal response codes: 200

2.9.4.1. Request

This table shows the URI parameters for the bulk adding slice id to nodes request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 2.13. Bulk adding slice ID to Nodes: JSON request

```
[{
  "nodeIndex": 2,
  "patch": [
    {
      "op": "add",
      "path": "/slices",
      "value": [
        {
          "sliceId": 501
        }
      ]
    }
  ]
},
{
  "nodeIndex": 4,
  "patch": [
    {
      "op": "add",
      "path": "/slices",
      "value": [
        {
          "sliceId": 501
        }
      ]
    }
  ]
}
]
```

2.9.4.2. Response

Example 2.14. Bulk adding slice ID to Nodes: JSON response

```
[
  {
    "topoObjectType": "node",
    "topologyIndex": 1,
    "id": "11.0.0.101",
    "nodeIndex": 2,
```

```

        "layer": "IP",
        "live": true,
        "AutonomousSystem": {
            "asNumber": 11
        },
        "routerId": "11.0.0.101",
        "protocols": {
            "ISIS": {
                "routerId": "11.0.0.101",
                "TERouterId": "11.0.0.101",
                "isoAddress": "0110.0000.0101",
                "area": "490011"
            },
            "NETCONF": {
                "operationalStatus": "Up",
                "operationalState": "Connected",
                "clientAddress": "172.16.18.101",
                "sessionParameters": {
                    "keepalive": 300
                },
                "clientCapabilities": [
                    "http://xml.juniper.net/dmi/system/1.0",
                    "http://xml.juniper.net/netconf/junos/1.0",
                    "urn:ietf:params:netconf:base:1.0",
                    "urn:ietf:params:netconf:capability:candidate:1.0",
                    "urn:ietf:params:netconf:capability:confirmed-commit:1.0",
                    "urn:ietf:params:netconf:capability:url:1.0?scheme=http,ftp,file",
                    "urn:ietf:params:netconf:capability:validate:1.0",
                    "urn:ietf:params:xml:ns:netconf:base:1.0?module=ietf-netconf&revision=2011-06-01",
                    "urn:ietf:params:xml:ns:netconf:capability:candidate:1.0",
                    "urn:ietf:params:xml:ns:netconf:capability:confirmed-commit:1.0",
                    "urn:ietf:params:xml:ns:netconf:capability:url:1.0?scheme=http,ftp,file",
                    "urn:ietf:params:xml:ns:netconf:capability:validate:1.0",
                    "urn:ietf:params:xml:ns:yang:ietf-inet-types?module=ietf-inet-types&revision=2013-07-15",
                    "urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring"
                ]
            },
            "PCEP": {
                "operationalStatus": "Up",
                "pccAddress": "172.16.18.101",
                "sessionParameters": {
                    "keepalive": 30,
                    "deadtimer": 125,
                    "maximumStackDepth": 5
                },
                "peerStatefullCapabilities": {
                    "lspUpdate": true,
                    "lspInstantiation": true,
                    "p2mp": true,
                    "p2mpUpdate": true,
                    "p2mpInstantiation": true,
                    "SRCapability": true
                }
            },
            "management": {

```

```
        "address": "172.16.18.101",
        "vendor": "JUNIPER",
        "operatingSystem": "junos",
        "operatingSystemVersion": "20.1R1.11"
    },
    "SR": {
        "enabled": true,
        "SRGBs": [
            {
                "start": 1000,
                "range": 32
            }
        ]
    },
    "hostName": "vmx101",
    "configurationDataSource": {
        "latest": {
            "href": "/traffic-engineering/api/topology/v2/1/nodes/bulk/
device/configurationDataSource/latest"
        }
    },
    "prefixes": [
        {
            "topoObjectType": "prefix",
            "address": "11.0.0.101",
            "length": 32,
            "SR": {
                "index": 1,
                "algo": 0,
                "flags": 64
            },
            "role": "routerId"
        }
    ],
    "slices": [
        {
            "sliceId": 501
        }
    ]
},
{
    "topoObjectType": "node",
    "topologyIndex": 1,
    "id": "11.0.0.103",
    "nodeIndex": 4,
    "layer": "IP",
    "live": true,
    "AutonomousSystem": {
        "asNumber": 11
    },
    "routerId": "11.0.0.103",
    "protocols": {
        "ISIS": {
            "routerId": "11.0.0.103",
            "TERouterId": "11.0.0.103",
            "isoAddress": "0110.0000.0103",
            "area": "490011"
        },
        "NETCONF": {
```

```

        "operationalStatus": "Up",
        "operationalState": "Connected",
        "clientAddress": "172.16.18.103",
        "sessionParameters": {
            "keepalive": 300
        },
        "clientCapabilities": [
            "http://xml.juniper.net/dmi/system/1.0",
            "http://xml.juniper.net/netconf/junos/1.0",
            "urn:ietf:params:netconf:base:1.0",
            "urn:ietf:params:netconf:capability:candidate:1.0",
            "urn:ietf:params:netconf:capability:confirmed-commit:1.0",
            "urn:ietf:params:netconf:capability:url:1.0?scheme=http,
ftp,file",
            "urn:ietf:params:netconf:capability:validate:1.0",
            "urn:ietf:params:xml:ns:netconf:base:1.0?module=ietf-
netconf&revision=2011-06-01",
            "urn:ietf:params:xml:ns:netconf:capability:candidate:1.0",
            "urn:ietf:params:xml:ns:netconf:capability:confirmed-
commit:1.0",
            "urn:ietf:params:xml:ns:netconf:capability:url:1.0?scheme=
http,ftp,file",
            "urn:ietf:params:xml:ns:netconf:capability:validate:1.0",
            "urn:ietf:params:xml:ns:yang:ietf-inet-types?module=ietf-
inet-types&revision=2013-07-15",
            "urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring"
        ]
    },
    "PCEP": {
        "operationalStatus": "Up",
        "pccAddress": "172.16.18.103",
        "sessionParameters": {
            "keepalive": 30,
            "deadtimer": 125,
            "maximumStackDepth": 5
        },
        "peerStatefullCapabilities": {
            "lspUpdate": true,
            "lspInstantiation": true,
            "p2mp": true,
            "p2mpUpdate": true,
            "p2mpInstantiation": true,
            "SRCapability": true
        }
    },
    "management": {
        "address": "172.16.18.103",
        "vendor": "JUNIPER",
        "operatingSystem": "junos",
        "operatingSystemVersion": "20.1R1.11"
    },
    "SR": {
        "enabled": true,
        "SRGBs": [
            {
                "start": 1000,
                "range": 32
            }
        ]
    }
}

```

```
    },
    "hostname": "vmx103",
    "configurationDataSource": {
      "latest": {
        "href": "/traffic-engineering/api/topology/v2/1/nodes/bulk/
device/configurationDataSource/latest"
      }
    },
    "prefixes": [
      {
        "topoObjectType": "prefix",
        "address": "11.0.0.103",
        "length": 32,
        "SR": {
          "index": 3,
          "algo": 0,
          "flags": 64
        },
        "role": "routerId"
      }
    ],
    "slices": [
      {
        "sliceId": 501
      }
    ]
  }
]
```

2.9.5. Bulk deleting slice ID from Nodes

Method	URI	Description
PATCH	/traffic-engineering/api/topology/v2/{topology_id}/nodes/bulk	Deleting slice id from nodes using patch with op = remove

Normal response codes: 200

2.9.5.1. Request

This table shows the URI parameters for the bulk deleting slice id from nodes request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 2.15. Bulk deleting slice ID from Nodes: JSON request

```
[{
  "nodeIndex": 2,
  "patch": [
    {
      "op": "remove",
      "path": "/slices",
      "value": []
    }
  ]
},
{
  "nodeIndex": 4,
  "patch": [
    {
      "op": "remove",
      "path": "/slices",
      "value": []
    }
  ]
}]
```

2.9.5.2. Response

Example 2.16. Bulk deleting slice ID from Nodes: JSON response

```
[
  {
    "topoObjectType": "node",
    "topologyIndex": 1,
    "id": "11.0.0.101",
    "nodeIndex": 2,
    "layer": "IP",
    "live": true,
    "AutonomousSystem": {
      "asNumber": 11
    },
    "routerId": "11.0.0.101",
    "protocols": {
      "ISIS": {
```

```

        "routerId": "11.0.0.101",
        "TERouterId": "11.0.0.101",
        "isoAddress": "0110.0000.0101",
        "area": "490011"
    },
    "NETCONF": {
        "operationalStatus": "Up",
        "operationalState": "Connected",
        "clientAddress": "172.16.18.101",
        "sessionParameters": {
            "keepalive": 300
        },
        "clientCapabilities": [
            "http://xml.juniper.net/dmi/system/1.0",
            "http://xml.juniper.net/netconf/junos/1.0",
            "urn:ietf:params:netconf:base:1.0",
            "urn:ietf:params:netconf:capability:candidate:1.0",
            "urn:ietf:params:netconf:capability:confirmed-commit:1.0",
            "urn:ietf:params:netconf:capability:url:1.0?scheme=http,ftp,file",
            "urn:ietf:params:netconf:capability:validate:1.0",
            "urn:ietf:params:xml:ns:netconf:base:1.0?module=ietf-netconf&revision=2011-06-01",
            "urn:ietf:params:xml:ns:netconf:capability:candidate:1.0",
            "urn:ietf:params:xml:ns:netconf:capability:confirmed-commit:1.0",
            "urn:ietf:params:xml:ns:netconf:capability:url:1.0?scheme=http,ftp,file",
            "urn:ietf:params:xml:ns:netconf:capability:validate:1.0",
            "urn:ietf:params:xml:ns:yang:ietf-inet-types?module=ietf-inet-types&revision=2013-07-15",
            "urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring"
        ]
    },
    "PCEP": {
        "operationalStatus": "Up",
        "pccAddress": "172.16.18.101",
        "sessionParameters": {
            "keepalive": 30,
            "deadtimer": 125,
            "maximumStackDepth": 5
        },
        "peerStatefullCapabilities": {
            "lspUpdate": true,
            "lspInstantiation": true,
            "p2mp": true,
            "p2mpUpdate": true,
            "p2mpInstantiation": true,
            "SRCapability": true
        }
    },
    "management": {
        "address": "172.16.18.101",
        "vendor": "JUNIPER",
        "operatingSystem": "junos",
        "operatingSystemVersion": "20.1R1.11"
    },
    "SR": {
        "enabled": true,
        "SRGBs": [

```



```
        {
            "start": 1000,
            "range": 32
        }
    ]
},
"hostName": "vmx101",
"configurationDataSource": {
    "latest": {
        "href": "/traffic-engineering/api/topology/v2/1/nodes/bulk/
device/configurationDataSource/latest"
    }
},
"prefixes": [
    {
        "topoObjectType": "prefix",
        "address": "11.0.0.101",
        "length": 32,
        "SR": {
            "index": 1,
            "algo": 0,
            "flags": 64
        },
        "role": "routerId"
    }
]
},
{
    "topoObjectType": "node",
    "topologyIndex": 1,
    "id": "11.0.0.103",
    "nodeIndex": 4,
    "layer": "IP",
    "live": true,
    "AutonomousSystem": {
        "asNumber": 11
    },
    "routerId": "11.0.0.103",
    "protocols": {
        "ISIS": {
            "routerId": "11.0.0.103",
            "TERouterId": "11.0.0.103",
            "isoAddress": "0110.0000.0103",
            "area": "490011"
        },
        "NETCONF": {
            "operationalStatus": "Up",
            "operationalState": "Connected",
            "clientAddress": "172.16.18.103",
            "sessionParameters": {
                "keepalive": 300
            },
            "clientCapabilities": [
                "http://xml.juniper.net/dmi/system/1.0",
                "http://xml.juniper.net/netconf/junos/1.0",
                "urn:ietf:params:netconf:base:1.0",
                "urn:ietf:params:netconf:capability:candidate:1.0",
                "urn:ietf:params:netconf:capability:confirmed-commit:1.0",
```

```

        "urn:ietf:params:netconf:capability:url:1.0?scheme=http,
ftp,file",
        "urn:ietf:params:netconf:capability:validate:1.0",
        "urn:ietf:params:xml:ns:netconf:base:1.0?module=ietf-
netconf&revision=2011-06-01",
        "urn:ietf:params:xml:ns:netconf:capability:candidate:1.0",
        "urn:ietf:params:xml:ns:netconf:capability:confirmed-
commit:1.0",
        "urn:ietf:params:xml:ns:netconf:capability:url:1.0?scheme=
http,ftp,file",
        "urn:ietf:params:xml:ns:netconf:capability:validate:1.0",
        "urn:ietf:params:xml:ns:yang:ietf-inet-types?module=ietf-
inet-types&revision=2013-07-15",
        "urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring"
    ]
},
"PCEP": {
    "operationalStatus": "Up",
    "pccAddress": "172.16.18.103",
    "sessionParameters": {
        "keepalive": 30,
        "deadtimer": 125,
        "maximumStackDepth": 5
    },
    "peerStatefullCapabilities": {
        "lspUpdate": true,
        "lspInstantiation": true,
        "p2mp": true,
        "p2mpUpdate": true,
        "p2mpInstantiation": true,
        "SRCapability": true
    }
},
"management": {
    "address": "172.16.18.103",
    "vendor": "JUNIPER",
    "operatingSystem": "junos",
    "operatingSystemVersion": "20.1R1.11"
},
"SR": {
    "enabled": true,
    "SRGBs": [
        {
            "start": 1000,
            "range": 32
        }
    ]
},
},
"hostName": "vmx103",
"configurationDataSource": {
    "latest": {
        "href": "/traffic-engineering/api/topology/v2/1/nodes/bulk/
device/configurationDataSource/latest"
    }
},
"prefixes": [
    {
        "topoObjectType": "prefix",
        "address": "11.0.0.103",

```

```
        "length": 32,  
        "SR": {  
            "index": 3,  
            "algo": 0,  
            "flags": 64  
        },  
        "role": "routerId"  
    }  
]  
]
```

2.9.6. Adding slice ID to Link

Method	URI	Description
PUT	/traffic-engineering/api/topology/v2/{topology_id}/links/{linkIndex}	Request Payload with sliceld.

2.9.6.1. Request

This table shows the URI parameters for the adding slice id to link request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{linkIndex}	Int	The unique linkIndex.

Example 2.17. Adding slice ID to Link: JSON request

```
{
  "endA": {
    "TEcolor": 6,
    "bandwidth": 10000000,
    "designParameters": {
      "canFail": true
    },
    "ipv4Address": {
      "topoObjectType": "ipv4",
      "address": "11.102.105.1"
    },
    "node": {
      "id": "11.0.0.102",
      "topoObjectType": "node",
      "topologyIndex": 1
    },
    "protocols": {
      "ISIS": {
        "metricL2": 10,
        "TEMetric": 10
      },
      "RSVP": {
        "bandwidth": 10000000
      }
    },
    "topoObjectType": "interface"
  },
  "endZ": {
    "TEcolor": 6,
    "bandwidth": 10000000,
    "designParameters": {
      "canFail": true
    },
    "ipv4Address": {
      "topoObjectType": "ipv4",
      "address": "11.102.105.2"
    },
    "node": {
```

```
    "id": "11.0.0.105",
    "topoObjectType": "node",
    "topologyIndex": 1
  },
  "protocols": {
    "ISIS": {
      "metricL2": 10,
      "TEMetric": 10
    },
    "RSVP": {
      "bandwidth": 10000000
    }
  },
  "topoObjectType": "interface"
},
"linkIndex": {linkIndex},
"live": true,
"name": "L11.0.0.102_11.102.105.1_11.0.0.105_11.102.105.2",
"operationalStatus": "Up",
"slices": [
  {
    "sliceId": 78
  }
],
"topoObjectType": "link",
"topologyIndex": 1
}
```

2.9.7. Delete slice ID from Link

Method	URI	Description
PUT	/traffic-engineering/api/topology/v2/{topology_id}/links/{linkIndex}	Request Payload without sliced.

2.9.7.1. Request

This table shows the URI parameters for the delete slice id from link request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{linkIndex}	Int	The unique linkIndex.

Example 2.18. Delete slice ID from Link: JSON request

```
{
  "endA": {
    "TEcolor": 6,
    "bandwidth": 10000000,
    "designParameters": {
      "canFail": true
    },
    "ipv4Address": {
      "topoObjectType": "ipv4",
      "address": "11.102.105.1"
    },
    "node": {
      "id": "11.0.0.102",
      "topoObjectType": "node",
      "topologyIndex": 1
    },
    "protocols": {
      "ISIS": {
        "metricL2": 10,
        "TEMetric": 10
      },
      "RSVP": {
        "bandwidth": 10000000
      }
    },
    "topoObjectType": "interface"
  },
  "endZ": {
    "TEcolor": 6,
    "bandwidth": 10000000,
    "designParameters": {
      "canFail": true
    },
    "ipv4Address": {
      "topoObjectType": "ipv4",
      "address": "11.102.105.2"
    },
    "node": {
```

```
    "id": "11.0.0.105",
    "topoObjectType": "node",
    "topologyIndex": 1
  },
  "protocols": {
    "ISIS": {
      "metricL2": 10,
      "TEMetric": 10
    },
    "RSVP": {
      "bandwidth": 10000000
    }
  },
  "topoObjectType": "interface"
},
"linkIndex": {linkIndex},
"live": true,
"name": "L11.0.0.102_11.102.105.1_11.0.0.105_11.102.105.2",
"operationalStatus": "Up",
"topoObjectType": "link",
"topologyIndex": 1
}
```

2.9.8. Adding LSP with slice

Method	URI	Description
POST	/traffic-engineering/api/topology/v2/{topology_id}/te-lsps	Slice Include All, Slice Include Any, or Slice Include Exclude.

3 options to add slice:

- Slice include all: This means that the path selected by Paragon should have all the slice id's associated with it. For example, if I want slice id 1, slice id 2, slice id 3 as part of the LSP then all the 3 slice id's should be present to get the LSP up. If there is no path with all 3 slice id's then LSP will be in an unknown state.
- Slice include any: If there are multiple slice id's, for instance, slice id 1, slice id 2, slice id 3, etc. then the decision to find the shortest path will be calculated which includes any of these slice by Paragon.
- Slice Exclude: As the name indicates, Paragon will route an LSP which will exclude the path with the mentioned slice id.

Network Slicing is working for these combinations as per Davinci 21.2 release:

ControlType can be as usual of 3 types i.e. PCEInitiated, PCC and Delegated. Provisioning-Type can be RSVP or SR. The main change is to tag LSP with slice ID with any of the 3 types i.e. slice include all, slice include any, slice exclude.

2.9.8.1. Request

This table shows the URI parameters for the adding lsp with slice request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 2.19. Adding LSP with slice: JSON request

Example 1:

- Provisioning type: RSVP
- Control type: PCC
- Slice option: Slice Include Any - 78

```
{
  "name": "netconf_rsvp",
  "creationConfigurationMethod": "NETCONF",
  "pathType": "primary",
  "provisioningType": "RSVP",
  "from": {
    "address": "11.0.0.101",
    "topoObjectType": "ipv4"
  },
}
```



```
"to": {
  "address": "11.0.0.103",
  "topoObjectType": "ipv4"
},
"plannedProperties": {
  "adminStatus": "Up",
  "bandwidth": "0",
  "holdingPriority": 7,
  "setupPriority": 7,
  "design": {
    "adminGroups": {},
    "slices": {
      "sliceIncludeAny": [
        {
          "sliceId": 78
        }
      ]
    }
  },
  "routingMethod": "routeByDevice"
},
"policy": {
  "bandwidthSizing": {
    "enabled": false
  }
}
}
```

Example 2:

- Provisioning type: SR
- Control type: PCEP
- Slice Include Any - 78, Slice Exclude - 55

```
{
  "name": "pcep_sr",
  "creationConfigurationMethod": "PCEP",
  "pathType": "primary",
  "provisioningType": "SR",
  "from": {
    "address": "11.0.0.101",
    "topoObjectType": "ipv4"
  },
  "to": {
    "address": "11.0.0.103",
    "topoObjectType": "ipv4"
  },
  "plannedProperties": {
    "adminStatus": "Up",
    "bandwidth": "0",
    "holdingPriority": 7,
    "setupPriority": 7,
    "design": {
      "adminGroups": {},
      "slices": {
        "sliceExclude": [
          {
            "sliceId": 55
          }
        ]
      }
    }
  }
}
```

```
    }  
  ],  
  "sliceIncludeAny": [  
    {  
      "sliceId": 78  
    }  
  ]  
},  
"routingMethod": "default"  
},  
"policy": {  
  "bandwidthSizing": {  
    "enabled": false  
  }  
}  
}  
}
```

3. Paragon PathFinder API

The Juniper Networks Paragon PathFinder API enables querying of topology, management of topology planning parameters, and management of TE-LSP.

P2MP groups and SRLG/Facilities are available through the API since NorthStar 4.2 , and you can use a notification from Paragon Pathfinder to dynamically push updates to REST API clients.

Use the P2MP REST API calls to create P2MP groups and manage the following P2MP group common attributes:

- Bandwidth
- Setup/Holding Priority
- Coloring Attributes

NOTE: P2MP groups cannot be created nor managed by TE-LSPs.

You can use the TE-LSP API calls or the new P2MP REST API calls to delete the P2MP leaves. For more information, see the [P2MP section](#)

You access the API using the configured Paragon address (as in `paragon.example.net`), through HTTPS port 443, on base URLs like the following:

- `https://paragon.example.net/traffic-engineering/api/`

All of the Paragon PathFinder RESTful APIs use JSON-formatted data that conforms to the [JSON-Schema](#) specification. The main schema is [topology_v2.json](#) , and it contains links to the other data types. All schema can be found [in this archive](#). Common data types are described in [common.json](#) . The following kinds of data and functionality are accessible and can be manipulated using the RESTful API:

- [Nodes](#): communication endpoints.
- [Links](#): lines or channels through which data is transmitted.
- [Topology](#): A collection of nodes and links.
- [TE-LSPs](#): Traffic-engineered label-switched paths.
- [Demands](#).
- [P2MP groups](#).
- [TE-Containers](#).
- [SRLGs/Facilities](#).
- [BGP Routes](#).
- [Analytics data](#).
- [Maintenances](#).
- [Path Computation](#).

- [Task scheduler](#).
- [Configuration](#).
- [VRFs](#).
- [IPE Policies](#).

You will find some example code associated with the API reference information. In addition, several examples that illustrate common usage can be found [later in this document](#). In our example requests, we use HTTP, however the same functionality is available over HTTPS.

Existing REST APIs that reference `plannedProperties/operationalStatus` needs to be changed to `plannedProperties/routingStatus` as backwards compatibility will not be maintained from this release going forward. The reason for the change is based on the feedback from users that under `plannedProperties`, `routingStatus` would be more clear than `operationalStatus`. For an example, please see "LSP update example" under TE-LSPs events section below.

Method	URI	Description
Notification API		
Topology		
GET	/topology/v2	Returns a list of the topologies that are available.
GET	/topology/v2/{topology_id}	Lists topological elements
DELETE	/topology/v2/{topology_id}	Deletes all of the topology planned data. The information acquired through BGP-LS reappears immediately.
GET	/topology/v2/{topology_id}/stream	See EventSource for format. The notifications send on that stream are only topologyEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).
	/topology/v2/{topology_id}/status	
GET	/topology/v2/{topology_id}/status/pce	Returns the status of the PCEP protocol adapter component.
GET	/topology/v2/{topology_id}/status/topologyAcquisition	Returns the status of the Network Topology Acquisition Daemon (NTAD, BGP-LSP protocol adapter) component.
GET	/topology/v2/{topology_id}/status/pathComputationServer	Returns the status of the Path Computation Server component.
GET	/topology/v2/{topology_id}/status/transportTopologyAcquisition	Returns the status of the Transport Network Topology Acquisition Daemon.
GET	/topology/v2/{topology_id}/status/analyticsCollection	Returns the status of the analytics data collection.
GET	/topology/v2/{topology_id}/status/analyticsDatabase	Returns the status of the analytics data collection database.
Node		
GET	/topology/v2/{topology_id}/nodes	Returns a full list of nodes.
POST	/topology/v2/{topology_id}/nodes	Creates a node using the following schema: node_v2.json#/definitions/createNode .
GET	/topology/v2/{topology_id}/nodes/search{?name,hostName,AS,query-Type}	Searches the list of nodes for specific URI parameters. For example, search?name=62.0.0.101 must return one node.
GET	/topology/v2/{topology_id}/nodes/stream	See EventSource for format. The notifications send on that stream are only nodeEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).

Method	URI	Description
GET	/topology/v2/{topology_id}/nodes/{nodeIndex}	Returns details for a node.
PUT	/topology/v2/{topology_id}/nodes/{nodeIndex}	Updates a node using the following schema: node_v2.json#/definitions/updateNode
PATCH	/topology/v2/{topology_id}/nodes/{nodeIndex}	Updates a node using a RFC6902 patch: json-patch.json . The result of the patch must conform to node_v2.json#/definitions/updateNode . The REST server remove all operational parameters like operationalStatus, ..etc. .
DELETE	/topology/v2/{topology_id}/nodes/{nodeIndex}	Deletes a node. (You cannot delete a live node; it reappears on the next update from Topology server.)
GET	/topology/v2/{topology_id}/nodes/{nodeIndex}/history{?start,end}	Returns the event history for a node.
POST	/topology/v2/{topology_id}/nodes/bulk	Creates several Nodes using the following JSON schema: node_v2.json#/definitions/createNodeList .
PUT	/topology/v2/{topology_id}/nodes/bulk	Updates several Nodes using the following JSON schema: node_v2.json#/definitions/nodeListUpdate .
PATCH	/topology/v2/{topology_id}/nodes/bulk	Updates several Nodes using the following JSON schema: node_v2.json#/definitions/nodeListPatch .
DELETE	/topology/v2/{topology_id}/nodes/bulk	Deletes a list of Nodess. The payload must conform to node_v2.json#/definitions/nodeListDelete
Links		
GET	/topology/v2/{topology_id}/links	Returns a full list of links.
POST	/topology/v2/{topology_id}/links	Creates a link using optional and required parameters defined in the following schema: link_v2.json#/definitions/createLink .
GET	/topology/v2/{topology_id}/links/search{?name,address,queryType}	Searches the link list based on URI parameters. For example, search?name=62.101.105 must return one Link.
GET	/topology/v2/{topology_id}/links/stream	See EventSource for format. The notifications send on that stream are only linkEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).
GET	/topology/v2/{topology_id}/links/{linkIndex}	Returns the details for a link.
PUT	/topology/v2/{topology_id}/links/{linkIndex}	Updates a planned link using the optional and required parameters defined in the following schema: link_v2.json#/definitions/updateLink .
PATCH	/topology/v2/{topology_id}/links/{linkIndex}	Updates a link using a RFC6902 patch: json-patch.json . The result of the patch must conform to link_v2.json#/definitions/updateLink . The REST server remove all operational parameters like operationalStatus, ..etc. .
DELETE	/topology/v2/{topology_id}/links/{linkIndex}	Deletes a link. Live links reappear on the next update from the Topology server.
GET	/topology/v2/{topology_id}/links/utilization	Returns a list of links only containing the endA/endZ utilization
GET	/topology/v2/{topology_id}/links/{linkIndex}/history{?start,end}	Returns the history for a Link.
TE-LSPs		
GET	/topology/v2/{topology_id}/te-lsps	Returns the TE-LSP list.
POST	/topology/v2/{topology_id}/te-lsps	Creates a TE-LSP using the following JSON schema: lsp.json#/definitions/createLSP . For example, protection and custom service mapping parameters set.
GET	/topology/v2/{topology_id}/te-lsps/search{?name,from,operationalStatus,queryType}	Performs a search in the LSP list based on the URI parameters. For example, "search?name=62.101.105" returns one link.

Method	URI	Description
GET	/topology/v2/{topology_id}/te-lsps/stream	See EventSource for format. The notifications send on that stream are only lspEvent or lspTopologyEvent. The data will contain a JSON document (see Paragon Pathfinder Notification API).
GET	/topology/v2/{topology_id}/te-lsps/{lspIndex}	Returns the details for a TE-LSP.
PUT	/topology/v2/{topology_id}/te-lsps/{lspIndex}	Updates a TE-LSP using the JSON schema: lsp.json#/definitions/updateLSP .
PATCH	/topology/v2/{topology_id}/te-lsps/{lspIndex}	Updates a TE-LSP using a RFC6902 patch: json-patch.json . The result of the patch must conform to lsp.json#/definitions/updateLsp . The REST server remove all operational parameters like operationalStatus, ..etc. . Using an empty patch (empty list) will result in the LSP to be re-provisioned without parameter changed.
DELETE	/topology/v2/{topology_id}/te-lsps/{lspIndex}	Deletes a TE-LSP. This function is supported only on the PCE-initiated LSPs. PCC-controlled and PCC-delegated LSPs cannot be deleted from Paragon Pathfinder. They must be deleted in the node.
POST	/topology/v2/{topology_id}/te-lsps/bulk	Creates several TE-LSPs using the following JSON schema: lsp.json#/definitions/createLSPList .
PUT	/topology/v2/{topology_id}/te-lsps/bulk	Updates several TE-LSPs using the following JSON schema: lsp.json#/definitions/lspListUpdate .
PATCH	/topology/v2/{topology_id}/te-lsps/bulk	Updates several TE-LSPs using the following JSON schema: lsp.json#/definitions/lspListPatch .
DELETE	/topology/v2/{topology_id}/te-lsps/bulk	Deletes a list of TE-LSPs. This function is supported only on the PCE-initiated LSPs. PCC-controlled and PCC-delegated LSPs cannot be deleted from Paragon Pathfinder. They must be deleted in the node.b The payload must conform to lsp.json#/definitions/lspListDelete
GET	/topology/v2/{topology_id}/te-lsps/{lspIndex}/history{?start, end}	Returns the history for a TE-LSP.
Demands		
GET	/topology/v2/{topology_id}/demands	Returns a full list of Demands.
POST	/topology/v2/{topology_id}/demands	Create a demand using the following JSON schema: demands.json#/definitions/createDemand .
GET	/topology/v2/{topology_id}/demands/stream	See EventSource for format. The notifications send on that stream are only demandEvent and demandTopologyEvent. The data will contain a JSON document (see Paragon Pathfinder Notification API).
GET	/topology/v2/{topology_id}/demands/{demandIndex}	Return the details of a specified demand.
PUT	/topology/v2/{topology_id}/demands/{demandIndex}	Modify a specific demand using the following JSON schema: demands.json#/definitions/updateDemand .
PATCH	/topology/v2/{topology_id}/demands/{demandIndex}	Patch a specific demand using a RFC6902 patch: json-patch.json . The result of the patch must conform to demands.json#/definitions/updateDemand . The REST server remove all operational parameters like operationalStatus, ..etc. .
DELETE	/topology/v2/{topology_id}/demands/{demandIndex}	Delete a specific demand.
POST	/topology/v2/{topology_id}/demands/bulk	Create a set of demands using the following JSON schema: demands.json#/definitions/createDemandList . The resulting list of demands (after the patch is applied) must also conform to demands.json#/definitions/demandListUpdate . The return code indicates the request acceptance.

Method	URI	Description
PUT	/topology/v2/{topology_id}/demands/bulk	Update a set of demands using the following JSON schema: demands.json#/definitions/demandListUpdate . The return code indicates the request acceptance.
PATCH	/topology/v2/{topology_id}/demands/bulk	Patch a set of demands using the following JSON schema: demands.json#/definitions/demandListPatch . The return code indicates the request acceptance.
DELETE	/topology/v2/{topology_id}/demands/bulk	Delete a set of demands. The return code indicates the request acceptance.
Task scheduler		
GET	/scheduler/v2/tasklist	Returns a full list of Tasks as per JSON schema: scheduler.json#/definitions/taskList
GET	/scheduler/v2/tasks	Returns a list of Tasks by filter task-type(?tasktype={tasktypeFilter}) as per JSON schema: scheduler.json#/definitions/taskList
POST	/scheduler/v2/updatetask	Update a task using the following JSON schema: scheduler.json#/definitions/updateTask . Returns response as per JSON schema: scheduler.json#/definitions/responseObject .
POST	/scheduler/v2/removeaddtask	Removes all the existing tasks with given taskType and adds a new Task using the following JSON schema: scheduler.json#/definitions/updateTask . Returns response as per JSON schema: scheduler.json#/definitions/responseObject .
POST	/scheduler/v2/deletetask	Delete one or more tasks using the following JSON schema: scheduler.json#/definitions/deleteTask . Returns response as per JSON schema: scheduler.json#/definitions/responseObject .
GET	/scheduler/v2/taskshistory/{taskId}	Returns task execution status history in HTML table format as per JSON schema: scheduler.json#/definitions/tasksHistoryList
GET	/scheduler/v2/taskstatus/{taskId}	Returns task execution status history as per JSON schema: scheduler.json#/definitions/tasksStatusList
Configuration		
GET	/config/config	Return a json object containing component configurations configuration.json#/definitions/northstar
POST	/config/config	Updates all component configurations using the following JSON schema: configuration.json#/definitions/northstar-post . Returns response as per JSON schema: configuration.json#/definitions/responseObject .
PATCH	/config/config	Updates configuration similar to plain-patch mechanism described in RFC7396. The content of the request body will be merged into Paragon PathFinder configuration. The limitation of this method is that it is not supporting deletion of attributes, for that the DELETE method should be used. Two known deviations from RFC are: arrays types are merged instead of getting replaced and to unset the empty types DELETE operation has to be used. config.json#/definitions/northstar-post . Returns response as per JSON schema: config.json#/definitions/responseObject .
DELETE	/config/config	Delete component configurations using the following JSON schema: config.json#/definitions/northstar . Returns response as per JSON schema: config.json#/definitions/responseObject .
Device Profiles		
GET	/netconf/v2/profiles	Gets all profiles.
POST	/netconf/v2/profiles	Creates new profiles.

Method	URI	Description
PUT	/netconf/v2/profiles	Updates profiles.
DELETE	/netconf/v2/profiles	Deletes profiles.
POST	/netconf/v2/collection/liveNetwork	Creates a new collection.
GET	/netconf/v2/collection/{id}	Gets the Status of a collection job.
TE-Containers		
GET	/topology/v2/{topology_id}/te-containers	List all TE-containers.
POST	/topology/v2/{topology_id}/te-containers	Creates a container.
GET	/topology/v2/{topology_id}/te-containers/stream	See EventSource for format. The notifications send on that stream are only teContainerEvent. The data will contain a JSON document (see Paragon Pathfinder Notification API).
GET	/topology/v2/{topology_id}/te-containers/{containerIndex}	Gets a TE Container.
PUT	/topology/v2/{topology_id}/te-containers/{containerIndex}	Updates a TE Container
PATCH	/topology/v2/{topology_id}/te-containers/{containerIndex}	Updates a TE Container using an RFC6902 document
DELETE	/topology/v2/{topology_id}/te-containers/{containerIndex}	Deletes a container.
POST	/topology/v2/{topology_id}/te-containers/bulk	Create a list of TE Container
PUT	/topology/v2/{topology_id}/te-containers/bulk	Modify a list of TE Container
PATCH	/topology/v2/{topology_id}/te-containers/bulk	Modify a list of TE Container using a PATCH
DELETE	/topology/v2/{topology_id}/te-containers/bulk	Deletes a list of containers.
Facilities		
GET	/topology/v2/{topology_id}/facilities	Gets all Facilities.
POST	/topology/v2/{topology_id}/facilities	Creates a facility.
GET	/topology/v2/{topology_id}/facilities/stream	See EventSource for format. The notifications send on that stream are only facilityEvent. The data will contain a JSON document (see Paragon Pathfinder Notification API).
GET	/topology/v2/{topology_id}/facilities/{facilityIndex}	Gets a Facility.
PUT	/topology/v2/{topology_id}/facilities/{facilityIndex}	Updates a facility
DELETE	/topology/v2/{topology_id}/facilities/{facilityIndex}	Deletes a facility.
P2MP		
GET	/topology/v2/{topology_id}/p2mp	Returns a full list of P2MP groups.
POST	/topology/v2/{topology_id}/p2mp	Create a P2MP group using the following schema: p2mp.json#/definitions/createP2mpGroup . The API request contains the destination list and the common planned properties. The common planned properties may include plannedProperties with flow mapping parameters.
GET	/topology/v2/{topology_id}/p2mp/stream	See EventSource for format. The notifications send on that stream are only p2mpEvent. The data will contain

Method	URI	Description
		a JSON document (see Paragon PathFinder Notification API).
GET	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}	Returns the details of a specified P2MP group.
PUT	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}	Modify a specific P2MP group, if the LSPs are not specified, the common properties are changed, not the LSPs. If the LSPs are specified, the set of LSPs will be modified. i.e LSP that is not in the new list will be remove, the other will be added/updated".
PATCH	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}	Modify a specific P2MP group using a PATCH document (RFC6906). The operation has the same behavior as the PUT method. An empty PATCH will re-provision the complete tree.
DELETE	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}	No response is received from the REST API unless an error occurs.
DELETE	/topology/v2/{topology_id}/p2mp/bulk	No response is received from the REST API unless an error occurs.
POST	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}	The POST URL accepts a list of new leaves. Use the following schema to create a P2MP leaf: p2mp.json#/definitions/createP2mpLeavesList .
GET	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}/{lspIndex}	Returns the details of a specified P2MP leaf.
DELETE	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}/{lspIndex}	No response is received from the REST API unless an error occurs.
DELETE	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}/bulk	No response is received from the REST API unless an error occurs.
VRF		
GET	/topology/v2/{topology_id}/vrfs	Returns the list of VRF summary: which node have vrf information and how many vrf are present.
GET	/topology/v2/{topology_id}/vrfs/{nodeIndex}	Returns the list of discovered VRF for that node.
BGP Routes		
GET	/topology/v2/{topology_id}/routes	List all nodes with BGP routes and the number of routes for that node.
GET	/topology/v2/{topology_id}/routes/{nodeIndex}	Gets the routes maintained by Paragon PathFinder for a node.
IPE Policies		
GET	/topology/v2/{topology_id}/ipe/policy	Get all IPE Policies
POST	/topology/v2/{topology_id}/ipe/policy	Create IPE Policies
PUT	/topology/v2/{topology_id}/ipe/policy	Update IPE Policies
DELETE	/topology/v2/{topology_id}/ipe/policy	Delete IPE Policies
GET	/topology/v2/{topology_id}/ipe/policy/{ipePolicyId}	Get single IPE policy
Analytics API		
POST	/statistics/v2/demands/bulk	The POST request accepts a JSON object describing the query parameters. - interval (string): Time duration for instance 1d, 1y, 24h,. - counter : Either a single counter or an array of counters. .

Method	URI	Description
		- aggregation (string): Aggregation function to use to aggregate values.the value must be one of: - avg - max - min - cardinality - sum - startTime (Either string or integer): Date time. . - demand (array): Array of demand objects. - endTime (Either string or integer): Date time. .
POST	/statistics/v2/interfaces/de-lay/{device_name}/{interface_name}	The POST request accepts a JSON object describing the query parameters. - counter (string): Name of the counter to aggregate by.. - endTime (Either string or integer): Date time. . - aggregation (string): Aggregation function to use to aggregate values.the value must be one of: - avg - max - min - cardinality - sum - startTime (Either string or integer): Date time. . - interval (string): Time duration for instance 1d, 1y, 24h,.
POST	/statistics/v2/interfaces/bulk	The POST request accepts a JSON object describing the query parameters. - interval (string): Time duration for instance 1d, 1y, 24h,. - counter : Either a single counter or an array of counters. . - aggregation (string): Aggregation function to use to aggregate values.the value must be one of: - avg - max - min - cardinality - sum - startTime (Either string or integer): Date time. . - interface (array): Array of interface objects.

Method	URI	Description
		• endTime (Either string or integer): Date time. .
POST	/statistics/v2/interfaces/fields	The POST request accepts a JSON object describing the query parameters. • endTime (Either string or integer): Date time. . • startTime (Either string or integer): Date time. .
POST	/statistics/v2/interfaces/top	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • device (array): Array of node objects. • endTime (Either string or integer): Date time. . • size (integer): Max number of interfaces top return.
POST	/statistics/v2/interfaces/child-traffic/{device_name}/{interface_name}	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,.
POST	/statistics/v2/interfaces/bulkdelay	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,.

Method	URI	Description
		• counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interface (array): Array of interface objects. • endTime (Either string or integer): Date time. . .
POST	/statistics/v2/interfaces/traffic	The POST request accepts a JSON object describing the query parameters. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,. .
POST	/statistics/v2/interfaces/traffic/{device_name}	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,.

Method	URI	Description
POST	/statistics/v2/interfaces/traffic/{device_name}/{interface_name}	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,.
POST	/statistics/v2/interfaces/topdelay	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • device (array): Array of node objects. • endTime (Either string or integer): Date time. . • size (integer): Max number of interfaces top return.
POST	/statistics/v2/childtraffic/bulk	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min

Method	URI	Description
		• cardinality • sum • startTime (Either string or integer): Date time. . • interface (array): Array of interface objects. • endTime (Either string or integer): Date time. . .
POST	/statistics/v2/jnxcos/bulk	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interface (array): Array of interface objects. • endTime (Either string or integer): Date time. . .
POST	/statistics/v2/delay/fields	The POST request accepts a JSON object describing the query parameters. • endTime (Either string or integer): Date time. . • startTime (Either string or integer): Date time. . .
POST	/statistics/v2/te-lsps/bulk	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • startTime (Either string or integer): Date time. . • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality

Method	URI	Description
		• sum • lsp (array): Array of lsp objects. • endTime (Either string or integer): Date time. .
POST	/statistics/v2/te-lsps/fields	The POST request accepts a JSON object describing the query parameters. • endTime (Either string or integer): Date time. . • startTime (Either string or integer): Date time. .
POST	/statistics/v2/te-lsps/top	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • device (array): Array of node objects. • endTime (Either string or integer): Date time. . • size (number): ???.
POST	/statistics/v2/te-lsps/bulkdelay	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • startTime (Either string or integer): Date time. . • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • lsp (array): Array of lsp objects. • endTime (Either string or integer): Date time. .

Method	URI	Description
POST	/statistics/v2/te-lsps/de-lay/{device_name}/{lsp_name}	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,.
POST	/statistics/v2/te-lsps/traffic	The POST request accepts a JSON object describing the query parameters. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,.
POST	/statistics/v2/te-lsps/traffic/{device_name}	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum

Method	URI	Description
		• startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,. .
POST	/statistics/v2/te-lsps/traffic/{device_name}/{lsp_name}	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,. .
POST	/statistics/v2/te-lsps/events/bulk	The POST request accepts a JSON object describing the query parameters. • from (number): TODO. • startTime (Either string or integer): Date time. . • filter (object): Generic filter object, ???. • lsp (array): Array of lsp objects. • endTime (Either string or integer): Date time. . • order : Result order (asc or desc).the value must be one of: • asc • desc • size (number): ???.
POST	/statistics/v2/te-lsps/events/{lsp_name}	This query does not accept any parameter.
POST	/statistics/v2/device/bulk	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. . • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg

Method	URI	Description
		• max • min • cardinality • sum • startTime (Either string or integer): Date time. . • device (array): Array of node objects. • endTime (Either string or integer): Date time. . .
POST	/statistics/v2/device/top	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • device (array): Array of node objects. • endTime (Either string or integer): Date time. . • size (number): ???. .
POST	/statistics/v2/ldp-lsps/bulk	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • startTime (Either string or integer): Date time. . • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • lsp (array): Array of lsp objects. • endTime (Either string or integer): Date time. .

Method	URI	Description
POST	/statistics/v2/ldp-lsps/traf-fic/{device_name}/{fec}	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,.
POST	/statistics/v2/net-flow/prefix_flow_demand	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • demand (array): Array of demand objects. • endTime (Either string or integer): Date time. .
POST	/statistics/v2/sid/bulk	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max

Method	URI	Description
		• min • cardinality • sum • startTime (Either string or integer): Date time. . • sid (array): Array of sid objects. • endTime (Either string or integer): Date time. . .
POST	/statistics/v2/sr-te-policy/bulk	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • policy (array): Array of policy objects. • endTime (Either string or integer): Date time. . .
POST	/statistics/v2/te-containers/traffic/bulk	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • containerLsp (array): Array of Container LSP objects. • endTime (Either string or integer): Date time. . .

Method	URI	Description
Transport Topology acquisition		
GET	/transport-controllers/v2	Returns a full list of the transport controllers using the following schema: transportController.json#/definitions/transportControllerList
POST	/transport-controllers/v2	Creates a transport controller using the following schema: transportController.json#/definitions/createTransportController
GET	/transport-controllers/v2/{transportControllerIndex}	Returns the configuration of a transport controller.
PUT	/transport-controllers/v2/{transportControllerIndex}	Updates a transport controller using the following schema: transportController.json#/definitions/updateTransportController The parameters are the same as for transport controller creation. A parameter change triggers reconnection to the transport controller using the new parameters.
DELETE	/transport-controllers/v2/{transportControllerIndex}	Deletes a transport controller configuration. The transport node, link, and circuits created from that transport controller are not deleted. You must delete them manually, if required.
Transport Topology devices		
GET	/transport-controller-groups/v2	Returns the full list of transport controller groups using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroupList .
POST	/transport-controller-groups/v2	Creates a group of transport controller devices using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroupDescription .
DELETE	/transport-controller-groups/v2	Deletes a group of transport controller devices using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroupDescription
GET	/transport-controller-groups/v2/{transportControllerGroupName}	Returns a list of transport controller groups using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroup The password is set by empty strings.
POST	/transport-controller-groups/v2/{transportControllerGroupName}	Creates devices in a group using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroup
PUT	/transport-controller-groups/v2/{transportControllerGroupName}	Updates devices in a group using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroup
DELETE	/transport-controller-groups/v2/{transportControllerGroupName}	Deletes devices from a group using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroup
Path Computation API		
POST	/topology/v2/{topology_id}/path-Computation	Use the following schema to request for a path computation: pathComputation.json#/definitions/pathComputationRequests . Response: pathComputation.json#/definitions/pathComputationResponses .
Maintenance API		
GET	/topology/v2/{topology_id}/maintenances	Returns a full list of Maintenance.
POST	/topology/v2/{topology_id}/maintenances	Create a new maintenance using the following schema: maintenance.json#/definitions/createMaintenance .
GET	/topology/v2/{topology_id}/maintenances/stream	See EventSource for format. The notifications send on that stream are only maintenanceEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).

Method	URI	Description
GET	/topology/v2/{topology_id}/maintenances/{maintenanceIndex}	Returns the details of a specified maintenance.
PUT	/topology/v2/{topology_id}/maintenances/{maintenanceIndex}	Modify a specific maintenance.
DELETE	/topology/v2/{topology_id}/maintenances/{maintenanceIndex}	Delete a specific maintenance.
RPCs		
Path Optimization RPC		
POST	/topology/v2/rpc/optimize	Trigger global path optimization.
GET	/topology/v2/rpc/optimize	Get last global path optimization results.
POST	/topology/v2/rpc/optimize	Trigger targeted path optimization.
Routing and Failure Simulation RPC		
POST	/topology/v2/rpc/simulation	Perform failure simulation analysis.
GET	/topology/v2/rpc/simulation	Returns the simulation list.
GET	/topology/v2/rpc/simulation/{uuid}	Returns the details of a specified simulation.
GET	/topology/v2/rpc/simulation/{uuid}/{reportName}	Returns specific report details of a selected simulation.
P2MP Tree design RPC		
POST	/topology/v2/rpc/diverseTreeDesign	Create 2 P2MP trees and perform Diverse P2MP Tree Design.
GET	/topology/v2/rpc/diverseTreeDesign	Return the list of simulation optimization RPC list.
GET	/topology/v2/rpc/diverseTreeDesign/{uuid}	Return the details of specified simulation optimization RPC result.
POST	/topology/v2/rpc/diverseTreeDesign	Create 2 P2MP trees and perform Diverse P2MP Tree Design with flow mapping.
Notification examples		

3.1. Notification API

Paragon PathFinder provides a [socket.io](#) interface on which object updates are sent. The socket.io namespace used is /restNotifications-v2, and the list of events is described in [Table 3.1, “Paragon PathFinder /restNotifications-v2 events” \[300\] the next table](#). Unless noted, the new value of an object (or the index of the object upon object removal) is sent in the event.

Table 3.1. Paragon PathFinder /restNotifications-v2 events

Event	Schema	Description
topologyEvent	topology_v2.json#/definitions/topologyNotification	Event affecting the complete topology, for instance when the administrator does a topology reset or resync. The notification sent can be as follows. First an add notification with empty topology indicating that the Topology has been re-synched or reset. Second an update notification with a topology object containing only node or link set. This indicates that the complete node (resp. link) should be re-synched. This may happen in high-load situation where the notification rate is too big (northstar.cfg *_state_refresh_threshold).

Event	Schema	Description
nodeEvent	topology_v2.json#/definitions/node-Notification	Node event notification.
linkEvent	topology_v2.json#/definitions/linkNotification	Link event notification.
lspEvent	topology_v2.json#/definitions/lspNotification	TE-LSP event notification.
lspTopologyEvent	topology_v2.json#/definitions/lsp-TopologyEvent	This notification is sent when there is a burst of LSP notification, it indicates that the complete set of TE-LSPs is potentially changed. See northstar.cfg lsp_state_refresh_threshold
demandEvent	topology_v2.json#/definitions/demandNotification	Demand event notification.
demandTopologyEvent	topology_v2.json#/definitions/lsp-TopologyEvent	This notification is sent when there is a burst of LSP or demand notification, it indicates that the complete set of TE-LSPs is potentially changed. See northstar.cfg lsp_state_refresh_threshold (This is intentionally the same as TE-LSPs).
p2mpEvent	topology_v2.json#/definitions/p2mpGroupNotification	P2MP group event notification. The lsp in the update are reduced to their lspIndex to reduce the size of the event
facilityEvent	topology_v2.json#/definitions/facilityNotification	Facility/SRLG event notification.
teContainerEvent	topology_v2.json#/definitions/containerNotification	TE Container event notification.
maintenanceEvent	topology_v2.json#/definitions/maintenanceNotification	Maintenance resource notification.
haEvent	topology_v2.json#/definitions/haHostNotification	Node state event notification. Only update (no add or remove) events are supported. The notification does not include the list of processes and only contains operational information.
healthEvent	topology_v2.json#/definitions/healthNotification	Node health event notification. Only update (no add or remove) events are supported. The notifications include utilization of CPU, disk, memory that exceed certain threshold, and processes status.
componentStatus	topology_v2.json#/definitions/status-ComponentNotification	Indicating LSP reconciliation during NS bringup or Switchover.
ipePolicyEvent	topology_v2.json#/definitions/ipePolicyNotification	IPE Policy event notification.

The following example is a simple python client for receiving all Paragon PathFinder Notifications:

```
#!/usr/bin/env python
from socketIO_client import SocketIO, BaseNamespace
import requests, json, sys
serverURL = 'https://paragon.example.net'
username = 'user'
password = 'password'
class NSNotificationNamespace(BaseNamespace):
    def on_connect(self):
```

```
    print('Connected to %s /restNotifications-v2'%serverURL)
    def on_event(key,name,data):
        print("Event: %r,data:%r"%(name,json.dumps(data)))
# First use IAM authentication API to get a token
payload = {"user":{"domain":"default","name":username},"password":password,
"methods":["PASSWORD"]}
r = requests.post('https://{server}/iam/authenticate'.format(server=server),
    json=payload, verify=False)
data = r.json()
if "id_token" not in data:
    print("Error: Invalid credentials")
    sys.exit(1)
headers= {'x-iam-token': "{id_token}".format(**data)}
socketIO = SocketIO(serverURL, verify=False,headers=headers, resource=
"traffic-engineering/api/socket.io")
ns = socketIO.define(NSNotificationNamespace, '/restNotifications-v2')
socketIO.wait()
```

3.2. Topology

Use the endpoints to retrieve the following information:

- Topology list
- Node list
- Links
- TE-LSPs
- Demands
- Facilities
- P2MP group
- TE-Containers
- Maintenances
- Interfaces
- Component status
- Topology status

Apply query parameter `q=JSONPath` to filter the objects listed under a specific endpoint. Example: The HTTP request GET `https://paragon.example.net/traffic-engineering/api/topology/v2/1/te-lsps?q=$.?(@.operationalStatus == "Down")` lists only the te-lsps with attribute `operationalStatus` value "Down". For more information on JSON, refer [JSONPath](#)

The topology schema is: [topology_v2.json](#) . The operations are:

- `https://paragon.example.net/traffic-engineering/api/topology/v2/topologies` [GET: get a list of topologies]

- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>> [GET: get all Nodes and Links]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/status> [GET: get status of all Paragon PathFinder components]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/status/multilayerTopology> [GET: get multi-layer topology status]

The topology is typically populated using BGP-LS peering with Paragon PathFinder, but it is possible to add nodes and links using the REST API for modelling purposes. Since NorthStar 6.0, two node attributes are used to refine how nodes added using the REST API are considered for path computation, typically to model CE and Site nodes. The first attribute is a new attribute (ipRole) that models how the node is used with regard to LSP termination/transit. The attribute accepts the values: Regular, Access and Core and are defined below. Access nodes cannot act as transit, while core nodes can be transit but cannot terminate LSPs from access nodes directly. Regular nodes can act as transit nodes. The second attribute is an existing attribute (Vendor) with special values (CE or SITE) and control how the LSP is installed in the network: CE and Site nodes are pruned from the signaled path. Please see [Nodes](#) section for example on how to create Site and CE nodes.

Method	URI	Description
GET	/topology/v2	Returns a list of the topologies that are available.
GET	/topology/v2/{topology_id}	Lists topological elements
DELETE	/topology/v2/{topology_id}	Deletes all of the topology planned data. The information acquired through BGP-LS reappears immediately.
GET	/topology/v2/{topology_id}/stream	See EventSource for format. The notifications send on that stream are only topologyEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).
	/topology/v2/{topology_id}/status	
GET	/topology/v2/{topology_id}/status/pce	Returns the status of the PCEP protocol adapter component.
GET	/topology/v2/{topology_id}/status/topologyAcquisition	Returns the status of the Network Topology Acquisition Daemon (NTAD, BGP-LSP protocol adapter) component.
GET	/topology/v2/{topology_id}/status/pathComputationServer	Returns the status of the Path Computation Server component.
GET	/topology/v2/{topology_id}/status/transportTopologyAcquisition	Returns the status of the Transport Network Topology Acquisition Daemon.
GET	/topology/v2/{topology_id}/status/analyticsCollection	Returns the status of the analytics data collection.
GET	/topology/v2/{topology_id}/status/analyticsDatabase	Returns the status of the analytics data collection database.

3.2.1. Get List of Topologies

Method	URI	Description
GET	/topology/v2	Returns a list of the topologies that are available.

Normal response codes: 200

3.2.1.1. Request

This operation does not accept a request body.

3.2.1.2. Response

Example 3.1. Get List of Topologies: JSON response

```
[
  {
    "topologyIndex": 1,
    "topoObjectType": "topology"
  }
]
```

3.2.2. Get Topological Elements

Method	URI	Description
GET	/topology/v2/{topology_id}	Lists topological elements

Normal response codes: 200

3.2.2.1. Request

This table shows the URI parameters for the get topological elements request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.2.2.2. Response

Example 3.2. Get Topological Elements: JSON response

The following examples show different set of results:

- The example shows three nodes, and two point-to-point links between node 11 and node 8. One node is PCEP-enabled (node 8, name 62.0.0.104) and one node is a pseudo node (node 2). A pseudo node does not have router-ids in its protocol object. It's pseudoNode attribute set to TRUE.

```
{
  "links": [
    {
      "endA": {
        "TEcolor": 0,
        "TEmetric": 10,
        "bandwidth": 10000000000,
        "ipv4Address": {
          "address": "62.104.107.1",
          "topoObjectType": "ipv4"
        },
      },
      "node": {
        "name": "62.0.0.104",
        "topoObjectType": "node",
        "topologyIndex": 1
      },
      "protocols": {
        "ISIS": {
          "TEMetric": 10,
          "level": "L2",
          "srlgs": [{"srlgValue": 407}]
        },
        "RSVP": {"bandwidth": 10000000000}
      },
      "srlgs": [{"srlgValue": 407 }],
      "topoObjectType": "interface",
      "unreservedBw": [9989999616, 9989999616, 9989999616,
616, 9489999872, 9489999872, 9489999872, 9489999872 ]
    },
  ],
}
```

```

    "endZ": {
      "TEcolor": 0,
      "TEmetric": 10,
      "bandwidth": 10000000000,
      "ipv4Address": {
        "address": "62.104.107.2",
        "topoObjectType": "ipv4"
      },
      "node": {
        "name": "62.0.0.107",
        "topoObjectType": "node",
        "topologyIndex": 1
      },
      "protocols": {
        "ISIS": {
          "TEMetric": 10,
          "level": "L2",
          "srlgs": [ { "srlgValue": 407 } ]
        },
        "RSVP": { "bandwidth": 10000000000 }
      },
      "srlgs": [{"srlgValue": 407}],
      "topoObjectType": "interface",
      "unreservedBw": [9989999616, 9989999616, 9989999616, 9989999616,
9989999616, 9989999616, 9989999616, 9989999616, 9489999872]
    },
    "linkIndex": 6,
    "name": "L62.104.107.1_62.104.107.2",
    "operationalStatus": "Up",
    "topoObjectType": "link",
    "topologyIndex": 1
  },
  {
    "endA": {
      "TEcolor": 0,
      "TEmetric": 10,
      "bandwidth": 10000000000,
      "ipv4Address": {
        "address": "62.114.117.1",
        "topoObjectType": "ipv4"
      },
      "node": {
        "name": "62.0.0.104",
        "topoObjectType": "node",
        "topologyIndex": 1
      },
      "protocols": {
        "ISIS": {
          "TEMetric": 10,
          "level": "L2",
          "srlgs": [{"srlgValue": 407}]
        },
        "RSVP": { "bandwidth": 10000000000 }
      },
      "srlgs": [{"srlgValue": 407} ],
      "topoObjectType": "interface",
      "unreservedBw": [ 10000000000, 10000000000, 10000000000,
10000000000, 10000000000, 10000000000, 10000000000, 10000000000, 10000000000]
    },
    "endZ": {

```

```

        "TEcolor": 0,
        "TEmetric": 10,
        "bandwidth": 10000000000,
        "ipv4Address": {
            "address": "62.114.117.2",
            "topoObjectType": "ipv4"
        },
        "node": {
            "name": "62.0.0.107",
            "topoObjectType": "node",
            "topologyIndex": 1
        },
        "protocols": {
            "ISIS": {
                "TEMetric": 10,
                "level": "L2",
                "srlgs": [{"srlgValue": 407}]
            },
            "RSVP": {"bandwidth": 10000000000}
        },
        "srlgs": [{"srlgValue": 407}],
        "topoObjectType": "interface",
        "unreservedBw": [9600000000, 6000000000, 9600000000, 9600000000,
9600000000, 9600000000, 9600000000, 9600000000]
    },
    "linkIndex": 7,
    "name": "L62.114.117.1_62.114.117.2",
    "operationalStatus": "Up",
    "topoObjectType": "link",
    "topologyIndex": 1
}
],
"nodes": [
    {
        "AutonomousSystem": {"asNumber": 62},
        "hostName": "vmx104-62",
        "layer": "IP",
        "name": "62.0.0.104",
        "nodeIndex": 8,
        "protocols": {
            "ISIS": {
                "TERouterId": "62.0.0.104",
                "area": "490062",
                "isoAddress": "0620.0000.0104",
                "routerId": "62.0.0.104"
            },
            "PCEP": {
                "pccAddress": "62.0.0.104"
            }
        },
        "topoObjectType": "node",
        "topologyIndex": 1
    },
    {
        "AutonomousSystem": {"asNumber": 62},
        "hostName": "vmx105-62-p107",
        "layer": "IP",
        "name": "62.0.0.107",
        "nodeIndex": 11,
        "protocols": {

```

```

        "ISIS": {
            "TERouterId": "62.0.0.107",
            "area": "490062",
            "isoAddress": "0620.0000.0107",
            "routerId": "62.0.0.107"
        },
        "topoObjectType": "node",
        "topologyIndex": 1
    },
    {
        "topoObjectType": "node",
        "topologyIndex": 1,
        "name": "0620.0000.0101.02",
        "nodeIndex": 2,
        "AutonomousSystem": {"asNumber": 62},
        "layer": "IP",
        "pseudoNode": true,
        "protocols": {
            "ISIS": { }
        }
    },
    "topoObjectType": "topology",
    "topologyIndex": 1
}

```

- The example shows two PE nodes (PE3 and PE4 - discovered from BGP-LS), two CE nodes (CE1 and CE2 - manually created using Node API) and three Sites (SiteA, SiteB and SiteC - manually created using Node API) nodes, and links between PE3 and PE4 (id: L10.0.1.10_10.0.1.9 - discovered from BGP-LS), and manually created link using Links API which are PE3 and SiteA (id: L0100.0000.0003_SiteA_PE3-SiteA), PE4 and SiteB (id: L0100.0000.0004_SiteB_PE4-SiteB), PE3 and CE1 (id: L0100.0000.0003_CE1_PE3-CE1), CE1 and SiteO (id: LCE1_SiteO_CE1-SiteO) and CE2 and SiteO (id: LCE2_SiteO_CE2-SiteO).

```

{
    "topologyIndex": 1,
    "topoObjectType": "topology",
    "nodes": [
        {
            "topoObjectType": "node",
            "topologyIndex": 1,
            "id": "0100.0000.0003",
            "nodeIndex": 3,
            "layer": "IP",
            "live": true,
            "AutonomousSystem": {
                "asNumber": 100
            },
            "routerId": "10.0.0.3",
            "protocols": {
                "ISIS": {
                    "routerId": "10.0.0.3",
                    "TERouterId": "10.0.0.3",
                    "isoAddress": "0100.0000.0003",
                    "area": "490001"
                },
                "NETCONF": {
                    "operationalStatus": "Up",

```

```

        "operationalState": "Connected",
        "clientAddress": "10.0.0.3",
        "sessionParameters": {
            "keepalive": 300
        },
        "clientCapabilities": [
            "http://xml.juniper.net/dmi/system/1.0",
            "http://xml.juniper.net/netconf/junos/1.0",
            "urn:ietf:params:netconf:base:1.0",
            "urn:ietf:params:netconf:capability:candidate:1.0",
            "urn:ietf:params:netconf:capability:confirmed-
commit:1.0",
            "urn:ietf:params:netconf:capability:url:1.0?scheme=
http,ftp,file",
            "urn:ietf:params:netconf:capability:validate:1.0",
            "urn:ietf:params:xml:ns:netconf:base:1.0",

            "urn:ietf:params:xml:ns:netconf:capability:candidate:1.0",

            "urn:ietf:params:xml:ns:netconf:capability:confirmed-commit:1.0",
            "urn:ietf:params:xml:ns:netconf:capability:url:1.0?
scheme=http,ftp,file",

            "urn:ietf:params:xml:ns:netconf:capability:validate:1.0",
            "urn:ietf:params:xml:ns:yang:ietf-netconf-
monitoring"
        ]
    },
    "PCEP": {
        "operationalStatus": "Up",
        "pccAddress": "10.0.0.3",
        "sessionParameters": {
            "keepalive": 0,
            "deadtimer": 0,
            "maximumStackDepth": 0
        },
        "peerStatefullCapabilities": {
            "lspUpdate": true,
            "lspInstantiation": true,
            "p2mp": true,
            "p2mpUpdate": true,
            "p2mpInstantiation": true,
            "SRCapability": false
        }
    },
    "management": {
        "address": "10.0.0.3",
        "vendor": "JUNIPER",
        "operatingSystem": "JUNOS",
        "operatingSystemVersion": "19.4R1.10"
    },
    "hostName": "PE3",
    "configurationDataSource": {
        "latest": {
            "href": "/NorthStar/API/v2/tenant/1/topology/1/nodes/3/
device/configurationDataSource/latest"
        }
    }
},

```

```

{
  "topoObjectType": "node",
  "topologyIndex": 1,
  "id": "0100.0000.0004",
  "nodeIndex": 4,
  "layer": "IP",
  "live": true,
  "AutonomousSystem": {
    "asNumber": 100
  },
  "routerId": "10.0.0.4",
  "protocols": {
    "ISIS": {
      "routerId": "10.0.0.4",
      "TERouterId": "10.0.0.4",
      "isoAddress": "0100.0000.0004",
      "area": "490001"
    },
    "NETCONF": {
      "operationalStatus": "Up",
      "operationalState": "Connected",
      "clientAddress": "10.0.0.4",
      "sessionParameters": {
        "keepalive": 300
      },
      "clientCapabilities": [
        "http://xml.juniper.net/dmi/system/1.0",
        "http://xml.juniper.net/netconf/junos/1.0",
        "urn:ietf:params:netconf:base:1.0",
        "urn:ietf:params:netconf:capability:candidate:1.0",
        "urn:ietf:params:netconf:capability:confirmed-
commit:1.0",
        "urn:ietf:params:netconf:capability:url:1.0?scheme=
http,ftp,file",
        "urn:ietf:params:netconf:capability:validate:1.0",
        "urn:ietf:params:xml:ns:netconf:base:1.0",
        "urn:ietf:params:xml:ns:netconf:capability:candidate:1.0",
        "urn:ietf:params:xml:ns:netconf:capability:confirmed-commit:1.0",
        "urn:ietf:params:xml:ns:netconf:capability:url:1.0?
scheme=http,ftp,file",
        "urn:ietf:params:xml:ns:netconf:capability:validate:1.0",
        "urn:ietf:params:xml:ns:yang:ietf-netconf-
monitoring"
      ]
    },
    "PCEP": {
      "operationalStatus": "Up",
      "pccAddress": "10.0.0.4",
      "sessionParameters": {
        "keepalive": 0,
        "deadtimer": 0,
        "maximumStackDepth": 0
      },
      "peerStatefullCapabilities": {
        "lspUpdate": true,
        "lspInstantiation": true,
        "p2mp": true,

```



```
        "p2mpUpdate": true,
        "p2mpInstantiation": true,
        "SRCapability": false
      }
    },
    "management": {
      "address": "10.0.0.4",
      "vendor": "JUNIPER",
      "operatingSystem": "JUNOS",
      "operatingSystemVersion": "19.4R1.10"
    }
  },
  "hostName": "PE4",
  "configurationDataSource": {
    "latest": {
      "href": "/NorthStar/API/v2/tenant/1/topology/1/nodes/4/
device/configurationDataSource/latest"
    }
  }
},
{
  "topoObjectType": "node",
  "topologyIndex": 1,
  "id": "SiteA",
  "nodeIndex": 11,
  "layer": "Service",
  "live": false,
  "routerId": "11.0.0.1",
  "ipRole": "Access",
  "protocols": {
    "ISIS": {
      "routerId": "11.0.0.1",
      "TERouterId": "11.0.0.1"
    },
    "management": {
      "vendor": "SITE"
    }
  },
  "hostName": "SiteA",
  "name": "SiteA"
},
{
  "topoObjectType": "node",
  "topologyIndex": 1,
  "id": "SiteB",
  "nodeIndex": 12,
  "layer": "Service",
  "live": false,
  "routerId": "11.0.0.2",
  "ipRole": "Access",
  "protocols": {
    "ISIS": {
      "routerId": "11.0.0.2",
      "TERouterId": "11.0.0.2"
    },
    "management": {
      "vendor": "SITE"
    }
  },
  "hostName": "SiteB",
```

```
      "name": "SiteB"
    },
    {
      "topoObjectType": "node",
      "topologyIndex": 1,
      "id": "Site0",
      "nodeIndex": 25,
      "layer": "Service",
      "live": false,
      "routerId": "11.0.0.15",
      "ipRole": "Access",
      "protocols": {
        "ISIS": {
          "routerId": "11.0.0.15",
          "TERouterId": "11.0.0.15"
        },
        "management": {
          "vendor": "SITE"
        }
      },
      "hostName": "Site0",
      "name": "Site0"
    },
    {
      "topoObjectType": "node",
      "topologyIndex": 1,
      "id": "CE1",
      "nodeIndex": 31,
      "layer": "Service",
      "live": false,
      "routerId": "11.0.0.101",
      "ipRole": "Regular",
      "protocols": {
        "ISIS": {
          "routerId": "11.0.0.101",
          "TERouterId": "11.0.0.101"
        },
        "management": {
          "vendor": "CE"
        }
      },
      "hostName": "CE1",
      "name": "CE1"
    },
    {
      "topoObjectType": "node",
      "topologyIndex": 1,
      "id": "CE2",
      "nodeIndex": 32,
      "layer": "Service",
      "live": false,
      "routerId": "11.0.0.102",
      "ipRole": "Regular",
      "protocols": {
        "ISIS": {
          "routerId": "11.0.0.102",
          "TERouterId": "11.0.0.102"
        },
        "management": {
          "vendor": "CE"
        }
      }
    }
  ]
}
```

```

    }
    },
    "hostname": "CE2",
    "name": "CE2"
  }
],
"links": [
  {
    "topoObjectType": "link",
    "linkIndex": 6,
    "id": "L10.0.1.10_10.0.1.9",
    "live": true,
    "topologyIndex": 1,
    "operationalStatus": "Up",
    "endZ": {
      "ipv4Address": {
        "topoObjectType": "ipv4",
        "address": "10.0.1.10"
      },
      "protocols": {
        "RSVP": {
          "bandwidth": 10000000
        },
        "ISIS": {
          "level": "L2",
          "TEMetric": 10
        }
      },
      "bandwidth": 10000000,
      "interfaceName": "ge-0/0/1.0",
      "TEcolor": 0,
      "TEmetric": 10,
      "unreservedBw": [
        10000000,
        10000000,
        10000000,
        10000000,
        10000000,
        10000000,
        10000000,
        10000000
      ],
      "node": {
        "topoObjectType": "node",
        "id": "0100.0000.0004",
        "topologyIndex": 1
      },
      "topoObjectType": "interface",
      "utilization": {
        "utilization": 0,
        "availableBandwidth": 10000000,
        "usedBandwidth": 0,
        "measuredDelay": 0.87,
        "interfaceUtilization": 0.00107,
        "interfaceTraffic": 10700,
        "tunnelUtilization": 0,
        "tunnelLiveUtilization": 0
      }
    },
    "endA": {

```

```

        "ipv4Address": {
          "topoObjectType": "ipv4",
          "address": "10.0.1.9"
        },
        "protocols": {
          "RSVP": {
            "bandwidth": 10000000
          },
          "ISIS": {
            "level": "L2",
            "TEMetric": 10
          }
        },
        "bandwidth": 10000000,
        "interfaceName": "ge-0/0/0.0",
        "TEMetric": 10,
        "unreservedBw": [
          10000000,
          10000000,
          10000000,
          10000000,
          10000000,
          10000000,
          10000000,
          10000000
        ],
        "node": {
          "topoObjectType": "node",
          "id": "0100.0000.0003",
          "topologyIndex": 1
        },
        "topoObjectType": "interface",
        "utilization": {
          "utilization": 0,
          "availableBandwidth": 10000000,
          "usedBandwidth": 0,
          "measuredDelay": 0.87,
          "interfaceUtilization": 0.00158,
          "interfaceTraffic": 15800,
          "tunnelUtilization": 0,
          "tunnelLiveUtilization": 0,
          "lastUpdate": "2020-04-30T17:07:08Z"
        }
      }
    },
    {
      "topoObjectType": "link",
      "linkIndex": 13,
      "id": "L0100.0000.0003_SiteA_PE3-SiteA",
      "live": false,
      "topologyIndex": 1,
      "operationalStatus": "Unknown",
      "name": "PE3-SiteA",
      "endZ": {
        "node": {
          "topoObjectType": "node",
          "id": "SiteA",
          "topologyIndex": 1
        },
        "topoObjectType": "interface",

```

```
        "utilization": {
          "utilization": 0,
          "availableBandwidth": 1000000000,
          "usedBandwidth": 0,
          "tunnelUtilization": 0,
          "tunnelLiveUtilization": 0
        }
      },
      "endA": {
        "node": {
          "topoObjectType": "node",
          "id": "0100.0000.0003",
          "topologyIndex": 1
        },
        "topoObjectType": "interface",
        "utilization": {
          "utilization": 0,
          "availableBandwidth": 1000000000,
          "usedBandwidth": 0,
          "tunnelUtilization": 0,
          "tunnelLiveUtilization": 0,
          "lastUpdate": "2020-04-30T07:34:55Z"
        }
      }
    },
    {
      "topoObjectType": "link",
      "linkIndex": 46,
      "id": "L0100.0000.0003_CE1_PE3-CE1",
      "live": false,
      "topologyIndex": 1,
      "operationalStatus": "Unknown",
      "name": "PE3-CE1",
      "endZ": {
        "node": {
          "topoObjectType": "node",
          "id": "CE1",
          "topologyIndex": 1
        },
        "topoObjectType": "interface",
        "utilization": {
          "utilization": 0,
          "availableBandwidth": 1000000000,
          "usedBandwidth": 0,
          "tunnelUtilization": 0,
          "tunnelLiveUtilization": 0
        }
      },
      "endA": {
        "node": {
          "topoObjectType": "node",
          "id": "0100.0000.0003",
          "topologyIndex": 1
        },
        "topoObjectType": "interface",
        "utilization": {
          "utilization": 0,
          "availableBandwidth": 1000000000,
          "usedBandwidth": 0,
          "tunnelUtilization": 0,
```

```
        "tunnelLiveUtilization": 0,
        "lastUpdate": "2020-04-30T07:34:55Z"
    }
}
},
{
    "topoObjectType": "link",
    "linkIndex": 62,
    "id": "LCE1_Site0_CE1-Site0",
    "live": false,
    "topologyIndex": 1,
    "operationalStatus": "Unknown",
    "name": "CE1-Site0",
    "endZ": {
        "node": {
            "topoObjectType": "node",
            "id": "Site0",
            "topologyIndex": 1
        },
        "topoObjectType": "interface",
        "utilization": {
            "utilization": 0,
            "availableBandwidth": 1000000000,
            "usedBandwidth": 0,
            "tunnelUtilization": 0,
            "tunnelLiveUtilization": 0
        }
    },
    "endA": {
        "node": {
            "topoObjectType": "node",
            "id": "CE1",
            "topologyIndex": 1
        },
        "topoObjectType": "interface",
        "utilization": {
            "utilization": 0,
            "availableBandwidth": 1000000000,
            "usedBandwidth": 0,
            "tunnelUtilization": 0,
            "tunnelLiveUtilization": 0,
            "lastUpdate": "2020-04-30T07:34:55Z"
        }
    }
}
},
{
    "topoObjectType": "link",
    "linkIndex": 63,
    "id": "LCE2_Site0_CE2-Site0",
    "live": false,
    "topologyIndex": 1,
    "operationalStatus": "Unknown",
    "name": "CE2-Site0",
    "endZ": {
        "node": {
            "topoObjectType": "node",
            "id": "Site0",
            "topologyIndex": 1
        },
        "topoObjectType": "interface",
```

```
        "utilization": {
          "utilization": 0,
          "availableBandwidth": 1000000000,
          "usedBandwidth": 0,
          "tunnelUtilization": 0,
          "tunnelLiveUtilization": 0
        }
      },
      "endA": {
        "node": {
          "topoObjectType": "node",
          "id": "CE2",
          "topologyIndex": 1
        },
        "topoObjectType": "interface",
        "utilization": {
          "utilization": 0,
          "availableBandwidth": 1000000000,
          "usedBandwidth": 0,
          "tunnelUtilization": 0,
          "tunnelLiveUtilization": 0,
          "lastUpdate": "2020-04-30T07:34:55Z"
        }
      }
    },
    {
      "topoObjectType": "link",
      "linkIndex": 15,
      "id": "L0100.0000.0004_SiteB_PE4-SiteB",
      "live": false,
      "topologyIndex": 1,
      "operationalStatus": "Unknown",
      "name": "PE4-SiteB",
      "endZ": {
        "node": {
          "topoObjectType": "node",
          "id": "SiteB",
          "topologyIndex": 1
        },
        "topoObjectType": "interface",
        "utilization": {
          "utilization": 0,
          "availableBandwidth": 1000000000,
          "usedBandwidth": 0,
          "tunnelUtilization": 0,
          "tunnelLiveUtilization": 0
        }
      },
      "endA": {
        "node": {
          "topoObjectType": "node",
          "id": "0100.0000.0004",
          "topologyIndex": 1
        },
        "topoObjectType": "interface",
        "utilization": {
          "utilization": 0,
          "availableBandwidth": 1000000000,
          "usedBandwidth": 0,
          "tunnelUtilization": 0,
```

```
    "tunnelLiveUtilization": 0,  
    "lastUpdate": "2020-04-30T07:34:55Z"  
  }  
}  
]  
}
```


3.2.3. Delete the Topology

Method	URI	Description
DELETE	/topology/v2/{topology_id}	Deletes all of the topology planned data. The information acquired through BGP-LS reappears immediately.

Normal response codes: 204

3.2.3.1. Request

This table shows the URI parameters for the delete the topology request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.2.4. Start a SSE Stream

Method	URI	Description
GET	/topology/v2/{topology_id}/stream	See EventSource for format. The notifications send on that stream are only topologyEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).

Normal response codes: 200

3.2.4.1. Request

This table shows the URI parameters for the start a sse stream request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.2.5.

Method	URI	Description
	/topology/v2/{topology_id}/status	

3.2.5.1. Request

This table shows the URI parameters for the request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.2.6. Get PCEP Protocol Adapter Status

Method	URI	Description
GET	/topology/v2/{topology_id}/status/pce	Returns the status of the PCEP protocol adapter component.

Normal response codes: 200

3.2.6.1. Request

This table shows the URI parameters for the get pcep protocol adapter status request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.2.6.2. Response

Example 3.3. Get PCEP Protocol Adapter Status: JSON response

```
{
  "component": "PCE",
  "status": "PCE is up.",
  "statusTimestamp": 0
}
```

3.2.7. Get BGP-LS Protocol Adapter Status

Method	URI	Description
GET	/topology/v2/{topology_id}/status/topologyAcquisition	Returns the status of the Network Topology Acquisition Daemon (NTAD, BGP-LSP protocol adapter) component.

Normal response codes: 200

3.2.7.1. Request

This table shows the URI parameters for the get bgp-ls protocol adapter status request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.2.7.2. Response

Example 3.4. Get BGP-LS Protocol Adapter Status: JSON response

```
{
  "component": "Topology acquisition",
  "status": "Connected to NTAD: 62.105.199.2 port: 450",
  "statusTimestamp": 1427167092212
}
```

3.2.8. Get Path Computation Server Status

Method	URI	Description
GET	/topology/v2/{topology_id}/status/pathComputationServer	Returns the status of the Path Computation Server component.

Normal response codes: 200

3.2.8.1. Request

This table shows the URI parameters for the get path computation server status request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.2.8.2. Response

Example 3.5. Get Path Computation Server Status: JSON response

```
{
  "component": "Path Computation Server",
  "status": "Active Path Stat: 3 up 0 down 0 detoured 0 being provisioned.
Link Stat: 19 up 0 down. Node Stat: 7 active nodes, 0 PCC nodes",
  "statusTimestamp": 1468153795771,
  "statusTimestampTime": "2016-07-10T12:29:55.771Z"
}
```

3.2.9. Get Transport Topology Acquisition Daemon Status

Method	URI	Description
GET	/topology/v2/{topology_id}/status/transportTopologyAcquisition	Returns the status of the Transport Network Topology Acquisition Daemon.

Normal response codes: 200

3.2.9.1. Request

This table shows the URI parameters for the get transport topology acquisition daemon status request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.2.9.2. Response

Example 3.6. Get Transport Topology Acquisition Daemon Status: JSON response

```
{
  "component": "Transport Topology acquisition",
  "status": "Up",
  "childs": [
    {
      "status": "Up; current peer: hostname='' ip='192.0.2.10'",
      "statusTimestamp": 1468153845392,
      "statusTimestampTime": "2016-07-10T12:30:45.392062Z",
      "component": "VendorX"
    }
  ],
  "statusTimestamp": 1468153853495,
  "statusTimestampTime": "2016-07-10T12:30:53.495Z"
}
```

3.2.10. Get Analytics Data Collection Component Status

Method	URI	Description
GET	/topology/v2/{topology_id}/status/analyticsCollection	Returns the status of the analytics data collection.

Normal response codes: 200

3.2.10.1. Request

This table shows the URI parameters for the get analytics data collection component status request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.2.10.2. Response

Example 3.7. Get Analytics Data Collection Component Status: JSON response

```
{
  "component": "Analytics data collection",
  "status": "host=pcs-server, status=OK",
  "statusTimestamp": "2017-07-27T04:09:05.239Z",
  "statusTimestamp": 1501128545239
}
```


3.2.11. Get Analytics Data Collection Database Status

Method	URI	Description
GET	/topology/v2/{topology_id}/status/analyticsDatabase	Returns the status of the analytics data collection database.

Normal response codes: 200

3.2.11.1. Request

This table shows the URI parameters for the get analytics data collection database status request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.2.11.2. Response

Example 3.8. Get Analytics Data Collection Database Status: JSON response

```
{
  "component": "Analytics database",
  "status": "Analytics Database Status:green Cluster NorthStar nodes count:1
(1 data nodes)",
  "statusTimestamp": 1501129483764,
  "statusTimestampTime": "2017-07-27T04:24:43.764Z"
}
```

3.3. Node

Use these endpoints for access to the nodes retrieved through BGP-LS and planned nodes.

The node schema is: [node_v2.json](#) . The operations are:

- https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/nodes [GET : get all nodes, POST : create a new node]
- https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/nodes/search URL parameters : name, AS, queryType=OR [GET : search nodes (name/as)]
- https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/nodes/<nodeIndex> [GET : get node, POST : modify node , DELETE]
- https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/nodes/<nodeIndex>/history [GET : node history]

As explained in the topology section, nodes can be created in the REST API, including CE and Site nodes. The examples for those are detailed in the "Create a Planned Node" section below.

Method	URI	Description
GET	/topology/v2/{topology_id}/nodes	Returns a full list of nodes.
POST	/topology/v2/{topology_id}/nodes	Creates a node using the following schema: node_v2.json#/definitions/createNode .
GET	/topology/v2/{topology_id}/nodes/ search{?name,hostName,AS,query- Type}	Searches the list of nodes for specific URI parameters. For example, search?name=62.0.0.101 must return one node.
GET	/topology/v2/{topology_id}/nodes/ stream	See EventSource for format. The notifications send on that stream are only nodeEvent. The data will contain a JSON document (see Paragon Pathfinder Notification API).
GET	/topology/v2/{topology_id}/nodes/ {nodeIndex}	Returns details for a node.
PUT	/topology/v2/{topology_id}/nodes/ {nodeIndex}	Updates a node using the following schema: node_v2.json#/definitions/updateNode
PATCH	/topology/v2/{topology_id}/nodes/ {nodeIndex}	Updates a node using a RFC6902 patch: json-patch.json . The result of the patch must conform to node_v2.json#/definitions/updateNode . The REST server remove all operational parameters like operationalStatus, ..etc. .
DELETE	/topology/v2/{topology_id}/nodes/ {nodeIndex}	Deletes a node. (You cannot delete a live node; it reappears on the next update from Topology server.)
GET	/topology/v2/{topology_id}/nodes/ {nodeIndex}/history{?start,end}	Returns the event history for a node.
POST	/topology/v2/{topology_id}/nodes/ bulk	Creates several Nodes using the following JSON schema: node_v2.json#/definitions/createNodeList .
PUT	/topology/v2/{topology_id}/nodes/ bulk	Updates several Nodes using the following JSON schema: node_v2.json#/definitions/nodeListUpdate .
PATCH	/topology/v2/{topology_id}/nodes/ bulk	Updates several Nodes using the following JSON schema: node_v2.json#/definitions/nodeListPatch .
DELETE	/topology/v2/{topology_id}/nodes/ bulk	Deletes a list of Nodes. The payload must conform to node_v2.json#/definitions/nodeListDelete

3.3.1. Get Nodes

Method	URI	Description
GET	/topology/v2/{topology_id}/nodes	Returns a full list of nodes.

Normal response codes: 200

3.3.1.1. Request

This table shows the URI parameters for the get nodes request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.3.1.2. Response

Example 3.9. Get Nodes: JSON response

```
[
  {
    "AutonomousSystem": {"asNumber": 62},
    "layer": "IP",
    "name": "0620.0000.0101.02",
    "nodeIndex": 2,
    "protocols": {
      "ISIS": {}
    },
    "pseudoNode": true,
    "topoObjectType": "node",
    "topologyIndex": 1
  },
  {
    "AutonomousSystem": {"asNumber": 62},
    "hostName": "vmx101-62",
    "layer": "IP",
    "name": "62.0.0.101",
    "nodeIndex": 1,
    "protocols": {
      "ISIS": {
        "TERouterId": "62.0.0.101",
        "area": "490062",
        "isoAddress": "0620.0000.0101",
        "routerId": "62.0.0.101"
      },
      "PCEP": {
        "pccAddress": "62.0.0.101"
      }
    },
    "topoObjectType": "node",
    "topologyIndex": 1
  },
  {
    {
```

```
    "AutonomousSystem": {"asNumber": 62},
    "hostName": "vmx105-62-p105",
    "layer": "IP",
    "name": "62.0.0.105",
    "nodeIndex": 9,
    "protocols": {
      "ISIS": {
        "TERouterId": "62.0.0.105",
        "area": "490062",
        "isoAddress": "0620.0000.0105",
        "routerId": "62.0.0.105"
      }
    },
    "topoObjectType": "node",
    "topologyIndex": 1
  },
  {
    "AutonomousSystem": {"asNumber": 62},
    "hostName": "vmx105-62-p106",
    "layer": "IP",
    "name": "62.0.0.106",
    "nodeIndex": 10,
    "protocols": {
      "ISIS": {
        "TERouterId": "62.0.0.106",
        "area": "490062",
        "isoAddress": "0620.0000.0106",
        "routerId": "62.0.0.106"
      }
    },
    "topoObjectType": "node",
    "topologyIndex": 1
  },
  {
    "AutonomousSystem": {"asNumber": 62},
    "hostName": "vmx105-62-p107",
    "layer": "IP",
    "name": "62.0.0.107",
    "nodeIndex": 11,
    "protocols": {
      "ISIS": {
        "TERouterId": "62.0.0.107",
        "area": "490062",
        "isoAddress": "0620.0000.0107",
        "routerId": "62.0.0.107"
      }
    },
    "topology": {
      "coordinates": {
        "type": "Point",
        "coordinates": [-0.025, -49.459999]
      }
    }
  },
  {
    "topoObjectType": "node",
    "topologyIndex": 1
  }
]
```

3.3.2. Create a Node

Method	URI	Description
POST	/topology/v2/{topology_id}/nodes	Creates a node using the following schema: node_v2.json#/definitions/createNode .

Normal response codes: 201

3.3.2.1. Request

This table shows the URI parameters for the create a node request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 3.10. Create a Node: JSON request

The following JSON sample shows the minimum required information to add a planned node. The Following parameters are required:

Table 3.2. Create Node Required Attributes

Attribute	Type	Description	Fixed
name	string	No	Node Name
topologyIndex	integer	Yes	1
topObjectType	string	Yes	node

The following parameters can be set:

- Name
- Autonomous System
- management address
- hostName
- nodeType
- design parameter: canFail
- Node coordinates
- ipRole: Regular or Access or Core
- vendor: CE or SITE

Create Node examples:

- In this example, create planned node

```
{
```

```
{
  "name": "PlannedNode",
  "topoObjectType": "node",
  "topologyIndex": 1
}
```

The response is:

```
{
  "layer": "IP",
  "name": "PlannedNode",
  "nodeIndex": 12,
  "protocols": {},
  "topoObjectType": "node",
  "topologyIndex": 1
}
```

- In this example, create Site node with Access role. The node will not be used as transit node during path computation (due to ipRole=Transit) and won't be included in P2MP tree signaling (due to vendor=SITE)

```
{
  "topoObjectType": "node",
  "topologyIndex": 1,
  "name": "SiteZ",
  "ipRole": "Access",
  "protocols": {
    "management": {
      "vendor": "SITE"
    }
  }
}
```

The response is:

```
{
  "nodeIndex": 46,
  "layer": "IP",
  "name": "SiteZ",
  "ipRole": "Access",
  "topoObjectType": "node",
  "live": false,
  "topologyIndex": 1,
  "location": "/NorthStar/API/v2/tenant/1/topology/1/nodes/46",
  "id": "SiteZ",
  "protocols": {
    "management": {
      "vendor": "SITE"
    }
  }
}
```

- In this example, create CE node with Regular role. Compared to the previous example (Site), this node will be used as transit for path computation (ipRole=Regular).

```
{
  "topoObjectType": "node",
  "topologyIndex": 1,
  "name": "CEZ1",
  "ipRole": "Regular",
  "protocols": {
```

```
    "management": {  
      "vendor": "CE"  
    }  
  }  
}
```

The response is:

```
{  
  "nodeIndex": 58,  
  "layer": "IP",  
  "name": "CEZ1",  
  "ipRole": "Regular",  
  "topoObjectType": "node",  
  "live": false,  
  "topologyIndex": 1,  
  "location": "/NorthStar/API/v2/tenant/1/topology/1/nodes/58",  
  "id": "CEZ1",  
  "protocols": {  
    "management": {  
      "vendor": "CE"  
    }  
  }  
}
```

3.3.2.2. Response

The responses are shown per example in the request section. In general the response is a JSON document conforming to [node_v2.json#/definitions/createNode](#).

This operation does not return a response body.

3.3.3. Search Nodes

Method	URI	Description
GET	/topology/v2/{topology_id}/nodes/search{?name,hostName,AS,query-Type}	Searches the list of nodes for specific URI parameters. For example, search?name=62.0.0.101 must return one node.

Normal response codes: 200

3.3.3.1. Request

This table shows the URI parameters for the search nodes request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This table shows the query parameters for the search nodes request:

Name	Type	Description
name	String (Optional)	Filters on node name. The parameter is treated as a regular expression. Simple strings such as 62.0.0.101 are accepted. JavaScript regexp (for example, /^62/) is also accepted.
hostName	Int (Optional)	Filters on hostname
AS	Int (Optional)	Filters on AS number
queryType	String (Optional)	The queryType parameter sets the logical operator to be used between the query parameters. By default, the queryType is AND. If queryType is set to OR, node is included in the result if any of the query parameters matches.

3.3.3.2. Response

Example 3.11. Search Nodes: JSON response

```
[
  {
    "AutonomousSystem": {"asNumber": 62},
    "layer": "IP",
    "name": "0620.0000.0101.02",
    "nodeIndex": 2,
    "protocols": {
      "ISIS": {}
    },
    "pseudoNode": true,
    "topoObjectType": "node",
    "topologyIndex": 1
  },
  {
    "AutonomousSystem": {"asNumber": 62},
    "hostName": "vmx101-62",
    "layer": "IP",
    "name": "62.0.0.101",
    "nodeIndex": 1,
    "protocols": {
```



```
        "ISIS": {
            "TERouterId": "62.0.0.101",
            "area": "490062",
            "isoAddress": "0620.0000.0101",
            "routerId": "62.0.0.101"
        },
        "PCEP": {
            "pccAddress": "62.0.0.101"
        }
    },
    "topoObjectType": "node",
    "topologyIndex": 1
},
{
    "AutonomousSystem": {"asNumber": 62},
    "hostName": "vmx105-62-p105",
    "layer": "IP",
    "name": "62.0.0.105",
    "nodeIndex": 9,
    "protocols": {
        "ISIS": {
            "TERouterId": "62.0.0.105",
            "area": "490062",
            "isoAddress": "0620.0000.0105",
            "routerId": "62.0.0.105"
        }
    }
},
    "topoObjectType": "node",
    "topologyIndex": 1
},
{
    "AutonomousSystem": {"asNumber": 62},
    "hostName": "vmx105-62-p106",
    "layer": "IP",
    "name": "62.0.0.106",
    "nodeIndex": 10,
    "protocols": {
        "ISIS": {
            "TERouterId": "62.0.0.106",
            "area": "490062",
            "isoAddress": "0620.0000.0106",
            "routerId": "62.0.0.106"
        }
    }
},
    "topoObjectType": "node",
    "topologyIndex": 1
},
{
    "AutonomousSystem": {"asNumber": 62},
    "hostName": "vmx105-62-p107",
    "layer": "IP",
    "name": "62.0.0.107",
    "nodeIndex": 11,
    "protocols": {
        "ISIS": {
            "TERouterId": "62.0.0.107",
            "area": "490062",
            "isoAddress": "0620.0000.0107",
            "routerId": "62.0.0.107"
        }
    }
}
```

```
    },
    "topology": {
      "coordinates": {
        "type": "Point",
        "coordinates": [-0.025, -49.459999]
      }
    },
    "topoObjectType": "node",
    "topologyIndex": 1
  }
]
```

3.3.4. Start a SSE Stream

Method	URI	Description
GET	/topology/v2/{topology_id}/nodes/stream	See EventSource for format. The notifications send on that stream are only nodeEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).

Normal response codes: 200

3.3.4.1. Request

This table shows the URI parameters for the start a sse stream request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.3.5. Get a Single Node

Method	URI	Description
GET	/topology/v2/{topology_id}/nodes/{nodeIndex}	Returns details for a node.

Normal response codes: 200

3.3.5.1. Request

This table shows the URI parameters for the get a single node request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{nodeIndex}	Int	The unique nodeIndex.

This operation does not accept a request body.

3.3.5.2. Response

Example 3.12. Get a Single Node: JSON response

```
{
  "AutonomousSystem": {
    "asNumber": 62
  },
  "hostname": "vmx101-62",
  "layer": "IP",
  "name": "62.0.0.101",
  "nodeIndex": 1,
  "protocols": {
    "ISIS": {
      "TERouterId": "62.0.0.101",
      "area": "490062",
      "isoAddress": "0620.0000.0101",
      "routerId": "62.0.0.101"
    },
    "PCEP": {
      "pccAddress": "62.0.0.101"
    }
  },
  "topoObjectType": "node",
  "topologyIndex": 1
}
```

3.3.6. Update Node Properties

Method	URI	Description
PUT	/topology/v2/{topology_id}/nodes/{nodeIndex}	Updates a node using the following schema: node_v2.json#/definitions/updateNode

Normal response codes: 202

3.3.6.1. Request

This table shows the URI parameters for the update node properties request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{nodeIndex}	Int	The unique nodeIndex.

Example 3.13. Update Node Properties : JSON request

```
{
  "layer": "IP",
  "id": "PlannedNode",
  "hostName" : "plannedNode.domain.example.com",
  "nodeIndex": 12,
  "protocols": {},
  "topoObjectType": "node",
  "topologyIndex": 1
}
```

3.3.6.2. Response

Example 3.14. Update Node Properties : JSON response

```
{
  "layer": "IP",
  "id": "PlannedNode",
  "hostName" : "plannedNode.domain.example.com",
  "nodeIndex": 12,
  "protocols": {},
  "topoObjectType": "node",
  "topologyIndex": 1
}
```

3.3.7. Patch Node

Method	URI	Description
PATCH	/topology/v2/{topology_id}/nodes/{nodeIndex}	Updates a node using a RFC6902 patch: json-patch.json . The result of the patch must conform to node_v2.json#/definitions/updateNode . The REST server remove all operational parameters like operationalStatus, ..etc. .

Normal response codes: 202

3.3.7.1. Request

This table shows the URI parameters for the patch node request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{nodeIndex}	Int	The unique nodeIndex.

Example 3.15. Patch Node: JSON request

The Patched update accepts the same parameters as the update

```
[{"op" : "add", "path":"/protocols/PCEP/extensions/lsp-association-protection", "value" : false}]
```

3.3.7.2. Response

Example 3.16. Patch Node: JSON response

```
{
  "layer": "IP",
  "id": "PlannedNode",
  "hostName" : "plannedNode.domain.example.com",
  "nodeIndex": 12,
  "protocols": {},
  "topoObjectType": "node",
  "topologyIndex": 1,
  "protocols" : {
    "PCEP" : {
      "extensions" : {
        "lsp-association-protection" : false
      }
    }
  }
}
```

3.3.8. Delete a Node

Method	URI	Description
DELETE	/topology/v2/{topology_id}/nodes/{nodeIndex}	Deletes a node. (You cannot delete a live node; it reappears on the next update from Topology server.)

Normal response codes: 204

3.3.8.1. Request

This table shows the URI parameters for the delete a node request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{nodeIndex}	Int	The unique nodeIndex.

This operation does not accept a request body.

3.3.9. Get the Node Event History

Method	URI	Description
GET	/topology/v2/{topology_id}/nodes/{nodeIndex}/history{?start,end}	Returns the event history for a node.

The history contains a list of Unix-timestamped events for the node resource.

Normal response codes: 200

3.3.9.1. Request

This table shows the URI parameters for the get the node event history request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{nodeIndex}	Int	The unique nodeIndex.

This table shows the query parameters for the get the node event history request:

Name	Type	Description
start	Int (Optional)	Start timestamp: Include events with the starting time and later.
end	Int (Optional)	End timestamp: Include events before (but not including) the ending timestamp.

3.3.9.2. Response

Example 3.17. Get the Node Event History: JSON response

```
[
  {
    "topoObjectType": "node",
    "topologyIndex": 1,
    "name": "62.0.0.101",
    "nodeIndex": 1,
    "AutonomousSystem": {
      "asNumber": 62
    },
    "layer": "IP",
    "protocols": {
      "ISIS": {
        "routerId": "62.0.0.101",
        "TERouterId": "62.0.0.101",
        "isoAddress": "0620.0000.0101",
        "area": "490062"
      }
    },
    "hostName": "vmx101-62",
    "timestamp": 1427130726939
  },
  {
    "topoObjectType": "node",
    "topologyIndex": 1,
```



```
"name": "62.0.0.101",
"nodeIndex": 1,
"AutonomousSystem": {
  "asNumber": 62
},
"layer": "IP",
"protocols": {
  "ISIS": {
    "routerId": "62.0.0.101",
    "TERouterId": "62.0.0.101",
    "isoAddress": "0620.0000.0101",
    "area": "490062"
  }
},
"hostName": "vmx101-62",
"timestamp": 1427130727025
},
{
  "topoObjectType": "node",
  "topologyIndex": 1,
  "name": "62.0.0.101",
  "nodeIndex": 1,
  "AutonomousSystem": {
    "asNumber": 62
  },
  "layer": "IP",
  "protocols": {
    "ISIS": {
      "routerId": "62.0.0.101",
      "TERouterId": "62.0.0.101",
      "isoAddress": "0620.0000.0101",
      "area": "490062"
    }
  },
  "hostName": "vmx101-62",
  "timestamp": 1427135395317
},
{
  "topoObjectType": "node",
  "topologyIndex": 1,
  "name": "62.0.0.101",
  "nodeIndex": 1,
  "AutonomousSystem": {
    "asNumber": 62
  },
  "layer": "IP",
  "protocols": {
    "ISIS": {
      "routerId": "62.0.0.101",
      "TERouterId": "62.0.0.101",
      "isoAddress": "0620.0000.0101",
      "area": "490062"
    }
  },
  "hostName": "CHI",
  "topology": {
    "coordinates": {
      "type": "Point",
      "coordinates": [
        4.29167,
```

```
    2.9
    ]
    }
  },
  "timestamp": 1427168112286
}
]
```

3.3.10. Create a List of Nodes

Method	URI	Description
POST	/topology/v2/{topology_id}/nodes/bulk	Creates several Nodes using the following JSON schema: node_v2.json#/definitions/createNodeList .

Normal response codes: 201

3.3.10.1. Request

This table shows the URI parameters for the create a list of nodes request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 3.18. Create a List of Nodes: JSON request

The request must contain a list of Nodes to be created. The individual node parameters are the same as creating an single node.

```
[
  {
    "name": "PlannedNode2",
    "topoObjectType": "node",
    "topologyIndex": 1
  },
  {
    "name": "PlannedNode3",
    "topoObjectType": "node",
    "topologyIndex": 1
  }
]
```

3.3.10.2. Response

Example 3.19. Create a List of Nodes: JSON response

```
[
  {
    "topoObjectType": "node",
    "topologyIndex": 1,
    "id": "PlannedNode2",
    "name": "PlannedNode2",
    "nodeIndex": 9,
    "layer": "IP",
    "live": false
  },
  {
    "topoObjectType": "node",
    "topologyIndex": 1,
    "id": "PlannedNode3",
    "name": "PlannedNode3",
    "nodeIndex": 10,
    "layer": "IP",
    "live": false
  }
]
```

3.3.11. Update a List of Nodes

Method	URI	Description
PUT	/topology/v2/{topology_id}/nodes/bulk	Updates several Nodes using the following JSON schema: node_v2.json#/definitions/nodeListUpdate .

Normal response codes: 202

3.3.11.1. Request

This table shows the URI parameters for the update a list of nodes request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 3.20. Update a List of Nodes: JSON request

The bulk update accepts a list of Node updates. Each entry requires the same parameters and logic as updating a single Node.

```
[
  {"nodeIndex": 9, "topoObjectType": "node", "topologyIndex": 1, "hostName" :
  "testHostName"},
  {"nodeIndex": 10, "topoObjectType": "node", "topologyIndex": 1, "comment" :
  "This is a comment"}
]
```

3.3.11.2. Response

Example 3.21. Update a List of Nodes: JSON response

The response contains a list of individual update responses (see Node update).

```
[
  {"topoObjectType": "node", "topologyIndex": 1, "id": "PlannedNode2",
  "nodeIndex": 9, "layer": "IP", "live": false, "hostName" : "testHostName"},
  {"topoObjectType": "node", "topologyIndex": 1, "id": "PlannedNode3",
  "nodeIndex": 10, "layer": "IP", "live": false, "comment" : "This is a comment"}
]
```

3.3.12. Update a List of Nodes using PATCH

Method	URI	Description
PATCH	/topology/v2/{topology_id}/nodes/bulk	Updates several Nodes using the following JSON schema: node_v2.json#/definitions/nodeListPatch .

Normal response codes: 202

3.3.12.1. Request

This table shows the URI parameters for the update a list of nodes using patch request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 3.22. Update a List of Nodes using PATCH : JSON request

The bulk PATCH accepts a list consisting of nodeIndex and patch:

```
[
  {"nodeIndex": 9, "patch" : [{"op" : "add", "path" : "/comment", "value":
    "Updated"}]},
  {"nodeIndex": 10, "patch" : [{"op" : "add", "path" : "/comment", "value":
    "Updated-1"}]}
]
```

3.3.12.2. Response

Example 3.23. Update a List of Nodes using PATCH : JSON response

The response contains a list of individual update responses.

```
[
  {"topoObjectType": "node", "topologyIndex": 1, "id": "PlannedNode2",
    "nodeIndex": 9, "layer": "IP", "live": false, "hostName" : "testHostName",
    "comment" : "Updated"},
  {"topoObjectType": "node", "topologyIndex": 1, "id": "PlannedNode3",
    "nodeIndex": 10, "layer": "IP", "live": false, "comment" : "Updated1"}
]
```

3.3.13. Delete a List of Nodes

Method	URI	Description
DELETE	/topology/v2/{topology_id}/nodes/bulk	Deletes a list of Nodes. The payload must conform to node_v2.json#/definitions/nodeListDelete

```
[
  {"nodeIndex": 9},
  {"nodeIndex": 10}
]
```

Normal response codes: 204

3.3.13.1. Request

This table shows the URI parameters for the delete a list of nodes request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.4. Links

Use these endpoints for access to links and information about them.

The link schema is: [link_v2.json](#) . The operations are:

- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/links> [GET : get all links, POST: create link]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/links/search> URL parameters : name, address, queryType=OR [GET : search links]]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/links/<linkId>> [GET: get property, PUT: modify link, DELETE]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/links/<linkId>/history> [GET : get link history]

Similar to Nodes, planned links can be added to Paragon PathFinder. Examples for the PE-CE links are detailed in the "Create a Planned Link" section below.

Method	URI	Description
GET	/topology/v2/{topology_id}/links	Returns a full list of links.
POST	/topology/v2/{topology_id}/links	Creates a link using optional and required parameters defined in the following schema: link_v2.json#/definitions/createLink .
GET	/topology/v2/{topology_id}/links/search{?name,address,queryType}	Searches the link list based on URI parameters. For example, search?name=62.101.105 must return one Link.

Method	URI	Description
GET	/topology/v2/{topology_id}/links/stream	See EventSource for format. The notifications send on that stream are only linkEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).
GET	/topology/v2/{topology_id}/links/{linkIndex}	Returns the details for a link.
PUT	/topology/v2/{topology_id}/links/{linkIndex}	Updates a planned link using the optional and required parameters defined in the following schema: link_v2.json#/definitions/updateLink .
PATCH	/topology/v2/{topology_id}/links/{linkIndex}	Updates a link using a RFC6902 patch: json-patch.json . The result of the patch must conform to link_v2.json#/definitions/updateLink . The REST server remove all operational parameters like operationalStatus, ..etc. .
DELETE	/topology/v2/{topology_id}/links/{linkIndex}	Deletes a link. Live links reappear on the next update from the Topology server.
GET	/topology/v2/{topology_id}/links/utilization	Returns a list of links only containing the endA/endZ utilization
GET	/topology/v2/{topology_id}/links/{linkIndex}/history{?start,end}	Returns the history for a Link.

3.4.1. Get Links

Method	URI	Description
GET	/topology/v2/{topology_id}/links	Returns a full list of links.

Normal response codes: 200

3.4.1.1. Request

This table shows the URI parameters for the get links request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.4.1.2. Response

Example 3.24. Get Links: JSON response

The example shows three links:

- One point-to-point link (link #6, named "L62.104.107.1_62.104.107.2").
- Two broadcast links (link 1 and 8). Broadcast links do not have protocols on one end (pseudo node end).

Broadcast links for MPLS-TE is not supported.

```
[
  {
    "endA": {
      "topoObjectType": "interface",
      "TEcolor": 0,
      "TEmetric": 10,
      "bandwidth": 10000000000,
      "ipv4Address": {
        "address": "62.104.107.1",
        "topoObjectType": "ipv4"
      },
      "node": {
        "id": "62.0.0.104",
        "topoObjectType": "node",
        "topologyIndex": 1
      },
      "protocols": {
        "ISIS": {
          "TEMetric": 10,
          "level": "L2",
          "srlgs": [{"srlgValue": 407}]
        },
        "RSVP": {"bandwidth": 10000000000 }
      },
      "srlgs": [{"srlgValue": 407}],
      "topoObjectType": "interface",
```



```
      "unreservedBw": [9989999616, 9989999616, 9989999616, 9989999616,
9489999872, 9489999872, 9489999872, 9489999872]
    },
    "endZ": {
      "topoObjectType": "interface",
      "TEcolor": 0,
      "TEmetric": 10,
      "bandwidth": 10000000000,
      "ipv4Address": {
        "address": "62.104.107.2",
        "topoObjectType": "ipv4"
      },
    },
    "node": {
      "id": "62.0.0.107",
      "topoObjectType": "node",
      "topologyIndex": 1
    },
    "protocols": {
      "ISIS": {
        "TEMetric": 10,
        "level": "L2",
        "srlgs": [{"srlgValue": 407}]
      },
      "RSVP": {"bandwidth": 10000000000}
    },
    "srlgs": [{"srlgValue": 407}],
    "topoObjectType": "interface",
    "unreservedBw": [9989999616, 9989999616, 9989999616, 9989999616,
9989999616, 9989999616, 9989999616, 9489999872]
  },
  "linkIndex": 6,
  "name": "L62.104.107.1_62.104.107.2",
  "operationalStatus": "Up",
  "topoObjectType": "link",
  "topologyIndex": 1
},
{
  "endA": {
    "topoObjectType": "interface",
    "node": {
      "topoObjectType": "node",
      "id": "0620.0000.0101.02",
      "topologyIndex": 1
    }
  },
  "endZ": {
    "topoObjectType": "interface",
    "TEcolor": 0,
    "TEmetric": 10,
    "bandwidth": 40000000000,
    "ipv4Address": {
      "address": "62.101.105.1",
      "topoObjectType": "ipv4"
    },
    "node": {
      "id": "62.0.0.101",
      "topoObjectType": "node",
      "topologyIndex": 1
    },
    "protocols": {
```

```

        "ISIS": {
            "TEMetric": 10,
            "level": "L2",
            "srlgs": [{"srlgValue": 100}]
        },
        "RSVP": {"bandwidth": 40000000000}
    },
    "srlgs": [{"srlgValue": 100}],
    "topoObjectType": "interface",
    "unreservedBw": [39990001664, 39489998848, 39489998848,
39489998848, 39489998848, 39489998848, 39489998848, 39489998848]
    },
    "linkIndex": 1,
    "name": "L_62.101.105.1",
    "operationalStatus": "Up",
    "topoObjectType": "link",
    "topologyIndex": 1
},
{
    "endA": {
        "topoObjectType": "interface",
        "node": {
            "topoObjectType": "node",
            "id": "0620.0000.0101.02",
            "topologyIndex": 1
        }
    },
    "endZ": {
        "topoObjectType": "interface",
        "TEcolor": 0,
        "TEmetric": 10,
        "bandwidth": 40000000000,
        "ipv4Address": {
            "address": "62.101.105.2",
            "topoObjectType": "ipv4"
        },
        "node": {
            "id": "62.0.0.105",
            "topoObjectType": "node",
            "topologyIndex": 1
        },
        "protocols": {
            "ISIS": {
                "TEMetric": 10,
                "level": "L2"
            },
            "RSVP": {"bandwidth": 40000000000}
        },
        "topoObjectType": "interface",
        "unreservedBw": [39570001920, 39570001920, 39570001920,
39070003200, 38570004480, 38570004480, 38570004480, 38570004480]
    },
    "linkIndex": 8,
    "name": "L_62.101.105.2",
    "operationalStatus": "Up",
    "topoObjectType": "link",
    "topologyIndex": 1
}
]

```

3.4.2. Create a Link

Method	URI	Description
POST	/topology/v2/{topology_id}/links	Creates a link using optional and required parameters defined in the following schema: link_v2.json#/definitions/createLink .

Normal response codes: 201

3.4.2.1. Request

This table shows the URI parameters for the create a link request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 3.25. Create a Link: JSON request

The example shows the creation of a numbered 40G planned link between node 62.0.0.102 and PlannedNode.

Table 3.3. Attributes required for Create Link

Attribute	Type	Description	Fixed
endA/node	string	No	Link source node
endZ/node	string	No	Link destination node
topologyIndex	integer	Yes	1
topoObjectType	string	Yes	link

Create Link examples:

- In this example, create planned link

```
{
  "topoObjectType": "link",
  "topologyIndex": 1,
  "endA": {
    "topoObjectType": "interface",
    "node": {
      "topoObjectType": "node", "name": "62.0.0.102", "topologyIndex": 1
    },
    "ipv4Address": { "topoObjectType": "ipv4", "address": "62.102.166.1"},
    "bandwidth": 40000000000,
    "TEmetric": 10,
    "protocols": {
      "RSVP": {
        "bandwidth": 40000000000
      }
    }
  },
  "endZ": {
    "topoObjectType": "interface",
    "node": {
```

```

    "topoObjectType": "node", "name": "PlannedNode", "topologyIndex": 1
  },
  "ipv4Address": { "topoObjectType": "ipv4", "address": "62.102.166.2"},
  "bandwidth": 40000000000,
  "TEmetric": 10,
  "protocols": {
    "RSVP": {
      "bandwidth": 40000000000
    }
  }
}
}

```

The response is:

```

{
  "topoObjectType": "link",
  "topologyIndex": 1,
  "name": "L62.102.166.1_62.102.166.2",
  "operationalStatus": "Planned",
  "linkIndex": 16,
  "endA": {
    "topoObjectType": "interface",
    "ipv4Address": {"topoObjectType": "ipv4", "address": "62.102.166.1"},
    "protocols": {
      "RSVP": {
        "bandwidth": 40000000000
      }
    }
  },
  "bandwidth": 40000000000,
  "node": {"topoObjectType": "node", "name": "62.0.0.102", "topologyIndex": 1}
},
  "endZ": {
    "topoObjectType": "interface",
    "ipv4Address": {"topoObjectType": "ipv4", "address": "62.102.166.2"},
    "protocols": {
      "RSVP": {
        "bandwidth": 40000000000
      }
    }
  },
  "bandwidth": 40000000000,
  "node": {"topoObjectType": "node", "name": "PlannedNode", "topologyIndex": 1}
}

```

- In this example, create a link between PE (id: 0100.0000.0003) and CE (id: CEZ1) node

```

{
  "topoObjectType": "link",
  "topologyIndex": 1,
  "endA": {
    "topoObjectType": "interface",
    "node": {
      "topoObjectType": "node",
      "id": "0100.0000.0003",
      "topologyIndex": 1
    }
  },
  "endZ": {
    "topoObjectType": "interface",

```

```
    "node":{
      "topoObjectType":"node",
      "id":"CEZ1",
      "topologyIndex":1
    }
  }
}
```

The response is:

```
{
  "endZ":{
    "node":{
      "topologyIndex":1,
      "id":"CEZ1",
      "topoObjectType":"node"
    },
    "topoObjectType":"interface"
  },
  "linkIndex":74,
  "endA":{
    "node":{
      "topologyIndex":1,
      "id":"0100.0000.0003",
      "topoObjectType":"node"
    },
    "topoObjectType":"interface"
  },
  "live":false,
  "operationalStatus":"Unknown",
  "topologyIndex":1,
  "location":"/NorthStar/API/v2/tenant/1/topology/1/links/74",
  "topoObjectType":"link",
  "id":"L0100.0000.0003-CEZ1"
}
```

- In this example, create a link between PE (id: 0100.0000.0003) and Site (id: SiteA1) node

```
{
  "topoObjectType":"link",
  "topologyIndex":1,
  "endA":{
    "topoObjectType":"interface",
    "node":{
      "topoObjectType":"node",
      "id":"0100.0000.0003",
      "topologyIndex":1
    }
  },
  "endZ":{
    "topoObjectType":"interface",
    "node":{
      "topoObjectType":"node",
      "id":"SiteA1",
      "topologyIndex":1
    }
  }
}
```

The response is:

```
{
  "endZ": {
    "node": {
      "topologyIndex": 1,
      "id": "SiteA1",
      "topoObjectType": "node"
    },
    "topoObjectType": "interface"
  },
  "linkIndex": 86,
  "endA": {
    "node": {
      "topologyIndex": 1,
      "id": "0100.0000.0003",
      "topoObjectType": "node"
    },
    "topoObjectType": "interface"
  },
  "live": false,
  "operationalStatus": "Unknown",
  "topologyIndex": 1,
  "location": "/NorthStar/API/v2/tenant/1/topology/1/links/86",
  "topoObjectType": "link",
  "id": "L0100.0000.0003_SiteA1"
}
```

- In this example, create a link between CE (id: CEZ1) and Site (id: SiteZ) node

```
{
  "topoObjectType": "link",
  "topologyIndex": 1,
  "endA": {
    "topoObjectType": "interface",
    "node": {
      "topoObjectType": "node",
      "id": "CEZ1",
      "topologyIndex": 1
    }
  },
  "endZ": {
    "topoObjectType": "interface",
    "node": {
      "topoObjectType": "node",
      "id": "SiteZ",
      "topologyIndex": 1
    }
  }
}
```

The response is:

```
{
  "endZ": {
    "node": {
      "topologyIndex": 1,
      "id": "SiteZ",
      "topoObjectType": "node"
    },
    "topoObjectType": "interface"
  }
}
```

```
    },
    "linkIndex": 98,
    "endA": {
      "node": {
        "topologyIndex": 1,
        "id": "CEZ1",
        "topoObjectType": "node"
      },
      "topoObjectType": "interface"
    },
    "live": false,
    "operationalStatus": "Unknown",
    "topologyIndex": 1,
    "location": "/NorthStar/API/v2/tenant/1/topology/1/links/98",
    "topoObjectType": "link",
    "id": "LCEZ1_SiteZ"
  }
}
```

3.4.2.2. Response

The responses are shown per example in the request section. In general the response is a JSON document conforming to [link_v2.json#/definitions/createLink](#).

This operation does not return a response body.

3.4.3. Search Links

Method	URI	Description
GET	/topology/v2/{topology_id}/links/search{?name,address,queryType}	Searches the link list based on URI parameters. For example, search?name=62.101.105 must return one Link.

Normal response codes: 200

3.4.3.1. Request

This table shows the URI parameters for the search links request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This table shows the query parameters for the search links request:

Name	Type	Description
name	String (Optional)	Filters based on the link name. The parameter is treated as a regular expression. Simple strings such as L62 are accepted. JavaScript regexp (for example, /^L62/) is also accepted
address	String (Optional)	Filters based on the link address (endA/ipv4Address/address and endZ/ipv4Address/address).
queryType	String (Optional)	The queryType parameter sets the logical operator to be used between the query parameters. By default, the queryType is AND. If the queryType is set to OR, a link is included in the results if any of the query parameters matches.

3.4.3.2. Response

Example 3.26. Search Links: JSON response

```
[
  {
    "endA": {
      "topoObjectType": "interface",
      "TEcolor": 0,
      "TEmetric": 10,
      "bandwidth": 10000000000,
      "ipv4Address": {
        "address": "62.104.107.1",
        "topoObjectType": "ipv4"
      },
      "node": {
        "id": "62.0.0.104",
        "topoObjectType": "node",
        "topologyIndex": 1
      },
      "protocols": {
        "ISIS": {
          "TEMetric": 10,
          "level": "L2",
          "srlgs": [{"srlgValue": 407}]
        },
        "RSVP": {"bandwidth": 10000000000 }
      }
    }
  }
]
```



```

    },
    "srlgs": [{"srlgValue": 407}],
    "topoObjectType": "interface",
    "unreservedBw": [9989999616, 9989999616, 9989999616, 9989999616,
9489999872, 9489999872, 9489999872, 9489999872]
  },
  "endZ": {
    "topoObjectType": "interface",
    "TEcolor": 0,
    "TEmetric": 10,
    "bandwidth": 10000000000,
    "ipv4Address": {
      "address": "62.104.107.2",
      "topoObjectType": "ipv4"
    },
  },
  "node": {
    "id": "62.0.0.107",
    "topoObjectType": "node",
    "topologyIndex": 1
  },
  "protocols": {
    "ISIS": {
      "TEMetric": 10,
      "level": "L2",
      "srlgs": [{"srlgValue": 407}]
    },
    "RSVP": {"bandwidth": 10000000000}
  },
  "srlgs": [{"srlgValue": 407}],
  "topoObjectType": "interface",
  "unreservedBw": [9989999616, 9989999616, 9989999616, 9989999616,
9989999616, 9989999616, 9989999616, 9489999872]
},
"linkIndex": 6,
"name": "L62.104.107.1_62.104.107.2",
"operationalStatus": "Up",
"topoObjectType": "link",
"topologyIndex": 1
},
{
  "endA": {
    "topoObjectType": "interface",
    "node": {
      "topoObjectType": "node",
      "id": "0620.0000.0101.02",
      "topologyIndex": 1
    }
  },
  "endZ": {
    "topoObjectType": "interface",
    "TEcolor": 0,
    "TEmetric": 10,
    "bandwidth": 40000000000,
    "ipv4Address": {
      "address": "62.101.105.1",
      "topoObjectType": "ipv4"
    },
  },
  "node": {
    "id": "62.0.0.101",
    "topoObjectType": "node",

```

```

        "topologyIndex": 1
      },
      "protocols": {
        "ISIS": {
          "TEMetric": 10,
          "level": "L2",
          "srlgs": [{"srlgValue": 100}]
        },
        "RSVP": {"bandwidth": 40000000000}
      },
      "srlgs": [{"srlgValue": 100}],
      "topoObjectType": "interface",
      "unreservedBw": [39990001664, 39489998848, 39489998848,
39489998848, 39489998848, 39489998848, 39489998848, 39489998848]
    },
    "linkIndex": 1,
    "name": "L_62.101.105.1",
    "operationalStatus": "Up",
    "topoObjectType": "link",
    "topologyIndex": 1
  },
  {
    "endA": {
      "topoObjectType": "interface",
      "node": {
        "topoObjectType": "node",
        "id": "0620.0000.0101.02",
        "topologyIndex": 1
      }
    },
    "endZ": {
      "topoObjectType": "interface",
      "TEcolor": 0,
      "TEmetric": 10,
      "bandwidth": 40000000000,
      "ipv4Address": {
        "address": "62.101.105.2",
        "topoObjectType": "ipv4"
      },
      "node": {
        "id": "62.0.0.105",
        "topoObjectType": "node",
        "topologyIndex": 1
      },
      "protocols": {
        "ISIS": {
          "TEMetric": 10,
          "level": "L2"
        },
        "RSVP": {"bandwidth": 40000000000}
      },
      "topoObjectType": "interface",
      "unreservedBw": [39570001920, 39570001920, 39570001920,
39070003200, 38570004480, 38570004480, 38570004480, 38570004480]
    },
    "linkIndex": 8,
    "name": "L_62.101.105.2",
    "operationalStatus": "Up",
    "topoObjectType": "link",
    "topologyIndex": 1
  }
}

```

```
}  
]
```

3.4.4. Start a SSE Stream

Method	URI	Description
GET	/topology/v2/{topology_id}/links/stream	See EventSource for format. The notifications send on that stream are only linkEvent. The data will contain a JSON document (see Paragon Pathfinder Notification API).

Normal response codes: 200

3.4.4.1. Request

This table shows the URI parameters for the start a sse stream request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon Pathfinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.4.5. Get a Single Link

Method	URI	Description
GET	/topology/v2/{topology_id}/links/{linkIndex}	Returns the details for a link.

Normal response codes: 200

3.4.5.1. Request

This table shows the URI parameters for the get a single link request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{linkIndex}	Int	The unique linkIndex.

This operation does not accept a request body.

3.4.5.2. Response

Example 3.27. Get a Single Link: JSON response

```
{
  "endA": {
    "TEcolor": 0,
    "TEmetric": 10,
    "delay" : 600,
    "bandwidth": 40000000000,
    "ipv4Address": {
      "address": "192.0.2.13",
      "topoObjectType": "ipv4"
    },
    "node": {
      "id": "192.0.2.3",
      "topoObjectType": "node",
      "topologyIndex": 1
    },
    "protocols": {
      "ISIS": {
        "TEMetric": 10,
        "level": "L2"
      },
      "RSVP": {
        "bandwidth": 40000000000
      }
    },
    "topoObjectType": "interface",
    "unreservedBw": [39970000896, 39970000896, 39970000896, 39469998080,
39469998080, 39469998080, 39469998080, 39469998080]
  },
  "endZ": {
    "TEcolor": 0,
    "TEmetric": 10,
    "delay": 600,
```

```
    "bandwidth": 40000000000,
    "ipv4Address": {
      "address": "192.0.2.14",
      "topoObjectType": "ipv4"
    },
    "node": {
      "id": "192.0.2.7",
      "topoObjectType": "node",
      "topologyIndex": 1
    },
    "protocols": {
      "ISIS": {
        "TEMetric": 10,
        "level": "L2"
      },
      "RSVP": {
        "bandwidth": 40000000000
      }
    },
    "unreservedBw": [39289999360, 38790000640, 38790000640, 38790000640,
38790000640, 38790000640, 38790000640, 38790000640]
  },
  "linkIndex": 1,
  "id": "L192.0.2.13_192.0.2.14",
  "operationalStatus": "Up",
  "topoObjectType": "link",
  "topologyIndex": 1
}
```

3.4.6. Update Link

Method	URI	Description
PUT	/topology/v2/{topology_id}/links/{linkIndex}	Updates a planned link using the optional and required parameters defined in the following schema: link_v2.json#/definitions/updateLink .

Normal response codes: 202

3.4.6.1. Request

This table shows the URI parameters for the update link request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{linkIndex}	Int	The unique linkIndex.

Example 3.28. Update Link : JSON request

The following example sets the link delay for the link L62.103.107.1_62.103.107.2

```
{
  "topoObjectType": "link", "topologyIndex": 1,
  "linkIndex": 4,
  "endA": {
    "topoObjectType": "interface",
    "ipv4Address": { "topoObjectType": "ipv4", "address": "62.103.107.1" },
    "protocols": {
      "ISIS": {
        "TEMetric": 10,
        "level": "L2",
        "metric": 10
      },
      "RSVP": { "bandwidth": 40000000000 }
    },
    "bandwidth": 40000000000,
    "TEMetric": 10,
    "TEcolor": 0,
    "delay": 600,
    "node": { "topoObjectType": "node", "name": "62.0.0.103", "topologyIndex": 1 }
  }, "endZ": {
    "topoObjectType": "interface",
    "ipv4Address": { "topoObjectType": "ipv4", "address": "62.103.107.2" },
    "protocols": {
      "ISIS": {
        "TEMetric": 10,
        "level": "L2",
        "metric": 10
      },
      "RSVP": { "bandwidth": 40000000000 }
    },
    "bandwidth": 40000000000,
    "TEMetric": 10,
    "TEcolor": 0,
    "delay": 600,

```

```
"node":{"topoObjectType":"node","name":"62.0.0.107","topologyIndex":1}
}
```

3.4.6.2. Response

Example 3.29. Update Link : JSON response

```
{
  "topoObjectType":"link", "topologyIndex":1,
  "name":"L62.103.107.1_62.103.107.2",
  "linkIndex":4,
  "operationalStatus": "Up",
  "endA":{"
    "topoObjectType":"interface",
    "ipv4Address" : {"topoObjectType":"ipv4","address":"62.103.107.1"},
    "protocols":{"
      "ISIS" : {
        "TEMetric":10,
        "level":"L2",
        "metric":10
      },
      "RSVP":{"bandwidth":400000000000}
    },
    "bandwidth":400000000000,
    "TEMetric":10,
    "TEcolor":0,
    "delay":600,
    "node":{"topoObjectType":"node","name":"62.0.0.103","topologyIndex":1}
  }, "endZ" : {
    "topoObjectType":"interface",
    "ipv4Address":{"topoObjectType":"ipv4","address":"62.103.107.2"},
    "protocols":{"
      "ISIS":{"
        "TEMetric":10,
        "level":"L2",
        "metric":10
      },
      "RSVP":{"bandwidth":400000000000}
    },
    "bandwidth":400000000000,
    "TEMetric":10,
    "TEcolor":0,
    "delay":600,
    "node":{"topoObjectType":"node","name":"62.0.0.107","topologyIndex":1}
  }
}
```


3.4.7. Patch Link

Method	URI	Description
PATCH	/topology/v2/{topology_id}/links/{linkIndex}	Updates a link using a RFC6902 patch: json-patch.json . The result of the patch must conform to link_v2.json#/definitions/updateLink . The REST server remove all operational parameters like operationalStatus, ..etc. .

Normal response codes: 202

3.4.7.1. Request

This table shows the URI parameters for the patch link request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{linkIndex}	Int	The unique linkIndex.

Example 3.30. Patch Link: JSON request

The Patched update accepts the same parameters as the update

```
[{"op" : "add", "path" : "/endZ/designParameters/userCost", "value": 40}]
```

3.4.7.2. Response

Example 3.31. Patch Link: JSON response

```
{
  "topoObjectType": "link",
  "linkIndex": 10,
  "id": "L11.101.202.1_11.101.202.2",
  "name": "L11.101.202.1_11.101.202.2",
  "live": false,
  "topologyIndex": 1,
  "operationalStatus": "Unknown",
  "endA": {
    "topoObjectType": "interface",
    "ipv4Address": {
      "topoObjectType": "ipv4",
      "address": "11.101.202.1"
    },
  },
  "node": {
    "topoObjectType": "node",
    "id": "11.0.0.101",
    "topologyIndex": 1
  },
},
"endZ": {
  "topoObjectType": "interface",
  "ipv4Address": {
    "topoObjectType": "ipv4",
    "address": "11.101.202.2"
  },
},
}
```

```
    "node": {  
      "topoObjectType": "node",  
      "id": "11.0.0.103",  
      "topologyIndex": 1  
    },  
    "designParameters" : {"userCost":40}  
  }  
}
```

3.4.8. Delete a Link

Method	URI	Description
DELETE	/topology/v2/{topology_id}/links/{linkIndex}	Deletes a link. Live links reappear on the next update from the Topology server.

Normal response codes: 204

3.4.8.1. Request

This table shows the URI parameters for the delete a link request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{linkIndex}	Int	The unique linkIndex.

This operation does not accept a request body.

3.4.9. Get Links utilization only

Method	URI	Description
GET	/topology/v2/{topology_id}/links/utilization	Returns a list of links only containing the endA/endZ utilization

Normal response codes: 200

3.4.9.1. Request

This table shows the URI parameters for the get links utilization only request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.4.9.2. Response

Example 3.32. Get Links utilization only: JSON response

The example shows three links:

- One point-to-point link (link #6, named "L62.104.107.1_62.104.107.2").
- Two broadcast links (link 1 and 8). Broadcast links do not have protocols on one end (pseudo node end).

Broadcast links for MPLS-TE is not supported.

```
[
  {
    "endA": {
      "topoObjectType": "interface",
      "node": {"id": "62.0.0.104", "topoObjectType": "node"},
      "topologyIndex": 1},
      "utilization" : {
        "lastUpdate" : "2016-03-24T01:37:24.192474Z",
        "utilization" : 10,
        "availableBandwidth" : 9000000000,
        "usedBandwidth" : 1000000000,
        "tunnelUtilization" : 15,
        "tunnelLiveUtilization" : 10
      }
    },
    "endZ": {
      "topoObjectType": "interface",
      "node": {"id": "62.0.0.107", "topoObjectType": "node"},
      "topologyIndex": 1},
      "utilization" : {
        "lastUpdate" : "2016-03-24T01:37:24.192474Z",
        "utilization" : 20,
        "availableBandwidth" : 8000000000,
        "usedBandwidth" : 2000000000,

```

```
        "tunnelUtilization"      : 15,
        "tunnelLiveUtilization" : 20
    },
    },
    "linkIndex": 6,
    "name": "L62.104.107.1_62.104.107.2",
    "topoObjectType": "link",
    "topologyIndex": 1
},
{
    "endA": {
        "topoObjectType": "interface",
        "node": {"id": "62.0.0.102", "topoObjectType": "node",
"topologyIndex": 1},
        "utilization" : {
            "lastUpdate" : "2016-03-24T01:37:24.192474Z",
            "utilization" : 20,
            "availableBandwidth"      : 8000000000,
            "usedBandwidth"           : 2000000000,
            "tunnelUtilization"       : 20,
            "tunnelLiveUtilization"   : 20
        }
    },
    "endZ": {
        "topoObjectType": "interface",
        "node": {"id": "62.0.0.105", "topoObjectType": "node",
"topologyIndex": 1},
        "utilization" : {
            "lastUpdate" : "2016-03-24T01:37:24.192474Z",
            "utilization" : 20,
            "availableBandwidth"      : 8000000000,
            "usedBandwidth"           : 2000000000,
            "tunnelUtilization"       : 15,
            "tunnelLiveUtilization"   : 20
        }
    },
    "linkIndex": 6,
    "name": "L62.102.105.1_62.102.105.2",
    "topoObjectType": "link",
    "topologyIndex": 1
}
]
```

3.4.10. Get the Link Event History

Method	URI	Description
GET	/topology/v2/{topology_id}/links/{linkIndex}/history{?start,end}	Returns the history for a Link.

The history contains a list of Unix-timestamped events for the link resource.

Normal response codes: 200

3.4.10.1. Request

This table shows the URI parameters for the get the link event history request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{linkIndex}	Int	The unique linkIndex.

This table shows the query parameters for the get the link event history request:

Name	Type	Description
start	Int (Optional)	Start timestamp: Include events with the starting time and later.
end	Int (Optional)	End timestamp: Include events before (but not including) the ending timestamp.

3.4.10.2. Response

Example 3.33. Get the Link Event History: JSON response

```
[
  {
    "topoObjectType": "link",
    "topologyIndex": 1,
    "name": "L62.103.107.1_62.103.107.2",
    "operationalStatus": "Up",
    "linkIndex": 4,
    "endA": {
      "topoObjectType": "interface",
      "ipv4Address": {"topoObjectType": "ipv4", "address": "62.103.107.1" },
      "protocols": {
        "ISIS": {
          "TEMetric": 10,
          "level": "L2",
          "metric": 10
        },
        "RSVP": {"bandwidth": 40000000000}
      },
      "bandwidth": 40000000000,
      "TEMetric": 10,
      "TEcolor": 0,
      "unreservedBw": [40000000000, 40000000000, 40000000000, 40000000000,
        40000000000, 40000000000, 40000000000, 40000000000],
    }
  }
]
```

```

    "node": {"topoObjectType": "node", "name": "62.0.0.103", "topologyIndex":
1}
  },
  "endZ": {
    "topoObjectType": "interface",
    "ipv4Address": {"topoObjectType": "ipv4", "address": "62.103.107.2"},
    "protocols": {
      "ISIS": {
        "TEMetric": 10,
        "level": "L2",
        "metric": 10
      },
      "RSVP": {"bandwidth": 40000000000}
    },
    "bandwidth": 40000000000,
    "TEMetric": 10,
    "TEcolor": 0,
    "unreservedBw": [39469998080, 38969999360, 38470000640, 38470000640,
37970001920, 37970001920, 37970001920, 37970001920],
    "node": {"topoObjectType": "node", "name": "62.0.0.107", "topologyIndex":
1}
  },
  "timestamp": 1427129349771
},
{
  "topoObjectType": "link",
  "topologyIndex": 1,
  "name": "L62.103.107.1_62.103.107.2",
  "operationalStatus": "Up",
  "linkIndex": 4,
  "endA": {
    "topoObjectType": "interface",
    "ipv4Address": {"topoObjectType": "ipv4", "address": "62.103.107.1"},
    "protocols": {
      "ISIS": {
        "TEMetric": 10,
        "level": "L2",
        "metric": 10
      },
      "RSVP": {"bandwidth": 40000000000}
    },
    "bandwidth": 40000000000,
    "TEMetric": 10,
    "TEcolor": 0,
    "unreservedBw": [39970000896, 39970000896, 39970000896, 38470000640,
38470000640, 38470000640, 38470000640, 38470000640],
    "node": {"topoObjectType": "node", "name": "62.0.0.103", "topologyIndex":
1}
  },
  "endZ": {
    "topoObjectType": "interface",
    "ipv4Address": {"topoObjectType": "ipv4", "address": "62.103.107.2"},
    "protocols": {
      "ISIS": {
        "TEMetric": 10,
        "level": "L2",
        "metric": 10
      },
      "RSVP": {"bandwidth": 40000000000}
    },
    "bandwidth": 40000000000,
    "TEMetric": 10,
    "TEcolor": 0,
    "unreservedBw": [39970000896, 39970000896, 39970000896, 38470000640,
38470000640, 38470000640, 38470000640, 38470000640],
    "node": {"topoObjectType": "node", "name": "62.0.0.107", "topologyIndex":
1}
  },
  "timestamp": 1427129349771
}

```

```

    "bandwidth": 40000000000,
    "TEmetric": 10,
    "TEcolor": 0,
    "unreservedBw": [39469998080, 38969999360, 38470000640, 38470000640,
37970001920, 37970001920, 37970001920, 37970001920],
    "node": {"topoObjectType": "node", "name": "62.0.0.107", "topologyIndex":
1}
  },
  "timestamp": 1427129349871
}
]

```

3.5. TE-LSPs

Use these endpoints to access TE-LSPs and the related information.

The TE-LSP schema is: [lsp.json](#) .

You can perform the following operations using the TE-LSP endpoints:

- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/te-lsps/> [GET : get all te-lsps, POST : create one LSP]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/te-lsps/search> search parameters : name={nameFilter}, from={fromIpv4} operStatus={operationalStatus} queryType=OR [GET : search LSPs ()]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/te-lsps/bulk> Bulk LSP operations: allows to create/update/delete a list of te-lsps [POST : create LSPs , PUT/PATCH : update lsps, DELETE : delete]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/te-lsps/<lspIndex>> [GET : get te-lsp, PUT/PATCH : update, DELETE : delete]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/te-lsps/<lspIndex>/history> [GET : get historical index of te-lsps]

Note: Use this API to create point-to-point LSPs only. The API supports RSVP and SR LSPs. Starting from Northstar release 4.1 onwards, binding SID SR LSPs are supported. Do not use this API to create the P2MP members. Ensure that you use the P2MP resources to create P2MP resources.

The TE-LSP API does allow diverse LSPs to be computed. This is controlled using the following set of attributes:

- **diversityGroup**: the LSPs are grouped together using a diversityGroup. If LSP1 and LSP2 should be diverse to each other, they should be in the same diversityGroup.
- **diversityLevel** (site, srlg or link): the level of requested diversity. Paragon PathFinder will try to compute path that reach that diversity level, if this cannot be achieved, it will try to route the path as diverse as possible, unless minimumDiversityLevel is specified.
- **minimumDiversityLevel** (site, srlg or link): the minimum acceptable diversity level. If Paragon PathFinder cannot achieve diversity as good or better than minimumDiversityLevel, the routing is considered as failed.

Method	URI	Description
GET	/topology/v2/{topology_id}/te-lsps	Returns the TE-LSP list.
POST	/topology/v2/{topology_id}/te-lsps	Creates a TE-LSP using the following JSON schema: lsp.json#/definitions/createLSP . For example, protection and custom service mapping parameters set.
GET	/topology/v2/{topology_id}/te-lsps/search{?name, from, operationalStatus, queryType}	Performs a search in the LSP list based on the URI parameters. For example, "search?name=62.101.105" returns one link.
GET	/topology/v2/{topology_id}/te-lsps/stream	See EventSource for format. The notifications send on that stream are only lspEvent or lspTopologyEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).
GET	/topology/v2/{topology_id}/te-lsps/{lspIndex}	Returns the details for a TE-LSP.
PUT	/topology/v2/{topology_id}/te-lsps/{lspIndex}	Updates a TE-LSP using the JSON schema: lsp.json#/definitions/updateLSP .
PATCH	/topology/v2/{topology_id}/te-lsps/{lspIndex}	Updates a TE-LSP using a RFC6902 patch: json-patch.json . The result of the patch must conform to lsp.json#/definitions/updateLsp . The REST server remove all operational parameters like operationalStatus, ..etc. . Using an empty patch (empty list) will result in the LSP to be re-provisioned without parameter changed.
DELETE	/topology/v2/{topology_id}/te-lsps/{lspIndex}	Deletes a TE-LSP. This function is supported only on the PCE-initiated LSPs. PCC-controlled and PCC-delegated LSPs cannot be deleted from Paragon PathFinder. They must be deleted in the node.
POST	/topology/v2/{topology_id}/te-lsps/bulk	Creates several TE-LSPs using the following JSON schema: lsp.json#/definitions/createLSPList .
PUT	/topology/v2/{topology_id}/te-lsps/bulk	Updates several TE-LSPs using the following JSON schema: lsp.json#/definitions/lspListUpdate .
PATCH	/topology/v2/{topology_id}/te-lsps/bulk	Updates several TE-LSPs using the following JSON schema: lsp.json#/definitions/lspListPatch .
DELETE	/topology/v2/{topology_id}/te-lsps/bulk	Deletes a list of TE-LSPs. This function is supported only on the PCE-initiated LSPs. PCC-controlled and PCC-delegated LSPs cannot be deleted from Paragon PathFinder. They must be deleted in the node.b The payload must conform to lsp.json#/definitions/lspListDelete
GET	/topology/v2/{topology_id}/te-lsps/{lspIndex}/history{?start, end}	Returns the history for a TE-LSP.

3.5.1. Get TE-LSPs

Method	URI	Description
GET	/topology/v2/{topology_id}/te-lsps	Returns the TE-LSP list.

The example contains 10 TE-LSPs:

- A simple delegated LSP (index 1, named LSP_Node101_Node102). A delegated LSP is configured in the PCC and delegated to Paragon PathFinder. Paragon PathFinder controls the path and some attributes of the LSP.
- A delegated LSP with a configured explicit path (index 2, name LSP_101_103). The path name in the router is Path_Node101_Node103_Strict_1.
- A protected tunnel with a primary and secondary path. The two paths (each represented as a TE-LSP) of a tunnel (index 3 and 4). The tunnel is named LSP_Node102_Node104_with_secondary and each path has a pathName. The tunnel name is used for correlation. One path (index3) is the primary path, while the other is the secondary path. The secondary path will be activated upon primary path failure. LSPs carrying traffic are marked as "Active" rather than "Up".
- A protected tunnel with a primary and secondary standby path (index 5 and 6). The model is similar to a tunnel with a secondary path, the difference is that the standby path has a pathType set to standby and the path is signaled in the control plane (making its operational status "Up").
- A set of three TE-LSPs in a TE++ configuration (index 7,8,9). The TE-LSPs are named TEplusplus-Node102-Node103-<index>. Correlation of the entries can be done using live-Properties, options, and TEPlusPlusId. Those LSPs are not delegated to Paragon PathFinder, as indicated by the attribute controlType set to PCC. Paragon PathFinder will not modify those LSPs.
- A PCE-Initiated LSP (index 10).

Normal response codes: 200

3.5.1.1. Request

This table shows the URI parameters for the get te-lsps request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.5.1.2. Response

Example 3.34. Get TE-LSPs: JSON response

```
[
  {
    "operationalStatus": "Active",
    "plannedProperties": {
      "bandwidth": "0",
```

```

        "setupPriority": 7,
        "holdingPriority": 0,
        "calculatedEro": [
            {
                "topoObjectType": "ipv4",
                "address": "11.105.107.2",
                "loose": false
            },
            {
                "topoObjectType": "ipv4",
                "address": "11.106.107.1",
                "loose": false
            }
        ],
        "routingStatus": "Down",
        "adminStatus": "Up",
        "design": {
            "routingMethod": "routeByDevice"
        },
        "lastStatusString": "[PCServer]<Down controller_state=Path found
on down lsp",
        "controllerStatus": {
            "status": "Path found on down lsp"
        },
        "correlatedRR0HopCount": 2
    },
    "name": "rsvp-105-106",
    "from": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.105"
    },
    "pathType": "primary",
    "to": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.106"
    },
    "lspIndex": 13,
    "controlType": "PCC",
    "provisioningType": "RSVP",
    "collectedProperties": {
        "bandwidth": "0",
        "setupPriority": 7,
        "holdingPriority": 0,
        "calculatedEro": [
            {
                "topoObjectType": "ipv4",
                "address": "11.0.0.105",
                "loose": false
            },
            {
                "topoObjectType": "ipv4",
                "address": "11.105.106.2",
                "loose": false
            },
            {
                "topoObjectType": "ipv4",
                "address": "11.0.0.106",
                "loose": false
            }
        ]
    },
    ],

```

```
        "routingStatus": "Up",
        "adminStatus": "Up",
        "correlatedRR0HopCount": 1
    },
    "tunnelId": 60274,
    "liveProperties": {
        "bandwidth": 0,
        "metric": 10,
        "setupPriority": 7,
        "holdingPriority": 0,
        "operationalStatus": "Active",
        "adminStatus": "Up",
        "ero": [],
        "rro": [
            {
                "topoObjectType": "ipv4",
                "address": "11.105.106.2",
                "protectionInUse": false,
                "protectionAvailable": false
            }
        ]
    },
    "controller": "External"
},
{
    "operationalStatus": "Active",
    "name": "11.0.0.101:11.0.0.104:300:vpls:vpn_200",
    "p2mpIndex": 184549480,
    "p2mpName": "11.0.0.104:300:vpls:vpn_200",
    "from": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.104"
    },
    "to": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.101"
    },
    "liveProperties": {
        "bandwidth": 0,
        "metric": 0,
        "setupPriority": 7,
        "holdingPriority": 0,
        "operationalStatus": "Active",
        "adminStatus": "Up",
        "ero": [],
        "rro": [
            {
                "topoObjectType": "ipv4",
                "address": "11.104.107.2",
                "protectionInUse": false,
                "protectionAvailable": false
            },
            {
                "topoObjectType": "ipv4",
                "address": "11.105.107.1",
                "protectionInUse": false,
                "protectionAvailable": false
            },
            {
                "topoObjectType": "ipv4",
```

```

        "address": "11.101.105.1",
        "protectionInUse": false,
        "protectionAvailable": false
    }
    ],
    },
    "controlType": "PCC",
    "controller": "External",
    "pathType": "primary",
    "lspIndex": 14
},
{
    "operationalStatus": "Active",
    "name": "11.0.0.103:11.0.0.104:300:vpls:vpn_200",
    "p2mpIndex": 184549480,
    "p2mpName": "11.0.0.104:300:vpls:vpn_200",
    "from": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.104"
    },
    "to": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.103"
    },
    "liveProperties": {
        "bandwidth": 0,
        "metric": 0,
        "setupPriority": 7,
        "holdingPriority": 0,
        "operationalStatus": "Active",
        "adminStatus": "Up",
        "ero": [],
        "rro": [
            {
                "topoObjectType": "ipv4",
                "address": "11.104.107.2",
                "protectionInUse": false,
                "protectionAvailable": false
            },
            {
                "topoObjectType": "ipv4",
                "address": "11.103.107.1",
                "protectionInUse": false,
                "protectionAvailable": false
            }
        ]
    },
    "controlType": "PCC",
    "controller": "External",
    "pathType": "primary",
    "lspIndex": 15
},
{
    "operationalStatus": "Active",
    "name": "11.0.0.106:11.0.0.104:300:vpls:vpn_200",
    "p2mpIndex": 184549480,
    "p2mpName": "11.0.0.104:300:vpls:vpn_200",
    "from": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.104"
    }
}

```

```
    },
    "to": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.106"
    },
    "liveProperties": {
      "bandwidth": 0,
      "metric": 0,
      "setupPriority": 7,
      "holdingPriority": 0,
      "operationalStatus": "Active",
      "adminStatus": "Up",
      "ero": [],
      "rro": [
        {
          "topoObjectType": "ipv4",
          "address": "11.104.107.2",
          "protectionInUse": false,
          "protectionAvailable": false
        },
        {
          "topoObjectType": "ipv4",
          "address": "11.105.107.1",
          "protectionInUse": false,
          "protectionAvailable": false
        },
        {
          "topoObjectType": "ipv4",
          "address": "11.105.106.2",
          "protectionInUse": false,
          "protectionAvailable": false
        }
      ]
    },
    "controlType": "PCC",
    "controller": "External",
    "pathType": "primary",
    "lspIndex": 16
  },
  {
    "operationalStatus": "Active",
    "plannedProperties": {
      "bandwidth": "0",
      "setupPriority": 7,
      "holdingPriority": 0,
      "calculatedEro": [
        {
          "topoObjectType": "ipv4",
          "address": "11.104.107.2",
          "loose": false
        },
        {
          "topoObjectType": "ipv4",
          "address": "11.105.107.1",
          "loose": false
        },
        {
          "topoObjectType": "ipv4",
          "address": "11.101.105.1",
          "loose": false
        }
      ]
    }
  }
}
```

```
    }
  ],
  "routingStatus": "Up",
  "adminStatus": "Up",
  "lastStatusString": "[PCServer]<PCC ACK request_id=417990091",
  "correlatedRR0HopCount": 3
},
"name": "Silver-104-101",
"from": {
  "topoObjectType": "ipv4",
  "address": "11.0.0.104"
},
"pathType": "primary",
"to": {
  "topoObjectType": "ipv4",
  "address": "11.0.0.101"
},
"lspIndex": 17,
"controlType": "Delegated",
"provisioningType": "RSVP",
"initiator": "PCC",
"controller": "NorthStar",
"collectedProperties": {
  "bandwidth": "0",
  "setupPriority": 7,
  "holdingPriority": 0,
  "calculatedEro": [
    {
      "topoObjectType": "ipv4",
      "address": "11.0.0.104",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.104.107.2",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.105.107.1",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.101.105.1",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.0.0.101",
      "loose": false
    }
  ]
},
"routingStatus": "Up",
"adminStatus": "Up",
"correlatedRR0HopCount": 3
},
"tunnelId": 32533,
"liveProperties": {
  "bandwidth": 0,
```

```
"metric": 30,
"setupPriority": 7,
"holdingPriority": 0,
"operationalStatus": "Active",
"adminStatus": "Up",
"ero": [
  {
    "topoObjectType": "ipv4",
    "address": "11.104.107.2",
    "loose": false
  },
  {
    "topoObjectType": "ipv4",
    "address": "11.105.107.1",
    "loose": false
  },
  {
    "topoObjectType": "ipv4",
    "address": "11.101.105.1",
    "loose": false
  }
],
"rro": [
  {
    "topoObjectType": "ipv4",
    "address": "11.104.107.2",
    "protectionInUse": false,
    "protectionAvailable": false
  },
  {
    "topoObjectType": "ipv4",
    "address": "11.105.107.1",
    "protectionInUse": false,
    "protectionAvailable": false
  },
  {
    "topoObjectType": "ipv4",
    "address": "11.101.105.1",
    "protectionInUse": false,
    "protectionAvailable": false
  }
]
},
{
  "operationalStatus": "Active",
  "plannedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 0,
    "calculatedEro": [
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.105.107.1",
        "loose": false
      }
    ]
  }
}
```



```
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.102.105.1",
      "loose": false
    }
  ],
  "routingStatus": "Up",
  "adminStatus": "Up",
  "lastStatusString": "[PCServer]<PCC ACK request_id=417990082",
  "correlatedRR0HopCount": 3
},
"name": "Silver-104-102",
"from": {
  "topoObjectType": "ipv4",
  "address": "11.0.0.104"
},
"pathType": "primary",
"to": {
  "topoObjectType": "ipv4",
  "address": "11.0.0.102"
},
"lspIndex": 18,
"controlType": "Delegated",
"provisioningType": "RSVP",
"initiator": "PCC",
"controller": "NorthStar",
"collectedProperties": {
  "bandwidth": "0",
  "setupPriority": 7,
  "holdingPriority": 0,
  "calculatedEro": [
    {
      "topoObjectType": "ipv4",
      "address": "11.0.0.104",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.104.107.2",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.105.107.1",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.102.105.1",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.0.0.102",
      "loose": false
    }
  ]
},
"routingStatus": "Up",
"adminStatus": "Up",
```

```
        "correlatedRR0HopCount": 3
      },
      "tunnelId": 29910,
      "liveProperties": {
        "bandwidth": 0,
        "metric": 30,
        "setupPriority": 7,
        "holdingPriority": 0,
        "operationalStatus": "Active",
        "adminStatus": "Up",
        "ero": [
          {
            "topoObjectType": "ipv4",
            "address": "11.104.107.2",
            "loose": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.105.107.1",
            "loose": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.102.105.1",
            "loose": false
          }
        ],
        "rro": [
          {
            "topoObjectType": "ipv4",
            "address": "11.104.107.2",
            "protectionInUse": false,
            "protectionAvailable": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.105.107.1",
            "protectionInUse": false,
            "protectionAvailable": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.102.105.1",
            "protectionInUse": false,
            "protectionAvailable": false
          }
        ]
      }
    },
    {
      "operationalStatus": "Active",
      "plannedProperties": {
        "bandwidth": "0",
        "setupPriority": 7,
        "holdingPriority": 0,
        "calculatedEro": [
          {
            "topoObjectType": "ipv4",
            "address": "11.104.107.2",
            "loose": false
```

```
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.103.107.1",
      "loose": false
    }
  ],
  "routingStatus": "Up",
  "adminStatus": "Up",
  "lastStatusString": "[PCServer]<PCC ACK request_id=417990087",
  "correlatedRR0HopCount": 2
},
"name": "Silver-104-103",
"from": {
  "topoObjectType": "ipv4",
  "address": "11.0.0.104"
},
"pathType": "primary",
"to": {
  "topoObjectType": "ipv4",
  "address": "11.0.0.103"
},
"lspIndex": 19,
"controlType": "Delegated",
"provisioningType": "RSVP",
"initiator": "PCC",
"controller": "NorthStar",
"collectedProperties": {
  "bandwidth": "0",
  "setupPriority": 7,
  "holdingPriority": 0,
  "calculatedEro": [
    {
      "topoObjectType": "ipv4",
      "address": "11.0.0.104",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.104.107.2",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.103.107.1",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.0.0.103",
      "loose": false
    }
  ]
},
"routingStatus": "Up",
"adminStatus": "Up",
"correlatedRR0HopCount": 2
},
"tunnelId": 32451,
"liveProperties": {
  "bandwidth": 0,
```

```

    "metric": 20,
    "setupPriority": 7,
    "holdingPriority": 0,
    "operationalStatus": "Active",
    "adminStatus": "Up",
    "ero": [
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.103.107.1",
        "loose": false
      }
    ],
    "rro": [
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.2",
        "protectionInUse": false,
        "protectionAvailable": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.103.107.1",
        "protectionInUse": false,
        "protectionAvailable": false
      }
    ]
  },
  {
    "operationalStatus": "Active",
    "plannedProperties": {
      "bandwidth": "0",
      "setupPriority": 7,
      "holdingPriority": 0,
      "calculatedEro": [
        {
          "topoObjectType": "ipv4",
          "address": "11.104.107.2",
          "loose": false
        },
        {
          "topoObjectType": "ipv4",
          "address": "11.105.107.1",
          "loose": false
        }
      ],
      "routingStatus": "Down",
      "adminStatus": "Up",
      "design": {
        "routingMethod": "routeByDevice"
      },
      "lastStatusString": "[PCServer]<Down controller_state=Path found
on down lsp",
      "controllerStatus": {
        "status": "Path found on down lsp"
      }
    }
  }
}

```

```
    },
    "correlatedRR0HopCount": 2
  },
  "name": "rsvp-104-105",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.104"
  },
  "pathType": "primary",
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.105"
  },
  "lspIndex": 20,
  "controlType": "PCC",
  "provisioningType": "RSVP",
  "collectedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 0,
    "calculatedEro": [
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.104",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.105.107.1",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.105",
        "loose": false
      }
    ],
    "routingStatus": "Up",
    "adminStatus": "Up",
    "correlatedRR0HopCount": 2
  },
  "tunnelId": 32072,
  "liveProperties": {
    "bandwidth": 0,
    "metric": 20,
    "setupPriority": 7,
    "holdingPriority": 0,
    "operationalStatus": "Active",
    "adminStatus": "Up",
    "ero": [],
    "rro": [
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.2",
        "protectionInUse": false,
```

```
        "protectionAvailable": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.105.107.1",
        "protectionInUse": false,
        "protectionAvailable": false
      }
    ]
  },
  "controller": "External"
}
```

3.5.2. Create a TE-LSP

Method	URI	Description
POST	/topology/v2/{topology_id}/te-lsp	Creates a TE-LSP using the following JSON schema: lsp.json#/definitions/createLSP . For example, protection and custom service mapping parameters set.

Normal response codes: 201

Error response codes: 400

3.5.2.1. Request

This table shows the URI parameters for the create a te-lsp request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 3.35. Create a TE-LSP : JSON request

Table 3.4. Create LSP Required Attributes.

Attribute	Type	Description	Fixed
name	string	No	Tunnel name
from/address	string	No	Tunnel ingress
to/address	string	No	Tunnel egress

The following examples show different set of parameters and results:

- LSP with a bandwidth of 100M

```
{
  "name": "Rest_LSP_1",
  "from": {
    "topoObjectType": "ipv4",
    "address": "62.0.0.101"
  },
  "to": {
    "topoObjectType": "ipv4",
    "address": "62.0.0.103"
  },
  "plannedProperties": {
    "bandwidth": "100M",
    "setupPriority": 7,
    "holdingPriority": 7
  }
}
```

- Primary and standby LSPs. The source (from), destination (to), and name must be the same. The standby (and secondary) LSPs must have a pathName attribute set in order to differentiate them. The Primary LSP may have a pathName set.

Primary LSP:

```
{
```

```

    "name": "REST_LSP_2",
    "from": {"address": "62.0.0.102", "topoObjectType": "ipv4"},
    "to": {"address": "62.0.0.104", "topoObjectType": "ipv4"},
    "pathType": "primary",
    "plannedProperties": {
      "bandwidth": "400M",
      "setupPriority": 7,
      "holdingPriority": 7
    }
  }
}

```

Standby LSP:

```

{
  "name": "REST_LSP_2",
  "from": {"address": "62.0.0.102", "topoObjectType": "ipv4"},
  "to": {"address": "62.0.0.104", "topoObjectType": "ipv4"},
  "pathType": "standby",
  "plannedProperties": {
    "pathName": "standby_path_for_second_LSP",
    "bandwidth": "100M",
    "setupPriority": 7,
    "holdingPriority": 7
  }
}

```

- LSP with a loose explicit route. The loose hop must follow a hop; if no other hop can be supplied, usual best practice is use of the ingress node ID.

```

{
  "name": "REST_LPS_with_explicit_route",
  "from": {"topoObjectType": "ipv4", "address": "62.0.0.101"},
  "to": {"topoObjectType": "ipv4", "address": "62.0.0.103"},
  "plannedProperties": {
    "bandwidth": 100000000,
    "setupPriority": 7,
    "holdingPriority": 7,
    "ero": [
      {
        "topoObjectType": "ipv4", "address": "62.0.0.101", "loose": false
      },
      {
        "topoObjectType": "ipv4", "address": "62.0.0.102", "loose": true
      }
    ]
  }
}

```

- Creates a TE LSP and associates it with a CCC VPN via userProperties.

```

{
  "name": "lintalphaomega",
  "creationConfigurationMethod": "NETCONF",
  "provisioningType": "RSVP",
  "pathType": "primary",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.101"
  },
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.103"
  }
}

```



```

    },
    "plannedProperties": {
      "bandwidth": "24k",
      "setupPriority": 7,
      "holdingPriority": 7,
      "userProperties": {
        "ccc-vpn-name": "vpnsisisi",
        "ccc-interface": "ge-0/0/7.987",
        "transmit-lsp": "lintalphaomega",
        "receive-lsp": "lintomegaalpha"
      }
    }
  }
}

```

The response is :

```

{
  "plannedProperties": {
    "bandwidth": "24K",
    "setupPriority": 7,
    "holdingPriority": 7,
    "pathName": "lintalphaomega_p0",
    "adminStatus": "Up",
    "preferredEro": [
      {
        "topoObjectType": "ipv4",
        "address": "11.101.105.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.105.107.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.103.107.1",
        "loose": false
      }
    ],
    "userProperties": {
      "ccc-vpn-name": "vpnsisisi",
      "ccc-interface": "ge-0/0/7.987",
      "transmit-lsp": "lintalphaomega",
      "receive-lsp": "lintomegaalpha"
    },
    "lastStatusString": "[ConfigServer]<Netconf provisioning order received",
    "correlatedRROHopCount": 0
  },
  "name": "lintalphaomega",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.101"
  },
  "pathType": "primary",
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.103"
  }
}

```

```
    "lspIndex": 95,  
    "controlType": "PCC",  
    "provisioningType": "RSVP"  
  }  
}
```

In the following NETCONF SR LSP examples, the following should be noted:

- Only Juniper routers are supported.
- Only NETCONF provisioning is supported.
- The binding SID label, if specified, is checked by the device based on the configured label range.
- Only one primary LSP path is supported.

SR LSP examples:

- In this example, the SR LSP has a node SID specified for the destination.

```
{  
  "name": "restSRNodeSID",  
  "creationConfigurationMethod": "NETCONF",  
  "provisioningType": "SR",  
  "pathType": "primary",  
  "from": {  
    "topoObjectType": "ipv4",  
    "address": "11.0.0.101"  
  },  
  "to": {  
    "topoObjectType": "ipv4",  
    "address": "11.0.0.103"  
  },  
  "plannedProperties": {  
    "bandwidth": "0",  
    "setupPriority": 7,  
    "holdingPriority": 7,  
    "ero": [  
      {  
        "topoObjectType": "ipv4",  
        "address": "11.0.0.101",  
        "loose": false  
      },  
      {  
        "topoObjectType": "ipv4",  
        "address": "11.0.0.103",  
        "loose": true  
      }  
    ],  
    "design": {  
      "routingMethod": "routeByDevice",  
      "adminGroups": {}  
    }  
  }  
}
```

The response is:

```
{
```

```

    "plannedProperties": {
      "bandwidth": "0",
      "setupPriority": 7,
      "holdingPriority": 7,
      "adminStatus": "Up",
      "ero": [
        {
          "topoObjectType": "ipv4",
          "address": "11.0.0.101",
          "loose": false
        },
        {
          "topoObjectType": "ipv4",
          "address": "11.0.0.103",
          "loose": true
        }
      ],
      "design": {
        "routingMethod": "routeByDevice"
      }
    },
    "name": "restSRNodeSID",
    "from": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.101"
    },
    "pathType": "primary",
    "to": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.103"
    },
    "lspIndex": 106,
    "controlType": "PCC",
    "provisioningType": "SR"
  }

```

- In this example, the SR LSP has a list of link SIDs specified to reach the destination.

```

{
  "name": "restSRLinkSIDs",
  "creationConfigurationMethod": "NETCONF",
  "provisioningType": "SR",
  "pathType": "primary",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.101"
  },
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.104"
  },
  "plannedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 7,
    "ero": [
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.101",
        "loose": false
      }
    ]
  }
}

```

```
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.101.105.2",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.105.106.2",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.106.107.2",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.104.107.1",
      "loose": false
    }
  ],
  "design": {
    "adminGroups": {}
  }
}
```

The response is:

```
{
  "plannedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 7,
    "adminStatus": "Up",
    "ero": [
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.101",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.101.105.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.105.106.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.106.107.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.1",

```

```

        "loose": false
      }
    ]
  },
  "name": "restSRLinkSIDs",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.101"
  },
  "pathType": "primary",
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.104"
  },
  "lspIndex": 107,
  "controlType": "PCC",
  "provisioningType": "SR"
}

```

In Paragon PathFinder, a privateForwardingAdjacency is comprised of a pair of binding SID SR LSPs from node A to node Z and from node Z to node A.

- In this example, the Binding SID SR LSP named restBindingSIDSRnodeSID from vmx105 to vmx 107 has a node SID specified for the destination.

```

{
  "name": "restBindingSIDSRnodeSID",
  "creationConfigurationMethod": "NETCONF",
  "provisioningType": "SR",
  "pathType": "primary",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.105"
  },
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.107"
  },
  "plannedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 7,
    "bindingSID": 1048048,
    "ero": [
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.105",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.107",
        "loose": true
      }
    ],
    "design": {
      "routingMethod": "routeByDevice",
      "adminGroups": {}
    }
  }
}

```

```
}

```

The response is:

```
{
  "plannedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 7,
    "adminStatus": "Up",
    "ero": [
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.105",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.107",
        "loose": true
      }
    ],
    "design": {
      "routingMethod": "routeByDevice"
    }
  },
  "name": "restBindingSIDSRnodeSID",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.105"
  },
  "pathType": "primary",
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.107"
  },
  "lspIndex": 108,
  "controlType": "PCC",
  "provisioningType": "SR"
}
```

- In this example, the Binding SID SR LSP named restBindingSIDSRnodeSID from vmx107 to vmx 105 has a node SID specified for the destination.

```
{
  "name": "restBindingSIDSRnodeSID",
  "creationConfigurationMethod": "NETCONF",
  "provisioningType": "SR",
  "pathType": "primary",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.107"
  },
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.105"
  },
  "plannedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,

```

```
"holdingPriority": 7,
"bindingSID": 1048048,
"ero": [
  {
    "topoObjectType": "ipv4",
    "address": "11.0.0.107",
    "loose": false
  },
  {
    "topoObjectType": "ipv4",
    "address": "11.0.0.105",
    "loose": true
  }
],
"design": {
  "routingMethod": "routeByDevice",
  "adminGroups": {}
}
}
```

The response is:

```
{
  "plannedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 7,
    "adminStatus": "Up",
    "ero": [
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.107",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.105",
        "loose": true
      }
    ],
    "design": {
      "routingMethod": "routeByDevice"
    }
  },
  "name": "restBindingSIDSRnodeSID",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.107"
  },
  "pathType": "primary",
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.105"
  },
  "lspIndex": 109,
  "controlType": "PCC",
  "provisioningType": "SR"
}
```

- In this example, the Binding SID SR LSP named restBindingSIDSRLinkSIDs from vmx105 to vmx 107 has a list of SIDs specified to reach the destination.

```
{
  "name": "restBindingSIDSRLinkSIDs",
  "creationConfigurationMethod": "NETCONF",
  "provisioningType": "SR",
  "pathType": "primary",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.105"
  },
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.107"
  },
  "plannedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 7,
    "bindingSID": 1048049,
    "ero": [
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.105",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.102.105.1",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.102.106.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.104.106.1",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.2",
        "loose": false
      }
    ],
    "design": {
      "adminGroups": {
      }
    }
  }
}
```

The response is:

```
{
  "plannedProperties": {
```



```

    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 7,
    "adminStatus": "Up",
    "ero": [
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.105",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.102.105.1",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.102.106.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.104.106.1",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.2",
        "loose": false
      }
    ]
  },
  "name": "restBindingSIDSRlinkSIDs",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.105"
  },
  "pathType": "primary",
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.107"
  },
  "lspIndex": 110,
  "controlType": "PCC",
  "provisioningType": "SR"
}

```

- In this example, the Binding SID SR LSP named restBindingSIDSRlinkSIDs from vmx107 to vmx 105 has a list of SIDs specified to reach the destination.

```

{
  "name": "restBindingSIDSRlinkSIDs",
  "creationConfigurationMethod": "NETCONF",
  "provisioningType": "SR",
  "pathType": "primary",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.107"
  },
  "to": {

```

```
    "topoObjectType": "ipv4",
    "address": "11.0.0.105"
  },
  "plannedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 7,
    "bindingSID": 1048049,
    "ero": [
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.107",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.1",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.104.106.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.102.106.1",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.102.105.2",
        "loose": false
      }
    ],
    "design": {
      "adminGroups": {
      }
    }
  }
}
```

The response is:

```
{
  "plannedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 7,
    "adminStatus": "Up",
    "ero": [
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.107",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.1",
```

```

        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.104.106.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.102.106.1",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.102.105.2",
        "loose": false
      }
    ]
  },
  "name": "restBindingSIDSRLinkSIDs",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.107"
  },
  "pathType": "primary",
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.105"
  },
  "lspIndex": 111,
  "controlType": "PCC",
  "provisioningType": "SR"
}

```

- In this example, the SR LSP named restSRoverPFA1 from vmx101 to vmx 103 has a list of SIDs specified to reach the destination; along the path, a privateForwardingAdjacency is used.

```

{
  "name": "restSRoverPFA1",
  "creationConfigurationMethod": "NETCONF",
  "provisioningType": "SR",
  "pathType": "primary",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.101"
  },
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.103"
  },
  "plannedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 7,
    "ero": [
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.101",
        "loose": false
      }
    ]
  }
}

```

```

    },
    {
      "topoObjectType": "ipv4",
      "address": "11.101.105.2",
      "loose": false
    },
    {
      "topoObjectType": "hopId",
      "id": "binding:0110.0000.0105:restBindingSIDSRLinkSIDs",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.103.107.1",
      "loose": false
    }
  ],
  "design": {
    "adminGroups": {
      }
    }
  }
}

```

The response is:

```

{
  "plannedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 7,
    "adminStatus": "Up",
    "ero": [
      {
        "topoObjectType": "ipv4",
        "address": "11.0.0.101",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.101.105.2",
        "loose": false
      },
      {
        "topoObjectType": "hopId",
        "id": "binding:0110.0000.0105:restBindingSIDSRLinkSIDs",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.103.107.1",
        "loose": false
      }
    ]
  },
  "name": "restSRoverPFA1",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.101"
  }
}

```

```
    },
    "pathType": "primary",
    "to": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.103"
    },
    "lspIndex": 112,
    "controlType": "PCC",
    "provisioningType": "SR"
  }
}
```

3.5.2.2. Response

Example 3.36. Create a TE-LSP : JSON response

The following example shows the creation of an LSP with a bandwidth of 100M:

```
{
  "lspIndex": 20,
  "name": "Rest_LSP_1",
  "from": {"address": "62.0.0.101", "topoObjectType": "ipv4"},
  "to": {"address": "62.0.0.103", "topoObjectType": "ipv4"},
  "controlType": "PCEInitiated",
  "plannedProperties": {
    "adminStatus": "Up",
    "bandwidth": "100M",
    "setupPriority": 7,
    "holdingPriority": 7,
    "lastStatusString": "Provisioning Order from REST Interface",
    "routingStatus": "Unknown"
  },
  "operationalStatus": "Unknown",
  "pathType": "primary"
}
```

3.5.3. Search LSPs

Method	URI	Description
GET	/topology/v2/{topology_id}/te-lsps/search{?name,from,operationalStatus,queryType}	Performs a search in the LSP list based on the URI parameters. For example, "search?name=62.101.105" returns one link.

Normal response codes: 200

3.5.3.1. Request

This table shows the URI parameters for the search lsps request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This table shows the query parameters for the search lsps request:

Name	Type	Description
name	String (Optional)	Filters on specified LSP Name, which is treated as a regular expression. Simple strings like "LP_" are acceptable, as are JavaScript regular expressions such as "/104\$/".
from	String (Optional)	Filters on the LSP from/address property.
operationalStatus	String (Optional)	Filters based on the LSP operationalStatus property.
queryType	String (Optional)	The queryType parameter sets the logical operator to be used between the query parameters. By default, the queryType is AND. If the queryType is set to OR, an LSP is included in the results if any of its parameter values match the query.

3.5.3.2. Response

Example 3.37. Search LSPs: JSON response

```
[
  {
    "operationalStatus": "Active",
    "plannedProperties": {
      "bandwidth": "0",
      "setupPriority": 7,
      "holdingPriority": 0,
      "calculatedEro": [
        {
          "topoObjectType": "ipv4",
          "address": "11.105.107.2",
          "loose": false
        },
        {
          "topoObjectType": "ipv4",
          "address": "11.106.107.1",
          "loose": false
        }
      ]
    },
    "routingStatus": "Down",
```

```

        "adminStatus": "Up",
        "design": {
            "routingMethod": "routeByDevice"
        },
        "lastStatusString": "[PCServer]<Down controller_state=Path found
on down lsp",
        "controllerStatus": {
            "status": "Path found on down lsp"
        },
        "correlatedRR0HopCount": 2
    },
    "name": "rsvp-105-106",
    "from": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.105"
    },
    "pathType": "primary",
    "to": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.106"
    },
    "lspIndex": 13,
    "controlType": "PCC",
    "provisioningType": "RSVP",
    "collectedProperties": {
        "bandwidth": "0",
        "setupPriority": 7,
        "holdingPriority": 0,
        "calculatedEro": [
            {
                "topoObjectType": "ipv4",
                "address": "11.0.0.105",
                "loose": false
            },
            {
                "topoObjectType": "ipv4",
                "address": "11.105.106.2",
                "loose": false
            },
            {
                "topoObjectType": "ipv4",
                "address": "11.0.0.106",
                "loose": false
            }
        ]
    },
    "routingStatus": "Up",
    "adminStatus": "Up",
    "correlatedRR0HopCount": 1
},
"tunnelId": 60274,
"liveProperties": {
    "bandwidth": 0,
    "metric": 10,
    "setupPriority": 7,
    "holdingPriority": 0,
    "operationalStatus": "Active",
    "adminStatus": "Up",
    "ero": [],
    "rro": [
        {

```

```

        "topoObjectType": "ipv4",
        "address": "11.105.106.2",
        "protectionInUse": false,
        "protectionAvailable": false
    }
}
},
"controller": "External"
},
{
    "operationalStatus": "Active",
    "name": "11.0.0.101:11.0.0.104:300:vpls:vpn_200",
    "p2mpIndex": 184549480,
    "p2mpName": "11.0.0.104:300:vpls:vpn_200",
    "from": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.104"
    },
    "to": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.101"
    },
    "liveProperties": {
        "bandwidth": 0,
        "metric": 0,
        "setupPriority": 7,
        "holdingPriority": 0,
        "operationalStatus": "Active",
        "adminStatus": "Up",
        "ero": [],
        "rro": [
            {
                "topoObjectType": "ipv4",
                "address": "11.104.107.2",
                "protectionInUse": false,
                "protectionAvailable": false
            },
            {
                "topoObjectType": "ipv4",
                "address": "11.105.107.1",
                "protectionInUse": false,
                "protectionAvailable": false
            },
            {
                "topoObjectType": "ipv4",
                "address": "11.101.105.1",
                "protectionInUse": false,
                "protectionAvailable": false
            }
        ]
    },
    "controlType": "PCC",
    "controller": "External",
    "pathType": "primary",
    "lspIndex": 14
},
{
    "operationalStatus": "Active",
    "name": "11.0.0.103:11.0.0.104:300:vpls:vpn_200",
    "p2mpIndex": 184549480,

```



```
"p2mpName": "11.0.0.104:300:vpls:vpn_200",
"from": {
  "topoObjectType": "ipv4",
  "address": "11.0.0.104"
},
"to": {
  "topoObjectType": "ipv4",
  "address": "11.0.0.103"
},
"liveProperties": {
  "bandwidth": 0,
  "metric": 0,
  "setupPriority": 7,
  "holdingPriority": 0,
  "operationalStatus": "Active",
  "adminStatus": "Up",
  "ero": [],
  "rro": [
    {
      "topoObjectType": "ipv4",
      "address": "11.104.107.2",
      "protectionInUse": false,
      "protectionAvailable": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.103.107.1",
      "protectionInUse": false,
      "protectionAvailable": false
    }
  ]
},
"controlType": "PCC",
"controller": "External",
"pathType": "primary",
"lspIndex": 15
},
{
  "operationalStatus": "Active",
  "name": "11.0.0.106:11.0.0.104:300:vpls:vpn_200",
  "p2mpIndex": 184549480,
  "p2mpName": "11.0.0.104:300:vpls:vpn_200",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.104"
  },
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.106"
  },
  "liveProperties": {
    "bandwidth": 0,
    "metric": 0,
    "setupPriority": 7,
    "holdingPriority": 0,
    "operationalStatus": "Active",
    "adminStatus": "Up",
    "ero": [],
    "rro": [
      {
```

```

        "topoObjectType": "ipv4",
        "address": "11.104.107.2",
        "protectionInUse": false,
        "protectionAvailable": false
    },
    {
        "topoObjectType": "ipv4",
        "address": "11.105.107.1",
        "protectionInUse": false,
        "protectionAvailable": false
    },
    {
        "topoObjectType": "ipv4",
        "address": "11.105.106.2",
        "protectionInUse": false,
        "protectionAvailable": false
    }
]
},
"controlType": "PCC",
"controller": "External",
"pathType": "primary",
"lspIndex": 16
},
{
    "operationalStatus": "Active",
    "plannedProperties": {
        "bandwidth": "0",
        "setupPriority": 7,
        "holdingPriority": 0,
        "calculatedEro": [
            {
                "topoObjectType": "ipv4",
                "address": "11.104.107.2",
                "loose": false
            },
            {
                "topoObjectType": "ipv4",
                "address": "11.105.107.1",
                "loose": false
            },
            {
                "topoObjectType": "ipv4",
                "address": "11.101.105.1",
                "loose": false
            }
        ],
        "routingStatus": "Up",
        "adminStatus": "Up",
        "lastStatusString": "[PCServer]<PCC ACK request_id=417990091",
        "correlatedRR0HopCount": 3
    },
    "name": "Silver-104-101",
    "from": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.104"
    },
    "pathType": "primary",
    "to": {
        "topoObjectType": "ipv4",

```

```
        "address": "11.0.0.101"
      },
      "lspIndex": 17,
      "controlType": "Delegated",
      "provisioningType": "RSVP",
      "initiator": "PCC",
      "controller": "NorthStar",
      "collectedProperties": {
        "bandwidth": "0",
        "setupPriority": 7,
        "holdingPriority": 0,
        "calculatedEro": [
          {
            "topoObjectType": "ipv4",
            "address": "11.0.0.104",
            "loose": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.104.107.2",
            "loose": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.105.107.1",
            "loose": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.101.105.1",
            "loose": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.0.0.101",
            "loose": false
          }
        ]
      },
      "routingStatus": "Up",
      "adminStatus": "Up",
      "correlatedRR0HopCount": 3
    },
    "tunnelId": 32533,
    "liveProperties": {
      "bandwidth": 0,
      "metric": 30,
      "setupPriority": 7,
      "holdingPriority": 0,
      "operationalStatus": "Active",
      "adminStatus": "Up",
      "ero": [
        {
          "topoObjectType": "ipv4",
          "address": "11.104.107.2",
          "loose": false
        },
        {
          "topoObjectType": "ipv4",
          "address": "11.105.107.1",
          "loose": false
        }
      ]
    }
  ]
}
```

```

    },
    {
      "topoObjectType": "ipv4",
      "address": "11.101.105.1",
      "loose": false
    }
  ],
  "rrro": [
    {
      "topoObjectType": "ipv4",
      "address": "11.104.107.2",
      "protectionInUse": false,
      "protectionAvailable": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.105.107.1",
      "protectionInUse": false,
      "protectionAvailable": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.101.105.1",
      "protectionInUse": false,
      "protectionAvailable": false
    }
  ]
},
{
  "operationalStatus": "Active",
  "plannedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 0,
    "calculatedEro": [
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.105.107.1",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.102.105.1",
        "loose": false
      }
    ],
    "routingStatus": "Up",
    "adminStatus": "Up",
    "lastStatusString": "[PCServer]<PCC ACK request_id=417990082",
    "correlatedRR0HopCount": 3
  },
  "name": "Silver-104-102",
  "from": {
    "topoObjectType": "ipv4",

```

```
        "address": "11.0.0.104"
      },
      "pathType": "primary",
      "to": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.102"
      },
      "lspIndex": 18,
      "controlType": "Delegated",
      "provisioningType": "RSVP",
      "initiator": "PCC",
      "controller": "NorthStar",
      "collectedProperties": {
        "bandwidth": "0",
        "setupPriority": 7,
        "holdingPriority": 0,
        "calculatedEro": [
          {
            "topoObjectType": "ipv4",
            "address": "11.0.0.104",
            "loose": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.104.107.2",
            "loose": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.105.107.1",
            "loose": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.102.105.1",
            "loose": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.0.0.102",
            "loose": false
          }
        ],
        "routingStatus": "Up",
        "adminStatus": "Up",
        "correlatedRR0HopCount": 3
      },
      "tunnelId": 29910,
      "liveProperties": {
        "bandwidth": 0,
        "metric": 30,
        "setupPriority": 7,
        "holdingPriority": 0,
        "operationalStatus": "Active",
        "adminStatus": "Up",
        "ero": [
          {
            "topoObjectType": "ipv4",
            "address": "11.104.107.2",
            "loose": false
          }
        ]
      }
    }
  ]
}
```

```

    },
    {
      "topoObjectType": "ipv4",
      "address": "11.105.107.1",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.102.105.1",
      "loose": false
    }
  ],
  "rrro": [
    {
      "topoObjectType": "ipv4",
      "address": "11.104.107.2",
      "protectionInUse": false,
      "protectionAvailable": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.105.107.1",
      "protectionInUse": false,
      "protectionAvailable": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.102.105.1",
      "protectionInUse": false,
      "protectionAvailable": false
    }
  ]
},
{
  "operationalStatus": "Active",
  "plannedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 0,
    "calculatedEro": [
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.103.107.1",
        "loose": false
      }
    ],
    "routingStatus": "Up",
    "adminStatus": "Up",
    "lastStatusString": "[PCServer]<PCC ACK request_id=417990087",
    "correlatedRR0HopCount": 2
  },
  "name": "Silver-104-103",
  "from": {
    "topoObjectType": "ipv4",

```

```
        "address": "11.0.0.104"
      },
      "pathType": "primary",
      "to": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.103"
      },
      "lspIndex": 19,
      "controlType": "Delegated",
      "provisioningType": "RSVP",
      "initiator": "PCC",
      "controller": "NorthStar",
      "collectedProperties": {
        "bandwidth": "0",
        "setupPriority": 7,
        "holdingPriority": 0,
        "calculatedEro": [
          {
            "topoObjectType": "ipv4",
            "address": "11.0.0.104",
            "loose": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.104.107.2",
            "loose": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.103.107.1",
            "loose": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.0.0.103",
            "loose": false
          }
        ]
      },
      "routingStatus": "Up",
      "adminStatus": "Up",
      "correlatedRR0HopCount": 2
    },
    "tunnelId": 32451,
    "liveProperties": {
      "bandwidth": 0,
      "metric": 20,
      "setupPriority": 7,
      "holdingPriority": 0,
      "operationalStatus": "Active",
      "adminStatus": "Up",
      "ero": [
        {
          "topoObjectType": "ipv4",
          "address": "11.104.107.2",
          "loose": false
        },
        {
          "topoObjectType": "ipv4",
          "address": "11.103.107.1",
          "loose": false
        }
      ]
    }
  }
}
```

```

    }
    ],
    "rrro": [
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.2",
        "protectionInUse": false,
        "protectionAvailable": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.103.107.1",
        "protectionInUse": false,
        "protectionAvailable": false
      }
    ]
  }
},
{
  "operationalStatus": "Active",
  "plannedProperties": {
    "bandwidth": "0",
    "setupPriority": 7,
    "holdingPriority": 0,
    "calculatedEro": [
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.105.107.1",
        "loose": false
      }
    ],
    "routingStatus": "Down",
    "adminStatus": "Up",
    "design": {
      "routingMethod": "routeByDevice"
    },
    "lastStatusString": "[PCServer]<Down controller_state=Path found
on down lsp",
    "controllerStatus": {
      "status": "Path found on down lsp"
    },
    "correlatedRRROHopCount": 2
  },
  "name": "rsvp-104-105",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.104"
  },
  "pathType": "primary",
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.105"
  },
  "lspIndex": 20,
  "controlType": "PCC",

```



```
    "provisioningType": "RSVP",
    "collectedProperties": {
      "bandwidth": "0",
      "setupPriority": 7,
      "holdingPriority": 0,
      "calculatedEro": [
        {
          "topoObjectType": "ipv4",
          "address": "11.0.0.104",
          "loose": false
        },
        {
          "topoObjectType": "ipv4",
          "address": "11.104.107.2",
          "loose": false
        },
        {
          "topoObjectType": "ipv4",
          "address": "11.105.107.1",
          "loose": false
        },
        {
          "topoObjectType": "ipv4",
          "address": "11.0.0.105",
          "loose": false
        }
      ],
      "routingStatus": "Up",
      "adminStatus": "Up",
      "correlatedRR0HopCount": 2
    },
    "tunnelId": 32072,
    "liveProperties": {
      "bandwidth": 0,
      "metric": 20,
      "setupPriority": 7,
      "holdingPriority": 0,
      "operationalStatus": "Active",
      "adminStatus": "Up",
      "ero": [],
      "rro": [
        {
          "topoObjectType": "ipv4",
          "address": "11.104.107.2",
          "protectionInUse": false,
          "protectionAvailable": false
        },
        {
          "topoObjectType": "ipv4",
          "address": "11.105.107.1",
          "protectionInUse": false,
          "protectionAvailable": false
        }
      ]
    },
    "controller": "External"
  }
]
```

3.5.4. Start a SSE Stream

Method	URI	Description
GET	/topology/v2/{topology_id}/te-lsps/stream	See EventSource for format. The notifications send on that stream are only lspEvent or lspTopologyEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).

Normal response codes: 200

3.5.4.1. Request

This table shows the URI parameters for the start a sse stream request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.5.5. Get a Single TE-LSP

Method	URI	Description
GET	/topology/v2/{topology_id}/te-lsps/{lspIndex}	Returns the details for a TE-LSP.

Normal response codes: 200

3.5.5.1. Request

This table shows the URI parameters for the get a single te-lsp request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{lspIndex}	Int	The unique lspIndex.

This operation does not accept a request body.

3.5.5.2. Response

Example 3.38. Get a Single TE-LSP: JSON response

```
{
  "lspIndex": 2,
  "name": "LP_101_103",
  "from": {"address": "62.0.0.101", "topoObjectType": "ipv4"},
  "to": {"address": "62.0.0.103", "topoObjectType": "ipv4"},
  "controlType": "Delegated",
  "plannedProperties": {
    "bandwidth": "10M",
    "setupPriority": 7,
    "holdingPriority": 0,
    "calculatedEro": [
      {"topoObjectType": "ipv4", "address": "62.101.105.2", "loose": false},
      {"topoObjectType": "ipv4", "address": "62.102.105.1", "loose": false},
      {"topoObjectType": "ipv4", "address": "62.102.106.2", "loose": false},
      {"topoObjectType": "ipv4", "address": "62.104.106.1", "loose": false},
      {"topoObjectType": "ipv4", "address": "62.104.107.2", "loose": false},
      {"topoObjectType": "ipv4", "address": "62.103.107.1", "loose": false}
    ],
    "pathName": "Path_Node101_Node103_Strict_1",
    "adminStatus": "Up",
    "routingStatus": "Up",
    "lastStatusString": "<Active PCS initialization"
  },
  "liveProperties": {
    "adminStatus": "Up",
    "bandwidth": 10000000,
    "ero": [
      {"address": "62.101.105.2", "loose": false, "topoObjectType":
"ipv4" },
      {"address": "62.105.107.2", "loose": false, "topoObjectType":
"ipv4" },
```

```
        {"address": "62.103.107.1", "loose": false, "topoObjectType":  
"ipv4" }  
      ],  
      "holdingPriority": 0,  
      "metric": 40,  
      "pathName": "Path_Node101_Node103_Strict_1",  
      "rro": [  
        {"address": "62.0.0.105", "protectionAvailable": true,  
"protectionInUse": false, "topoObjectType": "ipv4"},  
        {"address": "62.101.105.2", "protectionAvailable": true,  
"protectionInUse": false, "topoObjectType": "ipv4"},  
        {"address": "62.0.0.107", "protectionAvailable": false,  
"protectionInUse": false, "topoObjectType": "ipv4"},  
        {"address": "62.105.107.2", "protectionAvailable": false,  
"protectionInUse": false, "topoObjectType": "ipv4"},  
        {"address": "62.0.0.103", "protectionAvailable": false,  
"protectionInUse": false, "topoObjectType": "ipv4"},  
        {"address": "62.103.107.1", "protectionAvailable": false,  
"protectionInUse": false, "topoObjectType": "ipv4"}  
      ],  
      "setupPriority": 7  
    },  
    "operationalStatus": "Active",  
    "pathType": "primary",  
    "tunnelId": 56614  
  }  
}
```

3.5.6. Update a TE-LSP

Method	URI	Description
PUT	/topology/v2/{topology_id}/te-lsp/{lspIndex}	Updates a TE-LSP using the JSON schema: lsp.json#/definitions/updateLSP .

Normal response codes: 202

3.5.6.1. Request

This table shows the URI parameters for the update a te-lsp request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{lspIndex}	Int	The unique lspIndex.

Example 3.39. Update a TE-LSP : JSON request

The update accepts the same parameters as the create, except for parameters that cannot be modified (from, to, name, pathName, pathType).

```
{
  "lspIndex": 31,
  "name": "Rest_LSP_1",
  "from": {"topoObjectType": "ipv4", "address": "62.0.0.101" },
  "to": {"topoObjectType": "ipv4", "address": "62.0.0.103" },
  "pathType": "primary",
  "plannedProperties": {
    "bandwidth": "15M",
    "setupPriority": 7,
    "holdingPriority": 7
  }
}
```

3.5.6.2. Response

Example 3.40. Update a TE-LSP : JSON response

```
{
  "lspIndex": 20,
  "name": "Rest_LSP_1",
  "from": {"address": "62.0.0.101", "topoObjectType": "ipv4"},
  "to": {"address": "62.0.0.103", "topoObjectType": "ipv4"},
  "controlType": "PCEInitiated",
  "plannedProperties": {
    "adminStatus": "Up",
    "bandwidth": "100M",
    "setupPriority": 7,
    "holdingPriority": 7,
    "lastStatusString": "Provisioning Order from REST Interface",
    "routingStatus": "Unknown"
  },
  "operationalStatus": "Unknown",
  "pathType": "primary"
}
```

```
}
```



3.5.7. Patch a TE-LSP

Method	URI	Description
PATCH	/topology/v2/{topology_id}/te-lsps/{lspIndex}	Updates a TE-LSP using a RFC6902 patch: json-patch.json . The result of the patch must conform to lsp.json#/definitions/updateLsp . The REST server remove all operational parameters like operationalStatus, ..etc. . Using an empty patch (empty list) will result in the LSP to be re-provisioned without parameter changed.

Normal response codes: 202

3.5.7.1. Request

This table shows the URI parameters for the patch a te-lsp request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{lspIndex}	Int	The unique lspIndex.

Example 3.41. Patch a TE-LSP: JSON request

The Patched update accepts the same parameters as the create, except for parameters that cannot be modified (from, to, name, pathName, pathType).

```
[{ "op": "replace", "path": "/plannedProperties/bandwidth", "value": "100M" }]
```

3.5.7.2. Response

Example 3.42. Patch a TE-LSP: JSON response

```
{
  "lspIndex": 20,
  "name": "Rest_LSP_1",
  "from": {"address": "62.0.0.101", "topoObjectType": "ipv4"},
  "to": {"address": "62.0.0.103", "topoObjectType": "ipv4"},
  "controlType": "PCEInitiated",
  "plannedProperties": {
    "adminStatus": "Up",
    "bandwidth": "100M",
    "setupPriority": 7,
    "holdingPriority": 7,
    "lastStatusString": "Provisioning Order from REST Interface",
    "routingStatus": "Unknown"
  },
  "operationalStatus": "Unknown",
  "pathType": "primary"
}
```

3.5.8. Delete a TE-LSP

Method	URI	Description
DELETE	/topology/v2/{topology_id}/te-lsps/{lspIndex}	Deletes a TE-LSP. This function is supported only on the PCE-initiated LSPs. PCC-controlled and PCC-delegated LSPs cannot be deleted from Paragon PathFinder. They must be deleted in the node.

Normal response codes: 204

3.5.8.1. Request

This table shows the URI parameters for the delete a te-lsp request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{lspIndex}	Int	The unique lspIndex.

This operation does not accept a request body.

3.5.9. Create a List of TE-LSPs

Method	URI	Description
POST	/topology/v2/{topology_id}/te-lsps/bulk	Creates several TE-LSPs using the following JSON schema: lsp.json#/definitions/createLSPList .

Normal response codes: 201

3.5.9.1. Request

This table shows the URI parameters for the create a list of te-lsps request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 3.43. Create a List of TE-LSPs: JSON request

The request must contain a list of LSPs to be created. The LSP parameters are the same as creating an individual LSP. The following example shows the creation of two diverse LSPs at the same time.

```
[
  {
    "name": "REST_LSP_DIVERSE_1",
    "from": {"address": "62.0.0.102", "topoObjectType": "ipv4"},
    "to": {"address": "62.0.0.104", "topoObjectType": "ipv4"},
    "plannedProperties": {
      "bandwidth": "100M",
      "setupPriority": 7,
      "holdingPriority": 7,
      "design": {"diversityLevel": "srlg",
"diversityGroup": "DiverseGroup1"}
    },
  },
  {
    "name": "REST_LSP_DIVERSE_2",
    "from": {"address": "62.0.0.103", "topoObjectType": "ipv4"},
    "to": {"address": "62.0.0.101", "topoObjectType": "ipv4"},
    "plannedProperties": {
      "bandwidth": "100M",
      "setupPriority": 7,
      "holdingPriority": 7,
      "design": {"diversityLevel": "srlg",
"diversityGroup": "DiverseGroup1"}
    },
  }
]
```

3.5.9.2. Response

Example 3.44. Create a List of TE-LSPs: JSON response

```
[
  {
```

```
    "name": "REST_LSP_DIVERSE_1",
    "from": {"address": "62.0.0.102", "topoObjectType": "ipv4"},
    "to": {"address": "62.0.0.104", "topoObjectType": "ipv4"},
    "lspIndex": 21,
    "plannedProperties": {
      "bandwidth": "100M",
      "setupPriority": 7,
      "holdingPriority": 7,
      "adminStatus": "Up",
      "routingStatus": "Unknown",
      "design" : {"diversityLevel":"srlg",
"diversityGroup":"DiverseGroup1"},
      "lastStatusString": ">Provisioning Order from REST Interface"
    },
    "controlType": "PCEInitiated",
    "operationalStatus": "Unknown"
  },
  {
    "name": "REST_LSP_DIVERSE_2",
    "from": {"address": "62.0.0.103", "topoObjectType": "ipv4"},
    "to": {"address": "62.0.0.101", "topoObjectType": "ipv4"},
    "lspIndex": 22,
    "plannedProperties": {
      "bandwidth": "100M",
      "setupPriority": 7,
      "holdingPriority": 7,
      "adminStatus": "Up",
      "routingStatus": "Unknown",
      "design" : {"diversityLevel":"srlg",
"diversityGroup":"DiverseGroup1"},
      "lastStatusString": ">Provisioning Order from REST Interface"
    },
    "controlType": "PCEInitiated",
    "operationalStatus": "Unknown"
  }
]
```

3.5.10. Update a List of TE-LSPs

Method	URI	Description
PUT	/topology/v2/{topology_id}/te-lsps/bulk	Updates several TE-LSPs using the following JSON schema: lsp.json#/definitions/lspListUpdate .

Normal response codes: 202

3.5.10.1. Request

This table shows the URI parameters for the update a list of te-lsps request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

The bulk update accepts a list of LSP updates. Each entry requires the same parameters and logic as updating a single LSP.

This operation does not accept a request body.

3.5.10.2. Response

The response contains a list of individual update responses (see TE-LSP update).

This operation does not return a response body.

3.5.11. Update a List of TE-LSPs using PATCH

Method	URI	Description
PATCH	/topology/v2/{topology_id}/te-lsps/bulk	Updates several TE-LSPs using the following JSON schema: lsp.json#/definitions/lspListPatch .

Normal response codes: 202

3.5.11.1. Request

This table shows the URI parameters for the update a list of te-lsps using patch request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 3.45. Update a List of TE-LSPs using PATCH : JSON request

The bulk PATCH accepts a list consisting of lspIndex and patch:

```
[
  {"lspIndex" : 1,"patch":[{"op": "replace", "path": "/plannedProperties/
bandwidth", "value": "1M" }]},
  {"lspIndex" : 2,"patch":[{"op": "replace", "path": "/plannedProperties/
bandwidth", "value": "2M" }]}
]
```

3.5.11.2. Response

The response contains a list of individual update responses (see TE-LSP update).

This operation does not return a response body.

3.5.12. Delete a List of TE-LSPs

Method	URI	Description
DELETE	/topology/v2/{topology_id}/te-lsp/bulk	Deletes a list of TE-LSPs. This function is supported only on the PCE-initiated LSPs. PCC-controlled and PCC-delegated LSPs cannot be deleted from Paragon PathFinder. They must be deleted in the node.b The payload must conform to lsp.json#/definitions/lspListDelete

Normal response codes: 204

3.5.12.1. Request

This table shows the URI parameters for the delete a list of te-lsp request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.5.13. Get the LSP Event History

Method	URI	Description
GET	/topology/v2/{topology_id}/te-lsps/{lspIndex}/history{?start,end}	Returns the history for a TE-LSP.

The history contains a list of Unix-timestamped events for the LSP resource.

Normal response codes: 200

3.5.13.1. Request

This table shows the URI parameters for the get the lsp event history request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{lspIndex}	Int	The unique lspIndex.

This table shows the query parameters for the get the lsp event history request:

Name	Type	Description
start	Int (Optional)	Start timestamp: Include events with the starting time and later.
end	Int (Optional)	End timestamp: Include events before (but not including) the ending timestamp.

3.5.13.2. Response

Example 3.46. Get the LSP Event History: JSON response

```
[
  {
    "plannedProperties": {
      "bandwidth": "0",
      "setupPriority": 7,
      "holdingPriority": 0,
      "adminStatus": "Up",
      "routingStatus": "Down"
    },
    "controlType": "PCC",
    "eventStatusString": "<Down",
    "timestamp": 1427128941053,
    "operation": "State Change"
  },
  {
    "plannedProperties": {
      "bandwidth": "0",
      "setupPriority": 7,
      "holdingPriority": 0,
      "adminStatus": "Up",
      "routingStatus": "Down"
    },
    "controlType": "PCC",
```

```
"eventStatusString": "<Down PCS initialization",
"timestamp": 1427128941057,
"operation": "State Change"
},
{
  "plannedProperties": {
    "bandwidth": "10M",
    "setupPriority": 7,
    "holdingPriority": 0,
    "pathName": "VMX103_VMX101",
    "adminStatus": "Up",
    "routingStatus": "Down"
  },
  "controlType": "Delegated",
  "eventStatusString": "reprovision:provisioning new delegated lsp",
  "timestamp": 1427132006714,
  "operation": "State Change"
},
{
  "plannedProperties": {
    "bandwidth": "10M",
    "setupPriority": 7,
    "holdingPriority": 0,
    "pathName": "Node103_Node101",
    "adminStatus": "Up",
    "routingStatus": "Down"
  },
  "controlType": "Delegated",
  "eventStatusString": "Down",
  "timestamp": 1427132006720,
  "operation": "State Change"
},
{
  "plannedProperties": {
    "bandwidth": "10M",
    "setupPriority": 7,
    "holdingPriority": 0,
    "pathName": "Node103_Node101",
    "adminStatus": "Up",
    "routingStatus": "Down"
  },
  "controlType": "Delegated",
  "eventStatusString": "Down",
  "timestamp": 1427132006780,
  "operation": "State Change"
},
{
  "plannedProperties": {
    "bandwidth": "10M",
    "setupPriority": 7,
    "holdingPriority": 0,
    "pathName": "Node103_Node101",
    "adminStatus": "Up",
    "routingStatus": "Up"
  },
  "controlType": "Delegated",
  "eventStatusString": "Up",
  "timestamp": 1427132007053,
  "operation": "State Change"
},
```

```
{
  "plannedProperties": {
    "bandwidth": "10M",
    "setupPriority": 7,
    "holdingPriority": 0,
    "pathName": "Node103_Node101",
    "adminStatus": "Up",
    "routingStatus": "Up"
  },
  "controlType": "Delegated",
  "eventStatusString": "Active",
  "timestamp": 1427132007069,
  "operation": "State Change"
},
{
  "plannedProperties": {
    "bandwidth": "10M",
    "setupPriority": 7,
    "holdingPriority": 0,
    "pathName": "Node103_Node101",
    "adminStatus": "Up",
    "routingStatus": "Up"
  },
  "controlType": "Delegated",
  "eventStatusString": "Active",
  "timestamp": 1427132009764,
  "operation": "State Change"
},
{
  "plannedProperties": {
    "bandwidth": "10M",
    "setupPriority": 7,
    "holdingPriority": 0,
    "pathName": "Node103_Node101",
    "adminStatus": "Up",
    "routingStatus": "Up"
  },
  "controlType": "Delegated",
  "eventStatusString": "Active",
  "timestamp": 1427135406437,
  "operation": "State Change"
},
{
  "plannedProperties": {
    "bandwidth": "10M",
    "setupPriority": 7,
    "holdingPriority": 0,
    "pathName": "Path_Node101_Node103_Strict_1",
    "adminStatus": "Up",
    "routingStatus": "Down"
  },
  "controlType": "Delegated",
  "eventStatusString": "reprovision:Provision using planned data",
  "timestamp": 1427167092366,
  "operation": "State Change"
},
{
  "plannedProperties": {
    "bandwidth": "10M",
    "setupPriority": 7,
```



```

        "holdingPriority": 0,
        "pathName": "Path_Node101_Node103_Strict_1",
        "adminStatus": "Up",
        "routingStatus": "Down"
    },
    "controlType": "Delegated",
    "eventStatusString": "Down, PCS initialization",
    "timestamp": 1427167092372,
    "operation": "State Change"
  },
  {
    "plannedProperties": {
      "bandwidth": "10M",
      "setupPriority": 7,
      "holdingPriority": 0,
      "pathName": "Path_Node101_Node103_Strict_1",
      "adminStatus": "Up",
      "routingStatus": "Down"
    },
    "controlType": "Delegated",
    "eventStatusString": "Down",
    "timestamp": 1427167092475,
    "operation": "State Change"
  },
  {
    "plannedProperties": {
      "bandwidth": "10M",
      "setupPriority": 7,
      "holdingPriority": 0,
      "pathName": "Path_Node101_Node103_Strict_1",
      "adminStatus": "Up",
      "routingStatus": "Down"
    },
    "controlType": "Delegated",
    "eventStatusString": "Down",
    "timestamp": 1427167092516,
    "operation": "State Change"
  },
  {
    "plannedProperties": {
      "bandwidth": "10M",
      "setupPriority": 7,
      "holdingPriority": 0,
      "pathName": "Path_Node101_Node103_Strict_1",
      "adminStatus": "Up",
      "routingStatus": "Up"
    },
    "controlType": "Delegated",
    "eventStatusString": "Active",
    "timestamp": 1427167092865,
    "operation": "State Change"
  }
]

```

3.6. Demands

Use the following endpoints to access demands and the related information. Demands can be created based on netflow or LDP collection tasks.

The demand resources are described in [demands.json](#) .

Method	URI	Description
GET	/topology/v2/{topology_id}/demands	Returns a full list of Demands.
POST	/topology/v2/{topology_id}/demands	Create a demand using the following JSON schema: demands.json#/definitions/createDemand .
GET	/topology/v2/{topology_id}/demands/stream	See EventSource for format. The notifications send on that stream are only demandEvent and demandTopologyEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).
GET	/topology/v2/{topology_id}/demands/{demandIndex}	Return the details of a specified demand.
PUT	/topology/v2/{topology_id}/demands/{demandIndex}	Modify a specific demand using the following JSON schema: demands.json#/definitions/updateDemand .
PATCH	/topology/v2/{topology_id}/demands/{demandIndex}	Patch a specific demand using a RFC6902 patch: json-patch.json . The result of the patch must conform to demands.json#/definitions/updateDemand . The REST server remove all operational parameters like operationalStatus, ..etc. .
DELETE	/topology/v2/{topology_id}/demands/{demandIndex}	Delete a specific demand.
POST	/topology/v2/{topology_id}/demands/bulk	Create a set of demands using the following JSON schema: demands.json#/definitions/createDemandList . The resulting list of demands (after the patch is applied) must also conform to demands.json#/definitions/demandListUpdate . The return code indicates the request acceptance.
PUT	/topology/v2/{topology_id}/demands/bulk	Update a set of demands using the following JSON schema: demands.json#/definitions/demandListUpdate . The return code indicates the request acceptance.
PATCH	/topology/v2/{topology_id}/demands/bulk	Patch a set of demands using the following JSON schema: demands.json#/definitions/demandListPatch . The return code indicates the request acceptance.
DELETE	/topology/v2/{topology_id}/demands/bulk	Delete a set of demands. The return code indicates the request acceptance.

3.6.1. Gets all Demands

Method	URI	Description
GET	/topology/v2/{topology_id}/demands	Returns a full list of Demands.

Normal response codes: 200

3.6.1.1. Request

This table shows the URI parameters for the gets all demands request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.6.1.2. Response

Example 3.47. Gets all Demands: JSON response

```
[
  {
    "operationalStatus": "Unknown",
    "plannedProperties": {
      "bandwidth": "1.684666M",
      "setupPriority": 7,
      "holdingPriority": 7,
      "calculatedEro": [
        {
          "topoObjectType": "ipv4",
          "address": "11.103.107.2",
          "loose": false
        },
        {
          "topoObjectType": "ipv4",
          "address": "11.105.107.1",
          "loose": false
        },
        {
          "topoObjectType": "ipv4",
          "address": "11.101.105.1",
          "loose": false
        }
      ],
      "routingStatus": "Up",
      "pathName": "11.0.0.101",
      "adminStatus": "Up",
      "lastStatusString": "[PCServer]>demand update",
      "controllerStatus": {
        "status": ""
      },
      "correlatedRROHopCount": 3
    },
    "name": "vmx103_11.0.0.101/32_NONE_IP",
    "from": {
      "topoObjectType": "ipv4",
```

```
        "address": "11.0.0.103"
      },
      "pathType": "primary",
      "to": {
        "topoObjectType": "prefix",
        "address": "11.0.0.101",
        "length": 32
      },
      "demandIndex": 30,
      "liveProperties": {
        "bandwidth": "1.684666M"
      }
    },
    {
      "operationalStatus": "Unknown",
      "plannedProperties": {
        "bandwidth": "1.613733M",
        "setupPriority": 7,
        "holdingPriority": 7,
        "calculatedEro": [
          {
            "topoObjectType": "ipv4",
            "address": "11.104.107.2",
            "loose": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.105.107.1",
            "loose": false
          },
          {
            "topoObjectType": "ipv4",
            "address": "11.102.105.1",
            "loose": false
          }
        ],
        "routingStatus": "Up",
        "pathName": "11.0.0.102",
        "adminStatus": "Up",
        "lastStatusString": "[PCServer]>demand update",
        "controllerStatus": {
          "status": ""
        },
        "correlatedRR0HopCount": 3
      },
      "name": "vmx104_11.0.0.102/32_NONE_IP",
      "from": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.104"
      },
      "pathType": "primary",
      "to": {
        "topoObjectType": "prefix",
        "address": "11.0.0.102",
        "length": 32
      },
      "demandIndex": 31,
      "liveProperties": {
        "bandwidth": "1.613733M"
      }
    }
  ]
}
```

```
}  
]
```

3.6.2. Create a Demand

Method	URI	Description
POST	/topology/v2/{topology_id}/demands	Create a demand using the following JSON schema: demands.json#/definitions/createDemand .

Normal response codes: 200

3.6.2.1. Request

This table shows the URI parameters for the create a demand request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 3.48. Create a Demand: JSON request

```
{
  "plannedProperties":{
    "pathName":"11.0.0.11",
    "bandwidth":0,
    "setupPriority":7,
    "holdingPriority":7,
    "design":{
      "routingMethod":"default",
      "adminGroups":{

      }
    }
  },
  "name":"testprefix",
  "pathType":"primary",
  "from":{
    "topoObjectType":"ipv4",
    "address":"11.0.0.10"
  },
  "to":{
    "topoObjectType":"ipv4",
    "address":"11.0.0.11"
  },
  "prefix":{
    "topoObjectType":"ipv4",
    "address":"10.4.10.0",
    "length":24
  }
}
```

3.6.2.2. Response

Example 3.49. Create a Demand: JSON response

```
{
  "plannedProperties": {
    "bandwidth": 0,
```

```
        "setupPriority": 7,  
        "holdingPriority": 7,  
        "pathName": "11.0.0.11",  
        "adminStatus": "Up"  
    },  
    "name": "testprefix",  
    "from": {  
        "topoObjectType": "ipv4",  
        "address": "11.0.0.10"  
    },  
    "pathType": "primary",  
    "to": {  
        "topoObjectType": "ipv4",  
        "address": "11.0.0.11"  
    },  
    "demandIndex": 17,  
    "prefix": {  
        "topoObjectType": "prefix",  
        "address": "10.4.10.0",  
        "length": 24  
    }  
}
```

3.6.3. Start a SSE Stream

Method	URI	Description
GET	/topology/v2/{topology_id}/demands/stream	See EventSource for format. The notifications send on that stream are only demandEvent and demandTopologyEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).

Normal response codes: 200

3.6.3.1. Request

This table shows the URI parameters for the start a sse stream request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.6.4. Get a Specific Demand

Method	URI	Description
GET	/topology/v2/{topology_id}/demands/{demandIndex}	Return the details of a specified demand.

Normal response codes: 200

3.6.4.1. Request

This table shows the URI parameters for the get a specific demand request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{demandIndex}	Int	The unique demandIndex.

This operation does not accept a request body.

3.6.4.2. Response

Example 3.50. Get a Specific Demand: JSON response

```
{
  "operationalStatus": "Unknown",
  "plannedProperties": {
    "bandwidth": 0,
    "setupPriority": 7,
    "holdingPriority": 7,
    "pathName": "11.0.0.11",
    "adminStatus": "Up",
    "calculatedEro": [
      {
        "topoObjectType": "ipv4",
        "address": "10.1.10.1",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "10.1.2.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "10.2.11.11",
        "loose": false
      }
    ],
    "routingStatus": "Up",
    "correlatedRROHopCount": 3,
    "lastStatusString": "[PCServer]<demand orders update"
  },
  "name": "testprefix",
  "from": {
    "topoObjectType": "ipv4",
```

```
    "address": "11.0.0.10"
  },
  "pathType": "primary",
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.11"
  },
  "demandIndex": 27,
  "prefix": {
    "topoObjectType": "prefix",
    "address": "10.4.10.0",
    "length": 24
  }
}
```

3.6.5. Update a Specific Demand

Method	URI	Description
PUT	/topology/v2/{topology_id}/demands/{demandIndex}	Modify a specific demand using the following JSON schema: demands.json#/definitions/updateDemand .

Normal response codes: 200

3.6.5.1. Request

This table shows the URI parameters for the update a specific demand request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{demandIndex}	Int	The unique demandIndex.

Example 3.51. Update a Specific Demand: JSON request

```
{
  "plannedProperties": {
    "bandwidth": 4727,
    "setupPriority": 7,
    "holdingPriority": 7,
    "calculatedEro": [
      {
        "topoObjectType": "ipv4",
        "address": "10.1.10.1",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "10.1.2.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "10.2.11.11",
        "loose": false
      }
    ],
    "bindingLSP": "sr-color-test",
    "routingStatus": "Up",
    "pathName": "11.0.0.11",
    "adminStatus": "Up",
    "lastStatusString": "[PCServer]>new demand",
    "correlatedRROHopCount": 3
  },
  "name": "PE1_10.4.0.0/24_IP",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.10"
  },
  "pathType": "primary",
  "to": {
    "topoObjectType": "ipv4",
```

```
        "address": "11.0.0.11"
      },
      "demandIndex": 47,
      "prefix": {
        "topoObjectType": "prefix",
        "address": "10.4.0.0",
        "length": 24
      }
    }
  }
```

3.6.5.2. Response

Example 3.52. Update a Specific Demand: JSON response

```
{
  "plannedProperties": {
    "bandwidth": 4727,
    "setupPriority": 7,
    "holdingPriority": 7,
    "pathName": "11.0.0.11",
    "adminStatus": "Up",
    "bindingLSP": "sr-color-test"
  },
  "name": "PE1_10.4.0.0/24_IP",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.10"
  },
  "pathType": "primary",
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.11"
  },
  "demandIndex": 47,
  "prefix": {
    "topoObjectType": "prefix",
    "address": "10.4.0.0",
    "length": 24
  }
}
```

3.6.6. Update a Specific Demand using PATCH

Method	URI	Description
PATCH	/topology/v2/{topology_id}/demands/{demandIndex}	Patch a specific demand using a RFC6902 patch: json-patch.json . The result of the patch must conform to demands.json#/definitions/updateDemand . The REST server remove all operational parameters like operationalStatus, ..etc. .

Normal response codes: 200

3.6.6.1. Request

This table shows the URI parameters for the update a specific demand using patch request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{demandIndex}	Int	The unique demandIndex.

Example 3.53. Update a Specific Demand using PATCH: JSON request

```
[{ "op": "add", "path": "/plannedProperties/bindingLSP", "value": "sr-color-test" }]
```

3.6.6.2. Response

Example 3.54. Update a Specific Demand using PATCH: JSON response

```
{
  "plannedProperties": {
    "bandwidth": 140019,
    "setupPriority": 7,
    "holdingPriority": 7,
    "pathName": "11.0.0.11",
    "adminStatus": "Up",
    "bindingLSP": "sr-color-test"
  },
  "name": "PE1_10.4.0.0/24_IP",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.10"
  },
  "pathType": "primary",
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.11"
  },
  "demandIndex": 50,
  "prefix": {
    "topoObjectType": "prefix",
    "address": "10.4.0.0",
    "length": 24
  }
}
```


3.6.7. Delete a Specific Demand

Method	URI	Description
DELETE	/topology/v2/{topology_id}/demands/{demandIndex}	Delete a specific demand.

Normal response codes: 204

3.6.7.1. Request

This table shows the URI parameters for the delete a specific demand request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{demandIndex}	Int	The unique demandIndex.

This operation does not accept a request body.

3.6.8. Create a set of Demands

Method	URI	Description
POST	/topology/v2/{topology_id}/demands/bulk	Create a set of demands using the following JSON schema: demands.json#/definitions/createDemandList . The resulting list of demands (after the patch is applied) must also conform to demands.json#/definitions/demandListUpdate . The return code indicates the request acceptance.

Normal response codes: 200

3.6.8.1. Request

This table shows the URI parameters for the create a set of demands request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon Pathfinder version 2, the unique identifier is set to 1.

Example 3.55. Create a set of Demands: JSON request

```
[
  {
    "plannedProperties":{
      "pathName":"11.0.0.11",
      "bandwidth":"0",
      "setupPriority":7,
      "holdingPriority":7,
      "design":{
        "routingMethod":"default",
        "adminGroups":{

        }
      }
    },
    "name":"testprefix",
    "pathType":"primary",
    "from":{
      "topoObjectType":"ipv4",
      "address":"11.0.0.10"
    },
    "to":{
      "topoObjectType":"ipv4",
      "address":"11.0.0.11"
    },
    "prefix":{
      "topoObjectType":"ipv4",
      "address":"10.4.10.0",
      "length":24
    }
  },
  {
    "plannedProperties":{
      "pathName":"11.0.0.11",
      "bandwidth":"0",
      "setupPriority":7,
```



```
        "holdingPriority":7,
        "design":{
            "routingMethod":"default",
            "adminGroups":{
                }
        },
        "name":"testprefix2",
        "pathType":"primary",
        "from":{
            "topoObjectType":"ipv4",
            "address":"11.0.0.10"
        },
        "to":{
            "topoObjectType":"ipv4",
            "address":"11.0.0.11"
        },
        "prefix":{
            "topoObjectType":"ipv4",
            "address":"10.4.11.0",
            "length":24
        }
    }
}
```

3.6.8.2. Response

Example 3.56. Create a set of Demands: JSON response

```
[
  {
    "plannedProperties": {
      "bandwidth": 0,
      "setupPriority": 7,
      "holdingPriority": 7,
      "pathName": "11.0.0.11",
      "adminStatus": "Up"
    },
    "name": "testprefix",
    "from": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.10"
    },
    "pathType": "primary",
    "to": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.11"
    },
    "demandIndex": 51,
    "prefix": {
      "topoObjectType": "prefix",
      "address": "10.4.10.0",
      "length": 24
    }
  },
  {
    "plannedProperties": {
      "bandwidth": 0,
      "setupPriority": 7,
```

```
        "holdingPriority": 7,  
        "pathName": "11.0.0.11",  
        "adminStatus": "Up"  
    },  
    "name": "testprefix2",  
    "from": {  
        "topoObjectType": "ipv4",  
        "address": "11.0.0.10"  
    },  
    "pathType": "primary",  
    "to": {  
        "topoObjectType": "ipv4",  
        "address": "11.0.0.11"  
    },  
    "demandIndex": 52,  
    "prefix": {  
        "topoObjectType": "prefix",  
        "address": "10.4.11.0",  
        "length": 24  
    }  
}  
]
```

3.6.9. Update a set of Demands

Method	URI	Description
PUT	/topology/v2/{topology_id}/demands/bulk	Update a set of demands using the following JSON schema: demands.json#/definitions/demandListUpdate . The return code indicates the request acceptance.

Normal response codes: 200

3.6.9.1. Request

This table shows the URI parameters for the update a set of demands request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 3.57. Update a set of Demands: JSON request

```
[
  {
    "plannedProperties": {
      "bandwidth": 0,
      "setupPriority": 7,
      "holdingPriority": 7,
      "pathName": "11.0.0.11",
      "adminStatus": "Up"
    },
    "name": "testprefix",
    "from": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.10"
    },
    "pathType": "primary",
    "to": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.11"
    },
    "demandIndex": 51,
    "prefix": {
      "topoObjectType": "prefix",
      "address": "10.4.20.0",
      "length": 24
    }
  },
  {
    "plannedProperties": {
      "bandwidth": 0,
      "setupPriority": 7,
      "holdingPriority": 7,
      "pathName": "11.0.0.11",
      "adminStatus": "Up"
    },
    "name": "testprefix2",
    "from": {
      "topoObjectType": "ipv4",
```

```
        "address": "11.0.0.10"
      },
      "pathType": "primary",
      "to": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.11"
      },
      "demandIndex": 52,
      "prefix": {
        "topoObjectType": "prefix",
        "address": "10.4.21.0",
        "length": 24
      }
    }
  ]
```

3.6.9.2. Response

Example 3.58. Update a set of Demands: JSON response

```
[
  {
    "plannedProperties": {
      "bandwidth": 0,
      "setupPriority": 7,
      "holdingPriority": 7,
      "pathName": "11.0.0.11",
      "adminStatus": "Up"
    },
    "name": "testprefix",
    "from": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.10"
    },
    "pathType": "primary",
    "to": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.11"
    },
    "demandIndex": 51,
    "prefix": {
      "topoObjectType": "prefix",
      "address": "10.4.20.0",
      "length": 24
    }
  },
  {
    "plannedProperties": {
      "bandwidth": 0,
      "setupPriority": 7,
      "holdingPriority": 7,
      "pathName": "11.0.0.11",
      "adminStatus": "Up"
    },
    "name": "testprefix2",
    "from": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.10"
    },
    "pathType": "primary",
```

```
    "to": {  
      "topoObjectType": "ipv4",  
      "address": "11.0.0.11"  
    },  
    "demandIndex": 52,  
    "prefix": {  
      "topoObjectType": "prefix",  
      "address": "10.4.21.0",  
      "length": 24  
    }  
  }  
]
```

3.6.10. Patch a set of Demands

Method	URI	Description
PATCH	/topology/v2/{topology_id}/demands/bulk	Patch a set of demands using the following JSON schema: demands.json#/definitions/demandListPatch . The return code indicates the request acceptance.

Normal response codes: 200

3.6.10.1. Request

This table shows the URI parameters for the patch a set of demands request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 3.59. Patch a set of Demands: JSON request

```
[
  {
    "demandIndex" : 51,
    "patch": [
      {
        "op": "replace",
        "path": "/prefix/address",
        "value": "10.4.30.0"
      }
    ]
  },
  {
    "demandIndex" : 52,
    "patch": [
      {
        "op": "replace",
        "path": "/prefix/address",
        "value": "10.4.31.0"
      }
    ]
  }
]
```

3.6.10.2. Response

Example 3.60. Patch a set of Demands: JSON response

```
[
  {
    "plannedProperties": {
      "bandwidth": 0,
      "setupPriority": 7,
      "holdingPriority": 7,
      "pathName": "11.0.0.11",
      "adminStatus": "Up"
    },
    "name": "testprefix",
    "from": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.10"
    },
    "pathType": "primary",
    "to": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.11"
    },
    "demandIndex": 51,
    "prefix": {
      "topoObjectType": "prefix",
      "address": "10.4.30.0",
      "length": 24
    }
  }
]
```

```
    },  
    {  
      "plannedProperties": {  
        "bandwidth": 0,  
        "setupPriority": 7,  
        "holdingPriority": 7,  
        "pathName": "11.0.0.11",  
        "adminStatus": "Up"  
      },  
      "name": "testprefix2",  
      "from": {  
        "topoObjectType": "ipv4",  
        "address": "11.0.0.10"  
      },  
      "pathType": "primary",  
      "to": {  
        "topoObjectType": "ipv4",  
        "address": "11.0.0.11"  
      },  
      "demandIndex": 52,  
      "prefix": {  
        "topoObjectType": "prefix",  
        "address": "10.4.31.0",  
        "length": 24  
      }  
    }  
  ]  
}
```

3.6.11. Delete a set of Demands

Method	URI	Description
DELETE	/topology/v2/{topology_id}/demands/bulk	Delete a set of demands. The return code indicates the request acceptance.

Normal response codes: 204

3.6.11.1. Request

This table shows the URI parameters for the delete a set of demands request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 3.61. Delete a set of Demands: JSON request

```
[  
  {"demandIndex": 32},  
  {"demandIndex": 31}  
]
```

3.7. Task scheduler

Use these endpoints to work with task scheduler.

The scheduler resources are described in [scheduler.json](#).

The scheduler also provides a [socket.io](#) interface on which taskStatusUpdate event updates like task add/remove/update/status-change are sent. The socket.io namespace used is /taskScheduler and event name taskStatusUpdate.

Method	URI	Description
GET	/scheduler/v2/tasklist	Returns a full list of Tasks as per JSON schema: scheduler.json#/definitions/taskList
GET	/scheduler/v2/tasks	Returns a list of Tasks by filter task-type(?tasktype={tasktypeFilter}) as per JSON schema: scheduler.json#/definitions/taskList
POST	/scheduler/v2/updatetask	Update a task using the following JSON schema: scheduler.json#/definitions/updateTask . Returns response as per JSON schema: scheduler.json#/definitions/responseObject .
POST	/scheduler/v2/removeaddtask	Removes all the existing tasks with given taskType and adds a new Task using the following JSON schema: scheduler.json#/definitions/updateTask . Returns response as per JSON schema: scheduler.json#/definitions/responseObject .
POST	/scheduler/v2/deletetask	Delete one or more tasks using the following JSON schema: scheduler.json#/definitions/deleteTask . Returns response as per JSON schema: scheduler.json#/definitions/responseObject .

Method	URI	Description
GET	/scheduler/v2/taskshistory/{taskId}	Returns task execution status history in HTML table format as per JSON schema: scheduler.json#/definitions/tasksHistoryList
GET	/scheduler/v2/taskstatus/{taskId}	Returns task execution status history as per JSON schema: scheduler.json#/definitions/tasksStatusList

3.7.1. Gets all Tasks

Method	URI	Description
GET	/scheduler/v2/tasklist	Returns a full list of Tasks as per JSON schema: scheduler.json#/definitions/taskList

Normal response codes: 200

3.7.1.1. Request

This operation does not accept a request body.

3.7.1.2. Response

Example 3.62. Gets all Tasks: JSON response

```
[
  {
    "taskID": "7660df1e-b8fb-4f71-987e-a3d261c4d78f",
    "taskType": "Device Collection",
    "taskName": "dc",
    "createTime": "2019-04-26T09:45:54.745",
    "interval": 1,
    "scheduleType": 0,
    "startTime": "2019-04-26T09:45:54.745",
    "stopTime": null,
    "lastExecutionTime": "2019-04-26T09:45:57.755",
    "lastExecutionEndTime": "2019-04-26T09:46:30.762",
    "lastExecutionStatus": "Completed",
    "chainAfter": null,
    "taskparam": {
      "groupName": "liveNetwork",
      "idList": [],
      "jobId": 1556271955188,
      "opts": {
        "useMgmt": true,
        "concurrentJob": 16,
        "runParsing": true,
        "archiveRawData": true
      },
      "waitingList": [],
      "deviceGroups": [],
      "collection_commands": "config|interface|tunnel_path|transit_tunnel",
      "process_commands": "config,interface,tunnel_path,transit_tunnel,switch_cli,equipment_cli"
    },
    "lastExecutionLog": [
      "IP Address,Hostname,Status,Job Type,Severity",
      "11.0.0.101,vmx101-shiva,OK,config|interface|tunnel_path|transit_tunnel,INFO",
      "11.0.0.104,vmx104-shiva,OK,config|interface|tunnel_path|transit_tunnel,INFO",
      "11.0.0.103,vmx103-shiva,OK,config|interface|tunnel_path|transit_tunnel,INFO",
      "11.0.0.102,vmx102-shiva,OK,config|interface|tunnel_path|transit_tunnel,INFO",
    ]
  }
]
```

```
    ",All Devices,COMPLETE,Collection (Dir: /opt/northstar/data/
collection/7660df1e-b8fb-4f71-987e-a3d261c4d78f/1556271957755),INFO",
    ",All Devices,COMPLETE,Processing,INFO"
  ],
  "chainTaskGroup": null,
  "taskSize": 1
},
{
  "taskID": "fad3d27f-0e20-45b7-bbf1-c027509d75e0",
  "taskType": "Network Archive",
  "taskName": "NA4",
  "createTime": "2019-06-13T06:03:09.795",
  "interval": 0,
  "scheduleType": 0,
  "startTime": "2019-06-13T06:03:09.795",
  "stopTime": null,
  "lastExecutionTime": "2019-06-13T06:03:12.827",
  "lastExecutionEndTime": "2019-06-13T06:05:56.574",
  "lastExecutionStatus": "Completed",
  "chainAfter": null,
  "taskparam": {
    "opts": [
      "-l"
    ]
  },
  "lastExecutionLog": [
    "Details,Severity",
    "Parsed config files,INFO",
    "Parsed tunnel path and added to the spec file,INFO",
    "Added interface and tunnel traffic to the spec file,INFO",
    "Saved PCS tunnel to optunnel.x and added to the spec file,INFO"
  ],
  "chainTaskGroup": null,
  "taskSize": 1
}
]
```

3.7.2. Get tasks based on filter tasktype

Method	URI	Description
GET	/scheduler/v2/tasks	Returns a list of Tasks by filter task-type(?tasktype={tasktypeFilter}) as per JSON schema: scheduler.json#/definitions/taskList

Normal response codes: 200

3.7.2.1. Request

This operation does not accept a request body.

3.7.2.2. Response

Example 3.63. Get tasks based on filter tasktype: JSON response

```
[
  {
    "taskId": "7660df1e-b8fb-4f71-987e-a3d261c4d78f",
    "taskType": "Device Collection",
    "taskName": "dc",
    "createTime": "2019-04-26T09:45:54.745",
    "interval": 1,
    "scheduleType": 0,
    "startTime": "2019-04-26T09:45:54.745",
    "stopTime": null,
    "lastExecutionTime": "2019-04-26T09:45:57.755",
    "lastExecutionEndTime": "2019-04-26T09:46:30.762",
    "lastExecutionStatus": "Completed",
    "chainAfter": null,
    "taskparam": {
      "groupName": "liveNetwork",
      "idList": [],
      "jobId": 1556271955188,
      "opts": {
        "useMgmt": true,
        "concurrentJob": 16,
        "runParsing": true,
        "archiveRawData": true
      },
      "waitingList": [],
      "deviceGroups": [],
      "collection_commands": "config|interface|tunnel_path|transit_tunnel",
      "process_commands": "config,interface,tunnel_path,transit_tunnel,switch_cli,equipment_cli"
    },
    "lastExecutionLog": [
      "IP Address,Hostname,Status,Job Type,Severity",
      "11.0.0.101,vmx101-shiva,OK,config|interface|tunnel_path|transit_tunnel,INFO",
      "11.0.0.104,vmx104-shiva,OK,config|interface|tunnel_path|transit_tunnel,INFO",
      "11.0.0.103,vmx103-shiva,OK,config|interface|tunnel_path|transit_tunnel,INFO"
    ]
  }
]
```

```

        "11.0.0.102,vmx102-shiva,OK,config|interface|tunnel_path|
transit_tunnel,INFO",
        ",All Devices,COMPLETE,Collection (Dir: /opt/northstar/data/
collection/7660df1e-b8fb-4f71-987e-a3d261c4d78f/1556271957755),INFO",
        ",All Devices,COMPLETE,Processing,INFO"
    ],
    "chainTaskGroup": null,
    "taskSize": 1
},
{
    "taskId": "c51aea24-1c13-426a-89c9-fbb99ecc87de",
    "taskType": "Device Collection",
    "taskName": "future",
    "createTime": "2019-06-11T12:18:52.112",
    "interval": 0,
    "scheduleType": 1,
    "startTime": "2019-06-28T12:18:00.00",
    "stopTime": null,
    "lastExecutionTime": null,
    "lastExecutionEndTime": null,
    "lastExecutionStatus": "Scheduled",
    "chainAfter": null,
    "taskparam": {
        "groupName": "liveNetwork",
        "idList": [],
        "jobId": 1560255525511,
        "opts": {
            "useMgmt": true,
            "concurrentJob": 16,
            "runParsing": true,
            "archiveRawData": true
        },
        "waitingList": [],
        "deviceGroups": [],
        "collection_commands": "config|interface|tunnel_path|
transit_tunnel",
        "process_commands": "config,interface,tunnel_path,transit_tunnel,
switch_cli,equipment_cli"
    },
    "lastExecutionLog": [],
    "chainTaskGroup": null,
    "taskSize": 1
},
{
    "taskId": "70c71c06-73b4-4ea6-b4c8-48b831bb1781",
    "taskType": "Device Collection",
    "taskName": "dc1",
    "createTime": "2019-06-10T08:44:04.929",
    "interval": 0,
    "scheduleType": 0,
    "startTime": "2019-06-10T08:44:04.929",
    "stopTime": null,
    "lastExecutionTime": "2019-06-10T08:44:07.953",
    "lastExecutionEndTime": "2019-06-10T08:44:46.30",
    "lastExecutionStatus": "Completed",
    "chainAfter": null,
    "taskparam": {
        "groupName": "liveNetwork",
        "idList": [],
        "jobId": 1560156243979,

```

```

        "opts": {
            "useMgmt": true,
            "concurrentJob": 16,
            "runParsing": true,
            "archiveRawData": true
        },
        "waitingList": [],
        "deviceGroups": [],
        "collection_commands": "config|interface|tunnel_path|
transit_tunnel",
        "process_commands": "config,interface,tunnel_path,transit_tunnel,
switch_cli,equipment_cli"
    },
    "lastExecutionLog": [
        "Count,IP Address,Hostname,Status,Job Type,Severity",
        "1,11.0.0.101,vmx101-shiva,OK,config|interface|tunnel_path|
transit_tunnel,INFO",
        "2,11.0.0.104,vmx104-shiva,OK,config|interface|tunnel_path|
transit_tunnel,INFO",
        "3,11.0.0.103,vmx103-shiva,ACCESS_FAIL,config|interface|
tunnel_path|transit_tunnel,CRITIC",
        "4,11.0.0.102,vmx102-shiva,OK,config|interface|tunnel_path|
transit_tunnel,INFO",
        "-,-,All Devices,COMPLETE,Collection (Dir: /opt/northstar/data/
collection/70c71c06-73b4-4ea6-b4c8-48b831bb1781/1560156247953),INFO",
        "-,-,All Devices,COMPLETE,Processing,INFO",
        "Task result:,Devices attempted- 4 | success- 3 | access_fail- 1,,
",
    ],
    "chainTaskGroup": null,
    "taskSize": 4
},
{
    "taskId": "00e61ebd-adb2-4d97-a18f-b072f6f43504",
    "taskType": "Device Collection",
    "taskName": "First collection",
    "createTime": "2019-06-14T05:24:45.745",
    "interval": 1,
    "scheduleType": 4,
    "startTime": "2019-06-14T05:23:00.00",
    "stopTime": null,
    "lastExecutionTime": "2019-06-14T05:24:45.759",
    "lastExecutionEndTime": "2019-06-14T05:25:32.754",
    "lastExecutionStatus": "Scheduled",
    "chainAfter": null,
    "taskparam": {
        "groupName": "liveNetwork",
        "idList": [],
        "jobId": 1560489792572,
        "opts": {
            "useMgmt": true,
            "concurrentJob": 16,
            "runParsing": true,
            "archiveRawData": true
        },
        "waitingList": [],
        "deviceGroups": [],
        "collection_commands": "config|interface|tunnel_path|
transit_tunnel|switch_cli|equipment_cli",

```

```

        "process_commands": "config,interface,tunnel_path,transit_tunnel,
switch_cli,equipment_cli"
    },
    "lastExecutionLog": [
        "Count,IP Address,Hostname,Status,Job Type,Severity",
        "1,11.0.0.101,vmx101-shiva,OK,config|interface|tunnel_path|
transit_tunnel|switch_cli|equipment_cli,INFO",
        "2,11.0.0.104,vmx104-shiva,OK,config|interface|tunnel_path|
transit_tunnel|switch_cli|equipment_cli,INFO",
        "3,11.0.0.103,vmx103-shiva,ACCESS_FAIL,config|interface|
tunnel_path|transit_tunnel|switch_cli|equipment_cli,CRITIC",
        "4,11.0.0.102,vmx102-shiva,OK,config|interface|tunnel_path|
transit_tunnel|switch_cli|equipment_cli,INFO",
        "-,-,All Devices,COMPLETE,Collection (Dir: /opt/northstar/data/
collection/00e61ebd-adb2-4d97-a18f-b072f6f43504/1560489885759),INFO",
        "-,-,All Devices,COMPLETE,Processing,INFO",
        "Task result:,Devices attempted- 4 | success- 3 | access_fail- 1,,
",
    ],
    "chainTaskGroup": null,
    "taskSize": 4
},
{
    "taskID": "862b6dfb-4494-4230-ab4b-f40849eefc10",
    "taskType": "Device Collection",
    "taskName": "dc-1day",
    "createTime": "2019-02-05T09:09:38.237",
    "interval": 1,
    "scheduleType": 4,
    "startTime": "2019-02-05T09:09:38.237",
    "stopTime": null,
    "lastExecutionTime": "2019-06-13T09:09:38.278",
    "lastExecutionEndTime": "2019-06-13T09:10:13.691",
    "lastExecutionStatus": "Scheduled",
    "chainAfter": null,
    "taskparam": {
        "groupName": "liveNetwork",
        "idList": [],
        "jobId": 1549357765460,
        "opts": {
            "useMgmt": true,
            "concurrentJob": 16,
            "runParsing": true,
            "archiveRawData": true
        },
        "waitingList": [],
        "deviceGroups": [],
        "collection_commands": "config|interface|tunnel_path|
transit_tunnel",
        "process_commands": "config,interface,tunnel_path,transit_tunnel,
switch_cli,equipment_cli"
    },
    "lastExecutionLog": [
        "Count,IP Address,Hostname,Status,Job Type,Severity",
        "1,11.0.0.101,vmx101-shiva,OK,config|interface|tunnel_path|
transit_tunnel,INFO",
        "2,11.0.0.104,vmx104-shiva,OK,config|interface|tunnel_path|
transit_tunnel,INFO",
        "3,11.0.0.103,vmx103-shiva,ACCESS_FAIL,config|interface|
tunnel_path|transit_tunnel,CRITIC",

```

```
        "4,11.0.0.102,vmx102-shiva,OK,config|interface|tunnel_path|
transit_tunnel,INFO",
        "-,-,All Devices,COMPLETE,Collection (Dir: /opt/northstar/data/
collection/862b6dfb-4494-4230-ab4b-f40849eefc10/1560416978278),INFO",
        "-,-,All Devices,COMPLETE,Processing,INFO",
        "Task result:,Devices attempted- 4 | success- 3 | access_fail- 1,,
",
    ],
    "chainTaskGroup": "Device Collection_dc-1day",
    "taskSize": 4
}
]
```


3.7.3. Update a task

Method	URI	Description
POST	/scheduler/v2/updatetask	Update a task using the following JSON schema: scheduler.json#/definitions/updateTask . Returns response as per JSON schema: scheduler.json#/definitions/responseObject .

Normal response codes: 200

3.7.3.1. Request

Example 3.64. Update a task: JSON request

```
{
  "taskId": "00e61ebd-adb2-4d97-a18f-b072f6f43504",
  "taskType": "Device Collection",
  "taskName": "First collection",
  "interval": 1,
  "scheduleType": 4,
  "chainAfter": null,
  "chainTaskGroup": "",
  "taskparam": {
    "groupName": "liveNetwork",
    "idList": [],
    "opts": {
      "useMgmt": true,
      "concurrentJob": 16,
      "runParsing": true,
      "archiveRawData": true
    },
    "waitingList": [],
    "deviceGroups": [],
    "collection_commands": "config|interface|tunnel_path|transit_tunnel|
switch_cli|equipment_cli",
    "process_commands": "config,interface,tunnel_path,transit_tunnel,
switch_cli,equipment_cli"
  },
  "startTime": "2019-06-14T05:23:00.000Z"
}
```

3.7.3.2. Response

Example 3.65. Update a task: JSON response

```
{
  "taskId": "00e61ebd-adb2-4d97-a18f-b072f6f43504",
  "success": true
}
```

3.7.4. Remove and add a new Task

Method	URI	Description
POST	/scheduler/v2/removeaddtask	Removes all the existing tasks with given taskType and adds a new Task using the following JSON schema: scheduler.json#/definitions/updateTask . Returns response as per JSON schema: scheduler.json#/definitions/responseObject .

Normal response codes: 200

3.7.4.1. Request

Example 3.66. Remove and add a new Task: JSON request

```
{
  "taskId": "00e61ebd-adb2-4d97-a18f-b072f6f43504",
  "taskType": "Device Collection",
  "taskName": "First collection",
  "interval": 1,
  "scheduleType": 4,
  "chainAfter": null,
  "chainTaskGroup": "",
  "taskparam": {
    "groupName": "liveNetwork",
    "idList": [],
    "opts": {
      "useMgmt": true,
      "concurrentJob": 16,
      "runParsing": true,
      "archiveRawData": true
    },
    "waitingList": [],
    "deviceGroups": [],
    "collection_commands": "config|interface|tunnel_path|transit_tunnel|
switch_cli|equipment_cli",
    "process_commands": "config,interface,tunnel_path,transit_tunnel,
switch_cli,equipment_cli"
  },
  "startTime": "2019-06-14T05:23:00.000Z"
}
```

3.7.4.2. Response

Example 3.67. Remove and add a new Task: JSON response

```
{
  "taskId": "00e61ebd-adb2-4d97-a18f-b072f6f43504",
  "success": true
}
```

3.7.5. Delete tasks

Method	URI	Description
POST	/scheduler/v2/deletetask	Delete one or more tasks using the following JSON schema: scheduler.json#/definitions/deleteTask . Returns response as per JSON schema: scheduler.json#/definitions/responseObject .

Normal response codes: 200

3.7.5.1. Request

Example 3.68. Delete tasks: JSON request

```
[ "00e61ebd-adb2-4d97-a18f-b072f6f43504" ]
```

3.7.5.2. Response

Example 3.69. Delete tasks: JSON response

```
{  
  "success": true  
}
```

3.7.6. Gets task execution status history in HTML table format for a given taskId

Method	URI	Description
GET	/scheduler/v2/taskshistory/{taskId}	Returns task execution status history in HTML table format as per JSON schema: scheduler.json#/definitions/tasksHistoryList

Normal response codes: 200

3.7.6.1. Request

This operation does not accept a request body.

3.7.6.2. Response

Example 3.70. Gets task execution status history in HTML table format for a given taskId: JSON response

```
[
{
  "header": "1) 2019-06-14T07:03:18.040Z to 2019-06-14T07:03:42.616Z",
  "details": "
    <html>
      <body>
        <table cellpadding=0 cellspacing=0 border=1 width=100%>
          <tr bgcolor=#cccccc>
            <td nowrap>
              <b>IP Address</b>
            </td>
            <td nowrap>
              <b>Hostname</b>
            </td>
            <td nowrap>
              <b>Description</b>
            </td>
          </tr>
          <tr>
            <td nowrap>172.25.158.69</td>
            <td nowrap>vmx101-shiva-p107</td>
            <td nowrap>Collected 1 LSPs</td>
          </tr>
          <tr>
            <td nowrap>172.25.158.69</td>
            <td nowrap>vmx101-shiva</td>
            <td nowrap>Collected 9 LSPs</td>
          </tr>
          <tr>
            <td nowrap>172.25.158.69</td>
            <td nowrap>vmx101-shiva-p105</td>
            <td nowrap>Collected 2 LSPs</td>
          </tr>
          <tr>
            <td nowrap>172.25.158.69</td>
            <td nowrap>vmx101-shiva-p106</td>
            <td nowrap>Collected 1 LSPs</td>
          </tr>
        </table>
      </body>
    </html>
  "
}
```

```

        </tr>\n
        <tr>
            <td nowrap>172.25.158.70</td>
            <td nowrap>vmx104-shiva</td>
            <td nowrap>Collected 27 LSPs</td>
        </tr>\n
        <tr>
            <td nowrap>172.25.158.72</td>
            <td nowrap>vmx102-shiva</td>
            <td nowrap>Collected 55 LSPs</td>
        </tr>\n
        <tr bgcolor=#ff3333>
            <td nowrap>172.25.158.71</td>
            <td nowrap>vmx103-shiva</td>
            <td nowrap>No SNMP response received before timeout</td>
        </tr>\n
    </table>
</body>
</html>"}},
{
    "header": "2) 2019-06-14T06:58:18.040Z to 2019-06-14T06:58:42.722Z",
    "details": "
    <html>
        <body>
            <table cellpadding=0 cellspacing=0 width=\"100%\" border=1>\n
                <tr bgcolor=#cccccc>
                    <td nowrap>
                        <b>IP Address</b>
                    </td>
                    <td nowrap>
                        <b>Hostname</b>
                    </td>
                    <td nowrap>
                        <b>Description</b>
                    </td>
                </tr>\n
                <tr>
                    <td nowrap>172.25.158.69</td>
                    <td nowrap>vmx101-shiva-p107</td>
                    <td nowrap>Collected 1 LSPs</td>
                </tr>\n
                <tr>
                    <td nowrap>172.25.158.69</td>
                    <td nowrap>vmx101-shiva</td>
                    <td nowrap>Collected 9 LSPs</td>
                </tr>\n
                <tr>
                    <td nowrap>172.25.158.69</td>
                    <td nowrap>vmx101-shiva-p105</td>
                    <td nowrap>Collected 2 LSPs</td>
                </tr>\n
                <tr>
                    <td nowrap>172.25.158.69</td>
                    <td nowrap>vmx101-shiva-p106</td>
                    <td nowrap>Collected 1 LSPs</td>
                </tr>\n
                <tr>
                    <td nowrap>172.25.158.70</td>
                    <td nowrap>vmx104-shiva</td>

```

```

        <td nowrap>Collected 27 LSPs</td>
    </tr>\n
    <tr>
        <td nowrap>172.25.158.72</td>
        <td nowrap>vmx102-shiva</td>
        <td nowrap>Collected 55 LSPs</td>
    </tr>\n
    <tr bgcolor=#ff3333>
        <td nowrap>172.25.158.71</td>
        <td nowrap>vmx103-shiva</td>
        <td nowrap>No SNMP response received before timeout</td>
    </tr>\n
</table>
</body>
</html>"}},
{
  "header": "3) 2019-06-14T06:53:18.037Z to 2019-06-14T06:53:45.973Z",
  "details": "
    <html>
      <body>
        <table cellpadding=0 cellspacing=0 width=\"100%\" border=1>\n
          <tr bgcolor=#cccccc>
            <td nowrap>
              <b>IP Address</b>
            </td>
            <td nowrap>
              <b>Hostname</b>
            </td>
            <td nowrap>
              <b>Description</b>
            </td>
          </tr>\n
          <tr>
            <td nowrap>172.25.158.69</td>
            <td nowrap>vmx101-shiva-p107</td>
            <td nowrap>Collected 1 LSPs</td>
          </tr>\n
          <tr>
            <td nowrap>172.25.158.69</td>
            <td nowrap>vmx101-shiva</td>
            <td nowrap>Collected 9 LSPs</td>
          </tr>\n
          <tr>
            <td nowrap>172.25.158.69</td>
            <td nowrap>vmx101-shiva-p106</td>
            <td nowrap>Collected 1 LSPs</td>
          </tr>\n
          <tr>
            <td nowrap>172.25.158.69</td>
            <td nowrap>vmx101-shiva-p105</td>
            <td nowrap>Collected 2 LSPs</td>
          </tr>\n
          <tr>
            <td nowrap>172.25.158.70</td>
            <td nowrap>vmx104-shiva</td>
            <td nowrap>Collected 27 LSPs</td>
          </tr>\n
          <tr>
            <td nowrap>172.25.158.72</td>

```

```
        <td nowrap>vmx102-shiva</td>
        <td nowrap>Collected 55 LSPs</td>
    </tr>\n
    <tr bgcolor=#ff3333>
        <td nowrap>172.25.158.71</td>
        <td nowrap>vmx103-shiva</td>
        <td nowrap>No SNMP response received before timeout</td>
    </tr>\n
</table>
</body>
</html>"
}
]
```

3.7.7. Gets task execution status history for a given taskId

Method	URI	Description
GET	/scheduler/v2/taskstatus/{taskId}	Returns task execution status history as per JSON schema: scheduler.json#/definitions/tasksStatusList

Normal response codes: 200

3.7.7.1. Request

This operation does not accept a request body.

3.7.7.2. Response

Example 3.71. Gets task execution status history for a given taskId: JSON response

```
[
  {
    "taskId": "1767fcb4-59a0-48f8-8cb7-ad2612ceb606",
    "executionbeginTime": "2019-06-14 07:18:18.044Z",
    "executionendTime": "2019-06-14 07:18:41.763Z",
    "executionlog": [
      "IP Address,Hostname,Description,Severity",
      "172.25.158.69,vmx101-shiva-p107,Collected 1 LSPs,INFO",
      "172.25.158.69,vmx101-shiva,Collected 9 LSPs,INFO",
      "172.25.158.69,vmx101-shiva-p105,Collected 2 LSPs,INFO",
      "172.25.158.69,vmx101-shiva-p106,Collected 1 LSPs,INFO",
      "172.25.158.70,vmx104-shiva,Collected 27 LSPs,INFO",
      "172.25.158.72,vmx102-shiva,Collected 55 LSPs,INFO",
      "172.25.158.71,vmx103-shiva,No SNMP response received before
timeout,ERROR"
    ],
  },
  {
    "taskId": "1767fcb4-59a0-48f8-8cb7-ad2612ceb606",
    "executionbeginTime": "2019-06-14 07:13:18.042Z",
    "executionendTime": "2019-06-14 07:13:42.494Z",
    "executionlog": [
      "IP Address,Hostname,Description,Severity",
      "172.25.158.69,vmx101-shiva-p107,Collected 1 LSPs,INFO",
      "172.25.158.69,vmx101-shiva,Collected 9 LSPs,INFO",
      "172.25.158.69,vmx101-shiva-p106,Collected 1 LSPs,INFO",
      "172.25.158.69,vmx101-shiva-p105,Collected 2 LSPs,INFO",
      "172.25.158.70,vmx104-shiva,Collected 27 LSPs,INFO",
      "172.25.158.72,vmx102-shiva,Collected 55 LSPs,INFO",
      "172.25.158.71,vmx103-shiva,No SNMP response received before
timeout,ERROR"
    ],
  },
  {
    "taskId": "1767fcb4-59a0-48f8-8cb7-ad2612ceb606",
    "executionbeginTime": "2019-06-14 07:08:18.040Z",
    "executionendTime": "2019-06-14 07:08:44.478Z",
    "executionlog": [
```



```

        "IP Address,Hostname,Description,Severity",
        "172.25.158.69,vmx101-shiva-p107,Collected 1 LSPs,INFO",
        "172.25.158.69,vmx101-shiva,Collected 9 LSPs,INFO",
        "172.25.158.69,vmx101-shiva-p106,Collected 1 LSPs,INFO",
        "172.25.158.69,vmx101-shiva-p105,Collected 2 LSPs,INFO",
        "172.25.158.70,vmx104-shiva,Collected 27 LSPs,INFO",
        "172.25.158.72,vmx102-shiva,Collected 55 LSPs,INFO",
        "172.25.158.71,vmx103-shiva,No SNMP response received before
timeout,ERROR"
    ]
  },
  {
    "taskid": "1767fcb4-59a0-48f8-8cb7-ad2612ceb606",
    "executionbeginntime": "2019-06-14 07:03:18.040Z",
    "executionendtime": "2019-06-14 07:03:42.616Z",
    "executionlog": [
      "IP Address,Hostname,Description,Severity",
      "172.25.158.69,vmx101-shiva-p107,Collected 1 LSPs,INFO",
      "172.25.158.69,vmx101-shiva,Collected 9 LSPs,INFO",
      "172.25.158.69,vmx101-shiva-p105,Collected 2 LSPs,INFO",
      "172.25.158.69,vmx101-shiva-p106,Collected 1 LSPs,INFO",
      "172.25.158.70,vmx104-shiva,Collected 27 LSPs,INFO",
      "172.25.158.72,vmx102-shiva,Collected 55 LSPs,INFO",
      "172.25.158.71,vmx103-shiva,No SNMP response received before
timeout,ERROR"
    ]
  },
  {
    "taskid": "1767fcb4-59a0-48f8-8cb7-ad2612ceb606",
    "executionbeginntime": "2019-06-14 06:58:18.040Z",
    "executionendtime": "2019-06-14 06:58:42.722Z",
    "executionlog": [
      "IP Address,Hostname,Description,Severity",
      "172.25.158.69,vmx101-shiva-p107,Collected 1 LSPs,INFO",
      "172.25.158.69,vmx101-shiva,Collected 9 LSPs,INFO",
      "172.25.158.69,vmx101-shiva-p105,Collected 2 LSPs,INFO",
      "172.25.158.69,vmx101-shiva-p106,Collected 1 LSPs,INFO",
      "172.25.158.70,vmx104-shiva,Collected 27 LSPs,INFO",
      "172.25.158.72,vmx102-shiva,Collected 55 LSPs,INFO",
      "172.25.158.71,vmx103-shiva,No SNMP response received before
timeout,ERROR"
    ]
  }
]

```

3.8. Configuration

Endpoints for fetching/updating/deleting Northstar component configurations.

The configuration resources are described in [configuration.json](#).

The config also provide interface to update northstar configuration.

Method	URI	Description
GET	/config/config	Return a json object containing component configurations configuration.json#/definitions/northstar
POST	/config/config	Updates all component configurations using the follow- ing JSON schema: configuration.json#/definitions/north-

Method	URI	Description
		star-post . Returns response as per JSON schema: configuration.json#/definitions/responseObject .
PATCH	/config/config	Updates configuration similar to plain-patch mechanism described in RFC7396. The content of the request body will be merged into Paragon PathFinder configuration. The limitation of this method is that it is not supporting deletion of attributes, for that the DELETE method should be used. Two known deviations from RFC are: arrays types are merged instead of getting replaced and to unset the empty types DELETE operation has to be used. config.json#/definitions/northstar-post . Returns response as per JSON schema: config.json#/definitions/responseObject .
DELETE	/config/config	Delete component configurations using the following JSON schema: config.json#/definitions/northstar . Returns response as per JSON schema: config.json#/definitions/responseObject .

3.8.1. Get Paragon Pathfinder component configurations

Method	URI	Description
GET	/config/config	Return a json object containing component configurations configuration.json#/definitions/northstar

Normal response codes: 200

3.8.1.1. Request

This operation does not accept a request body.

3.8.1.2. Response

Example 3.72. Get Paragon Pathfinder component configurations: JSON response

```
{
  "analytics": {
    "data-publisher": {
      "max-queue-size": 3000000,
      "timeout": "30s"
    },
    "log-collector": {
      "heap-size": "2g",
      "jti-port": 3000,
      "jvision-ifd-port": 2000,
      "jvision-ifl-port": 2001,
      "jvision-lsp-port": 2002,
      "rpm-statistics-port": 1514
    },
    "model-driven-telemetry": {
      "log-destination": [
        {
          "name": "pipeline",
          "level": "info",
          "filename": "/opt/northstar/logs/pipeline.log"
        }
      ]
    },
    "netflowd": {
      "aggregate-by-prefix": {
        "disable": [
          null
        ]
      },
      "default-sampling-interval": 1,
      "enable-ssl": "",
      "log-destination": [
        {
          "name": "netflowd",
          "level": "info",
          "filename": "/opt/northstar/logs/netflowd.log"
        }
      ]
    },
    "publish-interval": "60s",
    "workers": 10
  }
}
```

```
    },
    "rollup-query-cutoff-interval": "7d"
  },
  "config-server": {
    "include-interface-type": [
      "links-if",
      "loopback-mgmt",
      "physical",
      "vrf-if"
    ]
  },
  "mladapter": {
    "polling-interval": 3600
  },
  "netconfd": {
    "device-connection-pool": {
      "batch-size": 10,
      "disconnect-delay": "300s",
      "netconf": {
        "keepalive-interval": "0s",
        "port": 830
      },
      "registry": {
        "check-delay": "45s",
        "publish-delay": "30s"
      },
      "publish-connected-device-status-delay": "300s",
      "reconnect-delay": "30s"
    },
    "encoding-template-folder": "/opt/northstar/netconfd/encoding_templates",
    "in-memory-datastore": {
      "connection-pool-size": 5
    },
    "log-destination": [
      {
        "name": "netconfd",
        "level": "info",
        "filename": "/opt/northstar/logs/netconfd.log"
      }
    ],
    "persistent-datastore": {
      "connection-pool-size": 5
    },
    "template-folder": "/opt/northstar/netconfd/templates"
  },
  "path-computation-server": {
    "link-flap": {
      "flap-count": 20,
      "flap-interval": "120s"
    }
  },
  "peer-engineering": {
    "egress": {
      "application": {
        "debug": false,
        "exec-pace-rate": "0.5"
      },
      "log-destination": [
        {
```

```
        "name": "log",
        "filename": "/opt/northstar/logs/epePlanner_info.log",
        "level": "info"
      }
    ],
    "northstar-rest-client": {
      "allow-rest-insecure": true,
      "host": "localhost",
      "port": 8443,
      "protocol": "https"
    },
    "notification": {
      "filename": "notif.json",
      "notification-types": [
        "amqp"
      ]
    }
  },
  "ingress": {
    "log-destination": [
      {
        "name": "ipe_app",
        "level": "info",
        "filename": "/opt/northstar/logs/ipe.log"
      }
    ]
  }
},
"programmable-rpd-client": {
  "retry-interval": "30s"
},
"system": {
  "health-monitor": {
    "heartbeat-holdown-timer": "15s",
    "heartbeat-interval": "5s",
    "history-ttl": "2d",
    "poll-interval": "10m"
  },
  "in-memory-datastore": {
    "connection-pool-size": 5,
    "reconnect-delay": 1000,
    "reconnect-retries": 10000
  },
  "messaging-bus": {
    "max-channels": 128,
    "reconnect-delay": 1000,
    "reconnect-retries": -1
  },
  "scheduler": {
    "distributed-task-queue": {
      "collector-max-queue-size": 0,
      "thread-pool-size": 10
    },
    "tasks": {
      "collection-cleanup": {
        "interval": "1d",
        "raw-data-retention-duration": "14d",
        "rollup-data-retention-duration": "180d"
      },
      "demand-reports": {
```

```

        "as-demands-bucket-size": 100
      },
      "device-collection": {
        "data-path": "/opt/northstar/data",
        "purge-cutoff": "7d",
        "timeout": "1800s"
      },
      "file-transfer": {
        "interval": "3600s"
      },
      "rollup": {
        "bulk-insert-record-count": 5000,
        "interval": "1h",
        "max-worker-process": 4
      },
      "scheduled-lsp-sizing": {
        "bandwidth-sizing": {
          "log-destination": [
            {
              "name": "bws_log",
              "filename": "bandwidth_sizing.log",
              "level": "info"
            }
          ],
          "stats-query-workers": 1
        },
        "container-lsp-normalization": {
          "log-destination": [
            {
              "name": "clsp_log",
              "filename": "container_lsp.log",
              "level": "info"
            }
          ]
        },
        "stats-collection-interval": "1m"
      },
      "snmp-collection": {
        "retries": 3,
        "timeout": "3s"
      }
    },
    "topology-server": {
      "application": {
        "element-state-mask": 23,
        "element-type-mask": 63
      },
      "application-tuning": {
        "lsp-topo-sync-timeout": "120s",
        "message-queue-high-watermark": 10000,
        "message-queue-low-watermark": 5000,
        "message-queue-overflow-watermark": 0,
        "pce-beacon-holddown": "90s",
        "pce-high-watermark-count-threshold": 10,
        "pce-high-watermark-rate": "-1.0",
        "pce-high-watermark-timeout": 60000,
        "pce-low-watermark-count-threshold": 3,
        "pce-low-watermark-rate": "-1.0",

```

```
        "pce-restart-holddown": "30s",
        "pce-watermark-interval": "10s",
        "resync-unresolved-node-threshold": 10
    },
    "data-persistence": {
        "debug-file-store-path": "/u/wandl/tmp/debug",
        "file-store-path": "/opt/northstar/data",
        "network-snapshot-store-interval": "3600s",
        "network-snapshot-store-path": "/opt/northstar/data/
network_archive/NorthStar",
        "persistent-storage-init-wait": "45s",
        "do-not-persist-lsps": ""
    },
    "log-destination": [
        {
            "name": "topology-server",
            "filename": "/opt/northstar/logs/toposerver.log",
            "level": "info"
        }
    ],
    "topology-acquisition": {
        "eor-timeout": "20s",
        "refresh-holddown": "300s",
        "retry-before-exit-count": 0,
        "retry-delay": "5s"
    }
}
```

3.8.2. Updates all component configurations

Method	URI	Description
POST	/config/config	Updates all component configurations using the following JSON schema: configuration.json#/definitions/northstar-post . Returns response as per JSON schema: configuration.json#/definitions/responseObject .

3.8.2.1. Request

Example 3.73. Updates all component configurations: JSON request

```
{
  "analytics": {
    "data-publisher": {
      "max-queue-size": 3000000,
      "timeout": "30s"
    },
    "log-collector": {
      "heap-size": "2g",
      "jti-port": 3000,
      "jvision-ifd-port": 2000,
      "jvision-ifl-port": 2001,
      "jvision-lsp-port": 2002,
      "rpm-statistics-port": 1514
    },
    "netflowd": {
      "aggregate-by-prefix": {
        "disable": [
          null
        ]
      },
      "default-sampling-interval": 1,
      "enable-ssl": "",
      "publish-interval": "60s",
      "workers": 10
    },
    "rollup-query-cutoff-interval": "7d"
  },
  "config-server": {
    "include-interface-type": [
      "physical",
      "loopback-mgmt",
      "vrf-if",
      "links-if"
    ]
  },
  "mladapter": {
    "polling-interval": 3600
  },
  "netconfd": {
    "device-connection-pool": {
      "batch-size": 10,
      "disconnect-delay": "300s",
      "netconf": {
        "keepalive-interval": "0s",
        "port": 830
      }
    }
  }
}
```



```
    },
    "publish-connected-device-status-delay": "300s",
    "reconnect-delay": "30s"
  },
  "encoding-template-folder": "/opt/northstar/netconfd/
encoding_templates",
  "in-memory-datastore": {
    "connection-pool-size": 5
  },
  "persistent-datastore": {
    "connection-pool-size": 5
  },
  "template-folder": "/opt/northstar/netconfd/templates"
},
"peer-engineering": {
  "egress": {
    "application": {
      "debug": false,
      "exec-pace-rate": "0.5"
    },
    "northstar-rest-client": {
      "allow-rest-insecure": true,
      "host": "localhost",
      "port": 8443,
      "protocol": "https"
    },
    "notification": {
      "filename": "notif.json",
      "notification-types": [
        "amqp"
      ]
    }
  },
  "programmable-rpd-client": {
    "retry-interval": "30s"
  },
  "system": {
    "health-monitor": {
      "heartbeat-holdown-timer": "15s",
      "heartbeat-interval": "5s",
      "history-ttl": "2d",
      "poll-interval": "10m"
    },
    "in-memory-datastore": {
      "connection-pool-size": 5,
      "reconnect-delay": 1000,
      "reconnect-retries": 10000
    },
    "messaging-bus": {
      "max-channels": 128,
      "reconnect-delay": 1000,
      "reconnect-retries": -1
    },
    "scheduler": {
      "distributed-task-queue": {
        "collector-max-queue-size": 0,
        "thread-pool-size": 10
      }
    }
  }
}
```

```
    "tasks": {
      "collection-cleanup": {
        "interval": "1d",
        "raw-data-retention-duration": "14d",
        "rollup-data-retention-duration": "180d"
      },
      "demand-reports": {
        "as-demands-bucket-size": 100
      },
      "device-collection": {
        "data-path": "/opt/northstar/data",
        "purge-cutoff": "7d",
        "timeout": "1800s"
      },
      "file-transfer": {
        "interval": "3600s"
      },
      "rollup": {
        "bulk-insert-record-count": 5000,
        "interval": "1h",
        "max-worker-process": 4
      },
      "scheduled-lsp-sizing": {
        "bandwidth-sizing": {
          "log-destination": [
            {
              "name": "bws_log",
              "filename": "bandwidth_sizing.log",
              "level": "info"
            }
          ]
        },
        "stats-query-workers": 1
      },
      "container-lsp-normalization": {
        "log-destination": [
          {
            "name": "clsp_log",
            "filename": "container_lsp.log",
            "level": "info"
          }
        ]
      },
      "stats-collection-interval": "1m"
    },
    "snmp-collection": {
      "retries": 3,
      "timeout": "3s"
    }
  },
  "topology-server": {
    "application": {
      "element-state-mask": 23,
      "element-type-mask": 63
    },
    "application-tuning": {
      "lsp-topo-sync-timeout": "120s",
      "message-queue-high-watermark": 10000,
      "message-queue-low-watermark": 5000,
    }
  }
}
```

```
    "message-queue-overflow-watermark": 0,  
    "pce-beacon-holddown": "90s",  
    "pce-high-watermark-count-threshold": 10,  
    "pce-high-watermark-rate": "-1.0",  
    "pce-high-watermark-timeout": 60000,  
    "pce-low-watermark-count-threshold": 3,  
    "pce-low-watermark-rate": "-1.0",  
    "pce-restart-holddown": "30s",  
    "pce-watermark-interval": "10s",  
    "resync-unresolved-node-threshold": 10  
  },  
  "data-persistence": {  
    "debug-file-store-path": "/u/wandl/tmp/debug",  
    "file-store-path": "/opt/northstar/data",  
    "network-snapshot-store-interval": "3600s",  
    "network-snapshot-store-path": "/opt/northstar/data/  
network_archive/NorthStar",  
    "persistent-storage-init-wait": "45s",  
    "persist-demand-events": ""  
  },  
  "topology-acquisition": {  
    "eor-timeout": "20s",  
    "refresh-holddown": "300s",  
    "retry-before-exit-count": 0,  
    "retry-delay": "5s"  
  }  
},  
"path-computation-server": {  
  "link-flap": {  
    "flap-count": 20,  
    "flap-interval": "120s"  
  }  
}  
}
```

3.8.3. Patch some component configurations

Method	URI	Description
PATCH	/config/config	Updates configuration similar to plain-patch mechanism described in RFC7396. The content of the request body will be merged into Paragon PathFinder configuration. The limitation of this method is that it is not supporting deletion of attributes, for that the DELETE method should be used. Two known deviations from RFC are: arrays types are merged instead of getting replaced and to unset the empty types DELETE operation has to be used. config.json#/definitions/northstar-post . Returns response as per JSON schema: config.json#/definitions/responseObject .

3.8.3.1. Request

Example 3.74. Patch some component configurations: JSON request

```
{
  "analytics": {
    "netflowd": {
      "enable-ssl": "",
      "aggregate-by-prefix": {
        "always": [ null ]
      },
      "default-sampling-interval": 1,
      "log-destination": [
        {
          "name": "netflowd-debug",
          "filename": "netflowd-debug2.log",
          "level": "debug"
        }
      ],
      "workers": 10
    }
  },
  "netconfd": {
    "device-connection-pool": {
      "batch-size": 100
    },
    "log-destination": [
      {
        "name": "netconfd2",
        "filename": "/opt/northstar/logs/netconfd-debug.log",
        "level": "debug"
      }
    ]
  },
  "path-computation-server": {
    "link-flap": {
      "flap-count": 10,
      "flap-interval": "100s"
    }
  },
  "topology-server": {
    "application": {
      "element-state-mask": 38,
```

```
        "element-type-mask": 60
      },
      "data-persistence": {
        "persist-demand-events": ""
      }
    }
  }
```

3.8.4. DELETE configuration

Method	URI	Description
DELETE	/config/config	Delete component configurations using the following JSON schema: config.json#/definitions/northstar . Returns response as per JSON schema: config.json#/definitions/responseObject .

3.8.4.1. Request

Example 3.75. DELETE configuration: JSON request

```
{
  "analytics": {
    "model-driven-telemetry": {
      "log-destination": [
        {
          "name": "pipeline2",
          "filename": "pipeline-debug.log",
          "level": "debug"
        }
      ]
    },
    "netflowd": {
      "enable-ssl": "",
      "aggregate-by-prefix": {
        "always": [
          null
        ]
      },
      "log-destination": [
        {
          "name": "netflowd3",
          "filename": "netflowd-debug.log",
          "level": "debug"
        }
      ]
    }
  },
  "topology-server": {
    "application": {
      "element-state-mask": 38,
      "element-type-mask": 60
    }
  }
}
```

3.9. Device Profiles

Use these endpoints to work with device profiles.

The device profile schema is: [deviceProfile.json](#) . The operations are:

- <https://paragon.example.net/traffic-engineering/api/netconf/v2/profiles/> [GET : get all device profile, POST : create new device profiles, PUT: update device profiles, DELETE : delete device profiles]

- <https://paragon.example.net/traffic-engineering/api/netconf/v2/collection/liveNetwork>
[POST: create a collection job with requested device profile ids]
- <https://paragon.example.net:8443/traffic-engineering/api/netconf/v2/collection/<id>>
[GET: get a collection job status with requested job id]

Method	URI	Description
GET	/netconf/v2/profiles	Gets all profiles.
POST	/netconf/v2/profiles	Creates new profiles.
PUT	/netconf/v2/profiles	Updates profiles.
DELETE	/netconf/v2/profiles	Deletes profiles.
POST	/netconf/v2/collection/liveNetwork	Creates a new collection.
GET	/netconf/v2/collection/{id}	Gets the Status of a collection job.

3.9.1. Get all Profiles

Method	URI	Description
GET	/netconf/v2/profiles	Gets all profiles.

Normal response codes: 200

3.9.1.1. Request

This operation does not accept a request body.

3.9.1.2. Response

Returns the following JSON document: [deviceProfile.json#/definitions/profileList](#) .

This operation does not return a response body.

3.9.2. Create New Profiles

Method	URI	Description
POST	/netconf/v2/profiles	Creates new profiles.

Normal response codes: 201

3.9.2.1. Request

The request must use the following JSON schema: [deviceProfile.json#/definitions/createProfileList](#) .

This operation does not accept a request body.

3.9.2.2. Response

Returns the following JSON document: [deviceProfile.json#/definitions/profileList](#)

This operation does not return a response body.

3.9.3. Update Profiles

Method	URI	Description
PUT	/netconf/v2/profiles	Updates profiles.

Normal response codes: 202

3.9.3.1. Request

The request must use the following JSON schema: [deviceProfile.json#/definitions/updateProfileList](#) and must contain an id in each profile: [deviceProfile.json#/definitions/profile](#)

This operation does not accept a request body.

3.9.3.2. Response

Returns the following JSON document: [deviceProfile.json#/definitions/profileList](#)

This operation does not return a response body.

3.9.4. Delete Profiles

Method	URI	Description
DELETE	/netconf/v2/profiles	Deletes profiles.

Normal response codes: 204

3.9.4.1. Request

The request must use the following JSON schema: [deviceProfile.json#/definitions/deleteProfileList](#) and must contain an id in each profile: [deviceProfile.json#/definitions/profile](#)

This operation does not accept a request body.

3.9.4.2. Response

Returns the following JSON document: [deviceProfile.json#/definitions/profileList](#)

This operation does not return a response body.

3.9.5. Create a New Collect

Method	URI	Description
POST	/netconf/v2/collection/liveNetwork	Creates a new collection.

Normal response codes: 201

3.9.5.1. Request

The request must use the following JSON schema: [deviceProfile.json#/definitions/startCollection](#) .

This operation does not accept a request body.

3.9.5.2. Response

Returns the following JSON document: [deviceProfile.json#/definitions/collectionStatus](#) .

This operation does not return a response body.

3.9.6. Get the Status of a Collection Job

Method	URI	Description
GET	/netconf/v2/collection/{id}	Gets the Status of a collection job.

Normal response codes: 200

3.9.6.1. Request

This table shows the URI parameters for the get the status of a collection job request:

Name	Type	Description
{collectionJobId}	Int	The unique identifier of the collection job.

This operation does not accept a request body.

3.9.6.2. Response

Returns the following JSON document: [deviceProfile.json#/definitions/collectionStatus](#) .

This operation does not return a response body.

3.10. TE-Containers

TE-Containers are related to TE-LSPs, similar to JunOS TE++ containers. The API allows the access to those TE-Containers parameter. A TE-Container has most of the TE-LSP parameters and split/merge parameters. A TE-Container will create a set of child TE-LSPs to adjust for traffic. The container normalization is handled as a separate global task.

The TE-containers schema is: [lsp-containers.json](#) . The operations are:

- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/te-containers/> [GET, POST]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/te-containers/<containerIndex>> [GET , PUT, PATCH , DELETE]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/te-containers/bulk> [POST , PUT, PATCH , DELETE] Bulk API

Method	URI	Description
GET	/topology/v2/{topology_id}/te-containers	List all TE-containers.
POST	/topology/v2/{topology_id}/te-containers	Creates a container.
GET	/topology/v2/{topology_id}/te-containers/stream	See EventSource for format. The notifications send on that stream are only teContainerEvent. The data will contain a JSON document (see Paragon Pathfinder Notification API).
GET	/topology/v2/{topology_id}/te-containers/{containerIndex}	Gets a TE Container.

Method	URI	Description
PUT	/topology/v2/{topology_id}/te-containers/{containerIndex}	Updates a TE Container
PATCH	/topology/v2/{topology_id}/te-containers/{containerIndex}	Updates a TE Container using an RFC6902 document
DELETE	/topology/v2/{topology_id}/te-containers/{containerIndex}	Deletes a container.
POST	/topology/v2/{topology_id}/te-containers/bulk	Create a list of TE Container
PUT	/topology/v2/{topology_id}/te-containers/bulk	Modify a list of TE Container
PATCH	/topology/v2/{topology_id}/te-containers/bulk	Modify a list of TE Container using a PATCH
DELETE	/topology/v2/{topology_id}/te-containers/bulk	Deletes a list of containers.

3.10.1. getAllTEContainers

Method	URI	Description
GET	/topology/v2/{topology_id}/te-containers	List all TE-containers.

Normal response codes: 200

3.10.1.1. Request

This table shows the URI parameters for the getalltecontainers request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.10.1.2. Response

Returns the following JSON document: [lsp-containers.json.json#/definitions/containerList](#) .

This operation does not return a response body.

3.10.2. createTEContainer

Method	URI	Description
POST	/topology/v2/{topology_id}/te-containers	Creates a container.

Normal response codes: 201

3.10.2.1. Request

This table shows the URI parameters for the createtecontainer request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

The input must conform to the [lsp-containers.json#/definitions/createContainer](#) schema.

This operation does not accept a request body.

3.10.2.2. Response

Follows [lsp-containers.json#/definitions/container](#) .

This operation does not return a response body.

3.10.3. Start a SSE Stream

Method	URI	Description
GET	/topology/v2/{topology_id}/te-containers/stream	See EventSource for format. The notifications send on that stream are only teContainerEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).

Normal response codes: 200

3.10.3.1. Request

This table shows the URI parameters for the start a sse stream request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.10.4. getATEContainer

Method	URI	Description
GET	/topology/v2/{topology_id}/te-containers/{containerIndex}	Gets a TE Container.

Normal response codes: 200

3.10.4.1. Request

This table shows the URI parameters for the getatecontainer request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{containerIndex}	Int	Program ID of a TE-container.

This operation does not accept a request body.

3.10.4.2. Response

Follows [lsp-containers.json#/definitions/container](#) .

This operation does not return a response body.

3.10.5. updateContainer

Method	URI	Description
PUT	/topology/v2/{topology_id}/te-containers/{containerIndex}	Updates a TE Container

Normal response codes: 201

3.10.5.1. Request

This table shows the URI parameters for the updatecontainer request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{containerIndex}	Int	Program ID of a TE-container.

The input must conform to the [lsp-containers.json#/definitions/updateContainer](#) schema.

This operation does not accept a request body.

3.10.5.2. Response

Returns the following JSON document: [lsp-containers.json#/definitions/container](#) .

This operation does not return a response body.

3.10.6. patchContainer

Method	URI	Description
PATCH	/topology/v2/{topology_id}/te-containers/{containerIndex}	Updates a TE Container using an RFC6902 document

Normal response codes: 201

3.10.6.1. Request

This table shows the URI parameters for the patchcontainer request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{containerIndex}	Int	Program ID of a TE-container.

The input must conform to the [rest-schemas/json-patch.json](#) schema, the produced document (Original resource +patch) must conform to the [lsp-containers.json#/definitions/updateContainer](#) schema.

This operation does not accept a request body.

3.10.6.2. Response

Returns the following JSON document: [lsp-containers.json#/definitions/container](#)

This operation does not return a response body.

3.10.7. deleteContainer

Method	URI	Description
DELETE	/topology/v2/{topology_id}/te-containers/{containerIndex}	Deletes a container.

The container must exist, no payload is expected or returned.

Normal response codes: 204

3.10.7.1. Request

This table shows the URI parameters for the deletecontainer request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{containerIndex}	Int	Program ID of a TE-container.

No content is expected.

This operation does not accept a request body.

3.10.7.2. Response

No content

This operation does not return a response body.

3.10.8. createContainerBulk

Method	URI	Description
POST	/topology/v2/{topology_id}/te-containers/bulk	Create a list of TE Container

Normal response codes: 200

3.10.8.1. Request

This table shows the URI parameters for the createcontainerbulk request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

The input must conform to the [lsp-containers.json#/definitions/containerCreateList](#) schema.

This operation does not accept a request body.

3.10.8.2. Response

Returns the following JSON document: [lsp-containers.json#/definitions/containerList](#) .

This operation does not return a response body.

3.10.9. updateContainerBulk

Method	URI	Description
PUT	/topology/v2/{topology_id}/te-containers/bulk	Modify a list of TE Container

Normal response codes: 200

3.10.9.1. Request

This table shows the URI parameters for the updatecontainerbulk request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

The input must conform to the [lsp-containers.json#/definitions/containerUpdateList](#) schema.

This operation does not accept a request body.

3.10.9.2. Response

Returns the following JSON document: [lsp-containers.json#/definitions/containerList](#) .

This operation does not return a response body.

3.10.10. patchContainerBulk

Method	URI	Description
PATCH	/topology/v2/{topology_id}/te-containers/bulk	Modify a list of TE Container using a PATCH

Normal response codes: 200

3.10.10.1. Request

This table shows the URI parameters for the patchcontainerbulk request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

The input must conform to the [lsp-containers.json#/definitions/containerListPatch](#) schema. the resulting container list (with patch applied) must also comply to the [lsp-containers.json#/definitions/containerUpdateList](#) schema.

This operation does not accept a request body.

3.10.10.2. Response

Returns the following JSON document: [lsp-containers.json#/definitions/containerList](#)

This operation does not return a response body.

3.10.11. deleteContainerBulk

Method	URI	Description
DELETE	/topology/v2/{topology_id}/te-containers/bulk	Deletes a list of containers.

Normal response codes: 200

3.10.11.1. Request

This table shows the URI parameters for the deletecontainerbulk request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

The containers must exist, the payload must conform to [lsp-containers.json#/definitions/containerListDelete](#).

This operation does not accept a request body.

3.10.11.2. Response

This operation does not return a response body.

3.11. Facilities

Use these endpoints to work with facilities.

The facilities schema is: [facilities.json](#) . The operations are:

- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/facilities/> [GET : get all facilities]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/facilities/<facilityIndex>> [GET : get a facility, PUT : update, DELETE: delete]

Method	URI	Description
GET	/topology/v2/{topology_id}/facilities	Gets all Facilities.
POST	/topology/v2/{topology_id}/facilities	Creates a facility.
GET	/topology/v2/{topology_id}/facilities/stream	See EventSource for format. The notifications send on that stream are only facilityEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).
GET	/topology/v2/{topology_id}/facilities/{facilityIndex}	Gets a Facility.
PUT	/topology/v2/{topology_id}/facilities/{facilityIndex}	Updates a facility

Method	URI	Description
DELETE	/topology/v2/{topology_id}/facilities/{facilityIndex}	Deletes a facility.

3.11.1. getAllFacilities

Method	URI	Description
GET	/topology/v2/{topology_id}/facilities	Gets all Facilities.

Returns the following JSON document: [facilities.json#/definitions/facilityList](#) .

3.11.1.1. Request

This table shows the URI parameters for the getAllFacilities request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.11.2. createAFacility

Method	URI	Description
POST	/topology/v2/{topology_id}/facilities	Creates a facility.

Returns the following JSON document: [facilities.json#/definitions/facility](#) .

3.11.2.1. Request

This table shows the URI parameters for the createafacility request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.11.3. Start a SSE Stream

Method	URI	Description
GET	/topology/v2/{topology_id}/facilities/stream	See EventSource for format. The notifications send on that stream are only facilityEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).

Normal response codes: 200

3.11.3.1. Request

This table shows the URI parameters for the start a sse stream request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.11.4. getAFacility

Method	URI	Description
GET	/topology/v2/{topology_id}/facilities/{facilityIndex}	Gets a Facility.

Returns the following JSON document: [facilities.json#/definitions/facility](#) .

3.11.4.1. Request

This table shows the URI parameters for the getafacility request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{facilityIndex}	Int	Program ID of a facility.

This operation does not accept a request body.

3.11.5. updateFacility

Method	URI	Description
PUT	/topology/v2/{topology_id}/facilities/{facilityIndex}	Updates a facility

Returns the following JSON document: [facilities.json#/definitions/facility](#) .

3.11.5.1. Request

This table shows the URI parameters for the updatefacility request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{facilityIndex}	Int	Program ID of a facility.

This operation does not accept a request body.

3.11.6. deleteFacility

Method	URI	Description
DELETE	/topology/v2/{topology_id}/facilities/{facilityIndex}	Deletes a facility.

3.11.6.1. Request

This table shows the URI parameters for the deletefacility request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{facilityIndex}	Int	Program ID of a facility.

This operation does not accept a request body.

3.12. P2MP

Use these endpoints to retrieve the P2MP lists and manage the P2MP groups. The complete set of endpoints and parameters is described in the schema [p2mp.json](#). P2MP group in Paragon PathFinder can be provisioned using NETCONF (default) or PCEP. Since Paragon PathFinder 5.1 flow mapping to MVPN using source and group is supported.

The API allows for group and leaf discovery, creation, modification, deletion and bulk deletion. Those operations are supported using the following endpoints:

- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/p2mp/> [GET: get all P2MP groups, POST: create a new P2MP group]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/p2mp/<p2mpGroupIndex>> [GET: get a P2MP group specified by index, PUT/PATCH: modify the P2MP group, DELETE: Delete the P2MP group and all its members leaves, POST: Create a list of leaves,]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/p2mp/<p2mpGroupIndex>/<lspIndex>> [GET: get a P2MP leaf by index, PUT: get a P2MP leaf specified by index, DELETE: remove a leaf]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/p2mp/<p2mpGroupIndex>/bulk> [DELETE: bulk remove a list of leaves]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/p2mp/<p2mpGroupIndex>/bulk> [DELETE only: delete a list of TE-LSP members]

A special use case for the PATCH method is to re-provision an entire tree (or a leaf) and do not change anything. To reprovision a tree a PATCH /traffic-engineering/api/topology/v2/1/p2mp/<p2mpGroupIndex> payload

[]

The same method is used for a leaf: PATCH /traffic-engineering/api/topology/v2/1/te-lsps/<lspIndex> payload

[]

. Another use case for PATCH is to add or replace flow mapping, the PATH examples show how to achieve those operations.

Since NorthStar 6.0, the P2MP tree leaves can also be CE or Site nodes. Paragon PathFinder path computation will compute the P2MP tree using those nodes but will configure them in the network terminating at the PE nodes. Please note that currently combination of PE and non-pE (CE or Site) nodes as destinations nodes in a P2MP tree is not supported. Use can either create P2MP tree where all destination nodes consist of PE nodes only or consist of CE nodes only. It cannot be combination of both (mixture of PE and CE nodes) in a single tree.

Method	URI	Description
GET	/topology/v2/{topology_id}/p2mp	Returns a full list of P2MP groups.
POST	/topology/v2/{topology_id}/p2mp	Create a P2MP group using the following schema: p2mp.json#/definitions/createP2mpGroup . The API request contains the destination list and the common planned properties. The common planned properties may include plannedProperties with flow mapping parameters.
GET	/topology/v2/{topology_id}/p2mp/stream	See EventSource for format. The notifications send on that stream are only p2mpEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).
GET	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}	Returns the details of a specified P2MP group.
PUT	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}	Modify a specific P2MP group, if the LSPs are not specified, the common properties are changed, not the LSPs. If the LSPs are specified, the set of LSPs will be modified. i.e LSP that is not in the new list will be remove, the other will be added/updated".
PATCH	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}	Modify a specific P2MP group using a PATCH document (RFC6906). The operation has the same behavior as the PUT method. An empty PATCH will re-provision the complete tree.
DELETE	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}	No response is received from the REST API unless an error occurs.
DELETE	/topology/v2/{topology_id}/p2mp/bulk	No response is received from the REST API unless an error occurs.
POST	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}	The POST URL accepts a list of new leaves. Use the following schema to create a P2MP leaf: p2mp.json#/definitions/createP2mpLeavesList .
GET	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}/{lspIndex}	Returns the details of a specified P2MP leaf.
DELETE	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}/{lspIndex}	No response is received from the REST API unless an error occurs.
DELETE	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}/bulk	No response is received from the REST API unless an error occurs.

3.12.1. Get P2MP groups

Method	URI	Description
GET	/topology/v2/{topology_id}/p2mp	Returns a full list of P2MP groups.

Normal response codes: 200

3.12.1.1. Request

This table shows the URI parameters for the get p2mp groups request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.12.1.2. Response

Example 3.76. Get P2MP groups: JSON response

```
[
  {
    "p2mpGroupIndex": 3,
    "p2mpIndex": 40,
    "name": "geeiamtree",
    "topoObjectType": "p2mpGroup",
    "from": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.101"
    },
    "links": [
      {
        "rel": "self",
        "href": "3/"
      }
    ]
  },
  {
    "p2mpGroupIndex": 4,
    "p2mpIndex": 43,
    "name": "mapletree",
    "topoObjectType": "p2mpGroup",
    "from": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.101"
    },
    "links": [
      {
        "rel": "self",
        "href": "4/"
      }
    ]
  }
]
```


3.12.2. Create a P2MP group

Method	URI	Description
POST	/topology/v2/{topology_id}/p2mp	Create a P2MP group using the following schema: p2mp.json#/definitions/createP2mpGroup . The API request contains the destination list and the common planned properties. The common planned properties may include plannedProperties with flow mapping parameters.

Normal response codes: 200

3.12.2.1. Request

This table shows the URI parameters for the create a p2mp group request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 3.77. Create a P2MP group: JSON request

While creating the P2MP group, ensure that either the following plannedProperties parameters of all the members are identical, or the group plannedProperties are applied. Do not set the plannedProperties in the members. If you do not provide the plannedProperties parameter, the default value is applied.

Table 3.5. P2MP group creation required Attributes.

Attribute	Type	Default	Description
name	string	None	P2MP group name name
from/address	string	none	P2MP group ingress ingress
plannedProperties/bandwidth	string or integer	0	P2MP members bandwidth
plannedProperties/setupPriority	integer	7	P2MP member setup priority
plannedProperties/holdingPriority	integer	0	P2MP member setup holdingPriority
plannedProperties/design/adminGroups/attributeIncludeAny	integer	0 (not set)	P2MP member setup administrative color include any
plannedProperties/design/adminGroups/attributeIncludeAll	integer	0 (not set)	P2MP member setup administrative color include all
plannedProperties/design/adminGroups/attributeExclude	integer	0 (not set)	P2MP member setup administrative color exclude
lsp	list	None	List of members with name and destination. The name must be unique on the ingress.

The following examples show different set of parameters and results:

- A P2MP tree with 3 Members using NETCONF.

```
{
```

```

    "name": "alphatree",
    "from": { "address": "11.0.0.101", "topoObjectType": "ipv4" },
    "plannedProperties": {
      "bandwidth": "99k",
      "setupPriority": 7,
      "holdingPriority": 7,
      "design": {
        "adminGroups": {
          "attributeExclude": 0,
          "attributeIncludeAll": 0,
          "attributeIncludeAny": 0
        },
        "diversityGroup": "twins"
      }
    },
    "lsp": [
      {
        "name": "alphatree-101102",
        "to": { "address": "11.0.0.102", "topoObjectType": "ipv4" }
      },
      {
        "name": "alphatree-101103",
        "to": { "address": "11.0.0.103", "topoObjectType": "ipv4" }
      },
      {
        "name": "alphatree-101104",
        "to": { "address": "11.0.0.104", "topoObjectType": "ipv4" }
      }
    ]
  }
}

```

The response is:

```

{
  "topoObjectType": "p2mpGroup",
  "name": "alphatree",
  "from": { "address": "11.0.0.101", "topoObjectType": "ipv4" },
  "lsp": [
    {
      "name": "alphatree-101102",
      "to": { "address": "11.0.0.102", "topoObjectType": "ipv4" },
      "plannedProperties": {
        "adminStatus": "Up",
        "bandwidth": "99K",
        "correlatedRR0HopCount": 0,
        "design": {
          "diversityGroup": "TWINS"
        }
      },
      "ero": [
        {
          "address": "11.101.105.2",
          "loose": false,
          "topoObjectType": "ipv4"
        },
        {
          "address": "11.102.105.1",
          "loose": false,
          "topoObjectType": "ipv4"
        }
      ]
    }
  ],

```

```

        "holdingPriority":0,
        "lastStatusString":"[ConfigServer]<Netconf provisioning order
received",
        "pathName":"alpatree-101102_p0",
        "setupPriority":7
    },
    "controlType":"PCC",
    "from":{
        "address":"11.0.0.101",
        "topoObjectType":"ipv4"
    },
    "lspIndex":91,
    "p2mpName":"alpatree",
    "pathType":"primary",
    "provisioningType":"RSVP"
},
{
    "name":"alpatree-101103",
    "to":{"address": "11.0.0.103", "topoObjectType": "ipv4"},
    "plannedProperties":{
        "adminStatus":"Up",
        "bandwidth":"99K",
        "correlatedRROHopCount":0,
        "design":{
            "diversityGroup":"TWINS"
        },
        "ero":[
            {
                "address":"11.101.105.2",
                "loose":false,
                "topoObjectType":"ipv4"
            },
            {
                "address":"11.105.107.2",
                "loose":false,
                "topoObjectType":"ipv4"
            },
            {
                "address":"11.103.107.1",
                "loose":false,
                "topoObjectType":"ipv4"
            }
        ],
        "holdingPriority":0,
        "lastStatusString":"[ConfigServer]<Netconf provisioning order
received",
        "pathName":"alpatree-101103_p0",
        "setupPriority":7
    },
    "controlType":"PCC",
    "from":{
        "address":"11.0.0.101",
        "topoObjectType":"ipv4"
    },
    "lspIndex":92,
    "p2mpName":"alpatree",
    "pathType":"primary",
    "provisioningType":"RSVP"
},
{

```

```

    "name": "alphatree-101104",
    "to": { "address": "11.0.0.104", "topoObjectType": "ipv4" },
    "plannedProperties": {
      "adminStatus": "Up",
      "bandwidth": "99K",
      "correlatedRROHopCount": 0,
      "design": {
        "diversityGroup": "TWINS"
      },
      "ero": [
        {
          "address": "dynamic",
          "loose": false,
          "topoObjectType": "ipv4"
        }
      ],
      "holdingPriority": 0,
      "lastStatusString": "[ConfigServer]<Netconf provisioning order
received",
      "pathName": "alphatree-101104_p0",
      "setupPriority": 7
    },
    "controlType": "PCC",
    "from": {
      "address": "11.0.0.101",
      "topoObjectType": "ipv4"
    },
    "lspIndex": 93,
    "p2mpName": "alphatree",
    "pathType": "primary",
    "provisioningType": "RSVP"
  }
]
}

```

- Create a P2MP group using PCEP and associate it with a Multicast VPN. This is the Paragon PathFinder 5.1 version using the flow mapping, prior version were using different encoding.

```

{
  "name": "alphatree",
  "from": { "address": "11.0.0.191", "topoObjectType": "ipv4" },
  "plannedProperties": {
    "bandwidth": "99k",
    "setupPriority": 7,
    "holdingPriority": 7,
    "flows": [
      {
        "topoObjectType": "flow",
        "vpn": { "rd": "11.0.0.105:65000", "topoObjectType": "vpnId" },
        "source": { "address": "11.0.0.101", "length": 32, "topoObjectType":
"prefix" },
        "group": { "address": "239.2.2.2", "length": 32, "topoObjectType":
"prefix" }
      },
      {
        "topoObjectType": "flow",
        "vpn": { "rd": "11.0.0.105:65000", "topoObjectType": "vpnId" },
        "source": { "address": "11.0.0.101", "length": 32, "topoObjectType":
"prefix" },

```

```

        "group":{"address": "239.3.3.3", "length": 32, "topoObjectType":
"prefix"}
    }
  ],
},
"lsps":[
  {
    "to":{"address": "11.0.0.213", "topoObjectType": "ipv4"}
  },
  {
    "to":{"address": "11.0.0.214", "topoObjectType": "ipv4"}
  }
],
"creationConfigurationMethod":"PCEP"
}

```

The response is:

```

{
  "topoObjectType":"p2mpGroup",
  "name":"alphatree",
  "from":{"address": "11.0.0.191", "topoObjectType": "ipv4"},
  "plannedProperties":{"
    "bandwidth":"99k",
    "setupPriority":7,
    "holdingPriority":7,
    "design":{"
      "adminGroups":{"
        "attributeExclude":0,
        "attributeIncludeAll":0,
        "attributeIncludeAny":0
      }
    },
    "flows":[
      {
        "topoObjectType":"flow",
        "vpn":{"rd": "11.0.0.105:65000", "topoObjectType": "vpnId"},
        "source":{"address": "11.0.0.101", "length": 32, "topoObjectType":
"prefix"},
        "group":{"address": "239.2.2.2", "length": 32, "topoObjectType":
"prefix"}
      },
      {
        "topoObjectType":"flow",
        "vpn":{"rd": "11.0.0.105:65000", "topoObjectType": "vpnId"},
        "source":{"address": "11.0.0.101", "length": 32, "topoObjectType":
"prefix"},
        "group":{"address": "239.3.3.3", "length": 32, "topoObjectType":
"prefix"}
      }
    ]
  },
  "lsps":[
    {
      "name":"alphatree_11.0.0.213",
      "to":{"address": "11.0.0.213", "topoObjectType": "ipv4"},
      "plannedProperties":{"
        "adminStatus":"Up",
        "bandwidth":"99k",
        "holdingPriority":7,

```



```

        "pathName": "alpatree_to213_p0",
        "setupPriority": 7
    },
    "controlType": "PCEInitiated",
    "from": {
        "address": "11.0.0.191",
        "topoObjectType": "ipv4"
    },
    "lspIndex": 145953,
    "p2mpName": "alpatree",
    "pathType": "primary",
    "provisioningType": "RSVP"
},
{
    "name": "alpatree_11.0.0.214",
    "to": { "address": "11.0.0.214", "topoObjectType": "ipv4" },
    "plannedProperties": {
        "adminStatus": "Up",
        "bandwidth": "99k",
        "holdingPriority": 7,
        "setupPriority": 7
    },
    "controlType": "PCEInitiated",
    "from": {
        "address": "11.0.0.191",
        "topoObjectType": "ipv4"
    },
    "lspIndex": 145954,
    "p2mpName": "alpatree",
    "pathType": "primary",
    "provisioningType": "RSVP"
}
]
}

```

- Create a P2MP group using PCEP and CE and Site nodes as destination nodes. This is Paragon PathFinder 6.0 version. CE and Sites nodes are SiteA1, SiteZ and CEY2. Please note that currently combination of PE and Site nodes as destinations nodes in a P2MP tree is not supported. Use can either create P2MP tree with all destination nodes consist of PE nodes only or consist of CE/Site nodes only. It cannot be combination of both (mixture of PE and CE/Sites nodes) in a single tree.

```

{
    "name": "pecenoippcep1",
    "creationConfigurationMethod": "PCEP",
    "from": {
        "topoObjectType": "ipv4",
        "address": "10.0.0.1"
    },
    "plannedProperties": {
        "bandwidth": "10"
    },
    "lsps": [
        {
            "to": {
                "topoObjectType": "ipv4",
                "address": "SiteA1"
            }
        }
    ],
}

```

```
{
  "to":{
    "topoObjectType":"ipv4",
    "address":"SiteZ"
  },
  {
    "to":{
      "topoObjectType":"ipv4",
      "address":"CEY2"
    }
  }
}
```

The response is:

```
{
  "from":{
    "address":"10.0.0.1",
    "topoObjectType":"ipv4"
  },
  "name":"pecenoippcep1",
  "lsp":{
    {
      "p2mpName":"pecenoippcep1",
      "initiator":"PCE",
      "from":{
        "address":"10.0.0.1",
        "topoObjectType":"ipv4"
      },
      "name":"pecenoippcep1_SiteA1",
      "pathType":"primary",
      "plannedProperties":{
        "adminStatus":"Up",
        "bandwidth":10,
        "holdingPriority":0,
        "setupPriority":7
      },
      "to":{
        "address":"SiteA1",
        "topoObjectType":"ipv4"
      },
      "controlType":"PCEInitiated",
      "provisioningType":"RSVP",
      "lspIndex":462
    },
    {
      "p2mpName":"pecenoippcep1",
      "initiator":"PCE",
      "from":{
        "address":"10.0.0.1",
        "topoObjectType":"ipv4"
      },
      "name":"pecenoippcep1_SiteZ",
      "pathType":"primary",
      "plannedProperties":{
        "adminStatus":"Up",
        "bandwidth":10,
        "holdingPriority":0,
```

```

        "setupPriority":7
      },
      "to":{
        "address":"SiteZ",
        "topoObjectType":"ipv4"
      },
      "controlType":"PCEInitiated",
      "provisioningType":"RSVP",
      "lspIndex":464
    },
    {
      "p2mpName":"pecenoippcep1",
      "initiator":"PCE",
      "from":{
        "address":"10.0.0.1",
        "topoObjectType":"ipv4"
      },
      "name":"pecenoippcep1_CEX2",
      "pathType":"primary",
      "plannedProperties":{
        "adminStatus":"Up",
        "bandwidth":10,
        "holdingPriority":0,
        "setupPriority":7
      },
      "to":{
        "address":"CEX2",
        "topoObjectType":"ipv4"
      },
      "controlType":"PCEInitiated",
      "provisioningType":"RSVP",
      "lspIndex":467
    }
  ],
  "creationConfigurationMethod":"PCEP",
  "topoObjectType":"p2mpGroup",
  "plannedProperties":{
    "bandwidth":10,
    "design":{
      "adminGroups":{
        "attributeIncludeAny":0,
        "attributeExclude":0,
        "attributeIncludeAll":0
      }
    }
  },
  "holdingPriority":0,
  "setupPriority":7
},
"location":"/NorthStar/API/v2/tenant/1/topology/1/p2mp/2",
"p2mpGroupIndex":2,
"provisioningType":"RSVP"
}

```

3.12.2.2. Response

The responses are shown per example in the request section. In general the response is a JSON document conforming to [p2mp.json#/definitions/p2mpGroup](#).

This operation does not return a response body.

3.12.3. Start a SSE Stream

Method	URI	Description
GET	/topology/v2/{topology_id}/p2mp/stream	See EventSource for format. The notifications send on that stream are only p2mpEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).

Normal response codes: 200

3.12.3.1. Request

This table shows the URI parameters for the start a sse stream request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.12.4. Get a Specific P2MP Group

Method	URI	Description
GET	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}	Returns the details of a specified P2MP group.

Normal response codes: 200

3.12.4.1. Request

This table shows the URI parameters for the get a specific p2mp group request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{p2mpGroupIndex}	Int	Signaling protocol P2MP id, which may be empty, and is the same for all TE-LSPs of a tunnel.

This operation does not accept a request body.

3.12.4.2. Response

Example 3.78. Get a Specific P2MP Group: JSON response

```
{
  "topoObjectType": "p2mpGroup",
  "p2mpGroupIndex": 8,
  "name": "geeiamtree",
  "from": { "address": "11.0.0.101", "topoObjectType": "ipv4" },
  "plannedProperties": {
    "bandwidth": "99k",
    "setupPriority": 7,
    "holdingPriority": 0,
    "design": {
      "adminGroups": {
        "attributeExclude": 0,
        "attributeIncludeAll": 0,
        "attributeIncludeAny": 0
      }
    }
  },
  "lsp": [
    {
      "name": "geeiamtree101-102",
      "to": { "address": "11.0.0.102", "topoObjectType": "ipv4" },
      "plannedProperties": {
        "adminStatus": "Up",
        "bandwidth": "99K",
        "calculatedEro": [
          {
            "address": "11.101.105.2",
            "loose": false,
            "topoObjectType": "ipv4"
          },
          {
            "address": "11.102.105.1",
            "loose": false,
```

```
        "topoObjectType":"ipv4"
      }
    ],
    "correlatedRROHopCount":2,
    "holdingPriority":0,
    "lastStatusString":"[PCServer]<Active",
    "pathName":"geeiamtree101-102_p0",
    "preferredEro":[
      {
        "address":"11.101.105.2",
        "loose":false,
        "topoObjectType":"ipv4"
      },
      {
        "address":"11.102.105.1",
        "loose":false,
        "topoObjectType":"ipv4"
      }
    ],
    "routingStatus":"Up",
    "setupPriority":7
  },
  "collectedProperties":{
    "adminStatus":"Up",
    "bandwidth":"99K",
    "calculatedEro":[
      {
        "address":"11.101.105.2",
        "loose":false,
        "topoObjectType":"ipv4"
      },
      {
        "address":"11.102.105.1",
        "loose":false,
        "topoObjectType":"ipv4"
      }
    ],
    "correlatedRROHopCount":2,
    "explicitPathName":"geeiamtree101-102_p0",
    "holdingPriority":0,
    "preferredEro":[
      {
        "address":"11.101.105.2",
        "loose":false,
        "topoObjectType":"ipv4"
      },
      {
        "address":"11.102.105.1",
        "loose":false,
        "topoObjectType":"ipv4"
      }
    ],
    "routingStatus":"Up",
    "setupPriority":7
  },
  "controlType":"PCC",
  "controller":"External",
  "from":{
    "address":"11.0.0.101",
    "topoObjectType":"ipv4"
```

```
    },
    "liveProperties": {
      "adminStatus": "Up",
      "bandwidth": 99000,
      "ero": [
        {
          "address": "11.101.105.2",
          "loose": false,
          "topoObjectType": "ipv4"
        },
        {
          "address": "11.102.105.1",
          "loose": false,
          "topoObjectType": "ipv4"
        }
      ],
      "holdingPriority": 0,
      "metric": 0,
      "operationalStatus": "Active",
      "pathName": "geeiamtree101-102_p0",
      "rro": [
        {
          "address": "11.101.105.2",
          "protectionAvailable": false,
          "protectionInUse": false,
          "topoObjectType": "ipv4"
        },
        {
          "address": "11.102.105.1",
          "protectionAvailable": false,
          "protectionInUse": false,
          "topoObjectType": "ipv4"
        }
      ],
      "setupPriority": 7
    },
    "lspIndex": 638,
    "operationalStatus": "Active",
    "p2mpIndex": 184549477,
    "p2mpName": "geeiamtree",
    "pathType": "primary",
    "provisioningType": "RSVP"
  },
  "p2mpIndex": 638
}
```

3.12.5. Update a Specific P2MP Group

Method	URI	Description
PUT	/topology/v2/{topology_id}/p2mp/{p2mpGroupId}	Modify a specific P2MP group, if the LSPs are not specified, the common properties are changed, not the LSPs. If the LSPs are specified, the set of LSPs will be modified. i.e LSP that is not in the new list will be remove, the other will be added/updated".

Normal response codes: 200

3.12.5.1. Request

This table shows the URI parameters for the update a specific p2mp group request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{p2mpGroupId}	Int	Signaling protocol P2MP id, which may be empty, and is the same for all TE-LSPs of a tunnel.

Example 3.79. Update a Specific P2MP Group: JSON request

```
{
  "topoObjectType": "p2mpGroup",
  "name": "alphytree",
  "from": { "address": "11.0.0.191", "topoObjectType": "ipv4" },
  "plannedProperties": {
    "bandwidth": "21k",
    "setupPriority": 3,
    "holdingPriority": 3
  },
  "lsp": [
    { "to": { "address": "11.0.0.102", "topoObjectType": "ipv4" } },
    { "to": { "address": "11.0.0.103", "topoObjectType": "ipv4" } },
    { "to": { "address": "11.0.0.104", "topoObjectType": "ipv4" } }
  ]
}
```

3.12.5.2. Response

Example 3.80. Update a Specific P2MP Group: JSON response

```
{
  "topoObjectType": "p2mpGroup",
  "name": "alphytree",
  "from": { "address": "11.0.0.191", "topoObjectType": "ipv4" },
  "plannedProperties": {
    "bandwidth": "21k",
    "setupPriority": 3,
    "holdingPriority": 3,
    "design": {
      "adminGroups": {
        "attributeExclude": 0,
        "attributeIncludeAll": 0,

```



```
        "attributeIncludeAny":0
      }
    },
    "lsps":[
      {
        "name":"alphatree_to213",
        "to":{"address": "11.0.0.213", "topoObjectType": "ipv4"},
        "plannedProperties":{"
          "adminStatus":"Up",
          "bandwidth":"21k",
          "design":{"
            "routingMethod":"routeByDevice"
          },
          "holdingPriority":3,
          "pathName":"alphatree_to213_p0",
          "setupPriority":3
        },
        "controlType":"PCC",
        "from":{"
          "address":"11.0.0.191",
          "topoObjectType":"ipv4"
        },
        "lspIndex":145948,
        "p2mpName":"alphatree",
        "pathType":"primary",
        "provisioningType":"RSVP"
      },
      {
        "name":"alphatree_to214",
        "to":{"address": "11.0.0.214", "topoObjectType": "ipv4"},
        "plannedProperties":{"
          "adminStatus":"Up",
          "bandwidth":"21k",
          "design":{"
            "routingMethod":"routeByDevice"
          },
          "holdingPriority":3,
          "pathName":"alphatree_to214_p0",
          "setupPriority":3
        },
        "controlType":"PCC",
        "from":{"
          "address":"11.0.0.191",
          "topoObjectType":"ipv4"
        },
        "lspIndex":145949,
        "p2mpName":"alphatree",
        "pathType":"primary",
        "provisioningType":"RSVP"
      }
    ]
  }
}
```

3.12.6. Update a Specific P2MP Group using a PATCH document

Method	URI	Description
PATCH	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}	Modify a specific P2MP group using a PATCH document (RFC6906). The operation has the same behavior as the PUT method. An empty PATCH will re-provision the complete tree.

Normal response codes: 200

3.12.6.1. Request

This table shows the URI parameters for the update a specific p2mp group using a patch document request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{p2mpGroupIndex}	Int	Signaling protocol P2MP id, which may be empty, and is the same for all TE-LSPs of a tunnel.

Example 3.81. Update a Specific P2MP Group using a PATCH document: JSON request

The following examples show different set of parameters and results:

- Modify the group bandwidth, setup and holding priority.

```
[{ "op": "add", "path": "/plannedProperties/bandwidth", "value": "21k" },
{ "op": "add", "path": "/plannedProperties/setupPriority", "value": 3 },
{ "op": "add", "path": "/plannedProperties/holdingPriority", "value": 3 }]
```

- Modify flow mapping: replace all (initial flows with groups 239.2.2.2 and 239.3.3.3) flows.

```
[
  {
    "op": "add",
    "path": "/plannedProperties/flows",
    "value": [
      {
        "topoObjectType": "flow",
        "vpn": { "rd": "11.0.0.105:65000", "topoObjectType": "vpnId" },
        "source": { "address": "11.0.0.101", "length": 32, "topoObjectType": "prefix" },
        "group": { "address": "239.4.4.4", "length": 32, "topoObjectType": "prefix" }
      },
      {
        "topoObjectType": "flow",
        "vpn": { "rd": "11.0.0.105:65000", "topoObjectType": "vpnId" },
        "source": { "address": "11.0.0.101", "length": 32, "topoObjectType": "prefix" }
      }
    ]
  }
]
```

```

      "group":{"address": "239.5.5.5", "length": 32, "topoObjectType":
"prefix"}
    }
  ]
}
]

```

- Modify flow mapping: insert a new flow (initial flows with groups 239.2.2.2 and 239.3.3.3) flows.

```

[
  {
    "op":"add",
    "path":"/plannedProperties/flows/0",
    "value": {
      "topoObjectType":"flow",
      "vpn":{"rd": "11.0.0.105:65000", "topoObjectType": "vpnId"},
      "source":{"address": "11.0.0.101", "length": 32, "topoObjectType":
"prefix"},
      "group":{"address": "239.1.1.1", "length": 32, "topoObjectType":
"prefix"}
    }
  }
]

```

3.12.6.2. Response

Example 3.82. Update a Specific P2MP Group using a PATCH document: JSON response

The response to the requests are

- Modify the group bandwidth, setup and holding priority.

```

{
  "topoObjectType":"p2mpGroup",
  "name":"alphytree",
  "from":{"address": "11.0.0.191", "topoObjectType": "ipv4"},
  "plannedProperties":{"
    "bandwidth":"21k",
    "setupPriority":3,
    "holdingPriority":3,
    "design":{"
      "adminGroups":{"
        "attributeExclude":0,
        "attributeIncludeAll":0,
        "attributeIncludeAny":0
      }
    }
  },
  "lsp": [
    {
      "name":"alphytree_to213",
      "to":{"address": "11.0.0.213", "topoObjectType": "ipv4"},
      "plannedProperties":{"
        "adminStatus":"Up",
        "bandwidth":"21k",
        "design":{"
          "routingMethod":"routeByDevice"

```

```

    },
    "holdingPriority":3,
    "pathName":"alpatree_to213_p0",
    "setupPriority":3
  },
  "controlType":"PCC",
  "from":{
    "address":"11.0.0.191",
    "topoObjectType":"ipv4"
  },
  "lspIndex":145948,
  "p2mpName":"alpatree",
  "pathType":"primary",
  "provisioningType":"RSVP"
},
{
  "name":"alpatree_to214",
  "to":{"address": "11.0.0.214", "topoObjectType": "ipv4"},
  "plannedProperties":{
    "adminStatus":"Up",
    "bandwidth":"21k",
    "design":{
      "routingMethod":"routeByDevice"
    },
    "holdingPriority":3,
    "pathName":"alpatree_to214_p0",
    "setupPriority":3
  },
  "controlType":"PCC",
  "from":{
    "address":"11.0.0.191",
    "topoObjectType":"ipv4"
  },
  "lspIndex":145949,
  "p2mpName":"alpatree",
  "pathType":"primary",
  "provisioningType":"RSVP"
}
]
}

```

- Modify flow mapping: replace all flows.

```

{
  "topoObjectType":"p2mpGroup",
  "name":"alpatree",
  "from":{"address": "11.0.0.191", "topoObjectType": "ipv4"},
  "plannedProperties":{
    "bandwidth":"99k",
    "setupPriority":7,
    "holdingPriority":7,
    "design":{
      "adminGroups":{
        "attributeExclude":0,
        "attributeIncludeAll":0,
        "attributeIncludeAny":0
      }
    }
  },
  "flows":[
    {

```

```

        "topoObjectType": "flow",
        "vpn": { "rd": "11.0.0.105:65000", "topoObjectType": "vpnId" },
        "source": { "address": "11.0.0.101", "length": 32, "topoObjectType":
"prefix" },
        "group": { "address": "239.4.4.4", "length": 32, "topoObjectType":
"prefix" }
    },
    {
        "topoObjectType": "flow",
        "vpn": { "rd": "11.0.0.105:65000", "topoObjectType": "vpnId" },
        "source": { "address": "11.0.0.101", "length": 32, "topoObjectType":
"prefix" },
        "group": { "address": "239.5.5.5", "length": 32, "topoObjectType":
"prefix" }
    }
]
},
"lsp": [
{
    "name": "alpatree_11.0.0.213",
    "to": { "address": "11.0.0.213", "topoObjectType": "ipv4" },
    "plannedProperties": {
        "adminStatus": "Up",
        "bandwidth": "99k",
        "holdingPriority": 7,
        "pathName": "alpatree_to213_p0",
        "setupPriority": 7
    },
    "controlType": "PCEInitiated",
    "from": {
        "address": "11.0.0.191",
        "topoObjectType": "ipv4"
    },
    "lspIndex": 145953,
    "p2mpName": "alpatree",
    "pathType": "primary",
    "provisioningType": "RSVP"
},
{
    "name": "alpatree_11.0.0.214",
    "to": { "address": "11.0.0.214", "topoObjectType": "ipv4" },
    "plannedProperties": {
        "adminStatus": "Up",
        "bandwidth": "99k",
        "holdingPriority": 7,
        "setupPriority": 7
    },
    "controlType": "PCEInitiated",
    "from": {
        "address": "11.0.0.191",
        "topoObjectType": "ipv4"
    },
    "lspIndex": 145954,
    "p2mpName": "alpatree",
    "pathType": "primary",
    "provisioningType": "RSVP"
}
]
}

```

- Modify flow mapping: insert a new flow flows.

```
{
  "topoObjectType": "p2mpGroup",
  "name": "alphatree",
  "from": { "address": "11.0.0.191", "topoObjectType": "ipv4" },
  "plannedProperties": {
    "bandwidth": "99k",
    "setupPriority": 7,
    "holdingPriority": 7,
    "design": {
      "adminGroups": {
        "attributeExclude": 0,
        "attributeIncludeAll": 0,
        "attributeIncludeAny": 0
      }
    }
  },
  "flows": [
    {
      "topoObjectType": "flow",
      "vpn": { "rd": "11.0.0.105:65000", "topoObjectType": "vpnId" },
      "source": { "address": "11.0.0.101", "length": 32, "topoObjectType": "prefix" },
      "group": { "address": "239.1.1.1", "length": 32, "topoObjectType": "prefix" }
    },
    {
      "topoObjectType": "flow",
      "vpn": { "rd": "11.0.0.105:65000", "topoObjectType": "vpnId" },
      "source": { "address": "11.0.0.101", "length": 32, "topoObjectType": "prefix" },
      "group": { "address": "239.2.2.2", "length": 32, "topoObjectType": "prefix" }
    },
    {
      "topoObjectType": "flow",
      "vpn": { "rd": "11.0.0.105:65000", "topoObjectType": "vpnId" },
      "source": { "address": "11.0.0.101", "length": 32, "topoObjectType": "prefix" },
      "group": { "address": "239.3.3.3", "length": 32, "topoObjectType": "prefix" }
    }
  ],
  "lsps": [
    {
      "name": "alphatree_11.0.0.213",
      "to": { "address": "11.0.0.213", "topoObjectType": "ipv4" },
      "plannedProperties": {
        "adminStatus": "Up",
        "bandwidth": "99k",
        "holdingPriority": 7,
        "pathName": "alphatree_to213_p0",
        "setupPriority": 7
      },
      "controlType": "PCEInitiated",
      "from": {
        "address": "11.0.0.191",
        "topoObjectType": "ipv4"
      }
    }
  ]
}
```

```
    "lspIndex":145953,  
    "p2mpName":"alphatree",  
    "pathType":"primary",  
    "provisioningType":"RSVP"  
  },  
  {  
    "name":"alphatree_11.0.0.214",  
    "to":{"address": "11.0.0.214", "topoObjectType": "ipv4"},  
    "plannedProperties":{"  
      "adminStatus":"Up",  
      "bandwidth":"99k",  
      "holdingPriority":7,  
      "setupPriority":7  
    },  
    "controlType":"PCEInitiated",  
    "from":{"  
      "address":"11.0.0.191",  
      "topoObjectType":"ipv4"  
    },  
    "lspIndex":145954,  
    "p2mpName":"alphatree",  
    "pathType":"primary",  
    "provisioningType":"RSVP"  
  }  
]  
}
```

3.12.7. Delete a P2MP Group

Method	URI	Description
DELETE	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}	No response is received from the REST API unless an error occurs.

Normal response codes: 204

3.12.7.1. Request

This table shows the URI parameters for the delete a p2mp group request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{p2mpGroupIndex}	Int	Signaling protocol P2MP id, which may be empty, and is the same for all TE-LSPs of a tunnel.

This operation does not accept a request body.

3.12.8. Delete list of P2MP Groups

Method	URI	Description
DELETE	/topology/v2/{topology_id}/p2mp/bulk	No response is received from the REST API unless an error occurs.

Normal response codes: 204

3.12.8.1. Request

This table shows the URI parameters for the delete list of p2mp groups request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{p2mpGroupIndex}	Int	Bulk process P2MP Groups.

Example 3.83. Delete list of P2MP Groups: JSON request

```
[  
  {"p2mpGroupIndex": 3},  
  {"p2mpGroupIndex": 4}  
]
```

3.12.9. Create a P2MP Leaf

Method	URI	Description
POST	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}	The POST URL accepts a list of new leaves. Use the following schema to create a P2MP leaf: p2mp.json#/definitions/createP2mpLeavesList .

Normal response codes: 200

3.12.9.1. Request

This table shows the URI parameters for the create a p2mp leaf request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{p2mpGroupIndex}	Int	Signaling protocol P2MP id, which may be empty, and is the same for all TE-LSPs of a tunnel.

Example 3.84. Create a P2MP Leaf: JSON request

```
[
  {
    "name": "alphyatree_to225",
    "to": { "address": "11.0.0.225", "topoObjectType": "ipv4" }
  }
]
```

3.12.9.2. Response

Example 3.85. Create a P2MP Leaf: JSON response

```
[
  {
    "name": "alphyatree_to225",
    "to": { "address": "11.0.0.225", "topoObjectType": "ipv4" },
    "plannedProperties": {
      "adminStatus": "Up",
      "bandwidth": "21k",
      "holdingPriority": 3,
      "pathName": "alphyatree_to225_p0",
      "setupPriority": 3
    },
    "controlType": "PCC",
    "from": {
      "address": "11.0.0.191",
      "topoObjectType": "ipv4"
    },
    "lspIndex": 145950,
    "p2mpName": "alphyatree",
    "pathType": "primary",
    "provisioningType": "RSVP"
  }
]
```

3.12.10. Get a Specific P2MP Leaf

Method	URI	Description
GET	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}/{lspIndex}	Returns the details of a specified P2MP leaf.

Normal response codes: 200

3.12.10.1. Request

This table shows the URI parameters for the get a specific p2mp leaf request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{p2mpGroupIndex}	Int	Signaling protocol P2MP id, which may be empty, and is the same for all TE-LSPs of a tunnel.
{lspIndex}	Int	The unique lspIndex.

This operation does not accept a request body.

3.12.10.2. Response

Example 3.86. Get a Specific P2MP Leaf: JSON response

```
{
  "operationalStatus": "Active",
  "plannedProperties": {
    "bandwidth": "99K",
    "setupPriority": 7,
    "holdingPriority": 0,
    "calculatedEro": [
      {
        "topoObjectType": "ipv4",
        "address": "11.101.105.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.102.105.1",
        "loose": false
      }
    ],
    "routingStatus": "Up",
    "pathName": "geeiamtree101-102_p0",
    "adminStatus": "Up",
    "preferredEro": [
      {
        "topoObjectType": "ipv4",
        "address": "11.101.105.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.102.105.1",
        "loose": false
      }
    ]
  }
}
```

```
    ],
    "lastStatusString": "[PCServer]<Active",
    "correlatedRR0HopCount": 2
  },
  "name": "geeiamtree101-102",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.101"
  },
  "pathType": "primary",
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.102"
  },
  "lspIndex": 638,
  "controlType": "PCC",
  "provisioningType": "RSVP",
  "p2mpName": "geeiamtree",
  "collectedProperties": {
    "bandwidth": "99K",
    "setupPriority": 7,
    "holdingPriority": 0,
    "calculatedEro": [
      {
        "topoObjectType": "ipv4",
        "address": "11.101.105.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.102.105.1",
        "loose": false
      }
    ]
  },
  "routingStatus": "Up",
  "explicitPathName": "geeiamtree101-102_p0",
  "adminStatus": "Up",
  "preferredEro": [
    {
      "topoObjectType": "ipv4",
      "address": "11.101.105.2",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.102.105.1",
      "loose": false
    }
  ],
  "correlatedRR0HopCount": 2
},
"p2mpIndex": 184549477,
"liveProperties": {
  "bandwidth": 99000,
  "metric": 0,
  "setupPriority": 7,
  "holdingPriority": 0,
  "operationalStatus": "Active",
  "adminStatus": "Up",
  "ero": [
```

```
    {
      "topoObjectType": "ipv4",
      "address": "11.101.105.2",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.102.105.1",
      "loose": false
    }
  ],
  "rrro": [
    {
      "topoObjectType": "ipv4",
      "address": "11.101.105.2",
      "protectionInUse": false,
      "protectionAvailable": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.102.105.1",
      "protectionInUse": false,
      "protectionAvailable": false
    }
  ],
  "pathName": "geeiamtree101-102_p0"
},
"controller": "External"
}
```

3.12.11. Delete a P2MP Leaf

Method	URI	Description
DELETE	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}/{lspIndex}	No response is received from the REST API unless an error occurs.

Normal response codes: 204

3.12.11.1. Request

This table shows the URI parameters for the delete a p2mp leaf request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{p2mpGroupIndex}	Int	Signaling protocol P2MP id, which may be empty, and is the same for all TE-LSPs of a tunnel.
{lspIndex}	Int	The unique lspIndex.

This operation does not accept a request body.

3.12.12. Delete list of P2MP leaves

Method	URI	Description
DELETE	/topology/v2/{topology_id}/p2mp/{p2mpGroupIndex}/bulk	No response is received from the REST API unless an error occurs.

Normal response codes: 204

3.12.12.1. Request

This table shows the URI parameters for the delete list of p2mp leaves request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{p2mpGroupIndex}	Int	Signaling protocol P2MP id, which may be empty, and is the same for all TE-LSPs of a tunnel.
{lspIndex}	Int	Bulk process P2MP leaves.

Example 3.87. Delete list of P2MP leaves: JSON request

```
[
  {"lspIndex": 145948},
  {"lspIndex": 145950}
]
```

3.13. VRF

The endpoints defined here allows to discover which VRFs are configured in the router. This endpoint has been introduced in NorthStar 5.1 and is read-only. The information from this endpoint is opulated using the device collection.

Those operations are supported using the following endpoints:

- https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/vrfs/ [GET: list the nodes with VRF information]
- https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/vrfs/<nodeIndex> [GET: list the VRF on a given node]

The detail of the information provided is defined in the associated json-schema [vrfs.json](#).

Method	URI	Description
GET	/topology/v2/{topology_id}/vrfs	Returns the list of VRF summary: which node have vrf information and how many vrf are present.
GET	/topology/v2/{topology_id}/vrfs/{nodeIndex}	Returns the list of discovered VRF for that node.

3.13.1. Get the nodes with VRF

Method	URI	Description
GET	/topology/v2/{topology_id}/vrfs	Returns the list of VRF summary: which node have vrf information and how many vrf are present.

Normal response codes: 200

3.13.1.1. Request

This table shows the URI parameters for the get the nodes with vrf request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.13.1.2. Response

Example 3.88. Get the nodes with VRF : JSON response

```
[
  {
    "topologyIndex":1,
    "topoObjectType":"vrfSummary",
    "links":[{"href": "1/", "rel": "vrfs"}],
    "node":{"id": "0110.0000.0102", "nodeIndex": 1, "topoObjectType": "node",
    "topologyIndex": 1},
    "entries":6
  },
  {
    "topologyIndex":1,
    "topoObjectType":"vrfSummary",
    "links":[{"href": "2/", "rel": "vrfs"}],
    "node":{"id": "0110.0000.0105", "nodeIndex": 2, "topoObjectType": "node",
    "topologyIndex": 1},
    "entries":1
  },
  {
    "topologyIndex":1,
    "topoObjectType":"vrfSummary",
    "links":[{"href": "3/", "rel": "vrfs"}],
    "node":{"id": "0110.0000.0106", "nodeIndex": 3, "topoObjectType": "node",
    "topologyIndex": 1},
    "entries":6
  },
  {
    "topologyIndex":1,
    "topoObjectType":"vrfSummary",
    "links":[{"href": "4/", "rel": "vrfs"}],
    "node":{"id": "0110.0000.0107", "nodeIndex": 4, "topoObjectType": "node",
    "topologyIndex": 1},
    "entries":3
  }
]
```


]



3.13.2. List the VRF on a given node

Method	URI	Description
GET	/topology/v2/{topology_id}/vrfs/{nodeIndex}	Returns the list of discovered VRF for that node.

Normal response codes: 200

3.13.2.1. Request

This table shows the URI parameters for the list the vrf on a given node request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{nodeIndex}	Int	The unique nodeIndex.

This operation does not accept a request body.

3.13.2.2. Response

Example 3.89. List the VRF on a given node: JSON response

```
[
  {
    "topologyIndex":1,
    "topoObjectType":"vrf",
    "node":{"id": "0110.0000.0102", "nodeIndex": 1, "topoObjectType": "node",
"topologyIndex": 1},
    "instanceName":"cherryblossom",
    "routeDistinguisher":"11.0.0.102:65000",
    "interfaces":[{"interfaceName": "ge-0/0/7.0"}],
    "importPolicies":[{"policyName": "65000:100"}],
    "exportPolicies":[{"policyName": "65000:100"}]
  },
  {
    "topologyIndex":1,
    "topoObjectType":"vrf",
    "node":{"id": "0110.0000.0102", "nodeIndex": 1, "topoObjectType": "node",
"topologyIndex": 1},
    "instanceName":"etree1",
    "routeDistinguisher":"11.0.0.102:301",
    "interfaces":[{"interfaceName": "ge-0/0/5.301"}],
    "importPolicies":[{"policyName": "6500:301"}],
    "exportPolicies":[{"policyName": "6500:301"}],
    "VLANID":100
  },
  {
    "topologyIndex":1,
    "topoObjectType":"vrf",
    "node":{"id": "0110.0000.0102", "nodeIndex": 1, "topoObjectType": "node",
"topologyIndex": 1},
    "instanceName":"l2vpn_3_ce_101",
    "routeDistinguisher":"",
    "interfaces":[{"interfaceName": "ge-0/0/5.3"}]
```

```

    },
    {
      "topologyIndex": 1,
      "topoObjectType": "vrf",
      "node": { "id": "0110.0000.0102", "nodeIndex": 1, "topoObjectType": "node",
"topologyIndex": 1},
      "instanceName": "vpn_0",
      "routeDistinguisher": "11.0.0.102:100",
      "interfaces": [{"interfaceName": "ge-0/0/1.0"}, {"interfaceName": "lo0.
1"}],
      "importPolicies": [{"policyName": "11:100"}],
      "exportPolicies": [{"policyName": "11:100"}],
      "lspMatchRegex": "^Silver.*"
    },
    {
      "topologyIndex": 1,
      "topoObjectType": "vrf",
      "node": { "id": "0110.0000.0102", "nodeIndex": 1, "topoObjectType": "node",
"topologyIndex": 1},
      "instanceName": "vpws_sh6_sh2_vlan1002",
      "routeDistinguisher": "11.0.0.102:2002",
      "interfaces": [{"interfaceName": "ge-0/0/5.1002"}],
      "importPolicies": [{"policyName": "6500:1002"}],
      "exportPolicies": [{"policyName": "6500:1002"}]
    },
    {
      "topologyIndex": 1,
      "topoObjectType": "vrf",
      "node": { "id": "0110.0000.0102", "nodeIndex": 1, "topoObjectType": "node",
"topologyIndex": 1},
      "instanceName": "vpn_200_ce_101",
      "routeDistinguisher": "",
      "interfaces": [{"interfaceName": "ge-0/0/5.200"}]
    }
  ]

```

3.14. BGP Routes

If PRDP is configured for a set of nodes, this API allows to see the BGP routes retrived by Paragon PathFinder.

The schema is: [route.json](#) . The operations are:

- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/routes/> [GET : get the list of nodes with bgp routes (summary only)]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/routes/<nodeIndex>> [GET : get the list of routes on a node]

Paragon PathFinder is limiting the number of routes fetched, the list may be incomplete.

Method	URI	Description
GET	/topology/v2/{topology_id}/routes	List all nodes with BGP routes and the number of routes for that node.
GET	/topology/v2/{topology_id}/routes/{nodeIndex}	Gets the routes maintained by Paragon PathFinder for a node.

3.14.1. getRoutesSummary

Method	URI	Description
GET	/topology/v2/{topology_id}/routes	List all nodes with BGP routes and the number of routes for that node.

Normal response codes: 200

3.14.1.1. Request

This table shows the URI parameters for the getroutessummary request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.14.1.2. Response

Example 3.90. : JSON response

Returns the following JSON document: [route.json.json#/definitions/summaryList](#) .

```
[
  {
    "topologyIndex": 1,
    "topologyObjectType": "routeSummary",
    "node": {
      "topoObjectType": "node",
      "nodeIndex": 7,
      "id": "0110.0000.0012",
      "topologyIndex": 1
    },
    "entries": 50
  },
  {
    "topologyIndex": 1,
    "topologyObjectType": "routeSummary",
    "node": {
      "topoObjectType": "node",
      "nodeIndex": 5,
      "id": "0110.0000.0010",
      "topologyIndex": 1
    },
    "entries": 98
  },
  {
    "topologyIndex": 1,
    "topologyObjectType": "routeSummary",
    "node": {
      "topoObjectType": "node",
      "nodeIndex": 6,
      "id": "0110.0000.0011",
      "topologyIndex": 1
    }
  }
]
```

```
    },  
    "entries": 50  
  }  
]
```

3.14.2. getNodeRoutes

Method	URI	Description
GET	/topology/v2/{topology_id}/routes/{nodeIndex}	Gets the routes maintained by Paragon PathFinder for a node.

Normal response codes: 200

3.14.2.1. Request

This table shows the URI parameters for the getnoderoutes request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{nodeIndex}	Int	Program ID of a node.

This operation does not accept a request body.

3.14.2.2. Response

Example 3.91. : JSON response

Follows [route.json#/definitions/routeList](#) .

```
[
  {
    "topologyIndex": 1,
    "topologyObjectType": "route",
    "node": {
      "topoObjectType": "node",
      "nodeIndex": 7,
      "id": "0110.0000.0012",
      "topologyIndex": 1
    },
    "table": "inet.0",
    "prefix": {
      "topoObjectType": "prefix",
      "address": "10.4.4.0",
      "length": 24
    },
    "pathCookie": 190819980,
    "asPath": [
      "3",
      "4"
    ],
    "protocolNextHops": [
      "11.0.0.22",
      "11.0.0.31"
    ],
    "protocol": "BGP",
    "routePreference": 170,
    "localPreference": 100,
    "vpnLabel": 0
  },
]
```

```

{
  "topologyIndex": 1,
  "topologyObjectType": "route",
  "node": {
    "topoObjectType": "node",
    "nodeIndex": 7,
    "id": "0110.0000.0012",
    "topologyIndex": 1
  },
  "table": "inet.0",
  "prefix": {
    "topoObjectType": "prefix",
    "address": "10.4.22.0",
    "length": 24
  },
  "pathCookie": 190821156,
  "asPath": [
    "2",
    "4"
  ],
  "protocolNextHops": [
    "11.0.0.22"
  ],
  "protocol": "BGP",
  "routePreference": 170,
  "localPreference": 100,
  "vpnLabel": 0
}
]

```

3.15. IPE Policies

When generation of external nodes and links through BGP policy configuration is enabled, this API allows creating, retrieving, updating and deleting IPE Policies. Note: This API is currently experimental and subject to change in future releases.

The schema is: [ipe.json](#) . The operations are:

- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/ipe/policy/> [GET : Gets all IPE policies, POST: Creates IPE Policies (bulk), PUT: Updates IPE Policies (bulk), DELETE: Deletes IPE Policies (bulk)]
- <https://paragon.example.net/traffic-engineering/api/topology/v2/<topology-id>/ipe/policy/<ipePolicyId>> [GET : Gets single IPE]

Method	URI	Description
GET	/topology/v2/{topology_id}/ipe/policy	Get all IPE Policies
POST	/topology/v2/{topology_id}/ipe/policy	Create IPE Policies
PUT	/topology/v2/{topology_id}/ipe/policy	Update IPE Policies
DELETE	/topology/v2/{topology_id}/ipe/policy	Delete IPE Policies
GET	/topology/v2/{topology_id}/ipe/policy/{ipePolicyId}	Get single IPE policy

3.15.1. getIPEPolicies

Method	URI	Description
GET	/topology/v2/{topology_id}/ipe/policy	Get all IPE Policies

Normal response codes: 200

3.15.1.1. Request

This table shows the URI parameters for the getipepolicies request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.15.1.2. Response

Example 3.92. : JSON response

```
{
  "ipe_policies": [
    {
      "id": 1,
      "node": {
        "id": "0110.0000.0011",
        "nodeIndex": 6,
        "topologyIndex": 1,
        "topoObjectType": "node"
      },
      "neighbor": "11.0.0.21",
      "prefix": "199.1.1.0/24",
      "priority": 1,
      "operationalStatus": "Up",
      "errorReason": "None"
    },
    {
      "id": 3,
      "node": {
        "id": "0110.0000.0011",
        "nodeIndex": 6,
        "topologyIndex": 1,
        "topoObjectType": "node"
      },
      "neighbor": "11.0.0.21",
      "prefix": "198.1.1.0/24",
      "priority": 3,
      "operationalStatus": "Up",
      "errorReason": "None"
    },
    {
      "id": 4,
      "node": {
```

```
        "id": "0110.0000.0012",
        "nodeIndex": 7,
        "topologyIndex": 1,
        "topoObjectType": "node"
    },
    "neighbor": "11.0.0.31",
    "prefix": "177.1.2.0/24",
    "priority": 5,
    "operationalStatus": "Up",
    "errorReason": "None"
}
]
```

3.15.2. createIPEPolicies

Method	URI	Description
POST	/topology/v2/{topology_id}/ipe/policy	Create IPE Policies

Normal response codes: 202

3.15.2.1. Request

This table shows the URI parameters for the createipepolicies request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Required schema: [ipe.json#/definitions/bulkIPECreate](#)

This operation does not accept a request body.

3.15.2.2. Response

This operation does not return a response body.

This operation does not return a response body.

3.15.3. updateIPEPolicies

Method	URI	Description
PUT	/topology/v2/{topology_id}/ipe/policy	Update IPE Policies

Normal response codes: 202

3.15.3.1. Request

This table shows the URI parameters for the updateipepolicies request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Required schema: [ipe.json#/definitions/bulkIPEUpdate](#)

This operation does not accept a request body.

3.15.3.2. Response

This operation does not return a response body.

This operation does not return a response body.

3.15.4. deleteIPEPolicies

Method	URI	Description
DELETE	/topology/v2/{topology_id}/ipe/policy	Delete IPE Policies

Normal response codes: 202

3.15.4.1. Request

This table shows the URI parameters for the deleteipepolicies request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Required schema: [ipe.json#/definitions/bulkIPEDelete](#)

This operation does not accept a request body.

3.15.4.2. Response

This operation does not return a response body.

This operation does not return a response body.

3.15.5. getIPEPolicy

Method	URI	Description
GET	/topology/v2/{topology_id}/ipe/policy/{ipePolicyId}	Get single IPE policy

Normal response codes: 200

3.15.5.1. Request

This table shows the URI parameters for the getipepolicy request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{ipePolicyId}	Int	Program ID of an IPE policy

This operation does not accept a request body.

3.15.5.2. Response

Example 3.93. : JSON response

```
{
  "id": 4,
  "node": {
    "id": "0110.0000.0012",
    "nodeIndex": 7,
    "topologyIndex": 1,
    "topoObjectType": "node"
  },
  "neighbor": "11.0.0.31",
  "prefix": "177.1.2.0/24",
  "priority": 5,
  "operationalStatus": "Up",
  "errorReason": "None"
}
```

3.16. Analytics API

The analytics REST API allows to query the analytics data collected by NorthStar controller. Each URL accepts a POST request with as set of parameters refining the query.

Each queried object has specific counters. Those counters are dynamically generated based on the data collected. The queries are

- Interfaces: /NorthStar/API/v2/tenant/1/statistics/interfaces/fields
- LSP/demands: /NorthStar/API/v2/tenant/1/statistics/te-lsps/fields
- Interface delay/loss: /NorthStar/API/v2/tenant/1/statistics/delay/fields

The information returned is a JSON object with a fields property containing a list of strings. For instance:

```
{ "fields": [ "average_rtt", "jitter", "loss_percent", "max_rtt", "min_rtt",
"one_way" ] }
```

Other common parameters are `startTime` and `endTime`, Those parameter represent an absolute or relative time.

Absolute values can be either a string (ISO8601 format, date is required, time is optional) or an integer (milliseconds since epoch). The special value `now` is accepted.

The Date mathematic consist of a plus or minus followed by a time unit: y (year), M(month), w (week), d (day), h or H (hours), m (minutes) or s (seconds). For instance a query with `startTime` set to `"now-1d"`, `endTime` `"now"` will retrieve the statistics for the last day. This field is based on [ElasticSearch Date Math](#)

Method	URI	Description
POST	/statistics/v2/demands/bulk	The POST request accepts a JSON object describing the query parameters. - interval (string): Time duration for instance 1d, 1y, 24h,. - counter : Either a single counter or an array of counters. . - aggregation (string): Aggregation function to use to aggregate values.the value must be one of: - avg - max - min - cardinality - sum - startTime (Either string or integer): Date time. . - demand (array): Array of demand objects. - endTime (Either string or integer): Date time. .
POST	/statistics/v2/interfaces/de-lay/{device_name}/{interface_name}	The POST request accepts a JSON object describing the query parameters. - counter (string): Name of the counter to aggregate by.. - endTime (Either string or integer): Date time. . - aggregation (string): Aggregation function to use to aggregate values.the value must be one of: - avg - max - min - cardinality - sum - startTime (Either string or integer): Date time. . - interval (string): Time duration for instance 1d, 1y, 24h,.

Method	URI	Description
POST	/statistics/v2/interfaces/bulk	The POST request accepts a JSON object describing the query parameters. - interval (string): Time duration for instance 1d, 1y, 24h.. - counter : Either a single counter or an array of counters. . - aggregation (string): Aggregation function to use to aggregate values.the value must be one of: - avg - max - min - cardinality - sum - startTime (Either string or integer): Date time. . - interface (array): Array of interface objects. - endTime (Either string or integer): Date time. .
POST	/statistics/v2/interfaces/fields	The POST request accepts a JSON object describing the query parameters. - endTime (Either string or integer): Date time. . - startTime (Either string or integer): Date time. .
POST	/statistics/v2/interfaces/top	The POST request accepts a JSON object describing the query parameters. - counter (string): Name of the counter to aggregate by.. - aggregation (string): Aggregation function to use to aggregate values.the value must be one of: - avg - max - min - cardinality - sum - startTime (Either string or integer): Date time. . - device (array): Array of node objects. - endTime (Either string or integer): Date time. . - size (integer): Max number of interfaces top return.
POST	/statistics/v2/interfaces/child-traffic/{device_name}/{interface_name}	The POST request accepts a JSON object describing the query parameters. - counter (string): Name of the counter to aggregate by.. - endTime (Either string or integer): Date time. .

Method	URI	Description
		- aggregation (string): Aggregation function to use to aggregate values.the value must be one of: - avg - max - min - cardinality - sum - startTime (Either string or integer): Date time. . - interval (string): Time duration for instance 1d, 1y, 24h,. .
POST	/statistics/v2/interfaces/bulkdelay	The POST request accepts a JSON object describing the query parameters. - interval (string): Time duration for instance 1d, 1y, 24h,. . - counter : Either a single counter or an array of counters. . - aggregation (string): Aggregation function to use to aggregate values.the value must be one of: - avg - max - min - cardinality - sum - startTime (Either string or integer): Date time. . - interface (array): Array of interface objects. - endTime (Either string or integer): Date time. .
POST	/statistics/v2/interfaces/traffic	The POST request accepts a JSON object describing the query parameters. - endTime (Either string or integer): Date time. . - aggregation (string): Aggregation function to use to aggregate values.the value must be one of: - avg - max - min - cardinality - sum - startTime (Either string or integer): Date time. . - interval (string): Time duration for instance 1d, 1y, 24h,. .

Method	URI	Description
POST	/statistics/v2/interfaces/traffic/{device_name}	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,. .
POST	/statistics/v2/interfaces/traffic/{device_name}/{interface_name}	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,. .
POST	/statistics/v2/interfaces/topdelay	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. .

Method	URI	Description
		• device (array): Array of node objects. • endTime (Either string or integer): Date time. . • size (integer): Max number of interfaces top return.
POST	/statistics/v2/childtraffic/bulk	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interface (array): Array of interface objects. • endTime (Either string or integer): Date time. .
POST	/statistics/v2/jnxcos/bulk	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interface (array): Array of interface objects. • endTime (Either string or integer): Date time. .
POST	/statistics/v2/delay/fields	The POST request accepts a JSON object describing the query parameters. • endTime (Either string or integer): Date time. . • startTime (Either string or integer): Date time. .

Method	URI	Description
POST	/statistics/v2/te-lsps/bulk	The POST request accepts a JSON object describing the query parameters. - interval (string): Time duration for instance 1d, 1y, 24h,. - startTime (Either string or integer): Date time. . - counter : Either a single counter or an array of counters. . - aggregation (string): Aggregation function to use to aggregate values.the value must be one of: - avg - max - min - cardinality - sum - lsp (array): Array of lsp objects. - endTime (Either string or integer): Date time. .
POST	/statistics/v2/te-lsps/fields	The POST request accepts a JSON object describing the query parameters. - endTime (Either string or integer): Date time. . - startTime (Either string or integer): Date time. .
POST	/statistics/v2/te-lsps/top	The POST request accepts a JSON object describing the query parameters. - counter (string): Name of the counter to aggregate by.. - aggregation (string): Aggregation function to use to aggregate values.the value must be one of: - avg - max - min - cardinality - sum - startTime (Either string or integer): Date time. . - device (array): Array of node objects. - endTime (Either string or integer): Date time. . - size (number): ???.
POST	/statistics/v2/te-lsps/bulkdelay	The POST request accepts a JSON object describing the query parameters. interval (string): Time duration for instance 1d, 1y, 24h,.

Method	URI	Description
		• startTime (Either string or integer): Date time. . • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • lsp (array): Array of lsp objects. • endTime (Either string or integer): Date time. . .
POST	/statistics/v2/te-lsps/delay/{device_name}/{lsp_name}	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,. .
POST	/statistics/v2/te-lsps/traffic	The POST request accepts a JSON object describing the query parameters. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,.

Method	URI	Description
POST	/statistics/v2/te-lsps/traffic/{device_name}	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,.
POST	/statistics/v2/te-lsps/traffic/{device_name}/{lsp_name}	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,.
POST	/statistics/v2/te-lsps/events/bulk	The POST request accepts a JSON object describing the query parameters. • from (number): TODO. • startTime (Either string or integer): Date time. . • filter (object): Generic filter object, ???. • lsp (array): Array of lsp objects. • endTime (Either string or integer): Date time. . • order : Result order (asc or desc).the value must be one of: • asc

Method	URI	Description
		• desc • size (number): ???. .
POST	/statistics/v2/te-lsps/events/{lsp_name}	This query does not accept any parameter.
POST	/statistics/v2/device/bulk	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • device (array): Array of node objects. • endTime (Either string or integer): Date time. . .
POST	/statistics/v2/device/top	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • device (array): Array of node objects. • endTime (Either string or integer): Date time. . • size (number): ???. .
POST	/statistics/v2/ldp-lsps/bulk	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,.

Method	URI	Description
		• startTime (Either string or integer): Date time. . • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • lsp (array): Array of lsp objects. • endTime (Either string or integer): Date time. . •
POST	/statistics/v2/ldp-lsps/traffic/{device_name}/{fec}	The POST request accepts a JSON object describing the query parameters. • counter (string): Name of the counter to aggregate by.. • endTime (Either string or integer): Date time. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • interval (string): Time duration for instance 1d, 1y, 24h,. •
POST	/statistics/v2/net-flow/prefix_flow_demand	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. .

Method	URI	Description
		• demand (array): Array of demand objects. • endTime (Either string or integer): Date time. . .
POST	/statistics/v2/sid/bulk	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • sid (array): Array of sid objects. • endTime (Either string or integer): Date time. . .
POST	/statistics/v2/sr-te-policy/bulk	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • counter : Either a single counter or an array of counters. . • aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • policy (array): Array of policy objects. • endTime (Either string or integer): Date time. . .
POST	/statistics/v2/te-containers/traffic/bulk	The POST request accepts a JSON object describing the query parameters. • interval (string): Time duration for instance 1d, 1y, 24h,. • counter : Either a single counter or an array of counters. .

Method	URI	Description
		• aggregation (string): Aggregation function to use to aggregate values.the value must be one of: • avg • max • min • cardinality • sum • startTime (Either string or integer): Date time. . • containerLsp (array): Array of Container LSP objects. • endTime (Either string or integer): Date time. .

3.17. Transport Topology acquisition

Enables configuration of the Paragon PathFinder connection to the transport controller for retrieval of abstracted transport topology and automation of addition of information to the IP layer.

The corresponding schema is: [transportController.json](#) .

The following tables show the set of parameters describing a transport controller configuration.

Table 3.6. Create Transport controller Attributes

Attribute	Type	Fixed	Description
name	string	No	Transport controller name
notifyUrl	string	No	The REST/RESTCONF URL publishing (via Server-Sent-event) topology notifications
profileName	string	No	The profile group name the Paragon PathFinder controller must use to connect to the transport controller instances.
topologyUrl	string	No	The URL providing the abstract topology, following the IETF model supported by Paragon PathFinder .
topoObjectType	string	Yes	transportController

The other parameters are:

Table 3.7. Create Transport controller optional Attributes

Attribute	Type	Fixed	Description
interfaceType	string	No	Indicates whether the transport controller interface follows the RESTCONF draft

Attribute	Type	Fixed	Description
			or a more simple REST interface. This does not affect the Paragon PathFinder Operation.
pollUrl	string	No	The URL on the server to poll server liveliness, by default /.well-known/host-meta.
reconnectTimeout	string	No	The time, in seconds, between two reconnect attempts between controller instances.
rootUrl	string	No	Default root URL for REST-CONF datastores.
srlgPrefix	string	No	Use the SRLG prefix to generate the SRLGs for the IP topology. If set, the SRLGs will be TSRLG_<srlgPrefix>_<SRLG>, otherwise TSRLG_<SRLG>. This permits you to either separate or merge two controller SRLG spaces. By default, SRLG is not set, or it has an empty string which corresponds to merging the SRLGs space.
topologyToUse	string	No	The transport controller might return several topologies. This field permits you to select a specific topology to apply as a filter to the model te-topology-id field.

Method	URI	Description
GET	/transport-controllers/v2	Returns a full list of the transport controllers using the following schema: transportController.json#/definitions/transportControllerList
POST	/transport-controllers/v2	Creates a transport controller using the following schema: transportController.json#/definitions/createTransportController
GET	/transport-controllers/v2/{transportControllerIndex}	Returns the configuration of a transport controller.
PUT	/transport-controllers/v2/{transportControllerIndex}	Updates a transport controller using the following schema: transportController.json#/definitions/updateTransportController The parameters are the same as for transport controller creation. A parameter change triggers reconnection to the transport controller using the new parameters.
DELETE	/transport-controllers/v2/{transportControllerIndex}	Deletes a transport controller configuration. The transport node, link, and circuits created from that transport controller are not deleted. You must delete them manually, if required.

3.17.1. Get Transport Controller List

Method	URI	Description
GET	/transport-controllers/v2	Returns a full list of the transport controllers using the following schema: transportController.json#/definitions/transportControllerList

Normal response codes: 200

3.17.1.1. Request

This operation does not accept a request body.

3.17.1.2. Response

Example 3.94. Get Transport Controller List: JSON response

```
[
  {
    "transportControllerIndex": 1,
    "name": "Nortel",
    "interfaceType": "RESTCONF",
    "notifyUrl": "/streams/NETCONF-JSON",
    "pollUrl": "",
    "profileName": "NortelProfile",
    "reconnectTimeout": 10,
    "rootUrl": "",
    "srlgPrefix": "",
    "topologyToUse": "",
    "topologyUrl": "/restconf/data/ietf-te-topology:te-topologies-state",
    "topoObjectType": "transportController"
  }
]
```

3.17.2. Create Transport Controller Configuration

Method	URI	Description
POST	/transport-controllers/v2	Creates a transport controller using the following schema: transportController.json#/definitions/createTransportController

Normal response codes: 201

3.17.2.1. Request

Example 3.95. Create Transport Controller Configuration: JSON request

```
{
  "name": "Nortel",
  "topologyModel": "ietf-te-topology-01",
  "interfaceType": "RESTCONF",
  "notifyUrl": "/streams/NETCONF-JSON",
  "profileName": "NortelProfile",
  "reconnectTimeout": 10,
  "topologyUrl": "/restconf/data/ietf-te-topology:te-topologies-state",
  "topoObjectType": "transportController"
}
```

3.17.2.2. Response

Example 3.96. Create Transport Controller Configuration: JSON response

```
{
  "name": "Nortel",
  "interfaceType": "RESTCONF",
  "topologyModel": "ietf-te-topology-01",
  "notifyUrl": "/streams/NETCONF-JSON",
  "profileName": "NortelProfile",
  "reconnectTimeout": 10,
  "topologyUrl": "/restconf/data/ietf-te-topology:te-topologies-state",
  "topoObjectType": "transportController",
  "transportControllerIndex": 1
}
```

3.17.3. Get Transport Controller Configuration

Method	URI	Description
GET	/transport-controllers/v2/{transportControllerIndex}	Returns the configuration of a transport controller.

Normal response codes: 200

3.17.3.1. Request

This table shows the URI parameters for the get transport controller configuration request:

Name	Type	Description
{transportControllerIndex}	Int	The unique transportControllerIndex.

This operation does not accept a request body.

3.17.3.2. Response

Example 3.97. Get Transport Controller Configuration: JSON response

Returns the following JSON document: [transportController.json#/definitions/transportController](#) .

```
{
  "name": "Nortel",
  "interfaceType": "RESTCONF",
  "topologyModel": "ietf-te-topology-01",
  "notifyUrl": "/streams/NETCONF-JSON",
  "profileName": "NortelProfile",
  "reconnectTimeout": 10,
  "topologyUrl": "/restconf/data/ietf-te-topology:te-topologies-state",
  "topoObjectType": "transportController",
  "transportControllerIndex": 1
}
```

3.17.4. Update Transport Controller Configuration

Method	URI	Description
PUT	/transport-controllers/v2/{transportControllerIndex}	Updates a transport controller using the following schema: transportController.json#/definitions/updateTransportController The parameters are the same as for transport controller creation. A parameter change triggers reconnection to the transport controller using the new parameters.

Normal response codes: 202

3.17.4.1. Request

This table shows the URI parameters for the update transport controller configuration request:

Name	Type	Description
{transportControllerIndex}	Int	The unique transportControllerIndex.

Example 3.98. Update Transport Controller Configuration: JSON request

```
{
  "name": "Nortel",
  "interfaceType": "RESTCONF",
  "notifyUrl": "/streams/NETCONF-JSON",
  "profileName": "NortelProfile",
  "reconnectTimeout": 10,
  "topologyUrl": "/restconf/data/ietf-te-topology:te-topologies-state",
  "srlgPrefix": "space1",
  "topoObjectType": "transportController",
  "transportControllerIndex": 1
}
```

3.17.4.2. Response

Example 3.99. Update Transport Controller Configuration: JSON response

```
{
  "name": "Nortel",
  "interfaceType": "RESTCONF",
  "notifyUrl": "/streams/NETCONF-JSON",
  "profileName": "NortelProfile",
  "reconnectTimeout": 10,
  "topologyUrl": "/restconf/data/ietf-te-topology:te-topologies-state",
  "srlgPrefix": "space1",
  "topoObjectType": "transportController",
  "transportControllerIndex": 1
}
```

3.17.5. Delete Transport Controller Configuration

Method	URI	Description
DELETE	/transport-controllers/v2/{transportControllerIndex}	Deletes a transport controller configuration. The transport node, link, and circuits created from that transport controller are not deleted. You must delete them manually, if required.

Normal response codes: 204

3.17.5.1. Request

This table shows the URI parameters for the delete transport controller configuration request:

Name	Type	Description
{transportControllerIndex}	Int	The unique transportControllerIndex.

This operation does not accept a request body.

3.18. Transport Topology devices

Manages a set of instances of a transport controller.

The corresponding schema is: [transportControllerEndpoint.json](#) .

The devices are managed as group of devices. For each device the following parameters can be managed:

Table 3.8. Create Transport controller Attributes

Attribute	Type	Fixed	Description
ipAddr	string	No	IP address to connect
accessMethod	string	No	Either HTTP or HTTPS
authMethod	string	No	Authentication method; can be BASIC or NOAUTH

The other parameters are:

Table 3.9. Create transport controller optional attributes

Attribute	Type	Fixed	Description
hostName	string	No	Device name.
httpPort	int	No	HTTP port.
login	string	Yes	Login to be used for the BASIC auth.
passwd	string	Yes	Encrypted password to be used for the BASIC auth.
retry	int	No	Number of retries for the heartbeat, before the device is considered down.

Attribute	Type	Fixed	Description
timeout	int	No	Request timeout for the device.

Method	URI	Description
GET	/transport-controller-groups/v2	Returns the full list of transport controller groups using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroupList .
POST	/transport-controller-groups/v2	Creates a group of transport controller devices using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroupDescription .
DELETE	/transport-controller-groups/v2	Deletes a group of transport controller devices using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroupDescription
GET	/transport-controller-groups/v2/{transportControllerGroupName}	Returns a list of transport controller groups using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroup The password is set by empty strings.
POST	/transport-controller-groups/v2/{transportControllerGroupName}	Creates devices in a group using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroup
PUT	/transport-controller-groups/v2/{transportControllerGroupName}	Updates devices in a group using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroup
DELETE	/transport-controller-groups/v2/{transportControllerGroupName}	Deletes devices from a group using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroup

3.18.1. Get Transport Controller Device Groups

Method	URI	Description
GET	/transport-controller-groups/v2	Returns the full list of transport controller groups using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroupList .

Normal response codes: 200

3.18.1.1. Request

This operation does not accept a request body.

3.18.1.2. Response

Example 3.100. Get Transport Controller Device Groups: JSON response

```
[
  {
    "name": "NortelProfile"
  },
  {
    "name": "Default"
  }
]
```

3.18.2. Create A Transport Controller Device Group

Method	URI	Description
POST	/transport-controller-groups/v2	Creates a group of transport controller devices using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroupDescription .

The request must contain the group name to be created, as shown in the example.

Normal response codes: 201

3.18.2.1. Request

Example 3.101. Create A Transport Controller Device Group: JSON request

```
{
  "name": "NortelProfile",
  "profileType": "Network Controllers"
}
```

3.18.2.2. Response

Example 3.102. Create A Transport Controller Device Group: JSON response

```
{
  "success": true
}
```

3.18.3. Delete a Transport Controller Device Group

Method	URI	Description
DELETE	/transport-controller-groups/v2	Deletes a group of transport controller devices using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroupDescription

The request must contain the group name to be deleted, as shown in the example.

Normal response codes: 201

3.18.3.1. Request

Example 3.103. Delete a Transport Controller Device Group: JSON request

```
{
  "name": "NortelProfile",
  "profileType": "Network Controllers"
}
```

3.18.3.2. Response

Example 3.104. Delete a Transport Controller Device Group: JSON response

```
{
  "success": true
}
```

3.18.4. Get Transport Controller Device Groups

Method	URI	Description
GET	/transport-controller-groups/v2/{transportControllerGroupName}	Returns a list of transport controller groups using the following schema: transportControllerEndpoint.json#/definitions/transportControllerGroup The password is set by empty strings.

Normal response codes: 200

3.18.4.1. Request

This table shows the URI parameters for the get transport controller device groups request:

Name	Type	Description
{transportControllerGroupName}	String	The name of the transport controller devices group.

This operation does not accept a request body.

3.18.4.2. Response

Example 3.105. Get Transport Controller Device Groups: JSON response

```
[
  {
    "id": "91172e83-7b45-422c-ac1e-1940fb9a7ddf",
    "accessMethod": "HTTP",
    "authMethod": "BASIC",
    "hostName": "",
    "httpPort": 80,
    "ipAddr": "10.0.1.3",
    "login": "Login2",
    "passwd": "",
    "retry": 3,
    "timeout": 2
  },
  {
    "id": "7bcf5b8a-30f4-46ad-9a51-29daac671587",
    "accessMethod": "HTTP",
    "authMethod": "BASIC",
    "hostName": "",
    "httpPort": 80,
    "ipAddr": "10.0.1.2",
    "login": "Login",
    "passwd": "",
    "retry": 3,
    "timeout": 2
  }
]
```

3.18.5. Create New Devices in a Group

Method	URI	Description
POST	/transport-controller-groups/v2/{transportControllerGroupName}	Creates devices in a group using the following schema: transportControllerEndpoint.json#/definitions/transport-ControllerGroup

The request contains a list of devices to be created.

Normal response codes: 201

3.18.5.1. Request

This table shows the URI parameters for the create new devices in a group request:

Name	Type	Description
{transportController-GroupName}	String	The name of the transport controller devices group.

Example 3.106. Create New Devices in a Group: JSON request

```
[
  {
    "accessMethod": "HTTP",
    "authMethod": "BASIC",
    "hostName": "",
    "httpPort": 80,
    "ipAddr": "10.0.1.2",
    "login": "Login",
    "passwd" : "XYXYXYXYXYX",
    "retry": 3,
    "timeout": 2
  },
  {
    "accessMethod": "HTTP",
    "authMethod": "BASIC",
    "hostName": "",
    "httpPort": 80,
    "ipAddr": "10.0.1.3",
    "login": "Login2",
    "passwd" : "XYXYXYXYXYX",
    "retry": 3,
    "timeout": 2
  }
]
```

3.18.5.2. Response

Example 3.107. Create New Devices in a Group: JSON response

```
{
  "success": true
}
```

3.18.6. Update Devices in the Group

Method	URI	Description
PUT	/transport-controller-groups/v2/{transportControllerGroupName}	Updates devices in a group using the following schema: transportControllerEndpoint.json#/definitions/transport-ControllerGroup

The request contains a list of devices to be updated. The id parameter is required.

Normal response codes: 201

3.18.6.1. Request

This table shows the URI parameters for the update devices in the group request:

Name	Type	Description
{transportController-GroupName}	String	The name of the transport controller devices group.

Example 3.108. Update Devices in the Group: JSON request

```
[
  {
    "id": "91172e83-7b45-422c-ac1e-1940fb9a7ddf",
    "accessMethod": "HTTP",
    "authMethod": "BASIC",
    "hostName": "",
    "httpPort": 80,
    "ipAddr": "10.0.1.3",
    "login": "Login3",
    "passwd": "",
    "retry": 3,
    "timeout": 2
  }
]
```

3.18.6.2. Response

Example 3.109. Update Devices in the Group: JSON response

```
{
  "success": true
}
```

3.18.7. Delete Devices in the Group

Method	URI	Description
DELETE	/transport-controller-groups/v2/{transportControllerGroupName}	Deletes devices from a group using the following schema: transportControllerEndpoint.json#/definitions/transport-ControllerGroup

The request contains a list of devices to be deleted. The id parameter is required, while other parameters are ignored.

Normal response codes: 201

3.18.7.1. Request

This table shows the URI parameters for the delete devices in the group request:

Name	Type	Description
{transportController-GroupName}	String	The name of the transport controller devices group.

Example 3.110. Delete Devices in the Group: JSON request

```
[
  {
    "id": "91172e83-7b45-422c-ac1e-1940fb9a7ddf"
  }
]
```

3.18.7.2. Response

Example 3.111. Delete Devices in the Group: JSON response

```
{
  "success": true
}
```

3.19. Path Computation API

The path computation APIs enable you to compute a path on the TE topology without creating an LSP. It supports the TE-LSP API constraints. The LSP is assumed to be an RSVP LSP by default. The SR-TE path is supported by using the provisioningType parameter, as described in [pathComputation.json](#), and illustrated in the following example:

NOTE: The path computation API for SR-TE provides only the path and does not provide the SIDs.

Method	URI	Description
POST	/topology/v2/{topology_id}/path-Computation	Use the following schema to request for a path computation: pathComputation.json#/definitions/pathComputationRequests . Response: pathComputation.json#/definitions/pathComputationResponses .

3.19.1. computePath

Method	URI	Description
POST	/topology/v2/{topology_id}/path-Computation	Use the following schema to request for a path computation: pathComputation.json#/definitions/pathComputationRequests . Response: pathComputation.json#/definitions/pathComputationResponses .

The request/response example illustrates a set of three path computation requests. The first request is successful. The two subsequent requests cannot be computed due to routing constraints because there is no path with fewer than two hops and no path that can carry 100G in the topology. All of the paths in the request list are computed in the order they are specified. This API does NOT re-shuffle the request order to optimize the solution or the constraints. No bandwidth is reserved by the path computation request. For example, if only 5G is available between two nodes, one path computation requesting two paths with 3G each will return a partial result, where only the first path is calculated.

Normal response codes: 201

3.19.1.1. Request

This table shows the URI parameters for the computepath request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 3.112. : JSON request

```
{
  "requests" :
  [
    {
      "from": {"topoObjectType": "ipv4", "address": "11.0.0.101"},
      "to" : {"topoObjectType": "ipv4", "address": "11.0.0.104"},
      "bandwidth" : 1000000, "design" : {"maxDelay": 5}, "extra" : "allowed"
    },
    {
      "from": {"topoObjectType": "ipv4", "address": "11.0.0.104"},
      "to" : {"topoObjectType": "ipv4", "address": "11.0.0.101"},
      "bandwidth" : 1000000, "design" : {"maxHop": 2}
    },
    {
      "from": {"topoObjectType": "ipv4", "address": "11.0.0.102"},
      "to" : {"topoObjectType": "ipv4", "address": "11.0.0.103"},
      "bandwidth" : "100G"
    }
  ]
}
```

3.19.1.2. Response

Example 3.113. : JSON response

```
{
  "result": "partial",
  "responses": [
    {
      "from": {"topoObjectType": "ipv4", "address": "11.0.0.101"},
      "to" : {"topoObjectType": "ipv4", "address": "11.0.0.104"},

```

```

    "bandwidth" : 1000000, "design" : {"maxDelay":5}, "extra" : "allowed",
    "status": "success",
    "path": [
      {"topoObjectType": "ipv4", "address": "11.101.105.2"},
      {"topoObjectType": "ipv4", "address": "11.105.107.2"},
      {"topoObjectType": "ipv4", "address": "11.114.117.1"}
    ]
  },
  {"from": {"topoObjectType": "ipv4", "address": "11.0.0.104"},
    "to" : {"topoObjectType": "ipv4", "address": "11.0.0.101"},
    "bandwidth" : 1000000, "design" : {"maxHop":2},
    "status": "noPathAvailable"
  },
  {"from": {"topoObjectType": "ipv4", "address": "11.0.0.102"},
    "to" : {"topoObjectType": "ipv4", "address": "11.0.0.103"},
    "bandwidth" : "100G",
    "status": "noPathAvailable"
  }
]
}

```

3.20. Maintenance API

Use these endpoints to work with maintenance.

The corresponding schema is: [maintenance.json](#) .

Method	URI	Description
GET	/topology/v2/{topology_id}/maintenances	Returns a full list of Maintenance.
POST	/topology/v2/{topology_id}/maintenances	Create a new maintenance using the following schema: maintenance.json#/definitions/createMaintenance .
GET	/topology/v2/{topology_id}/maintenances/stream	See EventSource for format. The notifications send on that stream are only maintenanceEvent. The data will contain a JSON document (see Paragon Pathfinder Notification API).
GET	/topology/v2/{topology_id}/maintenances/{maintenanceIndex}	Returns the details of a specified maintenance.
PUT	/topology/v2/{topology_id}/maintenances/{maintenanceIndex}	Modify a specific maintenance.
DELETE	/topology/v2/{topology_id}/maintenances/{maintenanceIndex}	Delete a specific maintenance.

3.20.1. Get Maintenance

Method	URI	Description
GET	/topology/v2/{topology_id}/maintenances	Returns a full list of Maintenance.

Normal response codes: 200

3.20.1.1. Request

This table shows the URI parameters for the get maintenance request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.20.1.2. Response

Example 3.114. Get Maintenance: JSON response

```
[
  {
    "topoObjectType": "maintenance",
    "topologyIndex": 1,
    "maintenanceIndex": 1,
    "user": "admin",
    "name": "1LinkMaint1",
    "status": "planned",
    "startTime": "2018-04-20T17:00:00Z",
    "endTime": "2018-04-20T17:30:00Z",
    "elements": [
      {
        "topoObjectType": "link",
        "index": 8,
        "id": "L11.105.107.1_11.105.107.2"
      }
    ]
  },
  {
    "topoObjectType": "maintenance",
    "topologyIndex": 1,
    "maintenanceIndex": 2,
    "user": "admin",
    "name": "1LinkMaint",
    "status": "planned",
    "startTime": "2018-04-20T17:00:00Z",
    "endTime": "2018-04-20T17:30:00Z",
    "elements": [
      {
        "topoObjectType": "link",
        "index": 8,
        "id": "L11.105.107.1_11.105.107.2"
      }
    ]
  }
]
```

```
    ]  
  }  
]
```

3.20.2. Create a new maintenance

Method	URI	Description
POST	/topology/v2/{topology_id}/maintenances	Create a new maintenance using the following schema: maintenance.json#/definitions/createMaintenance .

Normal response codes: 200

3.20.2.1. Request

This table shows the URI parameters for the create a new maintenance request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

Example 3.115. Create a new maintenance: JSON request

```
{
  "topoObjectType": "maintenance",
  "topologyIndex": 1,
  "startTime": "20180220T120700",
  "endTime": "20180220T120700",
  "name": "2LinksMaint",
  "elements": [
    {
      "topoObjectType": "link",
      "index": 1
    },
    {
      "topoObjectType": "link",
      "index": 3
    }
  ]
}
```

3.20.2.2. Response

Example 3.116. Create a new maintenance: JSON response

```
{
  "topoObjectType": "maintenance",
  "topologyIndex": 1,
  "maintenanceIndex": 3,
  "user": "admin",
  "name": "2LinksMaint",
  "status": "planned",
  "startTime": "2018-04-20T16:07:00Z",
  "endTime": "2018-04-20T16:15:00Z",
  "elements": [
    {
      "topoObjectType": "link",
      "index": 1,
      "id": "L11.101.105.1_11.101.105.2"
    }
  ],
}
```

```
    {  
      "topoObjectType": "link",  
      "index": 3,  
      "id": "L11.102.105.1_11.102.105.2"  
    }  
  ]  
}
```

3.20.3. Start a SSE Stream

Method	URI	Description
GET	/topology/v2/{topology_id}/maintenances/stream	See EventSource for format. The notifications send on that stream are only maintenanceEvent. The data will contain a JSON document (see Paragon PathFinder Notification API).

Normal response codes: 200

3.20.3.1. Request

This table shows the URI parameters for the start a sse stream request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.

This operation does not accept a request body.

3.20.4. Get a Specific Maintenance

Method	URI	Description
GET	/topology/v2/{topology_id}/maintenances/{maintenanceIndex}	Returns the details of a specified maintenance.

Normal response codes: 200

3.20.4.1. Request

This table shows the URI parameters for the get a specific maintenance request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{maintenanceIndex}	Int	The unique maintenanceIndex.

This operation does not accept a request body.

3.20.4.2. Response

Example 3.117. Get a Specific Maintenance: JSON response

```
{
  "topoObjectType": "maintenance",
  "topologyIndex": 1,
  "maintenanceIndex": 2,
  "user": "admin",
  "name": "1LinkMaint",
  "status": "planned",
  "startTime": "2018-04-20T17:00:00Z",
  "endTime": "2018-04-20T17:30:00Z",
  "elements": [
    {
      "topoObjectType": "link",
      "index": 8,
      "id": "L11.105.107.1_11.105.107.2"
    }
  ]
}
```


3.20.5. Update a Specific Maintenance

Method	URI	Description
PUT	/topology/v2/{topology_id}/maintenances/{maintenanceIndex}	Modify a specific maintenance.

Normal response codes: 200

3.20.5.1. Request

This table shows the URI parameters for the update a specific maintenance request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{maintenanceIndex}	Int	The unique maintenanceIndex.

Example 3.118. Update a Specific Maintenance: JSON request

```
{
  "topoObjectType": "maintenance",
  "topologyIndex": 1,
  "startTime": "20180408T154500",
  "endTime": "20180408T155000",
  "maintenanceIndex": 2,
  "name": "1LinkMaint",
  "elements": [
    {
      "topoObjectType": "link",
      "index": 1
    },
    {
      "topoObjectType": "link",
      "index": 2
    }
  ]
}
```

3.20.5.2. Response

Example 3.119. Update a Specific Maintenance: JSON response

```
{
  "topoObjectType": "maintenance",
  "topologyIndex": 1,
  "maintenanceIndex": 2,
  "user": "admin",
  "name": "1LinkMaint",
  "status": "planned",
  "startTime": "2018-04-08T20:45:00Z",
  "endTime": "2018-04-08T20:50:00Z",
  "elements": [
    {
      "topoObjectType": "link",
      "index": 1,

```

```
        "id": "L11.101.105.1_11.101.105.2"
      },
      {
        "topoObjectType": "link",
        "index": 2,
        "id": "L11.102.105.1_11.102.105.2"
      }
    ]
  }
```

3.20.6. Delete a Specific Maintenance

Method	URI	Description
DELETE	/topology/v2/{topology_id}/maintenances/{maintenanceIndex}	Delete a specific maintenance.

3.20.6.1. Request

This table shows the URI parameters for the delete a specific maintenance request:

Name	Type	Description
{topologyId}	Int	A unique identifier for the topology. In Paragon PathFinder version 2, the unique identifier is set to 1.
{maintenanceIndex}	Int	The unique maintenanceIndex.

This operation does not accept a request body.

3.21. RPCs

Remote Procedure Call (RPC) REST APIs allow subroutines that are written for Paragon PathFinder Planner to be utilized via the REST API. For instance Diverse P2MP Tree Design and Bandwidth Calendaring Simulation functionality can be called via these RPC type of REST APIs.

The corresponding schema is: [rpcs.json](#) .

3.22. Path Optimization RPC

These REST APIs are used to perform global or targeted path optimizations runs. The REST POST without a request body is used to call the global path optimization and with a request body to call the targeted path optimization functionalities. The REST GET is used to return the last global path optimization result that shows the difference between the previous vs. the optimized results. The REST GET doesn't support for targeted path optimization.

Method	URI	Description
POST	/topology/v2/rpc/optimize	Trigger global path optimization.
GET	/topology/v2/rpc/optimize	Get last global path optimization results.
POST	/topology/v2/rpc/optimize	Trigger targeted path optimization.

3.22.1. global_Optimize.post

Method	URI	Description
POST	/topology/v2/rpc/optimize	Trigger global path optimization.

The request body must be empty. The response returned by the server conforms to: [rpcs.json#/definitions/optimizationResponse](#).

Normal response codes: 201

3.22.1.1. Request

This operation does not accept a request body.

3.22.1.2. Response

Example 3.120. : JSON response

```
{
  "result ": "Path optimization started."
}
```

3.22.2. global_Optimize.get

Method	URI	Description
GET	/topology/v2/rpc/optimize	Get last global path optimization results.

The response returned by the server conforms to: [rpcs.json#/definitions/optimizationStatus](#).

Normal response codes: 201

3.22.2.1. Request

This operation does not accept a request body.

3.22.2.2. Response

Example 3.121. : JSON response

```
{
  "previous": {
    "bandwidth": 20000000,
    "hopCount": 98,
    "latency": 5
  },
  "optimized": {
    "bandwidth": 17000000,
    "hopCount": 85,
    "latency": 4
  },
  "numberOfOptimizedPaths": 6,
  "maxHopCount": 5,
  "maxLatency": 0.6,
  "tunnelCount": 21,
  "lastOptimizationTimeStamp": "2017-12-01T12:02:36.367Z"
}
```

3.22.3. targeted_Optimize.post

Method	URI	Description
POST	/topology/v2/rpc/optimize	Trigger targeted path optimization.

The request body must not be empty. The body consists of a list of elements. The elements can be either LSP, link or mixed. The elements are identified by the element index number. The response consists of a list of candidate elements to be optimized.

The response returned by the server conforms to: [rpcs.json#/definitions/optimizationResponse](#).

Normal response codes: 201

3.22.3.1. Request

Example 3.122. : JSON request

```
{
  "elements": [
    {
      "index": 1,
      "topoObjectType": "lsp"
    },
    {
      "index": 3,
      "topoObjectType": "lsp"
    },
    {
      "index": 47,
      "topoObjectType": "lsp"
    }
  ]
}
```

3.22.3.2. Response

Example 3.123. : JSON response

```
{
  "elements": [
    {
      "topoObjectType": "lsp",
      "index": 1
    },
    {
      "topoObjectType": "lsp",
      "index": 3
    }
  ]
}
```

3.23. Routing and Failure Simulation RPC

The routing and failure simulation APIs are used to simulate bandwidth calendaring (time based) and failure. Use the first REST POST function to perform routing simulation or failure

simulation. Use the subsequent three REST GET operations to return the reports to the calling program for further analysis.

NOTE: Starting from Northstar release 4.1, all the requests are timestamped.

Use the following API to simulate routes and failures:

Method	URI	Description
POST	/topology/v2/rpc/simulation	Perform failure simulation analysis.
GET	/topology/v2/rpc/simulation	Returns the simulation list.
GET	/topology/v2/rpc/simulation/{uuid}	Returns the details of a specified simulation.
GET	/topology/v2/rpc/simulation/{uuid}/{reportName}	Returns specific report details of a selected simulation.

3.23.1. Failure Simulation optimization RPC

Method	URI	Description
POST	/topology/v2/rpc/simulation	Perform failure simulation analysis.

The request body must follow the schema [rpcs.json#/definitions/simulationRequest](#) The response returned by the server conforms to: [rpcs.json#/definitions/simulationResponse](#) .

Normal response codes: 200

3.23.1.1. Request

Example 3.124. Failure Simulation optimization RPC: JSON request

The following example show different set of request:

- The following example shows maintenance simulation creation with exhaustive node, link and facility failure:

```
{
  "topologyIndex": 1,
  "elements":
  [
    "node",
    "link",
    "srlg",
    {
      "type": "maintenance",
      "maintenanceName": "1LinkMaint"
    }
  ]
}
```

- The following example shows maintenance simulation creation without exhaustive failure:

```
{
  "topologyIndex": 1,
  "elements":
  [
    {
      "type": "maintenance",
      "maintenanceName": "1LinkMaint"
    }
  ]
}
```

3.23.1.2. Response

Example 3.125. Failure Simulation optimization RPC: JSON response

The following example show different set of response based on the request above:

- The following example shows the response of maintenance simulation creation with exhaustive node, link and facility failure:


```
{
  "status": "success",
  "simulationId": "c203d890-24ee-4592-84b4-0975c7c0b79a",
  "results": {
    "links": [
      {
        "rel": "results",
        "href": "c203d890-24ee-4592-84b4-0975c7c0b79a"
      }
    ]
  },
  "topologyIndex": 1,
  "elements": [
    "node",
    "link",
    "srlg",
    {
      "type": "maintenance",
      "maintenanceName": "1LinkMaint"
    }
  ]
}
```

- The following example shows the response of maintenance simulation creation without exhaustive failure:

```
{
  "status": "success",
  "simulationId": "852c34b8-368d-4ae8-82d2-e9ed2dd8d223",
  "results": {
    "links": [
      {
        "rel": "results",
        "href": "852c34b8-368d-4ae8-82d2-e9ed2dd8d223"
      }
    ]
  },
  "topologyIndex": 1,
  "elements": [
    {
      "type": "maintenance",
      "maintenanceName": "1LinkMaint"
    }
  ]
}
```

3.23.2. Get Simulation optimization RPC list

Method	URI	Description
GET	/topology/v2/rpc/simulation	Returns the simulation list.

The response returned by the server conforms to: [rpcs.json#/definitions/simulationsList](#) .

Normal response codes: 200

3.23.2.1. Request

This operation does not accept a request body.

3.23.2.2. Response

Example 3.126. Get Simulation optimization RPC list: JSON response

```
{
  "topologyIndex": 1,
  "simulationReports": [
    {
      "status": "success",
      "simulationId": "c203d890-24ee-4592-84b4-0975c7c0b79a",
      "results": {
        "links": [
          {
            "rel": "results",
            "href": "c203d890-24ee-4592-84b4-0975c7c0b79a"
          }
        ]
      },
      "topologyIndex": 1,
      "elements": [
        "node",
        "link",
        "srlg",
        {
          "type": "maintenance",
          "maintenanceName": "1LinkMaint"
        }
      ]
    },
    {
      "simulationId": "20facea2-4ca6-4883-93da-0af22a0a4b75",
      "results": {
        "links": [
          {
            "rel": "results",
            "href": "20facea2-4ca6-4883-93da-0af22a0a4b75"
          }
        ]
      },
      "topologyIndex": 1,
      "elements": [
        {
          "type": "maintenance",
          "maintenanceName": "2LinkMaint"
        }
      ]
    }
  ]
}
```

```
}  
  ]  
  }  
  ]  
}
```

3.23.3. Get a specific simulation

Method	URI	Description
GET	/topology/v2/rpc/simulation/{uuid}	Returns the details of a specified simulation.

The response returned by the server conforms to: [rpcs.json#/definitions/simulationReports](#) .

Normal response codes: 200

3.23.3.1. Request

This table shows the URI parameters for the get a specific simulation request:

Name	Type	Description
{uuid}	String	rpc id.

This operation does not accept a request body.

3.23.3.2. Response

Example 3.127. Get a specific simulation: JSON response

```
{
  "status": "success",
  "simulationId": "c203d890-24ee-4592-84b4-0975c7c0b79a",
  "topologyIndex": 1,
  "elements": [
    "node",
    "link",
    "srlg",
    {
      "type": "maintenance",
      "maintenanceName": "1LinkMaint"
    }
  ],
  "reports": [
    {
      "reportName": "Report/L2_PathChange.r0",
      "links": [
        {
          "href": "Report/L2_PathChange.r0"
        }
      ]
    },
    {
      "reportName": "Report/L2_PATHDELAY.r0",
      "links": [
        {
          "href": "Report/L2_PATHDELAY.r0"
        }
      ]
    },
    {
      "reportName": "Report/L2_DVSIM.r0",
      "links": [
        {
```

```
        "href": "Report/L2_DVSIM.r0"
      }
    ],
  },
  {
    "reportName": "Report/L2_LinkUtilChange.r0",
    "links": [
      {
        "href": "Report/L2_LinkUtilChange.r0"
      }
    ]
  },
  {
    "reportName": "Report/L2_PeakSimLink.r0",
    "links": [
      {
        "href": "Report/L2_PeakSimLink.r0"
      }
    ]
  },
  {
    "reportName": "Report/L2_PeakSimRoute.r0",
    "links": [
      {
        "href": "Report/L2_PeakSimRoute.r0"
      }
    ]
  },
  {
    "reportName": "Report/L2_PHYDVSIM.r0",
    "links": [
      {
        "href": "Report/L2_PHYDVSIM.r0"
      }
    ]
  },
  {
    "reportName": "Report/Maintenance/maintsiminfo.1LinkMaint",
    "links": [
      {
        "href": "Report/Maintenance/maintsiminfo.1LinkMaint"
      }
    ]
  },
  {
    "reportName": "Report/L2_PeakSimSummary.r0",
    "links": [
      {
        "href": "Report/L2_PeakSimSummary.r0"
      }
    ]
  }
]
}
```

3.23.4. Get a specific report of one simulation

Method	URI	Description
GET	/topology/v2/rpc/simulation/{uuid}/{reportName}	Returns specific report details of a selected simulation.

The response returned by the server is a WANDL Report (Text file), see IPMPLSVIew report format documentation.

Normal response codes: 200

3.23.4.1. Request

This table shows the URI parameters for the get a specific report of one simulation request:

Name	Type	Description
{uuid}	String	rpc id.
{reportName}	String	report name.

This operation does not accept a request body.

3.23.4.2. Response

Example 3.128. Get a specific report of one simulation: application/textresponse

```
#####
#   LSP Path Changes
#####
##   Software Release= 4.0.0 20180327_72790_157, 64 bits,  Compilation Date=
    20180327
##   Customer= JNPR_Hanita-test
##   Platform=x86_64, OS=Linux 2.6.32-696.1.1.el6.x86_64
##   Report Date= 2018-3-28 14:44   Runcode=r0   User=pcs
Name,Node A,Node Z,Orig Hop Count,New Hop Count,Orig Path Cost,New Path Cost,
Orig Path,New Path,Protection,Orig Delay,New Delay,Delay Change %,Orig BW,New
BW,Type
Silver-101-103,vmx101,vmx103,3,4,30,60,11.101.105.2-11.105.107.2-11.103.107.1,
11.101.105.2-11.105.106.2-11.106.107.2-11.103.107.1,, 0.00, 0.00, 0.00%,0,
0,,
Silver-104-102,vmx104,vmx102,3,4,30,60,11.104.107.2-11.105.107.1-11.102.105.1,
11.104.107.2-11.106.107.1-11.105.106.1-11.102.105.1,, 0.00, 0.00, 0.00%,0,
0,,
rsvp-103-105,vmx103,vmx105,2,3,20,50,11.103.107.2-11.105.107.1,11.103.107.
2-11.106.107.1-11.105.106.1,, 0.00, 0.00, 0.00%,0,0,,
Silver-103-102,vmx103,vmx102,3,4,30,60,11.103.107.2-11.105.107.1-11.102.105.1,
11.103.107.2-11.106.107.1-11.105.106.1-11.102.105.1,, 0.00, 0.00, 0.00%,0,
0,,
11.0.0.103:11.0.0.106:200:vpls:vpn_200,vmx106,vmx103,3,2,30,40,11.105.106.
1-11.105.107.2-11.103.107.1,11.106.107.2-11.103.107.1,, 0.00, 0.00, 0.00%,
0,0,,
11.0.0.101:11.0.0.103:200:vpls:vpn_200,vmx103,vmx101,3,4,30,60,11.103.107.
2-11.105.107.1-11.101.105.1,11.103.107.2-11.106.107.1-11.105.106.1-11.101.105.
1,, 0.00, 0.00, 0.00%,0,0,,
rsvp-105-106,vmx105,vmx106,3,1,70,10,11.105.107.2-11.104.107.1-11.104.106.2,
11.105.106.2,, 0.00, 0.00, 0.00%,0,0,,
```

```

rsvp-107-105,vmx107,vmx105,1,2,10,40,11.105.107.1,11.106.107.1-11.105.106.1,,
0.00,0.00,0.00%,0,0,,
11.0.0.104:11.0.0.101:300:vp1s:vpn_200,vmx101,vmx104,3,4,30,60,11.101.105.
2-11.105.107.2-11.104.107.1,11.101.105.2-11.105.106.2-11.106.107.2-11.104.107.
1,,0.00,0.00,0.00%,0,0,,
tcg1tree-102107,vmx102,vmx107,2,-,20,-,11.102.105.2-11.105.107.2,#
vmx102--11.0.0.105--11.0.0.107,(path down),0.00,-,10.000K,10.000K,,
11.0.0.106:11.0.0.103:200:vp1s:vpn_200,vmx103,vmx106,3,2,30,40,11.103.107.
2-11.105.107.1-11.105.106.2,11.103.107.2-11.106.107.1,,0.00,0.00,0.00%,
0,0,,
11.0.0.101:11.0.0.104:300:vp1s:vpn_200,vmx104,vmx101,3,3,30,70,11.104.107.
2-11.105.107.1-11.101.105.1,11.104.106.2-11.105.106.1-11.101.105.1,,0.00,
0.00,0.00%,0,0,,
rsvp-106-105,vmx106,vmx105,3,1,70,10,11.104.106.1-11.104.107.2-11.105.107.1,
11.105.106.1,,0.00,0.00,0.00%,0,0,,
tcg1tree-102106,vmx102,vmx106,3,-,50,-,11.102.105.2-11.105.107.2-11.106.107.1,
# vmx102--11.0.0.105--11.0.0.107--11.0.0.106,(path down),0.00,-,10.000K,
10.000K,,
rsvp-104-105,vmx104,vmx105,2,3,20,50,11.104.107.2-11.105.107.1,11.104.107.
2-11.106.107.1-11.105.106.1,,0.00,0.00,0.00%,0,0,,
Silver-102-103,vmx102,vmx103,3,4,30,60,11.102.105.2-11.105.107.2-11.103.107.1,
11.102.105.2-11.105.106.2-11.106.107.2-11.103.107.1,,0.00,0.00,0.00%,0,
0,,
Silver-101-104,vmx101,vmx104,3,4,30,60,11.101.105.2-11.105.107.2-11.104.107.1,
11.101.105.2-11.105.106.2-11.106.107.2-11.104.107.1,,0.00,0.00,0.00%,0,
0,,
11.0.0.103:11.0.0.101:300:vp1s:vpn_200,vmx101,vmx103,3,4,30,60,11.101.105.
2-11.105.107.2-11.103.107.1,11.101.105.2-11.105.106.2-11.106.107.2-11.103.107.
1,,0.00,0.00,0.00%,0,0,,
Silver-103-101,vmx103,vmx101,3,4,30,60,11.103.107.2-11.105.107.1-11.101.105.1,
11.103.107.2-11.106.107.1-11.105.106.1-11.101.105.1,,0.00,0.00,0.00%,0,
0,,
Silver-104-101,vmx104,vmx101,3,4,30,60,11.104.107.2-11.105.107.1-11.101.105.1,
11.104.107.2-11.106.107.1-11.105.106.1-11.101.105.1,,0.00,0.00,0.00%,0,
0,,
tcg0tree-102104,vmx102,vmx104,3,-,30,-,11.102.105.2-11.105.107.2-11.104.107.1,
# vmx102--11.0.0.105--11.0.0.107--11.0.0.104,(path down),0.00,-,10.000K,
10.000K,,
11.0.0.106:11.0.0.104:300:vp1s:vpn_200,vmx104,vmx106,3,1,30,50,11.104.107.
2-11.105.107.1-11.105.106.2,11.104.106.2,,0.00,0.00,0.00%,0,0,,
11.0.0.104:11.0.0.106:200:vp1s:vpn_200,vmx106,vmx104,3,1,30,50,11.105.106.
1-11.105.107.2-11.104.107.1,11.104.106.1,,0.00,0.00,0.00%,0,0,,

```

3.24. P2MP Tree design RPC

These REST APIs are used for P2MP Tree Design Simulation. The first REST POST is utilized by the user to perform diverse p2mp tree design, given two trees in the same diversity group. The subsequent two REST GET operations are used to return the resulting list of P2MP trees. Since NS 5.0 the provisioningMethod (default NETCONF) can be specified and is used when useResultAsEro is set to true (The groups will be provisioned as a result of the design). The provisioningMethod must be set on the diverseTree request and will overwrite any set on the groups themselves. The fourth REST POST allows the user to perform diverse p2mp group using PCEP and associate it with a flow mapping. The POST REQUEST remains same for diverse p2mp group with or without flow mapping. Note, as the default provisioning method is NETCONF, we need to add additional field in the POST REQUEST ("creationConfigurationMethod": "PCEP"). This needs to be added globally as well as under "P2MPGroup", and "diverseP2MPGroup". Diverse p2mp with flow mapping is support-

ed since NorthStar 6.1.0. A non-diverse P2MP group with flow mapping is supported from NorthStar 5.1 version.

Following 2 statements are needed on the Junos router headend:

- set protocols pcep pce ns1 pce-traffic-steering.
- set routing-instances mvpn provider-tunnel external-controller pccd.

The tree design API does allow diverse P2MP trees to be computed. This is controlled using the following set of attributes:

- **diversityGroup**: the tree or leaves are grouped together using a diversityGroup. If TreeA and TreeB (LSP1 and LSP2) should be diverse to each other, they should be in the same diversityGroup (i.e the client should set the same value). Setting the diversityGroup on the Group will set it on all its leaves and Paragon PathFinder will pair the leaves based on their destination node/site.
- **diversityLevel** (site, srlg or link): the level of requested diversity. Paragon PathFinder will try to compute path that reach that diversity level, if this cannot be achieved, it will try to route the path as diverse as possible, unless minimumDiversityLevel is specified.
- **minimumDiversityLevel** (site, srlg or link): the minimum acceptable diversity level. If Paragon PathFinder cannot achieve diversity as good or better than minimumDiversityLevel, the routing is considered as failed. For DiverseTree design, this imply that the tree will not be provisioned if the minimumDiversityLevel cannot be achieved.

For instance a diverseTree request with diversityLevel=srlg and no minimumDiversityLevel may provision a tree with link diverse paths, while the same request with diversityLevel=minimumDiversityLevel=srlg will not provision the tree.

The Tree can also be terminated in CE or Site nodes, taking into account the PE to CE links for diversity constraints. The same restriction for P2MP groups detailed in [P2MP groups](#) applies.

Method	URI	Description
POST	/topology/v2/rpc/diverseTreeDesign	Create 2 P2MP trees and perform Diverse P2MP Tree Design.
GET	/topology/v2/rpc/diverseTreeDesign	Return the list of simulation optimization RPC list.
GET	/topology/v2/rpc/diverse-TreeDesign/{uuid}	Return the details of specified simulation optimization RPC result.
POST	/topology/v2/rpc/diverseTreeDesign	Create 2 P2MP trees and perform Diverse P2MP Tree Design with flow mapping.

3.24.1. P2MP Diverse Tree Design optimization RPC

Method	URI	Description
POST	/topology/v2/rpc/diverseTreeDesign	Create 2 P2MP trees and perform Diverse P2MP Tree Design.

The request body must follow the schema [rpcs.json#/definitions/diverseTreeRequest](#) The response returned by the server conforms to: [rpcs.json#/definitions/diverseTreeResult](#) .

Normal response codes: 200

3.24.1.1. Request

Example 3.129. P2MP Diverse Tree Design optimization RPC: JSON request

The following example show different set of request:

- The following example shows P2MP Diverse Tree Design optimization. Note that the values in the "name" field under "P2MPGroup" and under "diverseP2MPGroup" are required to be different from each other.

```
{
  "P2MPGroup": {
    "name": "hhtree",
    "from": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.102"
    },
    "plannedProperties": {
      "bandwidth": "99k",
      "design": {
        "diversityGroup": "hhzz"
      }
    },
    "lsps": [
      {
        "name": "hhtree-102104",
        "to": {
          "topoObjectType": "ipv4",
          "address": "11.0.0.104"
        }
      },
      {
        "name": "hhtree-102107",
        "to": {
          "topoObjectType": "ipv4",
          "address": "11.0.0.107"
        }
      }
    ]
  },
  "diverseP2MPGroup": {
    "name": "zztree",
    "from": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.102"
    },
  },
}
```

```

    "plannedProperties": {
      "bandwidth": "99k",
      "design": {
        "diversityGroup": "hhzz"
      }
    },
    "lsps": [
      {
        "name": "zztree-102104",
        "to": {
          "topoObjectType": "ipv4",
          "address": "11.0.0.104"
        }
      },
      {
        "name": "zztree-102107",
        "to": {
          "topoObjectType": "ipv4",
          "address": "11.0.0.107"
        }
      }
    ]
  },
  "ignoreExistingPath" : true,
  "useResultAsEro"      : true,
  "maxIterations" : 1000
}

```

The response is:

```

{
  "result": {
    "requestId": "fa87b57e-6edc-45a8-8b73-b06c79289f87",
    "status": "running",
    "maxIterations": 1000,
    "iterationsRun": 0
  }
}

```

- The following example shows P2MP Diverse Tree Design optimization using sites. Note that the values in the "name" field under "P2MPGroup" and under "diverseP2MPGroup" are required to be different from each other.

```

{
  "P2MPGroup":{
    "name":"tch0tree",
    "from":{
      "topoObjectType":"ipv4",
      "address":"11.0.0.101"
    },
    "plannedProperties":{
      "bandwidth":"10k",
      "design":{
        "diversityGroup":"h0h1"
      }
    }
  },
  "lsps":[
    {
      "name":"tch0tree-101104",
      "to":{

```

```

        "topoObjectType":"ipv4",
        "address":"11.0.0.104"
      },
    },
    {
      "name":"tch0tree-101107",
      "to":{
        "topoObjectType":"ipv4",
        "address":"11.0.0.107"
      }
    }
  ]
},
"diverseP2MPGroup":{
  "name":"tch1tree",
  "from":{
    "topoObjectType":"ipv4",
    "address":"11.0.0.102"
  },
  "plannedProperties":{
    "bandwidth":"10k",
    "design":{
      "diversityGroup":"h0h1"
    }
  },
},
"lsps":[
  {
    "name":"tch1tree-102106",
    "to":{
      "topoObjectType":"ipv4",
      "address":"11.0.0.106"
    }
  },
  {
    "name":"tch1tree-102107",
    "to":{
      "topoObjectType":"ipv4",
      "address":"11.0.0.107"
    }
  }
],
"ignoreExistingPath":true,
"useResultAsEro":true,
"maxIterations":1000,
"sites":[
  {
    "site":"site A",
    "nodes":[
      {"nodeIndex": 4}, {"nodeIndex": 6}
    ]
  }
]
}

```

The response is:

```

{
  "result": {
    "requestId": "f8c290ef-e850-45bb-9fd9-30089b095daa",
  }
}

```

```

        "status": "running",
        "maxIterations": 1000,
        "iterationsRun": 0
    }
}

```

- The following example shows P2MP Diverse Tree Design optimization with bandwidth calendaring. Note that the values in the "name" field under "P2MPGroup" and under "diverseP2MPGroup" are required to be different from each other.

```

{
  "P2MPGroup": {
    "name": "hhbwcaltree",
    "from": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.102"
    },
    "plannedProperties": {
      "bandwidth": "99k",
      "bandwidthCalendar": {
        "startTime": "20180430T0900",
        "endTime": "20180430T0905"
      },
      "design": {
        "diversityGroup": "hzbw"
      }
    },
    "lsp": [
      {
        "name": "hhbwcaltree-102104",
        "to": {
          "topoObjectType": "ipv4",
          "address": "11.0.0.104"
        }
      },
      {
        "name": "hhbwcaltree-102107",
        "to": {
          "topoObjectType": "ipv4",
          "address": "11.0.0.107"
        }
      }
    ]
  },
  "diverseP2MPGroup": {
    "name": "zzbwcaltree",
    "from": {
      "topoObjectType": "ipv4",
      "address": "11.0.0.102"
    },
    "plannedProperties": {
      "bandwidth": "99k",
      "bandwidthCalendar": {
        "startTime": "20180430T0900",
        "endTime": "20180430T0905"
      },
      "design": {
        "diversityGroup": "hzbw"
      }
    }
  }
}

```

```

    },
    "lsps": [
      {
        "name": "zzbwcaltree-102104",
        "to": {
          "topoObjectType": "ipv4",
          "address": "11.0.0.104"
        }
      },
      {
        "name": "zzbwcaltree-102107",
        "to": {
          "topoObjectType": "ipv4",
          "address": "11.0.0.107"
        }
      }
    ]
  },
  "ignoreExistingPath" : true,
  "useResultAsEro"      : true,
  "maxIterations" : 1000
}

```

The response is:

```

{
  "result": {
    "requestId": "f669b929-d82a-4bda-86bf-3ff623a794c9",
    "status": "running",
    "maxIterations": 1000,
    "iterationsRun": 0
  }
}

```

- The following example shows P2MP Diverse Tree Design optimization with minimum diversity level = site. Minimum diversity level options are site, srlg and link. Note that the values in the "name" field under "P2MPGroup" and under "diverseP2MPGroup" are required to be different from each other.

```

{
  "P2MPGroup":{
    "name":"tcb0pcepmindiv",
    "from":{
      "topoObjectType":"ipv4",
      "address":"11.0.0.101"
    },
    "plannedProperties":{
      "bandwidth":"10",
      "design":{
        "minimumDiversityLevel":"site",
        "diversityGroup":"tcb0tcb1pcepmindiv"
      }
    }
  },
  "lsps":[
    {
      "to":{
        "topoObjectType":"ipv4",
        "address":"11.0.0.104"
      }
    }
  ],
}

```

```

        {
          "to":{
            "topoObjectType":"ipv4",
            "address":"11.0.0.107"
          }
        }
      ],
    },
    "diverseP2MPGroup":{
      "name":"tcb1pcepmindiv",
      "from":{
        "topoObjectType":"ipv4",
        "address":"11.0.0.101"
      },
      "plannedProperties":{
        "bandwidth":"10",
        "design":{
          "minimumDiversityLevel":"site",
          "diversityGroup":"tcb0tcb1pcepmindiv"
        }
      },
    },
    "lsps":[
      {
        "to":{
          "topoObjectType":"ipv4",
          "address":"11.0.0.104"
        }
      },
      {
        "to":{
          "topoObjectType":"ipv4",
          "address":"11.0.0.107"
        }
      }
    ],
    "creationConfigurationMethod":"PCEP",
    "ignoreExistingPath":true,
    "useResultAsEro":true,
    "maxIterations":1000
  }
}

```

The response is:

```

{
  u'result':{
    u'status':u'running',
    u'maxIterations':1000,
    u'requestId':u'2862ddb5-2845-4971-8cd1-d8c9d56c281a',
    u'iterationsRun':0
  }
}

```

- The following example shows P2MP Diverse Tree Design optimization with CE and Site nodes as destination nodes. CE and Sites nodes are SiteA1, SiteZ and CEY2. Please note that currently combination of PE and CE/Site nodes as destinations nodes in a P2MP tree is not supported. Use can either create P2MP tree with all destination nodes consist of PE nodes only or consist of CE/Site nodes only. It cannot be combination of both (mixture of PE and CE/Sites nodes) in a single tree. Note that the values in the "name" field under

"P2MPGroup" and under "diverseP2MPGroup" are required to be different from each other.

```
{
  "P2MPGroup":{
    "name":"Atree",
    "from":{
      "topoObjectType":"ipv4",
      "address":"10.0.0.1"
    },
    "plannedProperties":{
      "bandwidth":"10k",
      "design":{
        "diversityGroup":"AB"
      }
    },
    "lsps":[
      {
        "to":{
          "topoObjectType":"ipv4",
          "address":"SiteA1"
        }
      },
      {
        "to":{
          "topoObjectType":"ipv4",
          "address":"SiteZ"
        }
      },
      {
        "to":{
          "topoObjectType":"ipv4",
          "address":"CEY2"
        }
      }
    ]
  },
  "diverseP2MPGroup":{
    "name":"Btree",
    "from":{
      "topoObjectType":"ipv4",
      "address":"10.0.0.1"
    },
    "plannedProperties":{
      "bandwidth":"10k",
      "design":{
        "diversityGroup":"AB"
      }
    },
    "lsps":[
      {
        "to":{
          "topoObjectType":"ipv4",
          "address":"SiteA1"
        }
      },
      {
        "to":{
          "topoObjectType":"ipv4",
          "address":"SiteZ"
        }
      }
    ]
  }
}
```

```
    },
    {
      "to": {
        "topoObjectType": "ipv4",
        "address": "CEY2"
      }
    }
  ],
  "creationConfigurationMethod": "PCEP",
  "ignoreExistingPath": true,
  "useResultAsEro": true,
  "maxIterations": 1000
}
```

The response is:

```
{
  "result": {
    "requestId": "4dde363c-abc0-471f-b76c-1b6df88bd9c9",
    "status": "running",
    "maxIterations": 1000,
    "iterationsRun": 0
  }
}
```

3.24.1.2. Response

The responses are shown per example in the request section. In general the response is a JSON document conforming to [rpcs.json#/definitions/diverseTreeResult](https://tools.ietf.org/html/draft-ietf-netconf-rpc-json-10#section-4).

This operation does not return a response body.

3.24.2. Get Simulation optimization RPC list

Method	URI	Description
GET	/topology/v2/rpc/diverseTreeDesign	Return the list of simulation optimization RPC list.

The response returned by the server conforms to: [rpcs.json#/definitions/diverseTreeResultList](#).

Normal response codes: 200

3.24.2.1. Request

This operation does not accept a request body.

3.24.2.2. Response

Example 3.130. Get Simulation optimization RPC list: JSON response

```
{
  "topologyIndex": "1",
  "rpcRequestIdList": [
    "f520e4e0-dcd7-4698-9575-6a2bcc8fd54b",
    "45eefe15-a668-4521-9d26-6159753fa948",
    "a9995c11-a5a7-4bf1-a0b1-dc9f65d38d19",
    "b0fa56f2-e178-4257-be47-ab12aab6043c",
    "cf168ccc-8bad-4c5d-8037-3dd5c6af7f53",
    "c4de3065-1b01-4a23-8aae-7a7860e5c4dc",
    "a39291c7-39de-4013-baab-8fa498c072fd",
    "69747e68-8a56-4b55-b36b-5167dd45d883",
    "248a2c33-f0c4-4e11-9275-8aa0aedad7df",
    "6dbe12b1-0cdc-4f85-8862-cf50aaa5b15b",
    "a27b2484-2181-457d-9424-1aa19cf6bcd0",
    "0f66bdf1-0346-49b8-956a-d6995446d861",
    "8fa7d9aa-8e42-4a44-bf93-f936df0ad084",
    "58f4da42-c20d-46b5-951d-bbdc80491da7"
  ]
}
```

3.24.3. Get a Specific Simulation optimization RPC Result

Method	URI	Description
GET	/topology/v2/rpc/diverse-TreeDesign/{uuid}	Return the details of specified simulation optimization RPC result.

The response returned by the server conforms to: [rpcs.json#/definitions/diverseTreeResult](#) .

Normal response codes: 200

3.24.3.1. Request

This table shows the URI parameters for the get a specific simulation optimization rpc result request:

Name	Type	Description
{uuid}	String	rpc id.

This operation does not accept a request body.

3.24.3.2. Response

Example 3.131. Get a Specific Simulation optimization RPC Result: JSON response

```
{
  "result": {
    "requestId": "f520e4e0-dcd7-4698-9575-6a2bcc8fd54b",
    "status": "partial",
    "P2MPGroup": {
      "name": "alphatree",
      "topoObjectType": "p2mpGroup",
      "from": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.101"
      },
      "lsps": [
        {
          "operationalStatus": "Unknown",
          "plannedProperties": {
            "bandwidth": "99K",
            "setupPriority": 7,
            "holdingPriority": 0,
            "calculatedEro": [
              {
                "topoObjectType": "ipv4",
                "address": "11.101.105.2",
                "loose": false
              },
              {
                "topoObjectType": "ipv4",
                "address": "11.105.106.2",
                "loose": false
              },
              {
                "topoObjectType": "ipv4",
                "address": "11.102.106.1",
                "loose": false
              }
            ]
          }
        }
      ]
    }
  }
}
```

```

    }
  ],
  "routingStatus": "Up",
  "pathName": "alphatree-101102_p0",
  "adminStatus": "Up",
  "design": {
    "diversityGroup": "TWINS"
  },
  "ero": [
    {
      "topoObjectType": "ipv4",
      "address": "11.101.105.2",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.105.106.2",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.102.106.1",
      "loose": false
    }
  ],
  "correlatedRR0HopCount": 3
},
"name": "alphatree-101102",
"from": {
  "topoObjectType": "ipv4",
  "address": "11.0.0.101"
},
"pathType": "primary",
"to": {
  "topoObjectType": "ipv4",
  "address": "11.0.0.102"
},
"lspIndex": 68,
"controlType": "PCC",
"provisioningType": "RSVP",
"p2mpName": "alphatree"
},
{
  "operationalStatus": "Unknown",
  "plannedProperties": {
    "bandwidth": "99K",
    "setupPriority": 7,
    "holdingPriority": 0,
    "calculatedEro": [
      {
        "topoObjectType": "ipv4",
        "address": "11.101.105.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.105.107.2",
        "loose": false
      }
    ]
  }
}

```

```

        "topoObjectType": "ipv4",
        "address": "11.103.107.1",
        "loose": false
    },
    ],
    "routingStatus": "Up",
    "pathName": "alpatree-101103_p0",
    "adminStatus": "Up",
    "design": {
        "diversityGroup": "TWINS"
    },
    "ero": [
        {
            "topoObjectType": "ipv4",
            "address": "11.101.105.2",
            "loose": false
        },
        {
            "topoObjectType": "ipv4",
            "address": "11.105.107.2",
            "loose": false
        },
        {
            "topoObjectType": "ipv4",
            "address": "11.103.107.1",
            "loose": false
        }
    ],
    "correlatedRROHopCount": 3
},
"name": "alpatree-101103",
"from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.101"
},
"pathType": "primary",
"to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.103"
},
"lspIndex": 69,
"controlType": "PCC",
"provisioningType": "RSVP",
"p2mpName": "alpatree"
},
{
    "operationalStatus": "Unknown",
    "plannedProperties": {
        "bandwidth": "99K",
        "setupPriority": 7,
        "holdingPriority": 0,
        "calculatedEro": [
            {
                "topoObjectType": "ipv4",
                "address": "11.101.105.2",
                "loose": false
            },
            {
                "topoObjectType": "ipv4",
                "address": "11.105.107.2",

```

```

        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.1",
        "loose": false
      }
    ],
    "routingStatus": "Up",
    "pathName": "alphatree-101104_p0",
    "adminStatus": "Up",
    "design": {
      "diversityGroup": "TWINS"
    },
    "ero": [
      {
        "topoObjectType": "ipv4",
        "address": "11.101.105.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.105.107.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.104.107.1",
        "loose": false
      }
    ],
    "correlatedRR0HopCount": 3
  },
  "name": "alphatree-101104",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.101"
  },
  "pathType": "primary",
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.104"
  },
  "lspIndex": 70,
  "controlType": "PCC",
  "provisioningType": "RSVP",
  "p2mpName": "alphatree"
}
]
},
"diverseP2MPGroup": {
  "name": "betatree",
  "topoObjectType": "p2mpGroup",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.102"
  },
  "lsps": [
    {
      "operationalStatus": "Unknown",

```

```
"plannedProperties": {
  "bandwidth": "99K",
  "setupPriority": 7,
  "holdingPriority": 0,
  "calculatedEro": [
    {
      "topoObjectType": "ipv4",
      "address": "11.102.105.2",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.101.105.1",
      "loose": false
    }
  ],
  "routingStatus": "Up",
  "pathName": "betatree-102101_p0",
  "adminStatus": "Up",
  "design": {
    "diversityGroup": "TWINS"
  },
  "ero": [
    {
      "topoObjectType": "ipv4",
      "address": "11.102.105.2",
      "loose": false
    },
    {
      "topoObjectType": "ipv4",
      "address": "11.101.105.1",
      "loose": false
    }
  ],
  "correlatedRR0HopCount": 2
},
"name": "betatree-102101",
"from": {
  "topoObjectType": "ipv4",
  "address": "11.0.0.102"
},
"pathType": "primary",
"to": {
  "topoObjectType": "ipv4",
  "address": "11.0.0.101"
},
"lspIndex": 65,
"controlType": "PCC",
"provisioningType": "RSVP",
"p2mpName": "betatree"
},
{
  "operationalStatus": "Unknown",
  "plannedProperties": {
    "bandwidth": "99K",
    "setupPriority": 7,
    "holdingPriority": 0,
    "calculatedEro": [
      {
        "topoObjectType": "ipv4",
```

```
        "address": "11.102.106.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.106.107.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.103.107.1",
        "loose": false
      }
    ],
    "routingStatus": "Up",
    "pathName": "betatree-102103_p0",
    "adminStatus": "Up",
    "design": {
      "diversityGroup": "TWINS"
    },
    "ero": [
      {
        "topoObjectType": "ipv4",
        "address": "11.102.106.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.106.107.2",
        "loose": false
      },
      {
        "topoObjectType": "ipv4",
        "address": "11.103.107.1",
        "loose": false
      }
    ],
    "correlatedRROHopCount": 3
  },
  "name": "betatree-102103",
  "from": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.102"
  },
  "pathType": "primary",
  "to": {
    "topoObjectType": "ipv4",
    "address": "11.0.0.103"
  },
  "lspIndex": 66,
  "controlType": "PCC",
  "provisioningType": "RSVP",
  "p2mpName": "betatree"
},
{
  "operationalStatus": "Unknown",
  "plannedProperties": {
    "bandwidth": "99K",
    "setupPriority": 7,
    "holdingPriority": 0,
```

```

        "calculatedEro": [
            {
                "topoObjectType": "ipv4",
                "address": "11.102.106.2",
                "loose": false
            },
            {
                "topoObjectType": "ipv4",
                "address": "11.104.106.1",
                "loose": false
            }
        ],
        "routingStatus": "Up",
        "pathName": "betatree-102104_p0",
        "adminStatus": "Up",
        "design": {
            "diversityGroup": "TWINS"
        },
        "ero": [
            {
                "topoObjectType": "ipv4",
                "address": "11.102.106.2",
                "loose": false
            },
            {
                "topoObjectType": "ipv4",
                "address": "11.104.106.1",
                "loose": false
            }
        ],
        "correlatedRROHopCount": 2
    },
    "name": "betatree-102104",
    "from": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.102"
    },
    "pathType": "primary",
    "to": {
        "topoObjectType": "ipv4",
        "address": "11.0.0.104"
    },
    "lspIndex": 67,
    "controlType": "PCC",
    "provisioningType": "RSVP",
    "p2mpName": "betatree"
}
]
}
},
"request": {
    "P2MPGroup": {
        "from": {
            "topoObjectType": "ipv4",
            "address": "11.0.0.101"
        },
        "name": "alphatree"
    },
    "diverseP2MPGroup": {
        "from": {

```



```
        "topoObjectType": "ipv4",  
        "address": "11.0.0.102"  
    },  
    "name": "betatree"  
},  
"ignoreExistingPath": true,  
"useResultAsEro": true  
}  
}
```

3.24.4. P2MP Diverse Tree Design optimization with flow mapping

Method	URI	Description
POST	/topology/v2/rpc/diverseTreeDesign	Create 2 P2MP trees and perform Diverse P2MP Tree Design with flow mapping.

The request body must follow the schema [rpcs.json#/definitions/diverseTreeRequest](#). The response returned by the server conforms to: [rpcs.json#/definitions/diverseTreeResult](#).

Normal response codes: 200

3.24.4.1. Request

Example 3.132. P2MP Diverse Tree Design optimization with flow mapping: JSON request

The following example shows Diverse P2MP Diverse Tree Design optimization with flow mapping. Note that the values in the "name" field under "P2MPGroup" and under "diverseP2MPGroup" are required to be different from each other.

```
{
  "P2MPGroup": {
    "creationConfigurationMethod": "PCEP",
    "name": "alphatree",
    "from": {"topoObjectType": "ipv4", "address": "11.0.0.105"},
    "plannedProperties": {
      "bandwidth": "99k",
      "flows": [
        {
          "topoObjectType": "flow",
          "vpn": {"topoObjectType": "vpnId", "rd": "11.0.0.105:65000"},
          "source": {"topoObjectType": "prefix", "address": "11.0.0.101", "length":
32},
          "group": {"topoObjectType": "prefix", "address": "239.2.2.2", "length":
32}
        }
      ],
      "design": {"diversityGroup": "Omega_diverse_group"}
    },
    "lsp": [
      {"to": {"topoObjectType": "ipv4", "address": "11.0.0.102"}},
      {"to": {"topoObjectType": "ipv4", "address": "11.0.0.106"}},
      {"to": {"topoObjectType": "ipv4", "address": "11.0.0.107"}}
    ]
  },
  "diverseP2MPGroup": {
    "creationConfigurationMethod": "PCEP",
    "name": "Betatree",
    "from": {"topoObjectType": "ipv4", "address": "11.0.0.105"},
    "plannedProperties": {
      "bandwidth": "99k",
      "flows": [
        {
```

```

    "topoObjectType": "flow",
    "vpn": {"topoObjectType": "vpnId", "rd": "11.0.0.105:65000"},
    "source": {"topoObjectType": "prefix", "address": "11.0.0.101", "length":
32},
    "group": {"topoObjectType": "prefix", "address": "239.3.3.3", "length":
32}
  }
],
  "design": {"diversityGroup": "Omega_diverse_group"}
},
  "lsps": [
    {"to": {"topoObjectType": "ipv4", "address": "11.0.0.102"}},
    {"to": {"topoObjectType": "ipv4", "address": "11.0.0.106"}},
    {"to": {"topoObjectType": "ipv4", "address": "11.0.0.107"}}
  ]
},
  "creationConfigurationMethod": "PCEP",
  "maxIterations": 1000
}

```

3.24.4.2. Response

Example 3.133. P2MP Diverse Tree Design optimization with flow mapping: JSON response

```

{
  "result": {
    "requestId": "6fc5ebe6-7763-4db7-a80b-0ca66e18ef66",
    "status": "running",
    "maxIterations": 1000,
    "iterationsRun": 0
  }
}

```

3.25. Notification examples

3.25.1. Node events

Example 3.134. Node update example

```

1
                                     {"action": "update", "notificationType": "node",
"object": {
    "topoObjectType": "node", "topologyIndex": 1, "id": "192.0.2.1",
"nodeIndex": 1, "layer": "IP", "AutonomousSystem": {"asNumber": 11},
    "protocols": {"ISIS": {
5      "routerId": "192.0.2.1",
        "TERouterId": "192.0.2.1",
        "isoAddress": "0110.0000.0101",
        "area": "490011"},
10      "PCEP": {"operationalStatus": "Up", "pccAddress": "192.0.2.1"}}}}}

```

Example 3.135. Node removal example

```

1
2      {"action": "remove", "notificationType":
"node", "object": {"nodeIndex": 1, "topologyIndex": 1, "id": "192.0.2.1",
"topoObjectType": "node"}}
3
4

```

3.25.2. Link events**Example 3.136. Link update example**

```

1
2      {"action": "remove", "notificationType": "link",
"object": {
3      "topoObjectType": "link", "topologyIndex": 1,
4      "linkIndex": 1, "id": "L192.0.2.13_192.0.2.14",
5      "endA": {
6      "topoObjectType": "interface", "TEcolor": 0, "TEmetric": 10,
"delay" : 600,
7      "bandwidth": 40000000000,
8      "node"      : {"id": "192.0.2.3", "topoObjectType": "node",
"topologyIndex": 1},
9      "ipv4Address": {"address": "192.0.2.13", "topoObjectType": "ipv4"},
10     "protocols": {"ISIS": { "TEmetric": 10, "level": "L2"},
11     "RSVP": {"bandwidth": 40000000000}},
12     "unreservedBw": [39970000896, 39970000896, 39970000896,
13     39469998080, 39469998080, 39469998080, 39469998080, 39469998080]
14     },
15     "endZ": {
16     "topoObjectType": "interface", "TEcolor": 0, "TEmetric": 10,
"delay" : 600,
17     "bandwidth": 40000000000,
18     "node"      : {"id": "192.0.2.7", "topoObjectType": "node",
"topologyIndex": 1},
19     "ipv4Address": {"address": "192.0.2.14", "topoObjectType": "ipv4"},
20     "protocols": {"ISIS": { "TEmetric": 10, "level": "L2"},
21     "RSVP": {"bandwidth": 40000000000}},
22     "unreservedBw": [39289999360, 38790000640, 38790000640,
23     38790000640, 38790000640, 38790000640, 38790000640, 38790000640]
24     },
25     "operationalStatus": "Up"}}

```

Example 3.137. Link removal example

```

1
2      {"action": "remove", "object": {"id": "192.
0.2.13", "topologyIndex": 1, "linkIndex": 1, "topoObjectType": "link"},
"notificationType": "link"},
3
4

```

3.25.3. TE-LSPs events

Example 3.138. LSP update example

```

1      {"action": "remove", "notificationType": "lsp",
"object":{
    "lspIndex": 2,
    "name": "LP_101_103",
    5    "from": {"address": "62.0.0.101", "topoObjectType": "ipv4"},
    "to": {"address": "62.0.0.103", "topoObjectType": "ipv4"},
    "controlType": "Delegated",
    "plannedProperties": {
    "bandwidth": "10M",
    10  "setupPriority": 7,
    "holdingPriority": 0,
    "calculatedEro": [
        {"topoObjectType": "ipv4", "address": "62.101.105.2", "loose": false},
        {"topoObjectType": "ipv4", "address": "62.102.105.1", "loose": false},
        15  {"topoObjectType": "ipv4", "address": "62.102.106.2", "loose": false},
        {"topoObjectType": "ipv4", "address": "62.104.106.1", "loose": false},
        {"topoObjectType": "ipv4", "address": "62.104.107.2", "loose": false},
        {"topoObjectType": "ipv4", "address": "62.103.107.1", "loose": false}
    ],
    20  "pathName": "Path_Node101_Node103_Strict_1",
    "adminStatus": "Up",
    "routingStatus": "Up",
    "lastStatusString": "<Active PCS initialization"
    },
    25  "liveProperties": {
        "adminStatus": "Up",
        "bandwidth": 10000000,
        "ero": [
            {"address": "62.101.105.2", "loose": false, "topoObjectType":
"ipv4" },
            30  {"address": "62.105.107.2", "loose": false, "topoObjectType":
"ipv4" },
            {"address": "62.103.107.1", "loose": false, "topoObjectType":
"ipv4" }
        ],
        "holdingPriority": 0,
        "metric": 40,
        35  "pathName": "Path_Node101_Node103_Strict_1",
        "rro": [
            {"address": "62.0.0.105", "protectionAvailable": true,
"protectionInUse": false, "topoObjectType": "ipv4"},
            {"address": "62.101.105.2", "protectionAvailable": true,
"protectionInUse": false, "topoObjectType": "ipv4"},
            {"address": "62.0.0.107", "protectionAvailable": false,
"protectionInUse": false, "topoObjectType": "ipv4"},
            40  {"address": "62.105.107.2", "protectionAvailable": false,
"protectionInUse": false, "topoObjectType": "ipv4"},
            {"address": "62.0.0.103", "protectionAvailable": false,
"protectionInUse": false, "topoObjectType": "ipv4"},
            {"address": "62.103.107.1", "protectionAvailable": false,
"protectionInUse": false, "topoObjectType": "ipv4"}
        ],
        "setupPriority": 7
    45  },

```

```

        "operationalStatus": "Active",
        "pathType": "primary",
        "tunnelId": 56614
    }}
50

```

Example 3.139. LSP removal example

```

1
2           {"action": "remove", "object": {"lspIndex": 74},
"notificationType": "lsp"},
3
4

```

3.25.4. P2MP Groups

Example 3.140. P2MP update example

The following example illustrates updating a P2MP group with a new LSP. The result is, in fact, two updates, one for the group and one for the newly added LSP. The results are similar when an LSP is removed from the group; the group is updated (or removed if no LSP remains), then the member itself is removed.

```

1
2           {"action": "update", "object": {"from":
{"address": "11.0.0.101", "topoObjectType": "ipv4"}, "name": "1-d",
"lsp": [{"lspIndex": 90}, {"lspIndex": 91}], "p2mpIndex": 184549477,
"topoObjectType": "p2mpGroup", "p2mpGroupIndex": 8}, "notificationType":
"p2mpGroup"},
3
4

```

```

1
2           {"action": "add", "object": {"p2mpName": "1-
d", "from": {"address": "11.0.0.101", "topoObjectType": "ipv4"}, "name":
"p2mpMember-02", "p2mpIndex": 184549477, "liveProperties": {"adminStatus":
"Up", "erro": [{"address": "11.101.105.2", "protectionAvailable": false,
"protectionInUse": false, "topoObjectType": "ipv4"}, {"address": "11.105.107.
2", "protectionAvailable": false, "protectionInUse": false, "topoObjectType":
"ipv4"}, {"address": "11.104.107.1", "protectionAvailable": false,
"protectionInUse": false, "topoObjectType": "ipv4"}], "setupPriority": 7,
"metric": 0, "bandwidth": 0, "holdingPriority": 0, "ero": [], "pathType":
"primary", "to": {"address": "11.0.0.104", "topoObjectType": "ipv4"},
"controller": "External", "operationalStatus": "Up", "controlType": "PCC",
"lspIndex": 91}, "notificationType": "lsp"},
3
4

```

Example 3.141. P2MP group removal example

When the Last LSP of the P2MP group is removed, the P2MP group is removed.

NOTE: LSP removal notification is described in the LSP notification examples.

```
1
2      {"action": "remove", "object": {"p2mpGroupIndex":
8, "topologyIndex": 1, "topoObjectType": "p2mpGroup"}, "notificationType":
"p2mpGroup"},
3
4
```

3.25.5. Node events

Example 3.142. Node update example

```
1
2      {"action": "update", "notificationType": "node",
"object": {
  "topoObjectType": "node", "topologyIndex": 1, "id": "192.0.2.1",
"nodeIndex": 1, "layer": "IP", "AutonomousSystem": {"asNumber": 11},
  "protocols": {"ISIS": {
5      "routerId": "192.0.2.1",
        "TERouterId": "192.0.2.1",
        "isoAddress": "0110.0000.0101",
        "area": "490011"},
10     "PCEP": {"operationalStatus": "Up", "pccAddress": "192.0.2.1"}}}}

```

Example 3.143. Node removal example

```
1
2      {"action": "remove", "notificationType":
"node", "object": {"nodeIndex": 1, "topologyIndex": 1, "id": "192.0.2.1",
"topoObjectType": "node"}}
3
4
```

3.25.6. Topology events

Example 3.144. Topology reset example

```
1
2      {"action": "add", "object": {"topoObjectType" :
"topology", "topologyIndex" : 1}, "notificationType": "topology"}
5
```

3.25.7. High Availability events

Example 3.145. Node status update

```
1                                     {"action": "add", "notificationType": "host",  
"object" : {  
    "hostname" : "node1.northstar.example.net",  
    "status" : "Up",  
5    "role" : "Active"  
    }}  
}}
```