

NSX-T Data Center Edge Bridge

This document was updated for NSX-T 3.1 (François Tallet @vmware)

Contents

- Use case for the NSX-T Edge Bridge 2
- Overview of the solution 3
 - DPDK-based performance 3
 - Extend an Overlay-backed Segment to a VLAN 3
 - Guest VLAN tagging scenarios 5
 - Segment associated to a single VLAN ID 5
 - Segment associated to a range of VLAN IDs 5
 - High Availability 7
 - Segment Load balancing 8
 - Firewall 10
 - Seamless integration with NSX-T routers 11
- Configuration example 11
 - Bridge Profile 13
 - Associating a Segment to a Bridge Profile 14
- High availability 16
 - Edge node availability 16
 - Bridge-specific Edge HA 17
 - Bridge failover scenarios 17
 - Failure of Edge1 18
 - Failure of the VLAN uplink of Edge1 18
 - Failure of the tunnel between Edge1 and Edge2 18
 - Failure of the path on the VLAN side 19
- Mac address tables updates 20
 - Populating the mac address tables 20
 - Updating the mac address tables after switchover 21
- Design Considerations 21
 - Edge form factor 21
 - Edge VM: Forged transmit and promiscuous or mac learning 21
 - Edge VM: Virtual Guest Tagging 22
 - Edge VM configuration example for the Bridge 22
 - Edge VM: Edge uplink protection 23
 - Redundant VLAN connectivity 24
 - Preemptive vs. non-preemptive 25
 - Performance 26
- CLI, Operations 26
- Limitations, caveats 29
 - NSX-T 3.1 limitations 29
 - VLAN conflict on the Edge 29
- Summary, TL;DR. 30

Use case for the NSX-T Edge Bridge

NSX-T Data Center leverages the overlay model, where layer 2 connectivity between virtual machines (VMs) is achieved using point-to-point tunnels over a traditional routed IP network. This provides the last building block for a complete virtualization of the datacenter, where a solution can be deployed programmatically with no dependency on the physical infrastructure.

Even in highly virtualized environments, customers often have some few workloads that cannot be virtualized, because of licensing or application-specific reasons. Those VLAN backed workloads typically communicate with overlay backed VMs at layer 3, through Tier0 gateways instantiated on the NSX-T edges. However, there are some scenarios where layer 2 connectivity is required between VMs and physical devices, and this paper introduces the NSX-T edge bridge, a service that can be instantiated on an edge for that purpose.

The most common use cases for the feature are:

- Physical to virtual/virtual to virtual migration. This is generally a temporary scenario where a VLAN backed environment is being virtualized to an overlay backed NSX data center. The NSX-T edge bridge is a simple way to maintain connectivity between the different components during the intermediate stages of the process.

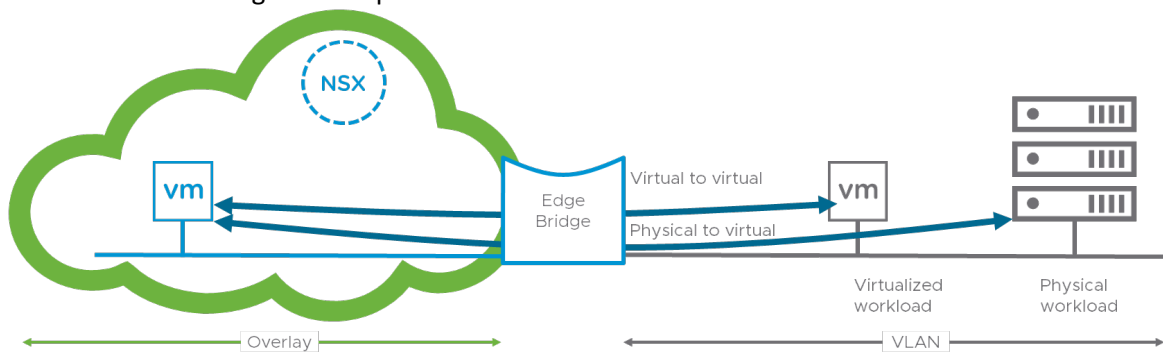


Figure 1: physical to virtual or virtual to virtual migration use case

Note that even if migration scenarios, which are temporary by nature, are a popular use case for the edge bridge, deploying a bridge as a permanent solution is fully supported.

- Integration of physical, non-virtualized appliances that require L2 connectivity to the virtualized environment. The most common example is a database server that requires L2 connectivity because L3 connectivity has not been validated and is not supported by the vendor. This could also be the case of a service appliance that need to be inserted inline, like a physical firewall or load balancer.

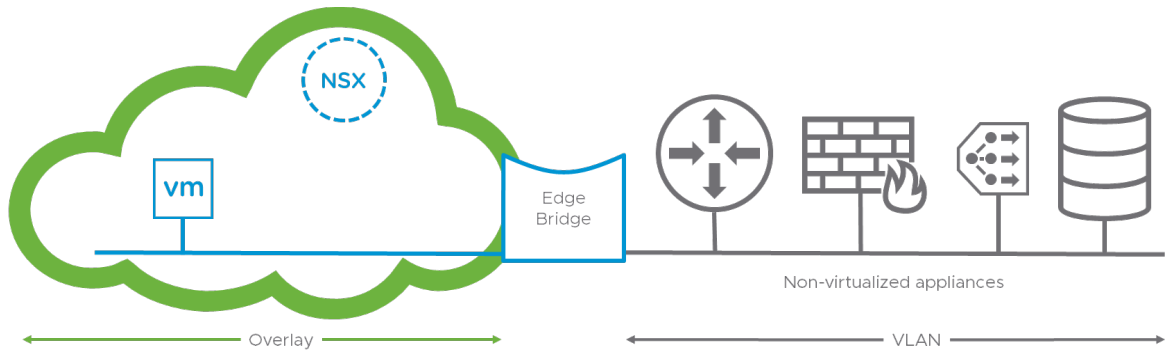


Figure 2: integration of non-virtualized appliances use case

Whether it is for migration purposes or for integration of non-virtualized appliances, if L2 adjacency is not needed, leveraging a gateway on the edges is typically more efficient, as routing allows for Equal Cost Multipathing (ECMP), which results in higher bandwidth and a better redundancy model.

Overview of the solution

The following sections present the capabilities of the NSX-T edge bridge.

DPDK-based performance

One of the main benefits of running a bridge on the NSX-T edge is the data plane performance. Indeed, the NSX-T edge is leveraging the Data Plane Development Kit (DPDK), providing low latency, high bandwidth and scalable traffic forwarding performance.

Extend an Overlay-backed Segment to a VLAN

In its most simple representation, the only thing the NSX-T edge bridge achieves is to convert an ethernet frame between two different representations: overlay and VLAN. In the overlay representation, the L2 frame and its payload are encapsulated in an IP-based format (NSX-T Data Center currently leverages GENEVE, Generic Network Virtualization Encapsulation). In the VLAN representation, the L2 frame include an 802.1Q VLAN tag. The edge bridge can be configured to make a one-to-one association between an overlay-backed segment identified by a VNI (a Virtual Network Identifier in the overlay header) and a specific VLAN ID.

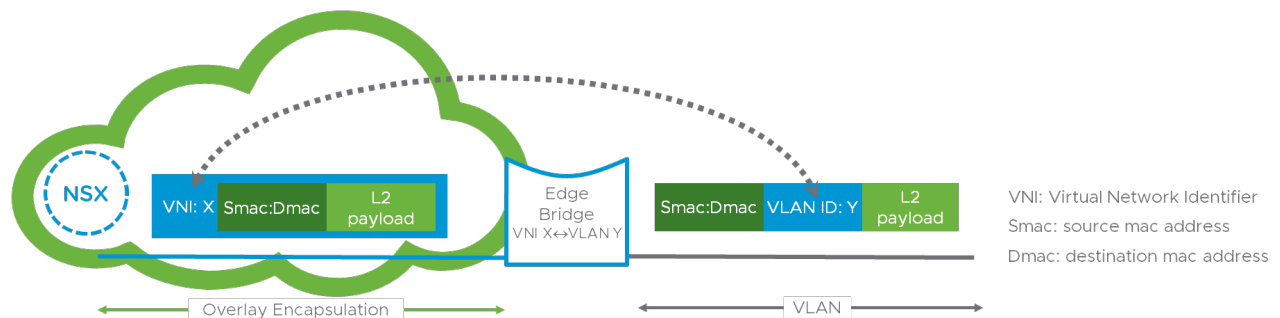


Figure 3: One-to-one association between segment with VNI X and VLAN with VLAN ID Y

Up to NSX-T 2.4 included, a single such edge bridge could be configured for a given segment. That meant that L2 traffic could only enter and leave the NSX overlay in a single location, thus preventing the possibility of a loop between VLAN and overlay.

NSX-T 2.5 introduced the capability of attaching several edge bridges to the same segment. This is convenient when multiple racks (or multiple data centers) need to have local L2 access to non-virtualized devices. Instantiating a bridge in each rack (resp. data center) allows relying on the overlay to connect at Layer 2 those racks (resp. data centers), without extending VLANs between them in the physical infrastructure. The following diagram is representing an NSX segment extended to two different VLANs, in different locations:

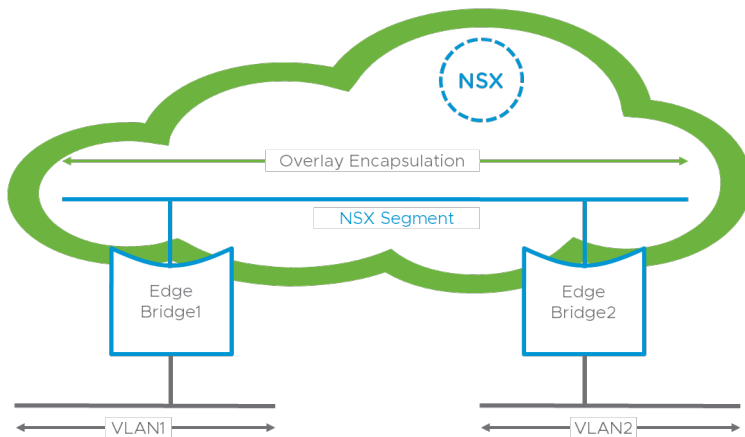


Figure 4: Bridging a segment in multiple locations

Note that VLAN1/VLAN2 don't refer to different VLAN IDs here but different VLANs (a VLAN being an L2 broadcast domain.) What matters is that there is no L2 connectivity in the physical network between those two VLANs, they might (or might not) use the same VLAN ID.

This point is extremely important as there is absolutely no loop prevention mechanism in NSX. If the administrator ends up bridging a segment multiple times to the same VLAN (i.e. the same L2 broadcast domain), a permanent bridging loop might be created, as represented in the diagram below:

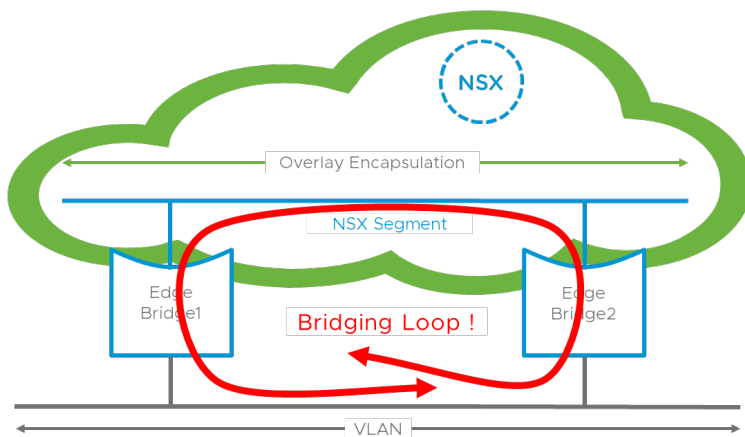


Figure 5: A misconfiguration can lead to a bridging loop

So be careful that the additional flexibility introduced in NSX-T 2.5 comes with additional responsibility. If you are not comfortable with this model, because you don't have enough control on the physical infrastructure for instance, make sure that you only attach a segment to a single edge bridge.

Guest VLAN tagging scenarios

When the user is doing Guest VLAN tagging, the VM inject dot1Q tagged traffic into the NSX segment they are attached to. The edge bridge can be configured in two different ways:

Segment associated to a single VLAN ID

The most common scenario is to make a 1:1 association between an NSX segment and a VLAN ID. This is the only option that has been presented in this document so far. In the diagram below, the NSX segment with VNI X has been associated to VLAN ID Y. If the frame carried by the NSX segment includes an 802.1Q tag, it is just ignored by the edge bridge and simply carried as part of the L2 payload.

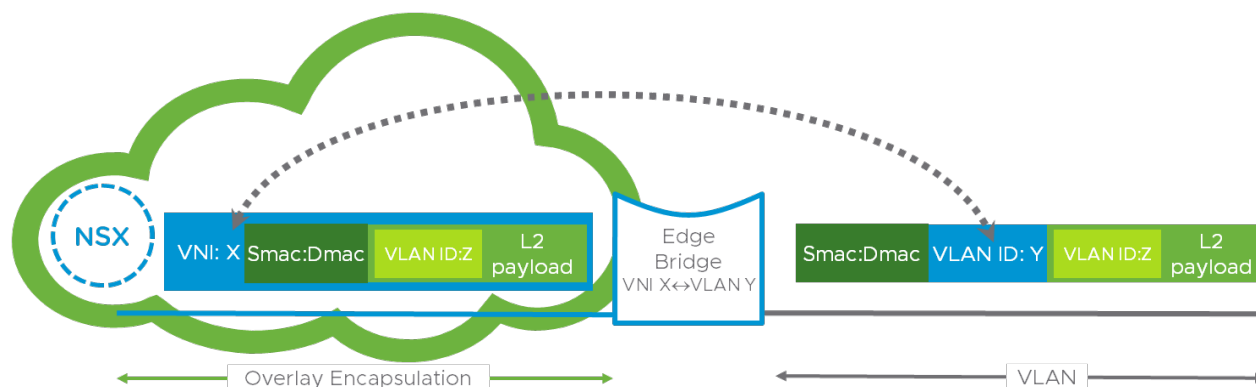


Figure 6: Guest VLAN Tagging with an NSX Segment associated to a single VLAN ID

That means that the resulting frame injected on the VLAN side is doubled tagged (something that is commonly referred to as a QinQ encapsulation.) Note that the behavior of the bridge is identical whether the frame carried by the NSX segment is tagged or not: the inner tag is completely unused by the bridge.

Segment associated to a range of VLAN IDs

Since NSX-T 3.0, an NSX segment attached to an edge bridge can be associated to a range of VLAN IDs. When doing so, the following two differences are introduced in the NSX bridge behavior compared to the case where the segment is mapped to a single VLAN ID:

- The 802.1Q tag within the frame carried by the NSX segment is parsed by the bridge. The bridge will only forward the traffic to VLAN if this 802.1Q tag is within the range configured for the segment.
- The NSX bridge does not add a VLAN tag when forwarding to the VLAN side. It just strips the overlay encapsulation, exposing the inner dot1Q tag as the VLAN tag of the frame.

The following diagram represents this process. An NSX segment with VNI X is mapped to the VLAN ID range VLAN A-VLAN B. Supposed that the NSX segment carries a dot1Q tagged frame with VLAN ID Z, with Z in the [A-B] range. In that case, the bridge will forward this frame to the VLAN side, stripping the

overlay encapsulation. This will appear as a dot1Q frame tagged with VLAN ID Z to the networking infrastructure receiving this traffic.

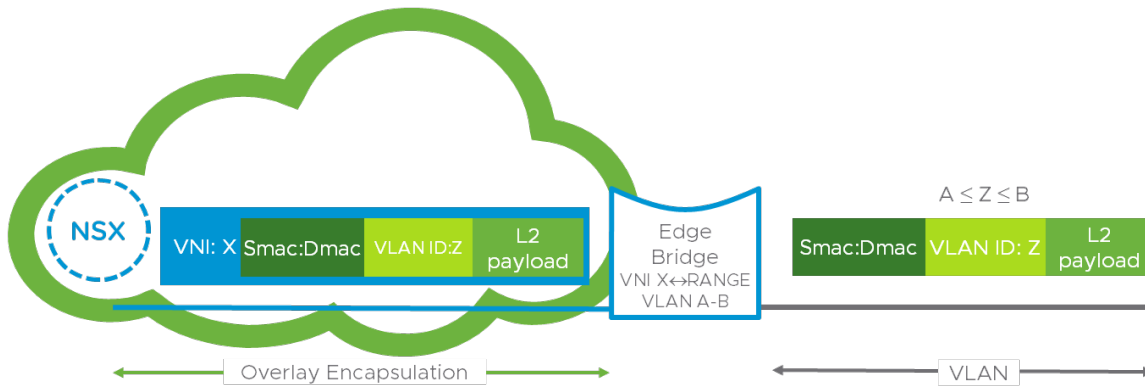


Figure 7: Guest VLAN Tagging with an NSX segment associated to a range of VLAN IDs

Of course, the symmetric forwarding behavior will happen in the opposite direction. If the edge bridge receives a tagged frame with VLAN ID Z, and Z is in the range [A,B] associated to VNI X, it will forward the unmodified to the overlay side, encapsulated with VNI X.

Multiple bridges can be configured on the same physical uplink of the edge, with VLAN ranges associated to different NSX segments. The following diagram represent this scenario, where NSX segments with VNI X and Y are associated to VLAN ID ranges [A,B] and respectively [C,D] on the edge.

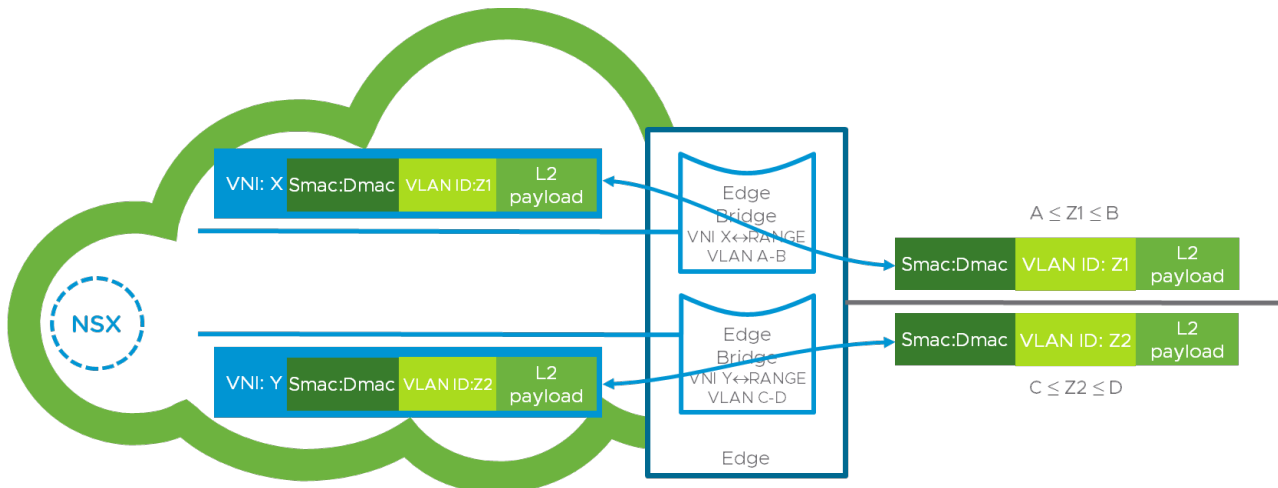


Figure 8: Multiple bridges configured on the same Edge VLAN uplink, with disjoint ranges of VLAN IDs

When traffic flow from the VLAN side to the Overlay side, the VLAN ID in the frame is used to identify the segment to which this frame will be bridge. Thus, the VLAN ranges configured on the edge bridge must not intersect, otherwise the bridge would not be able to make a deterministic decision as to which segment the traffic needs to be bridged to.

A frame with a VLAN Z that is not included in the mapping of any segment on this edge will be dropped by the bridge (whether it is received on the VLAN or overlay side). So in the above example, an

hypothetical frame with VLAN tag Z3, with Z3 outside of range [A,B] and [C,D], will be dropped by the bridge.

Finally, as it is traditionally used within VMware, VLAN 0 means “untagged” traffic. So, an untagged frame will be forwarded by a bridge to/from the segment that includes 0 in its configured VLAN range mapping.

High Availability

The edge bridge operates as an active/standby service. The bridge active in the data path is optionally backed by a unique, pre-determined standby bridge on a different edge.

NSX-T edges are deployed in a pool called an edge cluster. Within an edge cluster, the user can create a bridge profile, which essentially designates two edges as the potential hosts for a pair of redundant bridges. The bridge profile specifies which edge would be primary (i.e. the preferred host for the active bridge) and backup (the edge that will host the standby bridge). At the time of the creation of the bridge profile, no bridge is instantiated yet. The bridge profile is just a template for the creation of one or several bridge pairs.

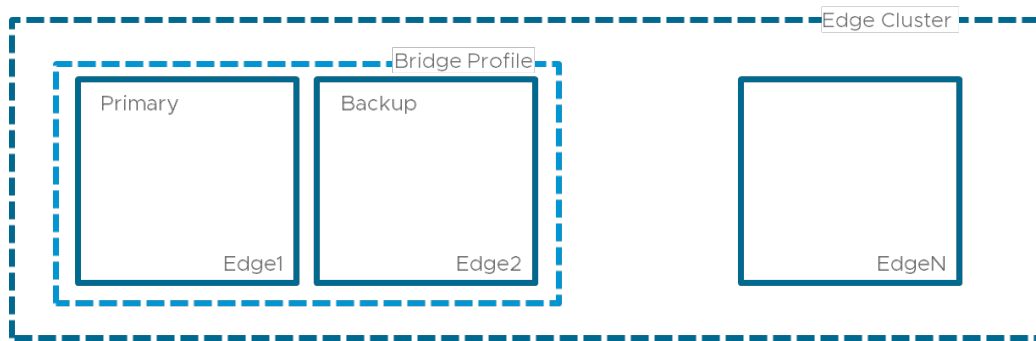


Figure 9: Bridge Profile, defining a redundant Edge Bridge (primary and backup)

Once a bridge profile is created, the user can attach a segment to it. By doing so, an active bridge instance is created on the primary edge, while a standby bridge is provisioned on the backup edge. NSX creates a bridge endpoint object, which represents the connectivity to this pair of bridges. The attachment of the segment to the bridge endpoint is represented by a dedicated logical port, as shown in the diagram below:

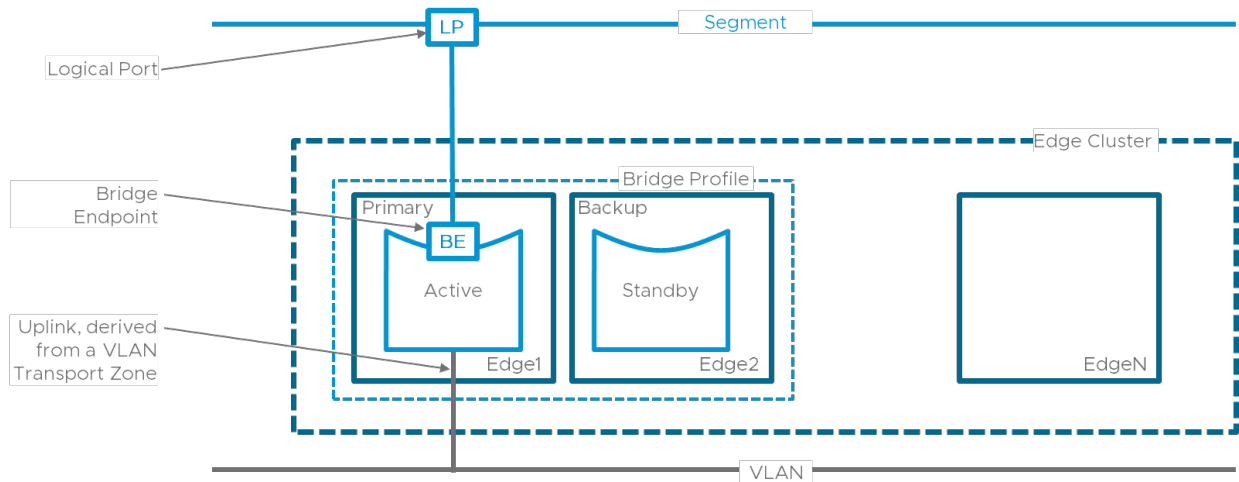


Figure 10: Primary Edge Bridge is active forwarding traffic between Segment and VLAN

At the time of the creation of the bridge profile, the user can select the failover mode. In the preemptive mode, the bridge on the primary edge will attempt to become the active bridge (forwarding traffic between overlay and VLAN) as soon as it is available. In the non-preemptive mode, the bridge on the primary edge will remain standby should it become available when the bridge on the backup edge is already active (redundancy will be discussed in detail in the high availability section of the document below.)

When associating a segment to a bridge profile, the user specifies a VLAN ID or a range of VLAN IDs, as well as a VLAN transport zone, that will determine the port of the edge that carries the VLAN traffic.

Segment Load balancing

Multiple bridge profiles can be configured, and a given edge can belong to several bridge profiles. By creating two separate bridge profiles, alternating active and backup edge in the configuration, the user can easily make sure that two edge nodes simultaneously bridge traffic between overlay and VLAN. The diagram below shows two edges with two pairs of redundant edge bridges (numbered 1 and 2.) The configuration defines the Primary 1 on Edge 1 and Primary 2 on Edge 2. With preemption configured, this ensures that when both edges are available, both are active for bridging traffic and traffic is load balanced on a per-segment basis.

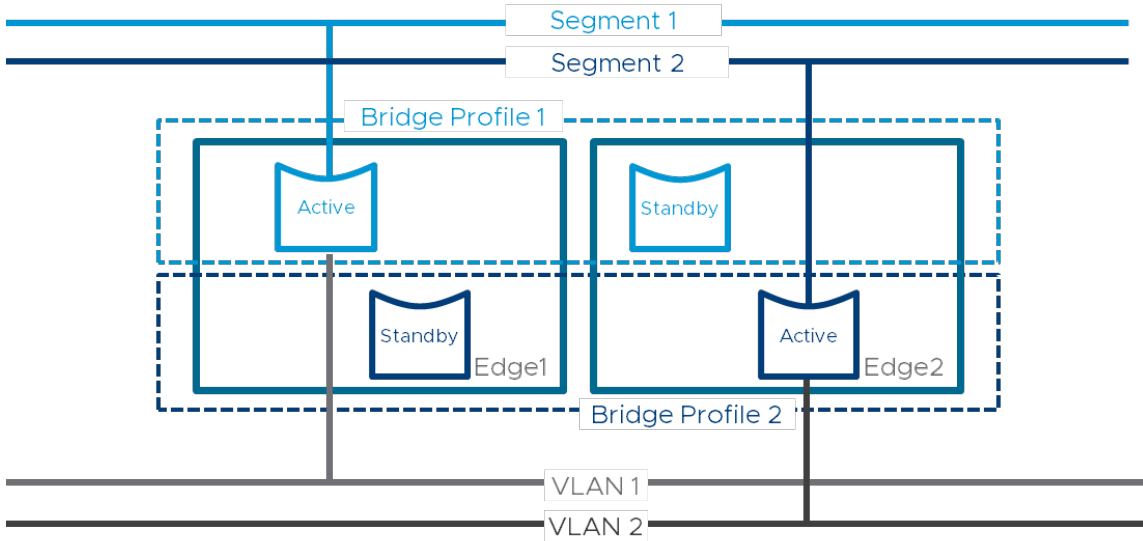
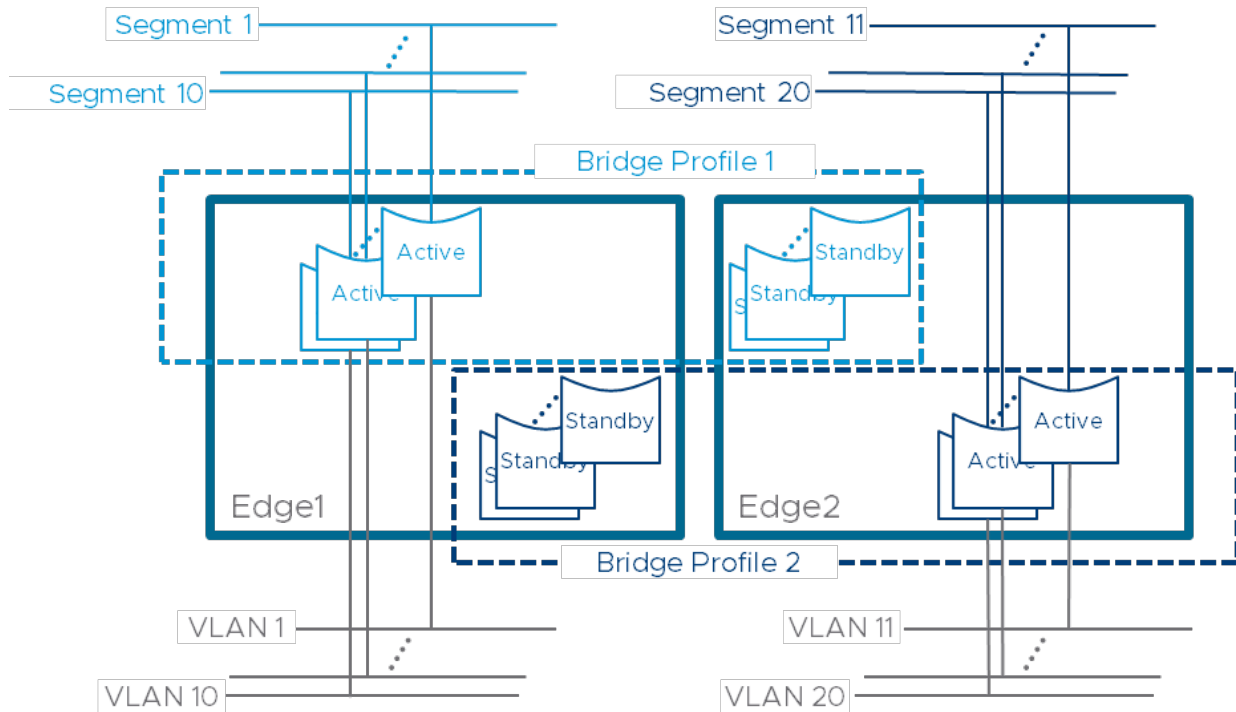


Figure 11: Load-balancing bridged traffic for two Segments over two Edges (Edge Cluster omitted for clarity.)

Note that multiple bridges can be associated to a single bridge profile. In the example below, there are 20 overlay segments that need to be bridged to VLAN across our two edges. There is no need to create 20 bridge profile. We can just map half of the segments to the bridge profile using Edge1 as primary and the other half to the bridge profile using Edge2 as primary. This way, the bridged traffic for all 20 segments is load balanced across the two edges.



Of course, further scale out can be achieved with more Edge nodes. The following diagram shows an example of three Edge Nodes active at the same time for three different segments. There is total flexibility in the configuration of the bridge profile within an edge cluster.

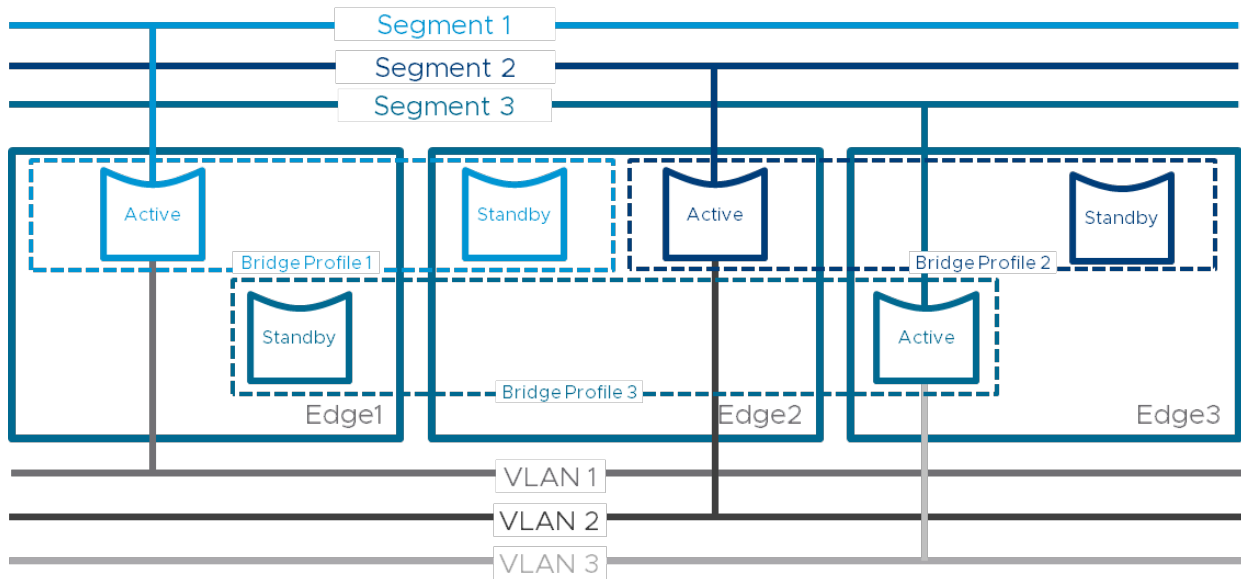


Figure 12: Load-balancing example across three Edge nodes (Bridge Profiles not shown for clarity.)

Note that if several bridge profiles can be configured on the same edges, a given bridge profile cannot specify more than two edge nodes.

Firewall

The traffic leaving and entering a segment via a bridge is subject to the bridge firewall. Rules are defined on a per-segment basis and are defined for the bridge as a whole, i.e. they apply to the active bridge instance, irrespective of the edge on which it is running.

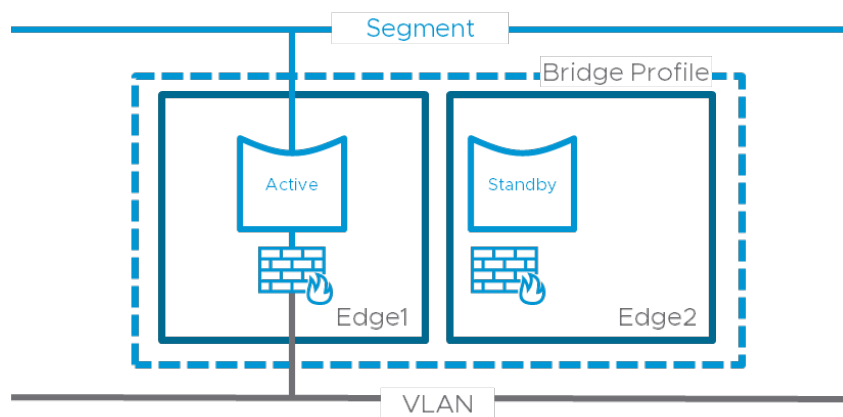


Figure 13: Edge Bridge Firewall

The firewall rules can leverage existing NSX-T grouping constructs, and there is currently a single firewall section available for those rules.

Seamless integration with NSX-T routers

Routing and bridging seamlessly integrate. Distributed routing is available to segments extended to VLAN by a bridge. The following diagram is a logical representation of a possible configuration leveraging T0 and T1 gateways along with edge bridges.

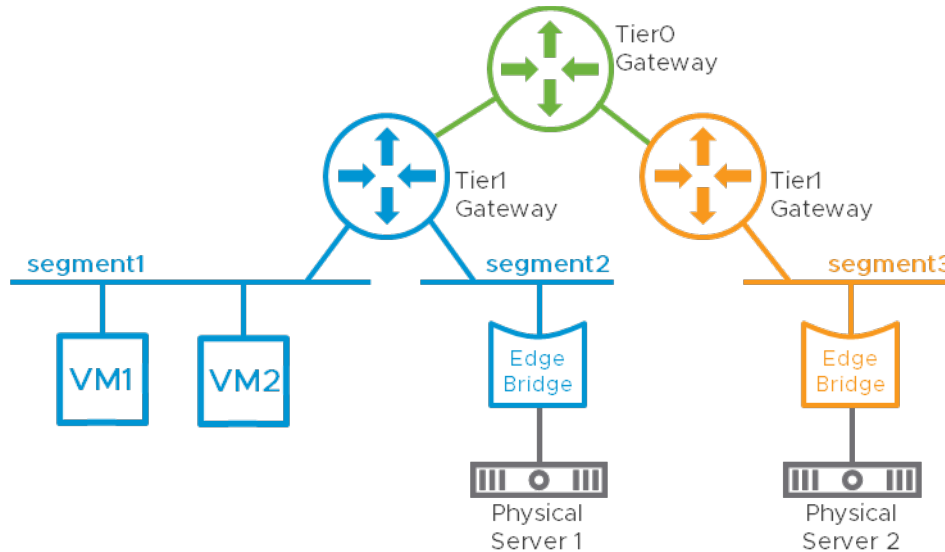


Figure 14: Integration with routing

In this above example, VM1, VM2, Physical Servers 1 and 2 have IP connectivity. Remarkably, through the edge bridges, Tier1 or Tier0 gateways can act as default gateways for physical devices. Note also that the distributed nature of NSX-T routing is not affected by the introduction of an edge bridge. ARP requests from physical workload for the IP address of an NSX router acting as a default gateway will be answered by the local distributed router on the edge where the bridge is active.

Configuration example

This section introduces the configuration workflow for an edge bridge using a minimalist example. The goal here is to provide L2 connectivity between a virtual machine VM1, attached to segment S1 and a physical server connected into VLAN 1000.

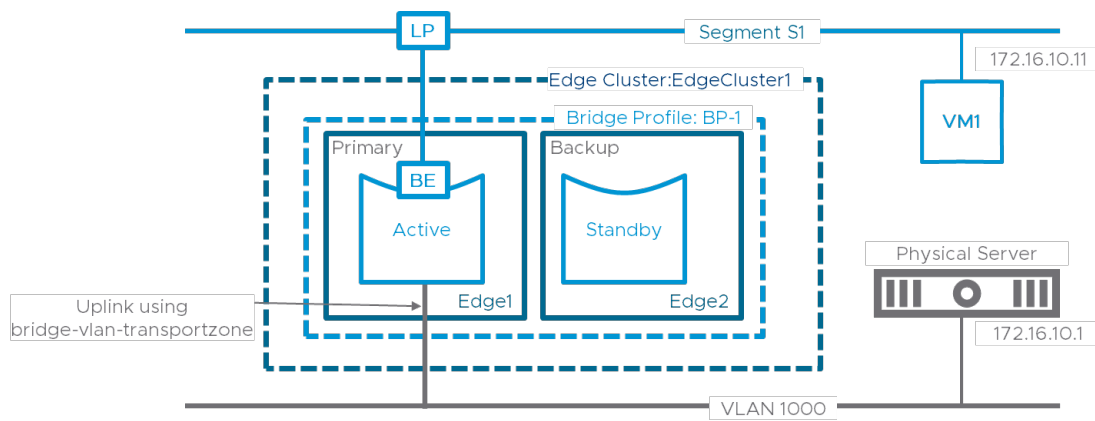


Figure 15: Logical lab topology

For this example, we’re going to assume that two edges, Edge1 and Edge2 are already deployed in an edge cluster “EdgeCluster1”. For the sake of simplicity, they only have one uplink, carrying both overlay and VLAN traffic. Many other combinations are possible, see the part specific to the VM form factor of the edge later in the document for more design considerations.

The N-VDS of the edges has its uplink attached to:

- An overlay transport zone called “nsx-overlay-transportzone”. This is the default overlay transport zone created by NSX-T 3.0, but it could be any other overlay transport zone defining the overlay segments you want to extend to VLAN through the bridge.
- A VLAN transport zone called “bridge-vlan-transportzone”. This VLAN transport zone was created for the edge bridges. You could potentially re-use the default NSX-T VLAN transport zone (called “nsx-vlan-transportzone”) but it’s better to avoid configuring a VLAN transport zone that might have some VLAN segments defined in it as they could conflict with the VLAN IDs the edge bridge is trying to use (check part “Limitations, caveats” at the end of the document.)

The following diagram represents a possible physical implementation of this lab where the two edges are in bare metal form factor.

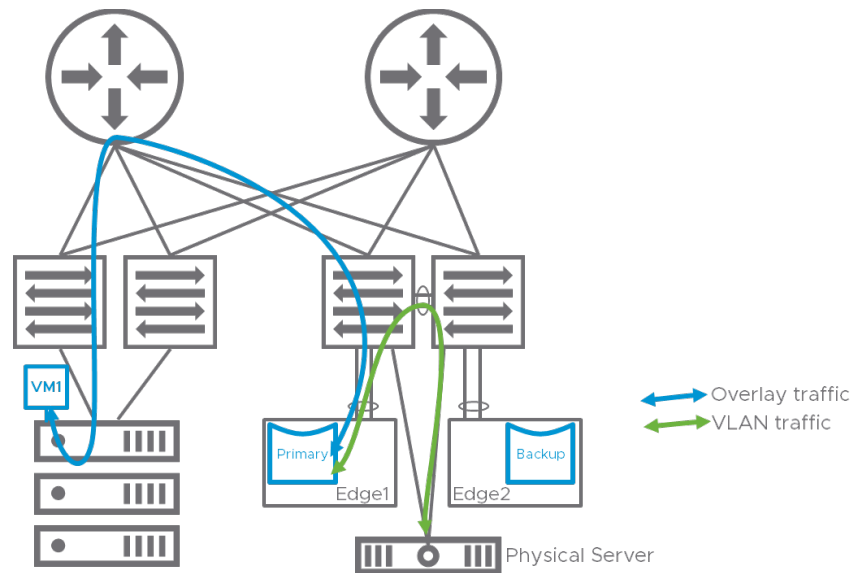


Figure 16: possible physical topology for the example lab.

The bare metal edges are connected via a LAG to a single top of rack switch. Both overlay and VLAN traffic is carried over this LAG. The traffic between VM1 and the physical server is pinned to the active bridge on Edge 1. The overlay part of the path of the traffic between VM1 and the physical server is represented with a solid blue line, while the VLAN part of the path is represented with a solid green line. Note that the overlay traffic is leveraging a routed spine in the network, while there is direct VLAN connectivity between the Edges and the physical server.

Bridge Profile

In order to realize this example setup, we first need to create a bridge profile that will identify Edge1 and Edge2 as the physical appliances where the bridging service must be deployed.

Here is the workflow in the NSX-T 3.0 UI:

1. Select “Networking”
2. Select “Segments” in the Connectivity tab on the left
3. Click “Edge Bridge Profiles”
4. and finally, click “+ADD EDGE BRIDGE PROFILE”

The sequence is represented in the figure below:

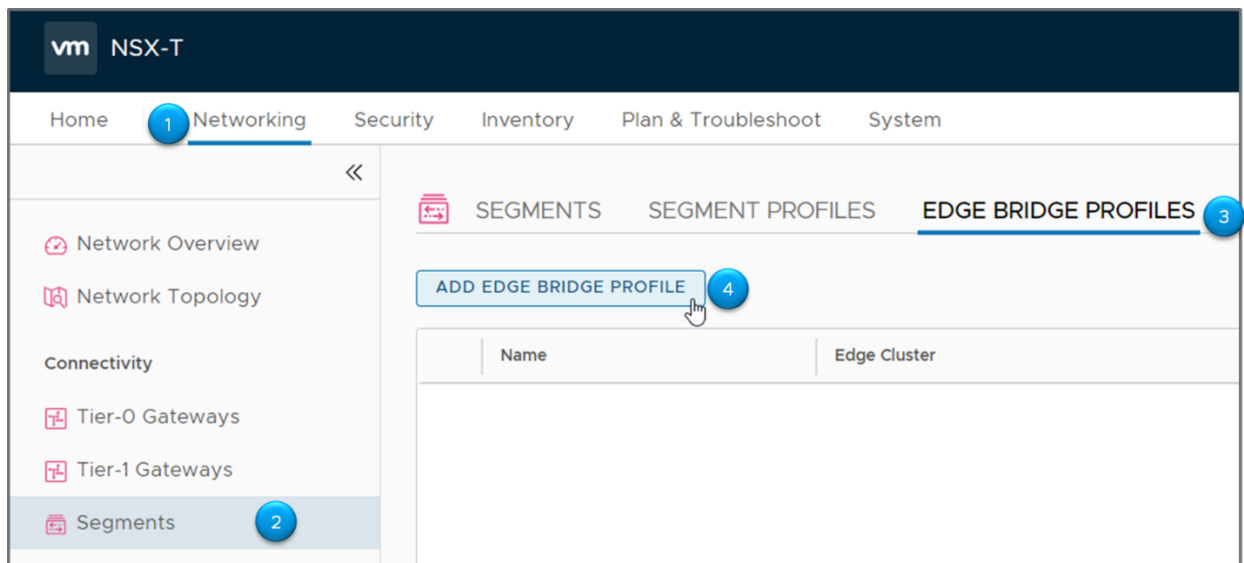


Figure 17: creating a Bridge Profile

The edge bridge profile definition is straightforward. The user just needs to identify the two edges involved within a specified edge cluster, as well as their role (primary/backup). The profile also defines the failover model, preemptive or non-preemptive.

- Name: BP-1, the name we give to this edge bridge profile
- Edge Cluster: EdgeCluster1, the edge cluster from which to pick the edges
- Primary Node: edge node that will be used as primary
- Backup Node: backup edge node
- Fail Over: we'll keep preemptive here, meaning that the primary node will take over as soon as it can.

The screenshot shows a configuration form for an edge bridge profile. At the top, there is a title 'ADD EDGE BRIDGE PROFILE' and a 'COLLAPSE ALL' button. Below the title is a search bar 'Filter by Name, Path and more'. The form is organized into several sections:

- Name:** BP-1
- Edge Cluster:** EdgeCluster1
- Primary Node:** dl20-edge1.ft.lab
- Backup Node:** dl20-edge2.ft.lab
- Fail Over:** Preemptive
- HA Mode:** Active Standby
- Description:** A text input field.
- Tags:** A section with 'Tag (I' and 'Scope' dropdowns, and a '+' button. Below it, it says 'Max 30 allowed. Click (+) to save.'

At the bottom left, there are 'SAVE' and 'CANCEL' buttons.

Figure 18: Bridge Profile options

Associating a Segment to a Bridge Profile

Once this edge bridge profile is created, it can be applied to one or more segments in order to extend them to VLAN.

Our goal is to extend segment S1 to which our VM1 is attached, to VLAN 1000 where the physical server is connected. We're going to use the edges specified in BP-1 to create the appropriate edge bridge. This configuration is simply achieved by editing segment S1.

Go to:

1. Networking
2. Segments
3. SEGMENTS and edit segment "S1".
4. In the S1 configuration box, click "Set" in the Edge Bridges section:

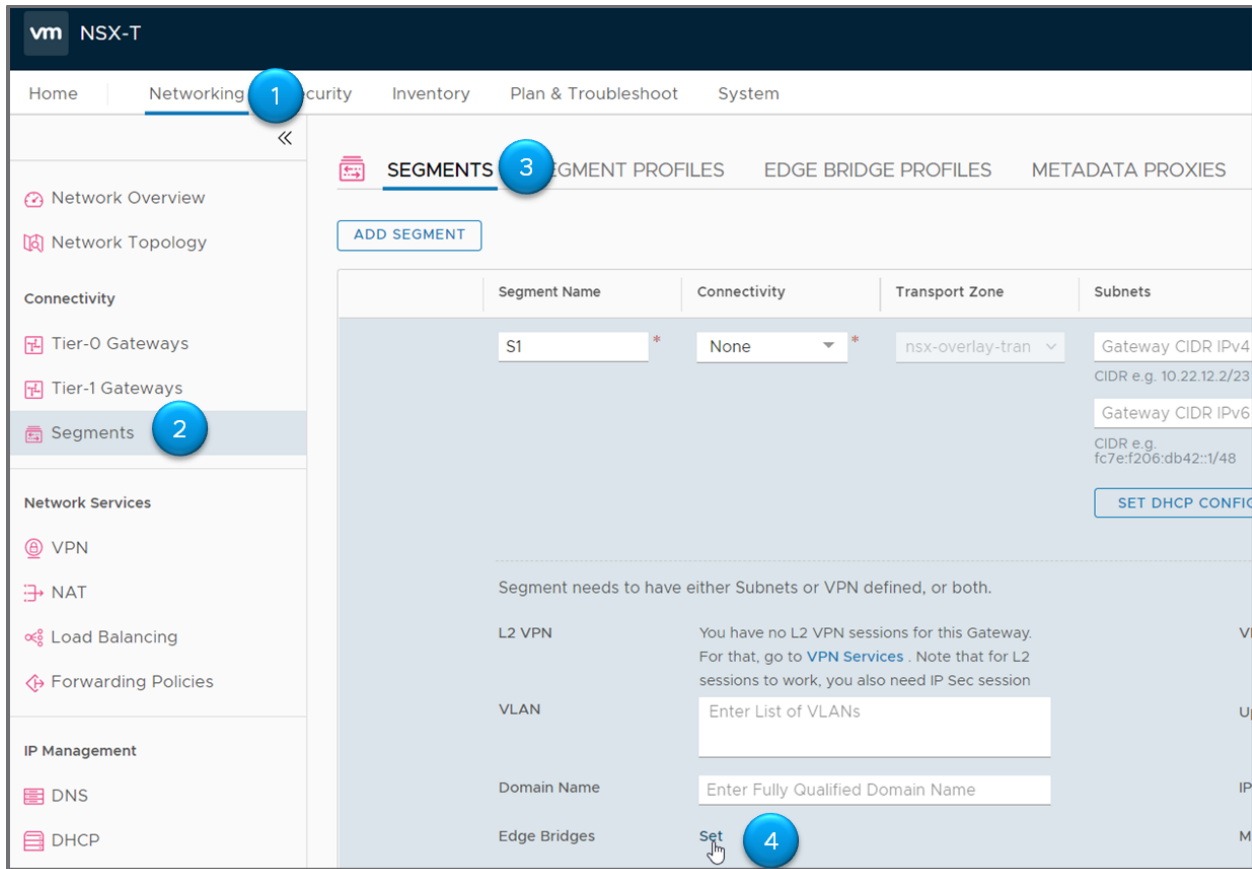


Figure 19: mapping a segment to a bridge profile

In the pop-up box “Bridge to S1”, click “ADD EDGE BRIDGE”:

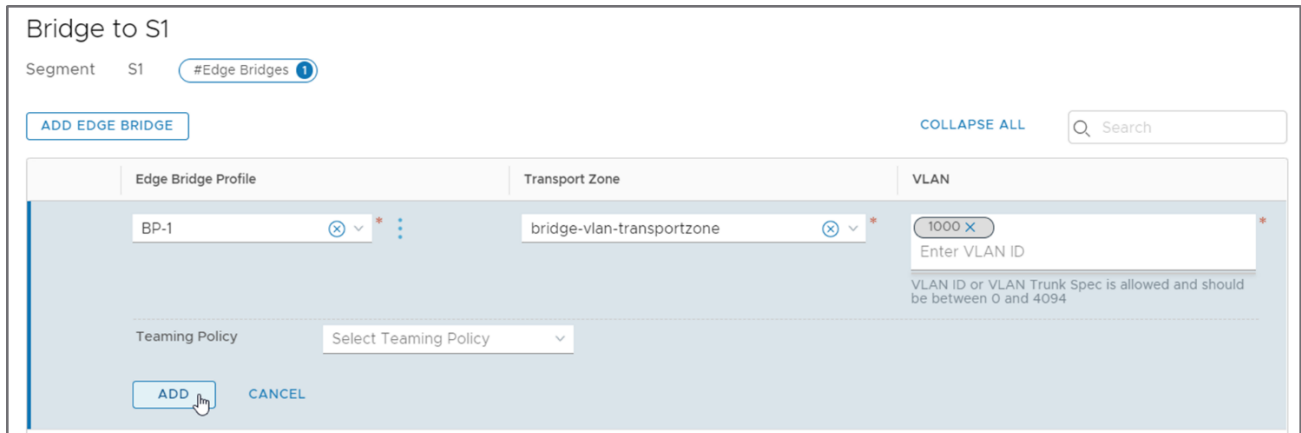


Figure 20: instantiate an Edge Bridge for a segment

- As expected, we’re going to use our previously defined “BP-1” bridge profile.

The overlay transport zone is implicitly defined: this is the one where our segment S1 is defined. Note also that the edges must be attached to this overlay transport zone.

- As mentioned earlier, we're going to use a dedicated VLAN transport zone called "bridge-vlan-transportzone" for our bridge. As a reminder, this VLAN transport zone is telling the bridge on which N-VDS of the edges the VLAN traffic needs to be forwarded. Be careful to select the transport zone to which the N-VDS of the edges in the Bridge Profile are attached as the UI is apparently not intelligent enough to filter the invalid VLAN transport zones.
- Then, we need to enter the VLAN ID we want to bridge our segment to, here VLAN 1000. If we were dealing with Guest VLAN tagged traffic in segment S1, that's where we could also enter a VLAN range.
- Finally, click ADD to complete our edge bridge definition. As you can see, the UI potentially allows you to add multiple edge bridges to segment S1, we'll only add one in this example.

A note on the "Select Teaming Policy" drop-down menu that we have not used:

In this example, our edges had an N-VDS with a single uplink attached to "bridge-vlan-transportzone". That's very convenient because, without getting into the details of how bridging works, the edge bridge only wants to send traffic on a single uplink. Now, suppose that this N-VDS had multiple uplinks. The administrator could then create a failover order "named teaming policy" in the "bridge-vlan-transportzone" that would identify in a deterministic way which uplink of the N-VDS should carry the VLAN traffic of the bridge. If the administrator does not select a named teaming policy here, the default teaming policy of the edge transport node would apply. Also, if the teaming policy applicable to bridged traffic is a "source port" teaming policy, then the first uplink defined in the teaming policy will be used.

Note: Check the "Limitations, caveats" part below for a current issue with the bridge not working with multiple VLAN uplinks.

High availability

The edge bridge is just a service like many others running on an edge and it leverages the edge High Availability (HA) capability available to all those services. The availability model has two layers:

- The first layer works at the level of the edge node. The goal here is to determine whether this edge node and its peers are operational.
- The second layer works per service. For a specific service on the edge node, when the peer edge node for this service is available, an additional state machine determines a more granular per-service state.

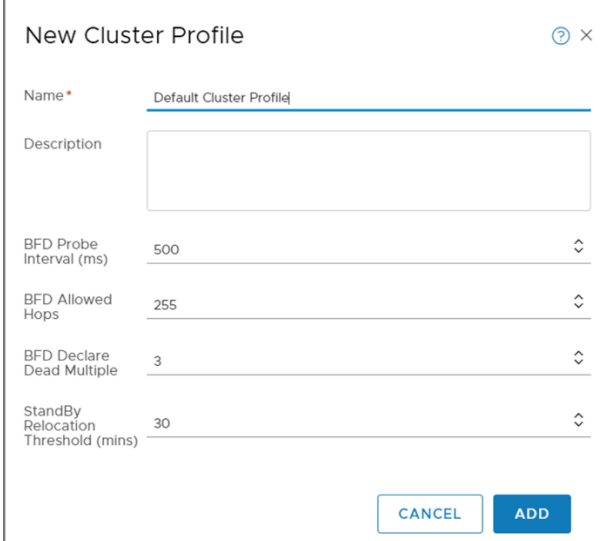
Edge node availability

An edge node is considered available if:

- It has a Tunnel End Point (TEP) interface up and operational. The TEP is the interface to the overlay.
- There is at least one tunnel in operational state. This means that this edge node is not entirely isolated and is peering with some other transport node: edge, or hypervisor.
- The edge node is not in maintenance mode.

There are other conditions based on the state of internal processes not listed here.

The determination of the state of the peer edge nodes depends on a protocol based on Bidirectional Forwarding Detection (BFD), transmitted over the management network and the overlay tunnels between edges. Basically, an edge establishes tunnels to its peers and retrieves their state via this BFD-based protocol. The different parameters for the BFD sessions can be set on a per edge cluster basis, according to the edge cluster profile. The following screenshot represents an edge cluster profile with default timers:



The screenshot shows a 'New Cluster Profile' dialog box with the following parameters:

Parameter	Value
Name *	Default Cluster Profile
Description	
BFD Probe Interval (ms)	500
BFD Allowed Hops	255
BFD Declare Dead Multiple	3
StandBy Relocation Threshold (mins)	30

Buttons: CANCEL, ADD

Figure 21: Edge Cluster Profile HA parameters

With these timers, the loss of an edge node can be detected by its peers in approximately three seconds. Bare metal edges can sustain higher probe rates than edge VMs and thus can converge faster.

Bridge-specific Edge HA

When both primary and backup edge nodes of an edge bridge are available, an additional per-application protocol allows determining which one will be actively forwarding bridged traffic. This per-application protocol is not BFD based: there can be hundreds of services peering across an edge cluster and sending periodic updates would represent an unacceptable amount of traffic and processor load. Instead, the per-application protocol only advertises changes to its peer. It is also run on the tunnels between edges. In the case of the edge bridge, this protocol is mainly used by the primary edge bridge instance to preempt the backup edge bridge instance and this can only be triggered when both edge node are available for a given edge bridge. The bridge specific edge HA can also trigger a switchover when the VLAN side port of the active edge bridge detects its link going down (something that can only be detected on the bare metal edge as the uplinks of an edge VM are internal to an ESXi host and don't go down.)

Bridge failover scenarios

The following diagram represents a very simple edge bridge scenario. A pair of edges host a primary and backup bridge. Those edges have one uplink connected to a physical switch for their VLAN traffic, and one uplink carrying their overlay traffic. Also represented is the overlay tunnel between the edges,

through which they exchange their status leveraging the BFD-based hello protocol mentioned in the previous section. At the bottom of the diagram, a physical server, attached to the physical switch, can communicate at Layer 2 with some VMs in the NSX domain.

Let us take this example to go over the different failure scenarios.

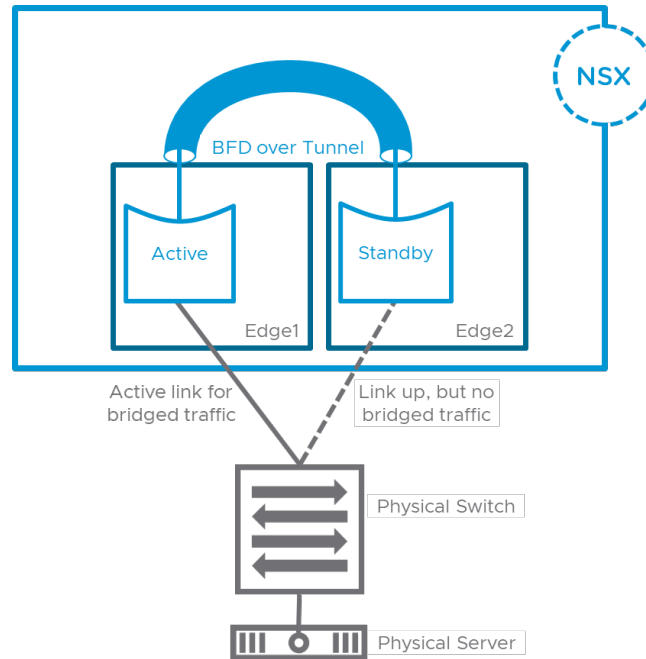


Figure 22: LAB topology for illustrating failure scenarios

Failure of Edge1

Edge2 does not hear from its peers via the BFD-based hello protocol. After timer expiration, Edge1 is declared dead and the bridge on Edge2 becomes active.

Failure of the VLAN uplink of Edge1

If the VLAN and overlay traffic are sharing the same physical uplink, the failure of this uplink will immediately bring down all the tunnels, the edge itself is then declared down and an edge switchover will take place immediately.

If the VLAN uplink goes down but overlay traffic is not impacted, the edge remains up and it is the per-edge bridge application protocol that will communicate the failure to its peer, triggering an immediate bridge failover.

Note that if Edge1 is in a VM form factor, the physical uplink on which VLAN traffic is transmitted can go down without bringing down the VLAN uplink vnic of the Edge1 VM. Some form of redundancy must be available within the host to cover for this failure scenario (check the part on the edge VM form factor below.)

Failure of the tunnel between Edge1 and Edge2

This failure scenario is unlikely considering that the tunnel and the management traffic between Edge1 and Edge2 run over a redundant IP infrastructure. Only multiple failures should lead to this state.

Anyway, in that scenario, both edges would consider their peer as defective and would end up active.

This dual-active scenario shouldn't open a bridging loop between virtual and physical. The BFD protocol running between Edge1 and Edge2 runs over a tunnel. If connectivity is broken for this BFD protocol, it is almost certainly also broken between Edge1 and Edge2 for all segments defined in the system. Indeed, the routing of the IP packets of the tunnel between the edges does not depend in any way on the VNI of the tunnel, so if one VNI is impacted (the one for the BFD protocol) all VNIs should be equally impacted.

However, there is one potential scenario where packet duplication could happen. Suppose that a hypervisor H still maintains connectivity to both Edges. This pathological case is represented below:

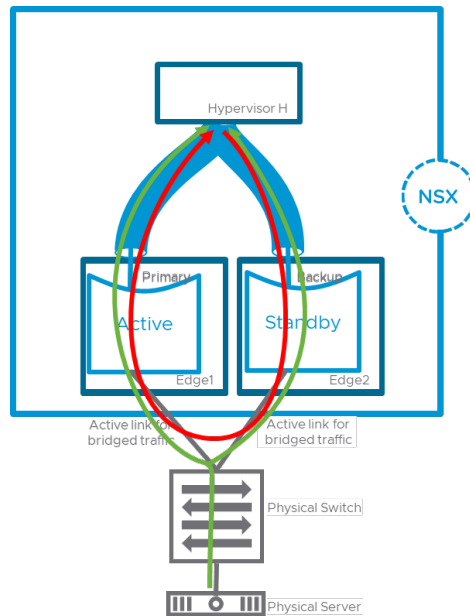


Figure 23: dual-active scenario (very unlikely)

Here, traffic initiated from the physical infrastructure could end up being duplicated for the VMs on H (green arrows). Also, traffic initiated from VM on this hypervisor H would be looped back to itself (red arrow).

Note that this case is very unlikely to happen if the physical infrastructure is leveraging a traditional routing protocol, where all possible forwarding path (both in the management and tunnel network) are considered between Edge1 and Edge2. This could however be possible if the physical infrastructure is using static routes or policy-based routing for example.

Failure of the path on the VLAN side

Finally, it is important to note that the edge bridge does not protect the path between Edge1 and Edge2 on the VLAN side.

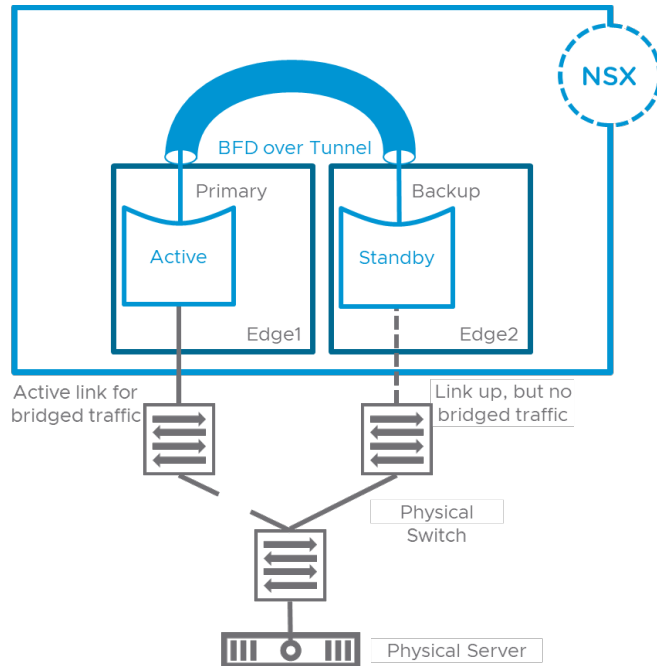


Figure 24: failure in the physical infrastructure

In the example represented above, the left uplink of the switch where the physical server is connected has failed. However, no failure has been reported by a link failure on Edge1 and the bridge on Edge1 remains active. As a result, the bridged traffic from the virtual world toward the physical server is still steered toward the left physical switch, where it is black holed. This failure scenario must be handled by some form of redundancy deployed at the level of the physical VLAN infrastructure (for example a redundant L2 connection between the two top of rack switches, or some uplink tracking functionality triggering a link down on the access port should a top of rack switch lose its uplink.)

Mac address tables updates

Two mac address tables are relevant when doing bridging between virtual and physical.

- The mac address tables within NSX that associate a mac address with a Tunnel End Point (TEP). Those are used when devices inside the NSX domain is trying to reach devices on the physical network (the VLAN side). For bridged traffic, those tables will point to the TEP of the edge hosting the active bridge.
- The mac address tables on the switches of the physical infrastructure. Those associate a mac address to a physical port, and in our case, this port is supposed to point in the direction of the edge where the active bridge reside.

Populating the mac address tables

The active bridge learns locally the source mac addresses of the traffic bridged into NSX. However, those mac addresses are not reported to the NSX Controllers. That means that within NSX, the mac address of a physical device reachable via an edge bridge has necessarily been learnt via data plane learning. This is the traditional Layer 2 flood and learn mechanism, also used by the switches in the physical infrastructure to learn the mac addresses originated from the NSX domain.

The aging time for the mac address learnt in NSX-T is a little bit above 10 minutes (10 minutes being the ARP cache timer), this timer is not configurable.

Updating the mac address tables after switchover

The switchover of the active bridge from one edge to another (because of a failure or a preempt) is an event that require updating both types of mac address tables mentioned earlier.

The following procedure is followed when a bridge becomes active:

- The active bridge syncs continuously the mac addresses it learns from the physical infrastructure to the standby bridge. When the standby takes over as active, it floods in the NSX domain RARP packets with a source mac address corresponding to each entry in the mac address table learnt from the physical infrastructure.
- The newly active bridge also injects RARP packets on the uplink toward the physical infrastructure with a source mac address corresponding to the mac address of each port reachable on the affected segment in the NSX domain (the list of those mac addresses is retrieved from the NSX-T controller). Those flooded packets will take care of updating the mac address tables in the physical infrastructure.

Design Considerations

Edge form factor

The edge bridge is available on both bare metal (BM) or Virtual Machine (VM) form factors. The use of the bridge in the bare metal form factor is relatively straightforward: the bridged traffic is sent on the uplinks of the N-VDS selected by VLAN transport zone specified on the bridge profile. There is no bridge-specific configuration necessary on the physical infrastructure where the bare metal edge attaches. This section is going to focus on a Bridge running on a VM form factor of the Edge.

Edge VM: Forged transmit and promiscuous or mac learning

For the VM form factor, it is important to remember that the edge bridge will end up sourcing traffic from several different mac addresses on its VLAN vNIC. This means, that this uplink vNIC must be connected to a DV port group allowing both:

- forged transmit and
- mac learning

Mac learning is available on the VDS as of vSphere 6.7. There is unfortunately no configuration for the feature available on the vCenter UI as of this writing, but it can be enabled via API or using powerCLI (check <https://www.virtuallyghetto.com/2018/04/native-mac-learning-in-vsphere-6-7-removes-the-need-for-promiscuous-mode-for-nested-esxi.html>.)

Mac learning is the preferred option, however, if your environment cannot meet the requirements for mac learning, it is possible to configure the edge port where the bridge sends VLAN traffic as a promiscuous port or a sink port. Check the following documentation page for the appropriate procedure: <https://docs.vmware.com/en/VMware-NSX-T-Data-Center/3.0/administration/GUID-F133B293-5DEA-4DC8-99DB-6EF004C8D8D7.html>

Note also that in that case, it is worth considering dedicating an edge uplink (vnic) to bridged traffic so that other kinds of traffic to/from the edge do not suffer from the performance impact related to promiscuous mode.

Edge VM: Virtual Guest Tagging

The edge bridge will be sending bridged traffic with an 802.1Q tag on its VLAN uplink. That means that this edge VM vNIC will have to be attached to a port group configured for Virtual Guest Tagging (VGT, i.e. the dvPortGroup shows as VLAN Trunk in the vCenter UI.)

Edge VM configuration example for the Bridge

The following Figure 25 represents an Edge VM dedicated to bridging and following the rules mentioned earlier in this document.

The Edge VM has four vNICs, but this design only uses 3:

- vNIC1 is dedicated to management traffic
- vNIC2 is the uplink of N-VDS1, the vSwitch that will be used for overlay traffic. The overlay dvPortGroup is using active/standby both pNICs of the host for redundancy.
- vNIC3 is the uplink of N-VDS2, the vSwitch that is attached to the VLAN transport zone where the bridged traffic will be sent. The “Bridge VLAN” dvPortGroup has the following configuration:
 - Virtual Guest Tagging is enabled so that it is possible to bridge to several segments to different VLAN IDs
 - Forged transmit and mac learning, so that the bridge can send traffic sourced from different mac addresses. If mac learning is not possible, promiscuous/sink port can be configured instead at the expense of degraded performance.
 - Active/standby teaming policy leveraging the same pNICs (but not necessarily in the same order) as the overlay dvPortGroup. That last point is important and will be detailed in the next part.

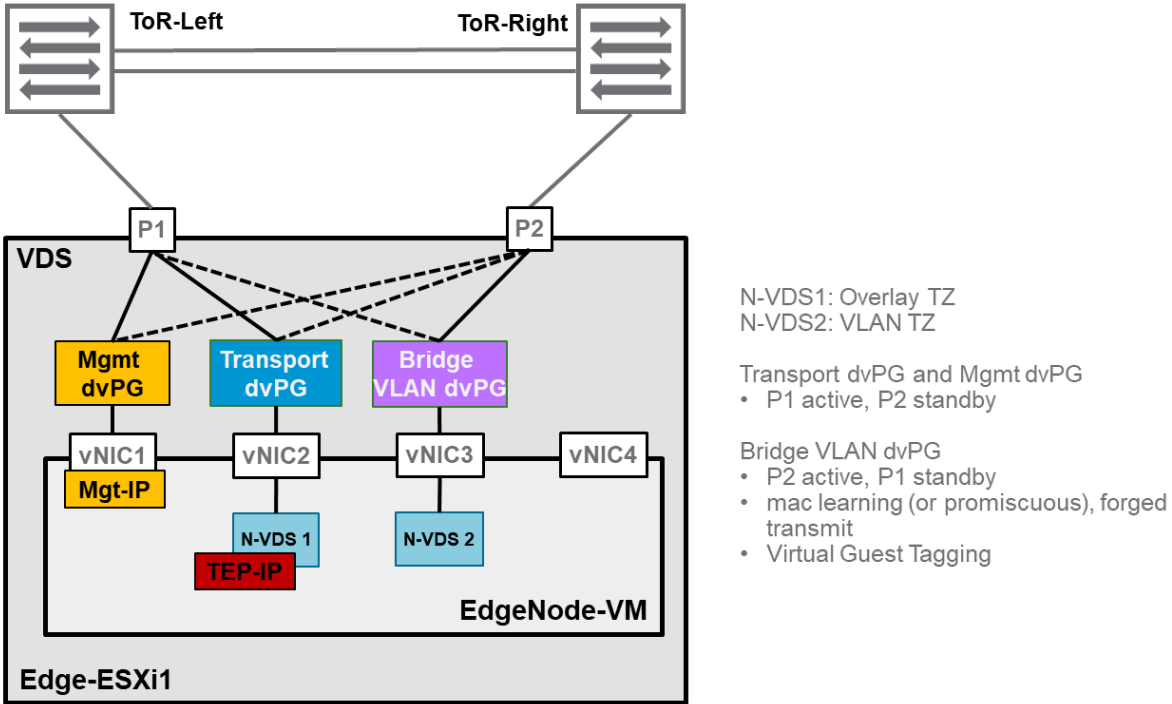


Figure 25: Edge VM design

Edge VM: Edge uplink protection

As we have seen, the edge bridge sends/receives two kinds of traffic on its uplinks: overlay traffic and VLAN traffic. This part discusses how to protect both against failure in the data path on the host.

The edge HA mechanism is exchanging BFD hellos over the tunnels between the different edges in the edge cluster. As a result, overlay traffic is protected against failure in the data path. In Figure 25 above, if both P1 and P2 went down on the host, all the tunnels between this edge VM and its peers would go down. As a result, this edge VM would be considered as failed by edge HA and another edge would take over the services it was running (including, but not limited to, the bridge service.)

Let us now focus on the VLAN uplink redundancy. As mentioned earlier, there is no protection mechanism for the VLAN uplinks on the bridge, and the edge VM does not detect links going down. However, the “Bridge VLAN” dvPortGroup has a teaming policy that protects against a single physical uplink of the host going down. Should both physical uplinks go down on the host, then Overlay traffic would be affected, the edge itself would be considered as down and the standby edge would takeover.

Again, because there is no protection for VLAN uplinks on the Edge Bridge, it is important that the VLAN traffic has access to the same redundant uplinks as the Overlay traffic. When all the uplinks used by VLAN traffic are down, it is imperative that overlay traffic is also affected so that there is edge switchover. Let us consider Figure 26, a slightly modified version of the or the recommended design, where this rule is not respected:

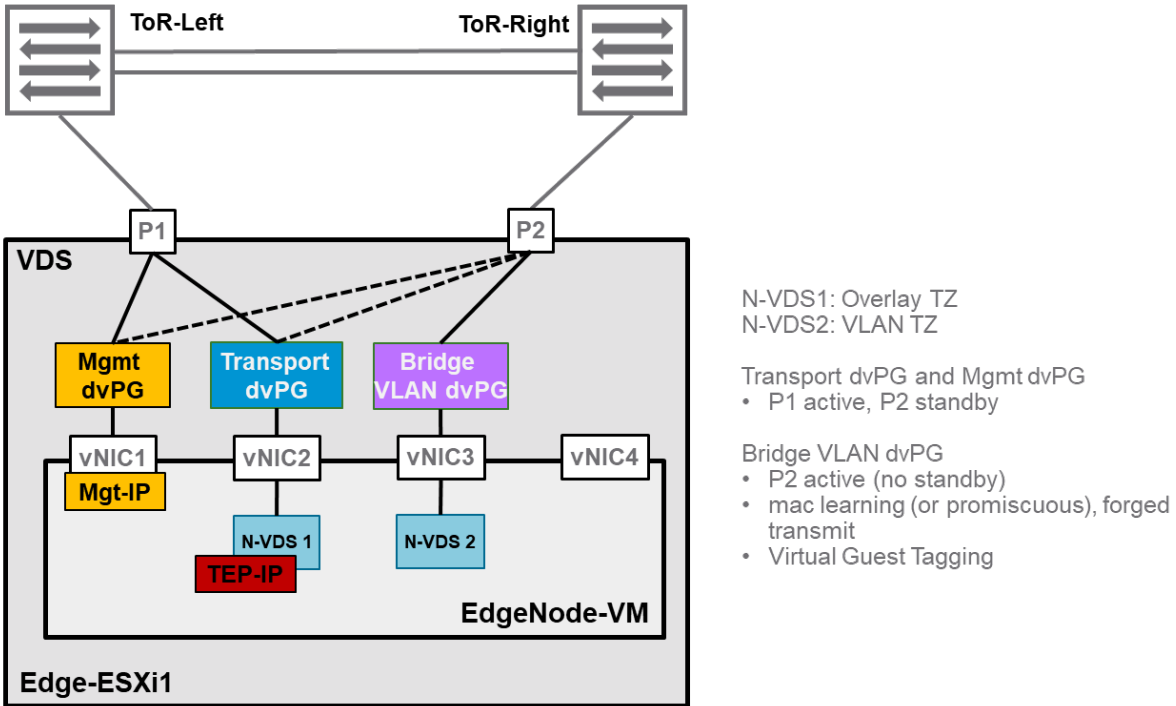


Figure 26: Non-redundant Edge VM design for the bridge (pathological example, do not use!)

Here, the “Bridge VLAN” dvPortGroup is only attached to port P2 on the host. Now suppose that P2 goes down. The bridge on the edge VM cannot send/receive VLAN traffic because there is no backup port covering for P2’s failure. Furthermore, because of the edge is a VM form factor, the edge uplink carrying the VLAN traffic of the bridge is not going down and the bridge HA mechanism cannot kick in and switchover to the standby bridge. Traffic is permanently black holed. The overlay traffic can still be sent over P1, so the edge HA is not triggering an edge switchover either.

Redundant VLAN connectivity

Remember that the edge bridge HA mechanism does not protect against connectivity problem in the VLAN infrastructure beyond the edge physical uplink.

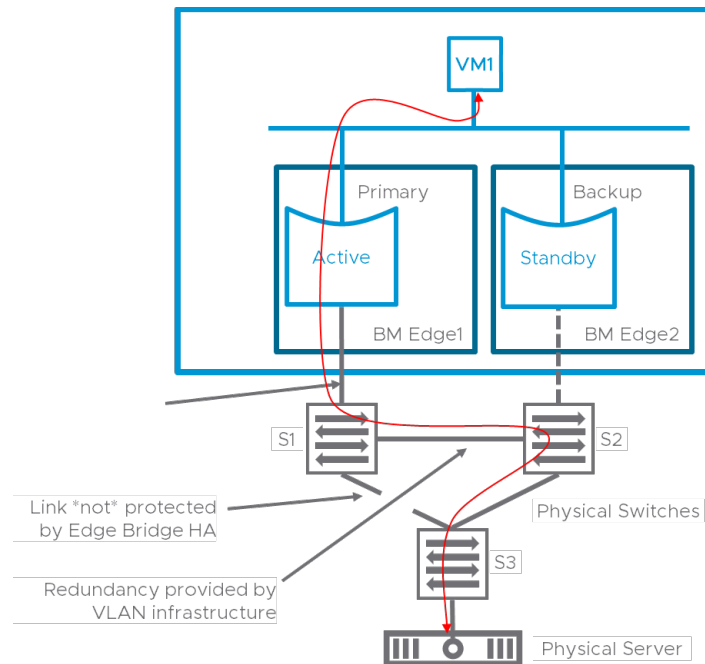


Figure 27: Physical bridging infrastructure must be redundant

In the above scenario, the failure of the uplink of bare metal Edge1 to physical switch S1 would trigger an edge bridge convergence where the bridge on Edge2 would become active. However, the failure of the path between physical switches S1 and S3 (as represented in the diagram) would have no impact on the edge bridge HA and would have to be recovered in the VLAN L2 domain itself. Here, we need to make sure that the alternate path S1-S2-S3 would become active thanks to some L2 control protocol in the bridged physical infrastructure for example.

Preemptive vs. non-preemptive

The choice is between precise bandwidth allocation on the uplinks and minimum disruption.

The preemptive model allows making sure that, when the system is fully operational, a bridge is using a specified uplink for its VLAN traffic. This is required for scaling out the solution, precisely distributing the load across several edge bridges and getting more aggregate bandwidth between virtual and physical by doing segment/VLAN load balancing.

The non-preemptive model maximizes availability. If the primary fails then recovers, it will not trigger a re-convergence that could lead to unnecessary packet loss by preempting the currently active backup. The drawback is that, after a recovered failure, the bridged traffic remains polarized on one edge, even if there were several bridge profiles defined on this pair of edges for segment/VLAN load balancing. As a result, with this design, one has to accept that bridged traffic will only be handled by a single bridge, even when multiple others are available. This is perfectly acceptable if availability is more important than available aggregate bandwidth.

NSX-T 2.5 introduces a CLI command (no API or UI yet) triggering a switchover to the primary when it is available, and the backup is currently forwarding traffic.

Performance

This document is purposefully avoiding the subject of the edge bridge performance. The rationale is that the bridge is a function running on the edge, and its capability is depending on the performance of the edge, a topic addressed by more specific papers. One can expect the edge bridge display the same performance as an edge gateway. There is however a very significant difference. By its nature, an NSX gateway is only dealing with traffic that can be routed, namely IPv4 and IPv6. The edge is optimized for this kind of traffic: it can identify and parse IP/UDP/TCP headers in order to distribute flows across multiple CPU cores on the edge platform. The edge bridge will benefit from those IP-centric optimization, but it is also responsible for forwarding any kind of ethernet traffic, irrespective of its upper layer protocol. Those non-IP packets will thus potentially experience lower throughput than IP-based packets through an edge bridge, as their handling might not be distributed optimally across CPU cores.

CLI, Operations

The command line interface for the bridge was reworked and simplified in NSX-T 3.1. Note that the old CLI commands are still available (so that scripts don't break) but hidden from the help.

You will be able to retrieve bridge information behind a single **"get bridge"** command on the edge:

```
edgenode-01a> get bridge
logical-switch          Logical switch
mac-sync-table         MAC Sync table
name                   Exact name
summary                Summary
vlan                   VLAN id <1-4094>
<CR>                  Execute command
|                      Output modifiers
```

When looking for bridging information on an edge, the first command to enter would be **"get bridge summary"**, which provides the bare minimum information on the bridge instances running on this edge.

In the following example, you can see that two bridges are instantiated on edgenode-01a. One is actively forwarding traffic, whereas the other is standby for a bridge running on a different edge. You also see the name of the overlay NSX segments and the VLAN ID to which they are bridged:

```
edgenode-01a> get bridge summary
Thu Jan 14 2021 UTC 21:13:11.064
UUID                               State      VLAN      Switch Name
88237255-73c9-431e-9f0e-690dfc9f5fcd Forwarding 172       web-seg
74946386-d928-4451-a724-83985ea0e5e3 Blocking  16        app-seg
```

From there, you can get the details of all the bridges on this edge by simply entering **"get bridge"** or you can drill down on a specific bridge from the list by identifying it via:

- the name of the segment being bridged,
ex: edgenode-01a> **get bridge name web-seg**
- the VLAN ID to which the segment is extended,
ex: edgenode-01a> **get bridge vlan 172**
- the UUID of the bridge itself

```
ex: edgenode-01a> get bridge 88237255-73c9-431e-9f0e-690dfc9f5fcd
```

The three above commands would produce the same output, detailing all the information related to the bridge extending segment “web-seg” to VLAN ID 172:

```
Thu Jan 14 2021 UTC 21:19:36.637
Bridge Port UUID      : 88237255-73c9-431e-9f0e-690dfc9f5fcd
Bridge Logical Switch UUID : 483ecb50-bcb2-59b7-95d3-e281a80c5fce
Attached Logical Switch UUID : 1e1e5739-683f-4f83-a3e2-ae8b884530b9
Attached Logical Switch Name : web-seg
VLAN ID              : 172
State                : Forwarding
Transport Zone       : 7879e179-652f-4e77-81a1-2c5dcbf6adf0
Device               : fp-eth0
Device State         : Up
Rank                 : 0
HA Failover Mode     : Preemptive
  State              : Active
  Event              : Remote State Updated
  Time               : 2021-01-14T18:53:23.205289
  Peer Node UUID     : 64968f68-62ac-4da3-8082-ac9b74da9e3c
  Peer HA State      : Standby
Last RARP            : 2021-01-14 18:53:24.266
RARP Count           : 5
```

Most of the fields are self-explanatory but we will detail them below. Note that many use UUID, which are not really user friendly, because the associated name is not easily retrieved.

Bridge Port UUID	UUID identifying the bridge
Bridge Logical Switch UUID	UUID identifying the VLAN logical switch
Attached Logical Switch UUID	UUID identifying the overlay segment being bridged
Attached Logical Switch Name	The name of the segment being bridged
VLAN ID	VLAN ID to which the overlay segment is bridged
State	State of the bridge, can be forwarding or blocking. When forwarding, the bridge is actively bridging traffic between overlay and VLAN. When blocking, no traffic is going through.
Transport Zone	the VLAN transport zone used to determine the VLAN uplink
Device	the VLAN uplink
Device State	State of the VLAN uplink (up/down)
Rank	When 0, this bridge is the primary bridge in the bridge profile, when 1, this bridge is configured as backup in the bridge profile.

HA Failover Mode	Preemptive or Non-Preemptive, as configured in the bridge profile associated to this bridge.
State	Active or Standby. This is the logical state of the bridge, that determines in its state machine whether it's going to be forwarding or blocking traffic.
Event	Event that triggered the last update in the HA state machine of the bridge.
Time	Timestamp for the above event.
Peer Node UUID	UUID of the edge node hosting the peer bridge. This line does not appear if the bridge profile does not specify a backup.
Peer HA State	State of the bridge on the peer edge. This line does not appear if the bridge profile does not specify a backup.
Last RARP	Last time the bridge sent RARP packets
RARP Count	Number of RARP packets sent.

As mentioned earlier in this document, the edge bridge is syncing some mac addresses it has learnt to its peer. This will be use for sending RARP packet to update the mac address tables during switchover. You can get some information about the mac addresses synced using the following `get bridge mac-sync-table` command.

```

edgenode-01a> get bridge mac-sync-table
Thu Jan 14 2021 UTC 21:23:33.452
MAC-SYNC Table
MAC                : 00:50:56:05:c0:7e
VNI                 : 67584
Bridge Port UUID   : 88237255-73c9-431e-9f0e-690dfc9f5fcd

```

The VNI referenced above is the the VNI associated to the overlay segment being bridged, the UUID identifies the bridge.

Finally, the CLI offers a single `set` command, to force switchover between redundant bridges. This command is only effective on the standby bridge created from a non-preemptive bridge profile. The following example shows the use of this command to put the bridge for VLAN 16 into forwarding state on edgenode-01a. You'll notice that the HA event is "Force Failover" in the bridge HA state machine:

```

edgenode-01a> get bridge vlan 16
Thu Jan 14 2021 UTC 21:18:23.012
Bridge Port UUID       : 74946386-d928-4451-a724-83985ea0e5e3
Bridge Logical Switch UUID : 9b3e53a4-0e86-5ef7-b062-f67cec76d817
Attached Logical Switch UUID : 26abe94d-a5af-48a1-9247-bc957a48f8e2
Attached Logical Switch Name : app-seg
VLAN ID                : 16
State                  : Blocked
Transport Zone         : 7879e179-652f-4e77-81a1-2c5dcbf6adf0
Device                 : fp-eth0
Device State           : Up

```

```

Rank : 1
HA Failover Mode : Non-Preemptive
  State : Standby
  Event : Force Failover
  Time : 2021-01-14T21:09:42.151733
  Peer Node UUID : 64968f68-62ac-4da3-8082-ac9b74da9e3c
  Peer HA State : Active
Last RARP : 2021-01-14 19:43:59.493
RARP Count : 3
edgenode-01a> set bridge 74946386-d928-4451-a724-83985ea0e5e3 state active
Successfully brought L2Bridge 74946386-d928-4451-a724-83985ea0e5e3 to active
state
edgenode-01a> get bridge 74946386-d928-4451-a724-83985ea0e5e3
Thu Jan 14 2021 UTC 21:20:50.932
Bridge Port UUID : 74946386-d928-4451-a724-83985ea0e5e3
Bridge Logical Switch UUID : 9b3e53a4-0e86-5ef7-b062-f67cec76d817
Attached Logical Switch UUID : 26abe94d-a5af-48a1-9247-bc957a48f8e2
Attached Logical Switch Name : app-seg
VLAN ID : 16
State : Forwarding
Transport Zone : 7879e179-652f-4e77-81a1-2c5dcbf6adf0
Device : fp-eth0
Device State : Up
Rank : 1
HA Failover Mode : Non-Preemptive
  State : Active
  Event : Force Failover
  Time : 2021-01-14T21:20:21.696614
  Peer Node UUID : 64968f68-62ac-4da3-8082-ac9b74da9e3c
  Peer HA State : Standby
Last RARP : 2021-01-14 21:20:22.717
RARP Count : 3

```

Limitations, caveats

NSX-T 3.1 limitations

- A segment can be attached to several bridge profiles. However, the UI does not allow configuring a bridge firewall for all those attachments.
- The data path does not allow bridging the same segment twice on the same Edge. The UI/API will not prevent the configuration, but you would experience BUM (broadcast, unknown unicast, multicast) traffic loss if you were attempting this configuration.
- A bridge should not be configured with multiple VLAN uplinks. This will be fixed as a bug in a future release. Check https://kb.vmware.com/s/article/81777?lang=en_US

VLAN conflict on the Edge

An edge uplink cannot have two VLAN segments with the same VLAN ID. This limitation is not directly related to the edge bridge, but its consequences will be apparent in some edge bridge related configuration.

When a user configures the attachment of segment S1 to VLAN X on a specific edge uplink, the edge automatically creates a local segment VLAN X on the N-VDS owning the uplink. This automatic creation will fail if there was already a VLAN segment configured with the same ID on this N-VDS. To be more precise, the configuration of the VLAN segment will succeed, but the realization of the segment will fail.

It is possible to hit this limitation when:

- Trying to configure attachments of several different segments to the same VLAN ID on the same uplink (only the first attachment will be operational.)
- Trying to configure the attachment of a segment to a VLAN ID that is already used by other services on this uplink. For example, if there is a Tier 0 router using this VLAN ID to send traffic on the uplink.

Summary, TL;DR.

This section is showing the different topics introduced in this paper as a list of bullet points, for quick reference. Note that for efficiency, the order of the bullet points does not necessarily match the structure of the document.

- The edge bridge allows extending a segment to VLAN.
- The edge bridge functionality is mainly for migration scenarios ($P \rightarrow V$ or $V \rightarrow V$) or for integrating non-virtualizable bare metal appliances that need Layer 2 adjacency with virtual machines (the typically case is a database server.)
- The edge bridge supports Guest VLAN Tagging on the overlay segment it is extending to VLAN.
 - If the bridge only includes a VLAN ID in its definition, a frame carried in an overlay segment already carrying an 802.1Q tag will be bridged to VLAN with an additional 802.1Q tag (QinQ). Basically, the inner tag is ignored by the bridging function.
 - If the bridge is defined with a VLAN range, then a frame carrying a VLAN tag will only be bridged if its VLAN ID is within the range defined for the bridge. No additional 802.1Q tag will be added when bridging this frame to the VLAN uplink.
- The edge bridge leverages DPDK for high performance forwarding.
- The traffic bridged in/out of the NSX-T domain is subject to an edge bridge firewall instance.
- The edge bridge integrates seamlessly with routing (centralized and distributed). A Distributed Router (running on the edge where a bridge is active) can be default gateway for a physical device.
- In the first NSX-T releases, a segment could only be attached to a single active edge bridge. This limitation was relaxed in NSX-T 2.5, thus allowing bridging between a segment and VLAN at different locations in the network (and also introducing the capability of misconfiguring bridging loops.)
- Two different segments can be bridged to the same VLAN ID, if their VLAN traffic is sent on different edge uplinks (same VLAN ID does not mean same VLAN.)
- The VLAN ID used for bridging on a specific uplink cannot collide with another VLAN segment used for other feature (for example, the VLAN ID of a Tier 0 uplink or another bridged segment)
- The edge bridge can be configured on both bare metal and VM form factor of the edge. Bear in mind that for a VM form factor:
 - The edge bridge will (most likely) send tagged VLAN traffic on its uplink: the port group to which this uplink is connected must be configured to accept this tag (Virtual Guest Tagging.)
 - The edge bridge will send VLAN traffic on behalf of several VMs on the virtual side, that means that the uplink of an edge doing bridging has to be connected to a port group

- allowing forged transmit and mac learning (promiscuous/sink port are less desirable alternatives to mac learning.)
- The VLAN uplink is not protected by a hello mechanism, it is thus better to use the same physical uplinks for VLAN and overlay in that case (to benefit from the overlay protection.)
 - Edge bridge redundancy
 - An edge bridge is defined based on a bridge profile. The bridge profile defines a primary edge, where bridging will be performed in priority and optionally a backup edge that will be used if the primary is not available. The bridge profile can only include two edges (not more.)
 - The edge bridge redundancy model is based on edge redundancy.
 - An edge can belong to different bridge profiles. This allows for VLAN/segment load balancing.
 - The edge bridge high availability protects against failure scenarios in the NSX domain and on the uplinks of the edge. It does not protect against failure in the physical VLAN infrastructure beyond the uplink.
 - Upon switchover between edge bridges, the newly active bridge will send flooded traffic (RARP) to update the mac address tables quickly.
 - The edge high availability model supports both preemptive or non-preemptive modes.
 - Preemptive ensures that bridged traffic flows through a specific edge when both edges are active.
 - Non-preempting minimize disruption by not triggering a switchover when the primary edge bridge comes back up after a failure and the backup bridge is active.
 - NSX-T 2.5 introduced a basic CLI command allowing a bridge to become active.
 - Caveats
 - An edge bridge should not be configured with multiple VLAN uplinks:
https://kb.vmware.com/s/article/81777?lang=en_US
 - VLAN conflicts on the Edge: two services cannot use the same VLAN ID on an edge uplink. That means that, for example, it is not possible to bridge traffic to a VLAN ID that is already used by a Tier 0 interface on an edge.
 - It is not possible to bridge the same segment twice on the same edge.