IDR Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 03, 2025

P. Huo
G. Chen
ByteDance
C. Lin
New H3C Technologies
W. Cheng
China Mobile
Syed Hasan Raza Naqvi
Broadcom
Yossi Kikozashvili
DriveNets
C. Q
ByteDance

March 03, 2025

Bgp Extension for Tunnel Egress Point
draft-hcl-idr-extend-tunnel-egress-point-04

Abstract

In AI networks, flow characteristics often exhibit a low number of flows but with high bandwidth per flow, making it easy to cause network congestion when using traditional flow-level load balancing methods. Currently, the direction of traffic scheduling focuses on load sharing individual packets of the same flow, which requires sorting based on the Tunnel Egress Point information from the remote end. This document describes the method of publishing Tunnel Egress Point through the BGP protocol.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 03, 2025.

Table of Contents

1. Introduction

With the widespread application of AI technology, the AI Computing Center has experienced rapid development and increased attention to potential issues within AI networks.

The characteristics of AI traffic exhibit a low number of flows with substantial bandwidth per flow, making traditional flow-level load balancing highly susceptible to multiple flows hashing to the same link, resulting in congestion on certain links while others remain idle. This leads to low network utilization and an inability to handle sudden surges in network traffic. Consequently, the need for a new load balancing scheduling model is imperative.

Presently, the direction of scheduling in AI networks involves sharing the load of multiple packets within each flow individually, enabling the "spraying" of individual flows across the entire path to enhance effective bandwidth utilization and better application of existing bandwidth.

However, sharing the load of individual packets within a flow can result in packet reordering for the same traffic. Therefore, it is necessary for the egress point to carry the egress features of the traffic to the ingress point, enabling packet sorting based on the egress features of the traffic to ensure the proper sequencing of multiple packets within the same flow.

This document describes the method of conveying the egress characteristics of routes as route attributes through the BGP protocol to inform the ingress server.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT","SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Motivation

As shown in the figure 1, Leaf devices are connected downwards to host devices and upwards to Spine devices.

When hosts communicate with each other, there are multiple different ECMP paths available for OSF-Egress to forward packets. For example, traffic from H1 to H8 can go through the path H1 -> Leaf1 -> Spine1 -> Leaf4 -> H8, or it can go through H1 -> Leaf1 -> Spine2 -> Leaf4

-> H8. In traditional load balancing, after hashing the traffic, the same path is chosen for forwarding for the same flow.

In AI networks, where there is less data per flow but each flow carries a larger payload, traditional load balancing strategies can lead to network congestion. To adapt to the characteristics of AI networks, when load balancing with ECMP, multiple small data packets can be combined into a larger packet for transmission, and large packets can be divided into relatively smaller packets for transmission. The combined data packets are then evenly distributed over ECMP paths to fully utilize the bandwidth of each path. However, this may result in packet reordering, so it is necessary to reorder the packets at the packet’s destination.

During sorting, all packets destined for the same end-point need to be sorted. For example, for two data packets from H1 to H8, they are sorted based on the destination (Leaf4 + H8) to ensure that the packets arrive at H8 in the correct order.

Therefore, it is necessary to synchronize end-point information from OSF-Egress to OSF-Ingress through the control plane. When sending packets, OSF-Ingress numbers the packets based on the end-point and selects different paths for "spraying" the packets.

The intermediate OSF-Forward device forwards packets towards the final destination device based on the end-point without concerning about packet order. Finally, OFP-Egress reorders packets based on the same end-point number and forwards them to the hosts.

This document primarily describes how the control plane delivers end-point information.

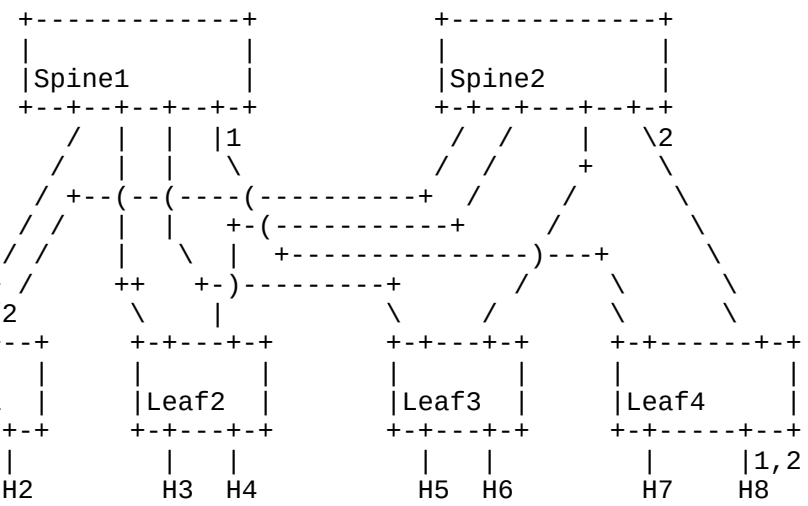

Figure 1: AI network

3. Terminology

The following terminologies are used in this document.

TEP: Tunnel Egress Point. This document

OSF: Open Scheduled Fabric. [draft-hcl-rtgwg-osf-framework-00]

OSF-Ingress: OSF Ingress router. [draft-hcl-rtgwg-osf-framework-00]

OSF-Egress: OSF Egress router. [draft-hcl-rtgwg-osf-framework-00]

OSF-Forwarder: OSF Forwarder Router. [draft-hcl-rtgwg-osf-framework-00]

4. Solution

As shown in Figure 2, in the Spin/Leaf network, each Leaf device, when advertising route prefixes externally, includes the Tunnel Egress Point information corresponding to these route prefixes.

When the entry Leaf device receives this route, it extracts the Tunnel Egress Point information and forwards it to the forwarding layer. The specific usage of the Tunnel Egress Point by the forwarding layer is beyond the scope of this document.

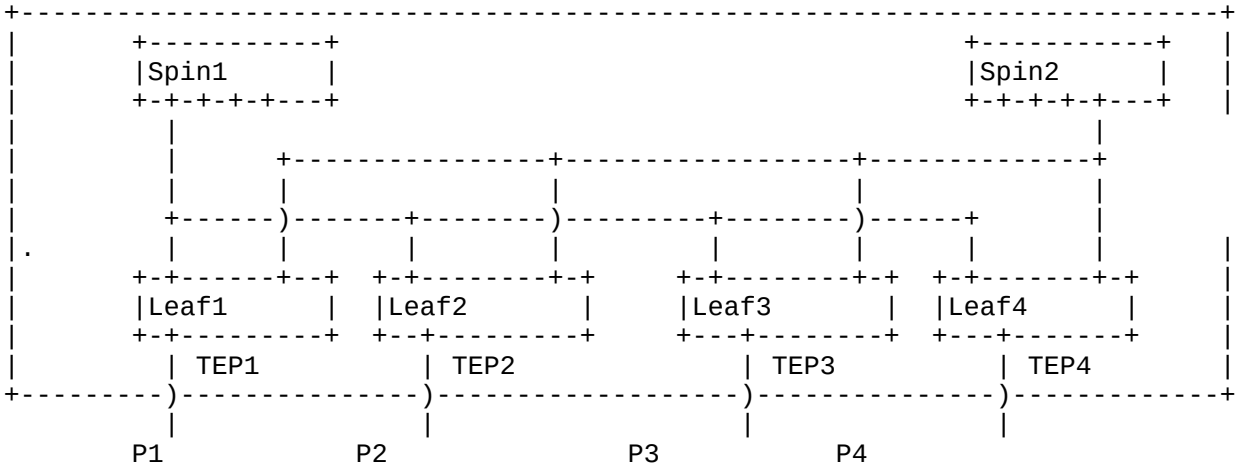

Figure 2: Spin/Leaf network

The forwarding paths for traffic are illustrated in Figure 3. For the same traffic from Leaf1 to Leaf2, there are two possible paths: Spin1->Leaf2 and Spin2->Leaf2. Different paths for the same traffic have the same Tunnel Egress Point information.

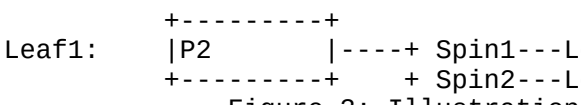

Figure 3: Illustration of Multiple Forwarding Paths

In addition to path information, to enable Leaf2 devices to directly forward packets without the need for secondary table lookups, Leaf1 devices can also prepare the required encapsulation information in advance. The encapsulation information is identified by an Encap ID and is included with the route when the Leaf device publishes it. Other devices, when forwarding packets, will include the Encap ID information if the route publisher has provided it.

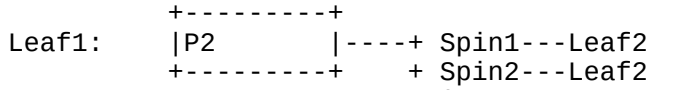

Figure 4

The specific synchronization process is as follows:

1) When Leaf2 devices announce routing information externally, they carry TEP2 information.

2) When Leaf2 devices announce the encapsulation information Encap ID1 to reach P2 externally.

3) When Leaf1 devices forward packets, they specify the forwarding path and the destination information TEP2. At the same time, based on the destination address P2, they specify the final encapsulation information Encap ID1 for sending.

4) The intermediate device independently determines the path to TEP2 and forwards the packet to TEP2.

5) TEP2, as the last hop router, directly encapsulates the packet according to the Encap ID1 and delivers the packet to P2.

To address network congestion, the data layer employs a Credit-based Flow Control mechanism to ensure that the egress end has sufficient capacity to receive data sent from the ingress end.

This necessitates the creation of a corresponding virtual queue (VOQ) at the ingress end for each transmission queue on the egress end. Typically, the establishment of these queues requires the control plane to assist the data plane in transmitting queue information corresponding to the physical interface from the egress end, aiming for enhanced interoperability. To this end, a new EVPN route type is needed to convey the information required for VOQ creation.

The specific process of entry synchronization and data handling is as follows:

1) The Egress device sends out port queue information to the Ingress device, which is only required during the initial establishment of BGP EVPN neighbor relationships.

2) The Ingress device creates the VOQ.

3) The Ingress device sends a Credit request.

4) The Egress device responds with a Credit ACK.

5) Data packets are transmitted.

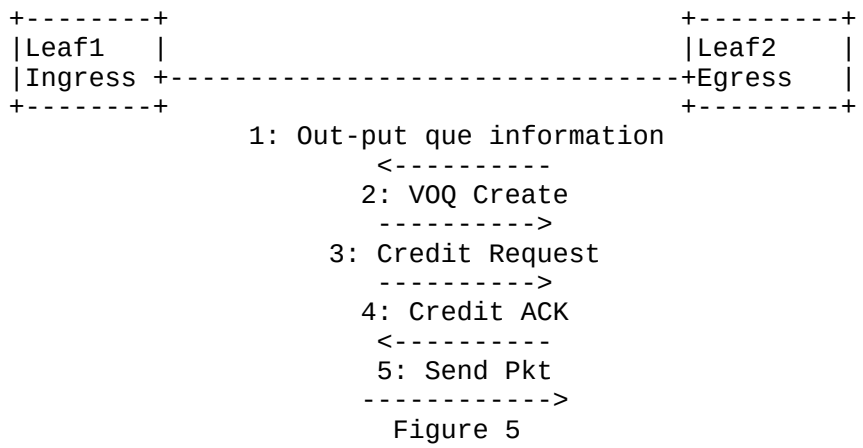

Figure 5

5. Protocol Extension

This section introduces the method of extending the BGP protocol to carry the Tunnel Egress Point information within the community attribute.

The Tunnel Egress Point information includes the Device Index and Port Index. The Device Index is globally unique and is used to distinguish different Leaf devices, while the Port Index is unique to the local device and is used to differentiate between different interfaces on the local device.

5.1. Extend for TEP(Tunnel Egress Point)

the TEP attribute is advertised as the path attribute type for BGP routes.

Add a new type, TEP type, to "BGP Tunnel Encapsulation Attribute Tunnel Types."

The TEP attribute is an optional transitive BGP path attribute.

```
 0               1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

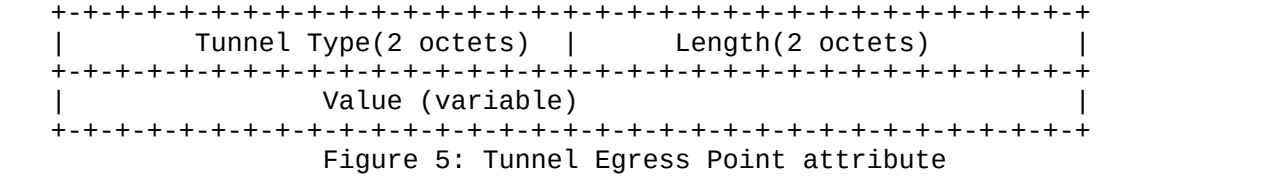

Figure 5: Tunnel Egress Point attribute

Tunnel Type: TBD, 2 octets. Identifies a type of tunnel. The field contains values from the IANA registry "BGP Tunnel Encapsulation Attribute Tunnel Types" [IANA-BGP-TUNNEL-ENCAP] [RFC9012]

Length: 2 octets, length of Value

Currently, two types of TEPs have been defined: one that carries a one Device ID and one Port ID attribute, and another that carries one Device ID multiple DevicePort attributes.

When the destination address is a unicast address, the corresponding destination node is a single node, and it carries a single DevicePort attribute. When the destination address is a broadcast address, the corresponding destination node is a group of nodes, and it carries one DeviceID and multiple PortID attributes.

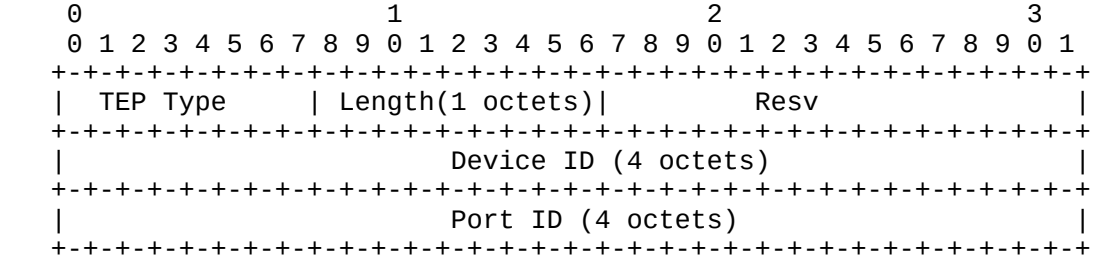

Figure 6: Single Port Index attribute

TEP Type 1: TBD1, Single Port Index, 1 octet

Length: 8, length of one DevicePort, 1 octet

Resv: 2 octets

Device ID: The Device ID, 4 octets

Port ID: The port ID, 4 octets

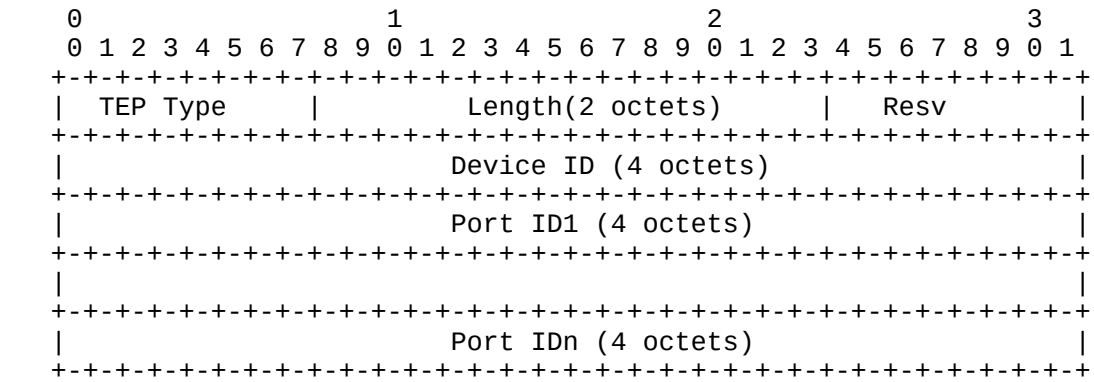

Figure 7: Multiple Port Index attribute

TEP Type 2: TBD2, Multiple Port Index

Resv: 1 octet

Length: DevicePort Total length

Device ID: The Device ID, 4 octets

Port ID: The port ID, 4 octets, can carry one or more, at least one.

5.2. Extend for Encap ID

The Encap ID occupies 2 bytes and is currently used only in EVPN networks. To facilitate the packaging of routes with the same attributes in BGP, the implementation includes the Encap ID as part of the NLRI information in EVPN routes. It reuses the Mpls ID2 field within the EVPN NLRI, eliminating the need for additional extensions, as shown below.

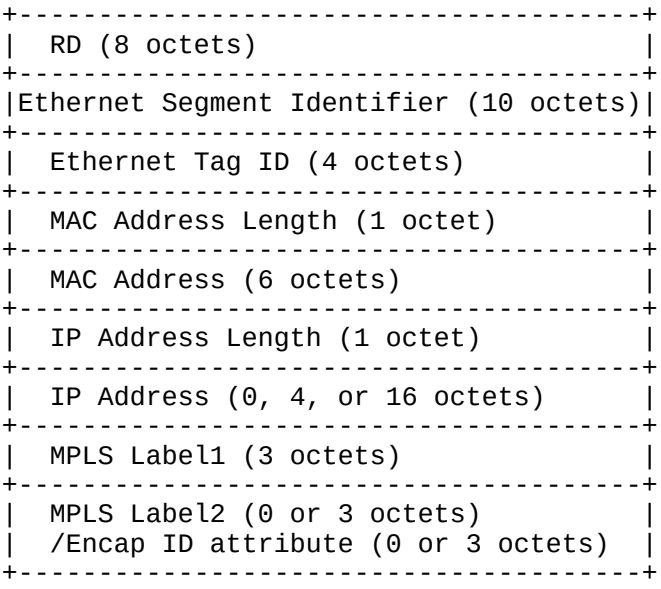

Figure 8

5.3. Extend for VOQ Information

To facilitate the transmission of all local interface-related VOQ information between two devices, it is necessary to introduce two new route types in EVPN. One route type represents device information, which includes details such as the device ID and the number of device interfaces. The other route type represents interface information, which includes the number of VOQs corresponding to a specific interface on the device. The first route type is referred to as the OSF-Device Route, and the second is referred to as the OSF-Interface Route.

After the EVPN neighbor relationship is established, the BGP of the Egress device must first transmit the OSF-Device Route to inform the peer of the total number of interfaces on the device. Subsequently, it transmits the OSF-Interface Route based on local interface information. The Ingress device, on the other hand, needs to parse the OSF-Device Route to determine how many OSF-Interface Routes it should receive, signifying the completion of the VOQ information transmission process. Once this process concludes, the Ingress device creates VOQ resources, thereby establishing the end-to-end Credit-based Flow Control mechanism.

The format of the OSF-Device Route is as shown in the figure below. It must include a 4-byte System ID, a 2-byte Interface Count, a 1-byte Flag, and a 1-byte reserved field. At this stage, both the Flag and Reserved fields must be set to 0.

OSF-Device Route:

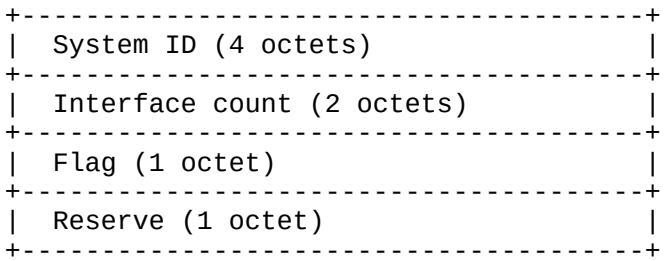

Figure 9

The format of the OSF-Interface Route is as shown in the figure below. The Device ID must match the one sent in the OSF-Device Route from the same device. The Interface ID is used to distinguish different interfaces on the same device, and it is possible for Interface IDs to be the same across different devices. The Interface Type is used to extend richer interface information. The Queue ID represents the specific VOQ, and a single interface may have multiple Queue IDs.

OSF-Interface Route:

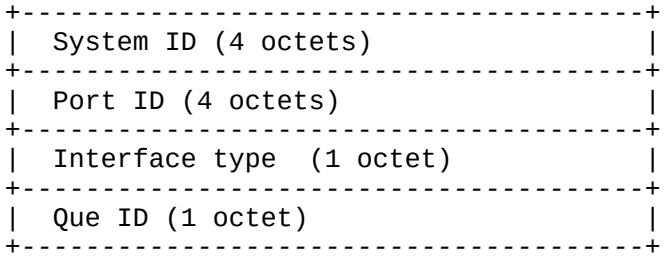

Figure 10

5.4. Implementation based on different types of networks

For the network shown in Figure 1, it can be a regular Layer 3 IP network or a Layer 2 network based on EVPN.

When advertising network route information, extended TEP attribute information is carried as path attribute.

If it is an EVPN network, the Encap ID is advertised with Type-2 MAC routes. When Leaf1 forwards a packet to a host under Leaf2’s P2, it first retrieves the TEP information based on the P2 route and then obtains the Encap ID information based on the host information. During packet encapsulation, both the TEP and Encap ID information are included in the packet sent to Leaf2.

The support for BGP Multicast VPN (MVPN) Services [RFC6513] with Tunnel Egress Point is outside the scope of this document.

6. Procedure

6.1. Procedure for IPv4/IPv6

When the control plane uses IPv4 or IPv6 unicast address families, the data plane does not require additional encapsulation extensions, except for sorting. The control plane only needs to add similar extensions like 5.1. The specific handling of the control plane and data plane is as follows.

Control Layer:

1) When OFP-Egress advertises IPv4/IPv6 prefix routes externally, the TEP attributes serve as path attribute types for these routes. For specific extension formats, refer to sections 5.1.

2) Upon receiving the prefix routes, OFP-Ingress updates the destination address and TEP information into the L3 forwarding table.

Forwarding Layer(details are not included in this document, the following is just the processing logic):

1) The L3 forwarding table records the TEP information.

2) During packet forwarding, OFP-Ingress sequences packets based on TEP information and embeds the TEP information and packet sequence number in the forwarded packet. How the TEP information and sequence number are carried within the forwarded packets is beyond the scope of this document.

3) OFP-Forward devices can choose to forward based on IP/IPv6 addresses or based on TEP info, without regard to packet disarray during forwarding.

4) Packets forwarded to OFP-Egress may arrive out of order due to differing intermediate paths.

5) OFP-Egress receives the packets, sorts them according to their sequence numbers, and if necessary, reassembles them, and forwards the packets to the server the original order sent by OFP-Ingress.

6) When delivering packets to the server, OFP-Egress adds the necessary encapsulation, and then delivers them to the server.

The information that the control plane’s OFP-Egress sends to the OFP-Ingress is shown in Figure 11.

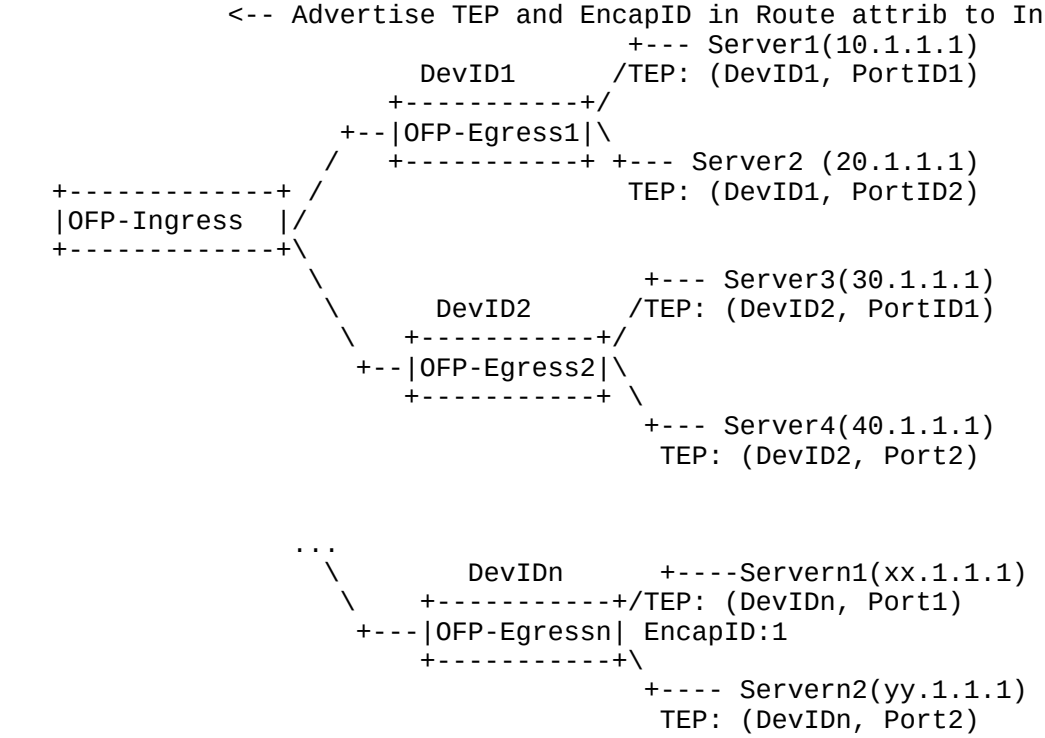

Figure 11

The data maintained by the OFP-Ingress is shown in Figure 12.

Prefix	TEP
10.1.1.1	DevID1,PortID1
20.1.1.1	DevID1,PortID2
30.1.1.1	DevID2,PortID1
40.1.1.1	DevID2,PortID2
xx.1.1.1	DevIDn,PortID1
yy.1.1.1	DevIDn,PortID1

Figure 12

The process of packet sending and reordering in the forwarding layer is shown in Figure 13. Here, p1, p2, and p3 are the three packets sent from OFP-Ingress to OFP-Egress. After being forwarded through multiple ECMP paths, they arrive at the OFP-Egress in the order p3, p2, p1. The OFP-Egress then reorders them based on the SequenceID, restoring the order to p1, p2, p3 before delivering them to the Server.

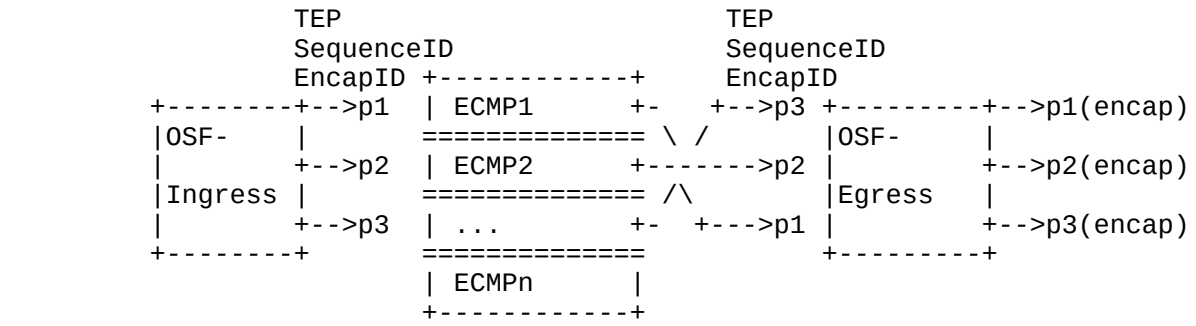

Figure 13

6.2. Procedure for EVPN

6.2.1. L2 Forwarding

Control Layer:

1) When OFP-Egress advertises Type 2 routes (MAC/IP advertisement) externally, it carries the TEP and Encap ID in the NLRI. For example, OFP-Egress1 advertise Type 2 route with MAC address (MAC1), IP Address (10.1.1.1), EncapID (1), TEP (DevID1, PortID1). For detailed formatting, see Figure 14.

2) Upon receiving the EVPN Type 2 routes, OFP-Ingress updates the destination address, TEP information, and EncapID into the L2 forwarding table.

Forwarding Layer:

1) The L2 forwarding table records the TEP and EncapID information.

2) During packet forwarding, OFP-Ingress sequences packets based on TEP information, embedding the TEP info and packet sequence number

in the forwarded packet. Additionally, it encapsulates the EncapID information within the packet. How the TEP information, sequence number, and EncapID are carried within forwarded packets is beyond the scope of this document.

3) OFP-Forward devices can choose to forward based on MAC or IP addresses, or based on TEP attributes, without regard to packet disarray during forwarding.

4) Packets forwarded to OFP-Egress may arrive out of order due to differing intermediate paths.

5) OFP-Egress receives the packets, sorts them according to their sequence numbers, and if necessary, reassembles them, and forwards the packets to the server the original order sent by OFP-Ingress.

6) When delivering packets to the server, OFP-Egress converts the packets according to EncapID information into local encapsulation formats, it then adds a L2-layer encapsulation to the packets based on the EncapID and forwards them to the server in sequence.

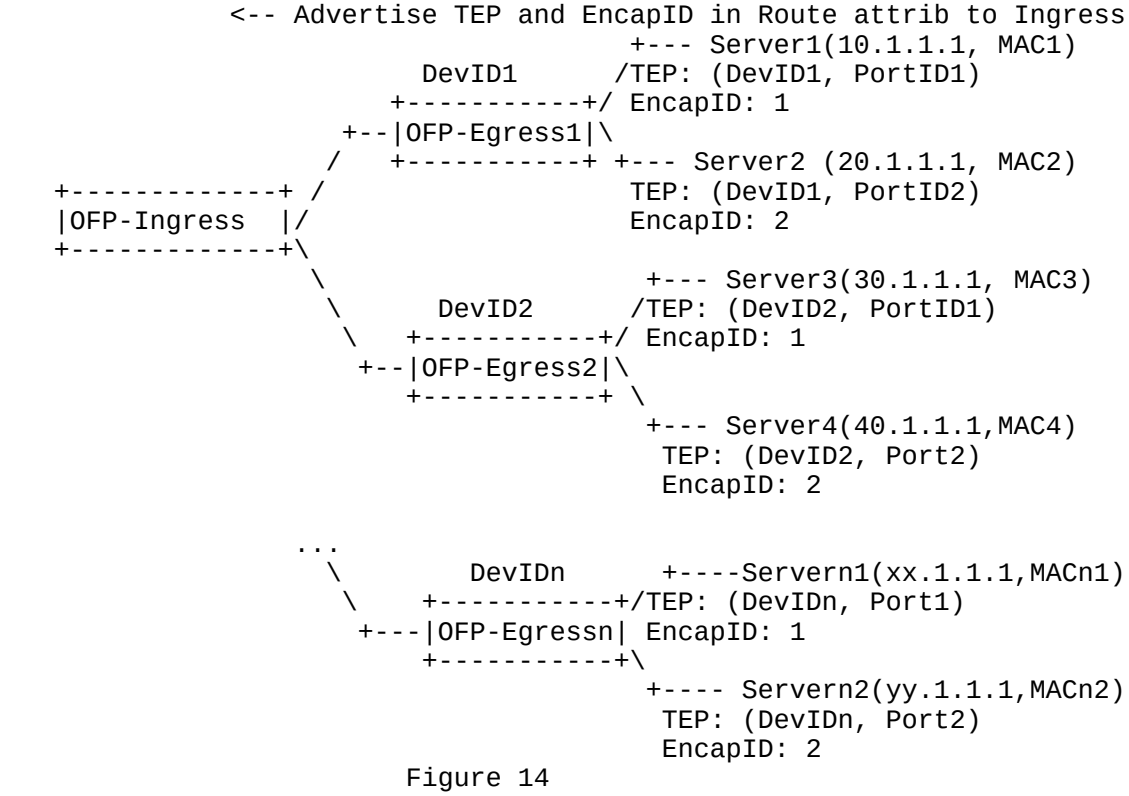

Figure 14

6.2.2. L3 Forwarding

Control Layer:

1) When OFP-Egress advertises Type 2 routes (MAC/IP advertisement) externally, it carries the TEP and Encap ID in the NLRI. When OFP-Egress advertises Type 5 routes (IP prefix), it carries TEP information. For example, OFP-Egress1 advertise Type 2 route with MAC address (MAC1), IP Address (10.1.1.1), EncapID (1). And advertise Type 5 route with IP address (100.1.1.0/24), GW IP address (10.1.1.1), TEP (DevID1, PortID1).

2) Upon receiving the EVPN Type 2 routes, OFP-Ingress updates the destination address, TEP information, and EncapID into the L2 forwarding table.

3) Upon receiving the EVPN Type 5 routes, OFP-Ingress looks up the corresponding Type 2 route using the Type 5 route’s GW IP address and inherits the Type 2 route’s EncapID. It then updates the

destination address, TEP information, and EncapID into the L3 forwarding table.

Forwarding Layer:

1) The L3 forwarding table records the TEP and EncapID information.

2) During packet forwarding, OFP-Ingress sequences packets based on TEP information, embedding the TEP info and packet sequence number in the forwarded packet. Additionally, it encapsulates the EncapID information within the packet. How the TEP information, sequence number, and EncapID are carried within forwarded packets is beyond the scope of this document.

3) OFP-Forward devices can choose to forward based on MAC or IP addresses, or based on TEP attributes, without regard to packet disarray during forwarding.

4) Packets forwarded to OFP-Egress may arrive out of order due to differing intermediate paths.

5) OFP-Egress receives the packets, sorts them according to their sequence numbers, and if necessary, reassembles them, and forwards the packets to the server the original order sent by OFP-Ingress.

6) When delivering packets to the server, OFP-Egress converts the packets according to EncapID information into local encapsulation formats, it then adds a L2-layer encapsulation to the packets based on the EncapID and forwards them to the server in sequence.

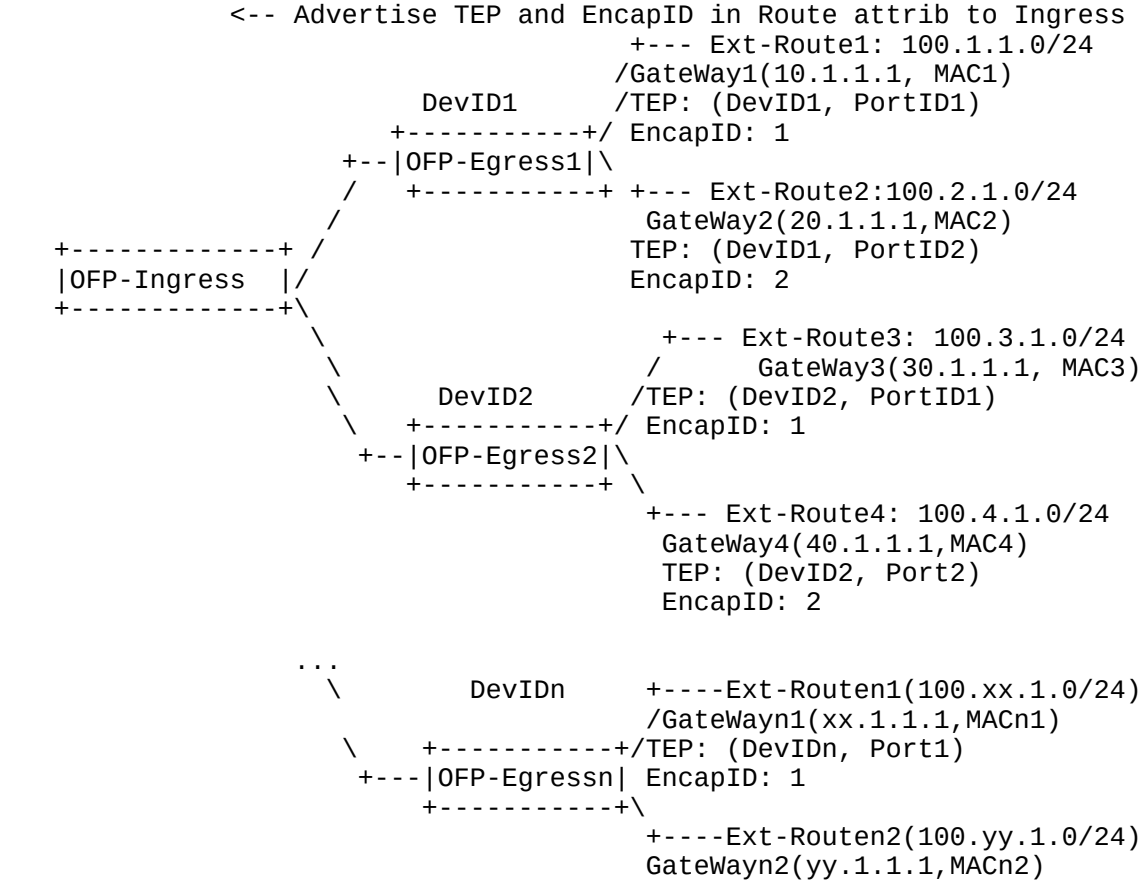

Figure 15

7. Deployment consideration

The Device ID of each Spin device must be globally unique, which can be ensured through configuration or by uniformly distributing guarantees through the controller.

8. IANA Considerations

This document registers the following in the "BGP Tunnel Encapsulation Attribute Tunnel Types" registry.[RFC9012]

TBD Tunnel Egress Point attribute

9. Security Considerations

TBD

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017

[RFC9012] K. Patel,

"The BGP Tunnel Encapsulation Attribute",

ISSN: 2070-1721, April 2021,

<https://datatracker.ietf.org/doc/rfc9012>.

10.2. Informative References

TBD.

Acknowledgments

TBD

Contributors

Jia Li
New H3C Technologies
China
Email: lij@h3c.com

Meng Li
New H3C Technologies
China
Email: li_meng_limeng@h3c.com

Jian Chen
New H3C Technologies
China
Email: jian_chen@h3c.com

Haina Zhong
New H3C Technologies
China
Email: zhonghaina.06454@h3c.com

Jincan Li
RuiJie
China
Email: lijincan@ruijie.com.cn

Yanrong Liang
RuiJie
China
Email: liangyanrong@ruijie.com.cn

Daniel Roytenberg
DriveNets
Email: danielro@drivenets.com

Eyal Hezi
DriveNets
Email: ehezi@drivenets.com

Alvin Yu Zhang
DriveNets
Email: azhang@drivenets.com

Yehonatan Lemberger
DriveNets
Email: ylemberger@drivenets.com

Yanjun Yang
Broadcom
Email: Yanjun.yang@broadcom.com

Authors’ Addresses

PengFei Huo
ByteDance
China
Email: huopengfei@bytedance.com

Gang Chen
ByteDance
China
Email: chengang.gary@bytedance.com

Changwang Lin
New H3C Technologies
China

Email: linchangwang.04414@h3c.com

Weiqiang Cheng
China Mobile
china

Email: chengweiqiang@chinamobile.com

Syed Hasan Raza Naqvi
Broadcom
Email: syed.naqvi@broadcom.com

Yossi Kikozashvili
DriveNets
Email: ykikozashvili@drivenets.com

Chenchen Qi
ByteDance
China
Email: qichenchen@bytedance.com