Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 2, 2018

D. Dolson

J. Snellman

M. Boucadair
C. Jacquenet
Orange

March 1, 2018

# An Inventory of Transport-centric Functions Provided by Middleboxes
## draft-dolson-transport-middlebox-02

Abstract

This document summarizes benefits that operators perceive to be provided by intermediary devices that provide functions apart from normal IP forwarding. Such intermediary devices are often called "middleboxes".

RFC3234 defines a taxonomy of middleboxes and issues in the Internet. Most of those middleboxes utilize or modify application-layer data. This document primarily focuses on devices that observe and act on information carried in the transport layer, and especially information carried in TCP packets.

A primary goal of this document is to provide information to working groups developing new transport protocols, to aid understanding of what might be gained or lost by design decisions that may affect (or be affected by) middlebox operation.

Table of Contents

## 1. Introduction

At IETF97, at a meeting regarding the Path Layer UDP Substrate (PLUS) protocol, a request was made for documentation about the benefits that might be provided by permitting middleboxes to have some visibility to transport-layer information.

From RFC3234 [RFC3234], "A middlebox is defined as any intermediary device performing functions other than the normal, standard functions of an IP router on the datagram path between a source host and destination host."

Middleboxes are usually (but not exclusively) deployed at locations permitting observation of bidirectional traffic flows. Such locations are typically points where stub networks connect to the Internet; e.g.,:

- Where a residential or business customer connects to its service provider(s), which may include multi-homing.
- On the Gi interface where a GGSN connects to a PDN (see section 3.1 of [RFC6459]).
- For the purposes of this document (and consistent with the RFC3234 definition), middlebox functions may be found in routers and switches in addition to dedicated devices.

The QUIC working group and PLUS BoF are debating the appropriate amount of information that end-points should expose to on-path network middleboxes and human trouble-shooters. (Some information used for debugging is discussed in <https://www.snellman.net/blog/archive/2016-12-01-quic-tou/>.) This document itemizes a variety of features provided by middleboxes and by ad hoc analysis performed by operators using packet analyzers.

Many of the techniques described in this document require stateful analysis of transport streams. A generic state machine is described in [I-D.trammell-plus-statefulness].

### 1.1. Operator Perspective

The Internet is complicated, and network operators are tasked with providing the network abstraction between end-points. Network operators are often the ones first called upon when applications fail

to function properly, often with user reports about application behaviors (not about packet behaviors). Therefore it isn't surprising that operators (wanting to be helpful) desire some visibility into flow information to identify how well the problem flows are progressing and how well other flows are progressing.

Advanced service functions (e.g., NATs, firewalls, etc.) are widely used to achieve various objectives such as IP address sharing, firewalling, avoiding covert channels, detecting and protect against ever increasing DDoS attacks, etc.

These sophisticated service functions are located in the network but also in service platforms, or intermediate entities (e.g., CDNs). Maintenance and diagnostics are complicated, particularly when responsibility is shared among various players.

Network Providers are challenged to deliver differentiated services as a competitive business advantage, while mastering the complexity of the applications, (continuously) evaluating the impacts on middleboxes, and enhancing customer's quality of experience.

Despite the complexity, removing all those functions is not an option because they are used to address constraints that are often typical of the current (and changing) Internet situation. Operators must deal with constraints such as global IPv4 address depletion and must support a plethora of services with different QoS, security, robustness, etc. requirements. Furthermore, environment-specific designs may require a number of service functions, such as those needed at the Gi interface of a mobile infrastructure [I-D.ietf-sfc-use-case-mobility].

## 1.2. Scope

Although many middleboxes observe and manipulate application-layer content (e.g., session boarder controllers [RFC5853]) they are out of scope for this document, the aim being to describe middleboxes using transport-layer features. An earlier document [I-D.mm-wg-effect-encrypt] describes the impact of pervasive encryption of application-layer data on network monitoring, protecting and troubleshooting.

This document is not intended to recommend (or prohibit) middlebox deployment. Many operators have found the value provided by middleboxes to outweigh the added cost and complexity; this document attempts to provide that perspective as a reference in discussion of protocol design trade-offs.

This document is not intended to discuss issues related to middleboxes. These issues are well-known and are recorded in existing documents such as [RFC3234] and [RFC6269]. This document aims to elaborate on the motivations leading operators to enable such functions in spite of complications.

This document takes an operator perspective that measurement and management of transport connections is of benefit to both parties: for the end-user to receive better quality of experience, and for the network operator to improve resource usage, the former being a consequence of the latter.

This document does not discuss whether exposing some data to on-path devices for network assistance purposes can be achieved by using in-band or out-of-band mechanisms.

### 1.3. Requirements Language

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

## 2. Measurements

A number of measurements can be made by network devices that are either in-line with the traffic (responsible for forwarding) or receiving off-line copy of traffic from a tap or file capture. These measurements can be used either by automated systems, or for manual network troubleshooting purposes (e.g., using packet analysis tools). The automated systems can further be classified as monitoring systems that compute performance indicators for large numbers of connections and generate aggregated reports from them, and active systems that make decisions on how to handle specific packets based on these performance indicators.

Long-term trends in these measurements can aid an operator in capacity planning. Short-term anomalies revealed by these measurements can identify network breakages, attacks in progress, or misbehaving devices/applications.

### 2.1. Packet Loss

It is very useful for an operator to be able to detect and isolate packet loss in a network.

Network problems and under-provisioning can be detected if packet loss is measurable. TCP packet loss can be detected by observing

gaps in sequence numbers, retransmitted sequence numbers, and SACK options. Packet loss can be detected per direction.

Gaps indicate loss upstream of the tap point; retransmissions indicate loss downstream of the tap. Selective acknowledgements (SACKs) can be used to detect either upstream or downstream packet loss (although some care needs to be taken to avoid mis-identifying packet reordering as packet loss), and to distinguish between upstream vs. downstream losses.

Packet loss measurements on both sides of the measurement point are an important component in precisely diagnosing insufficiently dimensioned devices or links in networks. Additionally, since packet losses are one of the two main ways for congestion to manifest (the other being queueing delay), packet loss is an important measurement for any middlebox that needs to make traffic handling decisions based on observed levels of congestion.

## 2.2. Round Trip Times

The ability to measure partial-path round-trip times is valuable in diagnosing network issues. Knowing if latency is poor on one side of the observation point or the other provides more information than is available at either end-point, which can only observe full round-trip times.

A TCP packet stream can be used to measure the round-trip time on each side of the measurement point. During the connection handshake, the SYN, SYNACK, and ACK timings can be used to establish a baseline RTT in each direction. Once the connection is established, the RTT between the server and the measurement point can only reliably be determined using TCP timestamps. On the side between the measurement point and the client, the exact timing of data segments and ACKs can be used as an alternative. For this latter method to be accurate when packet loss is present, the connection must use selective acknowledgements.

In many networks, congestion will show up as increasing packet queueing, and congestion-induced packet loss will only happen in extreme cases. RTTs will also show up as a much smoother signal than the discrete packet loss events. This makes RTTs a good way to identify individual subscribers for whom the network is a bottleneck at a given time, or geographical sites (such as cellular towers) that are experiencing large scale congestion.

The main limit of RTT measurement as a congestion signal is the difficulty of reliably distinguishing between the data segments being queued vs. the ACKs being queued.

## 2.3. Measuring Packet Reordering

If a network is reordering packets of transport connections, caused perhaps by ECMP misconfiguration (e.g., described in [RFC2991] and [RFC7690]), the end-points may react as if packet loss is occurring and retransmit packets or reduce forwarding rates. Therefore a network operator desires the ability to diagnose packet reordering.

For TCP, packet reordering can be detected by observing TCP sequence numbers per direction. See for example a number of standard packet reordering metrics in [RFC4737] and informational metrics in [RFC5236].

## 2.4. Throughput and Bottleneck Identification

Although throughput to or from an IP address can be measured without transport-layer measurements, the transport layer provides clues about what the end-points were attempting to do.

One way of quickly excluding the network as the bottleneck during troubleshooting is to check whether the speed is limited by the endpoints. For example, the connection speed might instead be limited by suboptimal TCP options, the sender's congestion window, the sender temporarily running out of data to send, the sender waiting for the receiver to send another request, or the receiver closing the receive window.

This data is also useful for middleboxes used to measure network quality of service. Connections, or portions of connections, that are limited by the endpoints do not provide an accurate measure of network's speed, and can be discounted or completely excluded in such analyses.

## 2.5. Congestion Responsiveness

Congestion control mechanisms continue to evolve. Tools exist that can interpret protocol sequence numbers (e.g., from TCP, RTP) to infer the congestion response of a flow. Such tools can be used by operators to help understand the impact of specific transport protocols on other traffic that shares their network. For example, analysing packet sequence numbers can be used to help understand whether an application flow backs-off its load in the face of persistent congestion (as TCP does), and hence to understand whether the behaviour is appropriate for sharing limited network capacity.

These tools can also be used to determine whether mechanisms are needed in the network to prevent flows from acquiring excessive

network capacity under severe congestion (e.g., by deploying rate-limiters or network transport circuit breakers [RFC8084]).

## 2.6. Attack Detection

When an application or network resource is under attack, it is useful to identify this situation from the network perspective, upstream of the attacked resource.

Although detection methods tend to be proprietary, attack detection from within the network may comprise:

- Identifying uncharacteristic traffic volumes or sources;
- Identifying congestion, possibly using techniques in Section 2.1 and Section 2.2;
- Identifying incomplete connections or transactions, from attacks which attempt to exhaust server resources;
- Fingerprinting based on whatever available fields are determined to be useful in discriminating an attack from desirable traffic.

Two trends in protocol design will make attack detection more difficult:

- the desire to encrypt transport-layer fields;
- the desire to avoid statistical fingerprinting by adding entropy in various forms.

While improving privacy, those approaches may hinder attack detection.

## 2.7. Packet Corruption

One notable source of packet loss is packet corruption. This corruption will generally not be detected until the checksums are validated by the endpoint, and the packet is dropped. This means that detecting the exact location where packets are lost is not sufficient when troubleshooting networks. An operator would like to find out where packets are being corrupted. IP and TCP checksum verification allows a measurement device to correctly distinguish between upstream packet corruption and normal downstream packet loss.

Transport protocol designers should consider whether a middlebox will be able to detect corrupted or tampered packets.

### 2.8. Application-Layer Measurements

Network health may also be gleaned from application-layer diagnosis. E.g.,

- DNS response times and retransmissions by correlating answers to queries.
- Various protocol-aware voice and video quality analysis.

Could this type of information be provided in a transport layer?

## 3. Functions Beyond Measurement: A Few Examples

This section describes features provided by in-line devices that go beyond measurement by modifying, discarding, delaying, or prioritizing traffic.

### 3.1. NAT

Network Address Translators (NATs) allow multiple devices to share a public address by dividing the transport-layer port space among the devices.

NAT behavior recommendations are found for UDP in BCP 127 [RFC4787] and for TCP in BCP 142 [RFC7857].

To support NAT, there must be transport-layer port numbers that can be modified by the network. The application-layer must not assume the port number was left unchanged (e.g., by including it in a checksum or signing it).

Address sharing is also used in the context of IPv6 transition. For example, DS-Lite AFTR [RFC6333], NAT64 [RFC6146], or MAP-* are features that are enabled in the network to allow for IPv4 service continuity over an IPv6 network.

Further, because of some multi-homing considerations, IPv6 prefix translation may be enabled by some enterprises by means of NPTv6 [RFC6296].

### 3.2. Firewall

Firewalls are pervasive and essential components that inspect incoming and outgoing traffic. Firewalls are usually the cornerstone of a security policy that is enforced in end-user premises and other locations to provide strict guarantees about traffic that may be authorized to enter/leave the said premises, as well as end-users who

may be assigned different clearance levels regarding which networks and portions of the Internet they may acess.

An important aspect of a firewall policy is differentiating internally-initiated from externally-initiated communications.

For TCP, this is easily done by tracking the TCP state machine. Furthermore, the ending of a TCP connection is indicated by RST or FIN flags.

For UDP, the firewall can be opened if the first packet comes from an internal user, but the closing is generally done by an idle timer of arbitrary duration, which might not match the expectations of the application.

Simple IPv6 firewall capabilities for customer premises equipment (both stateless and stateful) are described in [RFC6092].

A firewall functions better when it can observe the protocol state machine, described generally by Transport-Independent Path Layer State Management [I-D.trammell-plus-statefulness].

### 3.3. DDoS Scrubbing

In the context of a distributed denial-of-service (DDoS) attack, the purpose of a scrubber is to discard attack traffic while permitting useful traffic.  E.g., such a mitigator is described in [I-D.ietf-dots-architecture].

When attacks occur against constrained resources, some traffic will be discarded before reaching the intended destination.  A user receives better experience and a server runs more efficiently if a scrubber can discard attack traffic, leaving room for legitimate traffic.

Scrubbing must be provided by an on-path network device because neither end-point of a legitimate connection has any control over the source of the attack traffic.

Source-spoofed DDoS attacks can be mitigated at the source using BCP 38 ([RFC2827]), but it is more difficult if source address filtering cannot be applied.

In contrast to devices in the core of the Internet, middleboxes statefully observing bidirectional transport connections can reject source-spoofed TCP traffic based on the inability to provide sensible acknowledgement numbers to complete the three-way handshake.

Obviously this requires middlebox visibility into transport-layer state machine.

Middleboxes may also scrub on the basis of statistical classification: testing how likely a given packet is legitimate. As protocol designers add more entropy to headers and lengths, this test becomes less useful and the best scrubbing strategy becomes random drop.

### 3.4. Implicit Identification

In order to enhance the end-user's quality of experience, some operators deploy implicit identification features that rely upon the correlation of network-related information to access some local services. For example, service portals operated by some operators may be accessed immediately by end-users without any explicit identification for the sake of improved service availability. This is doable thanks to on-path devices that inject appropriate metadata that can be used by the remote server to enforce per-subscriber policies. The information can be injected at the application layer or at the transport layer (when an address sharing mechanism is in use).

An experimental implementation using a TCP option is described in [RFC7974].

For the intended use of implicit identification, it is more secure to have a trusted middlebox mark this traffic than to trust end-user devices.

### 3.5. Performance-Enhancing Proxies

Performance-Enhancing Proxies (PEPs) can improve performance in some types of networks by improving packet spacing or generating local acknowledgements, and are most commonly used in satellite and cellular networks. Transport-Layer PEPs are described in section 2.1.1 of [RFC3135].

PEPs allow central deployment of congestion control algorithms more suited to the specific network, most commonly use of delay-based congestion control. More advanced TCP PEPs deploy congestion control systems that treat all of a single end-user's TCP connections as a single unit, improving fairness and allowing faster reaction to changing network conditions.

Local acknowledgements generated by PEPs speed up TCP slow start by splitting the effective latency, and allow for retransmissions to be done from the PEP rather than from the actual sender, saving downlink

bandwidth on retransmissions. Local acknowledgements will also allow a PEP to maintain a local buffer of data appropriate to the actual network conditions, whereas the actual endpoints would often send too much or too little.

A PEP function requires transport-layer fields that allow chunks of data to be identified (e.g., TCP sequence numbers), acknowledgements to be identified (e.g., TCP ACK numbers), and acknowledgements to be created from the PEP.

Note that PEPs are only useful in some types of networks, and poor design could make performance worse.

## 3.6. Network Coding

Network Coding is a technique for improving transmission performance over low-bandwidth, long-latency links such as satellite links. Coding may involve lossless compression and/or adding redundancy to headers and payload. A Network Coding Taxonomy is provided by [I-D.irtf-nwcrg-network-coding-taxonomy]. It is typically deployed with network-coding gateways at each end of those links, with a network-coding tunnel between them via the slow/lossy/long-latency links.

Network coding implementations may be specific to TCP, taking advantage of known properties of the protocol.

The network coding gateways may employ some techniques of PEPs, such as creating acknowledgements of queued data, removing retransmissions and pacing data rates to reduce queue oscillation.

Note: this is not to be confused with transcoding, which performs lossy compression on transmitted media streams, and not in scope for this document.

## 3.7. Network-Assisted Bandwidth Aggregation

The Hybrid Access Aggregation Point (HAAP) is a middlebox that allows customers to aggregate the bandwidth of multiple access technologies [I-D.zhang-banana-problem-statement].

One of the approaches uses MPTCP proxies [I-D.nam-mptcp-deployment-considerations] to forward traffic along multiple paths. The MPTCP proxy operates at the transport layer while being located in the operator's network.

The support of multipath transport capabilities by communicating hosts remains a privileged target design so that such hosts can

directly use the available resources provided by a variety of access networks they can connect to. Nevertheless, network operators do not control end hosts while the support of MPTCP by content servers remains marginal.

Network-Assisted MPTCP deployment models are designed to facilitate the adoption of MPTCP for the establishment of multi-path communications without making any assumption about the support of MPTCP capabilities by communicating peers. Network-Assisted MPTCP deployment models rely upon MPTCP Conversion Points (MCPs) that act on behalf of hosts so that they can take advantage of establishing communications over multiple paths [I-D.boucadair-mptcp-plain-mode].

Note there are cases when end-to-end MPTCP cannot be used even though both client and server are MPTCP-compliant. An MPTCP proxy can provide multipath utilization in these cases. Examples of such cases are listed below:

1. The use of private IPv4 addresses in some access networks. Typically, additional subflows can not be added to the MPTCP connection without the help of an MCP.

2. The assignment of IPv6 prefixes only by some networks. If the server is IPv4-only, IPv6 subflows cannot be added to an MPTCP connection established with that server, by definition.

3. Subscription to some service offerings is subject to volume quota.

### 3.8. Prioritization and Differentiated Services

Bulk traffic may be served with a higher latency than interactive traffic with no reduction in throughput. This fact allows a middlebox function to improve response times in interactive applications by prioritizing, policing, or remarking interactive transport connections differently from bulk traffic transport connections. E.g., gaming traffic may be prioritized over email or software updates.

Middleboxes may identify different classes of traffic by inspecting multiple layers of header and length of payload.

### 3.9. Measurement-Based Shaping

Basic traffic shaping functionality requires no transport-layer information. All that is needed is a way of mapping each packet to a traffic shaper quota. For example, there may be a rate limit per 5-tuple or per subscriber IP address. However, such fixed traffic

shaping rules are wasteful as they end up rate limiting traffic even when the network has free resources available.

More advanced traffic shaping devices use transport layer metrics described in Section 2 to detect congestion on either a per-site or per-user level, and use different traffic shaping rules when congestion is detected. This type of device can overcome limitations of down-stream devices that behave poorly (e.g., by excessive buffering or sub-optimally dropping packets).

### 3.10. Fairness to End-User Quota

Several service offerings rely upon a volume-based charging model. Operators may assist end-users in conserving their data quota by deploying on-path functions that shape traffic that would otherwise be agressively transferred.

For example, a fast download of a video that won't be viewed completely by the subscriber may lead to quick exhaustion of the data quota. Limiting the video download rate conserves quota for the benefit of the end-user.

## 4. Acknowledgements

The authors thank Brian Trammell, Brian Carpenter, Mirja Kuehlewind, Kathleen Moriarty, and Gorry Fairhurst for their review and suggestions.

## 5. IANA Considerations

This memo includes no request to IANA.

## 6. Security Considerations

### 6.1. Confidentiality

This document intentionally excludes middleboxes that observe or manipulate application-layer data.

The measurements and functions described in this document can all be implemented without violating confidentiality. However, there is always the question of whether the fields and packet properties used to achieve operational benefits may also be used for harm.

In particular, we want to ask what confidentiality is lost by exposing transport-layer fields beyond what can be learned by observing IP-layer fields.

Sequence numbers: an observer can learn how much data is transferred.

Start/Stop indicators: an observer can count transactions for some applications.

Device fingerprinting: an observer may be more easily able to identify a device type when different devices use different default field values or options.

### 6.2. Active Attacks

Being able to observe sequence numbers or session identifiers may make it easier to modify or terminate a transport connection. E.g., observing TCP sequence numbers allows generation of a RST packet that terminates the connection. However, signing transport fields mitigates this attack. The attack and solution are described for the TCP authentication option [RFC5925].

### 6.3. More Information Can Improve Security

Proposition: network maintainability and security can be improved by providing firewalls and DDoS mechanisms with some information about transport connections. In contrast, it would be very difficult to secure a network in which every packet appears unique and filled with random bits.

For denial-of-service (DoS) attacks on bandwidth, the receiving end-point is usually on the wrong side of the constrained network link. This fact makes it seem reasonable to give some clues to allow a middlebox device to help out before the constrained link.

E.g., in a blind attack, an attacker cannot receive data from the target of the attack (section 4.6.3.2 of [RFC3552]). In the case of TCP, the blind attacker cannot complete the three-way handshake.

In the balance, some features providing the ability to mitigate/ filter attacks and fix broken networks will improve security vs. the scenario when all packets are completely opaque.

## 7. References

### 7.1. Normative References

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

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <https://www.rfc-editor.org/info/rfc2827>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/info/rfc3552>.

[RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, DOI 10.17487/RFC4737, November 2006, <https://www.rfc-editor.org/info/rfc4737>.

[RFC4787] Audet, F., Ed. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, DOI 10.17487/RFC4787, January 2007, <https://www.rfc-editor.org/info/rfc4787>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <https://www.rfc-editor.org/info/rfc5925>.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <https://www.rfc-editor.org/info/rfc6146>.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <https://www.rfc-editor.org/info/rfc6333>.

[RFC7857] Penno, R., Perreault, S., Boucadair, M., Ed., Sivakumar, S., and K. Naito, "Updates to Network Address Translation (NAT) Behavioral Requirements", BCP 127, RFC 7857, DOI 10.17487/RFC7857, April 2016, <https://www.rfc-editor.org/info/rfc7857>.

### 7.2. Informative References

[I-D.boucadair-mptcp-plain-mode]
Boucadair, M., Jacquenet, C., Bonaventure, O., Behaghel, D., stefano.secci@lip6.fr, s., Henderickx, W., Skog, R., Vinapamula, S., Seo, S., Cloetens, W., Meyer, U., Contreras, L., and B. Peirens, "Extensions for Network-Assisted MPTCP Deployment Models", draft-boucadair-mptcp-plain-mode-10 (work in progress), March 2017.

[I-D.ietf-dots-architecture]
Mortensen, A., Andreasen, F., Reddy, T., christopher_gray3@cable.comcast.com, c., Compton, R., and N. Teague, "Distributed-Denial-of-Service Open Threat Signaling (DOTS) Architecture", draft-ietf-dots-architecture-05 (work in progress), October 2017.

[I-D.ietf-sfc-use-case-mobility]
Haeffner, W., Napper, J., Stiemerling, M., Lopez, D., and J. Uttaro, "Service Function Chaining Use Cases in Mobile Networks", draft-ietf-sfc-use-case-mobility-07 (work in progress), October 2016.

[I-D.irtf-nwcrg-network-coding-taxonomy]
Adamson, B., Adjih, C., Bilbao, J., Firoiu, V., Fitzek, F., samah.ghanem@gmail.com, s., Lochin, E., Masucci, A., Montpetit, M., Pedersen, M., Peralta, G., Roca, V., Saxena, P., and S. Sivakumar, "Taxonomy of Coding Techniques for Efficient Network Communications", draft-irtf-nwcrg-network-coding-taxonomy-07 (work in progress), February 2018.

[I-D.mm-wg-effect-encrypt]
Moriarty, K. and A. Morton, "Effects of Pervasive Encryption on Operators", draft-mm-wg-effect-encrypt-22 (work in progress), February 2018.

[I-D.nam-mptcp-deployment-considerations]
Boucadair, M., Jacquenet, C., Bonaventure, O., Henderickx, W., and R. Skog, "Network-Assisted MPTCP: Use Cases, Deployment Scenarios and Operational Considerations", draft-nam-mptcp-deployment-considerations-01 (work in progress), December 2016.

[I-D.trammell-plus-statefulness]
Kuehlewind, M., Trammell, B., and J. Hildebrand, "Transport-Independent Path Layer State Management", draft-trammell-plus-statefulness-04 (work in progress), November 2017.

[I-D.zhang-banana-problem-statement]
Cullen, M., Leymann, N., Heidemann, C., Boucadair, M., Hui, D., Zhang, M., and B. Sarikaya, "Problem Statement: Bandwidth Aggregation for Internet Access", draft-zhang-banana-problem-statement-03 (work in progress), October 2016.

[RFC2991] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, DOI 10.17487/RFC2991, November 2000, <https://www.rfc-editor.org/info/rfc2991>.

[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>.

[RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, DOI 10.17487/RFC3234, February 2002, <https://www.rfc-editor.org/info/rfc3234>.

[RFC5236] Jayasumana, A., Piratla, N., Banka, T., Bare, A., and R. Whitner, "Improved Packet Reordering Metrics", RFC 5236, DOI 10.17487/RFC5236, June 2008, <https://www.rfc-editor.org/info/rfc5236>.

[RFC5853] Hautakorpi, J., Ed., Camarillo, G., Penfield, R., Hawrylyshen, A., and M. Bhatia, "Requirements from Session Initiation Protocol (SIP) Session Border Control (SBC) Deployments", RFC 5853, DOI 10.17487/RFC5853, April 2010, <https://www.rfc-editor.org/info/rfc5853>.

[RFC6092] Woodyatt, J., Ed., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, DOI 10.17487/RFC6092, January 2011, <https://www.rfc-editor.org/info/rfc6092>.

[RFC6269] Ford, M., Ed., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, DOI 10.17487/RFC6269, June 2011, <https://www.rfc-editor.org/info/rfc6269>.

[RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, DOI 10.17487/RFC6296, June 2011, <https://www.rfc-editor.org/info/rfc6296>.

[RFC6459] Korhonen, J., Ed., Soininen, J., Patil, B., Savolainen, T., Bajko, G., and K. Iisakkila, "IPv6 in 3rd Generation Partnership Project (3GPP) Evolved Packet System (EPS)", RFC 6459, DOI 10.17487/RFC6459, January 2012, <https://www.rfc-editor.org/info/rfc6459>.

[RFC7690] Byerly, M., Hite, M., and J. Jaeggli, "Close Encounters of the ICMP Type 2 Kind (Near Misses with ICMPv6 Packet Too Big (PTB))", RFC 7690, DOI 10.17487/RFC7690, January 2016, <https://www.rfc-editor.org/info/rfc7690>.

[RFC7974] Williams, B., Boucadair, M., and D. Wing, "An Experimental TCP Option for Host Identification", RFC 7974, DOI 10.17487/RFC7974, October 2016, <https://www.rfc-editor.org/info/rfc7974>.

[RFC8084] Fairhurst, G., "Network Transport Circuit Breakers", BCP 208, RFC 8084, DOI 10.17487/RFC8084, March 2017, <https://www.rfc-editor.org/info/rfc8084>.

## Authors' Addresses

David Dolson

Email: ddolson@acm.org

Juho Snellman

Email: jsnell@iki.fi

Mohamed Boucadair
Orange
4 rue du Clos Courtel
Rennes  35000
France

Email: mohamed.boucadair@orange.com

Christian Jacquenet
Orange
4 rue du Clos Courtel
Rennes  35000
France

Email: christian.jacquenet@orange.com