Maximize Bandwidth Utilization with Juniper Networks TE++

Lower Operations Overhead while Simplifying Network Planning

Table of Contents

Executive Summary

Demand for network bandwidth continues to rise, driven by a steady stream of new Internet users, network-connected devices, video and other bandwidth-intensive traffic, and services ranging from VPNs to cloud facilities and data center interconnection. For service providers, cloud operators, and enterprises alike, containing capital costs is a key concern.

Network operators are employing the traffic engineering (TE) capabilities of MPLS and other tools to help them use bandwidth more efficiently. The ability to engineer label-switched paths (LSPs) has made it easier for operators to address the requirements of video, data, and time-sensitive applications such as financial market updates, as well as meet service agreements. Multipath routing and auto-bandwidth capabilities have also helped operators use bandwidth more efficiently.

However, current MPLS-TE and auto-bandwidth solutions have limitations. Auto-bandwidth can dynamically adjust LSPs in response to fluctuating bandwidth demands and network events but, working in the context of a given path, it cannot extract every bit of bandwidth from all feasible combinations of network links.

Juniper Networks TE++ technology tackles the problem of network resource utilization and gives network operators a new tool for maximizing bandwidth. TE++ automatically sets up, distributes, and rebalances traffic from a given source across multiple equal-bandwidth LSPs in response to traffic changes and other network events.

By dynamically optimizing bandwidth utilization, Juniper TE++ helps customers maximize their infrastructure investments and lower capital expenses. By automating functions that are currently performed manually, Juniper TE++ lowers operations overhead and simplifies network planning and TE management.

Introduction: The Bandwidth Challenge

Ensuring that bandwidth is used efficiently has been an ongoing challenge for network operators. This challenge is tied to a classic problem known as “bin packing,” in which objects of different sizes or volumes must be packed into a finite number of bins or containers in a way that minimizes the number of bins used.

In the network context, the challenge is how to distribute traffic flows (considered as objects) along the available paths (considered as bins) in a way that makes the most efficient use of all network links. Inefficient bin packing leads to inefficient bandwidth utilization and increases the probability of blocking in the network, which can result in traffic loss.

In the context of MPLS, inefficient bin packing can occur for several reasons, including the order in which LSPs are signaled, the paths that ingress routers choose for LSPs, the failure of ingress routers to probe for available bandwidth, and a lack of coordination among routers.

What network operators want is an automated way to set up LSPs that use network bandwidth as efficiently as possible and dynamically respond to traffic changes and network events to continuously maximize bandwidth utilization.

The Bin Packing Problem in MPLS

One cause of inefficient bin packing is that each ingress router computes paths for the LSPs it originates without information about LSPs originated by other ingress routers. Consequently, each new LSP that’s set up must “work around” the LSPs already in place. Since TE LSPs are set up for an entire path, the bandwidth required for a TE LSP must be available end-to-end along that path in order for that LSP to be established.

Ingress devices may not be able to signal LSPs with their required bandwidth even though there is enough bandwidth in the network. The order in which ingress routers set up LSPs causes network capacity to be fragmented along different paths, such that a single LSP cannot fully utilize the capacity, resulting in some links being underutilized and overall inefficient bandwidth utilization.

Consider the sequence of LSPs in Table 1 below with reference to Figure 1.

Table 1: LSP Sequence

Time	Source	Destination	Demand	ERO
1	A	E	5	A, C, D, E
2	B	E	10	No-ERO

If LSP (A-E) is signaled and placed first (ERO = A-C-D-E), then the LSP (B-E) cannot be placed. However, if LSP (B-E) is signaled and placed first (ERO = B-C-D-E), LSP (A-E) can also be signaled (ERO = A-C-E).

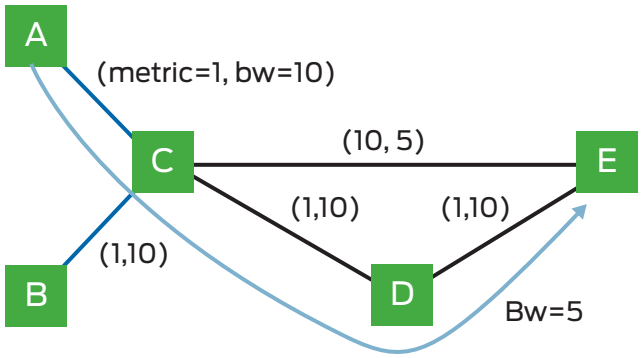

Figure 1: Each link is annotated with a tuple (cost, available bandwidth). There is an LSP between A and E requiring 5 MB of bandwidth and another one between B and E requiring 10 MB of bandwidth.

Inefficient bin packing can happen even if the order in which LSPs are set up isn't an issue. As shown in Figure 2, there is an LSP (with ingress at A and egress as E) that needs 20 MB of bandwidth. There is no single path with 20 MB of bandwidth available, so this LSP won't be established—even though the network has sufficient resources to satisfy an aggregate demand of 20 MB. This can be characterized as the "all-or-nothing" problem.

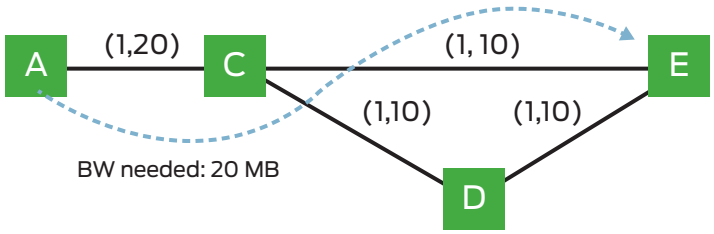

Figure 2: The all-or-nothing problem

It is a difficult problem for distributed ingress routers to optimally establish TE LSPs such that all the links are highly utilized or some other metric is optimized. Key reasons for bin packing inefficiency include:

- **All-or-nothing policy**: RSVP enables routers to determine the bandwidth available on a path and reserve it. However, when a traffic flow exceeds the bandwidth available on any given path, MPLS does not provide a mechanism for splitting that traffic across two or more paths whose combined capacity can accommodate that traffic flow. Unless an ingress router can find a path that provides the full amount of bandwidth requested, it won't establish an LSP. Because of this "all-or-nothing" behavior, the router must find a path with the suitable amount of bandwidth before it will set up an LSP.
- **Local ordering for CSPF**: The Constrained Shortest Path First (CSPF) scheduling at ingress is often based on certain LSP attributes, such as highest or lowest bandwidth first or highest priority first. This scheduling may not result in optimal placement of LSPs because network conditions keep changing. Furthermore, in an attempt to find a path that satisfies the required bandwidth, CSPF may find a path that has a higher metric or hop count than is ideal. Moving a traffic flow off the shortest path can affect application performance as well as consume bandwidth in more links than necessary.
- **Lack of global coordination**: In the absence of global coordination across all ingress routers, ingress devices cannot dynamically establish, break, and reestablish LSPs to continue to meet bandwidth and other constraints. Rather, ingress routers only react to failures, such as a preemption indication, RSVP signaling failure, or link failure.

Auto-Bandwidth and Equal-Cost Multipath Limitations

Network engineers and operators have been trying to solve the bandwidth inefficiency problem using MPLS RSVP-TE in conjunction with several other technologies, primarily auto-bandwidth mechanisms and multipath routing. Each has its benefits and drawbacks, but neither adequately addresses the bin packing problem.

Auto-Bandwidth Limitations

Due to the high costs involved, network operators want to avoid having to provision their networks for worst-case traffic demands. Rather, they want TE LSPs to reserve bandwidth to meet current traffic demands (along with other CSPF constraints), and they also want to control the maximum and minimum reserved bandwidth for such TE LSPs.

MPLS RSVP-TE supports a bandwidth adjustment feature that automatically adjusts the bandwidth allocation for TE LSPs based on their measured traffic load. This auto-bandwidth feature allows network operators to implement TE without knowing the traffic load and patterns in advance. Many organizations use auto-bandwidth because it can automatically accommodate changes in network traffic, providing greater flexibility and efficiency.

Vendors typically implement this auto-bandwidth capability so that it periodically adjusts an LSP's allocated bandwidth based on traffic samples; that is, auto-bandwidth estimates the current traffic rate from recent traffic samples collected by ingress routers at preconfigured intervals. Network operators can configure the sampling interval on a per-LSP basis as well as define the types of adjustments that are permitted. Ingress routers then use the auto-bandwidth feature to re-optimize the TE LSPs with updated bandwidth in a make-before-break fashion.

Auto-bandwidth lets TE LSPs reserve bandwidth according to actual usage, enabling the ingress device to use overflow or underflow parameters, for example, to adapt quickly to changing traffic. While auto-bandwidth has greatly simplified bandwidth management for TE LSPs, it does not solve the bin packing problem and has several drawbacks:

- **Properly setting the auto-bandwidth adjustment threshold is tricky**: The auto-bandwidth adjustment threshold dictates when, and if, LSPs need to be re-optimized and resized. If the threshold value is small, LSPs will be re-optimized and rerouted frequently, because a small change in bandwidth will be enough to cross threshold limits. LSPs may reroute even though there is enough bandwidth. Frequent re-optimization can cause router CPU usage to spike as the Routing Engine (RE) performs next-hop resolution for applications or protocols such as BGP.

  A large threshold value can make LSPs less responsive to traffic surges. When an LSP does respond, it may have to bridge a large gap between its current reservation and the actual current usage, which can result in the "all-or-nothing" problem. That is, the LSP's bandwidth requirements are so high that few paths in the network can satisfy them, so this LSP may not be set up at all. In the absence of a new path being found that can meet the higher bandwidth demand, the existing LSP continues to accept and forward traffic, resulting in an overloading of links and traffic being dropped.
- **Auto-bandwidth adjustments can lead to higher latency paths:** In an attempt to find a path that satisfies the required bandwidth, CSPF can select a path that has a higher metric or hop count than is ideal for that traffic flow. Moving a traffic flow off the shortest path can affect application performance as well as consume bandwidth in more links than necessary.

  Microsoft researchers[1] studied the latency on dozens of data centers connected with RSVP LSPs configured with auto-bandwidth. A variety of application traffic, from search engine queries to distributed storage, traversed the LSPs. The researchers found that auto-bandwidth and other LSP parameters were responsible for latency inflation. In particular, periodic optimization by auto-bandwidth regularly resulted in LSPs being rerouted to a higher delay path. Some 22% of data center pairs experienced significant latency spikes, and roughly 20% of the LSPs had spikes greater than 20 ms.

Auto-bandwidth is an improvement over fully manual bandwidth allocation, enabling TE LSPs to reserve bandwidth automatically according to their needs. However, the LSPs that are established are still subject to the "all-or-nothing" problem.

Network operators can work around the shortcomings of auto-bandwidth by provisioning multiple LSPs between the same pair of routers. However, this requires extra provisioning efforts, which drives up operations expenses and complicates management. Ideally, ingress devices should be able to address an application's or service's bandwidth needs by splitting that traffic across several LSPs whose combined bandwidth satisfies the application's or service's request.

ECMP and Multipath Drawbacks

Equal-cost multipath (ECMP) is a routing strategy where next-hop packet forwarding to a single destination can occur over multiple "best paths" that tie for top place in routing metric calculations. Multipath routing can also be non-equal cost.

Network operators often use multipath routing as a way to deploy multiple TE LSPs between an ingress and destination pair so that any aggregate traffic surge can be absorbed across multiple parallel TE LSPs. It's not necessary for these multiple LSPs to have the same routing cost or metric assigned.

[1]ACM SIGCOMM conference, IMC'11 Nov 2-4 2011 (http://dl.acm.org/citation.cfm?id=2068859)

Although multipath routing doesn't solve the bin packing problem, it does give network operators greater flexibility in controlling LSP setup and it increases reliability. However, network operators must manually configure all of the LSPs on a multipath, which creates considerable management overhead. Network operators need to monitor traffic across these LSPs and then add/delete LSPs manually as conditions on the network change (for example, in response to a traffic surge). Manually changing the configuration to achieve the benefits of multipath ECMP is cumbersome, and can be impractical for very large networks and particularly dynamic environments.

Network staff often turn to external software or write specific scripts to help with multipath configuration. In addition, they may collect traffic profiles from ingress routers and run LSP optimization scenarios offline to try to determine whether, and how, a configuration should be changed to improve bandwidth efficiency. These manual processes, however, create operational overhead and management complexity.

Networks are dynamic in nature, which makes it difficult to maximize bandwidth efficiency. The current bandwidth optimization mechanisms available to network operators have benefits but don't fully eliminate the need for manual calculations and configuration. Nor do they make network planning easier.

Operators need a bandwidth management solution that's easy to deploy and configure, and that responds dynamically to changes in the network by automatically setting up and tearing down LSPs as needed. To be truly effective, such a solution must address the bin packing problem.

Introducing Juniper Networks TE++ Solution

Juniper Networks developed its TE++ technology to address the problem of bin packing inefficiency and customers' need for a dynamic, automated way to maximize bandwidth utilization. Embedded in Juniper Networks® Junos® operating system, Juniper's TE++ allows for the creation, deletion, and elastic sizing of LSPs based on actual traffic patterns. It also significantly improves network bandwidth utilization without the need for additional provisioning efforts, helping customers reduce both CapEx and OpEx.

In developing TE++, Juniper leveraged the idea of multipath LSPs (MP LSPs) in RSVP multipath extensions proposed in the IETF draft "Multipath Label Switched Paths Signaled Using RSVP-TE"*. These RSVP extensions allow for the traffic engineering of MP LSPs that conform to TE constraints but follow multiple independent paths from the source to the destination. MP LSPs have the characteristics of ECMP and RSVP TE.

Juniper implements the multipath aspect of the MP LSP in the local context, with a simple algorithm, avoiding the need for protocol extensions. Consequently, there is no need to upgrade protocols across network devices.

In developing TE++, Juniper has defined a new construct, called a "container" LSP, which consists of multiple, dynamically created, point-to-point RSVP-TE LSPs called "member" LSPs. When network operators provision a container LSP, TE++ automatically establishes the member LSPs between ingress and egress in response to network conditions. A container LSP, as well as its member LSPs, can be used by protocols and services, such as BGP and VPNs, just as they currently use point-to-point LSPs.

Juniper TE++ enables each member LSP to potentially take a different, equal-bandwidth path to the same destination. By computing several small bandwidth LSPs to meet an application's or service's bandwidth requirements, TE++ eliminates the "all-or-nothing" problem and ensures that all traffic flows can be accommodated. By load balancing across multiple paths, TE++ inherits the standard benefits of ECMP, including the ability to absorb traffic surges.

Juniper TE++ directly addresses the challenge of inefficient bin packing. By enabling a container LSP to create several smaller bandwidth member TE LSPs, an ingress device is more likely to find feasible paths satisfying both CSPF constraints and bandwidth requirements.

To handle changes in network conditions and bandwidth needs, TE++ uses "splitting" and "merging" processes. In a process called "splitting," new member LSPs are added and the exiting member LSPs are re-signaled with updated bandwidth in a make-before-break manner. In a process called "merging," one or more member LSPs are removed dynamically when the traffic demand drops significantly, making unused bandwidth available for other container LSPs.

How TE++ Works

Network operators use Juniper TE++ templates to define a container LSP between an ingress and egress router. Container LSPs then manage the member LSPs at the ingress router; transit and egress routers do not know if two member LSPs are part of a container LSP. Each LSP member is enabled with auto-bandwidth.

The ingress router has the responsibility of computing the aggregate bandwidth of a container LSP from the bandwidth samples of member LSPs. The aggregate bandwidth computed could be the average of all samples, a particular percentile-based value, or the maximum of all samples, depending on sampling mode configuration.

* Kompella, K., Hellers, M., "Multi-path Label Switched Paths Signaled Using RSVP-TE," July 2013

Through a process called normalization, TE++ dynamically adjusts the number of members and the per-member bandwidth in response to the traffic situation in the network. Every time normalization occurs, ingress routers compute aggregate bandwidth for a container LSP using bandwidth samples taken since the last normalization event. They further compute the number of member LSPs needed for the container LSP and the new equal bandwidth allocations for each member LSP, as described below.

Normalization can be triggered in various ways, including the expiration of a timer, in response to network or signaling failures, or in response to a change in aggregate bandwidth. Juniper uses many bandwidth samples, reflecting 98% of link behavior, to estimate bandwidth usage, ensuring that estimates are accurate and stable.

The normalization result is based on configuration, traffic profile, and network topology. Normalization parameters do not influence auto-bandwidth behavior; therefore, based on the traffic profile, auto-bandwidth can resize each member to different bandwidths between two normalization events.

If a normalization result indicates the need for more member LSPs, the ingress router will signal new members and/or re-signal existing members with a smaller bandwidth, a "splitting" process that subdivides an existing container LSP into more component LSPs, each with a smaller bandwidth than prior to splitting. If normalization finds that fewer LSPs are needed, the ingress will apply a "merging" process by removing excess members and/or re-signaling the retained members with a higher bandwidth to shoulder the cumulative bandwidth.

The per-member upper limit on bandwidth above which splitting LSPs into smaller ones should be considered is called "splitting bandwidth," and the per-member lower limit on bandwidth below which the LSPs should be considered for merging is called "merging bandwidth."

The minimum and maximum value for per-member bandwidth to be used for signaling can be the same as the merging and splitting bandwidth values, respectively. Alternately, the values can be a range that is a subset of what is defined by the merging and splitting bandwidth, providing room for LSPs to grow or shrink via auto-bandwidth procedures while staying within the limits of merging-splitting bandwidth range.

When triggered periodically, normalization results in a new value for the number of members and per-member bandwidth if:

- The per-member utilization goes beyond splitting bandwidth or falls below merging bandwidth.
- The aggregate utilization changes more than the configured threshold.

The ingress computes the number of LSPs to signal by dividing the aggregate demand by the maximum signaling bandwidth. The ingress' attempt to bring up an LSP starts with this number of members and maximum signaling bandwidth per member. If, after a normalization event, the ingress determines this setup hasn't been successful, TE++ increases the number of LSPs by one and derives the appropriate per-member bandwidth to meet the aggregate demand.

The success of bring up is reviewed periodically (every normalization-retry-interval) and, if necessary, the number of members is incremented, while taking care not to fall outside the range of the configured minimum/maximum number of members. The maximum capacity that a container LSP can offer is based on the maximum number of member LSPs allowed by configuration, each reserving maximum-signaling-bandwidth. If the aggregate bandwidth cannot be met by the calculated configuration value (maximum number of LSPs * maximum bandwidth per LSP), the event will be logged and the container will continue to offer the maximum it can within the configuration and ECMP limitations.

All adjustments to existing member LSPs are done in a make-before-break manner to prevent any traffic loss. When a container LSP comes up from a "cold start," the minimum number of member LSPs are brought up with each member reserving the minimum-signaling bandwidth since there is no aggregate traffic estimated.

If the ingress is unable to meet aggregate bandwidth demand even after employing splitting, it can choose to stay in the pre-normalization state if the normalization mode is non-incremental. Or, if the normalization mode is incremental, the ingress can try to improve the LSP set to the extent possible. In other words, the ingress can offer a bandwidth capacity greater than the pre-normalization state, even though it still might not meet the current aggregate demand.

TE++ in Action

Using the Juniper TE++ template, network operators can easily configure a container LSP and indicate what properties member LSPs should have, such as adaptive, link protection, admin-group, etc. In addition, each container LSP must be configured with the following:

1. Information related to splitting/merging behavior (creation and deletion of members):
    a. Maximum/minimum number of member LSPs in the container
    b. Maximum/minimum signaled bandwidth for each member LSP
    c. Splitting and merging bandwidth (by default is the same as max/min signaling bandwidth)

2. Normalization events—computation of LSPs needed:
   a. Normalization interval
   b. Normalization mode (incremental/non-incremental)
   c. Failover normalization (whether to reshuffle if one of the members fails)
3. Sampling:
   a. Sampling mode
   b. Sampling cutoff threshold (which outlier samples need to be discarded)

Figures 3 through 5 illustrate the splitting and merging processes of Juniper TE++. Minimum-signaling bandwidth and maximum-signaling bandwidth are two thresholds used during normalization to determine how many LSPs there should be and the bandwidth associated with each member. Let's assume the following configuration:

- Minimum signaling bandwidth: 2 Gbps
- Maximum signaling bandwidth: 8 Gbps
- Merging bandwidth: 2 Gbps
- Splitting bandwidth: 9 Gbps

In each of the figures below, dashed lines indicate the old instance of an LSP that is in the process of getting removed.

In Figure 3, a container LSP creates a minimum number (1) LSP (A-E-1) with a minimum signaling bandwidth of 2 Gbps. As traffic begins moving over the LSP, Ingress A will compute the aggregate bandwidth from the samples. Let's say the LSP grows to 7 Gbps owing to auto-bandwidth adjustment.

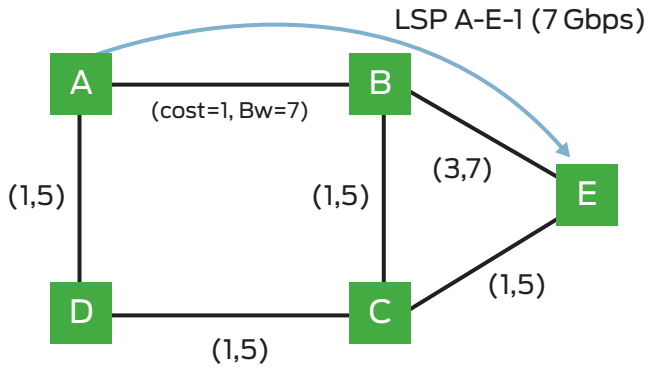

Figure 3: A container LSP signals a member LSP A-E-1 with adjusted bandwidth 7 Gbps along ERO (A-B-E).

In Figure 4, the LSP container traffic surges to 10 Gbps. Normalization could be triggered as a result of failure to reserve a 10 Gbps member LSP when failover normalization is configured, or it could be triggered due to normalization timer expiry. The computation would decide to split LSPs as the per-member utilization has crossed the splitting-bandwidth threshold of 9 Gbps. As a result of splitting, one new member LSP A-E-2 is created with 5 GB of bandwidth and the existing member LSP A-E-1 is re-signaled with 5 GB of bandwidth in a make-before-break way.

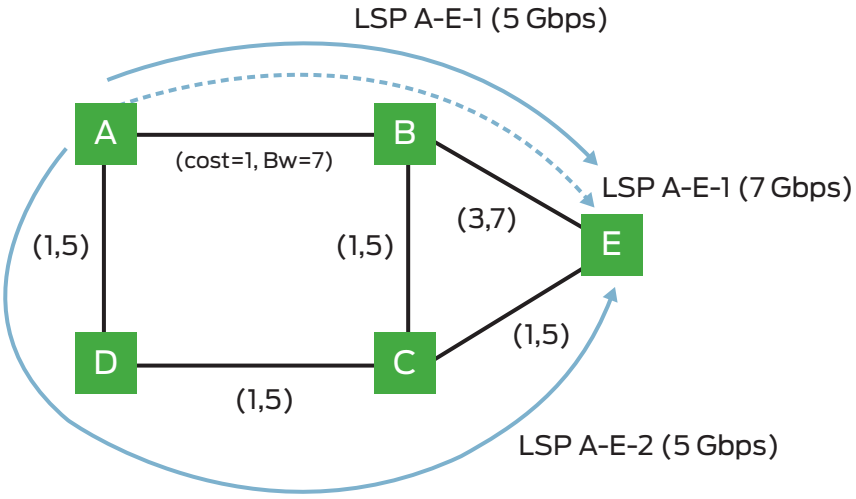

Figure 4: An increase in aggregate traffic causes splitting, a new member A-E-2 is created, and the old instance of A-E-1 (7 Gbps) is removed after re-signaling with 5 Gbps of bandwidth.

Figure 5 shows merging during the normalization process. The aggregate traffic on the container LSP has dropped to 4 Gbps, so when the normalization timer expires, merging occurs because the per-member utilization has hit the merging bandwidth of 2 Gbps. The container LSP deletes the member LSP-A-E-2 and re-signals the existing member A-E-1 with 4 Gbps of bandwidth.

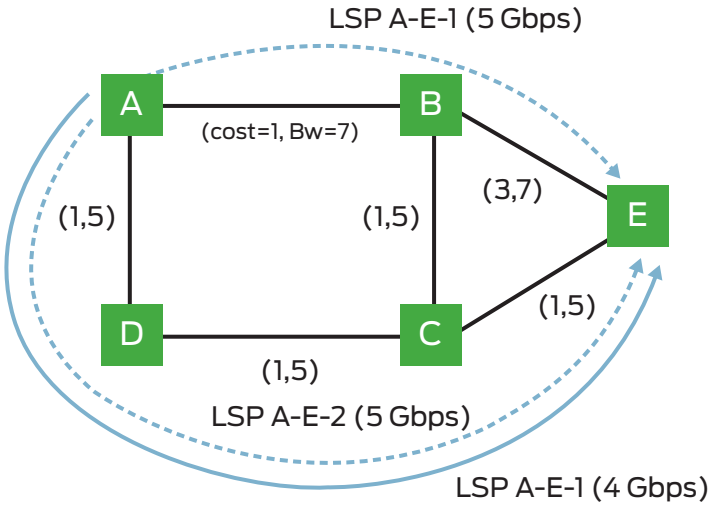

Figure 5: A decrease in traffic causes merging and removal of member LSP A-E-2 (5 GB). The old instance of A-E-1(5 GB) is removed after re-signaling with 4 GB.

It's important to note that there can be more than one correct path for each member LSP, whether a container is being set up from a cold start or following a splitting or merging decision. This flexible behavior helps ensure that bandwidth is used efficiently even as network conditions and application loads change.

Conclusion: Maximize Bandwidth Efficiency with TE++

Through its use of container LSPs, Juniper Networks TE++ provides a major step forward in resolving the bin packing problem. By splitting traffic across member LSPs and automatically adjusting LSPs in response to changing bandwidth demands and network events, TE++ ensures that bandwidth is used as efficiently as possible, lowering customers' CapEx.

TE++ also greatly simplifies operations. By operating automatically, Juniper TE++ significantly reduces or eliminates the manual configuration of LSPs. A network operator configures a single container LSP as opposed to multiple point-to-point RSVP LSPs. With TE++ container LSPs, there is a single entity to provision, manage, and monitor. Changes in topology are handled easily and autonomously by the ingress LSP, which adds, changes, or removes member LSPs to rebalance traffic, while maintaining the same TE constraints.

Juniper TE++ delivers these additional features and benefits:

- Use of multiple "good" paths instead of best effort or ECMP
- Efficient bin packing and load balancing of smaller LSPs over available paths
- Automatic measurement, computation, splitting, and merging of member LSPs within a container LSP
- Use of smaller LSPs (for example, less than 5% of aggregate link bandwidth) to allow routing with spare capacity even under failure conditions
- Increased reliability due to splitting bandwidth demand over N LSPs
- Partial protection at zero cost spare capacity

Customers benefit from lower OpEx as well as simpler network planning and management. End users benefit from more predictable application and service performance as TE++ ensures that LSPs are adjusted dynamically in response to application and network changes.

With Juniper TE++, network operators no longer have to wrestle with the challenges posed by the dynamic nature of networks. By automating configuration and bandwidth provisioning, Juniper TE++ gives customers the ability to maximize their investment in bandwidth.

About Juniper Networks

Juniper Networks is in the business of network innovation. From devices to data centers, from consumers to cloud providers, Juniper Networks delivers the software, silicon and systems that transform the experience and economics of networking. The company serves customers and partners worldwide. Additional information can be found at www.juniper.net.

Corporate and Sales Headquarters
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089 USA
Phone: 888.JUNIPER (888.586.4737)
or +1.408.745.2000
Fax: +1.408.745.2100
www.juniper.net

APAC and EMEA Headquarters
Juniper Networks International B.V.
Boeing Avenue 240
1119 PZ Schiphol-Rijk
Amsterdam, The Netherlands
Phone: +31.0.207.125.700
Fax: +31.0.207.125.701

2000587-001-EN Oct 2015