PANRG S. Dawkins, Ed.
Internet-Draft Wonder Hamster Internetworking
Intended status: Informational May 23, 2019
Expires: November 24, 2019

# Path Aware Networking: Obstacles to Deployment (A Bestiary of Roads Not Taken)

draft-irtf-panrg-what-not-to-do-03

## 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 technologies, most of which were unsuccessful, in order 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 the first meeting of the Path Aware Networking Research Group [PANRG], at IETF 99 [PANRG-99], Oliver 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 technologies and achieve a variety of goals over the past decade. At the end of this discussion, two things were abundantly clear.

- o The Internet community has accumulated considerable experience with many Path Aware technologies over a long period of time, and

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

The meta-lessons from that experience were

- o 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].

- o 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 (see Section 2), based on that catalog (see Section 4).

## 1.1. A Note About Path-Aware Technologies Included In This Document

This document does not catalog every technology about Path Aware Networking that was not implemented and deployed. Instead, we include enough technologies to provide background for the lessons included in Section 2 to guide researchers and protocol engineers in their work.

No shame is intended for the technologies included in this document. As shown in Section 2, the quality of specific technologies had little to do with whether they were deployed or not. Based on the technologies cataloged in this document, it is likely that when these technologies 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.

### 1.2. 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.

### 1.3. A Note for the Research Group

(RFC Editor: please remove this section before publication)

The editor and research group chairs are aware that the current version of this document is tilted toward transport-level Path Aware technologies, and would like to interact with other IETF protocol communities who have experience with Path Aware technologies.

It is worth looking at the Lessons Learned in Section 2 to see whether the Internet has changed in ways that would make some lessons less applicable for future protocol design.

### 1.4. A Note for the Editor

(Remove after taking these actions)

The to-do list for upcoming revisions includes

- o Confirm that the Summary of Lessons Learned makes sense and is complete, in consultation with the Research Group.
- o If the Research Group identifies technologies that provided lessons that aren't included in Section 2, solicit contributions for those technologies.
- o Provide better context for Section 2, to make sure that individual lessons aren't considered in isolation, and to distinguish between impediments to deployment and blockers for deployment.

### 1.5. 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 Setion 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.

Most of the Lessons Learned in Section 2 reflect considerations described in [RFC5218], [RFC7305], and [RFC8170].

## 2. Summary of Lessons Learned

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

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

It is useful to notice that sometimes an obstacle might impede deployment, while at other times, the same obstacle might prevent 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 technologies that were not deployed, and Path Aware technologies that were deployed, but were not deployed widely or quickly. See Section 4.6 and Section 4.6.3 as one example of this shifting boundary.

- o The benefit of Path Awareness must 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 a "best current practice" on today's Internet. (See Section 4.3, Section 4.5, and Section 4.4).

- o Providing benefits for early adopters can be key - if everyone must deploy a technology in order for the technology to provide benefits, or even to work at all, the technology is unlikely to be adopted. (See Section 4.2 and Section 4.3).

- o Adaptive end-to-end protocol mechanisms may respond to feedback quickly enough that the additional realizable benefit from a new Path Aware mechanism may be much smaller than anticipated (see Section 4.3 and Section 4.5).

- "Follow the money." If operators can't charge for a Path Aware technology to recover the costs of deploying it, the benefits to the operator must be really significant. Corollary: If operators charge for a Path Aware technology, the benefits to the user must be significant enough to justify the cost. (See Section 4.5, Section 4.1, and Section 4.2).

- Impact of a Path Aware technology requiring changes to operational practices can prevent deployment of promising technology. (See Section 4.6, including Section 4.6.3).

- Per-connection state in intermediate devices can be an impediment to adoption and deployment. This is especially true as we move from the edge of the network into the routing core (See Section 4.1 and Section 4.2).

- Many modern routers, especially high-end routers, have not been designed to make heavy use of in-band mechanisms such as IPv4 and IPv6 Router Alert Options (RAO), so operators can be reluctant to deploy technologies that rely on these mechanisms. (See Section 4.7).

- If the endpoints do not have any trust relationship with the intermediate devices along a path, operators can be reluctant to deploy technologies that rely on endpoints sending unauthenticated control signals to routers. (See Section 4.2 and Section 4.7. We also note this still remains a factor hindering deployment of DiffServ).

- If intermediate devices along the path can't be trusted, it's unlikely that endpoints will rely on signals from intermediate devices to drive changes to endpoint behaviors. (See Section 4.5, Section 4.4). The lowest level of trust is sufficient for a device issuing a message to 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 a network device could have a long or short term trust relationship with the sender it controls.

- Because the Internet is a distributed system, if the distance that information from distant hosts and routers travels to a Path Aware host or router is sufficiently large, the information may no longer represent the state and situation at the distant host or router when it is received. In this case, the benefit that a Path Aware technology provides likely decreases. (See Section 4.3).

- Providing a new feature/signal does not mean that it will be used. Endpoint stacks may not know how to effectively utilize Path-Aware

transport protocol technologies, because the technology may require information from applications to permit them to work effectively, but applications may not a-priori know that information. Even if the application does know that information, the de-facto API has no way of signaling the expectations of applications for the network path. Providing this awareness requires an API that signals more than the packets to be sent. TAPS is exploring such an API [TAPS-WG], yet even with such an API, policy is needed to bind the application expectations to the network characteristics. (See Section 4.1 and Section 4.2).

## 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.

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.

## 4. Contributions

Additional contributions that provide Lessons Learned beyond those already captured in Section 2 are welcomed.

### 4.1. Stream Transport (ST, ST2, ST2+)

The suggested references for IntServ 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 protocol numbers (IP, at that time, used protocol number 4, while ST2 used protocol 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.

#### 4.1.1. 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.

#### 4.1.2. 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 4.2.

### 4.2. 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 Mpbs) 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 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.

#### 4.2.1. 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

- 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 never 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.

#### 4.2.2. Lessons Learned.

The following lessons were learned:

- Any mechanism that requires a 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 technology 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].

### 4.3. Quick-Start TCP

The suggested references for Quick-Start TCP are:

- RFC 4782 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]

- RFC 5634 Quick-Start for the Datagram Congestion Control Protocol (DCCP) [RFC5634]

- 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. A corresponding mechanism was also specified for other congestion controllers (e.g., "Quick-Start for the Datagram Congestion Control Protocol (DCCP)" [RFC5634]). In these cases, a sender cannot easily determine an appropriate initial 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 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 devices 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.

### 4.3.1. 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 different packets, 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.

### 4.3.2. 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.

### 4.4. ICMP Source Quench

The suggested references for ICMP Source Quench are:

- o 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.

#### 4.4.1. 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.

### 4.4.2. 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 devices 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 device 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 device, 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.

## 4.5. 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 [RFC2581]. 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 TCP could be signaled when that link returned to service, and the sending TCP could retry immediately, without waiting for a full retransmission timeout (RTO) period.

### 4.5.1. 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 4.4, 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 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.

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.

### 4.5.2. Lessons Learned.

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

- o 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.

- o These reductions in scope were the direct result of an inability for hosts to trust or authenticate TRIGTRAN signals they received from the network.

- o 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.

## 4.6. Shim6

The suggested references for Shim6 are:

- o RFC5533 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 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 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.

#### 4.6.1. 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 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 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 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.

#### 4.6.2. 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.

#### 4.6.3. 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.

### 4.7. Next Steps in Signaling (NSIS)

The suggested references for NSIS are:

o the concluded working group charter [NSIS-CHARTER-2001]

o RFC 5971 GIST: General Internet Signalling Transport [RFC5971]

o RFC 5973 NAT/Firewall NSIS Signaling Layer Protocol (NSLP) [RFC5973]

o RFC 5974 NSIS Signaling Layer Protocol (NSLP) for Quality-of-Service Signaling [RFC5974]

o RFC 5981 "Authorization for NSIS Signaling Layer Protocols [RFC5981]

The Next Steps in Signaling (NSIS) Working Group worked on signaling technologies 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].

### 4.7.1. Reasons for Non-deployment

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

o Implementation-related aspects:

Although NSIS provides benefits with respect to flexibility, mobility, and security compared to other network signaling technologies, 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. One reason for the non-deployment of NSIS is the usage of the IPv4 and IPv6 Router Alert Option (RAO) to allow for an efficient interception of those path-coupled signaling messages: This option 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. End-to-end 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]).

o Operational Aspects:

End-to-end signaling technologies not only require trust between customers and their provider, but also among different providers. Especially, QoS signaling technologies 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 technologies. Furthermore, an end-to-end path typically involves several provider domains and these providers need to closely cooperate in cases of failures.

#### 4.7.2. 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:

- o the presence of asymmetric routes between endpoints and routers

- o the lack of security and trust at large in the Internet infrastructure

- o the presence of different trust boundaries

- o the effects of best-effort networks (e.g., robustness to packet loss)

- o 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".

### 4.8. IPv6 Flow Label

The suggested references for IPv6 Flow Label are:

- o 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 devices 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)).

#### 4.8.1. 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:

o Endpoint Implementation:

For network devices along a path to utilize the flow label there needs to be a non-zero value 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.

o Operational support in network devices:

A benefit can only be realized when a network device 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 device 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.

#### 4.8.2. 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:

o Before wide-scale support by endpoints, it would be impossible to rely on a non-zero flow label being set. Network devices 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.

o 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.

## 5. Security Considerations

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

## 6. IANA Considerations

This document makes no requests of IANA.

## 7. Acknowledgments

Initial material for Section 4.1 on ST2 was provided by Gorry Fairhurst.

Initial material for Section 4.2 on IntServ was provided by Ron Bonica.

Initial material for Section 4.3 on Quick-Start TCP was provided by Michael Scharf.

Initial material for Section 4.4 on ICMP Source Quench was provided by Gorry Fairhurst.

Initial material for Section 4.5 on Triggers for Transport (TRIGTRAN) was provided by Spencer Dawkins.

Section 4.6 on Shim6 builds on initial material describing obstacles provided by Erik Nordmark, with background added by Spencer Dawkins.

Initial material for Section 4.7 on Next Steps In Signaling (NSIS) was provided by Roland Bless and Martin Stiemerling.

Initial material for Section 4.8 on IPv6 Flow Labels was provided by Gorry Fairhurst.

Our thanks to C.M. Heard, Gorry Fairhurst, Joe Touch, Joeri de Ruiter, Roland Bless, Ruediger Geib, and Wes Eddy, who provided review comments on previous versions.

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 for the differences [Colossal-Cave].

## 8. Informative References

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

[GREASE] Thomson, M., "Long-term Viability of Protocol Extension Mechanisms", January 2019, <https://tools.ietf.org/html/draft-thomson-use-it-or-lose-it-03>.

[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/>.

[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-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., Sathiaseelan, 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/>.

[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>.

[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>.

[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>.

[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>.

[RFC5634] Fairhurst, G. and A. Sathiaseelan, "Quick-Start for the Datagram Congestion Control Protocol (DCCP)", RFC 5634, DOI 10.17487/RFC5634, August 2009, <https://www.rfc-editor.org/info/rfc5634>.

[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., 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>.

[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>.

[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>.

[TAPS-WG] "Transport Services Working Group Home Page", n.d., <https://datatracker.ietf.org/wg/taps/about/>.

[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)
Wonder Hamster Internetworking

Email: spencerdawkins.ietf@gmail.com