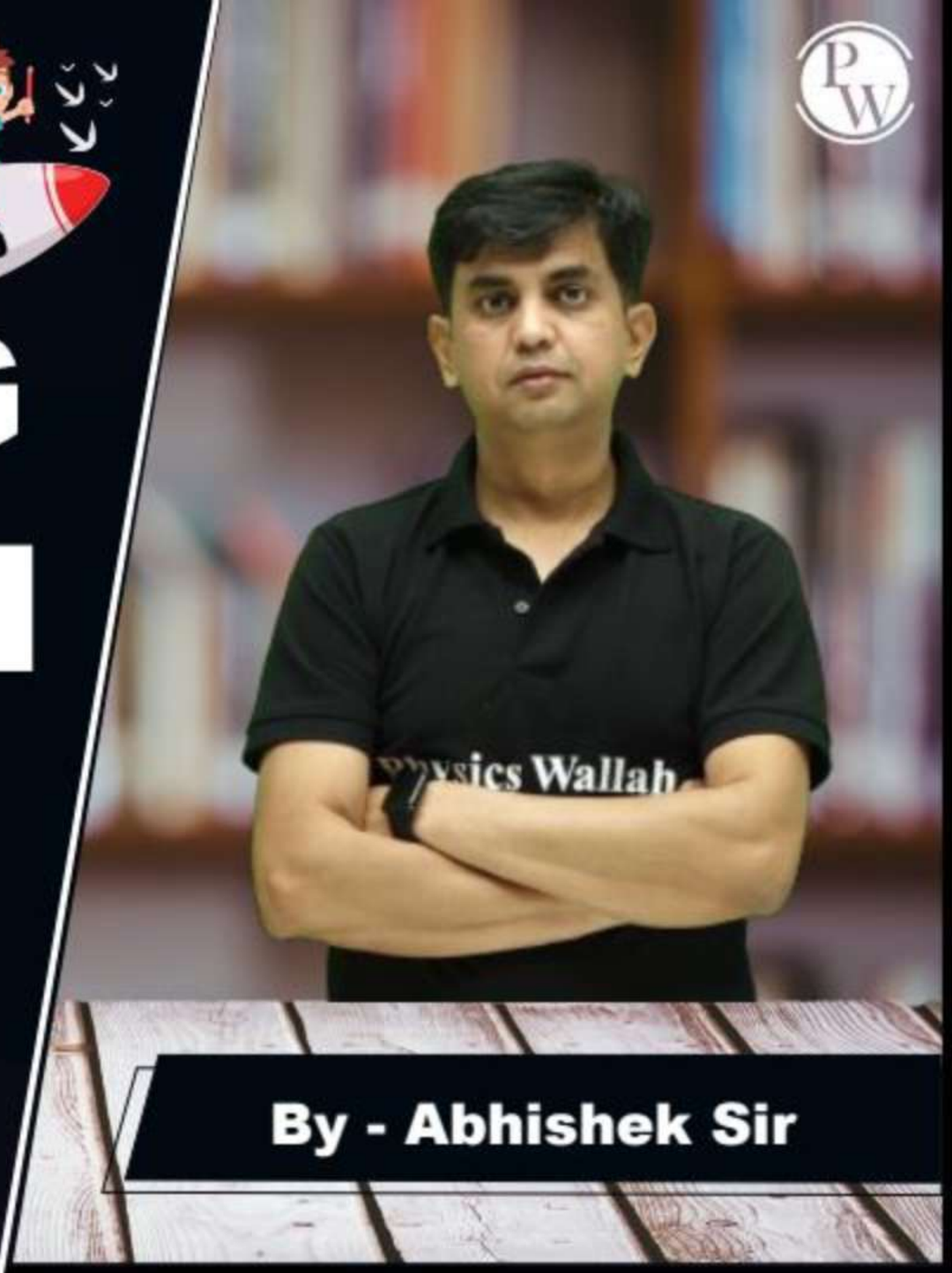
CS & IT ENGINEERING

Computer Network

IPv4 Header

Lecture No. - 06



Recap of Previous Lecture

Topic Fragmentation Offset

Topic Flag bits

Topic Time-to-Live

Topics to be Covered

Topic Header Checksum

Topic IPv4 Options

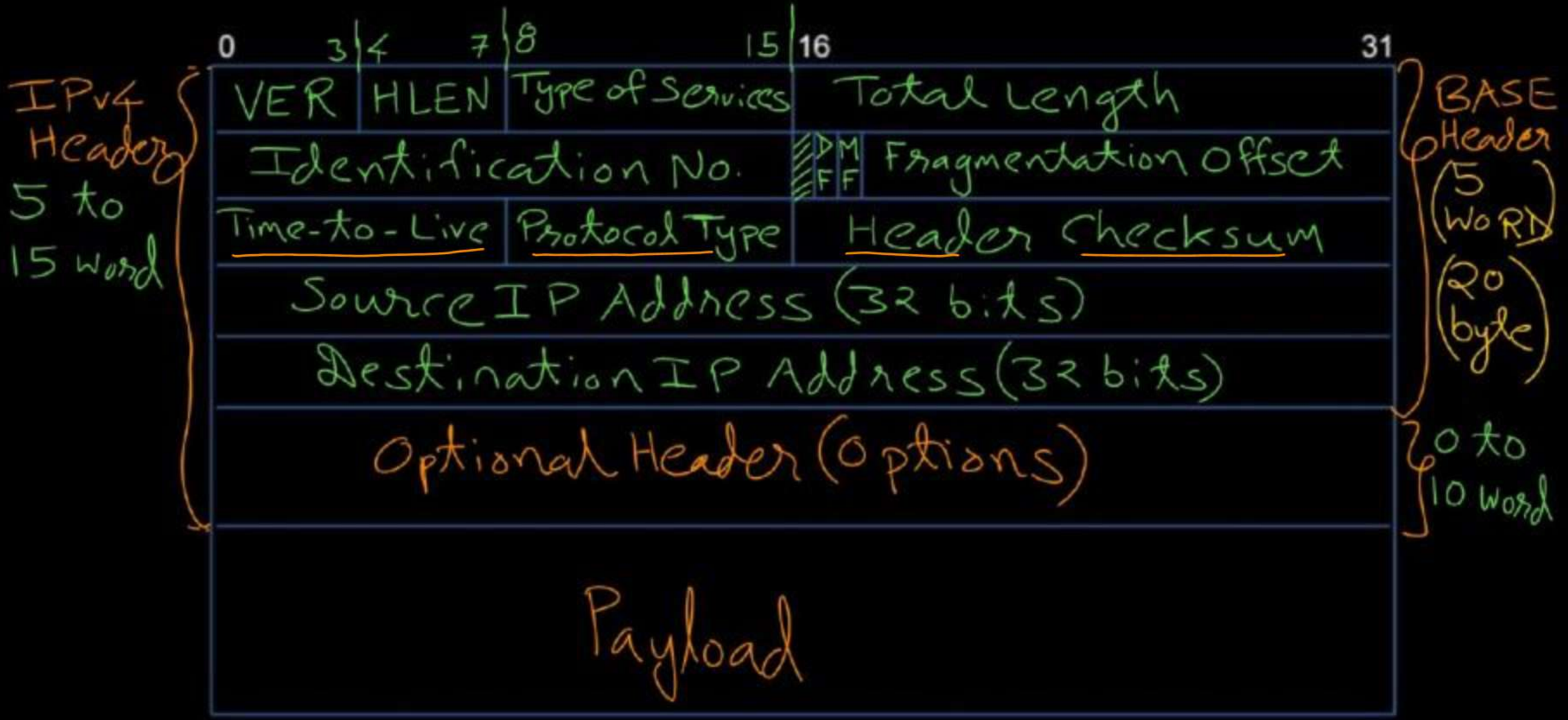
ABOUT ME

Hello, I'm Abhishek

- GATE CS AIR - 96
- M.Tech (CS) - IIT Kharagpur
- 12 years of GATE CS teaching experience

Telegram Link : https://t.me/abhisheksirCS_PW

Topic : IPv4 Packet Header



#Q. Consider an IP packet with a length of 4,500 bytes that includes a 20-byte IPv4 header and 40-byte TCP header. The packet is forwarded to an IPv4 router that supports a Maximum Transmission Unit (MTU) of 600 bytes. Assume that the length of the IP header in all the outgoing fragments of this packet is 20 bytes. Assume that the fragmentation offset value stored in the first fragment is 0. The fragmentation offset value stored in the third fragment is _____ . [GATE 2018]

TL = 4500 byte
Header Size = 20 byte

MTU Size = 600 byte
Header Size = 20 byte
New Payload Size = [MTU - Header Size]
= (600 - 20) = 580 byte
≈ 576 byte

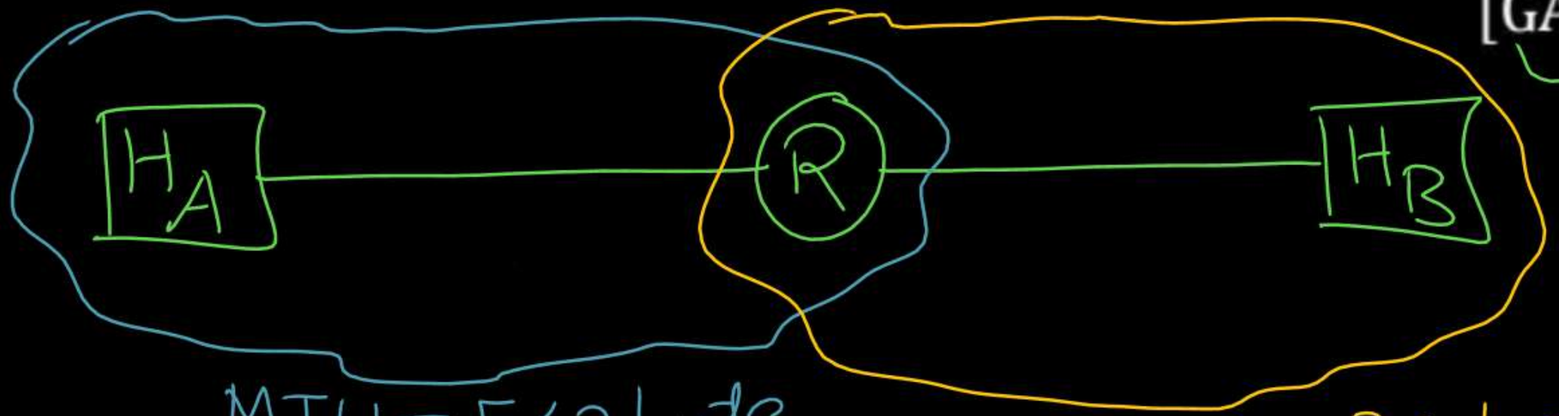
old offset = 0

offset value for 3rd fragment = old offset + $\left(\frac{2 * 576 \text{ byte}}{8}\right)$

Ans = 144

#Q. Consider sending an IP datagram of size 1420 bytes (including 20 bytes of IP header) from a sender to a receiver over a path of two links with a router between them. The first link (sender to router) has an MTU (Maximum Transmission Unit) size of 542 bytes, while the second link (router to receiver) has an MTU size of 360 bytes. The number of fragments that would be delivered at the receiver is _____ .

[GATE 2024, Set-1, 2-Mark]



Header size = 20 byte

MTU = 542 byte

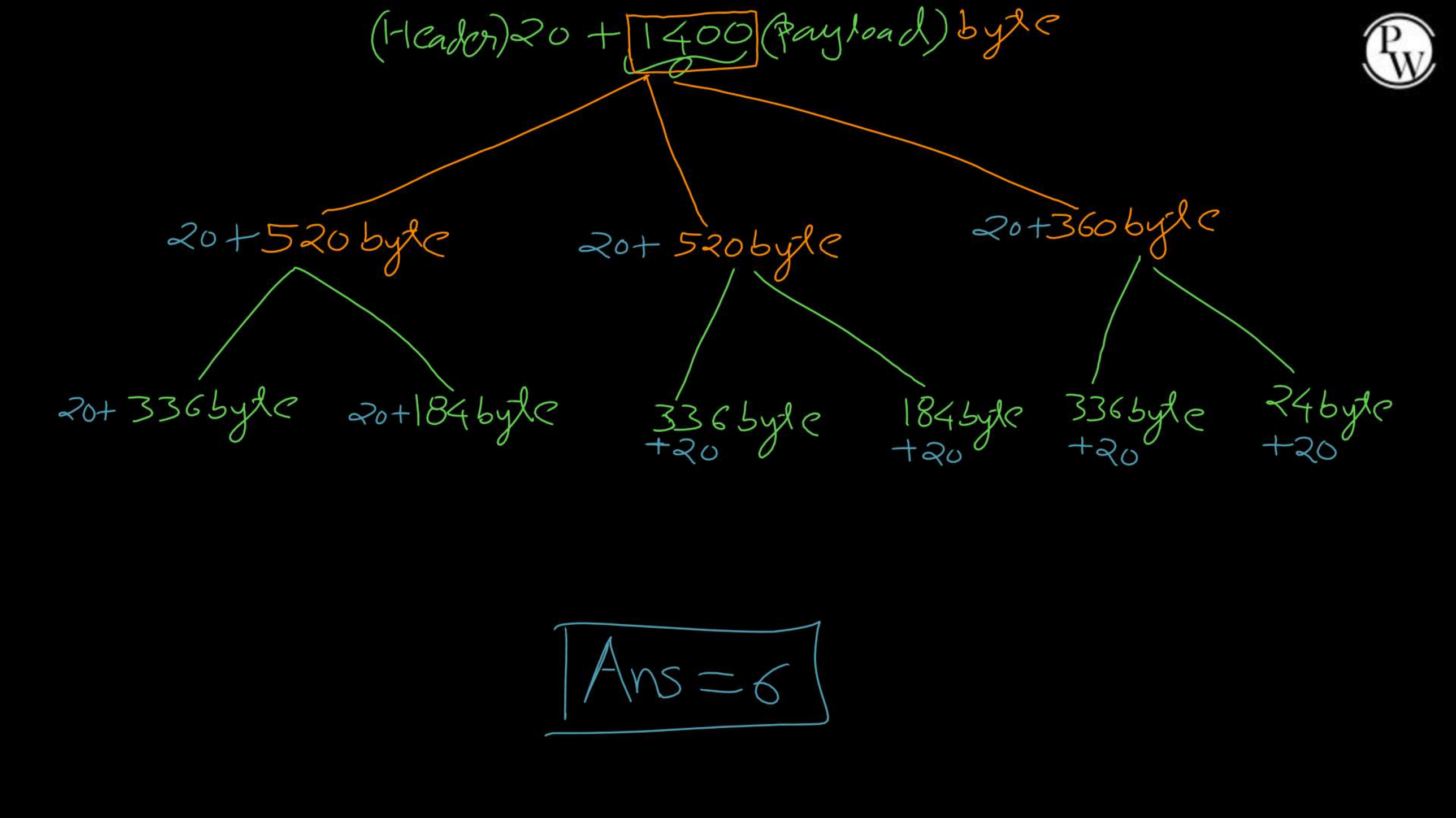
Payload Size = (542 - 20) byte

= 522 byte ≈ 520 byte

MTU = 360 byte

Payload Size = (360 - 20) byte

= 340 byte ≈ 336 byte


(Header) 20 + 1400 (Payload) byte
20+520 byte
20+520 byte
20+360 byte
20+336 byte
20+184 byte
336 byte +20
184 byte +20
336 byte +20
24 byte +20

Ans = 6

#Q. Consider two hosts P and Q connected through a router R. The maximum transfer unit (MTU) value of the link between P and R is 1500 bytes, and between R and Q is 820 bytes. A TCP segment of size 1400 bytes was transferred from P to Q through R, with IP identification value as 0x1234. Assume that the IP header size is 20 bytes. Further, the packet is allowed to be fragmented, i.e., Don't Fragment (DF) flag in the IP header is not set by P. Which of the following statements is/are correct?

[GATE 2021, Set-1, 2-Mark]

(A) Two fragments are created at R and the IP datagram size carrying the second fragment is 620 bytes. TRUE

(B) If the second fragment is lost, R will resend the fragment with the IP identification value 0×1234. FALSE

(C) If the second fragment is lost, P is required to resend the whole TCP segment. TRUE

(D) TCP destination port can be determined by analysing only the second fragment. FALSE

Ans: A & C

MTU = 1500 byte
Payload size = (1500 − 20) byte
= 1480 byte

MTU = 820 byte
Payload size =
(820 − 20) byte = 800 byte

Header size
= 20 byte

TCP Segment size = 1400 byte

1400 byte

20 + 1400 byte

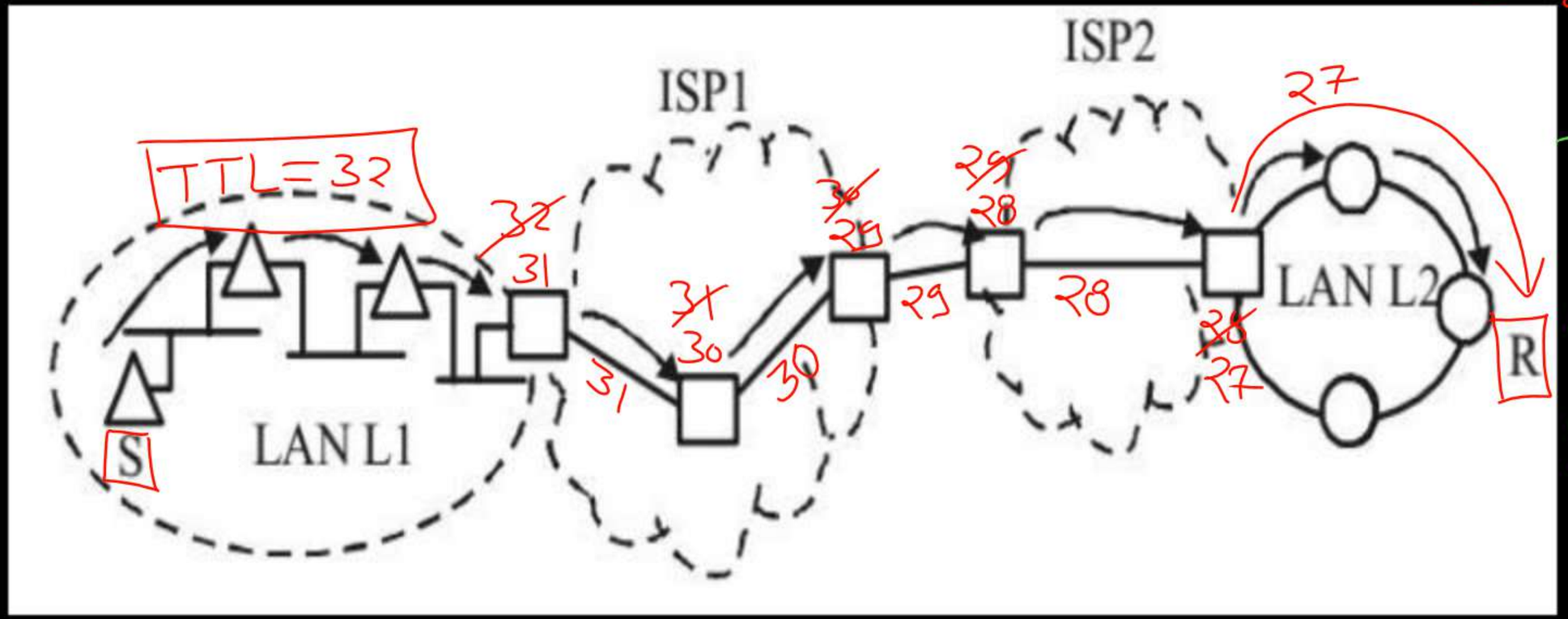
20 + 800

20 + 600 byte

#Q. In the diagram shown below L1 is an Ethernet LAN and L2 is a Token-Ring LAN. An IP packet originates from sender S and traverses to R, as shown. The link within each ISP, and across two ISPs, are all point to point optical links. The initial value of TTL is 32. The maximum possible value of TTL field when R receives the datagram is _____ .

Ans = 26

[GATE 2014]



Topic : Protocol Type

→ Protocol Type is 8-bit field in IPv4 Header

→ Indicates higher-level protocol

→ Indicates the type of protocol encapsulated in the payload

→ Demultiplexing of protocol at receiver

Topic : Protocol Type

Number	Protocol
1	ICMP
6	TCP
17	UDP
89	OSPF

Topic : Header Checksum

→ Header Checksum is 16-bit field in IPv4 Header

→ Internet Checksum

→ Ensures the integrity of the IPv4 header during routing

Topic : Header Checksum

→ Calculated over the IPv4 header only [including optional header]

→ Block Size is 16-bits

→ 16 bit one's complement of the one's complement sum of all 16 bit words in the IPv4 header.

Topic : Header Checksum

Block Size = 16 bits

While computing the checksum, the value of the checksum field should be initialized with zero.

IPv4 Header

VER	HLEN	Type of Service
Total Length		
Identification Number		
Flag	Fragmentation Offset	
Time-to-Live	Protocol Type	
Checksum [0 0 0 0 . . . . . 0 0 0 0]		
Source IP Address (16-bits)		
Source IP Address (16-bits)		
Destination IP Address (16-bits)		
Destination IP Address (16-bits)		

Topic : Checksum

→ Error detection technique

→ Both sender and receiver must agree on same block size (n)

→ n bit one's complement of the one's complement sum of all n bit words

Topic : Checksum

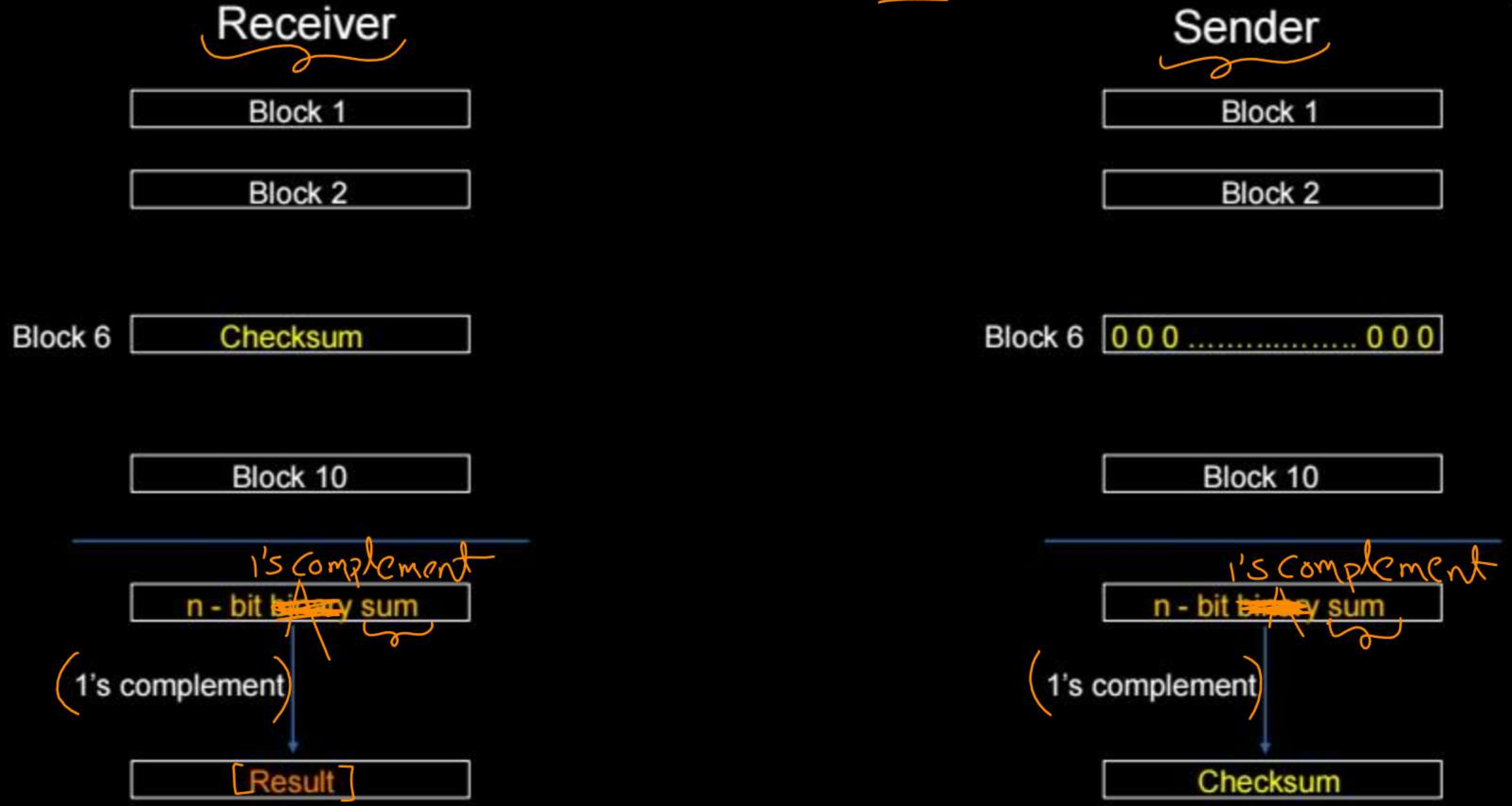
→ Sender generate n-bit Checksum from data blocks
and then send data along with checksum

→ Receiver check the "received data along with checksum" is balanced or not

→ While computing the checksum,
the value of the checksum field should be initialized with zero

→ While transmission,
checksum field should be updated with computed checksum

Block Size = n


Receiver
Block 1
Block 2
Block 6
Checksum
Block 10
1's complement
n - bit sum
(1's complement)
Result
Sender
Block 1
Block 2
Block 6
0 0 0 ..... 0 0 0
Block 10
1's complement
n - bit sum
(1's complement)
Checksum

Topic : Checksum

Result : [Computed at receiver]

if **Result == ZERO** :

then Receiver concluded "**No any error detected**"

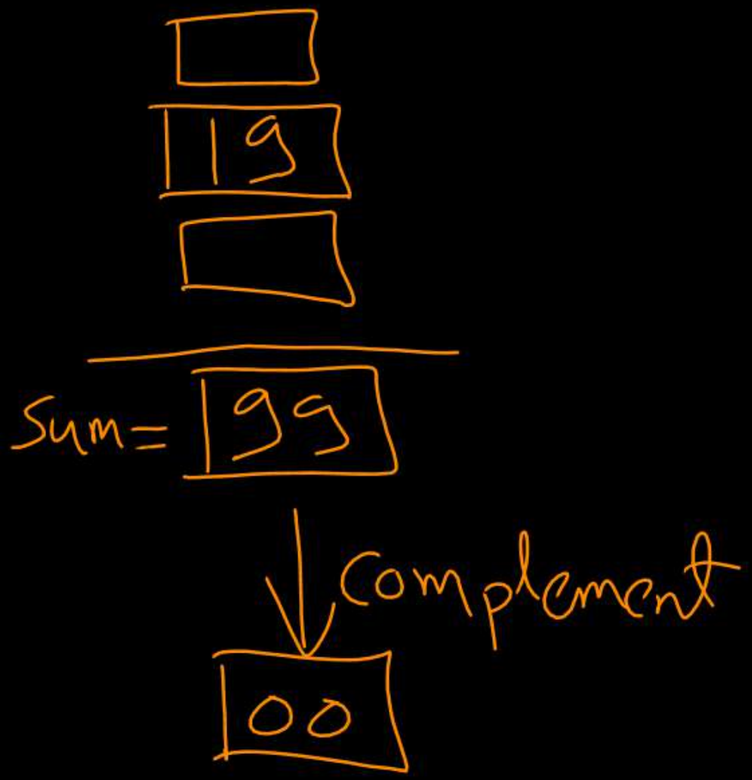
else

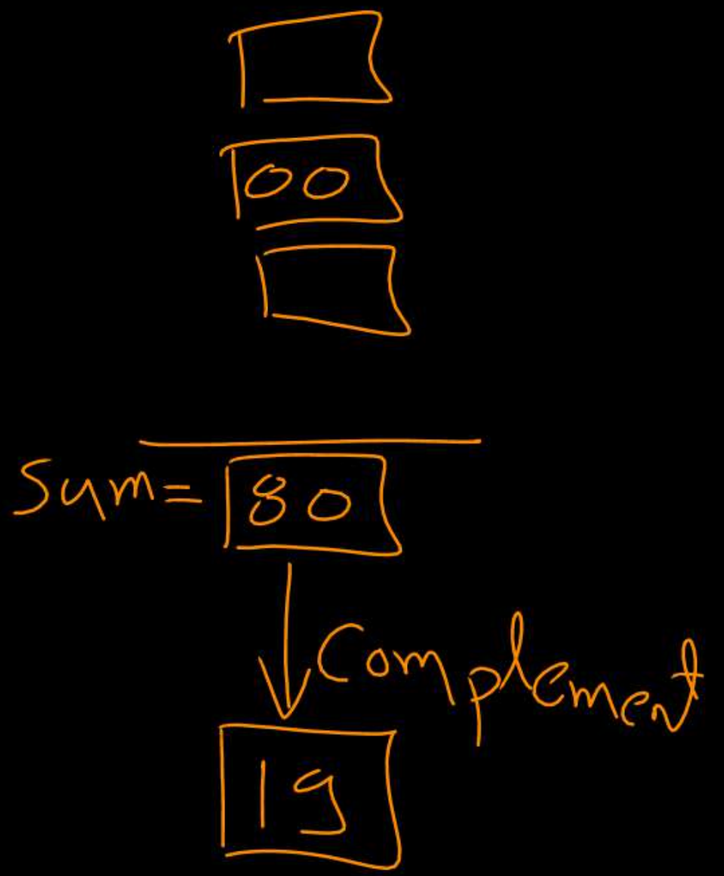
Receiver concluded "**Error detected**"

Two digit Range → 00 99

00 ------ 80 ----- 99

Complement





Example 1:-

Suppose block size = 5

```
   1
  11011
+    10
-------
  11101
```

Sender

10 10 10 1

```
  10110
  01101
  11011
  00000
  10011
  01010
-------
 10 11011
-------
  11101
```

checksum ↓ 1's comp. 00010

Example 1:-

Suppose block size = 5

Receiver

10 10 11 1

10110

01101

11011

00010

10011

01010

10 1 1 1 0 1

11111

1's comp

00000

Sender

10110

01101

11011

00000

10011

01010

10 11011

11101

1's comp

00010

Example 1:-

Suppose block size = 5

Receiver

```
  1 0 1 1 0
  0 1 1 0 1
  1 1 0 1 1
  0 0 0 1 0
  1 0 0 1 1
  0 1 0 1 0
-------------
1 0 1 1 1 0 1
-------------
  1 1 1 1 1
      ↓
  0 0 0 0 0
```

Sender

```
  1 0 1 1 0
  0 1 1 0 1
  1 1 0 1 1
  0 0 0 0 0
  1 0 0 1 1
  0 1 0 1 0
-------------
1 0 1 1 0 1 1
-------------
  1 1 1 0 1
      ↓
  0 0 0 1 0
```

Example 2:-

Suppose block size = 5

Sender

(11)(10)(11)(10)

1 1 1 1 0

0 1 1 1 1

1 1 0 1 1

0 0 0 0 0

1 1 0 1 1

0 1 0 1 1

1 1 0 1 1 1 0

1 0 0 0 1

1's comp.

checksum 0 1 1 1 0

Example 2:-

Suppose block size = 5

Receiver

4 11 4 10

1 1 1 1 0

0 1 1 1 1

1 1 0 1 1

0 1 1 1 0

1 1 0 1 1

0 1 0 1 1

1 1 1 1 0 0

1 1 1 1 1

1's comp

0 0 0 0 0

1 1 1 0 0

1 1

1 1 1 1 1

Sender

1 1 1 1 0

0 1 1 1 1

1 1 0 1 1

0 0 0 0 0

1 1 0 1 1

0 1 0 1 1

1 1 0 1 1 1 0

1 0 0 0 1

0 1 1 1 0

Example 2:-

Suppose block size = 5

Receiver

```
  1 1 1 1 0
  0 1 1 1 1
  1 1 0 1 1
  0 1 1 1 0
  1 1 0 1 1
  0 1 0 1 1
-----------
1 1 1 1 1 0 0
-----------
  1 1 1 1 1
    ↓
  0 0 0 0 0
```

Sender

```
  1 1 1 1 0
  0 1 1 1 1
  1 1 0 1 1
  0 0 0 0 0
  1 1 0 1 1
  0 1 0 1 1
-----------
1 1 0 1 1 1 0
-----------
  1 0 0 0 1
    ↓
  0 1 1 1 0
```

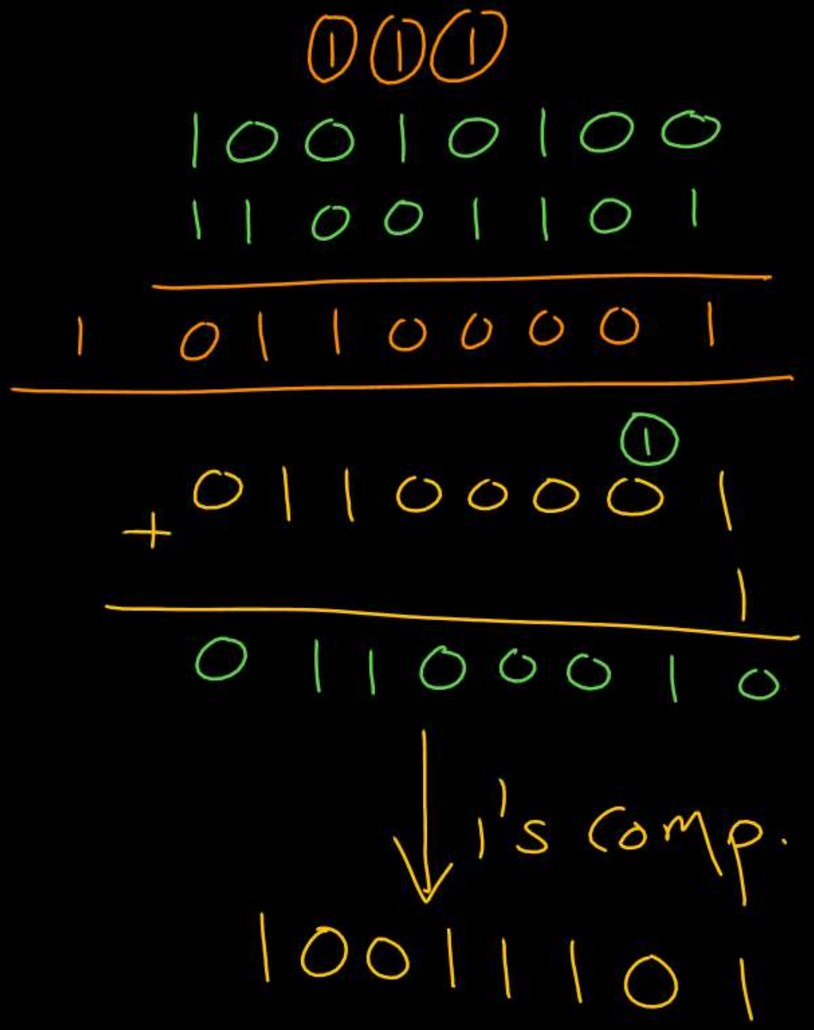
Topic : Checksum

Example 3 :-

String 1 = 1 0 0 1 0 1 0 0
String 2 = 1 1 0 0 1 1 0 1

Calculate checksum ?

Ans:
10011101

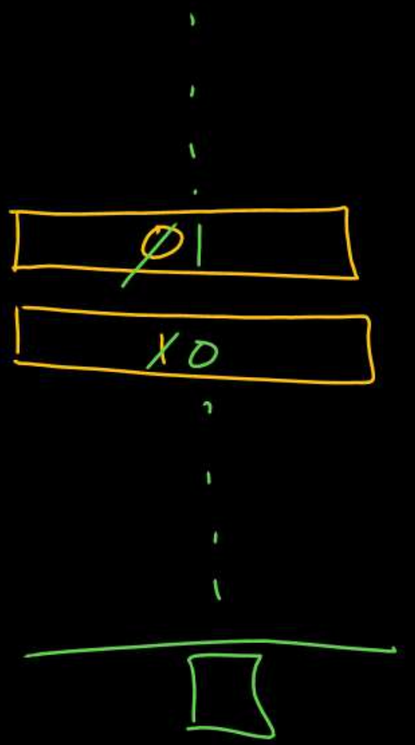


Topic : Checksum

→ Checksum detect "**all single bit error**"

→ In case of burst error,
checksum able to detect "**all odd number of errors**"

→ Checksum can be "all zero bits",
but checksum can never be "all one bits"



Example 4:-

Suppose block size = 5

Sender

1 0 0 0 0

0 1 0 0 0

0 0 1 0 0

0 0 0 0 0

0 0 0 1 0

0 0 0 0 1

---

1 1 1 1 1

---

1 1 1 1 1

↓ 1's comp.

0 0 0 0 0

Example 5:-

Suppose block size = 5

Sender

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

1's comp.

1 1 1 1 1

2 mins Summary

Topic Header Checksum

Topic ~~IPv4 Options~~

THANK - YOU