

6man Working Group  
Internet-Draft  
Updates: RFC [2460](#) (if approved)  
Intended status: Standards Track  
Expires: January 12, 2014

R. Bonica  
Juniper Networks  
W. Kumari  
Google, Inc.  
R. Bush  
Internet Initiative Japan  
H. Pfeifer  
Proton Labs  
July 11, 2013

**IPv6 Fragment Header Deprecation**  
**draft-bonica-6man-frag-deprecate-02**

Abstract

This memo deprecates IPv6 fragmentation and the IPv6 fragment header. It provides reasons for deprecation and updates [RFC 2460](#).

Requirements Language

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 12, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Case For Deprecation . . . . .	<a href="#">3</a>
<a href="#">2.1.</a>	Resource Conservation . . . . .	<a href="#">3</a>
<a href="#">2.2.</a>	Application Reliance on IPv6 Fragmentation . . . . .	<a href="#">3</a>
<a href="#">2.3.</a>	Attack Vectors . . . . .	<a href="#">5</a>
<a href="#">2.4.</a>	Operator Behavior . . . . .	<a href="#">6</a>
<a href="#">3.</a>	Applications That Rely on Fragmentation . . . . .	<a href="#">6</a>
<a href="#">3.1.</a>	DNSSEC . . . . .	<a href="#">7</a>
<a href="#">3.2.</a>	SIIT . . . . .	<a href="#">7</a>
<a href="#">3.3.</a>	OSPFv3 . . . . .	<a href="#">8</a>
<a href="#">3.4.</a>	DCCP and NFS . . . . .	<a href="#">8</a>
<a href="#">3.5.</a>	Tunneling . . . . .	<a href="#">8</a>
<a href="#">4.</a>	Recommendation . . . . .	<a href="#">8</a>
<a href="#">5.</a>	IANA Considerations . . . . .	<a href="#">8</a>
<a href="#">6.</a>	Security Considerations . . . . .	<a href="#">8</a>
<a href="#">7.</a>	Acknowledgements . . . . .	<a href="#">9</a>
<a href="#">8.</a>	References . . . . .	<a href="#">9</a>
<a href="#">8.1.</a>	Normative References . . . . .	<a href="#">9</a>
<a href="#">8.2.</a>	Informative References . . . . .	<a href="#">9</a>
	Authors' Addresses . . . . .	<a href="#">11</a>

## [1.](#) Introduction

Each link on the Internet is characterized by a Maximum Transmission Unit (MTU). A link's MTU represents the maximum packet size that can be conveyed over the link, without fragmentation. IPv6 [[RFC2460](#)] requires that every link in the Internet have an MTU of 1280 octets or greater. On any link that cannot convey a 1280-octet packet in one piece, link-specific fragmentation and reassembly must be provided at a layer below IPv6.

For any given source node, the path to a particular destination is characterized by a path MTU (PMTU). At a given source, the PMTU associated with a destination is equal to the minimum MTU of all of the links in the path between the source and the destination. Because every IPv6-enabled link must support an MTU or 1280 bytes or



greater, the PMTU between any two IPv6 nodes is also 1280 bytes or greater.

[RFC2460] strongly recommends that IPv6 nodes implement Path MTU Discovery (PMTUD) [[RFC1981](#)], in order to discover and take advantage of PMTU values greater than 1280 octets. However, a minimal IPv6 implementation (e.g., in a boot ROM) may simply restrict itself to sending packets no larger than 1280 octets, and omit implementation of PMTUD.

In order to send a packet larger than a path's MTU, a node may use the IPv6 Fragment header to fragment the packet at the source and have it reassembled at the destination(s). However, the use of such fragmentation is discouraged in any application that is able to adjust its packets to fit the measured path MTU (i.e., down to 1280 octets).

In IPv6, a packet can be fragmented only by the host that originates it. This constitutes a departure from the IPv4 [[RFC0791](#)] fragmentation strategy, in which a packet can be fragmented by its originator or by any router that it traverses en route to its destination.

This memo deprecates IPv6 fragmentation and the IPv6 fragment header. It provides reasons for deprecation and updates [[RFC2460](#)].

## **2. Case For Deprecation**

This section presents a case for deprecating the IPv6 Fragment Header.

### **2.1. Resource Conservation**

Packets that are fragmented at their source need to be reassembled at their destination. [[Kent87](#)] points out that the reassembly process is resource intensive. It consumes significant compute and memory resources. While the cited reference refers to IPv4 fragmentation and reassembly, many of its criticisms are equally applicable to IPv6.

By comparison, if a source node were to execute PMTUD procedures, and if applications were to avoid sending datagrams that would result in IP packets that exceed the PMTU, the task of reassembly could be avoided, altogether.

### **2.2. Application Reliance on IPv6 Fragmentation**

Today, a limited number of applications rely upon IPv6 fragmentation.



Most popular TCP implementations include PMTUD or an extension thereof, called Packetization Layer MTU Discovery (PMTUD) [[RFC4821](#)]. Therefore, in the nominal case, applications obtaining transport services from these TCP implementations never cause IPv6 fragments to be sent. However, some TCP implementations that include PMTUD do emit segments long enough to cause IPv6 fragmentation. This happens in the following circumstance:

- o The TCP implementation establishes two (or more) sessions to the same destination
- o Because the TCP implementation has not yet emitted any long segments, the underlying IPv6 implementation estimates the PMTU for destination to be equal to the MTU of the first link in the path to the destination. This estimate is incorrect, and will be revised, below.
- o The first TCP session submits a long segment to the underlying IPv6 implementation
- o The underlying IPv6 implementation determines that if it were to encapsulate this segment in an IPv6 header, the resulting packet would not exceed its current estimate of the PMTU for the destination. So, the underlying IPv6 implementation emits a non-fragmented IPv6 packet. This packet exceeds the actual PMTU for the destination
- o A downstream router discards the long packet and returns an ICMPv6 Packet Too Big (PTB) message.
- o The first TCP session reduces its Maximum Segment Size (MSS) to an appropriate value
- o The underlying IPv6 implementation reduces its estimate of the PMTU for the destination to an appropriate value
- o The second TCP session submits a long segment to the underlying IPv6 implementation. It does so without first querying the underlying IPv6 implementation to learn its estimate of the PMTU for the destination
- o The underlying IPv6 implementation determines that if it were to encapsulate this segment in an IPv6 header, the resulting packet would exceed its current estimate of the PMTU for the destination. So, the underlying IPv6 implementation emits multiple IPv6 fragments.



The authors suggest that the behavior described above may be sub-optimal, and that TCP implementations should leverage PMTU information that the underlying IPv6 implementation could provide.

Many UDP-based [\[RFC0768\]](#) applications follow the recommendations of [\[RFC5405\]](#). According to [\[RFC5405\]](#), "an application SHOULD NOT send UDP datagrams that result in IP packets that exceed the MTU of 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 itself to determine whether the path to a destination will support its desired message size without fragmentation. Applications that do not follow this recommendation to do PMTU 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." The effective MTU for IPv6 is 1280 bytes.

However, several applications are known to rely on IPv6 fragmentation. Some of these are mentioned in [Section 3](#).

### **[2.3. Attack Vectors](#)**

Security researchers have found and continue to find attack vectors that rely on IP fragmentation. For example, [\[I-D.ietf-6man-oversized-header-chain\]](#) and [\[I-D.ietf-6man-nd-extension-headers\]](#) describe variants of the tiny fragment attack [\[RFC1858\]](#). In this attack, a packet is crafted so that it can evade stateless firewall filters. The stateless firewall filter matches on fields drawn from the IPv6 header and an upper layer header. However, the packet is fragmented so that the upper layer header, or a significant part of that header, does not appear in the first fragment. Because a stateless firewall cannot parse payload beyond the first fragment, the packet evades detection by the firewall.

Security researcher have also studied reassembly algorithms on popular computing platforms, with the following goals:

- o to discover fragility in seldom exercised parts of the IP stack
- o to engineer flows that maximize resources consumed by the reassembly process

The Dawn and Rose Attacks [\[Hollis\]](#) are the products of such research.

All of the attack vectors mentioned above can be mitigated with firewalls and increasingly sophisticated reassembly algorithms.





However, the continued investment required to mitigate newly discovered vulnerabilities detracts from the cost effectiveness of IPv6 as a networking solution.

#### **2.4. Operator Behavior**

For reasons described above, and also articulated in [\[I-D.taylor-v6ops-fragdrop\]](#), many network operators filter all IPv6 fragments. Also, the default behavior of many currently deployed firewalls is to discard IPv6 fragments.

In one recent study [\[DeBoer\]](#), two researchers utilized a measurement network to measure fragment filtering. They sent packets, fragmented to the minimum MTU of 1280, to 502 IPv6 enabled and reachable probes. They found that during any given trial period, ten percent of the probes did not receive fragmented packets.

### **3. Applications That Rely on Fragmentation**

The following is a list of applications that are currently known to rely on IPv6 fragmentation:

- o DNSSEC [\[RFC4035\]](#).
- o SIIT [\[RFC6145\]](#)
- o OSPFv3 [\[RFC5340\]](#)
- o NFSv4 [\[RFC3530\]](#)
- o DCCP [\[RFC4340\]](#)

Some tunneling configurations also rely upon IPv6 fragmentation. See [Section 3.5](#) for details.

Each of these applications relies on 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 protocol implementations already take appropriate steps to avoid fragmentation.

Each of these applications will continue to emit IPv6 fragments, even after the IPv6 fragmentation header is deprecated. In order to achieve backwards compatibility, new IPv6 implementations will continue to support reassembly of incoming fragments. See for [Section 4](#) details.



### [3.1.](#) DNSSEC

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

When running over UDP, DNSSEC is likely to cause the generation of IPv6 fragments. By comparison, when running over TCP, DNSSEC is much less likely to cause the generation of IPv6 fragments.

When running over UDP, DNSSEC's reliance upon IPv6 fragmentation is fundamental. That reliance cannot be broken without changing the DNSSEC specification.

DNSSEC is an essential part of the Internet architecture. Therefore, this issue is for further study and must be resolved before IPv6 fragmentation can be deprecated.

### [3.2.](#) SIIT

[RFC6145] requires the following:

- o "When the IPv4 sender does not set the DF bit, the translator SHOULD always include an IPv6 Fragment Header to indicate that the sender allows fragmentation. The translator MAY provide a configuration function that allows the translator not to include the Fragment Header for the non-fragmented IPv6 packets".
- o "If the DF flag is not set and the IPv4 packet will result in an IPv6 packet larger than 1280 bytes, the packet SHOULD be fragmented so the resulting IPv6 packet (with Fragment Header added to each fragment) will be less than or equal to 1280 bytes."

These behaviors cannot be changed, and for these reasons, SIIT devices will continue to emit IPv6 fragments, even after IPv6 fragmentation has been deprecated.

SIIT also emits ICMPv6 PTB messages with MTU less than 1280. In that case, the originating IPv6 node is not required to reduce the size of subsequent packets to less than 1280, but must include a Fragment header in those packets so that SIIT can obtain a suitable Identification value to use in resulting IPv4 fragments. Note that this means the payload may have to be reduced to 1232 octets (1280 minus 40 for the IPv6 header and 8 for the Fragment header), and smaller still if additional extension headers are used.

This problem could be avoided if SIIT executed an alternative procedure. For example, rather than discarding the packet and



sending an ICMPv6 PTB message with MTU less than 1280, SIIT could generate a random number for use as the Identification value and forward the packet. This issue clearly requires further study.

### **3.3. OSPFv3**

OSPFv3 implementations may emit messages large enough to cause IPv6 fragmentation. However, in keeping with the recommendations of [\[RFC2460\]](#), and in order to optimize performance, most OSPFv3 implementation refrain from doing so. Many implementations simply restrict their maximum message size to some value that is safely below 1280.

### **3.4. DCCP and NFS**

Details TBD

### **3.5. Tunneling**

TBD

## **4. Recommendation**

This memo deprecates IPv6 fragmentation and the IPv6 fragment header. Application and transport layer protocols SHOULD support effective PLMTUD [\[RFC4821\]](#), since ICMP-based PMTUD [\[RFC1981\]](#) is unreliable. Any application or transport layer protocol that cannot support effective PMTUD MUST NOT in any circumstances send IPv6 packets that exceed the IPv6 minimum MTU of 1280 bytes.

IPv6 stacks and forwarding nodes MUST continue to support inbound fragmented IPv6 packets as specified in [\[RFC2460\]](#). However, this requirement exceeds the capability of some types of forwarding nodes such as firewalls and load balancers. Therefore implementers and operators need to be aware that on many paths through the Internet, IPv6 fragmentation will fail. Legacy applications and transport layer protocols that do not conform to the previous paragraph can expect connectivity failures as a result.

## **5. IANA Considerations**

IANA is requested to mark the Fragment Header for IPv6 (44) as deprecated in the Protocol Numbers registry.

## **6. Security Considerations**

Deprecation of the IPv6 Fragment Header will improve network security by eliminating attacks that rely on fragmentation.



## **7. Acknowledgements**

The author wishes to acknowledge Tore Anderson, Mark Andrews, Brian Carpenter, Havard Eidnes, Bob Hinden, Geoff Huston, George Michaelson, Simon Perreault, Arturo Servin, Mark Smith, Fred Templin, Willem Toorop, Glen Turner and Ole Troan for their review and constructive comments.

## **8. References**

### **8.1. Normative References**

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, [RFC 768](#), August 1980.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, [RFC 791](#), September 1981.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, [RFC 793](#), September 1981.
- [RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", [RFC 1981](#), August 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), December 1998.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", [RFC 4443](#), March 2006.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", [RFC 4821](#), March 2007.
- [RFC5405] Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", [BCP 145](#), [RFC 5405](#), November 2008.

### **8.2. Informative References**

- [DeBoer] De Boer, M. and J. Bosma, "Discovering Path MTU black holes on the Internet using RIPE Atlas", July 2012, <<http://www.nlnetlabs.nl/downloads/publications/pmtu-black-holes-msc-thesis.pdf>>.





- [Hollis] Hollis, K., "The Rose Attack Explained", , <[http://digital.net/~gandalf/Rose\\_Frag\\_Attack\\_Explained.htm](http://digital.net/~gandalf/Rose_Frag_Attack_Explained.htm)>.
- [I-D.ietf-6man-nd-extension-headers]  
Gont, F., "Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery", [draft-ietf-6man-nd-extension-headers-05](#) (work in progress), June 2013.
- [I-D.ietf-6man-oversized-header-chain]  
Gont, F. and V. Manral, "Security and Interoperability Implications of Oversized IPv6 Header Chains", [draft-ietf-6man-oversized-header-chain-02](#) (work in progress), November 2012.
- [I-D.ietf-6man-predictable-fragment-id]  
Gont, F., "Security Implications of Predictable Fragment Identification Values", [draft-ietf-6man-predictable-fragment-id-00](#) (work in progress), March 2013.
- [I-D.taylor-v6ops-fragdrop]  
Jaeggli, J., Colitti, L., Kumari, W., Vyncke, E., Kaeo, M., and T. Taylor, "Why Operators Filter Fragments and What It Implies", [draft-taylor-v6ops-fragdrop-01](#) (work in progress), June 2013.
- [Kent87] Kent, C. and J. Mogul, "Fragmentation Considered Harmful", In Proc. SIGCOMM '87 Workshop on Frontiers in Computer Communications Technology , August 1987.
- [RFC1858] Ziemba, G., Reed, D., and P. Traina, "Security Considerations for IP Fragment Filtering", [RFC 1858](#), October 1995.
- [RFC3530] Shepler, S., Callaghan, B., Robinson, D., Thurlow, R., Beame, C., Eisler, M., and D. Noveck, "Network File System (NFS) version 4 Protocol", [RFC 3530](#), April 2003.
- [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", [RFC 4035](#), March 2005.
- [RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", [RFC 4340](#), March 2006.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", [RFC 5340](#), July 2008.



[RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", [RFC 6145](#), April 2011.

#### Authors' Addresses

Ron Bonica  
Juniper Networks  
2251 Corporate Park Drive  
Herndon, Virginia 20170  
USA

Email: [rbonica@juniper.net](mailto:rbonica@juniper.net)

Warren Kumari  
Google, Inc.  
1600 Amphitheatre Parkway  
Mountainview, California 94043  
USA

Email: [warren@kumari.net](mailto:warren@kumari.net)

Randy Bush  
Internet Initiative Japan  
5147 Crystal Springs  
Bainbridge Island Washington  
USA

Email: [randy@psg.com](mailto:randy@psg.com)

Hagen Paul Pfeifer  
Protocollabs  
Munich 81379  
Germany

Email: [hagen.pfeifer@protocollabs.com](mailto:hagen.pfeifer@protocollabs.com)  
URI: <http://www.protocollabs.com>

