INTERNET-DRAFT
Intended Status: Standard Track

Sami Boutros
Ankur Dubey
VMware

Reshad Rahman
Cisco

Expires: May 31, 2018 November 27, 2017

Service Redundancy using BFD
draft-adubey-bfd-service-redundancy-01

Abstract

In a data center, when multiple routing/service nodes are providing single active redundancy for a set of L2, L3 and/or L4-L7 services. Both non-revertive and revertive fail over modes are required for the services. This draft describes a method to achieve the non-revertive and revertive fail over modes for services using Bidirectional Forwarding Detection (BFD).

Table of Contents

1 Introduction

This document describes how can a group of service/routing nodes in a data center providing single active redundancy for multiple L2/L3 and/or L4/L7 services, can use BFD protocol to support non-revertive as well as revertive fail over mode.

Typically, BFD is used between the group of service nodes to verify the connectivity as well as the aliveness of the service nodes. The assignment of which node in the group is the primary designated forwarder for a given service can be determined using a centralized or distributed control plane.

The use of BFD will be to communicate the set of services that are being currently active on a given service node to the other service nodes. On a given node failure, for a given service the backup node will take over. If the service was configured to have a non-revertive fail over mode, then the backup node should continue to perform the service forwarding even after the primary node recovers and comes back up. In order to do that, the backup node MUST inform the primary node that it is currently active for the service. This is achieved through the extension we are proposing to the BFD protocol as will be described in the following sections.

It is to be noted that for revertive fail over mode of operation, the primary node should be able to take over the active role from the backup node when the primary node goes back to an operational state. This can be as well communicated using the BFD session establishment between the primary node and the backup node.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Solution Overview

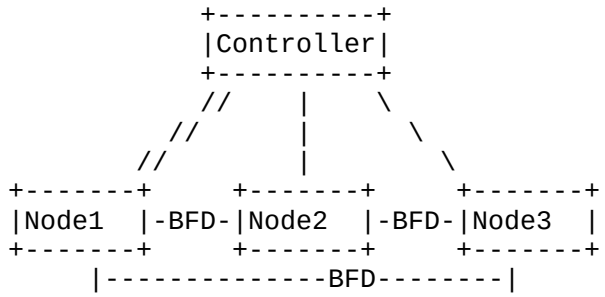


Figure 1:

Figure 1 shows 3 routing nodes using BFD to implement the single active redundancy for revertive and non-revertive services. More than 3 routing nodes can be used.

Multiple L2/L3 and/or L4/L7 services are offered in a data center by a set of routing/service nodes providing single active redundancy. The provisioning of the services can be done using a centralized control plane implemented in a controller or using a distributed dynamic control plane.

2.1 Node failover

An implementation MAY choose to support only node failover and not a per service failover. A node can be primary or backup for a given service. On a primary node failure, all non-revertive and revertive services will become active on the backup node.

In figure 1, lets assume that Node1 is the primary node for a set A of non-revertive services with node2 as backup, and another set B of non-revertive services with Node3 as backup. As well, Node1 is primary for a set C of revertive services with Node2 as backup and, another set D of revertive services with Node3 as backup.

If Node1 fails, Node2 and Node3 will set a new diag code in the BFD control packet. This diag code will inform Node1 that both Node2 and Node3 didn’t fail, and Node1 MUST NOT activate the non-revertive set of services A and B respectively, when it comes back up. The BFD control packet with the new diag code will be sent after the BFD session came up for at least twice the detection multiplier count.

Therefore, Node1 upon receiving the BFD control packet with the new diag code, MUST not attempt to activate the non-revertive services, but remain in standby state for the non-revertive services until the Node2 or Node3 that took over fails.

Revertive services are assumed to revert back to the primary node Node1, after the node recovers. Once the BFD session comes up between the primary and backup nodes, the backup node should stop forwarding for any revertive services. A node MUST start forwarding all revertive services for which it is configured as a primary once the BFD session comes up with the corresponding backup nodes. A node MUST stop forwarding for revertive services for which it is a backup once the BFD session comes up with the corresponding primary.

2.2 Per service failover for non-revertive services

An implementation MAY choose to support per service failover for non-revertive services. For example, in figure1, some non-revertive services could be active on Node1 while some non-revertive services could be active on Node2 or Node3 for better load balancing of services traffic. In this mode, every L2/L3 and/or L4/L7 non-revertive service will be identified by a unique ID known across the routing/service nodes providing the services.

A bitmap will be used to represent the non-revertive services, where each non-revertive service is represented by one bit in the bitmap. All the service nodes MUST have the same mapping of the bit position to the non-revertive service unique ID. The bitmap position and the unique service ID could be maintained by a network controller.

A node that is assigned as backup for a given non-revertive service node will take over as active in either of the following cases: 1) The node assigned as primary for this service failed. 2) This specific service failed on the primary node for this service.

In case 1, the BFD session will go down since it is a node failure. In case 2, BFD session between the nodes will remain up. In either scenarios, the node assigned as secondary will become active for the non-revertive service. In case 1, the secondary node will set the new diag code in the BFD control packets once the BFD session is established. The new diag code will be set in the BFD control packets for at least twice the detection multiplier count. In case 2, this diag code will be set in the next BFD control packets sent after the node takes over as Active for a given non-revertive service. If there is at least one non-revertive service for which this node is not active AND at least 1 non-revertive service for which it is active, the node will also send the bitmap in the BFD control packets payload. The bits identifying the active non-revertive services will

be set in this bitmap. The new diag code and the optional bitmap payload will be sent in the BFD control packets for at least twice the detection multiplier count.

Therefore, if a node receives a BFD control packet with the new diag code set but no payload in the BFD control packet, this means that it MUST NOT activate all non-revertive services for which this node is primary. Whereas, if a payload is present in the BFD control packet that has the new diag code set, the receiving node MUST NOT activate the non-revertive services indicated by the set bits in the bitmap.

Per service failover is not applicable to revertive services. They will behave the same way as described in section 2.1

3 Acknowledgements

4 Security Considerations

This document does not introduce any additional security constraints.

5 IANA Considerations

IANA is requested to assign a new diag code from the "BFD Diagnostic Codes"

Value	BFD Diagnostic Code Name
0xNN	Out-lived and optional BitMap BFD control packet payload for non-revertive services.

6 References

[RFC5880] D. Katz, D. Ward "Bidirectional Forwarding Detection (BFD)".

Authors' Addresses

Sami Boutros
VMware
Email: sboutros@vmware.com

Ankur Dubey
VMware
Email: adubey@vmware.com

Reshad Rahman
Cisco
Email: rrahman@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 6, 2018

J. Arkko
Ericsson
J. Tantsura
Nuagenetworks
J. Halpern
B. Varga
Ericsson

March 5, 2018

Considerations on Network Virtualization and Slicing
draft-arkko-arch-virtualization-01

Abstract

This document makes some observations on the effects of virtualization on Internet architecture, as well as provides some guidelines for further work at the IETF relating to virtualization.

This document also provides a summary of IETF technologies that relate to network virtualization. An understanding of what current technologies there exist and what they can or cannot do is the first step in developing plans for possible extensions.

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 http://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 6, 2018.

Table of Contents

1. Introduction

Network virtualization is network management pertaining to treating different traffic categories in separate virtual networks, with independent lifecycle management and resource, technology, and topology choices.

This document makes some observations on the effects of virtualization on Internet architecture, as well as provides some guidelines for further work at the IETF relating to virtualization.

This document also provides a summary of IETF technologies that relate to network virtualization. An understanding of what current technologies there exist and what they can or cannot do is the first step in developing plans for possible extensions.

In particular, many IETF discussions earlier in the summer of 2017 started from a top-down view of new virtualization technologies, but were often unable to explain the necessary delta to the wealth of existing IETF technology in this space. This document takes a different, bottom-up approach to the topic and attempts to document existing technology, and then identify areas of needed development.

In particular, whether one calls a particular piece of technology "virtualization", "slicing", "separation", or "network selection" does not matter at the level of a system. Any modern system will use several underlying technology components that may use different terms but provide some separation or management. So, for instance, in a given system you may use VLAN tags in an ethernet segment, MPLS or VPNs across the domain, NAIs to select the right AAA instance, and run all this top of virtualized operating system and software-based switches. As new needs are being recognised in the developing virtualization technology, what should drive the work is the need for specific capabilities rather than the need to distinghuish a particular term from another term.

2. Definitions

Network function virtualization is defined in Wikipedia as follows:

> "Network function virtualization or NFV is a network architecture concept that uses the technologies of IT virtualization to virtualize entire classes of network node functions into building blocks that may connect, or chain together, to create communication services.
>
> NFV relies upon, but differs from, traditional server-virtualization techniques, such as those used in enterprise IT. A virtualized network function, or VNF, may consist of one or more virtual machines running different software and processes, on top of standard high-volume servers, switches and storage devices, or even cloud computing infrastructure, instead of having custom hardware appliances for each network function."

We should not confuse NFV and network virtualization, the former, as the name suggests is about functions virtualization, and not the network.

The idea of network virtualization is almost as old as the networking technology itself. Network virtualization is hierarchical and multilayer in its nature, from layer 1 up to services on top. When talking about virtualization we usually define overlay to underlay relationship between different layers, bottom up. A VPN (Virtual Private Network) [RFC4026] is the most common form of network virtualization. The general benefits and desirability of VPNs have been described many times and in many places ([RFC4110] and [RFC4664]).

The only immutable infrastructure is the "physical" medium, that could be dedicated or "sliced" to provide services(VPNs) in a multi-tenant environment.

The term slicing has been used to describe a virtualization concept in planned 5G networks. The 3GPP architecture specification [TS-3GPP.23.501] defines network slices as having potentially different "supported features and network functions optimisations", and spanning functions from core network to radio access networks.

[I-D.king-teas-applicability-actn-slicing] defined slicing as "an approach to network operations that builds on the concept of network abstraction to provide programmability, flexibility, and modularity. It may use techniques such as Software Defined Networking (SDN) and Network Function Virtualization (NFV) to create multiple logical (virtual) networks, each tailored for a set of services that are sharing the same set of requirements, on top of a common network.

And, [I-D.geng-coms-problem-statement] defines slicing as a management mechanism that an service provider can use to allocate dedicated network resources from shared network infrastructures to a tenant.

3. General Observations

Software vs. Protocols

Many of the necessary tools for using virtualization are software, e.g., tools that enable running processes or entire machines in a virtual environment decoupled from physical machines and isolated from each other, virtual switches that connect systems together, management tools to set up virtual environments, and so on. From a communications perspective these tools operate largely in the same fashion as their real-world counterparts do, except that there may not be wires or other physical communication channels, and that connections can be made in the desired fashion.

In general, there is no reason for protocols to change just because a function or a connection exists on a virtual platform. However, sometimes there are useful underlying technologies that facilitiate connection to virtualized systems, or optimised or additional tools that are needed in the the virtualized environment.

For instance, many underlying technologies enable virtualization at hardware or physical networking level. For instance, Ethernet networks have Virtual LAN (VLAN) tags and mobile networks have a choice of Access Point Names (APNs). These techniques allow users and traffic to be put on specific networks, which in turn may comprise of virtual components.

Other examples of protocols providing helpful techniques include virtual private networking mechanisms or management mechanisms and data models that can assist in setting up and administering virtualized systems.

There may also be situations where scaling demands changes in protocols. An ability to replicate many instances may push the limits of protocol mechanisms that were designed primarily or originally for physical networks.

Selection vs. Creation and Orchestration

Two primary tasks in virtualization should be differentiated: selection of a particular virtual instance, and the tasks related to how that virtual instance was created and continues to be managed.

Selection involves choosing a particular virtual instance, or an entrypoint to a virtual network. In its simplest form, a customer could be hardwired by configuration to a particular virtual instance. In more complex cases, the connecting devices may have some settings that affect the choice. In the general case, both the connecting devices and the network they are connecting to it have a say in the choice.

The selection choice may even be dynamic in some cases. For instance, traffic pattern analysis may affect the selection.

Typically, however, connecting devices do not have a say in what the virtual instance does. This is directed by the network operator and its customers. An instance is specified, created, and needs to be continously managed and orchestrated. The creation can be manual and occur rarely, or be more dynamic, e.g., an instance can actually be instantiated automatically, and only when the first connecting device connects to it.

Protocols vs. Representations of Virtual Networks

Some of virtualization technology benefits from protocol support either in the data or control plane. But there are also management constructs, such as data models representing virtual services or networks and data models useful in the construction of such services.

There are also conceptual definitions that may be needed when constructing either protocols or data models or when discussing service agreements between providers and consumers.

4. Virtualization in 5G Networks

Goals for the support of virtualization in 5G relate to both the use of virtualized network functions to build the 5G network, and to enabling the separation of different user or traffic classes into separate network constructs called slices.

Slices enable a separation of concerns, allow the creation of dedicated services for special traffic types, allow faster evolution of the network mechanisms by easing gradual migration to new functionality, and enable faster time to market for new new functionality.

In 5G, slice selection happens as a combination of settings in the User Equipment (UE) and the network. Settings in the UE include, for instance, the Access Point Name (APN), Dedicated Core Network Indicator (DCN-ID) [TS-3GPP.23.401], and, with 5G, a slice indicator (Network Slice Selection Assistance Information or NSSAI) [TS-3GPP.23.501]. This information is combined with the information configured in the network for a given subscriber and the policies of the networks involved. Ultimately, a slice is selected.

A 5G access network carries a user’s connection attempt to the 5G core network and the Access Management Function (AMF) network function. This function collects information provided by the UE and the subscriber database from home network, and consults the Network Slice Selection Function (NSSF) to make a decision of the slice selected for the user. When the selection has been made, this may also mean that the connection is moved to a different AMF; enabling separate networks to have entirely different network-level service.

The creation and orchestration of slices does not happen at this signalling plane, but rather the slices are separately specified, created, and managed, typically with the help of an orchestrator function.

The exact mechanisms for doing this continue to evolve, but in any case involve multiple layers of technology, ranging from underlying virtualization software to network component configuration mechanisms and models (often in YANG) to higher abstraction level descriptions (often in TOSCA), to orchestrator software.

5. Overview of IETF Virtualization Technologies

General networking protocols are largely agnostic to virtualization. TCP/IP does not care whether it runs on a physical wire or on a computer-created connection between virtual devices.

As a result, virtualization generally does not affect TCP/IP itself or applications running on top. There are some exceptions, though, such as when the need to virtualize has caused previously held assumptions to break, and the Internet community has had to provide new solutions. For instance, early versions of the HTTP protocol assumed a single host served a single website. The advent of virtual hosting and pressure to not use large numbers of IPv4 addresses lead to HTTP 1.1 adopting virtual hosting, where the identified web host is indicated inside the HTTP protocol rather than inferred from the reception of a request at particular IP address [VirtualHosting] [RFC2616].

But where virtualization affects the Internet architecture and implementations is at lower layers, the physical and MAC layers, the systems that deal with the delivery of IP packets to the right destination, management frameworks controlling these systems, and data models designed to help the creation, monitoring, or management of virtualized services.

What follows is an overview of existing technologies and technologies currently under development that support virtualization in its various forms.

5.1. Selection of Virtual Instances

Some L2 technology allows the identification of traffic belonging to a particular virtual network or connection. For instance, Ethernet VLAN tags.

There are some IETF technologies that also allow similar identification of connections setup with the help of IETF protocols. For instance, Network Access Identifiers may identify a particular customer or virtual service within AAA, EAP or IKEv2 VPN connections.

5.2. Traffic Separation in VPNs

Technologies that assist separation and engineering of networks include both end-point and provider-based VPNs. End-point VPN tehchnologies include, for instance, IPsec-based VPNs [RFC4301].

For providing virtualized services, however, provider-based solutions are often the most relevant ones. L1VPN facilitates virtualization of the underlying L0 "physical" medium. L2[IEEE802.1Q] facilitates virtualization of the underlying Ethernet network Tunneling over IP (MPLS, GRE, VxLAN, IPinIP, L2TP, etc) facilitates virtualization of the underlying IP network - MPLS LSP’s - either traffic engineered or not belong here L2VPN facilitates virtualization of a L2 network L3VPN facilitates virtualization of a L3 network.

The IETF has defined a multiplicity of technologies that can be used for provider-based VPNs. The technologies choices available can be described along two axes, control mechanisms and dataplane encapsulation mechanisms. The two are not compeltely orthogonal.

In the data plane, for provider based VPNs, the first important observation is that the most obvious encapsulation is NOT used. While IPSec could be used for provider-based VPNs, it does not appear to be used in practice, and is not the focus for any of the available control mechanisms. Often, when end2end encryption is required it is used as an overlay over MPLS based L3VPN

The common encapsulation for provider-based VPNs is to use MPLS. This is particularly common for VPNs within one operator, and is sometimes supported across operators.

Keyed GRE can be used, particularly for cross-operator cases. However, it seems to be rare in practice.

The usage of MPLS for provider-based VPNs generally follows a pattern of using two (or more) MPLS labels, top (transport) label to represent the remote end point/egress provider-edge device, and bottom (service) label to signal the different VPNs on the remote end point. Using TE might result in a deeper label stack.

L2 VPNs could be signaled thru LDP[RFC4762] or MP-BGP[RFC4761], L3 VPN is signaled thru MP-BGP[RFC4364]

The LDP usage to control VPN establishment falls within the PALS working group, and is used to establish pseudo-wires to carry Ethernet (or lower layer) traffic. The Ethernet cases tend to be called VPLS (Virtual Private LAN Service) for multi-point connectivity and VPWS (Virtual Private Wire Service) for point-to-point connectivity. These mechanism do augment the data plane capabilites with control words that support additional features. In operation, LDP is used to signal the communicating end-points that are interested in communicating with each other in support of specific VPNs. Information about the MAC addresses used behind the provider edges is exchanged using classic Ethernet flooding technology. It has been proposed to use BGP to bootstrap the exchang eof information as to who the communicating endpoints are.

BGP can be used to establish Layer 2 or Layer 3 VPNs. Originally, the BGP based MPLS VPN technology was developed to support layer 3 VPNs. the BGP exchanges uses several different features in MP-BGP (specifically route distinguishers and route targets) to control the distribution of information about VPN end-points. The BGP information carries the VPN IP address prefixes, and the MPLS labels

to be used to represent the VPN. This technolgoy combination is generally known as L3VPN.

This usage of BGP for VPNs has been extended to support Layer 2 VPNs. This is known as EVPN. The BGP exchanges are used to carry the MAC address reachability behind each provider edge router, providing an Ethernet multipoint service without a need to flood unkown-destination Ethernet packets.

In theory, the BGP mechanisms can also be used to support other tunnels such as keyed GRE. That is not widely practiced.

There are also hybrid variations, such as adding an ARP / ND proxy service so that an L3VPN can be used with an L2 Access, when the only desired service is IP.

5.3. Traffic Engineering and QoS

Traffic Engineering (TE) is the term used to refer to techniques that enable operators to control how specific traffic flows are treated within their networks.

The TEAS working group works on enhancements to traffic-engineering capabilities for MPLS and GMPLS networks:

> TE is applied to packet networks via MPLS TE tunnels and LSPs. The MPLS-TE control plane was generalized to additionally support non-packet technologies via GMPLS. RSVP-TE is the signaling protocol used for both MPLS-TE and GMPLS.
>
> The TEAS WG is responsible for:
>
> * Traffic-engineering architectures for generic applicability across packet and non-packet networks.
>
> * Definition of protocol-independent metrics and parameters.
>
> * Functional specification of extensions for routing (OSPF, ISIS), for path computation (PCE), and RSVP-TE to provide general enablers of traffic-engineering systems.
>
> * Definition of control plane mechanisms and extensions to allow the setup and maintenance of TE paths and TE tunnels that span multiple domains and/or switching technologies.

A good example of work that is currently considered in the TEAS WG is the set of models that detail earlier IETF-developed topology models with both traffic engineering information and connection to what

services are running on top of the network [I-D.bryskin-teas-use-cases-sf-aware-topo-model] [I-D.bryskin-teas-sf-aware-topo-model]. These models enable reasoning about the state of the network with respect to those services, and to set up services with optimal network connectivity.

Traffic engineering is a common requirement for many routing systems, and also discussed, e.g., in the context of LISP.

5.4. Service Chaining

The SFC working group has defined the concept of Service Chaining:

> Today, common deployment models have service functions inserted on the data-forwarding path between communicating peers. Going forward, however, there is a need to move to a different model, where service functions, whether physical or virtualized, are not required to reside on the direct data path and traffic is instead steered through required service functions, wherever they are deployed.
>
> For a given service, the abstracted view of the required service functions and the order in which they are to be applied is called a Service Function Chain (SFC). An SFC is instantiated through selection of specific service function instances on specific network nodes to form a service graph: this is called a Service Function Path (SFP). The service functions may be applied at any layer within the network protocol stack (network layer, transport layer, application layer, etc.).

5.5. Management Frameworks and Data Models

There have been two working groups at the IETF, focusing on data models describing VPNs. The IETF and the industry in general is currently specifying a set of YANG models for network element and protocol configuration [RFC6020].

YANG is a powerful and versatile data modeling language that was designed from the requirements of network operators for an easy to use and robust mechanism for provisioning devices and services across networks. It was originally designed at the Internet Engineering Task Force (IETF) and has been so successful that it has been adopted as the standard for modeling design in many other standards bodies such as the Metro Ethernet Forum, OpenDaylight, OpenConfig, and others. The number of YANG modules being implemented for interfaces, devices, and service is growing rapidly.

(It should be noted that there are also other description formats, e.g., Topology and Orchestration Specification for Cloud Applications (TOSCA) [TOSCA-1.0] [TOSCA-Profile-1.1], common in many higher abstract level network service descriptions. The ONAP open source project plans to employ it for abstract mobile network slicing models, for instance.)

A service model is an abstract model, at a higher level than network element or protocol configuration. A service model for VPN service describes a VPN in a manner that a customer of the VPN service would see it.

It needs to be clearly understood that such a service model is not a configuration model. That is, it does not provide details for configuring network elements or protocols: that work is expected to be carried out in other protocol-specific working groups. Instead, service models contain the characteristics of the service as discussed between the operators and their customers. A separate process is responsible for mapping this customer service model onto the protocols and network elements depending on how the network operator chooses to realise the service.

The L2SM WG specifies a service model for L2-based VPNs:

> The Layer Two Virtual Private Network Service Model (L2SM) working group is a short-lived WG. It is tasked to create a YANG data model that describes a L2VPN service (a L2VPN customer service model). The model can be used for communication between customers and network operators, and to provide input to automated control and configuration applications.
>
> It is recognized that it would be beneficial to have a common base model that addresses multiple popular L2VPN service types. The working group derives a single data model that includes support for the following:
>
> * point-to-point Virtual Private Wire Services (VPWS),
>
> * multipoint Virtual Private LAN services (VPLS) that use LDP-signaled Pseudowires,
>
> * multipoint Virtual Private LAN services (VPLS) that use a Border Gateway Protocol (BGP) control plane as described in [RFC4761] and [RFC6624],
>
> * Ethernet VPNs specified in [RFC7432].

Other L2VPN service types may be included if there is consensus in the working group.

Similarly, the L3SM WG specified a sevice model for L3-based VPNs.

The Layer Three Virtual Private Network Service Model (L3SM) working group is a short-lived WG tasked to create a YANG data model that describes a L3VPN service (a L3VPN service model) that can be used for communication between customers and network operators, and to provide input to automated control and configuration applications.

It needs to be clearly understood that this L3VPN service model is not an L3VPN configuration model. That is, it does not provide details for configuring network elements or protocols. Instead it contains the characteristics of the service.

6. Architectural Observations

This section makes some observations about architectural trends and issues.

Role of Software

An obvious trend is that bigger and bigger parts of the functionality in a network is driven by software, e.g., orchestration or management tools that figure out how to control relatively simple network element functionality. The software components are where the intelligence is, and a smaller fraction of the intelligence resides in network elements, nor is the intelligence encoded in the behaviour rules of the protocols that the network elements use to communicate with each other.

Centralization of Functions

An interesting architectural trend is that virtualization and data /software driven networking technologies are driving network architectures where functionality moves towards central entities such as various controllers, path computation servers, and orchestration systems.

A natural consequence of this is the simplification (and perhaps commoditization) of network elements, while the "intelligent" or higher value functions migrate to the center.

The benefits are largely in the manageability, control, and speed of change. There are, however, potential pitfalls to be aware of as well. First off, networks need to continue to be operate even

under partial connectivity situations and breakage, and it is key that designs can handle those situations as well.

And it is important that network users and peers continue to be able to operate and connect in the distributed, voluntary manner that we have today. Today's virtualization technology is primarily used to manage single administrative domains and to offer specific service to others. One could imagine centralised models being taken too far as well, limiting the ability of other network owners to manage their own networks.

Tailored vs. general-purpose networking

The interest in building tailored solutions, tailored Quality-of-Service offerings vs. building general-purpose "low touch" networks seems to fluctuate over time.

It is important to find the right balance here. From an economics perspective, it may not be feasible to provide specialised service -- at least if it requires human effort -- for large fraction of use cases. Even if those are very useful in critical applications.

Need for descriptions

As networks deal more and more with virtual services, there arises a need to have generally understood, portable descriptions of these service. Hence the creation of YANG data models representing abstract VPN services, for instance.

We can also identify some potential architectural principles, such as:

Data model layering

Given the heterogenuity of networking technologies and the differing users that data models are being designed for, it seems difficult to provide a single-level model. It seems preferable to construct a layered set of models, for instance abstract, user-facing models that specify services that can then be mapped to concrete configuration model for networks. And these can in turn be mapped to individual network element configuration models.

Getting this layered design right is crucial for our ability to evolve a useful set of data models.

Ability to evolve modelling tools and mapping systems

The networks and their models are complex, and mapping from high abstraction level specifications to concrete network configurations is a hard problem.

It is important that each of the components can evolve on its own. It should be possible to plug in a new language that represents network models better. Or replace a software component that performs mapping between layers to one that works better.

While this should normally be possible, there's room to avoid too tight binding between the different aspects of a system. For instance, abstraction layers within software can shield the software from being too closely tied with a particular representation language.

Similarly, it would be an advantage to develop algorithms and mapping approaches separately from the software that actually does that, so that another piece of software could easily follow the same guidelines and provide an alternate implementation. Perhaps there's an opportunity for specification work to focus more on processing rules than protocol behaviours, for instance.

General over specific

In the quick pace of important developments, it is tempting to focus on specific concepts and service offerings such as 5G slicing.

But a preferrable approach seems to provide general-purpose tools that can be used by 5G and other networks, and whose longetivity exceeds that of a version of a specific offering. The quick development pace is likely driving the evolution of concepts in any case, and building IETF tools that provide the ability to deal with different technologies is most useful.

7. Further Work

There may be needs for further work in this area at the IETF. Before discussing the specific needs, it may be useful to classify the types of useful work that might come to question. And perhaps also outline some types of work that is not appropriate for the IETF.

The IETF works primarily on protocols, but in many cases also with data models that help manage systems, as well as operational guidance documents. But the IETF does not work on software, such as abstractions that only need to exist inside computers or ones that do not have an effect on protocols either on real or simulated "wires".

The IETF also does not generally work on system-level design. IETF is best at designing components, not putting those components together to achieve a particular purpose or build a specific application.

As a result, IETF's work on new systems employing virtualization techniques (such as 5G slicing concept) is more at the component improvement level than at the level of the concept. There needs to be a mapping between a vision of a system and how it utilizes various software, hardware, and protocol tools to achieve the particular virtualization capabilities it needs to. Developing a new concept does not necessarily mean that entirely new solutions are needed throughtout the stack. Indeed, systems and concepts are usually built on top of solid, well defined components such as the ones produced by the IETF.

That mapping work is necessarily something that those who want to achieve some new functionality need to do; it is difficult for others to take a position on what the new functionality is. But at the same time, IETF working groups and participants typically have a perspective on how their technology should develop and be extended. Those two viewpoints must meet.

The kinds of potential new work in this space falls generally in the following classes:

Virtualization selectors

Sometimes protocols need mechanisms that make it possible to use them as multiple instances. E.g., VLAN tags were added to Ethernet frames, NAIs were added to PPP and EAP, and so on. These cases are rare today, because most protocols and mechanisms have some kind of selector that can be used to run multiple instances or connect to multiple different networks.

Traffic engineering

A big reason for building specific networks for specific purposes is to provide an engineered service level on delay and other factors to the given customer. There are a number of different tools in the IETF to help manage and engineer networks, but it is also an area that continues to develop and will likely see new functionality.

Virtual service data models

Data models -- such as those described by L2SM or L3SM working groups can represent a "service" offered by a network, a setup built for a specific customer or purpose.

Some specific areas where work is likely needed include:

- The ability to manage heterogenous technologies, e.g., across SDN and traditionally built networks, or manage both general-purpose and very technology-specific parameters such as those associated with 5G radio.

- The ability to specify "statistical" rather than hard performance parameters. In some networks -- notably with wireless technology -- recent advances have made very high peak rates possible, but with increased bursty-ness of traffic and with potential bottlenecks on the aggregation parts of the networks. The ability to specify statistical performance in data models and in VPN configuration would be important, over different timescales and probabilities.

- Mapping from high abstraction level specifications to concrete network configurations.

  There is a lot of work on data models and templates at various levels and in different representations. There are also many systems built to manage these models and orchestrate network configuration. But the mapping of the abstract models to concrete network configurations remains a hard problem, and it certainly will need more work.

  There are even some questions about how to go about this. Is it enough that we specify models, and leave the mapping to "magic" of the software? Are the connections something that different vendors compete in producing good products in? Or are the mapping algorithms something that needs to be specified together, and their ability to work with different types of network equipment verified in some manner?

- Cross-domain: A big problem is that we have little tools for cross-domain management of virtualized networks and resources.

Finally, there is a question of where all this work should reside. There’s an argument that IETF-based virtualization technologies deserve proper management tools, including data models.

And there’s another argument that with the extensive use of virtualization technology, solutions that can manage many different networks should be general, and as such, potential IETF work

material. Yet, the IETF is not and should not be in the space of replacing various tools and open source toolkits that have been created for managing virtualization. It seems though that work on commonly usable data models at several layers of abstraction would be good work at the IETF.

Nevertheless, the IETF should understand where the broader community is and what tools they use for what purpose, and try to help by building on those components. Virtualization and slicing are sometimes represented as issues needing a single solution. In reality, they are an interworking of a number of different tools.

8. Acknowledgements

The authors would like to thank Gonzalo Camarillo, Gabriel Montenegro, Alex Galis, Adrian Farrell, Liang Geng, Yi Zhao, Hannu Flinck, Yi Zhao, Barry Leiba, Georg Mayer, Benoit Claise, Daniele Ceccarelli, Warren Kumari, Ted Hardie, and many others for interesting discussions in this problem space.

9. Informative References

[CC2015] claffy, kc. and D. Clark, "Adding Enhanced Services to the Internet: Lessons from History", September 2015 (https://www.caida.org/publications/papers/2015/adding_enhanced_services_internet/adding_enhanced_services_internet.pdf).

[I-D.bryskin-teas-sf-aware-topo-model] Bryskin, I. and X. Liu, "SF Aware TE Topology YANG Model", draft-bryskin-teas-sf-aware-topo-model-01 (work in progress), March 2018.

[I-D.bryskin-teas-use-cases-sf-aware-topo-model] Bryskin, I., Liu, X., Guichard, J., Lee, Y., Contreras, L., and D. Ceccarelli, "Use Cases for SF Aware Topology Models", draft-bryskin-teas-use-cases-sf-aware-topo-model-02 (work in progress), March 2018.

[I-D.geng-coms-problem-statement] 67, 4., Slawomir, S., Qiang, L., Matsushima, S., Galis, A., and L. Contreras, "Problem Statement of Supervised Heterogeneous Network Slicing", draft-geng-coms-problem-statement-00 (work in progress), September 2017.

[I-D.ietf-sfc-nsh]

Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", draft-ietf-sfc-nsh-28 (work in progress), November 2017.

[I-D.king-teas-applicability-actn-slicing]
King, D. and Y. Lee, "Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to Network Slicing", draft-king-teas-applicability-actn-slicing-01 (work in progress), July 2017.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, DOI 10.17487/RFC2616, June 1999, <https://www.rfc-editor.org/info/rfc2616>.

[RFC4026] Andersson, L. and T. Madsen, "Provider Provisioned Virtual Private Network (VPN) Terminology", RFC 4026, DOI 10.17487/RFC4026, March 2005, <https://www.rfc-editor.org/info/rfc4026>.

[RFC4110] Callon, R. and M. Suzuki, "A Framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs)", RFC 4110, DOI 10.17487/RFC4110, July 2005, <https://www.rfc-editor.org/info/rfc4110>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, DOI 10.17487/RFC4664, September 2006, <https://www.rfc-editor.org/info/rfc4664>.

[RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <https://www.rfc-editor.org/info/rfc4761>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6624] Kompella, K., Kothari, B., and R. Cherukuri, "Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling", RFC 6624, DOI 10.17487/RFC6624, May 2012, <https://www.rfc-editor.org/info/rfc6624>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <https://www.rfc-editor.org/info/rfc7432>.

[RFC8049] Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8049, DOI 10.17487/RFC8049, February 2017, <https://www.rfc-editor.org/info/rfc8049>.

[TOSCA-1.0] OASIS, "Topology and Orchestration Specification for Cloud Applications Version 1.0", OASIS OASIS Standard, http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html, November 2013.

[TOSCA-Profile-1.1] OASIS, "TOSCA Simple Profile in YAML Version 1.1", OASIS OASIS Standard, http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.1/TOSCA-Simple-Profile-YAML-v1.1.html, January 2018.

[TS-3GPP.23.401] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access; (Release 15)", 3GPP Technical Specification 23.401, December 2017.

[TS-3GPP.23.501] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security architecture and procedures for 5G System; (Release 15)", 3GPP Technical Specification 23.501, December 2017.

[VirtualHosting] Wikipedia, "Virtual Hosting", Wikipedia article https://en.wikipedia.org/wiki/Virtual_hosting, August 2017.

Authors' Addresses

Jari Arkko
Ericsson
Kauniainen 02700
Finland

Email: jari.arkko@piuha.net

Jeff Tantsura
Nuagenetworks

Email: jefftant.ietf@gmail.com

Joel Halpern
Ericsson

Email: joel.halpern@ericsson.com

Balazs Varga
Ericsson
Budapest 1097
Hungary

Email: balazs.a.varga@ericsson.com

Routing Area Working Group
Internet-Draft
Intended status: Informational
Expires: September 6, 2018

S. Bryant
J. Dong
Huawei
Z. Li
China Mobile
T. Miyasaka
KDDI Corporation

March 05, 2018

Enhanced Virtual Private Networks (VPN+)
draft-bryant-rtgwg-enhanced-vpn-02

Abstract

This draft describes a number of enhancements that need to be made to virtual private networks (VPNs) to support the needs of new applications, particularly applications that are associated with 5G services. A network enhanced with these properties may form the underpin of network slicing, but will also be of use in its own right.

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 6, 2018.

Copyright Notice

Table of Contents

1. Introduction

Virtual networks, often referred to as virtual private networks (VPNs) have served the industry well as a means of providing different groups of users with logically isolated access to a common network. The common or base network that is used to provide the VPNs

is often referred to as the underlay, and the VPN is often called an overlay.

Driven largely by needs surfacing from 5G, the concept of network slicing has gained traction. There is a need to create a VPN with enhanced characteristics. Specifically there is a need for a transport network supporting a set of virtual networks each of which provides the client with dedicated (private) networking, computing and storage resources drawn from a shared pool.
The tenant of such a network can require a degree of isolation and performance that previously could only be satisfied by dedicated networks. Additionally the tenant may ask for some level of control of their virtual network e.g. to customize the service paths in the network slice.

These properties cannot be met with pure overlay networks, as they require tighter coordination and integration between the underlay and the overlay network. This document introduces a new network service called enhanced VPN (VPN+). VPN+ refers to a virtual network which has dedicated network resources allocated from the underlay network. Unlike traditional VPN, an enhanced VPN can achieve greater isolation and guaranteed performance.

These new network layer properties, which have general applicability, may also be of interest as part of a network slicing solution.

This document specifies a framework for using the existing, modified and potential new networking technologies as components to provide an enhanced VPN (VPN+) service. Specifically we are concerned with:

- The design of the enhanced VPN data-plane
- The necessary protocols in both, underlay and the overlay of enhanced VPN, and
- The mechanisms to achieve integration between overlay and underlay
- The necessary method of monitoring an enhanced VPN
- The methods of instrumenting an enhanced VPN to ensure that the required tenant Service Level Agreement (SLA) is maintained

The required layer structure necessary to achieve this is shown in Section 4.1.

One use for enhanced VPNs is to create network slices with different isolation requirements. Such slices may be used to provide different tenants of vertical industrial markets with their own virtual network

with the explicit characteristics required. These slices may be "hard" slices providing a high degree of confidence that the VPN+ characteristics will be maintained over the slice life cycle, of they may be "soft" slices in which case some degree of interaction may be experienced.

2. 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 [RFC2119].

3. Overview of the Requirements

In this section we provide an overview of the requirements of an enhanced VPN.

3.1. Isolation between Virtual Networks

The requirement is to provide both hard and soft isolation between the tenants/applications using one enhanced VPN and the tenants/ applications using another enhanced VPN. Hard isolation is needed so that applications with exacting requirements can function correctly despite a flash demand being created on another VPN competing for the underlying resources. An example might be a network supporting both emergency services and public broadband multi-media services.

During a major incident the VPNs supporting these services would both be expected to experience high data volumes, and it is important that both make progress in the transmission of their data. In these circumstances the VPNs would require an appropriate degree of isolation to be able to continue to operate acceptably.

We introduce the terms hard (static) and soft (dynamic) isolation to cover cases such as the above. A VPN has soft isolation if the traffic of one VPN cannot be inspected by the traffic of another. Both IP and MPLS VPNs are examples of soft isolated VPNs because the network delivers the traffic only to the required VPN endpoints. However the traffic from one or more VPNs and regular network traffic may congest the network resulting in delays for other VPNs operating normally. The ability for a VPN to be sheltered from this effect is called hard isolation, and this property is required by some critical applications. Although these isolation requirements are triggered by the needs of 5G networks, they have general utility. In the remainder of this section we explore how isolation may be achieved in packet networks.

It is of course possible to achieve high degrees of isolation in the optical layer. However this is done at the cost of allocating resources on a long term basis and end-to-end basis. Such an arrangement means that the full cost of the resources must be borne by the service that is allocated the resources. On the other hand, isolation at the packet layer allows the resources to be shared amongst many services and only dedicated to a service on a temporary basis. This allows greater statistical multiplexing of network resources and amortizes the cost over many services, leading to better economy. However, the degree of isolation required by network slicing cannot easily be met with MPLS-TE packet LSPs as they guarantee long-term bandwidth, but not latency.

Thus some trade-off between the two approaches needs to be considered to provide the required isolation between virtual networks while still allows reasonable sharing inside each VPN.

The work of the IEEE project on Time Sensitive Networking is introducing the concept of packet scheduling where a high priority packet stream may be given a scheduled time slot thereby guaranteeing that it experiences no queuing delay and hence a reduced latency. However where no scheduled packet arrives its reserved time-slot is handed over to best effort traffic, thereby improving the economics of the network. Such a scheduling mechanism may be usable directly, or with extension to achieve isolation between multiple VPNs.

One of the key areas in which isolation needs to be provided is at the interfaces. If nothing is done the system falls back to the router queuing system in which the ingress places it on a selected output queue. Modern routers have quite sophisticated output queuing systems, traditionally these have not provided the type of scheduling system needed to support the levels of isolation needed for the applications that are the target of VPN+ networks. However some of the more modern approaches to queuing allow the construction of logical virtual channelized sub-interfaces (VCSI). With VCSIs there is only one physical interface, and routing sees a single adjacency, but the queuing system is used to provide virtual interfaces at various priorities. Sophisticated queuing systems of this type may be used to provide end-to-end virtual isolation between tenant’s traffic in an otherwise homogeneous network.

[FLEXE] provides the ability to multiplex multiple channels over an Ethernet link in a way that provides hard isolation. However it is a only a link technology. When packets are received by the downstream node they need to be processed in a way that preserves that isolation. This in turn requires a queuing and forwarding implementation that preserves the isolation, such as a sliced hardware system, or an LVI system of the type described above.

3.2. Diverse Performance Guarantees

There are several aspects to guaranteed performance, guaranteed maximum packet loss, guaranteed maximum delay and guaranteed delay variation.

Guaranteed maximum packet loss is a common parameter, and is usually addressed by setting the packet priorities, queue size and discard policy. However this becomes more difficult when the requirement is combine with the latency requirement. The limiting case is zero congestion loss, and than is the goal of the Deterministic Networking work that the IETF and IEEE are pursuing. In modern optical networks loss due to transmission errors is already asymptotic to zero due, but there is always the possibility of failure of the interface and the fiber itself. This can only be addressed by some form of packet duplication and transmission over diverse paths.

Guaranteed maximum latency is required in a number of applications particularly real-time control applications and some types of virtual reality applications. The work of the IETF Deterministic Networking (DetNet) Working Group is relevant, however the scope needs to be extended to methods of enhancing the underlay to better support the delay guarantee, and to integrate these enhancements with the overall service provision.

Guaranteed maximum delay variation is a service that may also be needed. Time transfer is one example of a service that needs this, although the fungible nature of time means that it might be delivered by the underlay as a shared service and not provided through different virtual networks. Alternatively a dedicated virtual network may be used to provide this as a shared service. The need for guaranteed maximum delay variation as a general requirement is for further study.

This leads to the concept that there is a spectrum of grades of service guarantee that need to be considered when deploying and enhanced VPN. As a guide to understanding the design requirements we can consider four types:

- Guaranteed latency,
- Enhanced delivery
- Assured bandwidth,
- Best effort

In Section 3.1 we considered the work of the IEEE Time Sensitive Networking (TSN) project and the work of the IETF DetNet Working group in the context of isolation. However this work is of greater relevance in assuring end-to-end packet latency. It is also of importance in considering enhanced delivery.

A service that is guaranteed latency has a latency upper bound provided by the network. It is important to note that assuring the upper bound is more important than achieving the minimum latency.

A service that is offered enhanced delivery is one in which the network (at layer 3) attempts to deliver the packet through multiple paths in the hope of avoiding transient congestion [I-D.ietf-detnet-dp-sol].

A useful mechanism to provide these guarantees is to use Flex Ethernet [FLEXE] as the underlay. This is a method of bonding Ethernets together and of providing time-slot based channelization over an Ethernet bearer. Such channels are fully isolated from other channels running over the same Ethernet bearer. As noted elsewhere this produces hard isolation but at the cost of making the reclamation of unused bandwidth harder.

These approaches can usefully be used in tandem. It is possible to use FlexE to provide tenant isolation, and then to use the TSN approach over FlexE to provide service performance guarantee inside the a slice/tenant VPN.

3.3. A Pragmatic Approach to Isolation

A key question to consider is whether whether it is possible to achieve hard isolation in packet networks? Packet networks were never designed to support hard isolation, just the opposite, they were designed to provide a high degree of statistical multiplexing and hence a significant economic advantage when compared to a dedicated, or a Time Division Multiplexing (TDM) network. However the key thing to bear in mind is that the concept of hard isolation needs to be viewed from the perspective of the application, and there is no need to provide any harder isolation than is required by the application. From a historical perspective it is good to think about pseudowires [RFC3985] which emulate services that in many would have had hard isolation in their native form. However experience has shown that in most cases an approximation to this requirement is sufficient for most uses.

Thus, for example, using FlexE or channelized sub-interface,together with packet scheduling as interface slicing, and optionally, also together with the slicing of node resources (Network Processor Unit

(NPU), etc.), it may be possible to provide a type of hard isolation that is adequate for many applications. Other applications may be satisfied with a classical VPN and reserved bandwidth, but yet others may require dedicated point to point fiber. The requirement is thus to qualify the needs of each application and provide an economic solution that satisfies those needs without over-engineering.

3.4. Integration

A solution to the enhanced VPN problem will need to provide seamless integration of both Overlay VPN and the underlay network resources. This needs be done in a flexible and scalable way so that it can be widely deployed in operator networks. Given the targeting of both this technology and service function chaining at mobile networks and in particular 5G the co-integration of service functions is a likely requirement.

3.5. Dynamic Configuration

It is necessary that new enhanced VPNs can be introduced to the network, modified, and removed from the network according to service demand. In doing so due regard must be given to the impact of other enhanced VPNs that are operational. An enhanced VPN that requires hard isolation must not be disrupted by the installation or modification of another enhanced VPN.

Whether modification of an enhanced VPN can be disruptive to that VPN, and in particular the traffic in flight is to be determined, but is likely to be a difficult problem to address.

The data-plane aspect of this are discussed further in Section 4.3.

The control-plane and management-plane aspects of this, particularly the garbage collection are likely to be challenging and are for further study.

As well as managing dynamic changes to the VPN in a seamless way, dynamic changes to the underlay and its transport network need to be managed in order to avoid disruption to sensitive services.

In addition to non-disruptively managing the network as a result of gross change such as the inclusion of a new VPN endpoint or a change to a link, consideration has to be given to the need to move VPN traffic as a result of traffic volume changes.

3.6. Customized Control Plane

In some cases it is desirable that an enhanced VPN has a custom control-plane, so that the tenant of the enhanced VPN can have some control to the resources and functions partitioned for this VPN. Each enhanced VPN may have its own dedicated controller, it may be provided with an interface to a control-plane that is shared with a set of other tenants, or it may be provided with an interface to the control-plane of the underlay provided by the underlay network operator.

Further detail on this requirement will be provided in a future version of the draft.

4. Architecture and Components of VPN+

Normally a number of enhanced VPN services will be provided by a common network infrastructure. Each enhanced VPN consists of both the overlay and a specific set of dedicated network resources and functions allocated in the underlay to satisfy the needs of the VPN tenant. The integration between overlay and underlay ensures the isolation and between different enhanced VPNs, and facilitates the guaranteed performance for different services.

An enhanced VPN needs to be designed with consideration given to:

- Isolation of enhanced VPN data plane.
- A scalable control plane to match the data plane isolation.
- The amount of state in the packet vs the amount of state in the control plane.
- Mechanism for diverse performance guarantee within an enhanced VPN
- Support of the required integration between network functions and service functions.

4.1. Communications Layering

The communications layering model use to build an enhanced VPN is shown in Figure 1.

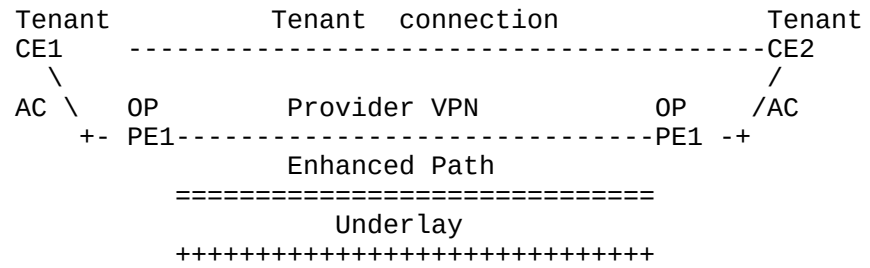


Figure 1: Communication Layering

The network operator is required to provide a tenant connection between the tenant's Customer Equipment (CE) (CE1 and CE2). These CEs attach to the Operator's Provider Edge Equipments (PE) (PE1 and PE2 respectively). The attachment circuits (AC) are outside the scope of this document other than to note that they obviously need to provide a connection of sufficient quality in terms of isolation, latency etc so as to satisfy the needs of the user. The subtlety to be aware of is that the ACs are often provided by a network rather than a fixed point to point connection and thus the considerations in this document may apply to the network that provides the AC.

A provider VPN is constructed between PE1 and PE2 to carry tenant traffic. This is a normal VPN, and provides one stage of isolation between tenants.

An enhanced path is constructed to carry the provider VPN using dedicated resources drawn from the underlay.

4.2. Multi-Point to Multi-point

At a VPN level connections are frequently multi-point-to-multi-point (MP2MP). As far as such services are concerned the underlay is also an abstract MP2MP medium. However when service guarantees are provided, such as with an enhanced VPN, each point to point path through the underlay needs to be specifically engineered to meet the required performance guarantees.

4.3. Candidate Underlay Technologies

A VPN is a network created by applying a multiplexing technique to the underlying network (the underlay) in order to distinguish the traffic of one VPN from that of another. A VPN path that travels by other than the shortest path through the underlay normally requires state in the underlay to specify that path. State is normally applied to the underlay through the use of the RSVP Signaling protocol, or directly through the use of an SDN controller, although

other techniques may emerge as this problem is studied. This state gets harder to manage as the number of VPN paths increases. Furthermore, as we increase the coupling between the underlay and the overlay to support the VPN which requires enhanced VPN service, this state will increase further.

In an enhanced VPN different subsets of the underlay resources are dedicated to different VPNs. Any enhanced VPN solution thus needs tighter coupling with underlay than is the case with classical VPNs. We cannot for example share the tunnel between enhanced VPNs which require hard isolation.

In the following sections we consider a number of candidate underlay solutions for proving the required VPN separation.

- FlexE
- Time Sensitive Networking
- Deterministic Networking
- Dedicated Queues

We then consider the problem of slice differentiation and resource representation. Candidate technologies are:

- MPLS
- MPLS-SR
- Segment Routing over IPv6 (SRv6)

4.3.1. FlexE

FlexE [FLEXE] is a method of creating a point-to-point Ethernet with a specific fixed bandwidth. FlexE supports the bonding of multiple links, which supports creating larger links out of multiple slower links in a more efficient way that traditional link aggregation. FlexE also supports the sub-rating of links, which allows an operator to only use a portion of a link. FlexE also supports the channelization of links, which allows one link to carry several lower-speed or sub-rated links from different sources.

If different FlexE channels are used for different services, then no sharing is possible between the services. This in turn means that it is not possible to dynamically re-distribute unused bandwidth to lower priority services increasing the cost of operation of the network. FlexE can on the other hand be used to provide hard

isolation between different tenants by providing hard isolation on an interface. The tenant can then use other methods to manage the relative priority of their own traffic.

Methods of dynamically re-sizing FlexE channels and the implication for enhanced VPN are under study.

4.3.2. Dedicated Queues

In an enhanced VPN providing multiple isolated virtual networks the conventional Diff-Serv based queuing system is insufficient for our purposes due to the limited number of queues which cannot differentiate between traffic of different VPNs and the range of service classes that each need to provide their tenants. This problem is particularly acute with an MPLS underlay due to the small number of traffic class services available. In order to address this problem and thus reduce the interference between VPNs, it is likely to be necessary to steer traffic of VPNs to dedicated input and output queues.

4.3.3. Time Sensitive Networking

Time Sensitive Networking (TSN) is an IEEE project that is designing a method of carrying time sensitive information over Ethernet. As Ethernet this can obviously be tunneled over a Layer 3 network in a pseudowire. However the TSN payload would be opaque to the underlay and thus not treated specifically as time sensitive data. The preferred method of carrying TSN over a layer 3 network is through the use of deterministic networking as explained in the following section of this document.

The machanisms defined in TSN can be used to meet the requirements of time sensitive services of an enhanced VPN.

4.3.4. Deterministic Networking

Deterministic Networking (DetNet) [I-D.ietf-detnet-architecture] is a technique being developed in the IETF to enhance the ability of layer 3 networks to deliver packets more reliably and with greater control over the delay. The design cannot use classical re-transmission techniques such as TCP since can add delay that is above the maximum tolerated by the applications. Even the delay improvements that are achieved with SCTP-PR are outside the bounds set by application demands. The approach is to pre-emptively send copies of the packet over various paths in the expectation that this minimizes the chance of all packets being lost, but to trim duplicate packets to prevent excessive flooding of the network and to prevent multiple packets being delivered to the destination. It also seeks to set an upper

bound on latency. Note that it is not the goal to minimize latency, and the optimum upper bound paths may not be the minimum latency paths.

DetNet is based on flows. It currently makes no comment on the underlay, and so at this stage must be assumed to use the base topology. To be of use in this application DetNet there needs to be a description of how to deal with the concept of flows within an enhanced VPN.

How we use DetNet in a multi-tenant (VPN) network, and how to improve the scalability of DetNet in a multi-tenant (VPN) network is for further study.

4.3.5. MPLS Traffic Engineering (MPLS-TE)

Normal MPLS runs on the base topology and has the concepts of reserving end to end bandwidth for an LSP, and of creating VPNs. VPN traffic can be run over RSVP-TE tunnels to provide reserved bandwidth for a specific VPN connection. This is rarely deployed in practice due to scaling and management overhead concerns.

4.3.6. Segment Routing

Segment Routing [I-D.ietf-spring-segment-routing] is a method that prepends instructions to packets at entry and sometimes at various points as it passes though the network. These instructions allow packets to be routed on paths other than the shortest path for various traffic engineering reasons. These paths can be strict or loose paths, depending on the compactness required of the instruction list and the degree of autonomy granted to the network (for example to support ECMP).

With SR, a path needs to be dynamically created through a set of resources by simply specifying the Segment IDs (SIDs), i.e. instructions rooted at a particular point in the network. Thus if a path is to be provisioned from some ingress point A to some egress point B in the underlay, A is provided with the A..B SID list and instructions on how to identify the packets to which the SID list is to be prepended.

By encoding the state in the packet, as is done in Segment Routing, state is transitioned out of the network.

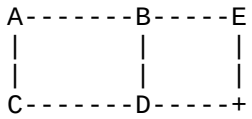


Figure 2: An SR Network Fragment

Consider the network fragment shown in Figure 2. To send a packet from A to E via B, D & E: Node A prepends the ordered list of SIDs:D, E to the packet and pushes the packet to B. SID list {B, D, E} can be used as a VPN path. Thus, to create a VPN, a set of SID Lists is created and provided to each ingress node of the VPN together with packet selection criteria. In this way it is possible to create a VPN with no state in the core. However this is at the expense of creating a larger packet with possible MTU and hardware restriction limits that need to be overcome.

Note in the above if A and E support multiple VPN an additional VPN identifier will need to be added to the packet, but this is omitted from this text for simplicity.

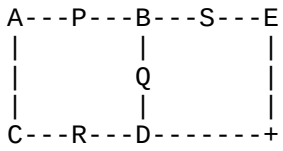


Figure 3: Another SR Network Fragment

Consider a further network fragment shown in Figure 3, and further consider VPN A+D+E.

A has lists: {P, B, Q, D}, {P, B, S, E}
D has lists: {Q, B, P, A}, {E}
E has lists: {S, B, P, A}, {D}

To create a new VPN C+D+B the following list are introduced:

C lists: {R, D}, {A, P, B}
D lists: {R, C}, {Q, B}
B lists: {Q, D}, {P, A, C}

Thus VPN C+D+B was created without touching the settings of the core routers, indeed it is possible to add endpoints to the VPNs, and move the paths around simply by providing new lists to the affected endpoints.

There are a number of limitations in SR as it is currently defined that limit its applicability to enhanced VPNs:

- Segments are shared between different VPNs,
- There is no reservation of bandwidth,
- There is limited differentiation in the data plane.

Thus some extensions to SR are needed to provide isolation between different enhanced VPNs. This can be achieved by including a finer granularity of state in the core in anticipation of its future use by authorized services. We therefore need to evaluate the balance between this additional state and the performance delivered by the network.

Both MPLS Segment Routing and SRv6 Segment Routing are candidate technologies for enhanced VPN.

With current segment routing, the instructions are used to specify the nodes and links to be traversed. However, in order to achieve the required isolation between different services, new instructions can be created which can be prepended to a packet to steer it through specific dedicated network resources and functions, e.g. links, queues, processors, services etc.

Clearly we can use traditional constructs to create a VPN, but there are advantages to the use of other constructs such as Segment Routing (SR) in the creation of virtual networks with enhanced properties.

Traditionally a traffic engineered path operates with a granularity of a link with hints about priority provided through the use of the traffic class field in the header. However to achieve the latency and isolation characteristics that are sought by VPN+ users, steering packets through specific queues resources will likely be required. The extent to which these needs can be satisfied through existing QoS mechanisms is to be determined. What is clear is that a fine control of which services wait for which, with a fine granularity of queue management policy is needed. Note that the concept of a queue is a useful abstraction for many types of underlay mechanism that may be used to provide enhanced latency support. From the perspective of the control plane and from the perspective of the segment routing the method of steering a packet to a queue that provides the required properties is a universal construct. How the queue satisfies the requirement is outside the scope of these aspect of the enhanced VPN system. Thus for example a FlexE channel, or time sensitive networking packet scheduling slot are abstracted to the same concept and bound to the data plane in a common manner.

We can introduce the specification of finer, deterministic, granularity to path selection through extensions to traditional path construction techniques such as RSVP-TE and MPLS-TP.

We can also introduce it by specifying the queue through an SR instruction list. Thus new SR instructions may be created to specify not only which resources are traversed, but in some cases how they are traversed. For example, it may be possible to specify not only the queue to be used but the policy to be applied when enqueuing and dequeuing.

This concept can be further generalized, since as well as queuing to the output port of a router, it is possible to queue to any resource, for example:

- A network processor unit (NPU)
- A Central Processing Unit (CPU) Core
- A Look-up engine such as TCAMs

4.4. Control Plane Considerations

It is expected that VPN+ would be based on a hybrid control mechanism, which takes advantage of the logically centralized controller for on-demand provisioning and global optimization, whilst still relies on distributed control plane to provide scalability, high reliability, fast reaction, automatic failure recovery etc. Extension and optimization to the distributed control plane is needed to support the enhanced properties of VPN+.

Where SR is used as a the data-plane construct it needs to be noted that it does not have the capability of reserving resources along the path nor do its currently specified distributed control plane (the link state routing protocols). An SDN controller can clearly do this, from the controllers point of view, and no resource reservation is done on the device. Thus if a distributed control plane is needed either in place of an SDN controller or as an assistant to it, the design of the control system needs to ensure that resources are uniquely allocated to the correct service, and no allocated to multiple services casing unintended resource conflict. This needs further study.

On the other hand an advantage of using an SR approach is that it provides a way of efficiently binding the network underlay and the enhanced VPN overlay. With a technology such as RSVP-TE LSPs, each virtual path in the VPN is bound to the underlay with a dedicated TE-LSP.

RSVP-TE could be enhanced to bind the VPN to specific resources within the underlay, but as noted elsewhere in this document there are concerns as to the scalability of this approach. With an SR-based approach to resource reservation (per-slice reservation), it is straightforward to create dedicated SR network slices, and the VPN can be bound to a particular SR network slice.

4.5. Application Specific Network Types

Although a lot of the traffic that will be carried over the enhanced VPN will likely be IPv4 or IPv6, the design has to be capable of carrying other traffic types. In particular the design SHOULD be capable of carrying Ethernet traffic. This is easily accomplished through the various pseudowire (PW) techniques [RFC3985]. Where the underlay is MPLS Ethernet can be carried over the enhanced VPN encapsulated according to the method specified in [RFC4448]. Where the underlay is IP Layer Two Tunneling Protocol - Version 3 (L2TPv3) [RFC3931] can be used with Ethernet traffic carried according to [RFC4719]. Encapsulations have been defined for most of the common layer two type for both PW over MPLS and for L2TPv3.

4.6. Integration with Service Functions

There is a significant overlap between the problem of routing a packet though a set of network resources and the problem of routing a packet through a set of compute resources. Service Function Chain technology is designed to forward a packet through a set of compute resources.

A future version of this document will discuss this further.

5. Scalability Considerations

For a packet to transit a network, other than on a best effort, shortest path basis, it is necessary to introduce additional state, either in the packet, or in the network of some combination of both.

There are at least three ways of doing this:

- Introduce the complete state into the packet. That is how SR does this, and this allows the controller to specify the precise series of forwarding and processing instructions that will happen to the packet as it transits the network. The cost of this is an increase in the packet header size. The cost is also that systems will have capabilities enabled in case they are called upon by a service. This is a type of latent state, and increases as we more precisely specify the path and resources that need to be exclusively available to a VPN.

- o Introduce the state to the network. This is normally done by creating a path using RSVP-TE, which can be extended to introduce any element that needs to be specified along the path, for example explicitly specifying queuing policy. It is of course possible to use other methods to introduce path state, such as via a Software Defined Network (SDN) controller, or possibly by modifying a routing protocol. With this approach there is state per path per path characteristic that needs to be maintained over its life-cycle. This is more state than is needed using SR, but the packet are shorter.

- o Provide a hybrid approach based on using binding SIDs to create path fragments, and bind them together with SR.

Dynamic creation of a VPN path using SR requires less state maintenance in the network core at the expense of larger VPN headers on the packet. The scaling properties will reduce roughly from a function of (N/2)^2 to a function of N, where N is the VPN path length in intervention points (hops plus network functions). Reducing the state in the network is important to VPN+, as VPN+ requires the overlay to be more closely integrated with the underlay than with traditional VPNs. This tighter coupling would normally mean that significant state needed to be created and maintained in the core. However, a segment routed approach allows much of this state to be spread amongst the network ingress nodes, and transiently carried in the packets as SIDs.

These approaches are for further study.

5.1. Maximum Stack Depth

One of the challenges with SR is the stack depth that nodes are able to impose on packets. This leads to a difficult balance between adding state to the network and minimizing stack depth, or minimizing state and increasing the stack depth.

5.2. RSVP scalability

The traditional method of creating a resource allocated path through an MPLS network is to use the RSVP protocol. However there have been concerns that this requires significant continuous state maintenance in the network. There are ongoing works to improve the scalability of RSVP-TE LSPs in the control plane [I-D.ietf-teas-rsvp-te-scaling-rec]. This will be considered further in a future version of this document.

There is also concern at the scalability of the forwarder footprint of RSVP as the number of paths through an LSR grows

[I-D.sitaraman-mpls-rsvp-shared-labels] proposes to address this by employing SR within a tunnel established by RSVP-TE. This work will be considered in a future version of this document.

6. OAM and Instrumentation

A study of OAM in SR networks has been documented in [I-D.ietf-spring-oam-usecase].

The enhanced VPN OAM design needs to consider the following requirements:

- Instrumentation of the underlay so that the network operator can be sure that the resources committed to a tenant are operating correctly and delivering the required performance.

- Instrumentation of the overlay by the tenant. This is likely to be transparent to the network operator and to use existing methods. Particular consideration needs to be given to the need to verify the isolation and the various committed performance characteristics.

- Instrumentation of the overlay by the network provider to proactively demonstrate that the committed performance is being delivered. This needs to be done in a non-intrusive manner, particularly when the tenant is deploying a performance sensitive application

- Verification of the conformity of the path to the service requirement. This may need to be done as part of a commissioning test.

These issues will be discussed in a future version of this document.

7. Enhanced Resiliency

Each enhanced VPN, of necessity, has a life-cycle, and needs modification during deployment as the needs of its user change. Additionally as the network as a whole evolves there will need to be garbage collection performed to consolidate resources into usable quanta.

Systems in which the path is imposed such as SR, or some form of explicit routing tend to do well in these applications because it is possible to perform an atomic transition from one path to another. However implementations and the monitoring protocols need to make sure that the new path is up before traffic is transitioned to it.

There are however two manifestations of the latency problem that are for further study in any of these approaches:

- The problem of packets overtaking one and other if a path latency reduces during a transition.

- The problem of the latency transient in either direction as a path migrates.

There is also the matter of what happens during failure in the underlay infrastructure. Fast reroute is one approach, but that still produces a transient loss with a normal goal of rectifying this within 50ms. An alternative is some form of N+1 delivery such as has been used for many years to support protection from service disruption. This may be taken to a different level using the techniques proposed by the IETF deterministic network work with multiple in-network replication and the culling of later packets.

In addition to the approach used to protect high priority packets, consideration has to be given to the impact of best effort traffic on the high priority packets during a transient. Specifically if a conventional re-convergence process is used there will inevitably be micro-loops and whilst some form of explicit routing will protect the high priority traffic, lower priority traffic on best effort shortest paths will micro-loop without the use of a loop prevention technology. To provide the highest quality of service to high priority traffic, either this traffic must be shielded from the micro-loops, or micro-loops must be prevented.

8. Security Considerations

All types of virtual network require special consideration to be given to the isolation between the tenants. However in an enhanced virtual network service hard isolation needs to be considered. If a service requires a specific latency then it can be damaged by simply delaying the packet through the activities of another tenant. In a network with virtual functions, depriving a function used by another tenant of compute resources can be just as damaging as delaying transmission of a packet in the network.

9. IANA Considerations

There are no requested IANA actions.

10. References

10.1. Normative 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>.

10.2. Informative References

[FLEXE] "Flex Ethernet Implementation Agreement", March 2016, <http://www.oiforum.com/wp-content/uploads/OIF-FLEXE-01.0.pdf>.

[I-D.ietf-detnet-architecture] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-04 (work in progress), October 2017.

[I-D.ietf-detnet-dp-sol] Korhonen, J., Andersson, L., Jiang, Y., Finn, N., Varga, B., Farkas, J., Bernardos, C., Mizrahi, T., and L. Berger, "DetNet Data Plane Encapsulation", draft-ietf-detnet-dp-sol-01 (work in progress), January 2018.

[I-D.ietf-spring-oam-usecase] Geib, R., Filsfils, C., Pignataro, C., and N. Kumar, "A Scalable and Topology-Aware MPLS Dataplane Monitoring System", draft-ietf-spring-oam-usecase-10 (work in progress), December 2017.

[I-D.ietf-spring-segment-routing] Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-15 (work in progress), January 2018.

[I-D.ietf-teas-rsvp-te-scaling-rec] Beeram, V., Minei, I., Shakir, R., Pacella, D., and T. Saad, "Techniques to Improve the Scalability of RSVP Traffic Engineering Deployments", draft-ietf-teas-rsvp-te-scaling-rec-09 (work in progress), February 2018.

[I-D.sitaraman-mpls-rsvp-shared-labels]
Sitaraman, H., Beeram, V., Parikh, T., and T. Saad, "Signaling RSVP-TE tunnels on a shared MPLS forwarding plane", draft-sitaraman-mpls-rsvp-shared-labels-03 (work in progress), December 2017.

[NETCALC] "Applicability of Network Calculus to DetNet", November 2017, <https://datatracker.ietf.org/meeting/100/materials/slides-100-detnet-applicability-of-network-calculus-to-detnet>.

[RFC3931] Lau, J., Ed., Townsley, M., Ed., and I. Goyret, Ed., "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, DOI 10.17487/RFC3931, March 2005, <https://www.rfc-editor.org/info/rfc3931>.

[RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <https://www.rfc-editor.org/info/rfc3985>.

[RFC4448] Martini, L., Ed., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, DOI 10.17487/RFC4448, April 2006, <https://www.rfc-editor.org/info/rfc4448>.

[RFC4719] Aggarwal, R., Ed., Townsley, M., Ed., and M. Dos Santos, Ed., "Transport of Ethernet Frames over Layer 2 Tunneling Protocol Version 3 (L2TPv3)", RFC 4719, DOI 10.17487/RFC4719, November 2006, <https://www.rfc-editor.org/info/rfc4719>.

Authors' Addresses

Stewart Bryant
Huawei

Email: stewart.bryant@gmail.com

Jie Dong
Huawei

Email: jie.dong@huawei.com

Zhenqiang Li
China Mobile

Email: lizhenqiang@chinamobile.com

Takuya Miyasaka
KDDI Corporation

Email: ta-miyasaka@kddi.com

rtgwg
Internet-Draft
Intended status: Informational
Expires: August 30, 2018

R. Gu
S. Hu
China Mobile
Michael. Wang
Huawei
Fangwei. Hu
ZTE Corporation

February 26, 2018

Deployment Model of Control Plane and User Plane Separated BNG
draft-cuspdt-rtgwg-cu-separation-bng-deployment-01

Abstract

This document introduces deployment model of BNG device with Control Plane and User Plane separation in order to give guidance of the deployment of CP and UP separated BNG devices in operators' network.

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 August 30, 2018.

Table of Contents

1. Introduction and Overview

BNG is an Ethernet-centric IP edge router and acts as the aggregation point for the user traffic with some additional functions such as address management and Radius cooperating with Radius systems and subscriber management. According to the rapid development of new services, such as 4K, IoT, etc. and the increasing numbers of distributed home broadband service users, high resource utilization, management of high-efficiency, and fast service provision are required. Thus a new architecture of BNG with CP and UP separation which is also called Cloud BNG is proposed in [BBF-CloudCO][TR-384].

The brandnew CP and UP separation architecture of BNG is constructed by Control Plane and User Plane, with concentrated CP responsible for control and management of UP's resrouces and subscribers' informaiton while distributed UP taking charge of policy implementation and traffic forwarding. The obvious advantages of this new architecture can be listed as belows.

Resource Utilization Improvement: Centralized Control Plane provides the capability of unified management of network resources and users information. CP has an overview of all the resources and can distribute resources as specific user required, thus resources can be totally controlled and balanced.

Management with High Effiency: Centralized CP provides unified management interface to the outside systems such as EMS, DHCP Server, Radius Server and etc. In this situation, management can be much easier for centralized CP as it's the only device facing to the outside systems.

Dynamic and Flexibility: CP can be virtualized as a VNF with MANO management in NFV, while UP can be a virtual machine or physical device as demand. Software-oriented CP can be designed with flexibility. CP can handle all the situations dynamically such as few users accessing and large numbers of users accessing.

Fast TTM: CP and UP can be deployed separately with CP deployed centrally and UP deployed in distribution closing to users. Thus according to different situations such as session overload or extremely high throughput, CP and UP can be extended separately as well. It can help shorten the time to marketing (TTM).

As noted that the new architecture of BNG consists with CP and UP separation, CP and UP are deployed due to practical requirements. This document gives out CU separation BNG deployment model according to the actual deployment.

2. Concept and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2.1. Terminology

BNG: Broadband Network Gateway. A broadband remote access server (BRAS, B-RAS or BBRAS) routes traffic to and from broadband remote access devices such as digital subscriber line access multiplexers (DSLAM) on an Internet service provider’s (ISP) network. BRAS can also be referred to as a Broadband Network Gateway (BNG).

CP: Control Plane. CP is a user control management component which support to manage UP’s resources such as the user entry and user’s QoS policy

UP: User Plane. UP is a network edge and user policy implementation component. The traditional router’s Control Plane and forwarding plane are both preserved on BNG devices in the form of a user plane.

TTM: Time to Market. It is the length of time it takes from a product or a service being conceived until its being available for sale.

MANO: Management and Orchestration. Functions are collectively provided by NFVO, VNFM and VIM.

VNF: Virtual Network Function. Implementation of a Network Function that can be deployed on a Network Function Virtualization Infrastructure (NFVI).

PNF: Physical Network Function

DHCP: Dynamic Host Configuration Protocol

PPPoE: Point to Point Protocol over Ethernet

IPoE: Internet Protocol over Ethernet

3. Deployment Model of BNG with CP and UP Separation

3.1. CP and UP of BNG deployment within only one district

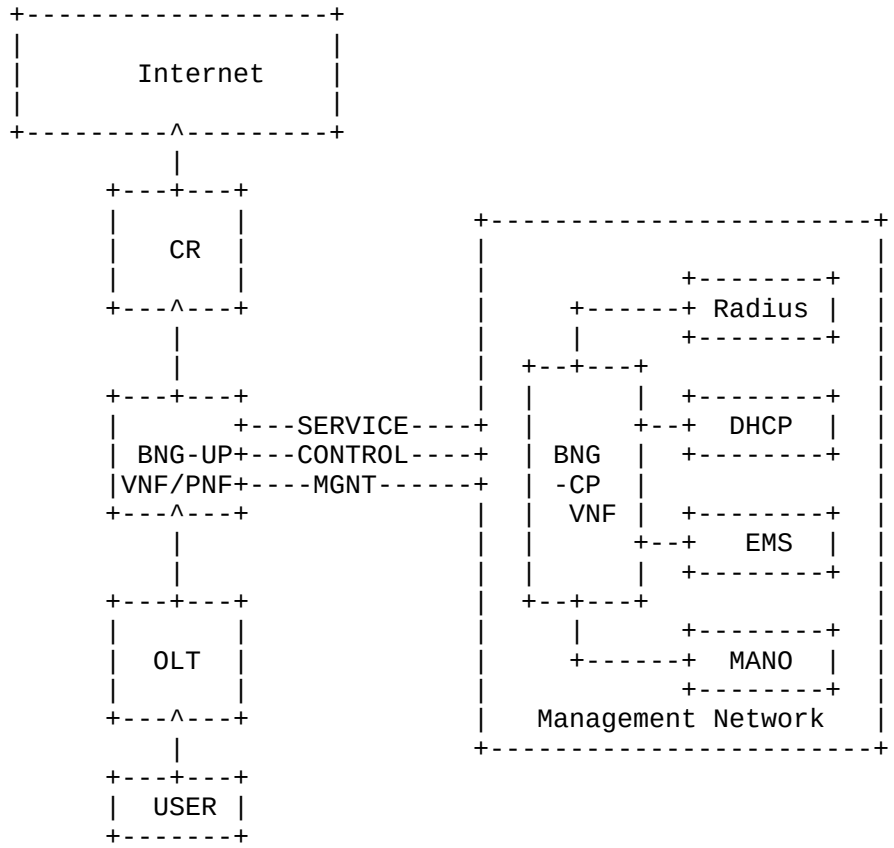
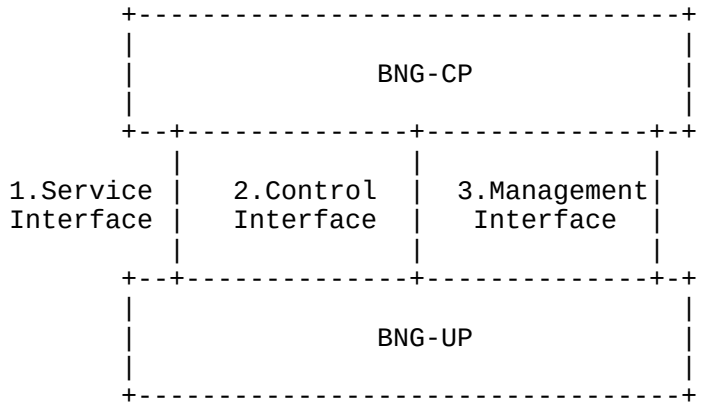


Figure 1: Cloud BNG Deployed in One Disctict

Take one district as an example, here BNG-CP and BNG-UP are separated deployed. As CP is computational intensive, virtualized CP acting as a VNF can meet the requirements of flexibility and fast calculation. UP is traffic intensive, which can be virtualized or stay physical depending on traffic. The virtualized UP with low expense and high flexibility can be suitable in light traffic. In hugh traffic, special hardware is needed with high traffic forwarding performance.

In order to fulfill the function of BNG, BNG-CP needs to communicate with outside systems such as Radius and many others in the management network. In addition, BNG-CP has different interfaces with BNG-UP. There are three kinds of interfaces between BNG-CP and BNG-UP according to its traffic catergories: Service Interface, Control Interface, and Management Interface.



Internal interfaces between the CP and UP of the BNG device

Service interface: The CP and UP use this interface to establish VXLAN tunnels with each other and transmit PPPoE and IPoE packets over the VXLAN tunnels for the authentication.

Control interface: The CP uses this interface to deliver service entries, and the UP uses this interface to report service events to the CP.

Management interface: The CP uses this interface to deliver basic configurations to the UP. This interface runs on NETCONF.

Several related drafts describing these interfaces in details exist. VXLAN-GPE extention draft for C/U separated BNG is related with Servie Interface in [I-D.draft-huang-nov3-vxlan-gpe-extension-for-vbng]. The draft YANG data model for CU separated BNG focuses on Management Interface, seeing in [I-D.draft-hu-rtgwg-cu-separation-

yang-model]. Another two drafts [I-D.draft-cuspdt-rtgwg-cusp-requirements] and [I-D.draft-cuspdt-rtgwg-cu-separation-infor-model] are related with control interface with information model abstraction and suitable protocol discussion.

3.2. CP and UP of BNG deployment within different districts

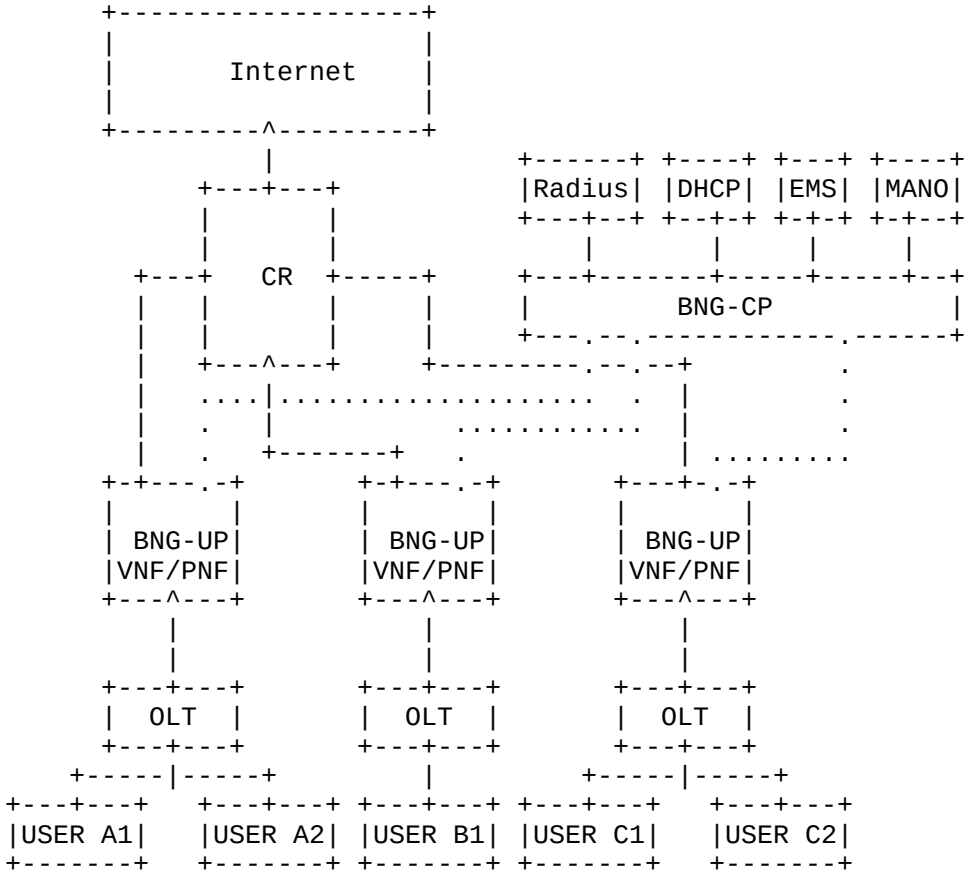


Figure 2: Cloud BNG Deployed in Several Discticts

If scubscribers are distributed in serveral districts, CP is deployed centrally with UP deployed in different districts closing to subscribers. Thus the deployment model can be a few complex. Take three districts A B C for example. Here three UPs are placed which share one CP. CP is usually deployed in Core Date Center such as in the province datacenter with UP in edge Date Centers such as datacenter in cities. In the Data Centers design, we have core data centers and edge data centers according to their location and

responsibility. Core datacenters are often planned in province for the control and management, while edge datacenters in cities or towns for easy service access.

In this scenario, centralized CP faces to the subsystems outside and communicate with all these UPs for the control and management.

Under the CP's control, the corresponding traffic is forwarded by UP to the Internet.

4. The Process of BNG with CP and UP in Home Broadband Service

Take a user Bob accessing to the Internet by Home Broadband Service as an example. The process includs the service traffic from user to the internet and signaling traffic between BNG-UP and BNG-CP. Below is the whole process.

(1)User Bob dialups with packets of PPPoE or IPoE from BNG-UP which will send to BNG-CP with its information. This belongs to signaling traffic.

(2)BNG-CP processes the dialup packets. Confirming with the outside neighboring systems in the management network, BNG-CP makes the decision to permit or deny of the dial through certification. In this step, BNG-CP manages resources and generates tables with information such as User Infor, IP Infor, QoS Info and etc. This belongs to signaling traffic.

(3)BNG-CP sends tables to the corresponding UP or choose one UP in corresponding UPs. This belongs to signaling traffic.

(4)BNG-UP receives the tables, matches rules and performs corresponding actions.

(5)If Bob is certificated and permitted, the UP forwards the traffic into the Internet with related policies such as limited bandwidth, etc. Otherwise, Bob is denied to access the Internet. This belongs to service traffic.

From Step 2 to Step 4, the information model defined in [I-D.draft-cuspdt-rtgwg-cu-separation-infor-model] can be used.

5. High Availability Consideration

As the BNG-CP takes the responsibility of control and management such as communicating with outside systems, generating flow tables and managing UP's resources, high availability of the key component should be considered. Some technology is adopted to ensure the

reliability, such as N+N or N+K active standby BNG-CP. N+N active standby means 1:1 backup for example, while N+K active standby means N:1 backup for example. When active CP fails, standby CP should take the role of active according to some mechanism. Actually in the deployment, resources should be reserved for the backup BNG-CP VNF.

6. Security Considerations

None.

7. IANA Considerations

None.

8. Normative 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>.

Authors' Addresses

Rong Gu
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing, Beijing 100053
China

Email: gurong_cmcc@outlook.com

Sujun Hu
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing, Beijing 100053
China

Email: shujun_hu@outlook.com

Michael Wang
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: wangzitao@huawei.com

Fangwei Hu
ZTE Corporation
No.889 Bibo Rd
Shanghai 201203
China

Phone: +86 21 68896273
Email: hu.fangwei@zte.com.cn

rtgwg
Internet-Draft
Intended status: Informational
Expires: January 3, 2019

S. Hu
China Mobile
V. Lopez
Telefonica
F. Qin
Z. Li
China Mobile
T. Chua
Singapore Telecommunications Limited
M. Wang
J. Song
Huawei

July 2, 2018

Requirements for Control Plane and User Plane Separated BNG Protocol draft-cuspdt-rtgwg-cusp-requirements-02

Abstract

This document introduces the Control Plane and User Plane separated BNG architecture and defines a set of associated terminology. What's more, this document focuses on defining a set of protocol requirements for the BNG-CP and BNG-UPs communication in the Control Plane and User Plane Separated BNG.

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 January 3, 2019.

Table of Contents

1. Introduction

BNG is an Ethernet-centric IP edge router, and the aggregation point for the user traffic. To provide centralized session management, flexible address allocation, high scalability for subscriber management capacity, and cost-efficient redundancy, the CU separated BNG is introduced [TR-384]. The CU separated Service Control Plane could be virtualized and centralized, which is responsible for user access authentication and setting forwarding entries to user planes. The routing control and forwarding plane, i.e. BNG user plane (local), could be distributed across the infrastructure.

This document introduces the Control Plane and User Plane separated BNG architecture and modeling. This document also defines the

protocol requirements for Control Plane and User Plane Separated BNG (CUSP).

2. Concept and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2.1. Terminology

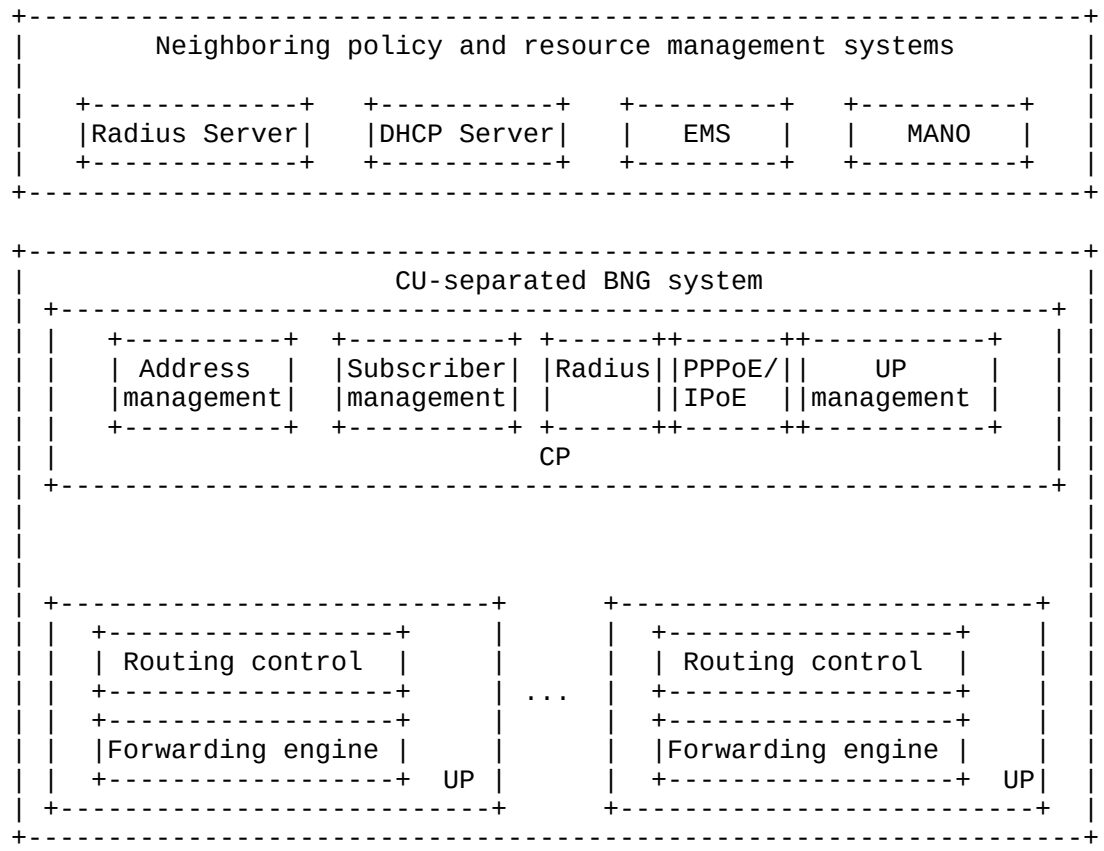
BNG: Broadband Network Gateway. A broadband remote access server (BRAS, B-RAS or BBRAS) routes traffic to and from broadband remote access devices such as digital subscriber line access multiplexers (DSLAM) on an Internet service provider's (ISP) network. BRAS can also be referred to as a Broadband Network Gateway (BNG).

CP: Control Plane. The CP is a user control management component which supports to manage UP's resources such as the user entry and user's QoS policy

UP: User Plane. UP is a network edge and user policy implementation component. The traditional router's Control Plane and forwarding plane are both preserved on BNG devices in the form of a user plane.

3. CU Separated BNG Model

The following figure describes the architecture of CU separated BNG



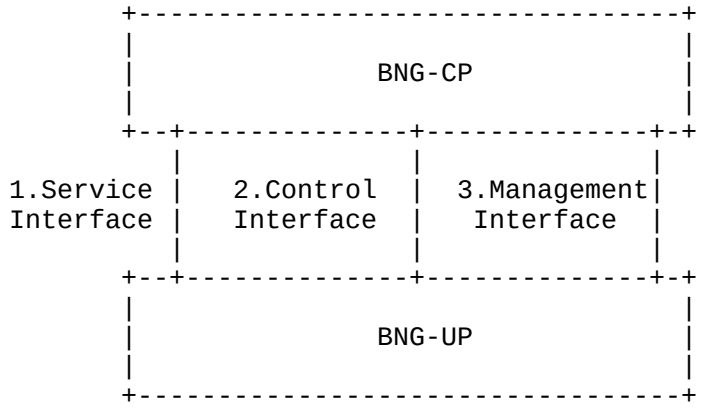
Architecture of CU Separated BNG

Briefly, a CU separated BNG is made up of a Control Plane (CP) and a set of User Planes (UPs) [TR-384], [I-D.cuspdt-rtgwg-cu-separation-bng-deployment]. The Control Plane is a user control management component which supports to manage UP's resources such as the user entry and user's QoS policy, for example, the access bandwidth and priority management. This Control Plane could be virtualized and centralized. The functional modules inside the BNG Service Control Plane can be implemented as VNFs and hosted in a NFVI. The User Plane Management module in the BNG control plane centrally manages the distributed BNG user planes (e.g. load balancing), as well as the setup, deletion, update, and maintenance of channels between control planes and user planes [TR-384], [I-D.cuspdt-rtgwg-cu-separation-bng-deployment]. And the User Plane (UP) is a network edge and user policy implementation component. It can support the forwarding plane functions on traditional BNG devices, such as traffic forwarding, QoS, and traffic statistics

collection, and it can also support the control plane functions on traditional BNG devices, such as routing, multicast, etc [TR-384], [I-D.cuspdt-rtgwg-cu-separation-bng-deployment].

3.1. Internal interfaces between the CP and UP

To support the communication between the Control Plane and User Plane, several interfaces are involved. Figure 2 illustrates the internal interfaces of CU Separated BNG.



Internal interfaces between the CP and UP of the BNG device

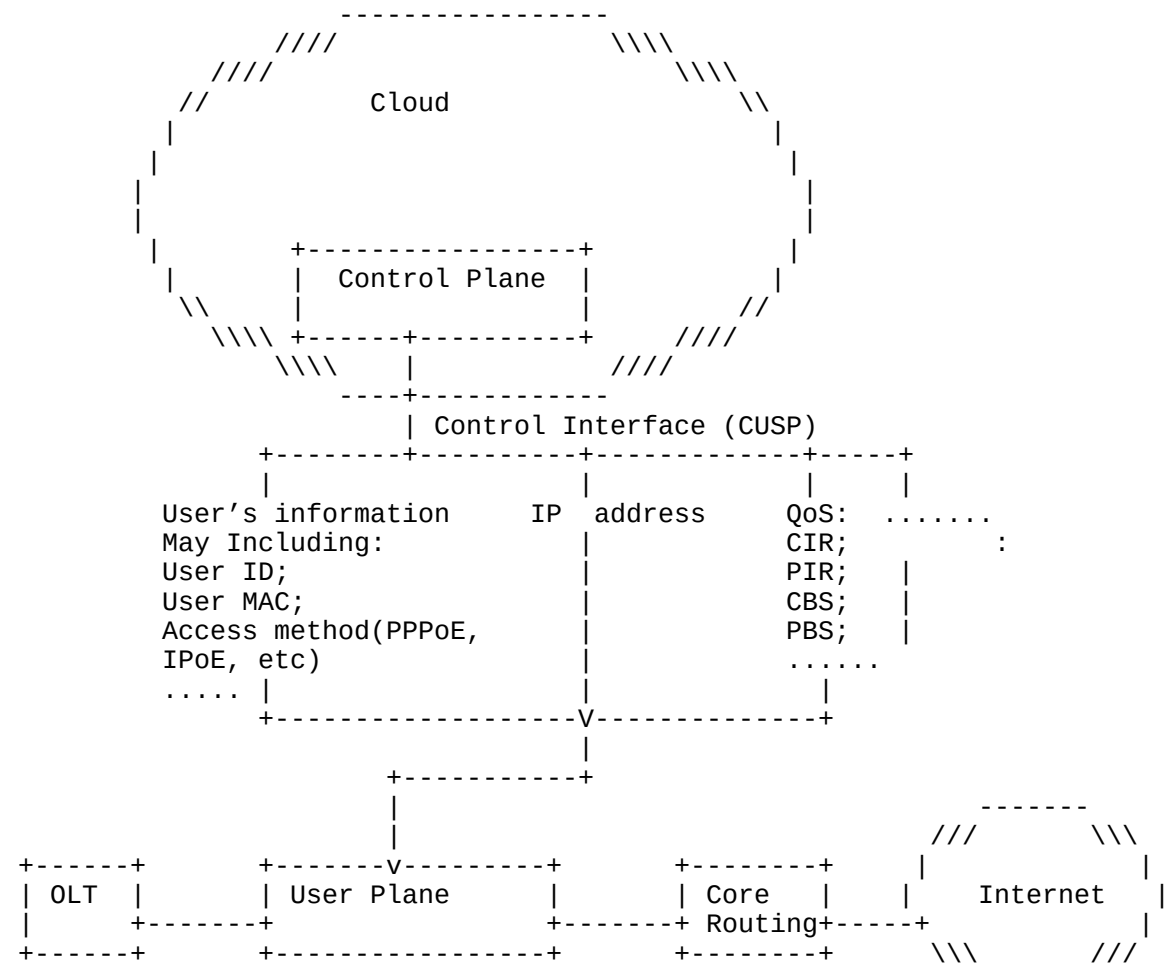
Service interface: The CP and UP use this interface to establish VXLAN tunnels with each other and transmit PPPoE and IPoE packets over the VXLAN tunnels.

Control interface: The CP uses this interface to deliver service entries, and the UP uses this interface to report service events to the CP.

Management interface: The CP uses this interface to deliver configurations to the UP. This interface runs NETCONF.

The CUSP (Control plane and User plane Separated BNG protocol) defines the control interface, and specifies the communication between the centralized control plane and user planes. This protocol should be designed to support establishing and maintaining a conversation between CP and UPs, and transporting the tables which specified in [draft-cuspdt-rtgwg-cu-separation-infor-model].

4. The usage of CU separation BNG protocol



CU Separation BNG protocol usage

As shown in above figure, when users access to the BNG network, the control plane solicits these users' information (such as user's ID, user's MAC, user's access methods, for example via PPPoE/IPoE), associates them with available bandwidth which are reported by User planes, and based on the service's requirement to generate a set of tables, which may include user's information, UP's IP segment, and QoS, etc. Then the control plane can transmit these tables to the

User planes. User planes receive these tables, parse them, match these rules, and then perform corresponding actions.

5. Control Plane and User Plane Separation Protocol Requirements

This section specifies some of the requirements that the CU separation protocol SHOULD support.

5.1. Transmit information tables

The Control Plane and User Plane Separation Protocol MUST allow the CP to send tables to each User Plane device.

a) The current BNG service requires that the UP should support at least 2000 users being accessed every second. And every user requires at least 2000 bytes. To achieve high performance, the CU Separation protocol SHOULD be lightweight.

b) CU separation protocol should support XML/binary data which serves as the encoding format. It allows user information data to be read, saved, and manipulated with tools specific to XML/binary.

c) In order to provide centralized session management, high scalability for subscriber management capacity, and cost-efficient redundancy, batching ability should be involved. The CU Separation protocol should be able to group an ordered set of commands to a UP device. Each such group of commands SHOULD be sent to the UP in as few messages as possible. Furthermore, the protocol MUST support the ability to specify if a command group MUST have all-or-nothing semantics.

d) The CU Separation protocol SHOULD be able to support at least hundreds of UP devices and tens of thousands of ports. For example, the protocol field sizes corresponding to UP or port numbers SHALL be large enough to support the minimum required numbers. This requirement does not relate to the performance of the system as the number of UPs or ports in the system grows.

5.2. Message Priority

The CU Separation protocol MUST provide a means to express the protocol message priorities.

5.3. Reliability

Heartbeat is a periodic signal generated by hardware or software to indicate normal operation or to synchronize other parts of network system.

In CU separation BNG, the heartbeat is sent between CP and UPs at a regular interval in the order of seconds. If the CP/UP does not receive a heartbeat for a time--usually a few heartbeat intervals--the CP/UP that should have sent the heartbeat is assumed to have failed.

The CU separation protocol should support some kind of heartbeat monitor mechanism. And this mechanism should have ability to distinguish whether the interruption is an actual failure. For example, in some scenarios (i.e. CP/UP update, etc), the connection between the UP and CP need to be interrupted. In this case, the interruption should not be reported.

5.4. Support for Secure Communication

As mentioned above, CP may send some information tables to the UP which may be critical to the network function (e.g, User Information, IPv4/IPv6 information) and may reflect the business information (e.g, QoS, service level agreements, etc). Therefore, it MUST be supported to ensure the integrity of all CU Separation protocol messages and protect against man-in-the-middle attacks.

And the CP Separation protocol should support multiple security mechanisms to satisfy various scenarios. For example, when the special lines are implemented between the CP and UPs, the key chain SHOULD be supported. And if some VPNs are deployed between the CP and UPS, the TLS SHOULD be supported. In case of the CP and UPs cross several domains (i.e. cross third-party network), the IPsec SHOULD be supported.

5.5. Version negotiation

The CU separated BNG may consist of different vendors’ devices. Since different vendors’ device may implement different versions of protocol, therefore, the CU separation protocol should provide some mechanisms to perform the version negotiation.

The version negotiation is the process that the CU separated BNG’s Control-Plane uses to evaluate the protocol versions supported by both the control-plane and the user-plane devices. Then a suitable protocol version is selected for communication in CUSP. The process is a "negotiation" because it requires identifying the most recent protocol version that is supported by both the control-plane and the user-plane devices.

5.6. Capability Exchange

The UP Capability Report displays the devices profile, service capability, and other assigned capabilities within the CU separated BNG. The CU separation protocol should provide some mechanism to exchange the UP device's capability

5.7. CP primary/backup capability

A backup CP for disaster recovery is required for the CU separated BNG network. And the CUSP should provide some mechanism to implement the backup CP:

a) In some scenarios, there may be two CP devices both declaring the primary CP. Thus the CUSP should support or associate with some mechanisms to determine which CP is the primary device.

b) In the scenario of the primary CP down, the CUSP should support switching between primary and backup CP.

5.8. Event Notification

The CUSP protocol SHOULD be able to asynchronously notify the CP of events on the UP such as failures and changes in available resources and capabilities. Some scenarios which may initiate the event notification list as follows.

a) Sending response message: As mentioned above, the control plane solicits users' information, associates them with available bandwidth, and generates a set of tables based on the service's requirement. Then the control plane transmits these tables to the conresponding User plane. The UP should respond with an event notification to inform the CP that the tables are received.

b) User trace: The user trace mechanism can support the Control Plane to trace and monitor the network status for users (for example the real-time bandwidth, etc) , debug the user's application. Therefore, the UPs SHOULD be able to notify the CP with the User trace message.

c) Sending statistics parameters: In CU separation BNG, the User-plane will report the traffic statistics parameters to the Control-plane, such as the ingress packets, ingress bytes, egress packets, egress bytes, etc. These parameters can help to measure the BNG network performance. Available network resources can be allocated basing on the statistics parameters by the BNG-CP. Therefore, the UPs SHOULD be able to notify the CP with statistics parameters.

d) Report the result of User Detect: "User Detect" message will be send periodically to detect user dial-up and disconnect. The UPs SHOULD be able to notify the CP with the result of User Detect.

5.9. Query Statistics

The CUSP protocol MUST provide a means for the CP to be able to query statistics (performance monitoring) from the UP.

6. Security Considerations

None.

7. IANA Considerations

None.

8. Normative References

[I-D.cuspdt-rtgwg-cu-separation-bng-deployment]
Gu, R., Hu, S., and Z. Wang, "Deployment Model of Control Plane and User Plane Separation BNG", draft-cuspdt-rtgwg-cu-separation-bng-deployment-00 (work in progress), October 2017.

[I-D.cuspdt-rtgwg-cu-separation-infor-model]
Wang, Z., Gu, R., Lopezalvarez, V., and S. Hu, "Information Model of Control-Plane and User-Plane separation BNG", draft-cuspdt-rtgwg-cu-separation-infor-model-00 (work in progress), February 2018.

[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>.

Authors' Addresses

Shujun Hu
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing, Beijing 100053
China

Email: shujun_hu@outlook.com

Victor Lopez
Telefonica
Sur 3 building, 3rd floor, Ronda de la Comunicacion s/n
Madrid 28050
Spain

Email: victor.lopezalvarez@telefonica.com

Fengwei Qin
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing, Beijing 100053
China

Email: qinfengwei@chinamobile.com

Zhenqiang Li
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing, Beijing 100053
China

Email: lizhenqiang@chinamobile.com

Tee Mong Chua
Singapore Telecommunications Limited
31 Exeter Road, #05-04 Comcentre Podium Block
Singapore City 239732
Singapore

Email: teemong@singtel.com

Michael Wang
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: wangzitao@huawei.com

Jun Song
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: song.jun@huawei.com

NETMOD
Internet-Draft
Intended status: Standards Track
Expires: April 29, 2018

D. Ding
F. Zheng
Huawei

October 26, 2017

YANG Data Model for ARP
draft-ding-netmod-arp-yang-model-00

Abstract

This document defines a YANG data model to describe Address Resolution Protocol (ARP) configurations. It is intended this model be used by service providers who manipulate devices from different vendors in a standard way.

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 April 29, 2018.

Copyright Notice

Table of Contents

1. Introduction

This document defines a YANG [RFC6020] data model for Address Resolution Protocol [RFC826] implementation and identification of some common properties within a device containing a Network Configuration Protocol (NETCONF) server. Devices that are managed by NETCONF and perhaps other mechanisms have common properties that need to be configured and monitored in a standard way.

The data model convers configuration of system parameters of ARP, such as static ARP entries, timeout for dynamic ARP entries, interface ARP, proxy ARP, and so on. It also provides information about running state of ARP implementations.

1.1. Terminology

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].

The following terms are defined in [RFC6241] and are not redefined here:

- client
- configuration data
- server

- state data

1.2. Tree Diagrams

A simplified graphical representation of the data model is presented in Section 3.

- Brackets "[" and "]" enclose list keys.
- Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).
- Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.
- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- Ellipsis ("...") stands for contents of subtrees that are not shown.

2. Problem Statement

This document defines a YANG [RFC7950] configuration data model that may be used to configure the ARP feature running on a system. YANG models can be used with network management protocols such as NETCONF [RFC6241] to install, manipulate, and delete the configuration of network devices.

The data model makes use of the YANG "feature" construct which allows implementations to support only those ARP features that lie within their capabilities. It is intended this model be used by service providers who manipulate devices from different vendors in a standard way.

This module can be used to configure the ARP applications for discovering the link layer address associated with a given Internet layer address.

3. Design of the Data Model

This data model intends to describe the processing that a protocol finds the hardware address, also known as Media Access Control (MAC) address, of a host from its known IP address. These tasks include, but are not limited to, adding a static entry in the ARP cache, configuring ARP cache entry timeout, and clearing dynamic entries from the ARP cache.

This data model has one top level container, ARP, which consists of several second level containers. Each of these second level containers describes a particular category of ARP handling, such as defining static mapping between an IP address (32-bit address) and a Media Access Control (MAC) address (48-bit address).

```
module: ietf-arp
   +--rw arp
      +--rw arp-static-tables
      |  +--rw arp-static-table* [vrf-name ip-address]
      |     +--rw vrf-name       arp:routing-instance-ref
      |     +--rw ip-address     inet:ipv4-address-no-zone
      |     +--rw mac-address    yang:mac-address
      |     +--rw if-name?       leafref
      +--rw arp-interfaces
      |  +--rw arp-interface* [if-name]
      |     +--rw if-name                    leafref
      |     +--rw expire-time?               uint32
      |     +--rw arp-learn-disable?         boolean
      |     +--rw proxy-enable?              boolean
      |     +--rw probe-interval?            uint8
      |     +--rw probe-times?               uint8
      |     +--rw probe-unicast?             boolean
      |     +--rw arp-gratuitous?            boolean
      |     +--rw arp-gratuitous-interval?   uint32
      |     +--rw arp-gratuitous-drop?       boolean
      |     +--rw arp-if-limits
      |        +--rw arp-if-limit* [vlan-id]
      |           +--rw vlan-id            uint16
      |           +--rw limit-number       uint32
      |           +--rw threshold-value?   uint32
      +--ro arp-tables
      |  +--ro arp-table* [vrf-name ip-address]
      |     +--ro vrf-name       arp:routing-instance-ref
      |     +--ro ip-address     inet:ipv4-address-no-zone
      |     +--ro mac-address?   yang:mac-address
      |     +--ro expire-time?   uint32
      |     +--ro if-name?       leafref
      +--ro arp-statistics
         +--ro global-statistics*
         |  +--ro requests-received?     uint32
         |  +--ro replies-received?      uint32
         |  +--ro gratuitous-received?   uint32
         |  +--ro requests-sent?         uint32
         |  +--ro replies-sent?          uint32
         |  +--ro gratuitous-sent?       uint32
         |  +--ro drops-received?        uint32
```

```
            |  +--ro total-received?        uint32
            |  +--ro total-sent?            uint32
            |  +--ro arp-dynamic-count?     uint32
            |  +--ro arp-static-count?      uint32
            +--ro arp-if-statistics* [if-name]
               +--ro if-name                leafref
               +--ro requests-received?     uint32
               +--ro replies-received?      uint32
               +--ro gratuitous-received?   uint32
               +--ro requests-sent?         uint32
               +--ro replies-sent?          uint32
               +--ro gratuitous-sent?       uint32
```

4. YANG Module

This section presents the YANG module for the ARP data model defined in this document.

```
<CODE BEGINS> file "ietf-arp@2017-10-18.yang"
module ietf-arp {
  namespace "urn:ietf:params:xml:ns:yang:ietf-arp";
  prefix arp;

 // import some basic types

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-yang-types {
    prefix yang;
  }

  import ietf-interfaces {
    prefix if;
  }

  import ietf-network-instance {
    prefix ni;
  }
  organization
    "IETF Netmod (Network Modeling) Working Group";
  contact
    "WG Web: <http://tools.ietf.org/wg/netmod/>
     WG List: <mailto: netmod@ietf.org>
```

```
     Editor: Xiaojian Ding
                dingxiaojian1@huawei.com
     Editor: Feng Zheng
                habby.zheng@huawei.com";
  description
    "Address Resolution Protocol (ARP) management, which includes
       static ARP configuration, dynamic ARP learning, ARP entry query,
       and packet statistics collection.";

  revision 2017-10-18 {
    description
      "Init revision";
    reference
      "RFC XXX: ARP (Address Resolution Protocol) YANG data model.";
  }

/*grouping*/

  grouping arp-prob-grouping {
    description
      "Common configuration for all ARP probe.";
    leaf probe-interval {
      type uint8 {
        range "1..5";
      }
          units "second";
      description
        "Interval for detecting dynamic ARP entries.";
    }
    leaf probe-times {
      type uint8 {
        range "0..10";
      }
      description
        "Number of aging probe attempts for a dynamic ARP entry. If
    a device does not receive an ARP reply message after the number
                of aging probe attempts reaches a specified number, the
                dynamic ARP entry is deleted.";
    }
    leaf probe-unicast {
      type boolean;
      default "false";
      description
        "Send unicast ARP aging probe messages for a dynamic ARP
                entry.";
    }
  }
```

```
  grouping arp-gratuitous-grouping {
    description
      "Configure gratuitous ARP.";
    leaf arp-gratuitous {
      type boolean;
      default "false";
      description
      "Enable or disable sending gratuitous-arp packet on
          interface.";
    }
    leaf arp-gratuitous-interval {
      type uint32 {
        range "1..86400";
      }
          units "second";
      description
        "The interval of sending gratuitous-arp packet on the
                interface.";
    }
    leaf arp-gratuitous-drop {
      type boolean;
      default "false";
      description
      "Drop the receipt of gratuitous ARP packets on the interface.";
    }
  }

  grouping arp-statistics-grouping {
    description "IP ARP statistics information";
    leaf requests-received {
      type uint32;
      description "Total ARP requests received";
    }
    leaf replies-received {
      type uint32;
      description "Total ARP replies received";
    }
    leaf gratuitous-received {
      type uint32;
      description "Total gratuitous ARP received";
    }
    leaf requests-sent {
      type uint32;
      description "Total ARP requests sent";
    }
    leaf replies-sent {
      type uint32;
      description "Total ARP replies sent";
```

```
    }
    leaf gratuitous-sent {
      type uint32;
      description "Total gratuituous ARP sent";
    }
  }

  /* Typedefs */

  typedef routing-instance-ref {
    type leafref {
      path "/ni:network-instances/ni:network-instance/ni:name";
    }
    description
      "This type is used for leafs that reference a routing instance
          configuration.";
  }

  /* Configuration data nodes */

  container arp {
    description
      "Address Resolution Protocol (ARP) management, which includes
          static ARP configuration, dynamic ARP learning, ARP entry
          query, and packet statistics collection.";

    container arp-static-tables {
      description
        "List of static ARP configurations.";
      list arp-static-table {
        key "vrf-name ip-address";
        description
          "Static ARP table. By default, the system ARP table is
                  empty, and address mappings are implemented by dynamic
                  ARP.";
        leaf vrf-name {
          type arp:routing-instance-ref;
          description
            "Name of a VPN instance. This parameter is used to
                        support the VPN feature. If this parameter is
                        set, it indicates that the ARP entry is in the
                        associated VLAN.";
        }
        leaf ip-address {
          type inet:ipv4-address-no-zone;
          description
            "IP address, in dotted decimal notation.";
        }
```

```
        leaf mac-address {
          type yang:mac-address;
          mandatory true;
          description
            "MAC address in the format of H-H-H, in which H is
                        a hexadecimal number of 1 to 4 bits. ";
        }
        leaf if-name {
          type leafref {
            path "/if:interfaces/if:interface/if:name";
          }
          description
            "Name of the ARP outbound interface.";
        }
      }
    }//End of arp-static-tables

    container arp-interfaces {
      description
        "List of ARP Interface configurations.";
      list arp-interface {
        key "if-name";
        description
          "ARP interface configuration, including the aging time,
                  probe interval, number of aging probe attempts, ARP
                  learning status, and ARP proxy.";
        leaf if-name {
          type leafref {
            path "/if:interfaces/if:interface/if:name";
          }
          description
            "Name of the interface that has learned dynamic ARP
                        entries.";
        }
        leaf expire-time {
          type uint32 {
            range "60..86400";
          }
                  units "second";
          description
            "Aging time of a dynamic ARP entry.";
        }
        leaf arp-learn-disable {
          type boolean;
          default "false";
          description
            "Whether dynamic ARP learning is disabled. If the value
                        is True, dynamic ARP learning is disabled. If the value
```

```
                    is False, dynamic ARP learning is enabled.";
        }
        leaf proxy-enable {
          type boolean;
          default "false";
          description
            "Enable proxy ARP.";
        }
        uses arp-prob-grouping;
        uses arp-gratuitous-grouping;

        container arp-if-limits {
          description
            "Maximum number of dynamic ARP entries that an interface
                    can learn.";
          list arp-if-limit {
            key "vlan-id";
            description
              "Maximum number of dynamic ARP entries that an
                          interface can learn. If the number of ARP entries that
                          an interface can learn changes and the number of the
                          learned ARP entries exceeds the changed value, the
                          interface cannot learn additional ARP entries. The
                          system prompts you to delete the excess ARP entries.";
             leaf vlan-id {
              type uint16 {
                range "0..4094";
              }
              description
                "ID of the VLAN where ARP learning is restricted.
                                This parameter can be set only on Layer 2 interf
aces
                                and sub-interfaces. Ethernet, GE, VE, and Eth-Tr
unk
                                interfaces can be both Layer 3 and Layer 2
                                interfaces. When they work in Layer 3 mode, they
                                cannot have VLANs configured. When they work in
Layer
                                2 mode, they must have VLANs configured. Etherne
t,
                                GE, and Eth-Trunk sub-interfaces can be both com
mon
                                and QinQ sub-interfaces. ";
            }
            leaf limit-number {
              type uint32 {
                range "1..65536";
              }
              mandatory true;
              description
                "Maximum number of dynamic ARP entries that an
                                interface can learn.";
            }
```

```
            leaf threshold-value {
              type uint32 {
                range "60..100";
              }
              must "not(not(../limit-number))"{
                           description
              "Upper boundary must be higher than lower boundary.";
                           }
              description
                "Alarm-Threshold for maximum number of ARP entries
                                that an interface can learn.";
            }
          }
        }//End of arp-if-limits
      }
    }// End of arp-interfaces

    container arp-tables {
        config false;
      description
        "List of ARP entries that can be queried.";
      list arp-table {
        key "vrf-name ip-address";
        description
          "Query ARP entries, including static, dynamic, and
                   interface-based ARP entries.";
        leaf vrf-name {
          type arp:routing-instance-ref;
          description
            "Name of the VPN instance to which an ARP entry
                        belongs.";
        }
        leaf ip-address {
          type inet:ipv4-address-no-zone;
          description
            "IP address, in dotted decimal notation.";
        }
        leaf mac-address {
          type yang:mac-address;
          description
            "MAC address in the format of H-H-H, in which H is a
                        hexadecimal number of 1 to 4 bits. ";
        }
        leaf expire-time {
          type uint32 {
            range "1..1440";
          }
          description
```

```
            "Aging time of a dynamic ARP entry. ";
        }
        leaf if-name {
          type leafref {
            path "/if:interfaces/if:interface/if:name";
          }
          description
            "Type and number of the interface that has learned ARP
                        entries.";
        }
      }
    }//End of arp-tables

    container arp-statistics {
      config false;
      description
        "List of ARP packet statistics.";
      list global-statistics {
        description
          "ARP packet statistics.";
        uses arp-statistics-grouping;
        leaf drops-received {
          type uint32 {
            range "0..4294967294";
          }
          description
            "Number of ARP packets discarded.";
        }
        leaf total-received {
          type uint32 {
            range "0..4294967294";
          }
          description
            "Total number of ARP received packets.";
        }
        leaf total-sent {
          type uint32 {
            range "0..4294967294";
          }
          description
            "Total number of ARP sent packets.";
        }
        leaf arp-dynamic-count {
          type uint32 {
            range "0..4294967294";
          }
          description
            "Number of dynamic ARP count.";
```

```
        }
        leaf arp-static-count {
          type uint32 {
            range "0..4294967294";
          }
          description
            "Number of static ARP count.";
        }
      }
      list arp-if-statistics {
        key "if-name";
        description
          "ARP statistics on interfaces. ARP statistics on all
                  interfaces are displayed in sequence.";
        leaf if-name {
          type leafref {
            path "/if:interfaces/if:interface/if:name";
          }
          description
            "Name of an interface where ARP statistics to be
                        displayed reside.";
        }
        uses arp-statistics-grouping;
      }
    }// End of arp-statistics
  }
}
<CODE ENDS>
```

5. Data Model Examples

This section presents a simple but complete example of configuring static ARP entries and interfaces, based on the YANG module specified in Section 4.

5.1. Static ARP entries

```
   Requirement:
   Enable static ARP entry configuration.
      <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
         <arp xmlns="urn:ietf:params:xml:ns:yang:ietf-arp">
            <arp-static-tables>
                   <vrf-name> __public__ </vrf-name>
                       <ip-address> 10.2.2.3 </ip-address>
                       <mac-address> 00e0-fc01-0000 </mac-address>
                       <if-name> GE1/0/1 </if-name>
                </arp-static-tables>
         </arp>
```

5.2. ARP interfaces

```
   Requirement:
   Enable static ARP interface configuration.

      <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
         <arp xmlns="urn:ietf:params:xml:ns:yang:ietf-arp">
            <arp-interfaces>
                   <if-name> GE1/0/1 </if-name>
                   <expire-time>1200</expire-time>
                       <arp-learn-disable>false</arp-learn-disable>
                   <proxy-enable>false</proxy-enable>
                       <probe-interval>5</probe-interval>
                       <probe-times>3</probe-times>
                       <probe-unicast>false</probe-unicast>
                       <arp-gratuitous>false</arp-gratuitous>
                   <arp-gratuitous-interval>60</arp-gratuitous-interval>
                       <arp-gratuitous-drop>false</arp-gratuitous-drop>
                       <arp-if-limits>
                          <vlan-id>3</vlan-id>
                          <limit-number>65535</limit-number>
                          <threshold-value>80</threshold-value>
                   </arp-if-limits>
             </arp-interfaces>
         </arp>
```

6. Security Considerations

The YANG module defined in this document is designed to be accessed via YANG based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. Both of these protocols have mandatory-to-implement secure transport layers (e.g., SSH, TLS) with mutual authentication.

The NETCONF access control model (NACM) [RFC6536] provides the means to restrict access for particular users to a pre-configured subset of all available protocol operations and content.

These are the subtrees and data nodes and their sensitivity/ vulnerability:

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations.

7. Conclusions

TBD.

8. References

8.1. Normative 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>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

8.2. Informative References

[RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826, November 1982, <https://www.rfc-editor.org/info/rfc826>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

Authors' Addresses

Xiaojian Ding
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: dingxiaojian1@huawei.com

Feng Zheng
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: habby.zheng@huawei.com

Routing Working Group
Internet-Draft
Intended status: Informational
Expires: December 13, 2018

F. Baker
C. Bowers
Juniper Networks
J. Linkova
Google
June 11, 2018

Enterprise Multihoming using Provider-Assigned Addresses without Network Prefix Translation: Requirements and Solution
draft-ietf-rtgwg-enterprise-pa-multihoming-07

Abstract

Connecting an enterprise site to multiple ISPs using provider-assigned addresses is difficult without the use of some form of Network Address Translation (NAT). Much has been written on this topic over the last 10 to 15 years, but it still remains a problem without a clearly defined or widely implemented solution. Any multihoming solution without NAT requires hosts at the site to have addresses from each ISP and to select the egress ISP by selecting a source address for outgoing packets. It also requires routers at the site to take into account those source addresses when forwarding packets out towards the ISPs.

This document attempts to define a complete solution to this problem. It covers the behavior of routers to forward traffic taking into account source address, and it covers the behavior of host to select appropriate source addresses. It also covers any possible role that routers might play in providing information to hosts to help them select appropriate source addresses. In the process of exploring potential solutions, this documents also makes explicit requirements for how the solution would be expected to behave from the perspective of an enterprise site network administrator .

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 December 13, 2018.

Table of Contents

1. Introduction

Site multihoming, the connection of a subscriber network to multiple upstream networks using redundant uplinks, is a common enterprise architecture for improving the reliability of its Internet connectivity. If the site uses provider-independent (PI) addresses, all traffic originating from the enterprise can use source addresses from the PI address space. Site multihoming with PI addresses is commonly used with both IPv4 and IPv6, and does not present any new technical challenges.

It may be desirable for an enterprise site to connect to multiple ISPs using provider-assigned (PA) addresses, instead of PI addresses.

Multihoming with provider-assigned addresses is typically less expensive for the enterprise relative to using provider-independent addresses. PA multihoming is also a practice that should be facilitated and encouraged because it does not add to the size of the Internet routing table, whereas PI multihoming does. Note that PA is also used to mean "provider-aggregatable". In this document we assume that provider-assigned addresses are always provider-aggregatable.

With PA multihoming, for each ISP connection, the site is assigned a prefix from within an address block allocated to that ISP by its National or Regional Internet Registry. In the simple case of two ISPs (ISP-A and ISP-B), the site will have two different prefixes assigned to it (prefix-A and prefix-B). This arrangement is problematic. First, packets with the "wrong" source address may be dropped by one of the ISPs. In order to limit denial of service attacks using spoofed source addresses, BCP38 [RFC2827] recommends that ISPs filter traffic from customer sites to only allow traffic with a source address that has been assigned by that ISP. So a packet sent from a multihomed site on the uplink to ISP-B with a source address in prefix-A may be dropped by ISP-B.

However, even if ISP-B does not implement BCP38 or ISP-B adds prefix-A to its list of allowed source addresses on the uplink from the multihomed site, two-way communication may still fail. If the packet with source address in prefix-A was sent to ISP-B because the uplink to ISP-A failed, then if ISP-B does not drop the packet and the packet reaches its destination somewhere on the Internet, the return packet will be sent back with a destination address in prefix-A. The return packet will be routed over the Internet to ISP-A, but it will not be delivered to the multihomed site because its link with ISP-A has failed. Two-way communication would require some arrangement for ISP-B to advertise prefix-A when the uplink to ISP-A fails.

Note that the same may be true with a provider that does not implement BCP 38, if his upstream provider does, or has no corresponding route. The issue is not that the immediate provider implements ingress filtering; it is that someone upstream does, or lacks a route.

With IPv4, this problem is commonly solved by using [RFC1918] private address space within the multi-homed site and Network Address Translation (NAT) or Network Address/Port Translation (NAPT) on the uplinks to the ISPs. However, one of the goals of IPv6 is to eliminate the need for and the use of NAT or NAPT. Therefore, requiring the use of NAT or NAPT for an enterprise site to multihome with provider-assigned addresses is not an attractive solution.

[RFC6296] describes a translation solution specifically tailored to meet the requirements of multi-homing with provider-assigned IPv6 addresses. With the IPv6-to-IPv6 Network Prefix Translation (NPTv6) solution, within the site an enterprise can use Unique Local Addresses [RFC4193] or the prefix assigned by one of the ISPs. As traffic leaves the site on an uplink to an ISP, the source address gets translated to an address within the prefix assigned by the ISP on that uplink in a predictable and reversible manner. [RFC6296] is currently classified as Experimental, and it has been implemented by several vendors. See Section 7.2, for more discussion of NPTv6.

This document defines routing requirements for enterprise multihoming using provider-assigned IPv6 addresses. We have made no attempt to write these requirements in a manner that is agnostic to potential solutions. Instead, this document focuses on the following general class of solutions.

Each host at the enterprise has multiple addresses, at least one from each ISP-assigned prefix. Each host, as discussed in Section 5.1 and [RFC6724], is responsible for choosing the source address applied to each packet it sends. A host SHOULD be able respond dynamically to the failure of an uplink to a given ISP by no longer sending packets with the source address corresponding to that ISP. Potential mechanisms for the communication of changes in the network to the host are Neighbor Discovery Router Advertisements, DHCPv6, and ICMPv6.

The routers in the enterprise network are responsible for ensuring that packets are delivered to the "correct" ISP uplink based on source address. This requires that at least some routers in the site network are able to take into account the source address of a packet when deciding how to route it. That is, some routers must be capable of some form of Source Address Dependent Routing (SADR), if only as described in [RFC3704]. At a minimum, the routers connected to the ISP uplinks (the site exit routers or SERs) must be capable of Source Address Dependent Routing. Expanding the connected domain of routers capable of SADR from the site exit routers deeper into the site network will generally result in more efficient routing of traffic with external destinations.

The document first looks in more detail at the enterprise networking environments in which this solution is expected to operate. It then discusses existing and proposed mechanisms for hosts to select the source address applied to packets. Finally, it looks at the requirements for routing that are needed to support these enterprise network scenarios and the mechanisms by which hosts are expected to select source addresses dynamically based on network state.

2. 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.

3. Enterprise Multihoming Requirements

3.1. Simple ISP Connectivity with Connected SERs

We start by looking at a scenario in which a site has connections to two ISPs, as shown in Figure 1. The site is assigned the prefix 2001:db8:0:a000::/52 by ISP-A and prefix 2001:db8:0:b000::/52 by ISP-B. We consider three hosts in the site. H31 and H32 are on a LAN that has been assigned subnets 2001:db8:0:a010::/64 and 2001:db8:0:b010::/64. H31 has been assigned the addresses 2001:db8:0:a010::31 and 2001:db8:0:b010::31. H32 has been assigned 2001:db8:0:a010::32 and 2001:db8:0:b010::32. H41 is on a different subnet that has been assigned 2001:db8:0:a020::/64 and 2001:db8:0:b020::/64.

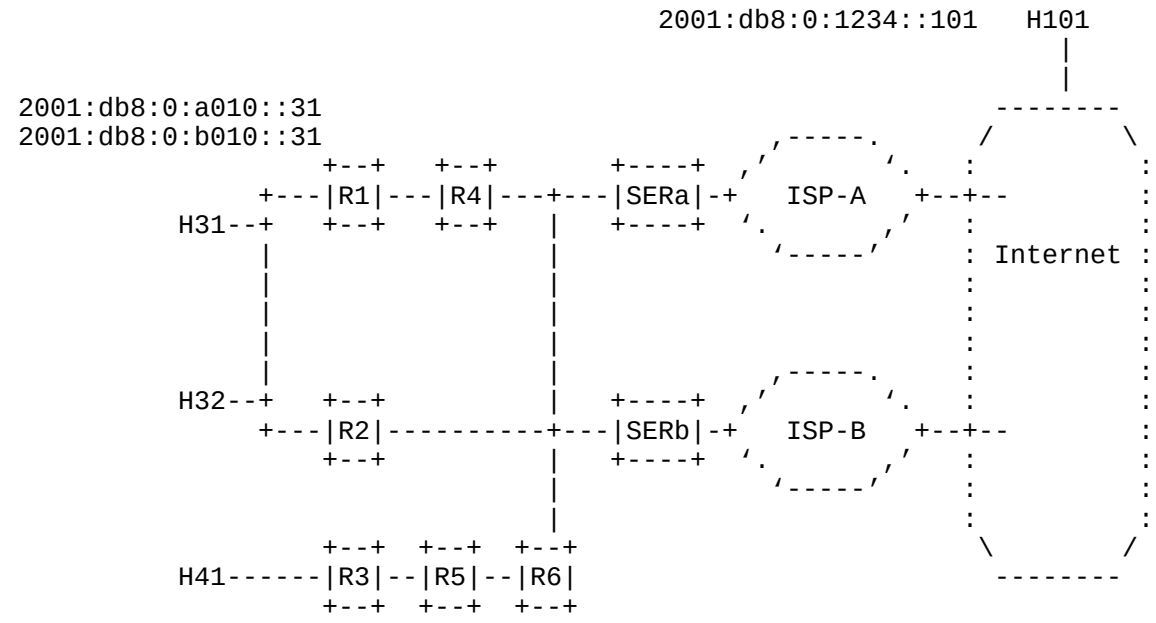


Figure 1: Simple ISP Connectivity With Connected SERs

We refer to a router that connects the site to an ISP as a site edge router(SER). Several other routers provide connectivity among the internal hosts (H31, H32, and H41), as well as connecting the internal hosts to the Internet through SERa and SERb. In this example SERa and SERb share a direct connection to each other. In Section 3.2, we consider a scenario where this is not the case.

For the moment, we assume that the hosts are able to make good choices about which source addresses through some mechanism that doesn't involve the routers in the site network. Here, we focus on primary task of the routed site network, which is to get packets efficiently to their destinations, while sending a packet to the ISP that assigned the prefix that matches the source address of the packet. In Section 5, we examine what role the routed network may play in helping hosts make good choices about source addresses for packets.

With this solution, routers will need form of Source Address Dependent Routing, which will be new functionality. It would be useful if an enterprise site does not need to upgrade all routers to

support the new SADR functionality in order to support PA multi-homing. We consider if this is possible and what are the tradeoffs of not having all routers in the site support SADR functionality.

In the topology in Figure 1, it is possible to support PA multihoming with only SERa and SERb being capable of SADR. The other routers can continue to forward based only on destination address, and exchange routes that only consider destination address. In this scenario, SERa and SERb communicate source-scoped routing information across their shared connection. When SERa receives a packet with a source address matching prefix 2001:db8:0:b000::/52 , it forwards the packet to SERb, which forwards it on the uplink to ISP-B. The analogous behaviour holds for traffic that SERb receives with a source address matching prefix 2001:db8:0:a000::/52.

In Figure 1, when only SERa and SERb are capable of source address dependent routing, PA multi-homing will work. However, the paths over which the packets are sent will generally not be the shortest paths. The forwarding paths will generally be more efficient as more routers are capable of SADR. For example, if R4, R2, and R6 are upgraded to support SADR, then can exchange source-scoped routes with SERa and SERb. They will then know to send traffic with a source address matching prefix 2001:db8:0:b000::/52 directly to SERb, without sending it to SERa first.

3.2. Simple ISP Connectivity Where SERs Are Not Directly Connected

In Figure 2, we modify the topology slightly by inserting R7, so that SERa and SERb are no longer directly connected. With this topology, it is not enough to just enable SADR routing on SERa and SERb to support PA multi-homing. There are two solutions to ways to enable PA multihoming in this topology.

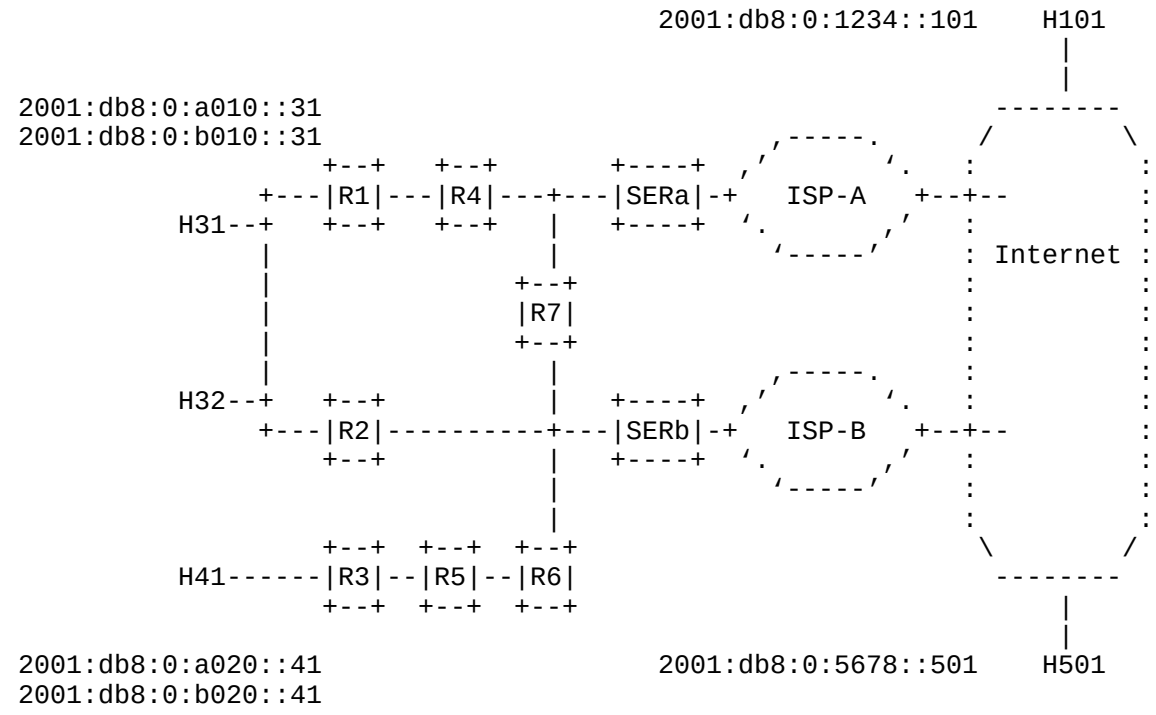


Figure 2: Simple ISP Connectivity Where SERs Are Not Directly Connected

One option is to effectively modify the topology by creating a logical tunnel between SERa and SERb, using GRE for example. Although SERa and SERb are not directly connected physically in this topology, they can be directly connected logically by a tunnel.

The other option is to enable SADR functionality on R7. In this way, R7 will exchange source-scoped routes with SERa and SERb, making the three routers act as a single SADR domain. This illustrates the basic principle that the minimum requirement for the routed site network to support PA multi-homing is having all of the site exit routers be part of a connected SADR domain. Extending the connected SADR domain beyond that point can produce more efficient forwarding paths.

3.3. Enterprise Network Operator Expectations

Before considering a more complex scenario, let’s look in more detail at the reasonably simple multihoming scenario in Figure 2 to understand what can reasonably be expected from this solution. As a

general guiding principle, we assume an enterprise network operator will expect a multihomed network to behave as close as to a single-homed network as possible. So a solution that meets those expectations where possible is a good thing.

For traffic between internal hosts and traffic from outside the site to internal hosts, an enterprise network operator would expect there to be no visible change in the path taken by this traffic, since this traffic does not need to be routed in a way that depends on source address. It is also reasonable to expect that internal hosts should be able to communicate with each other using either of their source addresses without restriction. For example, H31 should be able to communicate with H41 using a packet with S=2001:db8:0:a010::31, D=2001:db8:0:b010::41, regardless of the state of uplink to ISP-B.

These goals can be accomplished by having all of the routers in the network continue to originate normal unscoped destination routes for their connected networks. If we can arrange so that these unscoped destination routes get used for forwarding this traffic, then we will have accomplished the goal of keeping forwarding of traffic destined for internal hosts, unaffected by the multihoming solution.

For traffic destined for external hosts, it is reasonable to expect that traffic with an source address from the prefix assigned by ISP-A to follow the path to that the traffic would follow if there is no connection to ISP-B. This can be accomplished by having SERa originate a source-scoped route of the form (S=2001:db8:0:a000::/52, D=::/0) . If all of the routers in the site support SADR, then the path of traffic exiting via ISP-A can match that expectation. If some routers don’t support SADR, then it is reasonable to expect that the path for traffic exiting via ISP-A may be different within the site. This is a tradeoff that the enterprise network operator may decide to make.

It is important to understand how this multihoming solution behaves when an uplink to one of the ISPs fails. To simplify this discussion, we assume that all routers in the site support SADR. We first start by looking at how the network operates when the uplinks to both ISP-A and ISP-B are functioning properly. SERa originates a source-scoped route of the form (S=2001:db8:0:a000::/52, D=::/0), and SERb is originates a source-scoped route of the form (S=2001:db8:0:b000::/52, D=::/0). These routes are distributed through the routers in the site, and they establish within the routers two set of forwarding paths for traffic leaving the site. One set of forwarding paths is for packets with source address in 2001:db8:0:a000::/52. The other set of forwarding paths is for packets with source address in 2001:db8:0:b000::/52. The normal destination routes which are not scoped to these two source prefixes

play no role in the forwarding. Whether a packet exits the site via SERa or via SERb is completely determined by the source address applied to the packet by the host. So for example, when host H31 sends a packet to host H101 with (S=2001:db8:0:a010::31, D=2001:db8:0:1234::101), the packet will only be sent out the link from SERa to ISP-A.

Now consider what happens when the uplink from SERa to ISP-A fails. The only way for the packets from H31 to reach H101 is for H31 to start using the source address for ISP-B. H31 needs to send the following packet: (S=2001:db8:0:b010::31, D=2001:db8:0:1234::101).

This behavior is very different from the behavior that occurs with site multihoming using PI addresses or with PA addresses using NAT. In these other multi-homing solutions, hosts do not need to react to network failures several hops away in order to regain Internet access. Instead, a host can be largely unaware of the failure of an uplink to an ISP. When multihoming with PA addresses and NAT, existing sessions generally need to be re-established after a failure since the external host will receive packets from the internal host with a new source address. However, new sessions can be established without any action on the part of the hosts.

Another example where the behavior of this multihoming solution differs significantly from that of multihoming with PI address or with PA addresses using NAT is in the ability of the enterprise network operator to route traffic over different ISPs based on destination address. We still consider the fairly simple network of Figure 2 and assume that uplinks to both ISPs are functioning. Assume that the site is multihomed using PA addresses and NAT, and that SERa and SERb each originate a normal destination route for D=::/0, with the route origination dependent on the state of the uplink to the respective ISP.

Now suppose it is observed that an important application running between internal hosts and external host H101 experience much better performance when the traffic passes through ISP-A (perhaps because ISP-A provides lower latency to H101.) When multihoming this site with PI addresses or with PA addresses and NAT, the enterprise network operator can configure SERa to originate into the site network a normal destination route for D=2001:db8:0:1234::/64 (the destination prefix to reach H101) that depends on the state of the uplink to ISP-A. When the link to ISP-A is functioning, the destination route D=2001:db8:0:1234::/64 will be originated by SERa, so traffic from all hosts will use ISP-A to reach H101 based on the longest destination prefix match in the route lookup.

Implementing the same routing policy is more difficult with the PA multihoming solution described in this document since it doesn't use NAT. By design, the only way to control where a packet exits this network is by setting the source address of the packet. Since the network cannot modify the source address without NAT, the host must set it. To implement this routing policy, each host needs to use the source address from the prefix assigned by ISP-A to send traffic destined for H101. Mechanisms have been proposed to allow hosts to choose the source address for packets in a fine grained manner. We will discuss these proposals in Section 5. However, interacting with host operating systems in some manner to ensure a particular source address is chosen for a particular destination prefix is not what an enterprise network administrator would expect to have to do to implement this routing policy.

3.4. More complex ISP connectivity

The previous sections considered two variations of a simple multihoming scenario where the site is connected to two ISPs offering only Internet connectivity. It is likely that many actual enterprise multihoming scenarios will be similar to this simple example. However, there are more complex multihoming scenarios that we would like this solution to address as well.

It is fairly common for an ISP to offer a service in addition to Internet access over the same uplink. Two variation of this are reflected in Figure 3. In addition to Internet access, ISP-A offers a service which requires the site to access host H51 at 2001:db8:0:5555::51. The site has a single physical and logical connection with ISP-A, and ISP-A only allows access to H51 over that connection. So when H32 needs to access the service at H51 it needs to send packets with (S=2001:db8:0:a010::32, D=2001:db8:0:5555::51) and those packets need to be forward out the link from SERa to ISP-A.

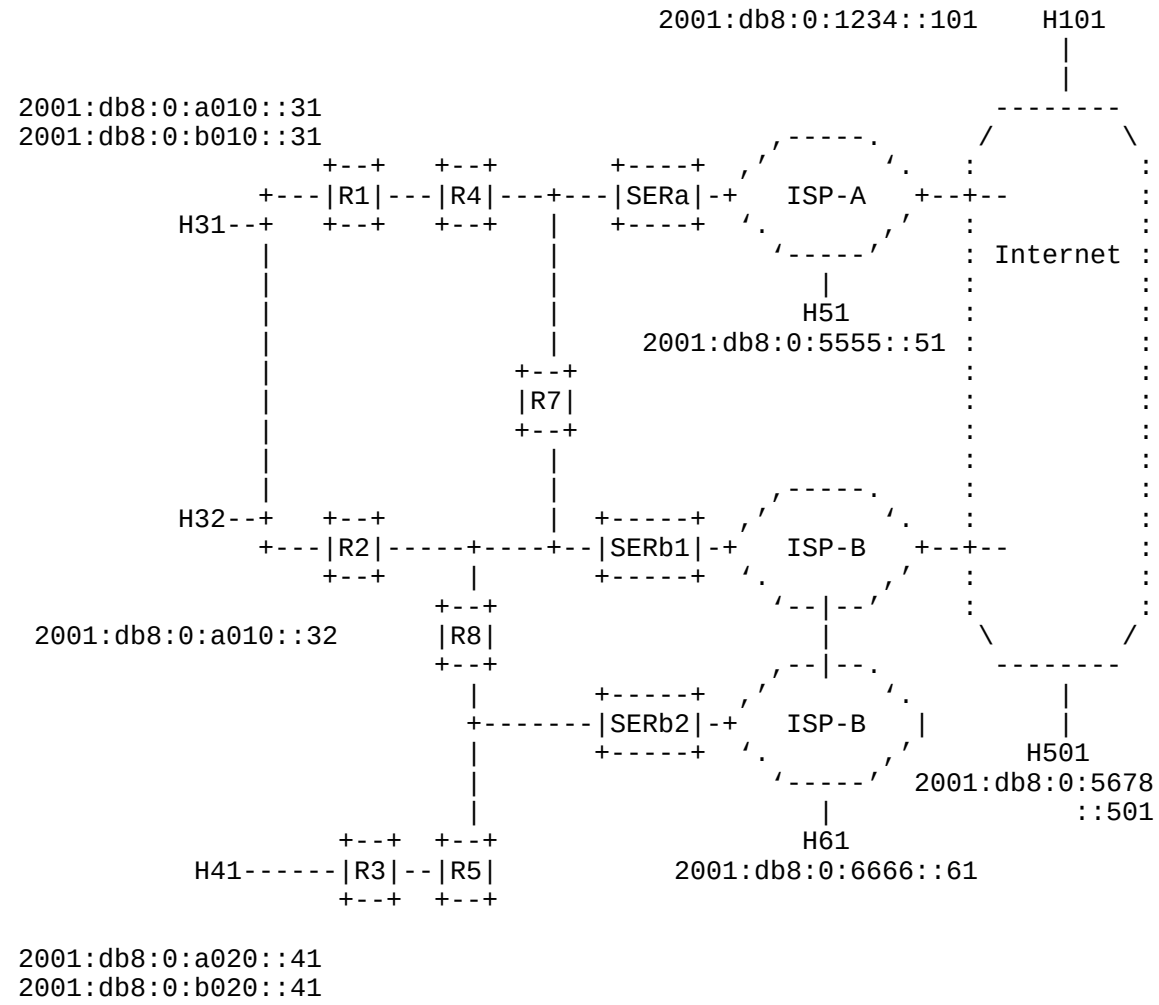


Figure 3: Internet access and services offered by ISP-A and ISP-B

ISP-B illustrates a variation on this scenario. In addition to Internet access, ISP-B also offers a service which requires the site to access host H61. The site has two connections to two different parts of ISP-B (shown as SERb1 and SERb2 in Figure 3). ISP-B expects Internet traffic to use the uplink from SERb1, while it expects it expects traffic destined for the service at H61 to use the uplink from SERb2. For either uplink, ISP-B expects the ingress traffic to have a source address matching the prefix it assigned to the site, 2001:db8:0:b000::/52.

As discussed before, we rely completely on the internal host to set the source address of the packet properly. In the case of a packet sent by H31 to access the service in ISP-B at H61, we expect the packet to have the following addresses: (S=2001:db8:0:b010::31, D=2001:db8:0:6666::61). The routed network has two potential ways of distributing routes so that this packet exits the site on the uplink at SERb2.

We could just rely on normal destination routes, without using source-prefix scoped routes. If we have SERb2 originate a normal unscoped destination route for D=2001:db8:0:6666::/64, the packets from H31 to H61 will exit the site at SERb2 as desired. We should not have to worry about SERa needing to originate the same route, because ISP-B should choose a globally unique prefix for the service at H61.

The alternative is to have SERb2 originate a source-prefix-scoped destination route of the form (S=2001:db8:0:b000::/52, D=2001:db8:0:6666::/64). From a forwarding point of view, the use of the source-prefix-scoped destination route would result in traffic with source addresses corresponding only to ISP-B being sent to SERb2. Instead, the use of the unscoped destination route would result in traffic with source addresses corresponding to ISP-A and ISP-B being sent to SERb2, as long as the destination address matches the destination prefix. It seems like either forwarding behavior would be acceptable.

However, from the point of view of the enterprise network administrator trying to configure, maintain, and trouble-shoot this multihoming solution, it seems much clearer to have SERb2 originate the source-prefix-scoped destination route correspond to the service offered by ISP-B. In this way, all of the traffic leaving the site is determined by the source-prefix-scoped routes, and all of the traffic within the site or arriving from external hosts is determined by the unscoped destination routes. Therefore, for this multihoming solution we choose to originate source-prefix-scoped routes for all traffic leaving the site.

3.5. ISPs and Provider-Assigned Prefixes

While we expect that most site multihoming involves connecting to only two ISPs, this solution allows for connections to an arbitrary number of ISPs to be supported. However, when evaluating scalable implementations of the solution, it would be reasonable to assume that the maximum number of ISPs that a site would connect to is five.

It is also useful to note that the prefixes assigned to the site by different ISPs will not overlap. This must be the case , since the provider-assigned addresses have to be globally unique.

3.6. Simplified Topologies

The topologies of many enterprise sites using this multihoming solution may in practice be simpler than the examples that we have used. The topology in Figure 1 could be further simplified by having all hosts directly connected to the LAN connecting the two site exit routers, SERa and SERb. The topology could also be simplified by having the uplinks to ISP-A and ISP-B both connected to the same site exit router. However, it is the aim of this draft to provide a solution that applies to a broad a range of enterprise site network topologies, so this draft focuses on providing a solution to the more general case. The simplified cases will also be supported by this solution, and there may even be optimizations that can be made for simplified cases. This solution however needs to support more complex topologies.

We are starting with the basic assumption that enterprise site networks can be quite complex from a routing perspective. However, even a complex site network can be multihomed to different ISPs with PA addresses using IPv4 and NAT. It is not reasonable to expect an enterprise network operator to change the routing topology of the site in order to deploy IPv6.

4. Generating Source-Prefix-Scoped Forwarding Tables

So far we have described in general terms how the routers in this solution that are capable of Source Address Dependent Routing will forward traffic using both normal unscoped destination routes and source-prefix-scoped destination routes. Here we give a precise method for generating a source-prefix-scoped forwarding table on a router that supports SADR.

1. Compute the next-hops for the source-prefix-scoped destination prefixes using only routers in the connected SADR domain. These are the initial source-prefix-scoped forwarding table entries.

2. Compute the next-hops for the unscoped destination prefixes using all routers in the IGP. This is the unscoped forwarding table.

3. Augment each less specific source-prefix-scoped forwarding table with all more specific source-prefix-scoped forwarding tables entries based on the following rule. If the destination prefix of the less specific source-prefix-scoped forwarding entry exactly matches the destination prefix of an existing more

specific source-prefix-scoped forwarding entry (including destination prefix length), then do not add the less specific source-prefix-scoped forwarding entry. If the destination prefix does NOT match an existing entry, then add the entry to the more source-prefix-scoped forwarding table. As the unscoped forwarding table is considered to be scoped to ::/0 this process starts with propagating routes from the unscoped forwarding table to source-prefix-scoped forwarding tables and then continues with propagating routes to more-specific-source-prefix-scoped forwarding tables should they exist.

The forward tables produced by this process are used in the following way to forward packets.

1. Select the most specific (longerst prefix match) source-prefix-scoped forwarding table that matches the source address of the packet (again, the unscoped forwarding table is considered to be scoped to ::/0).

2. Look up the destination address of the packet in the selected forwarding table to determine the next-hop for the packet.

The following example illustrates how this process is used to create a forwarding table for each provider-assigned source prefix. We consider the multihomed site network in Figure 3. Initially we assume that all of the routers in the site network support SADR. Figure 4 shows the routes that are originated by the routers in the site network.

```
Routes originated by SERa:
(S=2001:db8:0:a000::/52, D=2001:db8:0:5555/64)
(S=2001:db8:0:a000::/52, D=::/0)
(D=2001:db8:0:5555::/64)
(D=::/0)

Routes originated by SERb1:
(S=2001:db8:0:b000::/52, D=::/0)
(D=::/0)

Routes originated by SERb2:
(S=2001:db8:0:b000::/52, D=2001:db8:0:6666::/64)
(D=2001:db8:0:6666::/64)

Routes originated by R1:
(D=2001:db8:0:a010::/64)
(D=2001:db8:0:b010::/64)

Routes originated by R2:
(D=2001:db8:0:a010::/64)
(D=2001:db8:0:b010::/64)

Routes originated by R3:
(D=2001:db8:0:a020::/64)
(D=2001:db8:0:b020::/64)
```

Figure 4: Routes Originated by Routers in the Site Network

Each SER originates destination routes which are scoped to the source prefix assigned by the ISP that the SER connects to. Note that the SERs also originate the corresponding unscoped destination route. This is not needed when all of the routers in the site support SADR. However, it is required when some routers do not support SADR. This will be discussed in more detail later.

We focus on how R8 constructs its source-prefix-scoped forwarding tables from these route advertisements. R8 computes the next hops for destination routes which are scoped to the source prefix 2001:db8:0:a000::/52. The results are shown in the first table in Figure 5. (In this example, the next hops are computed assuming that all links have the same metric.) Then, R8 computes the next hops for destination routes which are scoped to the source prefix 2001:db8:0:b000::/52. The results are shown in the second table in Figure 5 . Finally, R8 computes the next hops for the unscoped destination prefixes. The results are shown in the third table in Figure 5.

```
forwarding entries scoped to
source prefix = 2001:db8:0:a000::/52
==========================================
D=2001:db8:0:5555/64      NH=R7
D=::/0                    NH=R7

forwarding entries scoped to
source prefix = 2001:db8:0:b000::/52
==========================================
D=2001:db8:0:6666/64      NH=SERb2
D=::/0                    NH=SERb1

unscoped forwarding entries
==========================================
D=2001:db8:0:a010::/64    NH=R2
D=2001:db8:0:b010::/64    NH=R2
D=2001:db8:0:a020::/64    NH=R5
D=2001:db8:0:b020::/64    NH=R5
D=2001:db8:0:5555::/64    NH=R7
D=2001:db8:0:6666::/64    NH=SERb2
D=::/0                    NH=SERb1
```

Figure 5: Forwarding Entries Computed at R8

The final step is for R8 to augment the less specific source-prefix-scoped forwarding entries with more specific source-prefix-scoped forwarding entries. As unscoped forwarding table is considered being scoped to ::/0 and both 2001:db8:0:a000::/52 and 2001:db8:0:b000::/52 are more specific prefixes of ::/0, the unscoped (scoped to ::/0) forwarding table needs to be augmented with both more specific source-prefix-scoped tables. If an less specific scoped forwarding entry has the exact same destination prefix as an more specific source-prefix-scoped forwarding entry (including destination prefix length), then the more specific source-prefix-scoped forwarding entry wins.

As as an example of how the source scoped forwarding entries are augmented, we consider how the two entries in the first table in Figure 5 (the table for source prefix = 2001:db8:0:a000::/52) are augmented with entries from the third table in Figure 5 (the table of unscoped or scoped for ::/0 forwarding entries). The first four unscoped forwarding entries (D=2001:db8:0:a010::/64, D=2001:db8:0:b010::/64, D=2001:db8:0:a020::/64, and D=2001:db8:0:b020::/64) are not an exact match for any of the existing entries in the forwarding table for source prefix 2001:db8:0:a000::/52. Therefore, these four entries are added to the final forwarding table for source prefix 2001:db8:0:a000::/52. The

result of adding these entries is reflected in first four entries the first table in Figure 6.

The next less specific scoped (scope is ::/0) forwarding table entry is for D=2001:db8:0:5555::/64. This entry is an exact match for the existing entry in the forwarding table for the more specific source prefix 2001:db8:0:a000::/52. Therefore, we do not replace the existing entry with the entry from the unscoped forwarding table. This is reflected in the fifth entry in the first table in Figure 6. (Note that since both scoped and unscoped entries have R7 as the next hop, the result of applying this rule is not visible.)

The next less specific prefix scoped (scope is ::/0) forwarding table entry is for D=2001:db8:0:6666::/64. This entry is not an exact match for any existing entries in the forwarding table for source prefix 2001:db8:0:a000::/52. Therefore, we add this entry. This is reflected in the sixth entry in the first table in Figure 6.

The next less specific prefix scoped (scope is ::/0) forwarding table entry is for D=::/0. This entry is an exact match for the existing entry in the forwarding table for more specific source prefix 2001:db8:0:a000::/52. Therefore, we do not overwrite the existing source-prefix-scoped entry, as can be seen in the last entry in the first table in Figure 6.

```
if source address matches 2001:db8:0:a000::/52
then use this forwarding table
==========================================
D=2001:db8:0:a010::/64    NH=R2
D=2001:db8:0:b010::/64    NH=R2
D=2001:db8:0:a020::/64    NH=R5
D=2001:db8:0:b020::/64    NH=R5
D=2001:db8:0:5555::/64    NH=R7
D=2001:db8:0:6666::/64    NH=SERb2
D=::/0                    NH=R7

else if source address matches 2001:db8:0:b000::/52
then use this forwarding table
==========================================
D=2001:db8:0:a010::/64    NH=R2
D=2001:db8:0:b010::/64    NH=R2
D=2001:db8:0:a020::/64    NH=R5
D=2001:db8:0:b020::/64    NH=R5
D=2001:db8:0:5555::/64    NH=R7
D=2001:db8:0:6666::/64    NH=SERb2
D=::/0                    NH=SERb1

else if source address matches ::/0 use this forwarding table
==========================================
D=2001:db8:0:a010::/64    NH=R2
D=2001:db8:0:b010::/64    NH=R2
D=2001:db8:0:a020::/64    NH=R5
D=2001:db8:0:b020::/64    NH=R5
D=2001:db8:0:5555::/64    NH=R7
D=2001:db8:0:6666::/64    NH=SERb2
D=::/0                    NH=SERb1
```

Figure 6: Complete Forwarding Tables Computed at R8

The forwarding tables produced by this process at R8 have the desired properties. A packet with a source address in 2001:db8:0:a000::/52 will be forwarded based on the first table in Figure 6. If the packet is destined for the Internet at large or the service at D=2001:db8:0:5555/64, it will be sent to R7 in the direction of SERa. If the packet is destined for an internal host, then the first four entries will send it to R2 or R5 as expected. Note that if this packet has a destination address corresponding to the service offered by ISP-B (D=2001:db8:0:5555::/64), then it will get forwarded to SERb2. It will be dropped by SERb2 or by ISP-B, since it the packet has a source address that was not assigned by ISP-B. However, this is expected behavior. In order to use the service offered by ISP-B, the host needs to originate the packet with a source address assigned by ISP-B.

In this example, a packet with a source address that doesn't match 2001:db8:0:a000::/52 or 2001:db8:0:b000::/52 must have originated from an external host. Such a packet will use the unscoped forwarding table (the last table in Figure 6). These packets will flow exactly as they would in absence of multihoming.

We can also modify this example to illustrate how it supports deployments where not all routers in the site support SADR. Continuing with the topology shown in Figure 3, suppose that R3 and R5 do not support SADR. Instead they are only capable of understanding unscoped route advertisements. The SADR routers in the network will still originate the routes shown in Figure 4. However, R3 and R5 will only understand the unscoped routes as shown in Figure 7.

```
Routes originated by SERa:
(D=2001:db8:0:5555::/64)
(D=::/0)

Routes originated by SERb1:
(D=::/0)

Routes originated by SERb2:
(D=2001:db8:0:6666::/64)

Routes originated by R1:
(D=2001:db8:0:a010::/64)
(D=2001:db8:0:b010::/64)

Routes originated by R2:
(D=2001:db8:0:a010::/64)
(D=2001:db8:0:b010::/64)

Routes originated by R3:
(D=2001:db8:0:a020::/64)
(D=2001:db8:0:b020::/64)
```

Figure 7: Routes Advertisements Understood by Routers that do no Support SADR

With these unscoped route advertisements, R5 will produce the forwarding table shown in Figure 8.

```
forwarding table
=========================================
D=2001:db8:0:a010::/64    NH=R8
D=2001:db8:0:b010::/64    NH=R8
D=2001:db8:0:a020::/64    NH=R3
D=2001:db8:0:b020::/64    NH=R3
D=2001:db8:0:5555::/64    NH=R8
D=2001:db8:0:6666::/64    NH=SERb2
D=::/0                    NH=R8
```

Figure 8: Forwarding Table For R5, Which Doesn't Understand Source-Prefix-Scoped Routes

Any traffic that needs to exit the site will eventually hit a SADR-capable router. Once that traffic enters the SADR-capable domain, then it will not leave that domain until it exits the site. This property is required in order to guarantee that there will not be routing loops involving SADR-capable and non-SADR-capable routers.

Note that the mechanism described here for converting source-prefix-scoped destination prefix routing advertisements into forwarding state is somewhat different from that proposed in [I-D.ietf-rtgwg-dst-src-routing]. The method described in this document is intended to be easy to understand for network enterprise operators while at the same time being functionally correct. Another difference is that the method in this document assumes that source prefix will not overlap. Other differences between the two approaches still need to be understood and reconciled.

An interesting side-effect of deploying SADR is if all routers in a given network support SADR and have a scoped forwarding table, then the unscoped forwarding table can be eliminated which ensures that packets with legitimate source addresses only can leave the network (as there are no scoped forwarding tables for spoofed/bogon source addresses). It would prevent accidental leaks of ULA/reserved/link-local sources to the Internet as well as ensures that no spoofing is possible from the SADR-enabled network.

5. Mechanisms For Hosts To Choose Good Source Addresses In A Multihomed Site

Until this point, we have made the assumption that hosts are able to choose the correct source address using some unspecified mechanism. This has allowed us to just focus on what the routers in a multihomed site network need to do in order to forward packets to the correct ISP based on source address. Now we look at possible mechanisms for hosts to choose the correct source address. We also look at what

role, if any, the routers may play in providing information that helps hosts to choose source addresses.

Any host that needs to be able to send traffic using the uplinks to a given ISP is expected to be configured with an address from the prefix assigned by that ISP. The host will control which ISP is used for its traffic by selecting one of the addresses configured on the host as the source address for outgoing traffic. It is the responsibility of the site network to ensure that a packet with the source address from an ISP is now sent on an uplink to that ISP.

If all of the ISP uplinks are working, the choice of source address by the host may be driven by the desire to load share across ISP uplinks, or it may be driven by the desire to take advantage of certain properties of a particular uplink or ISP. If any of the ISP uplinks is not working, then the choice of source address by the host can determine if packets get dropped.

How a host should make good decisions about source address selection in a multihomed site is not a solved problem. We do not attempt to solve this problem in this document. Instead we discuss the current state of affairs with respect to standardized solutions and implementation of those solutions. We also look at proposed solutions for this problem.

An external host initiating communication with a host internal to a PA multihomed site will need to know multiple addresses for that host in order to communicate with it using different ISPs to the multihomed site. These addresses are typically learned through DNS. (For simplicity, we assume that the external host is single-homed.) The external host chooses the ISP that will be used at the remote multihomed site by setting the destination address on the packets it transmits. For a sessions originated from an external host to an internal host, the choice of source address used by the internal host is simple. The internal host has no choice but to use the destination address in the received packet as the source address of the transmitted packet.

For a session originated by a host internal to the multi-homed site, the decision of what source address to select is more complicated. We consider three main methods for hosts to get information about the network. The two proactive methods are Neighbor Discovery Router Advertisements and DHCPv6. The one reactive method we consider is ICMPv6. Note that we are explicitly excluding the possibility of having hosts participate in or even listen directly to routing protocol advertisements.

First we look at how a host is currently expected to select the source and destination address with which it sends a packet.

5.1. Source Address Selection Algorithm on Hosts

[RFC6724] defines the algorithms that hosts are expected to use to select source and destination addresses for packets. It defines an algorithm for selecting a source address and a separate algorithm for selecting a destination address. Both of these algorithms depend on a policy table. [RFC6724] defines a default policy which produces certain behavior.

The rules in the two algorithms in [RFC6724] depend on many different properties of addresses. While these are needed for understanding how a host should choose addresses in an arbitrary environment, most of the rules are not relevant for understanding how a host should choose among multiple source addresses in mulitihomed envinronment when sending a packet to a remote host. Returning to the example in Figure 3, we look at what the default algorithms in [RFC6724] say about the source address that internal host H31 should use to send traffic to external host H101, somewhere on the Internet. Let’s look at what rules in [RFC6724] are actually used by H31 in this case.

There is no choice to be made with respect to destination address. H31 needs to send a packet with D=2001:db8:0:1234::101 in order to reach H101. So H31 have to choose between using S=2001:db8:0:a010::31 or S=2001:db8:0:b010::31 as the source address for this packet. We go through the rules for source address selection in Section 5 of [RFC6724]. Rule 1 (Prefer same address) is not useful to break the tie between source addresses, because neither the candidate source addresses equals the destination address. Rule 2 (Prefer appropriate scope) is also not used in this scenario, because both source addresses and the destination address have global scope.

Rule 3 (Avoid deprecated addresses) applies to an address that has been autoconfigured by a host using stateless address autoconfiguration as defined in [RFC4862]. An address autoconfigured by a host has a preferred lifetime and a valid lifetime. The address is preferred until the preferred lifetime expires, after which it becomes deprecated. A deprecated address is not used if there is a preferred address of the appropriate scope available. When the valid lifetime expires, the address cannot be used at all. The preferred and valid lifetimes for an autoconfigured address are set based on the corresponding lifetimes in the Prefix Information Option in Neighbor Discovery Router Advertisements. So a possible tool to control source address selection in this scenario would be for a host to make an address deprecated by having routers on that link, R1 and

R2 in Figure 3, send a Router Advertisement message contaning a Prefix Information Option for the source prefix to be discouraged (or prohibited) with the preferred lifetime set to zero. This is a rather blunt tool, because it discourages or prohibits the use of that source prefix for all destinations. However, it may be useful in some scenarios. For example, if all uplinks to a particular ISP fail, it is desirable to prevent hosts from using source addresses from that ISP address space.

Rule 4 (Avoid home addresses) does not apply here because we are not considering Mobile IP.

Rule 5 (Prefer outgoing interface) is not useful in this scenario, because both source addresses are assigned to the same interface.

Rule 5.5 (Prefer addresses in a prefix advertised by the next-hop) is not useful in the scenario when both R1 and R2 will advertise both source prefixes. However potentially this rule may allow a host to select the correct source prefix by selecting a next-hop. The most obvious way would be to make R1 to advertise itself as a default router and send PIO for 2001:db8:0:a010::/64, while R2 is advertising itself as a default router and sending PIO for 2001:db8:0:b010::/64. We'll discuss later how Rule 5.5 can be used to influence a source address selection in single-router topologies (e.g. when H41 is sending traffic using R3 as a default gateway).

Rule 6 (Prefer matching label) refers to the Label value determined for each source and destination prefix as a result of applying the policy table to the prefix. With the default policy table defined in Section 2.1 of [RFC6724], Label(2001:db8:0:a010::31) = 5, Label(2001:db8:0:b010::31) = 5, and Label(2001:db8:0:1234::101) = 5. So with the default policy, Rule 6 does not break the tie. However, the algorithms in [RFC6724] are defined in such as way that non-default address selection policy tables can be used. [RFC7078] defines a way to distribute a non-default address selection policy table to hosts using DHCPv6. So even though the application of rule 6 to this scenario using the default policy table is not useful, rule 6 may still be a useful tool.

Rule 7 (Prefer temporary addresses) has to do with the technique described in [RFC4941] to periodically randomize the interface portion of an IPv6 address that has been generated using stateless address autoconfiguration. In general, if H31 were using this technique, it would use it for both source addresses, for example creating temporary addresses 2001:db8:0:a010:2839:9938:ab58:830f and 2001:db8:0:b010:4838:f483:8384:3208, in addition to 2001:db8:0:a010::31 and 2001:db8:0:b010::31. So this rule would

prefer the two temporary addresses, but it would not break the tie between the two source prefixes from ISP-A and ISP-B.

Rule 8 (Use longest matching prefix) dictates that between two candidate source addresses the one which has longest common prefix length with the destination address. For example, if H31 were selecting the source address for sending packets to H101, this rule would not be a tie breaker as for both candidate source addresses 2001:db8:0:a101::31 and 2001:db8:0:b101::31 the common prefix length with the destination is 48. However if H31 were selecting the source address for sending packets H41 address 2001:db8:0:a020::41, then this rule would result in using 2001:db8:0:a101::31 as a source (2001:db8:0:a101::31 and 2001:db8:0:a020::41 share the common prefix 2001:db8:0:a000::/58, while for '2001:db8:0:b101::31 and 2001:db8:0:a020::41 the common prefix is 2001:db8:0:a000::/51). Therefore rule 8 might be useful for selecting the correct source address in some but not all scenarios (for example if ISP-B services belong to 2001:db8:0:b000::/59 then H31 would always use 2001:db8:0:b010::31 to access those destinations).

So we can see that of the 8 source selection address rules from [RFC6724], five actually apply to our basic site multihoming scenario. The rules that are relevant to this scenario are summarized below.

- Rule 3: Avoid deprecated addresses.
- Rule 5.5: Prefer addresses in a prefix advertised by the next-hop.
- Rule 6: Prefer matching label.
- Rule 8: Prefer longest matching prefix.

The two methods that we discuss for controlling the source address selection through the four relevant rules above are SLAAC Router Advertisement messages and DHCPv6.

We also consider a possible role for ICMPv6 for getting traffic-driven feedback from the network. With the source address selection algorithm discussed above, the goal is to choose the correct source address on the first try, before any traffic is sent. However, another strategy is to choose a source address, send the packet, get feedback from the network about whether or not the source address is correct, and try another source address if it is not.

We consider four scenarios where a host needs to select the correct source address. The first is when both uplinks are working. The second is when one uplink has failed. The third one is a situation

when one failed uplink has recovered. The last one is failure of both (all) uplinks.

5.2. Selecting Source Address When Both Uplinks Are Working

Again we return to the topology in Figure 3. Suppose that the site administrator wants to implement a policy by which all hosts need to use ISP-A to reach H01 at D=2001:db8:0:1234::101. So for example, H31 needs to select S=2001:db8:0:a010::31.

5.2.1. Distributing Address Selection Policy Table with DHCPv6

This policy can be implemented by using DHCPv6 to distribute an address selection policy table that assigns the same label to destination address that match 2001:db8:0:1234::/64 as it does to source addresses that match 2001:db8:0:a000::/52. The following two entries accomplish this.

Prefix	Precedence	Label
2001:db8:0:1234::/64	50	33
2001:db8:0:a000::/52	50	33

Figure 9: Policy table entries to implement a routing policy

This requires that the hosts implement [RFC6724], the basic source and destination address framework, along with [RFC7078], the DHCPv6 extension for distributing a non-default policy table. Note that it does NOT require that the hosts use DHCPv6 for address assignment. The hosts could still use stateless address autoconfiguration for address configuration, while using DHCPv6 only for policy table distribution (see [RFC3736]). However this method has a number of disadvantages:

- DHCPv6 support is not a mandatory requirement for IPv6 hosts, so this method might not work for all devices.

- Network administrators are required to explicitly configure the desired network access policies on DHCPv6 servers. While it might be feasible in the scenarion of a single multihomed network, such approach might have some scalability issues, especially if the centralized DHCPv6 solution is deployed to serve a large number of multiomed sites.

5.2.2. Controlling Source Address Selection With Router Advertisements

Neighbor Discovery currently has two mechanisms to communicate prefix information to hosts. The base specification for Neighbor Discovery (see [RFC4861]) defines the Prefix Information Option (PIO) in the

Router Advertisement (RA) message. When a host receives a PIO with the A-flag set, it can use the prefix in the PIO as source prefix from which it assigns itself an IP address using stateless address autoconfiguration (SLAAC) procedures described in [RFC4862]. In the example of Figure 3, if the site network is using SLAAC, we would expect both R1 and R2 to send RA messages with PIOs for both source prefixes 2001:db8:0:a010::/64 and 2001:db8:0:b010::/64 with the A-flag set. H31 would then use the SLAAC procedure to configure itself with the 2001:db8:0:a010::31 and 2001:db8:0:b010::31.

Whereas a host learns about source prefixes from PIO messages, hosts can learn about a destination prefix from a Router Advertisement containing Route Information Option (RIO), as specified in [RFC4191]. The destination prefixes in RIOs are intended to allow a host to choose the router that it uses as its first hop to reach a particular destination prefix.

As currently standardized, neither PIO nor RIO options contained in Neighbor Discovery Router Advertisements can communicate the information needed to implement the desired routing policy. PIO's communicate source prefixes, and RIO communicate destination prefixes. However, there is currently no standardized way to directly associate a particular destination prefix with a particular source prefix.

[I-D.pfister-6man-sadr-ra] proposes a Source Address Dependent Route Information option for Neighbor Discovery Router Advertisements which would associate a source prefix and with a destination prefix. The details of [I-D.pfister-6man-sadr-ra] might need tweaking to address this use case. However, in order to be able to use Neighbor Discovery Router Advertisements to implement this routing policy, an extension that allows a R1 and R2 to explicitly communicate to H31 an association between S=2001:db8:0:a000::/52 D=2001:db8:0:1234::/64 would be needed.

However, Rule 5.5 of the source address selection algorithm (discussed in Section 5.1 above), together with default router preference (specified in [RFC4191]) and RIO can be used to influence a source address selection on a host as described below. Let's look at source address selection on the host H41. It receives RAs from R3 with PIOs for 2001:db8:0:a020::/64 and 2001:db8:0:b020::/64. At that point all traffic would use the same next-hop (R3 link-local address) so Rule 5.5 does not apply. Now let's assume that R3 supports SADR and has two scoped forwarding tables, one scoped to S=2001:db8:0:a000::/52 and another scoped to S=2001:db8:0:b000::/52. If R3 generates two different link-local addresses for its interface facing H41 (one for each scoped forwarding table, LLA_A and LLA_B) and starts sending two different RAs: one is sent from LLA_A and

includes PIO for 2001:db8:0:a020::/64, another us sent from LLA_B and includes PIO for 2001:db8:0:b020::/64. Now it is possible to influence H41 source address selection for destinations which follow the default route by setting default router preference in RAs. If it is desired that H41 reaches H101 (or any destinations in the Internet) via ISP-A, then RAs sent from LLA_A should have default router preference set to 01 (high priority), while RAs sent from LLA_B should have preference set to 11 (low). Then LLA_A would be chosen as a next-hop for H101 and therefore (as per rule 5.5) 2001:db8:0:a020::41 would be selected as the source address. If, at the same time, it is desired that H61 is accessible via ISP-B then R3 should include a RIO for 2001:db8:0:6666::/64 to its RA sent from LLA_B. H41 would chose LLA_B as a next-hop for all traffic to H61 and then as per Rule 5.5, 2001:db8:0:b020::41 would be selected as a source address.

If in the above mentioned scenario it is desirable that all Internet traffic leaves the network via ISP-A and the link to ISP-B is used for accessing ISP-B services only (not as ISP-A link backup), then RAs sent by R3 from LLA_B should have Router Lifetime set to 0 and should include RIOs for ISP-B address space. It would instruct H41 to use LLA_A for all Internet traffic but use LLA_B as a next-hop while sending traffic to ISP-B addresses.

The description of the mechanism above assumes SADR support by the first-hop routers as well as SERs. However, a first-hop router can still provide a less flexible version of this mechanism even without implementing SADR. This could be done by providing configuration knobs on the first-hop router that allow it to generate different link-local addresses and to send individual RAs for each prefix.

The mechanism described above relies on Rule 5.5 of the default source address selection algorithm defined in [RFC6724]. [RFC8028] recommends that a host SHOULD select default routers for each prefix in which it is assigned an address. It also recommends that hosts SHOULD implement Rule 5.5. of [RFC6724]. Hosts following the recommendations specified in [RFC8028] therefore should be able to benefit from the solution described in this document. No standards need to be updated in regards to host behavior.

5.2.3. Controlling Source Address Selection With ICMPv6

We now discuss how one might use ICMPv6 to implement the routing policy to send traffic destined for H101 out the uplink to ISP-A, even when uplinks to both ISPs are working. If H31 started sending traffic to H101 with S=2001:db8:0:b010::31 and D=2001:db8:0:1234::101, it would be routed through SER-b1 and out the uplink to ISP-B. SERb1 could recognize that this is traffic is not

following the desired routing policy and react by sending an ICMPv6 message back to H31.

In this example, we could arrange things so that SERb1 drops the packet with S=2001:db8:0:b010::31 and D=2001:db8:0:1234::101, and then sends to H31 an ICMPv6 Destination Unreachable message with Code 5 (Source address failed ingress/egress policy). When H31 receives this packet, it would then be expected to try another source address to reach the destination. In this example, H31 would then send a packet with S=2001:db8:0:a010::31 and D=2001:db8:0:1234::101, which will reach SERa and be forwarded out the uplink to ISP-A.

However, we would also want it to be the case that SERb1 does not enforce this routing policy when the uplink from SERa to ISP-A has failed. This could be accomplished by having SERa originate a source-prefix-scoped route for (S=2001:db8:0:a000::/52, D=2001:db8:0:1234::/64) and have SERb1 monitor the presence of that route. If that route is not present (because SERa has stopped originating it), then SERb1 will not enforce the routing policy, and it will forward packets with S=2001:db8:0:b010::31 and D=2001:db8:0:1234::101 out its uplink to ISP-B.

We can also use this source-prefix-scoped route originated by SERa to communicate the desired routing policy to SERb1. We can define an EXCLUSIVE flag to be advertised together with the IGP route for (S=2001:db8:0:a000::/52, D=2001:db8:0:1234::/64). This would allow SERa to communicate to SERb that SERb should reject traffic for D=2001:db8:0:1234::/64 and respond with an ICMPv6 Destination Unreachable Code 5 message, as long as the route for (S=2001:db8:0:a000::/52, D=2001:db8:0:1234::/64) is present.

Finally, if we are willing to extend ICMPv6 to support this solution, then we could create a mechanism for SERb1 to tell the host what source address it should be using to successfully forward packets that meet the policy. In its current form, when SERb1 sends an ICMPv6 Destination Unreachable Code 5 message, it is basically saying, "This source address is wrong. Try another source address." In the absence of a clear indication which address to try next, the host will iterate over all addresses assigned to the interface (e.g. various privacy addresses) which would lead to significant delays and degraded user experience. It would be better is if the ICMPv6 message could say, "This source address is wrong. Instead use a source address in S=2001:db8:0:a000::/52.".

However using ICMPv6 for signalling source address information back to hosts introduces new challenges. Most routers currently have software or hardware limits on generating ICMP messages. An site administrator deploying a solution that relies on the SERs generating

ICMP messages could try to improve the performance of SERs for generating ICMP messages. However, in a large network, it is still likely that ICMP message generation limits will be reached. As a result hosts would not receive ICMPv6 back which in turn leads to traffic blackholing and poor user experience. To improve the scalability of ICMPv6-based signalling hosts SHOULD cache the preferred source address (or prefix) for the given destination (which in turn might cause issues in case of the corresponding ISP uplinks failure - see Section 5.3). In addition, the same source prefix SHOULD be used for other destinations in the same /64 as the original destination address. The source prefix SHOULD have a specific lifetime. Expiration of the lifetime SHOULD trigger the source address selection algorithm again.

Using ICMPv6 Code 5 message for influencing source address selection allows an attacker to exhaust the list of candidate source addresses on the host by sending spoofed ICMPv6 Code 5 for all prefixes known on the network (therefore preventing a victim from establishing a communication with the destination host). To protect from such attack hosts SHOULD verify that the original packet header included into ICMPv6 error message was actually sent by the host.

As currently standardized in [RFC4443], the ICMPv6 Destination Unreachable Message with Code 5 would allow for the iterative approach to retransmitting packets using different source addresses. As currently defined, the ICMPv6 message does not provide a mechanism to communication information about which source prefix should be used for a retransmitted packet. The current document does not define such a mechanism. However, we note that this might be a useful extension to define in a different document.

5.2.4. Summary of Methods For Controlling Source Address Selection To Implement Routing Policy

So to summarize this section, we have looked at three methods for implementing a simple routing policy where all traffic for a given destination on the Internet needs to use a particular ISP, even when the uplinks to both ISPs are working.

The default source address selection policy cannot distinguish between the source addresses needed to enforce this policy, so a non-default policy table using associating source and destination prefixes using Label values would need to be installed on each host. A mechanism exists for DHCPv6 to distribute a non-default policy table but such solution would heavily rely on DHCPv6 support by host operating system. Moreover there is no mechanism to translate desired routing/traffic engineering policies into policy tables on

DHCPv6 servers. Therefore using DHCPv6 for controlling address selection policy table is not recommended and SHOULD NOT be used.

At the same time Router Advertisements provide a reliable mechanism to influence source address selection process via PIO, RIO and default router preferences. As all those options have been standardized by IETF and are supported by various operating systems, no changes are required on hosts. First-hop routers in the enterprise network need to be able of sending different RAs for different SLAAC prefixes (either based on scoped forwarding tables or based on pre-configured policies).

SERs can enforce the routing policy by sending ICMPv6 Destination Unreachable messages with Code 5 (Source address failed ingress/ egress policy) for traffic that is being sent with the wrong source address. The policy distribution can be automated by defining an EXCLUSIVE flag for the source-prefix-scoped route which can be set on the SER that originates the route. As ICMPv6 message generation can be rate-limited on routers, it SHOULD NOT be used as the only mechanism to influence source address selection on hosts. While hosts SHOULD select the correct source address for a given destination the network SHOULD signal any source address issues back to hosts using ICMPv6 error messages.

5.3. Selecting Source Address When One Uplink Has Failed

Now we discuss if DHCPv6, Neighbor Discovery Router Advertisements, and ICMPv6 can help a host choose the right source address when an uplink to one of the ISPs has failed. Again we look at the scenario in Figure 3. This time we look at traffic from H31 destined for external host H501 at D=2001:db8:0:5678::501. We initially assume that the uplink from SERa to ISP-A is working and that the uplink from SERb1 to ISP-B is working.

We assume there is no particular routing policy desired, so H31 is free to send packets with S=2001:db8:0:a010::31 or S=2001:db8:0:b010::31 and have them delivered to H501. For this example, we assume that H31 has chosen S=2001:db8:0:b010::31 so that the packets exit via SERb to ISP-B. Now we see what happens when the link from SERb1 to ISP-B fails. How should H31 learn that it needs to start sending the packet to H501 with S=2001:db8:0:a010::31 in order to start using the uplink to ISP-A? We need to do this in a way that doesn’t prevent H31 from still sending packets with S=2001:db8:0:b010::31 in order to reach H61 at D=2001:db8:0:6666::61.

5.3.1. Controlling Source Address Selection With DHCPv6

For this example we assume that the site network in Figure 3 has a centralized DHCP server and all routers act as DHCP relay agents. We assume that both of the addresses assigned to H31 were assigned via DHCP.

We could try to have the DHCP server monitor the state of the uplink from SERb1 to ISP-B in some manner and then tell H31 that it can no longer use S=2001:db8:0:b010::31 by settings its valid lifetime to zero. The DHCP server could initiate this process by sending a Reconfigure Message to H31 as described in Section 19 of [RFC3315]. Or the DHCP server can assign addresses with short lifetimes in order to force clients to renew them often.

This approach would prevent H31 from using S=2001:db8:0:b010::31 to reach the a host on the Internet. However, it would also prevent H31 from using S=2001:db8:0:b010::31 to reach H61 at D=2001:db8:0:6666::61, which is not desirable.

Another potential approach is to have the DHCP server monitor the uplink from SERb1 to ISP-B and control the choice of source address on H31 by updating its address selection policy table via the mechanism in [RFC7078]. The DHCP server could initiate this process by sending a Reconfigure Message to H31. Note that [RFC3315] requires that Reconfigure Message use DHCP authentication. DHCP authentication could be avoided by using short address lifetimes to force clients to send Renew messages to the server often. If the host is not obtaining its IP addresses from the DHCP server, then it would need to use the Information Refresh Time option defined in [RFC4242].

If the following policy table can be installed on H31 after the failure of the uplink from SERb1, then the desired routing behavior should be achieved based on source and destination prefix being matched with label values.

Prefix	Precedence	Label
::/0	50	44
2001:db8:0:a000::/52	50	44
2001:db8:0:6666::/64	50	55
2001:db8:0:b000::/52	50	55

Figure 10: Policy Table Needed On Failure Of Uplink From SERb1

The described solution has a number of significant drawbacks, some of them already discussed in Section 5.2.1.

- DHCPv6 support is not required for an IPv6 host and there are operating systems which do not support DHCPv6. Besides that, it does not appear that [RFC7078] has been widely implemented on host operating systems.

- [RFC7078] does not clearly specify this kind of a dynamic use case where address selection policy needs to be updated quickly in response to the failure of a link. In a large network it would present scalability issues as many hosts need to be reconfigured in very short period of time.

- Updating DHCPv6 server configuration each time an ISP uplink changes its state introduces some scalability issues, especially for mid/large distributed scale enterprise networks. In addition to that, the policy table needs to be manually configured by administrators which makes that solution prone to human error.

- No mechanism exists for making DHCPv6 servers aware of network topology/routing changes in the network. In general DHCPv6 servers monitoring network-related events sounds like a bad idea as completely new functionality beyond the scope of DHCPv6 role is required.

5.3.2. Controlling Source Address Selection With Router Advertisements

The same mechanism as discussed in Section 5.2.2 can be used to control the source address selection in the case of an uplink failure. If a particular prefix should not be used as a source for any destinations, then the router needs to send RA with Preferred Lifetime field for that prefix set to 0.

Let’s consider a scenario when all uplinks are operational and H41 receives two different RAs from R3: one from LLA_A with PIO for 2001:db8:0:a020::/64, default router preference set to 11 (low) and another one from LLA_B with PIO for 2001:db8:0:a020::/64, default router preference set to 01 (high) and RIO for 2001:db8:0:6666::/64. As a result H41 is using 2001:db8:0:b020::41 as a source address for all Internet traffic and those packets are sent by SERs to ISP-B. If SERb1 uplink to ISP-B failed, the desired behavior is that H41 stops using 2001:db8:0:b020::41 as a source address for all destinations but H61. To achieve that R3 should react to SERb1 uplink failure (which could be detected as the scoped route (S=2001:db8:0:b000::/52, D=::/0) disappearance) by withdrawing itself as a default router. R3 sends a new RA from LLA_B with Router Lifetime value set to 0 (which means that it should not be used as default router). That RA still contains PIO for 2001:db8:0:b020::/64 (for SLAAC purposes) and RIO for 2001:db8:0:6666::/64 so H41 can reach H61 using LLA_B as a next-hop and 2001:db8:0:b020::41 as a source address. For all traffic

following the default route, LLA_A will be used as a next-hop and 2001:db8:0:a020::41 as a source address.

If all uplinks to ISP-B have failed and therefore source addresses from ISP-B address space should not be used at all, the forwarding table scoped S=2001:db8:0:b000::/52 contains no entries. Hosts can be instructed to stop using source addresses from that block by sending RAs containing PIO with Preferred Lifetime set to 0.

5.3.3. Controlling Source Address Selection With ICMPv6

Now we look at how ICMPv6 messages can provide information back to H31. We assume again that at the time of the failure H31 is sending packets to H501 using (S=2001:db8:0:b010::31, D=2001:db8:0:5678::501). When the uplink from SERb1 to ISP-B fails, SERb1 would stop originating its source-prefix-scoped route for the default destination (S=2001:db8:0:b000::/52, D=::/0) as well as its unscoped default destination route. With these routes no longer in the IGP, traffic with (S=2001:db8:0:b010::31, D=2001:db8:0:5678::501) would end up at SERa based on the unscoped default destination route being originated by SERa. Since that traffic has the wrong source address to be forwarded to ISP-A, SERa would drop it and send a Destination Unreachable message with Code 5 (Source address failed ingress/egress policy) back to H31. H31 would then know to use another source address for that destination and would try with (S=2001:db8:0:a010::31, D=2001:db8:0:5678::501). This would be forwarded to SERa based on the source-prefix-scoped default destination route still being originated by SERa, and SERa would forward it to ISP-A. As discussed above, if we are willing to extend ICMPv6, SERa can even tell H31 what source address it should use to reach that destination. The expected host behaviour has been discussed in Section 5.2.3. Potential issue with using ICMPv6 for signalling source address issues back to hosts is that uplink to an ISP-B failure immediately invalidates source addresses from 2001:db8:0:b000::/52 for all hosts which triggers a large number of ICMPv6 being sent back to hosts - the same scalability/rate limiting issues discussed in Section 5.2.3 would apply.

5.3.4. Summary Of Methods For Controlling Source Address Selection On The Failure Of An Uplink

It appears that DHCPv6 is not particularly well suited to quickly changing the source address used by a host in the event of the failure of an uplink, which eliminates DHCPv6 from the list of potential solutions. On the other hand Router Advertisements provides a reliable mechanism to dynamically provide hosts with a list of valid prefixes to use as source addresses as well as prevent particular prefixes to be used. While no additional new features are

required to be implemented on hosts, routers need to be able to send RAs based on the state of scoped forwarding tables entries and to react to network topology changes by sending RAs with particular parameters set.

The use of ICMPv6 Destination Unreachable messages generated by the SER (or any SADR-capable) routers seem like they have the potential to provide a support mechanism together with RAs to signal source address selection errors back to hosts, however scalability issues may arise in large networks in case of sudden topology change. Therefore it is highly desirable that hosts are able to select the correct source address in case of uplinks failure with ICMPv6 being an additional mechanism to signal unexpected failures back to hosts.

The current behavior of different host operating system when receiving ICMPv6 Destination Unreachable message with code 5 (Source address failed ingress/egress policy) is not clear to the authors. Information from implementers, users, and testing would be quite helpful in evaluating this approach.

5.4. Selecting Source Address Upon Failed Uplink Recovery

The next logical step is to look at the scenario when a failed uplink on SERb1 to ISP-B is coming back up, so hosts can start using source addresses belonging to 2001:db8:0:b000::/52 again.

5.4.1. Controlling Source Address Selection With DHCPv6

The mechanism to use DHCPv6 to instruct the hosts (H31 in our example) to start using prefixes from ISP-B space (e.g. S=2001:db8:0:b010::31 for H31) to reach hosts on the Internet is quite similar to one discussed in Section 5.3.1 and shares the same drawbacks.

5.4.2. Controlling Source Address Selection With Router Advertisements

Let’s look at the scenario discussed in Section 5.3.2. If the uplink(s) failure caused the complete withdrawal of prefixes from 2001:db8:0:b000::/52 address space by setting Preferred Lifetime value to 0, then the recovery of the link should just trigger new RA being sent with non-zero Preferred Lifetime. In another scenario discussed in Section 5.3.2, the SERb1 uplink to ISP-B failure leads to disappearance of the (S=2001:db8:0:b000::/52, D=::/0) entry from the forwarding table scoped to S=2001:db8:0:b000::/52 and, in turn, caused R3 to send RAs from LLA_B with Router Lifetime set to 0. The recovery of the SERb1 uplink to ISP-B leads to (S=2001:db8:0:b000::/52, D=::/0) scoped forwarding entry re-appearance and instructs R3 that it should advertise itself as a

default router for ISP-B address space domain (send RAs from LLA_B with non-zero Router Lifetime).

5.4.3. Controlling Source Address Selection With ICMP

It looks like ICMPv6 provides a rather limited functionality to signal back to hosts that particular source addresses have become valid again. Unless the changes in the uplink state a particular (S,D) pair, hosts can keep using the same source address even after an ISP uplink has come back up. For example, after the uplink from SERb1 to ISP-B had failed, H31 received ICMPv6 Code 5 message (as described in Section 5.3.3) and allegedly started using (S=2001:db8:0:a010::31, D=2001:db8:0:5678::501) to reach H501. Now when the SERb1 uplink comes back up, the packets with that (S,D) pair are still routed to SERa1 and sent to the Internet. Therefore H31 is not informed that it should stop using 2001:db8:0:a010::31 and start using 2001:db8:0:b010::31 again. Unless SERa has a policy configured to drop packets (S=2001:db8:0:a010::31, D=2001:db8:0:5678::501) and send ICMPv6 back if SERb1 uplink to ISP-B is up, H31 will be unaware of the network topology change and keep using S=2001:db8:0:a010::31 for Internet destinations, including H51.

One of the possible option may be using a scoped route with EXCLUSIVE flag as described in Section 5.2.3. SERa1 uplink recovery would cause (S=2001:db8:0:a000::/52, D=2001:db8:0:1234::/64) route to reappear in the routing table. In the absence of that route packets to H101 which were sent to ISP-B (as ISP-A uplink was down) with source addresses from 2001:db8:0:b000::/52. When the route re-appears SERb1 would reject those packets and sends ICMPv6 back as discussed in Section 5.2.3. Practically it might lead to scalability issues which have been already discussed in Section 5.2.3 and Section 5.4.3.

5.4.4. Summary Of Methods For Controlling Source Address Selection Upon Failed Uplink Recovery

Once again DHCPv6 does not look like reasonable choice to manipulate source address selection process on a host in the case of network topology changes. Using Router Advertisement provides the flexible mechanism to dynamically react to network topology changes (if routers are able to use routing changes as a trigger for sending out RAs with specific parameters). ICMPv6 could be considered as a supporting mechanism to signal incorrect source address back to hosts but should not be considered as the only mechanism to control the address selection in multihomed environments.

5.5. Selecting Source Address When All Uplinks Failed

One particular tricky case is a scenario when all uplinks have failed. In that case there is no valid source address to be used for any external destinations while it might be desirable to have intra-site connectivity.

5.5.1. Controlling Source Address Selection With DHCPv6

From DHCPv6 perspective uplinks failure should be treated as two independent failures and processed as described in Section 5.3.1. At this stage it is quite obvious that it would result in quite complicated policy table which needs to be explicitly configured by administrators and therefore seems to be impractical.

5.5.2. Controlling Source Address Selection With Router Advertisements

As discussed in Section 5.3.2 an uplink failure causes the scoped default entry to disappear from the scoped forwarding table and triggers RAs with zero Router Lifetime. Complete disappearance of all scoped entries for a given source prefix would cause the prefix being withdrawn from hosts by setting Preferred Lifetime value to zero in PIO. If all uplinks (SERa, SERb1 and SERb2) failed, hosts either lost their default routers and/or have no global IPv6 addresses to use as a source. (Note that 'uplink failure' might mean 'IPv6 connectivity failure with IPv4 still being reachable', in which case hosts might fall back to IPv4 if there is IPv4 connectivity to destinations). As a results intra-site connectivity is broken. One of the possible way to solve it is to use ULAs.

All hosts have ULA addresses assigned in addition to GUAs and used for intra-site communication even if there is no GUA assigned to a host. To avoid accidental leaking of packets with ULA sources SADR-capable routers SHOULD have a scoped forwarding table for ULA source for internal routes but MUST NOT have an entry for D=::/0 in that table. In the absence of (S=ULA_Prefix; D=::/0) first-hop routers will send dedicated RAs from a unique link-local source LLA_ULA with PIO from ULA address space, RIO for the ULA prefix and Router Lifetime set to zero. The behaviour is consistent with the situation when SERb1 lost the uplink to ISP-B (so there is no Internet connectivity from 2001:db8:0:b000::/52 sources) but those sources can be used to reach some specific destinations. In the case of ULA there is no Internet connectivity from ULA sources but they can be used to reach another ULA destinations. Note that ULA usage could be particularly useful if all ISPs assign prefixes via DHCP-PD. In the absence of ULAs uplinks failure hosts would lost all their GUAs upon prefix lifetime expiration which again makes intra-site communication impossible.

It should be noted that the Rule 5.5 (prefer a prefix advertised by the selected next-hop) takes precedence over the Rule 6 (prefer matching label, which ensures that GUA source addresses are preferred over ULAs for GUA destinations). Therefore if ULAs are used, the network adminstrator needs to ensure that while the site has an Internet connectivity, hosts do not select a roter which advertises ULA prefixes as their default router.

5.5.3. Controlling Source Address Selection With ICMPv6

In case of all uplinks failure all SERs will drop outgoing IPv6 traffic and respond with ICMPv6 error message. In the large network when many hosts are trying to reach Internet destinations it means that SERs need to generate an ICMPv6 error to every packet they receive from hosts which presents the same scalability issues discussed in Section 5.3.3

5.5.4. Summary Of Methods For Controlling Source Address Selection When All Uplinks Failed

Again, combining SADR with Router Advertisements seems to be the most flexible and scalable way to control the source address selection on hosts.

5.6. Summary Of Methods For Controlling Source Address Selection

To summarize the scenarios and options discussed above:

While DHCPv6 allows administrators to manipulate source address selection policy tables, this method has a number of significant disadvantages which eliminates DHCPv6 from a list of potential solutions:

1. It required hosts to support DHCPv6 and its extension (RFC7078);

2. DHCPv6 server needs to monitor network state and detect routing changes.

3. The use of policy tables requires manual configuration and might be extremely complicated, especially in the case of distributed network when large number of remote sites are being served by centralized DHCPv6 servers.

4. Network topology/routing policy changes could trigger simultaneous re-configuration of large number of hosts which present serious scalability issues.

The use of Router Advertisements to influence the source address selection on hosts seem to be the most reliable, flexible and scalable solution. It has the following benefits:

1. no new (non-standard) functionality needs to be implemented on hosts (except for [RFC4191] support);

2. no changes in RA format;

3. routers can react to routing table changes by sending RAs which would minimize the failover time in the case of network topology changes;

4. information required for source address selection is broadcast to all affected hosts in case of topology change event which improves the scalability of the solution (comparing to DHCPv6 reconfiguration or ICMPv6 error messages).

To fully benefit from the RA-based solution, first-hop routers need to implement SADR and be able to send dedicated RAs per scoped forwarding table as discussed above, reacting to network changes with sending new RAs. It should be noted that the proposed solution would work even if first-hop routers are not SADR-capable but still able to send individual RAs for each ISP prefix and react to topology changes as discussed above (e.g. via configuration knobs).

The RA-based solution relies heavily on hosts correctly implementing default address selection algorith as defined in [RFC6724]. While the basic (and most common) multihoming scenario (two or more Internet uplinks, no ’wall gardens’) would work for any host supporting the minimal implementation of [RFC6724], more complex use cases (such as "wall garden" and other scenarios when some ISP resources can only be reached from that ISP address space) require that hosts support Rule 5.5 of the default address selection algorithm. There is some evidence that not all host OSes have that rule implemented currently. However it should be noted that [RFC8028] states that Rule 5.5 SHOULD be implemented.

ICMPv6 Code 5 error message SHOULD be used to complement RA-based solution to signal incorrect source address selection back to hosts, but it SHOULD NOT be considered as the stand-alone solution. To prevent scenarios when hosts in multihomed envinronments incorrectly identify onlink/offlink destinations, hosts should treat ICMPv6 Redirects as discussed in [RFC8028].

5.7. Other Configuration Parameters

5.7.1. DNS Configuration

In mutihomed envinronment each ISP might provide their own list of DNS servers. E.g. in the topology show on Figure 3, ISP-A might provide recursive DNS server H51 2001:db8:0:5555::51, while ISP-B might provide H61 2001:db8:0:6666::61 as a recursive DNS server. [RFC8106] defines IPv6 Router Advertisement options to allow IPv6 routers to advertise a list of DNS recursive server addresses and a DNS Search List to IPv6 hosts. Using RDNSS together with 'scoped' RAs as described above would allow a first-hop router (R3 in the Figure 3) to send DNS server addresses and search lists provided by each ISP (or the corporate DNS servers addresses if the enterprise is running its own DNS servers).

As discussed in Section 5.5.2, failure of all ISP uplinks would cause deprecaction of all addresses assigned to a host from the address space if all ISPs. If any intra-site IPv6 connectivity is still desirable (most likely to be the case for any mid/large scare network), then ULAs should be used as discussed in Section 5.5.2. In such a scenario, the enterprise network should run its own recursive DNS server(s) and provide its ULA addresses to hosts via RDNSS in RAs send for ULA-scoped forwarding table as described in Section 5.5.2.

There are some scenarios when the final outcome of the name resolution might be different depending on:

- which DNS server is used;
- which source address the client uses to send a DNS query to the server (DNS split horizon).

There is no way currently to instruct a host to use a particular DNS server out of the configured servers list for resolving a particular name. Therefore it does not seem feasible to solve the problem of DNS server selection on the host (it should be noted that this particular issue is protocol-agnostic and happens for IPv4 as well). In such a scenario it is recommended that the enterprise run its own local recursive DNS server.

To influence host source address selection for packets sent to a particular DNS server the following requirements must be met:

- the host supports RIO as defined in [RFC4191];
- the routers send RIO for routes to DNS server addresses.

For example, if it is desirable that host H31 reaches the ISP-A DNS server H51 2001:db8:0:5555::51 using its source address 2001:db8:0:a010::31, then both R1 and R2 should send the RIO containing the route to 2001:db8:0:5555::51 (or covering route) in their 'scoped' RAs, containing LLA_A as the default router address and the PO for SLAAC prefix 2001:db8:0:a010::/64. In that case the host H31 (if it supports the Rule 5.5) would select LLA_A as a next-hop and then chose 2001:db8:0:a010::31 as the source address for packets to the DNS server.

It should be noted that [RFC8106] explicitly prohibits using DNS information if the RA router Lifetime expired: "An RDNSS address or a DNSSL domain name MUST be used only as long as both the RA router Lifetime (advertised by a Router Advertisement message) and the corresponding option Lifetime have not expired.". Therefore hosts might ignore RDNSS information provided in ULA-scoped RAs as those RAs would have router lifetime set to 0. However the updated version of RFC6106 ([RFC8106]) has that requirement removed.

6. Deployment Considerations

The solution described in this document requires certain mechanisms to be supported by the network infrastructure and hosts. It requires some routers in the enterprise site to support some form of Source Address Dependent Routing (SADR). It also requires hosts to be able to learn when the uplink to an ISP chages its state so the corresponding source addresses should (or should not) be used. Ongoing work to create mechanisms to accomplish this are discussed in this document, but they are still a work in progress.

The solution discussed in this document relies on the default address selection algorithm ([RFC6724]) Rule 5.5. While [RFC6724] consideres this rule as optional, the recent [RFC8028] recommends that a host SHOULD select default routers for each prefix in which it is assigned an address. It also recommends that hosts SHOULD implement Rule 5.5. of [RFC6724]. Therefore while RFC8028-compliant hosts already have mechanism to learn about ISP uplinks state changes and selecting the source addresses accordingly, many hosts do not have such mechanism supported yet.

It should be noted that multihomed enterprise network utilizing multipe ISP prefixes can be considered as a typical mutiple provisioning domain (mPVD) scenario, as desribed in [RFC7556]. This document defines a way for network to provide the PVD information to hosts indirectly, using the existing mechanisms. At the same time [I-D.ietf-intarea-provisioning-domains]takes one step further and describes a comprehensive mechanism for hosts to discover the whole set of configuration information associated with different PVD/ISPs.

[I-D.ietf-intarea-provisioning-domains] complements this document in terms of making hosts being able to learn about ISP uplink states and selecting the corresponding source addresses.

7. Other Solutions

7.1. Shim6

The Shim6 working group specified the Shim6 protocol [RFC5533] which allows a host at a multihomed site to communicate with an external host and exchange information about possible source and destination address pairs that they can use to communicate. It also specified the REAP protocol [RFC5534] to detect failures in the path between working address pairs and find new working address pairs. A fundamental requirement for Shim6 is that both internal and external hosts need to support Shim6. That is, both the host internal to the multihomed site and the host external to the multihomed site need to support Shim6 in order for there to be any benefit for the internal host to run Shim6. The Shim6 protocol specification was published in 2009, but it has not been widely implemented. Therefore Shim6 is not considered as a viable solution for enterprise multihoming.

7.2. IPv6-to-IPv6 Network Prefix Translation

IPv6-to-IPv6 Network Prefix Translation (NPTv6) [RFC6296] is not the focus of this document. NPTv6 suffers from the same fundamental issue as any other address translation approaches: it breaks end-to-end connectivity. Therefore NPTv6 is not considered as desirable solution and this document intentionally focuses on solving enterprise multihoming problem without any form of address translations.

With increasing interest and ongoing work in bringing path awareness to transport and application layer protocols hosts migth be able to determine the properties of the various network paths and choose among paths available to them. As selecting the correct source address is one of the possible mechanisms path-aware hosts may utilize, address translation negatively affects hosts path-awareness which makes NTPv6 even more undesirable solution.

7.3. Multipath Transport

Using multipath transport might solve the problems discussed in Section 5 it would allow hosts to use multiple source addresses for a single connection and switch between source addresses when a particular address becomes unavailable or a new address gets assigned to the host interface. Therefore if all hosts in the enterprise network are only using multipath transport for all connections, the

signalling solution described in Section 5 migth not be needed (it should be noted that the Source Address Dependent Routing would still be required to delver packets to the correct uplinks). Unfortunatelt when this document was written, mutlipath transport alone can not be considered a solition for the problem of selecting the source address in a multihomed envinronments. There are significant number of hosts which do not use mulipath transport currently and it seems unlikely that the situation is going to change in any foreseeable future. As the solution for enterprise multihoming needs to work for the least common denominator: hosts without multipath transport support. In addition, not all protocols are using multipath transport. While multipath transport would complement the solution described in Section 5, it could not be considered as a sole solution to the problem of source address selection in multihomed envinronments.

8. IANA Considerations

This memo asks the IANA for no new parameters.

9. Security Considerations

This document introduces no new security or privacy considerations. Security considerations of using stateless address autoconfiguration is discussed in [RFC4862].

10. Acknowledgements

The original outline was suggested by Ole Troan.

The authors would like to thank the following people (in alphabetical order) for their review and feedback: Olivier Bonaventure, Brian E Carpenter, Lorenzo Colitti, David Lamparter, Acee Lindem, Philip Matthewsu, Robert Raszuk, Dave Thaler.

11. References

11.1. Normative References

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <https://www.rfc-editor.org/info/rfc1122>.

[RFC1123] Braden, R., Ed., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, DOI 10.17487/RFC1123, October 1989, <https://www.rfc-editor.org/info/rfc1123>.

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <https://www.rfc-editor.org/info/rfc1918>.

[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>.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <https://www.rfc-editor.org/info/rfc2827>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <https://www.rfc-editor.org/info/rfc3315>.

[RFC3582] Abley, J., Black, B., and V. Gill, "Goals for IPv6 Site-Multihoming Architectures", RFC 3582, DOI 10.17487/RFC3582, August 2003, <https://www.rfc-editor.org/info/rfc3582>.

[RFC4116] Abley, J., Lindqvist, K., Davies, E., Black, B., and V. Gill, "IPv4 Multihoming Practices and Limitations", RFC 4116, DOI 10.17487/RFC4116, July 2005, <https://www.rfc-editor.org/info/rfc4116>.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <https://www.rfc-editor.org/info/rfc4191>.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <https://www.rfc-editor.org/info/rfc4193>.

[RFC4218] Nordmark, E. and T. Li, "Threats Relating to IPv6 Multihoming Solutions", RFC 4218, DOI 10.17487/RFC4218, October 2005, <https://www.rfc-editor.org/info/rfc4218>.

[RFC4219] Lear, E., "Things Multihoming in IPv6 (MULTI6) Developers Should Think About", RFC 4219, DOI 10.17487/RFC4219, October 2005, <https://www.rfc-editor.org/info/rfc4219>.

[RFC4242] Venaas, S., Chown, T., and B. Volz, "Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 4242, DOI 10.17487/RFC4242, November 2005, <https://www.rfc-editor.org/info/rfc4242>.

[RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, DOI 10.17487/RFC6296, June 2011, <https://www.rfc-editor.org/info/rfc6296>.

[RFC7157] Troan, O., Ed., Miles, D., Matsushima, S., Okimoto, T., and D. Wing, "IPv6 Multihoming without Network Address Translation", RFC 7157, DOI 10.17487/RFC7157, March 2014, <https://www.rfc-editor.org/info/rfc7157>.

[RFC7556] Anipko, D., Ed., "Multiple Provisioning Domain Architecture", RFC 7556, DOI 10.17487/RFC7556, June 2015, <https://www.rfc-editor.org/info/rfc7556>.

[RFC8028] Baker, F. and B. Carpenter, "First-Hop Router Selection by Hosts in a Multi-Prefix Network", RFC 8028, DOI 10.17487/RFC8028, November 2016, <https://www.rfc-editor.org/info/rfc8028>.

[RFC8106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 8106, DOI 10.17487/RFC8106, March 2017, <https://www.rfc-editor.org/info/rfc8106>.

[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>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

11.2. Informative References

[I-D.baker-ipv6-isis-dst-src-routing] Baker, F. and D. Lamparter, "IPv6 Source/Destination Routing using IS-IS", draft-baker-ipv6-isis-dst-src-routing-07 (work in progress), July 2017.

[I-D.baker-rtgwg-src-dst-routing-use-cases]
Baker, F., Xu, M., Yang, S., and J. Wu, "Requirements and Use Cases for Source/Destination Routing", draft-baker-rtgwg-src-dst-routing-use-cases-02 (work in progress), April 2016.

[I-D.boutier-babel-source-specific]
Boutier, M. and J. Chroboczek, "Source-Specific Routing in Babel", draft-boutier-babel-source-specific-03 (work in progress), July 2017.

[I-D.huitema-shim6-ingress-filtering]
Huitema, C., "Ingress filtering compatibility for IPv6 multihomed sites", draft-huitema-shim6-ingress-filtering-00 (work in progress), September 2005.

[I-D.ietf-intarea-provisioning-domains]
Pfister, P., Vyncke, E., Pauly, T., Schinazi, D., and W. Shao, "Discovering Provisioning Domain Names and Data", draft-ietf-intarea-provisioning-domains-02 (work in progress), June 2018.

[I-D.ietf-rtgwg-dst-src-routing]
Lamparter, D. and A. Smirnov, "Destination/Source Routing", draft-ietf-rtgwg-dst-src-routing-06 (work in progress), October 2017.

[I-D.pfister-6man-sadr-ra]
Pfister, P., "Source Address Dependent Route Information Option for Router Advertisements", draft-pfister-6man-sadr-ra-01 (work in progress), June 2015.

[I-D.xu-src-dst-bgp]
Xu, M., Yang, S., and J. Wu, "Source/Destination Routing Using BGP-4", draft-xu-src-dst-bgp-00 (work in progress), March 2016.

[PATRICIA]
Morrison, D., "Practical Algorithm to Retrieve Information Coded in Alphanumeric", Journal of the ACM 15(4) pp514-534, October 1968.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <https://www.rfc-editor.org/info/rfc3704>.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, DOI 10.17487/RFC3736, April 2004, <https://www.rfc-editor.org/info/rfc3736>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/info/rfc4862>.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <https://www.rfc-editor.org/info/rfc4941>.

[RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, DOI 10.17487/RFC5533, June 2009, <https://www.rfc-editor.org/info/rfc5533>.

[RFC5534] Arkko, J. and I. van Beijnum, "Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming", RFC 5534, DOI 10.17487/RFC5534, June 2009, <https://www.rfc-editor.org/info/rfc5534>.

[RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <https://www.rfc-editor.org/info/rfc6724>.

[RFC7078] Matsumoto, A., Fujisaki, T., and T. Chown, "Distributing Address Selection Policy Using DHCPv6", RFC 7078, DOI 10.17487/RFC7078, January 2014, <https://www.rfc-editor.org/info/rfc7078>.

[RFC7788] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", RFC 7788, DOI 10.17487/RFC7788, April 2016, <https://www.rfc-editor.org/info/rfc7788>.

[RFC8041] Bonaventure, O., Paasch, C., and G. Detal, "Use Cases and Operational Experience with Multipath TCP", RFC 8041, DOI 10.17487/RFC8041, January 2017, <https://www.rfc-editor.org/info/rfc8041>.

[RFC8305] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <https://www.rfc-editor.org/info/rfc8305>.

Appendix A. Change Log

Initial Version: July 2016

Authors' Addresses

Fred Baker
Santa Barbara, California 93117
USA

Email: FredBaker.IETF@gmail.com

Chris Bowers
Juniper Networks
Sunnyvale, California 94089
USA

Email: cbowers@juniper.net

Jen Linkova
Google
Mountain View, California 94043
USA

Email: furry@google.com

BFD Working Group
Internet-Draft
Updates: 5798 (if approved)
Intended status: Standards Track
Expires: November 25, 2018

G. Mirsky
ZTE Corp.
J. Tantsura

May 24, 2018

Bidirectional Forwarding Detection (BFD) for Multi-point Networks and Virtual Router Redundancy Protocol (VRRP) Use Case draft-mirsky-bfd-p2mp-vrrp-use-case-02

Abstract

This document discusses use of Bidirectional Forwarding Detection (BFD) for multi-point networks to provide Virtual Router Redundancy Protocol (VRRP) with sub-second Master convergence and defines the extension to bootstrap point-to-multipoint BFD session.

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 November 25, 2018.

Copyright Notice

Table of Contents

1. Introduction

The [RFC5798] is the current specification of the Virtual Router Redundancy Protocol (VRRP) for IPv4 and IPv6 networks. VRRPv3 allows for faster switchover to a Backup router. Using such capability with software-based implementation of VRRP is may prove challenging. But it still may be possible to deploy VRRP and provide sub-second detection of Master router failure by Backup routers.

Bidirectional Forwarding Detection (BFD) [RFC5880] had been originally defined detect failure of point-to-point (p2p) paths: single-hop [RFC5881], multihop [RFC5883]. Single-hop BFD may be used to enable Backup routers to detect failure of the Master router within 100 msec or faster. [I-D.nitish-vrrp-bfd] demonstrates how, with some extensions to [RFC5798], that can be achieved.

[I-D.ietf-bfd-multipoint] extends [RFC5880] for multipoint and multicast networks, which is precisely characterizes deployment scenarios for VRRP over LAN segment. This document demonstrates how point-to-multipoint (p2mp) BFD can enable faster detection of Master failure and thus minimize service disruption in a VRRP domain. The document also defines the extension to VRRP [RFC5798] to bootstrap a VRRP Backup router to join in p2mp BFD session.

1.1. Conventions used in this document

1.1.1. Terminology

BFD: Bidirectional Forwarding Detection

p2mp: Pont-to-Multipoint

VRRP: Virtual Router Redundancy Protocol

1.1.2. 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. Problem Statement

A router may be part of several Virtual Router Redundancy groups, as Master in some and as Backup in others. Supporting sub-second mode for VRRPv3 [RFC5798] for all these roles without specialized support in data plane may prove to be very challenging. BFD already has many implementations based on HW that are capable to support multiple sub-second session concurrently.

3. Applicability of p2mp BFD

[I-D.ietf-bfd-multipoint] may provide the efficient and scaleable solution for fast-converging environment that uses default route rather than dynamic routing. Each redundancy group presents itself as p2mp BFD session with its Master being the root and Backup routers being tails of the p2mp BFD session. Figure 1 displays the extension of VRRP [RFC5798] to bootstrap tail of the p2mp BFD session. Master

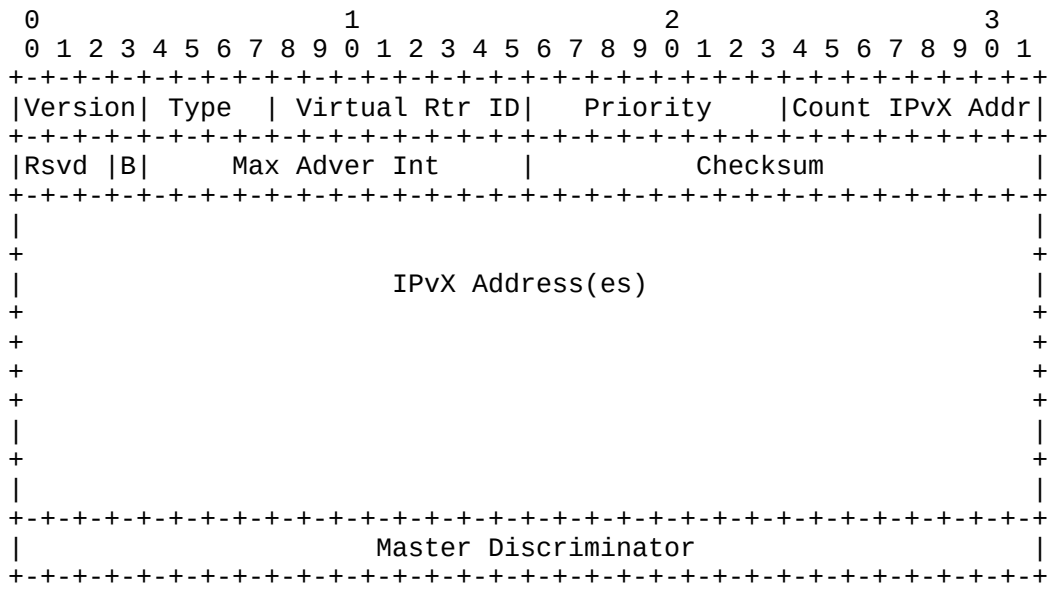


Figure 1: VRRP Extension to Bootstrap P2MP BFD session

where new fields are interpreted as:

B(FD) - one bit flag that indicates that the Master Discriminator field is appended to VRRP packet defined in [RFC5798];

Master Discriminator - My Discriminator value allocated by the root of the p2mp BFD session.

The Master router that is configured to use p2mp BFD to support faster convergence of VRRP starts transmitting BFD control packets with VRID as source IP address and My Discriminator. The same value of My Discriminator MUST be set as value of Master Discriminator field and BFD flag MUST be set in the VRRP packet. Backup router demultiplexes p2mp BFD test sessions based on VRID that it been configured with and the My Discriminator value it learns from the received VRRP packet. When a Backup router detects failure of the Master router it re-evaluates its role in the VRID. As result, the Backup router may become the Master router of the given VRID or continue as a Backup router. If the former is the case, then the new Master router MUST select My Discriminator and start transmitting p2mp BFD control packets using Master IP address as source IP address for p2mp BFD control packets. If the latter is the case, then the Backup router MUST wait for VRRP packet from the new VRRP Master router that will bootstrap new p2mp BFD session.

3.1. Multipoint BFD Encapsulation

The MultipointHead of p2mp BFD session when transmitting BFD control packet:

MUST set TTL value to 1 (though note that VRRP packets have TTL set to 255);

SHOULD use group address VRRP (’224.0.0.18’ for IPv4 and ’FF02:0:0:0:0:0:0:12’ for IPv6) as destination IP address

MAY use network broadcast address for IPv4 or link-local all nodes multicast group for IPv6 as destination IP address;

MUST set destination UDP port value to 3784 when transmitting BFD control packets, as defined in [I-D.ietf-bfd-multipoint];

MUST use Master IP address as source IP address.

4. IANA Considerations

This document makes no requests for IANA allocations. This section may be deleted by RFC Editor.

5. Security Considerations

Security considerations discussed in [RFC5798], [RFC5880], and [I-D.ietf-bfd-multipoint], apply to this document.

6. Acknowledgements

7. Normative References

[I-D.ietf-bfd-multipoint]
Katz, D., Ward, D., Networks, J., and G. Mirsky, "BFD for Multipoint Networks", draft-ietf-bfd-multipoint-16 (work in progress), April 2018.

[I-D.nitish-vrrp-bfd]
Gupta, N., Dogra, A., Docherty, C., Mirsky, G., and J. Tantsura, "Fast failure detection in VRRP with BFD", draft-nitish-vrrp-bfd-04 (work in progress), August 2016.

[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>.

[RFC5798] Nadas, S., Ed., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, DOI 10.17487/RFC5798, March 2010, <https://www.rfc-editor.org/info/rfc5798>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <https://www.rfc-editor.org/info/rfc5881>.

[RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, DOI 10.17487/RFC5883, June 2010, <https://www.rfc-editor.org/info/rfc5883>.

[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

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Jeff Tantsura

Email: jefftant.ietf@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 28, 2018

N. Gupta
A. Dogra
Cisco Systems, Inc.
C. Docherty
AT&T
G. Mirsky
J. Tantsura
Individual

January 24, 2018

Fast failure detection in VRRP with Point to Point BFD
draft-nitish-vrrp-bfd-p2p-02

Abstract

This document describes how Point to Point Bidirectional Forwarding Detection (BFD) can be used to support sub-second detection of a Master Router failure in the Virtual Router Redundancy Protocol (VRRP).

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 http://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 July 28, 2018.

Copyright Notice

Table of Contents

1. Introduction

The Virtual Router Redundancy Protocol (VRRP) provides redundant Virtual gateways in the Local Area Network (LAN), which is typically the first point of failure for end-hosts sending traffic out of the LAN. Fast failure detection of VRRP Master is critical in supporting high availability of services and improved Quality of Experience to users. In VRRP [RFC5798] specification, Backup routers depend on VRRP packets generated at a regular interval by the Master router, to detect the health of the VRRP Master. Faster failure detection can be achieved within VRRP protocol by reducing the Advertisement and Master Down Interval. However, sub second Advert timers, can put extra load on CPU and the network bandwidth which may not be desirable.

Since the VRRP protocol depends on the availability of Layer 3 IPv4 or IPv6 connectivity between redundant peers, the VRRP protocol can interact with the Layer 3 variant of BFD as described in [RFC5881] to achieve a much faster failure detection of the VRRP Master on the LAN. BFD, as specified by the [RFC5880] can provide a much faster failure detection in the range of 150ms, if implemented in the part of a Network device which scales better than VRRP when sub second Advert timers are used.

2. Requirements Language

In this document, several words are used to signify the requirements of the specification. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. [RFC2119]

3. Applicability of Point to Point BFD

BFD for IPv4 or IPv6 (Single Hop) [RFC5881] requires that in order for a BFD session to be formed both peers participating in a BFD session need to know its peer IPv4 or IPV6 address. This poses a unique problem with the definition of the VRRP protocol, that makes the use of BFD for IPv4 or IPv6 [RFC5881] more challenging. In VRRP it is only the Master router that sends Advert packets. This means that a Master router is not aware of any Backup routers, and Backup routers are only aware of the Master router. This also means that a Backup router is not aware of any other Backup routers in the Network.

Since BFD for IPv4 or IPv6 [RFC5881] requires that a session be formed by both peers using a full destination and source address, there needs to be some external means to provide this information to BFD on behalf of VRRP. Once the peer information is made available, VRRP can form BFD sessions with its peer Virtual Router. The BFD session for a given Virtual Router is identified as the Critical Path BFD Session, which is the session that forms between the current VRRP Master router, and the highest priority Backup router. When the Critical Path BFD Session identified by VRRP as having changed state from Up to Down, then this will be interpreted by the VRRP state machine on the highest priority Backup router as a Master Down event. A Master Down event means that the highest priority Backup peer will immediately become the new Master for the Virtual Router.

NOTE: At all times, the normal fail-over mechanism defined in the VRRP [RFC5798] will be unaffected, and the BFD fail-over mechanism will always resort to normal VRRP fail-over.

This draft defines the mechanism used by the VRRP protocol to build a peer table that will help in forming of BFD session and the detection of Critical Path BFD session. If the Critical Path BFD session were to go down, it will signal a Master Down event and make the most preferred Backup router as the VRRP Master router. This requires an extension to the VRRP protocol.

This can be achieved by defining a new type in the VRRP Advert packet, and allowing VRRP peers to build a peer table in any of the operational state, Master or Backup.

3.1. Extension to VRRP protocol

In this mode of operation VRRP peers learn the adjacent routers, and form BFD session between the learnt routers. In order to build the peer table, all routers send VRRP Advert packets whilst in any of the operational states (Master or Backup). Normally VRRP peers only send

Advert packets whilst in the Master state, however in this mode VRRP Backup peers will also send Advert packets with the type field set to BACKUP ADVERTISEMENT type defined in Section 3.3 of this document. The VRRP Master router will still continue to send packets with the Advert type as ADVERTISEMENT as defined in the VRRP protocol. This is to maintain inter-operability with peers complying to VRRP protocol.

Additionally, Advert packets sent from Backup Peers must not use the Virtual router MAC address as the source address. Instead it must use the Interface MAC address as the source address from which the packet is sent from. This is because the source MAC override feature is used by the Master to send Advert packets from the Virtual Router MAC address, which is used to keep the bridging cache on LAN switches and bridging devices refreshed with the destination port for the Virtual Router MAC.

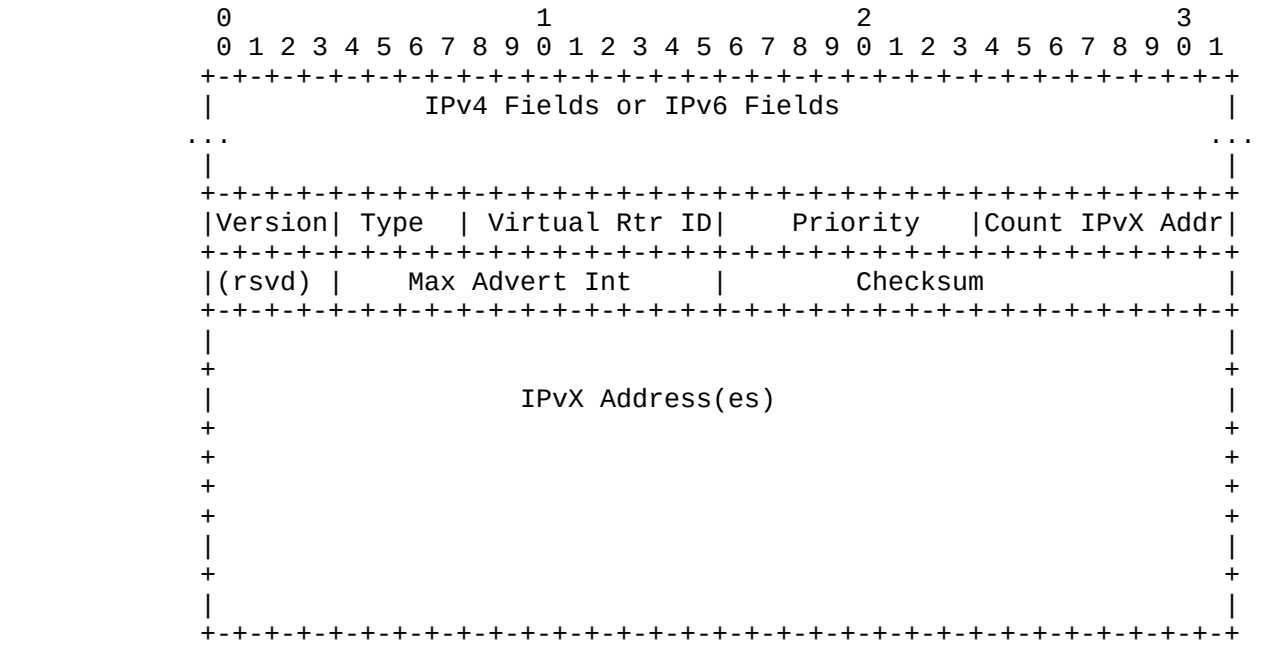
3.2. VRRP Peer Table

VRRP peers can now form the peer table by learning the source address in the ADVERTISEMENT or BACKUP ADVERTISEMENT packet sent by VRRP Master or Backup peers. This allows peers to create BFD sessions with other operational peers.

A peer entry should be removed from the peer table if Advert is not received from a peer for a period of (3 * the Advert interval).

3.3. VRRP BACKUP ADVERTISEMENT Packet Type

The following figure shows the VRRP packet as defined in VRRP [RFC5798] RFC.



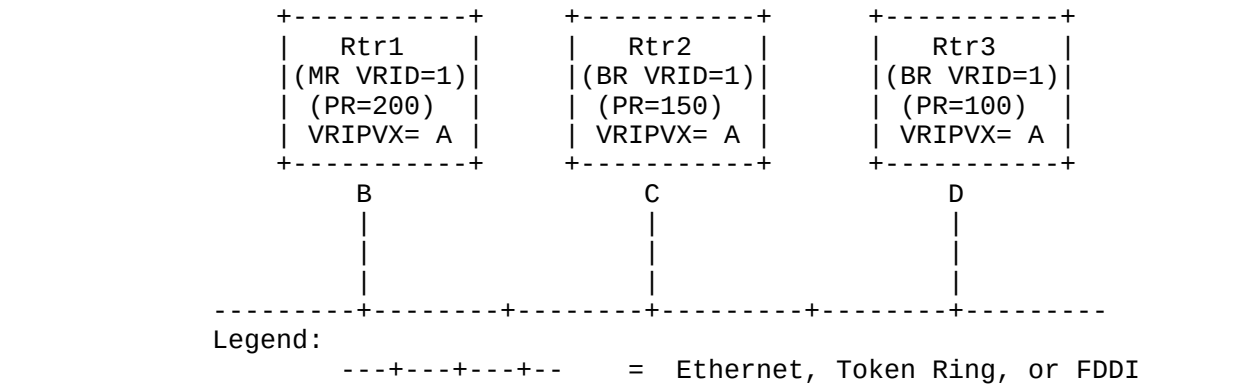
The type field specifies the type of this VRRP packet. The type field can have two values. Type 1 (ADVERTISEMENT) is used by the VRRP Master Router. Type 2 (BACKUP ADVERTISEMENT) is used by the VRRP Backup router. This is to distinguish the packets sent by the VRRP backup Router. VRRP Backup fills Backup_Advertisement_Interval in the Max Advert Int of BACKUP ADVERTISEMENT packet. Rest of the fields in Advert packet remain the same.

1 ADVERTISEMENT
2 BACKUP ADVERTISEMENT

A packet with unknown type MUST be discarded.

3.4. Sample configuration

The following figure shows a simple network with three VRRP routers implementing one virtual router.



In the above configuration there are three routers on the LAN protecting an IPv4 or IPv6 address associated to a Virtual Router ID 1. Rtr1 is the Master router since it has the highest priority compared to Rtr2 and Rtr3. Now if peer learning extension is enabled on all the peers. Rtr1 will send the Advert packet with type field set to 1. While Rtr2 and Rtr3 will send the Advert packet with type field set to 2. In the above configuration the peer table built at each router is shown below:

Rtr1 Peer table

Peer Address	Priority
C	150
D	100

Rtr2 Peer table

Peer Address	Priority
B	200
D	100

Rtr3 Peer table

Peer Address	Priority
B	200
C	150

Once the peer tables are formed, VRRP on each router can form a BFD sessions with the learnt peers.

3.5. Critical BFD session

The Critical BFD Session is determined to be the session between the VRRP Master and the next best VRRP Backup. Failure of the Critical BFD session indicates that the Master is no longer available and the most preferred Backup will now become Master.

In the above example the Critical BFD session is shared between Rtr1 and Rtr2. If the BFD Session goes from Up to Down state, Rtr2 can treat it as a Master down event and immediately assume the role of VRRP Master router for VRID 1 and Rtr3 will become the critical Backup. If the priorities of two Backup routers are same then the primary IPvX Address of the sender is used to determine the highest priority Backup. Where higher IPvX address has higher priority.

3.6. Protocol State Machine

3.6.1. Parameters Per Virtual Router

Following parameters are added to the VRRP protocol to support this mode of operation.

Backup_Advertisement_Interval	Time interval between BACKUP ADVERTISEMENTS (centiseconds). Default is 100 centiseconds (1 second).
Backup_Adver_Interval	Advertisement interval contained in BACKUP ADVERTISEMENTS received from the Backup (centiseconds). This value is saved by virtual routers used, to compute Backup_Down_Interval.
Backup_Down_Interval	Time interval for VRRP instance to declare Backup down (centiseconds). Calculated as (3 * Backup_Adver_Interval) for each VRRP Backup.
Critical_Backup	Procedure outlined in section 3.4 of this document is used to determine the Critical_Backup at each VRRP Instance.
Critical_BFD_Session	The Critical BFD Session is the session between the VRRP Master and Critical_Backup.

3.6.2. Timers

Following timers are added to the VRRP protocol to support this mode of operation.

Backup_Down_Timer	Timer that fires when BACKUP ADVERTISEMENT has not been heard from a backup peer for Backup_Down_Interval.
Backup_Adver_Timer	Timer that fires to trigger sending of BACKUP ADVERTISEMENT based on Backup_Advertisement_Interval.

3.6.3. VRRP State Machine with Point to Point BFD

Following State Machine replaces the state Machine outlined in section 6.4 of the VRRP protocol [RFC5798] to support this mode of operation. Please refer to the section 6.4 of [RFC5798] for State description.

3.6.3.1. Initialize

Following state machine replaces the state machine outlined in section 6.4.1 of [RFC5798]

(100) If a Startup event is received, then:

(105) - If the Priority = 255 (i.e., the router owns the IPvX address associated with the virtual router), then:

(110) + Send an ADVERTISEMENT

(115) + If the protected IPvX address is an IPv4 address, then:

(120) * Broadcast a gratuitous ARP request containing the virtual router MAC address for each IP address associated with the virtual router.

(125) + else // IPv6

(130) * For each IPv6 address associated with the virtual router, send an unsolicited ND Neighbor Advertisement with the Router Flag (R) set, the Solicited Flag (S) unset, the Override flag (O) set, the target address set to the IPv6 address of the virtual router, and the target link-layer address set to the virtual router MAC address.

(135) +endif // was protected addr IPv4?

(140) + Set the Adver_Timer to Advertisement_Interval

(145) + Transition to the {Master} state

(150) - else // rtr does not own virt addr

(155) + Set Master_Adver_Interval to Advertisement_Interval

(160) + Set the Master_Down_Timer to Master_Down_Interval

(165) + Set Backup_Adver_Timer to Backup_Advertisement_Interval

(170) + Transition to the {Backup} state

(175) -endif // priority was not 255

(180) endif // startup event was recv

3.6.3.2. Backup

Following state machine replaces the state machine outlined in section 6.4.2 of [RFC5798]

(300) While in this state, a VRRP router MUST do the following:

(305) - If the protected IPvX address is an IPv4 address, then:

(310) + MUST NOT respond to ARP requests for the IPv4 address(es) associated with the virtual router.

(315) - else // protected addr is IPv6

(320) + MUST NOT respond to ND Neighbor Solicitation messages for the IPv6 address(es) associated with the virtual router.

(325) + MUST NOT send ND Router Advertisement messages for the virtual router.

(330) -endif // was protected addr IPv4?

(335) - MUST discard packets with a destination link-layer MAC address equal to the virtual router MAC address.

(340) - MUST NOT accept packets addressed to the IPvX address(es) associated with the virtual router.

(345) - If a Shutdown event is received, then:

(350) + Cancel the Master_Down_Timer.

(355) + Cancel the Backup_Adver_Timer.

(360) + Cancel Backup_Down_Timers.

(365) + Remove Peer table.

(370) + If Critical_BFD_Session Exists:

(375) * Tear down the Critical_BFD_Session.

(380) + endif // Critical_BFD_Session Exists?

(385) + Send a BACKUP ADVERTISEMENT with Priority = 0.

(390) + Transition to the {Initialize} state.

```
(395) -endif // shutdown recv

(400) - If the Master_Down_Timer fires or
        If Critical_BFD_Session transitions from UP to DOWN, then:

   (405) + Send an ADVERTISEMENT

   (415) + If the protected IPvX address is an IPv4 address, then:

      (420) * Broadcast a gratuitous ARP request on that interface
      containing the virtual router MAC address for each IPv4
      address associated with the virtual router.

   (425) + else // ipv6

      (430) * Compute and join the Solicited-Node multicast
      address [RFC4291] for the IPv6 address(es) associated with
      the virtual router.

      (435) * For each IPv6 address associated with the virtual
      router, send an unsolicited ND Neighbor Advertisement with
      the Router Flag (R) set, the Solicited Flag (S) unset, the
      Override flag (O) set, the target address set to the IPv6
      address of the virtual router, and the target link-layer
      address set to the virtual router MAC address.

   (440) +endif // was protected addr ipv4?

   (445) + Set the Adver_Timer to Advertisement_Interval.

   (450) + If the Critical_BFD_Session exists:

      (455) @ Tear Critical_BFD_Session.

    (460) + endif // Critical_BFD_Session exists

   (465) + Calculate the Critical_Backup.

   (470) + If the Critical_Backup exists:

      (475) * BootStrap Critical_BFD_Session with the
              Critical_Backup.

   (480) + endif //Critical_Backup exists?

   (485) + Transition to the {Master} state.

(490) -endif // Master_Down_Timer fired
```

```
(485) - If an ADVERTISEMENT is received, then:

   (490) + If the Priority in the ADVERTISEMENT is zero, then:

      (495) * Set the Master_Down_Timer to Skew_Time.

      (500) * If the Critical_BFD_Session exists:

         (505) * Tear Critical_BFD_Session with the Master.

      (510) * endIf // Critical_BFD_Session exists

   (515) + else // priority non-zero

      (520) * If Preempt_Mode is False, or if the Priority in the
      ADVERTISEMENT is greater than or equal to the local
      Priority, then:

         (525) @ Set Master_Adver_Interval to Adver Interval
         contained in the ADVERTISEMENT.

         (530) @ Recompute the Master_Down_Interval.

         (535) @ Reset the Master_Down_Timer to
         Master_Down_Interval.

         (540) @ Determine Critical_Backup.

         (545) @ If Critical_BFD_Session does not exists and this
         instance is the Critical_Backup:

            (550) @+ BootStrap Critical_BFD_Session with Master.

         (555) @ endif //Critical_BFD_Session exists check

      (560) * else // preempt was true or priority was less

         (565) @ Discard the ADVERTISEMENT.

      (570) *endif // preempt test

   (575) +endif // was priority zero?

(580) -endif // was advertisement recv?

(585) - If a BACKUP ADVERTISEMENT is received, then:

   (590) + If the Priority in the BACKUP ADVERTISEMENT is zero,
```

```
                then:

            (595) * Cancel Backup_Down_Timer.

            (600) * Remove the Peer from Peer table.

         (605) + else // priority non-zero

            (610) * Update the peer table with peer information.

            (615) * Set Backup_Adver_Interval to Adver Interval
            contained in the BACKUP ADVERTISEMENT.

            (620) * Recompute the Backup_Down_Interval.

            (625) * Reset the Backup_Down_Timer to Backup_Down_Interval.

         (630) +endif // was priority zero?

         (635) + Recalculate Critical_Backup.

         (640) + If Critical_BFD_Session exists and this
         instance is not the Critical_Backup:

            (645) * Tear Down the Critical_BFD_Session.

         (650) + else If Critical_BFD_Session does not exists and this
         instance is the Critical_Backup:

            (655) * BootStrap Critical_BFD_Session with Master.

         (660) + endif // Critical_Backup change

      (665) -endif // was backup advertisement recv?

      (670) - If Backup_Down_Timer fires, then:

         (675) + Remove the Peer from Peer table.

         (680) + If Critical_BFD_Session does not exist:

            (685) @ Recalculate Critical_Backup.

            (690) @ If This instance is the Critical_Backup:

               (695) +@ BootStrap Critical_BFD_Session with Master.

            (700) @ endif // Critical_Backup change
```

(705) + endif // Critical_BFD_Session does not exist?

(710) -endif // Backup_Down_Timer fires?

(715) - If Backup_Adver_Timer fires, then:

(720) + Send a BACKUP ADVERTISEMENT.

(725) + Reset the Backup_Adver_Timer to Backup_Advertisement_Interval.

(730) -endif // Backup_Down_Timer fires?

(735) endwhile // Backup state

3.6.3.3. Master

Following state machine replaces the state machine outlined in section 6.4.3 of [RFC5798]

(800) While in this state, a VRRP router MUST do the following:

(805) - If the protected IPvX address is an IPv4 address, then:

(810) + MUST respond to ARP requests for the IPv4 address(es) associated with the virtual router.

(815) - else // ipv6

(820) + MUST be a member of the Solicited-Node multicast address for the IPv6 address(es) associated with the virtual router.

(825) + MUST respond to ND Neighbor Solicitation message for the IPv6 address(es) associated with the virtual router.

(830) + MUST send ND Router Advertisements for the virtual router.

(835) + If Accept_Mode is False: MUST NOT drop IPv6 Neighbor Solicitations and Neighbor Advertisements.

(840) -endif // ipv4?

(845) - MUST forward packets with a destination link-layer MAC address equal to the virtual router MAC address.

(850) - MUST accept packets addressed to the IPvX address(es) associated with the virtual router if it is the IPvX address owner or if Accept_Mode is True. Otherwise, MUST NOT accept these packets.

(855) - If a Shutdown event is received, then:

(860) + Cancel the Adver_Timer.

(865) + Send an ADVERTISEMENT with Priority = 0,

(870) + Cancel Backup_Down_Timers.

(875) + Remove Peer table.

(880) + If Critical_BFD_Session Exists:

(885) * Tear down Critical_BFD_Session

(890) + endif // If Critical_BFD_Session Exists

(895) + Transition to the {Initialize} state.

(900) -endif // shutdown recv

(905) - If the Adver_Timer fires, then:

(910) + Send an ADVERTISEMENT.

(915) + Reset the Adver_Timer to Advertisement_Interval.

(920) -endif // advertisement timer fired

(925) - If an ADVERTISEMENT is received, then:

(930) -+ If the Priority in the ADVERTISEMENT is zero, then:

(935) -* Send an ADVERTISEMENT.

(940) -* Reset the Adver_Timer to Advertisement_Interval.

(945) -+ else // priority was non-zero

(950) -* If the Priority in the ADVERTISEMENT is greater than the local Priority,

(955) -* or

```
            (960) -* If the Priority in the ADVERTISEMENT is equal to
            the local Priority and the primary IPvX Address of the
            sender is greater than the local primary IPvX Address, then:

               (965) -@ Cancel Adver_Timer

               (970) -@ Set Master_Adver_Interval to Adver Interval
               contained in the ADVERTISEMENT

               (975) -@ Recompute the Skew_Time

               (980) @ Recompute the Master_Down_Interval

               (985) @ Set Master_Down_Timer to Master_Down_Interval

               (990) If Critical_BFD_Session Exists:

                  (995) @+ Tear Critical_BFD_Session

               (960) @ endif //Critical_BFD_Session Exists?

               (965) @ Calculate Critical_Backup.

               (970) @ If this instance is Critical_Backup:

                  (975) @+ BootStrap Critical_BFD_Session with new
                           Master.

               (980) @ endif // am i Critical_Backup?

               (985) @ Transition to the {Backup} state

            (990) * else // new Master logic

               (995) @ Discard ADVERTISEMENT

            (1000) *endif // new Master detected

         (1005) +endif // was priority zero?

      (1010) -endif // advert recv

      (1015) - If a BACKUP ADVERTISEMENT is received, then:

         (1020) + If the Priority in the BACKUP ADVERTISEMENT is
                  zero, then:

            (1025) * Remove the Peer from peer table.
```

```
         (1030) + else: // priority non-zero

            (1035) * Update the Peer info in peer table.

            (1040) * Recompute the Backup_Down_Interval

            (1045) * Reset the Backup_Down_Timer to
                     Backup_Down_Interval

         (1050) + endif // priority in backup advert zero

         (1055) + Calculate the Critical_Backup

         (1060) + If Critical_BFD_Session doesnot exist:

            (1065) * BootStrap Critical_BFD_Session

         (1070) + else if Critical_BFD_Session exist and
                  Critical_Backup changes:

            (1075) + Tear Critical_BFD_Session with old Backup

            (1080) + BootStrap Critical_BFD_Session with Critical_Backup

         (1085) + endif //  Critical_BFD_Session check?

      (1090) - endif // backup advert recv

      (1095) - If Critical_BFD_Session transitions from UP to DOWN,
      then:
         (1100) + Cancel Backup_Down_Timer

         (1105) + Delete the Peer info from peer table

         (1200) + Calculate the Critical_Backup

         (1205) + BootStrap Critical_BFD_Session with Critical_Backup

      (1210) - endif // BFD session transition

   (1215) endwhile // in Master
```

4. Scalability Considerations

To reduce the number of packets generated at a regular interval, Backup Advert packets may be sent at a reduced rate as compared to Advert packets sent by the VRRP Master.

5. Operational Considerations

A VRRP peer that forms a member of this Virtual Router, but does not support this feature or extension must be configured with the lowest priority, and will only operate as the Router of last resort on failure of all other VRRP routers supporting this functionality.

It is recommended that mechanism defined by this draft, to interface VRRP with BFD should be used when BFD can support more aggressive monitoring timers than VRRP. Otherwise it is desirable not to interface VRRP with BFD for determining the health of VRRP Master.

This Draft does not preclude the possibility of the peer table being populated by means of manual configuration, instead of using the BACKUP ADVERTISEMENT as defined by the Draft.

6. Applicability to VRRPv2

The workings of this Draft can be extended to VRRPv2 [RFC3768], with the introduction of BACKUP ADVERTISEMENT and Peer Table as outlined in the Draft.

7. IANA Considerations

This document requests IANA to create a new name space that is to be managed by IANA. The document defines a new VRRP Packet Type. The VRRP Packet Types are discussed below.

a) Type 1 (ADVERTISEMENT) defined in section 5.2.2 of [RFC5798]
b) Type 2 (BACKUP ADVERTISEMENT) defined in section 3.3 of this document

7.1. A New Name Space for VRRP Packet Types

This document defines in Section 3.3 a "BACKUP ADVERTISEMENT" VRRP Packet Type. The new name space has to be created by the IANA and they will maintain this new name space. The field for this namespace is 4-Bits, and IANA guidelines for assignments for this field are as follows:

ADVERTISEMENT	1
BACKUP ADVERTISEMENT	2

Future allocations of values in this name space are to be assigned by IANA using the "Specification Required" policy defined in [IANA-CONS]

8. Security Considerations

Security considerations discussed in [RFC5798], [RFC5880], apply to this document. There are no additional security considerations identified by this draft.

9. Acknowledgements

The authors gratefully acknowledge the contributions of Gerry Meyer, and Mouli Chandramouli, for their contributions to the draft. The authors will also like to thank Jeffrey Haas, Maik Pfeil, Chris Bowers, Vengada Prasad Govindan and Alexander Vainshtein for their comments and suggestions.

10. Normative References

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, 2010.

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

[RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, 2010.

[RFC5798] Nadas, S., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, 2010.

[RFC3768] Hinden, R., "Virtual Router Redundancy Protocol (VRRP)", RFC 3768, 2004.

[IANA-CONS] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 2434, 1998.

Authors' Addresses

Nitish Gupta
Cisco Systems, Inc.
3265 CISCO Way
San Jose 95134
United States

Phone: +91 80 4429 2530
Email: nitisgup@cisco.com
URI: http://www.cisco.com/

Aditya Dogra
Cisco Systems, Inc.
Sarjapur Outer Ring Road
Bangalore 560103
India

Phone: +91 80 4429 2166
Email: addogra@cisco.com
URI: http://www.cisco.com/

Colin Docherty
AT&T
23 The Maltings
Haddington, Scotland EH414EF
United Kingdom

Email: colin.docherty@att.com

Greg Mirsky
Individual

Email: gregimirsky@gmail.com

Jeff Tantsura
Individual

Email: jefftant.ietf@gmail.com

i2rs
Internet-Draft
Intended status: Informational
Expires: May 2, 2018

M. Wang, Ed.
Huawei
R. Gu
China Mobile
Victor. Lopez
Telefonica
S. Hu
China Mobile

October 29, 2017

Information Model of Control-Plane and User-Plane separation BNG
draft-wcg-i2rs-cu-separation-infor-model-02

Abstract

To improve network resource utilization and reduce the operation expense, the Control-Plane and User-Plane separation conception is raised [I-D.gu-nfvrg-cloud-bng-architecture]. This document describes the information model for the interface between Control-Plane and User-Plane separation BNG. This information model may involve both control channel interface and configuration channel interface. The interface for control channel allows the Control-Plane to send several flow tables to the User-Plane, such as user's information table, user's interface table, and user's QoS table, etc. And it also allows the User-Plane to report the resources and statistics information to the Control-Plane. The interface for configuration channel is in charge of the version negotiation of protocols between the Control-Plane and User-Plane, the configuration for devices of Control-Plane and User-Plane, and the reports of User-Plane's capabilities, etc. The information model defined in this document enables defining a standardized data model. Such a data model can be used to define an interface to the CU separation BNG.

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 May 2, 2018.

Table of Contents

1. Introduction

The rapid development of new services, such as 4K, IoT, etc, and increasing numbers of home broadband service users present some new challenges for BNGs such as:

Low resource utilization: The traditional BNG acts as both a gateway for user access authentication and accounting and an IP network's Layer 3 edge. The mutually affecting nature of the tightly coupled control plane and forwarding plane makes it difficult to achieve the maximum performance of either plane.

Complex management and maintenance: Due to the large numbers of traditional BNGs, a network must have each device configured one at a time when deploying global service policies. As the network expands and new services are introduced, this deployment mode will cease to be feasible as it is unable to manage services effectively and rectify faults rapidly.

Slow service provisioning: The coupling of control plane and forwarding plane, in addition to a distributed network control mechanism, means that any new technology has to rely heavily on the existing network devices.

To address these challenges, cloud-based BNG with CU separation conception is raised [I-D.gu-nfvrg-cloud-bng-architecture]. The main idea of Control-Plane and User-Plane separation method is to extract and centralize the user management functions of multiple BNG devices, forming an unified and centralized control plane (CP). And the traditional router's Control Plane and Forwarding Plane are both preserved on BNG devices in the form of a user plane (UP).

This document describes an information model for the interface between Control-Plane and User-Plane separation BNG. This information model may involve both control channel interface and configuration channel interface. The interface for control channel allows the Control-Plane to send several flow tables to the User-Plane, such as user's information table, user's interface table, and user's QoS table, etc. And it also allows User-Plane to report the resources and statistics information to the Control-Plane. The interface for configuration channel is in charge of the version negotiation of protocols between the Control-Plane and User-Plane, the configuration for the devices of Control-Plane and User-Plane, and the report of User-Plane's capabilities, etc. The information model defined in this document enables defining a standardized data model. Such a data model can be used to define an interface to the CU separation BNG.

2. Concept and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2.1. Terminology

BNG: Broadband Network Gateway. A broadband remote access server (BRAS, B-RAS or BBRAS) routes traffic to and from broadband remote access devices such as digital subscriber line access multiplexers (DSLAM) on an Internet service provider's (ISP) network. BRAS can also be referred to as a Broadband Network Gateway (BNG).

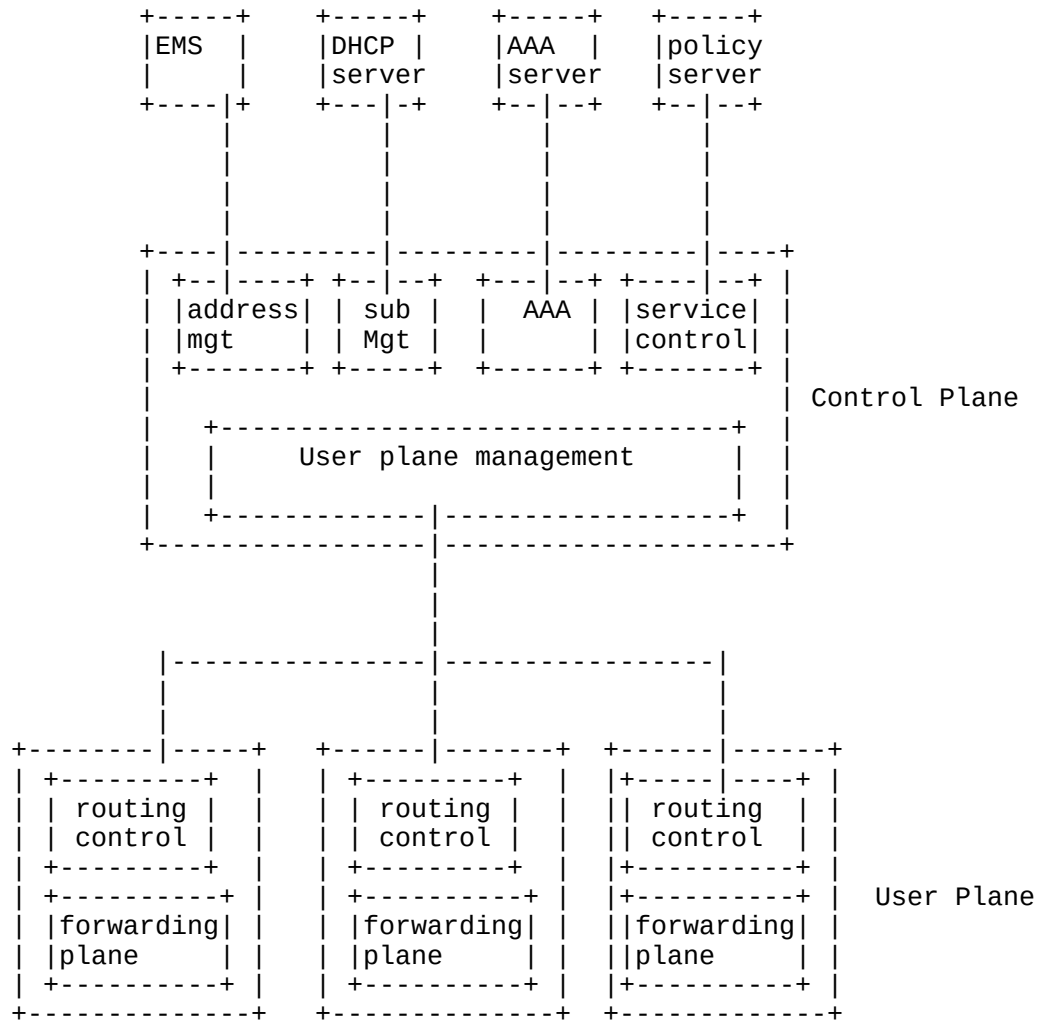
CP: Control Plane. CP is a user control management component which supports the management of UP's resources such as the user entry and forwarding policy

UP: User Plane. UP is a network edge and user policy implementation component. The traditional router's Control Plane and Forwarding Plane are both preserved on BNG devices in the form of a user plane.

3. Control Plane and User Plane separation BNG Information Model Overview

Briefly, a CU separation BNG is made up of a centralized CP and a set of UPs. The CP is a user control management component which supports to manage UP's resources such as the user entry and forwarding policy, for example, the access bandwidth and priority management. And the UP is a network edge and user policy implementation component. It can support the forwarding plane functions on traditional BNG devices, such as traffic forwarding, QoS, and traffic statistics collection, and it can also support the control plane functions on traditional BNG devices, such as routing, multicast, etc.

The following figure describes the architecture of CU separation BNG



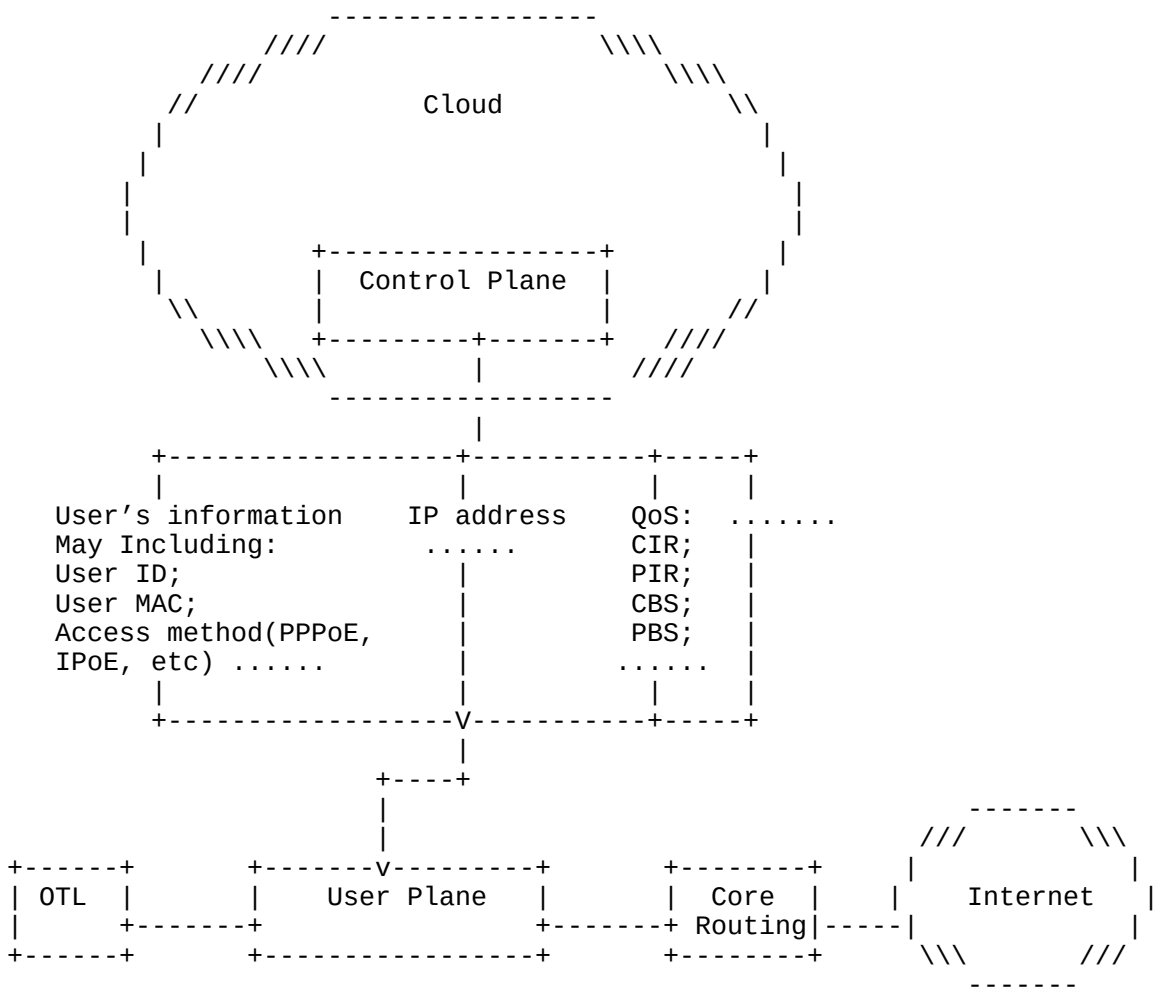
The CU separated BNG is shown in above figure. The BNG Control Plane could be virtualized and centralized, which provides significant benefits such as centralized session management, flexible address allocation, high scalability for subscriber management capacity, and cost-efficient redundancy, etc. The functional components inside the BNG Service Control Plane can be implemented as VNFs and hosted in a NFVI.

The User Plane Management module in the BNG control plane centrally manages the distributed BNG User Planes (e.g. load balancing), as well as the setup, deletion, maintenance of channels between Control

Planes and User Planes. Other modules in the BNG control plane, such as address management, AAA, and etc., are responsible for the connection with outside subsystems in order to fulfill the service. The routing control and forwarding Plane in the BNG User Plane (local) could be distributed across the infrastructure.

3.1. Service Data Model Usage

The idea of the information model is to propose a set of generic and abstract information models. The models are intended to be used in both Control Plane and User Planes. A typical scenario would be that this model can be used as a compendium for the interface between Control Plane and User Planes of CU separation BNG, that corresponding data model or TLVs can be defined to realize the communication between the Control Plane and User Planes.



CU Separation BNG

As shown in above figure, when users access to the BNG network, the control plane solicits these users' information (such as user's ID, user's MAC, user's access methods, for example via PPPoE/IPoE), associates them with available bandwidth which are reported by User planes, and based on the service's requirement to generate a set of tables, which may include user's information, user's IP address, and QoS, etc. Then the control plane can transmit these tables to the User planes. User planes receive these tables, parses it, matches these rules, and then performs corresponding actions.

4. Information Model

This section specifies the information model in Routing Backus-Naur Form [I-D.gu-nfvrg-cloud-bng-architecture]. This grammar intends to help readers better understand the English text description in order to derive a data model. However it may not provide all the details provided by the English text. When there is a lack of clarity in grammar the English text will take precedence.

This section describes information model that represents the concept of the interface of CU separation BNG which is languages and protocols neutral.

The following figure describes the Overview of Information Model for CU separation BNG.

```
<cu-separation-bng-infor-model>::=<control-plane-information-model>
                                  <user-plane-information-model>

<control-plane-information-model>::=<user-related-infor-model>
                                    <interface-related-infor-model>
                                    <device-related-infor-model>

<user-related-infor-model>::= <user-basic-information>
                              [<ipv4-informatiom>]|[<ipv6-information>]
                              [<qos-information>]

<user-basic-information> :: = <USER_ID> <MAC_ADDRESS>
                              [<ACCESS_TYPE>][<SESSION_ID>]
                              [<INNER_VLAN-ID>][<OUTER_VLAN_ID>]
                              <USER_INTERFACE>

<ipv4-informatiom>::=<USER_ID><USER_IPV4>
                      <MASK_LENGTH><GATEWAY>
                      <VRF>

<ipv6-information>::=<USER_ID>(<USER_IPV6>
                      <PREFIX_LEN>)|(<PD_ADDRESS><PD_PREFIX_LEN>)
                      <VRF>

<qos-information>::=<USER_ID>
                    (<CIR><PIR><CBS><PBS>)
                    [<QOS_PROFILE>]

<interface-related-infor-model>::=<interface-information>

<interface-information>::=<IFINDEX><BAS_ENABLE>
                           <service-type>
```

```
 <service-type>::=<PPP_Only><IPV4_TRIG>
                  <IPV6_TRIG><ND-TRIG>
                  <ARP_PROXY>

 <device-related-infor-model>::=<address-field-distribute>

 <address-field-distribute>::=<ADDRESS_SEGMENT><ADDRESS_SEGMENT_MASK>
                              <ADDRESS_SEGMENT_VRF><NEXT_HOP>
                              <IF_INDEX><MASK_LENGTH>

 <user-plane-information-model>::=<port-resources-infor-model>
                                  <traffic-statistics>

 <port-resource-information>::=<IF_INDEX><IF_NAME>
                               <IF_TYPE><LINK_TYPE>
                               <MAC_ADDRESS><IF_PHY_STATE>
                               <MTU>
 <traffic-statistics-information>::=<USER_ID><STATISTICS_TYPE>
                                    <INGRESS_STATIISTICS_PACKETS>
                                    <INGRESS STATISTICS_BYTES>
                                    <EGRESS_STATISTICS_PACKETS>
                                    <EGRESS_STATISTICS_BYTES>
```

4.1. Information Model for Control-Plane

This section describes information model for the Control-Plane (CP). As mentioned in section 3, the Control Plane is a user control management component which manages the user's information, User-Plane's resources and forwarding policy, etc. The control plane can generate several tables which contain a set of rules based on the resources and specific requirements of user's service. After that, the control plane sends the tables to User Planes, and User planes receive the tables, parse them, match the rules, and then perform corresponding actions.

The Routing Backus-Naur Form grammar below illustrates the Information model for Control-Plane:

```
<control-plane-information-model>::=<user-related-infor-model>
                                    <interface-related-infor-model>
                                    <device-related-infor-model>

<user-related-infor-model>::= <user-basic-information>
                              [<ipv4-informatiom>]|[<ipv6-information>]
                              [<qos-information>]

<user-basic-information> :: = <USER_ID> <MAC_ADDRESS>
                              [<ACCESS_TYPE>][<SESSION_ID>]
                              [<INNER_VLAN-ID>][<OUTER_VLAN_ID>]
                              <USER_INTERFACE>

<ipv4-informatiom>::=<USER_ID><USER_IPV4>
                      <MASK_LENGTH><GATEWAY>
                      <VRF>

<ipv6-information>::=<USER_ID>(<USER_IPV6>
                      <PREFIX_LEN>)|(<PD_ADDRESS><PD_PREFIX_LEN>)
                      <VRF>

<qos-information>::=<USER_ID>
                    (<CIR><PIR><CBS><PBS>)
                    [<QOS_PROFILE>]

<interface-related-infor-model>::=<interface-information>

<interface-information>::=<IFINDEX><BAS_ENABLE>
                           <service-type>

<service-type>::=<PPP_Only><IPV4_TRIG>
                 <IPV6_TRIG><ND-TRIG>
                 <ARP_PROXY>

<device-related-infor-model>::=<address-field-distribute>

<address-field-distribute>::=<ADDRESS_SEGMENT><ADDRESS_SEGMENT_MASK>
                             <ADDRESS_SEGMENT_VRF><NEXT_HOP>
                             <IF_INDEX><MASK_LENGTH>
```

user-related-infor-model: present the attributes which can describe the user’s profile, such as user’s basic information, qos, and IP address, etc.

interface-related-infor-model: present the attributes which relate to some physical/virtual interface. This model can be used to indicate which kinds of service can be supported by interfaces.

device-related-infor-model: present the attributes which relate to specific device. For example the control plane can manage and distribute the users, which belong to same subnet, to some specific devices. And the user plane's devices provide corresponding service for these users.

4.1.1. User-Related Information

The user related information are a bunch of attributes which may bind to specific users. For example, the control plane can use a unified ID to distinguish different users and distribute the IP address and QoS rules to a specific user. In this section, the user related information models are presented. The user related information models include the user information model, IPv4/IPv6 information model, QoS information model, etc.

The Routing Backus-Naur Form grammar below illustrates the user related information model:

```
<user-related-infor-model>::= <user-basic-information>
                              [<ipv4-informatiom>]|[<ipv6-information>]
                              [<qos-information>]

<user-basic-information> :: = <USER_ID> <MAC_ADDRESS>
                              [<ACCESS_TYPE>][<SESSION_ID>]
                              [<INNER_VLAN-ID>][<OUTER_VLAN_ID>]
                              <USER_INTERFACE>

<ipv4-informatiom>::=<USER_ID><USER_IPV4>
                      <MASK_LENGTH><GATEWAY>
                      <VRF>

<ipv6-information>::=<USER_ID>(<USER_IPV6>
                      <PREFIX_LEN>)|(<PD_ADDRESS><PD_PREFIX_LEN>)
                      <VRF>

<qos-information>::=<USER_ID>
                    (<CIR><PIR><CBS><PBS>)
                    [<QOS_PROFILE>]
```

4.1.1.1. User Basic Information Model

The User Basic Information model contains a set of attributes to describe the basic information of a specific user, such as user's mac address, access type (via PPPoE, IPoE, etc), inner vlan ID, outer vlan ID, etc.

The Routing Backus-Naur Form grammar below illustrates the user basic information model:

```
 <user-basic-information> :: = <USER_ID> <MAC_ADDRESS>
                               [<ACCESS_TYPE>][<SESSION_ID>]
                               [<INNER_VLAN-ID>][<OUTER_VLAN_ID>]
                               <USER_INTERFACE>
```

USER_ID: is the identifier of user. This parameter is a unique and mandatory, it can be used to distinguish different users.

MAC_ADDRESS: is the MAC address of the user.

ACCESS_TYPE: This attribute is an optional parameter. It can be used to indicate the protocol be used for user's accessing, such as PPPoE, IPoE, etc.

SESSION_ID: This attribute is an optional parameter. It can be used as the identifier of PPPoE session.

INNER_VLAN-ID: The identifier of user's inner VLAN.

OUTER_VLAN_ID: The identifier of user's outer VLAN.

USER_INTERFACE: This attribute specifies the binding interface of a specific user. The ifIndex of the interface MAY be included. This is the 32-bit ifIndex assigned to the interface by the device as specified by the Interfaces Group MIB [RFC2863]. The ifIndex can be utilized within a management domain to map to an actual interface, but it is also valuable in public applications [RFC5837]. The ifIndex can be used as an opaque token to discern which interface of User-Plane is providing corresponding service for specific user.

4.1.1.2. IPv4 Information Model

The IPv4 information model presents the user's IPv4 parameters. It is an optional constructs. The Routing Backus-Naur Form grammar below illustrates the user's IPv4 information model:

```
<ipv4-informatiom>::=<USER_ID><USER_IPV4>
                     <MASK_LENGTH><GATEWAY>
                     <VRF>
```

USER_ID: is the identifier of user. This parameter is unique and mandatory. This attribute is used to distinguish different users. And it collaborates with other IPv4 parameters to present the user's IPv4 information.

USER_IPV4: This attribute specifies the user's IPv4 address, and it's usually used in user plane discovery and ARP reply message.

MASK_LENGTH: This attribute specifies the user's subnet masks lengths which can identify a range of IP addresses that are on the same network.

GATEWAY: This attribute specifies the user's gateway, and it's usually used in User Plane discovery and ARP reply message.

VRF: is the identifier of VRF instance.

4.1.1.3. IPv6 Information Model

The IPv6 information model presents the user's IPv6 parameters. It is an optional constructs. The Routing Backus-Naur Form grammar below illustrates the user's IPv6 information model:

```
<ipv6-information>::=<USER_ID>(<USER_IPV6>
                     <PREFIX_LEN>)|(<PD_ADDRESS><PD_PREFIX_LEN>)
                     <VRF>
```

USER_ID: is the identifier of user. This parameter is unique and mandatory. This attribute is used to distinguish different users. And it collaborates with other IPv6 parameters to present the user's IPv4 information.

USER_IPV6: This attribute specifies the user's IPv6 address, and it usually be used in neighbor discovery (ND discovery).

PREFIX_LEN: This attribute specifies the user's subnet prefix lengths which can identify a range of IP addresses that are on the same network.

PD_ADDRESS: In IPv6 networking, DHCPv6 prefix delegation is used to assign a network address prefix and automate configuration and provisioning of the public routable addresses for the network. This attribute specifies the user's DHCPv6 prefix delegation address, and it's usually used in neighbor discovery (ND discovery).

PD_PREFIX_LEN: This attribute specifies the user's DHCPv6 delegation prefix length, and it's usually used in neighbor discovery (ND discovery).

VRF: is the identifier of VRF instance

4.1.1.4. QoS Information Model

In CU separation BNG, the Control-Plane (CP) generates the QoS table base on UP's bandwidth resources and specific QoS requirements of user's services. This table contains a set of QoS matching rules such as user's committed information rate, peak information rate, committed burst size, etc. And it is an optional constructs. The Routing Backus-Naur Form grammar below illustrates the user's qos information model:

```
<qos-information>::=<USER_ID>
                    (<CIR><PIR><CBS><PBS>)
                    [<QOS_PROFILE>]
```

USER_ID: is the identifier of user. This parameter is unique and mandatory. This attribute is used to distinguish different users. And it collaborates with other qos parameters to present the user's qos information.

CIR: In BNG network, the Committed Information Rate (CIR) is the bandwidth for a user guaranteed by an internet service provider to work under normal conditions. This attribute is used to indicate the user's committed information rate, and it usually collaborates with other qos attributes (such as PIR, CBS, PBS, etc) to present the user's QoS profile.

PIR: Peak Information Rate (PIR) is a burstable rate set on routers and/or switches that allows throughput overhead. This attribute is used to indicate the user's peak information rate, and it usually collaborate with other QoS attributes (such as CIR, CBS, PBS, etc) to present the user's QoS profile.

CBS: The Committed Burst Size (CBS) specifies the relative amount of reserved buffers for a specific ingress network's forwarding class queue or egress network's forwarding class queue. This attribute is used to indicate the user's committed burst size, and it usually collaborates with other qos attributes (such as CIR, PIR, PBS, etc) to present the user's QoS profile.

PBS: The Peak Burst Size (PBS) sepcifies the maximum size of the first token bucket. This attribute is used to indicate the user's peak burst size, and it usually collaborate with other qos attributes (such as CIR, PIR, CBS, etc) to present the user's QoS profile.

QOS_PROFILE: This attribute specifies the standard profile provided by the operator. It can be used as a QoS template which is defined

as a list of classes of services and associated properties. The properties may include:

- o Rate-limit: used to rate-limit the class of service. The value is expressed as a percentage of the global service bandwidth.

- o latency: used to define the latency constraint of the class. The latency constraint can be expressed as the lowest possible latency or a latency boundary expressed in milliseconds.

- o jitter: used to define the jitter constraint of the class. The jitter constraint can be expressed as the lowest possible jitter or a jitter boundary expressed in microseconds.

- o bandwidth: used to define a guaranteed amount of bandwidth for the class of service. It is expressed as a percentage.

4.1.2. Interface Related Information

This model contains the necessary information for the interface. It is used to indicate which kind of service can be supported by this interface. The Routing Backus-Naur Form grammar below illustrates the interface related information model:

```
 <interface-related-infor-model>::=<interface-information>

 <interface-information>::=<IFINDEX><BAS_ENABLE>
                           <service-type>

 <service-type>::=<PPP_Only><IPV4_TRIG>
                  <IPV6_TRIG><ND-TRIG>
                  <ARP_PROXY>
```

4.1.2.1. Interface Information Model

The interface model mentioned here is a logical construct that identifies a specific process or a type of network service. In CU separation BNG network, the Control-Plane (CP) generates the Interface-Infor table based on the available resources, which are received from the User-Plane (UP), and the specific requirements of user’s services.

The Routing Backus-Naur Form grammar below illustrates the interface information model:

```
<interface-information>::=<IFINDEX><BAS_ENABLE>
                          <service-type>

<service-type>::=<PPP_Only><IPV4_TRIG>
                 <IPV6_TRIG><ND-TRIG>
                 <ARP_PROXY>
```

IFINDEX: The IfIndex is the 32-bit index assigned to the interface by the device as specified by the Interfaces Group MIB [RFC2863]. The ifIndex can be utilized within a management domain to map to an actual interface, but it is also valuable in public applications. The ifIndex can be used as an opaque token to discern which interface of User-Plane is providing corresponding service for specific user.

BAS_ENABLE: This is a flag, and if it is TRUE, the BRAS is enabled on this interface.

PPP_Only: This is a flag, and if it is TRUE, the interface only supports PPP user.

IPV4_TRIG: This is a flag, and if it is TRUE, the interface supports that the user can be triggered to connect the internet by using IPv4 message.

IPV6_TRIG: This is a flag, and if it is TRUE, the interface supports that the user can be triggered to connect the internet by using IPv6 message.

ND-TRIG: This is a flag, and if it is TRUE, the interface supports that the user can be triggered to connect the internet by using neighbor discovery message.

ARP_PROXY: This is a flag, and if it is TRUE, the ARP PROXY is enabled on this interface.

4.1.3. Device Related Information

The device related information model presents the attributes which related to specific device. For example the control plane can manage and distribute the users, who belong to same subnet, to some specific devices. And then the user plane’s devices can provide corresponding service for these users. The Routing Backus-Naur Form grammar below illustrates the device related information model:

```
<device-related-infor-model>::=<address-field-distribute>

<address-field-distribute>::=<ADDRESS_SEGMENT><ADDRESS_SEGMENT_MASK>
                             <ADDRESS_SEGMENT_VRF><NEXT_HOP>
                             <IF_INDEX><MASK_LENGTH>
```

4.1.3.1. Address field distribute Table

In CU separation BNG information model, the Control-Plane (CP) generates and sends this Address field distribute table to UP. Based on this table, the user-plane's devices can be divided into several blocks, and each block is in charge of working for users with the same subnet. The Routing Backus-Naur Form grammar below illustrates the address field distribute information model:

```
<address-field-distribute>::=<ADDRESS_SEGMENT><ADDRESS_SEGMENT_MASK>
                             <ADDRESS_SEGMENT_VRF><NEXT_HOP>
                             <IF_INDEX><MASK_LENGTH>
```

4.2. Information Model for User Plane

This section describes information model for the interface of User Plane (UP). As mentioned in section 3, the UP is a network edge and user policy implementation component. It supports: Forwarding plane functions on traditional BNG devices, including traffic forwarding, QoS, and traffic statistics collection and Control plane functions on traditional BNG devices, including routing, multicast, and MPLS.

In CU separation BNG information model, the CP generates tables and provides the rules. The UP plays two roles:

1. It receives these tables, parses it, and matches these rules, then performs corresponding actions.

2. It also generates several tables to report the available resources (such as usable interfaces, etc) and statistical information to CP.

The Routing Backus-Naur Form grammar below illustrates the User Plane information model:

```
<user-plane-information-model>::=<port-resources-infor-model>
                                  <traffic-statistics>

port-resource-information>::=<IF_INDEX><IF_NAME>
                             <IF_TYPE><LINK_TYPE>
                             <MAC_ADDRESS><IF_PHY_STATE>
                             <MTU>
<traffic-statistics-information>::=<USER_ID><STATISTICS_TYPE>
                                   <INGRESS_STATIISTICS_PACKETS>
                                   <INGRESS STATISTICS_BYTES>
                                   <EGRESS_STATISTICS_PACKETS>
                                   <EGRESS_STATISTICS_BYTES>
```

4.2.1. Port Resources of UP

The User Plane can generate the network resource table, which contains a bunch of attributes to present the available network resources, for example the usable interfaces.

The Figure below illustrates the Port Resources Information Table of User-Plane:

```
<port-resource-information>::<IF_INDEX><IF_NAME>
                             <IF_TYPE><LINK_TYPE>
                             <MAC_ADDRESS><IF_PHY_STATE>
                             <MTU>
```

IFINDEX: IfIndex is the 32-bit index assigned to the interface by the device as specified by the Interfaces Group MIB [RFC2863]. The ifIndex can be utilized within a management domain to map to an actual interface, but it is also valuable in public applications. The ifIndex can be used as an opaque token to discern which interface of User-Plane is available.

IF_NAME: the textual name of the interface. The value of this object should be the name of the interface as assigned by the local device and should be suitable for use in commands entered at the device's 'console'. This might be a text name, such as 'le0' or a simple port number, such as '1', depending on the interface naming syntax of the device. If several entries in the ifTable together represent a single interface as named by the device, then each will have the same value of ifName.

IF_TYPE: the type of interface, such as Ethernet, GE, Eth-Trunk, etc.

LINK_TYPE: This attribute specifies the type of link, such as point-to-point, broadcast, multipoint, point-to-multipoint, private and public (accessibility and ownership), etc.

MAC_ADDRESS: This attribute specifies the available interface's MAC address.

IF_PHY_STATE: The current operational state of the interface. This is an enumeration type node:

1- Up: ready to pass packets;

2- Down

3- Testing: in some test mode;

4- Unknow: status cannot be determined for some reason;

5- Dormant;

6- Not present: some component is missing.

MTU: This attribute specifies the available interface's MTU (Maximum Transmission Unit).

4.2.2. Traffic Statistics Infor

The user-plane also generates the traffic statistics table to report the current traffic statistics.

The Figure below illustrates the Traffic Statistics Infor model of User-Plane:

```
<traffic-statistics-information>::=<USER_ID><STATISTICS_TYPE>
                                    <INGRESS_STATIISTICS_PACKETS>
                                    <INGRESS STATISTICS_BYTES>
                                    <EGRESS_STATISTICS_PACKETS>
                                    <EGRESS_STATISTICS_BYTES>
```

USER_ID: is the identifier of user. This parameter is unique and mandatory. This attribute is used to distinguish different users. And it collaborates with other statistics parameters such as ingress packets, egress packets, etc, to report the user's status profile.

STATISTICS_TYPE: This attributes specifies the traffic type such as IPv4, IPv6, etc.

INGRESS_STATIISTICS_PACKETS: This attribute specifies the Ingress Statistics Packets of specific user.

INGRESS STATISTICS_BYTES: This attribute specifies the Ingress Statistics Bytes of specific user.

EGRESS_STATISTICS_PACKETS: This attribute specifies the Egress Statistics Packets of specific user.

EGRESS_STATISTICS_BYTES: This attribute specifies the Egress Statistics Bytes of specific user.

5. Security Considerations

None.

6. IANA Considerations

None.

7. Normative References

[I-D.gu-nfvrg-cloud-bng-architecture]
Gu, R. and S. Hu, "Control and User Plane Separation Architecture of BNG", draft-gu-nfvrg-cloud-bng-architecture-01 (work in progress), July 2017.

[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>.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <https://www.rfc-editor.org/info/rfc2863>.

[RFC5837] Atlas, A., Ed., Bonica, R., Ed., Pignataro, C., Ed., Shen, N., and JR. Rivers, "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, DOI 10.17487/RFC5837, April 2010, <https://www.rfc-editor.org/info/rfc5837>.

Authors' Addresses

Michael Wang (editor)
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: wangzitao@huawei.com

Rong Gu
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing, Beijing 100053
China

Email: gurong_cmcc@outlook.com

Victor Lopez
Telefonica
Sur 3 building, 3rd floor, Ronda de la Comunicacion s/n
Madrid 28050
Spain

Email: victor.lopezalvarez@telefonica.com

Sujun Hu
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing, Beijing 100053
China

Email: shujun_hu@outlook.com