Workgroup: Network Working Group
Internet-Draft: draft-templin-6man-fragrep-07
Updates: RFC8200, RFC8201, RFC4443, RFC1191
(if approved)
Published: 29 March 2022
Intended Status: Standards Track
Expires: 30 September 2022
Authors: F. L. Templin, Ed.
Boeing Research & Technology

IPv6 Fragment Retransmission and Path MTU Discovery Soft Errors

Abstract

Internet Protocol version 6 (IPv6) provides a fragmentation and reassembly service for end systems allowing for the transmission of packets that exceed the path MTU. However, loss of individual fragments requires retransmission of original packets in their entirety leading to cascading reassembly failures. This document specifies an IPv6 fragment retransmission scheme that matches the loss unit to the retransmission unit. The document further specifies an update to Path MTU Discovery that distinguishes hard link size restrictions from reassembly congestion events.

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 https://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 30 September 2022.

Copyright Notice

Table of Contents

1. Introduction

Internet Protocol version 6 (IPv6) [RFC8200] provides a fragmentation and reassembly service similar to that found in IPv4 [RFC0791], with the exception that only the source host (i.e., and not routers on the path) may perform fragmentation. When an IPv6 packet is fragmented, the loss unit (i.e., a single IPv6 fragment) becomes smaller than the retransmission unit (i.e., the entire packet) which even under moderate loss conditions could result in cascading reassembly failures that degrade forward progress [RFC8900].

The presumed drawbacks of fragmentation are tempered by the fact that performance increases can often be realized when the source sends packets larger than the path MTU. This is due to the fact that larger packets result in fewer application system calls, plus transmission of a single large packet results in a burst of multiple IPv6 fragments separated by minimal inter-packet delays. These bursts yield high network utilization for the burst duration, while modern reassembly implementations have proven capable of accommodating the bursts. If the loss unit can somehow be made to match the retransmission unit, the performance benefits of IPv6 fragmentation can be realized.

This document therefore proposes an IPv6 fragment retransmission service where the source marks fragments as retransmission-eligible

while the destination may request retransmission of lost fragments. The service provides opportunistic best-effort retransmissions over an imaginary "link" extending from the source to the destination consistent with the Automatic Repeat Request (ARQ) function of common data links [RFC3366]. The service does not attempt to replace true end-to-end reliability, but instead often allows the destination to recover missing individual fragments of partial reassemblies before true end-to-end timers would cause retransmission of the entire packet.

The original packet source may be either co-located with or many IP network hops before the IPv6 fragmentation source. In the same fashion, the IPv6 reassembly destination may be either co-located with or many IP network hops before the final destination. When conditions suggest that an original source should begin sending smaller packets, the fragmentation source and/or reassembly destination can return a new type of ICMPv6/ICMPv4 Packet Too Big (PTB) message termed a PTB "soft error".

PTB "soft errors" are distinguished from classic "hard errors" by a non-zero PTB Code (ICMPv6) or unused (ICMPv4) field value. The fragmentation source can return rate-limited soft errors to recommend smaller packet sizes to the original source while fragmentation of large packets is producing excessive numbers of fragments. Similarly, the reassembly destination can return rate-limited soft errors (i.e., via the fragmentation source to the original source) while reassembly of large packets is causing excessive reassembly congestion. Original sources that receive these soft errors should reduce their packet sizes until the errors subside, but can begin to increase packet sizes again without delay until further soft or hard errors arrive.

The following sections discuss common use cases and operational considerations for applying IPv6 fragment retransmission and path MTU discovery soft errors. They further specify new codings for the IPv6 fragment header Reserved field, a new ICMPv6 message type and updates to ICMPv6/ICMPv4 PTB messages. This document therefore updates existing standards where necessary.

2. Terminology

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.

3. Common Use Cases

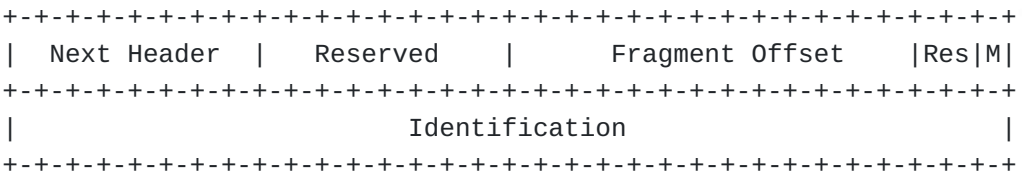
A common use case of interest is to improve the state of affairs for IPv6 encapsulation (i.e., "tunneling") [RFC2473] when the original source may be many IP hops away from the tunnel ingress, and the tunnel packet may be fragmented following encapsulation. The tunnel is seen as a "link" on the path from the original source to the final destination, and the goal is to increase link reliability in order to minimize wasteful end-to-end retransmissions.

When the original source and IPv6 fragmentation source are co-located on the same platform (physical or virtual) the window of opportunity for successful retransmission of individual fragments may be narrow unless the link persistence timeframe is carefully coordinated with upper layer retransmission timers. (In an uncoordinated case, upper layers may retransmit the entire packet before or at roughly the same time the IPv6 fragmentation source retransmits individual fragments, leading to increased congestion and wasted retransmissions.) However, the same retransmission facility can be applied to both the tunneled and end system source models.

Upper layer protocols of the original source can further assign a "Parcel ID" to groups of packets eligible for delivery to final destination applications as a larger aggregate instead of smaller individual packets (see: [I-D.templin-intarea-parcels]). The upper layer protocols supply the Parcel ID to lower layers which insert the value as discussed in Section 4, while the destination lower layer protocols deliver the Parcel ID to upper layers. Further details on parcel grouping are out of scope for this document.

4. IPv6 Fragmentation

IPv6 fragmentation is specified in Section 4.5 of [RFC8200] and is based on the IPv6 Fragment extension header formatted as shown below:



In this format:

* Next Header is a 1-octet IP protocol version of the next header following the Fragment Header.

* Reserved is a 1-octet reserved field set to 0 on transmission and ignored on reception.

*Fragment Offset is a 13-bit field that provides the offset (in 8-octet units) of the data portion that follows from the beginning of the packet.

*Res is a 2-bit field set to 0 on transmission and ignored on reception.

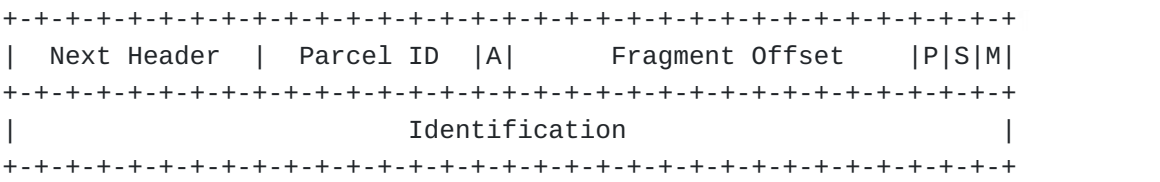
*M is the "More Fragments" bit telling whether additional fragments follow.

*Identification is a 32 bit numerical identification value for the entire IPv6 packet. The value is copied into each fragment of the same IPv6 packet.

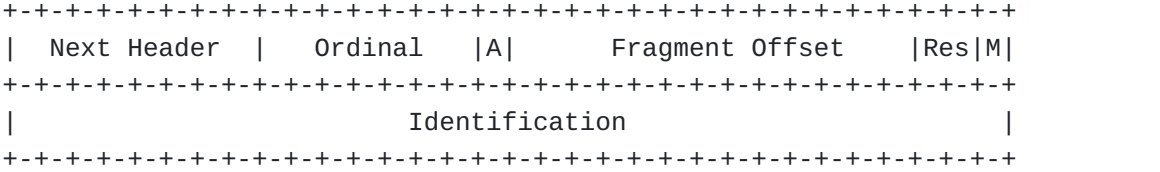
The fragmentation and reassembly specification in [RFC8200] can be considered as the standard method which adheres to the details of that RFC. This document presents an enhanced method that allows for retransmissions of individual fragments.

5. IPv6 Fragment Retransmission

Fragmentation implementations that follow this specification reuse the (formerly) Reserved field of the IPv6 Fragment Header. For first fragments (i.e., those with zero Fragment Offset) the 8-bit Reserved field is replaced with a 7-bit Parcel ID followed by a 1-bit A(RQ) flag, and the 2-bit Res field is replaced with a 1-bit P(arcel) flag followed by a 1-bit S(ub-parcels) flag as shown below:



For non-first fragments (i.e., those with non-zero Fragment Offset), the Reserved field is replaced with a 7-bit "Ordinal" field followed by a 1-bit A(RQ) flag as shown below:

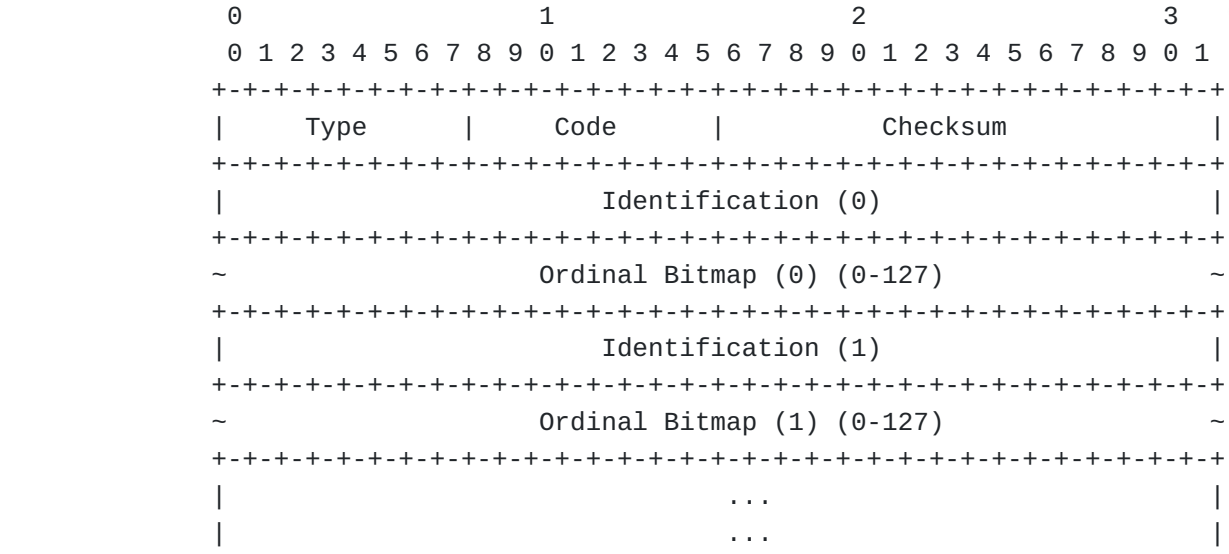


When a source that follows this specification fragments an IPv6 packet it sets the first fragment A flag to 1, then for IP parcels sets Parcel ID, P and S according to the processing and transmission procedures found in [I-D.templin-intarea-parcels] and [I-D.templin-6man-omni]. For non-parcels, the source instead sets Parcel ID, P and S to 0.

The source then sets the Ordinal value for each successive non-first fragment to a monotonically-increasing value beginning with 1, i.e., it sets Ordinal to '1' for the first non-first fragment, '2' for the second non-first fragment, '3' for the third non-first fragment, etc. up to either Ordinal '127' or the final fragment (whichever comes first) while also setting the A flag to 1. (If there are additional non-first fragments beyond Ordinal '127', the source instead sets their Ordinals to '0' to indicate that the fragment is not eligible for retransmission.)

When a destination that follows this specification receives IPv6 fragments with the A flag set, it infers that the source participates in the protocol and maintains a checklist of all Ordinal fragments received for a specific Identification number. (Note that receipt of any IPv6 fragments with the A flag set provides an implicit assertion that any lost Ordinals of the same packet are also eligible for retransmission.)

If the destination notices one or more Ordinals missing after most other Ordinals for the same Identification have arrived, it can prepare an ICMPv6 Fragmentation Report (FRAGREP) message [RFC4443] to send back to the source. The message is formatted as follows:



In this format, the destination prepares the FRAGREP message as a list of 20-octet (Identification(i), Bitmap(i)) pairs. The first 4 octets in each pair encode the Identification value for the IPv6 packet that is subject of the report, while the remaining 16 octets encode a 128-bit Bitmap of Ordinal fragments received for this Identification. For example, if the destination receives the first fragment (i.e., Ordinal number 0) plus non-first fragment Ordinals 1, 3, 4, 6, and 8 it sets Bitmap bits 0, 1, 3, 4, 6 and 8 to '1' and sets all other bits to '0'. The destination may include as many (Identification, Bitmap) pairs as necessary without causing the

entire message to exceed the minimum IPv6 MTU (i.e., 1280 octets); if additional pairs are necessary, the destination may prepare and send multiple messages.

The destination next transmits the FRAGREP message to the IPv6 fragment source. When the source receives the message, it examines each entry to determine the per-Identification Ordinal fragments that require retransmission. For example, if the source receives a Bitmap for Identification 0x12345678 with bits 0, 1, 3, 4, 6 and 8 set to '1', it would retransmit Ordinal fragments (0x12345678, 2), (0x12345678, 5) and (0x12345678, 7).

This implies that the source should retain a cache of recently transmitted fragments for a time that determines "link persistence" [RFC3366]. The link persistence should be at least as long as the round-trip time from the fragmentation source to the reassembly destination, plus an additional small delay to allow for processing overhead and/or delay variance. Then, if the source receives a FRAGREP message requesting retransmission of one or more Ordinals, it can retransmit any still in its cache. Otherwise, the Ordinal will incur a cache miss and the original source will eventually retransmit the original packet in its entirety. After processing all entries in the FRAGREP, the source discards the message.

The maximum-sized IPv6 packet that a source can submit for fragmentation is 65535 octets, and the minimum IPv6 path MTU is 1280 octets. Assuming the minimum IPv6 path MTU as the nominal size for non-final fragments, the number of Ordinals for each IPv6 packet should therefore easily fit within the available Bitmap bits when the fragments are transmitted over IPv6-only network paths. However, when the path may traverse one or more IPv4 networks (e.g., via tunneling) the path MTU may be significantly smaller. In that case, the number of IPv6 fragments needed may exceed the maximum number of Ordinal retransmission candidates.

When the number of IPv6 fragments exceeds 128, the source assigns an Ordinal value in the first 127 non-first fragments, but sets Ordinal to 0 in any remaining non-first fragments then transmits all fragments. When the destination receives the fragments, it may return a FRAGREP to request retransmission of the first fragment and/or any missing Ordinal non-first fragments, but may not request retransmission of non-first fragments with zero Ordinals for which the default behavior of best-effort delivery applies. However, all fragments are presented equally to the reassembly cache regardless of the (formerly) Reserved field settings, where the Reserved values are ignored and successful reassembly is likely.

Finally, transmission of IPv6 fragments over IPv6-only paths can be safely conducted without a fragmentation-layer integrity check since

IPv6 includes reassembly safeguards and a 32-bit Identification value. Conversely, transmission of IPv6 fragments over IPv4-only or mixed IPv6/IPv4 paths requires a fragmentation-layer integrity check inserted by the source before fragmentation and verified by the destination following reassembly since IPv4 provides only a 16-bit Identification and no reassembly safeguards. (In cases where the full path cannot be determined a priori, an integrity check should always be included as specified in AERO [I-D.templin-6man-aero] and OMNI [I-D.templin-6man-omni].)

6. Packet Too Big (PTB) Soft Errors

When an IPv6 fragmentation source forwards packets that produce what it considers as excessive numbers fragments (e.g., 32, 48, 64, more), the fragmentation source can also return PTB "soft errors" to the original source (subject to rate limiting). Either the fragmentation source or reassembly destination may also return PTB soft errors if the frequency of retransmissions or reassembly failures exceeds acceptable thresholds.

PTB soft errors are distinguished from ordinary "hard errors" through non-zero values in the ICMPv6 "Code" [RFC8201][RFC4443] or ICMPv4 "unused" [RFC1191] fields. The following values are currently defined:

* 0 - "PTB hard error" - Original sources that receive these messages obey the classic Path MTU Discovery (PMTUD) specifications found in [RFC8201][RFC1191].

* 1 - "PTB soft error (packet lost)" - Original sources that receive these messages should reduce their packet sizes while retransmitting the lost packet data, but need not wait the prescribed 10 minutes before attempting to again increase packet sizes.

* 2 - "PTB soft error (packet forwarded)" - Original sources that receive these messages should reduce their packet sizes without invoking retransmission, and also need not wait the prescribed 10 minutes before attempting to again increase packet sizes.

* 3-255 - reserved for future use.

PTB soft errors include as much of the invoking packet as possible without the message exceeding the minimum MTU (i.e., 1280 octets for IPv6 or 576 octets for IPv4). Original sources that recognize PTB soft errors should follow common logic to dynamically tune their packet sizes to obtain the best performance. In particular, an original source can gradually increase its packet sizes while PTB soft errors are suppressed then again reduce packet sizes when excessive soft errors arrive.

Original sources that do not recognize PTB soft errors (i.e., that do not examine the Code/unused field value) follow the same standards as for hard errors as described above and may therefore miss performance improvement opportunities.

7. Implementation Status

TBD.

8. IANA Considerations

A new ICMPv6 Message Type code for "Fragmentation Report (FRAGREP)" is requested. The registration procedure is "IETF Review" and the reference is this document [RFCXXXX].

The IANA is instructed to create new registries for "ICMPv6 Packet Too Big Code field" and "ICMPv4 Fragmentation Needed unused field" values. Both registries should have the following initial values:

Value	Sub-Type name	Reference
0	PTB hard error	[RFCXXXX]
1	PTB soft error (loss)	[RFCXXXX]
2	PTB soft error (no loss)	[RFCXXXX]
3-252	Unassigned	
253-254	Reserved for Experimentation	[RFCXXXX]
255	Reserved by IANA	[RFCXXXX]

Figure 1: Packet Too Big Code/unused Values

9. Security Considerations

Communications networking security is necessary to preserve confidentiality, integrity and availability.

10. Acknowledgements

This work was inspired by ongoing AERO/OMNI/DTN investigations along with recent innovations with IP Parcels.

.

11. References

11.1. Normative References

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

[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>.

[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

[I-D.templin-6man-aero] Templin, F. L., "Automatic Extended Route Optimization (AERO)", Work in Progress, Internet-Draft, draft-templin-6man-aero-40, 7 March 2022, <https://www.ietf.org/archive/id/draft-templin-6man-aero-40.txt>.

[I-D.templin-6man-omni] Templin, F. L., "Transmission of IP Packets over Overlay Multilink Network (OMNI) Interfaces", Work in Progress, Internet-Draft, draft-templin-6man-omni-55, 7 March 2022, <https://www.ietf.org/archive/id/draft-templin-6man-omni-55.txt>.

[I-D.templin-intarea-parcels] Templin, F. L., "IP Parcels", Work in Progress, Internet-Draft, draft-templin-intarea-parcels-09, 10 February 2022, <https://www.ietf.org/archive/id/draft-templin-intarea-parcels-09.txt>.

[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>.

[RFC3366] Fairhurst, G. and L. Wood, "Advice to link designers on link Automatic Repeat reQuest (ARQ)", BCP 62, RFC 3366, DOI 10.17487/RFC3366, August 2002, <https://www.rfc-editor.org/info/rfc3366>.

[RFC8900] Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", BCP 230, RFC 8900, DOI 10.17487/RFC8900, September 2020, <https://www.rfc-editor.org/info/rfc8900>.

Author's Address

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
United States of America

Email: fltemplin@acm.org