RTG Working Group
Internet-Draft
Intended status: Informational
Expires: March 29, 2020

R. Raszuk, Ed.
Bloomberg LP

September 26, 2019

IP Traffic Engineering Architecture with Network Programming
draft-raszuk-rtgwg-ip-te-np-00

Abstract

This document describes a control plane based IP Traffic Engineering Architecture where path information is kept in the control plane by selected nodes instead of being inserted into each packet on ingress of an administrative domain. The described proposal is also fully compatible with the concept of network programming.

It is positioned as a complimentary technique to native SRv6 and can be used when there are concerns with increased packet size due to depth of SID stack, possible concerns regarding exceeding MTU or more strict simplicity requirements typically seen in number of enterprise networks. The proposed solution is applicable to both IPv4 or IPv6 based networks.

As an additional added value, detection of end to end path liveness as well as dynamic path selection based on real time path quality is integrated from day one in the design.

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 March 29, 2020.

Table of Contents

1. Background

Ability to steer data over selected topological points often different from default IGP or BGP paths proves to provide substantial advantages to consumers of such data. The construction of controlled transit paths usually is driven by requirements to: offload excessively used default routing paths, construct disjointed paths for live-live dual streaming or create intra or inter-domain data distribution overlays using dynamic real time SLAs criteria often used along with per specific application mapping schema.

In addition to pure topological reasons there are often also requirements for special data flow processing to happen in selected network elements which by default would not be in the data path of the subject flows. Examples of this could be: firewall traffic screening, service function chaining, caching, deep packet inspection, etc ...

While there are some solutions available to allow traffic engineering in domains fully operated by single administrative entity there seems to be lack of proposals which could be used to control interconnections of sites over third party networks or Internet. As part of that category one could also list public cloud tenancies where ability to steer in/out traffic over other then default Internet routing could provide much better SLA characteristics or address some of the non purely technical requirements.

Another category of global networking which can significantly benefit from standards based IP TE solution is unified model of path engineering for Software Defined Wide Area Networks (SDWANs). One of the basic operational principles in selected SDWANs is point to point underlay selection based on the applied SLA characteristics. Adding ability to traffic engineer such underlay flows allows to bypass under performing underlay default paths or congestion points occurring even few autonomous systems away.

2. Terminology

The following abbreviations are used within this document:

- TE - Traffic Engineering
- AF - Address Family
- IPv4 - Internet Protocol version 4
- IPv6 - Internet Protocol version 6
- IGP - Interior Gateway Protocol
- EH - Extension Header
- RIR - Regional Internet Registry
- PCE - Path Computation Element
- UDP - User Datagram Protocol
- BGP - Border Gateway Protocol

- SRH - Segment Routing Header
- OWAMP - A One-way Active Measurement Protocol
- DOH - Destination Option Header
- PE - Provider Edge
- SE - Segment Endpoint
- SID - Segment Identifier (PREFIX+FUNCTION+4bits}
- NMS - Network Management System
- CoS - Class of Service
- PCE - Path Computation Element
- PCEP - Path Computation Element Communication Protocol
- SR-MPLS - Segment Routing with MPLS data plane
- SRv6 - SRv6 Network Programming
- RTT - Round Trip Time
- MTU - Maximum Transmission Unit
- MOS - Mean Opinion Score
- OAM - Operation, Administration, Maintenance
- MPLS - Multiprotocol Label Switching
- GID - Group Identifier

3. Introduction

Proposed architecture described in this specification defines a new forwarding paradigm which allows to create traffic engineered paths either centrally or in a distributed way. With the assistance of local provisioning tools or control plane such ordered set of paths are distributed to those network elements which will participate in data forwarding. In addition to basic packet forwarding the architecture also provides mechanism to execution arbitrary instructions at selected by operator network nodes which can include: routers, switches, firewalls, service processors, hosts etc ...

Authors have taken a clean slate approach to look at the possible options to engineer traffic within given administrative domain boundaries. The solution is applicable to both traditional "underlay" networks as well as administrative domains constructed with "overlays". It is also 100% transparent to operating network elements which would not participate in the traffic engineering solution while maintaining packet's entropy and fast connectivity restoration needs.

The proposed solution is constructed using either building blocks or ideas borrowed from the following technologies:

- Segment Routing Architecture [RFC8402]
- Destination/Source Routing [I-D.ietf-rtgwg-dst-src-routing]
- Generic Packet Tunneling in IPv6 Specification [RFC2473]
- IP Encapsulation within IP [RFC2003]
- Encapsulating IP in UDP [I-D.xu-intarea-ip-in-udp]
- Advertising Segment Routing Policies in BGP [I-D.ietf-idr-segment-routing-te-policy]
- BGP Vector Routing [I-D.patel-raszuk-bgp-vector-routing]
- A Path Computation Element (PCE) Based Architecture [RFC4655]
- PCEP Extensions for Segment Routing [I-D.ietf-pce-segment-routing]
- Topology Independent Fast Reroute using Segment Routing [I-D.ietf-rtgwg-segment-routing-ti-lfa]
- A One-way Active Measurement Protocol (OWAMP) [RFC4656]

It is also fully compatible with following specifications to embed network programming concept as is define in the below documents while in the same time provides a new alternate encoding model:

- Internet Protocol, Version 6 (IPv6) Specification [RFC8200]
- IPv6 Segment Routing Header (SRH) [I-D.ietf-6man-segment-routing-header]
- IPv4 Extension Headers and Flow Label [I-D.herbert-ipv4-eh]

- IPv4 Extension Headers and UDP Encapsulated Extension Headers [I-D.herbert-ipv4-udpencap-eh]

For the intradomain Traffic Engineering needs the introduced overhead is of fixed size and regardless of the amount of segment endpoints or links which need to be traversed as part of the engineered path is constant and equal to 28 octets for IPv4 and 40 octets for IPv6. If additional segment end or path end instructions are to be added into additional headers an extension header size will need to be included. Instructions however, can also be embedded into SID destination or reside above the encapsulation header. In those cases,3 the total length of the overhead remains fixed as stated above.

Interdomain Traffic Engineering depending on the deployment model could result in additional fixed 12 octets of the overhead. Overlay deployment models will be discussed in more details in below Data Plane section.

While the described architecture is applicable to both IPv4 and IPv6 networks the proposal could be split into separate documents each focusing on specifics corresponding only to a single address family if the community expresses such preference. However, due to the number of common AF agnostic characteristics it is advised to keep it within a single document.

Since the support of EH in IPv4 is planned to be introduced with a rather limited scope, the end segment or end path instructions could end up using other extension header types (for example: Destination Options) in IPv4 packets or could be encoded into the destination addresses itself. It has to be noted that IPv4 packets could be encapsulated in IPv6 when carried across a given domain. The document describes how the concept of network programming can be applied without use of extension headers.

The proposal does not enforce any new dependencies on IP address block allocations and is in full alignment to the current IETF and RIRs address structure and allocation policies.

The core of the defined functionality does not require any new protocol extensions. The solution attempts to maximize and reuse extensions already defined. If more optimal protocol solutions applicable to any of the defined functional blocks surface additional work will take place in corresponding area/wg.

Described architecture does not belong to segment routing family even if some terminology used to describe the proposal have been borrowed from it. Major difference is that by design it uses control plane or management plane to install per path state in the transit nodes

participating in the engineering of data paths instead of encoding set of TE midpoints into each packet on ingress.

While scaling aspects of any solution is a very important factor it needs to be put in perspective to the operational requirements as well as characteristics of the designs. It also needs to be noted that even basic IP routing is based on state in the network elements and scale of Internet routing is usually orders of magnitude higher then state of most traffic engineering needs. While looking at scaling factors of the complete solution variable size per packet overhead needs to be weighted against cost of additional per path fixed size state in control and data plane.

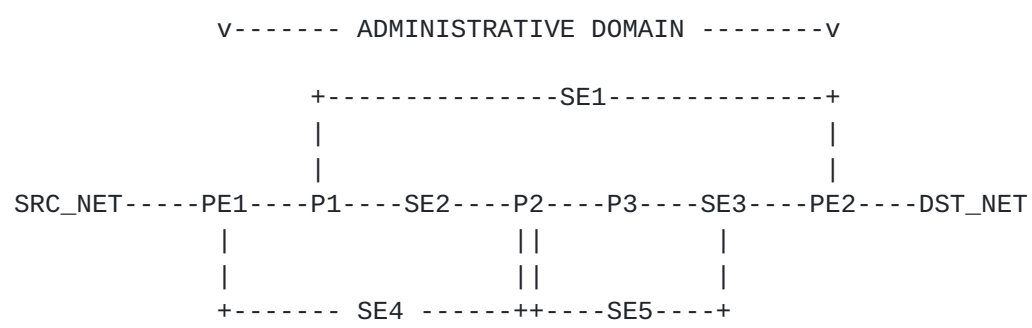
IP TE+NP design while allowing operator to create centrally computed and distribute strict end to end paths in number of deployments can be used in fully distributed mode. Traffic steering decisions can autonomously take place in any TE midpoint what is particularly useful with all SLA or performance based routing deployments.

If there is any comparison to be made between SR and IP TE+NP architectures putting aside other fundamental differences would be the assumption of constructing segment routing paths only by Binding SIDs (divided into static and variable parts) and only encoding them at each segment endpoint in least significant bits of source and destination address of the outer IP header.

4. Functional Description

For the purpose of this document the following term definitions will be used in capital letter notation:

- CLASSIFIER_ID: Identifier to set of rules used for mapping flows to TE paths. Length - 4 octets.
- PATH_GID_PFX: routable node prefix + locally significant PATH_GID value. Length - 4 or 16 octets.
- SID: routable node prefix + opt. function + opt. parameters + 4 bits (Lookup Type) - Length - 4 or 16 octets.
- PATH_LIST: ordered list of SIDs. Length N x 4 or N x 16 octets. N min = 1.



Basic Network Topology

Figure 1

Consider basic two requirements to be applied for some class of transit traffic T1 and T2:

- T1: PATH_A1: PE1--SE1--PE2

- T2: PATH_A2: PE1--SE4--SE5--PE2

IGP metric of all interfaces is set to 10 except interfaces attached to SE1, SE4 and SE5 nodes which are of metric of 100.

The shortest default path, in the example above, between PEs is: PE1--P1--SE2--P2--P3--SE3--PE2

In order to accomplish the stated requirements (for traffic classes T1 and T2 defined above) the following ordered path lists are created in the control plane and either locally configured on both ingress and segment endpoints or distributed by any of the control plane protocols discussed in subsequent sections:

CLASSIFIER_ID: T1	CLASSIFIER_ID: T2
PATH_GID: A1	PATH_GID: A2
PATH_LIST: SE1, PE2	PATH_LIST: SE4, SE5, PE2

There are few core elements of the design as listed below:

- Each PATH_GID_PFX contains unique routable IP prefix from one of the loopbacks of the corresponding ingress PE followed by PATH_GID value (PATH GROUP-ID). For example, if the loopback's prefix is a /64 IPv6 prefix there can be 2^64 unique paths originated at a given PE. If the loopback address is a /16 IPv4 prefix (for example used from [RFC1918] space) there can be 2^16 paths initiating at a given IPv4 PE. The choice of mapping scheme is local to the ingress PE and is assigned by the operator. Let's observe that in most cases to describe reachability to the

PATH_GID_PFX only a single IGP loopback prefix may need to be advertised from any ingress PE. It is also highly recommended that such loopback prefixes configured on all ingress nodes (ingress PEs) to be sourced from the same address block such that it can be described by single aggregate prefix.

- Each PATH_LIST consists of a number of SID elements. Each SID is a unique routable IP address from one of the loopbacks of the corresponding Segment Endpoint (SE) node. For example, if the loopback's prefix is a /64 IPv6 prefix there can be 2^(64-4) unique SID terminating on a given node. If the loopback address is a /16 IPv4 prefix (for example used from [RFC1918] space) there can be 2^(16-4) SIDs present on a given IPv4 node. As defined, a SID may represent not only a node's topological location in the network (via IP prefix reachability), but it may also, optionally, contain embedded functions with their parameters. In order to even further help the forwarding layer within a given domain, the last four bits can be consistently chosen to describe the lookup type required to correctly switch a given packet.

- Upon ingress to the domain, and after classification, packets are encapsulated into an additional outer IP header with the following elements corresponding to the non-default forwarding requirements:

Classified as T1 flows:	Classified as T2 flows:
Source address: PATH_A1_PFX	Source address: PATH_A2_PFX
Destination address: SID_SE1	Destination address: SID_SE4

In the case of IPv6 the encapsulation for the basic TE only requirement will consist of applying a fixed IPv6 40 octets header containing source and destination address as described above, the copy of original flow label, the copied and decremented hop limit count and, depending on the local policy, CoS setting (copy of original or setting local value). In the case of IPv4 scenario the 20 octets IP header will contain TTL copied and decremented from original packet, CoS (copy of original or setting of local value) + 8 octets UDP header allowing to improve entropy of flows bundled to travel within the provided TE path yet to still be able to utilize any ECMP along the path list.

- Encapsulated packets are natively forwarded via the network (by and through P nodes) till they arrive at the destination Segment Endpoint where the destination address gets swapped to the new destination address from the PATH_LIST kept in the local control and data plane. The lookup which returns new destination of the packet is a source-destination based lookup using both PATH_GID_PFX (with PATH_GID being encoded in the least significant

bits of the source address of the packet) and SID (encoded in destination address of the packet). That allows to maintain very good scaling property of the solution without SID state or SID number explosion. All functions descriptions which are encoded in the SIDs can be reused across any segment endpoint, if required, as they have only local significance.

- When packets arrive at the destination PE (last Segment Node) a similar lookup is performed which returns NULL as next segment what in turn will result into the decapsulation of the packet and regular destination based lookup of the destination address present in the inner IP header. As noted, a local optimization allows to encode the local lookup type in last 4 bits of any SID hence allowing to skip the first lookup if such optimization is enabled by the operator.

- The described lookup table is instantiated and maintained by either the control plane or by the local configuration of sets of path lists. For any given segment end node, only local SIDs (those where most significant prefix bits match locally configured prefixes) are populated to data plane along with PATH_GIDs they are attached to. That setup is all what is required to provide basic IP TE service. More elaboration on other SID values will be described within the embedded network programming section below.

5. Control plane

The proposed solution is based on classic IP reachability and does not require any new control plane extension. In its basic form, and in order to setup a few TE paths across the sample network in Figure 1, all is required is to apply two path lists on ingress and egress nodes as well as on three segment endpoints.

However depending on the required TE scale, on the network size, as well as on the TE path complexity, real production deployments will likely utilize automation in order to provision such configurations. Local NMS can be used successfully to provision all participating segment nodes with proper set of path lists. A separate document specification describing yang models for the solution will be provided.

Another alternative to propagate set of path lists can be enabled by using segment routing extensions for PCEP as described in [I-D.ietf-pce-segment-routing]. For the basic TE use cases path lists used are identical to SID lists for SR-MPLS or SRv6 technologies. The logic used by PCE to compute such paths within given domain can be directly leveraged by this architecture. The defined SR-ERO sub-object can be directly used to propagate path

lists not also to ingress and egress nodes, but also to all segment end points participating in given path list transit.

The described above methods offer a manual or automated way to distribute path lists from central locations using directed TCP sessions to all participating network elements. However, in order to even further reduce the complexity and increase rate of path list propagation across any domain a point to multipoint solution could be utilized. Also here like in former cases, existing extensions are available - specifically extension to BGP in order to Advertise Segment Routing Policies as described in [I-D.ietf-idr-segment-routing-te-policy]. Detailed encoding examples will be provided in subsequent versions of this document.

BGP constructs used for SR Policies propagation to ingress nodes can be used as is in order to propagate analogues path lists to all participating nodes in the network. A new SAFI has been defined (codepoint 73) to separate such propagation from any other address family as well as to uniquely define the NLRI format. For the purpose of dissemination path lists NLRI 4 octet Policy Color will carry CLASSIFIER_ID and 4 or 16 octet Endpoint field will carry the PATH_GID value. If PATH_GID is shorter then 4 or 16 octets the most significant bits of Endpoint field will be set to zero. Ordered list of SIDs will be propagated using Segment List Sub-TLVs (Type 3 for IPv4 and Type 9 for IPv6). Optionally other Sub-TLVs can be also included with propagation of path lists - for example: Preference Sub-TLV, Priority Sub-TLV, Name Sub-TLV etc...

As intra-domain BGP usually employs route reflection it is likely that participating nodes may receive many more path lists then required to be kept or installed into data plane. There are two optional solutions to reduce amount of unnecessary control plane information required to be kept any participating node which when applied on ingress will result in path lists inbound filtering: use of route target extended communities or filtering based on intersection of locally configured IP prefixes with either prefix part of Endpoint NLRI or prefix part of any SID carried in Segment List Sub-TLVs. Even if all path lists received would be accepted by BGP for operational and troubleshooting needs only those which are locally significant will be installed into data plane.

6. Data plane

There are three IP TE+NP deployment scenarios which may require different data plane encoding specific to the type of connectivity available for ingress, egress and TE transit nodes. The following three categories are covered by this specification:

Cat I - deployment within service provider or enterprise where all participating nodes are interconnected via links operated by the same organization using addressing scheme in control of such organization

Cat II - deployment where participating sites are interconnected over third party operated networks, where participating in IP TE nodes could allocate sufficient address block to be used as source address and still permit to encode entire PATH_GID space of the size chosen by the operator in the least significant bits of the addresses of such nodes

Cat III - deployment where participating nodes are interconnected over third party operated infrastructure where all what has been granted to such nodes are either host routes or prefixes with not enough bits left to encode PATH_GID

The below building blocks constitute the required minimum data plane functionality for this architecture:

Source+Destination Routing [I-D.ietf-rtgwg-dst-src-routing]

Choice of encapsulation:

IPv4 in IPv4+UDP [I-D.xu-intarea-ip-in-udp]

IPv6 or IPv4 in IPv6 [RFC2473]

The selection of normal destination only lookup or source+destination lookup is triggered by lookup of the destination address. Network elements which do not participate in the IP TE+NP service will perform destination only lookup and forward the packets. Network elements which do participate in the new architecture will perform destination address check and if that address matches the local prefix assigned to IP TE+NP service source+destination lookup will take place, otherwise standard destination only lookup will be performed.

For deployments falling into Cat III as classified above available address space does not allow to encode the PATH_GID as part of the source address. Therefore in such scenarios it is recommended to use additional GRE encapsulation where PATH_GID would be encoded in the 4 octet key field.

Proposed above GRE header encoding applicable only to Cat III deployments should in addition to already defined rules also follow described GRE encoding in the following specifications:

IPv4 in IPv4+UDP+GRE [RFC8086]

IPv4 or IPv6 in IPv6+GRE [RFC7676]

In Cat III deployments when source+destination lookup is performed PATH_GID from GRE key field should be used instead of packet's source address. For the case of IPv6 packet encapsulation 12 octets of zeros should be locally prepended to the key to perform source+destination lookup.

7. Network Programming

Control Plane Assisted Traffic Engineering is fully compatible with functions as described in [I-D.ietf-spring-srv6-network-programming] with one major difference. Instead of always inserting SIDs in a form of SRH on ingress and into each packet, there are few alternative ways proposed by this specification. One of them assumes that information about selected functions is added to the packet by the penultimate node of a given segment end node hop. SIDs defined in this document consist of routable prefix part and locally significant function/instruction part with optional parameters and lookup type. They can be 32 bit in the case of IPv4 or 128 bit long in the case of IPv6 with the length of the routable part being a local choice of the operator.

PATH_GID+SID lookup can return a simple pointer to the next segment node or can also result in any other local packet processing chain. While the routable part of the SID has domain-wide significance the function part has only local meaning to a given node on which it has been instantiated.

It needs to be observed that some network functions can, for practical purposes, only be instantiated of the ingress to the domain and as such can be attached to the packet during initial encapsulation by use of Segment Routing Header (SRH). The examples of such functions include L3VPN or EVPN or L2VPN demux labels which are to be used when packets arrive to the other side of the domain with or without TE.

To further simplify the processing of packets via the segment end nodes and relax the requirement for each transit node to inspect SRH (when added by ingress node) the document will recommend that each operator in the domain will reserve the last 4 bits of the SID to explicitly indicate the required lookup type (aka switching vector) on the outer packet header to occur:

Decimal value	Lookup Type
0	SRC-DST lookup only
1	SRH inspection + SRC-DST lookup
2	Decapsulation + Global lookup
3	SRH inspection + Decapsulation
4	reserved
..	..
15	reserved

Table 1: Recommended allocation of domain wide IPv6 SID_PFX actions

As this specification is only of informational category the proposed recommendation has non binding character and can be locally replaced by any different schema as chosen by the operator and made possible by implementations. For example the 4 bits may be placed in any other offset after the SID's routable prefix part. The proposed SID Lookup Types do not replace or interfere in any way with SRH SRv6 Endpoint Behaviors as defined in [I-D.ietf-spring-srv6-network-programming].

As defined today [RFC8200] mandates to inspect and process all extension headers in the IPv6 packet when packet's destination matches any of the locally configured IPv6 address. Therefor if present SRH will need to be inspected and processed at each segment end even if it is known by control plane that it does not contain any instructions to be executed at a given network element ahead of time. Authors will however still encourage recommended SID structure to be used for either troubleshooting reasons or for the future when IPv6 specification will relax the EH handling rules to accomodate such new deployment models.

As an alternative solution to avoid unnecessary processing of extension header by nodes which are not required to do so implementation can treat SID with last four bits set to zero as none local destination address. In such scenario source+destination lookup will instead of triggering local extension header processing invoke destination IPv6 NAT function as defined in [RFC6296]. The NAT rules which will be pre-programmed using information contained in the PATH_LIST will effectively result in destination address swap. Such NAT translation is to be of unidirectional character can can remain fully stateless.

Described solution also directly applies to the case of IPv4 in IPv6 encapsulation.

In the case of IPv4 in IPv4+UDP encapsulation the basic behaviour of embedding functions in SIDs does not change. However as to the moment of this writing the proposed IPv4 header extensions [I-D.herbert-ipv4-eh] and [I-D.herbert-ipv4-udpencap-eh] may only allow limited number of extension headers to be used (Hop-by-Hop Options and Destination Options). As such the recommended allocation table in the case of IPv4 requires slight adjustment:

Decimal value	Lookup Type
0	SRC-DST lookup only
1	DOH inspection + SRC-DST lookup
2	Decapsulation + Global lookup
3	DOH inspection + Decapsulation
4	reserved
..	..
15	reserved

Table 2: Recommended allocation of domain wide IPv4 SID_PFX actions

The specific syntax of Destination Option Header encoding when used with IPv4 encapsulation will be defined in subsequent versions of this document.

Existing services (ex: MPLS-VPNs [RFC4364]) are fully compatible as-is without any modifications to be transported over described IP TE architecture. Existing MPLS label can be used as service demux with full replacement of MPLS-Transport to IP-TE transport. In such scenario there is no longer need to rename service demux value into some new nomenclature to artificially force it to fit into SID space. Substitute of MPLS transport with new IP TE transport is essentially treated as basic IP-in-IP encapsulation and is seamless to the upper layer applications. That however in no way can prevent invention of new native services to only use new network programming paradigm.

8. Active Path Probing

One of the critical network metrics for a lot of applications running on the network is not only ability to reach the destination in a relatively congestion free fashion, but also the quality of the path which is traversed towards a destination. The latter is, unfortunately, very seldom used as selection criteria in number of TE implementations. Here authors recommend that, from day one, the operator has an option in order to define the minimum path quality metrics before it is considered for actual data plane use as both

relative or absolute set of values. Comparison with non TE path or other TE paths end to end metrics should also be available.

Today's network technologies focus on local protection as reaction to adjacent link or node failures. At the same time, there is a significant concern that they lack detection of any malfunctions of network elements' internal data plane itself which, as proven in number of production deployments, does occur.

Moreover, it also needs to be observed that most if not all of commonly used routing protocols focus on assuring loop free destination reachability via shortest or best path measured with static metrics without any consideration given to actual quality of end to end path towards given destination.

Traffic engineering allows to enable real time SLA evaluation of various TE paths. Results of such measurements can be used to automatically map traffic to such TE transport. Architecture described by this specification integrates such functionality provided an operator chooses to enable it.

It needs to be noted that packets used for diagnostics must traverse the exact same data plane and should be encapsulated in the identical header as the user packets. Such measurements not only detect path parameters but also end to end path availability.

While (N times path RTT - N times local detection interval) slower from local protection for vast majority of applications such end to end path liveness detection rate is both sufficient for applications and much simpler to implement and operate. It is also more attractive due to increased spectrum of types of failures which can be detected. Removed complexity required to be employed (example: node protection repair of adjacent segment nodes) is also an important consideration.

The choice of path probing protocol is left as the local operator's decision. However, it needs to be observed that such protocol suite should allow fast liveness detection as well as end to end path quality measurements reported to path headend (typically a network ingress node) as RTT, Jitter, Delay, MOS parameters as well as max MTU and sweep MTU path validation.

It is also completely valid to use more than one protocol - each in different frequency setting. As an example, one could use BFD multihop [RFC5883] with hardware offload to detect end to end path liveness while in the same time apply OWAMP [RFC4656] to collect more unidirectional path quality metrics. Recommendation for a single

integrated path liveness and quality reporting protocol will also be described in a separate IETF specification.

8.1. TI-LFA Local Protection

As stated in the TI-LFA specification for networks supporting segment routing [I-D.ietf-rtgwg-segment-routing-ti-lfa], protection of SR policy midpoints involves adjustments to segment list carried in the packets as well as proper selection of repair path in order to assure that protected packets can successfully reach the next SR policy segment node.

Based on the control plane distribution of complete PATH_LIST, similar protection is possible in the described architecture. Without any additional requirements to adjust any other fields in the packet header only destination address can be swapped. Current destination can be replaced by subsequent node's destination address on the PATH_LIST upon detection of neighboring node failure. That operation however, requires to maintain per path state at PLRs what while certainly possible may not be operator's preference.

Enabling local protection in segment engineered IP networks is clearly possible, however it needs additional processing and control plane information to be distributed and present on all nodes in the domain. Protection PATH_LISTs can be either computed centrally or by any node in the domain (including PLRs). Authors recommend this to remain a local operator decision and at the same time encourage to use end to end path protection scheme as first preference.

9. Solution advantages

The following key advantages can be used to characterize the described architecture:

- Native TE support for IPv4 and IPv6
- Very efficient use of available address space - no requirement for any new address allocations
- IGP impact - single prefix injection from ingress nodes of length chosen by operator
- Ability to aggregate injected prefixes at area or domain boundary with no impact to functionality
- No extensions to ISIS or OSPF routing protocols required

- Reuse of commonly available components (SRC-DST routing and IPinIP encapsulation)
- Integrated end to end path validation for reachability and quality
- For basic TE and PATH_LIST SID integrated network programming functions fixed overhead of 28 octets for IPv4 and 40 octets for IPv6.
- Full compatibility with SRH from SRv6 Network Programming concept
- No per user data flow state in any network element of the network except ingress (mapping only)
- No packet header size growth with the growing number of TE segment endpoints policies
- Support in all available hardware - no need for any new operations on the packet headers
- TI-LFA support when end to end path protection will not be sufficient
- Full native support of network services: L2VPNs, L3VPNs, EVPNs etc with single SID in SRH or native service level encapsulation
- Support of ingress, egress or transit nodes with available only single host address available on each such system

10. OAM

As result of use of IP encapsulation both traceroute as well as ping are natively supported within a given domain boundaries. ICMP or UDP OAM probes will be encapsulated in the exact same IPv4 or IPv6 header as user data packets therefore all replies will be sent to the domain ingress node.

No modifications to additional extension headers or even their presence is required for correct OAM operations.

If an OAM packet is originated externally to the domain, the ingress node will need to act as OAM proxy in relaying the responses to its original sources.

11. Deployment considerations

The solution is defined to be fully customizable by the operator. The path engineering as well as choice of numbering will likely differ domain to domain.

As all packets subject to this specification carry in their source address immutable PATH_GID. Together with locally assigned SIDs no further extensions are necessary to identify specific path flows at any point in the domain. The same tuple PATH_GIDs + SIDs can also be used to identify any path statistics (netflow records) at any point in the domain.

12. Security considerations

No new security issues are introduced by this specification.

13. IANA Considerations

No IANA allocations are required by this specification.

14. Acknowledgements

Authors would like to thank Tony Li, Stefano Previdi, Dirk Steinberg and Francois Clad for their valuable review and comments.

15. References

15.1. Normative References

[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>.

[RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, DOI 10.17487/RFC2003, October 1996, <https://www.rfc-editor.org/info/rfc2003>.

[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>.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <https://www.rfc-editor.org/info/rfc2473>.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <https://www.rfc-editor.org/info/rfc2784>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[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>.

[RFC7676] Pignataro, C., Bonica, R., and S. Krishnan, "IPv6 Support for Generic Routing Encapsulation (GRE)", RFC 7676, DOI 10.17487/RFC7676, October 2015, <https://www.rfc-editor.org/info/rfc7676>.

[RFC8086] Yong, L., Ed., Crabbe, E., Xu, X., and T. Herbert, "GRE-in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086, March 2017, <https://www.rfc-editor.org/info/rfc8086>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

15.2. Informative References

[I-D.herbert-ipv4-eh]
Herbert, T., "IPv4 Extension Headers and Flow Label", draft-herbert-ipv4-eh-01 (work in progress), May 2019.

[I-D.herbert-ipv4-udpencap-eh]
Herbert, T., "IPv4 Extension Headers and UDP Encapsulated Extension Headers", draft-herbert-ipv4-udpencap-eh-01 (work in progress), March 2019.

[I-D.ietf-6man-segment-routing-header]
Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-23 (work in progress), September 2019.

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-07 (work in progress), July 2019.

[I-D.ietf-pce-segment-routing]
Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "PCEP Extensions for Segment Routing", draft-ietf-pce-segment-routing-16 (work in progress), March 2019.

[I-D.ietf-rtgwg-dst-src-routing]
Lamparter, D. and A. Smirnov, "Destination/Source Routing", draft-ietf-rtgwg-dst-src-routing-07 (work in progress), March 2019.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., Francois, P., daniel.voyer@bell.ca, d., Clad, F., and P. Camarillo, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-01 (work in progress), March 2019.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-03 (work in progress), September 2019.

[I-D.patel-raszuk-bgp-vector-routing]
Raszuk, R., Patel, K., Pithawala, B., Sajassi, A., Osborne, E., Jalil, L., and J. Uttaro, "BGP vector routing.", draft-patel-raszuk-bgp-vector-routing-07 (work in progress), May 2016.

[I-D.xu-intarea-ip-in-udp]
Xu, X., Assarpour, H., Ma, S., daniel.bernier@bell.ca, d., Dukes, D., Lee, Y., and F. Yongbing, "Encapsulating IP in UDP", draft-xu-intarea-ip-in-udp-07 (work in progress), May 2018.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <https://www.rfc-editor.org/info/rfc4655>.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <https://www.rfc-editor.org/info/rfc4656>.

[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>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

Author's Address

Robert Raszuk (editor)
Bloomberg LP
731 Lexington Ave
New York City, NY 10022
USA

Email: robert@raszuk.net