April 2023

OPTIMIZING ROUTE LEAK FROM EVPN TO IP-VPN FOR SINGLE ACTIVE USE CASES

Mankamana Mishra

Jiri Chaloupka

Sameer Gulrajani

Nitin Kumar

OPTIMIZING ROUTE LEAK FROM EVPN TO IP-VPN FOR SINGLE ACTIVE USE CASES

AUTHORS:
Mankamana Mishra
Jiri Chaloupka
Sameer Gulrajani
Nitin Kumar

ABSTRACT

Optimizing the process by which host routes are leaked to an Internet Protocol (IP) virtual private network (VPN) unicast is of singular importance in a large network environment. To address this type of challenge, techniques are presented herein that support a dynamic means for limiting the host advertisements in a network where hosts are announced only on an as-needed basis. In effect, aspects of the presented techniques optimize what is leaked and optimize the total number of routes in a spine.

DETAILED DESCRIPTION

Optimizing the process by which host routes are leaked to an Internet Protocol (IP) virtual private network (VPN) unicast is of singular importance in a large network environment.

Techniques that address the above-described challenge are presented herein and will be described in detail in the narrative that follows. That discussion will make reference to elements of an exemplary arrangement (that depicts a desired network flow for a customer charter) that is shown in Figure 1, below.

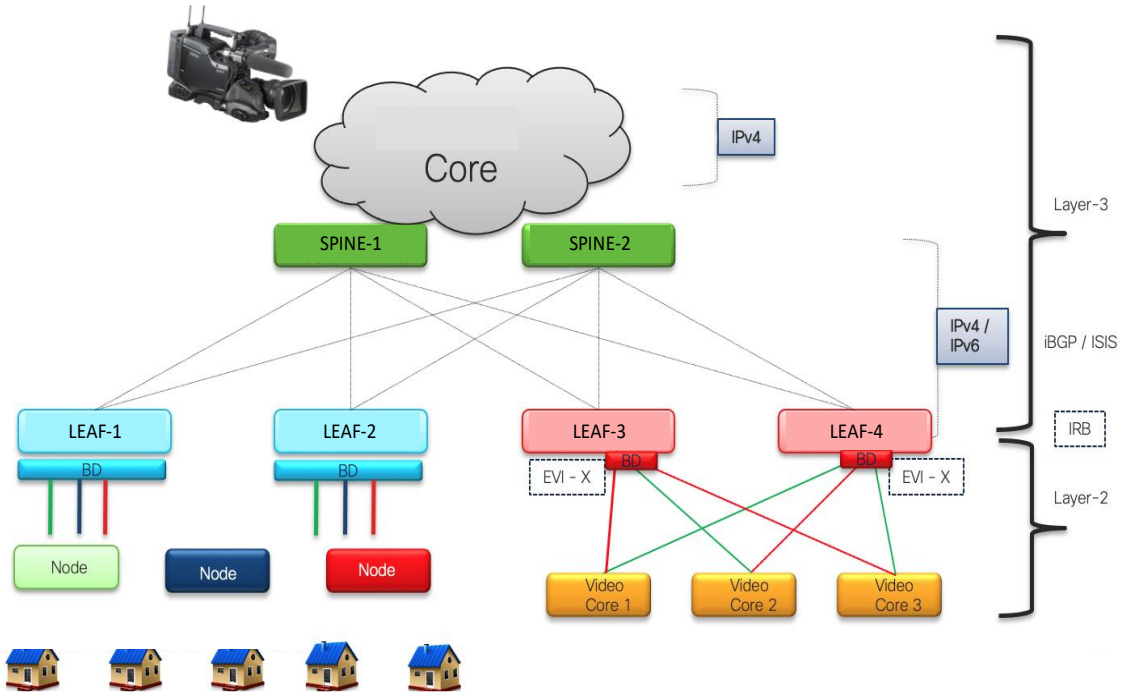

Figure 1: Exemplary Arrangement

As shown in Figure 1, above, there are two network layers. A charter core is an existing Internet Protocol version 4 (IPv4) network where the actual multicast sources are connected. Those sources provide a raw video feed.

As further shown in Figure 1, above, the raw video feeds are delivered to a video core, which is essentially a video processing center. The video core is a Layer 2 network consisting of some number of video processing virtual machines (VMs), data centers, and switches. Those third-party switches control the link state.

In Figure 1, above, the depicted spine and leaf are a new network, which may be used to deliver the video feed to different residential units after any video processing is completed.

In the above exemplary arrangement, it is important to note that a switch is controlling a port state because the customer wishes to run various analytics so they want to make sure that only one link is up at any given point in time.

Figure 2, below, makes use of the exemplary arrangement that was presented in Figure 1, above, to depict elements of a traffic flow.

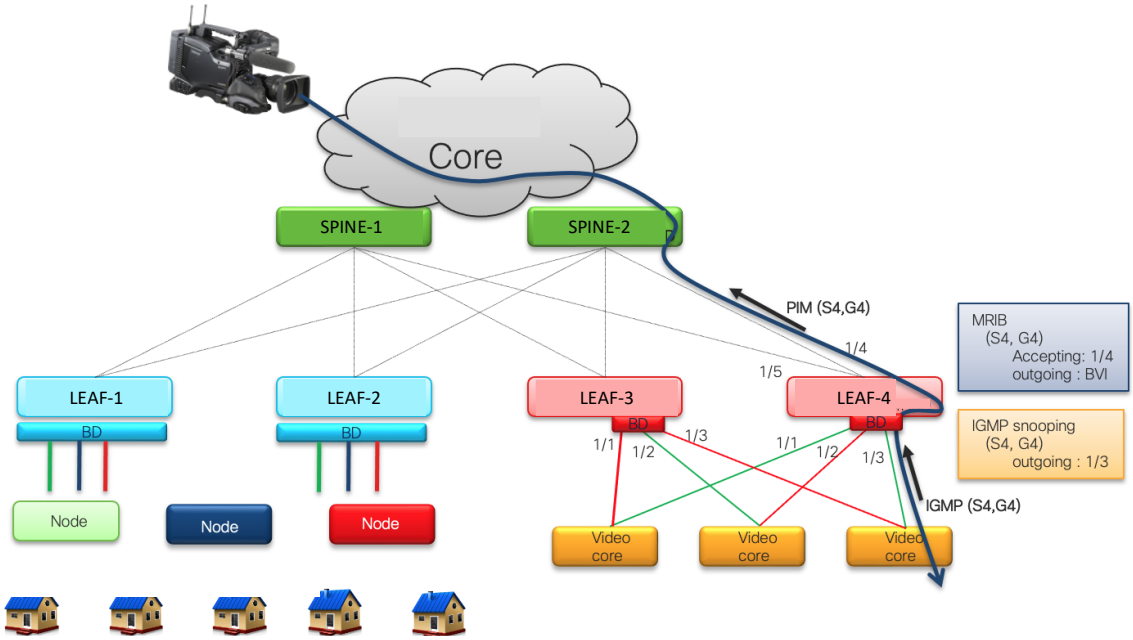

Figure 2: Illustrative Traffic Flow

As depicted in Figure 2, above, a video processing center may send a Protocol Independent Multicast (PIM) join message and then begin receiving a raw video feed.

Figure 3, below, again makes use of the exemplary arrangement that was presented in Figure 1, above, to depict elements of various processing activities that will be described below.

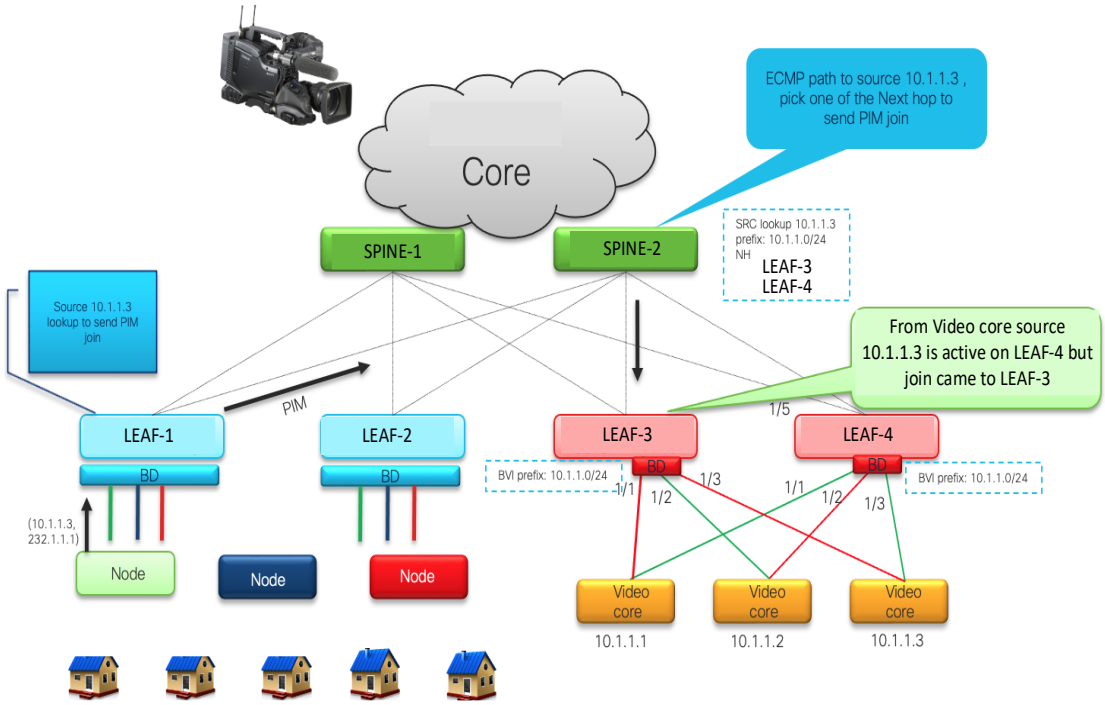

Figure 3: Illustrative Processing Activities

As depicted in Figure 3, above, after the video traffic has been processed it needs to be delivered to the different residential units. In support of that activity, a PIM-based routing protocol may be used to construct a multicast tree.

Consider the following sequence of events, which are highlighted in Figure 3, above, through the black arrows.

First, a left-most node wants the traffic which is being sourced from (10.1.1.3, G). Second, a PIM join message reaches one of the spines, for example Spine-2 in this case. Third, the spine looks at the source prefix and it identifies that there is an equal-cost multi-path (ECMP) routing path. Fourth, nothing prevents the PIM join message from landing on Leaf-3 and not getting the traffic, since the video core with address 10.1.1.3 is only active on Leaf-4 (since in the figure the link is red for Leaf-1).

An existing solution is available whereby host routes are leaked to an IP VPN unicast. As a result, a spine would know where, exactly, a host is present and it can send a join message to the appropriate ingress leaf. In reality, there may be a huge number of hosts in total (potentially on the order of hundreds of thousands) but the number of actual

multicast sources would be relatively low. As a result, a blanket route leak would lead to too many unnecessary routes being leaked to a spine.

Optimizing the above-described process is quite important. In general, the total number of hosts who are producing multicast traffic is much lower than the number of hosts in a site. A spine may also participate in many other parts of a network. Consequently, flooding the tables of a spine with unused host addresses will be too costly during any period of churn. For example, one large network in India had approximately two million routes in its Border Gateway Protocol (BGP) tables and during churn it was impacting some of the ongoing routing as well since BGP needed more time to process the whole route.

Techniques are presented herein that support a procedure for developing a dynamic route policy that is multicast-driven to ensure that only those routes are leaked to an IP VPN unicast which really belong to the multicast source. In effect, aspects of the presented techniques optimize what is leaked and optimize the total number of routes in a spine.

Aspects of the presented techniques support the creation of a dynamic policy for BGP which identifies the specific routes that need to be leaked to an IP VPN. By default, none of the hosts are leaked and a routing protocol will advertise only summarized routes. Once an actual source becomes active, a multicast paradigm may be employed to notify the routing protocols of which host needs to be advertised. Then the routing protocol can dynamically advertise the host address.

Although the above-described optimization may appear on the surface to be very simple, the impact of such an approach is quite significant for a network operator. Among other things, such an approach prevents so many issues which can occur in a network possibly due to scale.

In the case where there are P intermediary routers, if BGP is being used to advertise a route there are chances that it is not a point-to-point connection between both BGP peers. In such a case, when constructing an underlay tree, a middle node would not know how to reach the host. Consequently, when creating an underlay tree, a join reverse-path forwarding (RPF) vector or a recursive forwarding equivalence class (FEC) may be employed.

Under aspects of the techniques presented herein, if there is a leaf failure the video core (as depicted in the above example) will bring up another interface. This would lead to a situation where the new leaf is going to advertise a host route. This will lead to an RPF change and a PIM and multipoint Label Distribution Protocol (mLDP) tree will converge to the new location. Additionally, an access link failure will lead to a new leaf learning about a host and advertising a route.

Under further aspects of the techniques presented herein, an implementation may determine how quickly a host withdrawal must happen after a source is determined to be not active.

In summary, techniques have been presented herein that support a dynamic means for limiting the host advertisements in a network where hosts are announced only on an as-needed basis. In effect, aspects of the presented techniques optimize what is leaked and optimize the total number of routes in a spine.