

## Description

The sFlow VXLAN extension adds support for providing VXLAN-related information to sFlow packet samples, for VXLAN forwarded traffic. Specifically, for customer traffic ingressing on a CE-facing PE interface and forwarded into a VXLAN tunnel, the IP address of the source VTEP, the IP address of the destination VTEP and the VNI will be included in the sFlow datagram.

For IPv4 tunnels, the IP addresses of the source VTEP and the destination VTEP will be added to the sFlow datagram by using the `extended_ipv4_tunnel_egress` structure (shown below) as defined in the [sFlow Tunnel Structures](#) and populating in the `src_ip` and `dst_ip` fields respectively in the `sampled_ipv4` structure. For IPv6 tunnels, this information will be added with the `extended_ipv6_tunnel_egress` structure.

The VNI will be added to the sFlow datagram by using the `extended_vni_egress` structure as defined in the [sFlow Tunnel Structures](#).

```
struct sampled_ipv4 {
    unsigned int length;        /* The length of the IP packet excluding
                                lower layer encapsulations */
    unsigned int protocol;     /* IP Protocol type
                                (for example, TCP = 6, UDP = 17) */
    ip_v4 src_ip;              /* Source IP Address */
    ip_v4 dst_ip;              /* Destination IP Address */
    unsigned int src_port;     /* TCP/UDP source port number or equivalent */
    unsigned int dst_port;     /* TCP/UDP destination port number or equivalent */
    unsigned int tcp_flags;    /* TCP flags */
    unsigned int tos;          /* IP type of service */
}

/* opaque = flow_data; enterprise = 0; format = 1023 */
struct extended_ipv4_tunnel_egress {
    sampled_ipv4 header;
}

struct sampled_ipv6 {
    unsigned int length;        /* The length of the IP packet excluding
                                lower layer encapsulations */
    unsigned int protocol;     /* IP next header
                                (for example, TCP = 6, UDP = 17) */
    ip_v6 src_ip;              /* Source IP Address */
    ip_v6 dst_ip;              /* Destination IP Address */
    unsigned int src_port;     /* TCP/UDP source port number or equivalent */
    unsigned int dst_port;     /* TCP/UDP destination port number or equivalent */
}
```

```
valent */
    unsigned int tcp_flags; /* TCP flags */
    unsigned int priority; /* IP priority */
}

/* opaque = flow_data; enterprise = 0; format = 1025 */
struct extended_ipv6_tunnel_egress {
    sampled_ipv6 header;
}

/* opaque_flow_data; enterprise = 0; format = 1029 */
struct extended_vni_egress {
    unsigned int vni;
}
```

The additional data can be used by service providers to identify which customer traffic is flowing in their VXLAN deployment.

## Platform compatibility

This feature is available on all platforms that support both sFlow and VXLAN. Refer to the [EVPN IRB with VXLAN TOI](#) for the specific platform names.

## Configuration

The VXLAN extension is added to compatible samples by configuring sFlow on a device and then enabling the desired extension.

```
Arista(config)#sflow run
Arista(config)#sflow extension vxlan
```

The VXLAN extension is disabled by default.

## Show commands

The show sflow detail command may be used to examine the status of sFlow on the device,

including the VXLAN extension status. Here is an example of the output when the VXLAN extension is enabled:

```
Arista(config)#show sflow detail
...
Status
-----
...
Sample output interface: yes
Sample MPLS extension: no
Sample MPLS VC extension for VPLS traffic: yes
Sample switch extension: yes
Sample router extension: yes
Sample tunnel IPv4 egress extension: no
Sample VXLAN extension: yes
...
```

## Troubleshooting

For troubleshooting, it can be helpful to look at the show command from the previous section and enable the traces outlined in the tracing section, and refer to the sFlow agent logs.

## Tracing

Disclaimer: In some cases, enabling tracing can seriously impact the performance of the switch. Please use it cautiously and seek advice from an Arista representative before enabling this in any production environments.

The following traces can be enabled for debugging:

```
Arista(config)#trace Sflow setting SflowSm/0-8,BridgingSim*,RoutingSim*
```

## Limitations

- The VXLAN extension will only be added for traffic ingressing CE-facing PE interface (i.e. for traffic that has not yet been VXLAN-encapsulated).
- The VXLAN extension does not support broadcast, unknown unicast and multicast traffic.
- If multiple VTI is configured, only the Vxlan1 VTI will be used for VXLAN information

lookup.

- If multi-VTEP MLAG is configured, we will only report the local VTEP address of the MLAG device as the source VTEP.
- The VXLAN extension will not correctly provide VXLAN information if all-active multihoming is configured.
- The VXLAN extension does not support L2 subinterfaces.
- The VXLAN extension will not correctly provide VXLAN information if VLAN translation is configured on R3.