

PANRG
Internet-Draft
Intended status: Informational
Expires: April 21, 2019

T. Enhardt
TU Berlin
C. Kraehenbuehl
ETH Zuerich
October 18, 2018

A Vocabulary of Path Properties
draft-enghardt-panrg-path-properties-00

Abstract

This document defines and categorizes information about Internet paths that an endpoint might have or want to have. This information is expressed as properties of paths between two endpoints.

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 21, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Domain Properties	3
3. Backbone Properties	3
4. Dynamic Properties	4
5. Security Considerations	5
6. IANA Considerations	5
7. Informative References	6
Acknowledgments	6
Authors' Addresses	6

1. Introduction

Because the current Internet provides an IP-based best-effort bit pipe, endpoints have little information about paths to other endpoints. A Path Aware Network exposes information about one or multiple paths through the network to endpoints, so that endpoints can use this information.

Such path properties may be relatively dynamic, e.g. current Round Trip Time, close to the origin, e.g. nature of the access technology on the first hop, or far from the origin, e.g. list of ASes traversed.

Usefulness over time is fundamentally different for dynamic and non-dynamic properties. The merit of a momentary measurement of a dynamic path property diminishes greatly as time goes on, e.g. the merit of an RTT measurement from a few seconds ago is quite small, while a non-dynamic path property might stay relevant, e.g. a NAT can be assumed to stay on a path during the lifetime of a connection, as the removal of the NAT would break the connection.

Non-dynamic properties are further separated into (local) domain properties related to the first few hops of the connection, and backbone properties related to the remaining hops. Domain properties expose a high amount of information to endpoints and strongly influence the connection behavior while there is little influence and information about backbone properties.

Dynamic properties are not separated into domain and backbone properties, since most of these properties are defined for a complete path and it is difficult and seldom useful to define them on part of the path. There are exceptions such as dynamic wireless access properties, but these do not justify separation into different categories.

This document addresses the first of the questions in Path-Aware Networking [I-D.irtf-panrg-questions], which is a product of the PANRG in the IRTF.

2. Domain Properties

Domain path properties usually relate to the access network within the first hop or the first few hops. Endpoints can influence domain properties for example by switching from a WiFi to a cellular interface, changing their data plan to increase throughput, or moving closer to a wireless access point which increases the signal strength.

A large amount of information about domain properties exists. Properties related to configuration can be queried using provisioning domains (PvDs). A PvD is a consistent set of network configuration information as defined in [RFC7556], e.g., relating to a local network interface. This may include source IP address prefixes, IP addresses of DNS servers, name of an HTTP proxy server, DNS suffixes associated with the network, or default gateway IP address. As one PvD is not restricted to one local network interface, a PvD may also apply to multiple paths.

Access Technology present on the path: The lower layer technology on the first hop, for example, WiFi, Wired Ethernet, or Cellular. This can also be more detailed, e.g., further specifying the Cellular as 2G, 3G, 4G, or 5G technology, or the WiFi as 802.11a, b, g, n, or ac. These are just examples, this list is not exhaustive, and there is no common index of identifiers here. Note that access technologies further along the path may also be relevant, e.g., a cellular backbone is not only the first hop, and there may be a DSL line behind the WiFi.

Monetary Cost: This is information related to billing, data caps, etc. It could be the allowed monthly data cap, the start and end of a billing period, the monetary cost per Megabyte sent or received, etc.

3. Backbone Properties

Backbone path properties relate to non-dynamic path properties that are not within the endpoint's domain. They are likely to stay constant within the lifetime of a connection, since Internet "backbone" routes change infrequently. These properties usually change on the timescale of seconds, minutes, or hours, when the route changes.

Even if these properties change, endpoints can neither specify which backbone nodes to use, nor verify data was sent over these nodes. An endpoint can for example choose its access provider, but cannot choose the backbone path to a given destination since the access provider will make their own policy-based routing decision.

Presence of certain device on the path: Could be the presence of a certain kind of middlebox, e.g., a proxy, a firewall, a NAT.

Presence of a packet forwarding node or specific Autonomous System on a path:

Indicates that traffic goes through a certain node or AS, which might be relevant for deciding the level of trust this path provides.

Disjointness: How disjoint a path is from another path.

Path MTU: The end-to-end maximum transmission unit in one packet.

Transport Protocols available: Whether a specific transport protocol can be used to establish a connection over this path. An endpoint may know this because it has cached whether it could successfully establish, e.g., a QUIC connection, or an MPTCP subflow.

Protocol Features available: Whether a specific feature within a protocol is known to work over this path, e.g., ECN, or TCP Fast Open.

4. Dynamic Properties

Dynamic Path Properties are expected to change on the timescale of milliseconds. They usually relate to the state of the path, such as the currently available end-to-end bandwidth. Some of these properties may depend only on the first hop or on the access network, some may depend on the entire path.

Typically, Dynamic Properties can only be approximated and sampled, and might be made available in an aggregated form, such as averages or minimums. Dynamic Path Properties can be measured by the endpoint itself or somewhere in the network. See [ANRW18-Metrics] for a discussion of how to measure some dynamic path properties at the endpoint.

These properties may be symmetric or asymmetric. For example, an asymmetric property may be different in the upstream direction and in the downstream direction from the point of view of a particular host.

Available bandwidth: Maximum number of bytes per second that can be sent or received over this path. This depends on the available bandwidth at the bottleneck, and on crosstraffic.

Round Trip Time: Time from sending a packet to receiving a response from the remote endpoint.

Round Trip Time variation: Disparity of Round Trip Time values either over time or among multiple concurrent connections. A high RTT variation often indicates congestion.

Packet Loss: Percentage of sent packets that are not received on the other end.

Congestion: Whether there is any indication of congestion on the path.

Wireless Signal strength: Power level of the wireless signal being received. Lower signal strength, relative to the same noise floor, is correlated with higher bit error rates and lower modulation rates.

Wireless Modulation Rate: Modulation bitrate of the wireless signal. The modulation rate determines how many bytes are transmitted within a symbol on the wireless channel. A high modulation rate leads to a higher possible bitrate, given sufficient signal strength.

Wireless Channel utilization: Percentage of time during which there is a transmission on the wireless medium. A high channel utilization indicates a congested wireless network.

5. Security Considerations

Some of these properties may have security implications for endpoints. For example, a corporate policy might require to have a firewall on the path.

For properties provided by the network, their authenticity and correctness may need to be verified by an endpoint.

6. IANA Considerations

This document has no IANA actions.

7. Informative References

[ANRW18-Metrics]

Enghardt, T., Tiesel, P., and A. Feldmann, "Metrics for access network selection", Proceedings of the Applied Networking Research Workshop on - ANRW '18, DOI 10.1145/3232755.3232764, 2018.

[I-D.irtf-panrg-questions]

Trammell, B., "Open Questions in Path Aware Networking", draft-irtf-panrg-questions-00 (work in progress), April 2018.

[RFC7556]

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

Acknowledgments

Thanks to Paul Hoffman for feedback and editorial changes.

Authors' Addresses

Theresa Enghardt
TU Berlin

Email: theresa@inet.tu-berlin.de

Cyrill Kraehenbuehl
ETH Zuerich

Email: cyrill.kraehenbuehl@inf.ethz.ch

PANRG
Internet-Draft
Intended status: Informational
Expires: May 7, 2020

T. Enhardt
TU Berlin
C. Kraehenbuehl
ETH Zuerich
November 04, 2019

A Vocabulary of Path Properties
draft-enghardt-panrg-path-properties-03

Abstract

Path properties express information about paths across a network and the services provided via such paths. In a path-aware network, path properties may be fully or partially available to entities such as hosts. This document defines and categorizes path properties. Furthermore, the document specifies several path properties which might be useful to hosts or other entities.

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 7, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	2
3. Use Cases for Path Properties	4
3.1. Performance Monitoring and Enhancement	4
3.2. Path Selection	4
3.3. Traffic Configuration	5
4. Examples of Path Properties	6
5. Security Considerations	8
6. IANA Considerations	8
7. Informative References	8
Acknowledgments	10
Authors' Addresses	10

1. Introduction

In the current Internet architecture, hosts generally do not have information about forwarding paths through the network and about services associated with these paths. A path-aware network, as introduced in [I-D.irtf-panrg-questions], exposes information about paths to hosts or to other entities. This document defines such information as path properties, addressing the first of the questions in path-aware networking [I-D.irtf-panrg-questions].

As terms related to paths have different meanings in different areas of networking, first, this document provides a common terminology to define paths, path elements, and path properties. Then, this document provides some examples for use cases for path properties. Finally, the document lists several path properties that may be useful for the mentioned use cases.

2. Terminology

Node: An entity which processes packets, e.g., sends, receives, forwards, or modifies them. A node may be physical or virtual, e.g., a machine or a service function. A node may also be the collection of multiple entities which, as a collection, processes packets, e.g., an entire Autonomous System (AS).

Host: A node that generally executes application programs on behalf of user(s), employing network and/or Internet communication services in support of this function, as defined in [RFC1122].

Link: A medium or communication facility that connects two or more nodes with each other. A link enables a node to send packets to other nodes. Links can be physical, e.g., a WiFi network which connects an Access Point to stations, or virtual, e.g., a virtual switch which connects two virtual machines hosted on the same physical machine. A link is unidirectional and bidirectional communication can be modeled as two links between the same nodes in opposite directions.

Path element: Either a node or a link.

Path: A sequence of adjacent path elements over which a packet can be transmitted, starting and ending with a node. Paths are time-dependent, i.e., the sequence of path elements over which packets are sent from one node to another may change frequently. A path is defined between two nodes. For multicast or broadcast, a packet may be sent by one node and received by multiple nodes. In this case, the packet is sent over multiple paths at once, one path for each combination of sending and receiving node. Note that an entity may have only partial visibility of the path elements that comprise a path, and entities may treat path elements at different levels of abstraction.

Subpath: Given a path, a subpath is a sequence of adjacent path elements of this path.

Flow: An entity made of packets to which the traits of a path or set of subpaths may be applied in a functional sense. For example, a flow can consist of all packets sent within a TCP session with the same five-tuple between two hosts, or it can consist of all packets sent on the same physical link.

Property: A trait of one or a sequence of path elements, or a trait of a flow with respect to one or a sequence of path elements. An example of a link property is the maximum data rate that can be sent over the link. An example of a node property is the administrative domain that the node belongs to. An example of a property of a flow with respect to a subpath is the aggregated one-way delay of the flow being sent from one node to another node over this subpath. A property is thus described by a tuple containing the path element(s), the flow or an empty set if no packets are relevant for the property, the name of the property (e.g., maximum data rate), and the value of the property (e.g., 1Gbps).

Aggregated property: A collection of multiple values of a property into a single value, according to a function. A property can be aggregated over multiple path elements (i.e., a path), e.g., the

MTU of a path as the minimum MTU of all links on the path, over multiple packets (i.e., a flow), e.g., the median one-way latency of all packets between two nodes, or over both, e.g., the mean of the queueing delays of a flow on all nodes along a path. The aggregation function can be numerical, e.g., median, sum, minimum, it can be logical, e.g., "true if all are true", "true if at least 50\% of values are true", or an arbitrary function which maps multiple input values to an output value.

Observed property: A property that is observed for a specific path element or path, e.g., using measurements. For example, the one-way delay of a specific packet transmitted from one node to another node can be measured.

Assessed property: An approximate calculation or assessment of the value of a property. An assessed property includes the reliability of the calculation or assessment. The notion of reliability depends on the property. For example, a path property based on an approximate calculation may describe the expected median one-way latency of packets sent on a path within the next second, including the confidence level and interval. A non-numerical assessment may instead include the likelihood that the property holds.

3. Use Cases for Path Properties

When a path-aware network exposes path properties to hosts or other entities, these entities may use this information to achieve different goals. This section lists several use cases for path properties. Note that this is not an exhaustive list, as with every new technology and protocol, novel use cases may emerge, and new path properties may become relevant.

3.1. Performance Monitoring and Enhancement

Network operators can observe path properties (e.g., measured by on-path devices), to monitor Quality of Service (QoS) characteristics of recent end-user traffic on a path or subpath through their network. Such properties may help identify potential performance problems or trigger countermeasures to enhance performance.

3.2. Path Selection

Entities can choose what traffic to send over which path or subset of paths. Entities may select their paths to fulfill a specific goal, e.g., related to security or performance. As an example of security-related path selection, an entity may allow or disallow sending traffic over paths involving specific networks or nodes to enforce

traffic policies. In an enterprise network where all traffic has to go through a specific firewall, a path-aware host can implement this policy using path selection, in which case the host needs to be aware of paths involving that firewall. As an example of performance-related path selection, an entity may prefer paths with performance properties that best match its traffic, e.g., retrieving a small webpage as quickly as possible over a path with short One-Way Delays in both directions, or retrieving a large file over a path with high Link Capacities on all links. Note, there may be trade-offs between path properties (e.g., One-Way Delay and Link Capacity), and entities may influence these trade-offs with their choices. As a baseline, a path selection algorithm should aim to not perform worse than the default case most of the time.

Path selection can be done both by hosts and by entities within the network: A network (e.g., an AS) can adjust its path selection for internal or external routing based on the path properties. In BGP, the Multi Exit Discriminator (MED) attribute decides which path to choose if other attributes are equal; in a path aware network, instead of using this single MED value, other properties such as maximum or available/expected data rate could additionally be used to improve load balancing. A host might be able to select between a set of paths, either if there are several paths to the same destination (e.g., if the host is a mobile device with two wireless interfaces, both providing a path), or if there are several destinations, and thus several paths, providing the same service (e.g., Application-Layer Traffic Optimization (ALTO) [RFC5693], an application layer peer-to-peer protocol allowing hosts a better-than-random peer selection). Care needs to be taken when selecting paths based on path properties, as path properties that were previously measured may have become outdated and, thus, useless to predict the path properties of packets sent now.

3.3. Traffic Configuration

When sending traffic over a specific path, entities can adjust this traffic based on the properties of the path. For example, an entity may select an appropriate protocol depending on the capabilities of the on-path devices, or adjust protocol parameters to an existing path. An example of traffic configuration is a video streaming application choosing an (initial) video quality based on the achievable data rate, or the monetary cost to send data across a network, eventually on a given path, using a volume-based or flat-rate cost model.

Conversely, the selection of a protocol may influence the devices that will be involved in a path. For example, a 0-RTT Transport Converter [I-D.ietf-tcpm-converters] will be involved in a path only

when invoked by a host; such invocation will lead to the use of MPTCP or TCPinc capabilities while such use is not supported via the default forwarding path. Another example of traffic policies is a connection which may be composed of multiple streams; each stream with specific service requirements. A host may decide to invoke a given service function (e.g., transcoding) only for some streams while others are not processed by that service function.

4. Examples of Path Properties

This Section gives some examples of Path Properties which may be useful, e.g., for the use cases described in Section 3.

Path properties may be relatively dynamic, e.g., the one-way delay of a packet sent over a specific path, or non-dynamic, e.g., the MTU of an ethernet link which only changes infrequently. Usefulness over time differs depending on how dynamic a property is: The merit of a momentary measurement of a dynamic path property diminishes greatly as time goes on, e.g. the merit of an RTT measurement from a few seconds ago is quite small, while a non-dynamic path property might stay relevant for a longer period of time, e.g. a NAT typically stays on a specific path during the lifetime of a connection involving packets sent over this path.

From the point of view of a host, path properties may relate to path elements close to the host, i.e., within the first few hops, or they may include path elements far from the host, e.g. list of ASes traversed. The visibility of path properties to a specific entity may depend on factors such as the physical or network distance or the existence of trust or contractual relationships between the entity and the path element(s).

Furthermore, entities may or may not be able to influence the path elements on their path and their path properties. For example, a user might select between multiple potential adjacent links by selecting between multiple available WiFi Access Points. Or when connected to an Access Point, the user may move closer to enable their device to use a different access technology, potentially increasing the data rate available to the device. Another example is a user changing their data plan to reduce the Monetary Cost to transmit a given amount of data across a network.

Access Technology: The physical or link layer technology used for transmitting or receiving a flow on one or multiple path elements. The Access Technology may be given in an abstract way, e.g., as a WiFi, Wired Ethernet, or Cellular link. It may also be given as a specific technology, e.g., as a 2G, 3G, 4G, or 5G cellular link, or an 802.11a, b, g, n, or ac WiFi link. Other path elements

relevant to the access technology may include on-path devices, such as elements of a cellular backbone network. Note that there is no common registry of possible values for this property.

Monetary Cost: The price to be paid to transmit a specific flow across a network to which one or multiple path elements belong.

Service function: A service function that a path element applies to a flow, see [RFC7665]. Examples of abstract service functions include firewalls, Network Address Translation (NAT), and TCP optimizers.

Administrative Domain: The administrative domain, e.g., the ICP area, AS, or Service provider network to which a path element or subpath belongs.

Disjointness: For a set of two paths, the number of shared path elements can be a measure of intersection (e.g., Jaccard coefficient, which is the number of shared elements divided by the total number of elements). Conversely, the number of non-shared path elements can be a measure of disjointness (e.g., $1 - \text{Jaccard coefficient}$). A multipath protocol might use disjointness of paths as a metric to reduce the number of single points of failure.

Path MTU: The maximum size, in octets, of an IP packet that can be transmitted without fragmentation on a subpath.

Transport Protocols available: Whether a specific transport protocol can be used to establish a connection over a path or subpath. A host may cache its knowledge about recent successfully established connections using specific protocols, e.g., a QUIC connection, or an MPTCP subflow.

Protocol Features available: Whether a specific protocol feature is available over a path or subpath, e.g., Explicit Congestion Notification (ECN), or TCP Fast Open.

Some path properties express the performance of the transmission of a packet or flow over a link or subpath. Such transmission performance properties can be measured or approximated, e.g., by hosts or by path elements on the path. They might be made available in an aggregated form, such as averages or minimums. See [ANRW18-Metrics] for a discussion of how to measure some transmission performance properties at the host. Properties related to a path element which constitutes a single layer 2 domain are abstracted from the used physical and link layer technology, similar to [RFC8175].

Link Capacity: The link capacity is the maximum data rate at which data that was sent over a link can correctly be received at the node adjacent to the link. This property is analogous to the link capacity defined in [RFC5136] but not restricted to IP-layer traffic.

Link Usage: The link usage is the actual data rate at which data that was sent over a link is correctly received at the node adjacent to the link. This property is analogous to the link usage defined in [RFC5136] but not restricted to IP-layer traffic.

One-Way Delay: The one-way delay is the delay between a node sending a packet and another node on the same path receiving the packet. This property is analogous to the one-way delay defined in [RFC7679] but not restricted to IP-layer traffic.

One-Way Delay Variation: The variation of the one-way delays within a flow. This property is similar to the one-way delay variation defined in [RFC3393] but not restricted to IP-layer traffic and defined for packets on the same flow instead of packets sent between a source and destination IP address.

One-Way Packet Loss: Packets sent by a node but not received by another node on the same path after a certain time interval are considered lost. This property is analogous to the one-way loss defined in [RFC7680] but not restricted to IP-layer traffic. Metrics such as loss patterns [RFC3357] and loss episodes [RFC6534] can be expressed as aggregated properties.

5. Security Considerations

If nodes are basing policy or path selection decisions on path properties, they need to rely on the accuracy of path properties that other devices communicate to them. In order to be able to trust such path properties, nodes may need to establish a trust relationship or be able to verify the authenticity, integrity, and correctness of path properties received from another node.

6. IANA Considerations

This document has no IANA actions.

7. Informative References

[ANRW18-Metrics]

Enghardt, T., Tiesel, P., and A. Feldmann, "Metrics for access network selection", Proceedings of the Applied Networking Research Workshop on - ANRW '18, DOI 10.1145/3232755.3232764, 2018.

[I-D.ietf-tcpm-converters]

Bonaventure, O., Boucadair, M., Gundavelli, S., Seo, S., and B. Hesmans, "0-RTT TCP Convert Protocol", draft-ietf-tcpm-converters-13 (work in progress), October 2019.

[I-D.irtf-panrg-questions]

Trammell, B., "Current Open Questions in Path Aware Networking", draft-irtf-panrg-questions-03 (work in progress), October 2019.

[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>>.

[RFC3357] Koodli, R. and R. Ravikanth, "One-way Loss Pattern Sample Metrics", RFC 3357, DOI 10.17487/RFC3357, August 2002, <<https://www.rfc-editor.org/info/rfc3357>>.

[RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, DOI 10.17487/RFC3393, November 2002, <<https://www.rfc-editor.org/info/rfc3393>>.

[RFC5136] Chimento, P. and J. Ishac, "Defining Network Capacity", RFC 5136, DOI 10.17487/RFC5136, February 2008, <<https://www.rfc-editor.org/info/rfc5136>>.

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, DOI 10.17487/RFC5693, October 2009, <<https://www.rfc-editor.org/info/rfc5693>>.

[RFC6534] Duffield, N., Morton, A., and J. Sommers, "Loss Episode Metrics for IP Performance Metrics (IPPM)", RFC 6534, DOI 10.17487/RFC6534, May 2012, <<https://www.rfc-editor.org/info/rfc6534>>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.

- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/info/rfc7679>>.
- [RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <<https://www.rfc-editor.org/info/rfc7680>>.
- [RFC8175] Ratliff, S., Jury, S., Satterwhite, D., Taylor, R., and B. Berry, "Dynamic Link Exchange Protocol (DLEP)", RFC 8175, DOI 10.17487/RFC8175, June 2017, <<https://www.rfc-editor.org/info/rfc8175>>.

Acknowledgments

Thanks to the Path-Aware Networking Research Group for the discussion and feedback. Specifically, thanks to Mohamed Boudacair for the detailed review and various text suggestions, thanks to Brian Trammell for suggesting the flow definition, and thanks to Adrian Perrig and Matthias Rost for the detailed feedback. Thanks to Paul Hoffman for the editorial changes.

Authors' Addresses

Theresa Enghardt
TU Berlin

Email: theresa@inet.tu-berlin.de

Cyrill Kraehenbuehl
ETH Zuerich

Email: cyrill.kraehenbuehl@inf.ethz.ch

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: 11 November 2022

R. Hinden
Check Point Software
G. Fairhurst
University of Aberdeen
10 May 2022

IPv6 Minimum Path MTU Hop-by-Hop Option
draft-ietf-6man-mtu-option-15

Abstract

This document specifies a new IPv6 Hop-by-Hop option that is used to record the minimum Path MTU along the forward path between a source host to a destination host. The recorded value can then be communicated back to the source using the return Path MTU field in the option.

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 11 November 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Example Operation	3
1.2. Use of the IPv6 Hop-by-Hop Options Header	4
2. Motivation and Problem Solved	5
3. Requirements Language	6
4. Applicability Statements	6
5. IPv6 Minimum Path MTU Hop-by-Hop Option	6
6. Router, Host, and Transport Layer Behaviors	8
6.1. Router Behavior	8
6.2. Host Operating System Behavior	8
6.3. Transport Layer Behavior	9
6.3.1. Including the Option in an Outgoing Packet	10
6.3.2. Validation of the Packet that includes the Option	12
6.3.3. Receiving the Option	12
6.3.4. Using the Rtn-PMTU Field	13
6.3.5. Detecting Path Changes	14
6.3.6. Detection of Dropping Packets that include the Option	14
7. IANA Considerations	14
8. Security Considerations	14
8.1. Router Option Processing	15
8.2. Network Layer Host Processing	15
8.3. Validating use of the Option Data	16
8.4. Direct use of the Rtn-PMTU Value	16
8.5. Using the Rtn-PMTU Value as a Hint for Probing	17
8.6. Impact of Middleboxes	17
9. Experiment Goals	17
10. Implementation Status	18
11. Acknowledgments	18
12. Change log [RFC Editor: Please remove]	18
13. References	21
13.1. Normative References	21
13.2. Informative References	22
Appendix A. Examples of Usage	24
Authors' Addresses	26

1. Introduction

This document specifies a new IPv6 Hop-by-Hop (HBH) Option to record the minimum Maximum Transmission Unit (MTU) along the forward path between a source and a destination host. The source host creates a packet with this option and initializes the Min-PMTU field with the value of the MTU for the outbound link that will be used to forward the packet towards the destination host.

At each subsequent hop where the option is processed, the router compares the value of the Min-PMTU Field in the option and the MTU of its outgoing link. If the MTU of the link is less than the Min-PMTU, it rewrites the value in the option data with the smaller value. When the packet arrives at the destination host, the host can send the value of the minimum reported MTU for the path back to the source host using the Rtn-PMTU field in the option. The source host can then use this value as input to the method that sets the Path MTU (PMTU) used by upper layer protocols.

The IPv6 Minimum Path MTU Hop-by-Hop (MinPMTU HBH) Option is designed to work with packet sizes that can be specified in the IPv6 header. The maximum packet size that can be specified in an IPv6 header is 65,535 octets (2^{16}).

This method has the potential to complete Path MTU discovery in a single round trip time, even over paths that have successive links each with a lower MTU.

The mechanism defined in this document is focused on Unicast, it does not describe Multicast. That is left for future work.

1.1. Example Operation

The figure below illustrates the operation of the method. In this case, the path between the source host and the destination host comprises three links, the source has a link MTU of size MTU-S, the link between routers R1 and R2 has an MTU of size 9000 bytes, and the final link to the destination has an MTU of size MTU-D.

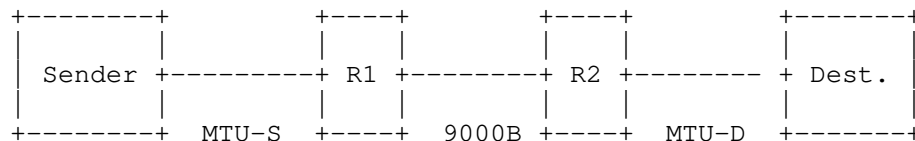


Figure 1

Three scenarios are described:

- * Scenario 1, considers all links to have an 9000 byte MTU and the method is supported by both routers. The initial Min-PMTU is not modified along the path, and therefore the PMTU is 9000 bytes.
- * Scenario 2, considers the link between R2 and destination host (MTU-D) to have an MTU of 1500 bytes. This is the smallest MTU, router R2 updates the Min-PMTU to 1500 bytes and the method

correctly updates the PMTU to 1500 bytes. Had there been another smaller MTU at a link further along the path that also supports the method, the lower MTU would also have been detected.

- * Scenario 3, considers the case where the router preceding the smallest link (R2) does not support the method, and the link to the destination host (MTU-D) has an MTU of 1500 bytes. Therefore, router R2 does not update the Min-PMTU to 1500 bytes. The method then fails to detect the actual PMTU.

In Scenarios 2 and 3, a lower PMTU would also fail to be detected in the case where PMTUD had been used and an ICMPv6 Packet Too Big (PTB) message had not been delivered to the sender [RFC8201].

These scenarios are summarized in the table below. "H" in R1 and/or R2 columns means the router understands the MinPMTU HBH option.

	MTU-S	MTU-D	R1	R2	Rec PMTU	Note
1	9000B	9000B	H	H	9000 B	Endpoints attempt to use a 9000 B PMTU.
2	9000B	1500B	H	H	1500 B	Endpoints attempt to use a 1500 B PMTU.
3	9000B	1500B	H	-	9000 B	Endpoints attempt to use a 9000 B PMTU, but need to implement a method to fall back to discover and use a 1500 B PMTU.

Figure 2

1.2. Use of the IPv6 Hop-by-Hop Options Header

IPv6 as specified in [RFC8200] allows nodes to optionally process the Hop-by-Hop header. Specifically, from Section 4:

- * The Hop-by-Hop Options header is not inserted or deleted, but may be examined or processed by any node along a packet's delivery path, until the packet reaches the node (or each of the set of nodes, in the case of multicast) identified in the Destination Address field of the IPv6 header. The Hop-by-Hop Options header, when present, must immediately follow the IPv6 header. Its presence is indicated by the value zero in the Next Header field of the IPv6 header.
- * NOTE: While [RFC2460] required that all nodes must examine and process the Hop-by-Hop Options header, it is now expected that nodes along a packet's delivery path only examine and process the Hop-by-Hop Options header if explicitly configured to do so.

The Hop-by-Hop Option defined in this document is designed to take advantage of this property of how Hop-by-Hop options are processed. Nodes that do not support this Option SHOULD ignore them. This can mean that the Min-PMTU value does not account for all links along a path.

2. Motivation and Problem Solved

The current state of Path MTU Discovery on the Internet is problematic. The mechanisms defined in [RFC8201] are known to not work well in all environments. It fails to work in various cases, including when nodes in the middle of the network do not send ICMPv6 PTB messages, or rate-limited ICMPv6 messages, or do not have a return path to the source host.

This results in many transport layer connections being configured to use smaller packets (e.g., 1280 bytes) by default and makes it difficult to take advantage of paths with a larger PMTU where they do exist. Applications that send large packets are forced to use IPv6 Fragmentation [RFC8200], which can reduce the reliability of Internet communication [RFC8900].

Encapsulations and network-layer tunnels further reduce the payload size available for a transport protocol to use. Also, some use-cases increase packet overhead, for example, Network Virtualization Using Generic Routing Encapsulation (NVGRE) [RFC7637] encapsulates L2 packets in an outer IP header and does not allow IP Fragmentation.

Sending larger packets can improve host performance, e.g., avoiding limits to packet processing by the packet rate. For example, the packet per second rate required to reach wire speed on a 10G link with 1280 byte packets is about 977K packets per second (pps), vs. 139K pps for 9000 byte packets.

The purpose of this document is to improve the situation by defining a mechanism that does not rely on reception of ICMPv6 Packet Too Big messages from nodes in the middle of the network. Instead, this provides information to the destination host about the minimum Path MTU, and sends this information back to the source host. This is expected to work better than the current RFC8201-based mechanisms.

A similar mechanism was proposed in 1988 for IPv4 in [RFC1063] by Jeff Mogul, C. Kent, Craig Partridge, and Keith McCloghrie. It was later obsoleted in 1990 by [RFC1191], the current deployed approach to Path MTU Discovery. In contrast, the method described in this document uses the Hop-by-Hop option of IPv6. It does not replace PMTUD [RFC8201], PLPPMTUD [RFC4821] or Datagram PLPMTUD [RFC8899], but rather is designed to compliment these methods.

3. 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.

4. Applicability Statements

The Path MTU option is designed for environments where there is control over the hosts and nodes that connect them, and where there is more than one MTU size in use. For example, in Data Centers and on paths between Data Centers, to allow hosts to better take advantage of a path that is able to support a large PMTU.

The design of the option is sufficiently simple that it can be executed on a router's fast path. A successful experiment depends on both implementation by host and router vendors and deployment by operators. The contained use-case of connections within and between Data Centers could be a driver for deployment.

The method could also be useful in other environments, including the general Internet, and offers advantage when this Hop-by-Hop Option is supported on all paths. The method is more robust when used to probe the path using packets that do not carry application data and when also paired with a method such as Packetization Layer PMTUD [RFC4821] or Datagram PLPMTUD [RFC8899].

5. IPv6 Minimum Path MTU Hop-by-Hop Option

The Minimum Path MTU Hop-by-Hop Option has the following format:

Option Type	Option Data Len	Option Data		
BBCTTTTT	00000100	Min-PMTU	Rtn-PMTU	R

Option Type (see Section 4.2 of [RFC8200]):

BB 00 Skip over this option and continue processing.

C 1 Option data can change en route to the packet's final destination.

TTTTT 10000 Option Type assigned from IANA [IANA-HBH].

Length: 4 The size of the value field in Option Data field supports PMTU values from 0 to 65,534 octets, the maximum size represented by the Path MTU option.

Min-PMTU: n 16-bits. The minimum MTU recorded along the path in octets, reflecting the smallest link MTU that the packet experienced along the path. A value less than the IPv6 minimum link MTU [RFC8200] MUST be ignored.

Rtn-PMTU: n 15-bits. The returned Path MTU field, carrying the 15 most significant bits of the latest received Min-PMTU field for the forward path. The value zero means that no Reported MTU is being returned.

R n 1-bit. R-Flag. Set by the source to signal that the destination host should include the received Rtn-PMTU field updated by the reported Min-PMTU value when the destination host is to send a PMTU Option back to the source host.

Figure 3

NOTE: The encoding of the final two octets (Rtn-PMTU and R-Flag) could be implemented by a mask of the latest received Min-PMTU value with 0xFFFE, discarding the right-most bit and then performing a logical 'OR' with the R-Flag value of the sender. This encoding fits in the minimum-sized Hop-by-Hop Option header.

6. Router, Host, and Transport Layer Behaviors

6.1. Router Behavior

Routers that are not configured to support Hop-by-Hop Options are not expected to examine or process the contents of this option [RFC8200].

Routers that support Hop-by-Hop Options, but are not configured to support this option SHOULD skip over this option and continue to processing the header [RFC8200].

Routers that support this option MUST compare the value of the Min-PMTU field with the MTU configured for the outgoing link. If the MTU of the outgoing link is less than the Min-PMTU, the router rewrites the Min-PMTU in the Option to use the smaller value. (The router processing is performed without checking the valid range of the Min-PMTU or the Rtn-PMTU fields.)

A router MUST ignore and MUST NOT change the Rtn-PMTU field or the R-Flag in the option.

6.2. Host Operating System Behavior

The PMTU entry associated with the destination in the host's destination cache [RFC4861] SHOULD be updated after detecting a change using the IPv6 Minimum Path MTU Hop-by-Hop Option. This cached value can be used by other flows that share the host's destination cache.

The value in the host destination cache SHOULD be used by PLPMTUD to select an initial PMTU for a flow. The cached PMTU is only increased by PLPMTUD when the Packetization Layer determines the path actually supports a larger PMTU [RFC4821] [RFC8899].

When requested to send an IPv6 packet with the MinPMTU HBH option, the source host includes the option in an outgoing packet. The source host MUST fill the Min-PMTU field with the MTU configured for the link over which it will send the packet on the next hop towards the destination host.

When a host includes the option in a packet it sends, the host SHOULD set the Rtn-PMTU field to the previously cached value of the received Minimum Path MTU for the flow in the Rtn-PMTU field (see Section 6.3.3). If this value is not set (for example, because there is no cached reported Min-PMTU value), the Rtn-PMTU field value MUST be set to zero.

The source host MAY request the destination host to return the reported Min-PMTU value by setting the R-Flag in the option of an outgoing packet. The R-Flag SHOULD NOT be set when the MinPMTU HBH Option was sent solely to provide requested feedback on the return Path MTU to avoid each response generating another response.

The destination host controls when to send a packet with this option in response to an R-flag, as well as which packets to include it in. The destination host MAY limit the rate at which it sends these packets.

A destination host only sets the R Flag if it wishes the source host to also return the discovered PMTU value for the path from the destination to the source.

The normal sequence of operation of the R-Flag using the terminology from the diagram in Figure 1 is:

1. The source sends a probe to the destination. The sender sets the R-Flag.
2. The destination responds by sending a probe including the received Min-PMTU as the Rtn-PMTU. A destination that does not wish to probe the return path sets the R-Flag to 0.

6.3. Transport Layer Behavior

This Hop-by-Hop option is intended to be used with a path MTU discovery method.

PLPMTUD [RFC9000] uses probe packets for two distinct functions:

- * Probe packets are used to confirm connectivity. Such probes can be of any size up to the PLPMTU. These probe packets are sent to solicit a response use the path to the remote node. These probe packets can carry the Hop-by-Hop PMTU option, providing the final size of the packet does not exceed the current PLPMTU. After validating that the packet originates from the path (section 4.6.1), the PLPMTUD method can use the reported size from the Hop-by-Hop option as the next search point when it resumes the search algorithm. (This use resembles the use of the PTB_SIZE information in section 4.6.2 of [RFC8899])
- * A second use of probe packets is to explore if a path supports a packet size greater than the current PLPMTU. If this probe packet is successfully delivered (as determined by the source host), then the PLPMTU is raised to the size of the successful probe. These probe packets do not usually set the Path MTU Hop-by-Hop option.

See section 1.2 of [RFC8899]. Section 4.1 of [RFC8899] also describes ways that a Probe Packet can be constructed, depending on whether the probe packets carry application data.

- * The PMTU Hop-by-Hop Option Probe can be sent on packets that include application data, but needs to be robust to potential loss of the packet (i.e., with the possibility that retransmission might be needed if the packet is lost).
- * Using a PMTU Probe on packets that do not carry application data will avoid the need for loss recovery if a router on the path drops packets that set this option. (This avoids the transport needing to retransmit a lost packet that includes this option.) This is the normal default format for both uses of probes.

6.3.1. Including the Option in an Outgoing Packet

The upper layer protocol can request the MinPMTU HBH option to be included in an outgoing IPv6 packet. A transport protocol (or upper layer protocol) can include this option only on specific packets used to test the path. This option does not need to be included in all packets belonging to a flow.

NOTE: Including this option in a large packet (e.g., one larger than the present PMTU) is not likely to be useful, since the large packet would itself be dropped by any link along the path with a smaller MTU, preventing the Min-PMTU information from reaching the destination host.

Discussion:

- * In the case of TCP, the option could be included in a packet that carries a TCP segment sent after the connection is established. A segment without data could be used, to avoid the need to retransmit this data if the probe packet is lost. The discovered value can be used to inform PLPMTUD [RFC4821].

NOTE: A TCP SYN can also negotiate the Maximum Segment Size (MSS), which acts as an upper limit to the packet size that can be sent by a TCP sender. If this option were to be included in a TCP SYN, it could increase the probability that the SYN segment is lost when routers on the path drop packets with this option (see Section 6.3.6), which could have an unwanted impact on the result of racing options [I-D.ietf-taps-arch] or feature negotiation.

- * The use with datagram transport protocols (e.g., UDP) is harder to characterize because applications using datagram transports range from very short-lived (low data-volume applications) exchanges, to longer (bulk) exchanges of packets between the source and destination hosts [RFC8085].
- * Simple-exchange protocols (i.e., low data-volume applications [RFC8085] that only send one or a few packets per transaction), might assume that the PMTU is symmetrical. That is, the PMTU is the same in both directions, or at least not smaller for the return path. This optimization does not hold when the paths are not symmetric.
- * The MinPMTU HBH option can be used with ICMPv6 [RFC4443]. This requires a response from the remote node and therefore is restricted to use with ICMPv6 echo messages. The MinPMTU HBH option could provide additional information about the PMTU that might be supported by a path. This could be use as a diagnostic tool to measure the PMTU of a path. As with other uses, the actual supported PMTU is only confirmed after receiving a response to a subsequent probe of the PMTU size.
- * A datagram transport can utilise DPLPMTUD [RFC8899]. For example, QUIC (see section 14.3 of [RFC9000]), can use DPLPMTUD to determine whether the path to a destination will support a desired maximum datagram size. When using the IPv6 MinPMTU HBH option, the option could be added to an additional QUIC PMTU Probe that is of minimal size (or one no larger than the currently supported PMTU size). Once the return Path MTU value in the MinPMTU HBH option has been learned, DPLPMTUD can be triggered to test for a larger PLPMTU using an appropriately sized PLPMTU Probe Packet (see section 5.3.1 of [RFC8899]).
- * The use of this option with DNS and DNSSEC over UDP is expected to work for paths where the PMTU is symmetric. The DNS server will learn the PMTU from the DNS query messages. If the Rtn-PMTU value is smaller, then a large DNSSEC response might be dropped and the known problems with PMTUD will then occur. DNS and DNSSEC over transport protocols that can carry the PMTU ought to work.
- * This method also can be used with Anycast to discover the PMTU of the path, but the use needs to be aware that the Anycast binding might change.

6.3.2. Validation of the Packet that includes the Option

An upper layer protocol (e.g., transport endpoint) using this option needs to provide protection from data injection attacks by off-path devices [RFC8085]. This requires a method to assure that the information in the Option Data is provided by a node on the path. This validates that the packet forms a part of an existing flow, using context available at the upper layer. For example, a TCP connection or UDP application that maintains the related state and uses a randomized ephemeral port would provide this basic validation to protect from off-path data injection, see Section 5.1 of [RFC8085]. IPsec [RFC4301] and TLS [RFC8446] provide greater assurance.

The upper layer discards any received packet when the packet validation fails. When packet validation fails, the upper layer **MUST** also discard the associated Option Data from the MinPMTU HBH option without further processing.

6.3.3. Receiving the Option

For a connection-oriented upper layer protocol, caching of the received Min-PMTU could be implemented by saving the value in the connection context at the transport layer. A connection-less upper layer (e.g., one using UDP), requires the upper layer protocol to cache the value for each flow it uses.

A destination host that receives a MinPMTU HBH Option with the R-Flag **SHOULD** include the MinPMTU HBH option in the next outgoing IPv6 packet for the corresponding flow.

A simple mechanism could only include this option (with the Rtn-PMTU field set) the first time this option is received or when it notifies a change in the Minimum Path MTU. This limits the number of packets including the option packets that are sent. However, this does not provide robustness to packet loss or recovery after a sender loses state.

Discussion:

- * Some upper layer protocols send packets less frequently than the rate at which the host receives packets. This provides less frequent feedback of the received Rtn-PMTU value. However, a host always sends the most recent Rtn-PMTU value.

6.3.4. Using the Rtn-PMTU Field

The Rtn-PMTU field provides an indication of the PMTU from on-path routers. It does not necessarily reflect the actual PMTU between the source and destination hosts. Care therefore needs to be exercised in using the Rtn-PMTU value. Specifically:

- * The actual PMTU can be lower than the Rtn-PMTU value because the Min-PMTU field was not updated by a router on the path that did not process the option.
- * The actual PMTU may be lower than the Rtn-PMTU value because there is a layer-2 device with a lower MTU.
- * The actual PMTU may be larger than the Rtn-PMTU value because of a corrupted, delayed or mis-ordered response. A source host **MUST** ignore a Rtn-PMTU value larger than the MTU configured for the outgoing link.
- * The path might have changed between the time when the probe was sent and when the Rtn-PMTU value received.

IPv6 requires that every link in the Internet have an MTU of 1280 octets or greater. A node **MUST** ignore a Rtn-PMTU value less than 1280 octets [RFC8200].

To avoid unintentional dropping of packets that exceed the actual PMTU (e.g., Scenario 3 in Section 1.1), the source host can delay increasing the PMTU until a probe packet with the size of the Rtn-PMTU value has been successfully acknowledged by the upper layer, confirming that the path supports the larger PMTU. This probing increases robustness, but adds one additional path round trip time before the PMTU is updated. This use resembles that of PTB messages in section 4.6 of DPLPMTUD [RFC8899] (with the important difference that a PTB message can only seek to lower the PMTU, whereas this option could trigger a probe packet to seek to increase the PMTU.)

Section 5.2 of [RFC8201] provides guidance on the caching of PMTU information and also the relation to IPv6 flow labels. Implementations should consider the impact of Equal Cost Multipath (ECMP) [RFC6438]. Specifically, whether a PMTU ought to be maintained for each transport endpoint, or for each network address.

6.3.5. Detecting Path Changes

Path characteristics can change and the actual PMTU could increase or decrease over time. For instance, following a path change when packets are forwarded over a link with a different MTU than that previously used. To bound the delay in discovering an increase in the actual PMTU, a host with a link MTU larger than the current PMTU SHOULD periodically send the MinPMTU HBH Option with the R-bit set. DPLPMTUD provides recommendations concerning how this could be implemented (see Section 5.3 of [RFC8899]). Since the option consumes less capacity than a full-sized probe packet, there can be advantage in using this to detect a change in the path characteristics.

6.3.6. Detection of Dropping Packets that include the Option

There is evidence that some middleboxes drop packets that include Hop-by-Hop options. For example, a firewall might drop a packet that carries an unknown extension header or option. This practice is expected to decrease as an option becomes more widely used. It could result in generation of an ICMPv6 message indicating the problem. This could be used to (temporarily) suspend use of this option.

A middlebox that silently discards a packet with this option results in dropping of any packet using the option. This dropping can be avoided by appropriate configuration in a controlled environment, such as within a data centre, but needs to be considered for Internet usage. Section 6.2 recommends that this option is not used on packets where loss might adversely impact performance.

7. IANA Considerations

IANA has assigned and registered an IPv6 Hop-by-Hop Option type with Temporary status from the "Destination Options and Hop-by-Hop Options" registry [IANA-HBH]. This assignment is shown in Section 5.

IANA is requested to update this registry to point to this document and remove the Temporary status.

8. Security Considerations

This section discusses the security considerations. It first reviews router option processing. It then reviews host processing when receiving this option at the network layer. It then considers two ways in which the Option Data can be processed, followed by two approaches for using the Option Data. Finally, it discusses middlebox implications related to use in the general Internet.

8.1. Router Option Processing

This option shares the characteristics of all other IPv6 Hop-by-Hop Options, in that if not supported at line rate it could be used to degrade the performance of a router. This option, while simple, is no different to other uses of IPv6 Hop-by-Hop options.

It is common for routers to ignore the Hop-by-Hop Option header or drop packets containing a Hop-by-Hop Option header. Routers implementing IPv6 according to [RFC8200] only examine and process the Hop-by-Hop Options header if explicitly configured to do so.

8.2. Network Layer Host Processing

A malicious attacker can forge a packet directed at a host that carries the MinPMTU HBH option. By design, the fields of this IP option can be modified by the network.

For comparison, the ICMPv6 Packet Too Big message used in [RFC8201] Path MTU Discovery, the source host has an inherent trust relationship with the destination host including this option. This trust relationship can be used to help verify the option. ICMPv6 Packet Too Big messages are sent from any router on the path to the destination host, the source host has no prior knowledge of these routers (except for the first hop router).

Reception of this packet will require processing as the network stack parses the packet before the packet is delivered to the upper layer protocol. This network layer option processing is normally completed before any upper layer protocol delivery checks are performed.

The network layer does not normally have sufficient information to validate that the packet carrying an option originated from the destination (or an on-path node). It also does not typically have sufficient context to demultiplex the packet to identify the related transport flow. This can mean that any changes resulting from reception of the option applies to all flows between a pair of endpoints.

These considerations are no different to other uses of Hop-by-Hop options, and this is the use case for PMTUD. The following section describes a mitigation for this attack.

8.3. Validating use of the Option Data

Transport protocols should be designed to provide protection from data injection attacks by off-path devices and mechanisms should be described in the Security Considerations for each transport specification (see Section 5.1 of the UDP Guidelines [RFC8085]). For example, a TCP or UDP application that maintains the related state and uses a randomized ephemeral port would provide basic protection. TLS [RFC8446] or IPsec [RFC4301] provide cryptographic authentication. An upper layer protocol that validates each received packet discards any packet when this validation fails. In this case, the host MUST also discard the associated Option Data from the MinPMTU HBH option without further processing (Section 6.3).

A network node on the path has visibility of all packets it forwards. By observing the network packet payload, the node might be able to construct a packet that might be validated by the destination host. Such a node would also be able to drop or limit the flow in other ways that could be potentially more disruptive. Authenticating the packet, for example, using IPsec [RFC4301] or TLS [RFC8446] mitigates this attack. Note that AH style authentication [RFC4302] while authenticating the payload and outer IPv6 header, does not check Hop-by-Hop options that change on route.

8.4. Direct use of the Rtn-PMTU Value

The simplest way to utilize the Rtn-PMTU value is to directly use this to update the PMTU. This approach results in a set of security issues when the option carries malicious data:

- * A direct update of the PMTU using the Rtn-PMTU value could result in an attacker inflating or reducing the size of the host PMTU for the destination. Forcing a reduction in the PMTU can decrease the efficiency of network use, might increase the number of packets/fragments required to send the same volume of payload data, and prevents sending an unfragmented datagram larger than the PMTU. Increasing the PMTU can result in black-holing (see Section 1.1 of [RFC8899]) when the source host sends packets larger than the actual PMTU. This persists until the PMTU is next updated.
- * The method can be used to solicit a response from the destination host. A malicious attacker could forge a packet that causes the destination to add the option to a packet sent to the source host. A forged value of Rtn-PMTU in the Option Data might also impact the remote endpoint, as described in the previous bullet. This persists until a valid MinPMTU HBH option is received. This attack could be mitigated by limiting the sending of the MinPMTU HBH option in reply to incoming packets that carry the option.

8.5. Using the Rtn-PMTU Value as a Hint for Probing

Another way to utilize the Rtn-PMTU value is to indirectly trigger a probe to determine if the path supports a PMTU of size Rtn-PMTU. This approach needs context for the flow, and hence assumes an upper layer protocol that validates the packet that carries the option (see Section 8.3). This is the case when used in combination with DPLPMTUD [RFC8899]. A set of security considerations result when an option carries malicious data:

- * If the forged packet carries a validated option with a non-zero Rtn-PMTU field, the upper layer protocol could utilize the information in the Rtn-PMTU field. A Rtn-PMTU larger than the current PMTU can trigger a probe for a new size.
- * If the forged packet carries a non-zero Min-PMTU field, the upper layer protocol would change the cached information about the path from the source. The cached information at the destination host will be overwritten when the host receives another packet that includes a MinPMTU HBH option corresponding to the flow.
- * Processing of the option could cause a destination host to add the MinPMTU HBH option to a packet sent to the source host. This option will carry a Rtn-PMTU value that could have been updated by the forged packet. The impact of the source host receiving this resembles that discussed previously.

8.6. Impact of Middleboxes

There is evidence that some middleboxes drop packets that include Hop-by-Hop options. For example, a firewall might drop a packet that carries an unknown extension header or option. This practice is expected to decrease as the option becomes more widely used. Methods to address this are discussed in Section 6.3.6.

When a forged packet causes a packet to be sent including the MinPMTU HBH option, and the return path does not forward packets with this option, the packet will be dropped Section 6.3.6. This attack is mitigated by validating the option data before use and by limiting the rate of responses generated. An upper layer could further mitigate the impact by responding to an R-Flag by including the option in a packet that does not carry application data.

9. Experiment Goals

This section describes the experimental goals of this specification.

A successful deployment of the method depends upon several components being implemented and deployed:

- * Support in the sending node (see Section 6.2). This also requires corresponding support in upper layer protocols (see Section 6.3).
- * Router support in nodes (see Section 6.1). The IETF continues to provide recommendations on the use of IPv6 Hop-by-Hop options, for example Section 2.2.2 of [RFC9099]. This document does not update the way router implementations configure support for Hop-by-Hop options.
- * Support in the receiving node (see Section 6.3.3).

Experience from deployment is an expected input to any decision to progress this specification from Experimental to IETF Standards Track. Appropriate inputs might include:

- * Reports of implementation experience;
- * Measurements of the number paths where the method can be used;
- * Measurements showing the benefit realized or the implications of using specific methods over specific paths.

10. Implementation Status

At the time this document was published there are two known implementations of the Path MTU Hop-by-Hop option. These are:

- * Wireshark dissector. This is shipping in production in Wireshark version 3.2 [WIRESHARK].
- * A prototype in the open source version of the FD.io Vector Packet Processing (VPP) technology [VPP]. At the time this document was published, the source code can be found [VPP_SRC].

11. Acknowledgments

Helpful comments were received from Tom Herbert, Tom Jones, Fred Templin, Ole Troan, Tianran Zhou, Jen Linkova, Brian Carpenter, Peng Shuping, Mark Smith, Fernando Gont, Michael Dougherty, Erik Kline, and other members of the 6MAN working group.

12. Change log [RFC Editor: Please remove]

draft-ietf-6man-mtu-option-15, 2022-May-10

- * Correcting an editing mistake in Appendix A.
- * Editorial Change.

draft-ietf-6man-mtu-option-14, 2022-April-15

- * Area Director Reviews:
 - Lars Eggert's Review: Fixed "nits".
 - Eric Vyncke's Review: Added that this work is focused on Unicast, removed Discussion from Section 6.1, revised text on PLPMTUD probing, changed SHOULD to MUST in Section 6.3.4, and fixed several NITs.
 - Alvaro Retana's Review: Changed SHOULD language to more general text in Section 6.1
 - ARTART Review: Added new Appendix "Examples of Usage" with diagrams showing examples of use.
 - Zaheduzzaman Sarker's Review: Fixed some editorial issues, and updated SHOULD language.
- * Editorial Changes.

draft-ietf-6man-mtu-option-13, 2022-February-28

- * Area Directorate Reviews:
 - SECDIR Review: Fixed "nit".
 - TSVART Review: Restructured Section 6 including making Transport Behavior more prominent, added text about ICMPv6 to Section 6.3.1, moved the text about prior work in RFC1063 to Section 2.
 - GENART Review: Added text to Section 1 that this option was designed to work with packet sizes that can be specified in the IPv6 Header.
- * Editorial Changes.

draft-ietf-6man-mtu-option-12, 2022-January-26

- * Clarified a few issues raised by AD review by Erik Kline AD review.

draft-ietf-6man-mtu-option-11, 2021-September-30

- * Clarifications and editorial changes to the Security Considerations section based on early AD review by Erik Kline.

draft-ietf-6man-mtu-option-10, 2021-September-27

- * Clarifications and editorial changes based on second chair review by Ole Troan.
- * Editorial changes.

draft-ietf-6man-mtu-option-09, 2021-September-23

- * Clarifications and editorial changes based on review by Michael Dougherty.

draft-ietf-6man-mtu-option-08, 2021-September-7

- * Clarifications and editorial changes based on chair review by Ole Troan.
- * Correction and clarifications based on review by Fernando Gont.

draft-ietf-6man-mtu-option-07, 2021-August-31

- * Added Experiment Goals section.
- * Added Implementation Status section.
- * Updated the IANA Considerations section to point to this document and remove Temporary status.
- * Clarifications and editorial changes based on review by Mark Smith.

draft-ietf-6man-mtu-option-06, 2021-August-7

- * Transport usage of the mechanism clarified in response to feedback and suggestions from Jen Linkova.
- * Restructured Section 6 to improve readability.
- * Editorial changes.

draft-ietf-6man-mtu-option-05, 2021-April-28

- * Editorial changes.

draft-ietf-6man-mtu-option-04, 2020-Oct-23

- * Fixes for typos.

draft-ietf-6man-mtu-option-03, 2020-Sept-14

- * Rewrite to make text and terminology more consistent.
- * Added the notion of validating the packet before use of the HBH option data.
- * Method aligned with the way common APIs send/receive HBH option data.
- * Added reference to DPLPMTUD and clarified upper layer usage.
- * Completed security considerations section.

draft-ietf-6man-mtu-option-02, 2020-March-9

- * Editorial changes to make text and terminology more consistent.

- * Added reference to DPLPMTUD.

draft-ietf-6man-mtu-option-01, 2019-September-13

- * Changes to show IANA assigned code point.
- * Editorial changes to make text and terminology more consistent.
- * Added a reference to RFC8200 in Section 2 and a reference to RFC6438 in Section 6.3.

draft-ietf-6man-mtu-option-00, 2019-August-9

- * First 6man w.g. draft version.
- * Changes to request IANA allocation of code point.
- * Editorial changes.

draft-hinden-6man-mtu-option-02, 2019-July-5

- * Changed option format to also include the Returned PMTU value and Return flag and made related text changes in Section 6.2 to describe this behavior.
- * ICMPv6 Packet Too Big messages are no longer used for feedback to the source host.
- * Added to Acknowledgements Section that a similar mechanism was proposed for IPv4 in 1988 in [RFC1063].
- * Editorial changes.

draft-hinden-6man-mtu-option-01, 2019-March-05

- * Changed requested status from Standards Track to Experimental to allow use of experimental option type (11110) to allow for experimentation. Removed request for IANA Option assignment.
- * Added Section 2 "Motivation and Problem Solved" section to better describe what the purpose of this document is.
- * Added appendix describing planned experiments and how the results will be measured.
- * Editorial changes.

draft-hinden-6man-mtu-option-00, 2018-Oct-16

- * Initial draft.

13. References

13.1. Normative References

[IANA-HBH] "Destination Options and Hop-by-Hop Options",
<<https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#ipv6-parameters-2>>.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [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>>.
- [RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <<https://www.rfc-editor.org/info/rfc8201>>.

13.2. Informative References

- [I-D.ietf-taps-arch] Pauly, T., Trammell, B., Brunstrom, A., Fairhurst, G., and C. Perkins, "An Architecture for Transport Services", Work in Progress, Internet-Draft, draft-ietf-taps-arch-12, 3 January 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-taps-arch-12>>.
- [RFC1063] Mogul, J., Kent, C., Partridge, C., and K. McCloghrie, "IP MTU discovery options", RFC 1063, DOI 10.17487/RFC1063, July 1988, <<https://www.rfc-editor.org/info/rfc1063>>.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <<https://www.rfc-editor.org/info/rfc1191>>.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<https://www.rfc-editor.org/info/rfc2460>>.
- [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>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.

- [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>>.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <<https://www.rfc-editor.org/info/rfc4821>>.
- [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>>.
- [RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <<https://www.rfc-editor.org/info/rfc6438>>.
- [RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", RFC 7637, DOI 10.17487/RFC7637, September 2015, <<https://www.rfc-editor.org/info/rfc7637>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8899] Fairhurst, G., Jones, T., Tüxen, M., Rüngeler, I., and T. Völker, "Packetization Layer Path MTU Discovery for Datagram Transports", RFC 8899, DOI 10.17487/RFC8899, September 2020, <<https://www.rfc-editor.org/info/rfc8899>>.
- [RFC8900] Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", BCP 230, RFC 8900, DOI 10.17487/RFC8900, September 2020, <<https://www.rfc-editor.org/info/rfc8900>>.
- [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <<https://www.rfc-editor.org/info/rfc9000>>.

- [RFC9099] Vyncke, É., Chittimaneni, K., Kaeo, M., and E. Rey,
"Operational Security Considerations for IPv6 Networks",
RFC 9099, DOI 10.17487/RFC9099, August 2021,
<<https://www.rfc-editor.org/info/rfc9099>>.
- [VPP] "VPP/What is VPP?",
<https://wiki.fd.io/view/VPP/What_is_VPP%3F>.
- [VPP_SRC] "VPP Source", <<https://gerriet.fd.io/r/c/vpp/+21948>>.
- [WIRESHARK] "Wireshark Network Protocol Analyzer",
<<https://www.wireshark.org>>.

Appendix A. Examples of Usage

This section provides examples that illustrate a use of the MinPMTU HBH option by a source using DPLPMTUD to discover the PLPMTU supported by a path. They consider a path where the on-path router has been configured with an outgoing MTU of d' . The source starts by transmission of packets of size a , and then uses DPLPMTUD to seek to increase the size in steps resulting in sizes of b, c, d, e , etc., (chosen by the search algorithm used by DPLPMTUD). The search algorithm terminates with a PLPMTU that is at least d and is less than or equal to d' .

The first example considers DPLPMTUD without using the MinPMTU HBH option. In this case, DPLPMTUD searches using an increasing size of probe packet. Probe packets of size (e) are sent, which are larger than the actual PMTU. In this example, PTB messages are not received from the routers and repeated unsuccessful probes result in the search phase completing. Packets of data are never sent with a size larger than the size of the last confirmed probe packet. ACKs of data packets are not shown.


```

----Packets of data size (a) ----->
----Probe size (b) ----->
<----- ACK of probe -----
----Packets of data size (b) ----->
----Probe size (c) ----->
<----- ACK of probe -----
----Packets of data size (c) ----->
----Probe size (d) ----->
<----- ACK of probe -----
----Packets of data size (d) ----->
<----- ACK of probe -----
...
----Probe size (e) -----X
      X----ICMPv6 PTB (d') --|
----Packets of data size (d) ----->
----Probe size (e) -----X (again)
      X----ICMPv6 PTB (d') --|
----Packets of data size (d) -----
...
etc, until MaxProbes are unsuccessful and search phase completes.
----Packets of data size (d) ----->

```

Figure 4

The second example considers DPLPMTUD with the MinPMTU HBH option set on a connectivity probe packet.

The IPv6 option is sent end-to-end, and the Min-PMTU is updated by a router on the path to d' , which is returned in a response that also sets the MinPMTU HBH option. Upon receiving Rtn-PMTU value is received, DPLPMTUD immediately sends a probe packet of the target size (d'). If the probe packet is confirmed for the path, the PLPMTU is updated, allowing the source to use data packets up to size d' . (The search algorithm is allowed to continue to probe to see if the path supports a larger size.) Packets of data are never sent with a size larger than the last confirmed probe size, d' .

```

----Packets of data size (a) ----->
----Connectivity probe with MinPMTU-
      +--updated to minPMTU=d'----->
<-----ACK with Rtn-PMTU=d'-----
----Packets of data size (a) ----->
----Probe size (d') ----->
<----- ACK of probe -----
----Packets of data size (d') ----->
Search phase completes.
----Packets of data size (d') ----->

```

Figure 5

The final example considers DPLPMTUD with the MinPMTU HBH option set on a connectivity probe packet, but shows the effect when this connectivity probe packet is dropped.

In this case, the packet with the MinPMTU HBH option is not received. DPLPMTUD searches using probe packets of increasing size, increasing the PLPMTU when the probes are confirmed. An ICMPv6 PTB message is received when the probed size exceeds the actual PMTU, indicating a PTB_SIZE of d'. DPLPMTUD immediately sends a probe packet of the target size (d'). If the probe packet is confirmed for the path, the PLPMTU is updated, allowing the source to use data packets up to size d'. If the ICMPv6 PTB message is not received, the DPLPMTU will be the last confirmed probe size, d.

```

----Packets of data size (a) ----->
----Connectivity probe with MinPMTU -----X
----Packets of data size (a) ----->
----Probe size (b) ----->
<----- ACK of probe -----
----Packets of data size (b) ----->
----Probe size (c) ----->
<----- ACK of probe -----
----Packets of data size (c) ----->
----Probe size (d) ----->
<----- ACK of probe -----
----Packets of data size (d) ----->
----Probe size (e) -----X
<--ICMPv6 PTB PTB_SIZE(d') -|
----Packets of data size (d) ----->
----Probe size (d') using target set by PTB_SIZE ----->
<----- ACK of probe -----
Search phase completes.
----Packets of data size (d') ----->

```

Figure 6

The number of probe rounds depends on the number of steps needed by the search algorithm, and is typically larger for a larger PMTU.

Authors' Addresses

Robert M. Hinden
 Check Point Software
 959 Skyway Road
 San Carlos, CA 94070
 United States of America

Email: bob.hinden@gmail.com

Godred Fairhurst
University of Aberdeen
School of Engineering
Fraser Noble Building
Aberdeen
AB24 3UE
United Kingdom
Email: gorrry@erg.abdn.ac.uk

Path Aware Networking RG
Internet-Draft
Intended status: Informational
Expires: 29 July 2022

B. Trammell
Google Switzerland GmbH
25 January 2022

Current Open Questions in Path Aware Networking
draft-irtf-panrg-questions-12

Abstract

In contrast to the present Internet architecture, a path-aware internetworking architecture has two important properties: it exposes the properties of available Internet paths to endpoints, and provides for endpoints and applications to use these properties to select paths through the Internet for their traffic. While this property of "path awareness" already exists in many Internet-connected networks within single domains and via administrative interfaces to the network layer, a fully path-aware internetwork expands these concepts across layers and across the Internet.

This document poses questions in path-aware networking open as of 2021, that must be answered in the design, development, and deployment of path-aware internetworks. It was originally written to frame discussions in the Path Aware Networking proposed Research Group (PANRG), and has been published to snapshot current thinking in this space.

Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at <https://github.com/panrg/questions>.

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 29 July 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction to Path-Aware Networking	2
1.1. Definitions	4
2. Questions	4
2.1. A Vocabulary of Path Properties	5
2.2. Discovery, Distribution, and Trustworthiness of Path Properties	5
2.3. Supporting Path Selection	6
2.4. Interfaces for Path Awareness	6
2.5. Implications of Path Awareness for the Transport and Application Layers	7
2.6. What is an Endpoint?	7
2.7. Operating a Path Aware Network	8
2.8. Deploying a Path Aware Network	8
3. Acknowledgments	9
4. Informative References	10
Author's Address	10

1. Introduction to Path-Aware Networking

In the current Internet architecture, the network layer provides a best-effort service to the endpoints using it, without verifiability of the properties of the path between the endpoints. While there are network layer technologies that attempt better-than-best-effort delivery, the interfaces to these are generally administrative as opposed to endpoint-exposed (e.g. Path Computation Element (PCE))

[RFC4655] and Software-Defined Wide Area Network (SD-WAN) approaches), and they are often restricted to single administrative domains. In this architecture, an application can assume that a packet with a given destination address will eventually be forwarded toward that destination, but little else.

A transport layer protocol such as TCP can provide reliability over this best-effort service, and a protocol above the network layer, such as Transport Layer Security (TLS) [RFC8446] can authenticate the remote endpoint. However, little, if any, explicit information about the path is available to the endpoints, and any assumptions made about that path often do not hold. These sometimes have serious impacts on the application, as in the case with BGP hijacking attacks.

By contrast, in a path-aware internetworking architecture, endpoints can select or influence the path(s) through the network used by any given packet or flow. The network and transport layers explicitly expose information about the path or paths available to the endpoints and to the applications running on them, so that they can make this selection. The Application Layer Traffic Optimization (ALTO) protocol [RFC7285] can be seen as an example of a path-awareness approach implemented in transport-layer terms on the present Internet protocol stack.

Path selection provides explicit visibility and control of network treatment to applications and users of the network. This selection is available to the application, transport, and/or network layer entities at each endpoint. Path control at the flow and subflow level enables the design of new transport protocols that can leverage multipath connectivity across disjoint paths through the Internet, even over a single physical interface. When exposed to applications, or to end-users through a system configuration interface, path control allows the specification of constraints on the paths that traffic should traverse, for instance to confound passive surveillance in the network core [RFC7624].

We note that this property of "path awareness" already exists in many Internet-connected networks within single domains. Indeed, much of the practice of network engineering using encapsulation at layer 3 can be said to be "path aware", in that it explicitly assigns traffic at tunnel endpoints to a given path within the network. Path-aware internetworking seeks to extend this awareness across domain boundaries without resorting to overlays, except as a transition technology.

This document presents a snapshot of open questions in this space that will need to be answered in order to realize a path-aware internetworking architecture; it is published to further frame discussions within and outside the Path Aware Networking Research Group, and is published with the rough consensus of that group.

1.1. Definitions

For purposes of this document, "path aware networking" describes endpoint discovery of the properties of paths they use for communication across an internetwork, and endpoint reaction to these properties that affects routing and/or data transfer. Note that this can and already does happen to some extent in the current Internet architecture; this definition expands current techniques of path discovery and manipulation to cross administrative domain boundaries and up to the transport and application layers at the endpoints.

Expanding on this definition, a "path aware internetwork" is one in which endpoint discovery of path properties and endpoint selection of paths used by traffic exchanged by the endpoint are explicitly supported, regardless of the specific design of the protocol features which enable this discovery and selection.

A "path", for the purposes of these definitions, is abstractly defined as a sequence of adjacent path elements over which a packet can be transmitted, where the definition of "path element" is technology-dependent. As this document is intended to pose questions rather than answer them, it assumes that this definition will be refined as part of the answer the first two questions it poses, about the vocabulary of path properties and how they are disseminated.

Research into path aware internetworking covers any and all aspects of designing, building, and operating path aware internetworks or the networks and endpoints attached to them. This document presents a collection of research questions to address in order to make a path aware Internet a reality.

2. Questions

Realizing path-aware networking requires answers to a set of open research questions. This document poses these questions, as a starting point for discussions about how to realize path awareness in the Internet, and to direct future research efforts within the Path Aware Networking Research Group.

2.1. A Vocabulary of Path Properties

The first question: how are paths and path properties defined and represented?

In order for information about paths to be exposed to an endpoint, and for the endpoint to make use of that information, it is necessary to define a common vocabulary for paths through an internetwork, and properties of those paths. The elements of this vocabulary could include terminology for components of a path and properties defined for these components, for the entire path, or for subpaths of a path. These properties may be relatively static, such as the presence of a given node or service function on the path; as well as relatively dynamic, such as the current values of metrics such as loss and latency.

This vocabulary and its representation must be defined carefully, as its design will have impacts on the properties (e.g., expressiveness, scalability, security) of a given path-aware internetworking architecture. For example, a system that exposes node-level information for the topology through each network would maximize information about the individual components of the path at the endpoints, at the expense of making internal network topology universally public, which may be in conflict with the business goals of each network's operator. Furthermore, properties related to individual components of the path may change frequently and may quickly become outdated. However, aggregating the properties of individual components to distill end-to-end properties for the entire path is not trivial.

2.2. Discovery, Distribution, and Trustworthiness of Path Properties

The second question: how do endpoints and applications get access to accurate, useful, and trustworthy path properties?

Once endpoints and networks have a shared vocabulary for expressing path properties, the network must have some method for distributing those path properties to the endpoints. Regardless of how path property information is distributed, the endpoints require a method to authenticate the properties -- to determine that they originated from and pertain to the path that they purport to.

Choices in distribution and authentication methods will have impacts on the scalability of a path-aware architecture. Possible dimensions in the space of distribution methods include in-band versus out-of-band, push versus pull versus publish-subscribe, and so on. There are temporal issues with path property dissemination as well, especially with dynamic properties, since the measurement or

elicitation of dynamic properties may be outdated by the time that information is available at the endpoints, and interactions between the measurement and dissemination delay may exhibit pathological behavior for unlucky points in the parameter space.

2.3. Supporting Path Selection

The third question: how can endpoints select paths to use for traffic in a way that can be trusted by the network, the endpoints, and the applications using them?

Access to trustworthy path properties is only half of the challenge in establishing a path-aware architecture. Endpoints must be able to use this information in order to select paths for specific traffic they send. As with the dissemination of path properties, choices made in path selection methods will also have an impact on the tradeoff between scalability and expressiveness of a path-aware architecture. One key choice here is between in-band and out-of-band control of path selection. Another is granularity of path selection (whether per packet, per flow, or per larger aggregate), which also has a large impact on the scalability/expressiveness tradeoff. Path selection must, like path property information, be trustworthy, such that the result of a path selection at an endpoint is predictable. Moreover, any path selection mechanism should aim to provide an outcome that is not worse than using a single path, or selecting paths at random.

Path selection may be exposed in terms of the properties of the path or the identity of elements of the path. In the latter case, a path may be identified at any of multiple layers (e.g. routing domain identifier, network layer address, higher-layer identifier or name, and so on). In this case, care must be taken to present semantically useful information to those making decisions about which path(s) to trust.

2.4. Interfaces for Path Awareness

The fourth question: how can interfaces among the network, transport, and application layers support the use of path awareness?

In order for applications to make effective use of a path-aware networking architecture, the control interfaces presented by the network and transport layers must also expose path properties to the application in a useful way, and provide a useful set of paths among which the application can select. Path selection must be possible based not only on the preferences and policies of the application developer, but of end-users as well. Also, the path selection interfaces presented to applications and end users will need to

support multiple levels of granularity. Most applications' requirements can be satisfied with the expression of path selection policies in terms of properties of the paths, while some applications may need finer-grained, per-path control. These interfaces will need to support incremental development and deployment of applications, and provide sensible defaults, to avoid hindering their adoption.

2.5. Implications of Path Awareness for the Transport and Application Layers

The fifth question: how should transport-layer and higher layer protocols be redesigned to work most effectively over a path-aware networking layer?

In the current Internet, the basic assumption that at a given time all traffic for a given flow will receive the same network treatment and traverse the same path or equivalent paths often holds. In a path aware network, this assumption is more easily violated. The weakening of this assumption has implications for the design of protocols above any path-aware network layer.

For example, one advantage of multipath communication is that a given end-to-end flow can be "sprayed" along multiple paths in order to confound attempts to collect data or metadata from those flows for pervasive surveillance purposes [RFC7624]. However, the benefits of this approach are reduced if the upper-layer protocols use linkable identifiers on packets belonging to the same flow across different paths. Clients may mitigate linkability by opting to not re-use cleartext connection identifiers, such as TLS session IDs or tickets, on separate paths. The privacy-conscious strategies required for effective privacy in a path-aware Internet are only possible if higher-layer protocols such as TLS permit clients to obtain unlinkable identifiers.

2.6. What is an Endpoint?

The sixth question: how is path awareness (in terms of vocabulary and interfaces) different when applied to tunnel and overlay endpoints?

The vision of path-aware networking articulated so far makes an assumption that path properties will be disseminated to endpoints on which applications are running (terminals with user agents, servers, and so on). However, incremental deployment may require that a path-aware network "core" be used to interconnect islands of legacy protocol networks. In these cases, it is the gateways, not the application endpoints, that receive path properties and make path selections for that traffic. The interfaces provided by this gateway are necessarily different than those a path-aware networking layer

provides to its transport and application layers, and the path property information the gateway needs and makes available over those interfaces may also be different.

2.7. Operating a Path Aware Network

The seventh question: how can a path aware network in a path aware internetwork be effectively operated, given control inputs from network administrators, application designers, and end users?

The network operations model in the current Internet architecture assumes that traffic flows are controlled by the decisions and policies made by network operators, as expressed in interdomain and intradomain routing protocols. In a network providing path selection to the endpoints, however, this assumption no longer holds, as endpoints may react to path properties by selecting alternate paths. Competing control inputs from path-aware endpoints and the routing control plane may lead to more difficult traffic engineering or nonconvergent forwarding, especially if the endpoints' and operators' notion of the "best" path for given traffic diverges significantly. The degree of difficulty may depend on the fidelity of information made available to path selection algorithms at the endpoints. Explicit path selection can also specify outbound paths, while BGP policies are expressed in terms of inbound traffic.

A concept for path aware network operations will need to have clear methods for the resolution of apparent (if not actual) conflicts of intent between the network's operator and the path selection at an endpoint. It will also need set of safety principles to ensure that increasing path control does not lead to decreasing connectivity; one such safety principle could be "the existence of at least one path between two endpoints guarantees the selection of at least one path between those endpoints."

2.8. Deploying a Path Aware Network

The eighth question: how can the incentives of network operators and end-users be aligned to realize the vision of path aware networking, and how can the transition from current ("path-oblivious") to path-aware networking be managed?

The vision presented in the introduction discusses path aware networking from the point of view of the benefits accruing at the endpoints, to designers of transport protocols and applications as well as to the end users of those applications. However, this vision requires action not only at the endpoints but also within the interconnected networks offering path aware connectivity. While the specific actions required are a matter of the design and

implementation of a specific realization of a path aware protocol stack, it is clear than any path aware architecture will require network operators to give up some control of their networks over to endpoint-driven control inputs.

Here the question of apparent versus actual conflicts of intent arises again: certain network operations requirements may appear essential, but are merely accidents of the interfaces provided by current routing and management protocols. For example, related (but adjacent) to path aware networking, the widespread use of the TCP wire image [RFC8546] in network monitoring for DDoS prevention appears in conflict with the deployment of encrypted transports, only because path signaling [RFC8558] has been implicit in the deployment of past transport protocols.

Similarly, incentives for deployment must show how existing network operations requirements are met through new path selection and property dissemination mechanisms.

The incentives for network operators and equipment vendors need to be made clear, in terms of a plan to transition [RFC8170] an internetwork to path-aware operation, one network and facility at a time. This plan to transition must also take into account that the dynamics of path aware networking early in this transition (when few endpoints and flows in the Internet use path selection) may be different than those later in the transition.

Aspects of data security and information management in a network that explicitly radiates more information about the network's deployment and configuration, and implicitly radiates information about endpoint configuration and preference through path selection, must also be addressed.

3. Acknowledgments

Many thanks to Adrian Perrig, Jean-Pierre Smith, Mirja Kuehlewind, Olivier Bonaventure, Martin Thomson, Shwetha Bhandari, Chris Wood, Lee Howard, Mohamed Boucadair, Thorben Krueger, Gorrry Fairhurst, Spencer Dawkins, Reese Enghardt, Laurent Ciavaglia, Stephen Farrell, and Richard Yang, for discussions leading to questions in this document, and for feedback on the document itself.

This work is partially supported by the European Commission under Horizon 2020 grant agreement no. 688421 Measurement and Architecture for a Middleboxed Internet (MAMI), and by the Swiss State Secretariat for Education, Research, and Innovation under contract no. 15.0268. This support does not imply endorsement.

4. Informative References

- [RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/rfc/rfc4655>>.
- [RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <<https://www.rfc-editor.org/rfc/rfc7285>>.
- [RFC7624] Barnes, R., Schneier, B., Jennings, C., Hardie, T., Trammell, B., Huitema, C., and D. Borkmann, "Confidentiality in the Face of Pervasive Surveillance: A Threat Model and Problem Statement", RFC 7624, DOI 10.17487/RFC7624, August 2015, <<https://www.rfc-editor.org/rfc/rfc7624>>.
- [RFC8170] Thaler, D., Ed., "Planning for Protocol Adoption and Subsequent Transitions", RFC 8170, DOI 10.17487/RFC8170, May 2017, <<https://www.rfc-editor.org/rfc/rfc8170>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/rfc/rfc8446>>.
- [RFC8546] Trammell, B. and M. Kuehlewind, "The Wire Image of a Network Protocol", RFC 8546, DOI 10.17487/RFC8546, April 2019, <<https://www.rfc-editor.org/rfc/rfc8546>>.
- [RFC8558] Hardie, T., Ed., "Transport Protocol Path Signals", RFC 8558, DOI 10.17487/RFC8558, April 2019, <<https://www.rfc-editor.org/rfc/rfc8558>>.

Author's Address

Brian Trammell
Google Switzerland GmbH
Gustav-Gull-Platz 1
CH- 8004 Zurich
Switzerland

Email: ietf@trammell.ch

PANRG
Internet-Draft
Intended status: Informational
Expires: 27 September 2021

S. Dawkins, Ed.
Tencent America
26 March 2021

Path Aware Networking: Obstacles to Deployment (A Bestiary of Roads Not
Taken)
draft-irtf-panrg-what-not-to-do-19

Abstract

At the first meeting of the Path Aware Networking Research Group, the research group agreed to catalog and analyze past efforts to develop and deploy Path Aware techniques, most of which were unsuccessful or at most partially successful, in order to extract insights and lessons for path-aware networking researchers.

This document contains that catalog and analysis.

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 27 September 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction
 - 1.1. What Do "Path" and "Path Awareness" Mean in this Document?
2. A Perspective On This Document
 - 2.1. Notes for the Reader
 - 2.2. A Note About Path-Aware Techniques Included In This Document
 - 2.3. Venue for Discussion of this Document
 - 2.4. Architectural Guidance

- 2.5. Terminology Used in this Document
- 2.6. Methodology for Contributions
- 3. Applying the Lessons We've Learned
- 4. Summary of Lessons Learned
 - 4.1. Justifying Deployment
 - 4.2. Providing Benefits for Early Adopters
 - 4.3. Providing Benefits During Partial Deployment
 - 4.4. Outperforming End-to-end Protocol Mechanisms
 - 4.5. Paying for Path Aware Techniques
 - 4.6. Impact on Operational Practices
 - 4.7. Per-connection State
 - 4.8. Keeping Traffic on Fast-paths
 - 4.9. Endpoints Trusting Intermediate Nodes
 - 4.10. Intermediate Nodes Trusting Endpoints
 - 4.11. Reacting to Distant Signals
 - 4.12. Support in Endpoint Protocol Stacks
 - 4.13. Planning For Failure
- 5. Future Work
- 6. Contributions
 - 6.1. Stream Transport (ST, ST2, ST2+)
 - 6.1.1. Reasons for Non-deployment
 - 6.1.2. Lessons Learned.
 - 6.2. Integrated Services (IntServ)
 - 6.2.1. Reasons for Non-deployment
 - 6.2.2. Lessons Learned.
 - 6.3. Quick-Start TCP
 - 6.3.1. Reasons for Non-deployment
 - 6.3.2. Lessons Learned
 - 6.4. ICMP Source Quench
 - 6.4.1. Reasons for Non-deployment
 - 6.4.2. Lessons Learned
 - 6.5. Triggers for Transport (TRIGTRAN)
 - 6.5.1. Reasons for Non-deployment
 - 6.5.2. Lessons Learned.
 - 6.6. Shim6
 - 6.6.1. Reasons for Non-deployment
 - 6.6.2. Lessons Learned
 - 6.6.3. Addendum on MultiPath TCP
 - 6.7. Next Steps in Signaling (NSIS)
 - 6.7.1. Reasons for Non-deployment
 - 6.7.2. Lessons Learned
 - 6.8. IPv6 Flow Label
 - 6.8.1. Reasons for Non-deployment
 - 6.8.2. Lessons Learned
 - 6.9. Explicit Congestion Notification (ECN)
 - 6.9.1. Reasons for Non-deployment
 - 6.9.2. Lessons Learned
- 7. Security Considerations
- 8. IANA Considerations
- 9. Acknowledgments
- 10. Informative References
- Author's Address

1. Introduction

This document describes the lessons that IETF participants have learned (and learned the hard way) about Path Aware Networking over a period of several decades, and provides an analysis of reasons why various Path Aware Networking techniques have seen limited or no deployment.

1.1. What Do "Path" and "Path Awareness" Mean in this Document?

One of the first questions reviewers of this document have asked is "what's the definition of a path, and what's the definition of path awareness?" That is not an easy question to answer for this document.

These terms have definitions in other [PANRG] documents, and are still the subject of some discussion in the research group, as of the date of this document. But because this document reflects work performed over several decades, the technologies described in Section 6 significantly predate the current definitions of "path" and "path aware" in use in the Path Aware Networking Research Group, and it is unlikely that all the contributors to Section 6 would have had the same understanding of these terms. Those technologies were considered "path aware" in early PANRG discussions, and so are included in this retrospective document.

It is worth noting that the definitions of "path" and "path aware" in [I-D.irtf-panrg-path-properties] would apply to path aware networking techniques at a number of levels of the Internet protocol architecture ([RFC1122], plus several decades of refinements), but the contributions received for this document tended to target the Transport Layer, and to treat a "path" constructed by routers as a "black box". It would be useful to consider how applicable the Lessons Learned cataloged in this document are, at other layers, and that would be a fine topic for follow-on research.

The current definition of "Path" in the Path Aware Networking Research Group appears in Section 2 ("Terminology") in [I-D.irtf-panrg-path-properties]. That definition is included here as a convenience to the reader.

Path: A sequence of adjacent path elements over which a packet can be transmitted, starting and ending with a node. A path is unidirectional. Paths are time-dependent, i.e., the sequence of path elements over which packets are sent from one node to another may change. A path is defined between two nodes. For multicast or broadcast, a packet may be sent by one node and received by multiple nodes. In this case, the packet is sent over multiple paths at once, one path for each combination of sending and receiving node; these paths do not have to be disjoint. Note that an entity may have only partial visibility of the path elements that comprise a path and visibility may change over time. Different entities may have different visibility of a path and/or treat path elements at different levels of abstraction.

The current definition of "Path Awareness", used by the Path Aware Networking Research Group, appears in Section 1.1 ("Definition") in [I-D.irtf-panrg-questions]. That definition is included here as a convenience to the reader.

For purposes of this document, "path aware networking" describes endpoint discovery of the properties of paths they use for communication, and endpoint reaction to these properties that affects routing and/or transmission; note that this can and already does happen to some extent in the current Internet architecture. Expanding on this definition, a "path aware internetwork" is one in which endpoint discovery of path properties and endpoint selection of paths used by traffic exchanged by the endpoint are explicitly supported, regardless of the specific design of the protocol features which enable this discovery and selection.

2. A Perspective On This Document

At the first meeting of the Path Aware Networking Research Group [PANRG], at IETF 99 [PANRG-99], Olivier Bonaventure led a discussion of "A Decade of Path Awareness" [PATH-Decade], on attempts, which were mostly unsuccessful for a variety of reasons, to exploit Path Aware techniques and achieve a variety of goals over the past decade. At the end of that discussion, two things were abundantly clear.

- * The Internet community has accumulated considerable experience with many Path Aware techniques over a long period of time, and
- * Although some path aware techniques have been deployed (for example, Differentiated Services, or DiffServ [RFC2475]), most of these techniques haven't seen widespread adoption and deployment. Even "successful" techniques like DiffServ can face obstacles that prevents wider usage. The reasons for non-adoption and limited adoption and deployment are many, and are worthy of study.

The meta-lessons from that experience were

- * Path aware networking has been more Research than Engineering, so establishing an IRTF Research Group for Path Aware Networking is the right thing to do [RFC7418].
- * Analyzing a catalog of past experience to learn the reasons for non-adoption would be a great first step for the Research Group.

Allison Mankin, as IRTF Chair, officially chartered the Path Aware Networking Research Group in July, 2018.

This document contains the analysis performed by that research group (Section 4), based on that catalog (Section 6).

This document represents the consensus of the Path Aware Networking Research Group.

2.1. Notes for the Reader

This Informational document discusses Path Aware protocol mechanisms considered, and in some cases standardized, by the Internet Engineering Task Force (IETF), and considers Lessons Learned from those mechanisms. The intention is to inform the work of protocol designers, whether in the IRTF, the IETF, or elsewhere in the Internet ecosystem.

As an Informational document published in the IRTF stream, this document has no authority beyond the quality of the analysis it contains.

2.2. A Note About Path-Aware Techniques Included In This Document

This document does not catalog every proposed path aware networking technique that was not adopted and deployed. Instead, we limited our focus to technologies that passed through the IETF community, and still identified enough techniques to provide background for the lessons included in Section 4 to inform researchers and protocol engineers in their work.

No shame is intended for the techniques included in this document. As shown in Section 4, the quality of specific techniques had little

to do with whether they were deployed or not. Based on the techniques cataloged in this document, it is likely that when these techniques were put forward, the proponents were trying to engineer something that could not be engineered without first carrying out research. Actual shame would be failing to learn from experience, and failing to share that experience with other networking researchers and engineers.

2.3. Venue for Discussion of this Document

(RFC Editor: please remove this section before publication)

Discussion of specific contributed experiences and this document in general should take place on the PANRG mailing list.

2.4. Architectural Guidance

As background for understanding the Lessons Learned contained in this document, the reader is encouraged to become familiar with the Internet Architecture Board's documents on "What Makes for a Successful Protocol?" [RFC5218] and "Planning for Protocol Adoption and Subsequent Transitions" [RFC8170].

Although these two documents do not specifically target path-aware networking protocols, they are helpful resources for readers seeking to improve their understanding of considerations for successful adoption and deployment of any protocol. For example, the Basic Success Factors described in Section 2.1 of [RFC5218] are helpful for readers of this document.

Because there is an economic aspect to decisions about deployment, the IAB Workshop on Internet Technology Adoption and Transition [ITAT] report [RFC7305] also provides food for thought.

Several of the Lessons Learned in Section 4 reflect considerations described in [RFC5218], [RFC7305], and [RFC8170].

2.5. Terminology Used in this Document

The terms Node and Element in this document have the meaning defined in [I-D.irtf-panrg-path-properties].

2.6. Methodology for Contributions

This document grew out of contributions by various IETF participants with experience with one or more Path Aware Networking techniques.

There are many things that could be said about the Path Aware networking techniques that have been developed. For the purposes of this document, contributors were requested to provide

- * the name of a technique, including an abbreviation if one was used
- * if available, a long-term pointer to the best reference describing the technique
- * a short description of the problem the technique was intended to solve
- * a short description of the reasons why the technique wasn't adopted

- * a short statement of the lessons that researchers can learn from our experience with this technique.

3. Applying the Lessons We've Learned

The initial scope for this document was roughly "what mistakes have we made in the decade prior to [PANRG-99], that we shouldn't make again". Some of the contributions in Section 6 predate the initial scope. The earliest Path-Aware Networking technique referred to in Section 6 is Section 6.1, published in the late 1970s. Given that the networking ecosystem has evolved continuously, it seems reasonable to consider how to apply these lessons.

The PANRG Research Group reviewed the Lessons Learned (Section 4) contained in the May 23, 2019 version of this document at IETF 105 [PANRG-105-Min], and carried out additional discussion at IETF 106 [PANRG-106-Min]. Table 1 provides the "sense of the room" about each lesson after those discussions. The intention was to capture whether a specific lesson seems to be

- * "Invariant" - well-understood and is likely to be applicable for any proposed Path Aware Networking solution.
- * "Variable" - has impeded deployment in the past, but might not be applicable in a specific technique. Engineering analysis to understand whether the lesson is applicable is prudent.
- * "Not Now" - this characteristic tends to turn up a minefield full of dragons, and prudent network engineers will wish to avoid gambling on a technique that relies on this, until something significant changes

Section 6.9 on ECN was added during the review and approval process, based on a question from Martin Duke. That section, along with its Lessons Learned and place in the "Invariant"/"Variable"/"Not Now" taxonomy, as contained in the March 8, 2021 version of this document, was discussed at [PANRG-110].

Lesson	Category
Justifying Deployment (Section 4.1)	Invariant
Providing Benefits for Early Adopters (Section 4.2)	Invariant
Providing Benefits during Partial Deployment (Section 4.3)	Invariant
Outperforming End-to-end Protocol Mechanisms (Section 4.4)	Variable
Paying for Path Aware Techniques (Section 4.5)	Invariant
Impact on Operational Practices (Section 4.6)	Invariant
Per-connection State (Section 4.7)	Variable
Keeping Traffic on Fast-paths (Section 4.8)	Variable
Endpoints Trusting Intermediate Nodes (Section 4.9)	Not Now
Intermediate Nodes Trusting Endpoints	Not Now

(Section 4.10)	
Reacting to Distant Signals (Section 4.11)	Variable
Support in Endpoint Protocol Stacks (Section 4.12)	Variable
Planning for Failure (Section 4.13)	Invariant

Table 1

"Justifying Deployment", "Providing Benefits for Early Adopters", "Paying for Path Aware Techniques", "Impact on Operational Practice", and "Planning for Failure" were considered to be invariant - the sense of the room was that these would always be considerations for any proposed Path Aware Technique.

"Providing Benefits During Partial Deployment" was added after IETF 105, during research group last call, and is also considered to be invariant.

For "Outperforming End-to-end Protocol Mechanisms", there is a trade-off between improved performance from Path Aware Techniques and additional complexity required by some Path Aware Techniques.

- * For example, if you can obtain the same understanding of path characteristics from measurements obtained over a few more round trips, endpoint implementers are unlikely to be eager to add complexity, and many attributes can be measured from an endpoint, without assistance from intermediate nodes.

For "Per-connection State", the key questions discussed in the research group were "how much state" and "where state is maintained".

- * IntServ (Section 6.2) required state at every intermediate node for every connection between two endpoints. As the Internet ecosystem has evolved, carrying many connections in a tunnel that appears to intermediate nodes as a single connection has become more common, so that additional end-to-end connections don't add additional state to intermediate nodes between tunnel endpoints. If these tunnels are encrypted, intermediate nodes between tunnel endpoints can't distinguish between connections, even if that were desirable.

For "Keeping Traffic on Fast-paths", we noted that this was true for many platforms, but not for all.

- * For backbone routers, this is likely an invariant, but for platforms that rely more on general-purpose computers to make forwarding decisions, this may not be a fatal flaw for Path Aware Networking techniques.

For "Endpoints Trusting Intermediate Nodes" and "Intermediate Nodes Trusting Endpoints", these lessons point to the broader need to revisit the Internet Threat Model.

- * We noted with relief that discussions about this were already underway in the IETF community at IETF 105 (see the Security Area Open Meeting minutes [SAAG-105-Min] for discussion of [I-D.arkko-arch-internet-threat-model] and [I-D.farrell-etm]), and the Internet Architecture Board has created a mailing list for continued discussions ([model-t]), but we recognize that there are

Path Aware Networking aspects of this effort, requiring research.

For "Reacting to Distant Signals", we noted that not all attributes are equal.

- * If an attribute is stable over an extended period of time, is difficult to observe via end-to-end mechanisms, and is valuable, Path Aware Techniques that rely on that attribute to provide a significant benefit become more attractive.
- * Analysis to help identify attributes that are useful enough to justify deployment of Path Aware techniques that make use of those attributes would be helpful.

For "Support in Endpoint Protocol Stacks", we noted that Path Aware applications must be able to identify and communicate requirements about path characteristics.

- * The de-facto sockets API has no way of signaling application expectations for the network path to the protocol stack.

4. Summary of Lessons Learned

This section summarizes the Lessons Learned from the contributed subsections in Section 6.

Each Lesson Learned is tagged with one or more contributions that encountered this obstacle as a significant impediment to deployment. Other contributed techniques may have also encountered this obstacle, but this obstacle may not have been the biggest impediment to deployment for those techniques.

It is useful to notice that sometimes an obstacle might impede deployment, while at other times, the same obstacle might prevent adoption and deployment entirely. The research group discussed distinguishing between obstacles that impede and obstacles that prevent, but it appears that the boundary between "impede" and "prevent" can shift over time - some of the Lessons Learned are based on both Path Aware techniques that were not deployed, and Path Aware techniques that were deployed, but were not deployed widely or quickly. See Section 6.6 and Section 6.6.3 as one example of this shifting boundary.

4.1. Justifying Deployment

The benefit of Path Awareness must be great enough to justify making changes in an operational network. The colloquial U.S. American English expression, "If it ain't broke, don't fix it" is a "best current practice" on today's Internet. (See Section 6.3, Section 6.4, Section 6.5, and Section 6.9, in addition to [RFC5218]).

4.2. Providing Benefits for Early Adopters

Providing benefits for early adopters can be key - if everyone must deploy a technique in order for the technique to provide benefits, or even to work at all, the technique is unlikely to be adopted widely or quickly. (See Section 6.2 and Section 6.3, in addition to [RFC5218]).

4.3. Providing Benefits During Partial Deployment

Some proposals require that all path elements along the full length

of the path must be upgraded to support a new technique, before any benefits can be seen. This is likely to require coordination between operators who control a subset of path elements, and between operators and end users if endpoint upgrades are required. If a technique provides benefits when only a part of the path has been upgraded, this is likely to encourage adoption and deployment. (See Section 6.2, Section 6.3, and Section 6.9, in addition to [RFC5218]).

4.4. Outperforming End-to-end Protocol Mechanisms

Adaptive end-to-end protocol mechanisms may respond to feedback quickly enough that the additional realizable benefit from a new Path Aware mechanism that tries to manipulate nodes along a path, or observe the attributes of nodes along a path, may be much smaller than anticipated (See Section 6.3 and Section 6.5).

4.5. Paying for Path Aware Techniques

"Follow the money." If operators can't charge for a Path Aware technique to recover the costs of deploying it, the benefits to the operator must be really significant. Corollary: If operators charge for a Path Aware technique, the benefits to users of that Path Aware technique must be significant enough to justify the cost. (See Section 6.1, Section 6.2, Section 6.5, and Section 6.9).

4.6. Impact on Operational Practices

Impact of a Path Aware technique requiring changes to operational practices can affect how quickly or widely a promising technique is deployed. The impacts of these changes may make deployment more likely, but often discourage deployment. (See Section 6.6, including Section 6.6.3).

4.7. Per-connection State

Per-connection state in intermediate nodes has been an impediment to adoption and deployment in the past, because of added cost and complexity. Often, similar benefits can be achieved with much less finely-grained state. This is especially true as we move from the edge of the network, further into the routing core (See Section 6.1 and Section 6.2).

4.8. Keeping Traffic on Fast-paths

Many modern platforms, especially high-end routers, have been designed with hardware that can make simple per-packet forwarding decisions ("fast-paths"), but have not been designed to make heavy use of in-band mechanisms such as IPv4 and IPv6 Router Alert Options (RAO) that require more processing to make forwarding decisions. Packets carrying in-band mechanisms are diverted to other processors in the router with much lower packet processing rates. Operators can be reluctant to deploy techniques that rely heavily on in-band mechanisms because they may significantly reduce packet throughput. (See Section 6.7).

4.9. Endpoints Trusting Intermediate Nodes

If intermediate nodes along the path can't be trusted, it's unlikely that endpoints will rely on signals from intermediate nodes to drive changes to endpoint behaviors. We note that "trust" is not binary - one, low, level of trust applies when a node issuing a message can confirm that it has visibility of the packets on the path it is

seeking to control [RFC8085] (e.g., an ICMP message included a quoted packet from the source). A higher level of trust can arise when an endpoint has established a short term, or even long term, trust relationship with network nodes. (See Section 6.4 and Section 6.5).

4.10. Intermediate Nodes Trusting Endpoints

If the endpoints do not have any trust relationship with the intermediate nodes along a path, operators have been reluctant to deploy techniques that rely on endpoints sending unauthenticated control signals to routers. (See Section 6.2 and Section 6.7). (We also note this still remains a factor hindering deployment of DiffServ).

4.11. Reacting to Distant Signals

Because the Internet is a distributed system, if the distance that information from distant path elements travels to a Path Aware host is sufficiently large, the information may no longer accurately represent the state and situation at the distant host or elements along the path when it is received locally. In this case, the benefit that a Path Aware technique provides will be inconsistent, and may not always be beneficial. (See Section 6.3).

4.12. Support in Endpoint Protocol Stacks

Just because a protocol stack provides a new feature/signal does not mean that applications will use the feature/signal. Protocol stacks may not know how to effectively utilize Path-Aware techniques, because the protocol stack may require information from applications to permit the technique to work effectively, but applications may not a-priori know that information. Even if the application does know that information, the de-facto sockets API has no way of signaling application expectations for the network path to the protocol stack. In order for applications to provide these expectations to protocol stacks, we need an API that signals more than the packets to be sent. (See Section 6.1 and Section 6.2).

4.13. Planning For Failure

If early implementers discover severe problems with a new feature, that feature is likely to be disabled, and convincing implementers to re-enable that feature can be very difficult, and can require years or decades. In addition to testing, partial deployment for a subset of users, implementing instrumentation that will detect degraded user experience, and even "failback" to a previous version or "failover" to an entirely different implementation are likely to be helpful. (See Section 6.9).

5. Future Work

By its nature, this document has been retrospective. In addition to considering how the Lessons Learned to date apply to current and future Path Aware networking proposals, it's also worth considering whether there is deeper investigation left to do.

- * We note that this work was based on contributions from experts on various Path Aware networking techniques, and all of the contributed techniques involved unicast protocols. We didn't consider how these lessons might apply to multicast, and, given anecdotal reports at the IETF 109 MOPS working group meeting of IP multicast offerings within data centers at one or more cloud

providers ([MOPS-109-Min]), it might be useful to think about path awareness in multicast, before we have a history of unsuccessful deployments to document.

- * The question of whether a mechanism supports admission control, based on either endpoints or applications, is associated with Path Awareness. One of the motivations of IntServ and a number of other architectures (e.g. Deterministic Networking, [RFC8655]) is the ability to "say no" to an application based on resource availability on a path, before the application tries to inject traffic onto that path and discovers the path does not have the capacity to sustain enough utility to meet the application's minimum needs. The question of whether admission control is needed comes up repeatedly, but we have learned a few useful lessons that, while covered implicitly in some of the lessons learned of the document, might be explained explicitly:
 - We have gained a lot of experience with application-based adaptation since the days where applications just injected traffic in-elastically into the network. Such adaptations seem to work well enough that admission control is of less value to these applications
 - There are end-to-end measurement techniques that can steer traffic at the application layer (Content Distribution Networks, multi-CDNs like Conviva [Conviva], etc.)
 - We noted in Section 4.12 that applications often don't know how to utilize Path Aware techniques. This includes not knowing enough about their admission control threshold to be able to ask accurately for the resources they need, whether this is because the application itself doesn't know, or because the application has no way to signal its expectations to the underlying protocol stack. To date, attempts to help them haven't gotten anywhere (e.g. the multiple-TSPEC additions to RSVP to attempt to mirror codec selection by applications [I-D.ietf-tsvwg-intserv-multiple-tspec] expired in 2013).
- * We note that this work took the then-current IP network architecture as given, at least at the time each technique was proposed. It might be useful to consider aspects of the now-current IP network architecture that ease, or impede, Path Aware networking techniques. For example, there is limited ability in IP to constrain bidirectional paths to be symmetric, and information-centric networking protocols such as Named Data Networking (NDN) and Content-Centric Networking (CCNx) ([RFC8793]) must force bidirectional path symmetry using protocol-specific mechanisms.

6. Contributions

Contributions on these Path Aware networking techniques were analyzed to arrive at the Lessons Learned captured in Section 4.

Our expectation is that most readers will not need to read through this section carefully, but we wanted to record these hard-fought lessons as a service to others who may revisit this document, so they'll have the details close at hand.

6.1. Stream Transport (ST, ST2, ST2+)

The suggested references for Stream Transport are:

- * ST - A Proposed Internet Stream Protocol [IEN-119]
- * Experimental Internet Stream Protocol, Version 2 (ST-II) [RFC1190]
- * Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+ [RFC1819]

The first version of Stream Transport, ST [IEN-119], was published in the late 1970's and was implemented and deployed on the ARPANET at small scale. It was used throughout the 1980's for experimental transmission of voice, video, and distributed simulation.

The second version of the ST specification (ST2) [RFC1190] [RFC1819] was an experimental connection-oriented internetworking protocol that operated at the same layer as connectionless IP. ST2 packets could be distinguished by their IP header version numbers (IP, at that time, used version number 4, while ST2 used version number 5).

ST2 used a control plane layered over IP to select routes and reserve capacity for real-time streams across a network path, based on a flow specification communicated by a separate protocol. The flow specification could be associated with QoS state in routers, producing an experimental resource reservation protocol. This allowed ST2 routers along a path to offer end-to-end guarantees, primarily to satisfy the QoS requirements for realtime services over the Internet.

6.1.1. Reasons for Non-deployment

Although implemented in a range of equipment, ST2 was not widely used after completion of the experiments. It did not offer the scalability and fate-sharing properties that have come to be desired by the Internet community.

The ST2 protocol is no longer in use.

6.1.2. Lessons Learned.

As time passed, the trade-off between router processing and link capacity changed. Links became faster and the cost of router processing became comparatively more expensive.

The ST2 control protocol used "hard state" - once a route was established, and resources were reserved, routes and resources existing until they were explicitly released via signaling. A soft-state approach was thought superior to this hard-state approach, and led to development of the IntServ model described in Section 6.2.

6.2. Integrated Services (IntServ)

The suggested references for IntServ are:

- * RFC 1633 Integrated Services in the Internet Architecture: an Overview [RFC1633]
- * RFC 2211 Specification of the Controlled-Load Network Element Service [RFC2211]
- * RFC 2212 Specification of Guaranteed Quality of Service [RFC2212]
- * RFC 2215 General Characterization Parameters for Integrated

Service Network Elements [RFC2215]

- * RFC 2205 Resource ReSerVation Protocol (RSVP) [RFC2205]

In 1994, when the IntServ architecture document [RFC1633] was published, real-time traffic was first appearing on the Internet. At that time, bandwidth was still a scarce commodity. Internet Service Providers built networks over DS3 (45 Mbps) infrastructure, and sub-rate (< 1 Mbps) access was common. Therefore, the IETF anticipated a need for a fine-grained QoS mechanism.

In the IntServ architecture, some applications can require service guarantees. Therefore, those applications use the Resource Reservation Protocol (RSVP) [RFC2205] to signal QoS reservations across network paths. Every router in the network that participates in IntServ maintains per-flow soft-state to a) perform call admission control and b) deliver guaranteed service.

Applications use Flow Specification (Flow Specs) [RFC2210] to describe the traffic that they emit. RSVP reserves capacity for traffic on a per Flow Spec basis.

6.2.1. Reasons for Non-deployment

Although IntServ has been used in enterprise and government networks, IntServ was never widely deployed on the Internet because of its cost. The following factors contributed to operational cost:

- * IntServ must be deployed on every router that is on a path where IntServ is to be used. Although it is possible to include a router that does not participate in IntServ along the path being controlled, if that router is likely to become a bottleneck, IntServ cannot be used to avoid that bottleneck along the path
- * IntServ maintained per flow state

As IntServ was being discussed, the following occurred:

- * For many expected uses, it became more cost effective to solve the QoS problem by adding bandwidth. Between 1994 and 2000, Internet Service Providers upgraded their infrastructures from DS3 (45 Mbps) to OC-48 (2.4 Gbps). This meant that even if an endpoint was using IntServ in an IntServ-enabled network, its requests would rarely, if ever, be denied, so endpoints and Internet Service Providers had little reason to enable IntServ.
- * DiffServ [RFC2475] offered a more cost-effective, albeit less fine-grained, solution to the QoS problem.

6.2.2. Lessons Learned.

The following lessons were learned:

- * Any mechanism that requires every participating onpath router to maintain per-flow state is not likely to succeed, unless the additional cost for offering the feature can be recovered from the user.
- * Any mechanism that requires an operator to upgrade all of its routers is not likely to succeed, unless the additional cost for offering the feature can be recovered from the user.

In environments where IntServ has been deployed, trust relationships with endpoints are very different from trust relationships on the Internet itself, and there are often clearly-defined hierarchies in Service Level Agreements (SLAs), and well-defined transport flows operating with pre-determined capacity and latency requirements over paths where capacity or other attributes are constrained.

IntServ was never widely deployed to manage capacity across the Internet. However, the technique that it produced was deployed for reasons other than bandwidth management. RSVP is widely deployed as an MPLS signaling mechanism. BGP reuses the RSVP concept of Filter Specs to distribute firewall filters, although they are called Flow Spec Component Types in BGP [RFC5575].

6.3. Quick-Start TCP

The suggested references for Quick-Start TCP are:

- * Quick-Start for TCP and IP [RFC4782]
- * Determining an appropriate initial sending rate over an underutilized network path [SAF07]
- * Fast Startup Internet Congestion Control for Broadband Interactive Applications [Sch11]
- * Using Quick-Start to enhance TCP-friendly rate control performance in bidirectional satellite networks [QS-SAT]

Quick-Start [RFC4782] is an Experimental TCP extension that leverages support from the routers on the path to determine an allowed initial sending rate for a path through the Internet, either at the start of data transfers or after idle periods. Without information about the path, a sender cannot easily determine an appropriate initial sending rate. The default TCP congestion control therefore uses the safe but time-consuming slow-start algorithm [RFC5681]. With Quick-Start, connections are allowed to use higher initial sending rates if there is significant unused bandwidth along the path, and if the sender and all of the routers along the path approve the request.

By examining the Time To Live (TTL) field in Quick-Start packets, a sender can determine if routers on the path have approved the Quick-Start request. However, this method is unable to take into account the routers hidden by tunnels or other network nodes invisible at the IP layer.

The protocol also includes a nonce that provides protection against cheating routers and receivers. If the Quick-Start request is explicitly approved by all routers along the path, the TCP host can send at up to the approved rate; otherwise TCP would use the default congestion control. Quick-Start requires modifications in the involved end-systems as well in routers. Due to the resulting deployment challenges, Quick-Start was only proposed in [RFC4782] for controlled environments.

The Quick-Start mechanism is a lightweight, coarse-grained, in-band, network-assisted fast startup mechanism. The benefits are studied by simulation in a research paper [SAF07] that complements the protocol specification. The study confirms that Quick-Start can significantly speed up mid-sized data transfers. That paper also presents router algorithms that do not require keeping per-flow state. Later studies [Sch11] comprehensively analyzes Quick-Start with a full Linux

implementation and with a router fast path prototype using a network processor. In both cases, Quick-Start could be implemented with limited additional complexity.

6.3.1. Reasons for Non-deployment

However, experiments with Quick-Start in [Sch11] revealed several challenges:

- * Having information from the routers along the path can reduce the risk of congestion, but cannot avoid it entirely. Determining whether there is unused capacity is not trivial in actual router and host implementations. Data about available capacity visible at the IP layer may be imprecise, and due to the propagation delay, information can already be outdated when it reaches a sender. There is a trade-off between the speedup of data transfers and the risk of congestion even with Quick-Start. This could be mitigated by only allowing Quick-Start to access a proportion of the unused capacity along a path.
- * For scalable router fast path implementation, it is important to enable parallel processing of packets, as this is a widely used method e.g. in network processors. One challenge is synchronization of information between packets that are processed in parallel, which should be avoided as much as possible.
- * Only some types of application traffic can benefit from Quick-Start. Capacity needs to be requested and discovered. The discovered capacity needs to be utilized by the flow, or it implicitly becomes available for other flows. Failing to use the requested capacity may have already reduced the pool of Quick-Start capacity that was made available to other competing Quick-Start requests. The benefit is greatest when senders use this only for bulk flows and avoid sending unnecessary Quick-Start requests, e.g. for flows that only send a small amount of data. Choosing an appropriate request size requires application-internal knowledge that is not commonly expressed by the transport API. How a sender can determine the rate for an initial Quick-Start request is still a largely unsolved problem.

There is no known deployment of Quick-Start for TCP or other IETF transports.

6.3.2. Lessons Learned

Some lessons can be learned from Quick-Start. Despite being a very light-weight protocol, Quick-Start suffers from poor incremental deployment properties, both regarding the required modifications in network infrastructure as well as its interactions with applications. Except for corner cases, congestion control can be quite efficiently performed end-to-end in the Internet, and in modern stacks there is not much room for significant improvement by additional network support.

After publication of the Quick-Start specification, there have been large-scale experiments with an initial window of up to 10 MSS [RFC6928]. This alternative "IW10" approach can also ramp-up data transfers faster than the standard congestion control, but it only requires sender-side modifications. As a result, this approach can be easier and incrementally deployed in the Internet. While theoretically Quick-Start can outperform "IW10", the improvement in completion time for data transfer times can, in many cases, be small.

After publication of [RFC6928], most modern TCP stacks have increased their default initial window.

6.4. ICMP Source Quench

The suggested references for ICMP Source Quench are:

- * INTERNET CONTROL MESSAGE PROTOCOL [RFC0792]

The ICMP Source Quench message [RFC0792] allowed an on-path router to request the source of a flow to reduce its sending rate. This method allowed a router to provide an early indication of impending congestion on a path to the sources that contribute to that congestion.

6.4.1. Reasons for Non-deployment

This method was deployed in Internet routers over a period of time, the reaction of endpoints to receiving this signal has varied. For low speed links, with low multiplexing of flows the method could be used to regulate (momentarily reduce) the transmission rate. However, the simple signal does not scale with link speed, or the number of flows sharing a link.

The approach was overtaken by the evolution of congestion control methods in TCP [RFC2001], and later also by other IETF transports. Because these methods were based upon measurement of the end-to-end path and an algorithm in the endpoint, they were able to evolve and mature more rapidly than methods relying on interactions between operational routers and endpoint stacks.

After ICMP Source Quench was specified, the IETF began to recommend that transports provide end-to-end congestion control [RFC2001]. The Source Quench method has been obsoleted by the IETF [RFC6633], and both hosts and routers must now silently discard this message.

6.4.2. Lessons Learned

This method had several problems:

First, [RFC0792] did not sufficiently specify how the sender would react to the ICMP Source Quench signal from the path (e.g., [RFC1016]). There was ambiguity in how the sender should utilize this additional information. This could lead to unfairness in the way that receivers (or routers) responded to this message.

Second, while the message did provide additional information, the Explicit Congestion Notification (ECN) mechanism [RFC3168] provided a more robust and informative signal for network nodes to provide early indication that a path has become congested.

The mechanism originated at a time when the Internet trust model was very different. Most endpoint implementations did not attempt to verify that the message originated from an on-path node before they utilized the message. This made it vulnerable to denial of service attacks. In theory, routers might have chosen to use the quoted packet contained in the ICMP payload to validate that the message originated from an on-path node, but this would have increased per-packet processing overhead for each router along the path, would have required transport functionality in the router to verify whether the quoted packet header corresponded to a packet the router had sent. In addition, section 5.2 of [RFC4443] noted ICMPv6-based attacks on

hosts that would also have threatened routers processing ICMPv6 Source Quench payloads. As time passed, it became increasingly obvious that the lack of validation of the messages exposed receivers to a security vulnerability where the messages could be forged to create a tangible denial of service opportunity.

6.5. Triggers for Transport (TRIGTRAN)

The suggested references for TRIGTRAN are:

- * TRIGTRAN BOF at IETF 55 [TRIGTRAN-55]
- * TRIGTRAN BOF at IETF 56 [TRIGTRAN-56]

TCP [RFC0793] has a well-known weakness - the end-to-end flow control mechanism has only a single signal, the loss of a segment, and TCP implementations since the late 1980s have interpreted the loss of a segment as evidence that the path between two endpoints may have become congested enough to exhaust buffers on intermediate hops, so that the TCP sender should "back off" - reduce its sending rate until it knows that its segments are now being delivered without loss [RFC5681]. More modern TCP stacks have added a growing array of strategies about how to establish the sending rate [RFC5681], but when a path is no longer operational, TCP would continue to retry transmissions, which would fail, again, and double their Retransmission Time Out (RTO) timers with each failed transmission, with the result that TCP would wait many seconds before retrying a segment, even if the path becomes operational while the sender is waiting for its next retry.

The thinking behind TRIGTRAN was that if a path completely stopped working because a link along the path was "down", somehow something along the path could signal TCP when that link returned to service, and the sending TCP could retry immediately, without waiting for a full retransmission timeout (RTO) period.

6.5.1. Reasons for Non-deployment

The early dreams for TRIGTRAN were dashed because of an assumption that TRIGTRAN triggers would be unauthenticated. This meant that any "safe" TRIGTRAN mechanism would have relied on a mechanism such as setting the IPv4 TTL or IPv6 Hop Count to 255 at a sender and testing that it was 254 upon receipt, so that a receiver could verify that a signal was generated by an adjacent sender known to be on the path being used, and not some unknown sender which might not even be on the path (e.g., "The Generalized TTL Security Mechanism (GTSM)" [RFC5082]). This situation is very similar to the case for ICMP Source Quench messages as described in Section 6.4, which were also unauthenticated, and could be sent by an off-path attacker, resulting in deprecation of ICMP Source Quench message processing [RFC6633].

TRIGTRAN's scope shrunk from "the path is down" to "the first-hop link is down".

But things got worse.

Because TRIGTRAN triggers would only be provided when the first-hop link was "down", TRIGTRAN triggers couldn't replace normal TCP retransmission behavior if the path failed because some link further along the network path was "down". So TRIGTRAN triggers added complexity to an already complex TCP state machine, and did not allow any existing complexity to be removed.

There was also an issue that the TRIGTRAN signal was not sent in response to a specific host that had been sending packets, and was instead a signal that stimulated a response by any sender on the link. This needs to scale when there are multiple flows trying to use the same resource, yet the sender of a trigger has no understanding how many of the potential traffic sources will respond by sending packets - if recipients of the signal back-off their responses to a trigger to improve scaling, then that immediately mitigates the benefit of the signal.

Finally, intermediate forwarding nodes required modification to provide TRIGTRAN triggers, but operators couldn't charge for TRIGTRAN triggers, so there was no way to recover the cost of modifying, testing, and deploying updated intermediate nodes.

Two TRIGTRAN BOFs were held, at IETF 55 [TRIGTRAN-55] and IETF 56 [TRIGTRAN-56], but this work was not chartered, and there was no interest in deploying TRIGTRAN unless it was chartered and standardized in the IETF.

6.5.2. Lessons Learned.

The reasons why this work was not chartered, much less deployed, provide several useful lessons for researchers.

- * TRIGTRAN started with a plausible value proposition, but networking realities in the early 2000s forced reductions in scope that led directly to reductions in potential benefits, but no corresponding reductions in costs and complexity.
- * These reductions in scope were the direct result of an inability for hosts to trust or authenticate TRIGTRAN signals they received from the network.
- * Operators did not believe they could charge for TRIGTRAN signaling, because first-hop links didn't fail frequently, and TRIGTRAN provided no reduction in operating expenses, so there was little incentive to purchase and deploy TRIGTRAN-capable network equipment.

It is also worth noting that the targeted environment for TRIGTRAN in the late 1990s contained links with a relatively small number of directly-connected hosts - for instance, cellular or satellite links. The transport community was well aware of the dangers of sender synchronization based on multiple senders receiving the same stimulus at the same time, but the working assumption for TRIGTRAN was that there wouldn't be enough senders for this to be a meaningful problem. In the 2010s, it is common for a single "link" to support many senders and receivers on a single link, likely requiring TRIGTRAN senders to wait some random amount of time before sending after receiving a TRIGTRAN signal, which would have reduced the benefits of TRIGTRAN even more.

6.6. Shim6

The suggested references for Shim6 are:

- * Shim6: Level 3 Multihoming Shim Protocol for IPv6 [RFC5533]

The IPv6 routing architecture [RFC1887] assumed that most sites on the Internet would be identified by Provider Assigned IPv6 prefixes,

so that Default-Free Zone routers only contained routes to other providers, resulting in a very small IPv6 global routing table.

For a single-homed site, this could work well. A multihomed site with only one upstream provider could also work well, although BGP multihoming from a single upstream provider was often a premium service (costing more than twice as much as two single-homed sites), and if the single upstream provider went out of service, all of the multihomed paths could fail simultaneously.

IPv4 sites often multihomed by obtaining Provider Independent prefixes, and advertising these prefixes through multiple upstream providers. With the assumption that any multihomed IPv4 site would also multihome in IPv6, it seemed likely that IPv6 routing would be subject to the same pressures to announce Provider Independent prefixes, resulting in a global IPv6 routing table that exhibited the same explosive growth as the global IPv4 routing table. During the early 2000s, work began on a protocol that would provide multihoming for IPv6 sites without requiring sites to advertise Provider Independent prefixes into the IPv6 global routing table.

This protocol, called Shim6, allowed two endpoints to exchange multiple addresses ("Locators") that all mapped to the same endpoint ("Identity"). After an endpoint learned multiple Locators for the other endpoint, it could send to any of those Locators with the expectation that those packets would all be delivered to the endpoint with the same Identity. Shim6 was an example of an "Identity/Locator Split" protocol.

Shim6, as defined in [RFC5533] and related RFCs, provided a workable solution for IPv6 multihoming using Provider Assigned prefixes, including capability discovery and negotiation, and allowing end-to-end application communication to continue even in the face of path failure, because applications don't see Locator failures, and continue to communicate with the same Identity using a different Locator.

6.6.1. Reasons for Non-deployment

Note that the problem being addressed was "site multihoming", but Shim6 was providing "host multihoming". That meant that the decision about what path would be used was under host control, not under edge router control.

Although more work could have been done to provide a better technical solution, the biggest impediments to Shim6 deployment were operational and business considerations. These impediments were discussed at multiple network operator group meetings, including [Shim6-35] at [NANOG-35].

The technical issues centered around concerns that Shim6 relied on the host to track all the connections, while also tracking Identity/Locator mappings in the kernel, and tracking failures to recognize that an available path has failed.

The operational issues centered around concerns that operators were performing traffic engineering on traffic aggregates. With Shim6, these operator traffic engineering policies must be pushed down to individual hosts.

In addition, operators would have no visibility or control over the decision of hosts choosing to switch to another path. They expressed

concerns that relying on hosts to steer traffic exposed operator networks to oscillation based on feedback loops, if hosts moved from path to path frequently. Given that Shim6 was intended to support multihoming across operators, operators providing only one of the paths would have even less visibility as traffic suddenly appeared and disappeared on their networks.

In addition, firewalls that expected to find a TCP or UDP transport-level protocol header in the IP payload would see a Shim6 Identity header instead, and would not perform transport-protocol-based firewalling functions because the firewall's normal processing logic would not look past the Identity header.

The business issues centered on reducing or removing the ability to sell BGP multihoming service to their own customers, which is often more expensive than two single-homed connectivity services.

6.6.2. Lessons Learned

It is extremely important to take operational concerns into account when a path-aware protocol is making decisions about path selection that may conflict with existing operational practices and business considerations.

6.6.3. Addendum on MultiPath TCP

During discussions in the PANRG session at IETF 103 [PANRG-103-Min], Lars Eggert, past Transport Area Director, pointed out that during charter discussions for the Multipath TCP working group [MP-TCP], operators expressed concerns that customers could use Multipath TCP to loadshare TCP connections across operators simultaneously and compare passive performance measurements across network paths in real time, changing the balance of power in those business relationships. Although the Multipath TCP working group was chartered, this concern could have acted as an obstacle to deployment.

Operator objections to Shim6 were focused on technical concerns, but this concern could have also been an obstacle to Shim6 deployment if the technical concerns had been overcome.

6.7. Next Steps in Signaling (NSIS)

The suggested references for Next Steps in Signaling (NSIS) are:

- * the concluded working group charter [NSIS-CHARTER-2001]
- * GIST: General Internet Signalling Transport [RFC5971]
- * NAT/Firewall NSIS Signaling Layer Protocol (NSLP) [RFC5973]
- * NSIS Signaling Layer Protocol (NSLP) for Quality-of-Service Signaling [RFC5974]
- * Authorization for NSIS Signaling Layer Protocols [RFC5981]

The NSIS Working Group worked on signaling techniques for network layer resources (e.g., QoS resource reservations, Firewall and NAT traversal).

When RSVP [RFC2205] was used in deployments, a number of questions came up about its perceived limitations and potential missing features. The issues noted in the NSIS Working Group charter

[NSIS-CHARTER-2001] include interworking between domains with different QoS architectures, mobility and roaming for IP interfaces, and complexity. Later, the lack of security in RSVP was also recognized ([RFC4094]).

The NSIS Working Group was chartered to tackle those issues and initially focused on QoS signaling as its primary use case. However, over time a new approach evolved that introduced a modular architecture using application-specific signaling protocols (the NSIS Signaling Layer Protocol (NSLP)) on top of a generic signaling transport protocol (the NSIS Transport Layer Protocol (NTLP)).

The NTLP is defined in [RFC5971]. Two NSLPs are defined: the NSIS Signaling Layer Protocol (NSLP) for Quality-of-Service Signaling [RFC5974] as well as the NAT/Firewall NSIS Signaling Layer Protocol (NSLP) [RFC5973].

6.7.1. Reasons for Non-deployment

The obstacles for deployment can be grouped into implementation-related aspects and operational aspects.

* Implementation-related aspects:

Although NSIS provides benefits with respect to flexibility, mobility, and security compared to other network signaling techniques, hardware vendors were reluctant to deploy this solution, because it would require additional implementation effort and would result in additional complexity for router implementations.

The NTLP mainly operates as path-coupled signaling protocol, i.e., its messages are processed at the intermediate node's control plane that are also forwarding the data flows. This requires a mechanism to intercept signaling packets while they are forwarded in the same manner (especially along the same path) as data packets. NSIS uses the IPv4 and IPv6 Router Alert Option (RAO) to allow for interception of those path-coupled signaling messages, and this technique requires router implementations to correctly understand and implement the handling of RAOs, e.g., to only process packet with RAOs of interest and to leave packets with irrelevant RAOs in the fast forwarding processing path (a comprehensive discussion of these issues can be found in [RFC6398]). The latter was an issue with some router implementations at the time of standardization.

Another reason is that path-coupled signaling protocols that interact with routers and request manipulation of state at these routers (or any other network element in general) are under scrutiny: a packet (or sequence of packets) out of the mainly untrusted data path is requesting creation and manipulation of network state. This is seen as potentially dangerous (e.g., opens up a Denial of Service (DoS) threat to a router's control plane) and difficult for an operator to control. Path-coupled signaling approaches were considered problematic (see also section 3 of [RFC6398]). There are recommendations on how to secure NSIS nodes and deployments (e.g., [RFC5981]).

* Operational Aspects:

NSIS not only required trust between customers and their provider, but also among different providers. Especially, QoS signaling techniques would require some kind of dynamic service level agreement support that would imply (potentially quite complex) bilateral

negotiations between different Internet service providers. This complexity was not considered to be justified and increasing the bandwidth (and thus avoiding bottlenecks) was cheaper than actively managing network resource bottlenecks by using path-coupled QoS signaling techniques. Furthermore, an end-to-end path typically involves several provider domains and these providers need to closely cooperate in cases of failures.

6.7.2. Lessons Learned

One goal of NSIS was to decrease the complexity of the signaling protocol, but a path-coupled signaling protocol comes with the intrinsic complexity of IP-based networks, beyond the complexity of the signaling protocol itself. Sources of intrinsic complexity include:

- * the presence of asymmetric routes between endpoints and routers
- * the lack of security and trust at large in the Internet infrastructure
- * the presence of different trust boundaries
- * the effects of best-effort networks (e.g., robustness to packet loss)
- * divergence from the fate sharing principle (e.g., state within the network).

Any path-coupled signaling protocol has to deal with these realities.

Operators view the use of IPv4 and IPv6 Router Alert Option (RAO) to signal routers along the path from end systems with suspicion, because these end systems are usually not authenticated and heavy use of RAOs can easily increase the CPU load on routers that are designed to process most packets using a hardware "fast path" and diverting packets containing RAO to a slower, more capable processor.

6.8. IPv6 Flow Label

The suggested references for IPv6 Flow Label are:

- * IPv6 Flow Label Specification [RFC6437]

IPv6 specifies a 20-bit field Flow Label field [RFC6437], included in the fixed part of the IPv6 header and hence present in every IPv6 packet. An endpoint sets the value in this field to one of a set of pseudo-randomly assigned values. If a packet is not part of any flow, the flow label value is set to zero [RFC3697]. A number of Standards Track and Best Current Practice RFCs (e.g., [RFC8085], [RFC6437], [RFC6438]) encourage IPv6 endpoints to set a non-zero value in this field. A multiplexing transport could choose to use multiple flow labels to allow the network to independently forward its subflows, or to use one common value for the traffic aggregate. The flow label is present in all fragments. IPsec was originally put forward as one important use-case for this mechanism and does encrypt the field [RFC6438].

Once set, the flow label can provide information that can help inform network nodes about subflows present at the transport layer, without needing to interpret the setting of upper layer protocol fields [RFC6294]. This information can also be used to coordinate how

aggregates of transport subflows are grouped when queued in the network and to select appropriate per-flow forwarding when choosing between alternate paths [RFC6438] (e.g. for Equal Cost Multipath Routing (ECMP) and Link Aggregation (LAG)).

6.8.1. Reasons for Non-deployment

Despite the field being present in every IPv6 packet, the mechanism did not receive as much use as originally envisioned. One reason is that to be useful it requires engagement by two different stakeholders:

* Endpoint Implementation:

For network nodes along a path to utilize the flow label there needs to be a non-zero value inserted in the field [RFC6437] at the sending endpoint. There needs to be an incentive for an endpoint to set an appropriate non-zero value. The value should appropriately reflect the level of aggregation the traffic expects to be provided by the network. However, this requires the stack to know granularity at which flows should be identified (or conversely which flows should receive aggregated treatment), i.e., which packets carry the same flow label. Therefore, setting a non-zero value may result in additional choices that need to be made by an application developer.

Although the standard [RFC3697] forbids any encoding of meaning into the flow label value, the opportunity to use the flow label as a covert channel or to signal other meta-information may have raised concerns about setting a non-zero value [RFC6437].

Before methods are widely deployed to use this method, there could be no incentive for an endpoint to set the field.

* Operational support in network nodes:

A benefit can only be realized when a network node along the path also uses this information to inform its decisions. Network equipment (routers and/or middleboxes) need to include appropriate support so they can utilize the field when making decisions about how to classify flows, or to inform forwarding choices. Use of any optional feature in a network node also requires corresponding updates to operational procedures, and therefore is normally only introduced when the cost can be justified.

A benefit from utilizing the flow label is expected to be increased quality of experience for applications - but this comes at some operational cost to an operator, and requires endpoints to set the field.

6.8.2. Lessons Learned

The flow label is a general purpose header field for use by the path. Multiple uses have been proposed. One candidate use was to reduce the complexity of forwarding decisions. However, modern routers can use a "fast path", often taking advantage of hardware to accelerate processing. The method can assist in more complex forwarding, such as ECMP and load balancing.

Although [RFC6437] recommended that endpoints should by default choose uniformly-distributed labels for their traffic, the specification permitted an endpoint to choose to set a zero value. This ability of endpoints to choose to set a flow label of zero has

had consequences on deployability:

- * Before wide-scale support by endpoints, it would be impossible to rely on a non-zero flow label being set. Network nodes therefore would need to also employ other techniques to realize equivalent functions. An example of a method is one assuming semantics of the source port field to provide entropy input to a network-layer hash. This use of a 5-tuple to classify a packet represents a layering violation [RFC6294]. When other methods have been deployed, they increase the cost of deploying standards-based methods, even though they may offer less control to endpoints and result in potential interaction with other uses/interpretation of the field.
- * Even though the flow label is specified as an end-to-end field, some network paths have been observed to not transparently forward the flow label. This could result from non-conformant equipment, or could indicate that some operational networks have chosen to re-use the protocol field for other (e.g. internal purposes). This results in lack of transparency, and a deployment hurdle to endpoints expecting that they can set a flow label that is utilized by the network. The more recent practice of "greasing" [GREASE] would suggest that a different outcome could have been achieved if endpoints were always required to set a non-zero value.
- * [RFC1809] noted that setting the choice of the flow label value can depend on the expectations of the traffic generated by an application, which suggests an API should be presented to control the setting or policy that is used. However, many currently available APIs do not have this support.

A growth in the use of encrypted transports, (e.g. QUIC [QUIC-WG]) seems likely to raise similar issues to those discussed above and could motivate renewed interest in utilizing the flow label.

6.9. Explicit Congestion Notification (ECN)

The suggested references for Explicit Congestion Notification (ECN) are:

- * Recommendations on Queue Management and Congestion Avoidance in the Internet [RFC2309]
- * A Proposal to add Explicit Congestion Notification (ECN) to IP [RFC2481]
- * The Addition of Explicit Congestion Notification (ECN) to IP [RFC3168]
- * Implementation Report on Experiences with Various TCP RFCs [vista-impl], slides 6 and 7
- * Implementation and Deployment of ECN [SallyFloyd]

In the early 1990s, the large majority of Internet traffic used TCP as its transport protocol, but TCP had no way to detect path congestion before the path was so congested that packets were being dropped, and these congestion events could affect all senders using a path, either by "lockout", where long-lived flows monopolized the queues along a path, or by "full queues", where queues remain full, or almost full, for a long period of time.

In response to this situation, "Active Queue Management" (AQM) was deployed in the network. A number of AQM disciplines have been deployed, but one common approach was that routers dropped packets when a threshold buffer length was reached, so that transport protocols like TCP that were responsive to loss would detect this loss and reduce their sending rates. Random Early Detection (RED) was one such proposal in the IETF. As the name suggests, a router using RED as its AQM discipline that detected time-averaged queue lengths passing a threshold would choose incoming packets probabilistically to be dropped [RFC2309]. In response to this situation, "Active Queue Management" (AQM) was deployed in the network. A number of AQM disciplines have been deployed, but one common approach was that routers dropped packets when a threshold buffer length was reached, so that transport protocols like TCP that were responsive to loss would detect this loss and reduce their sending rates. Random Early Detection (RED) was one such proposal in the IETF. As the name suggests, a router using RED as its AQM discipline that detected time-averaged queue lengths passing a threshold would choose incoming packets probabilistically to be dropped [RFC2309].

Researchers suggested that providing "explicit congestion notifications" to senders when routers along the path detected their queues were building, so that some senders would "slow down" as if a loss had occurred, so that the path queues had time to drain, and the path still had sufficient buffer capacity to accommodate bursty arrivals of packets from other senders. This was proposed as an Experiment in [RFC2481], and standardized in [RFC3168].

A key aspect of ECN was the use of IP header fields rather than IP options to carry explicit congestion notifications, since the proponents recognized that

Many routers process the "regular" headers in IP packets more efficiently than they process the header information in IP options.

Unlike most of the Path Aware technologies included in this document, the story of ECN continues to the present day, and encountered a large number of Lessons Learned during that time. The early history of ECN (non-)deployment provides Lessons Learned that were not captured by other contributions in Section 6, so that is the emphasis in this section of the document.

6.9.1. Reasons for Non-deployment

There are at least three sub-stories - ECN deployment in clients, ECN deployment in routers, and AQM deployment in operational networks. All three sub-stories mattered.

The proponents of ECN did so much right, anticipating many of the Lessons Learned now recognized in Section 4. They recognized the need to support incremental deployment (Section 4.2). They considered the impact on router throughput (Section 4.8). They even considered trust issues between end nodes and the network, both for non-compliant end nodes (Section 4.10) and non-compliant routers (Section 4.9).

They were rewarded with ECN being implemented in major operating systems, both for end nodes and for routers. A number of implementations are listed under "Implementation and Deployment of

ECN" at [SallyFloyd].

What they did not anticipate, was routers that would crash, when they saw bits 6 and 7 in the IPv4 TOS octet [RFC0791]/IPv6 Traffic Class field [RFC2460], which [RFC2481] redefined to be "currently unused", being set to a non-zero value.

As described in [vista-impl],

Intermediate Gateway Device problem #1: one of the most popular versions from one of the most popular vendors. When a data packet arrives with either ECT(0) or ECT(1) (indicating successful ECN capability negotiation) indicated, router crashed. Cannot be recovered at TCP layer (sic)

This implementation, which would be run on a significant percentage of Internet end nodes, was shipped with ECN disabled, as was true for several of the other implementations listed under "Implementation and Deployment of ECN" at [SallyFloyd]. Even if subsequent router vendors fixed these implementations, ECN was still disabled on end nodes, and given the tradeoff between the benefits of enabling ECN (somewhat better behavior during congestion) and the risks of enabling ECN (possibly crashing a router somewhere along the path), ECN tended to stay disabled on implementations that supported ECN for decades afterwards.

6.9.2. Lessons Learned

Of the contributions included in Section 6, ECN may be unique in providing these lessons:

- * Even if you do everything right, you may trip over implementation bugs in devices you know nothing about, that will cause severe problems that prevent successful deployment of your path aware technology.
- * After implementations disable your Path Aware technology, it may take years, or even decades, to convince implementers to re-enable it by default.

These two lessons, taken together, could be summarized as "you get one chance to get it right".

During discussion of ECN at [PANRG-110], we noted that "you get one chance to get it right" isn't quite correct today, because operating systems on so many host systems are frequently updated, and transport protocols like QUIC [I-D.ietf-quic-transport] are being implemented in user space, and can be updated without touching installed operating systems. Neither of these factors were true in the early 2000s.

We think that these restatements of the ECN Lessons Learned are more useful for current implementers:

- * Even if you do everything right, you may trip over implementation bugs in devices you know nothing about, that will cause severe problems that prevent successful deployment of your path aware technology. Testing before deployment isn't enough to ensure successful deployment. It is also necessary to "deploy gently", which often means deploying for a small subset of users to gain experience, and implementing feedback mechanisms to detect that user experience is being degraded.

- * After implementations disable your Path Aware technology, it may take years, or even decades, to convince implementers to re-enable it by default. This might be based on the difficulty of distributing implementations that enable it by default, but are just as likely to be based on the "bad taste in the mouth" that implementers have after an unsuccessful deployment attempt that degraded user experience.

With these expansions, the two lessons, taken together, could be more helpfully summarized as "plan for failure" - anticipate what your next step will be, if initial deployment is unsuccessful.

ECN deployment was also hindered by non-deployment of AQM in many devices, because of operator interest in QoS features provided in the network, rather than using the network to assist end systems in providing for themselves. But that's another story, and the AQM Lessons Learned are already covered in other contributions in Section 6.

7. Security Considerations

This document describes Path Aware techniques that were not adopted and widely deployed on the Internet, so it doesn't affect the security of the Internet.

If this document meets its goals, we may develop new techniques for Path Aware Networking that would affect the security of the Internet, but security considerations for those techniques will be described in the corresponding RFCs that specify them.

8. IANA Considerations

This document makes no requests of IANA.

9. Acknowledgments

Initial material for Section 6.1 on ST2 was provided by Gorrry Fairhurst.

Initial material for Section 6.2 on IntServ was provided by Ron Bonica.

Initial material for Section 6.3 on Quick-Start TCP was provided by Michael Scharf, who also provided suggestions to improve this section after it was edited.

Initial material for Section 6.4 on ICMP Source Quench was provided by Gorrry Fairhurst.

Initial material for Section 6.5 on Triggers for Transport (TRIGTRAN) was provided by Spencer Dawkins.

Section 6.6 on Shim6 builds on initial material describing obstacles provided by Erik Nordmark, with background added by Spencer Dawkins.

Initial material for Section 6.7 on Next Steps In Signaling (NSIS) was provided by Roland Bless and Martin Stiernerling.

Initial material for Section 6.8 on IPv6 Flow Labels was provided by Gorrry Fairhurst.

Initial material for Section 6.9 on Explicit Congestion Notification was provided by Spencer Dawkins.

Our thanks to Adrian Farrel, Bob Briscoe, C.M. Heard, David Black, Eric Kinnear, Erik Auerswald, Gorry Fairhurst, Jake Holland, Joe Touch, Joeri de Ruiter, Kireeti Kompella, Mohamed Boucadair, Roland Bless, Ruediger Geib, Theresa Enhardt, and Wes Eddy, who provided review comments on this document as a "work in process".

Mallory Knodel reviewed this document for the Internet Research Steering Group, and provided many helpful suggestions.

David Oran also provided helpful comments and text suggestions on this document during Internet Research Steering Group balloting. In particular, Section 5 reflects his review.

Benjamin Kaduk and Rob Wilton provided helpful comments during Internet Engineering Steering Group conflict review.

Special thanks to Adrian Farrel for helping Spencer navigate the twisty little passages of Flow Specs and Filter Specs in IntServ, RSVP, MPLS, and BGP. They are all alike, except when they are different [Colossal-Cave].

10. Informative References

[Colossal-Cave]

"Wikipedia Page for Colossal Cave Adventure", January 2019,
<https://en.wikipedia.org/wiki/Colossal_Cave_Adventure>.

[Conviva]

"Conviva Precision : Data Sheet", December 2020,
<<https://www.conviva.com/datasheets/precision-delivery-intelligence/>>.

[GREASE]

Thomson, M., "Long-term Viability of Protocol Extension Mechanisms", July 2019, <<https://tools.ietf.org/html/draft-iab-use-it-or-lose-it-00>>.

[I-D.arkko-arch-internet-threat-model]

Arkko, J., "Changes in the Internet Threat Model", Work in Progress, Internet-Draft, draft-arkko-arch-internet-threat-model-01, 8 July 2019, <<http://www.ietf.org/internet-drafts/draft-arkko-arch-internet-threat-model-01.txt>>.

[I-D.farrell-etm]

Farrell, S., "We're gonna need a bigger threat model", Work in Progress, Internet-Draft, draft-farrell-etm-03, 6 July 2019, <<http://www.ietf.org/internet-drafts/draft-farrell-etm-03.txt>>.

[I-D.ietf-quic-transport]

Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", Work in Progress, Internet-Draft, draft-ietf-quic-transport-34, 14 January 2021, <<http://www.ietf.org/internet-drafts/draft-ietf-quic-transport-34.txt>>.

[I-D.ietf-tsvwg-intserv-multiple-tspec]

Polk, J. and S. Dhesikan, "Integrated Services (IntServ) Extension to Allow Signaling of Multiple Traffic

Specifications and Multiple Flow Specifications in RSVIPv1", Work in Progress, Internet-Draft, draft-ietf-tsvwg-intserv-multiple-tspec-02, 25 February 2013, <<http://www.ietf.org/internet-drafts/draft-ietf-tsvwg-intserv-multiple-tspec-02.txt>>.

[I-D.irtf-panrg-path-properties]

Enghardt, T. and C. Krahenbuhl, "A Vocabulary of Path Properties", Work in Progress, Internet-Draft, draft-irtf-panrg-path-properties-01, 7 September 2020, <<http://www.ietf.org/internet-drafts/draft-irtf-panrg-path-properties-01.txt>>.

[I-D.irtf-panrg-questions]

Trammell, B., "Current Open Questions in Path Aware Networking", Work in Progress, Internet-Draft, draft-irtf-panrg-questions-08, 23 December 2020, <<http://www.ietf.org/internet-drafts/draft-irtf-panrg-questions-08.txt>>.

[IEN-119] Forgie, J., "ST - A Proposed Internet Stream Protocol", September 1979, <<https://www.rfc-editor.org/ien/ien119.txt>>.

[ITAT] "IAB Workshop on Internet Technology Adoption and Transition (ITAT)", December 2013, <<https://www.iab.org/activities/workshops/itat/>>.

[model-t] "Model-t -- Discussions of changes in Internet deployment patterns and their impact on the Internet threat model", n.d., <<https://www.iab.org/mailman/listinfo/model-t>>.

[MOPS-109-Min]

"Media Operations Working Group - IETF-109 Minutes", November 2020, <<https://datatracker.ietf.org/meeting/109/materials/minutes-109-mops-00>>.

[MP-TCP] "Multipath TCP Working Group Home Page", n.d., <<https://datatracker.ietf.org/wg/mptcp/about/>>.

[NANOG-35] "North American Network Operators Group NANOG-35 Agenda", October 2005, <<https://www.nanog.org/meetings/nanog35/agenda>>.

[NSIS-CHARTER-2001]

"Next Steps In Signaling Working Group Charter", March 2011, <<https://datatracker.ietf.org/doc/charter-ietf-nsis/>>.

[PANRG] "Path Aware Networking Research Group (Home Page)", n.d., <<https://irtf.org/panrg>>.

[PANRG-103-Min]

"Path Aware Networking Research Group - IETF-103 Minutes", November 2018, <<https://datatracker.ietf.org/doc/minutes-103-panrg/>>.

[PANRG-105-Min]

"Path Aware Networking Research Group - IETF-105 Minutes", July 2019, <<https://datatracker.ietf.org/doc/minutes-105-panrg/>>.

- [PANRG-106-Min] "Path Aware Networking Research Group - IETF-106 Minutes", November 2019,
<<https://datatracker.ietf.org/doc/minutes-106-panrg/>>.
- [PANRG-110] "Path Aware Networking Research Group - IETF-110", July 2017,
<<https://datatracker.ietf.org/meeting/110/sessions/panrg>>.
- [PANRG-99] "Path Aware Networking Research Group - IETF-99", July 2017,
<<https://datatracker.ietf.org/meeting/99/sessions/panrg>>.
- [PATH-Decade] Bonaventure, O., "A Decade of Path Awareness", July 2017,
<<https://datatracker.ietf.org/doc/slides-99-panrg-a-decade-of-path-awareness/>>.
- [QS-SAT] Secchi, R., Sathiaselalan, A., Potorti, F., Gotta, A., and G. Fairhurst, "Using Quick-Start to enhance TCP-friendly rate control performance in bidirectional satellite networks", 2009,
<<https://dl.acm.org/citation.cfm?id=3160304.3160305>>.
- [QUIC-WG] "QUIC Working Group Home Page", n.d.,
<<https://datatracker.ietf.org/wg/quic/about/>>.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981,
<<https://www.rfc-editor.org/info/rfc791>>.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981,
<<https://www.rfc-editor.org/info/rfc792>>.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981,
<<https://www.rfc-editor.org/info/rfc793>>.
- [RFC1016] Prue, W. and J. Postel, "Something a Host Could Do with Source Quench: The Source Quench Introduced Delay (SQuID)", RFC 1016, DOI 10.17487/RFC1016, July 1987,
<<https://www.rfc-editor.org/info/rfc1016>>.
- [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>>.
- [RFC1190] Topolcic, C., "Experimental Internet Stream Protocol: Version 2 (ST-II)", RFC 1190, DOI 10.17487/RFC1190, October 1990, <<https://www.rfc-editor.org/info/rfc1190>>.
- [RFC1633] Braden, R., Clark, D., and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, DOI 10.17487/RFC1633, June 1994,
<<https://www.rfc-editor.org/info/rfc1633>>.
- [RFC1809] Partridge, C., "Using the Flow Label Field in IPv6", RFC 1809, DOI 10.17487/RFC1809, June 1995,

<<https://www.rfc-editor.org/info/rfc1809>>.

- [RFC1819] Delgrossi, L., Ed. and L. Berger, Ed., "Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+", RFC 1819, DOI 10.17487/RFC1819, August 1995, <<https://www.rfc-editor.org/info/rfc1819>>.
- [RFC1887] Rekhter, Y., Ed. and T. Li, Ed., "An Architecture for IPv6 Unicast Address Allocation", RFC 1887, DOI 10.17487/RFC1887, December 1995, <<https://www.rfc-editor.org/info/rfc1887>>.
- [RFC2001] Stevens, W., "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", RFC 2001, DOI 10.17487/RFC2001, January 1997, <<https://www.rfc-editor.org/info/rfc2001>>.
- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<https://www.rfc-editor.org/info/rfc2205>>.
- [RFC2210] Wroclawski, J., "The Use of RSVP with IETF Integrated Services", RFC 2210, DOI 10.17487/RFC2210, September 1997, <<https://www.rfc-editor.org/info/rfc2210>>.
- [RFC2211] Wroclawski, J., "Specification of the Controlled-Load Network Element Service", RFC 2211, DOI 10.17487/RFC2211, September 1997, <<https://www.rfc-editor.org/info/rfc2211>>.
- [RFC2212] Shenker, S., Partridge, C., and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, DOI 10.17487/RFC2212, September 1997, <<https://www.rfc-editor.org/info/rfc2212>>.
- [RFC2215] Shenker, S. and J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements", RFC 2215, DOI 10.17487/RFC2215, September 1997, <<https://www.rfc-editor.org/info/rfc2215>>.
- [RFC2309] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, DOI 10.17487/RFC2309, April 1998, <<https://www.rfc-editor.org/info/rfc2309>>.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<https://www.rfc-editor.org/info/rfc2460>>.
- [RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, DOI 10.17487/RFC2475, December 1998, <<https://www.rfc-editor.org/info/rfc2475>>.
- [RFC2481] Ramakrishnan, K. and S. Floyd, "A Proposal to add Explicit Congestion Notification (ECN) to IP", RFC 2481, DOI 10.17487/RFC2481, January 1999, <<https://www.rfc-editor.org/info/rfc2481>>.

- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<https://www.rfc-editor.org/info/rfc3168>>.
- [RFC3697] Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, "IPv6 Flow Label Specification", RFC 3697, DOI 10.17487/RFC3697, March 2004, <<https://www.rfc-editor.org/info/rfc3697>>.
- [RFC4094] Manner, J. and X. Fu, "Analysis of Existing Quality-of-Service Signaling Protocols", RFC 4094, DOI 10.17487/RFC4094, May 2005, <<https://www.rfc-editor.org/info/rfc4094>>.
- [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>>.
- [RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-Start for TCP and IP", RFC 4782, DOI 10.17487/RFC4782, January 2007, <<https://www.rfc-editor.org/info/rfc4782>>.
- [RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, DOI 10.17487/RFC5082, October 2007, <<https://www.rfc-editor.org/info/rfc5082>>.
- [RFC5218] Thaler, D. and B. Aboba, "What Makes for a Successful Protocol?", RFC 5218, DOI 10.17487/RFC5218, July 2008, <<https://www.rfc-editor.org/info/rfc5218>>.
- [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>>.
- [RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <<https://www.rfc-editor.org/info/rfc5575>>.
- [RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <<https://www.rfc-editor.org/info/rfc5681>>.
- [RFC5971] Schulzrinne, H. and R. Hancock, "GIST: General Internet Signalling Transport", RFC 5971, DOI 10.17487/RFC5971, October 2010, <<https://www.rfc-editor.org/info/rfc5971>>.
- [RFC5973] Stiemerling, M., Tschofenig, H., Aoun, C., and E. Davies, "NAT/Firewall NSIS Signaling Layer Protocol (NSLP)", RFC 5973, DOI 10.17487/RFC5973, October 2010, <<https://www.rfc-editor.org/info/rfc5973>>.
- [RFC5974] Manner, J., Karagiannis, G., and A. McDonald, "NSIS Signaling Layer Protocol (NSLP) for Quality-of-Service Signaling", RFC 5974, DOI 10.17487/RFC5974, October 2010, <<https://www.rfc-editor.org/info/rfc5974>>.
- [RFC5981] Manner, J., Stiemerling, M., Tschofenig, H., and R. Bless,

- Ed., "Authorization for NSIS Signaling Layer Protocols", RFC 5981, DOI 10.17487/RFC5981, February 2011, <<https://www.rfc-editor.org/info/rfc5981>>.
- [RFC6294] Hu, Q. and B. Carpenter, "Survey of Proposed Use Cases for the IPv6 Flow Label", RFC 6294, DOI 10.17487/RFC6294, June 2011, <<https://www.rfc-editor.org/info/rfc6294>>.
- [RFC6398] Le Faucheur, F., Ed., "IP Router Alert Considerations and Usage", BCP 168, RFC 6398, DOI 10.17487/RFC6398, October 2011, <<https://www.rfc-editor.org/info/rfc6398>>.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <<https://www.rfc-editor.org/info/rfc6437>>.
- [RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <<https://www.rfc-editor.org/info/rfc6438>>.
- [RFC6633] Gont, F., "Deprecation of ICMP Source Quench Messages", RFC 6633, DOI 10.17487/RFC6633, May 2012, <<https://www.rfc-editor.org/info/rfc6633>>.
- [RFC6928] Chu, J., Dukkkipati, N., Cheng, Y., and M. Mathis, "Increasing TCP's Initial Window", RFC 6928, DOI 10.17487/RFC6928, April 2013, <<https://www.rfc-editor.org/info/rfc6928>>.
- [RFC7305] Lear, E., Ed., "Report from the IAB Workshop on Internet Technology Adoption and Transition (ITAT)", RFC 7305, DOI 10.17487/RFC7305, July 2014, <<https://www.rfc-editor.org/info/rfc7305>>.
- [RFC7418] Dawkins, S., Ed., "An IRTF Primer for IETF Participants", RFC 7418, DOI 10.17487/RFC7418, December 2014, <<https://www.rfc-editor.org/info/rfc7418>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.
- [RFC8170] Thaler, D., Ed., "Planning for Protocol Adoption and Subsequent Transitions", RFC 8170, DOI 10.17487/RFC8170, May 2017, <<https://www.rfc-editor.org/info/rfc8170>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.
- [RFC8793] Wissingh, B., Wood, C., Afanasyev, A., Zhang, L., Oran, D., and C. Tschudin, "Information-Centric Networking (ICN): Content-Centric Networking (CCNx) and Named Data Networking (NDN) Terminology", RFC 8793, DOI 10.17487/RFC8793, June 2020, <<https://www.rfc-editor.org/info/rfc8793>>.
- [SAAG-105-Min] "Security Area Open Meeting - IETF-105 Minutes", July

2019, <<https://datatracker.ietf.org/meeting/105/materials/minutes-105-saag-00>>.

[SAF07] Sarolahti, P., Allman, M., and S. Floyd, "Determining an appropriate sending rate over an underutilized network path", Computer Networking Volume 51, Number 7, May 2007.

[SallyFloyd] Floyd, S., "ECN (Explicit Congestion Notification) in TCP/IP", n.d., <<https://www.icir.org/floyd/ecn.html>>.

[Sch11] Scharf, M., "Fast Startup Internet Congestion Control for Broadband Interactive Applications", Ph.D. Thesis, University of Stuttgart, April 2011.

[Shim6-35] Meyer, D., Huston, G., Schiller, J., and V. Gill, "IAB IPv6 Multihoming Panel at NANOG 35", NANOG North American Network Operator Group, October 2005, <https://www.youtube.com/watch?v=ji6Y_rYHAQs>.

[TRIGTRAN-55] "Triggers for Transport BOF at IETF 55", July 2003, <<https://www.ietf.org/proceedings/55/239.htm>>.

[TRIGTRAN-56] "Triggers for Transport BOF at IETF 56", November 2003, <<https://www.ietf.org/proceedings/56/251.htm>>.

[vista-impl] Sridharan, M., Bansal, D., and D. Thaler, "Implementation Report on Experiences with Various TCP RFCs", November 2003, <<https://www.ietf.org/proceedings/68/slides/tsvarea-3/sld1.htm>>.

Author's Address

Spencer Dawkins (editor)
Tencent America
United States of America

Email: spencerdawkins.ietf@gmail.com