PANRG S. Dawkins, Ed.
Internet-Draft Huawei Technologies
Intended status: Informational June 18, 2018
Expires: December 20, 2018

Path Aware Networking: A Bestiary of Roads Not Taken
draft-dawkins-panrg-what-not-to-do-01

Abstract

At the first meeting of the proposed Path Aware Networking Research Group, Oliver Bonaventure led a discussion of our mostly-unsuccessful attempts to exploit Path Awareness to achieve a variety of goals, over the past decade. At the end of that discussion, the research group agreed to catalog and analyze these ideas, to extract insights and lessons for path-aware networking researchers.

This document contains that catalog and analysis.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

At IETF 99, the proposed Path Aware Networking Research Group [PANRG] held its first meeting [PANRG-99], and the first presentation in that session was "A Decade of Path Awareness" [PATH-Decade]. At the end of this discussion, two things were abundantly clear.

- The Internet community has accumulated considerable experience with many Path Awareness ideas over a long period of time, and

- Although some Path Awareness ideas have been successfully deployed (for example, Differentiated Services, or DiffServ [RFC2475]), most of these ideas haven’t seen widespread adoption. The reasons for this non-adoption are many, and are worthy of study.

The meta-lessons from this experience are

- Path Aware Networking is more Research than Engineering, so establishing an IRTF Research Group for Path Aware Networking is the right thing to do [RFC7418], and

- Cataloging and analyzing our experience to learn the reasons for non-adoption is a great first step for the proposed Research Group.

This document contains that catalog and analysis.

1.1. About this Document

This document is not intended to include every idea about Path Aware Networking that we can find. Instead, we include enough ideas to provide background for new lessons to guide researchers in their work, in order to add those lessons to Section 2.

1.2. A Note for Contributors (Consider removing after approval)

There is no shame to having your idea included in this document. When these proposals were made, we were trying to engineer something that was research. The document editor started with a subsection on his own idea. The only shame is not learning from experience, and not sharing that experience with other networking researchers and engineers.

This document is being built collaboratively. To contribute your experience, please send a Github pull request to https://github.com/panrg/draft-dawkins-panrg-what-not-to-do.

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

1.3. A Note for the Editor (Remove after taking these actions)

The to-do list for upcoming revisions includes

- Rearrange the Summary of Lessons Learned so that it flows (the current revision is more or less in the order of contributions).

- Tag the Lessons Learned so that they are tied to one or more specific contributions.

1.4. 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 on successful protocol adoption and deployment.

2. Summary of Lessons Learned

This section summarizes the Lessons Learned from the contributed sections in Section 4.

- The benefit of Path Awareness has to be great enough to overcome entropy for already-deployed devices. The colloquial American English expression, "If it ain’t broke, don’t fix it" is in full flower on today’s Internet.

- If intermediate devices along the path can’t be trusted, it’s difficult to rely on intermediate devices to drive changes to endpoint behaviors.

- If operators can’t charge for a Path Aware technology in order to recover the costs of deploying it, the benefits must be really significant.

- Impact of a Path Aware technology on operational practices can prevent deployment of promising technology.

- Per-connection state in intermediate devices is an impediment to adoption and deployment.

- Providing benefits for early adopters is key - if everyone must deploy a technology in order for the topology to provide benefits, or even to work at all, the technology is unlikely to be adopted.

- The Internet is a distributed system, so the more a technology relies on information propagated from distant hosts and routers, the less likely that information is to be accurate.

- Transport protocol technologies may require information from applications, in order to work effectively, but applications may not know the information they need to provide.

3. Template for Contributions

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

- the name of a technology, including an abbreviation if one was used
- if available, a long-term pointer to the best reference describing the technology
- a short description of the problem the technology was intended to solve
- a short description of the reasons why the technology wasn’t adopted
- a short statement of the lessons that researchers can learn from our experience with this technology.

4. Contributions

The editor has added some suggested subsections as a starting place, but others are solicited and welcome.

4.1. 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 a scarce commodity. Internet Service Providers built networks over DS3 (45 Mbps) infrastructure, and sub-rate (< 1 Mpbs) access was common. Therefore, the IETF anticipated a need for a fine-grained QoS mechanism.

In the IntServ architecture, some applications require service guarantees. Therefore, those applications use the Resource Reservation Protocol (RSVP) [RFC2205] to signal bandwidth reservations across the network. Every router in the network maintains per-flow state in order 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 bandwidth for traffic on a per Flow Spec basis.

4.1.1. Reasons for Non-deployment

IntServ was never widely deployed because of its cost. The following factors contributed to cost:

- IntServ must be deployed on every router within the QoS domain

- IntServ maintained per flow state

As IntServ was being discussed, the following occurred:

- 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)

- DiffServ [RFC2475] offered a more cost-effective, albeit less fine-grained, solution to the QoS problem.

4.1.2. Lessons Learned.

The following lessons were learned:

- Any mechanism that requires a router to maintain state is not likely to succeed.

- Any mechanism that requires an operator to upgrade all of its routers is not likely to succeed.

IntServ was never widely deployed. However, the technology that it produced was deployed for reasons other than bandwidth management. RSVP is widely deployed as an MPLS signaling mechanism. BGP uses Flow Specs to distribute firewall filters.

4.2. Quick-Start TCP

Quick-Start [RFC4782] is an experimental TCP extension that leverages support from the routers on the path to determine an allowed sending rate, either at the start of data transfers or after idle periods. In these cases, a TCP sender cannot easily determine an appropriate sending rate, given the lack of information about the path. The default TCP congestion control therefore uses the time-consuming slow-start algorithm. With Quick-Start, connections are allowed to use higher 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 Time To Live (TTL) fields, a sender can determine if all routers have approved the Quick-Start request. 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 has been being proposed in [RFC4782] for controlled environments such as intranets only.

The Quick-Start protocol 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.

4.2.1. Reasons for Non-deployment

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

- Having information from the routers along the path can reduce the risk of congestion, but it cannot avoid it entirely. Determining whether there is unused capacity is not trivial in actual router and host implementations. Data about available bandwidth visible at the IP layer may be imprecise, and due to the propagation delay, information can already be outdated when it reaches the sender. There is a trade-off between the speedup of data transfers and the risk of congestion even with Quick-Start.

- 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 different packets, which should be avoided as much as possible.

- Only selected applications can benefit from Quick-Start. For achieving an overall benefit, it is important that senders avoid sending unnecessary Quick-Start requests, e.g. for connections that will only send a small amount of data. This typically requires application-internal knowledge. It is a mostly unsolved question how a sender can indeed determine the data rate that Quick-Start shall request for.

After completion 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 TCP congestion control, but it only requires sender-side TCP modifications. As a result, this approach can be easier and incrementally deployed in the Internet. While theoretically Quick-Start can outperform "IW10", the absolute improvement of data transfer times is rather small in many cases. After publication of [RFC6928], most modern TCP stacks have increased their default initial window. There is no known deployment of Quick-Start TCP.

4.2.2. Lessons Learned

There are some lessons 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 TCP stacks there is not much room for significant improvement by additional network support.

4.3. Triggers for Transport (TRIGTRAN)

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 semi-modern TCPs (since the late 1980s) have interpreted the loss of a segment as evidence that the path between two endpoints has 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 [RFC2581]. More modern TCPs have added a growing array of strategies about how to establish the sending rate [RFC5681], but when a path is no longer operational, TCPs can wait many seconds before retrying a segment, even if the path becomes operational while the sender is waiting to retry.

The thinking in Triggers for Transport was that if a path completely stopped working because its first-hop link was "down", that somehow TCP could be signaled when the first-hop link returned to service, and the sending TCP could retry immediately, without waiting for a full Retransmission Time Out (RTO).

4.3.1. Reasons for Non-deployment

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 in the IETF.

4.3.2. Lessons Learned.

The reasons why this work was not chartered provide several useful lessons for researchers.

- TRIGTRAN triggers are only provided when the first-hop link is "down", so 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 didn't allow any existing complexity to be removed.

- The state of the art in the early 2000s was that TRIGTRAN triggers were assumed to be unauthenticated, so they couldn't be trusted to tell a sender to "speed up", only to "slow down". This reduced the potential benefit to implementers.

- intermediate forwarding devices 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 devices.

4.4. Shim6

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 routing table.

For a single-homed site, this could work well. A multi-homed 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 multi-homed 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 problems as the global IPv4 routing table. During the early 2000s, work began on a protocol that would provide the same benefits for multihomed IPv6 sites without requiring sites to advertise Provider Independent prefixes into the 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.

4.4.1. 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 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 technology issues centered around scaling concerns that Shim6 relied on the host to track all the TCP connections and the file descriptions with associated HTTP state, while also tracking Identity/Locator mappings in the kernel, and tracking failures to recognize that a backup path has failed.

The operator issues centered around concerns that operators were performing traffic engineering, but would have no visibility or control over hosts when they chose to begin using another path, and relying on hosts to engineer traffic exposed their networks to oscillation based on feedback loops, as hosts move from path to path.

At a minimum, traffic engineering policies must be pushed down to individual hosts. In addition, the usual concerns about firewalls that expected to find a transport-level protocol header in the IP payload, and won't be able to perform firewalling functions because its processing logic would have to look past the Identity header.

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

4.4.2. 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.

We also note that some path-aware networking ideas recycle. Although Shim6 did not achieve significant deployment, the IETF chartered a working group to specify "Multipath TCP" [MP-TCP] in 2009, and Multipath TCP allows TCP applications to control path selection, with many of the same advantages and disadvantages of Shim6.

4.5. Next Steps in Signaling (NSIS)

Write-up of Next Steps in Signaling (NSIS) [RFC5974]

Your description could be here.

5. Security Considerations

This document describes ideas 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 ideas for Path Aware Networking that would affect the security of the Internet, but security considerations for those ideas will be described in the corresponding RFCs that propose them.

6. IANA Considerations

This document makes no requests of IANA.

7. Acknowledgements

The section on IntServ was provided by Ron Bonica.

The section on Quick-Start TCP was provided by Michael Scharf.

The section on Shim6 builds on input provided by Erik Nordmark, with background added by Spencer Dawkins.

The section on Triggers for Transport (TRIGTRAN) was provided by Spencer Dawkins.

Review comments were provided by (your name could be here).

8. Informative References

[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>.

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

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

[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>.

[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>.

[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>.

[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>.

[RFC2581] Allman, M., Paxson, V., and W. Stevens, "TCP Congestion Control", RFC 2581, DOI 10.17487/RFC2581, April 1999, <https://www.rfc-editor.org/info/rfc2581>.

[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>.

[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>.

[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>.

[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>.

[RFC6928] Chu, J., Dukkipati, 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>.

[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>.

[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>.

[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.

[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>.

Author's Address

Spencer Dawkins (editor)
Huawei Technologies

Email: spencerdawkins.ietf@gmail.com

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

Table of Contents

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 tne 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 scalabilty/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 equivalend 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, Gorry 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