# SRM
# Institute of Science and Technology

## 21CSC302J-COMPUTER NETWORKS

## Unit- III

# Forwarding of IP Packets

# IP Fragmentation

The unit of communication at the network layer is a datagram.

- A datagram can travel through different networks.

- **Each router**
  - ***Decapsulates the IPv4 datagram from the frame it receives,***
  - ***Processes it, and then***
  - ***Encapsulates it in another frame.***

- The **format and size of the sent and received frame** ***depend on the protocol used by the physical network***

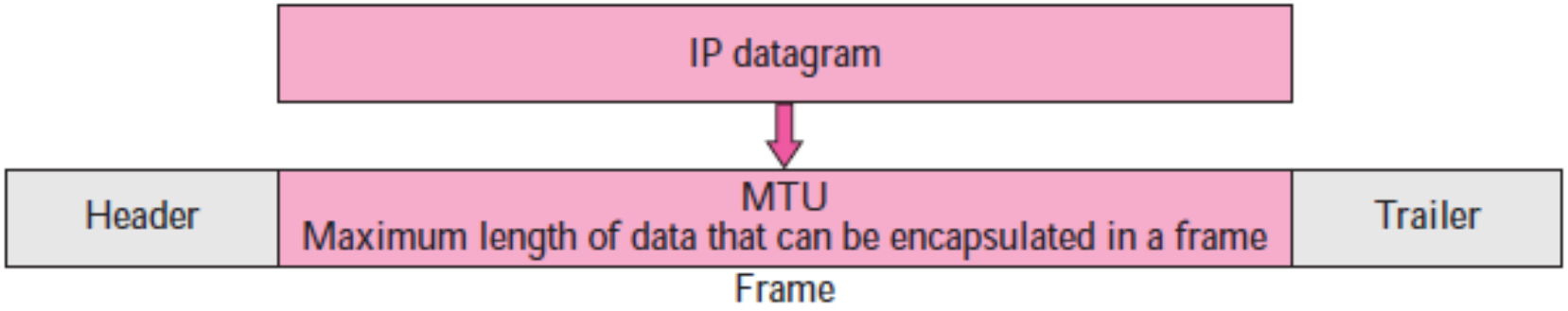
## For example

If a router connects a LAN to a WAN,

- *Receives a frame in the LAN format and*
- *Sends a frame in the WAN format.*

# Maximum Transfer Unit (MTU)

- Each data link layer protocol has *its own frame format*
- One of the fields defined in the format is *the maximum size of the data field.*
- When a datagram is encapsulated in a frame,
  - *The total size of the datagram must be less than the maximum size,*
  - *The restrictions imposed by the hardware and software used in the network*


IP datagram
Header
MTU
Maximum length of data that can be encapsulated in a frame
Trailer
Frame

# Fragmentation

- The value of the MTU *differs from one physical network protocol to another.*

For example

The value for the Ethernet LAN is 1500 bytes, for FDDI LAN is 4352 bytes, and for PPP is 296 bytes.

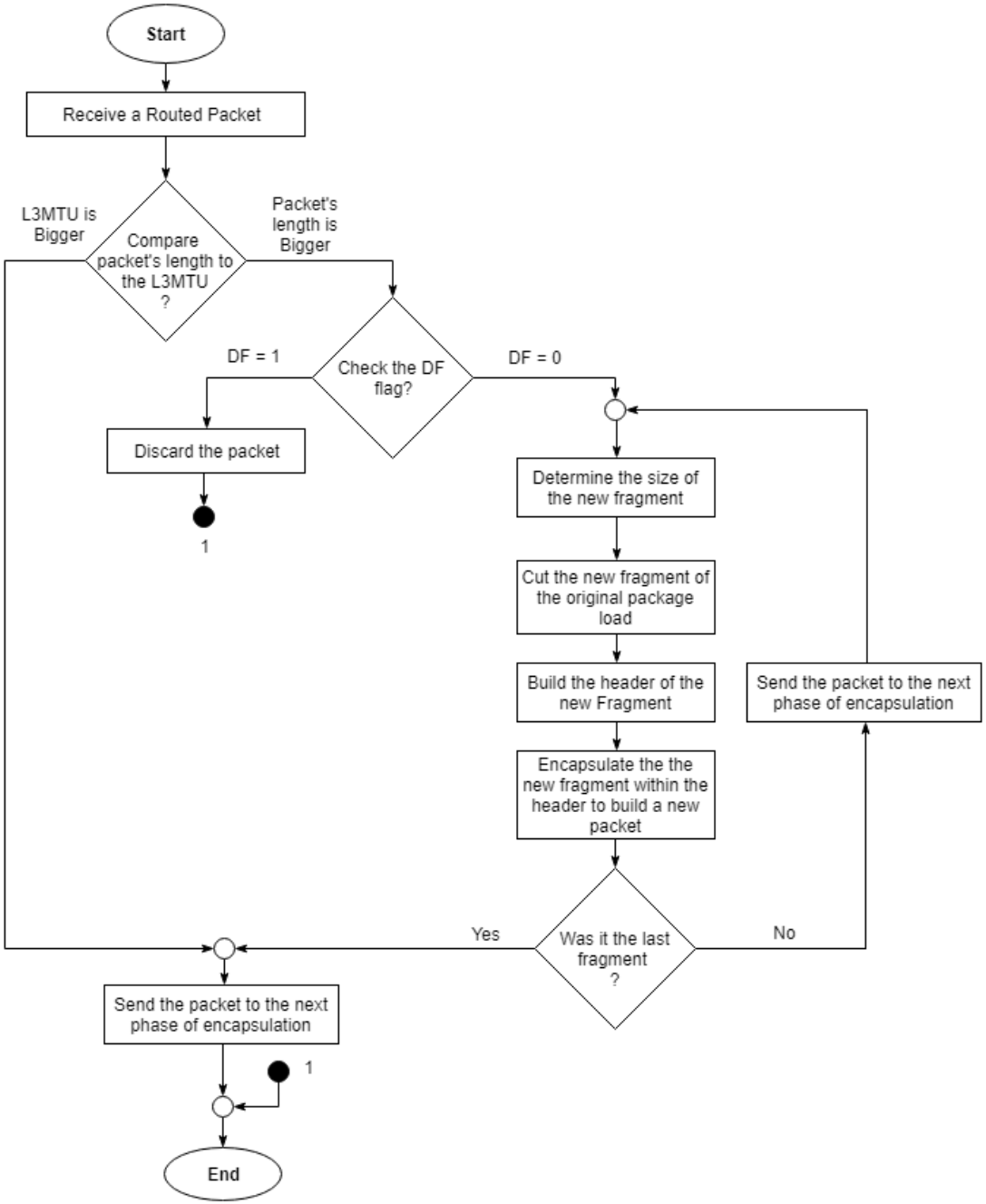
# Fragmentation

- To **make the IP protocol independent of the physical network**
  - ***Designers decided to make the maximum length of the IP datagram equal to 65,535 bytes.***

- This makes transmission more efficient if we use a protocol with an MTU of this size.

- However, for other physical networks, we must divide the datagram to make it possible to pass through these networks.

- This is called fragmentation.


Start
Receive a Routed Packet
Compare packet's length to the L3MTU ?
L3MTU is Bigger
Packet's length is Bigger
Check the DF flag?
DF = 1
DF = 0
Discard the packet
1
Determine the size of the new fragment
Cut the new fragment of the original package load
Build the header of the new Fragment
Encapsulate the the new fragment within the header to build a new packet
Send the packet to the next phase of encapsulation
Was it the last fragment ?
Yes
No
Send the packet to the next phase of encapsulation
1
End

# Maximum Transfer Unit (MTU)

- **The source *usually does not fragment the IP packet.***
- **The transport layer *segment the data into a size that can be accommodated by IP and the data link layer in use.***
- **When a datagram is fragmented**, ***each fragment has its own header with most of the fields repeated***, but some changed.
- A fragmented datagram may itself be fragmented if it encounters a network with an even smaller MTU.

# Maximum Transfer Unit (MTU)

- A datagram can be fragmented several times before it reaches the final destination.
- **A datagram can be fragmented *by the source host or any router in the path.***
- **The reassembly of the datagram**, ***is done only by the destination host*** because each fragment becomes an independent datagram.
- Whereas the fragmented datagram can travel through different routes

# Maximum Transfer Unit (MTU)

- We can **never control or guarantee** ***which route a fragmented datagram may take,***
- All of the fragments belonging to the same datagram should finally arrive at the destination host.
- So it is logical to do the reassembly at the final destination.
- **When a datagram is fragmented**, ***required parts of the header must be copied by all fragments***.

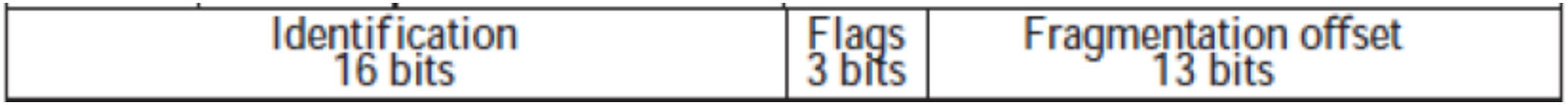
# Maximum Transfer Unit (MTU)

- The option field may or may not be copied
- **The host or router fragments a datagram must change the values of three fields**:
  - ***flags, fragmentation offset, and total length.***
- The rest of the fields must be copied.
- **The value of the checksum *must be recalculated*** regardless of fragmentation.

Only data in a datagram is fragmented.

# Fields Related to Fragmentation

- The fields that are related to fragmentation and reassembly of an IP 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 IP address must uniquely define a datagram as it leaves the source host.

- **To guarantee uniqueness**, ***the IP protocol uses a counter*** **to label the datagrams.**

- The counter is initialized to a positive number.

# Identification

- **When the IP protocol sends a datagram,**
  - ***Copy the current value of the counter to the identification field***
  - ***increments the counter by one.***

- As long as the counter is kept in the main memory, uniqueness is guaranteed.

- **When a datagram is fragmented**, ***the value in the identification field is copied into all fragments***.

- All fragments have the same identification number, which is also the same as the original datagram.

# Identification

- The identification number **helps the destination in reassembling the datagram.**

- It knows that all fragments having the same identification value should be assembled into one datagram.

# Flags

- This is a **three-bit field**.
- The **first bit is reserved (not used).**
- The **second bit** is called the ***do not fragment bit.***
  - ***D is 1, the machine must not fragment the datagram.***
  - ***D is 0, the datagram can be fragmented if necessary.***

# Flags

- The **third bit** is called the ***more fragment bit (M)***.
- **M is 1**, it means the datagram is not the last fragment; ***there are more fragments after this one***.
- **M is 0,** it means ***this is the last or only fragment***

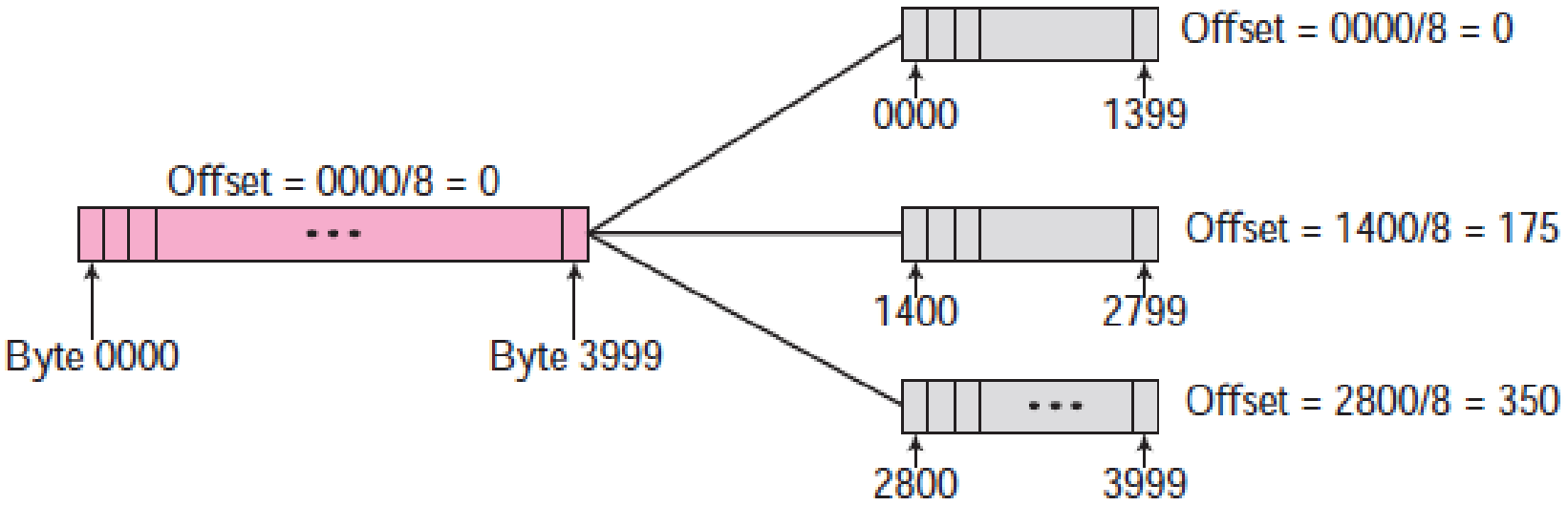
# Fragmentation offset

- 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.

- Figure shows a datagram with a data size of 4000 bytes fragmented into three fragments.

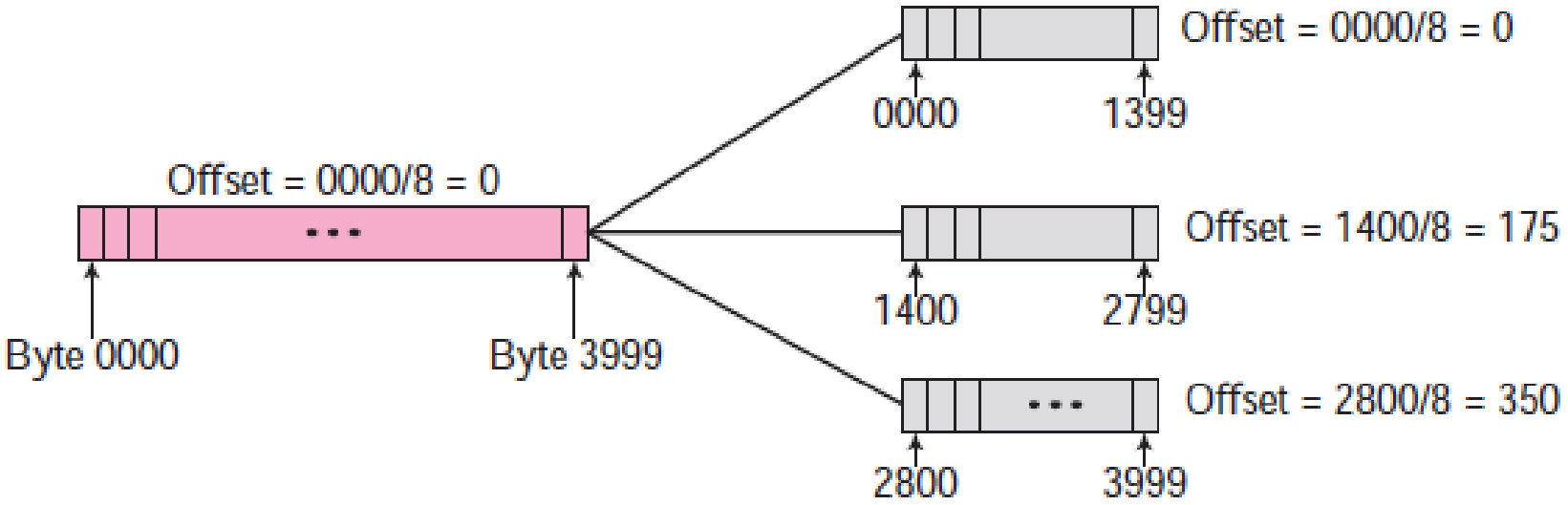
- The bytes in the original datagram are numbered 0 to 3999.




**Fragmentation Example**

# Fragmentation offset

- 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.




**Fragmentation Example**

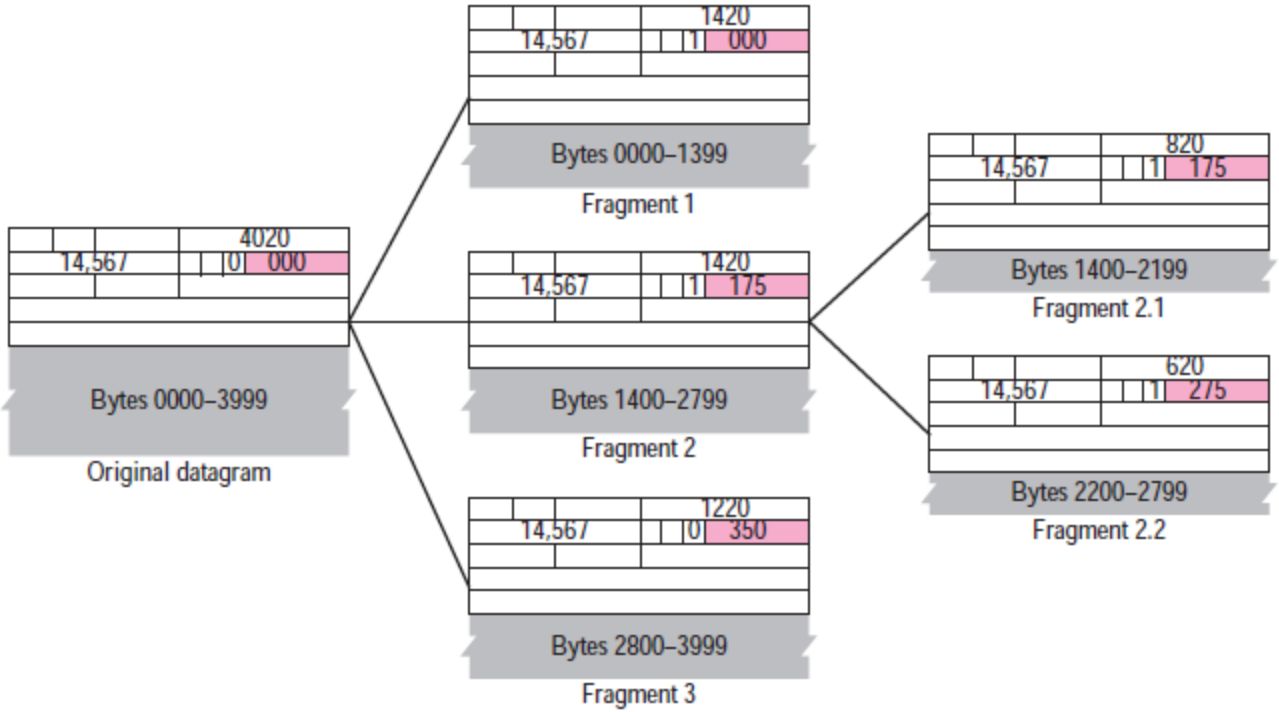
# Fragmentation offset

- **The value of the offset *is measured in units of 8 bytes.***

- This is done because the length of the offset field is only 13 bits long and cannot represent a sequence of bytes greater than 8191.

- This forces hosts or routers that fragment datagrams to choose the size of each fragment so that the first byte number is divisible by 8.

- Figure shows an expanded view of the fragments in the previous figure.




# Fragmentation Example

# Fragmentation offset

- **The value of the identification field *is the same in all fragments.***
- **The value of the flags field *with the more bit set for all fragments except the last.***
- **The value of the offset field *for each fragment is shown.***
- The figure also shows what happens if a fragment itself is fragmented.
- In this case the value of the offset field is always relative to the original datagram.

- The final destination host can reassemble the original datagram from the fragments received (if none of them is lost) using the following strategy:

The first fragment has an offset field value of zero.

Divide the length of the first fragment by 8. The second fragment has an offset value equal to that result.

Divide the total length of the first and second fragment by 8. The third fragment has an offset value equal to that result.

Continue the process. The last fragment has a more bit value of 0.

A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

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

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

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?

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 is the number of the first byte and the last byte?

# Thank You