NOTA TECNICA

IIT B4-01/2024

# The “Collapsed Spine IP Fabric” Network Topology with BGP/VXLAN

D. Puliero, L. Vasarelli

ISTITUTO DI INFORMATICA E TELEMATICA

Consiglio Nazionale delle Ricerche

# The “Collapsed Spine IP Fabric” Network Topology with BGP/VXLAN

*D. Puliero, L.Vasarelli*
Network Unit of Registro .it
Technology Unit “Digital Innovation”
Institute of Informatics and Telematics — Italian National Research Council
via G. Moruzzi, 1 — 56124 Pisa, Italy

## Abstract

This document presents a complete solution for the implementation of a network infrastructure for small data centers that still need high levels of resilience and reliability.
The EVPN framework, an evolution of L2VPNs, and the 3 different supported topologies are examined: 3-stage IP fabric, 5-stage IP fabric, and Collapsed Spine IP Fabric. All the main features for implementing multi-homing functionality in active-active mode (to have a full-mesh between the production devices) in the "Collapsed Spine IP Fabric" topology are analyzed and implemented. This specific topology is chosen for analysis because it can be resilient and reliable while keeping costs very low, a common problem in small businesses.
The EVPN framework provides two different protocols for data plane management: the MPLS protocol and the VXLAN protocol. This document focuses on the VXLAN protocol.
Having Juniper Networks devices available for experimentation, the commands are related to JunOS.

## General Concepts of the EVPN-VXLAN Framework

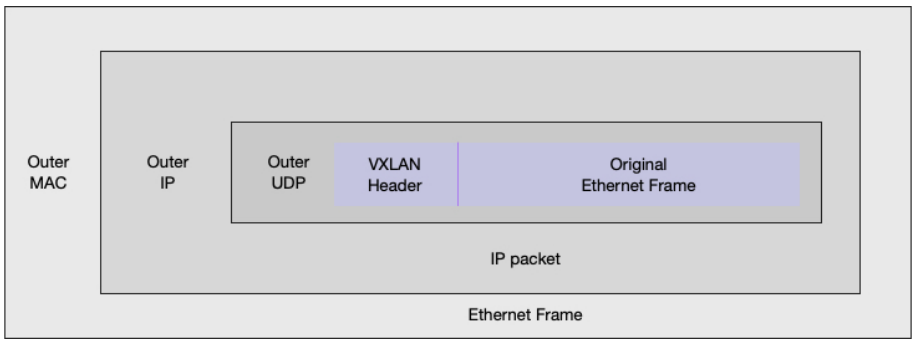
Considering the evolution of L2VPNs, this framework arises from the needs dictated by new applications in contexts such as data centers, cloud, campuses, etc.
The EVPN-VXLAN framework allows, essentially, to create an *overlay virtual network* extended over a network of any type (*underlay network*), in our case an IP network.
In the EVPN model, the BGP protocol (RFC 4271) is used to announce MAC addresses, therefore Mac Learning takes place both on the data plane (local *Mac Learning*) and on the control plane, using the BGP protocol (non-local *Mac Learning*).
In this new model, particular attention is paid to the implementation of *multi-homing connections*, *load balancing* and *loop* prevention.
The *overlay virtual networks* are implemented with the VXLAN (*Virtual eXtensible LAN*) protocol, RFC 7348. In this case, the Ethernet frames generated by the hosts (whether physical or virtual), belonging to the same VXLAN segment, are transported with a *MAC-in-UDP* encapsulation. In practice, for the transport of Ethernet frames, the *switched Ethernet network* is replaced with a "simple" IP network.
Of course, this brings many advantages:

- Absence of loops, with the natural consequence that the use *of Spanning Tree* is no longer necessary;
- Elimination of broadcast storm issues;
- Increased available bandwidth as there are no longer any ports in the *blocking* state;
- Optimal use of bandwidth using *load balancing* techniques typical of IP protocols.

The two main concepts of VXLAN operation are:

- **VNI** (*VXLAN Network Identifier*): the identifier of the various VXLAN segments (24 bits, therefore a total of 16.7 million different VXLAN segments), practically the equivalent of the VLAN tag (12 bits, therefore a maximum of 4096 VLANs that can be implemented);
- **VTEP** (*VXLAN Tunnel End Point*): the function that manages the origin and the termination of the VXLAN segment (they are the logical interfaces on which the VXLAN segments begin and end). In practice, the point where Ethernet frames are encapsulated and decapsulated.

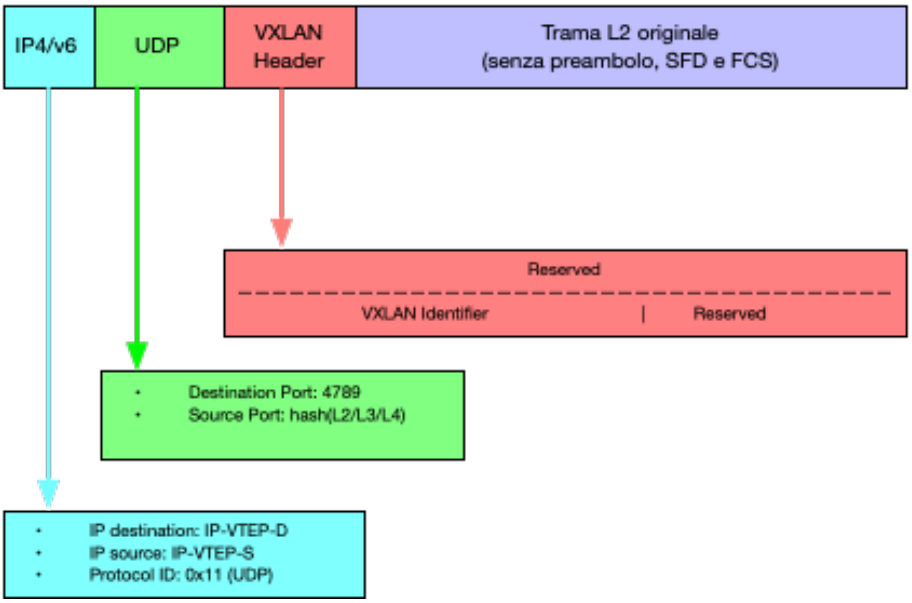
Below is a representation of a VXLAN packet:




*VXLAN Packet Format*

In details:

- Outer *Media Access Control* (MAC) destination address: MAC Address of the destination VTEP endpoint;
- Outer *Media Access Control* (MAC) source address: MAC Address of the source VTEP endpoint;
- Outer IP destination address: IP address of the destination VTEP endpoint;
- Outer IP source address: IP address of the source endpoint;
- Outer UDP: destination port (4789) and source port (L2/L3/L4 hash);
- VXLAN header includes:
    - A 24-bit field for the VNI;
    - Some reserved bits for future use.
-



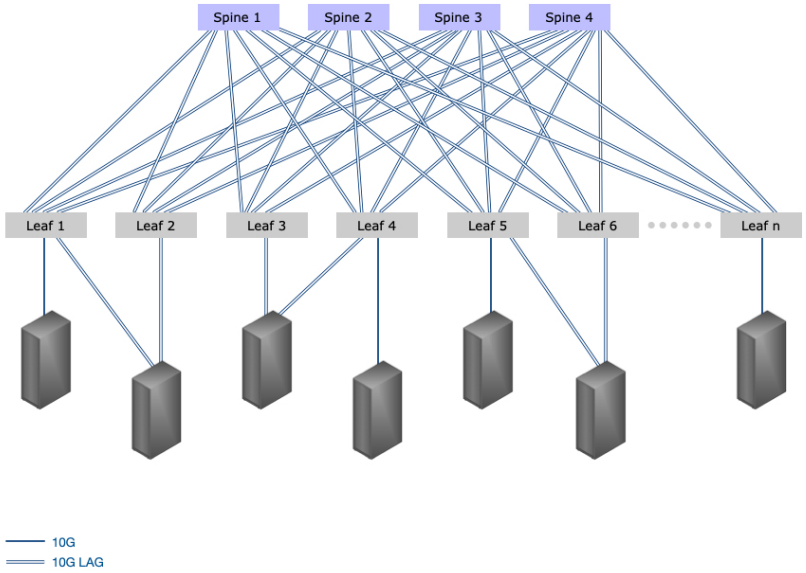

*VXLAN Packet Format in detail*

It is important to note that, since the VXLAN header adds at least 50 bytes (up to a maximum of 54 bytes) to the original Ethernet frame, the MTU parameter **must be** increased in the *underlay network* configuration.

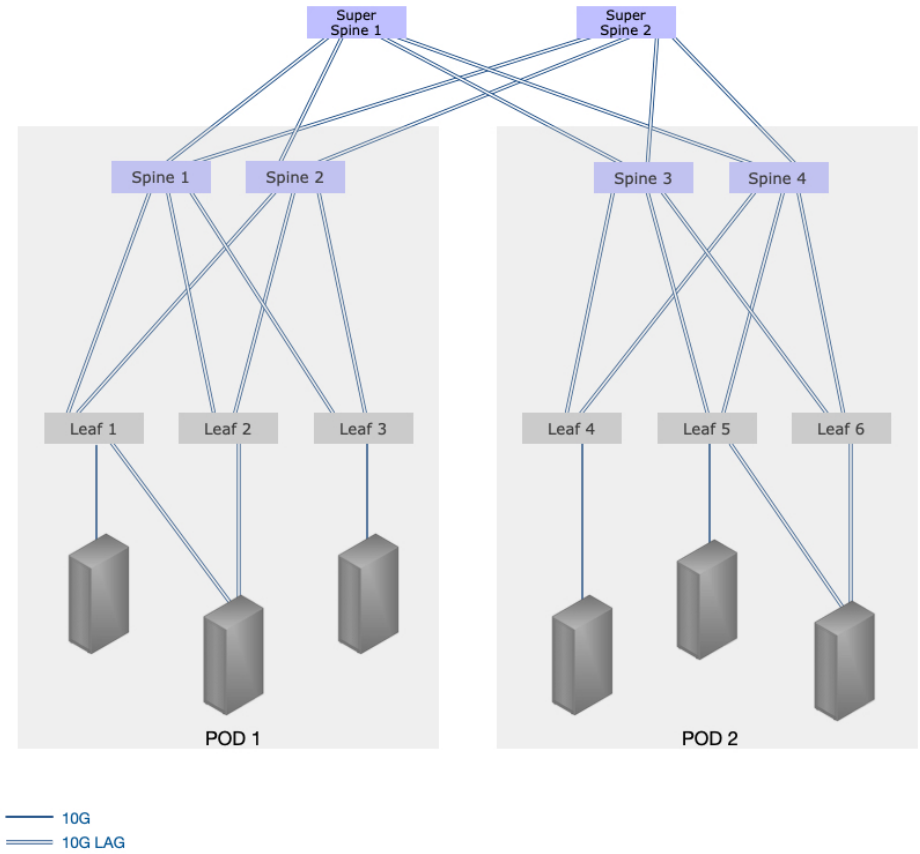
## Supported Topology for EVPN-VXLAN implementation

The following 3 models supported by this framework have been analyzed:

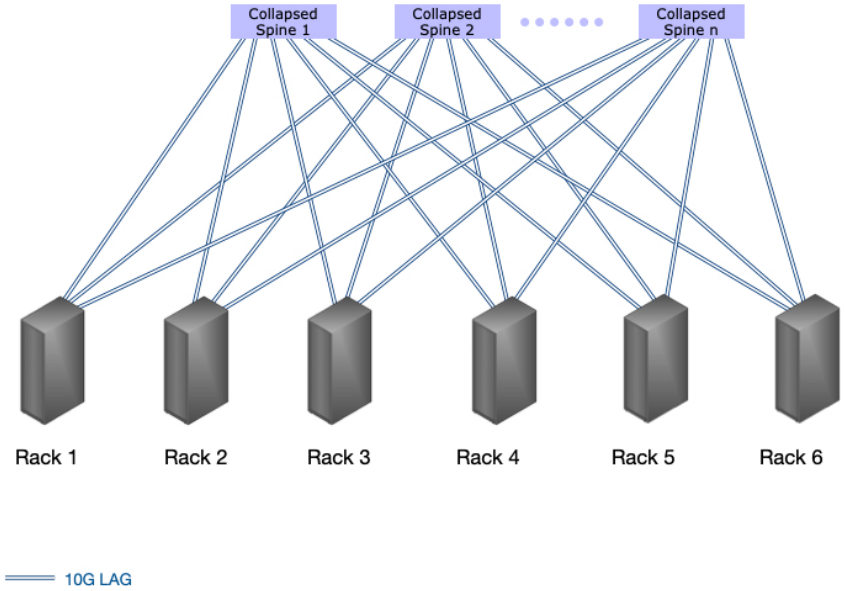
1. *3-stage IP fabric*: this model features a level with devices acting as *spine* and a level with devices acting as *leaf*




2. *5-stage IP fabric*: this is the natural evolution of the previous model in data centers where the fabrics are separate *Poinst of Delivery (PoDs)*. In this model, there is an even higher level of equipment called *super spines* that enable communication between the various *PODs*.



3. *Collapsed Spine IP Fabric*: in this model, the functions of the *leaf* are collapsed into the devices that act as spines.

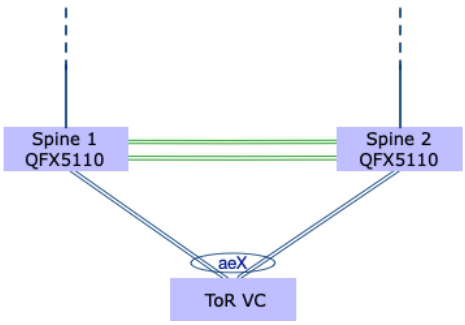



The .it Registry has chosen to experiment the *Collapsed Spine* topology.
As already anticipated, in this technical solution the functionalities of spines and leafs are implemented on the same device.

The devices used for the implementation of this topology are:
- n. 2 QFX5110 Juniper Networks;
- n. 4 QFX5100 Juniper Networks;
- n. 2 EX4300 Juniper Networks.

The two Juniper Networks QFX5110 devices act as *Collapsed Spine* while the 4 QFX5100 (in the configuration of two *virtual-chassis*) and the two EX4300 are the access devices on which both physical and virtual hosts are attested.
In order to increase reliability and resilience and to make the best use of the available bandwidth, the EVPN-VXLAN framework will be implemented with *multi-homing* connections in *active-active* mode. In this configuration, each ToR switch (the *virtual-chassis*) must be connected to each of the spines *with multi-chassis aggregated links (MC-LAG)*.
For *MC-LAG* implementation, the spines use the *Inter-Chassis Control Protocol (ICCP)*, RFC 7275.
However, it is important to highlight some limitations of MC-LAG implementation with ICCP:

- • it is a proprietary technology of Juniper Networks;
- • it does not support more than two spines.



# Configuration

## Configure the Underlay Network

### Configure the ToR switch

1. Configure the maximum number of aggregated links that you want to implement on a single device:

```
set chassis aggregated devices ethernet device count 5
```

2. Configure the interfaces that will be the physical contributors of the aggregated link towards the *spines*:

```
set interfaces xe-0/0/49:0 ether-options 802.3ad ae0
set interfaces xe-1/0/49:0 ether-options 802.3ad ae0
```

3. Configure the aggregated link towards the *spines*:

```
set interfaces ae0 description "to Spines"
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 encapsulation flexible-ethernet-services
set interfaces ae0 aggregated-ether-options minimum-links 1
set interfaces ae0 aggregated-ether-options lacp passive
set interfaces ae0 unit 15 encapsulation vlan-bridge
set interfaces ae0 unit 15 vlan-id 15
set interfaces ae0 unit 16 encapsulation vlan-bridge
set interfaces ae0 unit 16 vlan-id 16
set interfaces ae0 unit 17 encapsulation vlan-bridge
set interfaces ae0 unit 17 vlan-id 17
```

1. Configure the vlan:

```
set vlans test-15 interface ae0.15
set vlans test-16 interface ae0.16
set vlans test-17 interface ae0.17
```

Note: as you can see from the configuration above, the ToR switch can be any switch (even a low-cost one), as long as it supports the implementation of link aggregation (IEEE 802.3ad). In fact, no configuration related to the framework under consideration is implemented on the ToR switch. This allows for a significant reduction in costs.

## Configure Spine 1:

1. Configure the interfaces:

```
set interfaces xe-0/0/10 description "to spine-r2"
set interfaces xe-0/0/10 mtu 9216
set interfaces xe-0/0/10 unit 0 family inet address 192.168.0.1/30
set interfaces xe-0/0/11 description "to spine-r2"
set interfaces xe-0/0/11 mtu 9216
set interfaces xe-0/0/11 unit 0 family inet address 192.168.0.5/30
set interfaces lo0 description "Spine 1 loopback interface"
set interfaces lo0 unit 0 family inet address 192.168.168.1/32
```

2. Configure the EBGP underlay:

```
set protocols bgp group UNDERLAY type external
set protocols bgp group UNDERLAY description "UNDERLAY NETWORK connection"
set protocols bgp group UNDERLAY import underlay-import
set protocols bgp group UNDERLAY family inet unicast
set protocols bgp group UNDERLAY export underlay-export
set protocols bgp group UNDERLAY local-as 65001
set protocols bgp group UNDERLAY multipath multiple-as
set protocols bgp group UNDERLAY neighbor 192.168.0.2 description "Spine 2"
set protocols bgp group UNDERLAY neighbor 192.168.0.2 peer-as 65002
set protocols bgp group UNDERLAY neighbor 192.168.0.6 description "Spine 2"
set protocols bgp group UNDERLAY neighbor 192.168.0.6 peer-as 65002
set protocols bgp log-updown
set protocols bgp graceful-restart
```

3. Configure the import and export policies (to import and export the lo0 interfaces):

```
set policy-options policy-statement underlay-export term loopback from route-filter 192.168.168.0/24 orlonger
set policy-options policy-statement underlay-export term loopback then accept
set policy-options policy-statement underlay-export term default then reject
set policy-options policy-statement underlay-import term loopback from route-filter 192.168.168.0/24 orlonger
set policy-options policy-statement underlay-import term loopback then accept
set policy-options policy-statement underlay-import term default then reject
```

4. Enable ECMP and ECMP *fast reroute* protection. Also enable *load balancing*.
   It should be noted that, although the configuration statement is “load-balance per-packet”, the JunOS operating system performs load balancing per flow. As per the manufacturer's official documentation, to choose the physical link on which to forward the data flow, an hash is calculated based on the source address and the destination address.

```
set policy-options policy-statement ecmp-policy then load-balance per-packet
set routing-options forwarding-table export ecmp-policy
set routing-options forwarding-table ecmp-fast-reroute
```

Note: if a link goes down, ECMP uses the fast reroute protection function to forward packets on the other links without waiting for the routing table update, thus reducing packet loss.

5. Configure ARP table parameters to avoid MAC and MAC-IP synchronization issues in the EVPN-VXLAN environment:

```
set system arp aging-timer 5
set protocols l2-learning global-mac-table-aging-time 600
set protocols l2-learning global-mac-ip-table-aging-time 300
```

## Configure Spine 2:

1. Configure the interfaces:

```
set interfaces xe-0/0/10 description "to spine-r1"
set interfaces xe-0/0/10 mtu 9216
set interfaces xe-0/0/10 unit 0 family inet address 192.168.0.2/30
set interfaces xe-0/0/11 description "to spine-r1"
```

```
set interfaces xe-0/0/11 mtu 9216
set interfaces xe-0/0/11 unit 0 family inet address 192.168.0.6/30
set interfaces lo0 description "Spine 2 loopback interface"
set interfaces lo0 unit 0 family inet address 192.168.168.2/32
```

2. Configure EBGP underlay:

```
set protocols bgp group UNDERLAY type external
set protocols bgp group UNDERLAY description "UNDERLAY NETWORK connection"
set protocols bgp group UNDERLAY import underlay-import
set protocols bgp group UNDERLAY family inet unicast
set protocols bgp group UNDERLAY export underlay-export
set protocols bgp group UNDERLAY local-as 65002
set protocols bgp group UNDERLAY multipath multiple-as
set protocols bgp group UNDERLAY neighbor 192.168.0.1 description "Spine 1"
set protocols bgp group UNDERLAY neighbor 192.168.0.1 peer-as 65001
set protocols bgp group UNDERLAY neighbor 192.168.0.5 description "Spine 1"
set protocols bgp group UNDERLAY neighbor 192.168.0.5 peer-as 65001
set protocols bgp log-updown
set protocols bgp graceful-restart
```

3. Configure the import and export policies (to import and export the lo0 interfaces):

```
set policy-options policy-statement underlay-export term loopback from route-filter
192.168.168.0/24 orlonger
set policy-options policy-statement underlay-export term loopback then accept
set policy-options policy-statement underlay-export term default then reject
set policy-options policy-statement underlay-import term loopback from route-filter
192.168.168.0/24 orlonger
set policy-options policy-statement underlay-import term loopback then accept
set policy-options policy-statement underlay-import term default then reject
```

4. Enable ECMP and ECMP *fast reroute* protection. Also enable *load balancing*:

```
set policy-options policy-statement ecmp-policy then load-balance per-packet
set routing-options forwarding-table export ecmp-policy
set routing-options forwarding-table ecmp-fast-reroute
```

5. Configure ARP table parameters to avoid MAC and MAC-IP synchronization issues in the EVPN-VXLAN environment:

```
set system arp aging-timer 5
set protocols l2-learning global-mac-table-aging-time 600
set protocols l2-learning global-mac-ip-table-aging-time 300
```

## Check the status of UNDERLAY Network

1. Check the status of BGP sessions on the Spine 1:

```
show bgp neighbor 192.168.0.2
Peer: 192.168.0.2+179 AS 65002 Local: 192.168.0.1+49847 AS 65001
  Description: Spine 2
  Group: UNDERLAY              Routing-Instance: master
  Forwarding routing-instance: master
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ underlay-export ] Import: [ underlay-import ]
  Options: <Preference LogUpDown AddressFamily PeerAS Multipath LocalAS Refresh>
  Options: <MultipathAs>
  Options: <GracefulShutdownRcv>
  Address families configured: inet-unicast
  Holdtime: 90 Preference: 170
  Graceful Shutdown Receiver local-preference: 0
  Local AS: 65001 Local System AS: 0
…
```

```
show bgp neighbor 192.168.0.6
Peer: 192.168.0.6+53048 AS 65002 Local: 192.168.0.5+179 AS 65001
  Description: Spine 2
  Group: UNDERLAY          Routing-Instance: master
  Forwarding routing-instance: master
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ underlay-export ] Import: [ underlay-import ]
  Options: <Preference LogUpDown AddressFamily PeerAS Multipath LocalAS Refresh>
  Options: <MultipathAs>
  Options: <GracefulShutdownRcv>
  Address families configured: inet-unicast
  Holdtime: 90 Preference: 170
  Graceful Shutdown Receiver local-preference: 0
  Local AS: 65001 Local System AS: 0
…
```

2. Verify that the loopback interface of Spine 2 is received by Spine 1 via BGP from both neighbors:

```
root> show route receive-protocol bgp 192.168.0.2
inet.0: 9 destinations, 10 routes (9 active, 0 holddown, 0 hidden)
  Prefix                Nexthop         MED    Lclpref    AS path
* 192.168.168.2/32      192.168.0.2                       65002 I
inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

{master:0}
root> show route receive-protocol bgp 192.168.0.6
inet.0: 9 destinations, 10 routes (9 active, 0 holddown, 0 hidden)
  Prefix                Nexthop         MED    Lclpref    AS path
  192.168.168.2/32      192.168.0.6                       65002 I
inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

3. Verify that Spine 2 loopback is reachable from Spine 1:

```
root> ping 192.168.168.2
PING 192.168.168.2 (192.168.168.2): 56 data bytes
64 bytes from 192.168.168.2: icmp_seq=0 ttl=64 time=109.201 ms
64 bytes from 192.168.168.2: icmp_seq=1 ttl=64 time=109.707 ms
64 bytes from 192.168.168.2: icmp_seq=2 ttl=64 time=110.687 ms
64 bytes from 192.168.168.2: icmp_seq=3 ttl=64 time=109.109 ms
64 bytes from 192.168.168.2: icmp_seq=4 ttl=64 time=109.602 ms
^C
--- 192.168.168.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 109.109/109.661/110.687/0.561 ms
```

# Configure Overlay

## Configure Spine 1

1. Configure iBGP between *Spine 1* and *Spine 2*:

```
set protocols bgp group EVPN-Fabric type internal
set protocols bgp group EVPN-Fabric local-address 192.168.168.1
set protocols bgp group EVPN-Fabric family evpn signaling
set protocols bgp group EVPN-Fabric local-as 65100
set protocols bgp group EVPN-Fabric multipath
set protocols bgp group EVPN-Fabric bfd-liveness-detection minimum-interval 1000
set protocols bgp group EVPN-Fabric bfd-liveness-detection multiplier 3
set protocols bgp group EVPN-Fabric neighbor 192.168.168.2 description "lo0 Spine 2"
set protocols bgp group EVPN-Fabric vpn-apply-export
```

2. Configure VLAN and VLAN/VXLAN mapping:

```
set vlans VLAN-002 description "VLAN-002 test"
set vlans VLAN-002 vlan-id 2
set vlans VLAN-002 vxlan vni 5002
```

```
set vlans VLAN-003 description "VLAN-003 test"
set vlans VLAN-003 vlan-id 3
set vlans VLAN-003 vxlan vni 5003
set vlans VLAN-004 description "VLAN-004 test"
set vlans VLAN-004 vlan-id 4
set vlans VLAN-004 vxlan vni 5004
```

3. Configure *switch-options*:
    a. The source interface of the Virtual Tunnel Endpoint (VTEP). It corresponds to the loopback interface of Spine 1;
    b. The route-distinguisher for routes generated by the device;
    c. The route-target.

```
set switch-options vtep-source-interface lo0.0
set switch-options route-distinguisher 192.168.168.1:1
set switch-options vrf-target target:1:999
set switch-options vrf-target auto
```

4. Configure EVPN protocol:
    a. Configure VXLAN as data plane encapsulation;
    b. Configure the VNIs that are part of the EVPN-VXLAN MP-BGP domain.

```
set protocols evpn encapsulation vxlan
set protocols evpn extended-vni-list 5002
set protocols evpn extended-vni-list 5003
set protocols evpn extended-vni-list 5004
```

5. Since the chosen topology only allows for two spines, the *core isolation* feature must be disabled on both spines:

```
set protocols evpn no-core-isolation
```

## Configure Spine 2

1. Configure iBGP between *Spine 1* and *Spine 2*:

```
set protocols bgp group EVPN-Fabric type internal
set protocols bgp group EVPN-Fabric local-address 192.168.168.2
set protocols bgp group EVPN-Fabric family evpn signaling
set protocols bgp group EVPN-Fabric local-as 65100
set protocols bgp group EVPN-Fabric multipath
set protocols bgp group EVPN-Fabric bfd-liveness-detection minimum-interval 1000
set protocols bgp group EVPN-Fabric bfd-liveness-detection multiplier 3
set protocols bgp group EVPN-Fabric neighbor 192.168.168.1 description "lo0 Spine 1"
set protocols bgp group EVPN-Fabric vpn-apply-export
```

2. Configure VLAN and the mapping VLAN/VXLAN:

```
set vlans VLAN-002 description "VLAN-002 test"
set vlans VLAN-002 vlan-id 2
set vlans VLAN-002 vxlan vni 5002
```

```
set vlans VLAN-003 description "VLAN-003 test"
set vlans VLAN-003 vlan-id 3
set vlans VLAN-003 vxlan vni 5003
```

```
set vlans VLAN-004 description "VLAN-004 test"
set vlans VLAN-004 vlan-id 4
set vlans VLAN-004 vxlan vni 5004
```

3. Configure *switch-options*:
    a. The source interface of the Virtual Tunnel Endpoint (VTEP). It corresponds to the loopback interface of Spine 2;
    b. The route distinguisher for routes generated by the device;
    c. The route-target.

```
set switch-options vtep-source-interface lo0.0
set switch-options route-distinguisher 192.168.168.2:1
set switch-options vrf-target target:1:999
set switch-options vrf-target auto
```

4. Configure EVPN protocol:
    a. Configure VXLAN as encapsulation of data plane
    b. Configure VNIs that are part of the EVPN-VXLAN MP-BGP domain.

```
set protocols evpn encapsulation vxlan
set protocols evpn extended-vni-list 5002
set protocols evpn extended-vni-list 5003
set protocols evpn extended-vni-list 5004
```

```
set protocols evpn no-core-isolation
```

## Verify the Overlay

1. Verify if the iBGP session between the two spines is established:

```
show bgp neighbor 192.168.168.1
Peer: 192.168.168.1+65416 AS 65100 Local: 192.168.168.2+179 AS 65100
  Description: lo0 Spine 1
  Group: EVPN-Fabric          Routing-Instance: master
  Forwarding routing-instance: master
  Type: Internal    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference LocalAddress LogUpDown AddressFamily Multipath LocalAS Rib-group
Refresh>
  Options: <VpnApplyExport BfdEnabled>
  Options: <GracefulShutdownRcv>
  Address families configured: evpn
  Local Address: 192.168.168.1 Holdtime: 90 Preference: 170
  Graceful Shutdown Receiver local-preference: 0
  Local AS: 65100 Local System AS: 0
  Number of flaps: 0
  Peer ID: 192.168.168.2   Local ID: 192.168.168.1     Active Holdtime: 90
  Keepalive Interval: 30        Group index: 1    Peer index: 0    SNMP index: 2
…
```

2. Verify the VTEP source for the EVPN domain:

```
show ethernet-switching vxlan-tunnel-end-point source
Logical System Name     Id  SVTEP-IP      IFL  L3-Idx   SVTEP-Mode
<default>           0  192.168.168.1   lo0.0   0
  L2-RTT            Bridge Domain         VNID    MC-Group-IP
  default-switch      VLAN-002+2          5002    0.0.0.0
  default-switch      VLAN-003+3          5003    0.0.0.0
  default-switch      VLAN-004+4          5004    0.0.0.0
```

## NOTEs

Since we have chosen to apply security policies also on east-west inter-vlan traffic, the default gateway of each subnet is configured on the interface of the upstream firewall cluster. This allows for a much leaner and simpler configuration on the spines (especially in case of debugging) avoiding the configuration of *n* routing instances (one for each subnet/vlan) and, for each of them, the routing policies and BGP sessions towards the firewall cluster.
In the event that there is no need to implement inter-vlan security policies, routing instances can be created on the spine devices and configure the default gateways of each subnet within the specific routing instances on both spines. One of the advantages of this configuration is that inter-vlan routing is not performed by the firewall but by the spines, decreasing the latency.

## Configure Multi-homing EVPN

The EVPN multi-homing feature utilizes the Ethernet Segment Identifier (ESI), an attribute that enables *Link Aggregation Group Ethernet Segment Identifier* (ESI-LAG).
As the name suggests, ESI identifies a multihomed segment. The same ESI value must be configured on each spine connected to the same switchToR.

### Configure Spine 1

1. Configure:
    a. ae1 interface;
    b. ESI;
    c. lacp;
    d. The system-id (must be the same on both spines and is used by the ToR1 switch to indicate that the uplinks are part of the same LAG);
    e. Configure ae1 interface and allow it to forward VLAN traffic:

```
set interfaces ae1 description "to ToR1"
```

```
set interfaces ae1 mtu 9216
set interfaces ae1 esi 00:00:00:00:00:00:00:00:01:01
set interfaces ae1 esi all-active
set interfaces ae1 aggregated-ether-options link-speed 10g
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 aggregated-ether-options lacp system-id 00:00:00:00:01:01
set interfaces ae1 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-002
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-003
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-004
```

2. Configure the physical interface as physical contributors of the LAG ae1:

```
set interfaces xe-0/0/0 ether-options 802.3ad ae1
```

## Configure Spine 2

1. Configure:
    a. ae1 interface;
    b. ESI;
    c. lacp;
    d. The system-id (must be the same on both spines and is used by the ToR1 switch to indicate that the uplinks are part of the same LAG);
    e. Configure ae1 interface and allow it to forward VLAN traffic:

```
set interfaces ae1 description "to ToR1"
set interfaces ae1 mtu 9216
set interfaces ae1 esi 00:00:00:00:00:00:00:00:01:01
set interfaces ae1 esi all-active
set interfaces ae1 aggregated-ether-options link-speed 10g
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 aggregated-ether-options lacp system-id 00:00:00:00:01:01
set interfaces ae1 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-002
```

```
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-003
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-004
```

1. Configure the physical interface as physical contributors of the LAG ae1:

```
set interfaces xe-0/0/0 ether-options 802.3ad ae1
```

### Configure ToR1

1. Configure:
    a. The uplink interface and its physical contributors;
    b. vlan

```
set interfaces xe-0/0/10 ether-options 802.3ad ae0
set interfaces xe-0/0/11 ether-options 802.3ad ae0
set interfaces ae0 mtu 9216
set interfaces ae0 aggregated-ether-options minimum-links 1
set interfaces ae0 aggregated-ether-options link-speed 10g
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp periodic fast
set interfaces ae0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 0 family ethernet-switching vlan members VLAN-002
set interfaces ae0 unit 0 family ethernet-switching vlan members VLAN-003
set interfaces ae0 unit 0 family ethernet-switching vlan members VLAN-004
set vlans VLAN-002 vlan-id 2
set vlans VLAN-003 vlan-id 3
set vlans VLAN-004 vlan-id 4
```

## Verify EVPN Multi-homing

1. Check the status of the ae1 interface and the ESI value associated with the LAG:

```
show interfaces ae1
Physical interface: ae1, Enabled, Physical link is Up
  Interface index: 662, SNMP ifIndex: 559
```

```
  Description: to ToR1
  Link-level type: Ethernet, MTU: 9216, Speed: 10Gbps, BPDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Disabled, Minimum links needed: 1,
  Minimum bandwidth needed: 1bps
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Current address: 02:05:86:72:e0:ef, Hardware address: 02:05:86:72:e0:ef
  Ethernet segment value: 00:00:00:00:00:00:00:00:01:01, Mode: all-active
  Last flapped   : 2022-10-05 23:36:22 UTC (06:37:41 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 968 bps (0 pps)
…
```

2. Verify lacp:

```
show lacp interfaces ae1
Aggregated interface: ae1
   LACP state:      Role   Exp   Def  Dist  Col  Syn  Aggr  Timeout  Activity
     xe-0/0/0     Actor    No    No   Yes  Yes  Yes   Yes    Fast    Active
     xe-0/0/0   Partner    No    No   Yes  Yes  Yes   Yes    Fast    Active
   LACP protocol:       Receive State  Transmit State          Mux State
     xe-0/0/0              Current   Fast periodic Collecting distributing
```

3. Verify that the EVPN Multi-homing state, in the EVPN instance, is resolved.:

```
show evpn instance extensive
Instance: __default_evpn__
  Route Distinguisher: 192.168.168.1:0
  Number of bridge domains: 0
  Number of neighbors: 1
    Address          MAC    MAC+IP      AD      IM ES Leaf-label    Remote DCI Peer
    192.168.168.2      0       0       0       0       1
```

```
Instance: default-switch
  Route Distinguisher: 192.168.168.1:1
  Encapsulation type: VXLAN
  Duplicate MAC detection threshold: 5
  Duplicate MAC detection window: 180
  MAC database status               Local  Remote
    MAC advertisements:               0      0
    MAC+IP advertisements:            0      0
    Default gateway MAC advertisements:    0      0
  Number of local interfaces: 2 (2 up)
    Interface name  ESI                     Mode         Status    AC-Role
    .local..5       00:00:00:00:00:00:00:00:00:00  single-homed    Up       Root
    ae1.0           00:00:00:00:00:00:00:01:01  all-active      Up       Root
  Number of IRB interfaces: 0 (0 up)
  Number of protect interfaces: 0
  Number of bridge domains: 3
    VLAN  Domain ID   Intfs / up   IRB intf   Mode MAC sync  IM route label  IPv4 SG sync
IPv4 IM core nexthop  IPv6 SG sync  IPv6 IM core nexthop Trans Domain ID
    2    5002        1   1          Extended     Enabled  5002        Disabled
Disabled            5002
    3    5003        1   1          Extended     Enabled  5003        Disabled
Disabled            5003
    4    5004        1   1          Extended     Enabled  5004        Disabled
Disabled            5004
  Number of neighbors: 1
    Address          MAC    MAC+IP      AD      IM ES Leaf-label    Remote DCI Peer
    192.168.168.2        0       0       2       0       0
  Number of ethernet segments: 1
    ESI: 00:00:00:00:00:00:00:01:01
      Status: Resolved by IFL ae1.0
      Local interface: ae1.0, Status: Up/Forwarding
      Number of remote PEs connected: 1
        Remote PE       MAC label  Aliasing label  Mode
        192.168.168.2   0          0               all-active
      DF Election Algorithm: MOD based
```

```
    Designated forwarder: 192.168.168.1
    Backup forwarder: 192.168.168.2
    Last designated forwarder update: Oct 06 00:35:09
  Router-ID: 192.168.168.1
  SMET Forwarding: Disabled
```

4. Check the status of the members of the ae0 interface on the ToR1 switch:

```
show lacp interfaces ae0
Aggregated interface: ae0
  LACP state:      Role   Exp   Def  Dist  Col  Syn  Aggr  Timeout  Activity
    xe-0/0/10    Actor    No    No   Yes  Yes  Yes   Yes    Fast   Active
    xe-0/0/10  Partner    No    No   Yes  Yes  Yes   Yes    Fast   Active
    xe-0/0/11    Actor    No    No   Yes  Yes  Yes   Yes    Fast   Active
    xe-0/0/11  Partner    No    No   Yes  Yes  Yes   Yes    Fast   Active
  LACP protocol:        Receive State  Transmit State          Mux State
    xe-0/0/10             Current   Fast periodic Collecting distributing
    xe-0/0/11             Current   Fast periodic Collecting distributing
```