< draft-ietf-intarea-frag-fragile-13.txt	draft-ietf-intarea-frag-fragile-14.txt >
Internet Area WG R. Bonica	Internet Area WG R. Bonica
Internet-Draft Juniper Networks	Internet-Draft Juniper Networks
Intended status: Best Current Practice F. Baker	Intended status: Best Current Practice F. Baker
Expires: December 26, 2019 Unaffiliated	Expires: January 6, 2020 Unaffiliated
G. Huston	G. Huston
APNIC	APNIC
R. Hinden	R. Hinden
Check Point Software	Check Point Software
O. Troan	O. Troan
Cisco	Cisco
F. Gont	F. Gont
SI6 Networks	SI6 Networks
June 24, 2019	July 5, 2019
IP Fragmentation Considered Fragile	IP Fragmentation Considered Fragile
draft-ietf-intarea-frag-fragile-13	draft-ietf-intarea-frag-fragile-14
Abstract	Abstract
This document describes IP fragmentation and explains how it	This document describes IP fragmentation and explains how it
introduces fragility to Internet communication.	introduces fragility to Internet communication.
This document also proposes alternatives to IP fragmentation and	This document also proposes alternatives to IP fragmentation and
provides recommendations for developers and network operators.	provides recommendations for developers and network operators.
Status of This Memo	Status of This Memo
skipping to change at *page 1, line 43*	**skipping to change at *page 1, line 43***
Internet-Drafts are working documents of the Internet Engineering	Internet-Drafts are working documents of the Internet Engineering
Task Force (IETF). Note that other groups may also distribute	Task Force (IETF). Note that other groups may also distribute
working documents as Internet-Drafts. The list of current Internet-	working documents as Internet-Drafts. The list of current Internet-
Drafts is at https://datatracker.ietf.org/drafts/current/.	Drafts is at https://datatracker.ietf.org/drafts/current/.
Internet-Drafts are draft documents valid for a maximum of six months	Internet-Drafts are draft documents valid for a maximum of six months
and may be updated, replaced, or obsoleted by other documents at any	and may be updated, replaced, or obsoleted by other documents at any
time. It is inappropriate to use Internet-Drafts as reference	time. It is inappropriate to use Internet-Drafts as reference
material or to cite them other than as "work in progress."	material or to cite them other than as "work in progress."
This Internet-Draft will expire on December 26, 2019.	This Internet-Draft will expire on January 6, 2020.
Copyright Notice	Copyright Notice
Copyright (c) 2019 IETF Trust and the persons identified as the	Copyright (c) 2019 IETF Trust and the persons identified as the
document authors. All rights reserved.	document authors. All rights reserved.
This document is subject to BCP 78 and the IETF Trust's Legal	This document is subject to BCP 78 and the IETF Trust's Legal
Provisions Relating to IETF Documents	Provisions Relating to IETF Documents
(https://trustee.ietf.org/license-info) in effect on the date of	(https://trustee.ietf.org/license-info) in effect on the date of
publication of this document. Please review these documents	publication of this document. Please review these documents
carefully, as they describe your rights and restrictions with respect	carefully, as they describe your rights and restrictions with respect
to this document. Code Components extracted from this document must	to this document. Code Components extracted from this document must
include Simplified BSD License text as described in Section 4.e of	include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as	the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.	described in the Simplified BSD License.
Table of Contents	Table of Contents
1. Introduction 3	1. Introduction 3
1.1. IP-in-IP Tunnels 3	1.1. IP-in-IP Tunnels 3
2. IP Fragmentation 3	1.2. Requirements Language 3
2.1. Links, Paths, MTU and PMTU 3	2. IP Fragmentation 4
2.2. Fragmentation Procedures 5	2.1. Links, Paths, MTU and PMTU 4
	2.2. Fragmentation Procedures 6
2.3. Upper-Layer Reliance on IP Fragmentation 6	2.3. Upper-Layer Reliance on IP Fragmentation 6
3. Requirements Language 7	3. Increased Fragility 7
4. Increased Fragility 7	3.1. Virtual Reassembly 7
4.1. Policy-Based Routing 7	3.2. Policy-Based Routing 8
4.2. Network Address Translation (NAT) 8	3.3. Network Address Translation (NAT) 9
4.3. Stateless Firewalls 9	3.4. Stateless Firewalls 9
4.4. Equal Cost Multipath, Link Aggregate Groups and Stateless	3.5. Equal Cost Multipath, Link Aggregate Groups and Stateless
Load-Balancers 9	Load-Balancers 9
4.5. IPv4 Reassembly Errors at High Data Rates 10	3.6. IPv4 Reassembly Errors at High Data Rates 11
4.6. Security Vulnerabilities 11	3.7. Security Vulnerabilities 11
4.7. PMTU Blackholing Due to ICMP Loss 12	3.8. PMTU Blackholing Due to ICMP Loss 12
4.7.1. Transient Loss 12	3.8.1. Transient Loss 13
4.7.2. Incorrect Implementation of Security Policy 13	3.8.2. Incorrect Implementation of Security Policy 13
4.7.3. Persistent Loss Caused By Anycast 13	3.8.3. Persistent Loss Caused By Anycast 14
4.7.4. Persistent Loss Caused By Unidirectional Routing . . 14	3.8.4. Persistent Loss Caused By Unidirectional Routing . . 14
4.8. Blackholing Due To Filtering or Loss 14	3.9. Blackholing Due To Filtering or Loss 14
5. Alternatives to IP Fragmentation 15	4. Alternatives to IP Fragmentation 15
5.1. Transport Layer Solutions 15	4.1. Transport Layer Solutions 15
5.2. Application Layer Solutions 16	4.2. Application Layer Solutions 16
6. Applications That Rely on IPv6 Fragmentation 17	5. Applications That Rely on IPv6 Fragmentation 17
6.1. Domain Name Service (DNS) 17	5.1. Domain Name Service (DNS) 18
6.2. Open Shortest Path First (OSPF) 18	5.2. Open Shortest Path First (OSPF) 18
6.3. Packet-in-Packet Encapsulations 18	5.3. Packet-in-Packet Encapsulations 18
6.4. UDP Applications Enhancing Performance 18	5.4. UDP Applications Enhancing Performance 19
7. Recommendations 19	6. Recommendations 19
7.1. For Application and Protocol Developers 19	6.1. For Application and Protocol Developers 19
7.2. For System Developers 19	6.2. For System Developers 20
7.3. For Middle Box Developers 19	6.3. For Middle Box Developers 20
7.4. For ECMP, LAG and Load-Balancer Developers And Operators 20	6.4. For ECMP, LAG and Load-Balancer Developers And Operators 20
7.5. For Network Operators 20	6.5. For Network Operators 20
8. IANA Considerations 21	
9. Security Considerations 21	7. IANA Considerations 21
10. Acknowledgements 21	8. Security Considerations 21
11. References 21	9. Acknowledgements 21
11.1. Normative References 21	10. References 21
11.2. Informative References 23	10.1. Normative References 21
	10.2. Informative References 23

1. Introduction

Operational experience [Kent] [Huston] [RFC7872] reveals that IP fragmentation introduces fragility to Internet communication. This document describes IP fragmentation and explains the fragility it introduces. It also proposes alternatives to IP fragmentation and provides recommendations for developers and network operators.

While this document identifies issues associated with IP fragmentation, it does not recommend deprecation. Legacy protocols that depend upon IP fragmentation SHOULD be updated to remove that dependency. However, some applications and environments (see Section 6) require IP fragmentation. In these cases, the protocol will continue to rely on IP fragmentation, but the designer should to be aware that fragmented packets may result in blackholes; a design should include appropriate safeguards.

Rather than deprecating IP Fragmentation, this document recommends that upper-layer protocols address the problem of fragmentation at their layer, reducing their reliance on IP fragmentation to the greatest degree possible.

1.1. IP-in-IP Tunnels

This document acknowledges that in some cases, packets must be fragmented within IP-in-IP tunnels [I-D.ietf-intarea-tunnels]. Therefore, this document makes no additional recommendations regarding IP-in-IP tunnels.

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 routers. If a path contains more than one link, the links are connected in series and a router connects each link to the next.

Internet paths are dynamic. Assume that the path from one node to

1. Introduction

Operational experience [Kent] [Huston] [RFC7872] reveals that IP fragmentation introduces fragility to Internet communication. This document describes IP fragmentation and explains the fragility it introduces. It also proposes alternatives to IP fragmentation and provides recommendations for developers and network operators.

While this document identifies issues associated with IP fragmentation, it does not recommend deprecation. Legacy protocols that depend upon IP fragmentation SHOULD be updated to remove that dependency. However, some applications and environments (see Section 5) require IP fragmentation. In these cases, the protocol will continue to rely on IP fragmentation, but the designer should to be aware that fragmented packets may result in blackholes; a design should include appropriate safeguards.

Rather than deprecating IP Fragmentation, this document recommends that upper-layer protocols address the problem of fragmentation at their layer, reducing their reliance on IP fragmentation to the greatest degree possible.

1.1. IP-in-IP Tunnels

This document acknowledges that in some cases, packets must be fragmented within IP-in-IP tunnels [I-D.ietf-intarea-tunnels]. Therefore, this document makes no additional recommendations regarding IP-in-IP tunnels.

1.2. 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.

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 routers. If a path contains more than one link, the links are connected in series and a router connects each link to the next.

Internet paths are dynamic. Assume that the path from one node to

skipping to change at *page 5, line 34* skipping to change at *page 5, line 45*

whose length is equal to 576 bytes. However, the IPv4 minimum link MTU is not 576. Section 3.2 of RFC 791 explicitly states that the IPv4 minimum link MTU is 68 bytes. But for practical purposes, many network operators consider the IPv4 minimum link MTU to be 576 bytes, to minimize the requirement for fragmentation en route. So, for the purposes of this document, we assume that the IPv4 minimum path MTU is 576 bytes.

NOTE 2: 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.

NOTE 3:: The ICMP PTB message has two instantiations. In ICMPv4 [RFC0792], the ICMP PTB message is a Destination Unreachable message with Code equal to (4) fragmentation needed and DF set. This message was augmented by [RFC1191] to indicate 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. Fragmentation Procedures

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

[RFC0791] describes IPv4 fragmentation procedures. An IPv4 packet whose DF-bit is set to one can be fragmented by the source node, but cannot be fragmented by a downstream router. An IPv4 packet whose DF-bit is set to zero can be fragmented by the source node or by a downstream router. When an IPv4 packet is fragmented, all IP options appear in the first fragment, but only options whose "copy" bit is set to one appear in subsequent fragments.

[RFC8200] describes IPv6 fragmentation procedures. An IPv6 packet can be fragmented at the source node only. When an IPv6 packet is fragmented, all extension headers appear in the first fragment, but only per-fragment headers appear in subsequent fragments. Per-fragment headers include the following:

o The IPv6 header.

o The Hop-by-hop Options header (if present)

whose length is equal to 576 bytes. However, the IPv4 minimum link MTU is not 576. Section 3.2 of RFC 791 explicitly states that the IPv4 minimum link MTU is 68 bytes. But for practical purposes, many network operators consider the IPv4 minimum link MTU to be 576 bytes, to minimize the requirement for fragmentation en route. So, for the purposes of this document, we assume that the IPv4 minimum path MTU is 576 bytes.

NOTE 2: 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 (0) is fragmentable. An IPv4 packet whose DF-bit is set to (1) is non-fragmentable. All IPv6 packets are also non-fragmentable.

NOTE 3:: The ICMP PTB message has two instantiations. In ICMPv4 [RFC0792], the ICMP PTB message is a Destination Unreachable message with Code equal to (4) fragmentation needed and DF set. This message was augmented by [RFC1191] to indicate 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. Fragmentation Procedures

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

[RFC0791] describes IPv4 fragmentation procedures. An IPv4 packet whose DF-bit is set to (1) can be fragmented by the source node, but cannot be fragmented by a downstream router. An IPv4 packet whose DF-bit is set to (0) can be fragmented by the source node or by a downstream router. When an IPv4 packet is fragmented, all IP options appear in the first fragment, but only options whose "copy" bit is set to (1) appear in subsequent fragments.

[RFC8200] describes IPv6 fragmentation procedures. An IPv6 packet can be fragmented at the source node only. When an IPv6 packet is fragmented, all extension headers appear in the first fragment, but only per-fragment headers appear in subsequent fragments. Per-fragment headers include the following:

o The IPv6 header.

o The Hop-by-hop Options header (if present)

skipping to change at *page 7, line 20* skipping to change at *page 7, line 33*

[I-D.ietf-tsvwg-datagram-plpmtud] procedures.

[I-D.ietf-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 upper-layer peer and receiving acknowledgements. This strategy differs from PMTUD in that it relies on 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.	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 upper-layer peer and receiving acknowledgements. This strategy differs from PMTUD in that it relies on 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.
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. Increased Fragility	3. Increased Fragility
This section explains how IP fragmentation introduces fragility to Internet communication.	This section explains how IP fragmentation introduces fragility to Internet communication.
4.1. Policy-Based Routing	3.1. Virtual Reassembly Virtual reassembly is a procedure in which a device reassembles a packet, forwards its fragments, and discards the reassembled copy. In A+P and CGN, virtual reassembly is required in order to correctly translate fragment addresses. It can be useful in Section 3.2, Section 3.3, Section 3.4, and Section 3.5. Virtual reassembly in the network is problematic, however, because it is computationally expensive and because it holds state for indeterminate periods of time, is prone to errors and, is prone to attacks (Section 3.7). 3.2. Policy-Based Routing
IP Fragmentation causes problems for routers that implement policy-based routing.	IP Fragmentation causes problems for routers that implement policy-based routing.
When a router receives a packet, it identifies the next-hop on route to the packet's destination and forwards the packet to that next-hop. In order to identify the next-hop, the router interrogates a local data structure called the Forwarding Information Base (FIB).	When a router receives a packet, it identifies the next-hop on route to the packet's destination and forwards the packet to that next-hop. In order to identify the next-hop, the router interrogates a local data structure called the Forwarding Information Base (FIB).
Normally, the FIB contains destination-based entries that map a	Normally, the FIB contains destination-based entries that map a
skipping to change at ***page 8, line 42***	**skipping to change at** ***page 9, line 9***
When the router receives the second fragment of the packet, it interrogates the FIB again. This time, only the first FIB entry satisfies the query, because the second fragment contains no indication that the packet is destined for TCP port 80. Therefore, the router selects the first FIB entry and forwards the packet to 2001:db8::2.	When the router receives the second fragment of the packet, it interrogates the FIB again. This time, only the first FIB entry satisfies the query, because the second fragment contains no indication that the packet is destined for TCP port 80. Therefore, the router selects the first FIB entry and forwards the packet to 2001:db8::2.
Policy-based routing is also known as filter-based-forwarding.	Policy-based routing is also known as filter-based-forwarding.
4.2. Network Address Translation (NAT)	3.3. Network Address Translation (NAT)
IP fragmentation causes problems for Network Address Translation (NAT) devices. When a NAT device detects a new, outbound flow, it maps that flow's source port and IP address to another source port and IP address. Having created that mapping, the NAT device translates:	IP fragmentation causes problems for Network Address Translation (NAT) devices. When a NAT device detects a new, outbound flow, it maps that flow's source port and IP address to another source port and IP address. Having created that mapping, the NAT device translates:
o The Source IP Address and Source Port on each outbound packet.	o The Source IP Address and Source Port on each outbound packet.
o The Destination IP Address and Destination Port on each inbound packet.	o The Destination IP Address and Destination Port on each inbound packet.
A+P [RFC6346] and Carrier Grade NAT (CGN) [RFC6888] are two common NAT strategies. In both approaches the NAT device must virtually reassemble fragmented packets in order to translate and forward each fragment. (See NOTE 1.)	A+P [RFC6346] and Carrier Grade NAT (CGN) [RFC6888] are two common NAT strategies. In both approaches the NAT device must virtually reassemble fragmented packets in order to translate and forward each fragment. (See NOTE 1.)
Virtual reassembly in the network is problematic, because it is computationally expensive and because it is prone to attacks (Section 4.6). NOTE 1: Virtual reassembly is a procedure in which a device reassembles a packet, forwards its fragments, and discards the reassembled copy. In A+P and CGN, virtual reassembly is required in order to correctly translate fragment addresses. 4.3. Stateless Firewalls	3.4. Stateless Firewalls
As discussed in more detail in Section 4.6, IP fragmentation causes problems for stateless firewalls whose rules include TCP and UDP ports. Because port information is not available in the trailing fragments the firewall is limited to the following options:	As discussed in more detail in Section 3.7, IP fragmentation causes problems for stateless firewalls whose rules include TCP and UDP ports. Because port information is not available in the trailing fragments the firewall is limited to the following options:
o Accept all trailing fragments, possibly admitting certain classes of attack.	o Accept all trailing fragments, possibly admitting certain classes of attack.
o Block all trailing fragments, possibly blocking legitimate traffic.	o Block all trailing fragments, possibly blocking legitimate traffic.
Neither option is attractive.	Neither option is attractive.
4.4. Equal Cost Multipath, Link Aggregate Groups and Stateless Load-Balancers	3.5. Equal Cost Multipath, Link Aggregate Groups and Stateless Load-Balancers

IP fragmentation causes problems for Equal Cost Multipath (ECMP), Link Aggregate Groups (LAG) and other stateless load-balancing technologies. In order to assign a packet or packet fragment to a link, an intermediate node executes a hash (i.e., load-balancing) algorithm. The following paragraphs describe a commonly deployed hash algorithm.	IP fragmentation causes problems for Equal Cost Multipath (ECMP), Link Aggregate Groups (LAG) and other stateless load-balancing technologies. In order to assign a packet or packet fragment to a link, an intermediate node executes a hash (i.e., load-balancing) algorithm. The following paragraphs describe a commonly deployed hash algorithm.
If the packet or packet fragment contains a transport-layer header,	If the packet or packet fragment contains a transport-layer header,
skipping to change at *page 10, line 39*	**skipping to change at *page 10, line 46***
algorithm used to determine the outgoing component-link in an ECMP and/or LAG toward the next hop MUST minimally include the 3-tuple {dest addr, source addr, flow label} and MAY also include the remaining components of the 5-tuple."	algorithm used to determine the outgoing component-link in an ECMP and/or LAG toward the next hop MUST minimally include the 3-tuple {dest addr, source addr, flow label} and MAY also include the remaining components of the 5-tuple."
If the algorithm includes only the 3-tuple {dest addr, source addr, flow label}, it will assign all fragments belonging to a packet to the same link. (See [RFC6437] and [RFC7098]).	If the algorithm includes only the 3-tuple {dest addr, source addr, flow label}, it will assign all fragments belonging to a packet to the same link. (See [RFC6437] and [RFC7098]).
In order to avoid the problem described above, implementations SHOULD implement the recommendations provided in Section 7.4 of this document.	In order to avoid the problem described above, implementations SHOULD implement the recommendations provided in Section 6.4 of this document.
4.5. IPv4 Reassembly Errors at High Data Rates	3.6. IPv4 Reassembly Errors at High Data Rates
IPv4 fragmentation is not sufficiently robust for use under some conditions in today's Internet. At high data rates, the 16-bit IP identification field is not large enough to prevent frequent incorrectly assembled IP fragments, and the TCP and UDP checksums are insufficient to prevent the resulting corrupted datagrams from being delivered to higher protocol layers. [RFC4963] describes some easily reproduced experiments demonstrating the problem, and discusses some of the operational implications of these observations.	IPv4 fragmentation is not sufficiently robust for use under some conditions in today's Internet. At high data rates, the 16-bit IP identification field is not large enough to prevent duplicate IDs resulting in frequent incorrectly assembled IP fragments, and the TCP and UDP checksums are insufficient to prevent the resulting corrupted datagrams from being delivered to higher protocol layers. [RFC4963] describes some easily reproduced experiments demonstrating the problem, and discusses some of the operational implications of these observations.
These reassembly issues are not easily reproducible in IPv6 because the IPv6 identification field is 32 bits long.	These reassembly issues do not occur as frequently in IPv6 because the IPv6 identification field is 32 bits long.
4.6. Security Vulnerabilities	3.7. Security Vulnerabilities
Security researchers have documented several attacks that exploit IP fragmentation. The following are examples:	Security researchers have documented several attacks that exploit IP fragmentation. The following are examples:
o Overlapping fragment attacks [RFC1858][RFC3128][RFC5722]	o Overlapping fragment attacks [RFC1858][RFC3128][RFC5722]
o Resource exhaustion attacks	o Resource exhaustion attacks
o Attacks based on predictable fragment identification values [RFC7739]	o Attacks based on predictable fragment identification values [RFC7739]
skipping to change at *page 12, line 14*	**skipping to change at *page 12, line 28***
for an attacker to forge malicious IP fragments that would cause the reassembly procedure for legitimate packets to fail.	for an attacker to forge malicious IP fragments that would cause the reassembly procedure for legitimate packets to fail.
NIDS aims at identifying malicious activity by analyzing network traffic. Ambiguity in the possible result of the fragment reassembly process may allow an attacker to evade these systems. Many of these systems try to mitigate some of these evasion techniques (e.g. By computing all possible outcomes of the fragment reassembly process, at the expense of increased processing requirements).	NIDS aims at identifying malicious activity by analyzing network traffic. Ambiguity in the possible result of the fragment reassembly process may allow an attacker to evade these systems. Many of these systems try to mitigate some of these evasion techniques (e.g. By computing all possible outcomes of the fragment reassembly process, at the expense of increased processing requirements).
4.7. PMTU Blackholing Due to ICMP Loss	3.8. PMTU Blackholing Due to ICMP Loss
As mentioned in Section 2.3, upper-layer protocols can be configured to rely on PMTUD. Because PMTUD relies upon the network to deliver ICMP PTB messages, those protocols also rely on the networks to deliver ICMP PTB messages.	As mentioned in Section 2.3, upper-layer protocols can be configured to rely on PMTUD. Because PMTUD relies upon the network to deliver ICMP PTB messages, those protocols also rely on the networks to deliver ICMP PTB messages.
According to [RFC4890], ICMP PTB messages must not be filtered. However, ICMP PTB delivery is not reliable. It is subject to both transient and persistent loss.	According to [RFC4890], ICMP PTB messages must not be filtered. However, ICMP PTB delivery is not reliable. It is subject to both transient and persistent loss.
Transient loss of ICMP PTB messages can cause transient PMTU black holes. When the conditions contributing to transient loss abate, the network regains its ability to deliver ICMP PTB messages and connectivity between the source and destination nodes is restored. Section 4.7.1 of this document describes conditions that lead to transient loss of ICMP PTB messages.	Transient loss of ICMP PTB messages can cause transient PMTU black holes. When the conditions contributing to transient loss abate, the network regains its ability to deliver ICMP PTB messages and connectivity between the source and destination nodes is restored. Section 3.8.1 of this document describes conditions that lead to transient loss of ICMP PTB messages.
Persistent loss of ICMP PTB messages can cause persistent black holes. Section 4.7.2, Section 4.7.3, and Section 4.7.4 of this document describe conditions that lead to persistent loss of ICMP PTB messages.	Persistent loss of ICMP PTB messages can cause persistent black holes. Section 3.8.2, Section 3.8.3, and Section 3.8.4 of this document describe conditions that lead to persistent loss of ICMP PTB messages.
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 from any other source.	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 from any other source.
4.7.1. Transient Loss	3.8.1. Transient Loss
The following factors can contribute to transient loss of ICMP PTB messages:	The following factors can contribute to transient loss of ICMP PTB messages:
o Network congestion.	o Network congestion.
o Packet corruption.	o Packet corruption.
o Transient routing loops.	o Transient routing loops.
o ICMP rate limiting.	o ICMP rate limiting.

The effect of rate limiting may be severe, as RFC 4443 recommends strict rate limiting of IPv6 traffic.	The effect of rate limiting may be severe, as RFC 4443 recommends strict rate limiting of IPv6 traffic.
4.7.2. Incorrect Implementation of Security Policy	3.8.2. Incorrect Implementation of Security Policy
Incorrect implementation of security policy can cause persistent loss of ICMP PTB messages.	Incorrect implementation of security policy can cause persistent loss of ICMP PTB messages.
Assume that a Customer Premise Equipment (CPE) router implements the following zone-based security policy:	Assume that a Customer Premise Equipment (CPE) router implements the following zone-based security policy:
o Allow any traffic to flow from the inside zone to the outside zone.	o Allow any traffic to flow from the inside zone to the outside zone.
skipping to change at *page 13, line 40*	**skipping to change at *page 14, line 8***
allows the ICMP PTB to flow from the outside zone to the inside zone. If not, the implementation discards the ICMP PTB message.	allows the ICMP PTB to flow from the outside zone to the inside zone. If not, the implementation discards the ICMP PTB message.
When a incorrect implementation of the above-mentioned security policy receives an ICMP PTB message, it discards the packet because its source address is not associated with an existing flow.	When a incorrect implementation of the above-mentioned security policy receives an ICMP PTB message, it discards the packet because its source address is not associated with an existing flow.
The security policy described above is implemented incorrectly on many consumer CPE routers.	The security policy described above is implemented incorrectly on many consumer CPE routers.
4.7.3. Persistent Loss Caused By Anycast	3.8.3. Persistent Loss Caused By Anycast
Anycast can cause persistent loss of ICMP PTB messages. Consider the example below:	Anycast can cause persistent loss of ICMP PTB messages. Consider the example below:
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 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. 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 update its PMTU estimate.	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. 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 update its PMTU estimate.
4.7.4. Persistent Loss Caused By Unidirectional Routing	3.8.4. Persistent Loss Caused By Unidirectional Routing
Unidirectional routing can cause persistent loss of ICMP PTB messages. Consider the example below:	Unidirectional routing can cause persistent loss of ICMP PTB messages. Consider the example below:
A source node sends a packet to a destination node. All intermediate nodes maintain a route to the destination node, but do not maintain a route to the source node. In this case, when an intermediate node encounters an MTU issue, it cannot send an ICMP PTB message to the source node.	A source node sends a packet to a destination node. All intermediate nodes maintain a route to the destination node, but do not maintain a route to the source node. In this case, when an intermediate node encounters an MTU issue, it cannot send an ICMP PTB message to the source node.
4.8. Blackholing Due To Filtering or Loss	3.9. Blackholing Due To Filtering or Loss
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).	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	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
skipping to change at *page 15, line 7*	**skipping to change at *page 15, line 20***
Possible causes follow:	Possible causes follow:
o Hardware inability to process fragmented packets.	o Hardware inability to process fragmented packets.
o Failure to change vendor defaults.	o Failure to change vendor defaults.
o Unintentional misconfiguration.	o Unintentional misconfiguration.
o Intentional configuration (e.g., network operators consciously chooses to drop IPv6 fragments in order to address the issues raised in Section 4.1 through Section 4.7, above.)	o Intentional configuration (e.g., network operators consciously chooses to drop IPv6 fragments in order to address the issues raised in Section 3.2 through Section 3.8, above.)
5. Alternatives to IP Fragmentation	4. Alternatives to IP Fragmentation
5.1. Transport Layer Solutions	4.1. Transport Layer Solutions
The Transport Control Protocol (TCP) [RFC0793]) can be operated in a mode that does not require IP fragmentation.	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.	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.
skipping to change at *page 16, line 34*	**skipping to change at *page 16, line 47***
implement PLPMTUD to estimate the PMTU via[I-D.ietf-tsvwg-datagram-plpmtud]. This proposes procedures for performing PLPMTUD with UDP, UDP-Options, SCTP, QUIC and other datagram protocols.	implement PLPMTUD to estimate the PMTU via[I-D.ietf-tsvwg-datagram-plpmtud]. This proposes procedures for performing PLPMTUD with UDP, UDP-Options, SCTP, QUIC and other datagram protocols.

Currently, User Data Protocol (UDP) [RFC0768] lacks a fragmentation mechanism of its own and relies on IP fragmentation. However, [I-D.ietf-tsvwg-udp-options] proposes a fragmentation mechanism for UDP.

5.2. Application Layer Solutions

[RFC8085] recognizes that IP fragmentation reduces the reliability of Internet communication. It also recognizes that UDP lacks a fragmentation mechanism of its own and relies on IP fragmentation.

Therefore, [RFC8085] offers the following advice regarding applications the run over the UDP.

"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."

skipping to change at *page 17, line 25*

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 is sufficiently small to be supported by most current Internet paths, even though the IPv4 minimum link MTU is 68 bytes.

This advice applies equally to any application that runs directly over IP.

6. Applications That Rely on IPv6 Fragmentation

The following applications rely on IPv6 fragmentation:

o DNS [RFC1035]

o OSPFv3 [RFC2328][RFC5340]

o Packet-in-packet 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 IP fragmentation may exist. They are not specifically considered in the context of this document.

6.1. Domain Name Service (DNS)

DNS relies on UDP for efficiency, and the consequence is the use of IP fragmentation for large responses, as permitted by the DNS EDNS(0) options in the query. It is possible to mitigate the issue of fragmentation-based packet loss by having queries use smaller EDNS(0) UDP buffer sizes, or by having the DNS server limit the size of its UDP responses to some self-imposed maximum packet size that may be less than the preferred EDNS(0) UDP Buffer Size. In both cases, large responses are truncated in the DNS, signalling to the client to re-query using TCP to obtain the complete response. However, the

skipping to change at *page 18, line 20*

Larger DNS responses can normally be avoided by aggressively pruning the Additional section of DNS responses. One scenario where such pruning is ineffective is in the use of DNSSEC, where large key sizes act to increase the response size to certain DNS queries. There is no effective response to this situation within the DNS other than using smaller cryptographic keys and adoption of DNSSEC administrative practices that attempt to keep DNS response as short as possible.

6.2. Open Shortest Path First (OSPF)

OSPF implementations can emit messages large enough to cause fragmentation. However, in order to optimize performance, most OSPF implementations restrict their maximum message size to a value that will not cause fragmentation.

6.3. Packet-in-Packet Encapsulations

In this document, packet-in-packet encapsulations include IP-in-IP [RFC2003], Generic Routing Encapsulation (GRE) [RFC2784], GRE-in-UDP [RFC8086] and Generic Packet Tunneling in IPv6 [RFC2473]. [RFC4459] describes fragmentation issues associated with all of the above-mentioned encapsulations.

The fragmentation strategy described for GRE in [RFC7588] has been deployed for all of the above-mentioned encapsulations. This strategy does not rely on IP 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.

See [I-D.ietf-intarea-tunnels] for further discussion.

6.4. UDP Applications Enhancing Performance

Currently, User Data Protocol (UDP) [RFC0768] lacks a fragmentation mechanism of its own and relies on IP fragmentation. However, [I-D.ietf-tsvwg-udp-options] proposes a fragmentation mechanism for UDP.

4.2. Application Layer Solutions

[RFC8085] recognizes that IP fragmentation reduces the reliability of Internet communication. It also recognizes that UDP lacks a fragmentation mechanism of its own and relies on IP fragmentation.

Therefore, [RFC8085] offers the following advice regarding applications the run over the UDP.

"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."

skipping to change at *page 17, line 38*

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 is sufficiently small to be supported by most current Internet paths, even though the IPv4 minimum link MTU is 68 bytes.

This advice applies equally to any application that runs directly over IP.

5. Applications That Rely on IPv6 Fragmentation

The following applications rely on IPv6 fragmentation:

o DNS [RFC1035]

o OSPFv3 [RFC2328][RFC5340]

o Packet-in-packet 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 IP fragmentation may exist. They are not specifically considered in the context of this document.

5.1. Domain Name Service (DNS)

DNS relies on UDP for efficiency, and the consequence is the use of IP fragmentation for large responses, as permitted by the DNS EDNS(0) options in the query. It is possible to mitigate the issue of fragmentation-based packet loss by having queries use smaller EDNS(0) UDP buffer sizes, or by having the DNS server limit the size of its UDP responses to some self-imposed maximum packet size that may be less than the preferred EDNS(0) UDP Buffer Size. In both cases, large responses are truncated in the DNS, signalling to the client to re-query using TCP to obtain the complete response. However, the

skipping to change at *page 18, line 33*

Larger DNS responses can normally be avoided by aggressively pruning the Additional section of DNS responses. One scenario where such pruning is ineffective is in the use of DNSSEC, where large key sizes act to increase the response size to certain DNS queries. There is no effective response to this situation within the DNS other than using smaller cryptographic keys and adoption of DNSSEC administrative practices that attempt to keep DNS response as short as possible.

5.2. Open Shortest Path First (OSPF)

OSPF implementations can emit messages large enough to cause fragmentation. However, in order to optimize performance, most OSPF implementations restrict their maximum message size to a value that will not cause fragmentation.

5.3. Packet-in-Packet Encapsulations

In this document, packet-in-packet encapsulations include IP-in-IP [RFC2003], Generic Routing Encapsulation (GRE) [RFC2784], GRE-in-UDP [RFC8086] and Generic Packet Tunneling in IPv6 [RFC2473]. [RFC4459] describes fragmentation issues associated with all of the above-mentioned encapsulations.

The fragmentation strategy described for GRE in [RFC7588] has been deployed for all of the above-mentioned encapsulations. This strategy does not rely on IP 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.

See [I-D.ietf-intarea-tunnels] for further discussion.

5.4. UDP Applications Enhancing Performance

Some UDP applications rely on IP fragmentation to achieve acceptable levels of performance. These applications use UDP datagram sizes that are larger than the path MTU so that more data can be conveyed between the application and the kernel in a single system call.	Some UDP applications rely on IP fragmentation to achieve acceptable levels of performance. These applications use UDP datagram sizes that are larger than the path MTU so that more data can be conveyed between the application and the kernel in a single system call.
To pick one example, the Licklider Transmission Protocol (LTP), [RFC5326]which is in current use on the International Space Station (ISS), uses UDP datagram sizes larger than the path MTU to achieve acceptable levels of performance even though this invokes IP fragmentation. More generally, SNMP and video applications may transmit an application-layer quantum of data, depending on the network layer to fragment and reassemble as needed.	To pick one example, the Licklider Transmission Protocol (LTP), [RFC5326]which is in current use on the International Space Station (ISS), uses UDP datagram sizes larger than the path MTU to achieve acceptable levels of performance even though this invokes IP fragmentation. More generally, SNMP and video applications may transmit an application-layer quantum of data, depending on the network layer to fragment and reassemble as needed.
7. Recommendations	6. Recommendations
7.1. For Application and Protocol Developers	6.1. For Application and Protocol Developers
Developers SHOULD NOT develop new protocols or applications that rely on IP fragmentation. When a new protocol or application is deployed in an environment that does not fully support IP fragmentation, it SHOULD operate correctly, either in its default configuration or in a specified alternative configuration.	Developers SHOULD NOT develop new protocols or applications that rely on IP fragmentation. When a new protocol or application is deployed in an environment that does not fully support IP fragmentation, it SHOULD operate correctly, either in its default configuration or in a specified alternative configuration.
Developers MAY develop new protocols or applications that rely on IP fragmentation if the protocol or application is to be run only in environments where IP fragmentation is known to be supported.	Developers MAY develop new protocols or applications that rely on IP fragmentation if the protocol or application is to be run only in environments where IP fragmentation is known to be supported.
skipping to change at ***page 19, line 41***	**skipping to change at** ***page 20, line 5***
Protocols may be able to avoid IP fragmentation by using a sufficiently small MTU (e.g. The protocol minimum link MTU), disabling IP fragmentation, and ensuring that the transport protocol in use adapts its segment size to the MTU. Other protocols may deploy a sufficiently reliable PMTU discovery mechanism (e.g.,PLMPTUD).	Protocols may be able to avoid IP fragmentation by using a sufficiently small MTU (e.g. The protocol minimum link MTU), disabling IP fragmentation, and ensuring that the transport protocol in use adapts its segment size to the MTU. Other protocols may deploy a sufficiently reliable PMTU discovery mechanism (e.g.,PLMPTUD).
UDP applications SHOULD abide by the recommendations stated in Section 3.2 of [RFC8085].	UDP applications SHOULD abide by the recommendations stated in Section 3.2 of [RFC8085].
7.2. For System Developers	6.2. For System Developers
Software libraries SHOULD include provision for PLPMTUD for each supported transport protocol.	Software libraries SHOULD include provision for PLPMTUD for each supported transport protocol.
7.3. For Middle Box Developers	6.3. For Middle Box Developers
Middle boxes should process IP fragments in a manner that is consistent with [RFC0791] and [RFC8200]. In many cases, middle boxes must maintain state in order to achieve this goal.	Middle boxes should process IP fragments in a manner that is consistent with [RFC0791] and [RFC8200]. In many cases, middle boxes must maintain state in order to achieve this goal.
Price and performance considerations frequently motivate network operators to deploy stateless middle boxes. These stateless middle boxes may perform sub-optimally, process IP fragments in a manner that is not compliant with RFC 791 or RFC 8200, or even discard IP fragments completely. Such behaviors are NOT RECOMMENDED. If a middleboxes implements non-standard behavior with respect to IP fragmentation, then that behavior MUST be clearly documented.	Price and performance considerations frequently motivate network operators to deploy stateless middle boxes. These stateless middle boxes may perform sub-optimally, process IP fragments in a manner that is not compliant with RFC 791 or RFC 8200, or even discard IP fragments completely. Such behaviors are NOT RECOMMENDED. If a middleboxes implements non-standard behavior with respect to IP fragmentation, then that behavior MUST be clearly documented.
7.4. For ECMP, LAG and Load-Balancer Developers And Operators	6.4. For ECMP, LAG and Load-Balancer Developers And Operators
In their default configuration, when the IPv6 Flow Label is not equal to zero, IPv6 devices that implement Equal-Cost Multipath (ECMP) Routing as described in OSPF [RFC2328] and other routing protocols, Link Aggregation Grouping (LAG) [RFC7424], or other load-balancing technologies SHOULD accept only the following fields as input to their hash algorithm:	In their default configuration, when the IPv6 Flow Label is not equal to zero, IPv6 devices that implement Equal-Cost Multipath (ECMP) Routing as described in OSPF [RFC2328] and other routing protocols, Link Aggregation Grouping (LAG) [RFC7424], or other load-balancing technologies SHOULD accept only the following fields as input to their hash algorithm:
o IP Source Address.	o IP Source Address.
o IP Destination Address.	o IP Destination Address.
o Flow Label.	o Flow Label.
Operators SHOULD deploy these devices in their default configuration.	Operators SHOULD deploy these devices in their default configuration.
These recommendations are similar to those presented in [RFC6438] and [RFC7098]. They differ in that they specify a default configuration.	These recommendations are similar to those presented in [RFC6438] and [RFC7098]. They differ in that they specify a default configuration.
7.5. For Network Operators	6.5. For Network Operators
Operators MUST ensure proper PMTUD operation in their network, including making sure the network generates PTB packets when dropping packets too large compared to outgoing interface MTU. However, implementations MAY rate limit ICMP messages as per [RFC1812] and [RFC4443].	Operators MUST ensure proper PMTUD operation in their network, including making sure the network generates PTB packets when dropping packets too large compared to outgoing interface MTU. However, implementations MAY rate limit ICMP messages as per [RFC1812] and [RFC4443].
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 4.7, filtering ICMPv6 PTB packets causes PMTUD to fail. Many upper-layer protocols rely on PMTUD.	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 3.8, filtering ICMPv6 PTB packets causes PMTUD to fail. Many upper-layer protocols rely on PMTUD.
As per RFC 8200, network operators MUST NOT deploy IPv6 links whose MTU is less than 1280 bytes.	As per RFC 8200, network operators MUST NOT deploy IPv6 links whose MTU is less than 1280 bytes.
Network operators SHOULD NOT filter IP fragments if they are known to have originated at a domain name server or be destined for a domain name server. This is because domain name services are critical to operation of the Internet.	Network operators SHOULD NOT filter IP fragments if they are known to have originated at a domain name server or be destined for a domain name server. This is because domain name services are critical to operation of the Internet.
8. IANA Considerations	7. IANA Considerations
This document makes no request of IANA.	This document makes no request of IANA.

9. Security Considerations	8. Security Considerations
This document mitigates some of the security considerations associated with IP fragmentation by discouraging its use. 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.	This document mitigates some of the security considerations associated with IP fragmentation by discouraging its use. 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	9. Acknowledgements
Thanks to Mikael Abrahamsson, Brian Carpenter, Silambu Chelvan, Lorenzo Colitti, Gorry Fairhurst, Mike Heard, Tom Herbert, Tatuya Jinmei, Jen Linkova, Paolo Lucente, Manoj Nayak, Eric Nygren, Fred Templin and Joe Touch for their comments.	Thanks to Mikael Abrahamsson, Brian Carpenter, Silambu Chelvan, Lorenzo Colitti, Gorry Fairhurst, Mike Heard, Tom Herbert, Tatuya Jinmei, Jen Linkova, Paolo Lucente, Manoj Nayak, Eric Nygren, Fred Templin and Joe Touch for their comments.
11. References	10. References
11.1. Normative References	10.1. Normative References
[I-D.ietf-tsvwg-datagram-plpmtud] Fairhurst, G., Jones, T., Tuexen, M., Ruengeler, I., and T. Voelker, "Packetization Layer Path MTU Discovery for Datagram Transports", draft-ietf-tsvwg-datagram-plpmtud-08 (work in progress), June 2019.	[I-D.ietf-tsvwg-datagram-plpmtud] Fairhurst, G., Jones, T., Tuexen, M., Ruengeler, I., and T. Voelker, "Packetization Layer Path MTU Discovery for Datagram Transports", draft-ietf-tsvwg-datagram-plpmtud-08 (work in progress), June 2019.
[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.	[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.
skipping to change at *page 23, line 5*	**skipping to change at *page 23, line 19***
[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>.	[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>.	[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	10.2. Informative References
[Damas] Damas, J. and G. Huston, "Measuring ATR", April 2018, <http://www.potaroo.net/ispcol/2018-04/atr.html>.	[Damas] Damas, J. and G. Huston, "Measuring ATR", April 2018, <http://www.potaroo.net/ispcol/2018-04/atr.html>.
[Huston] Huston, G., "IPv6, Large UDP Packets and the DNS (http://www.potaroo.net/ispcol/2017-08/xtn-hdrs.html)", August 2017.	[Huston] Huston, G., "IPv6, Large UDP Packets and the DNS (http://www.potaroo.net/ispcol/2017-08/xtn-hdrs.html)", August 2017.
[I-D.ietf-intarea-tunnels] Touch, J. and M. Townsley, "IP Tunnels in the Internet	[I-D.ietf-intarea-tunnels] Touch, J. and M. Townsley, "IP Tunnels in the Internet
End of changes. 55 change blocks.	
110 lines changed or deleted	***118 lines changed or added***

This html diff was produced by rfcdiff 1.47. The latest version is available from http://tools.ietf.org/tools/rfcdiff/