Juniper Networks | Engineering Simplicity

Junos OS Evolved

AI-ML Data Center Feature Guide

Published
2025-12-19

JUNOS EVOLVED

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

Table of Contents

1

CHAPTER

Overview

IN THIS CHAPTER

AI-ML Data Center Overview

As artificial intelligence (AI) and machine learning (ML) applications expand, the networks supporting these AI-ML applications require increased capacity to handle large data flows. This requirement is particularly true for the data centers that store AI-ML data sets. Junos® OS Evolved offers a set of innovative features for AI-ML data centers. Network administrators can use this guide to learn how to configure these features to optimize operations inside AI-ML data center fabrics.

Generative AI and ML applications such as large language models (LLMs) are based on statistical analysis of data sets: the more often the computational model finds a pattern in the data, the more it reinforces that pattern in its output. Through this repetitive pattern finding, these models are able to accomplish tasks such as convincingly imitating human speech. However, a generative AI application is only as good as the data set it is trained on. The larger the data set, the more patterns the model is able to detect. For this reason, AI and ML applications require large data sets. These data sets are stored in data centers.

To increase the speed of training, AI and ML models are often trained within the data center network through parallel computing. Graphics processing unit (GPUs) are clustered together and hosted on server nodes that are distributed across the data center. Complex computations occur simultaneously on these GPU clusters. The network must synchronize the output from the GPUs within a cluster to create a fully trained model. This synchronization requires the continuous movement of large data flows, henceforth referred to as *elephant flows*, across the back end of the network.

The elephant flows in AI-ML data centers require robust networks. When dealing with elephant flows, an insufficient network quickly encounters problems such as traffic congestion, dropped packets, and link failures. These network problems are especially unacceptable when dealing with data that requires high levels of accuracy. One robust network design ideal for AI-ML data centers is the Rail-Optimized Stripe. This AI cluster architecture minimizes network disruption by moving data to a GPU on the same rail as the destination. An IP Clos architecture is another functional AI-ML data center fabric design.

Juniper Networks® QFX Series Switches running Junos OS Evolved are ideal candidates for both Rail-Optimized Stripe architectures and IP Clos network designs. For example, the QFX5220-32CD, QFX5230-64CD, QFX5240-64OD, and QFX5240-QD switches work well in both network types as leaf, spine, and superspine devices. These switches also function well as a group of leaf-spine switches called a point of distribution (POD). To build larger AI-ML clusters in your data center, you can use a superspine layer to interconnect different PODs. You can deploy these switches as a single POD or multiple PODs for maximum flexibility and network redundancy. In addition, these devices support advanced AI-ML features that solve many load balancing and traffic management problems common in AI-ML data centers.

RELATED DOCUMENTATION

- Juniper Validated Design (JVD): AI Data Center Network with Juniper Apstra, NVIDIA GPUs, and WEKA Storage
- AI Data Center Network with Juniper Apstra, AMD GPUs, and Vast Storage—Juniper Validated Design (JVD)
- AI Data Center Multitenancy with EVPN/VXLAN—Juniper Validated Design (JVD)
- Designing Data Centers for AI Clusters
- AI Data Center Networking

2

CHAPTER

Upgrade

IN THIS CHAPTER

Recommended Release

IN THIS SECTION

Junos OS Evolved Release 23.4X100D20 is the first Junos Evolved release introducing the EVPN-VXLAN multi-tenancy feature-set on QFX5230 and QFX5240-64OD/64QD switches for the AI-ML data center usecase. We recommend approaching the Juniper Networks representative or partner for the latest release recommendation.

For specific features, see Feature Explorer for platform and release support.

Platform Documentation

- QFX5220
- QFX5230-64CD
- QFX5240-64OD and QFX5240-QD

Additional Resources

- Port Checker
- Software Licenses for QFX Series Switches

Unified ISSU for AI-ML Data Centers

SUMMARY

Use unified in-service software upgrade (ISSU) to minimize traffic loss during the software upgrade process.

IN THIS SECTION

Overview

IN THIS SECTION

In an AI-ML data center deployment, the large data flows, also known as *elephant flows*, traveling through the network mean that even a low percentage of lost traffic can be a large number of packets. As the network administrator, you can use the unified in-service software upgrade (unified ISSU) feature to upgrade to a more recent release of Junos OS Evolved with no disruption on the control plane and minimal loss of traffic.

Benefits

- Eliminates network downtime during software image upgrades.
- Reduces operating costs while delivering higher service levels.
- Enables you to implement new features more quickly.

Configuration

IN THIS SECTION

Configuration Overview

When you are planning to perform a unified ISSU, choose a time when your network is as stable as possible. As with a normal upgrade, Telnet sessions, SNMP, and CLI access are briefly interrupted.

We recommend that you read the *Unified ISSU for Junos OS Evolved* topic to anticipate any special circumstances that might affect your upgrade.

For AI-ML data center deployments, the following configurations do not roll over to the upgraded operating system with unified ISSU. You must reconfigure these features after the upgrade is complete:

- Generic routing encapsulation (GRE) tunnels
- sFlow
- Port mirroring
- Multicast Internet Group Management Protocol (IGMP) snooping and Protocol Independent Multicast (PIM)
- Virtual Router Redundancy Protocol (VRRP)
- Link Aggregation Control Protocol (LACP)
- Bidirectional Forwarding Detection (BFD) protocol

Prepare to Run the Upgrade

1. Make sure that you have sufficient disk space for the upgrade and that a backup of the system is available. Save the system configuration and the information about how the system is handling traffic.

   You can do this by following the procedure at *Before You Upgrade or Reinstall Junos OS Evolved*.

You will need the information about the system configuration and how the system is handling traffic when you verify that the upgrade was performed correctly.

2. Download the software package from the Juniper Networks Support website at https://www.juniper.net/support/ and place the package on your local server.
3. If the BGP protocol is configured on the main routing instance or a specific routing instance, then configure BGP graceful restart. Set the restart time value to greater than or equal to 300 seconds.

   NOTE: Changing the restart-time for BGP graceful restart causes the existing BGP sessions to restart, which might cause disruptions. We recommend that you perform this action during a low network usage time to avoid traffic loss.

   Configure the following on the device you are upgrading as well as its BGP peers.

   To configure BGP graceful restart and the restart-time value on the main routing instance, issue the following commands:

```
[edit]
user@host# set routing-options graceful-restart
[edit]
user@host# set protocols bgp graceful-restart restart-time 300
```

   To configure BGP graceful restart and the restart-time value on a specific routing instance, issue the following commands:

```
[edit]
user@host# set routing-instances routing-instance routing-options graceful-restart
[edit]
user@host# set routing-instances routing-instance protocols bgp graceful-restart restart-time
300
```

4. If a Spanning Tree Protocol (STP) is configured, then configure the STP-enabled ports as edge ports and enable bridge protocol data unit (BPDU) protection.

   Depending on the type of STP configured, issue the following commands:

```
[edit]
user@host# set protocols (mstp | rstp | vstp) bpdu-block-on-edge
[edit]
user@host# set protocols (mstp | rstp | vstp) interface (interface-name | all) edge
```

5. Configure the value of the Address Resolution Protocol (ARP) aging timer to the maximum value of 240 minutes. Extending the aging timer to its maximum value gives the device time to upgrade between ARP updates.

```
[edit]
user@host# set system arp aging-timer 240
```

6. On the BGP peers of the device you are upgrading, set the number of ARP retry attempts to 300. If the number of retries is too low, the peer device might stop trying to reconnect before the upgrade is complete.

```
[edit]
user@peer-device# set system arp arp-retries 300
```

7. Copy the software image to the **/var/tmp/** directory of the device running Junos OS Evolved using the `scp` command.

   For example:

```
user@host> file copy scp://junos-evo-install-qfx-ms-x86-64-22.1R1-S1.2-EVO.iso /var/tmp/junos-evo-install-qfx-ms-x86-64-22.1R1-S1.2-EVO.iso
```

8. Validate the existing configuration against the new software image to check whether it supports unified ISSU by using the `request system software validate-restart` *package-name* command.

   For example:

```
user@host> request system software validate-restart /var/tmp/junos-evo-install-qfx-ms-x86-64-22.1R1-S1.2-EVO.iso
```

Run the Upgrade

After you have completed the tasks above, run the `request system software add` *package-name* `restart` command on the device that you want to upgrade.

For example:

```
user@host> request system software add /var/tmp/junos-evo-install-qfx-ms-x86-64-22.1R1-S1.2-EVO.iso restart
```

The system restarts or reboots to load the new software image. When the upgrade is complete, the device displays the login prompt.

Verify the Upgrade Was Successful

1. At the login prompt, log in and verify the release of the installed software, using the `show system software list` command.
2. Verify that the system is running properly and correctly handling traffic by repeating the steps in the procedure in *Before You Upgrade or Reinstall Junos OS Evolved*. Compare the information about the system configuration to what you collected before you installed the software package.
3. If you need to make any changes to the configuration after the upgrade, remember to back up the software and configuration using the `request system snapshot` command. See *Back up and Recover Software with Snapshots*.
4. If the unified ISSU fails for some reason, and if the CLI is still working, you can follow the steps in *Recover from a Failed Installation Attempt If the CLI Is Working* to install the software image.
5. Reconfigure any applicable features listed in the "Configuration Overview" on page 7 above. Commit your changes.

Platform Support

See Feature Explorer for platform and release support.

Related Documentation

- Unified ISSU for Junos OS Evolved

3

CHAPTER

Load Balancing

IN THIS CHAPTER

Load Balancing Overview for AI-ML Data Centers

Significant load balancing challenges arise when AI-ML data centers process elephant flows. If elephant flows are not load-balanced properly across the network, they are likely to cause traffic congestion. When traffic congestion does occur, ineffective load balancing can compound the problem by inadvertently directing traffic to already congested links. Junos OS Evolved offers several types of load-balancing configurations that are optimized for the challenges of elephant flows.

As the network administrator, you can configure three main types of load balancing on your network:

- **Static load balancing (SLB)**—In SLB, you configure certain types of traffic to always use certain links. SLB is the most basic type of load balancing.
- **Dynamic load balancing (DLB)**—DLB dynamically chooses the link for a traffic flow based on the size of the traffic queue and the local link bandwidth utilization. DLB also checks the health of a link before rerouting traffic. DLB is more effective at avoiding traffic congestion than SLB.

  DLB has several modes and types that allow for customization, including:

  - **Selective DLB**—Selectively enable DLB for certain per-packet scenarios and use SLB for others.
  - **Flowlet mode**—In *flowlet mode*, DLB tracks the status of flows using an inactivity timer. When the inactivity timer expires for a particular flow, DLB rechecks whether that link is still optimal for that flow. If the link is no longer optimal, DLB selects a new egress link.
  - **Reactive path rebalancing**—Use this enhancement to DLB to move the traffic to a better quality link even when *flowlet mode* is enabled.
- **Global load balancing (GLB)**—GLB is an improvement on DLB. While DLB takes into account only the local link bandwidth utilization, GLB has visibility into the bandwidth utilization of links at the next-to-next-hop (NNH) level. GLB can reroute traffic flows to avoid traffic congestion farther out in the network than what DLB can detect.

You can use these different load balancing techniques in parallel within your AI-ML data center fabric.

RELATED DOCUMENTATION

Load Balancing for Aggregated Ethernet Interfaces

Managing the Elephant in the Room for AI Data Centers

Load Balancing in the AI Data Center

Selective Dynamic Load Balancing (DLB)

IN THIS SECTION

Selective DLB Overview

IN THIS SECTION

With *selective DLB*, you no longer have to choose between DLB and SLB for all traffic traversing your device. You can configure your preferred DLB mode at the global level, configure a default type of load balancing, and then selectively enable or disable DLB for certain kinds of traffic.

Selective DLB is also useful when very large data flow, also called an elephant flow, encounters links that are too small for the entire data flow. In this scenario, selective DLB can calculate the optimal use of the links' available bandwidth in the data center fabric. When you enable selective per-packet DLB for the elephant flow, the algorithm directs the packets to the best-quality link first. As the link quality changes, the algorithm directs subsequent packets to the next best-quality link.

Use Feature Explorer to confirm platform and release support for specific features.

Benefits

- Improve your network handling of large data flows.
- Use per-packet and per-flow load balancing in the same traffic stream to improve performance.
- Customize load balancing based on any firewall filter match condition.

Selective DLB in AI-ML Data Centers

In AI-ML workloads, the majority of the application traffic uses Remote Direct Memory Access (RDMA) over Converged Ethernet version 2 (RoCEv2) for transport. Dynamic load balancing (DLB) is ideal for achieving efficient load balancing and preventing congestion in RoCEv2 networks. However, static load balancing (SLB) can be more effective for some types of traffic. Selective DLB solves this problem.

You can enable load balancing in two ways: per flow or per packet. Per-flow load balancing has been the most widely used because it handles the largest number of packets at a time. The device classifies packets that have the same 5-tuple packet headers as a single flow. The device gives all packets in the flow the same load balancing treatment. Flow-based load balancing works well for general TCP and UDP traffic because the traffic utilizes all links fairly equally. However, per-packet load balancing can reorder some packets, which can impact performance.

Many AI clusters connect the application to the network through smart network interface cards (SmartNICs) that can handle out-of-order packets. To improve performance, enable per-packet DLB on your network. Then enable DLB for only those endpoint servers that are capable of handling out-of-order packets. Your device looks at the RDMA operation codes (opcodes) in the BTH+ headers of these packets in real time. Using any firewall filter match condition, you can selectively enable or disable DLB based on these opcodes. Other flows continue to use default hash-based load balancing, also known as SLB.

Configuration

IN THIS SECTION

Configuration Overview

You can selectively enable DLB in two ways: disable DLB by default and selectively enable DLB on certain flows, or enable DLB globally and selectively disable DLB. In either case, you'll need to first configure DLB in *per-packet mode*. Per-packet is the DLB mode used wherever DLB is enabled. You cannot configure DLB in per-flow and per-packet mode on the same device at the same time.

This feature is compatible with flowlet mode. You can optionally enable this feature when DLB is configured in flowlet mode.

Topology

In the topology shown in Figure 1 on page 15, DLB is disabled by default. We have enabled DLB selectively on Flow2 in per-packet mode. Table 1 on page 15 summarizes the load balancing configuration on the two flows shown and the results of the load balancing applied on the flows:

Figure 1: Per-Flow and Per-Packet Load Balancing

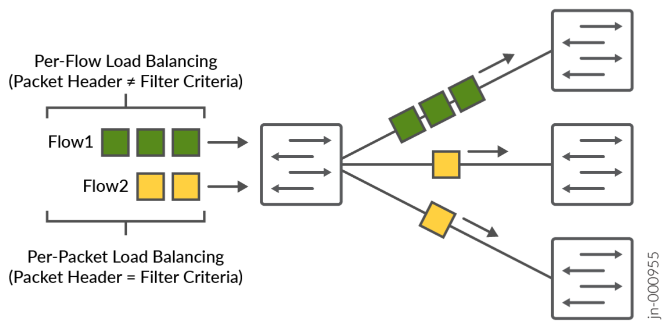


Table 1: Flow Behaviors

Flow	DLB Enabled?	Result
Flow1	No	The device uses the default load balancing configuration, which is per-flow mode. The flow is directed to a single device.
Flow2	Yes	The device uses the DLB configuration, which is per-packet mode. The device splits this flow into packets. DLB assigns each packet to a path that is based on the RDMA opcode in the packet header and the corresponding filter.

Disable DLB Globally and Selectively Enable DLB

In cases where very few packets will require DLB, you can disable DLB at the global level and selectively enable it per flow.

1. Enable DLB per-packet mode. Whenever DLB is enabled on a flow, DLB uses this mode to direct traffic.

```
set system packet-forwarding-options firewall profiles <inet | inet6 | ethernet-switching>
udf-profile-name
set forwarding-options enhanced-hash-key ecmp-dlb per-packet
```

2. Disable DLB globally by turning it off for all Ethernet types. By default, all packets will get hash-based load balancing (SLB).

```
set forwarding-options enhanced-hash-key ecmp-dlb ether-type none
```

3. Configure a firewall filter to match a specific RDMA opcode within the BTH+ header.

   This example matches based on `rdma-opcode` 10.

```
set firewall family inet filter filter-name term term-name from rdma-opcode 10
```

4. Enable per-packet DLB within that firewall filter to only apply DLB to those packets with the chosen RDMA opcode in the BTH+ header.

```
set firewall family inet filter filter-name term term-name then dynamic-load-balance enable
```

5. Other packets get the default load balancing method, which is SLB.

```
set firewall family inet filter filter-name term default then accept
```

Enable DLB Globally and Selectively Disable DLB

In cases where most packets will benefit from DLB, enable DLB at the global level for all packets and selectively disable it per packet.

1. Configure DLB at the global level in per-packet mode for all flows.

```
set system packet-forwarding-options firewall profiles <inet | inet6 | ethernet-switching>
udf-profile-name
set forwarding-options enhanced-hash-key ecmp-dlb per-packet
```

2. Configure a firewall filter to match a specific RDMA opcode within the BTH+ header.

This example matches based on rdma-opcode 10.

```
set firewall family inet filter filter-name term term-name from rdma-opcode 10
```

3. Disable per-packet DLB within that firewall filter for packets with the chosen RDMA opcode in the BTH+ header.

```
set firewall family inet filter filter-name term term-name then dynamic-load-balance disable
```

4. Other packets get the default load balancing method, which is DLB.

```
set firewall family inet filter filter-name term default then accept
```

5. Verify DLB is enabled as you expected using the following commands:

```
show forwarding-options enhanced-hash-key
```

```
show pfe filter hw profile-info
```

Example: Selectively Enable DLB with a Firewall Filter Match Condition

One of the benefits of selective DLB is that you can customize load balancing based on any firewall filter match condition. This example shows how to enable DLB based on a firewall filter that matches with RDMA queue pairs. Use this example to enable per-packet DLB only for those flows terminating on a network interface card (NIC) that supports packet reordering.

In a network that uses RoCEv2 for application traffic transport, an RDMA connection sends traffic on a send queue and receives traffic on a receive queue. These queues form the RDMA connection. Together, the send queue and receive queue are referred to as a queue pair. Each queue pair has an identifiable prefix. In this example, we use queue pair prefixes to control when DLB is enabled.

This example is configured on a QFX5240-64QD switch.

1. Create a user-defined field in a firewall for matching packets that is destined for a specific RDMA destination queue pair. Select a queue pair you know terminates on an NIC that is capable of reordering packets.

We named our firewall filter `sDLB`. The term `QP-match` matches on incoming packets with a destination queue pair with the following characteristics.

```
set firewall family inet filter sDLB term QP-match from flexible-match-range match-start
layer-4
set firewall family inet filter sDLB term QP-match from flexible-match-range byte-offset 13
set firewall family inet filter sDLB term QP-match from flexible-match-range bit-length 24
set firewall family inet filter sDLB term QP-match from flexible-match-range range 0x64
```

2. Configure the firewall filter to enable per-packet DLB on the queue pairs that match the filter.

   If the queue pair is not a match, the device uses the default load balancing type of SLB for that packet.

```
set firewall family inet filter sDLB term QP-match then dynamic-load-balance enable
```

3. Configure a counter that increments each time there is a match.

   The counter `QP-match-count` tracks how many packets were load balanced with DLB. You can use this information when troubleshooting.

```
set firewall family inet filter sDLB term QP-match then count QP-match-count
```

4. Enable your firewall filter on the relevant interface.

```
set interfaces et-0/0/5 unit 0 family inet filter input sDLB
```

5. Verify your firewall filter term is matching on packets coming through the device.

   The `QP-match-count` counter shows the number of bytes and packets that the firewall filter has redirected for load balancing with DLB.

```
user@device> show firewall

Filter: sDLB
Counters:
Name                              Bytes                Packets
QP-match-count               176695488320              552173401
```

Customize Egress Port Link Quality Metrics for DLB

IN THIS SECTION

Overview

IN THIS SECTION

Dynamic load balancing (DLB) selects an optimal link based on the quality of the link so that traffic flows are evenly distributed across your network. You (the network administrator) can customize the way DLB assigns quality metrics of egress ports so that DLB selects the optimal link.

DLB assigns each egress port that is part of equal-cost multipath (ECMP) to a quality band. Quality bands are numbered from 0 through 7, where 0 is the lowest quality and 7 is the highest quality. DLB tracks two metrics on each of the ports, and it uses these metrics to compute the link quality:

- Port load metric: The amount of traffic recently transmitted over each ECMP link, measured in bytes.
- Port queue metric: The amount of traffic enqueued on each ECMP link for transmission, measured in number of cells.

Based on the member port load and queue size, DLB assigns one of the quality bands to the member port. The port-to-quality band mapping changes based on the instantaneous port load and queue size metrics.

By default, DLB weighs the port load metric and port queue metric equally when evaluating link quality. You can configure DLB to base the link quality more heavily on the port load than the port queue, or vice versa. Configure the amount of weight DLB places on the port load using the `rate-weightage` statement at the `[edit forwarding-options enhanced-hash-key ecmp-dlb egress-quantization]` hierarchy level. DLB assigns the remaining weight percentage to the port queue. For example, if you configure the `rate-`

weightage value to be 80, DLB places 80% weight on the port load and 20% weight on the port queue when evaluating the quality of a link.

You can also configure port load thresholds that determine the upper and lower quality bands. The thresholds are percentages of the total port load that you configure using the min and max options. DLB assigns any egress port with a port load falling below this minimum to the highest quality band (7). Any port load larger than the maximum threshold falls into the lowest quality band (0). DLB divides the remaining port load quantities among quality bands 1 through 6.

For example, if you configure the minimum to be 10 and the maximum to be 70, DLB assigns any egress port with a port load that takes up less than 10 percent (%) of the total port load to quality band 7. DLB assigns any egress port with a port load taking up more than 70% of the total port load to quality band 0. DLB then assigns egress ports with port loads taking up 10% through 70% of the total port load to quality bands 1 through 6.

Use Feature Explorer to confirm platform and release support for specific features.

Benefits

- Optimized load balancing based on a port's load size and queues.
- Link quality parameters that suit your network needs.
- Flexible port assignment to quality bands based on real-time metrics.

Configuration

Configure the egress port quality metric.

1. Configure how much weight DLB puts on the port load metric, or amount of traffic, when determining the link quality.

   Range of rate-weightage: 0 through 100, where 100 means that DLB bases link quality 100% on the port load.

   When the rate weightage changes, the device repairs all ECMP DLB groups with the new egress quantization values for each of their egress links. During the transition between configurations, traffic can drop.

   ```
   set forwarding-options enhanced-hash-key ecmp-dlb egress-quantization rate-weightage rate-weightage
   ```

2. Configure the minimum port load in percentage.

DLB assigns any egress port with a port load falling below this minimum to the highest quality band (7). Range of `min`: 1 through 100 (percent).

```
set forwarding-options enhanced-hash-key ecmp-dlb egress-quantization min min
```

3. Configure the maximum port load in percentage.

   DLB assigns any egress port with a port load above this maximum to the lowest quality band (0). Range of `max`: 1 through 100 (percent).

```
set forwarding-options enhanced-hash-key ecmp-dlb egress-quantization max max
```

4. Verify the configuration was successful.

```
show forwarding-options enhanced-hash-key
```

Configure Flowset Table Size in DLB Flowlet Mode

IN THIS SECTION

Overview

IN THIS SECTION

Dynamic load balancing (DLB) is a load balancing technique that selects an optimal egress link based on link quality so that traffic flows are evenly distributed. You (the network administrator) can configure DLB in *flowlet mode*.

In flowlet mode, DLB tracks the flows by recording the last seen timestamp and the egress interface that DLB selected based on the optimal link quality. DLB records this information in the flowset table allocated to each ECMP group. The DLB algorithm maintains a given flow on a particular link until the last seen timestamp exceeds the inactivity timer. When the inactivity timer expires for a particular flow, DLB rechecks whether that link is still optimal for that flow. If the link is no longer optimal, DLB selects a new egress link and updates the flowset table with the new link and the last known timestamp of the flow. If the link continues to be optimal, the flowset table continues to use the same egress link.

You (the network administrator) can increase the flowset table size to change the distribution of the flowset table entries among the ECMP groups. The more entries an ECMP group has in the flowset table, the more flows the ECMP group can accommodate. In environments such as AI-ML data centers that must handle large numbers of flows, it is particularly useful for DLB to use a larger flowset table size. When each ECMP group can accommodate a large number of flows, DLB achieves better flow distribution across the ECMP member links.

The flowset table holds 32,768 total entries, and these entries are divided equally among the DLB ECMP groups. The flowset table size for each ECMP group ranges from 256 through 32,768. Use the following formula to calculate the number of ECMP groups:

```
32,768/(flowset size) = Number of ECMP groups
```

By default, the flowset size is 256 entries, so by default there are 128 ECMP groups.

Use Feature Explorer to confirm platform and release support for specific features.

Benefits

- Improve load distribution over egress links.
- Group flows to minimize how many calculations DLB has to make for each flow.
- Customize flowset table entry allocation for maximum efficiency.
- Increase the efficiency of flowlet mode.

Configuration

Be aware of the following when configuring the flowset table size:

- When you change the flowset size, the scale of ECMP DLB groups also changes. Allocating a flowset table size greater than 256 reduces the number of DLB-capable ECMP groups.
- When you commit this configuration, traffic can drop during the configuration change.
- DLB is not supported when a link aggregation group (LAG) is one of the egress members of ECMP.
- Only underlay fabrics support DLB.
- QFX5240 switch ports with a speed less than 50 Gbps do not support DLB.

1. Configure DLB in flowlet mode. See *Configuring Dynamic Load Balancing*.
2. Configure the flowset table size.

```
set forwarding-options enhanced-hash-key ecmp-dlb flowlet flowset-table-size value
```

3. Verify the configuration was successful.

```
show forwarding-options enhanced-hash-key
```

Reactive Path Rebalancing

IN THIS SECTION

Overview

IN THIS SECTION

Dynamic load balancing (DLB) is an important tool for handling the large data flows (also known as elephant flows) inherent in AI-ML data center fabrics. *Reactive path rebalancing* is an enhancement to existing DLB features.

In the flowlet mode of DLB, you (the network administrator) configure an inactivity interval. The traffic uses the assigned outgoing (egress) interface until the flow pauses for longer than the inactivity timer. If the outgoing link quality deteriorates gradually, the pause within the flow might not exceed the configured inactivity timer. In this case, classic flowlet mode does not reassign the traffic to a different link, so the traffic cannot utilize a better-quality link. Reactive path rebalancing addresses this limitation by enabling the user to move the traffic to a better-quality link even when flowlet mode is enabled.

The device assigns a quality band to each equal-cost multipath (ECMP) egress member link that is based on the traffic flowing through the link. The quality band depends on the port load and the queue buffer. The port load is the number of egress bytes transmitted. The queue buffer is the number of bytes waiting to be transmitted from the egress port. You can customize these attributes based on the traffic pattern flowing through the ECMP.

Use Feature Explorer to confirm platform and release support for specific features.

Benefits

- Scalable solution to link degradation
- Optimal use of bandwidth for large data flows
- Avoidance of load balancing inefficiencies due to long-lived flows

Configuration

IN THIS SECTION

Configuration Overview

Quality bands are numbered from 0 through 7, where 0 is the lowest quality and 7 is the highest quality. Based on the member port load and queue size, DLB assigns a quality band value to the member port. The port-to-quality band mapping changes based on instantaneous port load and queue size.

When both of the following conditions are met, reactive path rebalancing reassigns a flow to a higher-quality member link:

- A better-quality member link is available whose quality band is equal to or greater than the current member's quality band plus the configured reassignment *quality delta* value. The quality delta is the difference between the two quality bands. Configure the quality delta value using the `quality-delta` statement.

- The packet random value that the system generates is lower than the reassignment *probability threshold* value. Configure the probability threshold value using the `prob-threshold` statement.

Be aware of the following when using this feature:

- Reactive path rebalancing is a global configuration and applies to all ECMP DLB configurations in the system.

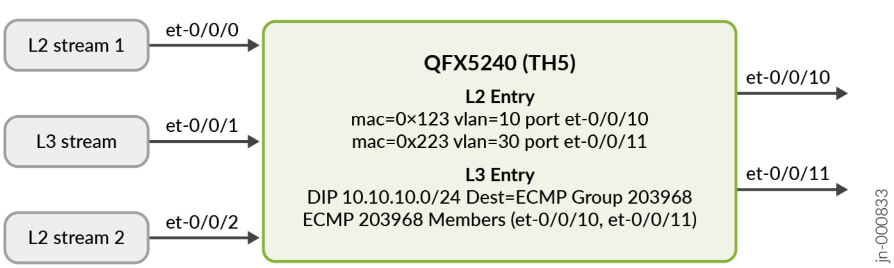
- You can configure egress quantization in addition to reactive path rebalancing to control the flow reassignment.

- Packet reordering can occur when the flow moves from one port to another. Configuring reactive path rebalancing can cause momentary out-of-order issues when the flow is reassigned to the new link.

Topology

In this topology, the device has three ingress ports and two egress ports. Two of the ingress streams are Layer 2 (L2) traffic and one is Layer 3 (L3) traffic. The figure shows the table entries forwarding the traffic to each of the egress ports. All the ingress and egress ports are of the same speed.

Figure 2: Reactive Path Rebalancing



In this topology, reactive path rebalancing works as follows:

1. Quality delta of 2 is configured.

2. L2 stream 1 (`mac 0x123`) enters ingress port et-0/0/0 with a rate of 10 percent. It exits through et-0/0/10. The egress link utilization of et-0/0/10 is 10 percent and the quality band value is 6.

3. The L3 stream enters port et-0/0/1 with a rate of 50 percent. It exits through et-0/0/11 and selects the optimal link from the ECMP member list. The egress link utilization of et-0/0/11 is 50 percent with a quality band value of 5.

4. L2 stream 2 (`mac 0x223`) enters port et-0/0/2 with a rate of 40 percent. It also exits through et-0/0/11. This further degrades the et-0/0/11 link quality band value to 4. Now the difference in the quality band values of both ECMP member links is 2.

5. The reactive path balancing algorithm now becomes operational because the difference in quality band values for ports et-0/0/10 and et-0/0/11 is equal to or higher than the configured quality delta of 2. The algorithm moves the L3 stream from et-0/0/11 to a better-quality member link, which in this case is et-0/0/10.

6. After the L3 steam moves to et-0/0/10, the et-0/0/10 link utilization increases to 60 percent with a decrease in quality band value to 5. L2 stream 2 continues to exit through et-0/0/11. The et-0/0/11 link utilization remains at 40 percent with an increase in quality band value to 5.

Configure Reactive Path Rebalancing

1. Configure DLB in flowlet mode. See *Configuring Dynamic Load Balancing*.
2. Configure the required difference (delta) in quality between the current stream member and the member available for reassignment.

   Optimal selection of the quality delta is very important. An incorrect delta can result in continuous reassignment of flow from one link to another.

   The range of the `quality-delta` statement is 0 through 8. Set it to 0 to disable reassignment of the flows.

   ```
   set forwarding-options enhanced-hash-key ecmp-dlb flowlet reassignment quality-delta reassign-
   quality-delta
   ```

3. Set the probability threshold that reactive path rebalancing uses to reassign the existing flow to a better available member link.

   Note the following when configuring the probability threshold:

   - When `quality-delta` is configured, `prob-threshold` defaults to 100.

   - The range of `prob-threshold` is 0 through 255. Set it to 0 to disable reassignment of the flows.

- A lower probability threshold value means that flows move to a higher-quality member link at a slower rate. For example, flows move to a higher-quality link more quickly with a probability threshold value of 200 than with a probability threshold value of 50.

```
set forwarding-options enhanced-hash-key ecmp-dlb flowlet reassignment prob-threshold
reassign-prob-threshold
```

4. Verify the configuration was successful.

```
show forwarding-options enhanced-hash-key
```

Global Load Balancing (GLB)

SUMMARY

Learn about GLB and how to configure GLB.

IN THIS SECTION

GLB Overview

IN THIS SECTION

Classic load balancing mechanisms use a hashing algorithm to decide the egress interface through which to send traffic. These algorithms operate the hash function on five tuples of the received packet. However, the algorithms do not consider the real-time utilization of the links through which they send packets. Even in DLB, the decision is completely local and the algorithm is unable to globally detect link utilization. If a node farther out is congested, that node might drop the packet. Global load balancing (GLB) is an enhancement to DLB that has visibility into congestion at the next-to-next-hop (NNH) level.

GLB takes into account the link utilization of remote links before deciding on the egress interface. Similarly to DLB, when one multipath leg experiences congestion, GLB can offload traffic to alternative legs to mitigate the congestion. Unlike DLB, GLB can reroute traffic flows on leaf devices to avoid traffic congestion on the spine level.

Use Feature Explorer to confirm platform and release support for specific features.

Benefits

- Reduces packet loss due to congestion and remote link failures
- Effectively load-balances large data flows in Clos topologies end-to-end to avoid congestion
- Is particularly useful in deployments where large data flows increase the likelihood of traffic congestion

GLB in AI-ML Data Centers

AI-ML data centers have less entropy and larger data flows than other networks. Because hash-based load balancing does not always effectively load-balance large data flows of traffic with less entropy, dynamic load balancing (DLB) is often used instead. However, DLB takes into account only the local link bandwidth utilization. For this reason, DLB can effectively mitigate traffic congestion only on the immediate next hop. GLB more effectively load-balances large data flows by taking traffic congestion on remote links into account.

Configure GLB

IN THIS SECTION

Considerations

Keep the following in mind when configuring GLB:

- GLB is supported only in a 3-Clos (leaf-spine-leaf) topology.
- All the devices in the 3-Clos topology must support GLB before you can configure GLB.
- The 3-Clos topology can have a maximum of 64 leaf devices when it supports GLB.
- GLB supports only one link between the same pair of devices (for example, a spine device and leaf device).

GLB does not support the following features:

- Integrated routing and bridging (IRB) interfaces between top-of-rack (ToR) and spine devices
- Multihomed servers
- GLB for overlay routes (IPv4 or IPv6)
- GLB for BGP routes learned in routing instances

Configure GLB

1. Configure DLB.

   The DLB configuration on each device in the fabric must be identical. See Dynamic Load Balancing for how to configure DLB.
2. Configure a node ID for each node.

   Each node must have a node ID. Keep the following in mind when configuring the node ID:

   - Configure the node ID at one of these hierarchy levels:

     ```
     [edit routing-optons router-id router-id]
     [edit protocols bgp bgp-identifier bgp-identifier]
     ```

   - If you configure the `bgp-identifier` statement, you must configure it globally, not at a `group` or `neighbor` hierarchy level.
   - The BGP identifier for each node must be unique within the fabric.
3. On spine devices, configure GLB in `helper-only` mode.

   In `helper-only` mode, BGP sends the NNH node (NNHN) capability for the route it advertises. BGP instructs the GLB application to monitor the link qualities of all local links with EBGP sessions and flood that information to all direct neighbors. Configure this option on the spine devices in a 3-Clos architecture.

```
set protocols bgp global-load-balancing helper-only
set forwarding-options enhanced-hash-key ecmp-dlb <flowlet | per-packet>
```

4. On leaf devices, configure GLB in `load-balancer-only` mode.

   In `load-balancer-only` mode, BGP does not send the NNHN capability for the route it advertises. The switch receives link qualities from neighboring nodes. It uses the combined link quality of next hops and NNHs to make load balancing decisions. Configure this option on the leaf devices of any Clos architecture.

```
set protocols bgp global-load-balancing load-balancer-only
set forwarding-options enhanced-hash-key ecmp-dlb <flowlet | per-packet>
```

5. Selectively disable GLB.

   After you globally configure GLB using the `global-load-balancing` statement, you can selectively disable it on a particular BGP group or peer. To selectively disable GLB, use the `no-global-load-balancing` statement at either of these hierarchy levels:

```
[edit protocols bgp group group-name]
```

```
[edit protocols bgp group group-name neighbor address]
```

   For example:

```
set protocols bgp group group-name no-global-load-balancing
```

6. Verify the configuration was successful using the following commands:

   - **show bgp global-load-balancing**
   - **show bgp global-load-balancing path**
   - **show bgp global-load-balancing path-monitor**
   - **show bgp global-load-balancing profile**

RELATED DOCUMENTATION

enhanced-hash-key

global-load-balancing

Configure GLB on 3-Clos IP Fabric with Multilinks

SUMMARY

In a Clos network, congestion on the first two next hops impacts the load balancing decisions of the local node and the previous hop nodes and triggers global load balancing (GLB). We support GLB on 3-Stage Clos topologies with multilink between spine and top-of-rack switches.

IN THIS SECTION

Dynamic load balancing (DLB) helps to avoid congested links to mitigate local congestion. However, DLB cannot address some congestion experienced by remote devices in the network. In these cases, global load balancing (GLB) extends DLB by modulating the local path selection using path quality perceived at downstream switches to mitigate congestion. GLB allows an upstream switch to avoid downstream congestion hotspots and select a better end-to-end path. In a Clos network, congestion on the first two next hops impacts the load balancing decisions of the local node and the previous hop nodes and triggers GLB. If the route has only one next-next-hop, a simple path quality profile is created. If the route has more than one next next-hop node, then a simple path quality profile is created for each next next-hop node.

Profile Sharing in Clos Network

In extensive Clos network configurations with many GPUs, the profile-sharing mechanism optimizes limited profile resources. In Clos networks with five or more stages, some nodes, like super spines exceed 64 next-next-hop nodes. We can reuse the profiles under specific conditions to support more than 64 next-next-hop nodes. When managing large-scale network fabrics, particularly in hyperscale AI/ML environments, efficiently utilizing the profile capacity is critical. The profile-sharing feature allows two next-next-hop nodes to use a single profile if their paths do not overlap. Ensuring non-overlapping paths maintains efficient routing without exceeding constraints, maximizing resource utilization.

Profile sharing involves stringent criteria to prevent configuration conflicts and performance degradation. Nodes must operate on distinct next-hop link ranges to ensure appropriate use of shared profiles, maintaining network robustness. Understanding these criteria is essential, as misconfiguration could lead to suboptimal routing or instability.

NOTE: Even with profile sharing, the total number of simple profiles must not be more than 1024. Maximum number of nexthops (paths) a HW profile can have is 352.

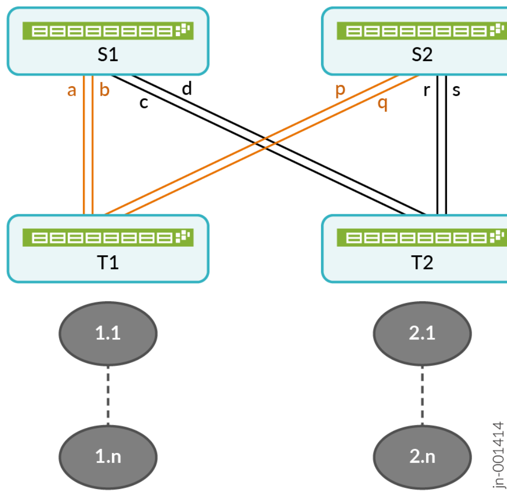
Transition from one profile space allocation to another profile space allocation might lead to over 64 hardware profiles in PFE during the transition. We strongly recommend to deactivate bgp global-load-balancing and wait until all profiles are cleared from PFE before changing the profile space allocation.

Benefits of GLB in 3-Clos Networks with Multilinks

- Mitigates congestion when AI-ML traffic that has elephant flows and lacks entropy causes congestion in the fabric.
- Efficient traffic distribution to ensure optimum link utilization. In a DC fabric, hashing is unable to ensure even load distribution over all ECMP links, which might result in underutilization of some links.
- Reduces packet loss in case of remote link failures.

In Figure 3 on page 33, S1 and S2 are spine nodes connecting to T1 and T2 top of rack (ToR) devices with multiple links a, b, c, d and p, q, r, s. S1 and S2 aggregate the quality of all available paths to a remote device and advertises the overall path quality to ToR devices. 1.1 to 1.n and 2.1 to 2.n are the hosts or routes behind the ToR devices T1 and T2 respectively. If one or more links go down, the spine continues to apply same aggregation logic to the remaining active links. The remote link state is only advertised as ‘down’ when all links in the multilink group are down.

Figure 3: GLB on 3 Clos IP Fabric with Multilinks



To configure GLB in a network with multiple paths between spine and top-of-rack switches on a 3-Clos IP fabric:

1. Configure a router ID for each node. Assign a BGP identifier if you prefer not to use the router-id as a GLB node ID.

```
[edit routing-options]
user@host# set router-id router-id
```

```
[edit protocols bgp]
user@host# set bgp-identifier bgp-identifier
```

> **NOTE**: If you change the `router-id` on a node configured with GLB, connected BGP neighbors are not updated with this change until you clear the BGP session on these peers.

2. Enable DLB on spine and leaf nodes in either flowlet or per-packet mode as per your network requirements.

Configure DLB on each router in the fabric. To achieve effective GLB, the DLB configuration on each router in the fabric must be identical.

```
[edit forwarding-options enhanced-hash-key]
user@host# set ecmp-dlb flowlet | per-packet
```

3. Enable GLB on spine and leaf devices. GLB must be configured on each router in the fabric.

```
[edit protocols bgp]
user@host# set global-load-balancing
```

   a. On spine devices, configure GLB in `helper-only` mode.

   ```
   [edit protocols bgp global-load-balancing]
    user@host# set helper-only
   ```

   In the helper mode, the node monitors the link quality.

   b. On leaf devices, configure GLB in `load-balancer-only` mode.

   ```
   [edit protocols bgp global-load-balancing]
   user@host# set protocols load-balancer-only
   ```

   In the load-balancer mode, GLB only receives link qualities from neighboring nodes and uses the combined link quality of next-hops and next-next-hops to make load balancing decisions.

4. Disable GLB on selected BGP peers or BGP groups.

```
[edit protocols bgp group group-name]
user@host# set no-global-load-balancing
```

```
[edit protocols bgp group group-name neighbor address]
user@host# set no-global-load-balancing
```

5. On spine devices, enable the GLB multilink mode. A spine can aggregate the link qualities of both the links to top-of-rack devices and send it to them. The aggregation is calculated in two ways:

a. Maximum of the active local links—Use this option if the network has links with different speeds.

```
[edit forwarding-options enhanced-hash-key]
user@host# set glb-multilink-mode max-val
```

b. Average of the local active links—Use this option if the speed of links is the same across devices. By default, the spine advertises the average quality of all the links.

```
[edit forwarding-options enhanced-hash-key]
user@host# set glb-multilink-mode avg-val
```

NOTE: When spine to top-of-rack devices are connected through multiple links, GLB multilink mode is enabled by default in average mode. If you manually change the GLB multilink mode, you must turn the power off and restart the PFE.

6. Verify the configuration using the following commands.

- **show global-load-balancing monitor-links** to display the details of all monitored links.

```
user@host> show global-load-balancing monitor-links


IFL-Index   BCM-Port   Dest-Router-Id   Encap-Id   Monitoring   MultiLink

1090         264       10.3.3.3           0          True         Yes
1092         265       10.3.3.3           0          True         Yes
1277         286       10.4.4.4           0          True         Yes
1282         287       10.4.4.4           0          True         Yes
1297         308       10.5.5.5           0          True         Yes
1298         309       10.5.5.5           0          True         Yes
```

- **show bgp global-load-balancing path-monitor** to display all path monitors that BGP has created and their installation status.

```
user@host> show bgp global-load-balancing path-monitor

        IfIndex     Local Node ID            Remote Node ID
          1090       10.2.2.2                 10.3.3.3
          1092       10.2.2.2                 10.3.3.3
```

```
            1277       10.2.2.2                  10.4.4.4
            1282       10.2.2.2                  10.4.4.4
            1297       10.2.2.2                  10.5.5.5
            1298       10.2.2.2                  10.5.5.5
```

- **show bgp global-load-balancing profile** on the leaf switch to display all GLB profiles and their installation status.

```
user@host> show bgp global-load-balancing profile
Profile(0x55882e6a8780) ID: 2002
  Lookahead profile count: 0
  Next-next-hop Node ID: 10.4.4.4
  Path count: 4
  Paths (Path ID, Next-hop IfIndex, Next-hop Node ID, Refcount):
    2006, 1074, 10.1.1.1, 1
    2005, 1075, 10.1.1.1, 1
    2004, 1076, 10.2.2.2, 1
    2003, 1077, 10.2.2.2, 1
Profile(0x55882e6a8680) ID: 2001
  Lookahead profile count: 0
  Next-next-hop Node ID: 5.5.5.5
  Path count: 4
  Paths (Path ID, Next-hop IfIndex, Next-hop Node ID, Refcount):
    2006, 1074, 10.1.1.1, 1
    2005, 1075, 10.1.1.1, 1
    2004, 1076, 10.2.2.2, 1
    2003, 1077, 10.2.2.2, 1
```

RELATED DOCUMENTATION

Global Load Balancing (GLB)

global-load-balancing

Queue-Pair Hashing for RDMA Flows

IN THIS SECTION

Overview

Load balancing algorithms use hashing mechanisms to select the best outgoing interface from a link aggregation group (LAG) or ECMP bundle. Most load balancing algorithms hash (that is, divide the traffic load among links) based on 5-tuple information. The 5-tuple information covers Layer 2 (L2), which includes the source and destination MAC addresses; Layer 3 (L3), which includes the source and destination IP addresses; and Layer 4 (L4), which includes the TCP/UDP ports. However, in AI-ML training networks, most traffic uses Remote Direct Memory Access (RDMA) over Converged Ethernet version 2 (RoCEv2) for transport. ROCEv2 traffic might not have variance on L4 ports. This leads to less entropy and could result in less efficient load balancing.

In a network that uses RoCEv2 for application traffic transport, an RDMA connection sends traffic on a send queue and receives traffic on a receive queue. These queues form the RDMA connection. Together, the send queue and receive queue are referred to as a queue pair. Each queue pair has an identifiable prefix. ROCEv2 uses the destination queue pair as the flow identification field.

You can enhance load balancing on your AI-ML training network by including the destination queue pair in the hash calculation. This is called queue-pair hashing. Adding the destination queue pair to the hash calculation increases the entropy of the system and improves the overall load balancing efficiency. On devices and releases that support this feature, queue-pair hashing is enabled by default for all ROCEv2 traffic.

NOTE: When you enable dynamic load balancing (DLB) on your device, queue-pair hashing is automatically disabled on link aggregation group (LAG) interfaces.

Configuration

IN THIS SECTION

Verify Queue-Pair Hashing

By default, the load balancing algorithm includes the RDMA queue pair as part of the hash calculation for RDMA traffic (both IPv4 and IPv6). No configuration is required.

To confirm the feature is enabled, use the following command and confirm that the “RDMA Queue Pair” field in the output is marked “yes”:

```
user@device> show forwarding-options enhanced-hash-key

Current RTAG7 Settings
-------------------------
   Hash-Mode                  :layer2-payload
   Hash-Seed                  :108202401
inet  RTAG7 settings:
----------------------
inet packet fields
  protocol                   :yes
  Destination IPv4 Addr      :yes
  Source IPv4 Addr           :yes
  destination L4 Port        :yes
  Source L4 Port             :yes
  Vlan id                    :no
  RDMA Queue Pair            :yes
inet non-packet fields
  incoming device            :yes
  incoming port              :yes
inet6  RTAG7 settings:
----------------------
inet6 packet fields
  next-header                :yes
```

```
  Destination IPv6 Addr     :yes
  Source IPv6 Addr          :yes
  destination L4 Port       :yes
  Source L4 Port            :yes
  Vlan id                   :no
  RDMA Queue Pair           :yes
inet6 non-packet fields
  incoming device           :yes
  incoming port             :yes
Hash-Parameter Settings for ECMP:
-----------------------------------
  Hash Function   = CRC16_BISYNC
  Hash offset base = 16
  Hash offset     = 4294967280
  Hash preprocess = 0
Hash-Parameter Settings for LAG:
-----------------------------------
  Hash Function   = CRC16_CCITT
  Hash offset base = 0
  Hash offset     = 0
  Hash preprocess = 0
Ecmp Resilient Hash   = Disabled
ECMP DLB Load Balancing Options:
---------------------------------------------------
  Load Balancing Method              : Disabled
  Inactivity Interval                : 0 (us)
  Flowset Table size                 : 0 (entries per ECMP)
  Reassignment Probability Threshold : 0
  Reassignment Quality Delta         : 0
  Egress Port Load Weight            : 0
  EgressBytes Min Threshold          : 1
  EgressBytes Max Threshold          : 100
  Sampling Rate                      : 62500
  Ether Type                         : All
```

Disable Queue-Pair Hashing

To exclude the queue pair from the load balancing hash calculation, configure the following:

1. Check the hash mode configured on your device using the `show forwarding-options enhanced-hash-key` command.

In this example, the hash mode is the default layer2-payload. The output has been shortened for clarity.

```
user@device> show forwarding-options enhanced-hash-key

SENT: Ukern command: show forwarding-options enhanced-hash-key
Current RTAG7 Settings
-------------------------
   Hash-Mode                 :layer2-payload
[...]
```

2. Configure the hash mode using the hash-mode statement.

```
user@device# set forwarding-options enhanced-hash-key hash-mode hash-mode
```

Even though the device already recognizes layer2-payload as the hash mode in this example as shown in the previous step, you must still configure it in the CLI. Otherwise, the commit fails.

```
user@device# set forwarding-options enhanced-hash-key hash-mode layer2-payload
```

3. Configure the no-queue-pair option for either IPv4 or IPv6.

```
user@device# set forwarding-options enhanced-hash-key (inet | inet6) no-queue-pair
```

4. Commit the configuration.

```
user@device# commit
```

Platform Support

See Feature Explorer for platform and release support.

Related Documentation

- *Load Balancing for Aggregated Ethernet Interfaces*

BGP Deterministic Path Forwarding in a CLOS Network

IN THIS SECTION

BGP Deterministic Path Forwarding Overview

SUMMARY

BGP deterministic path forwarding (DPF) divides a physical fabric into multiple logical fabrics, where different flows are mapped to different logical fabrics to serve the requirements of the flows. Single-hop EBGP's best effort service might not meet all data center flow requirements, especially for drop and latency-sensitive AI-ML flows. BGP DPF selects a path based on the specified logical fabric to ensure optimum link utilization.

IN THIS SECTION

BGP DPF is an alternative to RSVP-TE for data center IP fabrics. Data centers usually use single-hop EBGP for IPv4 and IPv6 routing. This is a simple and scalable hop-by-hop EBGP routing that provides a single best effort service for all traffic flows. BGP DPF divides a physical fabric into multiple logical fabrics. This allows both IPv4, and IPv6 traffic to be mapped to different logical fabrics. You can use separate logical fabrics for load balancing across links based on available bandwidth, and with different service level agreements (SLA). You can configure DPF to map flows to multiple logical fabrics to avoid fate sharing.

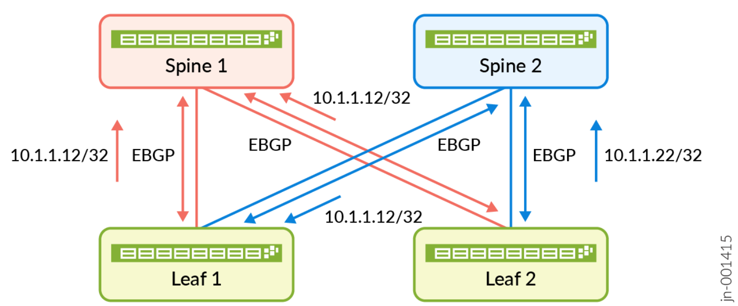
BGP DPF colors the single-hop EBGP session on each link with a fabric color. For example, if a link belongs to the red fabric, the EBGP session over the link is assigned a red BGP color community. A single-hop EBGP neighbor can be assigned a color at the global, group or neighbor level. A route with no color community assigned can be advertised over any colored or uncolored EBGP sessions.

NOTE:

- If there is a color mismatch on the receiver side, the receiver marks the route as hidden.
- All colored routes can be advertised over an uncolored BGP neighbor. To allow certain routes over only certain colors, do not have a mix of uncolored and colored fabrics.
- You can advertise uncolored routes over all colored fabrics. To advertise matching colored routes over a colored fabric, do not have uncolored routes, except for those carrying light control traffic only.

In Figure 4 on page 42, Spine1 and Spine 2 are connected to leaf 1 and leaf 2 devices with multiple connections. BGP DPF colors the single-hop EBGP session on each link with a fabric color. Spine 1 belongs to the red fabric, the EBGP sessions over the red link are also colored red. Spine 2 belongs to the blue fabric, the EBGP sessions over the link are also colored blue. BGP advertises IP routes over the EBGP session based on color matching. BGP advertises a route over a red EBGP session if it only has the red color community or no color community. BGP advertises a route over a blue EBGP session if it only has the blue color community or no color community.

Figure 4: BGP DPF: Divide a physical fabric using colored EBGP sessions



Benefits of BGP DPF

- A lightweight traffic engineering solution for IP fabrics.
- Correlation of underlay with overlay that isolates large elephant flows from small mice flows.

NOTE: We do not support Egress Link Protection (ELP) and Assisted Replication with `fabric-color` configuration. However, we still support these features for uncolored routes.

Configure BGP Deterministic Path Forwarding in a CLOS Network

This example shows how to configure BGP DPF in a 3-CLOS IP fabric:

1. Configure EBGP sessions between the spine and leaf nodes.
2. Specify a color for BGP neighbors at the global level. You can also configure fabric color at the BGP group or neighbor levels. When a neighbor is colored, only routes with the same color or no assigned color are advertised through this neighbor. Also, only routes with the same color or no assigned color are allowed to be received by this neighbor. Routes with any other colors are marked as hidden.

   ```
   [edit protocols bgp]
   user@host# set red-group
   ```

   ```
   [edit protocols bgp group group-name]
   user@host# set fabric-color com-name
   ```

   ```
   [edit protocols bgp group group-name neighbor address]
   user@host# set fabric-color com-name
   ```

3. Enable route advertisement with fabric color. These routes are advertised automatically based on the color configurations. They do not need an export policy to advertise these routes. If a color is configured, the route is advertised over the EBGP neighbors of the same color. The color community is also added for the route advertised over the colored fabric. When both color and backup-color are configured for a fabric-advertise route, an AIGP metric of 0 is added to the route. This metric signals the receiver to prefer the route when advertised over primary color neighbors.

   ```
   [edit protocols bgp]
   user@host# set fabric-advertise route address color name
   ```

   NOTE: If you do not configure a color, the route is advertised to all EBGP peers.

4. (Optional) Configure route advertisement with a backup color.

```
[edit protocols bgp fabric-advertise route address color name]
user@host# set backup-color name
```

5. Verify the configuration using the following commands.

   - Use the `show bgp summary fabric-color` *`color`* command to display neighbors with the specified fabric color only.

```
user@host> show bgp summary fabric-color red
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 3 Peers: 3 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.evpn.0
                       2          2          0          0          0          0
inet.0
                       4          3          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|
#Active/Received/Accepted/Damped...
10.1.1.2                  3         14         12       0       0        4:00 Establ
  inet.0: 2/2/2/0
```

   - Use the `show bgp neighbor fabric-color` *`color`* command to display neighbors with the specified fabric color only.

```
user@host> show bgp neighbor fabric-color red
Peer: 10.1.1.2+179 AS 3        Local: 10.1.1.1+54513 AS 2
  Group: underlay-red          Routing-Instance: master
  Forwarding routing-instance: master
  Type: External    State: Established    Flags: <Sync>
  …
  Fabric-color: red [color:0:1 ]
```

- Use the `show bgp fabric-advertise` command to display all advertise routes configured and their colors.

```
user@host> show bgp fabric-advertise

    Address                                    Color               Backup Color
    10.1.1.2                                    -                   -
    10.1.1.12                                   red [color:0:1 ]    -
    10.1.1.22                                   blue [color:0:2 ]   grey [color:0:3 ]
```

- Use the `show route` command to display the route details of the advertising protocol.

```
user@host> show route 200.1.1.1 advertising-protocol bgp 10.23.23.1 detail

inet.0: 36 destinations, 56 routes (36 active, 0 holddown, 0 hidden)

 * 192.168.1.1/32 (1 entry, 1 announced)

 BGP group red-ipv4 type External

      Nexthop: Self

      AS path: [64503] 64523 E

      Communities: color:0:1

      AIGP: 0
```

```
 user@host> show route 200.1.1.1 advertising-protocol bgp 10.23.23.5 detail


inet.0: 36 destinations, 56 routes (36 active, 0 holddown, 0 hidden)

 * 192.168.1.1/32 (1 entry, 1 announced)

 BGP group blue-ipv4 type External
```

```
        Nextop: Self
        AS path: [64503] 64523 E
        Communities: color:0:2
```

SEE ALSO

Global Load Balancing (GLB)

global-load-balancing

Weighted Packet Spray

IN THIS SECTION

Overview

IN THIS SECTION

Weighted Packet Spray (WPS) is a load balancing technique that distributes traffic unevenly across multiple links based on their configured link bandwidth values. It operates at the route level: weighted packet spray applies only when all paths for a given route have valid link bandwidth (LBW) attributes. If any path in that route lacks LBW information, that route alone falls back to regular ECMP, while other routes remain unaffected. WPS can be selectively applied through firewall filters using the `weighted-packet-spray` option, allowing specific flows to be sprayed while other flows continue to use flow-based

Weighted ECMP. Consistent implementation across network tiers ensures compatibility and stable load balancing.

Use Feature Explorer to confirm platform and release support for specific features.

Benefits

- Enhances traffic distribution by allocating flows proportionally to link bandwidth.
- Improves efficiency and resilience during link failures.
- Reduces packet loss and congestion across the network.
- Allows selective application through firewall filters for tailored traffic management.
- Complements Weighted ECMP by distributing packets proportionally to link bandwidth, instead of evenly across all links.

  Ensures consistent performance and bandwidth utilization when applied uniformly across tiers.

Configuration

Use the `weighted-packet-spray` firewall filter action to selectively enable Packet Spray only for certain flows while the rest will use flow based weighted ECMP. In the `weighted-packet-spray` load balancing method the packets will be randomly placed on the member links according to the weights. Hence, the behavior will be similar to the Packet Spray load balancing technique but with the additional feature of also honoring the weights of the links. This is especially useful when for example in a network topology involving multiple spines and leaves, if a link is down, then this link will receive less or no traffic compared to the links that are up.

The following is an example of a configuration where the `weighted-packet-spray` action is applied to packets matching an `rdma-opcode` range

```
set firewall family inet filter WEIGHTED_PKT_SPRAY term packet_spray from rdma-opcode 6-11
set firewall family inet filter WEIGHTED_PKT_SPRAY term packet_spray then count COUNT_PKT_SPRAY
set firewall family inet filter WEIGHTED_PKT_SPRAY term packet_spray then weighted-packet-spray
set firewall family inet filter WEIGHTED_PKT_SPRAY term rest then count COUNT_REST
set firewall family inet filter WEIGHTED_PKT_SPRAY term rest then accept
set interfaces irb unit 2012 family inet filter input WEIGHTED_PKT_SPRAY
```

4
CHAPTER

Traffic Management

IN THIS CHAPTER

Traffic Management Overview for AI-ML Data Centers

Traffic management is a reactive technique to handle traffic congestion as it occurs. One of the key types of traffic management for AI-ML data centers is priority-based flow control (PFC). Before configuring AI-ML traffic management features, configure class-of-service (CoS) features on your device. Read on to learn about the AI-ML traffic management features that Junos OS Evolved offers.

RELATED DOCUMENTATION

CoS Support on QFX Series Switches and EX4600 Line of Switches
Understanding CoS Flow Control (Ethernet PAUSE and PFC)

PFC Watchdog

IN THIS SECTION

PFC Watchdog Overview

IN THIS SECTION

Priority-based flow control (PFC) allows independent flow control for each class of service to ensure that congestion does not result in frame loss. PFC pause frames instruct the link partner to halt packet transmission. These frames can propagate through the network, causing traffic on the PFC streams to stop in what is known as a *PFC pause storm*. Use the *PFC watchdog* to detect and to resolve PFC pause storms.

The PFC watchdog monitors PFC-enabled ports for PFC pause storms. The PFC watchdog intervenes when a PFC-enabled port receives PFC pause frames for an extended period of time and is unable to schedule any of the data packets on PFC-enabled queues. The PFC watchdog mitigates the situation by disabling the queue where the PFC pause storm was detected for a set length of time. This length of time, called the recovery time, is configurable. After the recovery time passes, the PFC watchdog reenables the affected queue.

You can monitor the number of PFC pause storms that have been detected and recovered, as well as the number of packets that have been dropped, on a particular interface.

In lossless Ethernet fabrics such as those used in AI-ML data centers, the PFC watchdog plays a critical role in keeping the fabric lossless even during congestion.

Use Feature Explorer to confirm platform and release support for specific features.

Benefits

- Quickly detect and resolve PFC pause storms.
- Maintain lossless traffic links.
- Improve link quality.

Understanding PFC Watchdog

IN THIS SECTION

The PFC watchdog has three key functions: detection, mitigation, and restoration.

Detection

The PFC watchdog checks the status of PFC queues at regular intervals called *polling intervals*. If the PFC watchdog finds a PFC queue with a non-zero pause timer, it compares the queue's current transmit counter register to the last recorded value. If the PFC queue has not transmitted any packets since the last polling interval, the PFC watchdog checks if there are any packets in the queue. If there are packets on the queue that are not being transmitted and there are no flow control frames on that port, the PFC watchdog detects a stall condition.

The PFC watchdog monitors the PFC-enabled queues periodically for continuous PFC pause assertion by the downstream device when the queue is empty. If this occurs, PFC watchdog detects a stall condition. The system must detect this stall condition within a specified amount of time. This length of time is determined by how you configure two statements: `poll-interval` and `detection`.

The PFC watchdog checks the status of PFC queues at regular intervals. Configure this interval in milliseconds using the `poll-interval` statement. The PFC watchdog checks the status of the queues once per polling interval. The default interval is 100 ms. The minimum interval is 100 ms and the maximum is 1000 ms.

The PFC watchdog must detect stall conditions for at least two consecutive polling intervals before it determines that a PFC queue has stalled. Configure the `detection` statement to control how many polling intervals the PFC watchdog waits before it mitigates the stalled traffic. The default is two polling intervals. The maximum number is 10 polling intervals.

The total detection time is the length of the polling interval multiplied by the number of polling intervals.

Mitigation

When the PFC watchdog detects that a PFC queue has stalled, it moves the queue to the mitigation state. First it disables the queue where it detected the PFC pause storm for a period of time called the recovery time.

Configure the `pfc-watchdog-action` statement to specify the action that the PFC watchdog takes to mitigate the traffic congestion. The only option is the drop action. It drops all queued packets and all newly arriving packets for the stalled PFC queue. The system monitors all packet drops on the PFC queue during the recovery time.

Restoration

When the recovery time ends, the PFC watchdog collects the ingress drop counters and any other drop counters associated with disabling the PFC queue. The PFC watchdog maintains a count of the packets lost during the last recovery and the total number of lost packets due to PFC mitigation since the device was started. The PFC watchdog then restores the queue and re-enables PFC.

Use the recovery statement to configure how long the PFC watchdog disables the affected queue. The minimum recovery period is 200 ms and the maximum is 10,000 ms. After the recovery time passes, the PFC watchdog re-enables PFC on the affected queues.

Configure PFC Watchdog

You can enable the PFC watchdog on all PFC-enabled queues. The PFC watchdog recovery is a global setting, so it requires the same action on all ports to function. When you configure the PFC watchdog on multiple ports, make sure all ports are configured with the same type of action (drop or forward). By default, all ports use the drop action.

Enabling PFC watchdog on the congestion notification profile without configuring other options enables the PFC watchdog with the default values. By default, the polling interval is 100 ms, the detection period is set to 2 (that is, two polling intervals, or 200 ms), and the recovery time is 200 ms.

PFC watchdog only works for PFC queues. To designate a queue as a PFC queue, use the flow-control-queue statement with the queue number. For example:

```
set class-of-service congestion-notification-profile cnp output ieee-802.1 code-point 011 flow-control-queue 3
set class-of-service congestion-notification-profile cnp output ieee-802.1 code-point 100 flow-control-queue 4
```

1. Enable PFC watchdog. Use the pfc-watchdog statement at the [edit class-of-service congestion-notification-profile *profile-name*] hierarchy level:

   ```
   set class-of-service congestion-notification-profile profile-name pfc-watchdog
   ```

2. Configure the polling interval in milliseconds. The polling interval is how often the PFC watchdog checks the status of PFC queues.

   ```
   set class-of-service congestion-notification-profile profile-name pfc-watchdog poll-interval time
   ```

3. Configure the detection interval number. The detection interval number is how many polling intervals the PFC watchdog waits before it mitigates the stalled traffic.

```
set class-of-service congestion-notification-profile profile-name pfc-watchdog detection
polling-interval-number
```

4. Specify the action that the PFC watchdog takes to mitigate the traffic congestion.

```
set class-of-service congestion-notification-profile profile-name pfc-watchdog watchdog-
action drop
```

5. Configure the recovery time in milliseconds. The recovery time is how long the PFC watchdog disables the affected queue for before it restores PFC.

```
set class-of-service congestion-notification-profile cnp-name pfc-watchdog recovery time
```

6. Verify your configuration with the `show class-of-service congestion-notification-profile` *`profile-name`* command.

   The detection time shown is the polling interval multiplied by the detection interval number. In this case, the polling interval is 100 milliseconds, so the configured number of detection intervals was two.

```
user@device> show class-of-service congestion-notification-profile cnp-profile

Name: cnp, Index: 0
Type: Input
Cable Length: 100
Type: Output
  Priority     Flow-Control-Queues
  011
               3
  Priority     Flow-Control-Queues
  100
               4
PFC Watchdog : enabled
PFC-action : drop
Polling Interval : 100 ms
Detection Time : 200 ms
Recovery Time : 200 ms
```

Use the PFC Watchdog for Monitoring

You can track PFC watchdog events in the system log. The device logs PFC watchdog detection and recovery events in the system log with a timestamp. You can identify these logs from the following messages:

- `CDA PfcWd: PFC Watchdog detection enabled on ifd: et-0/0/16 Poll Interval:100ms Detection Period:200ms Recovery Interval:200ms`—PFC watchdog was enabled on a new port.
- `CDA PfcWd: PFC Storm Detected! on ifd:et-0/0/16 Queue: 3 Priority: 3 BLOCKED for AutoRecovery Recovery Time: 200ms`—PFC watchdog detected a stall condition.
- `CDA PfcWd: PFC Storm Recovered on Port ifd:et-0/0/16 Queue: 3 Priority: 3 UNBLOCKED after AutoRecovery Recovery Time: 200ms`—PFC watchdog restores the PFC queue and the queue recovers from the PFC pause storm.

You can also monitor the PFC watchdog statistics on a particular interface. Use the following command to view the number of PFC pause storms that have been detected and recovered, as well as the number of packets that have been dropped, on the PFC queues on an interface:

```
user@device> show interfaces interface extensive
...

Priority Flow Control Watchdog Statistics:
                 Detected    Recovered  LastPacketDropCount  TotalPacketDropCount
Queue :  0       0           0          0                    0
Queue :  1       0           0          0                    0
Queue :  2       0           0          0                    0
Queue :  3       0           0          0                    0
Queue :  4       0           0          0                    0
Queue :  5       0           0          0                    0
Queue :  6       0           0          0                    0
Queue :  7       0           0          0                    0

...
```

RELATED DOCUMENTATION

Configuring CoS PFC (Congestion Notification Profiles)

congestion-notification-profile

DSCP-based PFC for Layer 3 Untagged Traffic

IN THIS SECTION

You can configure DSCP-based PFC to support lossless behavior for untagged traffic across Layer 3 connections to Layer 2 subnetworks for protocols such as Remote Direct Memory Access (RDMA) over converged Ethernet version 2 (RoCEv2).

Overview

With DSCP-based PFC, pause frames are generated to notify the peer that the link is congested based on a configured 6-bit Distributed Services code point (DSCP) value in the Layer 3 IP header of incoming traffic, rather than a 3-bit IEEE 802.1p code point in the Layer 2 VLAN header.

Because PFC can only send pause frames corresponding to PFC priority code points, the 6-bit configured DSCP value must be mapped to a 3-bit PFC priority to use in pause frames when DSCP-based PFC is triggered. Configuring the mapping involves mapping the PFC priority value to a no-loss forwarding class when you map the forwarding class to a queue, defining a congestion notification profile to enable PFC on traffic with the desired DSCP value, and configuring a DSCP classifier to associate the PFC priority-mapped forwarding class (along with the loss priority) with the configured DSCP value on which to trigger PFC pause frames.

The peer device should have output PFC and a corresponding flow control queue configured to match the PFC priority configuration on the device.

Use Feature Explorer to confirm platform and release support for specific features.

DSCP-based PFC for Layer 3 Untagged Traffic in AI-ML Data Centers

AI and ML applications are rapidly expanding in data centers. When dealing with AI and ML workloads and large data sets, one critical challenge is handling the size of the data. Offloading the computation to graphics processing units (GPUs) can significantly speed up this task. However, the data size and the model, especially with large language models (LLMs), often exceed the memory capacity of a single GPU. As a result, you commonly require multiple GPUs to achieve reasonable job completion times, especially for training.

The performance of an AI data center depends on the number of GPUs that are used and the efficiency of the network that connects them. Slowdowns in the network can lead to underutilization of GPUs and longer job completion times. Ethernet-based networks are becoming more popular as an alternative to InfiniBand for AI data center networking. One solution is the Remote Direct Memory Access (RDMA) over Converged Ethernet version 2 (RoCEv2) network.

RoCEv2 involves encapsulating RDMA protocol packets within UDP packets for transport over Ethernet networks. The RoCEv2 protocol utilizes priority-based flow control (PFC) to establish a drop-free network, while *data center quantized congestion notification* (DCQCN) provides end-to-end congestion control for RoCEv2. Junos OS Evolved supports DCQCN by combining explicit congestion notification (ECN) and PFC to enable end-to-end lossless AI Ethernet networking.

To support lossless IPv6 traffic across Layer 3 (L3) connections to Layer 2 (L2) subnetworks, you can configure PFC to operate using 6-bit Differentiated Services code point (DSCP) values from L3 headers of untagged VLAN traffic. You can use PFC with DSCP as an alternative to IEEE 802.1p priority values in L2 VLAN-tagged packet headers. You need DSCP-based PFC to support RoCEv2.

Benefits

- Utilize Ethernet-based networks for AI-ML data center networking.
- Improve network efficiency for large data sets.
- Enable end-to-end lossless AI-ML Ethernet networking.

Configuration

To configure DSCP-based PFC:

1. Map a lossless forwarding class to a PFC priority—a 3-bit value represented in decimal form (0-7)—to use in the PFC pause frames.

You must also assign an output queue to the forwarding class with the `queue-num` option. The `no-loss` option is required in this case to support lossless behavior for DSCP-based PFC, and the `pfc-priority` statement specifies the priority value mapping, as follows:

```
[edit class-of-service]
user@device# set forwarding-classes class class-name queue-num queue-number no-loss
user@device# set forwarding-classes class class-name pfc-priority pfc-priority
```

2. Define an input congestion notification profile to enable PFC on traffic specified by the desired 6-bit DSCP value. Optionally configure the maximum receive unit (MRU) and cable length (used to determine PFC buffer headroom space reserved for the link):

> **NOTE**: You cannot configure both DSCP-based PFC and IEEE 802.1p PFC under the same congestion notification profile.

```
[edit class-of-service]
user@device# set congestion-notification-profile name input dscp code-point code-point-bits
pfc mru mru-value
user@device# set congestion-notification-profile name cable-length cable-length-value
```

3. Set up a DSCP classifier for the configured DSCP value and no-loss forwarding class mapped in the previous steps:

```
[edit class-of-service]
user@device# set classifiers dscp classifier-name forwarding-class class-name loss-priority
level code-points code-point-bits
```

4. Assign the classifier and congestion notification profile set up in the previous steps to an interface on which you are enabling DSCP-based PFC:

```
[edit class-of-service]
user@device# set interfaces interface-name classifiers dscp classifier-name
user@device# set interfaces interface-name congestion-notification-profile profile-name
```

5. Review your configuration.

For example, with the following sample commands configuring DSCP-based PFC for interface xe-0/0/1, PFC pause frames will be generated with PFC priority 3 when incoming traffic with DSCP value 110000 becomes congested:

```
set interfaces xe-0/0/1 unit 0 family inet address 10.1.1.2/24
set class-of-service forwarding-classes class fc1 queue-num 1 no-loss
set class-of-service forwarding-classes class fc1 pfc-priority 3
set class-of-service congestion-notification-profile dpfc-cnp input dscp code-point 110000 pfc
set class-of-service classifiers dscp dpfc forwarding-class fc1 loss-priority low code-points
110000
set class-of-service interfaces xe-0/0/1 congestion-notification-profile dpfc-cnp
set class-of-service interfaces xe-0/0/1 classifiers dscp dpfc
```

Configuration for PTX10000 Series Routers

1. PTX10000 Series routers have separate buffer spaces for lossy and lossless queues, with 10percent of the total buffer spaces reserved for lossless queues by default. If necessary, adjust the amount of buffer space reserved for lossless queus.

   You adjust the percent of buffer space reserved for lossless queues on a per-FPC basis:

   ```
   [edit chassis]
   user@device# set fpc fpc-slot no-loss buffer percentage percent
   ```

2. Map a lossless forwarding class to a PFC priority—a 3-bit value represented in decimal form (0-7)—to use in the PFC pause frames.

   You must also assign an output queue to the forwarding class with the `queue-num` option. The `no-loss` option is required in this case to support lossless behavior for DSCP-based PFC, and the `pfc-priority` statement specifies the priority value mapping, as follows:

   ```
   [edit class-of-service]
   user@device# set forwarding-classes class class-name queue-num queue-number no-loss
   user@device# set forwarding-classes class class-name pfc-priority pfc-priority
   ```

3. Define an input congestion notification profile to enable PFC on traffic specified by the desired 6-bit DSCP value. Optionally configure the maximum receive unit (MRU) and cable length (used to determine PFC buffer headroom space reserved for the link):

NOTE: You cannot configure both DSCP-based PFC and IEEE 802.1p PFC under the same congestion notification profile.

```
[edit class-of-service]
user@device# set congestion-notification-profile name input dscp code-point code-point-bits
pfc mru mru-value
user@device# set congestion-notification-profile name cable-length cable-length-value
```

Include the PFC account(s) and assign a PFC account to each code point.

```
[edit class-of-service]
user@device# set congestion-notification-profile name input pfc-account account-name pfc-
priority priority
user@device# set congestion-notification-profile name input pfc-account account-name xoff
value
user@device# set congestion-notification-profile name input pfc-account account-name xon value
user@device# set congestion-notification-profile name input dscp code-point code-point-bits
pfc-account account-name
```

4. Set up a DSCP classifier for the configured DSCP value and no-loss forwarding class mapped in the previous steps:

```
[edit class-of-service]
user@device# set classifiers dscp classifier-name forwarding-class class-name loss-priority
level code-points code-point-bits
```

5. Assign the classifier and congestion notification profile set up in the previous steps to an interface on which you are enabling DSCP-based PFC:

```
[edit class-of-service]
user@device# set interfaces interface-name classifiers dscp classifier-name
user@device# set interfaces interface-name congestion-notification-profile profile-name
```

6. Review your configuration.

   For example, with the following sample commands configuring DSCP-based PFC for interface xe-0/0/1, PFC pause frames will be generated with PFC priority 3 when incoming traffic with DSCP value 110000 reaches a delay equal to XOFF, which is set to 5000 microseconds, and a resume frame is sent with the delay falls back below XON, which is set to 2500 microseconds:

```
set chassis 0 no-loss buffer percentage 25
set interfaces xe-0/0/1 unit 0 family inet address 10.1.1.2/24
set class-of-service forwarding-classes class fc1 queue-num 1 no-loss
set class-of-service forwarding-classes class fc1 pfc-priority 3
set class-of-service congestion-notification-profile dpfc-cnp input cable-length 1000
set class-of-service congestion-notification-profile dpfc-cnp input pfc-account pfca1 pfc-priority 3
set class-of-service congestion-notification-profile dpfc-cnp input pfc-account pfca1 xoff 5000
set class-of-service congestion-notification-profile dpfc-cnp input pfc-account pfca1 xon 2500
set class-of-service congestion-notification-profile dpfc-cnp input dscp code-point 110000 pfc
set class-of-service congestion-notification-profile dpfc-cnp input dscp code-point 110000 pfc-account
pfca1
set class-of-service classifiers dscp dpfc forwarding-class fc1 loss-priority low code-points 110000
set class-of-service interfaces xe-0/0/1 congestion-notification-profile dpfc-cnp
set class-of-service interfaces xe-0/0/1 classifiers dscp dpfc
```

Verify the configuration.

1. Check the ingress port.

```
show interfaces interface-name extensive | match Priority
```

```
show interfaces queue interface-name
```

2. Display the DSCP-based input congestion notification profile.

```
show class-of-service congestion-notification-profile cnp name
```

3. Display which forwarding classes are mapped to each PFC priority.

```
show class-of-service forwarding-classes
```

RELATED DOCUMENTATION

Understanding PFC Using DSCP at Layer 3 for Untagged Traffic

Configuring CoS PFC (Congestion Notification Profiles)

Defining CoS BA Classifiers (DSCP, DSCP IPv6, IEEE 802.1p)

Defining CoS Forwarding Classes

Customize PFC X-ON Threshold and Per-Queue Alpha Values

IN THIS SECTION

Overview

IN THIS SECTION

When you configure a congestion notification profile on an ingress port, lossless traffic is mapped to lossless priority groups. You configure these priority groups with priority-based flow control (PFC) *X-OFF* and *X-ON* thresholds. In case of congestion at an egress port, these priority groups ensure that the ingress port generates the PFC frames toward the peer based on the configured thresholds. When the shared occupancy or receive buffer on an ingress priority group reaches its PFC X-OFF limit, the corresponding priority group transmits the PFC pause frame to the egress peer. The peer temporarily stops transmitting packets to give the device time to resolve the traffic congestion.

The X-ON threshold is a buffer limit that is shared by the priority group. When the buffer usage on the ingress priority group drops below this PFC X-ON limit, the priority group sends a PFC message to the peer so it can resume packet transmission. Make sure the device has enough time to resolve the congestion. You must also ensure that traffic is not paused for long enough to cause disruption to your network. To optimize the downtime during a PFC pause storm, adjust the X-ON threshold through the congestion notification profile (CNP).

You can globally adjust the limit of buffers that each queue can consume from the shared buffer pool. The shared buffer pool is based on a dynamic threshold setting called the alpha value. You can configure a scheduler with different dynamic buffer threshold values for different queues, thereby controlling the shared buffer access by individual queues.

Benefits

- Ensure that the device has enough time to resolve traffic congestion without disrupting your network.
- Customize device responses to PFC pause storms.
- Globally adjust the limits of buffers for ease of configuration.

Considerations

There are some additional considerations to keep in mind when configuring this feature with a high XON offset.

As part of the PFC feature, the hardware supports a PFC refresh functionality. When a priority group experiences congestion and the current buffer utilization exceeds the PFC XOFF threshold, the device sends a PFC XOFF frame to the peer device. If the buffer utilization does not fall back to the PFC XON threshold within the default PFC refresh time, the port generates a new PFC XOFF refresh frame and sends it to the peer device. For a 100G port, the default refresh time is 262 microseconds. This is why multiple PFC XOFF frames may be observed before a PFC XON frame is sent.

This behavior is expected for priority groups with a higher PFC XON offset. However, the PFC refresh timer operates on a per-port basis. Therefore, when the per-port PFC refresh timer expires, the port triggers PFC refresh XOFF frames for all priority groups that are in the XOFF state at that time. The hardware cannot distinguish which priority group's refresh timer has expired. As a result, even for a priority group with the default XON offset, the device might send multiple refresh XOFF frames continuously. This behavior is due to the expiration of the port-level PFC refresh timer. The many XOFF frames could cause the peer device to detect a PFC storm for that priority group. This could activate the PFC watchdog.

We recommend that if you set a very high XON offset for any priority group on a port, you should configure the peer device with a longer PFC watchdog detection timer. For example, if you set a PFC XON offset value of 10,000 for a priority group, the peer device should have a PFC watchdog detection timer of at least 10 milliseconds.

Configuration

1. Enable PFC.

   Map the `code-point` configuration to no-loss queues.

   ```
   set class-of-service congestion-notification-profile profile-name input dscp code-point code-point-bits pfc
   ```

2. Specify the number of X-ON threshold offset cells before the peer resumes transmission from the dynamic shared buffer.

   The range is 0 through 100000:

   ```
   set class-of-service congestion-notification-profile profile-name input dscp code-point code-point-bits xon value
   ```

3. Configure the per-queue alpha value through a scheduler.

   Per-queue alpha values are not supported for lossless queues. The range of the threshold for the maximum buffer share for a queue at the egress buffer partition is 0 through 10:

   ```
   set class-of-service schedulers scheduler-name buffer-dynamic-threshold value
   ```

4. Verify the shared buffer configuration.

   ```
   show class-of-service shared-buffer
   ```

5. Verify the buffer profile configuration.

   ```
   show class-of-service dedicated-buffer-profile
   ```

6. Verify the configuration on the interface.

```
show class-of-service interface interface-name
```

RELATED DOCUMENTATION

PFC

xon (Input Congestion Notification)

buffer-dynamic-threshold

Increase Shared Buffer Pool by Reducing Dedicated Buffer

IN THIS SECTION

Overview

IN THIS SECTION

Each device partitions its buffer into dedicated and shared buffers. The dedicated buffer is exclusive to each port, and only that port can use its dedicated buffer. The shared buffer is shared across all ports. When ports have little traffic, their dedicated buffer space is unused. Ports with a lot of traffic cannot use that unused buffer space as long as it is dedicated to other ports. You can effectively increase the global shared buffer space, and therefore the buffer of busy ports, by decreasing the dedicated buffer space from the default value.

You can also define a dedicated buffer profile to increase or decrease the dedicated buffer that is allocated to an individual port. Buffer profiles for individual ports are particularly useful for decreasing dedicated buffer space on unused or down ports, thereby increasing dedicated buffer space available to active ports.

Benefits

- Allow the device to allocate buffer space more efficiently among ports.
- Increase the buffer space available to active ports.
- Busy traffic ports can use more of the buffer space according to their dynamic-threshold value.

Configuration

> **NOTE**: Modify the dedicated buffer settings with caution to prevent traffic loss due to buffer misconfiguration.

1. Configure the dedicated buffer.

   To avoid all ports contending with the shared buffers and to address line-rate traffic, you cannot reduce dedicated buffers below 15 percent of the default value.

   Range of `percent`: 15 through 100 (percent).

   ```
   set class-of-service dedicated-buffer ingress percent percent
   set class-of-service dedicated-buffer egress percent percent
   ```

2. Configure the dedicated buffer profile.

   If the dedicated buffer configured as part of the `dedicated-buffer-profile` statement exceeds the total available dedicated buffers, the configuration is not effective. The configuration commits but the device logs a system logging error and does not program the configuration in the hardware.

   Range of `buffer-size`: 20 through 50,000 (absolute value in cells).

   ```
   set class-of-service dedicated-buffer-profile profile-name ingress buffer-size (none |
   absolute-value-in-cells)
   ```

```
set class-of-service dedicated-buffer-profile profile-name egress buffer-size (none |
absolute-value-in-cells)
```

3. Configure the dedicated buffer profile on a specific interface.

   You can configure this feature only on physical interfaces. You cannot attach dedicated buffer profiles to aggregated Ethernet parent ports.

```
set class-of-service interface interface-name dedicated-buffer-profile profile-name
```

4. Verify the configuration is correct.

```
show class-of-service dedicated-buffer-profile
```

RELATED DOCUMENTATION

Configuring Ingress and Egress Dedicated Buffers

dedicated-buffer

dedicated-buffer-profile

ECN Packets per Queue

IN THIS SECTION

Overview

IN THIS SECTION

Explicit congestion notification (ECN) enables two endpoint devices on TCP/IP-based networks to send end-to-end congestion notifications to each other. Without ECN, devices respond to network congestion by dropping TCP/IP packets. The dropped packets signal the occurrence of network congestion. In contrast, ECN marks packets to signal network congestion without dropping the packets. ECN reduces packet loss by making the sending device decrease the transmission rate until the congestion clears.

Packets may be delayed as the device decreases the transmission rate until congestion clears. To account for how many packets are delayed, you can use the `show interfaces queue` command to view the amount of ECN congestion experienced (CE) traffic in the queue.

Benefits

- Identify the packets that have experienced congestion.
- Helps in identifying if traffic is going to reach the queue buffer limits.
- Enables quick troubleshooting of network congestion points.

Configuration

1. Configure ECN.

   ECN is disabled by default. For how to configure ECN, see Example: Configuring Static and Dynamic ECN.
2. Enable ECN on both endpoints and on all of the intermediate devices between the endpoints.

   ECN must be enabled this way for ECN to work properly. Any device in the transmission path that does not support ECN breaks the end-to-end ECN functionality.
3. Use the `show interfaces queue` command to view the amount of traffic that has experienced congestion.

   The `ECN-CE packets` field shows the number of packets that have experienced congestion, while the `ECN-CE bytes` field shows the number of total bytes in those packets.

The per-queue ECN counters `ECN-CE packets` and `ECN-CE bytes` only count packets that experienced congestion on the local switch.

For example:

```
show interfaces queue et-0/0/5 forwarding-class network-control

Physical interface: et-0/0/5, up, Physical link is Up
  Interface index: 1262, SNMP ifIndex: 974
Forwarding classes: 12 supported, 9 in use
Egress queues: 12 supported, 9 in use
Queue: 3, Forwarding classes: network-control1
  Queued:
    Packets              :             15239998               856158 pps
    Bytes                :           2225039708            999992904 bps
  Transmitted:
    Packets              :             15239998               856158 pps
    Bytes                :           2225039708            999992904 bps
    Tail-dropped packets :                    0                    0 pps
    Tail-dropped bytes   :                    0                    0 bps
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
    ECN-CE packets       :              8577686               482043 pps
    ECN-CE bytes         :           1252342156             70378315 bps
```

IPv6 Wildcard Mask Match Conditions

IN THIS SECTION

Overview

Firewall filters are an important component of network security. You (the network administrator) might have many similar IP addresses in your network that you want to allow or disallow. Configuring each address individually can be time-consuming and inefficient. A more efficient option is to use wildcard masks. To dynamically apply firewall filters to similar IPv6 addresses, configure match conditions based on wildcard masks for your firewall filters.

By default, any IPv6 address you configure in a firewall filter has a subnet mask of 128, which means the address must match exactly. To get a partial match on an address, use a subnet mask where ffff includes that portion of the address and 0000 excludes that portion of the address.

Configuration

Follow these steps to use wildcard masks for IPv6 addresses.

1. Create an IPv6 stateless firewall filter. In this example, the filter is called `f1`.

```
[edit]
user@host# edit firewall family inet6 filter f1
```

2. Specify that a packet matches if it is from a source address or going to a destination address that matches the condition after the subnet mask is applied.

   In this example, the firewall filter matches on packets with a source address of the form 2001:db8:0:0:****:1111, where **** stands for any value. The firewall filter also matches on packets with a destination address of 2001:db8:0:0:****:2222, where **** stands for any value.

```
[edit firewall family inet6 filter f1]
user@host# set term 1 from source-address 2001:db8::1111/ ffff:ffff:0:ffff::ffff
user@host# set term 1 from destination-address 2001:db8::2222/ ffff:ffff:0:ffff::ffff
```

3. Specify that matched packets should be counted, logged to the buffer on the Packet Forwarding Engine, and accepted.

```
[edit firewall family inet6 filter f1]
user@host# set term 1 then count cnt1
user@host# set term 1 then log
user@host# set term 1 then accept
```

4. Specify that for packets not matching the specified source or destination addresses, the firewall filter should count them separately.

```
[edit firewall family inet6 filter f1]
user@host# set term 2 then count default-match
```

5. Apply the firewall filter to a particular port. In this example, the interface et-0/0/16:4 has address 2001:db8:3c4d:3::2222. We apply the firewall filter to packets that enter the port et-0/0/16:4. Any packets destined for that port match the f1 filter and are logged accordingly.

```
[edit]
user@host# set interfaces et-0/0/16:4 unit 0 family inet6 filter input f1
```

6. Commit the configuration.

```
[edit]
user@host# commit
```

7. Verify your configuration was successful.

   In the example below, the counter `cnt1` shows the number of packets that matched the firewall filter's match condition. The `default-match` counter shows the packets that did not match the condition.

```
user@device> show firewall
Filter: f1
Counters:
Name                                    Bytes                 Packets
cnt1                                    1057536               8262
default-match                           0                     0
```

Platform Support

See Feature Explorer for platform and release support.

Related Documentation

- *Guidelines for Configuring Firewall Filters*

Dropped Packet Notifications to Aid in System Performance Tuning

SUMMARY

The dropped-packet notification feature enables you to see detailed information about what is causing particular packet drops. Having that information, in real time, allows you to tune up your system's performance.

IN THIS SECTION

Dropped-Packet Notifications

Packet drops are common occurrences on network switches and routers. Debugging packet drops can be complex and time-consuming. The packet-processing pipeline supports a limited set of drop counters, but these counters are insufficient for debugging complex packet-drop issues. Debugging difficulties can result in high mean times to recovery (MTTRs).

A feature called *dropped-packet notification*, also referred to as mirror on drop (MoD), can help you debug packet drops in real time. These types of packet drops are monitored:

- Stateless ingress—Packets dropped due to processing in the ingress pipeline
- Stateless MMU—Packets dropped due to congestion in the MMU

Packets that are dropped by the IP and MMU are sampled for mirroring so that they do not overwhelm regular network traffic.

The dropped-notification feature generates reports for the drops in PSAMP format and sends them to a third-party entity, such as a collector.

Meta information—such as packet-drop reason and the congestion point (switch, port, or queue)—about the dropped packet is carried in the PSAMP file, and only the first cell of the packet is mirrored.

Samples of Dropped-Packet Notification Configurations

You configure much of the dropped-packet notification feature at the `[edit forwarding-options mirror-profile]` hierarchy level.

> **NOTE**: In the following sample configurations, all commands are optional except for the first one, for the switch ID.

Stateless MMU flow-unaware mode (`[edit forwarding-options analyzer]` hierarchy):

```
user@host# set forwarding-options switch-id id-value-range
user@host# set forwarding-options mirror-profile profile-name mirror-on-drop modtype stateless
user@host# set forwarding-options mirror-profile profile-name mirror-on-drop mmu
user@host# set forwarding-options mirror-profile profile-name mirror-on-drop mmu sample-rate
integer-value
user@host# set forwarding-options mirror-profile profile-name mirror-on-drop mmu drop-reasons
drop-reasons-list
```

Drop-reason options for MMU dropped-packet notifications:

- `ingress-limit`
- `egress-queue-limit`
- `egress-wred-drop`

Stateless ingress flow-unaware mode (`[edit forwarding-options analyzer]` hierarchy):

The ingress configuration is the same as for the MMU configuration except for the following ingress-specific configuration:

```
user@host# set forwarding-options mirror-profile profile-name mirror-on-drop ingress
```

By default, the 18 currently supported L3 ingress drop reasons are enabled internally.

Also by default, a sampling threshold value of 0 (zero) is used internally—all packets that are dropped are mirrored to the collector.

TAP Aggregation Enhancements

SUMMARY

Test access point (TAP) aggregation provides N:M (any-to-any) packet replication, allowing you to capture different types of data in real time so that you quickly see what is happening in your network. Enhancements to the TAP aggregation feature provide timestamping and ACL filtering, as well as an updated hierarchy location for the TAP aggregation interfaces configuration.

IN THIS SECTION

TAP Aggregation ACL Filtering, Timestamps, and TAP Aggregation Interfaces Update

You can enhance data accuracy by inserting a timestamp into a TAP aggregation packet that shows exactly when the data packet was captured. You configure the TAP aggregation feature to insert a timestamp in packets at data capture, before the packets are sent to the tool interfaces for analysis. For timestamping to work, you must configure the PTP reference clock on the TAP aggregation switch, and PTP must be running when the timestamp is inserted. Your tap aggregation switch must also sync the PTP FPGA's recovered time-of-day with the system chip's time-of-day.

You can apply ingress access control list (ACL) user-defined filtering (UDF) filtering on tap interfaces, enabling you to selectively choose specific traffic to be sent to the tool interfaces on a TAP aggregation switch. (You can attach different analytical tools to different tool interfaces.) If the ACL match and the TAP aggregation rule conflict, the ACL match takes precedence.

Also on supported devices, you can now configure the TAP aggregation interfaces under the `[edit interfaces]` hierarchy level.

Table 2: Enhancements to the TAP Aggregation Feature

Feature/Update	Description	Notes
Timestamp option for TAP aggregation packets	Configure timestamping to see exactly when the data packet was captured.	Include the `timestamp` CLI statement in the configuration.
Ingress ACL UDF filtering function on a tap interface	Configure filtering on a tap interface to choose specific traffic to be sent to the tool interfaces.	
Tap group and tool group interfaces configuration change	Configure the TAP aggregation interfaces directly under the `[edit interfaces]` hierarchy.	Include the `mode` CLI statement in the configuration.

NOTE: Support for 256-way equal-cost multipath (ECMP) provides a larger number of fabrics when more logical interfaces are on a spine and increases capacity to 512 next hops. This increase in possible connections improves latency and optimizes data flow.

Add Interfaces to a New TAP Aggregation Configuration

If you are configuring a new TAP aggregation setup, use these instructions when you configure the TAP aggregation interfaces:

1. Add an interface to a tap group:

```
[edit]
user@host#  set interfaces interface-name unit 0 mode tap group tap-group-name
```

2. Add an interface to a tool group:

```
[edit]
user@host#  set interfaces interface-name unit 0 mode tool group tool-group-name
```

Reconfigure the Interfaces Configuration in Your Existing TAP Aggregation Configuration

Reconfigure the interfaces in your existing TAP aggregation setup by following these steps:

1. Delete any existing interface configuration under the `[edit forwarding-options]` hierarchy.
2. Add an interface to a tap group:

```
[edit]
user@host#  set interfaces interface-name unit 0 mode tap group tap-group-name
```

3. Add an interface to a tool group:

```
[edit]
user@host#  set interfaces interface-name unit 0 mode tool group tool-group-name
```

Configure Timestamping in TAP Aggregation

Before you begin, ensure that you have configured a PTP reference clock. See Precision Time Protocol (PTP) Overview.

Configure timestamping in your TAP aggregation setup:

- Enable timestamping per interface:

```
[edit]
user@host# set interfaces interface-name timestamp ingress
```

Configure Ingress ACL Filtering in TAP Aggregation

Configure ingress ACL filtering on tap interfaces for your TAP aggregation setup. If the ACL rule conflicts with the TAP aggregation rule, the ACL rule takes precedence. See Overview of Firewall Filters (QFX Series).

RELATED DOCUMENTATION

TAP Aggregation for Network Monitoring

Load Balancing for a BGP Session

BFD for AI-ML Data Centers

IN THIS SECTION

Hardware Assisted Inline BFD Overview

IN THIS SECTION

The Bidirectional Forwarding Detection (BFD) protocol is a simple hello mechanism that detects failures in a network. BFD enables sub-second detection and convergence, preserving the continuity of AI-ML workloads. Hardware-assisted inline BFD sessions run on the ASIC firmware. The Routing Engine creates BFD sessions and passes them to the ASIC firmware for processing. The device uses existing paths to forward any BFD events that need to be processed by protocol processes.

With hardware-assisted inline BFD, the firmware handles most of the BFD protocol processing. The ASIC firmware processes the packets more quickly than the software, so hardware-assisted inline BFD is faster than regular inline BFD. We support this feature for single-hop and multihop IPv4 and IPv6 BFD sessions.

We support hardware-assisted inline BFD sessions for both underlay and overlay. For example, you can run BFD sessions between EVPN overlay BGP peers.

To enable hardware-assisted inline BFD, use the `set routing-options ppm inline-processing-enable` CLI command.

Use Feature Explorer to confirm platform and release support for specific features.

Benefits of Hardware-Assisted Inline BFD

- BFD enables faster detection and recovery from link failures.
- Hardware-assisted inline BFD reduces latency and decreases the load on the Routing Engine.
- BFD timers are adaptive. You can adjust them to be more or less aggressive.

How to Adjust BFD Timers

The BFD failure detection timers are adaptive and can be adjusted to be faster or slower. The lower the BFD failure detection timer value, the faster the failure detection and vice versa.

- To specify the BFD timers, include the `detection-time` statement:

```
bfd-liveness-detection {
    detection-time {
        threshold  milliseconds;
    }
}
```

  Specify the threshold value. This is the maximum time interval for detecting a BFD neighbor. If the transmit interval is greater than this value, the device triggers a trap. The range is 1 through 255,000 milliseconds.

- A `holddown-interval` value sets the minimum time that the BFD session must remain up before sending a state change notification.

  To specify the hold-down interval, include the `holddown-interval` statement:

```
bfd-liveness-detection {
    holddown-interval milliseconds;
}
```

You can configure a number in the range from 0 through 255,000 milliseconds, and the default is 0. If the BFD session goes down and then comes back up during the hold-down interval, the timer is restarted.

- The `minimum-interval` value indicates the time interval for transmitting and receiving data.

  This value represents the minimum interval at which the local routing device transmits BFD packets, as well as the minimum interval in which the routing device expects to receive a reply from a neighbor with which it has established a BFD session.

  To specify the minimum transmit and receive intervals for failure detection, include the `minimum-interval` statement:

```
bfd-liveness-detection {
    minimum-interval milliseconds;
}
```

  You can configure a number in the range from 1 through 255,000 milliseconds.

  NOTE: BFD is an intensive protocol that consumes system resources. Specifying a BFD minimum interval below 1 second on devices that do not support hardware assisted inline BFD can cause BFD flapping.

- The `minimum-receive-interval` statement specifies the minimum interval in which the local routing device expects to receive a reply from a neighbor with which it has established a BFD session:

```
bfd-liveness-detection {
    minimum-receive-interval milliseconds;
}
```

  You can configure a number in the range from 1 through 255,000 milliseconds.

SEE ALSO

Understanding How BFD Detects Network Failures

Drop Congestion Notification

SUMMARY

Drop Congestion Notification (DCN) is a congestion management technique based on packet trimming. Rather than dropping a packet when congestion occurs, the device experiencing congestion trims the packet's payload to a much smaller size. The device then transmits the trimmed packet through a high-priority queue toward its destination. End hosts identify the specific DCN packet that was dropped due to congestion, and quickly requesting retransmission of the lost packet. The original sender host, upon receiving the DCN feedback, immediately re-transmits the exact packet and adjusts the flow rate.

IN THIS SECTION

Overview

IN THIS SECTION

In typical Ethernet based networks, when buffers are full due to congestion, packets are dropped from the egress queues. For protocols like TCP, this results in the need for packet retransmission. To enable retransmissions, end hosts must manage a complex state machine, which can negatively impact overall network throughput.

An alternative approach is for switches to implement congestion notification mechanisms, such as Explicit Congestion Notification (ECN). ECN allows the network to signal congestion to the receiver before packet loss occurs. However, ECN also introduces some reduction in throughput, as the congestion must be detected and marked before the buffers are completely full.

Lossless networks incorporate features like Priority Flow Control (PFC) to prevent packet loss during congestion. PFC works signaling the peer device to reduce its transmission rate. However, PFC also introduces challenges, including issues such as Head-of-Line (HOL) blocking, PFC deadlock, and congestion propagation.

Drop Congestion Notification (DCN) is a congestion management technique based on packet trimming. Rather than dropping a packet when congestion occurs, the device experiencing congestion trims the packet's payload to a much smaller size (just the packet header and a small part of the payload). The device then transmits the trimmed packet through a high-priority queue toward its destination. Subsequent hops in the network recognize DCN-marked packets and direct them to high-priority queues as well. End hosts must be capable of processing the trimmed DCN packets, identifying the specific packets that were dropped due to congestion, and quickly requesting retransmission of those lost packets. The original sender host, upon receiving the DCN feedback, immediately re-transmits the exact packet and adjusts the flow rate.

With DCN enabled, the device trims a dropped packet to the size of a single cell. Trimming is most effective when it is applied to large data packets (e.g. 4KB), because it reduces the data rate significantly. Conversely, trimming packets just fractionally larger than a single cell (206 bytes) provides little data reduction, and can even be detrimental if a large fraction of bandwidth is used by trimmed packets.

Benefits

- Provide lower end-to-end latency using regular Ethernet fabric.
- No need for the end host must to maintain a state machine for an extended period to detect missing packets before initiating a retransmission request.
- With DCN, the end host can promptly request retransmission of the exact missing packets, enabling faster flow completion.

Configuration

DCN works only for UDP unicast traffic. Enabling a device to provide DCN transit requires you to set a UDP port number as the DCN protocol number and a strict-high priority forwarding class and queue for the DCN trimmed packets.

Enabling the device to also create DCN trimmed packets in case of congestion requires you to also enable DCN on one or more ingress interfaces.

1. Set the customer-defined L4 UDP port number as the DCN protocol number. The device uses this to identify DCN packets.

   ```
   set class-of-service drop-congestion-notification udp-port port-number
   ```

2. Set a strict-high priority forwarding class for all DCN trimmed packets. The forwarding class must map to a unicast queue.

```
set class-of-service drop-congestion-notification forwarding-class class-name
```

> **NOTE**: We recommend reserving a strict-high priority queue for only DCN. There is no hardware counter for DCN trimmed packets. However, if you use a dedicated queue for DCN trimmed packets, the existing queue statistics command can show DCN statistics.

3. (Optional) To enable the device to create DCN trimmed packets in case of congestion, enable DCN on individual ports.

```
set class-of-service interface ingress-interface-name drop-congestion-notification
```

> **NOTE**: You can use a wildcard, for example `et-*`, to enable DCN on all ports on the device.
>
> This step is not required if you only need the device to provide DCN transit.

Below is an example configuration:

```
set class-of-service drop-congestion-notification forwarding-class dcn
set class-of-service drop-congestion-notification udp-port 13742
set class-of-service interface et-0/0/0 drop-congestion-notification
```

Run the following commands to verify that DCN is enabled:

1.
```
user@host> show class-of-service drop-congestion-notification
Queue-num : 7
UDP Port Number : 13742
```

2.
```
user@host> show class-of-service interface et-0/0/0
Physical interface: et-0/0/0, Index: 1205
Maximum usable queues: 10, Queues in use: 9
Exclude aggregate overhead bytes: disabled
Logical interface aggregate statistics: disabled
Scheduler map: default
```

```
Congestion-notification: Disabled
Drop congestion notification : Enabled

Logical interface: et-0/0/0.0, Index: 1003
Object Name Type Index
Classifier dscp-default dscp 1
```

RELATED DOCUMENTATION

Data Center Quantized Congestion Notification (DCQCN)

5

CHAPTER

BGP

IN THIS CHAPTER

BGP Overview for AI-ML Data Centers

SUMMARY

Learn the benefits of BGP for an AI-ML data center deployment.

BGP is an exterior gateway protocol (EGP) that routers in different autonomous systems (ASs) use to exchange routing information. BGP routing information includes the complete route to each destination. BGP uses the routing information to maintain a database of network reachability information, which it exchanges with other BGP systems. BGP uses the network reachability information to construct a graph of AS connectivity. This AS connectivity graph enables BGP to remove routing loops and enforce policy decisions at the AS level.

BGP enables policy-based routing (PBR). You can use routing policies to choose among multiple paths to a destination and to control the redistribution of routing information.

In AI-ML data center deployments, you can use BGP to effectively exchange equal-cost multipath (ECMP) link or load balancing-related control plane information between layers of the Clos network. The features described in this guide use this information to improve the performance of your AI-ML data center network.

RELATED DOCUMENTATION

BGP Configuration Overview

Improve Network Resiliency Using Multiple ECMP BGP Peers

IN THIS SECTION

Overview

IN THIS SECTION

Equal-cost multipath (ECMP) is a network routing strategy that allows for traffic of the same session, or flow, to be transmitted across multiple paths of equal cost. A flow is traffic with the same source and destination. The ECMP process identifies routers that are legitimate equal-cost next hops toward the flow's destination. The device then uses load balancing to evenly distribute traffic across these multiple equal-cost next hops. ECMP is a mechanism that enables you (the network administrator) to load-balance traffic and increase bandwidth by fully utilizing otherwise unused bandwidth on links to the same destination.

You often use ECMP with BGP. Each BGP route can have multiple ECMP next hops. The BGP export policy determines whether to advertise the BGP route to these next hops. As the network administrator, you can control the advertisement and withdrawal of BGP prefixes to and from these ECMP peers. The BGP export policy determines whether to advertise a BGP prefix based on the number of ECMP BGP peers the policy receives the prefix from.

You can configure the BGP export policy to withdraw a BGP route unless it receives the BGP route prefix from a minimum number of ECMP BGP peers. Requiring the BGP route to have multiple ECMP BGP peers creates better resiliency in case of link failures.

Benefits

- Improves resiliency of your network
- Prevents accidental overloading of links
- Assists with load balancing

Configuration

The BGP export policy compares the number of ECMP next hops for the BGP route against the value you configure with the `from nexthop-ecmp` statement at either of these hierarchies: `[edit policy-options policy-statement policy-name]` or `[edit policy-options policy-statement policy-name term term-name]`.

The options for this statement are:

- *value*: The exact number of ECMP gateways (1 through 512) required to meet the condition.
- `equal`: The number of gateways must be equal to the configured value.
- `greater-than`: The number of gateways must be greater than the configured value.
- `greater-than-equal`: The number of gateways must be greater than or equal to the configured value.
- `less-than`: The number of gateways must be less than the configured value.
- `less-than-equal`: The number of gateways must be less than or equal to the configured value.

1. Configure the BGP export policy to compare the number of ECMP next hops for the BGP route against the value you configure with the `from nexthop-ecmp` statement.

   In this example, the policy term `min-ecmp` finds a match when a route has less than two ECMP BGP peers.

   ```
   set policy-options policy-statement policy-name term min-ecmp from nexthop-ecmp less-than 2
   ```

2. Configure the BGP export policy to stop advertising BGP route prefixes if the number of ECMP next hops doesn't match the conditions you configured.

   ```
   set policy-options policy-statement policy-name term min-ecmp then reject
   set policy-options policy-statement policy-name term default then accept
   ```

3. Apply the policy to routes being exported from the routing table into BGP.

   ```
   set protocols bgp group group-name export policy-name
   ```

4. Confirm that you have validated the value to be in line with the configured BGP ECMP peers in the policy.

   ```
   show policy policy-name
   ```

5. Check whether the BGP route has been advertised to or withdrawn from the desired upstream BGP peer.

```
show route advertising-protocol bgp peer-advertised [detail]
```

RELATED DOCUMENTATION

Configuring Consistent Load Balancing for ECMP Groups

BGP Link-Bandwidth Community

IN THIS SECTION

Overview

IN THIS SECTION

Within a BGP implementation, a link-bandwidth extended community encodes the bandwidth of a given next hop. BGP assists in load-balancing traffic by communicating the speeds of BGP links to remote peers. When you (the network administrator) combine a link-bandwidth community with multipath, the load-balancing algorithm of your choice distributes traffic flows across the set of next hops proportional to their relative bandwidths.

When the BGP link-bandwidth extended community is a transitive attribute across autonomous systems (ASs), the BGP group advertises the link-bandwidth extended community to neighboring ASs. You can choose to use the BGP link-bandwidth community as a nontransitive attribute so routers drop the link-bandwidth community at the AS boundary. The BGP group does not advertise nontransitive link-bandwidth communities to external BGP (EBGP) neighbors.

You can also configure BGP to automatically sense the bandwidth and import the community at a group or neighbor level. Using this link-bandwidth autosense feature, your network can automatically set the link-bandwidth value to the speed of the interface over which the device received the BGP route.

Only per-packet load balancing supports the BGP link-bandwidth community.

Benefits

- With multipath enabled, link-bandwidth provides weighted equal-cost multipath (WECMP) for unequal load balancing.
- Ensures high-bandwidth links carry more flows than low-bandwidth links.
- Reduces the likelihood of traffic congestion.

Configuration

IN THIS SECTION

Bandwidth

By default, the link-bandwidth community is transitive. You can use either of these statements to configure the link-bandwidth community as transitive:

```
set policy-options community name members bandwidth:value
```

```
set policy-options community name members bandwidth-transitive:value
```

To make it nontransitive, use the following configuration:

```
set policy-options community policy-name members bandwidth-non-transitive:value
```

Nontransitive Override

You can override a nontransitive configuration so that a BGP group sends the link-bandwidth extended community over an EBGP session even when link-bandwidth is nontransitive. To send the nontransitive link-bandwidth community across an EBGP neighbor, include the following configuration:

```
set protocols bgp group group-name send-non-transitive-link-bandwidth
```

The `send-non-transitive-link-bandwidth` statement does not differentiate between the originated link-bandwidth community and one that has been received and readvertised. When you enable this option, BGP advertises all nontransitive link-bandwidth communities to the EBGP neighbor.

Aggregate Bandwidth

By default, the aggregate link-bandwidth community is transitive. You can use either of these statements to configure the link-bandwidth community as transitive:

```
set policy-options policy-statement name then aggregate-bandwidth
```

```
set policy-options policy-statement name then aggregate-bandwidth transitive
```

To make it nontransitive, use the following configuration:

```
set policy-options policy-statement policy-name then aggregate-bandwidth non-transitive
```

To divide the total link-bandwidth by the number of peers in the advertising group, enable the `divide-equal` statement:

```
set policy-options policy-statement policy-name then aggregate-bandwidth divide-equal
```

Autosense

You can only enable autosense for single-hop EBGP sessions.

1. Configure autosense for the BGP group.

   Configure the `auto-sense` statement at the `neighbor` hierarchy to detect and store the bandwidth toward that BGP neighbor. Configure it at the `group` hierarchy to detect and store the bandwidth for all neighbors under that BGP group:

   ```
   set protocols bgp group group-name link-bandwith auto-sense
   set protocols bgp group group-name neighbor link-bandwith auto-sense
   ```

2. Configure the import policy with `auto-link-bandwidth` set to `transitive` or `non-transitive`. If you do not specify, by default `auto-link-bandwidth` is transitive:

   ```
   set protocols bgp group group-name import policy-name
   set policy-options policy-statement policy-name then auto-link-bandwidth non-transitive
   ```

3. (Optional) To suppress frequent changes in the link-bandwidth value when bandwidth increases, you can configure the autosense hold-down timer. The hold-down timer is only triggered when the bandwidth increases. By default, the timer is set to 60 seconds:

   ```
   set protocols bgp group group-name link-bandwith auto-sense hold-down time-in-seconds
   ```

Verification

Verify the configuration was successful using the following commands:

- **show route receive-protocol bgp *peer-ip-address* extensive**
- **show route advertising-protocol bgp *peer-ip-address* extensive**
- **show route *address* extensive**
- **show bgp neighbor *address***

RELATED DOCUMENTATION

auto-sense

group (Protocols BGP)

policy-statement

Load Balancing for a BGP Session

Advertising Aggregate Bandwidth Across External BGP Links for Load Balancing Overview

6
CHAPTER

EVPN-VXLAN

IN THIS CHAPTER

EVPN-VXLAN for AI-ML Data Centers

IN THIS SECTION

Overview of EVPN-VXLAN for AI-ML Data Centers

IN THIS SECTION

This document covers the steps necessary to configure Ethernet VPN-Virtual Extensible LAN (EVPN-VXLAN) in an artificial intelligence (AI) and machine learning (ML) data center fabric.

Features and Benefits of an AI-ML Data Center

- Improve scalability: You can enable multitenancy within the same data center using an IP fabric overlay.
- Improve productivity: You can run different AI workloads (multiple large language models (LLMs) for different tenants) in the same data center.
- Improve security: You can isolate L2 at the local top-of-rack (ToR) level with multiple MAC-VRF instances, or L3 at the ToR level with multiple EVPN Type 5 routing instances (IP-VRF-to-IP-VRF model). See the configuration section for examples of these use cases.
- Reduce configuration efforts: You can extend the tenants' logical context between different ToR switches in different points of delivery (PODs) without changing the configuration of the intermediate spine or superspine devices.

Use Feature Explorer to confirm platform and release support for specific features.

Configuration

IN THIS SECTION

Configuration Overview

We'll look at two use cases relevant to this topic. The first use case is running two MAC-VRF instances on the same device in a data center. The second use case is running two EVPN Type 5 VRF instances on the same device in a data center.

Use Case #1: Two MAC-VRF instances on the same device:

- Separate MAC-VRF instances help to isolate the AI data center tenants at the L2 level, and extend this isolation using the EVPN-VXLAN overlay.
- The intermediate AI data center spine and superspine devices don't require provisioning each new AI data center tenant.
- The L2 connectivity is closer to the actual service connection.
- AI data center tenants can be in the same MAC-VRF L2 EVPN instance (EVI) when you configure the tenants with the `vlan-aware` EVPN service type.

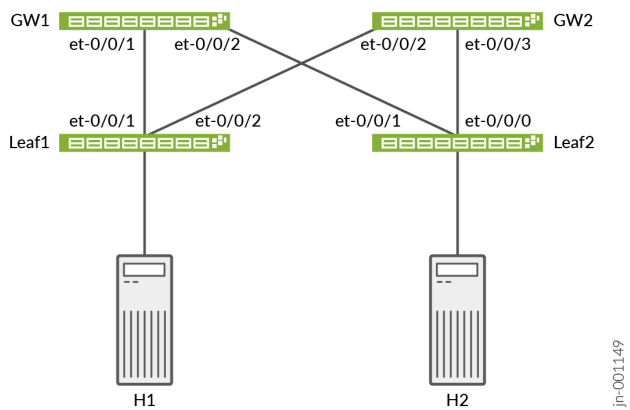
Use Case #2: Two EVPN Type 5 IP-VRF instances on the same device:

- Multiple EVPN Type 5 routing instances can isolate the AI data center tenants at the L3 routing level. Pure Type 5 routing can also extend the context within a POD or between PODs.
- EVPN signaling exchanges between the ToR switches of the AI data center automatically establish VXLAN tunnels for Type 5 routes.

Topology

The topology for these examples uses QFX5240-64QD switches for both the spine and leaf layers. The network is an edge-routed bridging (ERB) architecture.

Figure 5: Topology



How to Configure Two MAC-VRFs

Use the following steps as a guide to configure two MAC-VRF instances on the same leaf node. We use actual values for example purposes. You should customize these steps with relevant values for your implementation.

1. Configure a MAC-VRF routing instance.

```
set routing-instances myMACVRF101 instance-type mac-vrf
```

2. Configure the EVPN protocol with VXLAN encapsulation and supporting statements.

```
set routing-instances myMACVRF101 protocols evpn encapsulation vxlan
set routing-instances myMACVRF101 protocols evpn default-gateway no-gateway-community
set routing-instances myMACVRF101 protocols evpn extended-vni-list 5101
```

3. Configure a virtual tunnel endpoint (VTEP) interface.

```
set routing-instances myMACVRF101 vtep-source-interface lo0.0
```

4. Configure a service type. We use `vlan-aware` for this example. `vlan-aware` allows configuring more than one VLAN.

```
set routing-instances myMACVRF101 service-type vlan-aware
```

5. Configure an interface for the routing instance.

```
set routing-instances myMACVRF101 interface et-0/0/4.0
```

6. Configure a route distinguisher (RD) and a VRF target.

```
set routing-instances myMACVRF101 route-distinguisher 10.203.113.10:101
set routing-instances myMACVRF101 vrf-target target:1:9101
```

7. Configure one or more VLANs.

```
set routing-instances myMACVRF101 vlans vlan101 vlan-id 101
set routing-instances myMACVRF101 vlans vlan101 l3-interface irb.101
set routing-instances myMACVRF101 vlans vlan101 vxlan vni 5101
```

8. Configure a second MAC-VRF routing instance. The full configuration is displayed. Note the differences in VLANs, RD, VRF-target, and interfaces. This routing instance uses the `vlan-based` service type, limiting the configuration to a single VLAN. Either `vlan-based` or `vlan-aware` are valid choices.

```
set routing-instances myMACVRF102 instance-type mac-vrf
set routing-instances myMACVRF102 protocols evpn encapsulation vxlan
set routing-instances myMACVRF102 protocols evpn default-gateway no-gateway-community
set routing-instances myMACVRF102 protocols evpn extended-vni-list 5102
set routing-instances myMACVRF102 vtep-source-interface lo0.0
set routing-instances myMACVRF102 service-type vlan-based
set routing-instances myMACVRF102 interface et-0/0/5.0
set routing-instances myMACVRF102 route-distinguisher 10.203.113.10:102
set routing-instances myMACVRF102 vrf-target target:1:9102
set routing-instances myMACVRF102 vlans vlan102 vlan-id 102
set routing-instances myMACVRF102 vlans vlan102 l3-interface irb.102
set routing-instances myMACVRF102 vlans vlan102 vxlan vni 5102
```

Verification

Verify that routing is working as expected. Note that verification requires other network devices to be configured, and your outputs will vary.

```
user@device> show route table myMACVRF101.evpn.0 active-path

myMACVRF101.evpn.0: 10 destinations, 15 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2:10.203.113.10:101::5101::00:10:94:00:00:05/304 MAC/IP
                   *[EVPN/170] 04:43:28
                       Indirect
2:10.203.113.10:101::5101::6c:62:fe:b9:3b:3d/304 MAC/IP
                   *[EVPN/170] 05:23:00
                       Indirect
2:10.203.113.11:101::5101::00:10:94:00:00:06/304 MAC/IP
                   *[BGP/170] 04:36:14, localpref 100, from 10.203.113.14
                      AS path: 65101 64513 I, validation-state: unverified
                       to 192.0.2.11 via et-0/0/1.0, Push 5101
                       to 192.0.2.9 via et-0/0/0.0, Push 5101
                    >  to 192.0.2.13 via et-0/0/2.0, Push 5101
2:10.203.113.11:101::5101::6c:62:fe:b9:22:3d/304 MAC/IP
                   *[BGP/170] 04:36:26, localpref 100, from 10.203.113.13
                      AS path: 65101 64513 I, validation-state: unverified
                       to 192.0.2.11 via et-0/0/1.0, Push 5101
                       to 192.0.2.9 via et-0/0/0.0, Push 5101
                    >  to 192.0.2.13 via et-0/0/2.0, Push 5101
```

```
user@device> show mac-vrf forwarding vlans

Routing instance        VLAN name             Tag          Interfaces
default-switch          default               1

myMACVRF101             vlan101               101
                                                           et-0/0/4.0*
                                                           vtep-53.32773*
myMACVRF102             vlan102               102
```

```
                                                            et-0/0/5.0
                                                            vtep-54.32773*
```

```
user@device> show ethernet-switching table vlan-id 101

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static, C -
Control MAC
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC,
           B - Blocked MAC)


Ethernet switching table : 3 entries, 3 learned
Routing instance : myMACVRF101
   Vlan                MAC                 MAC       GBP    Logical                SVLBNH/
Active
   name                address             flags     tag    interface              VENH Index
source
   vlan101             00:10:94:00:00:05   D                et-0/0/4.0
   vlan101             00:10:94:00:00:06   DR               vtep-53.32773
10.203.113.11
   vlan101             6c:62:fe:b9:22:3d   DRP              vtep-53.32773
10.203.113.11
```

How to Configure Two Type 5 IP-VRFs

Use the following steps as a guide to configuring two Type 5 IP-VRFs on the same leaf node. We use actual values for example purposes. You should customize these steps with relevant values for your implementation.

1. Configure a VRF routing instance.

```
set routing-instances RT5-IPVRF1 instance-type vrf
```

2. Configure the EVPN protocol with Type 5 support.

```
set routing-instances RT5-IPVRF1 protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances RT5-IPVRF1 protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances RT5-IPVRF1 protocols evpn ip-prefix-routes vni 1100
set routing-instances RT5-IPVRF1 protocols evpn ip-prefix-routes export my-t5-export-vrf1
```

3. Configure routing options.

```
set routing-instances RT5-IPVRF1 routing-options static route 192.168.10.10/32 discard
set routing-instances RT5-IPVRF1 routing-options multipath
```

4. Configure interfaces, RD, and VRF target.

```
set routing-instances RT5-IPVRF1 interface irb.101
set routing-instances RT5-IPVRF1 interface lo0.1
set routing-instances RT5-IPVRF1 route-distinguisher 10.203.113.10:200
set routing-instances RT5-IPVRF1 vrf-target target:1100:1100
set routing-instances RT5-IPVRF1 vrf-table-label
```

5. Configure a second Type 5 IP-VRF on the same leaf node. The full configuration is displayed. Note the differences in VLANs, RD, VRF-target, and interfaces.

```
set routing-instances RT5-IPVRF2 instance-type vrf
set routing-instances RT5-IPVRF2 routing-options static route 192.168.20.20/32 discard
set routing-instances RT5-IPVRF2 routing-options multipath
set routing-instances RT5-IPVRF2 protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances RT5-IPVRF2 protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances RT5-IPVRF2 protocols evpn ip-prefix-routes vni 2100
set routing-instances RT5-IPVRF2 protocols evpn ip-prefix-routes export my-t5-export-vrf2
set routing-instances RT5-IPVRF2 interface irb.102
set routing-instances RT5-IPVRF2 interface lo0.2
set routing-instances RT5-IPVRF2 route-distinguisher 10.203.113.10:202
set routing-instances RT5-IPVRF2 vrf-target target:2100:2100
set routing-instances RT5-IPVRF2 vrf-table-label
```

6. The routing policy supporting each VRF is shown here.

```
set policy-options policy-statement loopback-advertise term loo from route-filter
10.203.113.10/32 exact
set policy-options policy-statement loopback-advertise term loo then accept

set policy-options policy-statement my-t5-export-vrf1 term term1 from route-filter
192.168.10.10/32 exact
set policy-options policy-statement my-t5-export-vrf1 term term1 then accept
set policy-options policy-statement my-t5-export-vrf1 term term2 from route-filter
10.10.101.0/24 orlonger
```

```
set policy-options policy-statement my-t5-export-vrf1 term term2 from route-filter
192.168.101.1/32 exact
set policy-options policy-statement my-t5-export-vrf1 term term2 then accept

set policy-options policy-statement my-t5-export-vrf2 term term1 from route-filter
192.168.20.20/32 exact
set policy-options policy-statement my-t5-export-vrf2 term term1 then accept
set policy-options policy-statement my-t5-export-vrf2 term term2 from route-filter
10.10.102.0/24 orlonger
set policy-options policy-statement my-t5-export-vrf2 term term2 from route-filter
192.168.102.1/32 exact
set policy-options policy-statement my-t5-export-vrf2 term term2 then accept

set policy-options policy-statement pplb then load-balance per-packet
```

Verification

Verify that routing is working as expected. Note that verification requires other network devices to be configured, and your outputs will vary.

```
user@device> show bgp summary

Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 3 Peers: 6 Down peers: 1
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                      11         11          0          0          0          0
bgp.evpn.0
                      34         17          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
192.0.2.9             65534        713        699       0       0     5:17:25 Establ
  inet.0: 4/4/4/0
192.0.2.11            65534        709        699       0       0     5:17:25 Establ
  inet.0: 3/3/3/0
192.0.2.13            65534        713        699       0       0     5:17:25 Establ
  inet.0: 4/4/4/0
198.51.100.5          65512         18         24       0       1     5:40:07 Idle
10.203.113.13          65101        724        705       0       0     5:13:39 Establ
  bgp.evpn.0: 14/17/17/0
  myMACVRF101.evpn.0: 3/5/5/0
```

```
  myMACVRF102.evpn.0: 3/3/3/0
  __default_evpn__.evpn.0: 0/0/0/0
  RT5-IPVRF1.evpn.0: 4/5/5/0
  RT5-IPVRF2.evpn.0: 4/4/4/0
10.203.113.14          65101        687        679       0       0     5:04:10 Establ
  bgp.evpn.0: 3/17/17/0
  myMACVRF101.evpn.0: 2/5/5/0
  myMACVRF102.evpn.0: 0/3/3/0
  __default_evpn__.evpn.0: 0/0/0/0
  RT5-IPVRF1.evpn.0: 1/5/5/0
  RT5-IPVRF2.evpn.0: 0/4/4/0
```

```
user@device> show route table RT5-IPVRF1.evpn.0

RT5-IPVRF1.evpn.0: 10 destinations, 15 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

5:10.203.113.10:200::0::10.10.101.0::24/248
                   *[EVPN/170] 05:28:52
                       Fictitious
5:10.203.113.10:200::0::10.10.101.1::32/248
                   *[EVPN/170] 05:28:52
                       Fictitious
5:10.203.113.10:200::0::10.10.101.10::32/248
                   *[EVPN/170] 04:49:20
                       Fictitious
5:10.203.113.10:200::0::192.168.10.10::32/248
                   *[EVPN/170] 05:32:20
                       Fictitious
5:10.203.113.10:200::0::192.168.101.1::32/248
                   *[EVPN/170] 05:32:20
                       Fictitious
5:10.203.113.11:200::0::10.10.101.0::24/248
                   *[BGP/170] 04:42:18, localpref 100, from 10.203.113.13
                      AS path: 65101 64513 I, validation-state: unverified
                    >  to 192.0.2.11 via et-0/0/1.0, Push 1100
                       to 192.0.2.9 via et-0/0/0.0, Push 1100
                       to 192.0.2.13 via et-0/0/2.0, Push 1100
                    [BGP/170] 04:42:06, localpref 100, from 10.203.113.14
                      AS path: 65101 64513 I, validation-state: unverified
                    >  to 192.0.2.11 via et-0/0/1.0, Push 1100
```

```
            to 192.0.2.9 via et-0/0/0.0, Push 1100
            to 192.0.2.13 via et-0/0/2.0, Push 1100
```

RELATED DOCUMENTATION

Understanding EVPN with VXLAN Data Plane Encapsulation

IPv6 Underlay for an IPv4 EVPN-VXLAN Fabric

IN THIS SECTION

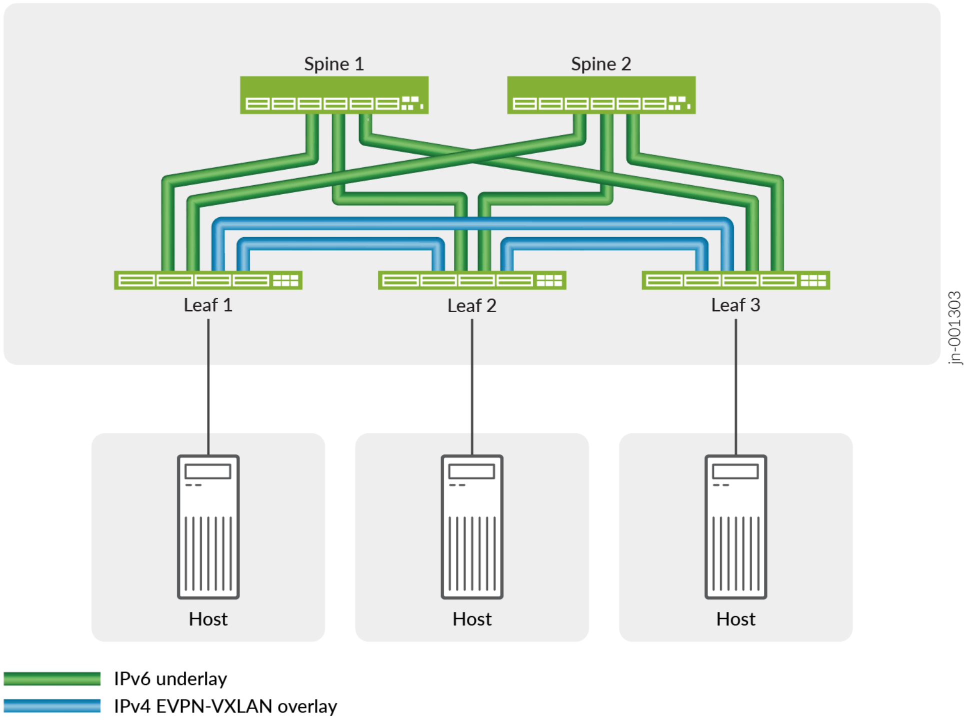
Overview

When configuring an IPv6 underlay, an additional challenge might be present for a network with an existing IPv4 EVPN-VXLAN overlay. In such a network, the virtual tunnel endpoint (VTEP) address is IPv4. The VXLAN overlay extends virtual tunnels (VTs) between VTEPs over the underlying IP fabric. To add an IPv6 underlay, the network needs to distribute this IPv4 address to the IPv6-only underlay. Certain platforms and releases support configuring your AI-ML data center to accomplish this when the IPV6 underlay is external BGP (EBGP).

Topology

This configuration is suited for a network with the following topology:

Figure 6: Network topology of IPv4 EVPN-VXLAN overlay with an IPv6 underlay



Configuration

Follow these steps to send IPV4 network layer reachability information (NLRI) over the IPv6 address family in your EBGP underlay.

1. Enable BGP to send IPV4 NLRI over the IPv6 address family.

```
user@host# set protocols bgp group bgp-underlay family inet unicast extended-nexthop
```

2. Verify that the IPv4 loopback address, which functions as the VTEP address, is reachable over the IPv6 address.

   In this example, the VTEP address is 192.168.255.255.

```
user@host> show route forwarding-table destination 192.168.255.255

Routing table: default.inet
Internet:
Destination              Type  RtRef  Next hop       Type  Index  NhRef  Netif
```

```
192.168.255.255/32      user    0                   ulst    9074    1
2001:db8:3c4d:5::ucst   6000    1     et-0/0/3.0    sftw    9070    1       et-0/0/3.0
2001:db8:3c4d:6::ucst   6001    1     et-0/0/12.0   sftw    9073    1       et-0/0/12.0
```

```
user@host> show route table macvrf1.evpn.0

macvrf1.evpn.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2001:db8:3c4d:7::/248 IM
                   *[EVPN/170] 00:21:22
                       Indirect
2001:db8:3c4d:8::/248 IM
                   *[BGP/170] 00:02:51, localpref 100, from 192.168.1.0
                      AS path: 65102 65003 I, validation-state: unverified
>  to 2001:db8:3c4d:5:: via et-0/0/3.0
                       to 2001:db8:3c4d:6:: via et-0/0/12.0
                    [BGP/170] 00:02:47, localpref 100, from 192.168.2.0
                      AS path: 65101 65003 I, validation-state: unverified
>  to 2001:db8:3c4d:5:: via et-0/0/3.0
                       to 2001:db8:3c4d:6:: via et-0/0/12.0
```

3. You can now run IPv4 EVPN-VXLAN on an IPV6-only underlay.

Platform Support

See Feature Explorer for platform and release support.

Related Documentation

- *EVPN-VXLAN with an IPv6 Underlay*

BGP Auto-Discovery Underlay for EVPN-VXLAN

IN THIS SECTION

A BGP auto-discovery underlay, as defined by RFC 5549, auto-discovers eBGP peer neighbors using the link-local IPv6 addresses. Enabling this underlay for EVPN-VXLAN helps simplify your EVPN-VXLAN network. When running RFC 5549-defined BGP unnumbered peering in the underlay BGP, you can use EVPN Type 2 media access control (MAC) virtual routing and forwarding (VRF) or EVPN-VXLAN Type 5 IP VRF for the overlay. While the underlay links run the IPv6 link-local, the EVPN-VXLAN overlay can still use the IPv4 tunnel termination using the embedded advertisement. You can use EVPN-VXLAN Type 2 and Type 5 routing instances on the same fabric as BGP unnumbered peering.

Use Feature Explorer to confirm platform and release support for specific features.

Benefits of BGP Auto-Discovery Underlay

- Simplified underlay BGP establishment without enabling Point-to-Point Protocol addresses from leaf to spine devices.
- Faster onboarding of new leaf devices onto the EVPN-VXLAN fabric.

RELATED DOCUMENTATION

BGP Unnumbered EVPN Fabric

7

CHAPTER

Interfaces

IN THIS CHAPTER

802.1X Authentication on Layer 2 Interfaces

IN THIS SECTION

Overview

IN THIS SECTION

The IEEE 802.1X standard for port-based network access control (PNAC) provides a mechanism to authenticate users of devices attached to a LAN port. The 802.1X standard verifies the user's credentials in a local or remote user database. The authentication mechanism allows only users with the correct credentials to access the network. It denies access for all other users, thereby controlling network access.

The three basic components of a network with 802.1X authentication are:

- Authenticator port access entity (PAE): A switch or router port to which a client connects. Authenticator PAEs form the control gate that blocks all traffic to and from the clients until 802.1X authenticates the clients.
- Supplicants: Clients that are trying to access the network and need to be authenticated. Supplicants connect to authenticator PAEs.
- Authentication server: The back-end database containing information about the users that are allowed to connect to the network. When a supplicant attempts to log in, 802.1X sends the supplicant's credentials to this server for authentication.

After the authentication server authenticates the supplicant's credentials, the device stops blocking access on the PAE. The device opens the interface to the supplicant and allows it to access the network. You (the network administrator) can configure 802.1X on Layer 2 (L2) interfaces.

The 802.1X IEEE standard allows you to use any authentication server for client authentication. RADIUS servers are most commonly used because those servers are easy to configure. RADIUS servers also provide the option to define proprietary, or vendor-specific, attributes. The device and the server can exchange these attributes.

Benefits

- Authenticate users.
- Prevent bad actors from accessing your network.
- Control network access.

Configuration

1. Configure the L2 interface.

   For example:

   ```
   set interfaces et-0/0/0 unit 0 family ethernet-switching interface-mode access
   set interfaces et-0/0/0 unit 0 family ethernet-switching vlan members v10
   set vlans v10 vlan-id 10
   ```

2. Enable 802.1X authentication using the `authenticator` statement.

   a. Single-supplicant mode:

   ```
   set protocols dot1x authenticator interface et-0/0/0.0 supplicant single
   ```

   b. Single-secure-supplicant mode:

   ```
   set protocols dot1x authenticator interface et-0/0/0.0 supplicant single-secure
   ```

   c. Multiple-supplicant mode:

   ```
   set protocols dot1x authenticator interface et-0/0/0.0 supplicant multiple
   ```

3. Create the 802.1X profiles and associate the profiles to 802.1X, the RADIUS authentication server, and the RADIUS accounting server.

For example:

```
set access profile dot1x-auth-profile authentication-order radius
set access profile dot1x-auth-profile radius authentication-server address
set access profile dot1x-auth-profile radius accounting-server address
set protocols dot1x authenticator authentication-profile-name dot1x-auth-profile
set access profile dot1x-accounting authentication-order radius
set access profile dot1x-accounting accounting order radius
```

4. Configure the RADIUS authentication server.

   For example:

```
set access radius-server address port 1812
set access radius-server address secret secret
set access radius-server address timeout 3
set access radius-server address retry 3
set access radius-server address source-address source-address
```

5. Verify the configuration using the following commands.

   - **show vlans**
   - **show ethernet-switching table**
   - **show mac-vrf forwarding mac-table**
   - **show dot1x interface detail**

Optics Pre-FEC BER Rate

Pre-forward error correction (pre-FEC) bit error rate (BER) provides an insight into the link quality. Typically, the device uses FEC-corrected code words to calculate the pre-FEC BER. Certain devices with an ASIC serializer/deserializer (SerDes) support the FEC histogram, which provides a detailed view of link quality in terms of the number of symbol error corrections taking place per FEC code word.

With four-level pulse-amplitude modulation (PAM4) and a high-speed SerDes such as the 56G or 112G, the eye spacing is reduced to one third of the nonreturn to zero (NRZ). Due to less spacing between the voltage levels, the signal-to-noise ratio (SNR) is typically low, making the link very susceptible to noise.

To compensate for that, devices can use stricter FEC algorithms. When these algorithms are used, the FEC-corrected code words counter usually increments. In these conditions, you can use the pre-FEC BER and FEC histogram to forecast the link quality.

On devices that support this feature, it is enabled by default. To display the pre-FEC BER and the FEC histogram, use the `show interfaces` *`interface-name`* command. This command displays FEC corrected and un-corrected code words counters, the pre-FEC BER, and the FEC histogram.

Use Feature Explorer to confirm platform and release support for specific features.

RELATED DOCUMENTATION

Understanding Pre-FEC BER Monitoring and BER Thresholds

FEC Histogram and Statistics

IN THIS SECTION

You can use forward error correction (FEC) counters and the FEC histogram to forecast the link quality histogram and statistics. To display the this information, use the `show interfaces` *`interface-name`* command. This command displays the FEC corrected and un-corrected code words counters, pre-FEC BER, and the FEC histogram.

Use Feature Explorer to confirm platform and release support for specific features.

Output

In the output of this command, `Total Code Words` is the harvested count. `Code Words` is the instantaneous count.

For example:

```
user@device> show interfaces et-0/0/26

Physical interface: et-0/0/26, Enabled, Physical link is Up
  Interface index: 1214, SNMP ifIndex: 626
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 400Gbps, BPDU Error: None, Loop
Detect PDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled, Source filtering:
Disabled,
  Flow control: Disabled, Auto-negotiation: Disabled, Media type: Fiber
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  CoS queues     : 0 supported, 0 maximum usable queues
  Current address: e4:f2:7c:82:cb:58, Hardware address: e4:f2:7c:82:cb:58
  Last flapped   : 2025-02-05 21:54:51 PST (02:20:48 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 86054410200 bps (84037509 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC Mode  :                  FEC119
    FEC Codeword size                      544
    FEC Codeword rate                    0.945
  Ethernet FEC statistics               Errors
    FEC Corrected Errors              90245707
    FEC Uncorrected Errors                   0
    FEC Corrected Errors Rate            10838
    FEC Uncorrected Errors Rate              0
    PRE FEC BER                 0.00010699043195928
  Symbol Error Per Code Word        Code Words    Total Code Words    Last Changed
         Bin0                        158197020        659994893109    00:00:01 ago
         Bin1                            21923            90120080    00:00:01 ago
         Bin2                               27              124983    00:00:01 ago
         Bin3                                0                 641           Never
         Bin4                                0                   5           Never
         Bin5                                0                   0           Never
         Bin6                                0                   0           Never
         Bin7                                0                   0           Never
         Bin8                                0                   0           Never
```

```
        Bin9                                   0                    0            Never
        Bin10                                  0                    0            Never
        Bin11                                  0                    0            Never
        Bin12                                  0                    0            Never
        Bin13                                  0                    0            Never
        Bin14                                  0                    0            Never
        Bin15                                  0                    0            Never
 Interface transmit statistics: Disabled
 Link Degrade :
   Link Monitoring                   :  Disable

 Logical interface et-0/0/26.0 (Index 1003) (SNMP ifIndex 760)
   Flags: Up SNMP-Traps Encapsulation: Ethernet-Bridge DF
   Input packets : 0
   Output packets: 0
   Protocol ethernet-switching, MTU: 1514
     Flags: Is-Primary
```

RELATED DOCUMENTATION

Forward Error Correction (FEC)