

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

J. Arkko
Ericsson
T. Hardie
Cisco
T. Pauly
Apple
M. Kühlewind
Ericsson
25 October 2021

Considerations on Application - Network Collaboration Using Path Signals
[draft-arkko-iab-path-signals-collaboration-01](#)

Abstract

Encryption and other security mechanisms are on the rise on all layers of the stack, protecting user data and making network operations more secured. Further, encryption is also a tool to address ossification that has been observed over time. Separation of functions into layers and enforcement of layer boundaries based on encryption supports selected exposure to those entities that are addressed by a function on a certain layer. A clear separation supports innovation and also enables new opportunities for collaborative functions. [RFC 8558](#) describes path signals as messages to or from on-path elements. This document states principles for designing mechanisms that use or provide path signals and calls for actions on specific valuable cases.

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	2
2.	Past Guidance	4
3.	Principles	5
3.1.	Intentional Distribution	6
3.2.	Minimum Set of Entities	7
3.3.	Consent of Parties	7
3.4.	Minimum Information	8
3.5.	Carrying Information	9
3.6.	Protecting Information and Authentication	9
4.	Further Work	10
5.	Acknowledgments	11
6.	Informative References	11
	Authors' Addresses	14

[1.](#) Introduction

Encryption, besides its important role in security in general, provides a tool to control information access and protects against ossification by avoiding unintended dependencies and requiring active maintenance. The increased deployment of encryption provides an opportunity to reconsider parts of Internet architecture that have rather used implicit derivation of input signals for on-path functions than explicit signaling, as recommended by [RFC 8558](#) [[RFC8558](#)].

[RFC 8558](#) defines the term path signals as signals to or from on-path elements. Today path signals are often implicit, e.g. derived from in-clear end-to-end information by e.g. examining transport protocols. For instance, on-path elements use various fields of the TCP header [[RFC0793](#)] to derive information about end-to-end latency as well as congestion. These techniques have evolved because the information was simply available and use of this information is

easier and therefore also cheaper than any explicit and potentially complex cooperative approach.

As such, applications and networks have evolved their interaction without comprehensive design for how this interaction should happen or which information would be desired for a certain function. This has lead to a situation where sometimes information is used that maybe incomplete, incorrect, or only indirectly representative of the information that was actually desired. In addition, dependencies on information and mechanisms that were designed for a different function limits the evolvability of the protocols in question.

The unplanned interaction ends up having several negative effects:

- * Ossifying protocols by introducing unintended parties that may not be updating
- * Creating systemic incentives against deploying more secure or private versions of protocols
- * Basing network behaviour on information that may be incomplete or incorrect
- * Creating a model where network entities expect to be able to use rich information about sessions passing through

For instance, features such as DNS resolution or TLS setup have been used beyond their original intent, such as in name-based filtering. MAC addresses have been used for access control, captive portal implementations that employ taking over cleartext HTTP sessions, and so on.

Increased deployment of encryption can and will change this situation. For instance, QUIC replaces TCP for various application and protects all end-to-end signals to only be accessible by the endpoint, ensuring evolvability [[RFC9000](#)]. QUIC does expose information dedicated for on-path elements to consume by design explicit signal for specific use cases, such as the Spin bit for latency measurements or connection ID that can be used by load balancers [[I-D.ietf-quic-manageability](#)] but information is limited to only those use cases. Each new use cases requires additional action.

Explicit signals that are specifically designed for the use of on-path function leave all other information is appropriately protected. This enables an architecturally clean approach and evolvability, while allowing an information exchange that is important for improving the quality of experience for users and efficient management of the network infrastructure built for them.

This draft discusses different approaches for explicit collaboration and provides guidance on architectural principles to design new mechanisms. [Section 2](#) discusses past guidance. [Section 3](#) discusses principles that good design can follow. This section also provides some examples and explanation of situations that not following the principles can lead to. [Section 4](#) points to topics that need more to be looked at more carefully before any guidance can be given.

2. Past Guidance

Incentives are a well understood problem in general but perhaps not fully internalised for various designs attempting to establish collaboration between applications and path elements. The principle is that both receiver and sender of information must acquire tangible and immediate benefits from the communication, such as improved performance.

A related issue is understanding whether a business model or ecosystem change is needed. For instance, relative prioritization between different flows of a user or an application does not require agreements or payments. But requesting prioritization over other people's traffic may imply that you have to pay for that which may not be easy even for a single provider let alone across many.

But on to more technical aspects.

The main guidance in [\[RFC8558\]](#) is to be aware that implicit signals will be used whether intended or not. Protocol designers should consider either hiding these signals when the information should not be visible, or using explicit signals when it should be.

[\[RFC9049\]](#) discusses many past failure cases, a catalogue of past issues to avoid. It also provides relevant guidelines for new work, from discussion of incentives to more specific observations, such as the need for outperforming end-to-end mechanisms ([Section 4.4](#)), considering the need for per-connection state ([Section 4.6](#)), taking into account the latency involved in reacting to distant signals, and so on.

There are also more general guidance documents, e.g., [\[RFC5218\]](#) discusses protocol successes and failures, and provides general advice on incremental deployability etc. Internet Technology Adoption and Transition (ITAT) workshop report [\[RFC7305\]](#) is also recommended reading on this same general topic. And [\[RFC6709\]](#) discusses protocol extensibility, and provides general advice on the importance of global interoperability and so on.

3. Principles

This section attempts to provide some architecture-level principles that would help future designers and recommend useful models to apply.

A large number of our protocol mechanisms today fall into one of two categories: authenticated and private communication that is only visible to the end-to-end nodes; and unauthenticated public communication that is visible to all nodes on a path. [RFC 8558](#) explores the line between data that is protected and path signals.

There is a danger in taking a position that is too extreme towards either exposing all information to the path, or hiding all information from the path.

Exposed information encourages pervasive monitoring, which is described in [RFC 7258](#) [[RFC7258](#)]. Exposed information may also be used for commercial purposes, or form a basis for filtering that the applications or users do not desire.

But a lack of all path signaling, on the other hand, may be harmful to network management, debugging, or the ability for networks to provide the most efficient services. There are many cases where elements on the network path can provide beneficial services, but only if they can coordinate with the endpoints. It also affects the ability of service providers and others observe why problems occur [[RFC9075](#)].

This situation is sometimes cast as an adversarial tradeoff between privacy and the ability for the network path to provide intended functions. However, this is perhaps an unnecessarily polarized characterization as a zero-sum situation. Not all information passing implies loss of privacy. For instance, performance information or preferences do not require disclosing user or application identity or what content is being accessed, network congestion status information does not have reveal network topology or the status of other users, and so on.

This points to one way to resolve the adversity: the careful of design of what information is passed.

Another approach is to employ explicit trust and coordination between endpoints and network devices. VPNs are a good example of a case where there is an explicit authentication and negotiation with a network path element that's used to optimize behavior or gain access to specific resources.

The goal of improving privacy and trust on the Internet does not necessarily need to remove the ability for network elements to perform beneficial functions. We should instead improve the way that these functions are achieved. Our goals should be:

- * To ensure that information is distributed intentionally, not accidentally;
- * to understand the privacy and other implications of any distributed information;
- * to ensure that the information distribution targets the intended parties; and
- * to gate the distribution of information on the consent of the relevant parties

These goals for distribution apply equally to senders, receivers, and path elements.

We can establish some basic questions that any new network path functions should consider:

- * What is the minimum set of entities that need to be involved?
- * What is the minimum information each entity in this set needs?
- * Which entities must consent to the information exchange?

If we look at many of the ways network path functions are achieved today, we find that many if not most of them fall short the standard set up by the questions above. Too often, they send unnecessary information or fail to limit the scope of distribution or providing any negotiation or consent.

Going forward, new standards work in the IETF needs to focus on addressing this gap by providing better alternatives and mechanisms for providing path functions. Note that not all of these functions can be achieved in a way that preserves a high level of user privacy from the network; in such cases, it is incumbent upon us to not ignore the use case, but instead to define the high bar for consent and trust, and thus define a limited applicability for those functions.

3.1. Intentional Distribution

This guideline is best expressed in [RFC 8558](#):

"Fundamentally, this document recommends that implicit signals should be avoided and that an implicit signal should be replaced with an explicit signal only when the signal's originator intends that it be used by the network elements on the path. For many flows, this may result in the signal being absent but allows it to be present when needed."

This guideline applies also in the other direction as well. For instance, a network element should not unintentionally leak information that is visible to endpoints. An explicit decision is needed for a specific information to be provided, along with analysis of the security and privacy implications of that information.

3.2. Minimum Set of Entities

It is recommended that a design identify the minimum number of entities needed to share a specific signal required for an identified function. In some cases this will be a very limited set, e.g. when the application needs to provide a signal to a specific gateway function. In other cases, such as congestion control, a signal might be shared with every router along the path, since each should be aware of the congestion.

While it is tempting to consider removing these limitations in the context of closed, private networks, each interaction is still best considered separately, rather than simply allowing all information exchanges within the closed network. Even in a closed network with carefully managed components there may be compromised components, as evidenced in the most extreme way by the Stuxnet worm that operated in an airgapped network. Most "closed" networks have at least some needs and means to access the rest of the Internet, and should not be modeled as if they had an impenetrable security barrier.

3.3. Consent of Parties

Consent and trust must determine the distribution of information. The set of entities that need to consent is determined by the scope and specificity of the information being shared.

Three distinct types of consent are recommended for collaboration or information sharing:

- * A corollary of the intentional distribution is that the sender needs to agree to sending the information. Or that the requester for an action needs to agree to make a request; it should not be an implicit decision by the receiver that information was sent or a request was made, just because a packet happened to be formed in a particular way.

- * At the same time, the recipient of information or the target of a request should agree to wishing to receive the information. It should not be burdened with extra processing if it does not have willingness or a need to do so. This happens naturally in most protocol designs, but has been a problem for some cases where "slow path" packet processing is required or implied, and the recipient or router did not have willingness for this.
- * Internet communications are not made for the applications, they are ultimately made on behalf of users. Information relating to the users is something that both networks and applications should be careful with, and not be shared without the user's consent. This is not always easy, as the interests of the user and (for instance) application developer may not always coincide; some applications may wish to collect more information about the user than the user would like.

As a result, typically an application's consent is not the same as the user's consent.

3.4. Minimum Information

Parties should provide only the information that is needed for the other party to perform the collaboration task that is desired by this party, and not more. This applies to information sent by an application about itself, information sent about users, or information sent by the network.

An architecture can follow the guideline from [RFC 8558](#) in using explicit signals, but still fail to differentiate properly between information that should be kept private and information that should be shared.

In looking at what information can or cannot easily be passed, we can look at both information from the network to the application, and from the application to the network.

For the application to the network direction, user-identifying information can be problematic for privacy and tracking reasons. Similarly, application identity can be problematic, if it might form the basis for prioritization or discrimination that the that application provider may not wish to happen. It may also have undesirable economic consequences, such as extra charges for the consumer from a priority service where a regular service would have worked.

On the other hand, as noted above, information about general classes of applications may be desirable to be given by application

providers, if it enables prioritization that would improve service, e.g., differentiation between interactive and non-interactive services.

For the network to application direction there is similarly sensitive information, such as the precise location of the user. On the other hand, various generic network conditions, predictive bandwidth and latency capabilities, and so on might be attractive information that applications can use to determine, for instance, optimal strategies for changing codecs. However, information given by the network about load conditions and so on should not form a mechanism to provide a side-channel into what other users are doing.

While information needs to be specific and provided on a per-need basis, it is often beneficial to provide declarative information that, for instance, expresses application needs than makes specific requests for action.

3.5. Carrying Information

There is a distinction between what information is passed and how it is carried. The actually sent information may be limited, while the mechanisms for sending or requesting information can be capable of sending much more.

There is a tradeoff here between flexibility and ensuring the minimality of information in the future. The concern is that a fully generic data sharing approach between different layers and parties could potentially be misused, e.g., by making the availability of some information a requirement for passing through a network.

This is undesirable, and our recommendation is to employ very targeted, minimal information carriage mechanisms.

3.6. Protecting Information and Authentication

Some simple forms of information often exist in cleartext form, e.g., ECN bits from routers are generally not authenticated or integrity protected. This is possible when the information exchanges are advisory in their nature, and do not carry any significantly sensitive information from the parties.

In other cases it may be necessary to establish a secure channel for communication with a specific other party, e.g., between a network element and an application. This channel may need to be authenticated, integrity protected and encrypted. This is necessary, for instance, if the particular information or request needs to be shared in confidentiality only with a particular, trusted node, or there's

a danger of an attack where someone else may forge messages that could endanger the communication.

However, it is important to note that authentication does not equal trust. Whether a communication is with an application server or network element that can be shown to be associated with a particular domain name, it does not follow that information about the user can be safely sent to it.

In some cases, the ability of a party to show that it is on the path can be beneficial. For instance, an ICMP error that refers to a valid flow may be more trustworthy than any ICMP error claiming to come from an address.

Other cases may require more substantial assurances. For instance, a specific trust arrangement may be established between a particular network and application. Or technologies such as confidential computing can be applied to provide an assurance that information processed by a party is handled in an appropriate manner.

4. Further Work

This is a developing field, and it is expected that our understanding continues to grow. The recent changes with regards to much higher use of encryption at different protocol layers, the consolidation or more and more traffic to the same destinations, and so on have also greatly impacted the field.

While there are some examples of modern, well-designed collaboration mechanisms, clearly more work is needed. Many complex cases would require significant developments in order to become feasible.

Some of the most difficult areas are listed below. Research on these topics would be welcome.

- * Business arrangements. Many designs - for instance those related to quality-of-service - involve an expectation of paying for a service. This is possible and has been successful within individual domains, e.g., users can pay for higher data rates or data caps in their ISP networks. However, it is a business-wise much harder proposition for end-to-end connections across multiple administrative domains [[Claffy2015](#)] [[RFC9049](#)].
- * Secure communications with path elements. This has been a difficult topic, both from the mechanics and scalability point view, but also because there is no easy way to find out which parties to trust or what trust roots would be appropriate. Some application-network element interaction designs have focused on

information (such as ECN bits) that is distributed openly within a path, but there are limited examples of designs with secure information exchange with specific nodes.

- * The use of path signals for reducing the effects of denial-of-service attacks, e.g., in the form of modern "source quench" designs.
- * Ways of protecting information when held by network elements or servers, beyond communications security. For instance, host applications commonly share sensitive information about the user's actions with other nodes, starting from basic data such as domain names learned by DNS infrastructure or source and destination addresses and protocol header information learned by all routers on the path, to detailed end user identity and other information learned by the servers. Some solutions are starting to exist for this but are not widely deployed, at least not today [[Oblivious](#)] [[PDOT](#)] [[I-D.arkko-dns-confidential](#)] [[I-D.thomson-http-oblivious](#)]. These solutions address also very specific parts of the issue, and more work remains.
- * Sharing information from networks to applications. Some proposals have been made in this space (see, e.g., [[I-D.flinck-mobile-throughput-guidance](#)]) but there are no successful or deployed mechanisms today.

5. Acknowledgments

The authors would like to thank everyone at the IETF, the IAB, and our day jobs for interesting thoughts and proposals in this space. Fragments of this document were also in [[I-D.per-app-networking-considerations](#)] and [[I-D.arkko-path-signals-information](#)] that were published earlier. We would also like to acknowledge [[I-D.trammell-stackevo-explicit-coop](#)] for presenting similar thoughts. Finally, the authors would like to thank Adrian Farrell, Toerless Eckert, and Jeffrey Haas for useful feedback in the IABOPEN session at IETF-111.

6. Informative References

[Claffy2015]

kc Claffy, . and D. Clark, "Adding Enhanced Services to the Internet: Lessons from History", TPRC 43: The 43rd Research Conference on Communication, Information and Internet Policy Paper , April 2015.

[I-D.arkko-dns-confidential]

Arkko, J. and J. Novotny, "Privacy Improvements for DNS

Resolution with Confidential Computing", Work in Progress, Internet-Draft, [draft-arkko-dns-confidential-02](https://www.ietf.org/archive/id/draft-arkko-dns-confidential-02), 2 July 2021, <<https://www.ietf.org/archive/id/draft-arkko-dns-confidential-02.txt>>.

[I-D.arkko-path-signals-information]

Arkko, J., "Considerations on Information Passed between Networks and Applications", Work in Progress, Internet-Draft, [draft-arkko-path-signals-information-00](https://www.ietf.org/archive/id/draft-arkko-path-signals-information-00), 22 February 2021, <<https://www.ietf.org/archive/id/draft-arkko-path-signals-information-00.txt>>.

[I-D.flinck-mobile-throughput-guidance]

Jain, A., Terzis, A., Flinck, H., Sprecher, N., Arunachalam, S., Smith, K., Devarapalli, V., and R. B. Yanai, "Mobile Throughput Guidance Inband Signaling Protocol", Work in Progress, Internet-Draft, [draft-flinck-mobile-throughput-guidance-04](https://www.ietf.org/archive/id/draft-flinck-mobile-throughput-guidance-04), 13 March 2017, <<https://www.ietf.org/archive/id/draft-flinck-mobile-throughput-guidance-04.txt>>.

[I-D.ietf-quic-manageability]

Kuehlewind, M. and B. Trammell, "Manageability of the QUIC Transport Protocol", Work in Progress, Internet-Draft, [draft-ietf-quic-manageability-13](https://www.ietf.org/archive/id/draft-ietf-quic-manageability-13), 2 September 2021, <<https://www.ietf.org/archive/id/draft-ietf-quic-manageability-13.txt>>.

[I-D.per-app-networking-considerations]

Colitti, L. and T. Pauly, "Per-Application Networking Considerations", Work in Progress, Internet-Draft, [draft-per-app-networking-considerations-00](https://www.ietf.org/archive/id/draft-per-app-networking-considerations-00), 15 November 2020, <<https://www.ietf.org/archive/id/draft-per-app-networking-considerations-00.txt>>.

[I-D.thomson-http-oblivious]

Thomson, M. and C. A. Wood, "Oblivious HTTP", Work in Progress, Internet-Draft, [draft-thomson-http-oblivious-02](https://www.ietf.org/archive/id/draft-thomson-http-oblivious-02), 24 August 2021, <<https://www.ietf.org/archive/id/draft-thomson-http-oblivious-02.txt>>.

[I-D.trammell-stackevo-explicit-coop]

Trammell, B., "Architectural Considerations for Transport Evolution with Explicit Path Cooperation", Work in Progress, Internet-Draft, [draft-trammell-stackevo-explicit-coop-00](https://www.ietf.org/archive/id/draft-trammell-stackevo-explicit-coop-00), 23 September 2015, <<https://www.ietf.org/archive/id/draft-trammell-stackevo-explicit-coop-00.txt>>.

[Oblivious]

Schmitt, P., "Oblivious DNS: Practical privacy for DNS queries", Proceedings on Privacy Enhancing Technologies 2019.2: 228-244 , 2019.

[PDoT]

Nakatsuka, Y., Pavard, A., and G. Tsudik, "PDoT: Private DNS-over-TLS with TEE Support", Digit. Threat.: Res. Pract., Vol. 2, No. 1, Article 3, <https://dl.acm.org/doi/fullHtml/10.1145/3431171> , February 2021.

[RFC0793]

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

[RFC5218]

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

[RFC6709]

Carpenter, B., Aboba, B., Ed., and S. Cheshire, "Design Considerations for Protocol Extensions", [RFC 6709](#), DOI 10.17487/RFC6709, September 2012, <<https://www.rfc-editor.org/info/rfc6709>>.

[RFC7258]

Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", [BCP 188](#), [RFC 7258](#), DOI 10.17487/RFC7258, May 2014, <<https://www.rfc-editor.org/info/rfc7258>>.

[RFC7305]

Lear, E., Ed., "Report from the IAB Workshop on Internet Technology Adoption and Transition (ITAT)", [RFC 7305](#), DOI 10.17487/RFC7305, July 2014, <<https://www.rfc-editor.org/info/rfc7305>>.

[RFC8558]

Hardie, T., Ed., "Transport Protocol Path Signals", [RFC 8558](#), DOI 10.17487/RFC8558, April 2019, <<https://www.rfc-editor.org/info/rfc8558>>.

[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>>.

[RFC9049]

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

[RFC9075] Arkko, J., Farrell, S., Kühlewind, M., and C. Perkins,
"Report from the IAB COVID-19 Network Impacts Workshop
2020", [RFC 9075](#), DOI 10.17487/RFC9075, July 2021,
<<https://www.rfc-editor.org/info/rfc9075>>.

Authors' Addresses

Jari Arkko
Ericsson

Email: jari.arkko@ericsson.com

Ted Hardie
Cisco

Email: ted.ietf@gmail.com

Tommy Pauly
Apple

Email: tpauly@apple.com

Mirja Kühlewind
Ericsson

Email: mirja.kuehlewind@ericsson.com

