

QUIC Working Group
Internet-Draft
Intended status: Informational
Expires: December 4, 2021

S. Dawkins
Tencent America LLC
June 02, 2021

Path Selection for Multiple Paths In QUIC
draft-dawkins-quic-multipath-selection-01

Abstract

In QUIC working group discussions about proposals to use multiple paths, an obvious question came up - given multiple paths, how does QUIC select paths to send packets over?

The answer to that question may inform decisions in the QUIC working group about the scope of any multipath extensions considered for experimentation and adoption.

This document is intended to summarize aspects of path selection from those contributions and conversations.

It is recognized that path selection is not the only important open question about QUIC Multipath, but other open questions are out of scope for this document.

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 December 4, 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
1.1. Why We Should Look at Path Selection Strategies Now	3
1.2. Notes for Readers	4
1.3. Minimal Terminology	4
1.4. Contribution and Discussion Venues for this draft.	5
2. Background for this document	5
3. Overview of Proposed Path Selection Strategies	6
3.1. Active-Standby	7
3.2. Latency Versus Bandwidth	7
3.3. Bandwidth Aggregation/Load Balancing	7
3.4. Minimum RTT Difference	7
3.5. Round-Trip-Time Thresholds	7
3.6. RTT Equivalence	7
3.7. Priority-based	7
3.8. Redundant	8
3.9. Control Plane Versus Data Plane	8
3.10. Combinations of Strategies	8
4. Implications for QUIC Multipath	8
4.1. Selecting a Single Path Among Multiple Validated Paths ("Traffic Switching")	8
4.2. Selecting Multiple Active Paths ("Traffic Splitting")	9
4.3. Arbitrary Combinations	9
5. Next Steps	10
6. IANA Considerations	10
7. Security Considerations	10
8. Acknowledgments	10
9. Informative References	10
Author's Address	13

1. Introduction

In QUIC working group [QUIC-charter] discussions about proposals to use multiple paths, an obvious question came up – given multiple paths, how does QUIC select paths to send packets over?

The answer to that question may inform decisions in the QUIC working group about the scope of any multipath extensions considered for experimentation and adoption.

This document is intended to summarize aspects of path selection from those contributions and conversations.

It is recognized that path selection is not the only important open question about QUIC Multipath, but other open questions are out of scope for this document.

1.1. Why We Should Look at Path Selection Strategies Now

One of the first questions that's come up in discussions about multiple paths for QUIC has been about path selection. As soon as an implementation has multiple paths available, it must decide how to use these paths. The [RFC9000] answer, relying on connection migration, is "if you have multiple paths available, you can validate more than one connection at a time, but you only send on one connection at a time, and you migrate to another connection when you decide sending on the current connection is no longer appropriate. How you decide to migrate to another connection is up to you".

That has been a fine answer for many of the implementers who have worked on the first version of QUIC, and have deployed it in their networks. For other implementers, targeting other use cases and other networking environments, it may not be sufficient.

To take only one example, one of several presentations at [QUIC-interim-20-10] described aspects of 3GPP Access Traffic Steering, Switch and Splitting support (ATSSS), which contained four "Steering Modes" as part of Rel-16 in 2019 [TS23501], each of which corresponding roughly to a path selection strategy described in Section 3 of this document. A study on "ATSSS Phase 2" [TR23700-93] included four more Steering Modes for Rel-17, expected to be finalized in mid-2021, and none of these eight (so far) Steering Modes are based on QUIC - they are based on Multipath TCP ([RFC8684] or simple IP packet forwarding. And if that were not enough, a proposal for a study on "ATSSS Phase 3" [S2-2104599] was provided to the SA2 145-e meeting in May 2021. Some of the ATSSS strategies rely in 5G network internals and don't translate to the broader Internet, but most do translate, and 3GPP participants certainly aren't the only people thinking about path selection strategies.

Since the various proposals presented at [QUIC-interim-20-10] were developed in isolation from each other, the path selection strategies that they reflect may be similar between proposals, but not quite the

same, and none of the proposals presented had more than two strategies in common with any other proposal.

Given the number of path selection strategies being considered, implemented, and even deployed in any number of venues, and the potential for combinatorial explosion as described in Section 3.10, it seems that identifying common aspects of path selection strategies, sooner rather than later, is important.

1.2. Notes for Readers

This document is an informational Internet-Draft, not adopted by any IETF working group, and does not carry any special status within the IETF.

Please note well that this document reflects the author's current understanding of past working group discussions and proposals. Contributions that add or improve considerations are welcomed, as described in Section 1.4.

1.3. Minimal Terminology

In this document, "QUIC multipath" is only used as shorthand for "QUIC using multiple paths". It does not refer to a specific proposal.

In this document, "path selection strategy" means the policy that a QUIC sender uses to guide its choice between multiple interfaces of a QUIC connection for "the next packet".

This document adopts three terms, stolen from [TS23501], that seemed helpful in previous discussions about multipath in the QUIC working group.

- o Traffic Steering - selecting an initial path (in [RFC9000], this would be "validating a connection, and then using it". Although an [RFC9000] client can validate multiple connections, the client will only use one validated connection at a time.
- o Traffic Switching - selecting a different validated path (in [RFC9000], this is something like "migrating to a new validated connection", although whether connection migration as defined in [RFC9000]) would be sufficient is discussed in Section 4).
- o Traffic Splitting - using multiple validated paths simultaneously (this would almost certainly require an extension beyond connection migration as defined in [RFC9000]).

"Traffic Steering" does not require any extension to [RFC9000], and is not discussed further in this document. The focus will be on "Traffic Switching" and "Traffic Splitting".

1.4. Contribution and Discussion Venues for this draft.

(Note to RFC Editor - if this document ever reaches you, please remove this section)

This document is under development in the Github repository at <https://github.com/SpencerDawkins/quic-multipath-selection>.

Readers are invited to open issues and send pull requests with contributed text for this document, but since the document is intended to guide discussion for the QUIC working group, substantial discussion of this document should take place on the QUIC working group mailing list (quic@ietf.org). Mailing list subscription and archive details are at <https://www.ietf.org/mailman/listinfo/quic>.

2. Background for this document

A number of individual draft proposals for "QUIC over multiple paths" have been submitted to the IETF QUIC and INTAREA working groups, dating back as far as 2017. The author thinks that the complete list is as follows (and reminders for proposals he missed are welcomed):

- o [I-D.an-multipath-quic]
- o [I-D.an-multipath-quic-application-policy]
- o [I-D.an-multipath-quic-traffic-distribution]
- o [I-D.chan-quic-owl]
- o [I-D.deconinck-quic-multipath]
- o [I-D.deconinck-quic-multipath],
- o [I-D.huitema-quic-mpath-req]
- o [I-D.huitema-quic-mpath-option])
- o [I-D.liu-multipath-quic]
- o [I-D.piraux-intarea-quic-tunnel-session]

[I-D.bonaventure-iccr-g-schedulers] has also been submitted to the Internet Congestion Control Research Group [ICCRG-charter] in the

Internet Research Task Force. It contains specific proposals for implementing some multipath schedulers, and includes some discussion of path selection relevant to this document.

One point of confusion in QUIC working group discussions was that the various proposals (also using Multipath TCP [RFC8684], so not all proposals were QUIC-specific) discussed in working group meetings and on the QUIC mailing list were from various proponents who weren't solving the same problem. This meant that no two of the use cases presented at the QUIC working group virtual interim on QUIC Multipath [QUIC-interim-20-10] were relying on the same strategies.

It seemed useful to collect the path selection strategies described in those proposals, to look for common elements, and to write them down in one place, to allow more focused discussion. [I-D.dawkins-quic-what-to-do-with-multipath] was intended to summarize, at a high level, various proposals for the use of multipath capabilities in QUIC, both inside the IETF and outside the IETF, in order to identify elements that were common across proposals. This draft tries to describe the impact of these various strategies on potential QUIC Multipath extensions.

One element that is certainly worth considering is whether the path selection strategies that have been proposed can be satisfied using a small number of "building block" strategies.

3. Overview of Proposed Path Selection Strategies

The following strategies were discussed at [QUIC-interim-20-10], and afterwards on the QUIC mailing list. These are summarized in this section, are described in more detail in [I-D.dawkins-quic-what-to-do-with-multipath], and are attributed to various proposals in that document.

- o Active-Standby - described in Section 3.1
- o Latency Versus Bandwidth - described in Section 3.2
- o Bandwidth Aggregation/Load Balancing - described in Section 3.3
- o Minimum RTT Difference - described in Section 3.4
- o Round-Trip-Time Thresholds - described in Section 3.5
- o RTT Equivalence - described in Section 3.6
- o Priority-based - described in Section 3.7

- o Redundant - described in Section 3.8
- o Control Plane Versus Data Plane - described in Section 3.9
- o Combinations of Strategies - described in Section 3.10

3.1. Active-Standby

The traffic associated with a specific flow will be sent via a specific path (the 'active path') and switched to another path (called 'standby path') when the active access is unavailable.

3.2. Latency Versus Bandwidth

Some traffic might be sent over a network path with lower latency and other traffic might be sent over a different network path with higher bandwidth.

3.3. Bandwidth Aggregation/Load Balancing

Traffic is sent using all available paths simultaneously, so that all available bandwidth is utilized, likely based on something like weighted round-robin path selection. This strategy is often used for bulk transfers.

3.4. Minimum RTT Difference

Traffic is sent over the path with the lowest smoothed RTT among all available paths, in order to minimize latency for latency-sensitive flows.

3.5. Round-Trip-Time Thresholds

Traffic is sent over the first path with a smoothed RTT that meets a certain threshold.

3.6. RTT Equivalence

When multiple paths each have sufficiently similar smoothed RTTs, traffic is sent over all of these paths. This is similar to "Bandwidth Aggregation/Load Balancing", with the additional qualification that not all available paths are used for this traffic.

3.7. Priority-based

Priorities are assigned to each path (often by association with network interfaces). Traffic is sent on a highest-priority path

until it becomes congested, and then "overflows" onto a lower-priority path.

3.8. Redundant

Traffic is replicated over two or more paths. This strategy could be used continuously, but is more commonly used when measured network conditions indicate that redundant sending may be necessary to increase the likelihood that at least one copy of each packet will arrive at the receiver.

3.9. Control Plane Versus Data Plane

An application might stream media over one or more available paths (based on one of the other strategies named in this section), but might send ACK traffic or retransmission over a path specifically chosen for that purpose. This is more likely to be beneficial if the path characteristics differ significantly between available paths - for example, satellite uplink/downlink stations connected by both higher-bandwidth Low Earth Orbit satellite paths and lower-bandwidth cellular or landline paths.

3.10. Combinations of Strategies

In addition to the strategies described above, it is also possible to combine these strategies. For example, a scheduler might use load-balancing over three paths, but when one of the paths becomes unavailable, the scheduler might switch to the two paths that are still available, in a way similar to Active-Standby. This is very much an example chosen at random - potentially, there are many combinations that could be useful.

4. Implications for QUIC Multipath

This section summarizes potential implications for "Multipath QUIC" of path selection strategies described in Section 3, dividing them between "Traffic Switching" (Section 4.1) and "Traffic Splitting" (Section 4.2).

4.1. Selecting a Single Path Among Multiple Validated Paths ("Traffic Switching")

If a sender using Active-Standby (described in Section 3.1) does not perform frequent path switching, it can likely be supported using connection migration as defined in [RFC9000] without change.

- o The caveat here is that connection migration can include the also-implicit assumption that an endpoint can free up resources

associated with the previously-active path. If connection migration happens often enough, the endpoint may spend considerable time "thrashing" between allocating resources and quickly freeing them. Of course, if a sender is frequently selecting a new path for connection migration, this probably degenerates into one of the other path selection strategies.

Some path selection strategies could be supported by a mechanism as simple as the one proposed in [I-D.huitema-quic-mpath-option], which replaces "the implicit signaling of path migration through data transmission, by means of a new PATH_OPTION frame" (this isn't intended to imply the proposal is simple, only the explicit signaling), if the receiver uses this option to notify the sender of the preferred path. For example, Minimum RTT Difference (described in Section 3.4) and Round-Trip-Time Thresholds (described in Section 3.5) likely fall into this category.

Some path selection strategies are exploiting a relatively long-lived difference between paths - for example, Latency Versus Bandwidth (described in Section 3.2), Priority-based (described in Section 3.7), and Control Plane Versus Data Plane (described in Section 3.9) may fall into this category. One might wonder why these senders would need to use a single "multipath connection", rather than multiple [RFC9000] connections, for these cases, but if there is a reason to use a single multipath connection, a mechanism similar to the one proposed in [I-D.huitema-quic-mpath-option] could also be used in these cases.

4.2. Selecting Multiple Active Paths ("Traffic Splitting")

Some path selection strategies are treating more than one path as a set of active paths, whether the sender is performing "Traffic Splitting" (as defined in Section 1.3)), as is the case for Bandwidth Aggregation/Load Balancing (described in Section 3.3) and RTT Equivalence (described in Section 3.6), or simply transmitting the same packet across multiple paths, as is the case for Redundant (described in Section 3.8).

For these cases, a more complex mechanism is likely required.

4.3. Arbitrary Combinations

Because it is simple enough to imagine various combinations of strategies (as described in Section 3.10), it seems important to understand what basic building blocks are required in order to support the strategies that seem common across a variety of use cases, because interactions between strategies may have significant

implications for QUIC Multipath that might not arise when considering strategies in isolation.

This seems especially important because existing proposals for QUIC Multipath don't use the same vocabulary to describe path selection strategies, so implementations may not behave in the same way, even if they are each using a strategy that seems to be common.

5. Next Steps

If this discussion is useful, it may also be useful to take the next step, and identify potential building blocks that can be used to construct the path selection strategies described in Section 4.1 and Section 4.2.

6. IANA Considerations

This document does not make any request to IANA.

7. Security Considerations

QUIC-specific security considerations are discussed in Section 21 of [RFC9000].

Section 6 of [I-D.ietf-quic-datagram] discusses security considerations specific to the use of the Unreliable Datagram Extension to QUIC.

Some "Multipath QUIC"-specific security considerations can be found in the corresponding section of [I-D.deconinck-quic-multipath].

Having said that, it may be best to repeat the security considerations section from [I-D.huitema-quic-mpath-option]: "TBD. There are probably ways to abuse this."

8. Acknowledgments

Your name could appear here. Please comment and contribute, as per Section 1.4.

9. Informative References

[I-D.an-multipath-quic]

An, Q., Liu, Y., Ma, Y., and Z. Li, "Multipath Extension for QUIC", draft-an-multipath-quic-00 (work in progress), October 2020.

- [I-D.an-multipath-quic-application-policy]
An, Q., Li, Z., and Y. Liu, "Enabling application policy-awareness in Multipath QUIC", draft-an-multipath-quic-application-policy-00 (work in progress), July 2020.
- [I-D.an-multipath-quic-traffic-distribution]
An, Q., Liu, D., and Y. Liu, "Key Components for Multipath QUIC Traffic Distribution", draft-an-multipath-quic-traffic-distribution-00 (work in progress), March 2020.
- [I-D.bonaventure-iccrq-schedulers]
Bonaventure, O., Piraux, M., Coninck, Q. D., Baerts, M., Paasch, C., and M. Amend, "Multipath schedulers", draft-bonaventure-iccrq-schedulers-01 (work in progress), September 2020.
- [I-D.chan-quic-owl]
Chan, H. A., Wei, A., Song, F., and H. Zhang, "One Way Latency Considerations for Multipath in QUIC", draft-chan-quic-owl-01 (work in progress), March 2017.
- [I-D.dawkins-quic-what-to-do-with-multipath]
Dawkins, S., "What To Do With Multiple Active Paths in QUIC", draft-dawkins-quic-what-to-do-with-multipath-03 (work in progress), January 2021.
- [I-D.deconinck-quic-multipath]
Coninck, Q. D. and O. Bonaventure, "Multipath Extensions for QUIC (MP-QUIC)", draft-deconinck-quic-multipath-07 (work in progress), May 2021.
- [I-D.huitema-quic-mpath-option]
Huitema, C., "QUIC Multipath Negotiation Option", draft-huitema-quic-mpath-option-00 (work in progress), October 2020.
- [I-D.huitema-quic-mpath-req]
Huitema, C., "QUIC Multipath Requirements", draft-huitema-quic-mpath-req-01 (work in progress), January 2018.
- [I-D.ietf-quic-datagram]
Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", draft-ietf-quic-datagram-02 (work in progress), February 2021.

- [I-D.liu-multipath-quic]
Liu, Y., Ma, Y., Huitema, C., An, Q., and Z. Li,
"Multipath Extension for QUIC", draft-liu-multipath-
quic-03 (work in progress), March 2021.
- [I-D.piraux-intarea-quic-tunnel-session]
Piraux, M., Bonaventure, O., and A. Masputra, "Session
mode for multiple QUIC Tunnels", draft-piraux-intarea-
quic-tunnel-session-00 (work in progress), November 2020.
- [ICCRG-charter]
"IRTF Internet Congestion Control Research Group Charter",
n.d., <<https://datatracker.ietf.org/rg/iccr/about/>>.
- [QUIC-charter]
"IETF QUIC Working Group Charter", n.d.,
<<https://datatracker.ietf.org/wg/quic/about/>>.
- [QUIC-interim-20-10]
"IETF QUIC Working Group Virtual Interim Meeting - October
2020", October 2020, <<https://github.com/quicwg/wg-materials/tree/master/interim-20-10>>.
- [RFC8684] Ford, A., Raiciu, C., Handley, M., Bonaventure, O., and C.
Paasch, "TCP Extensions for Multipath Operation with
Multiple Addresses", RFC 8684, DOI 10.17487/RFC8684, March
2020, <<https://www.rfc-editor.org/info/rfc8684>>.
- [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based
Multiplexed and Secure Transport", RFC 9000,
DOI 10.17487/RFC9000, May 2021,
<<https://www.rfc-editor.org/info/rfc9000>>.
- [S2-2104599]
Lenovo, Motorola Mobility, ., "Study on Access Traffic
Steering, Switching and Splitting support in the 5G system
architecture; Phase 3", 2021,
<[https://www.3gpp.org/ftp/tsg_sa/WG2_Arch/
TSGS2_145E_Electronic_2021-05/Docs/S2-2104599.zip](https://www.3gpp.org/ftp/tsg_sa/WG2_Arch/TSGS2_145E_Electronic_2021-05/Docs/S2-2104599.zip)>.
- [TR23700-93]
3GPP (3rd Generation Partnership Project), ., "Technical
Specification Group Services and System Aspects; Study on
access traffic steering, switch and splitting support in
the 5G System (5GS) architecture; Phase 2 (Release 17)",
2021, <[https://www.3gpp.org/ftp/Specs/
archive/23_series/23.700-93/](https://www.3gpp.org/ftp/Specs/archive/23_series/23.700-93/)>.

[TS23501] 3GPP (3rd Generation Partnership Project), ., "Technical Specification Group Services and System Aspects; System Architecture for the 5G System; Stage 2 (Release 16)", 2019, <https://www.3gpp.org/ftp/Specs/archive/23_series/23.501/>.

Author's Address

Spencer Dawkins
Tencent America LLC

Email: spencerdawkins.ietf@gmail.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 Ha , 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:

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

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 Enhardt
Netflix
Email: ietf@tenhardt.net

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

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 28 April 2022

T. Jones
G. Fairhurst
University of Aberdeen
N. Kuhn
CNES
J. Border
Hughes Network Systems, LLC
E. Stephan
Orange
25 October 2021

Enhancing Transport Protocols over Satellite Networks
draft-jones-tsvwg-transport-for-satellite-02

Abstract

IETF transport protocols such as TCP, SCTP and QUIC are designed to function correctly over any network path. This includes networks paths that utilise a satellite link or network. While transport protocols function, the characteristics of satellite networks can impact performance when using the defaults in standard mechanisms, due to the specific characteristics of these paths.

[RFC2488] and [RFC3135] describe mechanisms that enable TCP to more effectively utilize the available capacity of a network path that includes a satellite system. Since publication, both application and transport layers and satellite systems have evolved. Indeed, the development of encrypted protocols such as QUIC challenges currently deployed solutions, for satellite systems the capacity has increased and commercial systems are now available that use a range of satellite orbital positions.

This document follows the terminology proposed in [I-D.irtf-panrg-path-properties] to describe the current characterises of common satellite paths. This document also describes considerations when implementing and deploying reliable transport protocols that are intended to work efficiently over paths that include a satellite system. It discusses available network mitigations and offers advice to designers of protocols and operators of satellite networks.

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 28 April 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	4
2. SATCOM terminology	5
3. Satellite Systems	6
3.1. Geosynchronous Earth Orbit (GEO)	7
3.2. Low Earth Orbit (LEO)	8
3.3. Medium Earth Orbit (MEO)	8
3.4. Hybrid Network Paths	9
4. Satellite System Characteristics	9
4.1. Impact of delay	11
4.1.1. Larger Bandwidth Delay Product	11
4.1.2. Variable Link Delay	12
4.1.3. Impact of delay on protocol feedback	12
4.2. Intermittent connectivity	12
5. On-Path Mitigations	12
5.1. Link-Level Forward Error Correction and ARQ	12
5.2. PMTU Discovery	12
5.3. Quality of Service (QoS)	13
5.4. Split-TCP PEP	13
5.5. Application Proxies	14
6. Generic Transport Protocol Mechanisms	14

6.1.	Transport Initialization	15
6.2.	Getting up to Speed	15
6.3.	Sizing of Maxium Congestion Window	15
6.4.	Reliability (Loss Recovery/Repair)	16
6.4.1.	Packet Level Forward Error Correction	16
6.5.	Flow Control	16
6.6.	ACK Traffic Reduction	17
6.7.	Multi-Path	17
7.	Protocol Specific Mechanisms	18
7.1.	TCP Protocol Mechanisms	18
7.1.1.	Transport Initialization	18
7.1.2.	Getting Up To Speed	18
7.1.3.	Size of Windows	18
7.1.4.	Reliability	18
7.1.5.	ACK Reduction	18
7.2.	QUIC Protocol Mechanisms	18
7.2.1.	Transport initialization	18
7.2.2.	Getting up to Speed	18
7.2.3.	Size of Windows	18
7.2.4.	Reliability	18
7.2.5.	Asymmetry	18
7.2.6.	Packet Level Forward Error Correction	19
7.2.7.	Split Congestion Control	19
8.	Discussion	19
8.1.	Mitigation Summary	19
9.	Acknowledgments	20
10.	Security Considerations	20
11.	Informative References	20
Appendix A.	Example Network Profiles	23
A.1.	LEO	23
A.2.	MEO	23
A.3.	GEO	23
A.3.1.	Small public satellite broadband access	24
A.3.2.	Medium public satellite broadband access	24
A.3.3.	Congested medium public satellite broadband access	25
A.3.4.	Variable medium public satellite broadband access	26
A.3.5.	Loss-free large public satellite broadband access	26
A.3.6.	Lossy large public satellite broadband access	27
Appendix B.	Revision Notes	28
Authors' Addresses	28

1. Introduction

Satellite communications (SATCOM) systems have long been used to support point-to-point links and specialised networks. The predominate current use today is to support Internet Protocols. Typical example applications include: use as an access technology for remote locations, backup and rapid deployment of new services, transit networks, backhaul of various types of IP networks, and provision to mobile environments (maritime, aircraft, etc.).

In most scenarios, the satellite IP network segment forms only one part of the end-to-end path used by an Internet transport protocol. This means that user traffic can experience a path that includes a satellite network combined with a wide variety of other network technologies (Ethernet, cable modems, WiFi, cellular, radio links, etc). Although a user can sometimes know the presence of a satellite service, a typical user does not deploy special software or applications when a satellite network is being used. Users are often unaware of the technologies underpinning the links forming a network path.

Satellite path characteristics have an effect on the operation of transport protocols, such as TCP, SCTP or QUIC. Transport Protocol performance can be affected by the magnitude and variability of the network delay. When transport protocols perform poorly the link utilization can be low. Techniques and recommendations have been made that can improve the performance of transport protocols when the path includes as satellite network.

The end-to-end performance of an application using an Internet path can be impacted by the path characteristics, such as the Bandwidth-Delay Product (BDP) of the links and network devices forming the path. It can also be impacted by underlying mechanisms used to manage the radio resources.

Performance can be impacted at several layers. For instance, the page load time for a complex page can be much larger when a path includes a satellite system. A significant contribution to the reduced performance arises from the initialisation and design of transport mechanisms.

Although mechanisms are designed for use across Internet paths, not all designs are performant when used over the wide diversity of path characteristics that can occur. This document therefore considers the implications of Internet paths that include a satellite system. The analysis and conclusions might also apply to other network systems that also result in characteristics that differ from typical Internet paths.

RFC2488 specifies an Internet Best Current Practices for the Internet Community, relating to use of the standards-track Transmission Control Protocol (TCP) mechanisms over satellite channels [RFC2488]. A separate RFC, [RFC2760], identified research issues and proposed mitigations for satellite paths.

Since the publication of these RFCs many TCP mechanisms have become widely used. In particular, this includes a series of mitigation based on Performance Enhancing Proxies (PEPs) [RFC3135] that split the protocol at the transport layer. Although PEPs are now a common component of satellite systems, their use slows the deployment of new transport protocols and mechanisms (each of which demands an update to the PEP functionality). This has made it difficult for new protocol extensions to achieve comparable performance over satellite channels. In addition, protocols with strong requirements on authentication and privacy such as QUIC [RFC9000] are not able to be split using a PEP and mitigation, and need to therefore use other methods.

XXX Note from the current authors: This document currently focuses on Geosynchronous Earth Orbit (GEO) satellite systems, the authors solicit feedback and experience from users and operators of satellite systems using other orbits. XXX

2. SATCOM terminology

This section follows the terminology proposed in [I-D.irtf-panrg-path-properties] to describe a generic SATCOM system for broadband access. This description is inline with the one proposed in [RFC8975].

A generic SATCOM system could contain the following entities:

- * A: A Host providing the end service (e.g. web server);
- * B: A Node being the point-of-presence for the SATCOM system;
- * C: A Node gathering network functions (e.g. firewall, PEP, caching services, etc.);
- * D: A Node gathering MAC and PHY functionalities (a.k.a. the satellite gateway);
- * E: A Node being one of the satellite (if there are several satellites) (this node could include network layer functions);
- * F: A Node receiving the signal from the satellite (a.k.a. the satellite terminal);

- * G: A Host providing the end service (e.g. web browser).

These entities would be interconnected with path elements which properties differ from one SATCOM system to another. [I-D.irtf-panrg-path-properties] provides properties that can be discussed to describe the path. These properties are exploited throughout the whole document to describe SATCOM systems.

While the paths interconnecting the entities (1) A to B, (2) B to C, (3) C to D and (4) F to G are quite generic for all the systems, and not specific for SATCOM systems, some properties need to be discussed:

- * Protocol Features available
- * Transport Protocols available
- * Transparency

The paths (1) D to E and (2) D to F are quite specific to SATCOM systems. In particular, the following elements, provided by [I-D.irtf-panrg-path-properties], are in the scope of this document and deserve some description:

- * Symmetric Path
- * Disjointness
- * Transparency
- * Link Capacity
- * Link Usage
- * One-Way Delay
- * One-Way Delay Variation
- * One-Way Packet Loss

3. Satellite Systems

Satellite communications systems have been deployed using many space orbits, including low earth orbit, medium earth orbits, geosynchronous orbits, elliptical orbits and more. This document considers the characteristics of all satellite networks.

- * Many communications satellites are located at Geostationary Orbit (GEO) with an altitude of approximately 36,000 km [Sta94]. At this altitude the orbit period is the same as the Earth's rotation period. Therefore, each ground station is always able to "see" the orbiting satellite at the same position in the sky. The propagation time for a radio signal to travel twice that distance (corresponding to a ground station directly below the satellite) is 239.6 milliseconds (ms) [Mar78]. For ground stations at the edge of the view area of the satellite, the distance traveled is $2 \times 41,756$ km for a total propagation delay of 279.0 ms [Mar78]. These delays are for one ground station-to-satellite-to-ground station route (or "hop"). Therefore, the propagation delay for a packet and the corresponding reply (one round-trip time or RTT) could be at least 558 ms. The RTT is not based solely due to satellite propagation time and will be increased by other factors, such as the serialisation time, including any FEC encoding/ARQ delay and propagation time of other links along the network path and the queueing delay in network equipment. The delay is increased when multiple hops are used (i.e. communications is relayed via a gateway) or if inter-satellite links are used. As satellites become more complex and include on-board processing of signals, additional delay can be added.
- * Communications satellites can also be built to use a Low Earth Orbit (LEO) [Stu95] [Mon98]. The lower orbits require the use of constellations of satellites for constant coverage. In other words, as one satellite leaves the ground station's sight, another satellite appears on the horizon and the channel is switched to it. The propagation delay to a LEO orbit ranges from several milliseconds when communicating with a satellite directly overhead, to as much as 20 ms when the satellite is on the horizon. Some of these systems use inter-satellite links and have variable path delay depending on routing through the network.
- * Another orbital position use a Medium Earth Orbit (MEO) [Mar78]. These orbits lie between LEO and GEO.

3.1. Geosynchronous Earth Orbit (GEO)

The characteristics of systems using Geosynchronous Earth Orbit (GEO) satellites differ from paths only using terrestrial links in their path characteristics:

- * A large propagation delay of at least 250ms one-way delay;
- * Use of radio resource management (often using techniques similar to cellular mobile or DOCSIS cable networks, but differ to accommodate the satellite propagation delay);

- * Links can be highly asymmetric (in terms of capacity, one-way delay and in their cost of operation, see Appendix A for example scenarios).

As an example. GEO systems use the DVB-S2 specifications [EN 302 307-1], published by the European Telecommunications Standards Institute (ETSI), where the key concept is to ensure both a good usage of the satellite resource and a Quasi Error Free (QEF) link. These systems typically monitor the link quality in real-time, with the help of known symbol sequences, included along with regular packets, which enable an estimation of the current signal-to-noise ratio. This estimation is then feedback allowing the transmitting link to adapt its coding rate and modulation to the actual transmission conditions.

3.2. Low Earth Orbit (LEO)

There is an important variability of LEO systems. Depending on the locations of the gateways on the ground, routing within the constellation may be necessary to bring to packets down to the ground. Depending on the routes currently available for an end user, high levels of jitter may occur (from 40ms to 140ms with the Iridium constellation). This may lead to out-of-order delivery of packets.

XXX The authors solicit feedback and experience from users and operators of satellite systems in LEO orbits. XXX

3.3. Medium Earth Orbit (MEO)

MEO systems such as O3B combines advantages and drawbacks from both LEO and GEO systems.

MEO systems can have a large coverage and with limited number of satellites required providing a broad service. The usage of powerful satellites enables provision of high data rates.

MEO systems have the drawback, from a transport protocol perspective, that the BDP can be very high due to the altitude of such constellations (8 063 km for O3B) and there may be delay variations when the satellite changes (every 45 minutes with O3B). The latter can be dealt with by means of double antennas terminals.

XXX The authors solicit feedback and experience from users and operators of satellite systems in MEO orbits. XXX

3.4. Hybrid Network Paths

XXX The authors solicit feedback and experience from users and operators of satellite systems in hybrid network scenarios. XXX

4. Satellite System Characteristics

There is an inherent delay in the delivery of a packet over a satellite system due to the finite speed of light and the altitude of communications satellites.

Satellite links are dominated by two fundamental characteristics, as described below:

- * **Packet Loss:** The strength of any radio signal falls in proportion to the square of the distance traveled. For a satellite link the square of the distance traveled is large and so the signal becomes weak before reaching its destination. This results in a low signal-to-noise ratio. Some frequencies are particularly susceptible to atmospheric effects such as rain attenuation. For mobile applications, satellite channels are especially susceptible to multi-path distortion and shadowing (e.g., blockage by buildings). Typical bit error ratios (BER) for a satellite link today are on the order of 1 error per 10 million bits (1×10^{-7}) or less frequent. Advanced error control coding (e.g., Reed Solomon or LDPC) can be added to existing satellite services and is currently being used by many services. Satellite performance approaching fiber will become more common using advanced error control coding in new systems. However, many legacy satellite systems will continue to exhibit higher physical layer BER than newer satellite systems. TCP uses all packet drops as signals of network congestion and reduces its window size in an attempt to alleviate the congestion. In the absence of knowledge about why a packet was dropped (congestion or corruption), TCP must assume the drop was due to network congestion to avoid congestion collapse [Jac88] [FF98]. Therefore, packets dropped due to corruption cause TCP to reduce the size of its sliding window, even though these packet drops do not signal congestion in the network.
- * **Bandwidth:** The radio spectrum is a limited natural resource, there is a restricted amount of bandwidth available to satellite systems which is typically controlled by licenses. This scarcity makes it difficult to trade bandwidth to solve other design problems. Satellite-based radio repeaters are known as transponders. Traditional C-band transponder bandwidth is typically 36 MHz to accommodate one color television channel (or 1200 voice channels). Ku-band transponders are typically around 50 MHz. Furthermore, one satellite may carry a few dozen transponders. Not only is

bandwidth limited by nature, but the allocations for commercial communications are limited by international agreements so that this scarce resource can be used fairly by many different communications applications. Typical carrier frequencies for current, point-to-point, commercial, satellite services are 6 GHz (uplink) and 4 GHz (downlink), also known as C-band, and 14/12 GHz (Ku band). Services also utilise higher bands, including 30/20 GHz (Ka-band). XXX JB: I think we need add Ka-band details. You cannot get 250 Mbps out of a C-band or Ku-band transponder. Outbound Ka-band transponders range from 100 to 500 MHz. Inbound Ka-band transponders range from 50 to 250 MHz.XXX

- * Link Design: It is common to consider the satellite network segment composed of a forward link and a return link. The forward link (also called "downlink") is the link from the ground station of the satellite to the user terminal. The return link (also called "uplink") is the link in the opposite direction. These two links can have different capacities and employ different technologies to carry the IP packets. On the forward link, the satellite gateway often manages all the available capacity, possibly with several carriers, to communicate with a set of remote terminals. A carrier is a single Time-Division-Multiplexing (TDM) channel that multiplexes packets addressed to specific terminals. There are trade-offs in terms of overall system efficiency and performance observed by a user. Most systems incur additional delay to ensure overall system performance.

- * **Shared Medium Access:** In common with other radio media, satellite capacity can be assigned for use by a link for a period of time, for the duration of communication, for a per-packet or per burst of packets, or accessed using contention mechanisms. Packets sent over a shared radio channels need to be sent in frames that need to be allocated resources (bandwidth, power, time) for their transmission. This results in a range of characteristics that are very different to a permanently assigned medium (such as an Ethernet link using an optical fibre). On the return link, satellite resource is typically dynamically shared among the terminals. Two access methods can be distinguished: on-demand access or contention access. In the former, a terminal receives dedicated transmission resources (usually to send to the gateway). In the latter, some resources are reserved for contention access, where a set of terminals are allowed to compete to obtain transmission resource. Dedicated access, which is more common in currently deployed systems, can be through a Demand Assigned Multiple Access (DAMA) mechanism, while contention access techniques are usually based on Slotted Aloha (SA) and its numerous derivatives. More information on satellite links characteristics can be found in [RFC2488] [IJSCN17].

Satellite systems have several characteristics that differ from most terrestrial channels. These characteristics may degrade the performance of TCP. These characteristics include:

4.1. Impact of delay

Even for characteristics shared with terrestrial paths, the impact on a satellite link could be amplified by the path RTT. For example, paths using a satellite system can also exhibit a high loss-rate (e.g., a mobile user or a user behind a Wi-Fi link), where the additional delay can impact transport mechanisms.

4.1.1. Larger Bandwidth Delay Product

Although capacity is often less than in many terrestrial systems, the bandwidth delay product (BDP) defines the amount of data that a protocol is permitted to have "in flight" at any one time to fully utilize the available capacity. In flight means data that is transmitted, but not yet acknowledged.

The delay used in this equation is the path RTT and the bandwidth is the capacity of the bottleneck link along the network path. Because the delay in some satellite environments is higher, protocols need to keep a larger number of packets in flight.

This also impacts the size of window/credit needed to avoid flow control mechanisms throttling the sender rate.

4.1.2. Variable Link Delay

In some satellite environments, such as some Low Earth Orbit (LEO) constellations, the propagation delay to and from the satellite varies over time.

Even when the propagation delay varies only very slightly, the effects of medium access methods can result in significant variation in the link delay. Whether or not this will have an impact on performance of a well-designed transport is currently an open question.

4.1.3. Impact of delay on protocol feedback

The link delay of some satellite systems may require more time for a transport sender to determine whether or not a packet has been successfully received at the final destination. This delay impacts interactive applications as well as loss recovery, congestion control, flow control, and other algorithms (see Section 6).

4.2. Intermittent connectivity

In some non-GEO satellite orbit configurations, from time to time Internet connections need to be transferred from one satellite to another or from one ground station to another. This hand-off might cause excessive packet loss or reordering if not properly performed.

5. On-Path Mitigations

This section describes mitigations that operate on the path, rather than with the transport endpoints.

5.1. Link-Level Forward Error Correction and ARQ

XXX Common, but includes adaptive ModCod and sometimes ARQ - which can reduce the loss at the expense of decreasing the available capacity. XXX

5.2. PMTU Discovery

XXX Packet Size can impact performance and mitigations (such as PEP/ Application Proxy) can interact with end-to-end PMTUD XXX

5.3. Quality of Service (QoS)

Links where packets are sent over radio channels exhibit various trade-offs in the way the signal is sent on the communications channel. These trade-offs are not necessarily the same for all packets, and network traffic flows can be optimised by mapping these onto different types of lower layer treatment (packet queues, resource management requests, resource usage, and adaption to the channel using FEC, ARQ, etc). Many systems differentiate classes of traffic to manage these QoS trade-offs.

5.4. Split-TCP PEP

High BDP networks commonly break the TCP end-to-end paradigm to adapt the transport protocol. Splitting a TCP connection allows adaptation for a specific use-case and to address the issues discussed in Section 2. Satellite communications commonly deploy Performance Enhancing Proxy (PEP) for compression, caching and TCP acceleration services [RFC3135]. Their deployment can result in significant performance improvement (e.g., a 50% page load time reduction in a SATCOM use-case [ICCRG100]).

[NCT13] and [RFC3135] describe the main functions of a SATCOM TCP split solution. For traffic originated at a gateway to an endpoint connected via a satellite terminal, the TCP split proxy intercepts TCP SYN packets, acting on behalf of the endpoint and adapts the sending rate to the SATCOM scenario. The split solution can specifically tune TCP parameters to the satellite link (latency, available capacity).

When a proxy is used on each side of the satellite link, the transport protocol can be replaced by a protocol other than TCP, optimized for the satellite link. This can be tuned using a priori information about the satellite system and/or by measuring the properties of the network segment that includes the satellite system.

Split connections can also recover from packet loss that is local to the part of the connection on which the packet losses occur. This eliminates the need for end-to-end recovery of lost packets.

One important advantage of a TCP split solution is that it does not require any end-to-end modification and is independent of both the client and server sides.

Split-TCP comes with a significant drawback: TCP splitters are often unable to track end-to-end improvements in protocol mechanisms (e.g., RACK, ECN, TCP Fast Open) or new protocols that share a wire format with TCP (MPTCP [RFC6824]). The set of methods configured in a split

proxy usually continue to be used, until the split solution is finally updated. This can delay/negate the benefit of any end-to-end improvements, contributing to ossification of the transport system.

5.5. Application Proxies

Authenticated proxies:

- * Split some functions, so the proxy needs to agree on the formats/ semantics of the protocol info that is changed
- * Need a trust relationship - need to be authenticated, and understand what is modified
- * Proxy needs to be on-path, which places constraints on the routing via the box
- * Need to discover the device, which might vary by user - by service - etc.

6. Generic Transport Protocol Mechanisms

This section outlines transport protocol mechanisms that may be necessary to tune or optimize in satellite or hybrid satellite/terrestrial networks to better utilize the available capacity of the link. These mechanisms may also be needed to fully utilize fast terrestrial channels. Furthermore, these mechanisms do not fundamentally hurt performance in a shared terrestrial network. Each of the following sections outlines one mechanism and why that mechanism may be needed.

- * Transport initialization: the connection handshake (in TCP the 3-way exchange) takes a longer time to complete, delaying the time to send data (several transport protocol exchanges may be needed, such as TLS);
- * Size of congestion window required: to fully exploit the bottleneck capacity, a high BDP requires a larger number of in-flight packets;
- * Size of receiver (flow control) window required: to fully exploit the bottleneck capacity, a high BDP requires a larger number of in-flight packets;
- * Reliability: transport layer loss detection and repair can incur a single or multiple RTTs (the performance of end-to-end retransmission is also impacted when using a high RTT path);

- * Getting up to speed: many congestion control methods employ an exponential increase in the sending rate during slow start (for path capacity probing), a high RTT will increase the time to reach the maximum possible rate;
- * Asymmetry: when the links are asymmetric the return path may modify the rate and/timing of transport acknowledgment traffic, potentially changing behaviour (e.g., limiting the forward sending rate).

6.1. Transport Initialization

Many transport protocols now deploy 0-RTT mechanisms [REF] to reduce the number of RTTs required to establish a connection. QUIC has an advantage that the TLS and TCP negotiations can be completed during the transport connection handshake. This can reduce the time to transmit the first data.

6.2. Getting up to Speed

Results of [IJSCN19] illustrate that it can still take many RTTs for a CC to increase the sending rate to fill the bottleneck capacity. The delay in getting up to speed can dominate performance for a path with a large RTT, and requires the congestion and flow controls to accommodate the impact of path delay.

One relevant solution is tuning of the initial window described in [I-D.irtf-iccr-g-sallantin-initial-spreading] , which has been shown to improve performance both for high BDP and more common BDP [CONEXT15] [ICC16] . Such a solution requires using sender pacing to avoid generating bursts of packets in a network.

6.3. Sizing of Maximum Congestion Window

Size of windows required: to fully exploit the bottleneck capacity, a high BDP requires a larger number of in-flight packets.

The number of in-flight packets required to fill a bottleneck capacity, is dependent on the BDP. Default values of maximum windows may not be suitable for a SATCOM context.

Such as presented in [PANRG105] , only increasing the initial congestion window is not the only way that can improve QUIC performance in a SATCOM context: increasing maximum congestion windows can also result in much better performance. Other protocol mechanisms also need to be considered, such as flow control at the stream level in QUIC.

6.4. Reliability (Loss Recovery/Repair)

The time for end systems to perform packet loss detection and recovery/repair is a function of the path RTT.

The RTT also determines the time needed by a server to react to a congestion event. Both can impact the user experience. For example, when a user uses a Wi-Fi link to access the Internet via SATCOM terminal.

A solution could be to opportunistically retransmit packets even if they have not been detected as lost but the congestion control allows to transmit more packets.

6.4.1. Packet Level Forward Error Correction

XXX Packet level FEC can mitigate loss/re-ordering, with a trade-off in capacity. XXX

End-to-end packet Forward Error Correction offers an alternative to retransmission with different trade offs in terms of utilised capacity and repair capability.

The benefits of introducing FEC need to be weighed against the additional overhead introduced by end-to-end FEC and the opportunity to use link-local ARQ and/or link-adaptive FEC. A transport connections can suffer link-related losses from a particular link (e.g., Wi-Fi), but also congestion loss (e.g. router buffer overflow in a satellite operator ground segment or along an Internet path).

6.5. Flow Control

Flow Control mechanisms allow the receiver to control the amount of data a sender can have in flight at any time. Flow Control allows the receiver to allocate the smallest buffer sizes possible improving memory usage on receipt.

The sizing of initial receive buffers requires a balance between keeping receive memory allocation small while allowing the send window to grow quickly to help ensure high utilization. The size of receive windows and their growth can govern the performance of the protocol if updates are not timely.

Many TCP implementations deploy Auto-scaling mechanisms to increase the size of the largest receive window over time. If these increases are not timely then sender traffic can stall while waiting to be notified of an increase in receive window size. XXX QUIC? XXX

Multi-streaming Protocols such as QUIC implement Flow Control using credit-based mechanisms that allow the receiver to prioritise which stream is able to send and when. Credit-based systems, when flow credit allocations are not timely, can stall sending when credit is exhausted.

6.6. ACK Traffic Reduction

When the links are asymmetric, for various reasons, the return path may modify the rate and/timing of transport acknowledgment traffic, potentially changing behaviour (e.g., limiting the forward sending rate).

Asymmetry in capacity (or in the way capacity is granted to a flow) can lead to cases where the transmission in one direction of communication is restricted by the transmission of the acknowledgment traffic flowing in the opposite direction. A network segment could present limitations in the volume of acknowledgment traffic (e.g., limited available return path capacity) or in the number of acknowledgment packets (e.g., when a radio-resource management system has to track channel usage), or both.

TCP Performance Implications of Network Path Asymmetry [RFC3449] describes a range of mechanisms that have been used to mitigate the impact of path asymmetry, primarily targeting operation of TCP.

Many mitigations have been deployed in satellite systems, often as a mechanism within a PEP. Despite their benefits over paths with high asymmetry, most mechanisms rely on being able to inspect and/or modify the transport layer header information of TCP ACK packets. This is not possible when the transport layer information is encrypted (e.g., using an IP VPN).

One simple mitigation is for the remote endpoint to send compound acknowledgments less frequently. A rate of one ACK for every RTT/4 can significantly reduce this traffic. The QUIC transport specification may evolve to allow the ACK Ratio to be adjusted.

6.7. Multi-Path

XXX This includes between different satellite systems and between satellite and terrestrial paths XXX

7. Protocol Specific Mechanisms

7.1. TCP Protocol Mechanisms

7.1.1. Transport Initialization

7.1.2. Getting Up To Speed

One relevant solution is tuning of the initial window described in [I-D.irtf-iccr-g-sallantin-initial-spreading][RFC6928], which has been shown to improve performance both for high BDP and more common BDP [CONEXT15] [ICC16]. This requires sender pacing to avoid generating bursts of packets to the network.

7.1.3. Size of Windows

7.1.4. Reliability

7.1.5. ACK Reduction

Mechanisms are being proposed in TCPM for TCP [REF].

7.2. QUIC Protocol Mechanisms

7.2.1. Transport initialization

QUIC has an advantage that the TLS and transport protocol negotiations can be completed during the transport connection handshake. This can reduce the time to transmit the first data. Moreover, using 0-RTT may further reduce the connection time for users reconnecting to a server.

7.2.2. Getting up to Speed

Getting up to speed may be easier with the usage of the 0-RTT-BDP extension proposed in [I-D.kuhn-quic-0rtt-bdp].

7.2.3. Size of Windows

7.2.4. Reliability

7.2.5. Asymmetry

The QUIC transport specification may evolve to allow the ACK Ratio to be adjusted.

Default could be adapted following [I-D.fairhurst-quic-ack-scaling] or using extensions to tune acknowledgement strategies [I-D.iyengar-quic-delayed-ack].

7.2.6. Packet Level Forward Error Correction

Network coding as proposed in [I-D.swett-nwcrq-coding-for-quic] and [I-D.roca-nwcrq-rlc-fec-scheme-for-quic] could help QUIC recover from link or congestion loss.

Another approach could utilise QUIC tunnels such as proposed in the MASQUE WG to apply packet FEC to all or a part of the end-to-end path or enable local retransmissions.

7.2.7. Split Congestion Control

Splitting the congestion control requires the deployment of application proxies.

8. Discussion

Many of the issues identified for high BDP paths already exist when using an encrypted transport service over a path that employs encryption at the IP layer. This includes endpoints that utilise IPsec at the network layer, or use VPN technology over a satellite network segment. Users are unable to benefit from enhancement within the satellite network segment, and often the user is unaware of the presence of the satellite link on their path, except through observing the impact it has on the performance they experience.

One solution would be to provide PEP functions at the termination of the security association (e.g., in a VPN client). Another solution could be to fall-back to using TCP (possibly with TLS or similar methods being used on the transport payload). A different solution could be to deploy and maintain a bespoke protocol tailored to high BDP environments. In the future, we anticipate that fall-back to TCP will become less desirable, and methods that rely upon bespoke configurations or protocols will be unattractive. In parallel, new methods such as QUIC will become widely deployed. The opportunity therefore exists to ensure that the new generation of protocols offer acceptable performance over high BDP paths without requiring operating tuning or specific updates by users.

8.1. Mitigation Summary

XXX A Table will be inserted here XXX

9. Acknowledgments

The authors would like to thank Mark Allman, Daniel R. Glover and Luis A. Sanchez the authors of RFC2488 from which the format and descriptions of satellite systems in this document have taken inspiration.

The authors would like to thank Christian Huitema, Igor Lubashev, Alexandre Ferrieux, Francois Michel, Emmanuel Lochin, github user sedrubal and the participants of the IETF106 side-meeting on QUIC for high BDP for their useful feedback.

10. Security Considerations

This document does not propose changes to the security functions provided by the QUIC protocol. QUIC uses TLS encryption to protect the transport header and its payload. Security is considered in the "Security Considerations" of cited IETF documents.

11. Informative References

- [CONEXT15] Li, Q., Dong, M., and P B. Godfrey, "Halfback: Running Short Flows Quickly and Safely", ACM CoNEXT , 2015.
- [FF98] Floyd, S. and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", IEEE Transactions on Networking 10.1109/90.79302, 1999.
- [I-D.fairhurst-quic-ack-scaling]
Fairhurst, G., Custura, A., and T. Jones, "Changing the Default QUIC ACK Policy", Work in Progress, Internet-Draft, draft-fairhurst-quic-ack-scaling-04, 15 March 2021, <<https://www.ietf.org/archive/id/draft-fairhurst-quic-ack-scaling-04.txt>>.
- [I-D.irtf-iccr-g-sallantin-initial-spreading]
Sallantin, R., Baudoin, C., Arnal, F., Dubois, E., Chaput, E., and A. Beylot, "Safe increase of the TCP's Initial Window Using Initial Spreading", Work in Progress, Internet-Draft, draft-irtf-iccr-g-sallantin-initial-spreading-00, 15 January 2014, <<https://www.ietf.org/archive/id/draft-irtf-iccr-g-sallantin-initial-spreading-00.txt>>.

- [I-D.irtf-panrg-path-properties]
Enghardt, T. and C. Krähenbühl, "A Vocabulary of Path Properties", Work in Progress, Internet-Draft, draft-irtf-panrg-path-properties-03, 9 July 2021, <<https://www.ietf.org/archive/id/draft-irtf-panrg-path-properties-03.txt>>.
- [I-D.iyengar-quick-delayed-ack]
Iyengar, J. and I. Swett, "Sender Control of Acknowledgement Delays in QUIC", Work in Progress, Internet-Draft, draft-iyengar-quick-delayed-ack-02, 2 November 2020, <<https://www.ietf.org/archive/id/draft-iyengar-quick-delayed-ack-02.txt>>.
- [I-D.kuhn-quick-0rtt-bdp]
Kuhn, N., Stephan, E., Fairhurst, G., Jones, T., and C. Huitema, "Transport parameters for 0-RTT connections", Work in Progress, Internet-Draft, draft-kuhn-quick-0rtt-bdp-09, 7 June 2021, <<https://www.ietf.org/archive/id/draft-kuhn-quick-0rtt-bdp-09.txt>>.
- [I-D.roca-nwcrp-rlc-fec-scheme-for-quick]
Roca, V., Michel, F., Swett, I., and M. Montpetit, "Sliding Window Random Linear Code (RLC) Forward Erasure Correction (FEC) Schemes for QUIC", Work in Progress, Internet-Draft, draft-roca-nwcrp-rlc-fec-scheme-for-quick-03, 9 March 2020, <<https://www.ietf.org/archive/id/draft-roca-nwcrp-rlc-fec-scheme-for-quick-03.txt>>.
- [I-D.swett-nwcrp-coding-for-quick]
Swett, I., Montpetit, M., Roca, V., and F. Michel, "Coding for QUIC", Work in Progress, Internet-Draft, draft-swett-nwcrp-coding-for-quick-04, 9 March 2020, <<https://www.ietf.org/archive/id/draft-swett-nwcrp-coding-for-quick-04.txt>>.
- [ICC16] Sallantin, R., Baudoin, C., Chaput, E., Arnal, F., Dubois, E., and A-L. Beylot, "Reducing web latency through TCP IW: Be smart", IEEE ICC , 2016.
- [ICCRG100] Kuhn, N., "MPTCP and BBR performance over Internet satellite paths", IETF ICCRG 100, 2017.
- [IJSCN17] Ahmed, T., Dubois, E., Dupe, JB., Ferrus, R., Gelard, P., and N. Kuhn, "Software-defined satellite cloud RAN", International Journal of Satellite Communications and Networking , 2017.

- [IJSCN19] Thomas, L., Dubois, E., Kuhn, N., and E. Lochin, "Google QUIC performance over a public SATCOM access", International Journal of Satellite Communications and Networking , 2019.
- [Jac88] Jacobson, V., "Congestion Avoidance and Control", ACM SIGCOMM 88, 1988.
- [Mar78] Martin, J., "Communications Satellite Systems", Prentice Hall 78, 1978.
- [Mon98] Montpetit, M.J., "TELEDESIC: Enabling The Global Community Interaccess", International Wireless Symposium 98, 1998.
- [NCT13] Pirovano, A. and F. Garcia, "A new survey on improving TCP performances over geostationary satellite link", Network and Communication Technologies , 2013.
- [PANRG105] Kuhn, N., Stephan, E., Border, J., and G. Fairhurst, "QUIC Over In-sequence Paths with Different Characteristics", IRTF PANRG 105, 2019.
- [RFC2488] Allman, M., Glover, D., and L. Sanchez, "Enhancing TCP Over Satellite Channels using Standard Mechanisms", BCP 28, RFC 2488, DOI 10.17487/RFC2488, January 1999, <<https://www.rfc-editor.org/info/rfc2488>>.
- [RFC2760] Allman, M., Ed., Dawkins, S., Glover, D., Griner, J., Tran, D., Henderson, T., Heidemann, J., Touch, J., Kruse, H., Ostermann, S., Scott, K., and J. Semke, "Ongoing TCP Research Related to Satellites", RFC 2760, DOI 10.17487/RFC2760, February 2000, <<https://www.rfc-editor.org/info/rfc2760>>.
- [RFC3135] Border, J., Kojo, M., Griner, J., Montenegro, G., and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", RFC 3135, DOI 10.17487/RFC3135, June 2001, <<https://www.rfc-editor.org/info/rfc3135>>.
- [RFC3449] Balakrishnan, H., Padmanabhan, V., Fairhurst, G., and M. Sooriyabandara, "TCP Performance Implications of Network Path Asymmetry", BCP 69, RFC 3449, DOI 10.17487/RFC3449, December 2002, <<https://www.rfc-editor.org/info/rfc3449>>.

- [RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, DOI 10.17487/RFC6824, January 2013, <<https://www.rfc-editor.org/info/rfc6824>>.
- [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>>.
- [RFC8975] Kuhn, N., Ed. and E. Lochin, Ed., "Network Coding for Satellite Systems", RFC 8975, DOI 10.17487/RFC8975, January 2021, <<https://www.rfc-editor.org/info/rfc8975>>.
- [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <<https://www.rfc-editor.org/info/rfc9000>>.
- [Sta94] Stallings, W., "Data and Computer Communications", MacMillian 4th edition, 1994.
- [Stu95] Sturza, M.A., "Architecture of the TELEDESIC Satellite System", International Mobile Satellite Conference 95, 1995.

Appendix A. Example Network Profiles

This proposes sampler profiles and a set of regression tests to evaluate transport protocols over SATCOM links and discusses how to ensure acceptable protocol performance.

XXX These test profiles currently focus on the measuring performance and testing for regressions in the QUIC protocol. The authors solicit input to adapt these tests to apply to more transport protocols. XXX

A.1. LEO

A.2. MEO

A.3. GEO

This section proposes a set of regression tests for QUIC that consider high BDP scenarios. We define by:

- * Download path: from Internet to the client endpoint;

- * Upload path: from the client endpoint to a server (e.g., in the Internet).

A.3.1. Small public satellite broadband access

The tested scenario has the following path characteristics:

- * Satellite downlink path: 10 Mbps
- * Satellite uplink path: 2 Mbps
- * No emulated packet loss
- * RTT: 650 ms
- * Buffer size : BDP

During the transmission of 100 MB on both download and upload paths, the test should report the upload and download time of 2 MB, 10 MB and 100 MB.

Initial thoughts of the performance objectives for QUIC are the following:

- * 3 s for downloading 2 MB
- * 10 s for downloading 10 MB
- * 85 s for downloading 100 MB
- * 10 s for uploading 2 MB
- * 50 s for uploading 10 MB
- * 420 s for uploading 100 MB

A.3.2. Medium public satellite broadband access

The tested scenario has the following path characteristics:

- * Satellite downlink path: 50 Mbps
- * Satellite uplink path: 10 Mbps
- * No emulated packet loss
- * RTT: 650 ms

- * Buffer size : BDP

During the transmission of 100 MB on the download path, the test should report the download time for 2 MB, 10 MB and 100 MB. Then, to assess the performance of QUIC with the 0-RTT extension and its variants, after 10 seconds, repeat the transmission of 100 MB on the download path where the download time for 2 MB, 10 MB and 100 MB is recorded.

Initial thoughts of the performance objectives for QUIC are the following:

- * 3 s for the first downloading 2 MB
- * 5 s for the first downloading 10 MB
- * 20 s for the first downloading 100 MB
- * TBD s for the second downloading 2 MB
- * TBD s for the second downloading 10 MB
- * TBD s for the second downloading 100 MB

A.3.3. Congested medium public satellite broadband access

There are cases where the uplink path is congested or where the capacity of the uplink path is not guaranteed.

The tested scenario has the following path characteristics:

- * Satellite downlink path: 50 Mbps
- * Satellite uplink path: 0.5 Mbps
- * No emulated packet loss
- * RTT: 650 ms
- * Buffer size : BDP

During the transmission of 100 MB on the download path, the test should report the download time for 2 MB, 10 MB and 100 MB.

Initial thoughts of the performance objectives for QUIC are the following:

- * 3 s for downloading 2 MB

- * 5 s for downloading 10 MB
- * 20 s for downloading 100 MB

A.3.4. Variable medium public satellite broadband access

There are cases where the downlink path is congested or where, due to link layer adaptations to rain fading, the capacity of the downlink path is variable.

The tested scenario has the following path characteristics:

- * Satellite downlink path: 50 Mbps - wait 5s - 10 Mbps
- * Satellite uplink path: 10 Mbps
- * No emulated packet loss
- * RTT: 650 ms
- * Buffer size : BDP

During the transmission of 100 MB on the download path, the test should report the download time for 2 MB, 10 MB and 100 MB.

Initial thoughts of the performance objectives for QUIC are the following:

- * TBD s for downloading 2 MB
- * TBD s for downloading 10 MB
- * TBD s for downloading 100 MB

A.3.5. Loss-free large public satellite broadband access

The tested scenario has the following path characteristics:

- * Satellite downlink path: 250 Mbps
- * Satellite uplink path: 6 Mbps
- * No emulated packet loss
- * RTT: 650 ms
- * Buffer size : BDP

During the transmission of 100 MB on the download path, the test should report the download time for 2 MB, 10 MB and 100 MB. Then, to assess the performance of QUIC with the 0-RTT extension and its variants, after 10 seconds, repeat the transmission of 100 MB on the download path where the download time for 2 MB, 10 MB and 100 MB is recorded.

Initial thoughts of the performance objectives for QUIC are the following:

- * 3 s for the first downloading 2 MB
- * 5 s for the first downloading 10 MB
- * 8 s for the first downloading 100 MB
- * TBD s for the second downloading 2 MB
- * TBD s for the second downloading 10 MB
- * TBD s for the second downloading 100 MB

A.3.6. Lossy large public satellite broadband access

The tested scenario has the following path characteristics:

- * Satellite downlink path: 250 Mbps
- * Satellite uplink path: 6 Mbps
- * Emulated packet loss on both downlink and uplink paths:
 - Uniform random transmission link losses: 1%
- * RTT: 650 ms
- * Buffer size : BDP

During the transmission of 100 MB on the download path, the test should report the download time for 2 MB, 10 MB and 100 MB.

Initial thoughts of the performance objectives for QUIC are the following:

- * 3 s for downloading 2 MB (uniform transmission link losses)
- * 6 s for downloading 10 MB (uniform transmission link losses)

- * 10 s for downloading 100 MB (uniform transmission link losses)

Appendix B. Revision Notes

Note to RFC-Editor: please remove this entire section prior to publication.

Individual draft -00:

- * Comments and corrections are welcome directly to the authors or via the <https://github.com/uaaerg/draft-jones-transport-for-satellite> github repo in the form of pull requests and issues.

Individual draft -01:

- * Explained Terms Forward and return link
- * Rearranged text to help the document read better
- * Fix typos and inaccuracies
- * Added a mention of MPTCP

Authors' Addresses

Tom Jones
University of Aberdeen

Email: tom@erg.abdn.ac.uk

Godred Fairhurst
University of Aberdeen

Email: gorry@erg.abdn.ac.uk

Nicolas Kuhn
CNES

Email: nicolas.kuhn@cnes.fr

John Border
Hughes Network Systems, LLC

Email: border@hns.com

Emile Stephan
Orange

Email: emile.stephan@orange.com