Juniper Networks | Engineering Simplicity

Scale-Out IPsec Solution for Enterprises — Juniper Validated Design (JVD)

Published
2025-05-26

Table of Contents

Scale-Out IPsec Solution for Enterprises — Juniper Validated Design (JVD)

Juniper Networks Validated Designs provide you with a comprehensive, end-to-end blueprint for deploying Juniper solutions in your network. These designs are created by Juniper's expert engineers and tested to ensure they meet your requirements. Using a validated design, you can reduce the risk of costly mistakes, save time and money, and ensure that your network is optimized for maximum performance.

About this Document

This document covers the Juniper Scale-Out Security Services Solution delivering a scalable solution for security services, scaling on your business needs, to enable security at high speed and high rate without using very large chassis. This solution can scale easily from small virtual to large security performances and scaling needs.

The summary of the solution platforms is as follows:

Table 1: Solution Platforms Summary

Solution Platforms Summary		
Solution	**Forwarding Layer**	**Service Layer**
Scale-Out Security Services for Enterprises	MX304 Universal Edge Router	SRX4600 vSRX

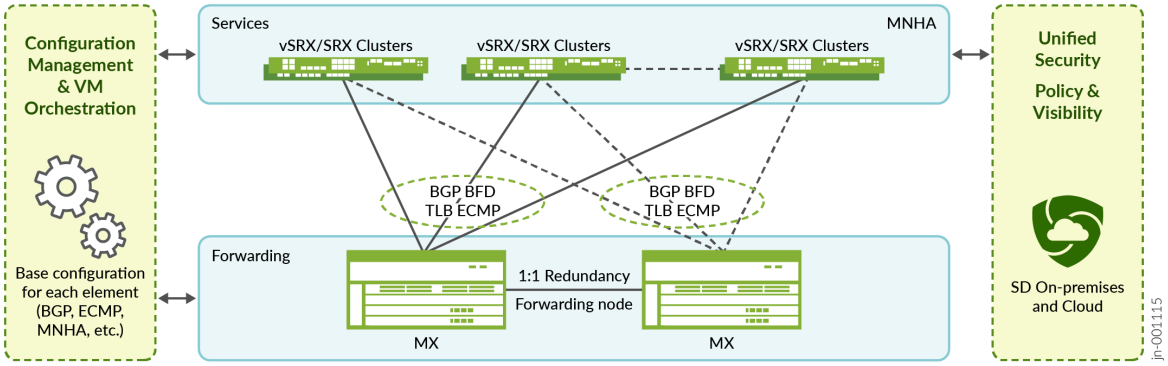
Solution Benefits

IN THIS SECTION

The Juniper Scale-Out Security Services Solution is a scalable IPsec Security Gateway (IPSEC) for use in central offices or for data centres in enterprises or managed security providers. The security complex leverages the scale-out network architecture and automation with a tight integration between routing and security services elements represented by MX universal routers and SRX Series Firewalls. This provides the best routing and security stacks of both worlds for optimal performance and total cost of ownership. The scale-out approach offers advantages over scale-up or integrates security engines directly into routing domain, including:

- Highly scalable IPsec systems with respect to number of tunnels and IPv4/IPv6 prefixes
- Pay-as-you-grow approach
- Flexibility to handle unpredictable traffic growth
- High availability with sub-second restoration for IPsec Security Associations
- Optimal operational preferences for a choice of physical or virtual nodes
- Improved time to market for security services on new platforms
- Flexible placement for security services in the network

Figure 1: Juniper Scale-Out General Architecture



This solution is equally applicable for the green-field deployments or as a nested solution on top of an existing MX Series Routers in the centralized or distributed networks allowing flexibility in placement of the services across enterprises and data centers infrastructure.

The Scale-Out Security Services Solution provides a scale out model for enabling high-capacity IPsec Gateway services combining Juniper MX Series modular and compact routers with Juniper vSRX and SRX4600 security products (Virtual Network Functions or Appliances). Generally, a solution includes three layers: security services layer, forwarding layer, and management and control layer, which enable consistent traffic flows through the service complex in both directions, addresses high availability requirements and simplified operations and management of multiple systems constitute the solution.

This JVD focuses on first two layers only, which include the following functional elements and solution building blocks:

Security Services Layer

- IPsec security services (terminating IPsec from branch/data centers/MSS/users)
- Stateful firewall (not focused as such however the SRX Series Firewall handles all traffic in a stateful way, even within IPsec)
- High availability function (using MNHA aka Multinode High Availability (MNHA))

Forwarding Layer

- Router forwarding plane with virtual routing instance (“external” and “internal”)

- Load balancing between multiple nodes of the security service layer
- High availability function
- Might include a distribution forwarding layer optionally

Use Case and Reference Architecture

IN THIS SECTION

Solution Functional Elements

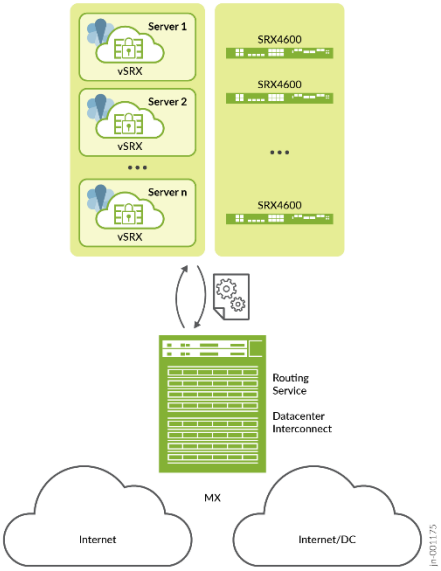
This JVD is part of a series of JVDs using the same scale-out principle. However, that is detailed for each use case, each one having some particularities applying to your use cases. The following use cases are described with a common part and specific parts.

This JVD document covers enterprise use case for IPsec Security Gateway.

Juniper Scale-Out Security Services Solution architecture includes two main functional blocks:

- The security services device is formed by standalone vSRX virtual network functions or SRX4600 Firewall or a redundant pair of the same device. This section covers a standalone case, former section covers details on redundant solution architectures.
- The MX Series Routers as load balancer routers provide 100G or 400G interfaces to the vSRX servers or the SRX4600s forming the services complexes. Both access side and Internet side peering are enabled through MX Series Router dedicated ports are used for high throughput.

Figure 2: Scale Out Solution Functional Blocks



With the new Trio 6 MX10004 and 10008 systems, capacity per slot is up to 9.6 Tbps and with compact MX304 systems, capacity per system is up to 4.8 Tbps, enabling a high number of 100G ports. An MX304 router can provide up to 48 x 100G interfaces and an LC9600 line card in a modular MX10000 system, up to 96 x 100G ports.

To optimize port usage, it is recommended to implement an intermediate distribution layer with two (or more) QFX Series switches to aggregate multiple SRX (and vSRX Series on compute servers) Series Firewalls nodes into a bundled 400GE links on the MX Series Router. In such case, the aggregate links terminate from the MX Series Routers onto the distribution layer rather than on physical SRX Series Firewalls (or the computes for vSRX).

SRX4600 offers a 400Gbps throughput capacity of up to 4 x 100GE interfaces in a 1RU appliance, perfectly fitting the use case interconnection with the MX Series Router.

If vSRX is the choice for the security element, it rolls out on top of the KVM or VMware virtual network function, running on open compute servers. You can bring your own server based on prescribed server specifications (CPU cores, memory, Linux OS, KVM versions). For more information about server specifications, see the vSRX server specifications in the references.

vSRX is a Virtual Network Function (VNF) running on KVM or VMware hypervisors, with a flexible compute allocation of cores (up to 32) and memory (up to 64G). vSRX can use virtio or SR-IOV (Single Root I/O Virtualization) with smart NICs such as Mellanox ConnectX 6. On the available platforms, hardware acceleration can be leveraged for IKE and IPsec encryption (such as AES-NI for DH and RSA algorithms).

An external BGP (eBGP) protocol with BFD provides a routing and control function between network elements while you can implement load-balancing using following two approaches:

- Equal-cost multipath (ECMP) load-balancing function with Consistent Hashing (CHASH)
- RE based traffic load balancer function (TLB) on MX Series Routers

Two routing instances – Internet and Internal – are used on MX Series Router to peer with corresponding network segments of the enterprise or data center network infrastructure and the security node. eBGP enables scalable and flexible exchange of routing information for the Internet side routing and the internal networks. The failure detection is based on BFD with timers as low as 100ms, enabling fast reconvergence and fast automatic adjustment for the ECMP load balancing.

The IPsec services on the Internet are load balanced between services nodes dynamically based on ECMP CHASH with source IPv4 or IPv6 addresses. The IKE and IPsec termination point (the common IPsec gateway) needs to be distributed through eBGP to the external peers. For the decrypted traffic coming out of the IPsec tunnels on the Internal side, an eBGP routing and BFD failure detection is required. Source based IPv4 or IPv6 ECMP CHASH is used on the Internet side with the IPsec services.

Essentially ECMP CHASH limits the impact on existing traffic flows in the event of service-node failure or addition of new service node to the scale-out architecture. In case of a service-node failure event, only the impacted flows are rehashed and rebalanced to other nodes; while in case of a service-node addition, an equal number or flows from each node in the cluster are rehashed and rebalanced to the new member in the cluster, limiting the impact while maintaining the equal-cost load balancing.

This architecture effectively scales the service complex with tens of service nodes (SRX/vSRX), with efficient load-balancing of flows between service nodes, and minimizing the effect (blast radius) due to a single node failure. The eBGP routing on the MX Series Router scales beyond Internet tables to millions of routes if required and easily beyond.

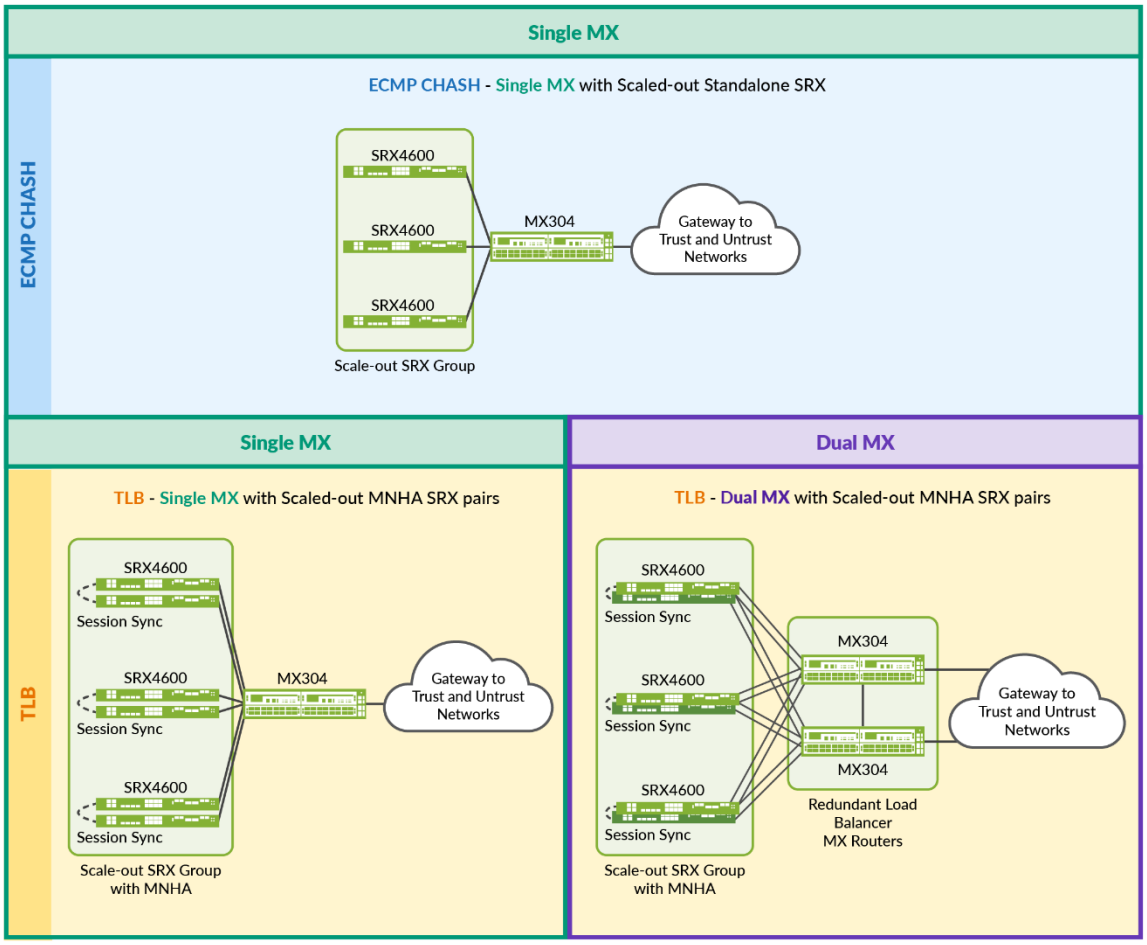
Solution Deployment Scenarios

Following the Scale-Out Security Services solution architecture, you can consider a few deployment scenarios where the MX Series Routers and SRX Series Firewalls are connected in either standalone or redundant pairs (see topologies). It uses network redundancy mechanisms to provide flow resiliency between the MX Series Router Forwarding Layer and SRX Series Router Service's. Also, the BFD protocol is used to achieve a quicker failover mechanism on routing when any other failure occurs. If SRX's MNHA provides session synchronization (stateful sessions and IPsec security associations) between two nodes, then the existing traffic and tunnels can continue uninterrupted.

Figure 3 on page 7 shows the three main topologies covered in the JVD, combining standalone/dual MX Series Router with standalone/MNHA for SRX Series Firewalls, each on a particular load balancing

mechanism (ECMP or TLB). It uses three SRX Series Firewalls for the first topology and doubles them to three pairs for the other topologies.

Figure 3: Validated Topologies



There are numerous trade-offs with each of the architectural choices. In general, complexity increases as more redundancies are added. For example, SRX MNHA Pairs introduce some requirements like a network link for high availability communications. Based on dependencies, load balancing method is used on the MX Series Router (namely ECMP CHASH or TLB). This selection of topologies covers the most important considerations from simple to more redundancy scenarios.

- ECMP CHASH is simpler to use, leverages standard protocols and well know ECMP mechanism, which might be a preferable option for some enterprise network operations department, though this method is somewhat limited when it comes to failover capabilities.
- TLB is the latest load balancing capability (at the time of publishing this JVD), which leverages services to load balancing, offers better redundancy capabilities and multiplies with different local

groups. It is useful when there is a need to combine different use cases with the same architecture. Though this method might not be backward compatible with older Junos OS releases.

Table 2: Validated Features Combination

Load-Balancing Method	Junos OS Release for MX	Number of MX	Security Features	SRX standalone	SRXs MNHA cluster
ECMP with Consistent Hashing	23.4R2	Single MX	IPSEC	**Yes**	**No**
Traffic Load Balancer (TLB)	23.4R2	Single MX	IPSEC	**Yes**	**Yes**
with Health Checking		Dual MX	IPSEC	**Yes**	**Yes**

NOTE: The Scale-Out solution only uses standard mechanisms and protocols between the components and does not require any special proprietary protocols. The exception is in the way load balancing is implemented internally (how the MX Series Router handles and distributes sessions). From a networking point of view, this solution uses standard protocols.

The following networking features are deployed and validated in this JVD:

- Dynamic routing using **BGP**
- Dynamic fault detection using **BFD**
- Load balancing of sessions across multiple SRX Series Firewalls (standalone or high availability)
- Load balancing using ECMP **CHASH (first appeared in Junos OS Release 13.3R3)**
- Load balancing using Traffic Load Balancer (TLB) on the MX Series Router (**first appeared in Junos OS Release 16.1R6)**
- MX Series Routers redundancy between two MX Series Routers with TLB
- MX Series Routers redundancy using Dynamic Routing between two MX Series Routers with TLB
- SRX Series Firewalls redundancy using MNHA as Active/Backup with sessions synchronization
- Dual stack solution with IPv4 and IPv6 (for outer IPsec and for inner tunnelled traffic)

- IPsec (auto VPN with responder only mode). AES-256-GCM is used for IPsec Encryption
- Stateful firewall (SFW) is inherently used for traffic inside the IPsec tunnels with simple long protocol sessions (HTTP, UDP)

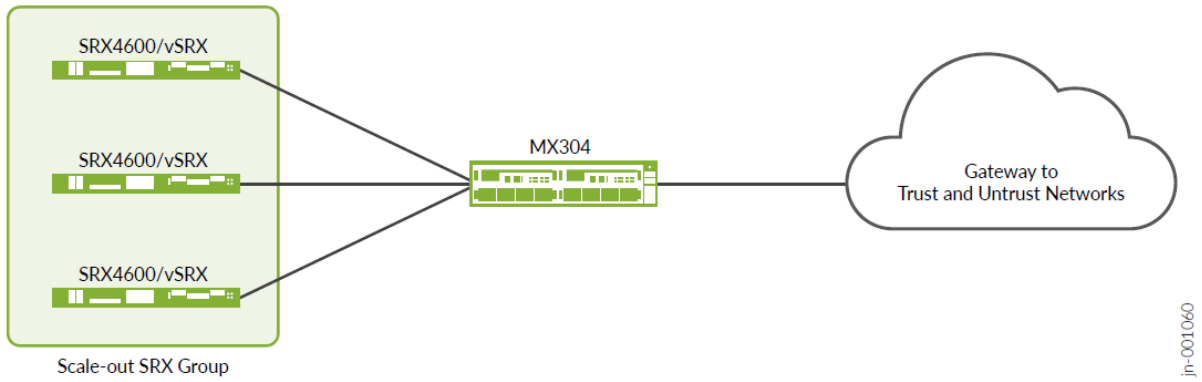
Each JVD is tested with the following platforms:

- Routing and Load Balancer: **MX304 with Junos OS Release 23.4R2**
- Security Services: **vSRX and SRX4600 with Junos OS Release 23.4R2**

Deployment Scenario 1 – ECMP CHASH – Single MX Series Router with Scaled Out Multiple Standalone SRX Series Firewalls

This topology is the simplest, however, the least redundant. The resiliency is provided at MX Series Router hardware – redundant RE, PSU, and so on. There is no protection against MX Series Router failure. Deployment allows protection against service node failure by redistributing traffic flows between two remaining security-nodes. Though there is no session synchronization between the SRXs which leads to longer restoration time for the affected flows (IPsec sessions need to reestablish).

Figure 4: Deployment Scenario 1 – ECMP CHASH - Single MX, Multiple Standalone SRX Firewalls



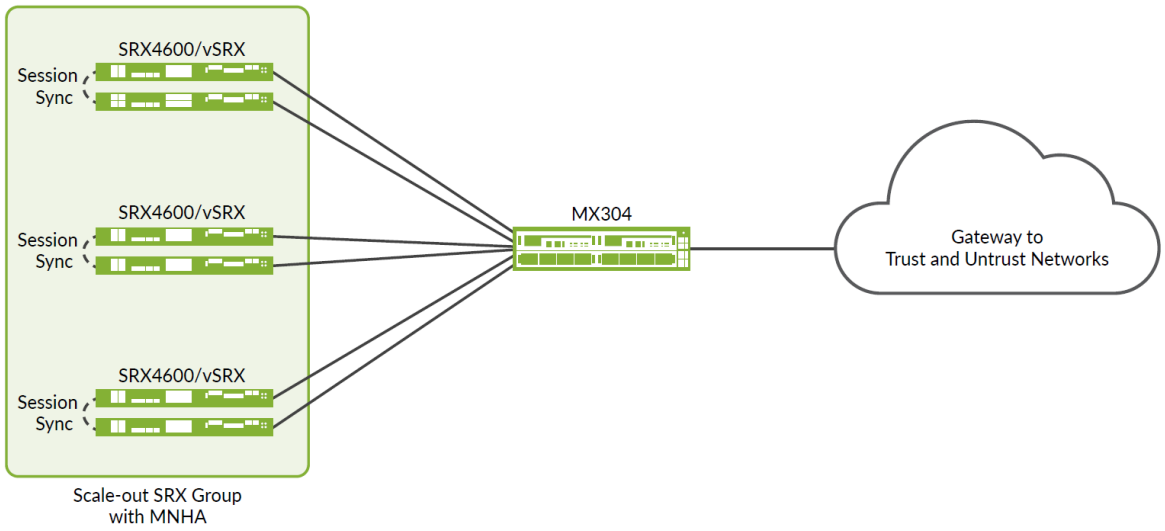
Network operators that are not concerned about stateful failover can augment security service capacities by adding more SRX Series Firewalls in this simple manner as application sessions are short lived anyway (a redundancy mechanism is handled at the application level, so session sync between two different firewalls is not required).

- Pros: Simplicity and scaling with each individual SRX Series Firewalls
- Cons: No redundancy

Deployment Scenario 2 – TLB – Single MX Series Router with multiple SRX MNHA pairs

This topology offers redundancy for the SRX Series Firewalls and not for the MX Series Router. This topology has a second Routing Engine (RE) installed in the appropriate slot; however, this does not use two MX Series Routers in that case. All SRX Series Firewalls are in MNHA pairs to offer sessions synchronization and failover capabilities.

Figure 5: Topology 3 – TLB - Single MX, Multiple SRX MNHA Pairs

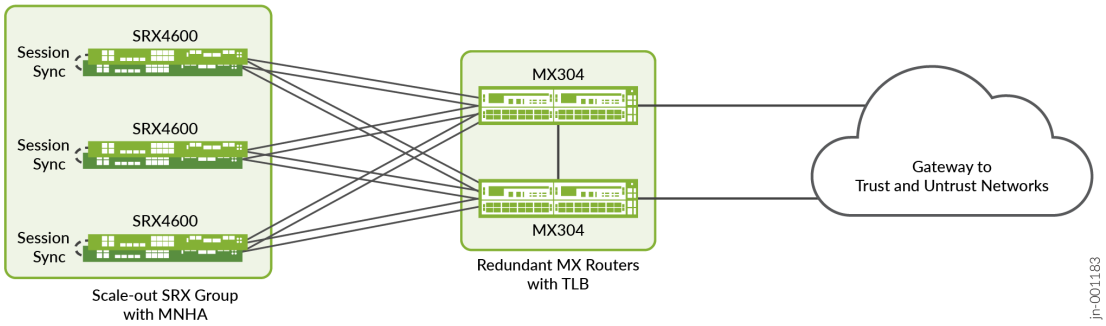


- Pros: Redundancy and scaling with each SRX Series Firewalls pair
- Cons: No redundancy on the router (except using dual RE)

Deployment Scenario 3 – TLB – Dual MX Series Router with Multiple SRX MNHA Pairs

This last topology offers the most redundancy both for MX Series Routers and SRX Series Firewalls and takes advantage of having all components used at the same time. Any failover scenario can be covered using BGP routes announced and BFD accelerating the error detection.

Figure 6: Topology 4 – TLB – Dual MX, Multiple SRX MNHA Pairs



MX Series Router can handle traffic on any of the two routers, while SRX Series Firewalls are used either in Active/Backup role or in Active/Active role, making use of both nodes at the same time. This augments the capacity of the network during normal operation. However, this leaves one node active at a time when a failure occurs (considering a single MNHA cluster).

Each SRX Series Firewall is connected to both MX Series Routers. If any of one node fails within a cluster, all other SRX Series Firewalls pairs might have an independent failover from the other SRX Series Firewalls pairs and the MX Series Router.

- Pros: Full redundancy and scaling for MX Series Router and SRX Series Firewalls pairs.
- Cons: More interfaces used on the MX Series Router if directly connected. Then, an optional distribution layer can cover more connectivity needs when SRX Series Firewall count augments.

Supported Platforms and Positioning

IN THIS SECTION

To review the software versions and platforms on which this JVD was validated by Juniper Networks, see the Validated Platforms and Software section in this document.

Test Optics

The fiber optic transceivers used in the test bed are:

- QSFP-100GBASE-SR4: between MX304 and SRX4600s
- QSFP28-100G-AOC-3M: between MX304 and servers hosting vSRXs

This JVD is validated with the fiber optics reference above. However, the technical validation is larger in regard to hardware compatible optics. For more information, see the references for Juniper's Hardware Compatibility Tool.

- For SRX4600**:** https://apps.juniper.net/hct/product/?prd=SRX4600
- For MX304: https://apps.juniper.net/hct/product/?prd=MX304
- For MX10004: https://apps.juniper.net/hct/product/?prd=MX10004

vSRX Setup and Sizing

This JVD focuses only on the functional aspect of the solution. It does not matter whether powerful servers are tested for hosting the vSRX(s) and which vSRX size is used here. For real world performances, high end servers (such as Dell or HPE servers with Intel Gold or AMK 9K CPUs, 256GB RAM and ConnectX6 or X7 or later interfaces) with large vSRX sizes are proposed (such as 16 vCPU and 32GB RAM). For vSRX requirements information, see the following Juniper documentation:

- VMware ESXi: https://www.juniper.net/documentation/us/en/software/vsrx/vsrx-consolidated-deployment-guide/vsrx-kvm/topics/concept/security-vsrx-kvm-understanding.html
- KVM: https://www.juniper.net/documentation/us/en/software/vsrx/vsrx-consolidated-deployment-guide/vsrx-vmware/topics/concept/security-vsrx-vmware-overview.html

Test Objectives

IN THIS SECTION

JVD is a cross-functional collaboration between Juniper solution architects and test teams to develop coherent multidimensional solutions for domain-specific use cases. The JVD team comprises technical leaders in the industry with a wealth of experience supporting your complex use cases. The scenarios selected for validation are based on industry standards to solve critical business needs with practical network and solution designs.

The key goals of the JVD initiative include:

- Validate overall solution integrity and resilience
- Support configuration and design guidance
- Deliver practical, validated, and deployable solutions

A reference architecture is selected after consultation with Juniper Networks global theaters and a deep analysis of your use cases. The design concepts that are deployed use best practices and leverage relevant technologies to deliver the solution. Key performance indicators (KPIs) are identified as part of an extensive test plan that focuses on functionality, performance integrity, and service delivery.

Once the physical infrastructure that is required to support the validation is built, the design is sanity-checked and optimized. Our test teams conduct a series of rigorous validations to prove solution viability, capturing, and recording results. Throughout the validation process, our engineers engage with software developers to quickly address any issues found.

Test Goals

The test objective is to validate the Scale-Out architecture, showing the various topologies with single/dual MX Series Routers and multiple SRX Series Firewalls, and demonstrate its ability to respond to various use cases while being able to scale. Different possibilities are offered to cover the routing with the two main load balancing methods, using different platform sizes for MX Series Routers and/or SRX Series Firewalls, in addition to using high availability of the various components.

Additional goals are to demonstrate Scale-Out capability of the solution, which allows linear performance and logical scale (stateful traffic flows) growth in the process of new SRX/vSRX Series Firewalls addition to the security services complex.

Use this JVD validate system behavior under the following administrative events, with a general expectation to have no or little effect on the traffic:

- Adding a new SRX Series Firewall to the service layer - a redistribution of traffic to get an even distribution, minor percentage of traffic disturbance [depends on the number of SRX Series Firewalls next-hops] is seen on all other SRX Series Firewall due to change in next-hops and then the hash.
- Removing a SRX Series Firewall from the service layer - traffic redistribution only for those associated with this removed SRX Series Firewall.
- Having a SRX Series Firewall failover to its peer (MNHA case) and return to normal state - no traffic disruption expected, sessions and IPsec Security Associations are preserved.
- Having an MX Series Router failover (dual MX Series Router) - no traffic disruption expected.
- Variation among these themes and failure scenarios - no traffic disruption expected.

The following networking features are deployed and validated in this JVD:

- Dynamic Routing using BGP
- Dynamic fault detection using BFD
- Load Balancing of sessions across multiple SRX Series Firewalls in standalone or high availability
- Load Balancing using ECMP CHASH, first appeared in Junos OS Release 13.3R3
- Load Balancing using Traffic Load Balancer on the MX Series Router first appeared in Junos OS Release 16.1R6
- MX Series Router redundancy using BGP Dynamic Routing between two MX Series Router with TLB
- SRX Series Firewalls redundancy using Multi-Node High Availability (MNHA) as Active/Backup with sessions and IPsec security associations synchronization
- Dual stack solution with IPv4 and IPv6 (for outer IPsec and for inner tunneled traffic)
- IKE/ IPsec tunnel negotiation using AES-GCM encryption protocols as responder mode (waiting for IPsec peers, SRX Series Firewalls IPsec configured with auto-vpn mode)
- Stateful firewall is implicitly used with simple long protocol sessions (HTTP, UDP) for traffic inside the IPsec tunnels
- Dead Peer Detection (DPD) helps in detecting unreachable IKE peers. It helps to maintain a link active while no traffic flows inside and detect end to end VPN reachability issues

Individual tests of failure, failover, upgrades, and downgrades are used to show how the architecture behaves and verify its ability to react to failures, including resiliency of MX Series Router and SRX Series Firewalls in various conditions. It also shows the consistency of the traffic distribution and its possibility to scale when adding new service nodes.

Fixed performance/scaling of traffic is tested across each platform with the idea of showing linearity in the process of SRX Series Firewalls addition (standalone or MNHA pair) to the Scale-Out solution.

Test Non-Goals

Maximum scale and performance of the individual network elements constitute the solution.

There are no preferred specifications for the hypervisor hosting the vSRX, nor specific vSRX sizes (in vCPU/vRAM/vNIC quantity). Simple vSRX is sufficient for testing the required features. Note that vSRX runs on many hypervisors including: ESXi, KVM, Microsoft for on-prem. Though vSRX can also be deployed in public clouds (AWS, Azure and GCP), the purpose of the architecture is not to run with vSRX in those external clouds where it is questionable to consider the networking plumbing to get them connected.

> **NOTE**: This JVD does not mention about automation. However, automation is used to build and test the solution with various use cases and tests.

Following features and functions are not included in the current JVD:

- Automated onboarding of the vSRX
- Security Director Cloud or on-prem
- Application and Advanced Security features (App ID, IDP, URL filtering and other Layer 7)
- IKE/IPsec tunnel negotiation using protocols other than AES-GCM or other initiator mode (to other peers)
- IKE using Public Key Infrastructure (PKI) is not used; however, it works the same

Event Testing

SRX Series Firewalls failure events:

- MX Series Router to SRX Series Firewalls link failures

- SRX Series Firewalls reboot
- SRX Series Firewalls power off
- Complete MNHA pair power off
- IKE/IPsec failures

MX Series Router failure events:

- Reboot MX Series Router
- Restart routing process
- Restart traffic-dird deamon
- Restart Network-monitor deamon
- Restart sdk-process
- GRES (Graceful REStart of routing daemon)
- ECMP/TLB next-hop addition/deletion (adding/deleting new scale out SRX MNHA pair)
- SRD based cli switchover between MX Series Router (ECMP)

Traffic recovery is validated post all failure scenarios.

UDP traffic generated using IXnetwork for all the failure related test cases is used to measure the failover convergence time.

Tested Traffic Profiles

Tested traffic profiles are composed of multiple simultaneous flows showing the same for either a standalone SRX Series Firewalls or a SRX MNHA pair in Active/Backup mode.

Table 3: Tested Traffic Profiles Per SRX (or MNHA Pair)

Tunnel Count/ MNHA-Pair	Packet Size	Traffic Type	Throughput	Platform
1000	SECGW-IMIX	UDP	40Gbps	SRX4600
1000	SECGW-IMIX	UDP	40Gbps	vSRX, CPU/ vSRX :90%

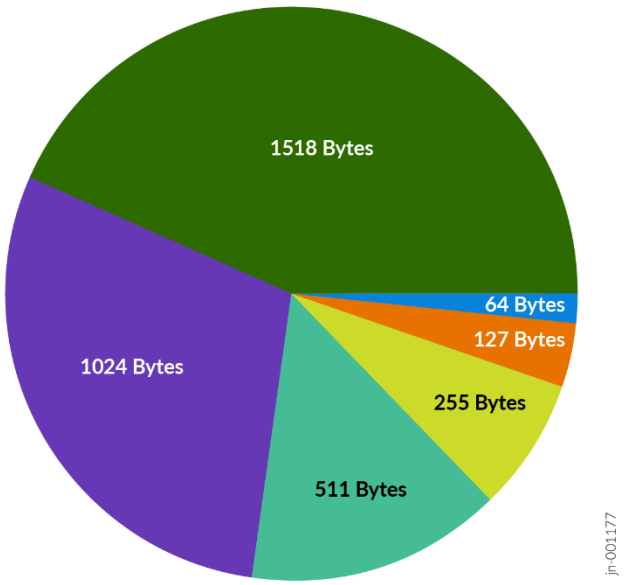
NOTE: These performances are not at the maximum capability for each platform however, it is a steady performance representative to test among multiple SRX4600/vSRX in similar conditions.

Packet size is using the Internet mix with average packet size of ~700bytes. The “Packet Size: Weight” distribution is as follows:

- 64: 8
- 127:36
- 255:11
- 511:4
- 1024:2
- 1518:39

NOTE: Lab used end to end 9000 for MTU to prevent fragmentation.

Figure 7: Packet Size: Weight Distribution



Test Bed Configuration

Contact your Juniper representative to obtain a full archive of the test bed configuration used for this JVD.

Solution Architecture

IN THIS SECTION

Traffic Path in IPsec Scale-Out Solution

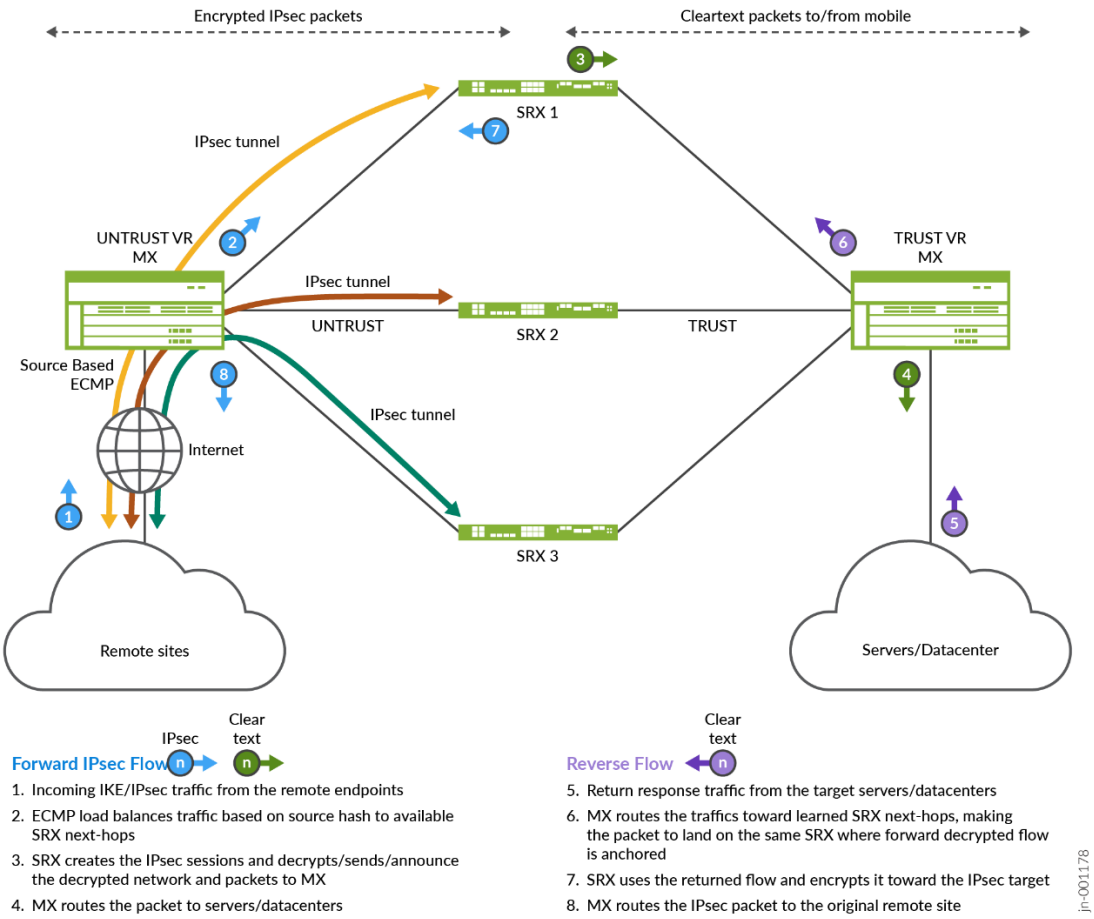
The scale-out solution is based on BGP as dynamic routing protocol. It enables all the MX Series Router and SRX Series Firewalls to learn from their surrounding networks, however, most importantly to exchange path information of the network traffic that is sent from the MX Series Router across each SRX Series Firewalls to the next MX Series Router.

This protocol enables the exchange of network paths for the external user subnets coming from IPsec peers and the specific internal networks. When each SRX Series Firewalls announces its own IKE/IPsec termination gateway to its BGP peers, each with the same “network cost”, the load balancing algorithm can then use those routes for load balancing across each SRX Series Firewalls. In case of IPsec traffic,

the Internet side needs to announce the IP address used for establishing the tunnel (the IKE destination at minimum, usually both IKE – phase 1 - and IPsec – phase 2 - negotiation use the exact same address), on the other side the inner traffic transported over IPsec (the IP addresses negotiated in a traffic selector of each party).

The following diagram shows how traffic flows are distributed from an MX Series Router to multiple SRX Series Firewalls using ECMP load balancing method for the IKE/IPsec traffic started from the ENB in a mobile network. The SRX Series Firewalls are in a symmetric sandwich between the two MX Series Routers in the diagram, whether those MX Series Routers are a single physical node configured with two routing instances (more typical) or two physical MX Series Router nodes on each side, the routing principle stays the same as if two routing nodes are used, maintaining the traffic flow distribution that is consistent in both directions. However, on one side of the SRX Series Firewalls is encrypted traffic and on the other side the clear text traffic to/from the mobile endpoint.

Figure 8: Flow Distribution with IPsec



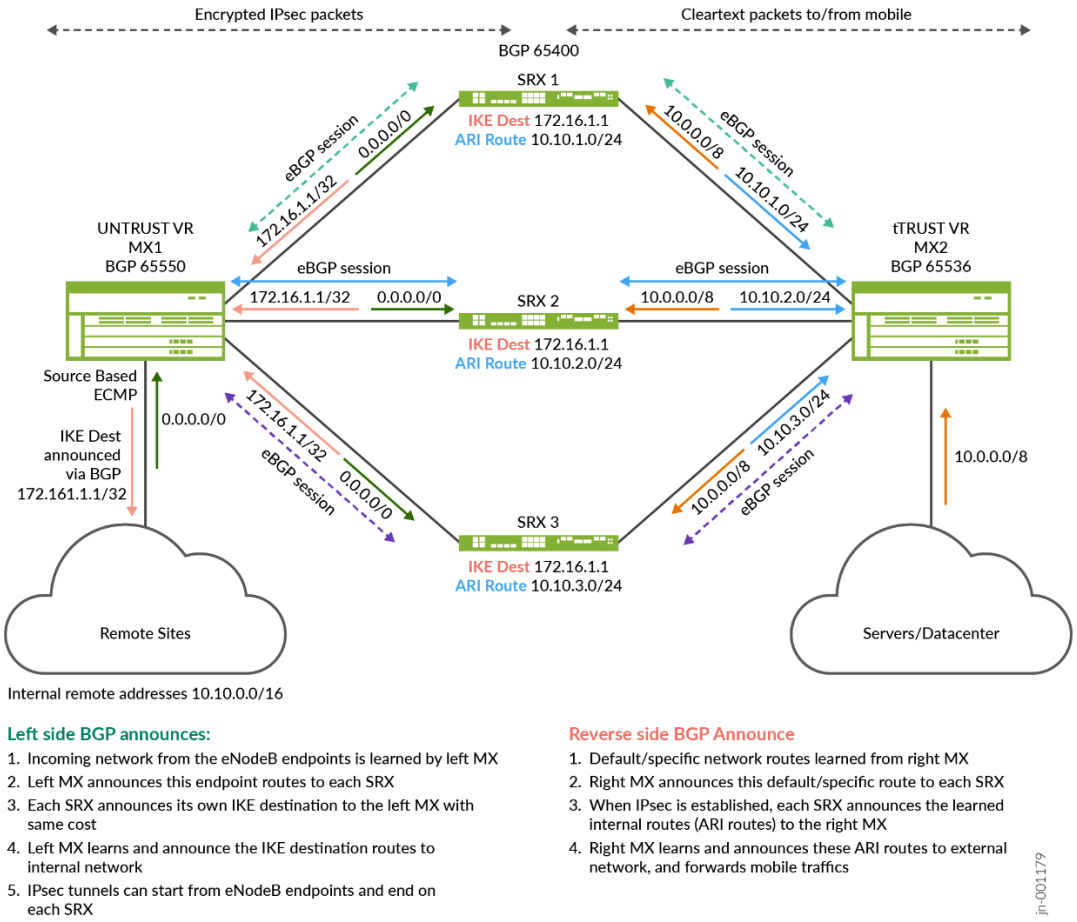
The MX Series Router on the left side uses UNTRUST-VR routing instance to forward traffic to each SRX Series Firewalls. On the left side, only IPsec traffic is seen. The only IP addresses to announce are the ones used by the remote sites (IPsec gateways or users) and the same IKE gateway IP address is used by each SRX Series Firewalls. The routes on this side are announced through BGP to the next hop, making its path available on each MX Series Router through each SRX Series Firewalls (with same cost for load balancing).

The MX Series Router on the right side uses TRUST-VR to receive traffic from each SRX Series Firewalls and forward it to the next-hop toward the target resources. When an IPsec tunnel is established on the left side (remote site to SRX), it negotiates (as part of the IPsec Security Association) an inner IP address(es) assigned to the remote entity. This is the IP address that is announced to the router on the right side, making the return path unique toward the specific SRX hosting that IPsec security association (the diagram shows a simple network IP address with a /24 prefix for IPv4, and an IPv6 shows a /120 prefix for example).

Routes are announced through BGP, each MX Series Router with their own BGP Autonomous System (AS) and peer with the SRX Series Firewalls on their two sides (TRUST and UNTRUST zones in a single routing instance). The MX Series Router peers with any other routers bringing connectivity to the Internet and servers/data center.

Figure 9 on page 21 shows how the traffic comes through the remote gateway. This starts an IPsec negotiation with one of the SRX Series Firewalls (destination being selected by the load balancing mechanism), then transported over IPsec to the SRX Series Firewalls. This SRX Series Firewalls then decrypts the packet and sends the content to the next hop. The return path across the right SRX Series Firewalls to the ARI route (Auto Route Injection) is announced by the SRX Series Firewalls to the MX Series Router on the right side. Each route is announced through BGP for making every network reachable.

Figure 9: Network BGP Within/Outside IPsec



NOTE: In the above diagram and later configuration examples, the “publicly” announced IKE Gateway address is 172.16.1.1/32 (though part of private space RFC1918) for the sake of the demonstration. All other 10.0.0.0/8 addresses are considered private IP addresses as per the same RFC.

Introduction to SRX Series Firewalls Multinode High Availability

For more information, see an extract from the public documentation on MNHA https://www.juniper.net/documentation/us/en/software/junos/high-availability/topics/topic-map/mnha-introduction.html .

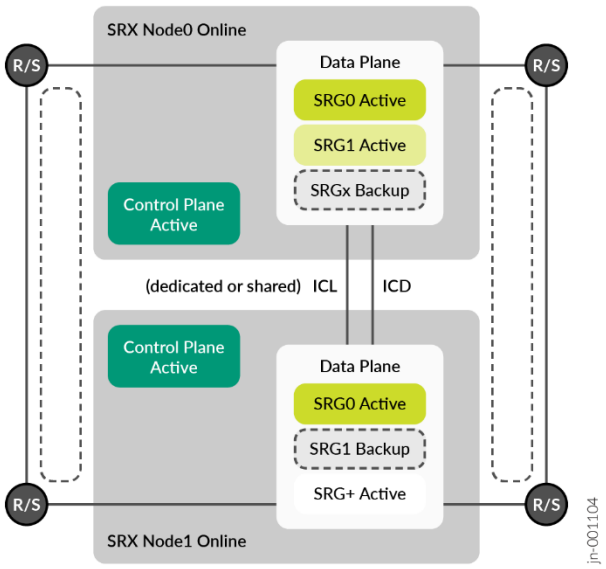
Juniper Networks SRX Series Firewalls support a new solution, Multinode High Availability (MNHA), to address high availability requirements for modern networks. In this solution, both the control plane and the data plane of the participating devices (nodes) are active at the same time. Thus, the solution provides inter-chassis resiliency.

The participating devices are either co-located or geographically separated to different rooms or buildings. Having nodes with high availability across geographical locations ensures resilient service. If a disaster affects one physical location, MNHA can fail over to a node in another physical location, thereby ensuring continuity.

In MNHA, both SRX Series Firewalls have an active control plane and communicate their status over an Inter Chassis Link (ICL) that can be direct or routed across the network. This allows the nodes to be geo-dispersed while synchronizing the sessions and IKE security associations. Also, they do not share a common configuration, and this enables different IP addresses settings on both SRX Series Firewalls. Use the commit sync mechanism for the elements of configuration to be same on both the platforms.

The SRX Series Firewalls uses one or more services redundancy groups (SRGs) for the data plane that can be either active or backup (for SRG1 and above). An exception is the SRG group 0 (zero) that is always active on both. This is a group that can be used natively by scale-out solution to load balance the traffic across both SRX Series Firewalls at the same time. However, some interest exists for the other modes where it can be Active/Backup for SRG1 and Backup/Active for SRG2. This is like always active SRG0, however can also add some routing information (like BGP as-path-prepend) under certain conditions. SRG1/+ offers more health checking of its surrounding environment that can be leveraged to make an SRGn group active/backup/ineligible.

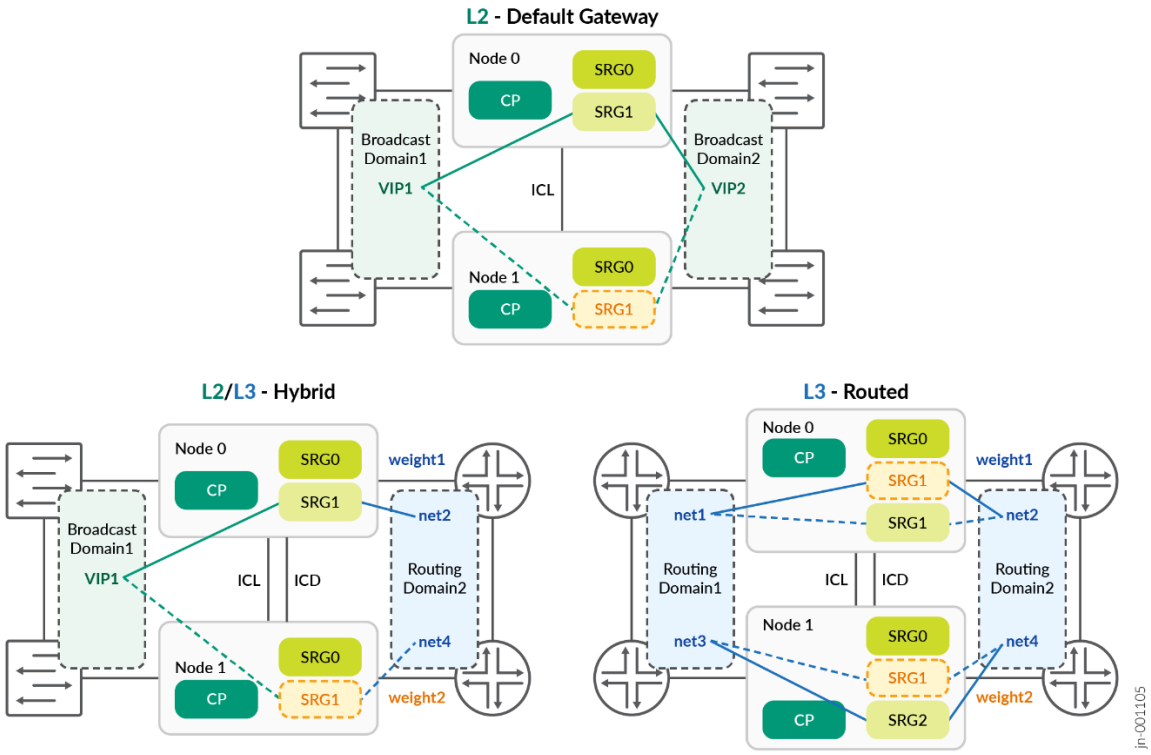
Figure 10: Munit Node High Availability General Architecture



MNHA can select a network mode between the following three possibilities:

- Default Gateway or L2 mode: It uses the same network segment at L2 on the different sides of the SRX Series Firewalls (for example, TRUST/UNTRUST) and both SRX Series Firewalls share a common IP / MAC address on each network segment. It does not mean the SRX Series Firewalls is in switching mode, it does route between its interfaces, however, shares the same broadcast domain on one side with the other SRX Series Firewalls, and same on the other side as well.
- Hybrid mode or mix of L2 and L3: It uses an L2 (broadcast domain) and IP address on one side of the SRX Series Firewalls (for example, TRUST) and routing on the other side (for example, UNTRUST) then having different IP subnets on the second side.
- Routing mode or L3: The JVD uses this architecture where each side of the SRX Series Firewalls is using a different IP address, even between the SRX Series Firewalls (no common IP subnet) and all communication with rest of the network happens through routing. This mode is perfect for scale-out communication using BGP with the MX Series Router.

Figure 11: MNHA Network Modes

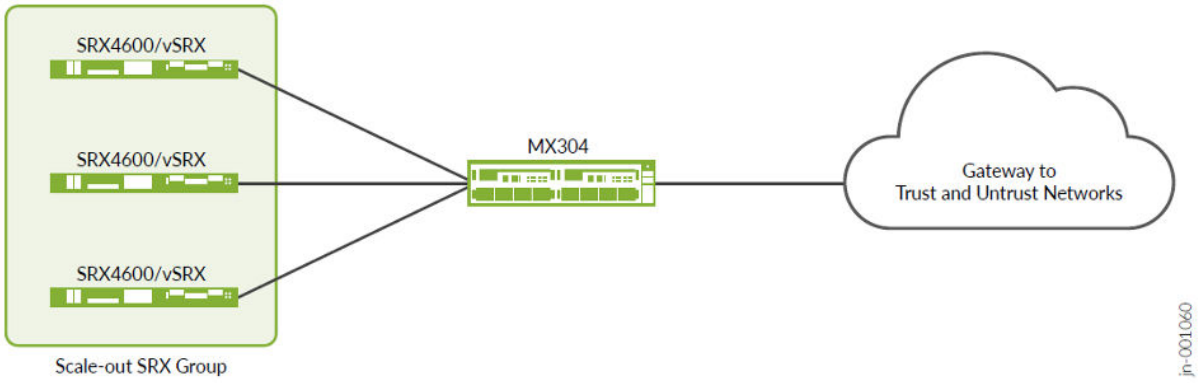


Whether using SRG0 Active/Active, or SRG1 Active/Backup (single one active at a time), or a combination of SRG1 Active/Backup and SRG2 Backup/Active, this simply uses one or two SRX Series Firewalls in a cluster at the same time.

ECMP Consistent Hashing (CHASH) Load Balancing Overview

This feature relates to the topology (single MX Series Router, multiple standalone SRX Series Firewalls) used with ECMP (dual MX Series Router and/or SRX Series Firewalls is not possible with this load balancing method).

Figure 12: Topology 1 - ECMP CHASH



ECMP Consistent Hashing in MX Series Router

Equal Cost Multi Path (ECMP) is a network routing strategy that allows traffic of the same session, or flow — that is, traffic with the same source and destination — to be transmitted across multiple paths of equal cost. It is a mechanism that allows to load balance traffic and increase bandwidth by fully utilizing bandwidth otherwise the unused bandwidth links to the same destination.

When forwarding a packet, the routing technology must decide which next-hop path to use. In deciding, the device considers the packet header fields that identify a flow. When ECMP is used, next-hop paths of equal cost are identified based on routing metric calculations and hash algorithms. That is, routes of equal cost have the same preference and metric values, and the same cost to the network. The ECMP process identifies a set of routers, each of which is a legitimate equal cost next hop towards the destination. The routes that are identified are referred to as an ECMP set. An ECMP set is formed when the routing table contains multiple next-hop addresses for the same destination with equal cost (routes of equal cost have the same preference and metric values). If there is an ECMP set for the active route, Junos OS uses a hash algorithm to choose one of the next-hop addresses in the ECMP set to install in the forwarding table. You can configure Junos OS so that the multiple next-hop entries in an ECMP set are installed in the forwarding table. On Juniper Networks devices, one can perform per-packet load balancing to spread traffic across multiple paths between routing devices.

The following example is of learned routes and forwarding table for the same destination (assuming traffic target is exact address 172.16.1.1/32 and SRX Series Firewalls BGP peers are 10.1.1.0, 10.1.1.8 and 10.1.1.16):

```
user@MX> show route 172.16.1.1/32
trust-vr.inet.0: 30 destinations, 33 routes (30 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
172.16.1.1/32        *[BGP/170] 4d 04:52:53, MED 10, localpref 100
                      AS path: 64500 64500 I, validation-state: unverified
                       to 10.1.1.0 via ae1.0      ## learning routes from BGP peer SRX1
                     >  to 10.1.1.8 via ae2.0      ## learning routes from BGP peer SRX2
                        to 10.1.1.16 via ae3.0     ## learning routes from BGP peer SRX3
user@MX> show route forwarding-table destination 10.0.2.0/24 table trust-vr
Routing table: trust-vr.inet
Internet:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
172.16.1.1/32       user     0                    ulst  1048580     2
                              10.1.1.0           ucst      801     4 ae1.0      ## to SRX1
                              10.1.1.8           ucst      798     5 ae2.0      ## to SRX2
                              10.1.1.16          ucst      799     5 ae3.0      ## to SRX3
```

With scale-out architecture where stateful security devices are connected, maintained symmetricity of the flows in the security devices is the primary objective. The symmetricity means traffic from a subscriber (remote user or remote site) to the same subscriber must always go through the same SRX Series Firewalls (which maintains the subscriber state). To reach the same SRX Series Firewalls, the traffic must be hashed onto the same link towards that SRX Series Firewalls in both directions.

A subscriber is identified by the source IP address in the upstream direction (client to server) and by the destination IP address in the downstream direction (server to client). MX Series Routers do symmetric hashing i.e. for a given (sip, dip) tuple, same hash is calculated irrespective of the direction of the flow i.e. even if sip and dip are swapped. However, this is not enough for our requirement as it requires all flows from a subscriber to reach the same SRX Series Firewalls – so hash only on source-ip address (and not destination-ip address) in one direction and vice versa in the reverse direction.

However, in the present IPsec use case, traffic is not the same on both sides of the firewall. On the left side there is IPsec, coming from remote sites and terminating on the SRX Series Firewalls, and the inner traffic from the SRX Series Firewalls on the right side going to the internal servers or data center. However, the symmetry of the traffic needs to be true. The SRX Series Firewalls receiving the initial IKE/IPsec request establishes a tunnel with the source of that tunnel (the remote site), and in the IPsec negotiation (IKE phase 2) is also negotiated with source/destination IP address (i.e. the traffic selector or encryption domain depending on the language used). In the remote site term, this source IP address or subnet negotiated in this traffic selector is the one that is then used and announced through BGP to the next MX Series Router in the chain (this is the ARI route, aka Auto Route Injection). This makes the return traffic to that remote site reach the correct SRX Series Firewalls and then route that traffic back to the proper IPsec tunnel to its destination.

By default, when a failure occurs in one or more paths, the hashing algorithm recalculates the next hop for all paths, typically resulting in the redistribution of all flows. Consistent load balancing enables you to override this behavior so that only flows for links that are inactive are redirected. All existing active flows are maintained without disruption. In such an environment, the redistribution of all flows when a

link fails potentially results in significant traffic loss or a loss of service to SRX Series Firewalls whose links remain active. However, consistent load balancing maintains all active links and instead remaps only those flows affected by one or more link failures. This feature ensures that flows connected to links remain active and continue uninterrupted.

This feature applies to topologies where members of an ECMP group are external BGP neighbors in a single-hop BGP session. Consistent load balancing does not apply when you add a new ECMP path or modify an existing path in any way. The new SRX Series Firewalls design is implemented where SRX devices are added gracefully with an intent of equal redistribution from each active SRX Series Firewalls. Hence, it causes minimal impact to existing ECMP flows. For example, if there are four active SRX Series Firewalls carrying 25% of total flows on each link and a 5th SRX Series Firewalls (previously unseen) is added, 5% of flows from each existing SRX Series Firewalls move to the new SRX Series Firewalls. Hence results in 20% of flow re-distribution from an existing four SRX Series Firewalls to new one.

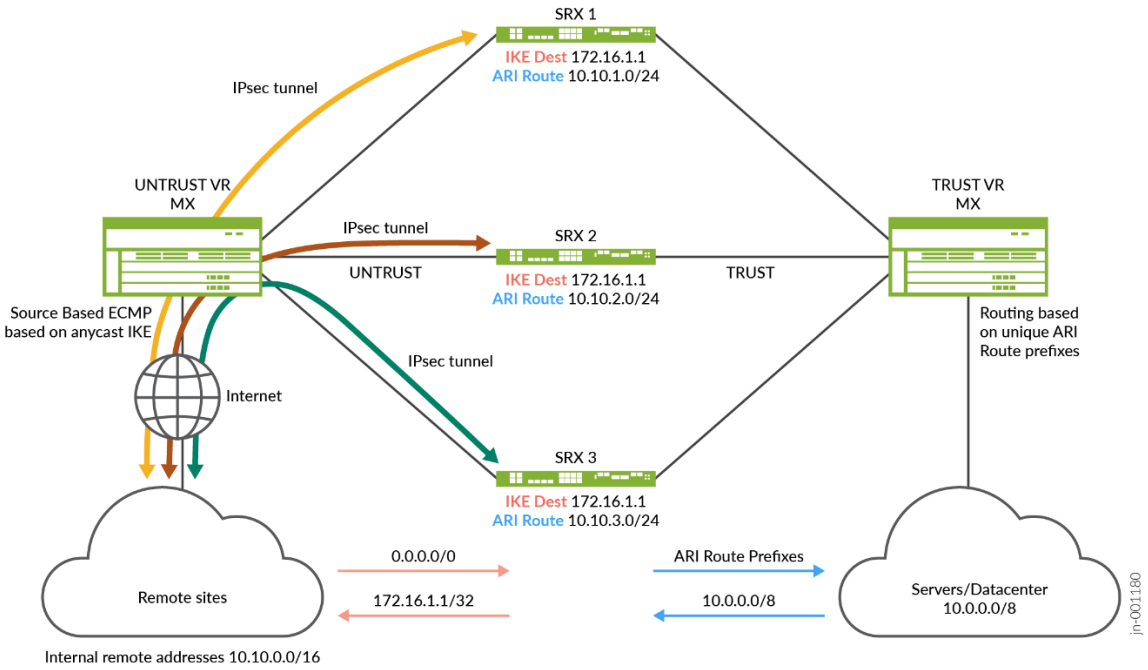
In case of traffic redistribution (loss of a single SRX Series Firewalls or addition of a new SRX Series Firewall) the IPsec peer renegotiates to that “new” peer IKE gateway as the Security Association does not exist yet.

In the case of SRX MNHA pair, any failover (if losing its SRX Series Firewalls other node) from one to another in the same pair reuses the existing synchronized IPsec Security Association and no renegotiation happens.

ECMP CHASH Usage in Topology 1 (Single MX Series Router, Scale-Out SRXs) for IPsec:

NOTE: IPsec use case usually accepts IPsec connections from remote sites (it can be remote users). The connections from the left slide as shown in Figure 13 on page 28 and then their respective internal traffic transported over IPsec then decapsulated and send to the right side, typically to internal servers or a private data center.

Figure 13: Topology 1 - ECMP CHASH - IPsec Use Case



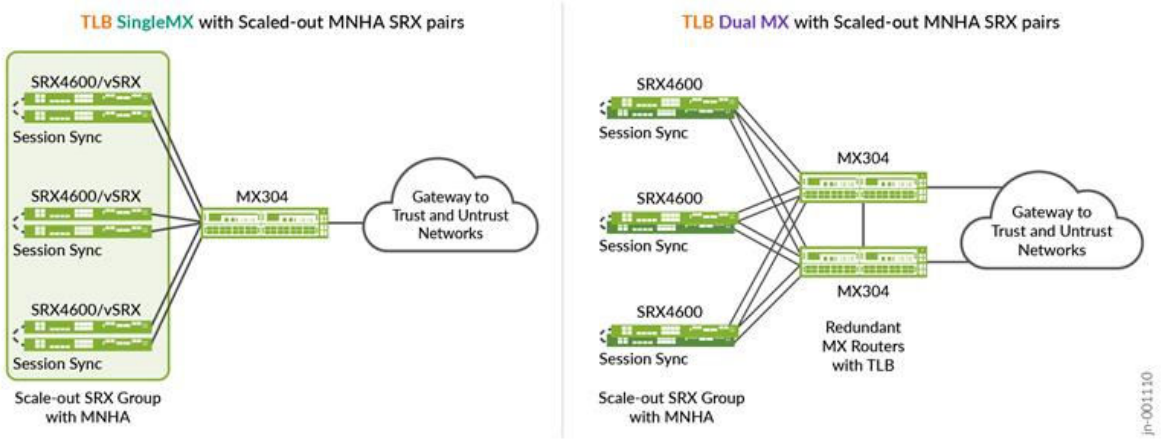
- All the scale-out SRX Series Firewalls connected to MX Series Router are configured with EBGP connections.
- All the scale-out SRX Series Firewalls need to be configured with auto-vpn config and with the same anycast IP address hosted on loopback interface as IKE endpoint IP address. All the SRX Series Firewalls are in IPsec responder only mode.
- IPsec Tunnels getting initiated behind MX Series Router from IPsec initiator uses same SRX IKE endpoint IP with unique traffic-selectors. This traffic selector is used by SRX Series Firewalls to install unique ARI routes to attract the data return traffic from the server to the right IPsec tunnel.
- A Load-balancing policy with source-hash for anycast IP address route is configured in the forwarding-table.
- MX Series Router receives anycast IP address route on UNTRUST side and advertised using EBGP to MX Series router on the UNTRUST side. MX Series Router imports this route on the UNTRUST instance using load-balancing consistent-hash policy.
- MX Series Router on the UNTRUST side has an ECMP route for anycast IP address.
- IKE traffic initiated from IPsec initiator router reaches MX Series Router on UNTRUST instance and hits ECMP anycast IP address route and takes any one ECMP next hop to SRX Series Firewalls based on the calculated source IP address-based hash value.

- SRX Series Firewalls anchors the IKE session and installs the ARI route.
- SRX Series Firewalls advertises the ARI route towards the TRUST direction of MX Series Router.
- IPsec data traffic initiated from clients behind IPsec initiator router goes through the IPsec tunnel and reaches the anchored IPsec tunnel on the SRX Series Firewalls. Clear-text packets coming out of tunnel are routed towards the TRUST direction to reach the server.
- IPsec data reply traffic from server towards client reaches the MX Series Router on the TRUST direction and then gets routed through unique ARI route to the SRX Series Firewalls where tunnel is anchored.
- SRX Series Firewalls encrypt the traffic and send the traffic over the tunnel to the IPsec initiator and then to the client.
- When any SRX Series Firewalls goes down, CHASH on the MX Series Router ensures that IPsec sessions on the other SRX Series Firewalls are not disturbed and only IPsec sessions on the down SRX Series Firewalls are redistributed.

Traffic Load Balancer Overview

This feature relates to topology 2 (single MX Series Router, scale-out SRX MNHA pairs) and topology 3 (dual MX Series Routers and scale-out SRX MNHA pairs).

Figure 14: Topologies 2 and 3 – TLB – IPsec Use Cases

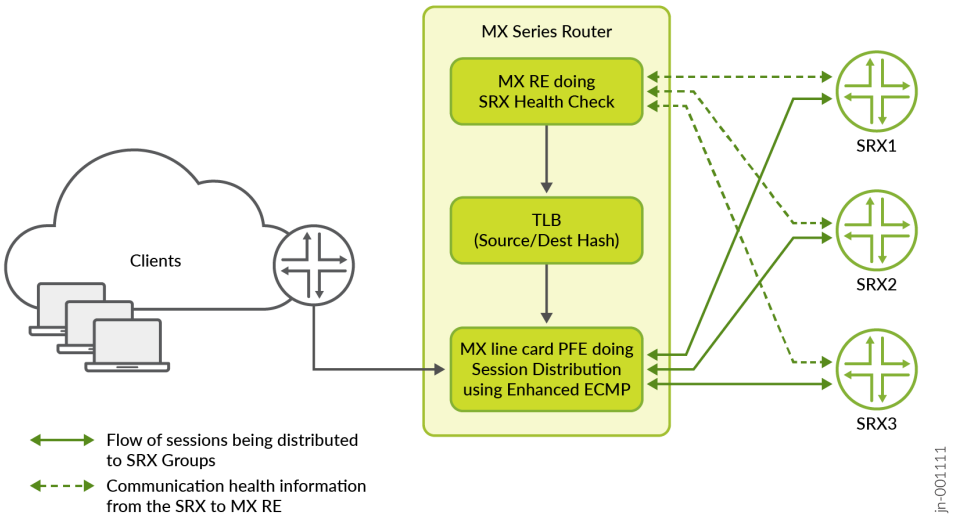


Traffic Load Balancer in MX Series Router

Traffic Load Balancer (TLB) functionality provides stateless translated or non-translated traffic load balancer, as an inline PFE service in the MX Series Routers. Load balancing in this context is a method where incoming transit traffic is distributed across configured servers that are in service. This is a stateless load balancer, as there is no state created for any connection, and so there are no scaling limitations. Throughput can be close to line rate. TLB has two modes of load balancing i.e., translated (L3) and non-translated Direct Server Return (L3).

For the scale-out solution, the TLB mode non-translated Direct Server Return (L3) is used. As part of TLB configuration, there is a list of available SRX Series Firewalls addresses and the MX Series Router PFE programs a selector table based on this SRX Series Firewalls. TLB does a health check (ICMP usually however it can do HTTP, Custom, and TCP checks) for each of the SRX Series Firewalls individually. TLB health check is done using MX Series Router routing engine. If the SRX Series Firewalls pass the health check, TLB installs a specific IP address route or wild card IP address (TLB config option) route in the routing table with next-hop as composite next-hop. Composite next-hop in the PFE is programmed with all the available SRX Series Firewalls in the selector table. Filter based forwarding is used to push the "Client to Server" traffic to the TLB where it hits the TLB installed specific IP address route or wild card IP address route to get the traffic sprayed across the available SRX Series Firewalls with source or destination hash. "Server to Client" is directly routed back to client instead of going through the TLB.

Figure 15: TLB Work in RE and PFE



NOTE: TLB has been used in the Junos OS and MX Series Routers family for a few years now (as early as Junos OS Release 16.1R6) and you are using it successfully on large server farms with around 20,000 servers.

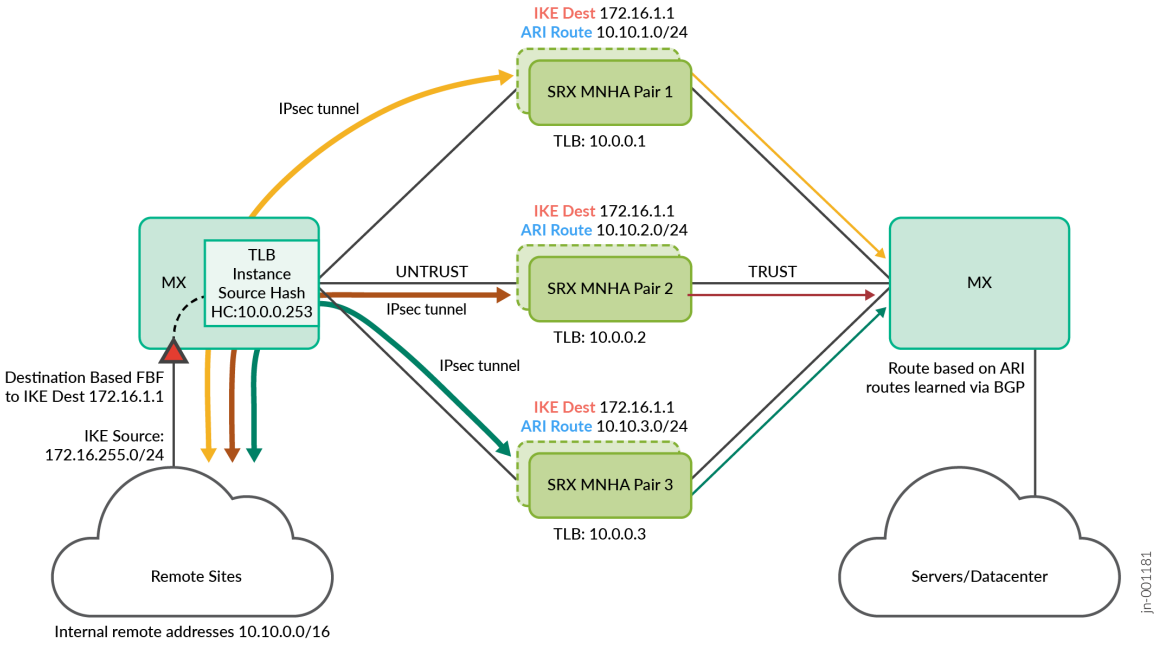
TLB uses the control part and the health check on MS-MPC or MX-SPC3 service cards on MX240/480/960 and MX2000 chassis before data plane or PFE is already on the line cards. It is not running on the RE as it is implemented on MX304/MX10000 chassis.

For more information see, https://www.juniper.net/documentation/us/en/software/junos/interfaces-next-gen-services/interfaces-adaptive-services/topics/concept/tdf-tlb-overview.html

Using TLB in MX Series Router for Scale-Out SRX Series Firewalls Solution with IPsec

In this scenario, the source of IPsec traffic is some remote sites (for an Enterprise or remote users) that reside on the left side of Figure 16 on page 32. When this remote site connects using IPsec to the SRX Series Firewalls, it is redirected by load balancing to one of the SRX Series Firewalls as TLB handles it. It can be represented the other way around for an enterprise, however, the principle stays the same, only interface IP address routing-instance and zone naming might change. A unique anycast IP address is used for all IKE/IPsec connections, hosted on each SRX Series Firewalls.

Figure 16: Topology 3 - Scale-Out IPsec with TLB



- All SRX Series Firewalls are configured with BGP to establish an eBGP peering sessions with MX Series Router-nodes.
- All the scale-out SRX Series Firewalls need to be configured with auto-vpn config and with the same anycast IP address as IKE endpoint IP address. All SRX Series Firewalls are an IPsec responder only mode.
- IPsec clients getting initiated behind MX Series Routers use the same SRX IKE endpoint IP address with unique traffic-selectors. This traffic-selector is used by SRX Series Firewalls to install unique ARI routes (Auto Route Injection) to attract the data return traffic to the right IPsec tunnel from the server. The ARI routes need to be unique also.
- MX Series Routers are configured with TLB on the IPsec VR routing instance to do the load balancing of IKE traffic coming from MX Series Router towards scale-out SRX Series Firewalls.
- All the scale-out SRX Series Firewalls connected to MX Series Routers are configured with unique IP addresses, which is used by MX TLB to do the health check and build up the selector table in the PFE. PFE uses this selector table to load balance the packet across the available next hops. This health check is reachable through BGP connection. Anycast IP address used for IKE endpoint is reachable through this Unique IP address on each SRX Series Firewalls.
- Filter based forwarding based on source IP address match is used in MX Series Router to push IPsec specific traffic to the TLB IPsec forwarding instance.

- TLB Forwarding instance has a default route with the next hop as a list of SRX Series Firewalls. TLB installs this default route when its health check passes with at least one SRX Series Firewalls.
- TLB does source-based hash load balancing across all the available SRX Series Firewalls next-hop devices.
- Load balanced IPsec tunnel sessions get anchored on any available SRX Series Firewalls and it installs the ARI route. Then packet gets decrypted, and it's routed to reach the server through MX Series Router over TRUST routing instance.

For the return traffic coming from server to client direction on the MX TRUST routing instance, Unique ARI routes are used to route the traffic back to same SRX Series Firewalls where the IPsec tunnel is anchored.

- SRX Series Firewalls use the same IPsec tunnel session to encrypt the packet and route the IPsec traffic towards MX Series Router on the UNTRUST VR direction.
- MX Series Router routes the IPsec traffic back to IPsec Initiators.

Configuration Example for ECMP CHASH

The following sample configurations are proposed to understand the elements making this solution work, including configurations for both MX Series Router and some SRX Series Firewalls. It contains a lot of repetitive statements. It shows Junos OS hierarchical view.

Source-hash for forward flow is common for all ECMP based solutions or TLB based solutions. CHASH is used during any next-hop failure where it helps an existing session on an active next-hop to remain undisturbed, while sessions on down next-hop is redistributed over other active next-hop. This CHASH behavior is pre-built in the TLB solution. However, in ECMP based solution you must configure this CHASH configuration explicitly using BGP import policy.

NOTE: The following sample configuration examples consider the "publicly" announced IKE Gateway address as 172.16.1.1/32 (though part of private space RFC1918) for the sake of the demonstration, as well as the remote site "public" IP addresses in the 172.16.255.0/24 range. All other 10.0.0.0/8 addresses are considered private addresses as per the same RFC.

The following MX Series Router configuration is an example for ECMP load balancing using source hash on the **UNTRUST** side (only to IKE gateway unicast address shared by each SRX Series Firewalls):

```
### MX sample routing configuration:
policy-options {
```

```
    prefix-list ipsec_sites_v4 {
        172.16.255.0/24;                        ### source IPsec remote sites
    }
    prefix-list IPsecGW_v4 {
        172.16.1.1/32;                  ## same target IPsec gateway on each SRX(s)
    }
    policy-statement pfe_lb_hash {
        term source_hash {
            from {
                 prefix-list-filter IPsecGW_v4 exact; ### target IPsec SRX gateway
            }
            then {
                load-balance source-ip-only;       ### when match, then LB per src-ip
                accept;
            }
        }
        term ALL-ELSE {
            then {
                load-balance per-packet;                ### per packet for anything else
                accept;
            }
        }
    }
}
routing-options {
    forwarding-table {
        export pfe_lb_hash;
    }
}
```

The following MX Series Router configuration is an example for specific forward traffic with ECMP CHASH on the UNTRUST side (on the IPsec encrypted traffic side):

```
### MX sample UNTRUST configuration:
policy-options {
    policy-statement pfe_consistent_hash {
        from {
            prefix-list-filter IPsecGW_v4 exact;   ### The same IKE target on each SRX
        }
        then {
            load-balance consistent-hash;              ### Load Balancing mechanism
            accept;
```

```
        }
    }
    policy-statement untrust-to-trust-export {   ### Export external routes to SRX(s)
        term 1 {
            from {
                protocol [ bgp static ];
                prefix-list-filter ipsec_sites_v4;   ### The source IKE/IPsec
            }
            then {
                next-hop self;
                accept;
            }
        }
        term 2 {
            then reject;
        }
    }
}
routing-instances {
    UNTRUST_VR {
        instance-type virtual-router;
        routing-options {
            autonomous-system 65550;
        }
        protocols {
            bgp {
                group MX-TO-GATEWAY {   ### BGP Peering with external gateway
                    type external;
                    peer-as 65551;
                    local-as 65550;
                    ...
                }
                group MX-TO-SRXS {              ### BGP Peering with all SRX (untrust)
                    type external;
                    import pfe_consistent_hash;  ### apply LB CHASH toward SRX(s)
                    export untrust-to-trust-export; ### Remote sites routes to SRX(s)
                    peer-as 64500;
                    local-as 65550;
                    multipath;
                    bfd-liveness-detection {
                        minimum-interval 300;
                        minimum-receive-interval 300;
                        multiplier 3;
```

```
                    }
                    neighbor 10.1.1.2;                  ### PEERING WITH SRX1
                    neighbor 10.1.1.10;                         ### PEERING WITH SRX2
                    ...                                         ### ANY OTHER SRX/VSRX
            }
        }
        interface ae...;
        ...
    }
}
```

The following MX Series Router configuration is an example for specific forward traffic on the **TRUST** side with decrypted mobile traffic (only remote sites negotiated IP address coming from the Auto-Route-Injection Traffic Selectors, ARI-TS, need to be announced):

```
### MX sample TRUST configuration:
policy-options {
    prefix-list remote_sites_v4 {
        10.10.0.0/16;                   ### remote site internal subnets
    }
    policy-statement srx_ari_route_export{ ### Export remote routes from SRX(s) to MX
        term 1 {
            from {
                protocol bgp;           ### Remotes sites learned via SRX
                prefix-list-filter remote_sites_v4 orlonger;
            }
            then {
                next-hop self;
                accept;
            }
        }
        term 2 {
            then reject;
        }
    }
    policy-statement trust-to-untrust-export {  ### Export internal routes to SRX(s)
        term 1 {
            from protocol [ bgp static ];
            then {
                next-hop self;
                accept;
            }
```

```
        }
        term 2 {
            then reject;
        }
    }
}
routing-instances {
    TRUST_VR {
        instance-type virtual-router;
        routing-options {
            autonomous-system 65536;
        }
        protocols {
            bgp {
                group MX-TO-MX-IBGP {   ### BGP Peering with internal gateway
                    type internal;
                    export srx_ari_route_export;        ### Export remote sites to GW
                    ...
                }
                group MX-TO-SRXS {              ### BGP Peering with all SRX (trust)
                    type external;
                    export trust-to-untrust-export; ### Export internal to SRX(s)
                    peer-as 64500;
                    local-as 65536;
                    multipath;
                    bfd-liveness-detection {
                        minimum-interval 300;
                        minimum-receive-interval 300;
                        multiplier 3;
                    }
                    neighbor 10.1.1.0;                  ### PEERING WITH SRX1
                    neighbor 10.1.1.8;                  ### PEERING WITH SRX2
                    ...                                         ### ANY OTHER SRX/VSRX
                }
            }
        }
        interface ae1.0;
        interface ae2.20;
        ...
    }
}
```

After reviewing the MX Series Router configuration, consider the following example showing SRX1 configuration for IPsec on the **UNTRUST** side (includes security zone). Very similar configuration applies to all next SRX Series Firewalls, including **same IKE Loopback address** and **same BGP AS number**, however, different IP address for their own network addresses.

```
### SRX sample UNTRUST configuration:
policy-options {
    policy-statement ike_endpoint_export_policy {
        term 1 {
            from {
                protocol direct;
                route-filter 172.16.1.1/32 exact;       ### SRX loopback IKE target IP
            }
            then {
                next-hop self;
                accept;
            }
        }
        term 2 {
            then reject;
        }
    }
}
routing-instances {
    VR-1 {
        instance-type virtual-router;
        protocols {
            bgp {
                group srx-to-mx1_UNTRUST {
                    type external;
                    export ike_endpoint_export_policy;  ### announces IKE Gateway to MX
                    local-as 64500;
                    bfd-liveness-detection {
                        minimum-interval 300;
                        minimum-receive-interval 300;
                        multiplier 3;
                    }
                    neighbor 10.1.1.3 {
                        local-address 10.1.1.2;
                        peer-as 65550;
                    }
                }
```

```
            }
        }
        interface ae1.1;                        ### Interface assigned to UNTRUST zone
        interface lo0.0;                        ### Loopback interface used as IKE Gateway
    }
}
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 172.16.1.1/32;  ### Loopback IP used as IKE Gateway
                address 172.16.0.101/32;        ### Loopback IP used as healthcheck on SRX1
               # address 172.16.0.102/32;       ### Loopback IP used as healthcheck on SRX2
               # address 172.16.0.103/32;       ### Loopback IP used as healthcheck on SRX3
            }
        }
    }
}
security {
    zones {
        security-zone untrust {
            interfaces {
                ae1.1 {
                    host-inbound-traffic {
                        system-services {
                            ping;
                        }
                        protocols {
                            bgp;
                            bfd;
                        }
                    }
                }
                lo0.0 {
                    host-inbound-traffic {
                        system-services {
                            ping;
                            ike;                ### Loopback terminating IKE/IPsec
                        }
                        protocols {
                            bgp;
                            bfd;
                        }
```

```
                    }
                }
            }
        }
    }
}
```

The following example shows SRX1 configuration for security gateway (referred to as SECGW in code) on the **TRUST** side (using the single and same VR as above):

```
### SRX sample TRUST configuration:
policy-options {
    policy-statement ari_export_trust {
        term 1 {
            from {
                protocol ari-ts;        ### Auto Route Injection from IPsec negociations
                route-filter 10.10.0.0/16 orlonger; ### SRX announce remote sites
            }
            then accept;
        }
        term 2 {
            then reject;
        }
    }
}
routing-instances {
    VR-1 {
        instance-type virtual-router;
        protocols {
            bgp {
                group srx-to-mx1_TRUST {
                    type external;
                    export ari_export_trust;    ### announces remote sites to MX
                    local-as 64500;
                    bfd-liveness-detection {
                        minimum-interval 300;
                        minimum-receive-interval 300;
                        multiplier 3;
                    }
                    neighbor 10.1.1.1 {
                        local-address 10.1.1.0;
                        peer-as 65536;
```

```
                    }
                }
            }
        }
        interface ae1.0;                        ### Interface assigned to TRUST zone
        interface st0.0;                        ### Tunnel interface from IPsec tunnel
    }
}
interfaces {
    st0 {
        unit 0 {
            family inet;
        }
    }
}
security {
    zones {
        security-zone trust {
            interfaces {
                ae1.0 {
                    host-inbound-traffic {
                        system-services {
                            ping;
                        }
                        protocols {
                            bgp;
                            bfd;
                        }
                    }
                }
                st0.0;                  ### Tunnel interface from IPsec tunnel
            }
        }
     }
}
```

The following example shows SRX1 configuration for security gateway at the security level (IKE/IPsec listening settings – example with PSK here - and security policies):

```
### SRX sample IKE/IPsec configuration:
security {
    ike {
```

```
        proposal IKE_PROP {
            authentication-method pre-shared-keys;    ### PSK example and could be PKI
            dh-group group2;
            authentication-algorithm sha1;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 3600;
        }
        policy IKE_POLICY {
            proposals IKE_PROP;
            pre-shared-key ascii-text "###someverylongsecretkeyhere"; ## SECRET-DATA
        }
        gateway avpn_ike_gw {
            ike-policy IKE_POLICY;
            dynamic {
                hostname .juniper.net;
                ike-user-type group-ike-id;             ### Shared IKE id with peers
            }
            dead-peer-detection {
                probe-idle-tunnel;
                interval 10;
                threshold 3;
            }
            local-identity hostname srx.juniper.net;
            external-interface lo0.0;           ### Loopback used for IKE/IPsec
            local-address 172.16.1.1;
            version v2-only;
        }
    }
    ipsec {
        proposal IPSEC_PROP {
            protocol esp;
            encryption-algorithm aes-256-gcm;
            lifetime-seconds 3600;
        }
        policy IPSEC_POLICY {
            proposals IPSEC_PROP;
        }
        vpn avpn_ipsec_vpn {
            bind-interface st0.0;
            ike {
                gateway avpn_ike_gw;
                ipsec-policy IPSEC_POLICY;
            }
```

```
            traffic-selector ts {
                local-ip 0.0.0.0/0;
                remote-ip 0.0.0.0/0;
            }
        }
        anti-replay-window-size 512;
    }
    address-book {
        global {
            address IPsecGW 172.16.1.1/32;              ### IPsecGW address
            address RemoteIKE 172.16.255.0/24;  ### IPsec Remote site subnet
            address remote_sites 10.10.0.0/16;  ### Internal remote sites subnet
            address datacenter 10.0.0.0/8;              ### Internal datacenter
        }
    }
    policies {
        from-zone untrust to-zone untrust {             ### permit IKE/IPsec to IPsecGW
            policy incoming-vpn {
                match {
                    source-address RemoteIKE ;
                    destination-address IPsecGW;
                    application any;
                }
                then {
                    permit;                             ### permit and log
                    log {
                        session-close;
                    }
                }
            }
        from-zone trust to-zone trust {                 ### inbound for remote sites
            policy t2u-permit {
                match {
                    source-address remote_sites;                ### remote sites
                    destination-address datacenter;     ### to datacenter
                    application any;
                }
                then {
                    permit;
                    log {
                        session-close;
                    }
                }
```

```
            }
        }
        default-policy {
            deny-all;
        }
        pre-id-default-policy {
            then {
                log {
                    session-close;
                }
            }
        }
    }
}
```

NOTE: These configurations can also use IPv6.

When running tests, some ECMP CHASH outputs can show the route selections. Notice the IKE anycast IP address for the gateway through different BGP peers on the UNTRUST side:

```
user@MX> show route table untrust-vr.inet.0 172.16.1.1/32 active-path
          TRUST_VR.inet.0: 12 destinations, 14 routes (12 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              172.16.1.1/32       *[BGP/170] 03:14:10, localpref 100
                                   AS path: 64500 I, validation-state: unverified
                                    to 10.1.1.2 via ae1.1
                                  >  to 10.1.1.10 via ae2.1
                                     to 10.1.1.18 via ae3.1
```

And the inner IP address coming out of the IPsec tunnels (allocated to each connected mobile, then showing /32) announced to the TRUST router:

```
user@MX> show route table trust-vr.inet.0 10.10.0.0/16
        UNTRUST_VR.inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              10.10.3.0/24       *[BGP/170] 03:13:30, MED 5, localpref 100
                                   AS path: 64500 I, validation-state: unverified
                                  >  to 10.1.1.18 via ae3.0
              10.10.2.0/24       *[BGP/170] 03:13:31, MED 5, localpref 100
```

```
                              AS path: 64500 I, validation-state: unverified
                            >  to 10.1.1.10 via ae2.0
             10.10.1.0/24       *[BGP/170] 02:12:57, MED 5, localpref 100
                              AS path: 64500 I, validation-state: unverified
                            >  to 10.1.1.2 via ae1.0
```

NOTE: This configuration is also available in the CSDS configuration example as this uses the exact same technology and configuration for the ECMP CHASH. For more information, see https://www.juniper.net/documentation/us/en/software/connected-security-distributed-services/csds-deploy/topics/example/configure-csds-ecmp-chash-singlemx-standalonesrx-scaledout-nat-statefulfw.html (some IP or AS might have changed).

Configuration Example for TLB

Like ECMP CHASH, TRUST-VR/UNTRUST-VR are similar in the TLB use case, with BGP peering with the SRX Series Firewalls on each side, however, different configuration is needed for the TLB services, including additional routing-instances and less policy statements.

Source-hash for forward flow and destination-hash for reverse flow is common for all the solutions based on ECMP or TLB. CHASH is used during any next-hop failures where it helps an existing session on active next-hops not to get disturbed and sessions only on down next-hops gets re-distributed over other active next-hops. This CHASH behavior is pre-built in the TLB solution.

General load balancing strategy for anything except TLB:

```
### MX sample configuration:
system {                                        ### internal services needed for TLB
    processes {
        sdk-service enable;
    }
}
policy-options {
    prefix-list clients_v4 {
        172.16.255.0/24;                        ### source IPsec remote sites
    }
    prefix-list IPsecGW_v4 {
        172.16.1.1/32;                  ### target IPsec gateway on SRX(s)
```

```
    }
}
```

The following MX Series Router configuration is an example for specific forward and return traffic. TLB uses forwarding as a new routing-instance type.

```
### MX sample ROUTING-INSTANCES configuration:
routing-instances {
    UNTRUST_VR {
        instance-type virtual-router;
        ### BGP Peering with next router toward IPsec remote sites
        ### BGP Peering with each SRX on the UNTRUST side (similar to ECMP CHASH)
        interface ae...;
        interface ...;
        interface lo0.0;        ### Used for TLB health check toward SRX(s)
    }
    TRUST_VR {
        instance-type virtual-router;
        ### BGP Peering with next router toward inside
        ### BGP Peering with each SRX on the TRUST side (similar to ECMP CHASH)
        interface ae...;
        interface ...;
    }
    srx-tproxy-fi {             ### additional forwarding instance redirecting to TLB
        instance-type forwarding;
    }
}
```

The following configuration example shows how traffic is redirected to TLB instance using Filter Based Forwarding (associated with routing-instance srx-tproxy-fi) to extract that specific traffic for load balancing it to each SRX Series Firewalls:

```
### MX sample Filter configuration:
firewall {
    family inet {
        filter IPSEC_LB {                       ### The FBF to redirect traffics to TLB
            term IPSEC {
                from {
                    destination-address {
                        172.16.1.1/32;          ### when going to IKE gateway
                    }
```

```
                }
                then {
                    count ipsec_tlb_traffic;
                    routing-instance srx-tproxy-fi; ### then forwarding instance used
                }
            }
            term other_traffic {
                then {
                    count other_traffic;
                    accept;
                }
            }
        }
    }
}
interfaces {
    ### Aggregate and vlan tagging used on AE (not shown here)
    ae1 {
        unit 80 {
            family inet {
                filter {
                    input IPSEC_LB;             ### Where incoming FBF filter is applied
                }
                address 10.80.1.2/31;   ### Interface facing external GW
            }
        }
    }
}
```

TLB uses the following interface loopbacks for health checking to the SRX Series Firewalls:

```
### MX sample loopback configuration:
interfaces {
    lo0 {
        unit 0 {
            ip-address-owner service-plane; ### not for RE-TLB but used on other MX
            description "TLB Health-Check Source IP Address for UNTRUST VR";
            family inet {
                address 172.16.1.253/32;
            }
        }
```

```
    }
}
```

And the TLB service part (Example, with the IPsec service, only TRUST side TLB instance is used as ARI route, which is announced for return traffic):

```
### MX sample TLB configuration:
services {
    traffic-load-balance {
        routing-engine-mode;                    ### Important for MX304/MX10K to enable TLB
        instance ipsec_lb {                     ### TLB instance for IPsec traffics
            interface lo0.0;
            client-vrf UNTRUST_VR;
            server-vrf UNTRUST_VR;
            group mnha_srx_group {
                real-services [ MNHA_SRX1 MNHA_SRX2 ];  ### selected SRXs in TLB group
                routing-instance UNTRUST_VR;
                health-check-interface-subunit 0;
                network-monitoring-profile icmp-profile;
            }
            real-service MNHA_SRX1 {
                address 172.16.0.101;   ### address used on SRX1 or MNHA pair1
            }
            real-service MNHA_SRX2 {
                address 172.16.0.102;   ### address used on SRX2 or MNHA pair2
            }
            ...
            virtual-service srx_untrust_vs {
                mode direct-server-return;
                address 172.16.1.1;
                routing-instance srx-tproxy-fi;  ### Using routes from this VR
                group mnha_srx_group;        ### and sending them to that TLB group
                load-balance-method {
                    hash {
                        hash-key {
                            source-ip;       ### using source-ip as hash
                        }
                    }
                }
            }
        }
    network-monitoring {               ### monitor via icmp, http, tcp, udp, ssl/tls…
```

```
        profile icmp-profile {
            icmp;
            probe-interval 1;
            failure-retries 5;
            recovery-retries 1;
        }
    }
}
```

After MX Series Router configuration, the following sample SRX1 configuration is for IPsec security gateway.

In case of SRX MNHA pair, same loopback IP address is shared to failover in case of any event on the active device. This specific loopback IKE gateway IP address is announced by BGP to the MX Series Router peer (on TRUST side). The following example shows SRX1 configuration for MNHA and loopback export:

```
### SRX sample MNHA configuration:
chassis {
    high-availability {
        local-id {
            1;
            local-ip 10.2.0.1;
        }
        peer-id 2 {
            peer-ip 10.2.0.2;
            interface lo0.0;
            liveness-detection {
                minimum-interval 1000;
                multiplier 3;
            }
        }
        services-redundancy-group 0 {
            peer-id {
                2;
            }
        }
        services-redundancy-group 1 {
            deployment-type routing;    ### Full routing mode with BGP peers
            peer-id {
                2;
            }
```

```
            activeness-probe {
                dest-ip {
                    10.1.1.1;
                    src-ip 10.10.10.1;
                }
            }
            monitor {
                bfd-liveliness 10.1.1.1 {
                    src-ip 10.1.1.0;
                    session-type singlehop;
                    interface ge-0/0/1.0;
                }
            }
            active-signal-route {               ### Used to announce Active state
                10.2.2.1;
                routing-instance MNHA-VR;
            }
            backup-signal-route {               ### Used to announce Backup state
                10.4.4.1;
                routing-instance MNHA-VR;
            }
            prefix-list ike_lo0;                ### Announces this IKE prefix when Active
            managed-services ipsec;
            preemption;
            activeness-priority 200;
        }
    }
}
policy-options {
    prefix-list ike_lo0 {                       ### Loopback address for IKE gateway
        172.16.1.1/32;                  ### MNHA will announce it when active
    }
    prefix-list active_probe_ip {               ### Loopback address for TLB healthchecks
        172.16.0.101/32;                        ### lo0 for each SRX 101, 102, 103…
    }
    policy-statement ari_export_untrust {
        term 1 {
            from {
                protocol ari-ts;
                condition active_route_exists;
            }
            then accept;                        ### Announce ARI routes via current AS
        }
```

```
        term 2 {
            from {
                protocol ari-ts;
                condition backup_route_exists;
            }
            then {
                as-path-prepend 64500;  ### Announce ARI routes with prepanded AS
                accept;
            }
        }
        term default {
            then reject;
        }
    }
    policy-statement loopback_export_trust {
        term 1 {
            from {
                prefix-list active_probe_ip; ### Announce loopbacks conditionnally
                condition active_route_exists;
            }
            then accept;
        }
        term 2 {
            from {
                prefix-list active_probe_ip; ### Announce loopbacks with prepanded AS
                condition backup_route_exists;
            }
            then {
                as-path-prepend 64500;
                accept;
            }
        }
        term default {
            then reject;
        }
    }
    condition active_route_exists {                    ### Used to test Active state
        if-route-exists {
            address-family {
                inet {
                    10.2.2.1/32;
                    table MNHA-VR.inet.0;
                }
```

```
            }
        }
    }
    condition backup_route_exists {                    ### Used to test Backup state
        if-route-exists {
            address-family {
                inet {
                    10.4.4.1/32;
                    table MNHA-VR.inet.0;
                }
            }
        }
    }
}
routing-instances {
    MNHA-VR {
        instance-type virtual-router;
    }
}
```

When running the tests, some TLB is seen as the group usage and packets/bytes to each SRX Series Firewalls:

```
### MX sample TLB statistics:
user@MX> show services traffic-load-balance statistics instance srx-tproxy-fi
Traffic load balance instance name    :  ipsec_lb
Multi services interface name         :  lo0.0
Interface state                       :  UP
Interface type                        :  Multi services
Route hold timer                      :  180
Active real service count             :  2
Total real service count              :  2
Traffic load balance virtual svc name :  mnha_srx_vip
IP address                            :  172.16.1.1
Virtual service mode                  :  Direct Server Return mode
Routing instance name                 :  srx-tproxy-fi
Traffic load balance group name       :  mnha_srx_group
Health check interface subunit        :  0
Demux Nexthop index                   :  N/A (612)
Nexthop index                         :  613
Up time                               :  05:58:01
Total packet sent count               :  1919281813
```

```
Total byte sent count                 :  1151527984611
Real service     Address       Sts  Packet Sent  Byte Sent     Packet Recv    Byte Recv
MNHA_SRX1        172.16.0.101       UP  947865774    731089457241
MNHA_SRX2        172.16.0.102       UP  9244758620    502478110497
```

Common Configurations for ECMP CHASH and TLB

Some elements of configuration need to be in place for both load balancing methods. The following sample configurations are for TRUST and UNTRUST VR and the peering with each SRX Series Firewalls. It also shows some other less seen configuration elements.

The following sample shows a common configuration when using dual MX Series Router topology: Both MX Series Router calculate the same hash value when both have same number of next hops.

```
forwarding-options {
    enhanced-hash-key {
        symmetric;
    }}
```

Results Summary and Analysis

All the test results are summarized in different documents detailing all aspects of the testing. This JVD shows that scale-out can leverage the use of important functions both on the MX Series Routers and SRX Series Firewalls for their respective target usage:

- The MX Series Router is used as a load balancer with different options, ECMP CHASH and TLB.
- The SRX Series Firewalls are used as an IPsec security service with simple integration with the MX Series Router.
- Both physical SRX Series Firewalls and virtual SRX Series Firewalls are used the same way.
- Simple network integration using BGP and BFD helps in fast convergence time.
- Though no scale is tested, the simplicity of adding a new service node shows that this architecture can help to scale in many directions (performances, scaling, and so on) by simply adding new service node without disturbing the global service.

ECMP CHASH shows steady restoration (time in milliseconds.)

NOTE: With ECMP, all the SRX Series Firewalls need to be of the same model, whereas with TLB, it is not mandatory to have same devices, for example some SRX Series Firewalls in an IPsec group and other vSRX Series Firewalls in different IPsec groups. The number of groups is around 2,000 per MX Series Router and the number of SRX Series Firewalls members are around 256.

With TLB being used mainly on MX Series Router platforms, it also works with non-tested MX Series Router models, where TLB uses a control function on the RE (like MX304) or on a service card (for example, MS-MPC for MX240). TLB has been in Junos OS since Junos OS Release 18.1R1 when BGP acquired multipath function. This connection with BGP offers a good solution for service providers who often use it internally and externally.

TLB use case works with fast restoration timers and shows more flexibility in deployment options (aka single or dual MX Series Routers), as well as a better handling of SRX Series Firewalls in the MNHA cluster.

SRX Series Firewalls features leveraged in this JVD focus on IPsec security gateway, leveraging easy central gateway with standard security protocols like AES-GCM (Advanced Encryption Standard in Galois Counter Mode using 128 bits or 256 bits key length). However, any other encryption mode can also be used. AES-GCM became the preferred de facto standard for the security industry. One can add any method to it from IKEv1 and IKEv2, using Pre shared Keys or PKI (digital certificates). A SRX Series Firewall, and moreover multiple scale-out SRX Series Firewalls are used to terminate multiple tunnels with various methods at the same time with many other remote sites or remote users.

SRX Series Firewalls are used as a terminating, and not initiating, all the IPsec VPN from remote entities as it listens and wait for remote entities -- that all have the ability to contact the gateway (like from an initial DNS request) – to start the IKE negotiation to negotiate an IPsec VPN. This applies to all the SRX Series Firewalls in that group listening on the same configuration that needs a very simple and single IKE/IPsec entry in its configuration. The opposite is complex to contact from those SRX Series Firewalls to many remote sites, as it requires a specific configuration for each remote entity (configuration multiplied by the number of remote sites) and more complex to tell to a single SRX Series Firewall to start the IKE/IPsec negotiation outbound (and not all of them); these firewalls cannot share the same IKE/IPsec configuration, then removing the interest of scaling out.

NOTE: SRX Series Firewalls are always performing a stateful firewall in addition to IPsec for both its IPsec headers. However, for the tunneled traffic that is encrypted/decrypted, it uses same Layer 4 to Layer 7 protocol and application firewall, if this is set to be checked in the security policies configuration. This document is at the IKE/IPsec configuration level. However, it does

not mention some security policies for the received (and decrypted) traffic coming through IPsec from the remote sites.

The scale-out solution is considered as an alternative to the monolithic scale-up approach. It uses the chassis based SRX Series Firewalls or security services on MX240/480/960 with MX-SPC3 service cards independently. However, nothing prevents such architectures from benefitting from both leverage possibilities to add new services and the power of those existing platforms. The existing smaller platforms like the MX304 and SRX4600 help to create smaller footprint architectures.

On the management front, automation is used to build and test the solution with the various use cases and tests. In summary, scripting is used with Junos OS access using Netconf. Lots of scripting already exists in the field (or Juniper automation places like GitHub) using Ansible, Terraform, Python, PyEZ (Python Easy for Junos OS), etc. Some advanced users have scripted Junos OS, and API that are available to integrate with the existing management framework.

The Security Director (on-prem or cloud) has an important place for delivering common configuration to the security service layer (like security policies, address objects, NAT pools, etc.). This gives visibility to the security events and logs generated by each SRX Series Firewalls.

Junos OS integration with BGP (peering between the MX Series Router and the SRX Series Firewalls, including the right BFD timers) allows you to create a matching environment with Juniper solutions working seamlessly together. The redundancy of each router and security solution allows you to maintain steady traffic while providing addition of new capacities in a simple way. Similar configuration statements on both routers (MX Series Router) and security (SRX Series Firewalls) provide a simple and seamless management of this solution.

Additional Resources

- Service Redundancy Daemon (SRD) https://www.juniper.net/documentation/us/en/software/junos/interfaces-adaptive-services/topics/topic-map/service-redundancy-daemon.html
- Equal-Cost Multipath (ECMP) https://www.juniper.net/documentation/us/en/software/junos/interfaces-ethernet-switches/sampling-forwarding-monitoring/topics/concept/policy-per-packet-load-balancing-overview.html
- Load Balancing Using Source or Destination IP Only https://www.juniper.net/documentation/us/en/software/junos/routing-policy/topics/task/load-balancing-using-src-or-dst-ip-only-configuring.html
- ECMP Consistent Hashing - Consistent Load Balancing for ECMP Groups https://www.juniper.net/documentation/us/en/software/junos/interfaces-ethernet-switches/topics/topic-map/understanding-ecmp-groups.html

- Traffic Load Balancing (TLB) https://www.juniper.net/documentation/us/en/software/junos/interfaces-next-gen-services/interfaces-adaptive-services/topics/concept/tdf-tlb-overview.html
- Junos OS Symmetrical Load Balancing https://community.juniper.net/blogs/moshiko-nayman/2024/06/19/junos-symmetrical-load-balancing
- Multinode High Availability https://www.juniper.net/documentation/us/en/software/junos/high-availability/topics/topic-map/mnha-introduction.html
- IKE and IPsec VPN Overview https://www.juniper.net/documentation/us/en/software/junos/vpn-ipsec/topics/topic-map/security-ipsec-vpn-overview.html
- Connected Security Distributed Services https://www.juniper.net/documentation/us/en/software/connected-security-distributed-services/csds-deploy/topics/concept/csds-overview.html
- ECMP Consistent Hashing with Stateful traffic flow https://www.juniper.net/documentation/us/en/software/connected-security-distributed-services/csds-deploy/topics/concept/csds-ecmp-chash-singlemx-standalonesrx-scaledout-statefulfw.html
- Automation and communities
  https://github.com/orgs/Juniper/repositories?type=all

  https://community.juniper.net/home/techpost

Revision History

Table 4: Revision History

Date	Version	Description
December 2024	MSE-SCALEOUT-IPSEC-ENT-01-01	Initial publish