                      IP Fragmentation Considered Fragile
                      draft-bonica-intarea-frag-fragile-01

Abstract

   This document provides an overview of IP fragmentation. It explains
   how IP fragmentation works and why it is required. As part of that
   explanation, this document also explains how IP fragmentation reduces
   the reliability of Internet communication.

   This document also proposes alternatives to IP fragmentation.
   Finally, it provides recommendations for application developers and
   network operators.

Copyright Notice

Table of Contents

1. Introduction

   Operational experience [RFC7872] [Huston] reveals that IP
   fragmentation reduces the reliability of Internet communication.
   This document provides an overview of IP fragmentation. It explains
   how IP fragmentation works and why it is required. As part of that
   explanation, this document also explains how IP fragmentation reduces
   the reliability of Internet communication.

   This document also proposes alternatives to IP fragmentation.
   Finally, it provides recommendations for application developers and
   network operators.

2. IP Fragmentation

2.1. Links, Paths, MTU and PMTU

   An Internet path connects a source node to a destination node. A
   path can contain links and intermediate systems. If a path contains
   more than one link, the links are connected in series and an
   intermediate system connects each link to the next. An intermediate
   system can be a router or a middle box.

   Internet paths are dynamic. Assume that the path from one node to
   another contains a set of links and intermediate systems. If the
   network topology changes, that path can also change so that it
   includes a different set of links and intermediate systems.

   Each link is constrained by the number of bytes that it can convey in
   a single IP packet. This constraint is called the link Maximum
   Transmission Unit (MTU). IPv4 [RFC0791] requires every link to have
   an MTU of 68 bytes or greater. IPv6 [RFC8200] requires every link to
   have an MTU of 1280 bytes or greater. These are called the IPv4 and
   IPv6 minimum link MTU's.

   Each Internet path is constrained by the number of bytes that it can
   convey in a IP single packet. This constraint is called the Path MTU
   (PMTU). For any given path, the PMTU is equal to the smallest of its
   link MTU's. Because Internet paths are dynamic, PMTU is also
   dynamic.

   For reasons described below, source nodes estimate the PMTU between
   themselves and destination nodes. A source node can produce
   extremely conservative PMTU estimates in which:

   o The estimate for each IPv4 path is equal to IPv4 minimum link MTU
      (68 bytes).

o The estimate for each IPv6 path is equal to the IPv6 minimum link
   MTU (1280 bytes).

While these conservative estimates are guaranteed to be less than or
equal to the actual MTU, they are likely to be much less than the
actual PMTU. This may adversely affect upper-layer protocol
performance.

By executing Path MTU Discovery (PMTUD) [RFC1191] [RFC8201]
procedures, a source node can maintain a less conservative, running
estimate of the PMTU between itself and a destination node.
According to these procedures, the source node produces an initial
PMTU estimate. This initial estimate is equal to the MTU of the
first link along the path to the destination node. It can be greater
than the actual PMTU.

Having produced an initial PMTU estimate, the source node sends non-
fragmentable IP packets to the destination node. If one of these
packets is larger than the actual PMTU, a downstream router will not
be able to forward the packet through the next link along the path.
Therefore, the downstream router drops the packet and send an
Internet Control Message Protocol (ICMP) [RFC0792] [RFC4443] Packet
Too Big (PTB) message to the source node. The ICMP PTB message
indicates the MTU of the link through which the packet could not be
forwarded. The source node uses this information to refine its PMTU
estimate.

PMTUD produces a running estimate of the PMTU between a source node
and a destination node. Because PMTU is dynamic, at any given time,
the PMTU estimate can differ from the actual PMTU. In order to
detect PMTU increases, PMTUD occasionally resets the PMTU estimate to
the MTU of the first link along path to the destination node. It
then repeats the procedure described above.

Furthermore, PMTUD has the following characteristics:

o It relies on the network's ability to deliver ICMP PTB messages to
   the source node.

o It is susceptible to attack because ICMP messages are easily
   forged [RFC5927].

FOOTNOTE: According to RFC 0791, every IPv4 host must be capable of
receiving a packet whose length is equal to 576 bytes. However, the
IPv4 minimum link MTU is not 576. Section 3.2 of RFC 0791 explicitly
states that the IPv4 minimum link MTU is 68 bytes.

FOOTNOTE: In the paragraphs above, the term "non-fragmentable packet"
is introduced. A non-fragmentable packet can be fragmented at its
source. However, it cannot be fragmented by a downstream node. An
IPv4 packet whose DF-bit is set to zero is fragmentable. An IPv4
packet whose DF-bit is set to one is non-fragmentable. All IPv6
packets are also non-fragmentable.

FOOTNOTE: In the paragraphs above, the term "ICMP PTB message" is
introduced. The ICMP PTB message has two instantiations. In ICMPv4
[RFC0792], the ICMP PTB message is Destination Unreachable message
with Code equal to (4) fragmentation needed and DF set. This message
was augmented by [RFC1191] to indicates the MTU of the link through
which the packet could not be forwarded. In ICMPv6 [RFC4443], the
ICMP PTB message is a Packet Too Big Message with Code equal to (0).
This message also indicates the MTU of the link through which the
packet could not be forwarded.

2.2. Upper-layer Protocols

When an upper-layer protocol submits data to the underlying IP
module, and the resulting IP packet's length is greater than the
PMTU, IP fragmentation may be required. IP fragmentation divides a
packet into fragments. Each fragment includes an IP header and a
portion of the original packet.

[RFC0791] describes IPv4 fragmentation procedures. IPv4 packets
whose DF-bit is set to one cannot be fragmented. IPv4 packets whose
DF-bit is set to zero can be fragmented at the source node or by any
downstream router. [RFC8200] describes IPv6 fragmentation
procedures. IPv6 packets can be fragmented at the source node only.

IPv4 fragmentation differs slightly from IPv6 fragmentation.
However, in both IP versions, the upper-layer header appears in the
first fragment only. It does not appear in subsequent fragments.

Upper-layer protocols can operate in the following modes:

o Do not rely on IP fragmentation.

o Rely on IP source fragmentation only (i.e., fragmentation at the
   source node).

o Rely on IP source fragmentation and downstream fragmentation
   (i.e., fragmentation at any node along the path).

Upper-layer protocols running over IPv4 can operate in the first and
third modes (above). Upper-layer protocols running over IPv6 can
operate in the first and second modes (above).

Upper-layer protocols that operate in the first two modes (above) require access to the PMTU estimate. In order to fulfil this requirement, they can

o Estimate the PMTU to be equal to the IPv4 or IPv6 minimum link MTU.

o Access the estimate that PMTUD produced.

o Execute PMTUD procedures themselves.

o Execute Packetization Layer PMTUD (PLPMTUD) [RFC4821] [I-D.fairhurst-tsvwg-datagram-plpmtud] procedures.

According to PLPMTUD procedures, the upper-layer protocol maintains a running PMTU estimate. It does so by sending probe packets of various sizes to its peer and receiving acknowledgements. This strategy differs from PMTUD in that it relies of acknowledgement of received messages, as opposed to ICMP PTB messages concerning dropped messages. Therefore, PLPMTUD does not rely on the network's ability to deliver ICMP PTB messages to the source.

An upper-layer protocol that does not rely on IP fragmentation never causes the underlying IP module to emit

o A fragmentable IP packet (i.e., an IPv4 packet with the DF-bit set to zero).

o An IP fragment.

o A packet whose length is greater than the PMTU estimate.

However, when the PMTU estimate is greater than the actual PMTU, the upper-layer protocol can cause the underlying IP module to emit a packet whose length is greater than the actual PMTU. When this occurs, a downstream router drops the packet and the source node refines its PMTU estimate, employing either PMTUD or PLPMTUD procedures.

When an upper-layer protocol that relies on IP source fragmentation only submits data to the underlying IP module, and the resulting packet is larger than the PMTU estimate, the underlying IP module fragments the packet and emits the fragments. However, the upper-layer protocol never causes the underlying IP module to emit

o A fragmentable IP packet.

o A packet whose length is greater than the PMTU estimate.

When the PMTU estimate is greater than the actual PMTU, the upper-
layer protocol can cause the underlying IP module to emit a packet
whose length is greater than the actual PMTU. When this occurs, a
downstream router drops the packet and the source node refines its
PMTU estimate, employing either PMTUD or PLPMTUD procedures.

An upper-layer protocol that relies on IP source fragmentation and
downstream fragmentation can cause the underlying IP module to emit

o A fragmentable IP packet.

o An IP fragment.

o A packet whose length is greater than the PMTU estimate.

A protocol that relies on IP source fragmentation and downstream
fragmentation does not require access to the PMTU estimate. For
these protocols, the underlying IP module:

o Fragments all packets whose length exceeds the MTU of the first
   link along the path to the destination.

o Sets the DF-bit to zero, so that downstream nodes can fragment the
   packet.

3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
"OPTIONAL" in this document are to be interpreted as described in BCP
14 [RFC2119] [RFC8174] when, and only when, they appear in all
capitals, as shown here.

4. IP Fragmentation Reduces Reliability

This section explains how IP fragmentation reduces the reliability of
Internet communication.

4.1. Middle Box Failures

Many middle boxes require access to the transport-layer header.
However, when a packet is divided into fragments, the transport-layer
header appears in the first fragment only. It does not appear in
subsequent fragments. This omission can prevent middle boxes from
delivering their intended services.

For example, assume that a router diverts selected packets from their
normal path towards network appliances that support deep packet

   inspection and lawful intercept. The router selects packets for
   diversion based upon the following 5-tuple:

   o IP Source Address.

   o IP Destination Address.

   o IPv4 Protocol or IPv6 Next Header.

   o transport-layer source port.

   o transport-layer destination port.

   IP fragmentation causes this selection algorithm to behave
   suboptimally, because the transport-layer header appears only in the
   first fragment of each packet.

   In another example, a middle box remarks a packet's Differentiated
   Services Code Point [RFC2474] based upon the above mentioned 5-tuple.
   IP fragmentation causes this process to behave suboptimally, because
   the transport-layer header appears only in the first fragment of each
   packet.

   In all of the above-mentioned examples, the middle box cannot deliver
   its intended service without reassembling fragmented packets.

4.2. Partial Filtering

   IP fragments cause problems for firewalls whose filter rules include
   decision making based on TCP and UDP ports. As the port information
   is not in the trailing fragments the firewall may elect to accept all
   trailing fragments, which may admit certain classes of attack, or may
   elect to block all trailing fragments, which may block otherwise
   legitimate traffic, or may elect to reassemble all fragmented
   packets, which may be inefficient and negatively affect performance.

4.3. Suboptimal Load Balancing

   Many stateless load-balancers require access to the transport-layer
   header. Assume that a load-balancer distributes flows among parallel
   links. In order to optimize load balancing, the load-balancer sends
   every packet or packet fragment belonging to a flow through the same
   link.

   In order to assign a packet or packet fragment to a link, the load-
   balancer executes an algorithm. If the packet or packet fragment
   contains a transport-layer header, the load balancing algorithm
   accepts the following 5-tuple as input:

o IP Source Address.

o IP Destination Address.

o IPv4 Protocol or IPv6 Next Header.

o transport-layer source port.

o transport-layer destination port.

However, if the packet or packet fragment does not contain a
transport-layer header, the load balancing algorithm accepts only the
following 3-tuple as input:

o IP Source Address.

o IP Destination Address.

o IPv4 Protocol or IPv6 Next Header.

Therefore, non-fragmented packets belonging to a flow can be assigned
to one link while fragmented packets belonging to the same flow can
be divided between that link and another. This can cause suboptimal
load balancing.

4.4. Security Vulnerabilities

Security researchers have documented several attacks that rely on IP
fragmentation. The following are examples:

o Overlapping fragment attack [RFC1858] [RFC5722]

o Resource exhaustion attacks (such as the Rose Attack)

o Attacks based on predictable fragment Identification values
   [RFC7739]

o Attacks based on bugs in the implementation of the fragment
   reassembly algorithm

o Evasion of Network Intrusion Detection Systems (NIDS) [Ptacek1998]

In the overlapping fragment attack, an attacker constructs a series
of packet fragments. The first fragment contains an IP header, a
transport-layer header, and some transport-layer payload. This
fragment complies with local security policy and is allowed to pass
through a stateless firewall. A second fragment, having a non-zero
offset, overlaps with the first fragment. The second fragment also

passes through the stateless firewall. When the packet is
reassembled, the transport layer header from the first fragment is
overwritten by data from the second fragment. The reassembled packet
does not comply with local security policy. Had it traversed the
firewall in one piece, the firewall would have rejected it.

A stateless firewall cannot protect against the overlapping fragment
attack. However, destination nodes can protect against the
overlapping fragment attack by implementing the reassembly procedures
described in RFC 1858 and RFC 8200. These reassembly procedures
detect the overlap and discard the packet.

The fragment reassembly algorithm is a stateful procedure for an
otherwise stateless protocol. As such, it can be exploited for
resource exhaustion attacks. An attacker can construct a series of
fragmented packets, with one fragment missing from each packet such
that the reassembly process cannot complete. Thus, this attack
causes resource exhaustion on the destination node, possibly denying
reassembly services to other flows. This type of attack can be
mitigated by flushing fragment reassembly buffers when necessary, at
the expense of possibly dropping legitimate fragments.

An IP fragment contains an "Identification" field that, together with
the IP Source Address and Destination Address of a packet, identifies
fragments that correspond to the same original datagram, such that
they can be reassembled together by the receiving host. Many
implementations have employed predictable values for the
Identification field, thus making it easy for an attacker to forge
malicious IP fragments that would cause the reassembly procedure for
legitimate packets to fail.

Over the years multiple IPv4 and IPv6 implementations have been found
to have flaws in their implementation of the IP fragment reassembly
algorithm, typically resulting in buffer overflows. These buffer
overflows have been exploitable for denial of service and remote code
execution attacks.

NIDS aims at identifying malicious activity by analyzing network
traffic. Ambiguity in the possible result of the fragment reasembly
process may allow an attacker to evade these systems. Many of these
systems try to mitigate some of these evasion techniques by e.g.
computing all possible outcomes of the fragment reassembly process,
at the expense of increased processing requirements.

4.5. Blackholing Due to ICMP Loss

   As stated above, an upper-layer protocol requires access the PMTU
   estimate if it:

   o Does not rely on IP fragmentation.

   o Relies on IP source fragmentation only (i.e., fragmentation at the
      source node).

   In order to satisfy this requirement, the upper-layer protocol can:

   o Estimate the PMTU to be equal to the IPv4 or IPv6 minimum link
      MTU.

   o Access the estimate that PMTUD produced.

   o Execute PMTUD procedures itself.

   o Execute PLPMTUD procedures.

   PMTUD relies upon the network's ability to deliver ICMP PTB messages
   to the source node. Therefore, if an upper-layer protocol relies on
   PMTUD for its PMTU estimate, it also relies on the networks ability
   to deliver ICMP PTB messages to the source node.

   [RFC4890] states that the PTB messages must not be filtered.
   However, ICMP delivery is not reliable. It is subject to transient
   loss and, in some configurations, more persistent delivery issues.

   ICMP rate limiting, network congestion and packet corruption can
   cause transient loss. The effect of rate limiting may be severe, as
   RFC 4443 recommends strict rate limiting of IPv6 traffic.

   While transient loss causes PMTUD to perform less efficiently, it
   does not cause PMTUD to fail completely. When the conditions
   contributing to transient loss abate, the network regains its ability
   to deliver ICMP PTB messages and PMTUD regains its ability to
   function.

   By contrast, more persistent delivery issues cause PMTUD to fail
   completely. Consider the following example:

   A DNS client sends a request to an anycast address. The network
   routes that DNS request to the nearest instance of that anycast
   address (i.e., a DNS Server). The DNS server generates a response
   and sends it back to the DNS client. While the response does not

exceed the DNS server's PMTU estimate, it does exceed the actual
PMTU.

A downstream router drops the packet and sends an ICMP PTB message
the packet's source (i.e., the anycast address). The network routes
the ICMP PTB message to the anycast instance closest to the
downstream router. Sadly, that anycast instance may not be the DNS
server that originated the DNS response. It may be another DNS
server with the same anycast address. The DNS server that originated
the response may never receive the ICMP PTB message and may never
updates it PMTU estimate.

The problem described in this section is specific to PMTUD. It does
not occur when the upper-layer protocol obtains its PMTU estimate
from PLPMTUD or any other source.

Furthermore, the problem described in this section occurs when the
upper-layer protocol does not rely on IP fragmentation, as well as
when the upper-layer protocol relies on IP source fragmentation only.

4.6. Blackholing Due To Filtering

In RFC 7872, researchers sampled Internet paths to determine whether
they would convey packets that contain IPv6 extension headers.
Sampled paths terminated at popular Internet sites (e.g., popular
web, mail and DNS servers).

The study revealed that at least 28% of the sampled paths did not
convey packets containing the IPv6 Fragment extension header. In
most cases, fragments were dropped in the destination autonomous
system. In other cases, the fragments were dropped in transit
autonomous systems.

Another recent study [Huston] confirmed this finding. It reported
that 37% of sampled endpoints used IPv6-capable DNS resolvers that
were incapable of receiving a fragmented IPv6 response.

It is difficult to determine why network operators drop fragments.
In some cases, packet drop may be caused by misconfiguration. In
other cases, network operators may consciously choose to drop IPv6
fragments, in order to address the issues raised in Section 4.1
through Section 4.5, above.

5. Alternatives to IP Fragmentation

5.1. Transport Layer Solutions

   The Transport Control Protocol (TCP) [RFC0793]) can be operated in a
   mode that does not require IP fragmentation.

   Applications submit a stream of data to TCP. TCP divides that stream
   of data into segments, with no segment exceeding the TCP Maximum
   Segment Size (MSS). Each segment is encapsulated in a TCP header and
   submitted to the underlying IP module. The underlying IP module
   prepends an IP header and forwards the resulting packet.

   If the TCP MSS is sufficiently small, the underlying IP module never
   produces a packet whose length is greater than the actual PMTU.
   Therefore, IP fragmentation is not required.

   TCP offers the following mechanisms for MSS management:

   o Manual configuration

   o PMTUD

   o PLPMTUD

   For IPv6 nodes, manual configuration is always applicable. If the
   MSS is manually configured to 1220 bytes and the packet does not
   contain extension headers, the IP layer will never produce a packet
   whose length is greater than the IPv6 minimum link MTU (1280 bytes).
   However, manual configuration prevents TCP from taking advantage of
   larger link MTU's.

   RFC 8200 strongly recommends that IPv6 nodes implement PMTUD, in
   order to discover and take advantage of path MTUs greater than 1280
   bytes. However, as mentioned in Section 2.1, PMTUD relies upon the
   network's ability to deliver ICMP PTB messages. Therefore, PMTUD is
   applicable only in environments where the risk of ICMP PTB loss is
   acceptable.

   By contrast, PLPMTUD does not rely upon the network's ability to
   deliver ICMP PTB messages. However, in many loss-based TCP
   congestion control algorithms, the dropping of a packet may cause the
   TCP control algorithm to drop the congestion control window, or even
   re-start with the entire slow start process. For high capacity, long
   RTT, large volume TCP streams, the deliberate probing with large
   packets and the consequent packet drop may impose too harsh a penalty
   on total TCP throughput for it to be a viable approach. [RFC4821]
   defines PLPMTUD procedures for TCP.

While TCP will never cause the underlying IP module to emit a packet that is larger than the PMTU estimate, it can cause the underlying IP module to emit a packet that is larger than the actual PMTU. If this occurs, the packet is dropped, the PMTU estimate is updated, the segment is divided into smaller segments and each smaller segment is submitted to the underlying IP module.

The Datagram Congestion Control Protocol (DCCP) [RFC4340] and the Stream Control Protocol (SCP) [RFC4960] also can be operated in a mode that does not require IP fragmentation. They both accept data from an application and divide that data into segments, with no segment exceeding a maximum size. Both DCCP and SCP offer manual configuration, PMTUD and PLPMTUD as mechanisms for managing that maximum size. [I-D.fairhurst-tsvwg-datagram-plpmtud] proposes PLPMTUD procedures for DCCP and SCP.

5.2. Application Layer Solutions

[RFC8085] recognizes that IP fragmentation reduces the reliability of Internet communication. Therefore, it offers the following advice regarding applications the run over the User Data Protocol (UDP) [RFC0768].

"An application SHOULD NOT send UDP datagrams that result in IP packets that exceed the Maximum Transmission Unit (MTU) along the path to the destination. Consequently, an application SHOULD either use the path MTU information provided by the IP layer or implement Path MTU Discovery (PMTUD) itself to determine whether the path to a destination will support its desired message size without fragmentation."

RFC 8085 continues:

"Applications that do not follow the recommendation to do PMTU/ PLPMTUD discovery SHOULD still avoid sending UDP datagrams that would result in IP packets that exceed the path MTU. Because the actual path MTU is unknown, such applications SHOULD fall back to sending messages that are shorter than the default effective MTU for sending (EMTU_S in [RFC1122]). For IPv4, EMTU_S is the smaller of 576 bytes and the first-hop MTU. For IPv6, EMTU_S is 1280 bytes. The effective PMTU for a directly connected destination (with no routers on the path) is the configured interface MTU, which could be less than the maximum link payload size. Transmission of minimum-sized UDP datagrams is inefficient over paths that support a larger PMTU, which is a second reason to implement PMTU discovery."

RFC 8085 assumes that for IPv4, an EMTU_S of 576 is sufficiently small, even though the IPv4 minimum link MTU is 68 bytes.

   This advice applies equally to application that run directly over IP.

6. Applications That Rely on IPv6 Fragmentation

   The following applications rely on IPv6 fragmentation:

   o DNS [RFC1035]

   o OSPFv3 [RFC5340]

   o IP Encapsulations

   Each of these applications relies on IPv6 fragmentation to a varying
   degree. In some cases, that reliance is essential, and cannot be
   broken without fundamentally changing the protocol. In other cases,
   that reliance is incidental, and most implementations already take
   appropriate steps to avoid fragmentation.

   This list is not comprehensive, and other protocols that rely on IPv6
   fragmentation may exist. They are not specifically considered in the
   context of this document.

6.1. DNS

   DNS can obtain transport services from either UDP or TCP. Superior
   performance and scaling characteristics are observed when DNS runs
   over UDP.

   DNS Servers that execute DNSSEC [RFC4035] procedures are more likely
   to generate large responses. Therefore, when running over UDP, they
   are more likely to cause the generation of IPv6 fragments. DNS's
   reliance upon IPv6 fragmentation is fundamental and cannot be broken
   without changing the DNS specification.

   DNS is an essential part of the Internet architecture. Therefore,
   this issue is for further study and must be resolved before DNSSEC
   can be deployed successfully in IPv6 only networks.

6.2. OSPFv3

   OSPFv3 implementations can emit messages large enough to cause IPv6
   fragmentation. However, in keeping with the recommendations of
   RFC8200, and in order to optimize performance, most OSPFv3
   implementations restrict their maximum message size to the IPv6
   minimum link MTU.

6.3. IP Encapsulations

   In this document, IP encapsulations include IP-in-IP [RFC2003],
   Generic Routing Encapsulation (GRE) [RFC2784], GRE-in-UDP [RFC8086]
   and Generic Packet Tunneling in IPv6 [RFC2473]. The fragmentation
   strategy described for GRE in [RFC7588] has been deployed for all of
   the above-mentioned IP encapsulations. This strategy does not rely
   on IPv6 fragmentation except in one corner case. (see Section 3.3.2.2
   of RFC 7588 and Section 7.1 of RFC 2473). Section 3.3 of [RFC7676]
   further describes this corner case.

7. Recommendation

7.1. For Application Developers

   Application developers SHOULD NOT develop applications that rely on
   IPv6 fragmentation.

   Application-layer protocols then depend upon IPv6 fragmentation
   SHOULD be updated to break that dependency.

7.2. For Network Operators

   As per RFC 4890, network operators MUST NOT filter ICMPv6 PTB
   messages unless they are known to be forged or otherwise
   illegitimate. As stated in Section 4.5, filtering ICMPv6 PTB packets
   causes PMTUD to fail. Many upper-layer protocols rely on PMTUD.

8. IANA Considerations

   This document makes no request of IANA.

9. Security Considerations

   This document mitigates some of the security considerations
   associated with IP fragmentation by discouraging the use of IP
   fragmentation. It does not introduce any new security
   vulnerabilities, because it does not introduce any new alternatives
   to IP fragmentation. Instead, it recommends well-understood
   alternatives.

10. Acknowledgements

   TBD

11. References

11.1. Normative References

   [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768,
              DOI 10.17487/RFC0768, August 1980,
              <https://www.rfc-editor.org/info/rfc768>.

   [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791,
              DOI 10.17487/RFC0791, September 1981,
              <https://www.rfc-editor.org/info/rfc791>.

   [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5,
              RFC 792, DOI 10.17487/RFC0792, September 1981,
              <https://www.rfc-editor.org/info/rfc792>.

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

   [RFC1035] Mockapetris, P., "Domain names - implementation and
              specification", STD 13, RFC 1035, DOI 10.17487/RFC1035,
              November 1987, <https://www.rfc-editor.org/info/rfc1035>.

   [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191,
              DOI 10.17487/RFC1191, November 1990,
              <https://www.rfc-editor.org/info/rfc1191>.

   [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>.

   [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet
              Control Message Protocol (ICMPv6) for the Internet
              Protocol Version 6 (IPv6) Specification", STD 89,
              RFC 4443, DOI 10.17487/RFC4443, March 2006,
              <https://www.rfc-editor.org/info/rfc4443>.

   [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU
              Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007,
              <https://www.rfc-editor.org/info/rfc4821>.

   [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage
              Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085,
              March 2017, <https://www.rfc-editor.org/info/rfc8085>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <https://www.rfc-editor.org/info/rfc8174>.

   [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6
              (IPv6) Specification", STD 86, RFC 8200,
              DOI 10.17487/RFC8200, July 2017,
              <https://www.rfc-editor.org/info/rfc8200>.

   [RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed.,
              "Path MTU Discovery for IP version 6", STD 87, RFC 8201,
              DOI 10.17487/RFC8201, July 2017,
              <https://www.rfc-editor.org/info/rfc8201>.

11.2. Informative References

   [Anderson2001]
              Anderson, J., "An Analysis of Fragmentation Attacks",
              2001, <http://www.ouah.org/fragma.html>.

   [Huston]   Huston, G., "IPv6, Large UDP Packets and the DNS
              (http://www.potaroo.net/ispcol/2017-08/xtn-hdrs.html)",
              August 2017.

   [I-D.fairhurst-tsvwg-datagram-plpmtud]
              Fairhurst, G., Jones, T., Tuexen, M., and I. Ruengeler,
              "Packetization Layer Path MTU Discovery for Datagram
              Transports", draft-fairhurst-tsvwg-datagram-plpmtud-02
              (work in progress), December 2017.

   [Ptacek1998]
              Ptacek, T. and T. Newsham, "Insertion, Evasion and Denial
              of Service: Eluding Network Intrusion Detection", 1998,
              <http://www.aciri.org/vern/Ptacek-Newsham-Evasion-98.ps>.

   [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts -
              Communication Layers", STD 3, RFC 1122,
              DOI 10.17487/RFC1122, October 1989,
              <https://www.rfc-editor.org/info/rfc1122>.

   [RFC1858] Ziemba, G., Reed, D., and P. Traina, "Security
              Considerations for IP Fragment Filtering", RFC 1858,
              DOI 10.17487/RFC1858, October 1995,
              <https://www.rfc-editor.org/info/rfc1858>.

   [RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003,
              DOI 10.17487/RFC2003, October 1996,
              <https://www.rfc-editor.org/info/rfc2003>.

   [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in
              IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473,
              December 1998, <https://www.rfc-editor.org/info/rfc2473>.

   [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black,
              "Definition of the Differentiated Services Field (DS
              Field) in the IPv4 and IPv6 Headers", RFC 2474,
              DOI 10.17487/RFC2474, December 1998,
              <https://www.rfc-editor.org/info/rfc2474>.

   [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P.
              Traina, "Generic Routing Encapsulation (GRE)", RFC 2784,
              DOI 10.17487/RFC2784, March 2000,
              <https://www.rfc-editor.org/info/rfc2784>.

   [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S.
              Rose, "Protocol Modifications for the DNS Security
              Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005,
              <https://www.rfc-editor.org/info/rfc4035>.

   [RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram
              Congestion Control Protocol (DCCP)", RFC 4340,
              DOI 10.17487/RFC4340, March 2006,
              <https://www.rfc-editor.org/info/rfc4340>.

   [RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering
              ICMPv6 Messages in Firewalls", RFC 4890,
              DOI 10.17487/RFC4890, May 2007,
              <https://www.rfc-editor.org/info/rfc4890>.

   [RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol",
              RFC 4960, DOI 10.17487/RFC4960, September 2007,
              <https://www.rfc-editor.org/info/rfc4960>.

   [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF
              for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008,
              <https://www.rfc-editor.org/info/rfc5340>.

   [RFC5722] Krishnan, S., "Handling of Overlapping IPv6 Fragments",
              RFC 5722, DOI 10.17487/RFC5722, December 2009,
              <https://www.rfc-editor.org/info/rfc5722>.

   [RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927,
              DOI 10.17487/RFC5927, July 2010,
              <https://www.rfc-editor.org/info/rfc5927>.

   [RFC7588] Bonica, R., Pignataro, C., and J. Touch, "A Widely
              Deployed Solution to the Generic Routing Encapsulation
              (GRE) Fragmentation Problem", RFC 7588,
              DOI 10.17487/RFC7588, July 2015,
              <https://www.rfc-editor.org/info/rfc7588>.

   [RFC7676] Pignataro, C., Bonica, R., and S. Krishnan, "IPv6 Support
              for Generic Routing Encapsulation (GRE)", RFC 7676,
              DOI 10.17487/RFC7676, October 2015,
              <https://www.rfc-editor.org/info/rfc7676>.

   [RFC7739] Gont, F., "Security Implications of Predictable Fragment
              Identification Values", RFC 7739, DOI 10.17487/RFC7739,
              February 2016, <https://www.rfc-editor.org/info/rfc7739>.

   [RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu,
              "Observations on the Dropping of Packets with IPv6
              Extension Headers in the Real World", RFC 7872,
              DOI 10.17487/RFC7872, June 2016,
              <https://www.rfc-editor.org/info/rfc7872>.

   [RFC8086] Yong, L., Ed., Crabbe, E., Xu, X., and T. Herbert, "GRE-
              in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086,
              March 2017, <https://www.rfc-editor.org/info/rfc8086>.

Appendix A. Contributors' Address

Authors' Addresses

   Ron Bonica
   Juniper Networks
   2251 Corporate Park Drive
   Herndon, Virginia 20171
   USA

   Email: rbonica@juniper.net

   Fred Baker
   Unaffiliated
   Santa Barbara, California 93117
   USA

   Email: FredBaker.IETF@gmail.com

Geoff Huston
APNIC
6 Cordelia St
Brisbane, 4101 QLD
Australia

Email: gih@apnic.net

Robert M. Hinden
Check Point Software
959 Skyway Road
San Carlos, California 94070
USA

Email: bob.hinden@gmail.com

Ole Troan
Cisco
Philip Pedersens vei 1
N-1366 Lysaker
Norway

Email: ot@cisco.com

Fernando Gont
SI6 Networks
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires
Argentina

Email: fgont@si6networks.com