IPv6 RFC8200 Fragmentation Errata

Bob Hinden

IETF106 Singapore

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
- 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.

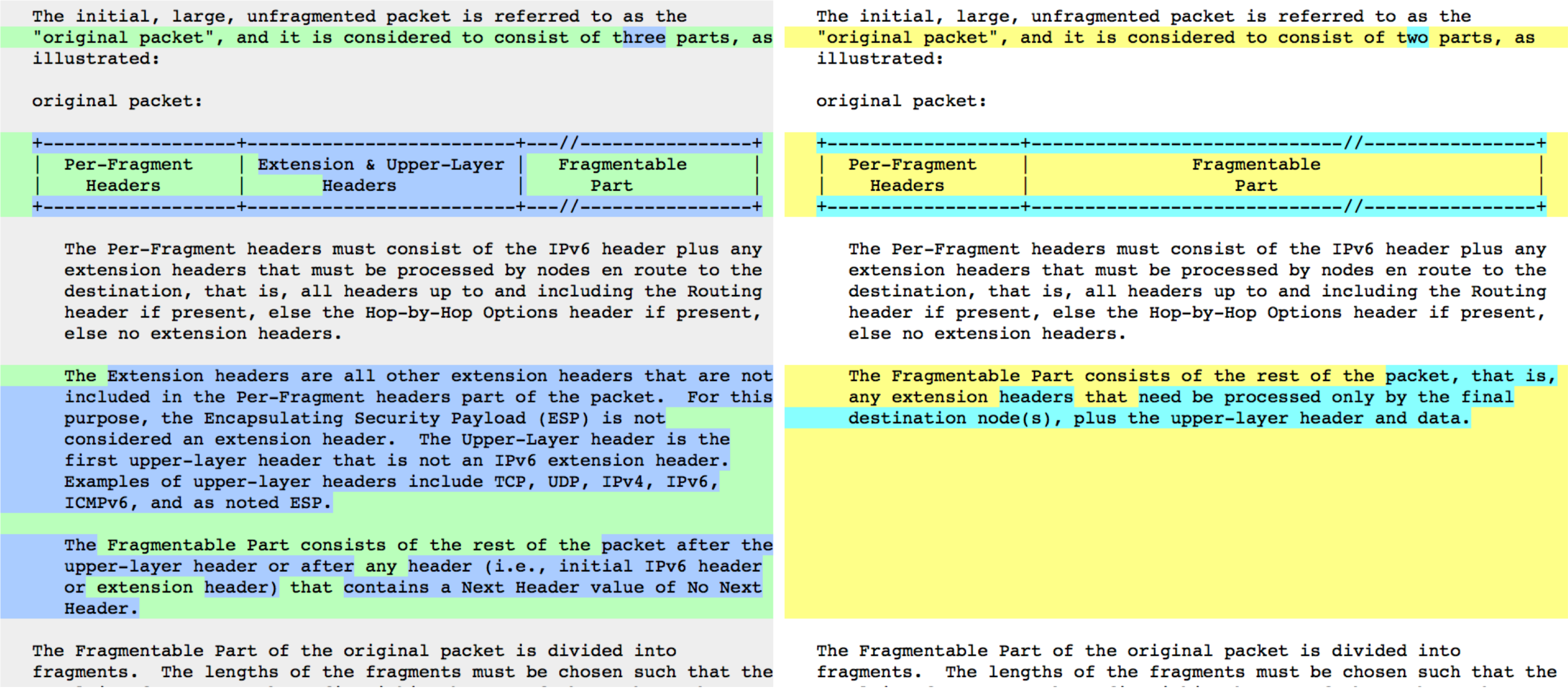
Next Steps

- Proposed revision to Section 4.5 of RFC800
  - Discussed at IETF 105
  - Proposed changes and diff discussed on IPv6 mailing list
- Change generally accepted with one change:
  - Old

    (4) The first fragment.
  - New

    (4) The remainder of the first fragment.

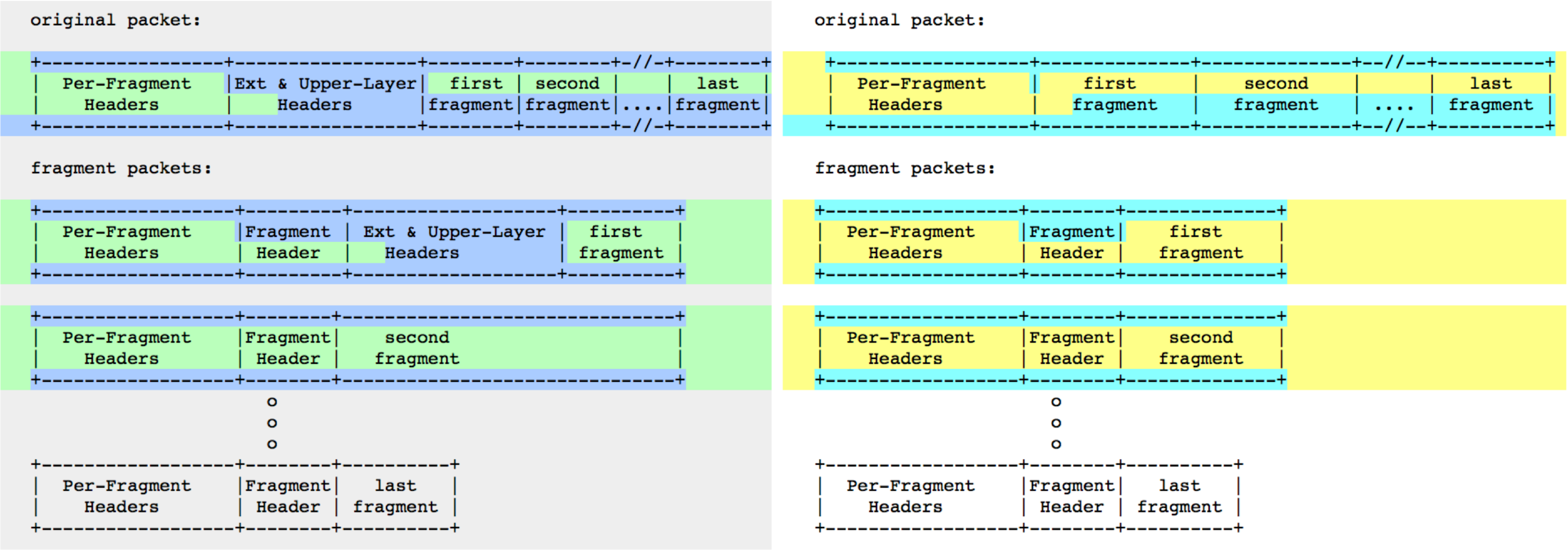
Diff (1 of 4)



```
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:

+------------------+-------------------------+---//----------------+
|  Per-Fragment    | Extension & Upper-Layer |  Fragmentable       |
|    Headers       |        Headers          |     Part            |
+------------------+-------------------------+---//----------------+

   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
```

```
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:

+------------------+----------------------------//-----------------+
|  Per-Fragment    |                Fragmentable                   |
|    Headers       |                    Part                       |
+------------------+----------------------------//-----------------+

   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
```

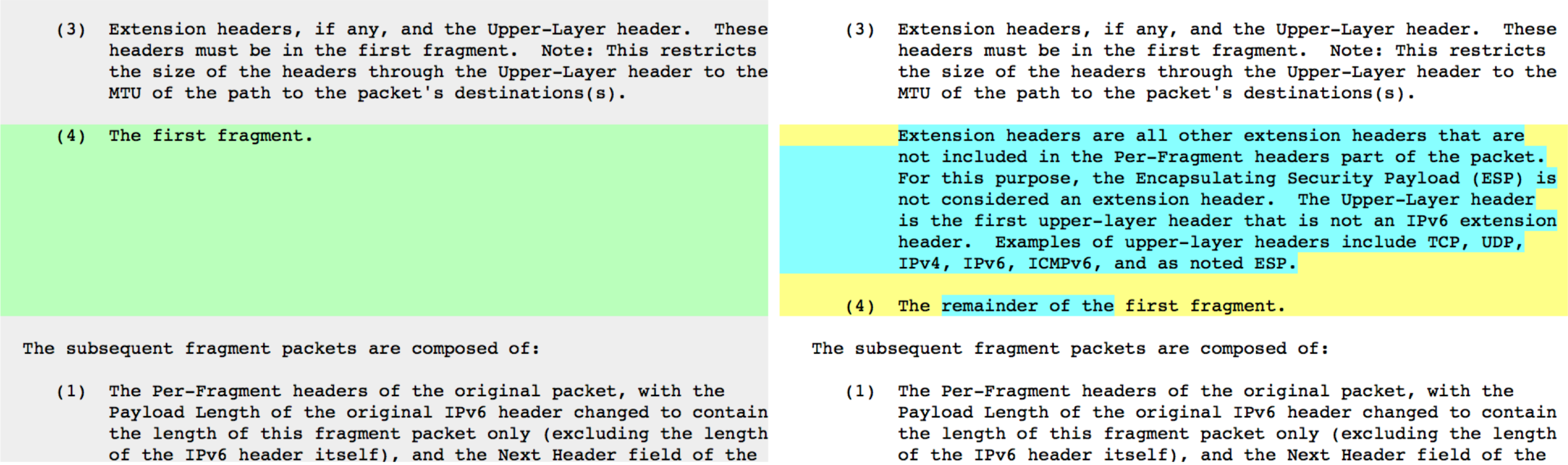
Diff (2 of 4)



```
original packet:

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

fragment packets:

+------------------+---------+-------------------+----------+
|  Per-Fragment    |Fragment | Ext & Upper-Layer |  first   |
|    Headers       | Header  |   Headers         | fragment |
+------------------+---------+-------------------+----------+

+------------------+--------+-------------------------------+
|  Per-Fragment    |Fragment|    second                     |
|    Headers       | Header |   fragment                    |
+------------------+--------+-------------------------------+
                       o
                       o
                       o
+------------------+--------+----------+
|  Per-Fragment    |Fragment|   last   |
|    Headers       | Header | fragment |
+------------------+--------+----------+
```

```
original packet:

+------------------+--------------+--------------+--//--+----------+
|  Per-Fragment    |    first     |    second    |      |   last   |
|    Headers       |   fragment   |   fragment   | .... | fragment |
+------------------+--------------+--------------+--//--+----------+

fragment packets:

+------------------+--------+--------------+
|  Per-Fragment    |Fragment|    first     |
|    Headers       | Header |   fragment   |
+------------------+--------+--------------+

+------------------+--------+--------------+
|  Per-Fragment    |Fragment|    second    |
|    Headers       | Header |   fragment   |
+------------------+--------+--------------+
                       o
                       o
                       o
+------------------+--------+----------+
|  Per-Fragment    |Fragment|   last   |
|    Headers       | Header | fragment |
+------------------+--------+----------+
```

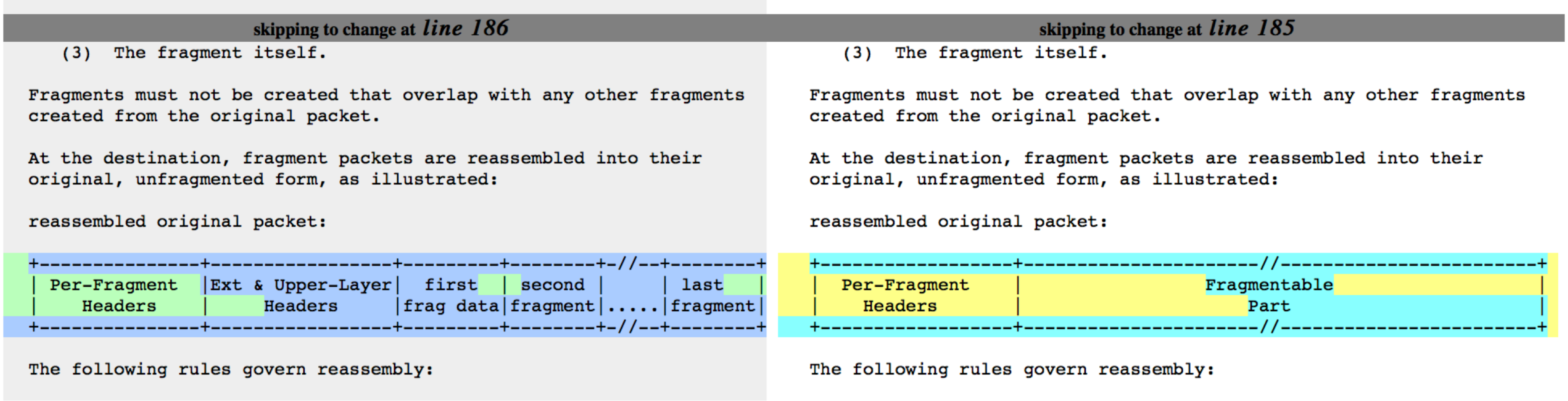
Diff (3 of 4)



```
   (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 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
```

```
   (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 remainder of 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
```

Diff (4 of 4)



skipping to change at *line 186*

```
   (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:

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

The following rules govern reassembly:
```

skipping to change at *line 185*

```
   (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:

+------------------+----------------------//------------------------+
|  Per-Fragment    |                 Fragmentable                   |
|    Headers       |                     Part                       |
+------------------+----------------------//------------------------+

The following rules govern reassembly:
```

Proposed Text

4.5. Fragment Header

The Fragment header is used by an IPv6 source to send a packet larger than would fit in the path MTU to its destination. (Note: unlike IPv4, fragmentation in IPv6 is performed only by source nodes, not by routers along a packet's delivery path -- see [RFC8200].) The Fragment header is identified by a Next Header value of 44 in the immediately preceding header and has the following format:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Header  |   Reserved    |      Fragment Offset    |Res|M|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Identification                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
   Next Header          8-bit selector.  Identifies the initial header
                        type of the Fragmentable Part of the original
                        packet (defined below).  Uses the same values
                        as the IPv4 Protocol field [IANA-PN].

   Reserved             8-bit reserved field.  Initialized to zero for
                        transmission; ignored on reception.

   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 of the original packet.
```

Proposed Text

Res — 2-bit reserved field. Initialized to zero for transmission; ignored on reception.

M flag — 1 = more fragments; 0 = last fragment.

Identification — 32 bits. See description below.

In order to send a packet that is too large to fit in the MTU of the path to its destination, a source node may divide the packet into fragments and send each fragment as a separate packet, to be reassembled at the receiver.

For every packet that is to be fragmented, the source node generates an Identification value. The Identification must be different than that of any other fragmented packet sent recently* with the same Source Address and Destination Address. If a Routing header is present, the Destination Address of concern is that of the final destination.

* "recently" means within the maximum likely lifetime of a 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].

Proposed Text

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

original packet:

```
+------------------+-------------------------+---//----------------+
|  Per-Fragment    |~~Extension & Upper-Layer~~|  Fragmentable       |
|    Headers       |       ~~Headers~~           |      Part           |
+------------------+-------------------------+---//----------------+
```

OLD

```
+------------------+-----------------------------//----------------+
|  Per-Fragment    |            Fragmentable                       |
|    Headers       |                Part                           |
+------------------+-----------------------------//----------------+
```

NEW

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.~~

MOVED TO (3)

Proposed Text

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

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

OLD

```
+------------------+--------------+---------------+--//--+----------+
|  Per-Fragment    |    first     |    second     |      |   last   |
|   Headers        |   fragment   |   fragment    | .... | fragment |
+------------------+--------------+---------------+--//--+----------+
```

NEW

Proposed Text

```
 fragment packets:

+------------------+---------+-------------------+----------+
|  Per-Fragment    |Fragment | ~~Ext & Upper-Layer~~|  first   |
|    Headers       | Header  |    ~~Headers~~        | fragment |
+------------------+---------+-------------------+----------+
```

OLD

```
+------------------+--------+--------------+
|  Per-Fragment    |Fragment|    first     |
|    Headers       | Header |   fragment   |
+------------------+--------+--------------+
```

NEW

```
+------------------+--------+--------------+
|  Per-Fragment    |Fragment|    second    |
|    Headers       | Header |   fragment   |
+------------------+--------+--------------+
                      o
                      o
                      o
+------------------+--------+----------+
|  Per-Fragment    |Fragment|   last   |
|    Headers       | Header | 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.
```

Proposed Text

(2) A Fragment header containing:

The Next Header value that identifies the first header after the Per-Fragment headers of the original packet.

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

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.

MOVED HERE

(4) The remainder of the first fragment.

Proposed Text

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.

(2) A Fragment header containing:

> The Next Header value that identifies the first header after the Per-Fragment headers of the original packet.
>
> A Fragment Offset containing the offset of the fragment, in 8-octet units, relative to the start of the Fragmentable Part of the original packet.
>
> An M flag value of 0 if the fragment is the last ("rightmost") one, else an M flag value of 1.
>
> The Identification value generated for the original packet.

(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:

Proposed Text

```
reassembled original packet:

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

OLD

```
+------------------+----------------------//-------------------------+
|  Per-Fragment    |                    Fragmentable                  |
|    Headers       |                        Part                      |
+------------------+----------------------//-------------------------+
```

NEW

```
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
   zero), with the following two changes:

      The Next Header field of the last header of the Per-Fragment
      headers is obtained from the Next Header field of the first
      fragment's Fragment header.
```

Proposed Text

The Payload Length of the reassembled packet is computed from the length of the Per-Fragment headers and the length and offset of the last fragment. For example, a formula for computing the Payload Length of the reassembled original packet is:

```
   PL.orig = PL.first - FL.first - 8 + (8 * FO.last) + FL.last


   where
   PL.orig  =  Payload Length field of reassembled packet.
   PL.first =  Payload Length field of first fragment packet.
   FL.first =  length of fragment following Fragment header of
               first fragment packet.
   FO.last  =  Fragment Offset field of Fragment header of last
               fragment packet.
   FL.last  =  length of fragment following Fragment header of
               last fragment packet.
```

The 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 is computed from its Fragment Offset value.

The Fragment header is not present in the final, reassembled packet.

Proposed Text

If the fragment is a whole datagram (that is, both the Fragment Offset field and the M flag are zero), then it does not need any further reassembly and should be processed as a fully reassembled packet (i.e., updating Next Header, adjust Payload Length, removing the Fragment header, etc.). Any other fragments that match this packet (i.e., the same IPv6 Source Address, IPv6 Destination Address, and Fragment Identification) should be processed independently.

The following error conditions may arise when reassembling fragmented packets:

- If insufficient fragments are received to complete reassembly of a packet within 60 seconds of the reception of the first-arriving fragment of that packet, reassembly of that packet must be abandoned and all the fragments that have been received for that packet must be discarded. If the first fragment (i.e., the one with a Fragment Offset of zero) has been received, an ICMP Time Exceeded -- Fragment Reassembly Time Exceeded message should be sent to the source of that fragment.

- If the length of a fragment, as derived from the fragment packet's Payload Length field, is not a multiple of 8 octets and the M flag of that fragment is 1, then that fragment must be discarded and an ICMP Parameter Problem, Code 0, message should be sent to the source of the fragment, pointing to the Payload Length field of the fragment packet.

- If the length and offset of a fragment are such that the Payload Length of the packet reassembled from that fragment

Proposed Text

would exceed 65,535 octets, then that fragment must be discarded and an ICMP Parameter Problem, Code 0, message should be sent to the source of the fragment, pointing to the Fragment Offset field of the fragment packet.

- o If the first fragment does not include all headers through an Upper-Layer header, then that fragment should be discarded and an ICMP Parameter Problem, Code 3, message should be sent to the source of the fragment, with the Pointer field set to zero.

- o If any of the fragments being reassembled overlap with any other fragments being reassembled for the same packet, reassembly of that packet must be abandoned and all the fragments that have been received for that packet must be discarded, and no ICMP error messages should be sent.

  It should be noted that fragments may be duplicated in the network. Instead of treating these exact duplicate fragments as overlapping fragments, an implementation may choose to detect this case and drop exact duplicate fragments while keeping the other fragments belonging to the same packet.

The following conditions are not expected to occur frequently but are not considered errors if they do:

The number and content of the headers preceding the Fragment header of different fragments of the same original packet may differ. Whatever headers are present, preceding the Fragment

Proposed Text

header in each fragment packet, are processed when the packets arrive, prior to queueing the fragments for reassembly. Only those headers in the Offset zero fragment packet are retained in the reassembled packet.

The Next Header values in the Fragment headers of different fragments of the same original packet may differ. Only the value from the Offset zero fragment packet is used for reassembly.

Other fields in the IPv6 header may also vary across the fragments being reassembled. Specifications that use these fields may provide additional instructions if the basic mechanism of using the values from the Offset zero fragment is not sufficient. For example, Section 5.3 of [RFC3168] describes how to combine the Explicit Congestion Notification (ECN) bits from different fragments to derive the ECN bits of the reassembled packet.

Plan for Errata

- Plan agreed with our Area Director
- Open a new Errata that describes the errors in the Fragmentation text in Section 4.5 of RFC8200 and include the revised text.
    - Basically replace Section 4.5
    - Status of “Verified”
- Close Errata 5170, 5171, 5172, 5173 as “Rejected" with a pointer to the new Errata.

Proposed Errata

- Current Text
  - Section 4.5 of RFC8200
- Proposed Text
  - (described here)
- Note
  - Fragment Offset defined as pointing to “Fragmentable Part”, should have pointed to “Extension & Upper-Layer Headers”
  - This errata replaces Errata 5170, 5171, 5172, 5173.
  - This change has been reviewed by the IETF 6MAN working group.

QUESTIONS / COMMENTS?