                    Beneficial Functions of Middleboxes
                    draft-dolson-plus-middlebox-benefits-02

Abstract

   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.

   This document summarizes benefits provided to the Internet 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, in particular the PLUS and
   QUIC working groups, to aid understanding of what might be gained or
   lost by design decisions that may affect (or be affected by)
   middlebox operation.

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 http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any

   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on August 19, 2017.

Copyright Notice

Table of Contents

1. **Introduction**

   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.,:

   o Where a residential or business customer connects to its service
      provider(s), which may include multi-homing.

   o On the Gi interface where a GGSN connects to a PDN (see section
      3.1 of [RFC6459]).

   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].

   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 benefits of
   middleboxes using transport-layer features. This document advocates
   for transport connections to be measured and managed by the network
   for the benefit of 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.

1.1. 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

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

   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. It is therefore
   beneficial to be able to diagnose packet reordering from within a
   network.

   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. DDoS 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, DDoS attack
detection is fundamentally one of:

o detecting protocol violations by tracking the transport-layer
   state machine or application-layer messaging; or

o anomaly detection by noticing atypical traffic patterns taken from
   measurements.

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

o the desire to encrypt transport-layer communication and sequence
   numbers;

o the desire to avoid statistical fingerprinting by adding entropy
   in various forms.

Those desires assist in the worthy goal of improved privacy, but also
serve to defeat DDoS detection.

2.6. 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. It should also be possible
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.

2.7. Application-Layer Measurements

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

o DNS response times and retransmissions by correlating answers to queries.

o 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.

Arguably many users within various types of organizations would not
have been granted Internet access if not for safety provided by
firewalls.

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, there is obviously
a huge benefit in being able to scrub well.

Furthermore, this is solely a task for 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 compressing traffic or adding redundancy for transmission over low-bandwidth, long-latency links such as satellite links. One method is to deploy network-coding gateways at each end of those links, with a network-coding tunnel between them via the slow/lossy/long-latency links.

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.

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 that an MPTCP proxy can be beneficial even if both the client
and the server are MPTCP-compliant. 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 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 and Brian Carpenter 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 benefits 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
these 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,
              <http://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, <http://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,
              <http://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,
              <http://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, <http://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, <http://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, <http://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,
              <http://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,
              <http://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, "An MPTCP Option for
              Network-Assisted MPTCP", draft-boucadair-mptcp-plain-
              mode-09 (work in progress), October 2016.

[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-01 (work in progress), October 2016.

[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-02 (work in progress),
          December 2016.

[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,
          <http://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,
          <http://www.rfc-editor.org/info/rfc3135>.

[RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and
          Issues", RFC 3234, DOI 10.17487/RFC3234, February 2002,
          <http://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,
          <http://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,
              <http://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,
              <http://www.rfc-editor.org/info/rfc6092>.

   [RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix
              Translation", RFC 6296, DOI 10.17487/RFC6296, June 2011,
              <http://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,
              <http://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,
              <http://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,
              <http://www.rfc-editor.org/info/rfc7974>.

Authors' Addresses

   David Dolson
   Sandvine
   408 Albert Street
   Waterloo, ON N2L 3V3
   Canada

   Phone: +1 519 880 2400
   Email: ddolson@sandvine.com

Juho Snellman
Sandvine
Seestrasse 5
Zurich 8002
Switzerland

Email: jsnellman@sandvine.com

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