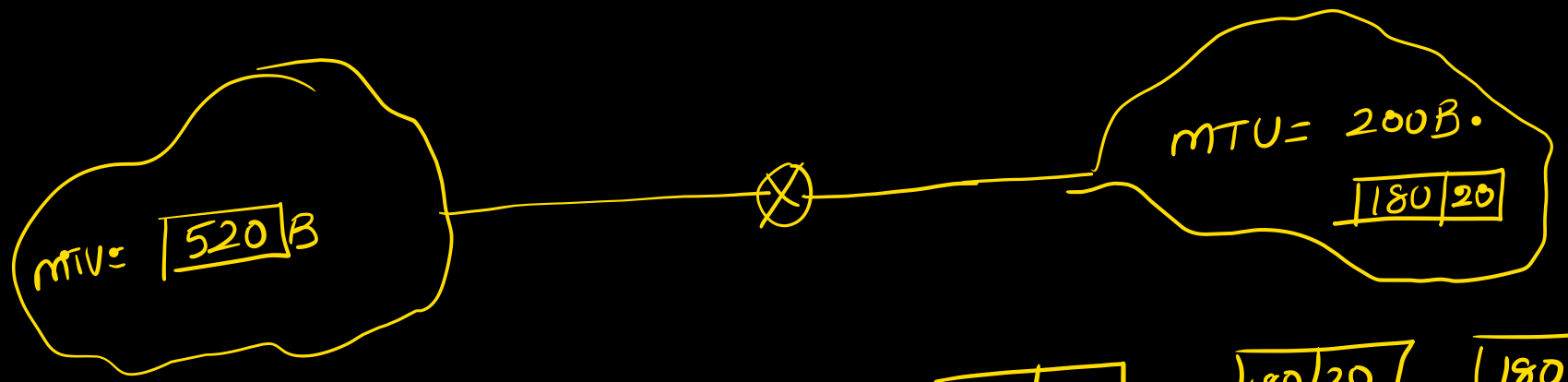
Hi everyone :

i



Data Header.

65515 20

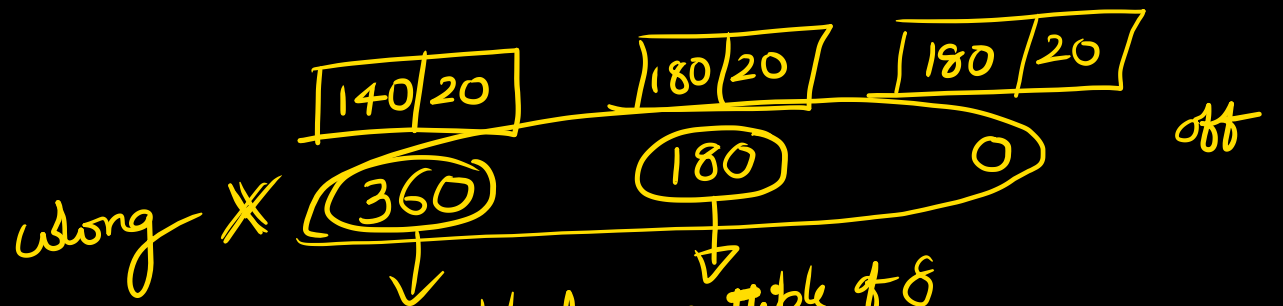
11 20 ... ...

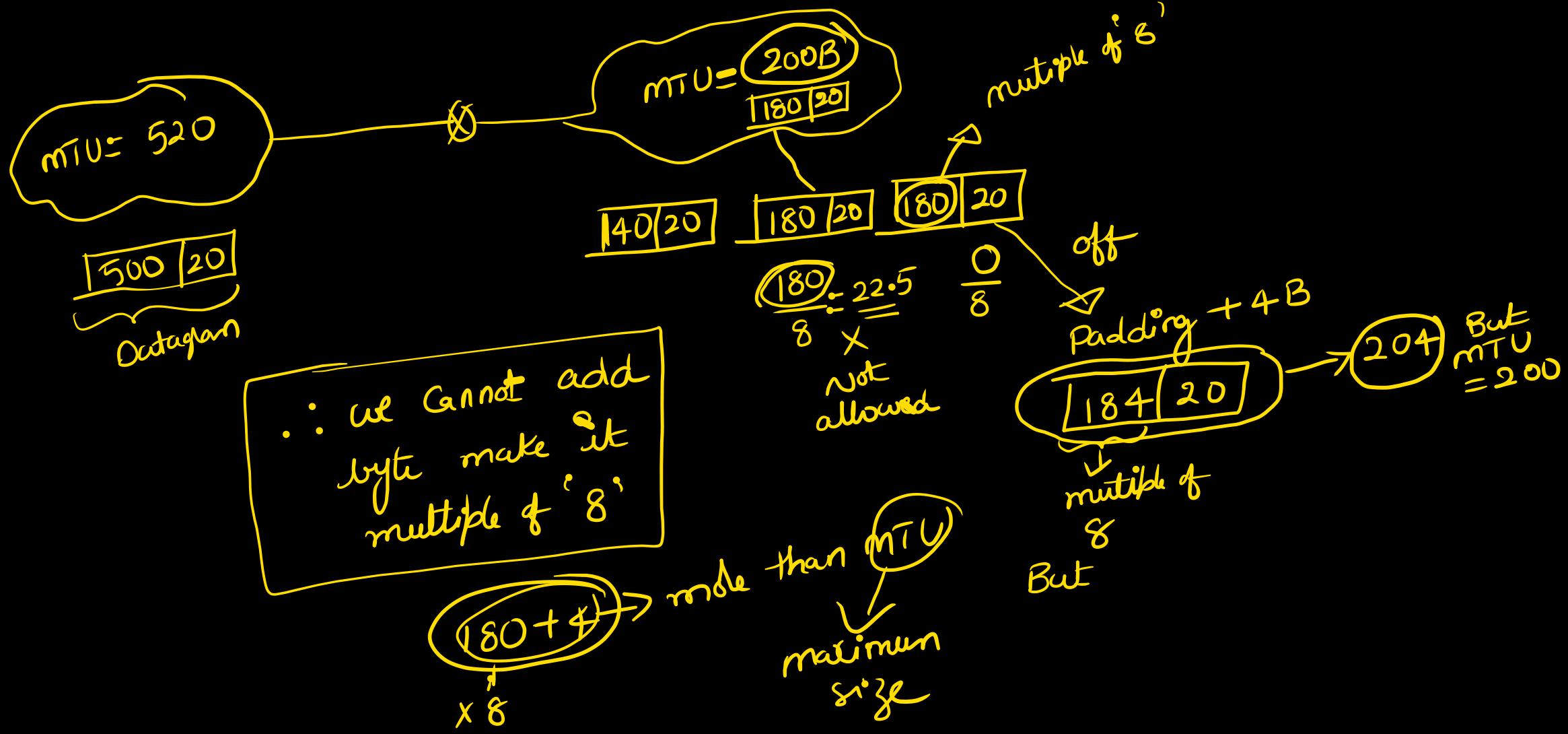
65514

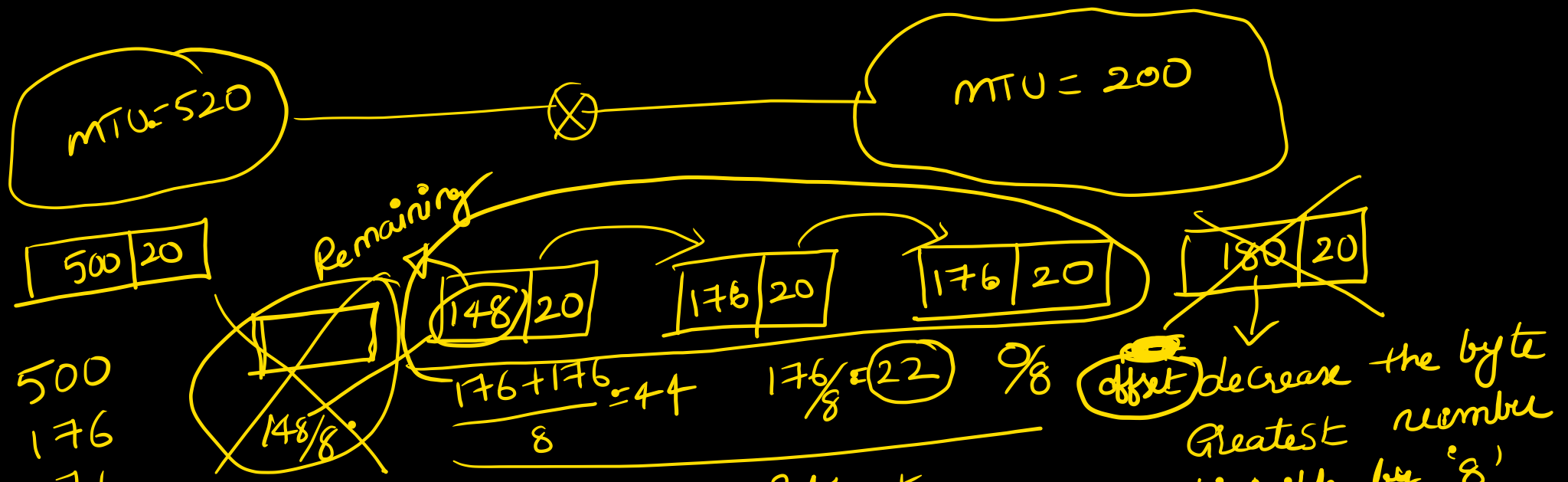
FO= 65515

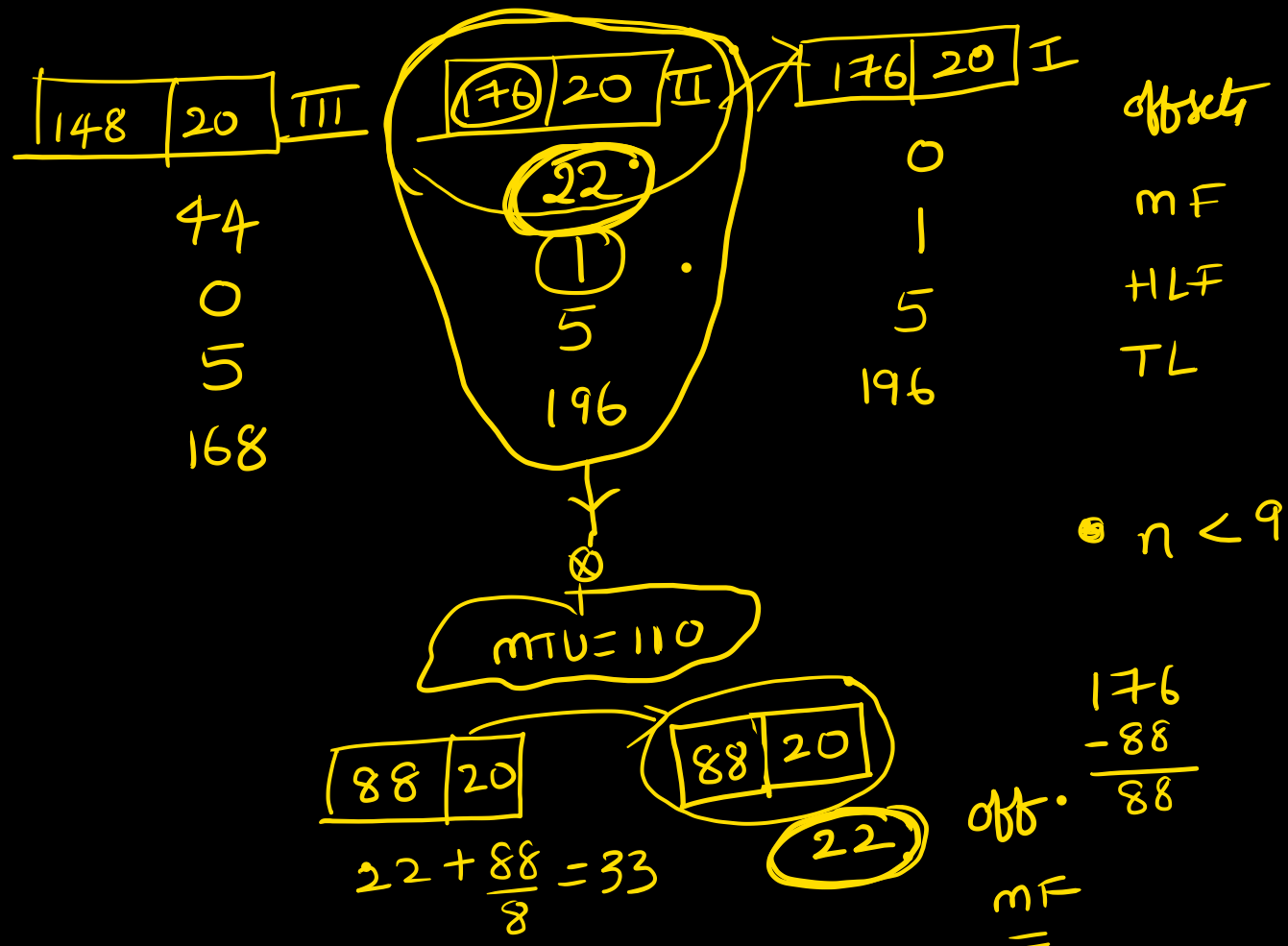
But FOF= 13bit ≅ 16K.

∴ Scaling = 65515 / 16K = 8

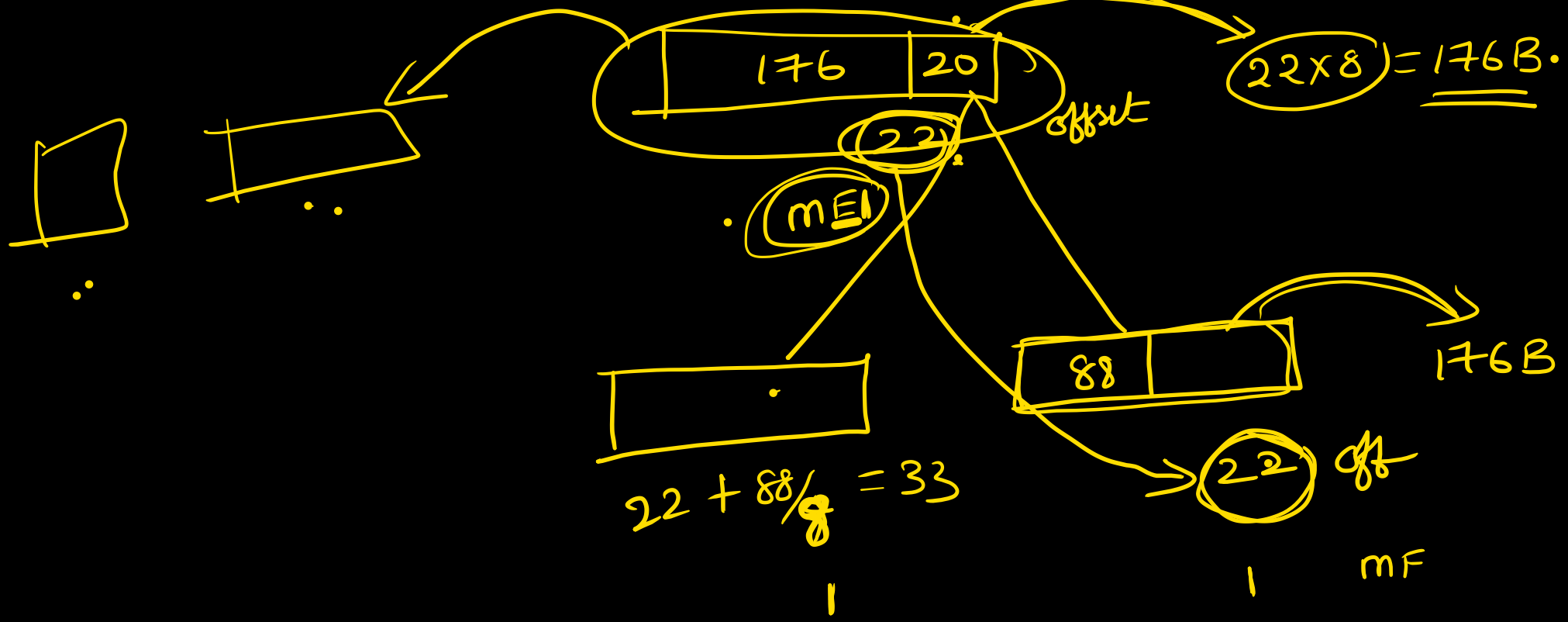



MTU = 520
MTU = 200B
180 20
multiple of '8'
500 20
Datagram
140 20
180 20
180 20
off
180 / 8 = 22.5
0/8
8 X
Not allowed
Padding +4B
184 20
204
But MTU = 200
multiple of 8
But
∴ we Cannot add byte make it multiple of '8'
180+4
more than MTU
maximum size
X 8

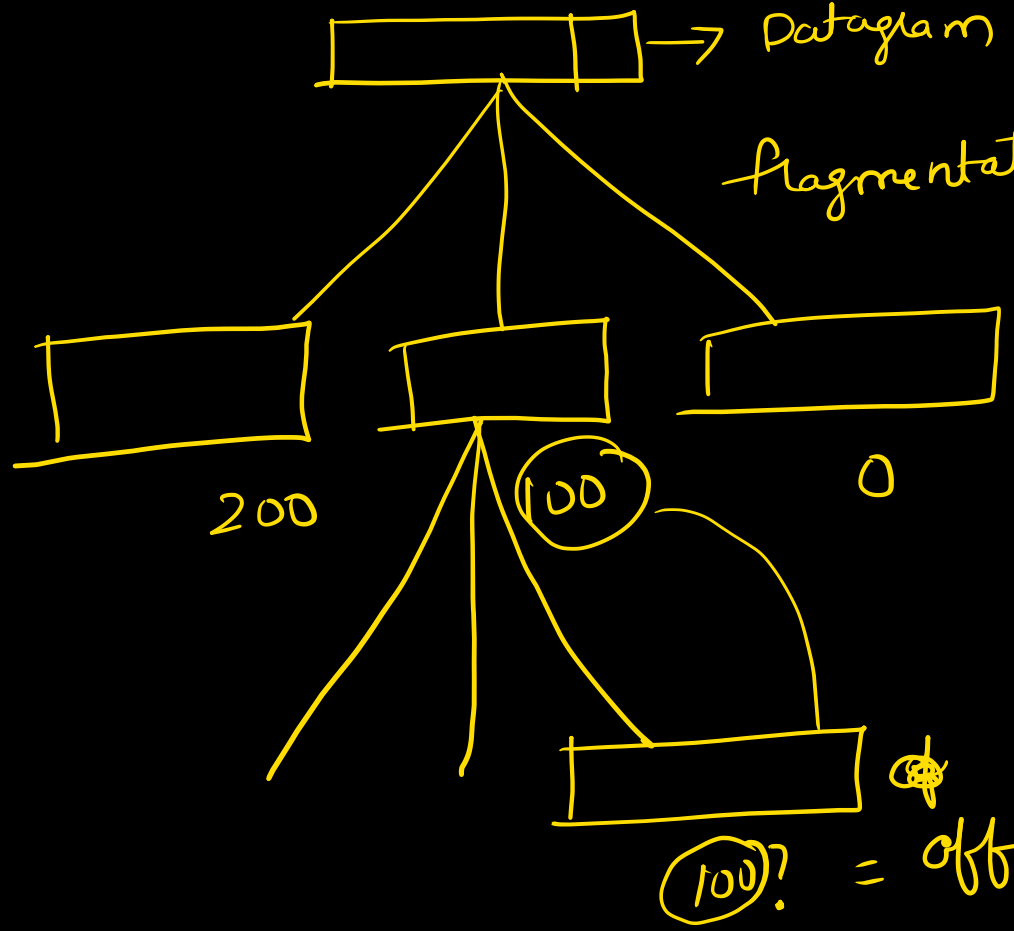

MTU=520
MTU = 200
500 | 20
Remaining
148 | 20
176 | 20
176 | 20
180 | 20
148/8
176+176 / 8 = 44
176/8 = 22
9/8
offset
decrease the byte
Greatest number
divisible by '8'
less than 180.
500
- 176
- 176
148 → Not divisible by '8', but it's ok.
why. Last fragment need not be divisible by 8.
offset
How many fragments?
3



- $n < 90 \rightarrow$ multiple of 8

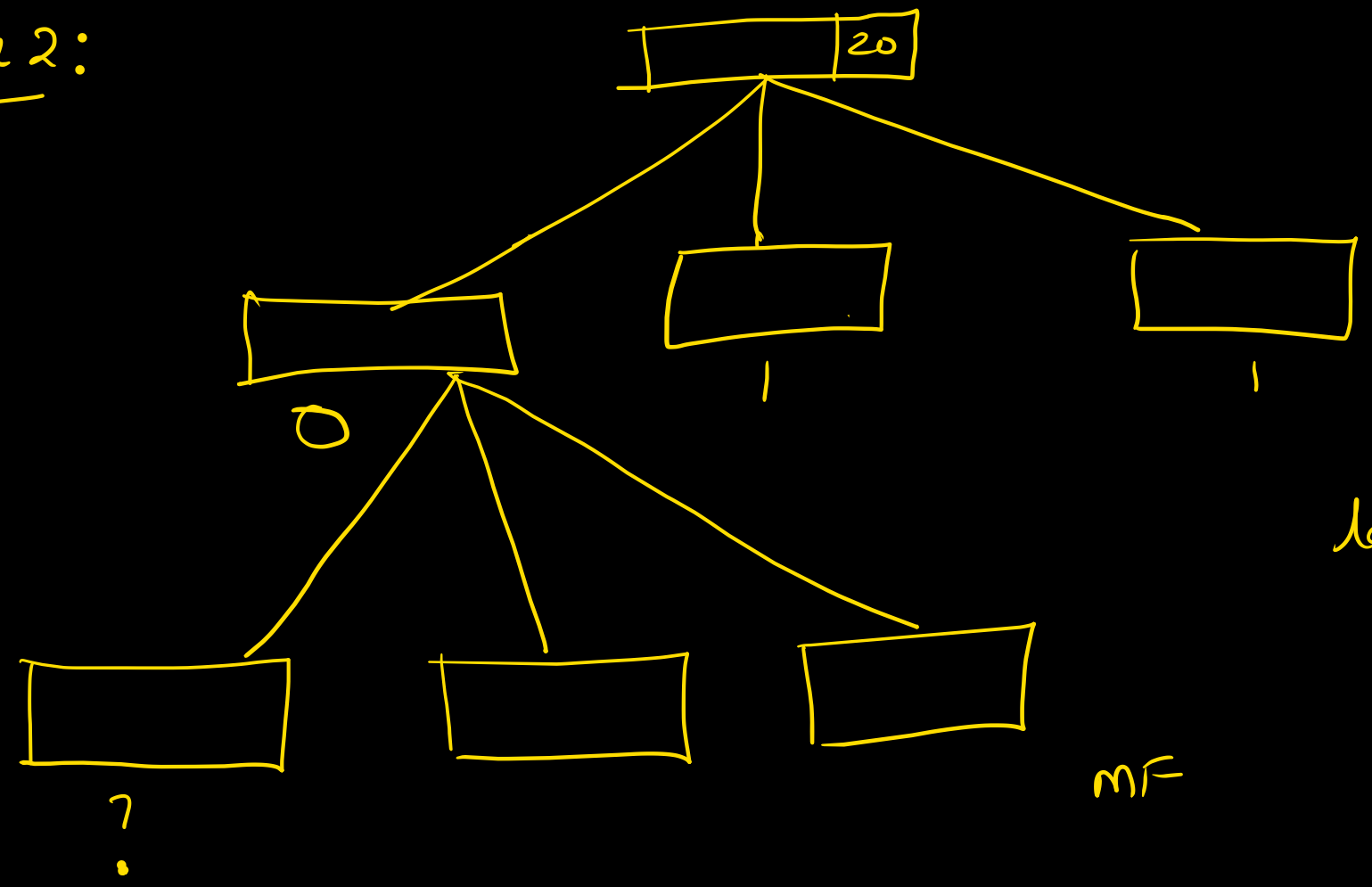

176
20
22x8 = 176B.
22
offset
MEI
88
176B
22 + 88/8 = 33
22
MF
1
1

Cases:



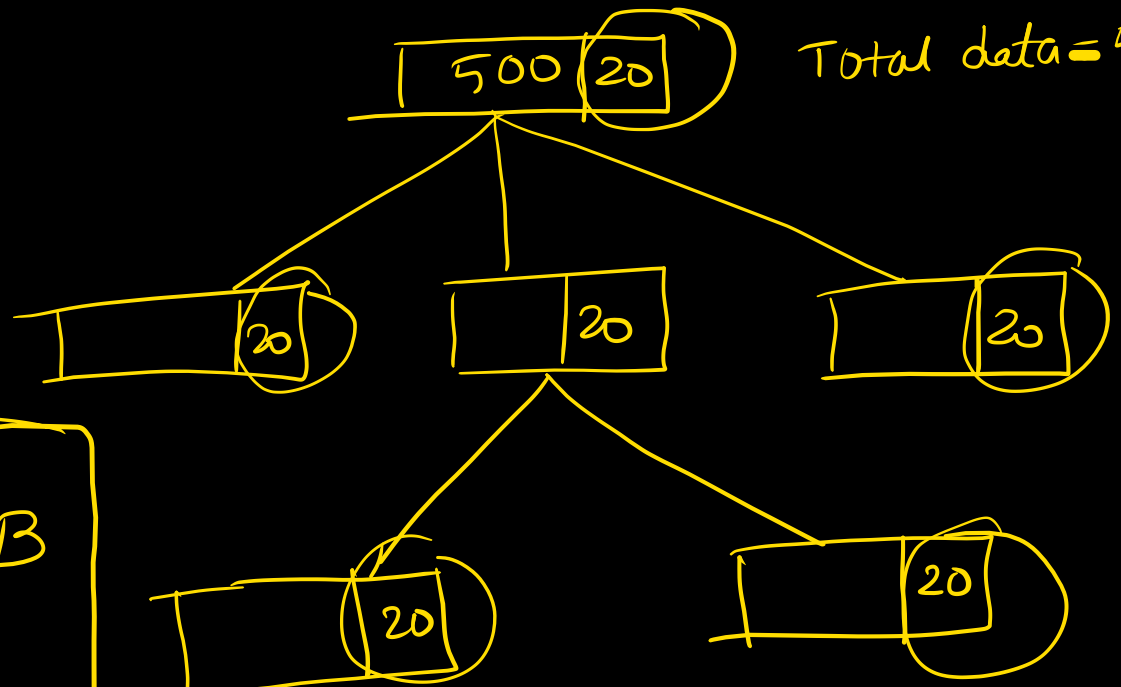
First fragment will get parents FO.

Case 2:



last fragments MF
= fauntz MF

Previous example:



Total data = 500B

? Fragments

original packet → 1H

all the fragments → 4H

∴ over head = (4 - 1)

= 3 headers

= 60 B

$\eta = \frac{\text{useful data}}{\text{Total data}}$

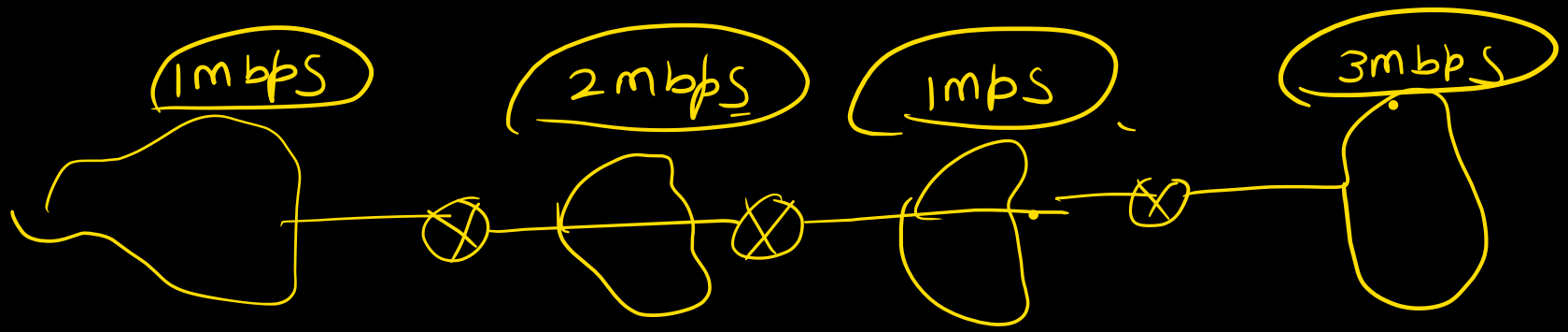
$= \frac{500B}{500B + 4*20B}$

4 fragments

$= \frac{500}{560}$

BW of Throughput = BW utilization

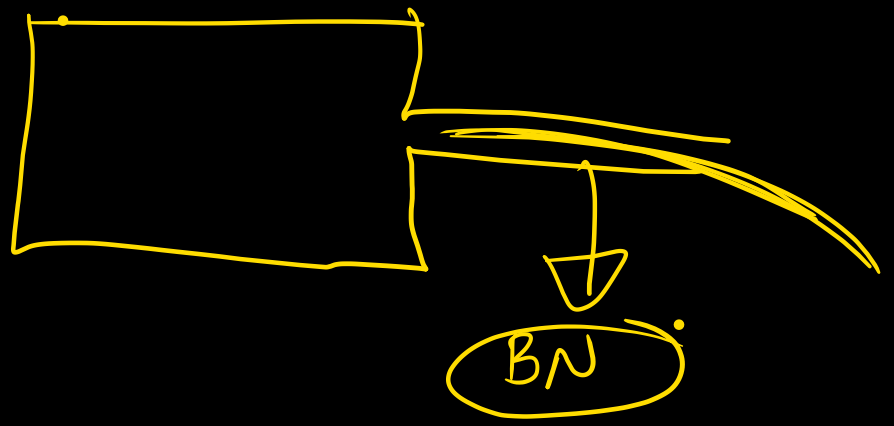
$= \eta * B$

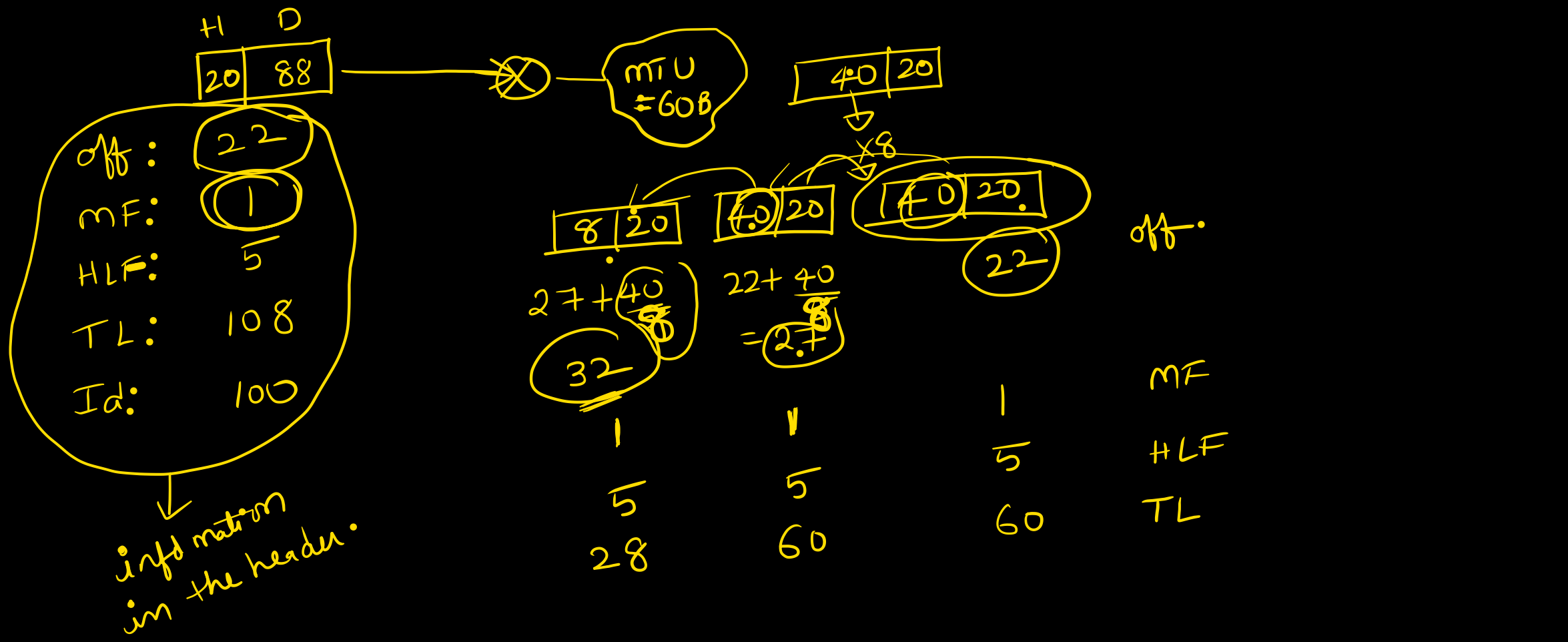


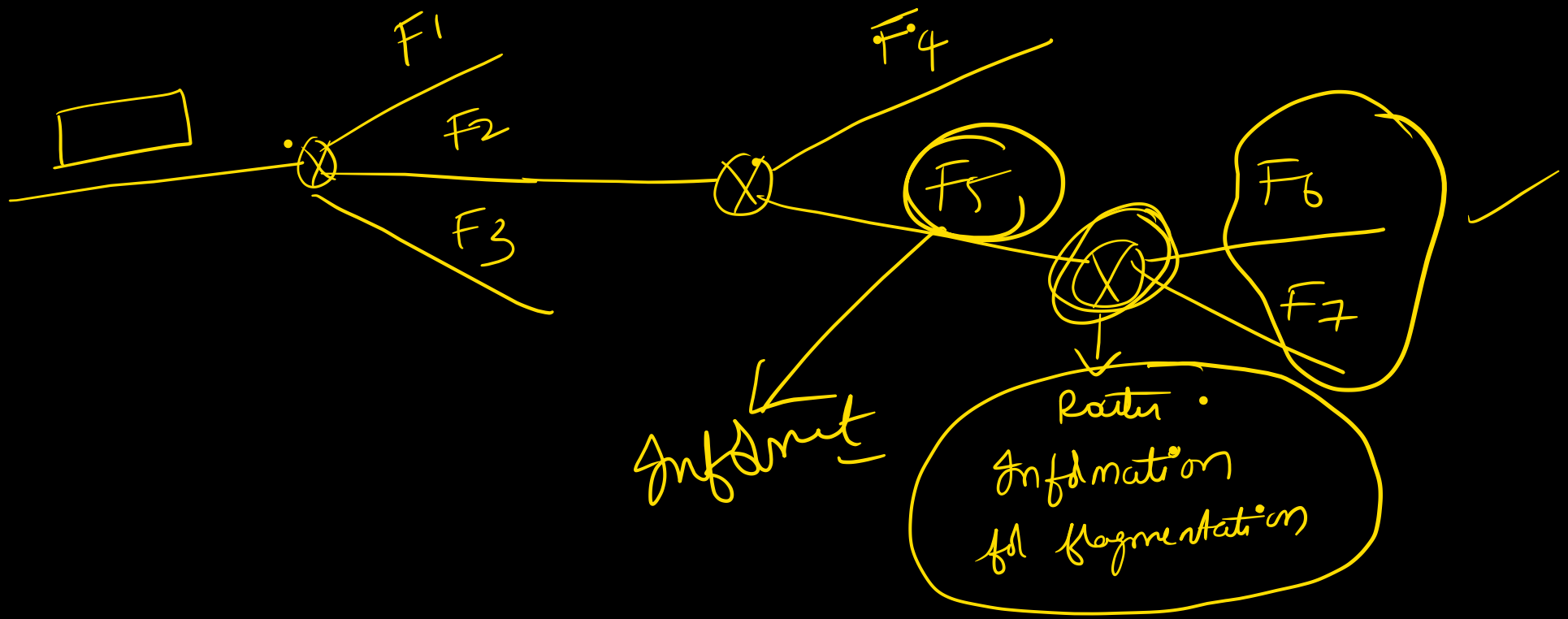
Q = BW utili = η * BW

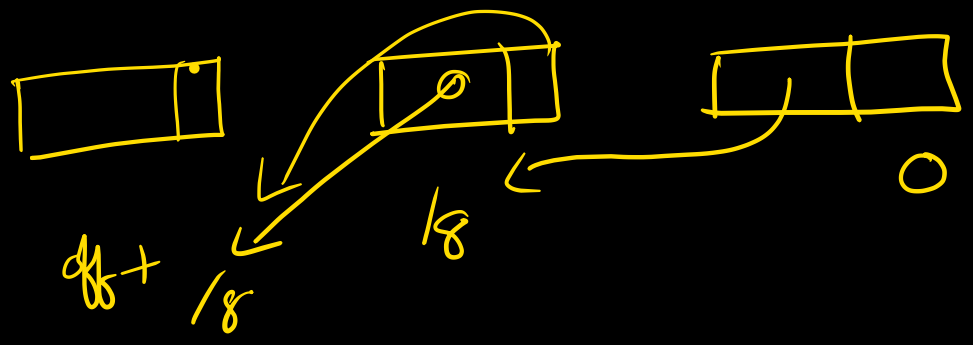
= minimum

Bottleneck.




HI
D
20
88
MTU
=60B
40
20
x8
8
20
40
20
40
20
22
off:
MF:
1
HLF:
5
TL:
108
Id:
100
information
in the header.
27+40/8
=32
22+40/8
=27
32
1
5
28
1
5
60
1
5
60
off.
MF
HLF
TL


F1
F2
F3
F4
F5
F6
F7
Router
Information
for
Fragmentation


0xff+
18
18
0

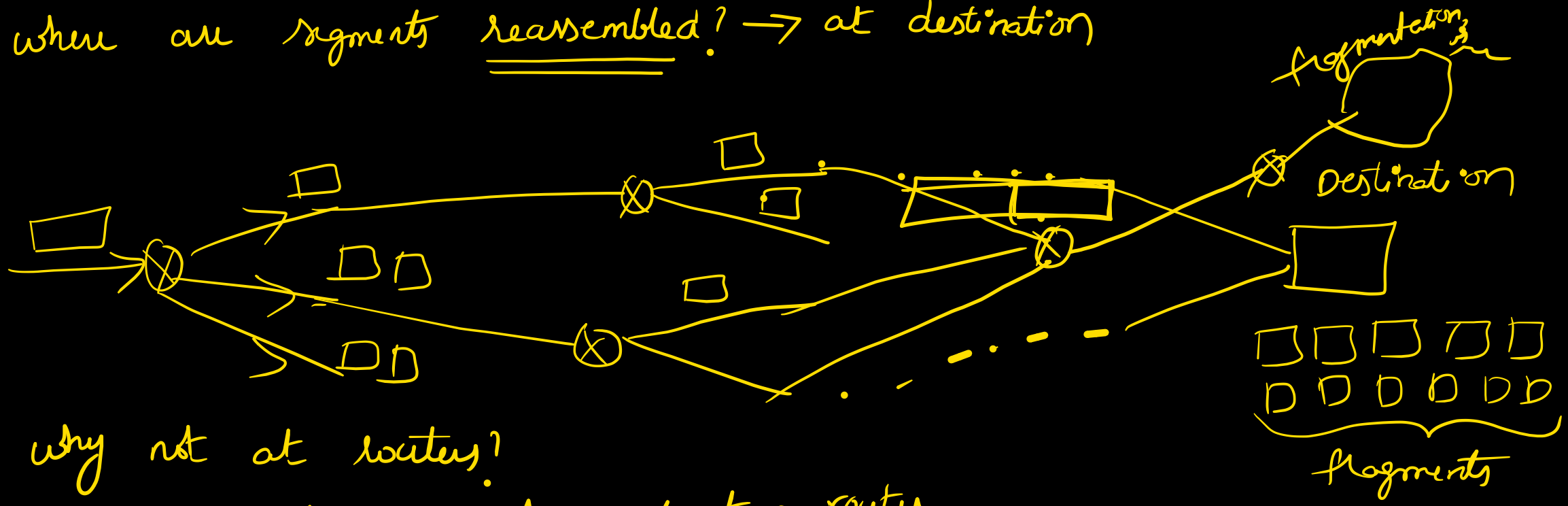
where is fragmentation done?

→ Routers, not at the source

↓ why?

Source does segmentation.

2) where are segments reassembled? → at destination

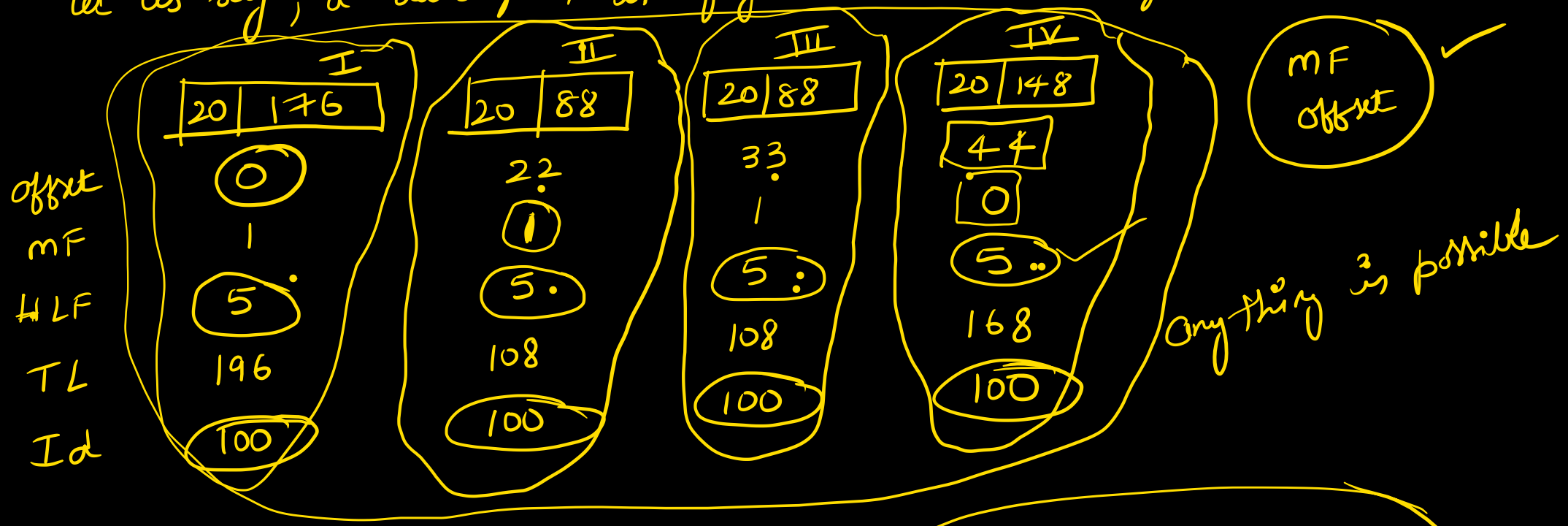


why not at routers?

→ All fragments may not meet at a router

→ may be fragmentation is required in future

let us say, a datagram is fragmented into 4 fragments:



First: (2) → understand that fragmentation is done.

Destination

1, 2, 3, 4

4, 3, 1, 2

⋮

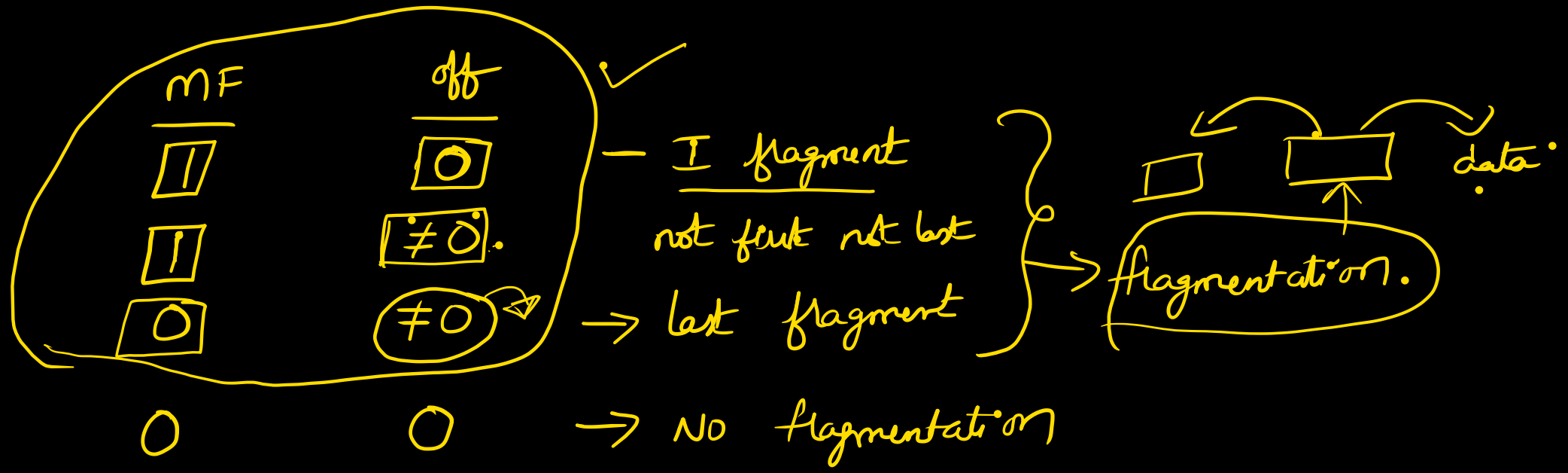
Sometimes some fragments are lost.

1 3 4 → 2 is missing

1 3 2 → 4 is missing

Reassembly at destination:

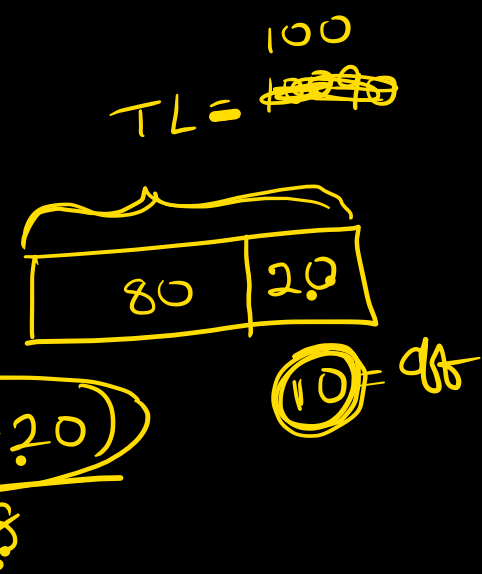
1) Destination should identify that datagram is fragmented. ✓



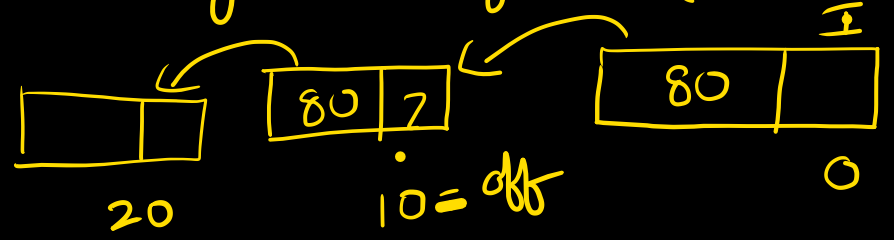
2) Destination should identify & collect all fragments belonging to the same data gram

→ using the identification number.

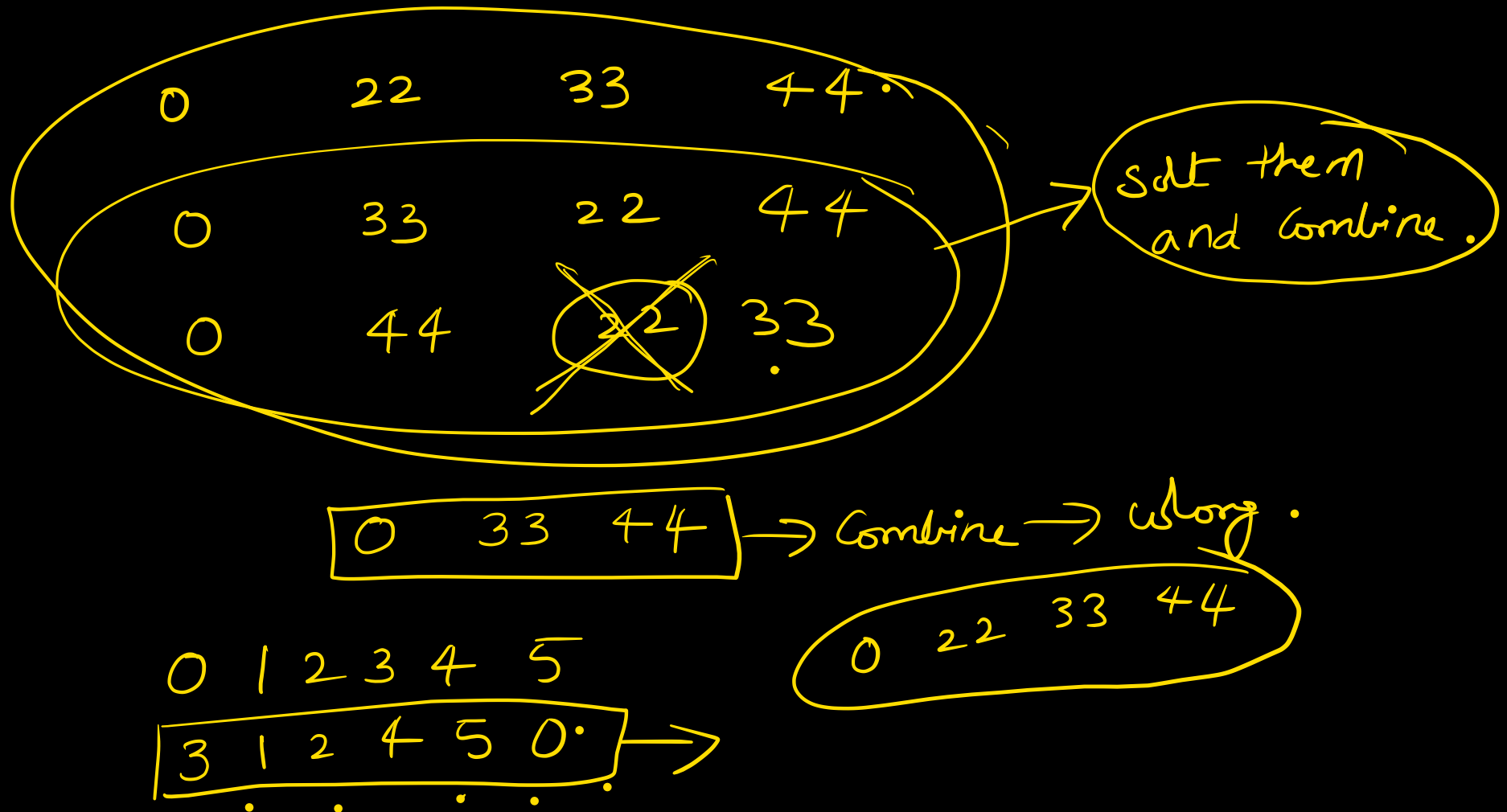
3) Identify the I fragment (offset = 0)



4) Identify Subsequent (next) packets.



Formula: offset of next fragment = off of Current fragment + $\frac{TL - HL \times 4}{8}$


0 22 33 44
0 33 22 44
0 44 22 33
Solve them and Combine.
0 33 44
Combine
0 22 33 44
0 1 2 3 4 5
3 1 2 4 5 0

5) Repeat it until MF = 0 ✓

→ NO more fragments.

discarded.

↓

TCP will take care

IP is unreliable

and

no acks.