

Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 4, 2022

Z. Du
P. Liu
China Mobile
December 31, 2021

Gateway Based Trust Relationship Between the Endpoint and the
Intermediate Node
draft-du-panrg-gateway-based-trust-relationship-01

Abstract

This document describes a mechanism about establishing trust relationship between the endpoint and the intermediate node along the path based on the gateway of the endpoint.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 4, 2022.

Copyright Notice

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

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

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Proposed Mechanism for the Trust Problem	3
3. IANA Considerations	4
4. Security Considerations	4
5. Acknowledgements	4
6. References	4
6.1. Normative References	4
6.2. Informative References	5
Authors' Addresses	5

1. Introduction

In future, many new services would emerge in the network, such as the 5G URLLC (Ultra Reliable Low Latency Communication) service, and the holographic type communications. Many of the new services need a higher QoS (Quality of Service) level than the current Internet services, and some of them have a critical SLA (Service-Level Agreement) requirement. The SLA differences between the new services and traditional services would become larger and larger. However, current networks can only provide the Best Effort bearing, in which all the traffic are treated as the same kind. In summary, current networks are short of negotiation abilities between the network and the applications. PANRG in the IRTF has proposed a research direction to enable the path aware networking. A lot of analyses have been done in the [RFC9049], which explains reasons why various Path Aware techniques have seen limited or no deployment.

One of the reasons is that it is hard to establish a trust relationship between the Endpoint and Intermediate Node. In the current network structure, the Endpoints only needs to be aware of the each other, and assume that the network can provide a good connection service for them. On the other hand, traditionally, Intermediate Nodes only need to support IP forwarding and do not need to be aware of up-layer information. In addition, the network nodes work in a per-packet model, not a per-flow model. Also in the [RFC9049], it is said that "per-connection state in intermediate nodes has been an impediment to adoption and deployment".

However, we can find that the gateway of the Endpoint is able to maintain a per-connection state and a trust-relationship for each

user. For example, the users in the fixed network need to be authorized by the BNG (Broadband Network Gateway), and the BNG also needs to do the accounting for each user. It is hard and unnecessary to make every intermediate node along the path has the same ability as the BNG; however, if they can have some communication with the BNG, perhaps they can make a better path choice for the user. Following this direction, this document proposes a mechanism about how to enable the communication between the BNG and the Head-End node in the network, because the Head-End node is the main node to select the path for a flow in the network. If any future work on the trust relationship between the Endpoint and the Intermediate Node is considered, the mechanism in this document can be a reference.

2. Proposed Mechanism for the Trust Problem

As shown in the Figure 1, in the fixed network, the BNG works as the gateway for the Client, and provides the Internet connection service for the Applications. The Client and Server are the EndPoints, and the BNG, Head-End, Mid-Node, End-Node are the nodes along the path from the Client to the Server. There are three paths, i.e., A, B, C, with different properties such as high bandwidth or low latency, between the Head-End and the End-Node in the network.

By default, all the traffic from the APPs are forwarded from the Head-End to the End-Node with the same treatment in the network. In the Head-end, perhaps a load balance mechanism can be enabled, but normally without any per-flow mechanism, because the Head-End does not know the requirements of each flow. If the Applications need different treatments in the network, and the Head-End can schedule the traffic to a proper path, the user can have a better experience and the network resource can be used more efficiently.

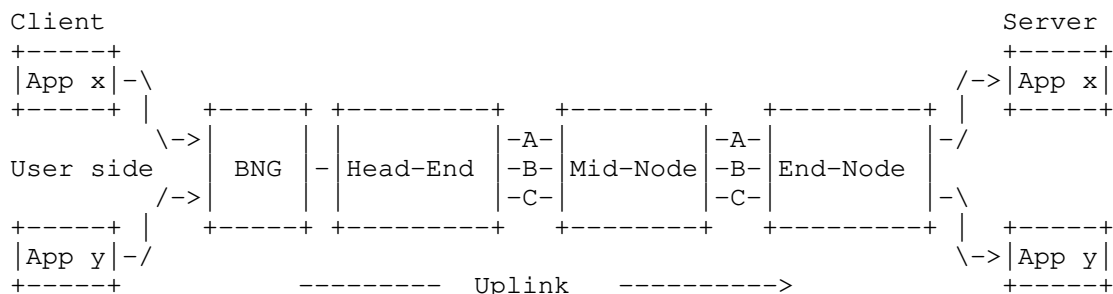


Figure 1: Path-aware Mechanism in the Fixed Network

The following paragraphs are about the trust problems and the potential solutions for them.

The first problem is the path information collection for the Endpoints. The Endpoints should be able to trust the path information that the Intermediate Nodes signal. As a first step, we only consider the situation that information is limited and does not need to be updated frequently. In this case, if the Head-End needs to inform the Endpoints something, it can send the information with its signature generated by using a private key. The Endpoints can check the information using the corresponding public key. For example, the public key can be obtained by the Endpoint in the authentication procedure.

The second problem is the Head-End should trust the Endpoints if it receives some path selection suggestions from the Endpoints. In this case, we think that the BNG has authenticated the Endpoints, so that the BNG can send some information to the Head-End indicating that the Endpoint is not a fake one. For example, the BNG and the Head-End can using an IPSec to transfer the traffic that needs specific treatment. Another option is that the BNG can forward the traffic that needs specific treatment with its signature generated by using a private key. The Head-End can check the information using the corresponding public key of the BNG.

The reason that we do not suggest that the Endpoints make the signature is because their number is much larger than the number of BNGs. We do not think the Head-End can handle a large number of keys. Meanwhile, in this mechanism, the Intermediate Node does not need to maintain per-connection state.

3. IANA Considerations

TBD.

4. Security Considerations

TBD.

5. Acknowledgements

TBD.

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

6.2. Informative References

[RFC9049] Dawkins, S., Ed., "Path Aware Networking: Obstacles to Deployment (A Bestiary of Roads Not Taken)", RFC 9049, DOI 10.17487/RFC9049, June 2021, <<https://www.rfc-editor.org/info/rfc9049>>.

Authors' Addresses

Zongpeng Du
China Mobile
No.32 XuanWuMen West Street
Beijing 100053
China

Email: duzongpeng@foxmail.com

Peng Liu
China Mobile
No.32 XuanWuMen West Street
Beijing 100053
China

Email: liupengyjy@chinamobile.com

PANRG
Internet-Draft
Intended status: Informational
Expires: 16 July 2022

J. Garcia-Pardo
C. Kraehenbuehl
B. Rothenberger
A. Perrig
ETH Zuerich
12 January 2022

Dynamically Recreatable Keys
draft-garciapardo-panrg-drkey-02

Abstract

DRKey is a pragmatic Internet-scale key-establishment system that allows any host to locally obtain a symmetric key to enable a remote service to perform source-address authentication, and enables first-packet authentication. The remote service can itself locally derive the same key with efficient cryptographic operations.

DRKey was developed with path aware networks in mind, but it is also applicable to today's Internet. It can be incrementally deployed and it offers incentives to the parties using it independently of its dissemination in the network.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 16 July 2022.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
1.1. Outline	3
2. Terminology	4
3. Key Derivation	5
3.1. Overview	6
3.2. Assumptions	6
3.3. Key Hierarchy	7
4. Key Establishment	8
4.1. First Level Key Establishment	8
4.2. Second or Third Level Key Establishment	10
4.3. Key Server Discovery	10
4.4. Key Expiration	11
5. Packet Authentication	11
5.1. High-Speed DNS Authentication	12
5.2. EDNS(0) Authentication Option	12
6. Deployment	12
6.1. Deployment Incentives	13
6.2. Key-Server Latency	13
6.3. Network Mobility	14
6.4. Lightning Filter System as a DRKey Deployment	14
7. Security Considerations	14
7.1. DRKey and Trust in ASes	14
7.2. Authentication within an AS	15
7.3. Adversary Model	15
8. IANA Considerations	16
Authors' Addresses	16

1. Introduction

In today's Internet, denial-of-service (DoS) attacks often use reflection and amplification techniques enabled by connectionless protocols like DNS or NTP and the possibility of source-address spoofing. The main goal of DRKey is to provide a highly efficient global first-packet authentication system. DRKey provides packet authentication at the network layer based on the network address (i.e., the IP address in the current Internet or the combination of AS number and local address in SCION), and not based on a higher-

level identity such as a domain name or web-server identity.

To obtain strong guarantees with high efficiency on a per-packet basis, an authentication system based on symmetric cryptography is required. DRKey does not rely on in-band protocols to negotiate keys, so it is able to authenticate already the first packet received from a host. DRKey also does not store the symmetric keys for all potential senders, as it would be infeasible in an Internet-scale system.

The core property achieved by DRKey is to enable a service to rapidly derive a symmetric key to perform network-address authentication for an arbitrary source host. This enables services such as DNS or NTP to instantly authenticate the first request originating from a client, thus providing a defense against reflection-based DoS attacks. The key can also be used to authenticate the payload of the request and reply, which is particularly useful for DNS which by default does not include any authentication.

The prototype system enables the server to derive the symmetric key within two AES operations, which corresponds to 18 ns on a commodity server platform, and authenticate the first packet within 85 ns on commodity hardware. Such speeds cannot be achieved with protocols based on asymmetric cryptography that require multiple messages to be exchanged to establish a shared session key. For example, DRKey outperforms RSA 1024-based source authentication by a factor of more than 220, even under the assumption that the service already knows the client's public key. In addition to providing highly efficient network address verification, DRKey can also be used to authenticate Diffie-Hellman (DH) keys in a protocol such as TCPcrypt.

1.1. Outline

The main ideas behind DRKey are as follows. Autonomous systems (ASes) can obtain certificates for their AS number and IP address range from a public-key infrastructure (PKI), i.e., SCION's control-plane PKI in a SCION deployment or the Resource Public Key Infrastructure (RPKI) in today's Internet. DRKey uses such a PKI to bootstrap its own symmetric-key infrastructure. DRKey key servers are set up in all deploying ASes and contact each other on a regular basis to set up symmetric keys between pairs of ASes. These symmetric keys are then used as a root keys to efficiently derive a hierarchy of symmetric per-host and per-service keys. The hardware implementation of the AES block cipher on most modern CPUs (Intel, AMD, ARM), allows such a key derivation in about four to seven times faster than a single DDR4 DRAM memory fetch. The approach described ensures rapid key derivation on the server side, whereas a slower key fetch is required by the client to a local key server. This one-

sidedness makes the source authentication for the receiving side as efficient as possible and ensures that DRKey does not introduce new DoS attack vectors. DRKey is incrementally deployable and provides immediate benefits to deploying entities.

A fundamental tradeoff in DRKey is the additional trust requirements of end hosts in their local AS: as the key server is able to derive the end-to-end symmetric key, this key cannot be used directly to achieve secrecy between two end hosts. However, DRKey keys can be used to authenticate that the source host indeed belongs to the claimed AS, which suffices to resolve DoS attacks.

2. Terminology

AS: Autonomous System. A one-entity managed network.

SCION: A Path-Aware inter-networking architecture.

Network Node: An entity that processes packets.

Key Server: An entity connected to the network, that contains cryptographic keys, and is able to provide such keys to their respective hosts, granted they have the required permissions.

End Host: A node in the network that executes programs in behalf of users. Users usually have full control of their end hosts.

PRF: Pseudo Random Function. Function that has a low time complexity to evaluate, but which inverse is very expensive to obtain, making it infeasible to compute. PRF may have as parameters a key and a value to which the function is applied.

DRKey Secret Value: A sequence of bytes kept in secret by the AS, inside the Key Server. The validity of the secret value is configurable per AS, and dictates the validity of other keys derived from it. The secret value is either random, or derived via a PRF from a random or secret sequence of bytes only known by the AS. Secret values are the root of the DRKey key hierarchy. A secret value for AS A is denoted as SV_A . More generally, a secret value can be bound to a standard protocol p (denoted as SV_A^p). Non-standard protocols do not have their own secret value.

DRKey Key Arrow Notation: In DRKey, level 1 and level 2 keys exist to allow the authentication of the communication between one source entity a and one destination entity b . The key is derived by one side and copied to the other. The side that derives the key is the source of the arrow in the DRKey key notation. So the key $K_{\{b \rightarrow a\}}$ denotes a key that is derived at b 's side and

obtained on a's side, independently of the flow of the packets. The source side of the arrow is also called the "fast side", and the destination, the "slow side". The fast side is typically a server, and the slow side an end host.

DRKey Level 1 Key: A key derived from a protocol bound secret value, by specifying the source and destination AS IDs of the ASes involved in the communication. The level 1 key can be derived by applying a PRF keyed on the secret value, to the identifiers of the source and destination ASes of the derivation. A level 1 key between fast side AS A and slow one AS B is denoted as $K_{\{A \rightarrow B\}}^p$ for a standard protocol "p", or $K_{\{A \rightarrow B\}}^*$ for non-standard ones.

DRKey Level 2 Key: A key derived from a level 1 key, and used to authenticate the source of packets from end-hosts to infrastructure nodes, or to further derive level 3 keys. A level 2 key is derived by applying a PRF keyed on the level 1 key to the identifiers of the source and destination of the communication. These identifiers can be the AS ID plus the IP address for the slow side, and the AS ID or the AS ID plus the IP address for the fast side of the DRKey protected communication. All level 2 keys are anchored to a protocol, identified by a string. We distinguish two possible level 2 keys, depending on the fast and slow sides of the key. (1) A level 2 key between the fast side AS A and the slow side end host Hb in AS B for standard protocol "p" is denoted as $K_{\{A \rightarrow B:Hb\}}^p$. (2) A level 2 key between the fast side endhost Ha in AS A and the slow side AS B for standard protocol "p" is denoted as $K_{\{A:Ha \rightarrow B\}}^p$. For non-standard protocols the notation is the same but replacing p with *,p.

DRKey Level 3 Key: A key derived from a level 2 host-to-AS key, used to authenticate the source of end-host to end-host packets. A level 2 key between the fast side endhost Ha in AS A and the slow side end host Hb in AS B for standard protocol "p" is denoted as $K_{\{A:Ha \rightarrow B:Hb\}}^p$. For non-standard protocols the notation is the same but replacing p with *,p.

MAC: Message Authentication Code is a sequence of bytes that authenticates and protects the integrity of a message. Modifying the sender identity or the content of the message is detected by the MAC.

3. Key Derivation

To convey an intuition of the operation of the DRKey system, a high-level overview is provided first.

3.1. Overview

The basic use case of DRKey is when a host H_a in AS A desires to communicate with a server H_b in AS B, and H_b wants to authenticate the network address of H_a using a symmetric key. ASes A and B have set up one DRKey key server each, K_{Sa} and K_{Sb} respectively. Each AS randomly selects a local secret value, SV_a and SV_b , which is only shared with trustworthy entities (in particular the key servers) in the same AS. The secret values are never shared outside the AS. The secret value will serve as the root of a symmetric-key hierarchy, where keys of a level are derived from keys of the preceding level. In DRKey, the keys are derived using a CMAC with AES, which is an efficient pseudorandom function (PRF). The key derivation used by K_{Sb} in the example is: $K_{\{B \rightarrow A\}} = \text{PRF}_{\{SV_b\}}(A)$.

Thanks to the key-secrecy property of a secure PRF, $K_{\{B \rightarrow A\}}$ can be shared with another entity without disclosing SV_b . The arrow notation indicates the secret value used to derive the key. Thus $K_{\{B \rightarrow *\}}$ would typically be used if AS B is on the performance critical side, where $*$ denotes the set of remote ASes.

To continue with the example, K_{Sa} will prefetch keys $K_{\{* \rightarrow A\}}$ from key servers in other ASes, including $K_{\{B \rightarrow A\}}$ from K_{Sb} . In the example, the server H_b is trustworthy, and can thus obtain the secret value SV_b to derive keys quickly. When H_a wants to authenticate to H_b , it contacts its local key server K_{Sa} and requests the key $K_{\{B:H_b \rightarrow A:Ha\}}$, which K_{Sa} can locally derive from $K_{\{B \rightarrow A\}}$. H_a can now directly use this symmetric key for authenticating to H_b .

The important property of DRKey is that H_b can rapidly derive $H_{\{B:H_b \rightarrow A:Ha\}}$ by using SV_b and performing two PRF operations. The one-wayness of the key-derivation function allows a key server to delegate key derivation to specific entities. The key derivation process exhibits an asymmetry, meaning that the delegated entity H_b can directly derive a required key, whereas host H_a is required to fetch the corresponding key from its local key server. As opposed to other systems that rely on a dedicated server for key generation and distribution (such as Kerberos), this delegation mechanism allows entities to directly obtain a symmetric key without communication to the key server.

3.2. Assumptions

- * There exists an AS-level PKI, that authenticates the public key of an asymmetric key pair for each participating AS E and certifies its network resources; e.g. the SCION control-plane PKI certifying AS numbers for a deployment in SCION and RPKI certifying AS numbers and IP address ranges for a deployment in today's Internet.
- * To verify the expiration time of keys and messages, DRKey relies on time synchronization among ASes with a precision on the order of several seconds. This can be achieved using a secure time-synchronization protocol such as Roughtime.
- * There exists an authentication mechanism for end hosts within an AS. This is needed for access control.

3.3. Key Hierarchy

The DRKey key-establishment framework uses a key hierarchy consisting of four levels:

- * 0th-Level (AS-internal). On the zeroth level of the hierarchy, each AS A randomly generates a local AS-specific secret value SVa . The secret value represents the per-AS basis of the key hierarchy and is renewed frequently (e.g., daily). In addition, an AS can generate protocol-specific secret values: $SVa^p = \text{PRF}_{\{SVa\}}("p")$ for a standard protocol p, where "p" is its ASCII encoding. The purpose of these values is that they can be shared with specific services, such as DNS servers, that cannot be trusted with SVa but should still be able to efficiently derive additional keys. This construction introduces additional communication and storage overhead, so only widely used protocols such as DNS or NTP would have their own secret values. Non-standard arbitrary protocols will not have their own independent secret value, and thus it won't be shareable among services. For these protocols, their level 1 keys will be derived from a special secret value denoted as SVa^* , only used for the derivation purpose.
- * 1st-Level (AS-to-AS). By using key derivation, an AS A can derive different symmetric keys using a PRF from the special local secret value SVa^* or a protocol-specific secret value SVa^p . These derived keys, which are shared between AS A and a second AS B, form the first level of the key hierarchy and are called first-level keys: $K_{\{A \rightarrow B\}}^x = \text{PRF}_{\{SVa^x\}}(B)$. The input to the PRF is the (globally unique) AS number of AS B. The value of x will be either p for standard protocols or * for arbitrary ones. The first-level keys are periodically exchanged between key servers of different ASes.

- * 2nd-Level (AS-to-host, host-to-AS). Using the symmetric keys of the first level of the hierarchy, second-level keys are derived to provide symmetric keys for authentication (AS-to-host cases) or further derivation (host-to-AS cases) into the third level keys. Second-level keys can be established between:
 - An AS as fast side, and an end-host as slow, for a standard protocol p : $K_{\{A \rightarrow B: Hb\}}^p = \text{PRF}_{\{K_{\{A \rightarrow B\}}^p\}}(0 || Hb)$
 - An end-host as fast side, and an AS as slow, for a standard protocol p : $K_{\{A: Ha \rightarrow B\}}^p = \text{PRF}_{\{K_{\{A \rightarrow B\}}^p\}}(1 || Ha)$
 - An AS as fast side, and an end-host as slow, for a non-standard, arbitrary protocol p : $K_{\{A \rightarrow B: Hb\}}^{\{*, p\}} = \text{PRF}_{\{K_{\{A \rightarrow B\}}^*\}}(0 || Hb || "p")$
 - An end-host as fast side, and an AS as slow, for a non-standard, arbitrary protocol p : $K_{\{A: Ha \rightarrow B\}}^{\{*, p\}} = \text{PRF}_{\{K_{\{A \rightarrow B\}}^*\}}(1 || Ha || "p")$
- * 3rd-Level (host-to-host). These keys are derived from the second level host-to-AS, DRKeys. Depending on the protocol type, we observe two cases:
 - Standard protocol p : the PRF is keyed on the level 2 host-to-AS drkey: $K_{\{A: Ha \rightarrow B: Hb\}}^p = \text{PRF}_{\{K_{\{A: Ha \rightarrow B\}}^p\}}(Hb)$
 - Non-standard, arbitrary protocol p : the PRF is keyed on the level 2 host-to-AS drkey: $K_{\{A: Ha \rightarrow B: Hb\}}^{\{*, p\}} = \text{PRF}_{\{K_{\{A: Ha \rightarrow B\}}^{\{*, p\}}\}}(Hb)$

4. Key Establishment

There are two types of key establishment: first level, and second or third level key establishment, depending on the level of the key in the hierarchy.

4.1. First Level Key Establishment

Key exchange is offloaded to the key servers deployed in each AS. The key servers are not only responsible for first-level key establishment, they also derive second-level keys and provide them to hosts within the same AS. To exchange a first-level key, the key servers of corresponding ASes perform the key exchange protocol. The key exchange is initialized by key server KSb by sending the request:

$\text{req} = A || B || \text{val_time} || TS || [p]$

Where TS denotes a timestamp of the current time and val_time specifies a point in time at which the requested key is valid. If an optional protocol p is supplied, the protocol-specific first-level key $K'_{\{A \rightarrow B\}^p}$ is requested, otherwise the general $K_{\{A \rightarrow B\}}$ is. The requested key may not be valid at the time of request, either because it already expired or because it will become valid in the future. For example, prefetching future keys allows for seamless transition to the new key. The request token is signed with B's private key to prove authenticity of the request.

Upon receiving the initial request, K_{Sa} checks the signature for authenticity and the timestamp for expiration. If the request has not yet expired, the key server K_{Sa} will reply with an encrypted and signed first-level key derived from the local secret value S_{Va} or, if an optional protocol p was supplied in the request, S_{Va}^p:

```
key = PRF_{SVa} (B)
or
key = PRF_{SVap} (B)
```

```
repl = {A || key}_{PK_B} || exp_time || TS
```

The term $\{A || key\}_{PK_B}$ indicates that the concatenation of A with the key is encrypted with asymmetric cryptography using B's public key. The reply token is signed with A's private key.

Once the requesting key server K_{Sb} has received the key, it shares it among other local key servers to ensure a consistent view. Each key server can now respond to queries by entities within the same AS requesting second-level keys. Alternatively, the proposed first-level key exchange protocol could also make use of TLS 1.3 with client certificates to secure the key exchange.

All first-level keys for other ASes are prefetched such that second-level keys can be derived without delay. However, on-demand key exchange between ASes is also possible. For example, in case a key server is missing a first-level key that is required for the derivation of a second-level key, the key server initiates a key exchange. ASes that contain a large number of end hosts benefit from prefetching most first-level keys, as they are likely to communicate with a large set of destination ASes. In today's Internet there exist around 68000 active ASes. Thus, setting up symmetric keys between all entities requires minor effort and state. To avoid explicit revocation, the shared keys are short-lived and new keys are established frequently (e.g., daily). Subsequent key exchanges to establish a new first-level key can use the current key as a first line of defense to avoid signature-flooding attacks.

4.2. Second or Third Level Key Establishment

End hosts request a second-level key from their local key server with the following request format:

```
format = {type, requestID, protocol, source, destination}
```

An end host H_a in AS A uses this format for issuing the following request to its local key server KS_a :

```
format || val_time || TS
```

It is assumed that this request and the reply are sent over a secure channel. Similar to the first-level key exchange, `val_time` specifies a point in time at which the requested key is valid. The key server only replies with a key to requests with a valid timestamp and only if the querying host is authorized to use the key. An authorized host must either be an end point of the communication that is authenticated using the second-level key or authorized separately by the AS.

If the end host requested a third level key, it must now be derived. It is done so from the obtained second level key.

4.3. Key Server Discovery

When a key server wants to contact another key server in a remote AS, it needs to know the IP address of the remote server.

In the SCION architecture, the concept of service addresses can be used to anycast to a key server in a specific AS.

For the regular Internet, RPKI can be used, which connects Internet resource information to a trust anchor. As the deployment numbers of RPKI are growing, the RPKI certificate can be extended with the IP address of the key server (e.g., by encoding it into the common name field or creating a separate extension). Using this mechanism, each AS publishes a list of IP addresses (or an IP anycast address) that is publicly accessible and shared by all key servers. The routing infrastructure will direct the packets to the topologically nearest key server. This mapping from an AS identifier to an IP address is verifiable through RPKI to prevent unauthorized announcements of key servers. In case RPKI has not been fully deployed, key-server discovery could also work using a DNS entry that maps a domain to IP addresses of key servers.

4.4. Key Expiration

Shared symmetric keys are short-lived (i.e., up to one day lifetime) to avoid the additional complication of explicit key revocation. However, letting all keys expire at the same time would lead to peaks in key requests. Such peaks are avoided by spreading out key expiration, which in turn leads to spreading out the fetching requests. To this end, a deterministic mapping offset $(A, B) \rightarrow [0, t)$ is introduced. This function uniformly maps the AS identifiers of the source in AS A and the destination in AS B to a range between 0 and the maximum lifetime t of SVA. This mapping is computed using a (non-cryptographic) hash function:

$$\text{offset}(A, B) = H(A \parallel B) \bmod t$$

The offset is then used to determine the validity period of a key by determining the secret value SVA^j that is used to derive $K_{\{A \rightarrow B\}}$ at the current sequence j as follows:

$$[\text{start}(\text{SVA}^j) + \text{offset}(A, B) , \text{start}(\text{SVA}^{j+1}) + \text{offset}(A, B))$$

I.e., depending on the destination B, the secret value SVA can be different, even when chosen for the same point in time.

5. Packet Authentication

The DRKey keys enable source authentication of every packet. To send DRKey source authenticated packets from end host H_a located in AS A to endhost H_b located in AS B, end host H_a first obtains the second level key $K_{\{B:H_b \rightarrow A\}}^p$ from the key server located in its AS A, KS_a . With it derives the third level key $K_{\{B:H_b \rightarrow A:H_a\}}^p$, which is used to authenticate. For a packet pkt , the source H_a then calculates the authentication tag by computing the MAC keyed on the third level key $K_{\{B:H_b \rightarrow A:H_a\}}^p$, over an immutable part of the packet pkt . This immutable part of pkt can include parts of the layer-3 and layer-4 headers, and optionally the layer-4 payload. It is important to only include immutable fields as the verification would otherwise fail at the destination.

The packet received at the destination is used to determine the source address H_a and source AS.

- * In SCION these are part of the regular header, thus no extra information is needed other than the tag itself.
- * In the current internet, 4 bytes containing the AS ID, plus the tag are added to the packet.

The destination H_b then derives or obtains the key $K_{\{B:H_b \rightarrow A:Ha\}^p}$ and uses it with the same MAC function to recalculate the authentication tag. The tag is then compared to the one present in the packet.

5.1. High-Speed DNS Authentication

A protocol specific secret value is used SV_b^p , with $p = \text{"DNS"}$. The level 1 key for a slow side A is derived directly in the DNS server:

$$K_{\{B \rightarrow A\}^p} = \text{PRF}_{\{SV_b^p\}}(A)$$

This first level key is exchanged with other AS via the level 1 key requests, as described in Section 4.1. For a DNS query from a end host H_a , located in AS A, to a DNS server located in AS B, the first level key is derived as described above, and then the second level key is derived:

$$K_{\{B \rightarrow A:Ha\}^p} = \text{PRF}_{\{K_{\{B \rightarrow A\}^p\}}}(0 \parallel H_a)$$

How to compute the authentication tag and obtain the AS ID is described in Section 5.

5.2. EDNS(0) Authentication Option

DRKey can use EDNS(0) to avoid breaking the existing DNS resolvers and authoritative servers. With a DRKey custom extension that includes the total query/response length, the source AS number, a timestamp, and the per packet MAC. The per-packet MAC for DNS queries and responses is computed the DNS header and all fields contained in the extension. Using the DRKey EDNS(0) option, packet authentication for every DNS packet introduces 28 bytes of header overhead.

6. Deployment

DRKey allows incremental deployment, as key servers could be gradually deployed in each AS. Already in the incremental deployment phase, DRKey prevents the addresses of upgraded ASes from being spoofed at other upgraded destination ASes. Early adopters can immediately profit from DRKey's security properties. Authenticating a packet requires packet modification either at the end host, or at a network appliance such as a middlebox or border router. Adding the MAC at the end host does not increase the request size en-route.

When updating end hosts is not possible in the short-term, DRKey can be implemented on a middlebox that computes a per-packet MAC and modifies applicable bypassing packets.

Packet verification at the destination AS can be performed by a middlebox as well. If a destination does not understand DRKey traffic, it could fail to process this traffic. In this case, the sending host contacts its local key server and asks if the destination AS supports DRKey. The key server might have previously derived second-level keys for an end host in the corresponding AS or might forward the query to a key server in the destination AS. If an AS supports DRKey, then it may deploy a middlebox that performs the DRKey operations in case the end host does not support it. This will prevent sending authenticated traffic to a destination host that does not support DRKey. In the worst case, the end host could fall back to legacy traffic.

In case of operational failures (e.g., a single key server fails), the end entity will try to contact another key server in the same AS. If all key servers fail, the system could fall back to the current system with unauthenticated traffic.

6.1. Deployment Incentives

Since DRKey can be deployed on commodity hardware and integrates well with the current Internet infrastructure, the deployment obstacle for DRKey is low. DRKey traffic can be recognized outside of ASes that have deployed DRKey and can thus be prioritized by servers. This means that even when relatively few companies deploy DRKey to authenticate packets at their services (e.g., popular open DNS resolvers of Google or Cloudflare), there are strong incentives for ISPs to deploy DRKey and provide its services to their customers.

6.2. Key-Server Latency

The initial connection setup depends on the latency of the connection between the client and the key server. To lower latency, DRKey's key servers should be positioned in an AS at a similar location as local DNS resolvers. As even public resolvers have an average query latency of less than 20 ms traversing the Internet, it is expected that the latency of a local key derivation will be in the order of a few ms. In most cases locally fetching a key will result in a lower latency than a full round-trip between client and server. For ASes that are geographically dispersed, multiple key servers may be deployed (e.g., co-located with an access router or per point-of-presence).

6.3. Network Mobility

Network mobility allows entities to move from one AS to another while maintaining communication sessions. In DRKey, key derivations are based on the current location of the entity in the Internet. Therefore, as soon as an entity moves to another AS, it needs to contact the key server of the new AS and fetch new second-level keys. Because local key derivation is fast and the latency of obtaining a key is small, the overhead is minimal.

6.4. Lightning Filter System as a DRKey Deployment

The Lightning Filter (LF) mechanism is a novel system that is intended to complement traditional firewalls by enabling cryptographically authenticated traffic shaping, based on the autonomous system of the source host of the traffic. This reduces significantly the load on the traditional firewall during denial-of-service attacks, and even allows LF to be the only network defense mechanism for a specific sub-network, e.g. by securing a DMZ that exposes external services to untrusted networks.

The core principle of the LF system relies on DRKey, using the high speed source authentication that DRKey enables. This way, the system can authenticate each packet, assuring that it came from the host it claimed to.

In case a breach is detected, the network administrators can immediately add the host and/or the origin AS to a blacklist, preventing packets originating there from reaching past the Lightning Filter.

7. Security Considerations

7.1. DRKey and Trust in ASes

The keys provided by DRKey do not provide full end-to-end authenticity or secrecy properties: The source and destination ASes are able to derive the keys and could thus perform an active attack. This attack is limited to these two ASes; active attacks by intermediate ASes are not possible. DRKey always enables AS-level source authentication and host-level source authentication under the additional assumption of an honest source AS.

7.2. Authentication within an AS

To achieve secrecy as well as end-host authentication for communication between end hosts and key servers, an AS needs an intra-domain end-host and/or user-authentication system. Different authentication mechanisms based on the operational environment are discussed:

- * Authentication using TLS. In order to securely exchange second-level DRKey keys between end hosts and key server, the end host can establish a secure TLS channel to the key server. The identity of the communicating parties is authenticated using public-key cryptography for both the key server and the end host. Thus, the key server can uniquely identify the end host and verify its legitimacy to obtain a second-level key.
- * Deployment in ISPs. If the corresponding AS is an ISP, we assume that they can identify their customers (e.g., terminal connection number or router that has been deployed by the ISP). In this case, only an attacker that is present at the customers local network can gain access to learn keys.
- * Company / University. For ASes that are under the control of companies or universities, login credentials or other local authentication mechanisms can be used to identify the user. This gives companies the ability to run their own web servers and have full control over their key material.
- * Mobile Devices. For mobile devices such as smart phones that are connected to the Internet through a mobile telecommunication network, clients could be authenticated by the telecom provider, for example using the International Mobile Equipment Identity (IMEI).

7.3. Adversary Model

The adversary can deviate from the protocol and attempt to violate its security goals. The Dolev-Yao model is assumed, where the adversary can reside at arbitrary locations within the network. The adversary can passively eavesdrop on messages and also actively tamper with the communication by injecting, dropping, delaying, or altering packets. However, the adversary has no efficient way of breaking cryptographic primitives such as signatures, pseudorandom functions (PRFs), and message authentication codes (MACs). It is assumed that there exists a secure channel between end hosts and a key server within the same AS.

Assuming the mentioned capabilities, the goal of the adversary is to obtain cryptographic keys of other parties to forge authenticated messages. By compromising an entity, the adversary learns all cryptographic keys and settings stored. The adversary can also control how the entity behaves, including fabrication, replay, and modification of packets. Both end hosts and network nodes compromises are considered.

8. IANA Considerations

This document has no IANA actions.

Authors' Addresses

Juan A. Garcia-Pardo
ETH Zuerich

Email: juan.garcia@inf.ethz.ch

Cyrill Kraehenbuehl
ETH Zuerich

Email: cyrill.kraehenbuehl@inf.ethz.ch

Benjamin Rothenberger
ETH Zuerich

Email: benjamin.rothenberger@inf.ethz.ch

Adrian Perrig
ETH Zuerich

Email: adrian.perrig@inf.ethz.ch

PANRG
Internet-Draft
Intended status: Informational
Expires: 8 September 2022

T. Enhardt
Netflix
C. Krähenbühl
ETH Zürich
7 March 2022

A Vocabulary of Path Properties
draft-irtf-panrg-path-properties-05

Abstract

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

Discussion Venues

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

Discussion of this document takes place on the "Path-Aware Networking Research Group" mailing list (PANRG), which is archived at <https://mailarchive.ietf.org/arch/browse/panrg/>. Subscription information is at <https://www.ietf.org/mailman/listinfo/panrg/>.

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 September 2022.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Terminology	3
2.1. Terminology usage for specific technologies	5
3. Use Cases for Path Properties	6
3.1. Path Selection	6
3.2. Protocol Selection	7
3.3. Service Invocation	7
4. Examples of Path Properties	8
5. Security Considerations	11
6. IANA Considerations	12
7. Informative References	12
Acknowledgments	14
Authors' Addresses	14

1. Introduction

The current Internet architecture does not explicitly support endpoint discovery of forwarding paths through the network as well as the discovery of properties and services associated with these paths. Path-aware networking, as defined in Section 1.1 of [I-D.irtf-panrg-questions], 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". This document provides a generic definition of path properties, addressing the first of the questions in path-aware networking [I-D.irtf-panrg-questions].

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

Note that this document does not assume that any of the listed path properties are actually available to any entity. The question of how entities can discover and distribute path properties in a trustworthy way is out of scope for this document.

2. Terminology

Entity: A physical or virtual device or function, or a collection of devices or functions, which plays a role related to path-aware networking for particular paths and flows. An entity can be on-path or off-path: On the path, an entity may participate in forwarding the flow, i.e., what may be called data plane functionality. On or off the path, an entity may influence aspects of how the flow is forwarded, i.e., what may be called control plane functionality, such as Path Selection or Service Invocation. An entity influencing forwarding aspects is usually aware of path properties, e.g., by observing or measuring them or by learning them from another entity.

Node: An on-path entity which processes packets, e.g., sends, receives, forwards, or modifies them. A node may be physical or virtual, e.g., a physical device, a service function provided as a virtual element, or even a single queue within a switch. A node may also be an entity which consists of a collection of devices or functions, e.g., an entire Autonomous System (AS).

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

Path element: Either a node or a link. For example, a path element can be an Abstract Network Element (ANE) as defined in [I-D.ietf-alto-path-vector].

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. For example, a path may be given as a sequence of physical nodes and the links connecting these nodes, or it may be given as a sequence of logical nodes such as a sequence of ASes or an Explicit Route Object (ERO). Similarly, the representation of a path and its properties, as it is known to a specific entity, may be more complex and include details about the physical layer technology, or it may be more abstract and only consist of a specific source and destination which is known to be reachable from that source.

Endpoint: The endpoints of a path are the first and the last node on the path. For example, an endpoint can be a host as defined in [RFC1122], which can be a client (e.g., a node running a web browser) or a server (e.g., a node running a web server).

Reverse Path: The path that is used by a remote node in the context of bidirectional communication.

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

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

Property: A trait of one or a sequence of path elements, or a trait

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

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

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

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

2.1. Terminology usage for specific technologies

The terminology defined in this document is intended to be general and applicable to existing and future path-aware technologies. Using this terminology, a path-aware technology can define and consider specific path elements and path properties on a specific level of abstraction. For instance, a technology may define path elements as IP routers, e.g., in source routing ([RFC1940]). Alternatively, it may consider path elements on a different layer of the Internet

Architecture ([RFC1122]) or as a collection of entities not tied to a specific layer, such as an AS or an ERO. Even within a single path-aware technology, specific definitions might differ depending on the context in which they are used. For example, the endpoints might be the communicating hosts in the context of the transport layer, ASes that contain the hosts in the context of routing, or specific applications in the context of the application layer.

3. Use Cases for Path Properties

When a path-aware network exposes path properties to endpoints or other entities, these entities may use this information to achieve different goals. This section lists several use cases for path properties.

Note that this is not an exhaustive list, as with every new technology and protocol, novel use cases may emerge, and new path properties may become relevant. Moreover, for any particular technology, entities may have visibility of and control over different path elements and path properties, and consider them on different levels of abstraction. Therefore, a new technology may implement an existing use case related to different path elements or on a different level of abstraction.

3.1. Path Selection

Nodes may be able to send flows via one (or a subset) out of multiple possible paths, and an entity may be able to influence the decision which path(s) to use. Path Selection may be feasible if there are several paths to the same destination (e.g., in case of a mobile device with two wireless interfaces, both providing a path), or if there are several destinations, and thus several paths, providing the same service (e.g., Application-Layer Traffic Optimization (ALTO) [RFC5693], an application layer peer-to-peer protocol allowing endpoints a better-than-random peer selection). Care needs to be taken when selecting paths based on path properties, as path properties that were previously measured may not be helpful in predicting current or future path properties and such path selection may lead to unintended feedback loops.

Entities may select their paths to fulfill a specific goal, e.g., related to security or performance. As an example of security-related path selection, an entity may allow or disallow sending flows over paths involving specific networks or nodes to enforce traffic policies. In an enterprise network where all traffic has to go through a specific firewall, a path-aware entity can implement this policy using path selection. As an example of performance-related path selection, an entity may prefer paths with performance

properties that best match application requirements. For example, for sending a small delay sensitive query, the entity may select a path with a short One-Way Delay, while for retrieving a large file, it may select a path with high Link Capacities on all links. Note, there may be trade-offs between path properties (e.g., One-Way Delay and Link Capacity), and entities may influence these trade-offs with their choices. As a baseline, a path selection algorithm should aim to not perform worse than the default case most of the time.

Path selection can be done either by the communicating node(s) or by other entities within the network: A network (e.g., an AS) can adjust its path selection for internal or external routing based on path properties. In BGP, the Multi Exit Discriminator (MED) attribute is used in the decision-making process to select which path to choose among those having the same AS PATH length and origin [RFC4271]; in a path-aware network, instead of using this single MED value, other properties such as Link Capacity or Link Usage could additionally be used to improve load balancing or performance [I-D.ietf-idr-performance-routing].

3.2. Protocol Selection

Before sending data over a specific path, an entity may select an appropriate protocol or configure protocol parameters depending on path properties. For example, an endpoint may cache state on whether a path allows the use of QUIC [I-D.ietf-quic-transport] and if so, first attempt to connect using QUIC before falling back to another protocol when connecting over this path again. A video streaming application may choose an (initial) video quality based on the achievable data rate or the monetary cost of sending data (e.g., volume-base or flat-rate cost model).

3.3. Service Invocation

In addition to path or protocol selection, an entity may choose to invoke additional functions in the context of Service Function Chaining [RFC7665], which may influence what nodes are on the path. For example, a 0-RTT Transport Converter [I-D.ietf-tcpm-converters] will be involved in a path only when invoked by an endpoint; such invocation will lead to the use of MPTCP or TCPinc capabilities while such use is not supported via the default forwarding path. Another example is a connection which is composed of multiple streams where each stream has specific service requirements. An endpoint may decide to invoke a given service function (e.g., transcoding) only for some streams while others are not processed by that service function.

4. Examples of Path Properties

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

Within the context of any particular technology, available path properties may differ as entities have insight into and are able to influence different path elements and path properties. For example, an endpoint may have some visibility into path elements that are on a low level of abstraction and close, e.g., individual nodes within the first few hops, or it may have visibility into path elements that are far away and/or on a higher level of abstraction, e.g., the list of ASes traversed. This visibility may depend on factors such as the physical or network distance or the existence of trust or contractual relationships between the endpoint and the path element(s). A path property can be defined relative to individual path elements, a sequence of path elements, or "end-to-end", i.e., relative to a path that comprises of two endpoints and a single virtual link connecting them.

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

Access Technology: The physical or link layer technology used for transmitting or receiving a flow on one or multiple path elements. If known, the Access Technology may be given as an abstract link type, e.g., as Wi-Fi, Wired Ethernet, or Cellular. It may also be given as a specific technology used on a link, e.g., 2G, 3G, 4G, or 5G cellular, or 802.11a, b, g, n, or ac Wi-Fi. Other path elements relevant to the access technology may include nodes related to processing packets on the physical or link layer, such as elements of a cellular backbone network. Note that there is no common registry of possible values for this property.

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

Service function: A service function that a path element applies to

a flow, see [RFC7665]. Examples of abstract service functions include firewalls, Network Address Translation (NAT), and TCP optimizers. Some stateful service functions, such as NAT, need to observe the same flow in both directions, e.g., by being an element of both the path and the reverse path.

Transparency: When a node performs an action A on a flow F, the node is transparent to F with respect to some (meta-)information M if the node performs A independently of M. M can for example be the existence of a protocol (header) in a packet or the content of a protocol header, payload, or both. A can for example be blocking packets or reading and modifying (other protocol) headers or payloads. Transparency can be modeled using a function f, which takes as input F and M and outputs the action taken by the node. If a taint analysis shows that the output of f is not tainted (impacted) by M or if the output of f is constant for arbitrary values of M, then the node is considered to be transparent. An IP router could be transparent to transport protocol headers such as TCP/UDP but not transparent to IP headers since its forwarding behavior depends on the IP headers. A firewall that only allows outgoing TCP connections by blocking all incoming TCP SYN packets regardless of their IP address is transparent to IP but not to TCP headers. Finally, a NAT that actively modifies IP and TCP/UDP headers based on their content is not transparent to either IP or TCP/UDP headers. Note that according to this definition, a node that modifies packets in accordance with the endpoints, such as a transparent HTTP proxy, as defined in [RFC2616], and a node listening and reacting to implicit or explicit signals, see [RFC8558], are not considered transparent.

Administrative Domain: The identity of an individual or an organization that owns a path element (or several path elements). Examples of administrative domains are an IGP area, an AS, or a service provider network.

Routing Domain Identifier: An identifier indicating the routing domain of a path element. Path elements in the same routing domain are in the same administrative domain and use a common routing protocol to communicate with each other. An example of a routing domain identifier is the globally unique autonomous system number (ASN) as defined in [RFC1930].

Disjointness: For a set of two paths or subpaths, the number of

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

Symmetric Path: Two paths are symmetric if the path and its reverse path consist of the same path elements on the same level of abstraction, but in reverse order. For example, a path which consists of layer 3 switches and links between them and a reverse path with the same path elements but in reverse order are considered "routing" symmetric, as the same path elements on the same level of abstraction (IP forwarding) are traversed in the opposite direction.

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

Transport Protocols available: Whether a specific transport protocol can be used to establish a connection over a path or subpath, e.g., whether the path is QUIC-capable or MPTCP-capable, based on cached knowledge.

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

Some path properties express the performance of the transmission of a packet or flow over a link or subpath. Such transmission performance properties can be measured or approximated, e.g., by endpoints or by path elements on the path, or they may be available as cost metrics, see [I-D.ietf-alto-performance-metrics]. Transmission performance properties may be made available in an aggregated form, such as averages or minimums. Properties related to a path element which constitutes a single layer 2 domain are abstracted from the used physical and link layer technology, similar to [RFC8175].

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

Link Usage: The link usage is the actual data rate at which data

that was sent over a link is correctly received at the node adjacent to the link. This property is analogous to the link usage defined in [RFC5136] but not restricted to IP-layer traffic.

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

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

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

5. Security Considerations

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

Security related properties such as confidentiality and integrity protection of payloads are difficult to characterize since they are only meaningful with respect to a threat model which depends on the use case, application, environment, and other factors. Likewise, properties for trust relations between entities cannot be meaningfully defined without a concrete threat model, and defining a threat model is out of scope for this draft. Properties related to confidentiality, integrity, and trust are orthogonal to the path terminology and path properties defined in this document. Such properties are tied to the communicating nodes and the protocols they use (e.g., client and server using HTTPS, or client and remote network node using VPN) while the path is typically oblivious to them. Intuitively, the path describes what function the network applies to packets, while confidentiality, integrity, and trust describe what function the communicating parties apply to packets.

6. IANA Considerations

This document has no IANA actions.

7. Informative References

[I-D.ietf-alto-path-vector]

Gao, K., Lee, Y., Randriamasy, S., Yang, Y. R., and J. J. Zhang, "An ALTO Extension: Path Vector", Work in Progress, Internet-Draft, draft-ietf-alto-path-vector-24, 7 March 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-alto-path-vector-24>>.

[I-D.ietf-alto-performance-metrics]

Wu, Q., Yang, Y. R., Lee, Y., Dhody, D., Randriamasy, S., and L. M. C. Murillo, "ALTO Performance Cost Metrics", Work in Progress, Internet-Draft, draft-ietf-alto-performance-metrics-26, 2 March 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-alto-performance-metrics-26>>.

[I-D.ietf-idr-performance-routing]

Xu, X., Hegde, S., Talaulikar, K., Boucadair, M., and C. Jacquenet, "Performance-based BGP Routing Mechanism", Work in Progress, Internet-Draft, draft-ietf-idr-performance-routing-03, 22 December 2020, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-performance-routing-03>>.

[I-D.ietf-quic-transport]

Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", Work in Progress, Internet-Draft, draft-ietf-quic-transport-34, 14 January 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34>>.

[I-D.ietf-tcpm-converters]

Bonaventure, O., Boucadair, M., Gundavelli, S., Seo, S., and B. Hesmans, "0-RTT TCP Convert Protocol", Work in Progress, Internet-Draft, draft-ietf-tcpm-converters-19, 22 March 2020, <<https://datatracker.ietf.org/doc/html/draft-ietf-tcpm-converters-19>>.

- [I-D.irtf-panrg-questions]
Trammell, B., "Current Open Questions in Path Aware Networking", Work in Progress, Internet-Draft, draft-irtf-panrg-questions-12, 25 January 2022,
<<https://datatracker.ietf.org/doc/html/draft-irtf-panrg-questions-12>>.
- [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/rfc/rfc1122>>.
- [RFC1930] Hawkinson, J. and T. Bates, "Guidelines for creation, selection, and registration of an Autonomous System (AS)", BCP 6, RFC 1930, DOI 10.17487/RFC1930, March 1996,
<<https://www.rfc-editor.org/rfc/rfc1930>>.
- [RFC1940] Estrin, D., Li, T., Rekhter, Y., Varadhan, K., and D. Zappala, "Source Demand Routing: Packet Format and Forwarding Specification (Version 1)", RFC 1940,
DOI 10.17487/RFC1940, May 1996,
<<https://www.rfc-editor.org/rfc/rfc1940>>.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616,
DOI 10.17487/RFC2616, June 1999,
<<https://www.rfc-editor.org/rfc/rfc2616>>.
- [RFC3357] Koodli, R. and R. Ravikanth, "One-way Loss Pattern Sample Metrics", RFC 3357, DOI 10.17487/RFC3357, August 2002,
<<https://www.rfc-editor.org/rfc/rfc3357>>.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393,
DOI 10.17487/RFC3393, November 2002,
<<https://www.rfc-editor.org/rfc/rfc3393>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271,
DOI 10.17487/RFC4271, January 2006,
<<https://www.rfc-editor.org/rfc/rfc4271>>.
- [RFC5136] Chimento, P. and J. Ishac, "Defining Network Capacity", RFC 5136, DOI 10.17487/RFC5136, February 2008,
<<https://www.rfc-editor.org/rfc/rfc5136>>.

- [RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, DOI 10.17487/RFC5693, October 2009, <<https://www.rfc-editor.org/rfc/rfc5693>>.
- [RFC6534] Duffield, N., Morton, A., and J. Sommers, "Loss Episode Metrics for IP Performance Metrics (IPPM)", RFC 6534, DOI 10.17487/RFC6534, May 2012, <<https://www.rfc-editor.org/rfc/rfc6534>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/rfc/rfc7665>>.
- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/rfc/rfc7679>>.
- [RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <<https://www.rfc-editor.org/rfc/rfc7680>>.
- [RFC8175] Ratliff, S., Jury, S., Satterwhite, D., Taylor, R., and B. Berry, "Dynamic Link Exchange Protocol (DLEP)", RFC 8175, DOI 10.17487/RFC8175, June 2017, <<https://www.rfc-editor.org/rfc/rfc8175>>.
- [RFC8558] Hardie, T., Ed., "Transport Protocol Path Signals", RFC 8558, DOI 10.17487/RFC8558, April 2019, <<https://www.rfc-editor.org/rfc/rfc8558>>.

Acknowledgments

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

Authors' Addresses

Theresa Enghardt
Netflix
Email: ietf@tenghardt.net

Cyrill Krähenbühl
ETH Zürich
Email: cyrill.kraehenbuehl@inf.ethz.ch

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

B. Trammell
Google Switzerland GmbH
25 January 2022

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

Abstract

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

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

Discussion Venues

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

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 29 July 2022.

Copyright Notice

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

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

Table of Contents

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

1. Introduction to Path-Aware Networking

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

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

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

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

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

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

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

1.1. Definitions

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

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

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

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

2. Questions

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

2.1. A Vocabulary of Path Properties

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

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

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

2.2. Discovery, Distribution, and Trustworthiness of Path Properties

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

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

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

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

2.3. Supporting Path Selection

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

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

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

2.4. Interfaces for Path Awareness

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

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

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

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

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

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

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

2.6. What is an Endpoint?

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

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

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

2.7. Operating a Path Aware Network

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

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

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

2.8. Deploying a Path Aware Network

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

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

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

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

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

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

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

3. Acknowledgments

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

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

4. Informative References

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

Author's Address

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

Email: ietf@trammell.ch

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

S. Dawkins, Ed.
Tencent America
26 March 2021

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

Abstract

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

This document contains that catalog and analysis.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 27 September 2021.

Copyright Notice

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

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

Table of Contents

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

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

1. Introduction

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

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

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

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

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

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

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

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

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

2. A Perspective On This Document

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

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

The meta-lessons from that experience were

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

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

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

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

2.1. Notes for the Reader

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

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

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

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

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

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

2.3. Venue for Discussion of this Document

(RFC Editor: please remove this section before publication)

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

2.4. Architectural Guidance

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

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

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

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

2.5. Terminology Used in this Document

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

2.6. Methodology for Contributions

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

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

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

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

3. Applying the Lessons We've Learned

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

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

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

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

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

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

Table 1

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

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

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

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

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

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

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

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

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

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

Path Aware Networking aspects of this effort, requiring research.

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

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

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

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

4. Summary of Lessons Learned

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

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

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

4.1. Justifying Deployment

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

4.2. Providing Benefits for Early Adopters

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

4.3. Providing Benefits During Partial Deployment

Some proposals require that all path elements along the full length

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

4.4. Outperforming End-to-end Protocol Mechanisms

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

4.5. Paying for Path Aware Techniques

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

4.6. Impact on Operational Practices

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

4.7. Per-connection State

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

4.8. Keeping Traffic on Fast-paths

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

4.9. Endpoints Trusting Intermediate Nodes

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

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

4.10. Intermediate Nodes Trusting Endpoints

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

4.11. Reacting to Distant Signals

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

4.12. Support in Endpoint Protocol Stacks

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

4.13. Planning For Failure

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

5. Future Work

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

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

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

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

6. Contributions

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

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

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

The suggested references for Stream Transport are:

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

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

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

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

6.1.1. Reasons for Non-deployment

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

The ST2 protocol is no longer in use.

6.1.2. Lessons Learned.

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

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

6.2. Integrated Services (IntServ)

The suggested references for IntServ are:

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

Service Network Elements [RFC2215]

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

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

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

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

6.2.1. Reasons for Non-deployment

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

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

As IntServ was being discussed, the following occurred:

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

6.2.2. Lessons Learned.

The following lessons were learned:

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

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

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

6.3. Quick-Start TCP

The suggested references for Quick-Start TCP are:

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

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

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

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

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

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

6.3.1. Reasons for Non-deployment

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

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

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

6.3.2. Lessons Learned

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

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

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

6.4. ICMP Source Quench

The suggested references for ICMP Source Quench are:

- * INTERNET CONTROL MESSAGE PROTOCOL [RFC0792]

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

6.4.1. Reasons for Non-deployment

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

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

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

6.4.2. Lessons Learned

This method had several problems:

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

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

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

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

6.5. Triggers for Transport (TRIGTRAN)

The suggested references for TRIGTRAN are:

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

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

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

6.5.1. Reasons for Non-deployment

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

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

But things got worse.

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

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

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

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

6.5.2. Lessons Learned.

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

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

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

6.6. Shim6

The suggested references for Shim6 are:

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

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

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

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

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

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

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

6.6.1. Reasons for Non-deployment

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

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

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

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

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

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

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

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

6.6.2. Lessons Learned

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

6.6.3. Addendum on MultiPath TCP

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

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

6.7. Next Steps in Signaling (NSIS)

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

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

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

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

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

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

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

6.7.1. Reasons for Non-deployment

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

* Implementation-related aspects:

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

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

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

* Operational Aspects:

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

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

6.7.2. Lessons Learned

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

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

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

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

6.8. IPv6 Flow Label

The suggested references for IPv6 Flow Label are:

- * IPv6 Flow Label Specification [RFC6437]

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

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

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

6.8.1. Reasons for Non-deployment

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

* Endpoint Implementation:

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

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

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

* Operational support in network nodes:

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

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

6.8.2. Lessons Learned

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

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

had consequences on deployability:

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

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

6.9. Explicit Congestion Notification (ECN)

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

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

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

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

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

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

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

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

6.9.1. Reasons for Non-deployment

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

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

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

ECN" at [SallyFloyd].

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

As described in [vista-impl],

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

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

6.9.2. Lessons Learned

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

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

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

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

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

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

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

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

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

7. Security Considerations

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

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

8. IANA Considerations

This document makes no requests of IANA.

9. Acknowledgments

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

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

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

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

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

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

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

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

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

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

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

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

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

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

10. Informative References

[Colossal-Cave]

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

[Conviva]

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

[GREASE]

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

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

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

[I-D.farrell-etm]

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

[I-D.ietf-quic-transport]

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

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

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

Specifications and Multiple Flow Specifications in RSVPv1", Work in Progress, Internet-Draft, draft-ietf-tsvwg-intserv-multiple-tspec-02, 25 February 2013, <<http://www.ietf.org/internet-drafts/draft-ietf-tsvwg-intserv-multiple-tspec-02.txt>>.

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

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

[I-D.irtf-panrg-questions]

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

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

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

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

[MOPS-109-Min]

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

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

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

[NSIS-CHARTER-2001]

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

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

[PANRG-103-Min]

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

[PANRG-105-Min]

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

- [PANRG-106-Min] "Path Aware Networking Research Group - IETF-106 Minutes", November 2019,
<<https://datatracker.ietf.org/doc/minutes-106-panrg/>>.
- [PANRG-110] "Path Aware Networking Research Group - IETF-110", July 2017,
<<https://datatracker.ietf.org/meeting/110/sessions/panrg>>.
- [PANRG-99] "Path Aware Networking Research Group - IETF-99", July 2017,
<<https://datatracker.ietf.org/meeting/99/sessions/panrg>>.
- [PATH-Decade] Bonaventure, O., "A Decade of Path Awareness", July 2017,
<<https://datatracker.ietf.org/doc/slides-99-panrg-a-decade-of-path-awareness/>>.
- [QS-SAT] Secchi, R., Sathiaselalan, A., Potorti, F., Gotta, A., and G. Fairhurst, "Using Quick-Start to enhance TCP-friendly rate control performance in bidirectional satellite networks", 2009,
<<https://dl.acm.org/citation.cfm?id=3160304.3160305>>.
- [QUIC-WG] "QUIC Working Group Home Page", n.d.,
<<https://datatracker.ietf.org/wg/quic/about/>>.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981,
<<https://www.rfc-editor.org/info/rfc791>>.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981,
<<https://www.rfc-editor.org/info/rfc792>>.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981,
<<https://www.rfc-editor.org/info/rfc793>>.
- [RFC1016] Prue, W. and J. Postel, "Something a Host Could Do with Source Quench: The Source Quench Introduced Delay (SQuID)", RFC 1016, DOI 10.17487/RFC1016, July 1987,
<<https://www.rfc-editor.org/info/rfc1016>>.
- [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989,
<<https://www.rfc-editor.org/info/rfc1122>>.
- [RFC1190] Topolcic, C., "Experimental Internet Stream Protocol: Version 2 (ST-II)", RFC 1190, DOI 10.17487/RFC1190, October 1990, <<https://www.rfc-editor.org/info/rfc1190>>.
- [RFC1633] Braden, R., Clark, D., and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, DOI 10.17487/RFC1633, June 1994,
<<https://www.rfc-editor.org/info/rfc1633>>.
- [RFC1809] Partridge, C., "Using the Flow Label Field in IPv6", RFC 1809, DOI 10.17487/RFC1809, June 1995,

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

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

- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<https://www.rfc-editor.org/info/rfc3168>>.
- [RFC3697] Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, "IPv6 Flow Label Specification", RFC 3697, DOI 10.17487/RFC3697, March 2004, <<https://www.rfc-editor.org/info/rfc3697>>.
- [RFC4094] Manner, J. and X. Fu, "Analysis of Existing Quality-of-Service Signaling Protocols", RFC 4094, DOI 10.17487/RFC4094, May 2005, <<https://www.rfc-editor.org/info/rfc4094>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-Start for TCP and IP", RFC 4782, DOI 10.17487/RFC4782, January 2007, <<https://www.rfc-editor.org/info/rfc4782>>.
- [RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, DOI 10.17487/RFC5082, October 2007, <<https://www.rfc-editor.org/info/rfc5082>>.
- [RFC5218] Thaler, D. and B. Aboba, "What Makes for a Successful Protocol?", RFC 5218, DOI 10.17487/RFC5218, July 2008, <<https://www.rfc-editor.org/info/rfc5218>>.
- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, DOI 10.17487/RFC5533, June 2009, <<https://www.rfc-editor.org/info/rfc5533>>.
- [RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <<https://www.rfc-editor.org/info/rfc5575>>.
- [RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <<https://www.rfc-editor.org/info/rfc5681>>.
- [RFC5971] Schulzrinne, H. and R. Hancock, "GIST: General Internet Signalling Transport", RFC 5971, DOI 10.17487/RFC5971, October 2010, <<https://www.rfc-editor.org/info/rfc5971>>.
- [RFC5973] Stiemerling, M., Tschofenig, H., Aoun, C., and E. Davies, "NAT/Firewall NSIS Signaling Layer Protocol (NSLP)", RFC 5973, DOI 10.17487/RFC5973, October 2010, <<https://www.rfc-editor.org/info/rfc5973>>.
- [RFC5974] Manner, J., Karagiannis, G., and A. McDonald, "NSIS Signaling Layer Protocol (NSLP) for Quality-of-Service Signaling", RFC 5974, DOI 10.17487/RFC5974, October 2010, <<https://www.rfc-editor.org/info/rfc5974>>.
- [RFC5981] Manner, J., Stiemerling, M., Tschofenig, H., and R. Bless,

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

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

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

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

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

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

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

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

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

Author's Address

Spencer Dawkins (editor)
Tencent America
United States of America

Email: spencerdawkins.ietf@gmail.com

Path Aware Networking Research Group
Internet-Draft
Intended status: Informational
Expires: August 26, 2021

S. Zheng
P. Liu
Z. Chen
China Mobile
February 22, 2021

Required path properties for applying path aware networking in
integrated space-terrestrial networks
draft-zheng-panrg-path-properties-istn-00

Abstract

Integrated space-terrestrial networks are heterogeneous networks with various path characteristic, and usually belong to different administrative domains. Therefore integrated space-terrestrial networks can be seen as a use case of path-aware networking. This memo introduces requirements on path properties when applying path-aware-network in integrated space-terrestrial networks.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 26, 2021.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Terminology and Abbreviation	3
3. Path properties	3
4. Fine granular properties	3
4.1. node properties	4
4.2. Link properties	4
5. Summary	5
6. Security Considerations	5
7. IANA Considerations	5
8. Normative References	6
Authors' Addresses	6

1. Introduction

In the integrated space-terrestrial networks, endpoint is capable to access space networks, mobile networks, and fixed networks. These heterogeneous networks have essential difference on characteristics and come from different service providers, which makes it difficult to carry out unified management and control. Furthermore, different with ground networks, the quality of links in space is fluctuating, the network topology changes dynamically, and the resources of space node is limited. It is necessary to come out a system to release the burden of networks (especially space nodes with limited resource) and leaving the complex function to endpoint. In other words, the path-aware network may help to cope with the dynamics of this kind of network.

According to the definition of [RFC5136], a path is a series of links that connect a series of nodes from the source node to destination. The properties of path can be seen from the overall point of view, or decomposed into node properties and link properties. Corresponding granular path awareness can be performed in the basis of the capability of the endpoint and/or the required quality of service. This memo will describe the required path properties from different granularity in integrated space-terrestrial networks.

2. Terminology and Abbreviation

Integrated space-terrestrial Networks (ISTN): A network system that comprehensively utilizes a variety of communication network technologies including space networks and terrestrial networks to achieve global coverage. The integrated system includes ground segment and space segment. The ground segment includes terrestrial network nodes such as ground stations, terminals, servers controllers and terrestrial links such as cable, fiber. Space segment includes space node such as satellites and space links such as laser and radio.

3. Path properties

The path properties describe the overall properties of the whole path from an end-to-end perspective.

Space and ground networks share some common properties, but due to the essential differences between the space network and the terrestrial network on characteristics such as mobility, link stability, resources etc., some additional properties are required to support path selection at the endpoint.

Common path properties

1. Properties in path

properties[I-D.irtf-panrg-path-properties], such as one way delay and one way packet loss.

Additional path properties in space

1. Available time: path available time; due to the topological dynamics of the space link, the path in the world-ground integrated network is not always available. Therefore, it is necessary to set an available time for each path;

4. Fine granular properties

In addition to the fluctuating latency, and bandwidth, the complex space environment will lead to unpredictable wireless link disconnection. The mobility of space nodes will lead to periodic dynamic topology change. Therefore, the performance of the path changes more frequently, and the fine granular properties can help the integrated space-terrestrial networks to quickly locate unpredictable faults and find the optimal alternative link instead of discarding the entire path. For example, path properties can be decomposed into node properties and link properties.

4.1. node properties

Common properties of nodes

1. Node computing resources: computing resources available on ground nodes/space nodes. When the available computing resource is less, it indicates that the node is heavy-loaded, and the path that contains the node should be avoided when selecting a path.

2. Node storage resources: available storage resources of ground nodes/space nodes.

Additional node properties in space

1. Node power: This is actually the most important property of space, because the energy of satellite in space comes from solar panels, which make the node energy fluctuating with time. If the power of the satellite node is not sufficient to support additional computing/communication functions, the satellite node is not available; it can be simply set to 0/1 to indicate whether the node supports additional computing/communication functions.

2. Available interfaces of the node. The interface that can be used to establish a link, it may contain a set of information indicating the direction of interface and available next hop. This property can be used to derive the topology information. The specific link status is excluded and needs to query the link properties described below.

3. The future available interfaces of the node. The movement of satellite nodes is periodic. Periodicity can be used to predict the topology in the future to help make routing decisions. This property can be sent in different manners, depending on the mechanism the system used to deal with the network mobility. This property can be sent in each time slot if the system uses snapshot. Or to reduce the interaction cost, event triggered property notification can be used, that is the notification only executes when the available interfaces change due to unexpected event.

4.2. Link properties

Common link properties

1. Propagation delay: When a data packet propagates from the source node to the destination node, the time required for the transmission from the beginning to the end of the link is the propagation delay. Data packets are propagated at the propagation rate of the link, and its rate depends on the physical medium of the link. The propagation delay is equal to the ratio of the distance between the nodes and the

propagation rate. As the distance between the nodes changes as space node moves, the delay changes as well.

2.Link media: the link media can be laser/cable/radio etc., and the different media can have different priority and cost, which should be used to do the path selection decision.

3.Quality of link: This property can be indicated by bit error rate or packet loss rate, depending on the network system.

Additional link properties in space

1. Available time: When the nodes at both ends of a link are constantly moving relative to each other, the link may be unavailable because the nodes move out of mutual visible area. Therefore, it is necessary to know the available time of the link.

2. Link status: different from bit error rate, this property indicates the state of link, for example, when the link is temporarily unavailable due to space environment, it can be set in leave and; when the link is unavailable due to mobility, it can be set to down . The link state information may not come from space node itself but from ground measurement and control station.

5. Summary

Integrated space-terrestrial Networks can take advantage of the PAN and can be seen as a typical use cases. When PAN is introduced into ISTN, it will have some different requirements on the path properties, and this memo study the first question in [I-D.irtf-panrg-questions] by list and explain some potential path properties.

6. Security Considerations

It should be noticed that under the Integrated space-terrestrial Networks background, the topology information comes from different operators, they may not willing to expose their network information to other operators or other 3rd parties, so it is crucial to find a way to supply the information to end user while not expose to others.

7. IANA Considerations

This document has no requests to IANA.

8. Normative References

- [I-D.irtf-panrg-path-properties]
Enghardt, T. and C. Krahenbuhl, "A Vocabulary of Path Properties", draft-irtf-panrg-path-properties-01 (work in progress), September 2020.
- [I-D.irtf-panrg-questions]
Trammell, B., "Current Open Questions in Path Aware Networking", draft-irtf-panrg-questions-08 (work in progress), December 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5136] Chimento, P. and J. Ishac, "Defining Network Capacity", RFC 5136, DOI 10.17487/RFC5136, February 2008, <<https://www.rfc-editor.org/info/rfc5136>>.

Authors' Addresses

Shaowen Zheng
China Mobile
Beijing 100053
China

Email: zhengshaowen@chinamobile.com

Peng Liu
China Mobile
Beijing 100053
China

Email: liupengyjy@chinamobile.com

Danyang Chen
China Mobile
Beijing 100053
China

Email: chendanyang@chinamobile.com