Computer Network

Lecture-32

Dharmendra Kumar (Associate Professor)
Department of Computer Science and Engineering
United College of Engineering and Research,
Prayagraj

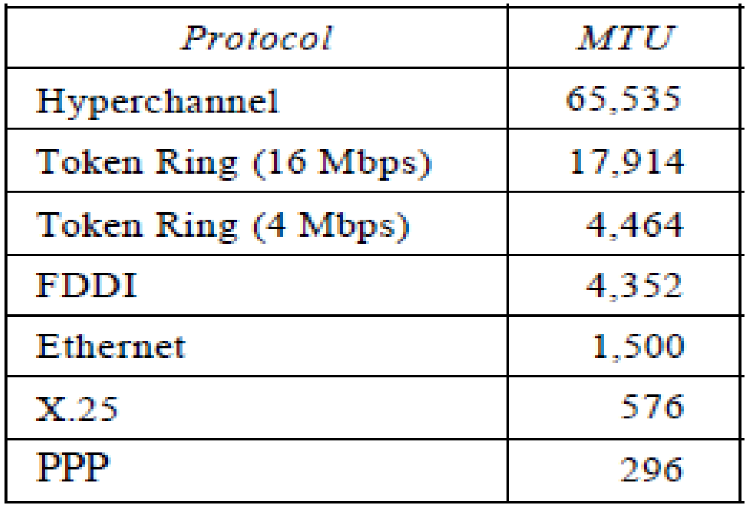
IPv4

Fragmentation

Maximum Transfer Unit (MTU)

Each data link layer protocol has its own frame format in most protocols. One of the fields defined in the format is the maximum size of the data field. In other words, when a datagram is encapsulated in a frame, the total size of the datagram must be less than this maximum size, which is defined by the restrictions imposed by the hardware and software used in the network.

The value of the MTU depends on the physical network protocol. Following table shows the values for some protocols:-



Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

IPv4

- If we divide the datagram into fragments to make it possible to pass through the networks, then this process is called the **fragmentation**.
- When a datagram is fragmented, each fragment has its own header with most of the fields repeated, but with some changed. A fragmented datagram may itself be fragmented if it encounters a network with an even smaller MTU. In other words, a datagram can be fragmented several times before it reaches the final destination.

IPv4

Fields Related to Fragmentation

The fields that are related to fragmentation and reassembly of an IPv4 datagram are the identification, flags, and fragmentation offset fields.

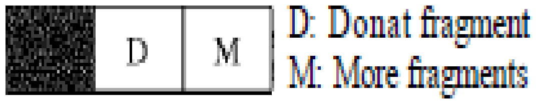
Identification:

- ❖ This 16-bit field identifies a datagram originating from the source host. The combination of the identification and source IPv4 address must uniquely define a datagram as it leaves the source host.
- ❖ When a datagram is fragmented, the value in the identification field is copied to all fragments. In other words, all fragments have the same identification number, the same as the original datagram.
- ❖ The identification number helps the destination in reassembling the datagram. It knows that all fragments having the same identification value must be assembled into one datagram.

IPv4

Flags:

- This is a 3-bit field.
- The first bit is reserved.

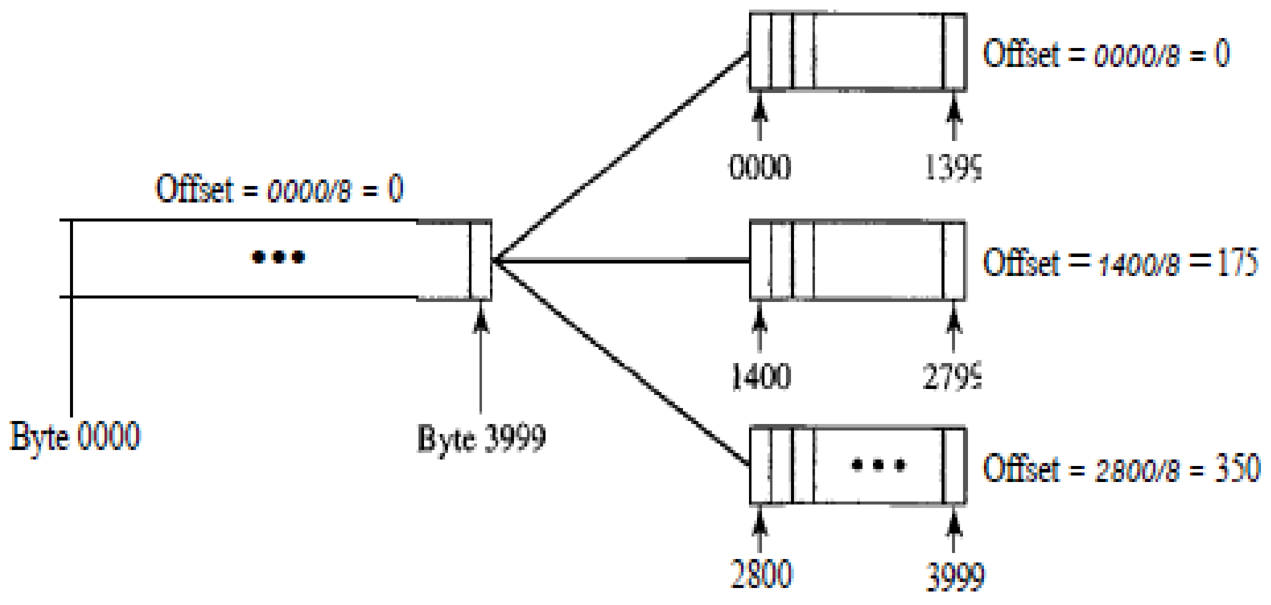


- The second bit is called the do not fragment bit. If its value is 1, the machine must not fragment the datagram. If it cannot pass the datagram through any available physical network, it discards the datagram and sends an ICMP error message to the source host. If its value is 0, the datagram can be fragmented if necessary.
- The third bit is called the more fragment bit. If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment.

IPv4

Fragmentation offset: This 13-bit field shows the relative position of this fragment with respect to the whole datagram. It is the offset of the data in the original datagram measured in units of 8 bytes.

Example: Following figure shows a datagram with a data size of 4000 bytes fragmented into three fragments.

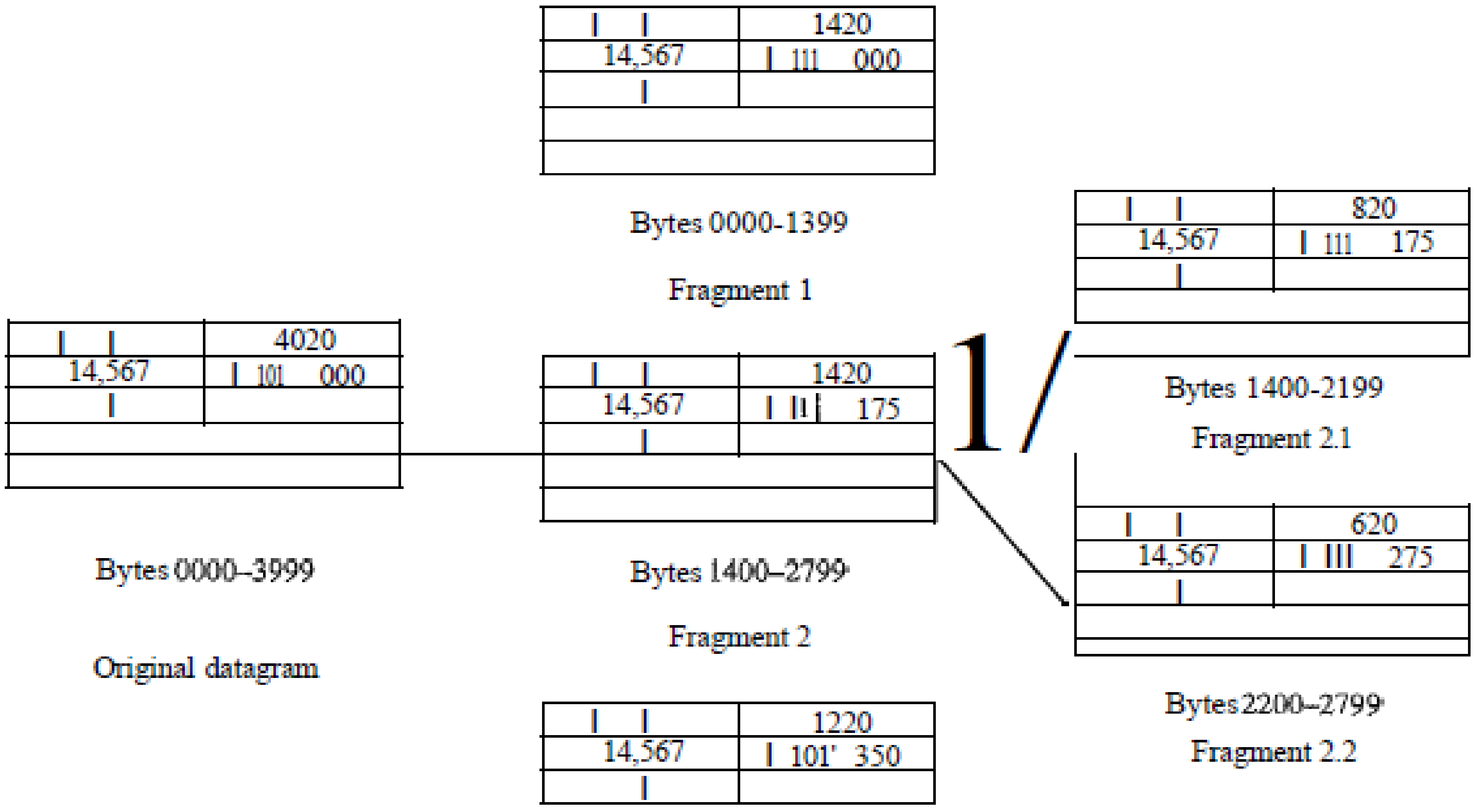


IPv4

The bytes in the original datagram are numbered 0 to 3999.

- The first fragment carries bytes 0 to 1399. The offset for this datagram is 0/8 = 0.
- The second fragment carries bytes 1400 to 2799; the offset value for this fragment is 1400/8 = 175.
- Finally, the third fragment carries bytes 2800 to 3999. The offset value for this fragment is 2800/8 =350.

IPv4


1420
14,567
1 111 000
Bytes 0000-1399
Fragment 1
4020
14,567
1 101 000
Bytes 0000–3999
Original datagram
1420
14,567
1 111 175
Bytes 1400–2799
Fragment 2
820
14,567
1 111 175
Bytes 1400-2199
Fragment 2.1
620
14,567
1 111 275
Bytes 2200–2799
Fragment 2.2
1220
14,567
1 101' 350
Bytes 2800-3999
Fragment 3

IPv4

If each fragment follows a different path and arrives out of order, the final destination host can reassemble the original datagram from the fragments received (if none of them is lost) by using the following strategy:

1. The first fragment has an offset field value of zero.
2. Divide the length of the first fragment by 8. The second fragment has an offset value equal to that result.
3. Divide the total length of the first and second fragments by 8. The third fragment has an offset value equal to that result.
4. Continue the process. The last fragment has a more bit value of 0.

IPv4

Example

A packet has arrived with an M bit value of O. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 0, it means that there are no more fragments; the fragment is the last one. However, we cannot say if the original packet was fragmented or not. A non fragmented packet is considered the last fragment.

Example

A packet has arrived with an M bit value of 1. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 1, it means that there is at least one more fragment. This fragment can be the first one or a middle one, but not the last one. We don't know if it is the first one or a middle one; we need more information (the value of the fragmentation offset).

IPv4

Example

A packet has arrived with an M bit value of 1 and a fragmentation offset value of 0. Is this the first fragment, the last fragment, or a middle fragment?

Solution

Because the M bit is l, it is either the first fragment or a middle one. Because the offset value is 0, it is the first fragment.

Example

A packet has arrived in which the offset value is 100. What is the number of the first byte? Do we know the number of the last byte?

Solution

To find the number of the first byte, we multiply the offset value by 8. This means that the first byte number is 800. We cannot determine the number of the last byte unless we know the length of the data.

IPv4

Example

A packet has arrived in which the offset value is 100, the value of HLEN is 5, and the value of the total length field is 100. What are the numbers of the first byte and the last byte?

Solution

The first byte number is 100 x 8 = 800. The total length is 100 bytes, and the header length is 20 bytes (5 x 4), which means that there are 80 bytes in this datagram. If the first byte number is 800, the last byte number must be 879.