Internet Research Task Force (IRTF)
Request for Comments: 9049
Category: Informational
ISSN: 2070-1721

S. Dawkins, Ed.
Tencent America

June 2021

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

Abstract

This document is a product of the Path Aware Networking Research Group (PANRG). At the first meeting of the PANRG, the Research Group agreed to catalog and analyze past efforts to develop and deploy Path Aware techniques, most of which were unsuccessful or at most partially successful, in order to extract insights and lessons for Path Aware networking researchers.

This document contains that catalog and analysis.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Research Task Force (IRTF). The IRTF publishes the results of Internet-related research and development activities. These results might not be suitable for deployment. This RFC represents the consensus of the Path Aware Networking Research Group of the Internet Research Task Force (IRTF). Documents approved for publication by the IRSG are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9049.

Table of Contents

1. Introduction

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

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

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

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

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

It is worth noting that the definitions of "Path" and "Path Aware" in [PANRG-PATH-PROPERTIES] would apply to Path Aware techniques at a number of levels of the Internet protocol architecture ([RFC1122], plus several decades of refinements), but the contributions received for this document tended to target the transport layer and to treat a "Path" constructed by routers as opaque. It would be useful to consider how applicable the Lessons Learned cataloged in this document are, at other layers, and that would be a fine topic for follow-on research.

The current definition of "Path" in the Path Aware Networking Research Group appears in Section 2 ("Terminology") in [PANRG-PATH-PROPERTIES]. That definition is included here as a convenience to the reader.

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

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

> For purposes of this document, "path aware networking" describes endpoint discovery of the properties of paths they use for communication 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.

2. A Perspective on This Document

At the first meeting of the Path Aware Networking Research Group [PANRG], at IETF 99 [PANRG-99], Olivier Bonaventure led a discussion of "A Decade of Path Awareness" [PATH-Decade], on attempts, which

were mostly unsuccessful for a variety of reasons, to exploit Path Aware techniques and achieve a variety of goals over the past decade. At the end of that discussion, two things were abundantly clear.

* The Internet community has accumulated considerable experience with many Path Aware techniques over a long period of time, and

* Although some Path Aware techniques have been deployed (for example, Differentiated Services, or Diffserv [RFC2475]), most of these techniques haven't seen widespread adoption and deployment. Even "successful" techniques like Diffserv can face obstacles that prevent wider usage. The reasons for non-adoption and limited adoption and deployment are many and are worthy of study.

The meta-lessons from that experience were as follows:

* Path Aware networking has been more Research than Engineering, so establishing an IRTF Research Group for Path Aware networking was the right thing to do [RFC7418].

* Analyzing a catalog of past experience to learn the reasons for non-adoption would be a great first step for the Research Group.

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

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

2.1. Notes for the Reader

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

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

2.2. A Note about Path Aware Techniques Included in This Document

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

No shame is intended for the techniques included in this document. As shown in Section 4, the quality of specific techniques had little to do with whether they were deployed or not. Based on the techniques cataloged in this document, it is likely that when these techniques were put forward, the proponents were trying to engineer something that could not be engineered without first carrying out research. Actual shame would be failing to learn from experience and failing to share that experience with other networking researchers and engineers.

2.3. Architectural Guidance

As background for understanding the Lessons Learned contained in this document, the reader is encouraged to become familiar with the Internet Architecture Board's documents on "What Makes for a

Successful Protocol?" [RFC5218] and "Planning for Protocol Adoption and Subsequent Transitions" [RFC8170].

Although these two documents do not specifically target Path Aware networking protocols, they are helpful resources for readers seeking to improve their understanding of considerations for successful adoption and deployment of any protocol. For example, the basic success factors described in Section 2.1 of [RFC5218] are helpful for readers of this document.

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

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

2.4. Terminology Used in This Document

The terms "node" and "element" in this document have the meaning defined in [PANRG-PATH-PROPERTIES].

2.5. Methodology for Contributions

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

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

* the name of a technique, including an abbreviation if one was used.

* if available, a long-term pointer to the best reference describing the technique.

* a short description of the problem the technique was intended to solve.

* a short description of the reasons why the technique wasn’t adopted.

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

3. Applying the Lessons We’ve Learned

The initial scope for this document was roughly "What mistakes have we made in the decade prior to [PANRG-99], that we shouldn’t make again?" Some of the contributions in Section 6 predate the initial scope. The earliest Path Aware technique referred to in Section 6 is [IEN-119], which was published in the late 1970s; see Section 6.1. Given that the networking ecosystem has evolved continuously, it seems reasonable to consider how to apply these lessons.

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

* "Invariant" - well-understood and is likely to be applicable for any proposed Path Aware networking solution.

* "Variable" - has impeded deployment in the past but might not be applicable in a specific technique. Engineering analysis to understand whether the lesson is applicable is prudent.

* "Not Now" - a characteristic that tends to turn up a minefield full of dragons. Prudent network engineers will wish to avoid gambling on a technique that relies on this, until something significant changes.

Section 6.9 on Explicit Congestion Notification (ECN) was added during the review and approval process, based on a question from Martin Duke. Section 6.9, as contained in the March 8, 2021 draft version of this document, was discussed at [PANRG-110] and is summarized in Section 4.13, describing a new Lesson Learned.

Lesson	Category
Justifying Deployment (Section 4.1)	Invariant
Providing Benefits for Early Adopters (Section 4.2)	Invariant
Providing Benefits during Partial Deployment (Section 4.3)	Invariant
Outperforming End-to-End Protocol Mechanisms (Section 4.4)	Variable
Paying for Path Aware Techniques (Section 4.5)	Invariant
Impact on Operational Practices (Section 4.6)	Invariant
Per-Connection State (Section 4.7)	Variable
Keeping Traffic on Fast Paths (Section 4.8)	Variable
Endpoints Trusting Intermediate Nodes (Section 4.9)	Not Now
Intermediate Nodes Trusting Endpoints (Section 4.10)	Not Now
Reacting to Distant Signals (Section 4.11)	Variable
Support in Endpoint Protocol Stacks (Section 4.12)	Variable
Planning for Failure (Section 4.13)	Invariant

Table 1

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

"Providing Benefits during Partial Deployment" was added after IETF 105, during Research Group Last Call, and is also considered to be Invariant.

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

* For example, if you can obtain the same understanding of path characteristics from measurements obtained over a few more round

trips, endpoint implementers are unlikely to be eager to add complexity, and many attributes can be measured from an endpoint, without assistance from intermediate nodes.

For "Per-Connection State", the key questions discussed in the Research Group were "how much state" and "where state is maintained".

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

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

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

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

* We noted with relief that discussions about this were already underway in the IETF community at IETF 105 (see the Security Area Open Meeting minutes [SAAG-105-Min] for discussion of [INTERNET-THREAT-MODEL] and [FARRELL-ETM]), and the Internet Architecture Board has created a mailing list for continued discussions [model-t], but we recognize that there are Path Aware networking aspects of this effort, requiring research.

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

* If an attribute is stable over an extended period of time, is difficult to observe via end-to-end mechanisms, and is valuable, Path Aware techniques that rely on that attribute to provide a significant benefit become more attractive.

* Analysis to help identify attributes that are useful enough to justify deployment of Path Aware techniques that make use of those attributes would be helpful.

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

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

4. Summary of Lessons Learned

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

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

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

4.1. Justifying Deployment

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

4.2. Providing Benefits for Early Adopters

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

4.3. Providing Benefits during Partial Deployment

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

4.4. Outperforming End-to-End Protocol Mechanisms

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

4.5. Paying for Path Aware Techniques

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

4.6. Impact on Operational Practices

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

4.7. Per-Connection State

Per-connection state in intermediate nodes has been an impediment to

adoption and deployment in the past, because of added cost and complexity. Often, similar benefits can be achieved with much less finely grained state. This is especially true as we move from the edge of the network, further into the routing core. (See Sections 6.1 and 6.2.)

4.8. Keeping Traffic on Fast Paths

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

4.9. Endpoints Trusting Intermediate Nodes

If intermediate nodes along the path can’t be trusted, it’s unlikely that endpoints will rely on signals from intermediate nodes to drive changes to endpoint behaviors. We note that "trust" is not binary -- one low level of trust applies when a node receiving a message can confirm that the sender of the message has visibility of the packets on the path it is seeking to control [RFC8085] (e.g., an ICMP Destination Unreachable message [RFC0792] that includes the Internet Header + 64 bits of Original Data Datagram payload from the source). A higher level of trust can arise when an endpoint has established a short-term, or even long-term, trust relationship with network nodes. (See Sections 6.4 and 6.5.)

4.10. Intermediate Nodes Trusting Endpoints

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

4.11. Reacting to Distant Signals

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

4.12. Support in Endpoint Protocol Stacks

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

4.13. Planning for Failure

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

5. Future Work

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

* We note that this work was based on contributions from experts on various Path Aware techniques, and all of the contributed techniques involved unicast protocols. We didn't consider how these lessons might apply to multicast, and, given anecdotal reports at the IETF 109 Media Operations (MOPS) Working Group meeting of IP multicast offerings within data centers at one or more cloud providers [MOPS-109-Min], it might be useful to think about Path Awareness in multicast, before we have a history of unsuccessful deployments to document.

* The question of whether a mechanism supports admission control, based on either endpoints or applications, is associated with Path Awareness. One of the motivations of IntServ and a number of other architectures (e.g., Deterministic Networking [RFC8655]) is the ability to "say no" to an application based on resource availability on a path, before the application tries to inject traffic onto that path and discovers the path does not have the capacity to sustain enough utility to meet the application's minimum needs. The question of whether admission control is needed comes up repeatedly, but we have learned a few useful lessons that, while covered implicitly in some of the Lessons Learned provided in this document, might be explained explicitly:

  - We have gained a lot of experience with application-based adaptation since the days where applications just injected traffic inelastically into the network. Such adaptations seem to work well enough that admission control is of less value to these applications.

  - There are end-to-end measurement techniques that can steer traffic at the application layer (Content Delivery Networks (CDNs), multi-CDNs like Conviva [Conviva], etc.).

  - We noted in Section 4.12 that applications often don't know how to utilize Path Aware techniques. This includes not knowing enough about their admission control threshold to be able to ask accurately for the resources they need, whether this is because the application itself doesn't know or because the application has no way to signal its expectations to the underlying protocol stack. To date, attempts to help them haven't gotten anywhere (e.g., the multiple-TSPEC (Traffic Specification) additions to RSVP to attempt to mirror codec selection by applications [INTSERV-MULTIPLE-TSPEC] expired in 2013).

* We note that this work took the then-current IP network architecture as given, at least at the time each technique was proposed. It might be useful to consider aspects of the now-

current IP network architecture that ease, or impede, Path Aware techniques. For example, there is limited ability in IP to constrain bidirectional paths to be symmetric, and information-centric networking protocols such as Named Data Networking (NDN) and Content-Centric Networking (CCNx) [RFC8793] must force bidirectional path symmetry using protocol-specific mechanisms.

6. Contributions

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

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

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

The suggested references for Stream Transport are:

* "ST - A Proposed Internet Stream Protocol" [IEN-119]

* "Experimental Internet Stream Protocol: Version 2 (ST-II)" [RFC1190]

* "Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+" [RFC1819]

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

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

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

6.1.1. Reasons for Non-deployment

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

The ST2 protocol is no longer in use.

6.1.2. Lessons Learned

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

The ST2 control protocol used "hard state" -- once a route was established, and resources were reserved, routes and resources

existed until they were explicitly released via signaling. A soft-state approach was thought superior to this hard-state approach and led to development of the IntServ model described in Section 6.2.

6.2. Integrated Services (IntServ)

The suggested references for IntServ are:

* "Integrated Services in the Internet Architecture: an Overview" [RFC1633]

* "Specification of the Controlled-Load Network Element Service" [RFC2211]

* "Specification of Guaranteed Quality of Service" [RFC2212]

* "General Characterization Parameters for Integrated Service Network Elements" [RFC2215]

* "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification" [RFC2205]

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

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

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

6.2.1. Reasons for Non-deployment

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

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

* IntServ maintained per-flow state.

As IntServ was being discussed, the following occurred:

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

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

6.2.2. Lessons Learned

The following lessons were learned:

* Any mechanism that requires every participating on-path 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. There are often clearly defined hierarchies in Service Level Agreements (SLAs) governing well-defined transport flows operating with predetermined capacity and latency requirements over paths where capacity or other attributes are constrained.

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

6.3. Quick-Start TCP

The suggested references for Quick-Start TCP are:

* "Quick-Start for TCP and IP" [RFC4782]

* "Determining an appropriate sending rate over an underutilized network path" [SAF07]

* "Fast Startup Internet Congestion Control for Broadband Interactive Applications" [Sch11]

* "Using Quick-Start to enhance TCP-friendly rate control performance in bidirectional satellite networks" [QS-SAT]

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

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

The protocol also includes a nonce that provides protection against cheating routers and receivers. If the Quick-Start request is explicitly approved by all routers along the path, the TCP host can send at up to the approved rate; otherwise, TCP would use the default congestion control. Quick-Start requires modifications in the involved end systems as well as 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 analyze Quick-Start with a full Linux implementation and with a router fast-path prototype using a network processor. In both cases, Quick-Start could be implemented with limited additional complexity.

6.3.1. Reasons for Non-deployment

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

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

* For scalable router fast-path implementations, it is important to enable parallel processing of packets, as this is a widely used method, e.g., in network processors. One challenge is synchronization of information between packets that are processed in parallel, which should be avoided as much as possible.

* Only some types of application traffic can benefit from Quick-Start. Capacity needs to be requested and discovered. The discovered capacity needs to be utilized by the flow, or it implicitly becomes available for other flows. Failing to use the requested capacity may have already reduced the pool of Quick-Start capacity that was made available to other competing Quick-Start requests. The benefit is greatest when senders use this only for bulk flows and avoid sending unnecessary Quick-Start requests, e.g., for flows that only send a small amount of data. Choosing an appropriate request size requires application-internal knowledge that is not commonly expressed by the transport API. How a sender can determine the rate for an initial Quick-Start request is still a largely unsolved problem.

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

6.3.2. Lessons Learned

Some lessons can be learned from Quick-Start. Despite being a very lightweight protocol, Quick-Start suffers from poor incremental deployment properties regarding both a) the required modifications in network infrastructure and b) 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 segments

[RFC6928]. This alternative "IW10" approach can also ramp up data transfers faster than the standard congestion control, but it only requires sender-side modifications. As a result, this approach can be easier and incrementally deployed in the Internet. While theoretically Quick-Start can outperform "IW10", the improvement in completion time for data transfer times can, in many cases, be small. After publication of [RFC6928], most modern TCP stacks have increased their default initial window.

6.4. ICMP Source Quench

The suggested reference for ICMP Source Quench is:

* "Internet Control Message Protocol" [RFC0792]

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

6.4.1. Reasons for Non-deployment

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

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

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

6.4.2. Lessons Learned

This method had several problems.

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

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

The mechanism originated at a time when the Internet trust model was very different. Most endpoint implementations did not attempt to verify that the message originated from an on-path node before they utilized the message. This made it vulnerable to Denial-of-Service (DoS) attacks. In theory, routers might have chosen to use the quoted packet contained in the ICMP payload to validate that the message originated from an on-path node, but this would have increased per-packet processing overhead for each router along the path and 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 DoS opportunity.

6.5. Triggers for Transport (TRIGTRAN)

The suggested references for TRIGTRAN are:

* TRIGTRAN BOF at IETF 55 [TRIGTRAN-55]

* TRIGTRAN BOF at IETF 56 [TRIGTRAN-56]

TCP [RFC0793] has a well-known weakness -- the end-to-end flow control mechanism has only a single signal, the loss of a segment, detected when no acknowledgment for the lost segment is received at the sender. There are multiple reasons why the sender might not have received an acknowledgment for the segment. To name several, the segment could have been trapped in a routing loop, damaged in transmission and failed checksum verification at the receiver, or lost because some intermediate device discarded the packet, or any of a variety of other things could have happened to the acknowledgment on the way back from the receiver to the sender. TCP implementations since the late 1980s have made the "safe" decision and have interpreted the loss of a segment as evidence that the path between two endpoints may have become congested enough to exhaust buffers on intermediate hops, so that the TCP sender should "back off" -- reduce its sending rate until it knows that its segments are now being delivered without loss [RFC5681].

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

6.5.1. Reasons for Non-deployment

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

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

But things got worse.

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

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

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

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

6.5.2. Lessons Learned

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

* TRIGTRAN started with a plausible value proposition, but networking realities in the early 2000s forced reductions in scope that led directly to reductions in potential benefits but no corresponding reductions in costs and complexity.

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

* Operators did not believe they could charge for TRIGTRAN signaling, because first-hop links didn't fail frequently and TRIGTRAN provided no reduction in operating expenses, so there was little incentive to purchase and deploy TRIGTRAN-capable network equipment.

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

6.6. Shim6

The suggested reference for Shim6 is:

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

The IPv6 routing architecture [RFC1887] assumed that most sites on the Internet would be identified by Provider Assigned IPv6 prefixes, so that Default-Free Zone routers only contained routes to other providers, resulting in a very small IPv6 global routing table.

For a single-homed site, this could work well. A multihomed site with only one upstream provider could also work well, although BGP multihoming from a single upstream provider was often a premium

service (costing more than twice as much as two single-homed sites), and if the single upstream provider went out of service, all of the multihomed paths could fail simultaneously.

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

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

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

6.6.1. Reasons for Non-deployment

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

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

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

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

In addition, operators would have no visibility or control over the decision of hosts choosing to switch to another path. They expressed concerns that relying on hosts to steer traffic exposed operator networks to oscillation based on feedback loops, if hosts moved from path to path frequently. Given that Shim6 was intended to support multihoming across operators, operators providing only one of the paths would have even less visibility as traffic suddenly appeared and disappeared on their networks.

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

would not look past the Identity header. The firewall would perform its default action, which would most likely be to drop packets that don't match any processing rule.

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

6.6.2. Lessons Learned

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

6.6.3. Addendum on Multipath TCP

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

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

6.7. Next Steps in Signaling (NSIS)

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

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

* "GIST: General Internet Signalling Transport" [RFC5971]

* "NAT/Firewall NSIS Signaling Layer Protocol (NSLP)" [RFC5973]

* "NSIS Signaling Layer Protocol (NSLP) for Quality-of-Service Signaling" [RFC5974]

* "Authorization for NSIS Signaling Layer Protocols" [RFC5981]

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

When RSVP [RFC2205] was used in deployments, a number of questions came up about its perceived limitations and potential missing features. The issues noted in the NSIS Working Group charter [NSIS-CHARTER-2001] include interworking between domains with different QoS architectures, mobility and roaming for IP interfaces, and complexity. Later, the lack of security in RSVP was also recognized [RFC4094].

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

NTLP is defined in [RFC5971]. Two types of NSLPs are defined: an NSLP for QoS signaling [RFC5974] and an NSLP for NATs/firewalls [RFC5973].

6.7.1. Reasons for Non-deployment

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

* Implementation-related aspects:

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

  NTLP mainly operates as a path-coupled signaling protocol, i.e., its messages are processed at the control plane of each intermediate node that is also forwarding the data flows. This requires a mechanism to intercept signaling packets while they are forwarded in the same manner (especially along the same path) as data packets. NSIS uses the IPv4 and IPv6 Router Alert Option (RAO) to allow for interception of those path-coupled signaling messages, and this technique requires router implementations to correctly understand and implement the handling of RAOs, e.g., to only process packets 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 DoS threat to a router’s control plane) and difficult for an operator to control. Path-coupled signaling approaches were considered problematic (see also Section 3 of [RFC6398]). There are recommendations on how to secure NSIS nodes and deployments (e.g., [RFC5981]).

* Operational Aspects:

  NSIS not only required trust between customers and their provider, but also among different providers. In particular, QoS signaling techniques would require some kind of dynamic SLA support that would imply (potentially quite complex) bilateral negotiations between different Internet Service Providers. This complexity was not considered to be justified, and increasing the bandwidth (and thus avoiding bottlenecks) was cheaper than actively managing network resource bottlenecks by using path-coupled QoS signaling techniques. Furthermore, an end-to-end path typically involves several provider domains, and these providers need to closely cooperate in cases of failures.

6.7.2. Lessons Learned

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

* the presence of asymmetric routes between endpoints and routers.

* the lack of security and trust at large in the Internet infrastructure.

* the presence of different trust boundaries.

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

* divergence from the fate-sharing principle (e.g., state within the network).

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

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

6.8. IPv6 Flow Labels

The suggested reference for IPv6 Flow Labels is:

* "IPv6 Flow Label Specification" [RFC6437]

IPv6 specifies a 20-bit 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 pseudorandomly 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 either independently forward its subflows or use one common value for the traffic aggregate. The flow label is present in all fragments. IPsec was originally put forward as one important use case for this mechanism and does encrypt the field [RFC6438].

Once set, the flow label can provide information that can help inform network nodes about subflows present at the transport layer, without needing to interpret the setting of upper-layer protocol fields [RFC6294]. This information can also be used to coordinate how aggregates of transport subflows are grouped when queued in the network and to select appropriate per-flow forwarding when choosing between alternate paths [RFC6438] (e.g., for Equal-Cost Multipath (ECMP) routing and Link Aggregation Groups (LAGs)).

6.8.1. Reasons for Non-deployment

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

* Endpoint Implementation:

  For network nodes along a path to utilize the flow label, there needs to be a non-zero value inserted in the field [RFC6437] at the sending endpoint. There needs to be an incentive for an endpoint to set an appropriate non-zero value. The value should appropriately reflect the level of aggregation the traffic expects

to be provided by the network. However, this requires the stack to know granularity at which flows should be identified (or, conversely, which flows should receive aggregated treatment), i.e., which packets carry the same flow label. Therefore, setting a non-zero value may result in additional choices that need to be made by an application developer.

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

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

* Operational support in network nodes:

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

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

6.8.2. Lessons Learned

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

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

* Before wide-scale support by endpoints, it would be impossible to rely on a non-zero flow label being set. Network nodes therefore would need to also employ other techniques to realize equivalent functions. An example of a method is one assuming semantics of the source port field to provide entropy input to a network-layer hash. This use of a 5-tuple to classify a packet represents a layering violation [RFC6294]. When other methods have been deployed, they increase the cost of deploying standards-based methods, even though they may offer less control to endpoints and result in potential interaction with other uses/interpretation of the field.

* Even though the flow label is specified as an end-to-end field, some network paths have been observed to not transparently forward the flow label. This could result from non-conformant equipment or could indicate that some operational networks have chosen to reuse 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 that 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 [RFC9000]) seems likely to raise issues similar to those discussed above and could motivate renewed interest in utilizing the flow label.

6.9. Explicit Congestion Notification (ECN)

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

* "Recommendations on Queue Management and Congestion Avoidance in the Internet" [RFC2309]

* "A Proposal to add Explicit Congestion Notification (ECN) to IP" [RFC2481]

* "The Addition of Explicit Congestion Notification (ECN) to IP" [RFC3168]

* "Implementation Report on Experiences with Various TCP RFCs" [vista-impl], slides 6 and 7

* "Implementation and Deployment of ECN" (at [SallyFloyd])

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

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

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

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

> Many routers process the "regular" headers in IP packets more

efficiently than they process the header information in IP options.

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

6.9.1. Reasons for Non-deployment

ECN deployment relied on three factors -- support in client implementations, support in router implementations, and deployment decisions in operational networks.

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

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

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

As described in [vista-impl] ("IGD" stands for "Intermediate Gateway Device"),

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

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

6.9.2. Lessons Learned

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

* Even if you do everything right, you may trip over implementation bugs in devices you know nothing about, that will cause severe problems that prevent successful deployment of your Path Aware technology.

* After implementations disable your Path Aware technology, it may take years, or even decades, to convince implementers to re-enable

it by default.

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

During discussion of ECN at [PANRG-110], we noted that "you get one chance to get it right" isn't quite correct today, because operating systems on so many host systems are frequently updated, and transport protocols like QUIC [RFC9000] are being implemented in user space and can be updated without touching installed operating systems. Neither of these factors were true in the early 2000s.

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

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

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

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

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

7. Security Considerations

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

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

8. IANA Considerations

This document has no IANA actions.

9. Informative References

[Colossal-Cave]
Wikipedia, "Colossal Cave Adventure", June 2021, <https://en.wikipedia.org/w/index.php?title=Colossal_Cave_Adventure&oldid=1027119625>.

[Conviva] "Conviva Precision : Data Sheet", January 2021,

<https://www.conviva.com/datasheets/precision-delivery-intelligence/>.

[FARRELL-ETM]
Farrell, S., "We’re gonna need a bigger threat model", Work in Progress, Internet-Draft, draft-farrell-etm-03, 6 July 2019, <https://datatracker.ietf.org/doc/html/draft-farrell-etm-03>.

[GREASE] Thomson, M., "Long-term Viability of Protocol Extension Mechanisms", Work in Progress, Internet-Draft, draft-iab-use-it-or-lose-it-00, 7 August 2019, <https://datatracker.ietf.org/doc/html/draft-iab-use-it-or-lose-it-00>.

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

[INTERNET-THREAT-MODEL]
Arkko, J., "Changes in the Internet Threat Model", Work in Progress, Internet-Draft, draft-arkko-arch-internet-threat-model-01, 8 July 2019, <https://datatracker.ietf.org/doc/html/draft-arkko-arch-internet-threat-model-01>.

[INTSERV-MULTIPLE-TSPEC]
Polk, J. and S. Dhesikan, "Integrated Services (IntServ) Extension to Allow Signaling of Multiple Traffic Specifications and Multiple Flow Specifications in RSVPv1", Work in Progress, Internet-Draft, draft-ietf-tsvwg-intserv-multiple-tspec-02, 25 February 2013, <https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-intserv-multiple-tspec-02>.

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

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

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

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

[NANOG-35] "NANOG 35 Agenda", North American Network Operators’ Group (NANOG), October 2005, <https://archive.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", <https://irtf.org/panrg>.

[PANRG-103-Min]

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

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

[PANRG-106-Min]
"Path Aware Networking Research Group - IETF 106 Minutes", November 2019, <https://datatracker.ietf.org/doc/minutes-106-panrg/>.

[PANRG-110]
"Path Aware Networking Research Group - IETF 110", March 2021, <https://datatracker.ietf.org/meeting/110/session/panrg>.

[PANRG-99] "Path Aware Networking Research Group - IETF 99", July 2017, <https://datatracker.ietf.org/meeting/99/session/panrg>.

[PANRG-PATH-PROPERTIES]
Enghardt, T. and C. Krhenbhl, "A Vocabulary of Path Properties", Work in Progress, Internet-Draft, draft-irtf-panrg-path-properties-02, 22 February 2021, <https://datatracker.ietf.org/doc/html/draft-irtf-panrg-path-properties-02>.

[PANRG-QUESTIONS]
Trammell, B., "Current Open Questions in Path Aware Networking", Work in Progress, Internet-Draft, draft-irtf-panrg-questions-09, 16 April 2021, <https://datatracker.ietf.org/doc/html/draft-irtf-panrg-questions-09>.

[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., Potort, F., Gotta, A., and G. Fairhurst, "Using Quick-Start to enhance TCP-friendly rate control performance in bidirectional satellite networks", DOI 10.1002/sat.929, May 2009, <https://dl.acm.org/citation.cfm?id=3160304.3160305>.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

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

[RFC1016] Prue, W. and J. Postel, "Something a Host Could Do with Source Quench: The Source Quench Introduced Delay (SQuID)", RFC 1016, DOI 10.17487/RFC1016, July 1987, <https://www.rfc-editor.org/info/rfc1016>.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts -

Communication Layers", STD 3, RFC 1122,
DOI 10.17487/RFC1122, October 1989,
<https://www.rfc-editor.org/info/rfc1122>.

[RFC1190] Topolcic, C., "Experimental Internet Stream Protocol:
Version 2 (ST-II)", RFC 1190, DOI 10.17487/RFC1190,
October 1990, <https://www.rfc-editor.org/info/rfc1190>.

[RFC1633] Braden, R., Clark, D., and S. Shenker, "Integrated
Services in the Internet Architecture: an Overview",
RFC 1633, DOI 10.17487/RFC1633, June 1994,
<https://www.rfc-editor.org/info/rfc1633>.

[RFC1809] Partridge, C., "Using the Flow Label Field in IPv6",
RFC 1809, DOI 10.17487/RFC1809, June 1995,
<https://www.rfc-editor.org/info/rfc1809>.

[RFC1819] Delgrossi, L., Ed. and L. Berger, Ed., "Internet Stream
Protocol Version 2 (ST2) Protocol Specification - Version
ST2+", RFC 1819, DOI 10.17487/RFC1819, August 1995,
<https://www.rfc-editor.org/info/rfc1819>.

[RFC1887] Rekhter, Y., Ed. and T. Li, Ed., "An Architecture for IPv6
Unicast Address Allocation", RFC 1887,
DOI 10.17487/RFC1887, December 1995,
<https://www.rfc-editor.org/info/rfc1887>.

[RFC2001] Stevens, W., "TCP Slow Start, Congestion Avoidance, Fast
Retransmit, and Fast Recovery Algorithms", RFC 2001,
DOI 10.17487/RFC2001, January 1997,
<https://www.rfc-editor.org/info/rfc2001>.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S.
Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1
Functional Specification", RFC 2205, DOI 10.17487/RFC2205,
September 1997, <https://www.rfc-editor.org/info/rfc2205>.

[RFC2210] Wroclawski, J., "The Use of RSVP with IETF Integrated
Services", RFC 2210, DOI 10.17487/RFC2210, September 1997,
<https://www.rfc-editor.org/info/rfc2210>.

[RFC2211] Wroclawski, J., "Specification of the Controlled-Load
Network Element Service", RFC 2211, DOI 10.17487/RFC2211,
September 1997, <https://www.rfc-editor.org/info/rfc2211>.

[RFC2212] Shenker, S., Partridge, C., and R. Guerin, "Specification
of Guaranteed Quality of Service", RFC 2212,
DOI 10.17487/RFC2212, September 1997,
<https://www.rfc-editor.org/info/rfc2212>.

[RFC2215] Shenker, S. and J. Wroclawski, "General Characterization
Parameters for Integrated Service Network Elements",
RFC 2215, DOI 10.17487/RFC2215, September 1997,
<https://www.rfc-editor.org/info/rfc2215>.

[RFC2309] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering,
S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G.,
Partridge, C., Peterson, L., Ramakrishnan, K., Shenker,
S., Wroclawski, J., and L. Zhang, "Recommendations on
Queue Management and Congestion Avoidance in the
Internet", RFC 2309, DOI 10.17487/RFC2309, April 1998,
<https://www.rfc-editor.org/info/rfc2309>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6
(IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460,
December 1998, <https://www.rfc-editor.org/info/rfc2460>.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, DOI 10.17487/RFC2475, December 1998, <https://www.rfc-editor.org/info/rfc2475>.

[RFC2481] Ramakrishnan, K. and S. Floyd, "A Proposal to add Explicit Congestion Notification (ECN) to IP", RFC 2481, DOI 10.17487/RFC2481, January 1999, <https://www.rfc-editor.org/info/rfc2481>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

[RFC3697] Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, "IPv6 Flow Label Specification", RFC 3697, DOI 10.17487/RFC3697, March 2004, <https://www.rfc-editor.org/info/rfc3697>.

[RFC4094] Manner, J. and X. Fu, "Analysis of Existing Quality-of-Service Signaling Protocols", RFC 4094, DOI 10.17487/RFC4094, May 2005, <https://www.rfc-editor.org/info/rfc4094>.

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

[RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-Start for TCP and IP", RFC 4782, DOI 10.17487/RFC4782, January 2007, <https://www.rfc-editor.org/info/rfc4782>.

[RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, DOI 10.17487/RFC5082, October 2007, <https://www.rfc-editor.org/info/rfc5082>.

[RFC5218] Thaler, D. and B. Aboba, "What Makes for a Successful Protocol?", RFC 5218, DOI 10.17487/RFC5218, July 2008, <https://www.rfc-editor.org/info/rfc5218>.

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

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <https://www.rfc-editor.org/info/rfc5575>.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <https://www.rfc-editor.org/info/rfc5681>.

[RFC5971] Schulzrinne, H. and R. Hancock, "GIST: General Internet Signalling Transport", RFC 5971, DOI 10.17487/RFC5971, October 2010, <https://www.rfc-editor.org/info/rfc5971>.

[RFC5973] Stiemerling, M., Tschofenig, H., Aoun, C., and E. Davies, "NAT/Firewall NSIS Signaling Layer Protocol (NSLP)", RFC 5973, DOI 10.17487/RFC5973, October 2010, <https://www.rfc-editor.org/info/rfc5973>.

[RFC5974] Manner, J., Karagiannis, G., and A. McDonald, "NSIS Signaling Layer Protocol (NSLP) for Quality-of-Service Signaling", RFC 5974, DOI 10.17487/RFC5974, October 2010, <https://www.rfc-editor.org/info/rfc5974>.

[RFC5981] Manner, J., Stiemerling, M., Tschofenig, H., and R. Bless, Ed., "Authorization for NSIS Signaling Layer Protocols", RFC 5981, DOI 10.17487/RFC5981, February 2011, <https://www.rfc-editor.org/info/rfc5981>.

[RFC6294] Hu, Q. and B. Carpenter, "Survey of Proposed Use Cases for the IPv6 Flow Label", RFC 6294, DOI 10.17487/RFC6294, June 2011, <https://www.rfc-editor.org/info/rfc6294>.

[RFC6398] Le Faucheur, F., Ed., "IP Router Alert Considerations and Usage", BCP 168, RFC 6398, DOI 10.17487/RFC6398, October 2011, <https://www.rfc-editor.org/info/rfc6398>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <https://www.rfc-editor.org/info/rfc6438>.

[RFC6633] Gont, F., "Deprecation of ICMP Source Quench Messages", RFC 6633, DOI 10.17487/RFC6633, May 2012, <https://www.rfc-editor.org/info/rfc6633>.

[RFC6928] Chu, J., 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>.

[RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <https://www.rfc-editor.org/info/rfc8655>.

[RFC8793] Wissingh, B., Wood, C., Afanasyev, A., Zhang, L., Oran, D., and C. Tschudin, "Information-Centric Networking (ICN): Content-Centric Networking (CCNx) and Named Data Networking (NDN) Terminology", RFC 8793, DOI 10.17487/RFC8793, June 2020, <https://www.rfc-editor.org/info/rfc8793>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[SAAG-105-Min] "Security Area Open Meeting - IETF 105 Minutes", July 2019, <https://datatracker.ietf.org/meeting/105/materials/minutes-105-saag-00>.

[SAF07] Sarolahti, P., Allman, M., and S. Floyd, "Determining an appropriate sending rate over an underutilized network path", Computer Networks: The International Journal of Computer and Telecommunications Networking, Volume 51, Number 7, DOI 10.1016/j.comnet.2006.11.006, May 2007, <https://dl.acm.org/doi/10.1016/j.comnet.2006.11.006>.

[SallyFloyd] Floyd, S., "ECN (Explicit Congestion Notification) in TCP/IP", June 2009, <https://www.icir.org/floyd/ecn.html>.

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

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

[TRIGTRAN-55] "Triggers for Transport BOF at IETF 55", November 2002, <https://www.ietf.org/proceedings/55/239.htm>.

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

[vista-impl] Sridharan, M., Bansal, D., and D. Thaler, "Implementation Report on Experiences with Various TCP RFCs", March 2007, <https://www.ietf.org/proceedings/68/slides/tsvarea-3/sld1.htm>.

Acknowledgments

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Author’s Address

Spencer Dawkins (editor)
Tencent America
United States of America

Email: spencerdawkins.ietf@gmail.com