SPRING
Internet-Draft
Intended status: Experimental
Expires: 20 February 2021

K. Fang
Cisco Systems, Inc.
Y. Li
Google, Inc.
F. Cai
X. Jiang
Cisco Systems, Inc.

19 August 2020

Distributed KV Store based Routing protocol for SR over UDP(SRoU) draft-zartbot-srou-control-00

Abstract

This document defines the Distributed KV store based routing protocol for Segment Routing over UDP.

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 20 February 2021.

Copyright Notice

Table of Contents

1. Introduction

This draft provides a contol plane support for SRoU(Segment Routing over UDP).

Discussion of this work is encouraged to happen on GitHub repository which contains the draft: https://github.com/zartbot/draft-quic-sr (https://github.com/zartbot/draft-quic-sr)

1.1. Specification of Requirements

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.

1.2. Motivation

SRoU support udp transport session over internet, but it lack of reachability detection and routing control, existing routing protocol like BGP-EVPN did not provide Dynamic NAT traversal capability.

This document provide a distributed KV store based routing protocol for SRoU.

1.3. Overview

The routing protocol is based on source routing, each of the ingress node cloud get the overlay prefix and dest location mapping from distributed KV store, then the ingress node could fetch linkstate database from this KV store and execute A* algorithm to search the candidate path which meet the SLA requirement.

2. Node abstraction and registration

Each Node has the following attribute

1. Role: the system contains different node type, role attribute is a uint16 value which contains:

Type	Name	Section
0x0	STUN	This node is used as a STUN server to help other nodes discovery their public address.This node must deploy with a public internet address or behind static 1:1 NAT
0x1	Fabric	This node type is used as a interim node to relay the SRoU traffic, this node MUST initial TWAMP link probe to other Fabric node and report linkstate to KV Store.
0x2	Linecard	This node type is used to connect existing network, it could use TWAMP probe other Fabric Node or Linecard node

Table 1: Node Role

1. SiteID: uint32 number, defined the node which belongs to same site or Automomous System.

2. SystemName: unique string type to indicate a node.

3. Label: unique 24bit value, allocation algorithm is described in the following section.

4. Location: Optional filed. It contains two float32 value(latitude and longitude) to indicate the Geo location.

2.1. Node Label allocation

Each node initial TLS session to Distributed KV Store, and fetch a distributed lock with key "/lock/systemlabel". The node will fetch prefix "/systemlabel" to get all label mapping once it get the lock. Then it will assign the smallest unpresent int "X" in the list as it's system label, and register it to KV store by key="/systemlabel/X", then it could release the distributed lock. All of the fabric node MUST listen the "/systemlabel" to update it's local node mapping table, Linecard node may fetch the "/systemlabel" key when it need to optimize the local route.

This System Label could be used for cSID encoding or VPN based client linecard node convert to it's tunnel address.

2.2. Node registration

Each node will send Key="/node/role/systemName" and Value="SiteID,SystemLabel, Lat,Long" to the distributed KV store.

3. SRoU Locator and Route

Each node may have multiple underlay socket which may behind the dynamic NAT, it MUST fetch the STUN list from "/node/stun" and "/service/stun" to get the STUN server address list, then send the SRoU OAM-STUN packet to the random selected stun server to get the public address.

Once the socket get the public address, it will encode the udp socket info as a SRoU Locator:

"SystemName/Color/LocalIP:Port/PublicIP:Port/LocalInterface/TXBW/RXBW"

If the local socket has public address and port information, it could be added in the service list.

The node MUST update it local servicelist to distributed KV store by: Key= "/service/role/systemName" Value= "SRoULocator1,SRoULocator2"

4. Node Keepalive

Each KV pair registration MUST have a leasetime and keepalive timer, Once the Node out of service and disconnected, the KV store MUST withdraw the KV pair after lease timeout.

5. Link State

Each Fabric Node must watch the "/service/fabric" key prefix to update its local SRoU Service list database. It MUST initial TWAMP session over the service udp socket to measure the link performance and reachablity.

Linkstate measurement result COULD send to the KV store to construct the linkstate Database by the following Key Value type:

Key="/stats/linkstate/SRC_SRoU_Locator->DST_SRoU_Locator" value= TWAMP measured jitter/delay/loss result and underlay interface load.

The Node CPU,Memory usage also could be updated by: Key="/stats/node/SystemName" Value="CPULoad,MemoryUsage"

An telemetry analytics node could watch key prefix ="/stats" for assurance and AIOps based routing optimization.

6. Sercurity Key

Each node may update it node key or per socket key , or per session pair key to the KV Store:

Key="/key/SystemName" Value="Key1,Key2"

Key="/key/socket/SRoU_Locator" Value="Key1,Key2"

Key="/key/session/SRC_SRoU_Locator->DST_SRoU_Locator" Value="Key1,Key2"

During Rekey, the node must update both OldKey and newKey to the KV Store and accept both Key in a while to wait the entire system sync to the new key.

7. Overlay Routing

RouteDistinguish could encode by SystemName + local VNID The overlay routing prefix is encoded as below:

Type-2 EVPN Route Key="/route/2/exportRT/RD/MAC/IP" Value="VNID/SystemName/PolicyTag"

Type-5 EVPN Route

Key="/route/5/exportRT/RD/IPPrefix/IPMask" Value="VNID/SystemName/PolicyTag"

Each of the linecard node could based on import RT list to watch key prefix ="/route/2/importRT" and "/route/5/importRT" to sync the routing table.

Each linecard node could selective fetch the "/stats/linkstate" to get the toplogy information and execute flexibile algorithm(SPF,A* search) to calculate the candidate path, then enforce it to its forwarding table.

8. Control Policy

8.1. Route control

Inspired by BGP FlowSpec,Network operator could update the control policy to the entire system by using:

Key="/control/RT/2/SRC_MAC/SRC_IP/DST_MAC/DST_IP"
Key="/control/RT/5/SRC_Prefix/SRC_Mask/DST_Prefix/DST_Mask"
Value="Action" /"SR Locator list"

8.2. Access Control

Each node may use the SRoU flowID field as a token based access control. This token could grant or revoke by a policy engine.

Key="/token/permit/flowid" Key="/token/block/flowid"

Each node could sync this table to execute the access control policy.

8.3. User identity

Each of the endpoint may have it's identity or group policy tags, it could be updated by

key="/identity/userid/user_device_id" value="group policy tags"

Group policy could be updated and store in ETCD by

key="/policy/src_grp/dst_grp" value="actions"

9. Distributed KV Store

ETCD is used in our prototype, we deploy an etcd cluster in main datacenter and place many of the proxy node on public cloud to make sure the node could be available connect to the entire system. In some on-prem deployment, each of the nodes could act as a ETCD proxy to help other node register to KV store.

10. Security Considerations

All of the control connection is TLS based and MUST validate the server and client certification.

11. IANA Considerations

Acknowledgements

The following people provided substantial contributions to this document:

* Yijen Wang, Cisco Systems, Inc.

Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

Authors' Addresses

Kevin Fang
Cisco Systems, Inc.

Email: zartbot.ietf@gmail.com

Yinghao Li
Google, Inc.

Email: liyinghao@gmail.com

Feng Cai
Cisco Systems, Inc.

Email: fecai@cisco.com

Xing Jiang
Cisco Systems, Inc.

Email: jamjiang@cisco.com