IP Packet Fragmentation Tutorial

Dr. E. Garcia, admin@miislita.com

Published: November 16, 2009. Last Update: November 16, 2009. http://www.miislita.com/

Abstract–This tutorial covers IP fragmentation, data payloads, IP packet and header lengths, maximum transmission unit (*MTU*), and fragmentation offset (*FO*). Expressions describing these concepts are systematically derived and explained.

Keywords: IP fragmentation, data payloads, IP packet, IP headers, maximum transmission unit, *MTU*, fragmentation offset, *FO*

1 Data Payloads

The data payload (*DP*) of an IP packet (IP datagram) is defined as the packet length (*PL*) minus the length of its IP header (*IPHL*),

$$DP = PL - IPHL \quad \text{(Eq 1)}$$

Per RFCs 791 and 829, the largest packet allowed by TCP/IP standards is 65,535 (i.e., $2^{16} - 1$) octets, the minimum *IPHL* is 20 octets, and the maximum *IPHL* is 60 octets. An octet is a group of 8 elements. In computer sciences, an octet is a group of 8 bits called a byte. Octet is also the French word for byte. Thus, the largest data payload of an IP packet that conforms to TCP/IP standards is 65,515 bytes. Since such long datagrams are impractical for most hosts and networks, some limiting values have been adopted. One of these limiting values is the so-called *Maximum Transmission Unit* size or *MTU*.

2 *MTU* Values

MTU is defined as the largest IP packet that can be transmitted without fragmentation. However, 'maximum' is a relative term, subject to technology requirements. For instance according to RFC 791 (emphasis added), "All hosts must be prepared to accept datagrams of up to 576 octets (whether they arrive whole or in fragments). It is recommended that hosts **only send datagrams larger than 576 octets** if they have assurance that the destination is prepared to accept the larger datagrams.". This is a long established TCP/IP rule.

Thus, hosts should not send packets with *MTU* values greater than 576 bytes, unless they know that the destination can accept larger packets. As network and media types evolve, vendors have indeed adopted *MTU* values greater than the 576 mark. Per RFC 1191, the default *MTU* for *Ethernet v2* networks is 1,500 bytes. Table 1 lists *MTU*s for different *Ethernet* versions, networks, and media types.

Table 1. *MTU* values for several Ethernet versions, networks, and media types (RFCs 879, 1191, and 2675).

Ethernet version	*802.3 (v1)*	*v2*	*802.11*	*802.5*	*Jumbo Frames*	
MTU, in bytes	*1,492*	*1,500*	*2,272*	*4,464*	*1,500 – 9,000*	
Media types	*Internet IPv4*	*ARPANET, MILNET*	*Internet IPv6*	*Proteon PRONET*	*Token Ring*	*FDDI*
MTU, in bytes	*at least 68*	*1,007*	*at least 1,280*	*2,046*	*4,096*	*4,500*

As nearly all IP over *Ethernet* implementations are *Ethernet v2* the 1,500 mark is used in this tutorial, unless stated otherwise. Since the default IP header is 20 bytes, a working data payload of 1,500 – 20 = 1,480 bytes is assumed. Techniques for determining experimentally *MTU* values are discussed in the ***MTU and MSS Tutorial*** (http://www.miislita.com/internet-engineering/mtu-mss-tutorial.pdf) and elsewhere (1 – 2).

3 IP Fragmentation

IP packets larger than the permitted *MTU* must be divided into *n* smaller packets called fragments, each with their own header length, packet length (*pl*), and data payload (*dp*).

$$DP = dp_1 + dp_2 \ldots + dp_n = \textstyle\sum_i^n dp_i \quad \text{(Eq 2)}$$

$$DP = (pl_1 - IPHL) + (pl_2 - IPHL) \ldots + (pl_n - IPHL) \quad \text{(Eq 3)}$$

Equation 3 can be rewritten as

$$DP = (MTU - IPHL) + (MTU - IPHL) \ldots + dp_n \quad \text{(Eq 4)}$$

$$DP = (n-1)(MTU - IPHL) \ldots + dp_n \quad \text{(Eq 5)}$$

From Equation 5, two important concepts can be derived: (1) fragment offsets, and (2) number of constituting fragments.

4 Fragment Offsets

Given a sequence of fragments from the same packet, the fragment offset (*FO*) of fragment *i* is defined as the sum of data payloads of previous fragments, expressed as a multiple of 8. Since *i = 1, 2, 3 …n*, Equation 5 leads to

$$FO_i = \frac{DP - dp_i}{8} = \frac{PL - IPHL - dp_i}{8} \quad \text{(Eq 6)}$$

$$FO_i = (i-1)\frac{(MTU - IPHL)}{8} \quad \text{(Eq 7)}$$

Table 2 depicts the expected sequence of *FO* values for *MTU* = 1,500 bytes and *IPHL* = 20 bytes.

Table 2. Sequence of *FO* values for *MTU* = 1,500 bytes and *IPHL* = 20 bytes

i	*1*	*2*	*3*	*4*	…	*n*
FO_i	*0*	*185*	*370*	*555*	…	*(n - 1)*185*

Once a packet is fragmented, each fragment is tagged with the destination address and a sequence number. As the fragments traverse the network between the source and destination hosts the best current path available to each fragment is used. In this way, if a network link goes down, not all the fragments have to be re-sent.

At the destination host, the fragments might arrive at different times, out of sequence, or even through the same path or different paths. When all the fragments reach the destination, these are re-ordered according to the sequence numbers and *FO's* and the original packet reconstructed. Since all the fragments must arrive at the destination host using the best current path available and lost fragments must be re-sent prior to reassembling, intermediate routers should not attempt to reassemble the fragments back to the original packet.

5 Number of Fragments

When a packet is fragmented, normally the length and data payload of the last fragment is not known in advance. All it can be said is that $pl_n \leq MTU$ and $dp_n \leq MTU - IPHL$. Equation 5 then leads to two possible scenarios:

Scenario 1: if $dp_n = MTU - IPHL$

$$DP = n * (MTU - IPHL) \quad \text{(Eq 8)}$$

$$PL - IPHL = n * (MTU - IPHL) \quad \text{(Eq 9)}$$

$$\frac{Dp}{MTU - IPHL} = \frac{PL - IPHL}{MTU - IPHL} = n \quad \text{(Eq 10)}$$

Scenario 2: if $dp_n < MTU - IPHL$

$$DP > (n-1) * (MTU - IPHL) \quad \text{(Eq 11)}$$

$$PL - IPHL > (n-1) * (MTU - IPHL) \quad \text{(Eq 12)}$$

$$\frac{Dp}{MTU - IPHL} = \frac{PL - IPHL}{MTU - IPHL} > n - 1 \quad \text{(Eq 13)}$$

$$\frac{Dp}{MTU - IPHL} + 1 = \frac{PL - IPHL}{MTU - IPHL} + 1 > n \quad \text{(Eq 14)}$$

Thus in order for dp_n be equal to *MTU – IPHL*, the ratio *(PL – IPHL)/(MTU – IPHL)* must be a whole number (integer); otherwise the ratio must be incremented by 1 and *n* be calculated by rounding off the result to the nearest integer.

Moreover, dropping the *IPHL* term from Equations 13 and 14 allows one to predict the following lower and upper bounds for *n*:

$$PL/MTU \leq n \leq PL/MTU + 1 \quad \text{(Eq 15)}$$

Last but not least, an IP packet encapsulates a TCP packet such that *DP = TCPHL + MSS*. Here *TCPHL* is the length of the TCP header and *MSS* is the data payload of the TCP packet, also known as the *Maximum Segment Size*. The ***MTU and MSS Tutorial*** (http://www.miislita.com/internet-engineering/mtu-mss-tutorial.pdf) expands on this. Figure 1 and the last section of this tutorial illustrate all these concepts.

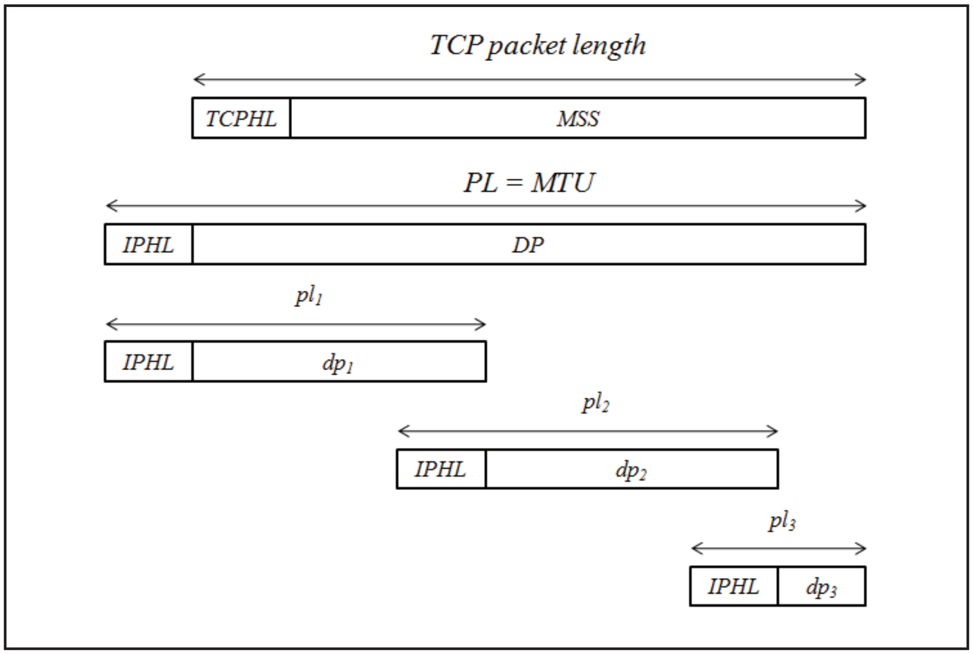

Figure 1. Fragmentation example where $MTU = PL = pl_1 = pl_2 > pl_3$ and $DP = dp_1 + dp_2 + dp_3 = PL - IPHL$.

6 Sample Calculations

Problem

A 3,000-byte IP packet with a default *IPHL* of 20 bytes must be sent over an *Ethernet v2* network.

1. What is the data payload of the packet?
2. In how many fragments this packet must be decomposed?
3. What are the *FO's* of the fragments?
4. What are the data payloads of the fragments?
5. How long the fragments are?
6. What are the expected lower and upper bounds of *n*?

Solution

For *Ethernet v2*, *MTU* = 1,500 bytes.

1. From Equation 1, the data payload of the packet is

   DP = 3,000 – 20 = 2,980 bytes

2. From Equation 10,

   n = (3,000 – 20)/(1,500 – 20) = 2.0135…

   Since this result is not a whole number, it is incremented by 1 and rounded off to the nearest integer; i.e.

   n = 2.0135… + 1 = 3.0135… = 3

3. From Equation 7, the fragment offsets are:

   FO_1 = (1 – 1)*(1,500 – 20)/8 = 0
   FO_2 = (2 – 1)*(1,500 – 20)/8 = 185
   FO_3 = (3 – 1)*(1,500 – 20)/8 = 370

4. From Equations 2 through 5, the data payloads of the fragments are:

   dp_1 = 1,500 – 20 = 1480 bytes
   dp_2 = 1,500 – 20 = 1480 bytes
   $dp_3 = DP - (dp_1 + dp_2) = DP - 8*FO_3$ = 2980 – 2,960 = 20 bytes

5. The lengths of the fragments are:

   $pl_1 = dp_1 + IPHL$ = 1,500 bytes
   $pl_2 = dp_2 + IPHL$ = 1,500 bytes
   $pl_3 = dp_3 + IPHL$ = 40 bytes

6. From Equation 15, the expected lower and upper bounds of n are:

   $PL/MTU = 3{,}000/1{,}500 = 2$

   $PL/MTU + 1 = 3$

   $2 \leq n \leq 3$

   Therefore regardless of whether *IPHL* is 20 or 60 bytes, the packets must be broken up into no more than 3 fragments.

7 Exercises

1. A 2,400-byte packet with an IP header of 20 bytes must be broken into fragments no larger than 620 bytes.

   a. In how many fragments should the packet be decomposed?
   b. What is the data payload of the packet?
   c. What are the data payloads, lengths, and *FO's* of the fragments?

2. A packet is decomposed into 3 fragments, each with a header IP length of 20 bytes. The length of last fragment is 300 bytes and its *FO* is 150.

   a. What is the network *MTU*?
   b. What is the data payload of the last fragment?
   c. What is the combined data payload of the fragments excluding last fragment?
   d. What are the *FO's* of the fragments?
   e. What is the length and data payload of the original packet?

3. Suppose that in Exercise 2, a second packet must be sent as a group of three fragments. Calculate the minimum and maximum size allowed for this packet.

References

1. Garcia, E. *MTU and MSS Tutorial* Mi Islita.com; 2009.
   [On-line]. Available: http://www.miislita.com/internet-engineering/mtu-mss-tutorial.pdf
2. Internet Engineering Task Force.
   [On-line]. Available: http://www.ietf.org