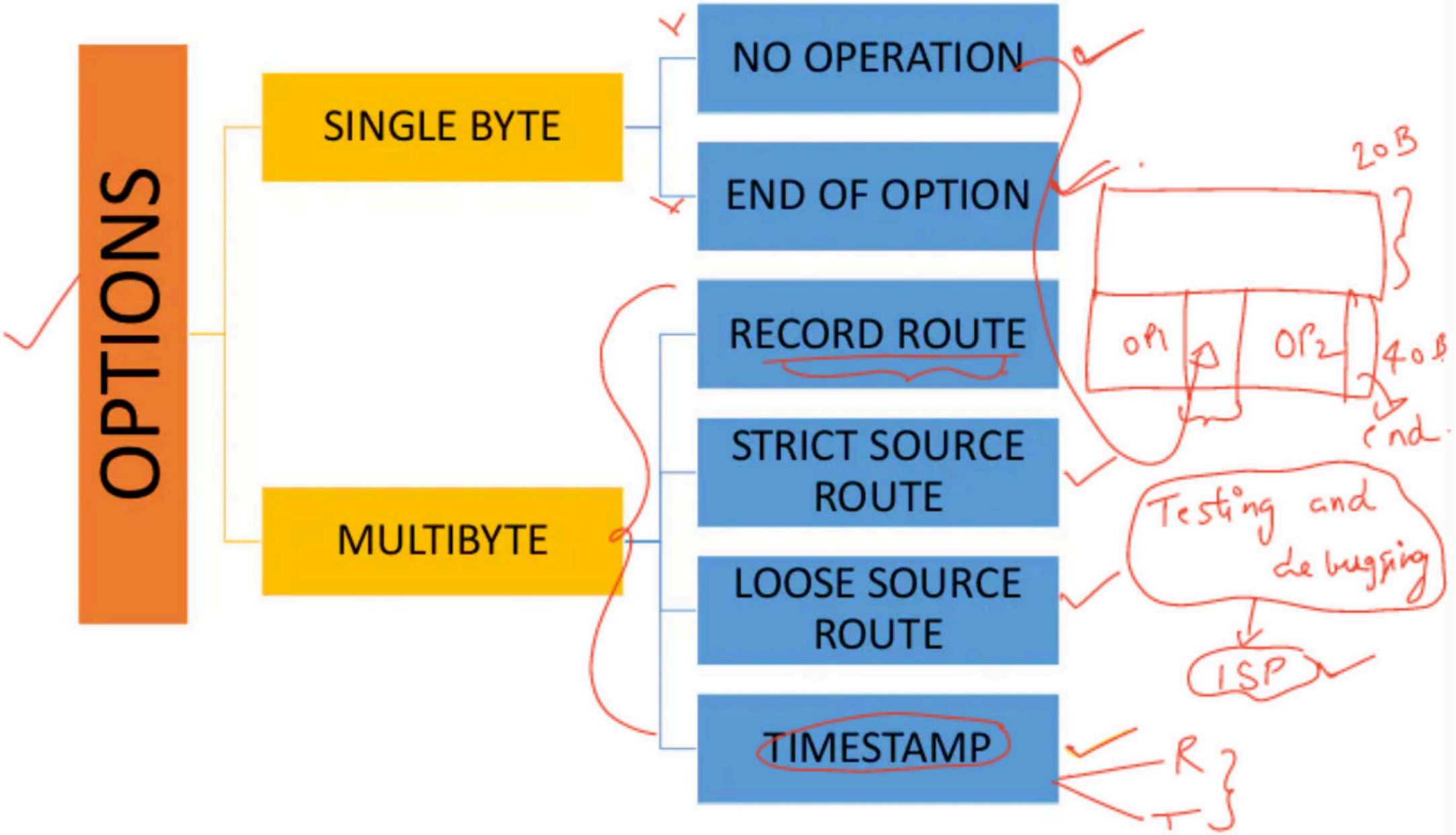
Fragmentation with Numerical Example - Part II

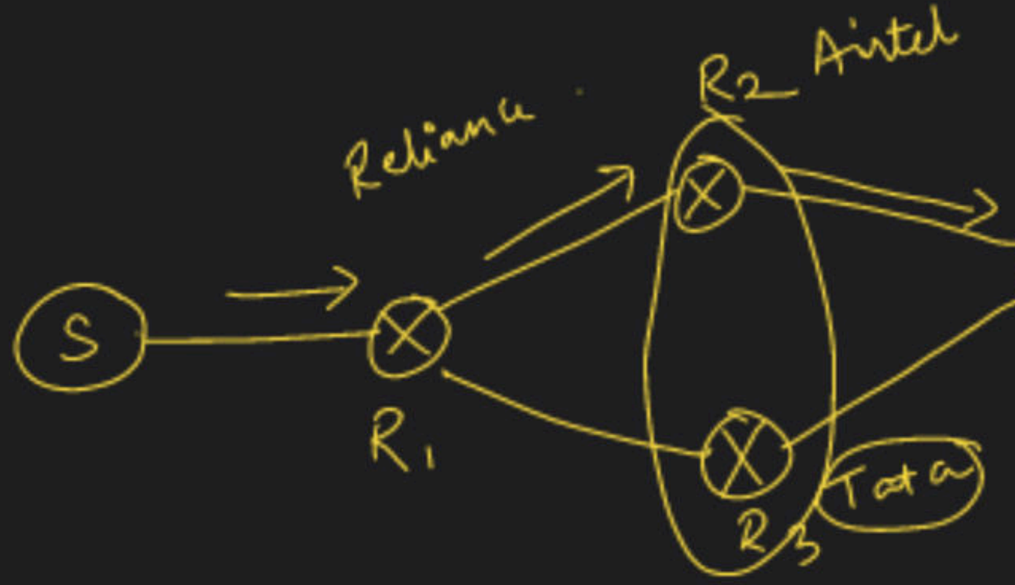
Complete Course on Computer Networks - Part III

Ravindrababu Ravula • Lesson 4 • Mar 26, 2021

26-03-2021 Classes by Ravindrababu Ravula	
Lecture Name	Time
Summary of ER to RDB Conversion & GATE Questions	6:00 AM
Fragmentation with Numerical Example - Part II	7:00 AM
OS Lecture 36 | Paging | Multi Paging | Part 8	8:05 AM
Synchronization Practice Questions Part 2| GATE Practice Questions	12:00PM
Logic Functions Practice Questions Set - 1| GATE Practice Questions	2:00PM
ER and Relational Models Practice Questions | Database Practice Questions | GATE Practice Questions	5:00PM
Asymptotic Notations and Recurrence Relations Practice Questions | GATE Practice Questions	6:00PM

IPV4 OPTIONS FIELD


OPTIONS
SINGLE BYTE
MULTIBYTE
NO OPERATION
END OF OPTION
RECORD ROUTE
STRICT SOURCE ROUTE
LOOSE SOURCE ROUTE
TIMESTAMP
20B
40B
opt
OP2
end
Testing and debugging
ISP
R
T



Strict Source Routing ✓

R_1 R_2 R_4

Loose SR

✓ ✓

R_1 R_4

1) Comp
2) Security

"9"

40 byt

The header of the IPv4 datagram is made of two parts: a fixed part and a variable part. The fixed part is 20 bytes long and was discussed in the previous section. The variable part comprises the options that can be a maximum of 40 bytes. Options, as the name implies, are not required for a datagram. They can be used for network testing and debugging. Although options are not a required part of the IPv4 header, option processing is required of the IPv4 software. This means that all implementations must be able to handle options if they are present in the header.

- No Operation:

A no-operation option is a I-byte option used as a filler between options.

- End of Option

An end-of-option option is a I-byte option used for padding at the end of the option field. It, however, can only be used as the last option.

- Record Route

A record route option is used to record the Internet routers that handle the datagram. It can list up to nine router addresses. It can be used for debugging and management purposes.

- Strict Source Route

A strict source route option is used by the source to predetermine a route for the datagram as it travels through the Internet. Dictation of a route by the source can be useful for several purposes. The sender can choose a route with a specific type of service, such as minimum delay or maximum throughput. Alternatively, it may choose a route that is safer or more reliable for the sender's purpose. For example, a sender can choose a route so that its datagram does not travel through a competitor's network.

If a datagram specifies a strict source route, all the routers defined in the option must be visited by the datagram. A router must not be visited if its IPv4 address is not listed in the datagram. If the datagram visits a router that is not on the list, the datagram is discarded and an error message is issued. If the datagram arrives at the destination and some of the entries were not visited, it will also be discarded and an error message issued.

- Loose Source Route

A loose source route option is similar to the strict source route, but it is less rigid. Each router in the list must be visited, but the datagram can visit other routers as well.

- Timestamp

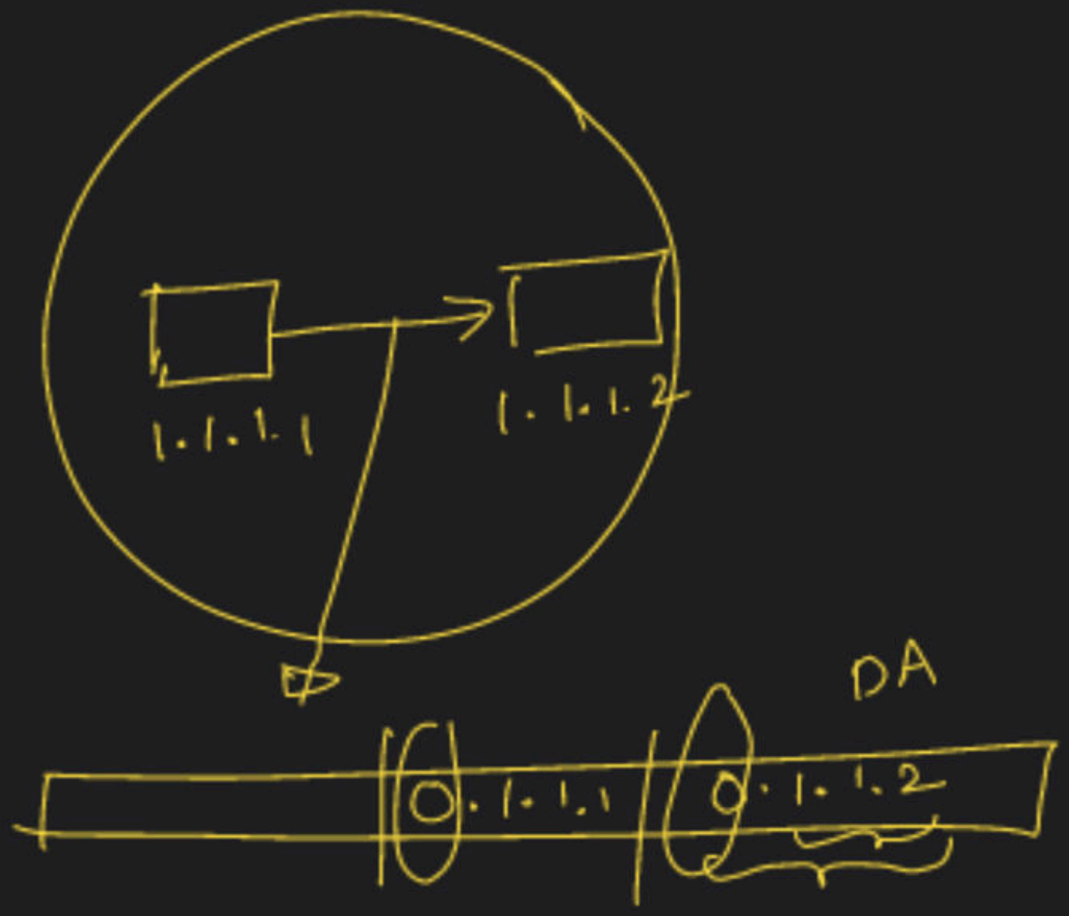
A timestamp option is used to record the time of datagram processing by a router. The time is expressed in milliseconds from midnight, Universal time or Greenwich meantime. Knowing the time a datagram is processed can help users and managers track the behavior of the routers in the Internet. We can estimate the time it takes for a datagram to go from one ~outer to another. We say estimate because, although all routers may use Universal time, their local clocks may not be synchronized.

NID	HID	
✔ valid	✔ valid	VIP 1.1.1.2
✔	0'S	NID 10.0.0.0
✔	1'S	DBA 10.255.255.255
1'S	1'S	LBA 255.255.255.255
1'S	0'S	NM OR SM
0'S	✔	HOST WITHIN A N/W
0'S	0'S	I DON'T HAVE IP
127 ✓	✔	LOOP BACK ADDRESS

127

1-126

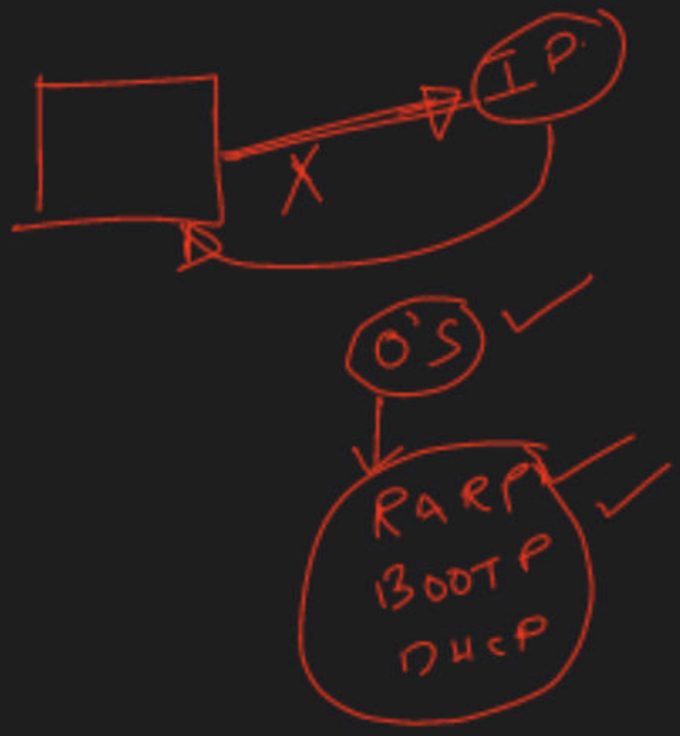
128



IP -> S/w number

↓

Ram

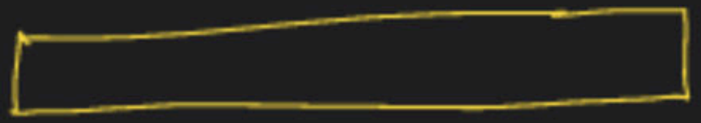


Example: Segmentation and Fragmentation

AL

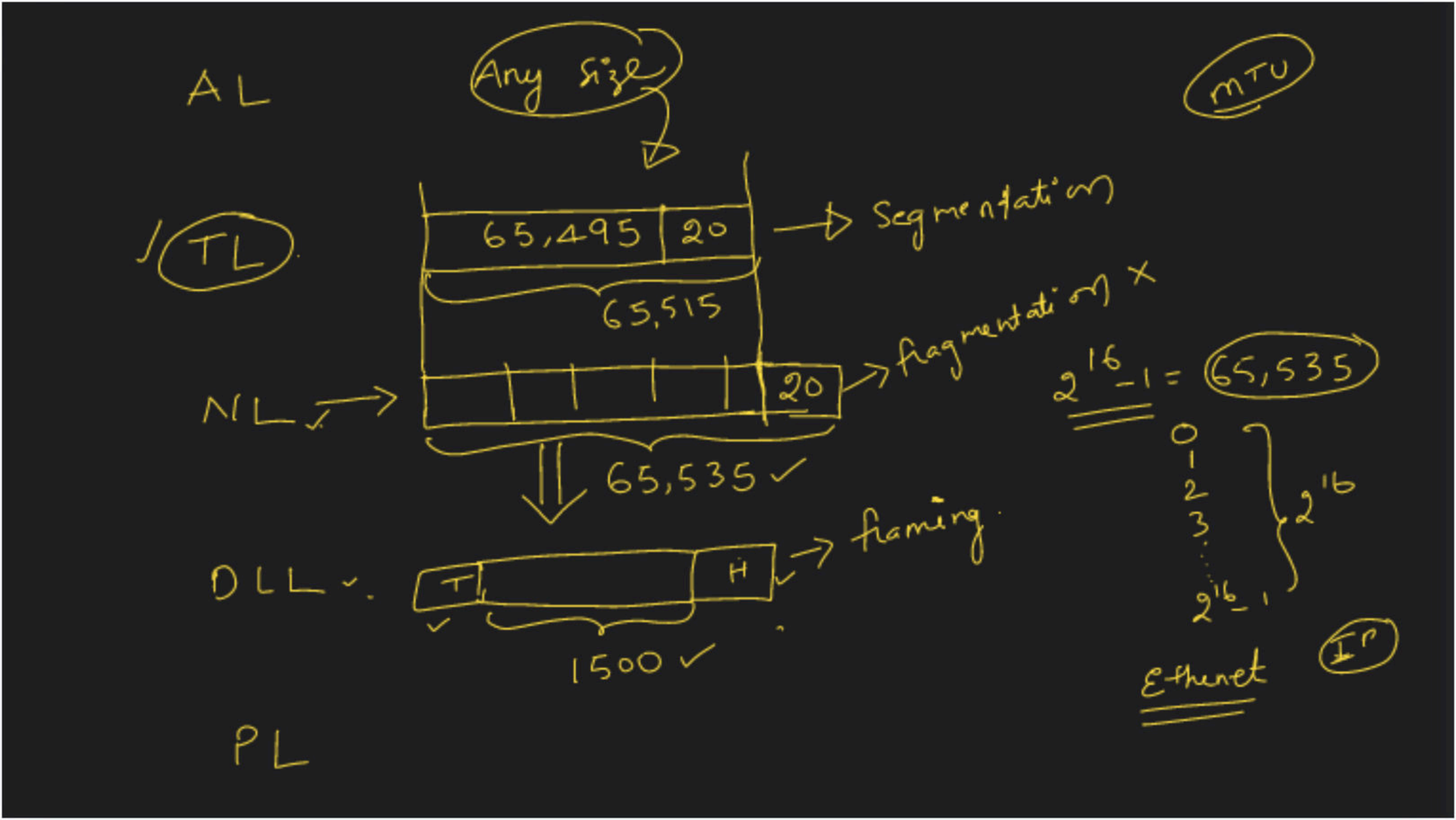
TL

NL



DLL

PL


AL
Any Size
MTU
TL
65,495
20
Segmentation
65,515
NIL
20
fragmentation ×
$2^{16}-1$ = 65,535
65,535
0
1
2
3
2^{16}
$2^{16}-1$
DLL
T
H
framing
1500
Ethernet
IP
PL

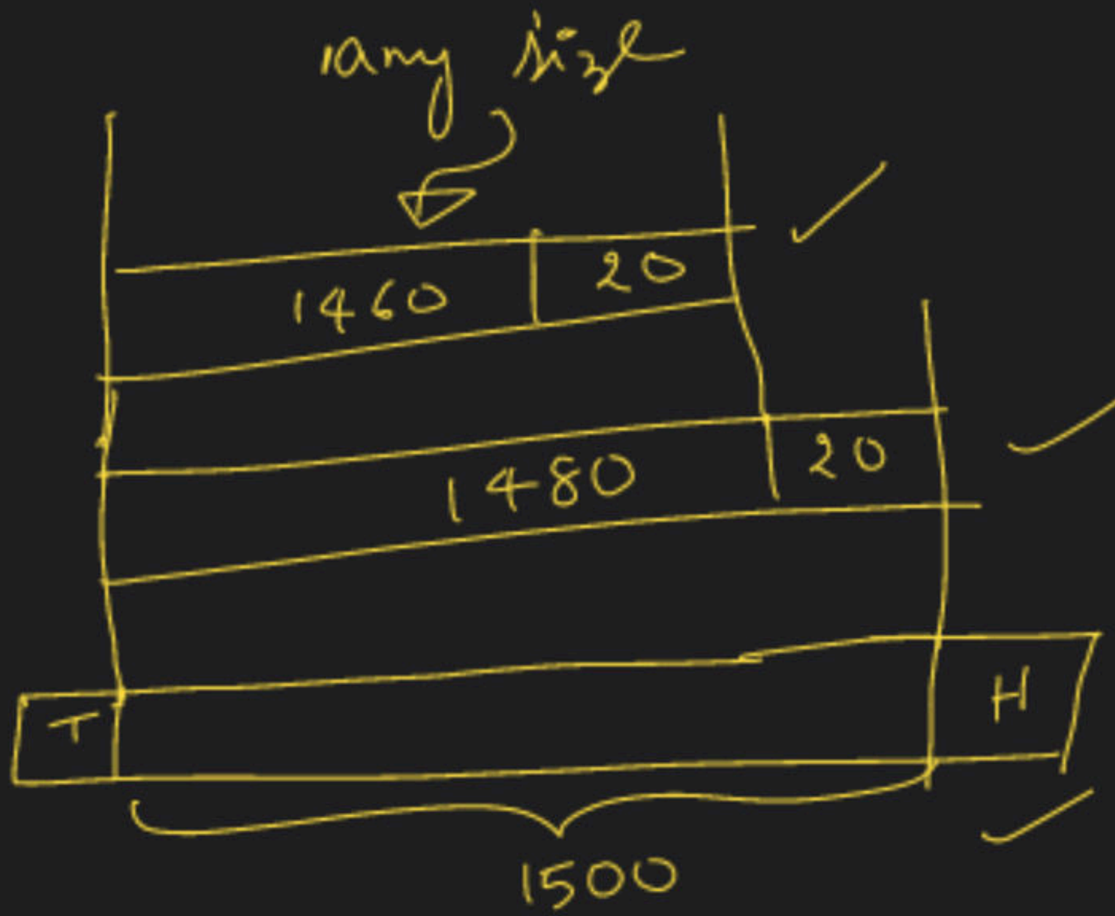
AL

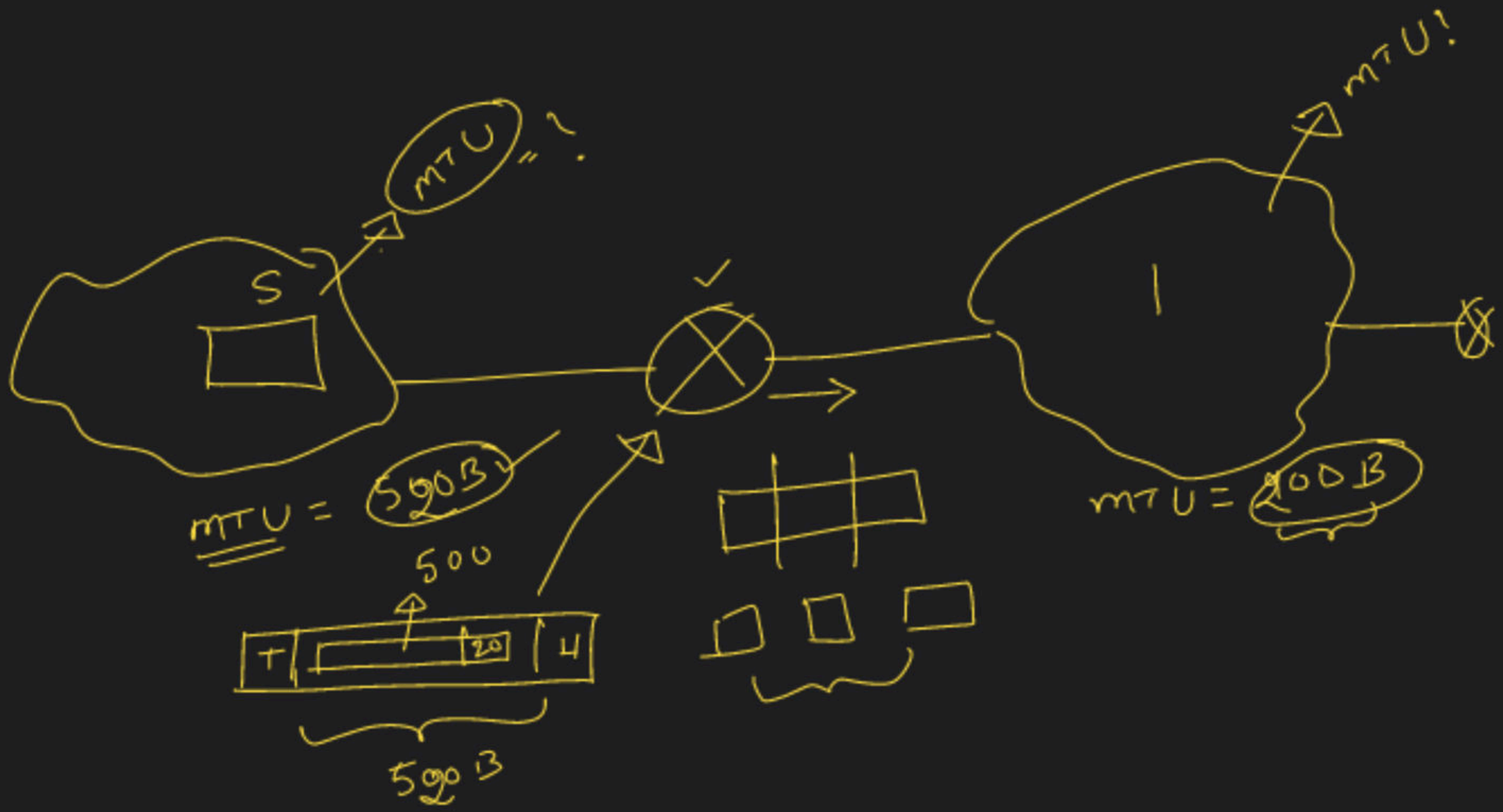
TL

NL

DLL

PL




MTU = ?
S
MTU = 590B
500
T
20
4
590B
MTU!
1
MTU = 200B

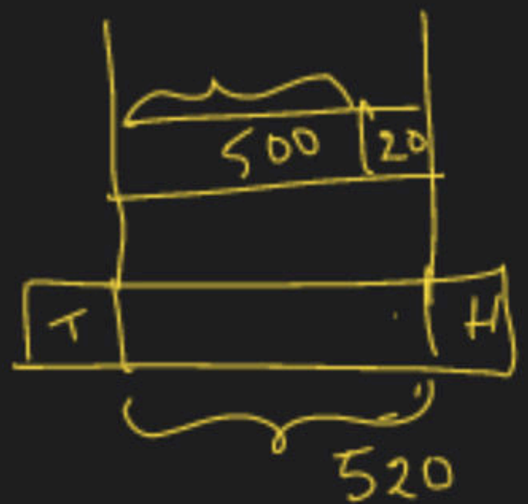
AL

TL

NL

DLL

PL



MTU → Data

520

H → 20.

D → 500

SEGMENTATION AND FRAGMENTATION

This occurs during the original creation of the packets when a set of data doesn't fit within the "Maximum Segment Size (MSS)".
The data is then divided into multiple segments referred as "Protocol Data Unit". This process is known as **Segmentation.**

In order to avoid Fragmentation (which we will see further) , note that
(Number of bytes in the data segment + the header) < MTU

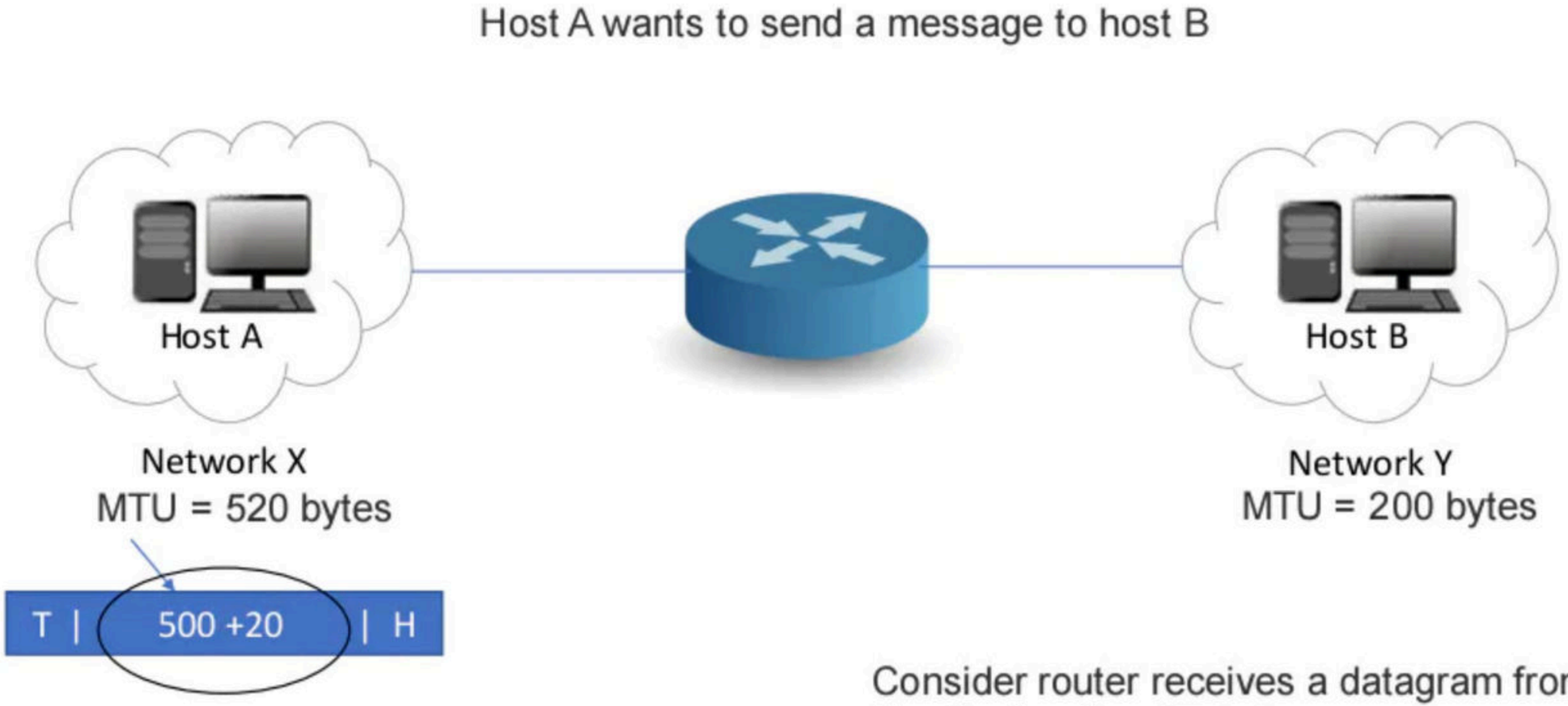
Fragmentation occurs during the original creation of frames where the network layer must send packets down to the Data Link Layer for transmission. Some Data Link Layer technologies have limits on the length of the data that can be sent. In short some links have smaller MTU (Maximum Transmission Unit).
If the packet that is to be sent is larger than the MTU then it is divided into pieces.
This process is known as fragmentation.
These pieces are reassembled once they arrive at the network layer of the destination.
As mentioned earlier Fragmentation can be avoided if,
(Number of bytes in the data segment + the header) < MTU

Computer Networks

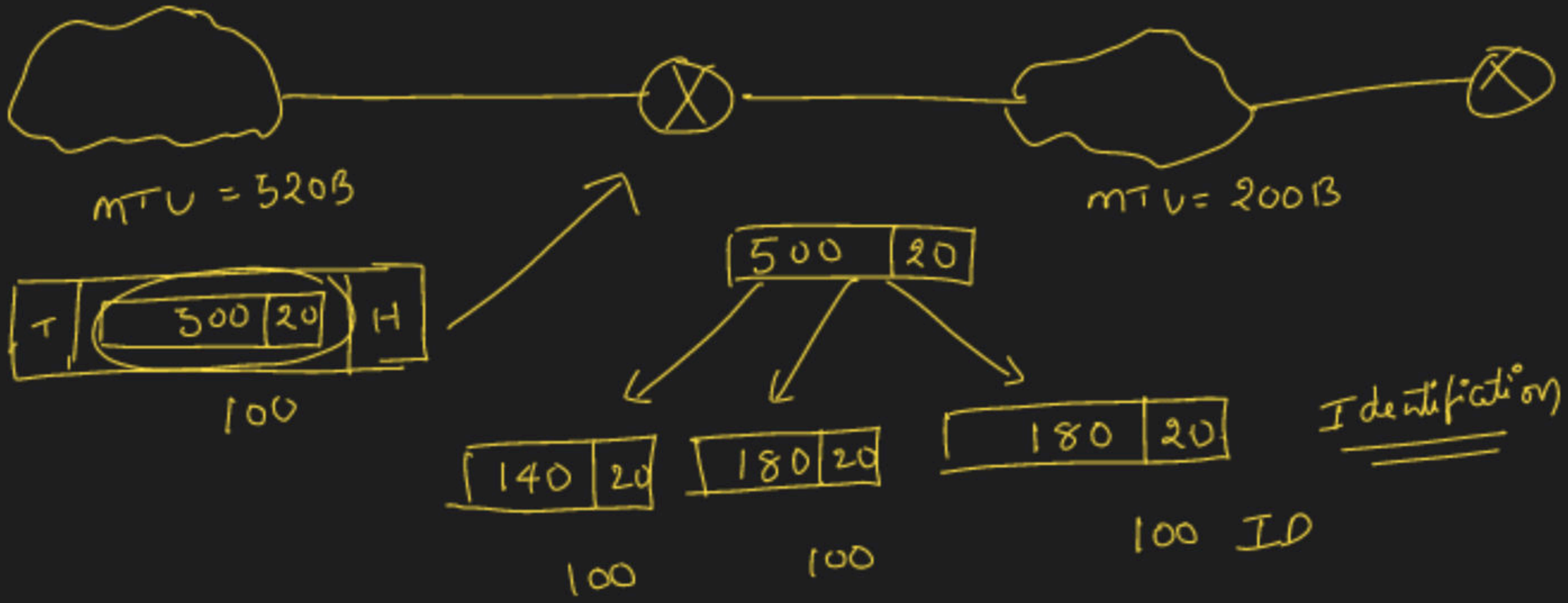
Fragmentation

Lets us discuss some examples of IP fragmentation to understand how the fragmentation is actually carried out.

EXAMPLE 1

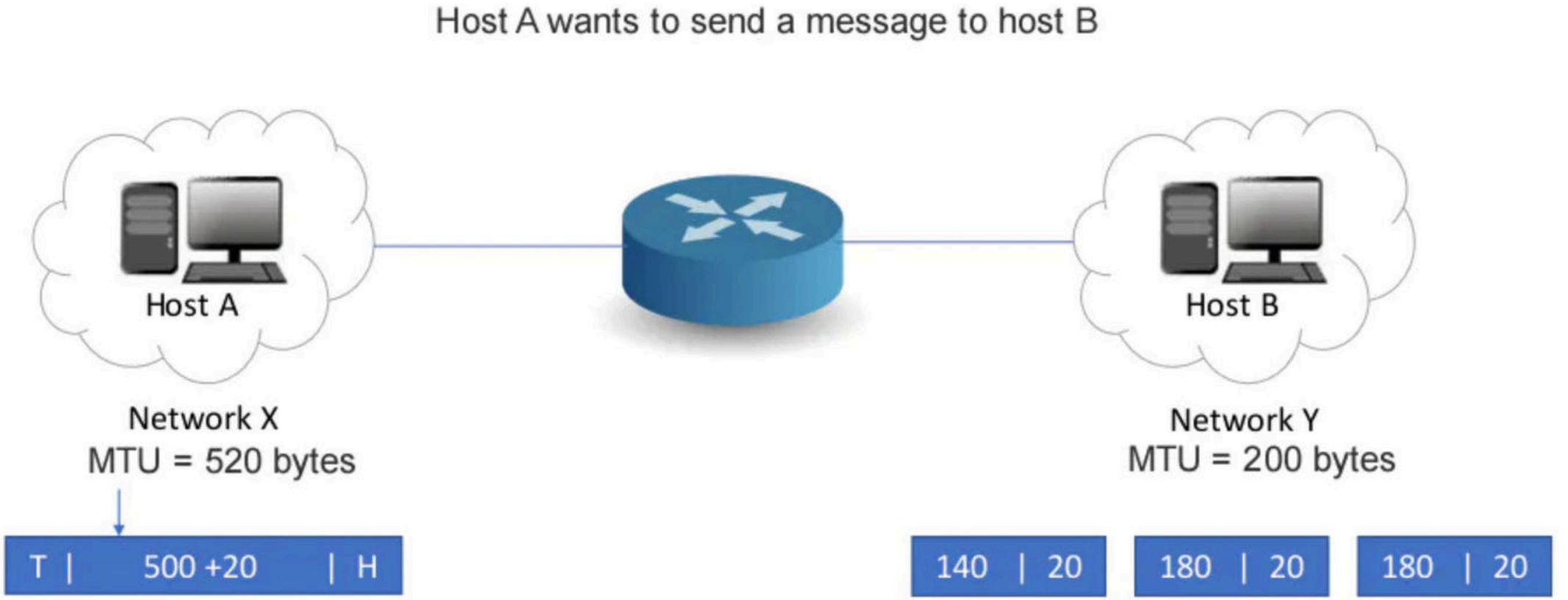


Consider router receives a datagram from host A having-
Header length = 20 bytes
Payload length = 500 bytes
Total length = 520 bytes


MTU = 520B
MTU = 200B
T
300
20
H
100
500
20
140
20
180
20
180
20
100
100
100 ID
Identification

Lets us discuss some examples of IP fragmentation to understand how the fragmentation is actually carried out.

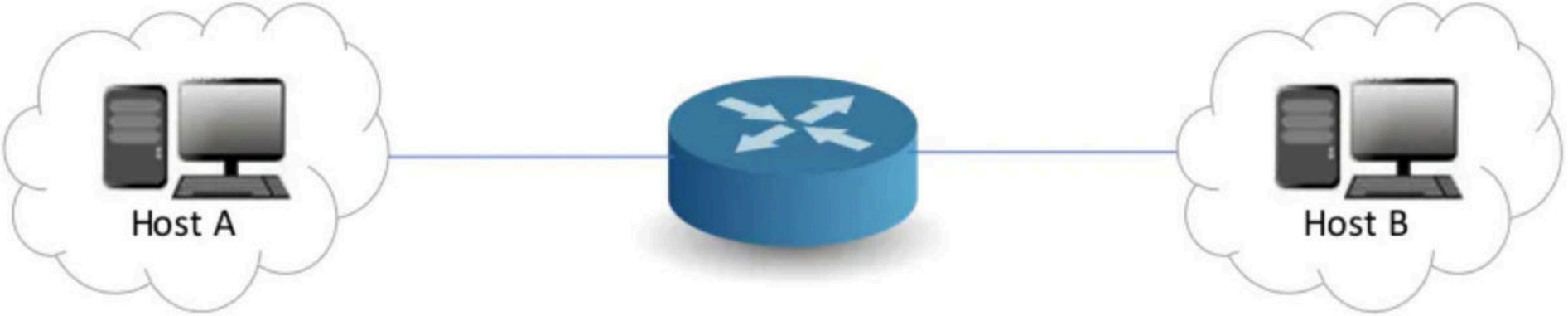
EXAMPLE 1



The amount of data sent in one fragment is chosen such that-
It is as large as possible but less than or equal to MTU.
It is a multiple of 8 so that pure decimal value can be obtained for the fragment offset field.

NOTE

- It is not compulsory for the last fragment to contain the amount of data that is a multiple of 8.
- This is because it does not have to decide the fragment offset value for any other fragment.



Network X
MTU = 520 bytes

500 +20

Network Y
MTU = 200 bytes

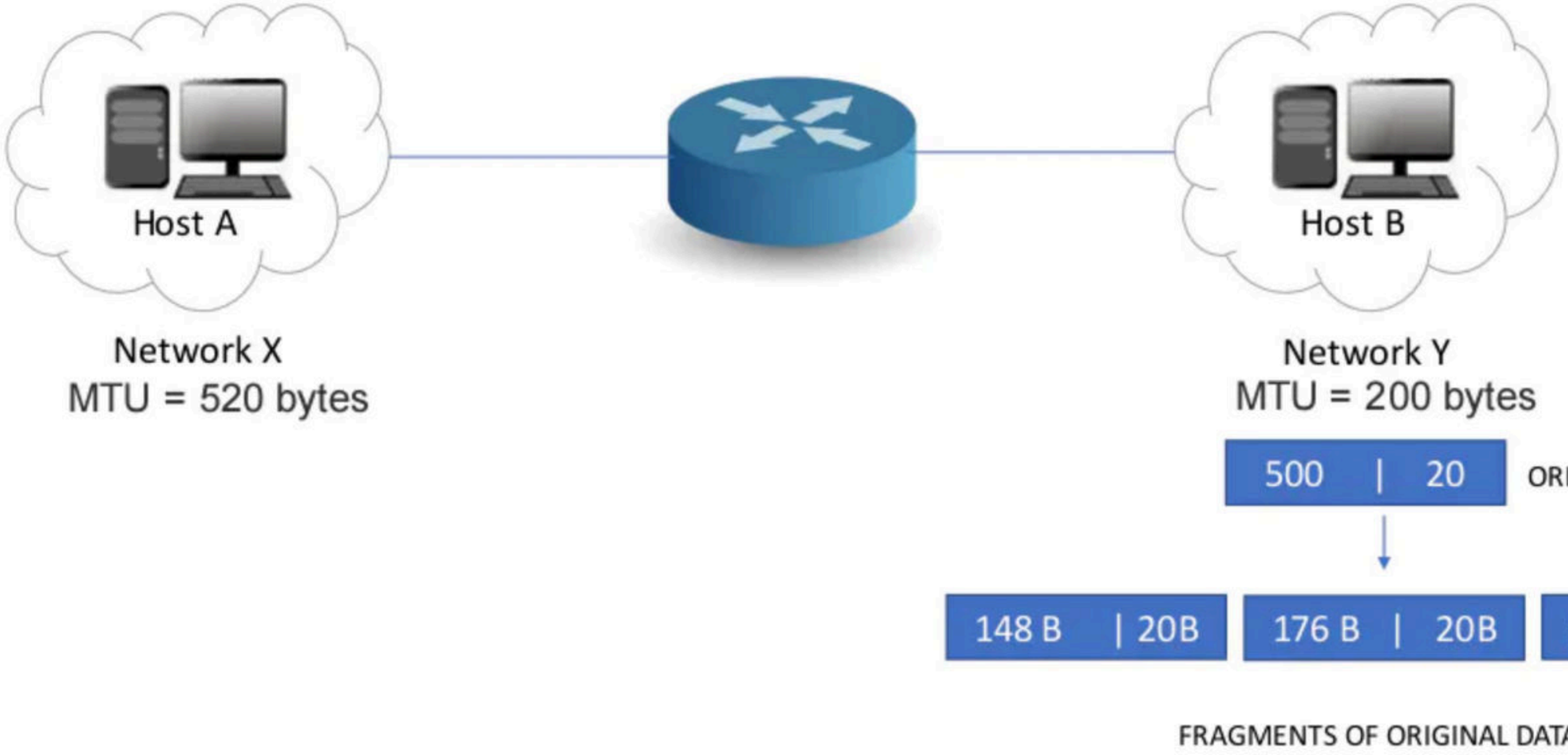
Following the above rule,
Router decides to send maximum 176 bytes of data in one fragment.
This is because it is the greatest value that is a multiple of 8 and less than MTU.

Router creates three fragments of the original datagram where-

First fragment contains the data = 176 bytes

Second fragment contains the data = 176 byes

Third fragment contains the data = 148 bytes



The information contained in the IP header of each fragment is-

Header Information Of 1st Fragment-

- Header length field value = 20 / 4 = 5
- Total length field value = 176 + 20 = 196
- MF bit = 1
- Fragment offset field value = 0
- Header checksum is recalculated.
- Identification number is same as that of original datagram.

Header Information Of 2nd Fragment-

- Header length field value = 20 / 4 = 5
- Total length field value = 176 + 20 = 196
- MF bit = 1
- Fragment offset field value = 176 / 8 = 22
- Header checksum is recalculated.
- Identification number is same as that of original datagram.

Header Information Of 3rd Fragment-

- Header length field value = 20 / 4 = 5
- Total length field value = 148 + 20 = 168
- MF bit = 0
- Fragment offset field value = (176 + 176) / 8 = 44
- Header checksum is recalculated.
- Identification number is same as that of original datagram.

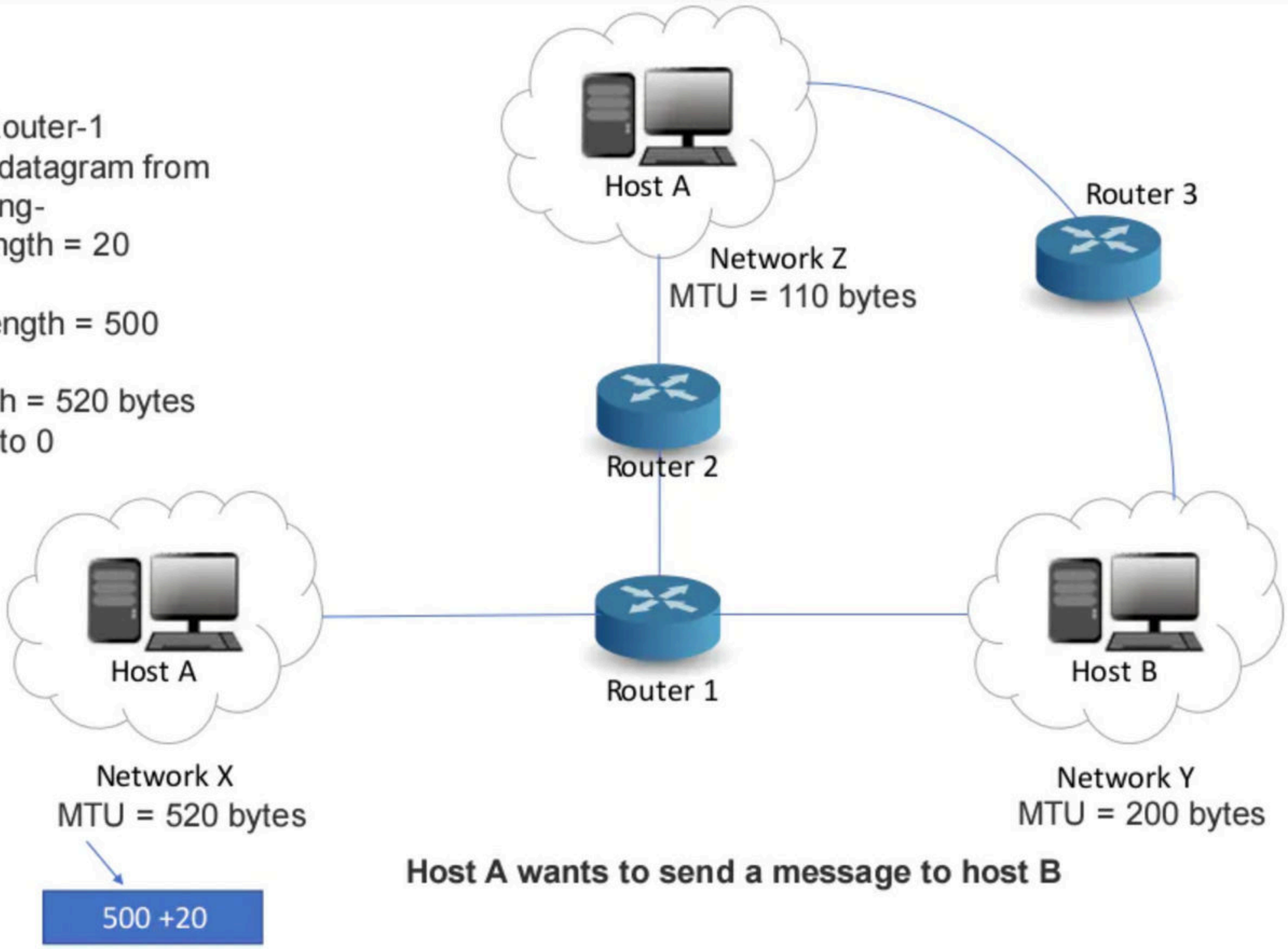
Router transmits all the fragments.

At destination side,
- Receiver receives 3 fragments of the datagram.
- Reassembly algorithm is applied to combine all the fragments to obtain the original datagram.

EXAMPLE 2

Consider Router-1 receives a datagram from host A having-

- Header length = 20 bytes
- Payload length = 500 bytes
- Total length = 520 bytes
- DF bit set to 0

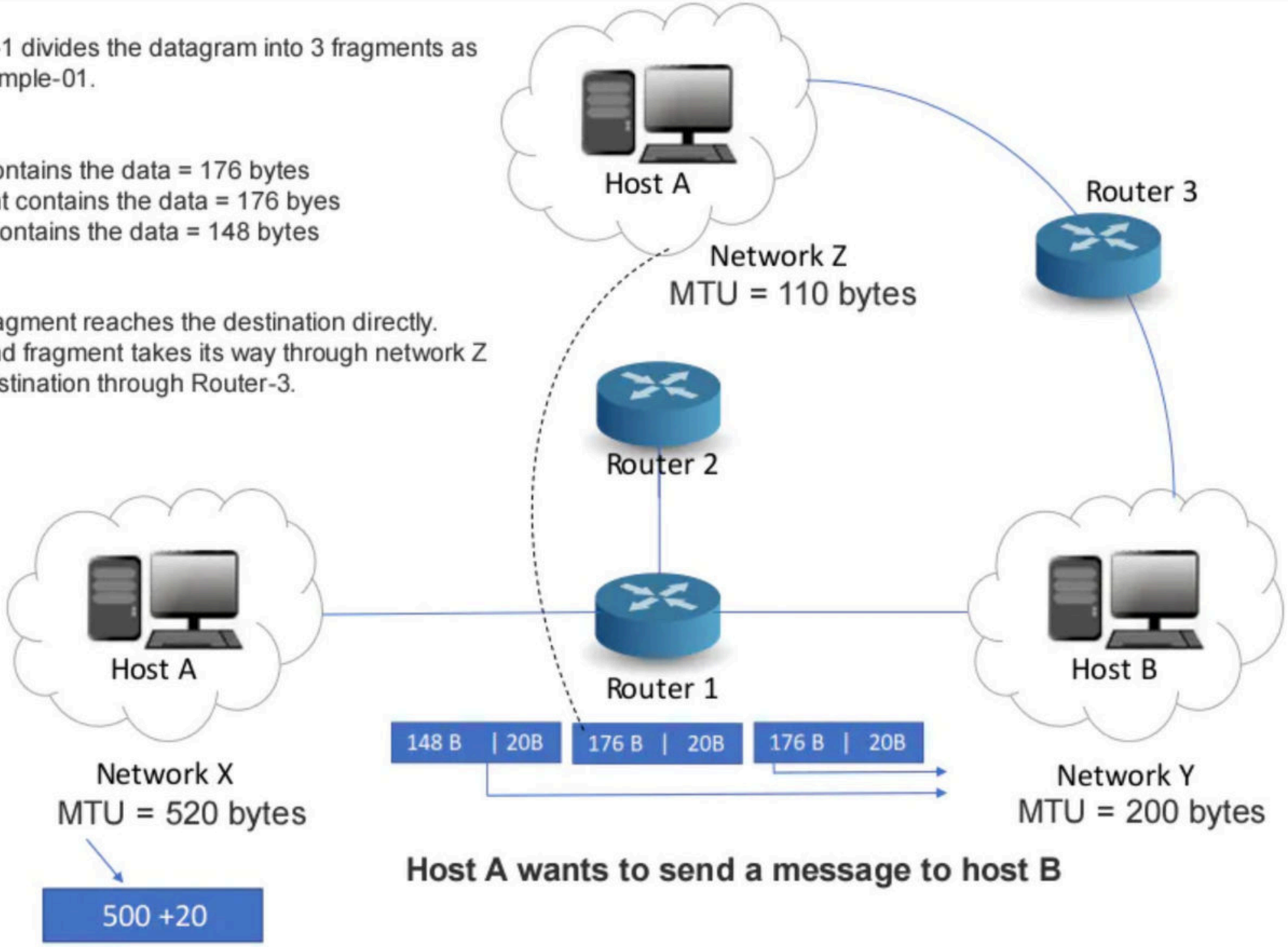


Host A wants to send a message to host B

Consider Router-1 divides the datagram into 3 fragments as discussed in Example-01.

Then,
- First fragment contains the data = 176 bytes
- Second fragment contains the data = 176 byes
- Third fragment contains the data = 148 bytes

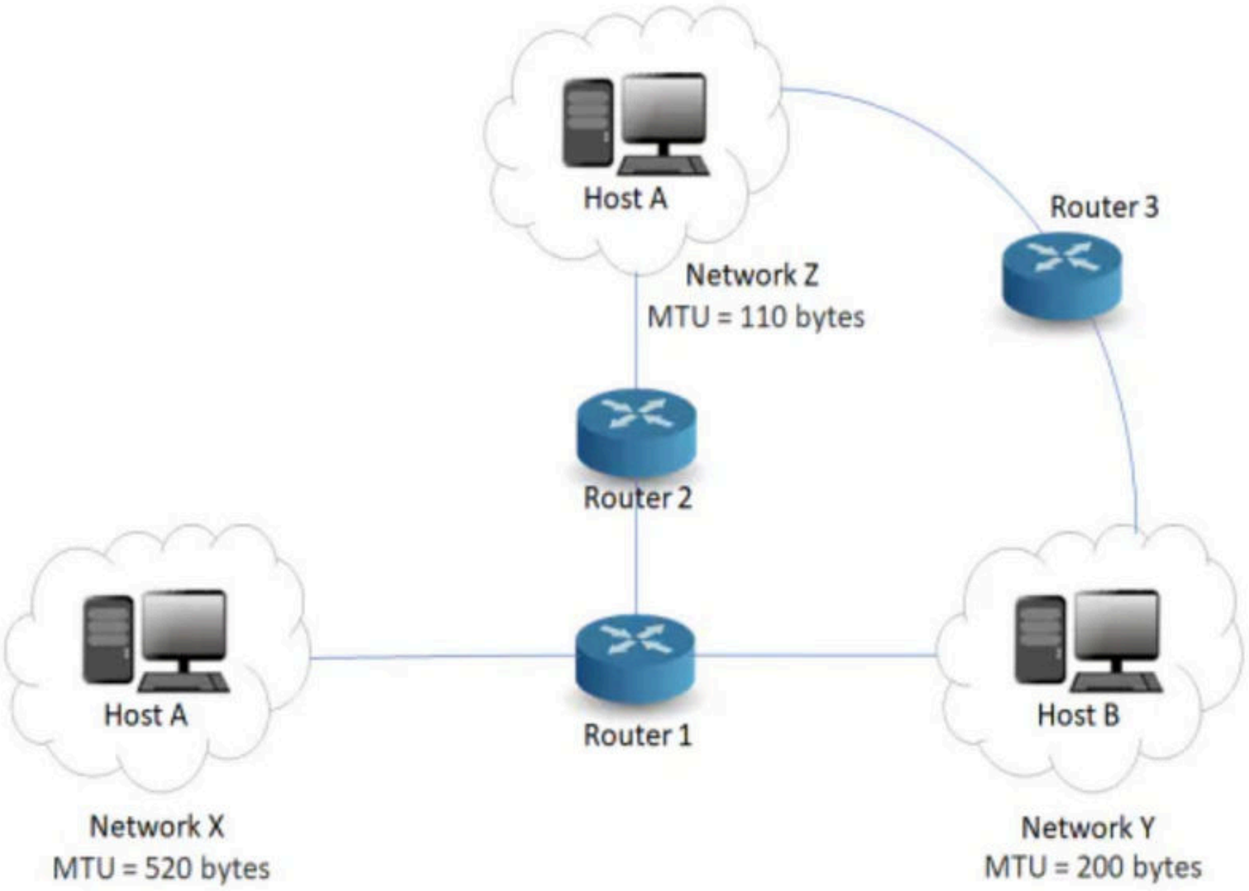
Now, consider-
- First and third fragment reaches the destination directly.
- However, second fragment takes its way through network Z and reach the destination through Router-3.



Now, let us discuss the journey of fragment-2 and how it finally reaches the destination.

Router-2 receives a datagram (second fragment of original datagram) where-

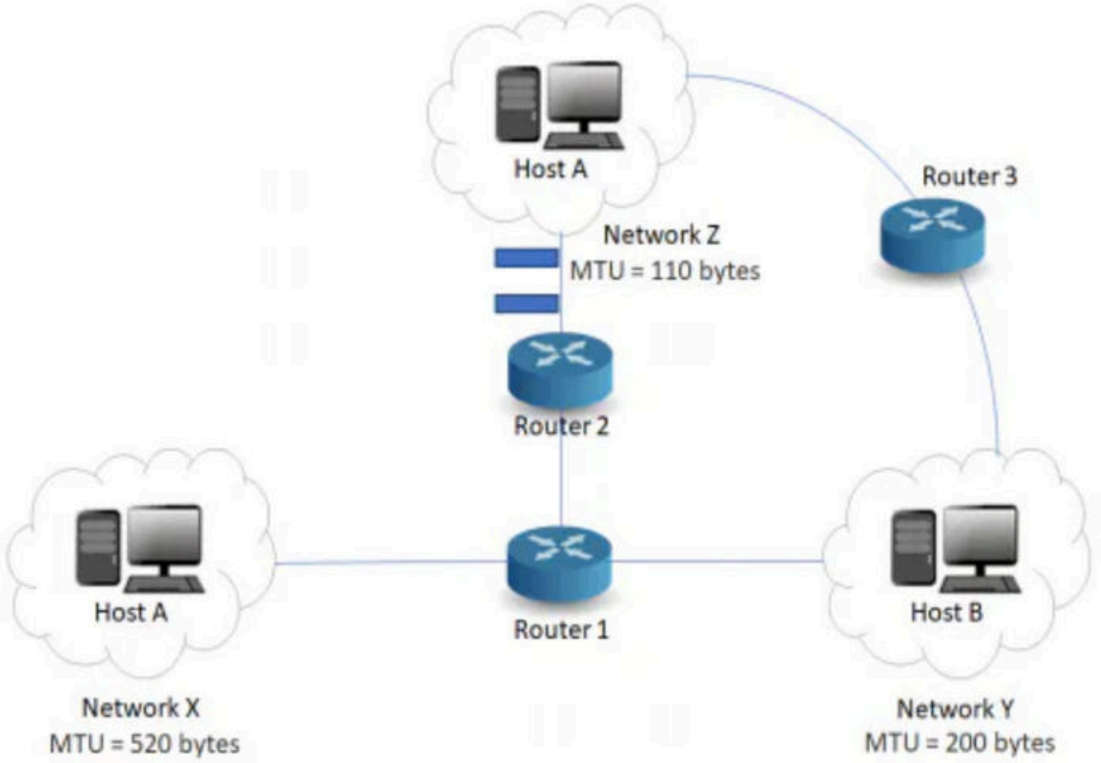
- Header length = 20 bytes
- Payload length = 176 bytes
- Total length = 196 bytes
- DF bit set to 0



Step-01:

Router-2 examines the datagram and finds-
•Size of the datagram = 196 bytes
•Destination is network Z having MTU = 110 bytes
•DF bit is set to 0

Router-2 concludes-
•Size of the datagram is greater than MTU.
•So, it will have to divide the datagram into fragments.
•DF bit is set to 0.
•So, it is allowed to create fragments of the datagram.



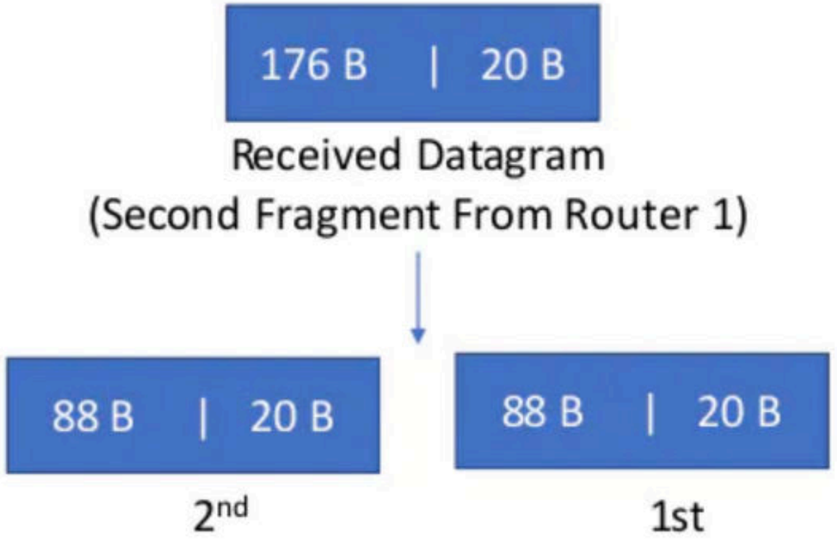
Step-02:

Router-2 decides the amount of data that it should transmit in each fragment.

Router-2 knows-
•MTU of the destination network = 110 bytes.
•So, maximum total length of any fragment can be only 110 bytes.
•Out of 110 bytes, 20 bytes will be taken by the header.
•So, maximum amount of data that can be sent in any fragment = 90 bytes.

Following the rule,
•Router-2 decides to send maximum 88 bytes of data in one fragment.
•This is because it is the greatest value that is a multiple of 8 and less than MTU.

Router-2 creates two fragments of the received datagram where-



The information contained in the IP header of each fragment is-

Header Information Of 1st Fragment-

- Header length field value = 20 / 4 = 5
- Total length field value = 88 + 20 = 108
- MF bit = 1
- Fragment offset field value = 176 / 8 = 22
- Header checksum is recalculated.
- Identification number is same as that of original datagram.

NOTE-

- This fragment is **NOT** the first fragment of the original datagram.
- It is the first fragment of the datagram received by Router-2.
- The datagram received by Router-2 is the second fragment of the origina datagram.
- This datagram will serve as the second fragment of the original datagram
- Therefore, fragment offset field is set according to the first fragment of th original datagram.

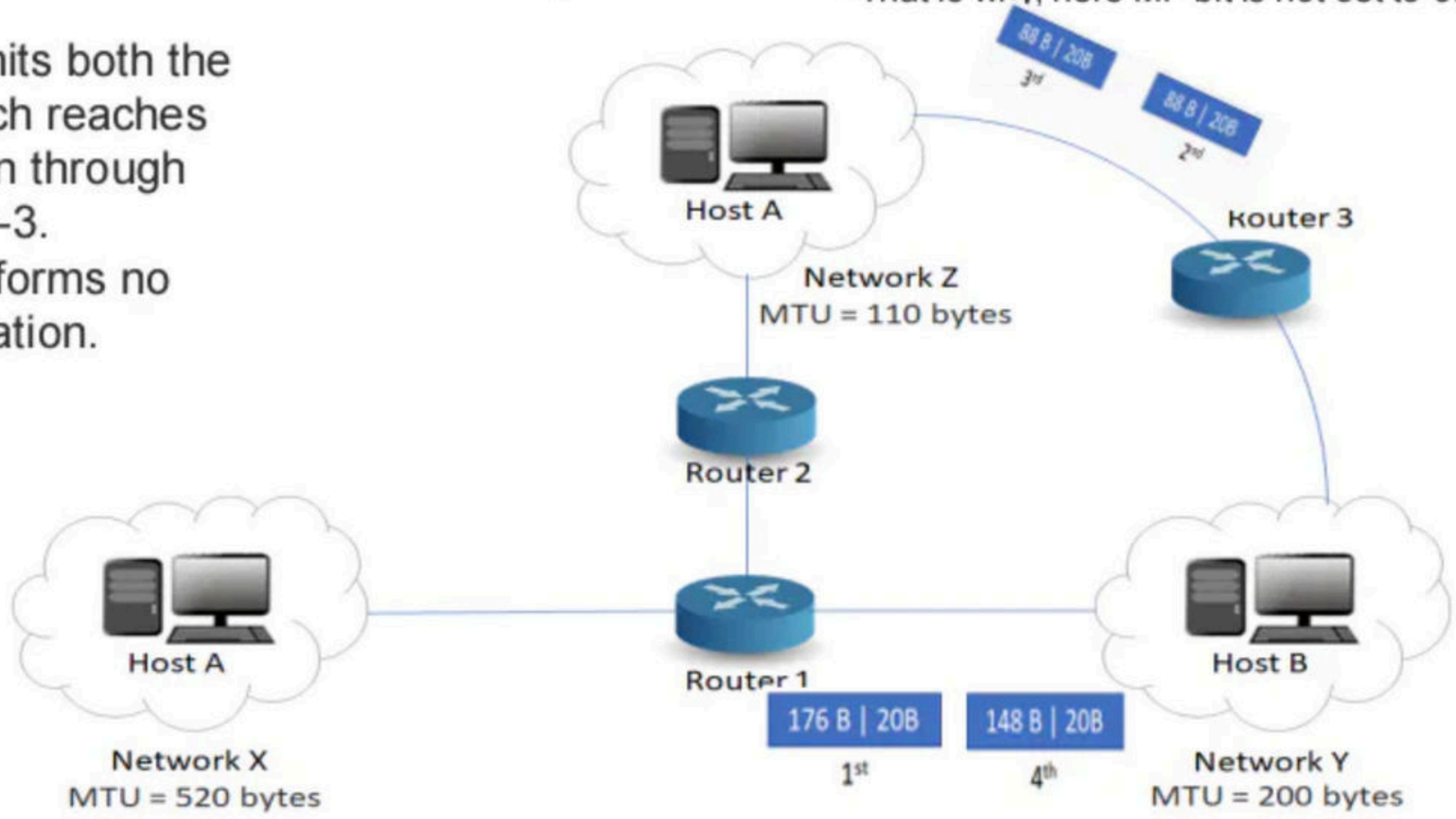
Header Information Of 2nd Fragment-

- Header length field value = 20 / 4 = 5
- Total length field value = 88 + 20 = 108
- MF bit = 1
- Fragment offset field value = (176 + 88) / 8 = 33
- Header checksum is recalculated.
- Identification number is same as that of original datagram.

Router-2 transmits both the fragments which reaches the destination through Router-3.
Router-3 performs no fragmentation.

NOTE-

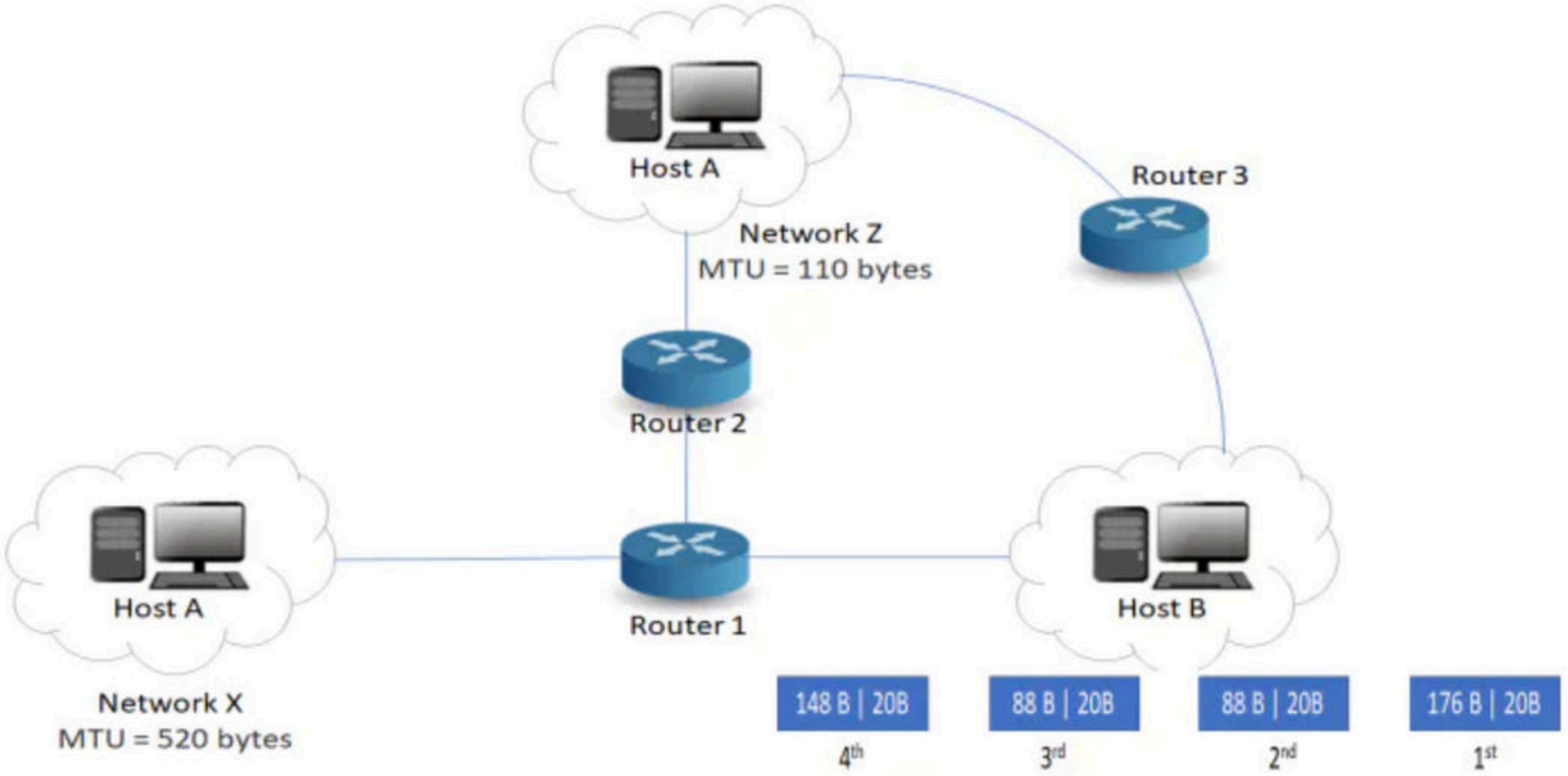
- This fragment is **NOT** the last fragment of the original datagram.
- It is the last fragment of the datagram received by Router-2.
- The datagram received by Router-2 is the second fragment of the original datagram.
- This datagram will serve as the third fragment of the original datagram.
- There is another fragment of the original datagram that follows it.
- That is why, here MF bit is not set to 0.



At destination side,

•Receiver receives 4 fragments of the datagram.

•Reassembly algorithm is applied to combine all the fragments to obtain the original datagram.



Fragment Offset field value for the next subsequent fragment
= (Payload length of the current fragment / 8) + Offset field value of the current fragment
= (Total length – Header length / 8) + Offset field value of the current fragment

Fragmentation Overhead

- Fragmentation of datagram increases the overhead.
- This is because after fragmentation, IP header has to be attached with each fragment.

Total Overhead
= (Total number of fragmented datagrams – 1) x size of IP header

Efficiency = Useful bytes transferred / Total bytes transferred
OR
Efficiency = Data without header / Data with header

Bandwidth Utilization or Throughput = Efficiency x Bandwidth

NOTE:

MF bit	Offset value	Represents
1	0	1st Fragment
1	!=0	Intermediate Fragment
0	!=0	Last Fragment
0	0	No Fragmentation

Reassembly is not done at the routers because-

All the fragments may not meet at the router.

Fragmented datagrams may reach the destination through independent paths.

There may be a need for further fragmentation.

Computer Networks

Reassembly Algorithm

Reassembly Algorithm

Receiver applies the following steps for reassembly of all the fragments-

1.It identifies whether datagram is fragmented or not using MF bit and Fragment offset field.

2.It identifies all the fragments belonging to the same datagram using identification field.

3.It identifies the first fragment. Fragment with offset field value = 0 is the first fragment.

4.It identifies the subsequent fragments using total length, header length and fragment offset.

5.It repeats step-04 until MF bit = 0.