IPv6 RFC8200 Fragmentation Errata

Bob Hinden

IETF105 Montreal

Background

- RFC8200 Internet Protocol, Version 6 (IPv6) Specification published as “Internet Standard” July 2017
  - Included fragmentation updates from RFC5722, RFC6946, RFC7112, and RFC8021
  - Required extensive changes to fragmentation text in Section 4.5
- Errata 5170, 5171, 5172, 5173 filed 2017-10-29

Errata 5170, 5171, 5172

- Problem is that Fragment Offset defined as pointing to “Fragmentable Part”, should have pointed to “Extension & Upper-Layer Headers”
- Text needs to change in four places
  - Errata missed one place
- Errata proposed other text changes beyond the problem.

Errata 5173

- Proposed change in figure of reassembled original packet.

```
+--------------+-----------------+---------+--------+-//--+--------+
| Per-Fragment | Ext & Upper-Layer| first   | second |     | last   |
| Headers      | Headers          |frag data|fragment|.....|fragment|
+--------------+-----------------+---------+--------+-//--+--------+
```

to:

```
+--------------+-----------------+---------+--------+-//--+--------+
| Per-Fragment | Ext & Upper-Layer| first   | second |     | last   |
| Headers      | Headers          |fragment |fragment|.....|fragment|
+--------------+-----------------+---------+--------+-//--+--------+
```

- Change is not correct, it is only the fragment data, unlike the following fragments.

Proposed Text Changes (1)

- Page 16, 1st paragraph

- Fragment Offset

  13-bit unsigned integer. The offset, in 8-octet units, of the data following this header, relative to the start of the ~~Fragmentable Part~~ **Extension & Upper-Layer Headers** of the original packet.

Proposed Text Changes (2)

- Page 18, 2nd paragraph from bottom

- A Fragment Offset containing the offset of the fragment, in 8-octet units, relative to the start of the ~~Fragmentable Part~~ **Extension & Upper-Layer Headers** of the original packet. The Fragment Offset of the first ("leftmost") fragment is 0.

Proposed Text Changes (3)

- Page 19, 4th paragraph from bottom

- A Fragment Offset containing the offset of the in 8-octet units, relative to the start of the ~~Fragmentable Part~~ **Extension & Upper-Layer Headers** of the original packet.

Proposed Text Changes (4)

- Page 20, last paragraph

- The **Extension & Upper-Layer Headers and** Fragmentable Part of the reassembled packet is constructed from the fragments following the Fragment headers in each of the fragment packets. The length of each fragment is computed by subtracting from the packet's Payload Length the length of the headers between the IPv6 header and fragment itself; its relative position in ~~Fragmentable Part~~ **the reassembled original packet** is computed from its Fragment Offset value.

Other Related Changes

- Page 15, 2nd paragraph

- Next Header Definition

  8-bit selector. Identifies the initial header type of the ~~Fragmentable Part~~ **Next Header value that identifies the first header after the Per-Fragment headers of the original packet** (defined below). Uses the same values as the IPv4 Protocol field [IANA-PN].

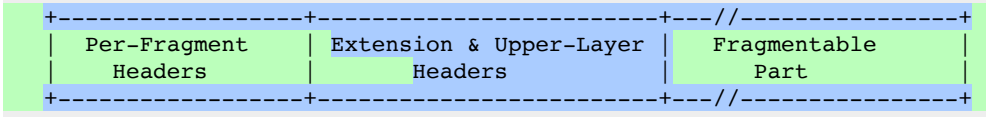
Other Approach

- Ole reviewed the changes and proposed an alternative approach
- Problem was caused by adding **Extension & Upper-Layer Headers** to figures and invalidating Fragment Offset / Fragmentable Part text.
- Fix is to limit text about **Extension & Upper-Layer Headers** to text that describes creating the first fragment. Keep original figures.

packet, including transit time from source to destination and time spent awaiting reassembly with other fragments of the same packet. However, it is not required that a source node knows the maximum packet lifetime. Rather, it is assumed that the requirement can be met by implementing an algorithm that results in a low identification reuse frequency. Examples of algorithms that can meet this requirement are described in [RFC7739].

The initial, large, unfragmented packet is referred to as the "original packet", and it is considered to consist of three parts, as illustrated:

original packet:



The Per-Fragment headers must consist of the IPv6 header plus any extension headers that must be processed by nodes en route to the destination, that is, all headers up to and including the Routing header if present, else the Hop-by-Hop Options header if present, else no extension headers.

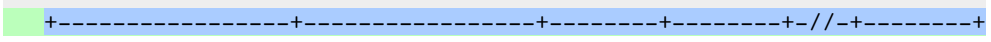
The Extension headers are all other extension headers that are not included in the Per-Fragment headers part of the packet. For this purpose, the Encapsulating Security Payload (ESP) is not considered an extension header. The Upper-Layer header is the first upper-layer header that is not an IPv6 extension header. Examples of upper-layer headers include TCP, UDP, IPv4, IPv6, ICMPv6, and as noted ESP.

The Fragmentable Part consists of the rest of the packet after the upper-layer header or after any header (i.e., initial IPv6 header or extension header) that contains a Next Header value of No Next Header.

The Fragmentable Part of the original packet is divided into fragments. The lengths of the fragments must be chosen such that the resulting fragment packets fit within the MTU of the path to the packet's destination(s). Each complete fragment, except possibly the last ("rightmost") one, is an integer multiple of 8 octets long.

The fragments are transmitted in separate "fragment packets" as illustrated:

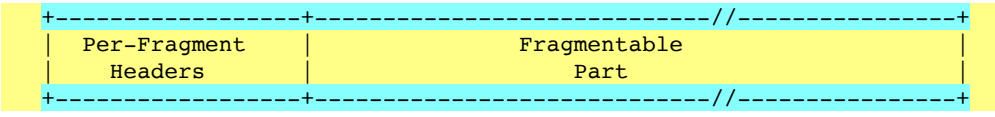
original packet:



packet, including transit time from source to destination and time spent awaiting reassembly with other fragments of the same packet. However, it is not required that a source node knows the maximum packet lifetime. Rather, it is assumed that the requirement can be met by implementing an algorithm that results in a low identification reuse frequency. Examples of algorithms that can meet this requirement are described in [RFC7739].

The initial, large, unfragmented packet is referred to as the "original packet", and it is considered to consist of two parts, as illustrated:

original packet:



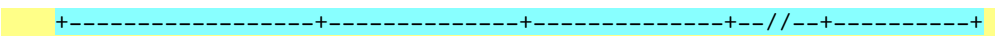
The Per-Fragment headers must consist of the IPv6 header plus any extension headers that must be processed by nodes en route to the destination, that is, all headers up to and including the Routing header if present, else the Hop-by-Hop Options header if present, else no extension headers.

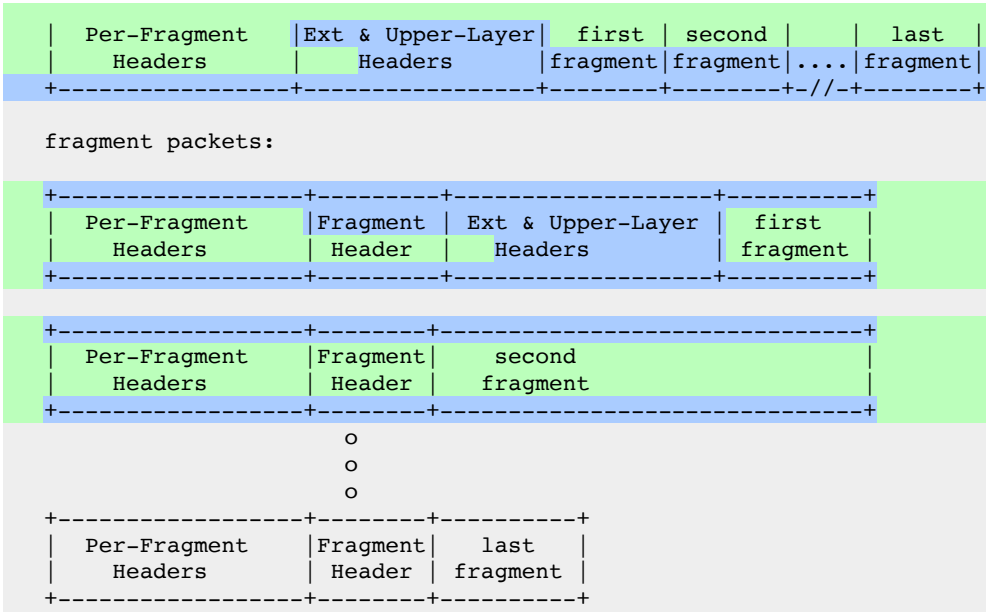
The Fragmentable Part consists of the rest of the packet, that is, any extension headers that need be processed only by the final destination node(s), plus the upper-layer header and data.

The Fragmentable Part of the original packet is divided into fragments. The lengths of the fragments must be chosen such that the resulting fragment packets fit within the MTU of the path to the packet's destination(s). Each complete fragment, except possibly the last ("rightmost") one, is an integer multiple of 8 octets long.

The fragments are transmitted in separate "fragment packets" as illustrated:

original packet:





The first fragment packet is composed of:

(1) The Per-Fragment headers of the original packet, with the Payload Length of the original IPv6 header changed to contain the length of this fragment packet only (excluding the length of the IPv6 header itself), and the Next Header field of the last header of the Per-Fragment headers changed to 44.

(2) A Fragment header containing:

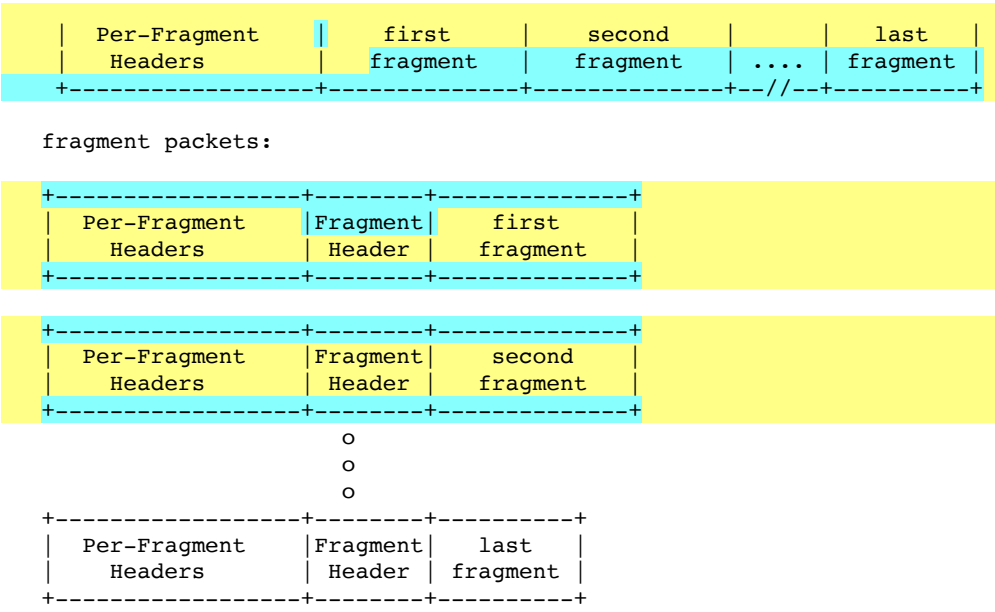
skipping to change at *page 5, line 32*

An M flag value of 1 as this is the first fragment.

The Identification value generated for the original packet.

(3) Extension headers, if any, and the Upper-Layer header. These headers must be in the first fragment. Note: This restricts the size of the headers through the Upper-Layer header to the MTU of the path to the packet's destinations(s).

(4) The first fragment.



The first fragment packet is composed of:

(1) The Per-Fragment headers of the original packet, with the Payload Length of the original IPv6 header changed to contain the length of this fragment packet only (excluding the length of the IPv6 header itself), and the Next Header field of the last header of the Per-Fragment headers changed to 44.

(2) A Fragment header containing:

skipping to change at *page 5, line 25*

An M flag value of 1 as this is the first fragment.

The Identification value generated for the original packet.

(3) Extension headers, if any, and the Upper-Layer header. These headers must be in the first fragment. Note: This restricts the size of the headers through the Upper-Layer header to the MTU of the path to the packet's destinations(s).

Extension headers are all other extension headers that are not included in the Per-Fragment headers part of the packet. For this purpose, the Encapsulating Security Payload (ESP) is not considered an extension header. The Upper-Layer header is the first upper-layer header that is not an IPv6 extension header. Examples of upper-layer headers include TCP, UDP, IPv4, IPv6, ICMPv6, and as noted ESP.

(4) The first fragment.

The subsequent fragment packets are composed of:

(1) The Per-Fragment headers of the original packet, with the Payload Length of the original IPv6 header changed to contain the length of this fragment packet only (excluding the length of the IPv6 header itself), and the Next Header field of the last header of the Per-Fragment headers changed to 44.

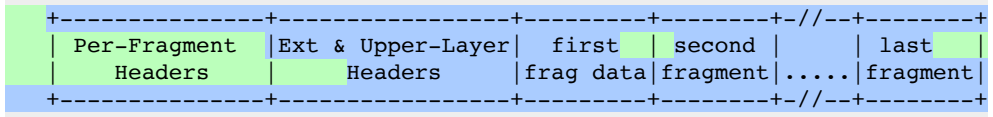
skipping to change at *page 6, line 21*

(3) The fragment itself.

Fragments must not be created that overlap with any other fragments created from the original packet.

At the destination, fragment packets are reassembled into their original, unfragmented form, as illustrated:

reassembled original packet:



The following rules govern reassembly:

An original packet is reassembled only from fragment packets that have the same Source Address, Destination Address, and Fragment Identification.

The Per-Fragment headers of the reassembled packet consists of all headers up to, but not including, the Fragment header of the first fragment packet (that is, the packet whose Fragment Offset is

skipping to change at *page 10, line 5*

This section describes change history made in each Internet Draft that went into producing this version. The numbers identify the Internet-Draft version in which the change was made.

Individual Internet Drafts

00) The purpose of this version is to establish a baseline from RFC8200 Section 4.5. It is based on the XML received from the RFC Editor.

Author's Address

The subsequent fragment packets are composed of:

(1) The Per-Fragment headers of the original packet, with the Payload Length of the original IPv6 header changed to contain the length of this fragment packet only (excluding the length of the IPv6 header itself), and the Next Header field of the last header of the Per-Fragment headers changed to 44.

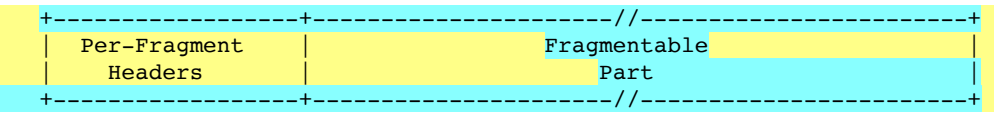
skipping to change at *page 6, line 21*

(3) The fragment itself.

Fragments must not be created that overlap with any other fragments created from the original packet.

At the destination, fragment packets are reassembled into their original, unfragmented form, as illustrated:

reassembled original packet:



The following rules govern reassembly:

An original packet is reassembled only from fragment packets that have the same Source Address, Destination Address, and Fragment Identification.

The Per-Fragment headers of the reassembled packet consists of all headers up to, but not including, the Fragment header of the first fragment packet (that is, the packet whose Fragment Offset is

skipping to change at *page 9, line 48*

This section describes change history made in each Internet Draft that went into producing this version. The numbers identify the Internet-Draft version in which the change was made.

Individual Internet Drafts

00) The purpose of this version is to establish a baseline from RFC8200 Section 4.5. It is based on the XML received from the RFC Editor.

01) This version includes proposed new version of the IPv6 Fragmentation Header text to resolve the issues in reported errata.

Author's Address

Next Steps

- Chairs think that alternative approach is better

- Working group to review proposed changes
  - Good idea to do implementation based on proposed text. There might be other issues.

- Errata 5173
  - Reject, it is not correct.

Errata 5170, 5171, 5172

- Choices for handling
  1. Accept Errata with changed text described here, with "**Held for Document Update**" status.
  2. Reject these errata and create new errata with agreed new text
     - New Errata tools under development
  3. Publish new RFC updating Section 4.5 of RFC8200
  4. Publish RFC8200bis with these changes.

My Recommendations

- Either:
  - Show changes in Errata, has the advantage of keeping the changes in the same RFC, or
  - Proceed with new RFC that updates RFC8200
- *[Or, deprecate IPv6 Fragmentation……???]*
- Do not do RFC8200bis
  - That would open up everything for changes
  - Sends the wrong message to larger community that IPv6 isn't stable.

QUESTIONS / COMMENTS?