IP

(slides derived from past EE122 sections)
(multi-homing slides taken from Nick Feamster)

Today

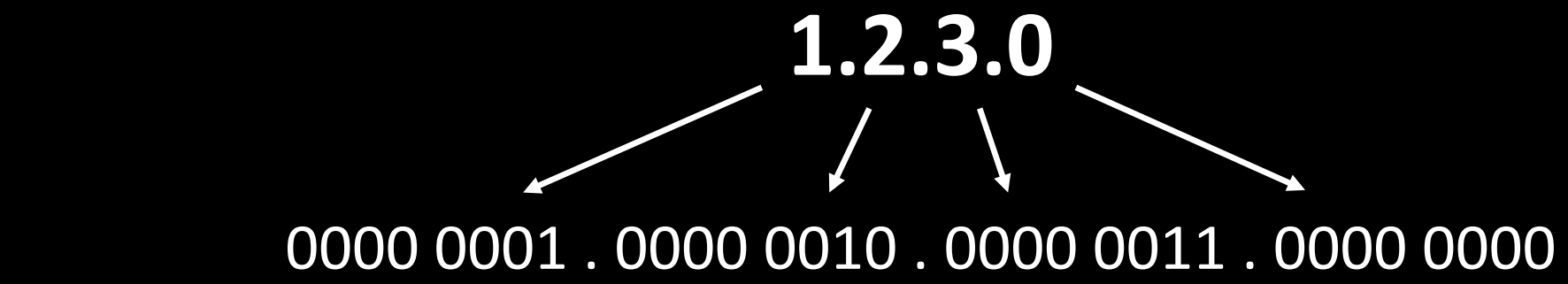
IP Addressing (Q0, Q1)

- multi-homing

IP Header (Q2)

IP Fragmentation (Q3)

IP Addressing


1.2.3.0
0000 0001 . 0000 0010 . 0000 0011 . 0000 0000

1.2.3.0 / 24


0000 0001 . 0000 0010 . 0000 0011 . 0000 0000
Network Portion
Host Portion

10.192.128.0 / 18

Q0

192.168.0.0 / 13

1100 0000 . 1010 1000 . 0000 0000 . 0000 0000

Network Portion

Host Portion

Q0

1100 0000 . 1010 1000 . 0000 0000 . 0000 0000

1111 1111 . 1111 1000 . 0000 0000 . 0000 0000

Subnet Mask

255.248.0.0

Q0

Smallest Address

1100 0000 . 1010 1000 . 0000 0000 . 0000 0000

Largest Address

1100 0000 . 1010 1111 . 1111 1111 . 1111 1111

Address Range

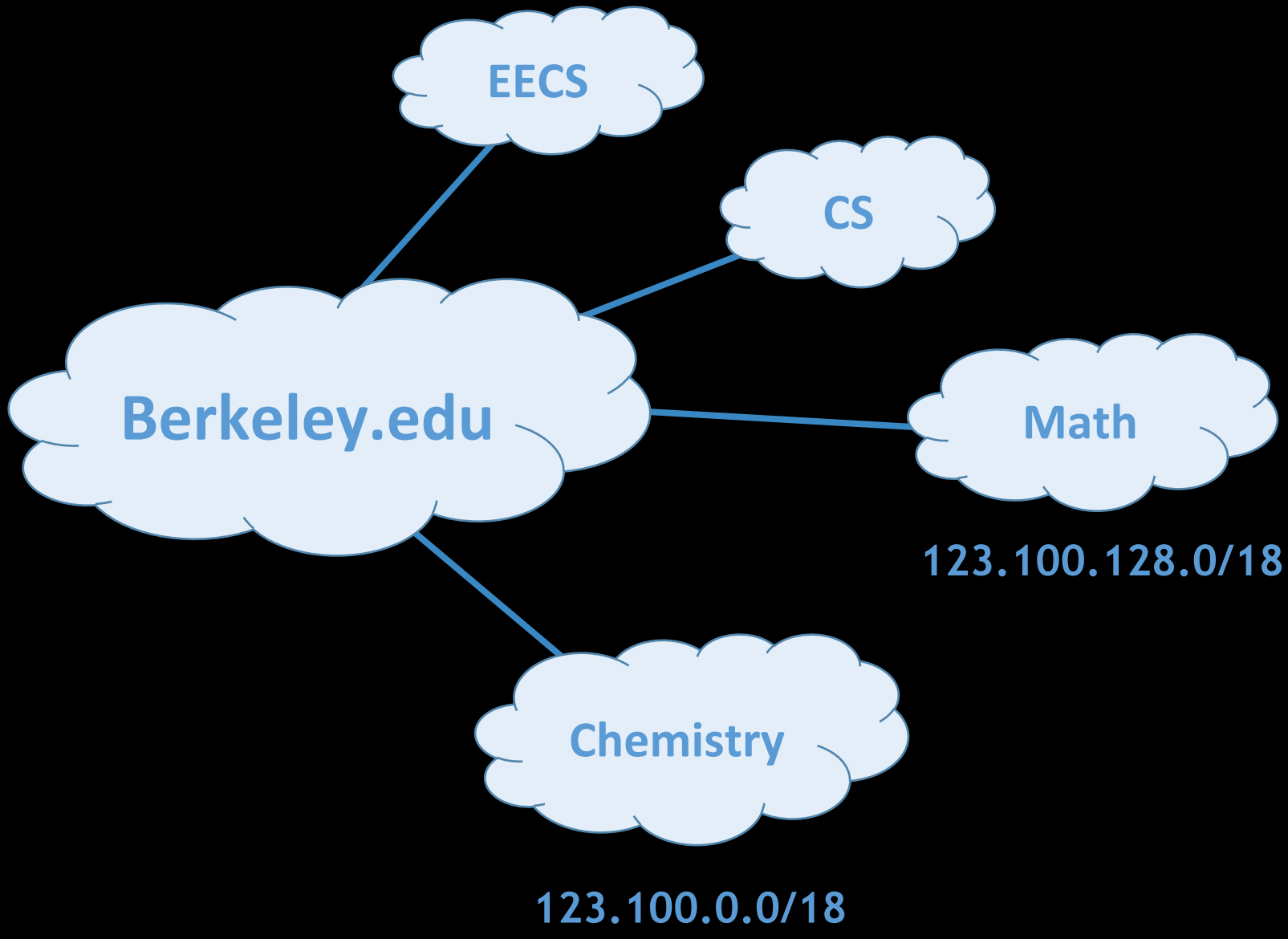
192.168.0.0 → 192.175.255.255

Q0

192.168.0.0 / 13

123.100.0.5	First quad does not match
192.128.69.5	Second quad is less than the minimum
192.168.244.8	Match!
192.175.100.0	Match!
192.176.3.4	Second quad is greater than the maximum

Q1


EECS
CS
Berkeley.edu
Math
123.100.128.0/18
Chemistry
123.100.0.0/18

Q1a

Smallest Address

01111011 . 01100100 . 1000 0000 . 0000 0000

Largest Address

01111011 . 01100100 . 1011 1111 . 1111 1111

Address Range

123.100.128.0 → 123.100.191.255

2^{14} addresses

Q1b

123.100.192.0/18

0111 1011 . 0110 0100 . 1100 0000 . 0000 0000

0111 1011 . 0110 0100 . 1100 0000 . 0000 0000
0111 1011 . 0110 0100 . 1110 0000 . 0000 0000

123.100.192.0/19
123.100.224.0/19

Q1c

123.100.192.0/18	0111 1011 . 0110 0100 . 1100 0000 . 0000 0000
123.100.128.0/18	0111 1011 . 0110 0100 . 1000 0000 . 0000 0000
123.100.0.0/18	0111 1011 . 0110 0100 . 0000 0000 . 0000 0000
123.100.0.0/16	0111 1011 . 0110 0100 . 0000 0000 . 0000 0000

Q1d

$2^5 < 50 < 2^6$

Need at least 6 bits for host portion → /26
Well, easy. Just assign 123.100.0.0/26

Problem?

123.100.0.0/26 is part of 123.100.0.0/18!
This takes a chunk out of Chemistry's address space.

Q1d

So, what's left?

123.100.192.0/18	0111 1011 . 0110 0100 . 1100 0000 . 0000 0000
123.100.128.0/18	0111 1011 . 0110 0100 . 1000 0000 . 0000 0000
123.100.0.0/18	0111 1011 . 0110 0100 . 0000 0000 . 0000 0000
123.100.64.0/18	0111 1011 . 0110 0100 . 0100 0000 . 0000 0000

Allocate from this chunk

123.100.64.0/26

What is Multihoming?

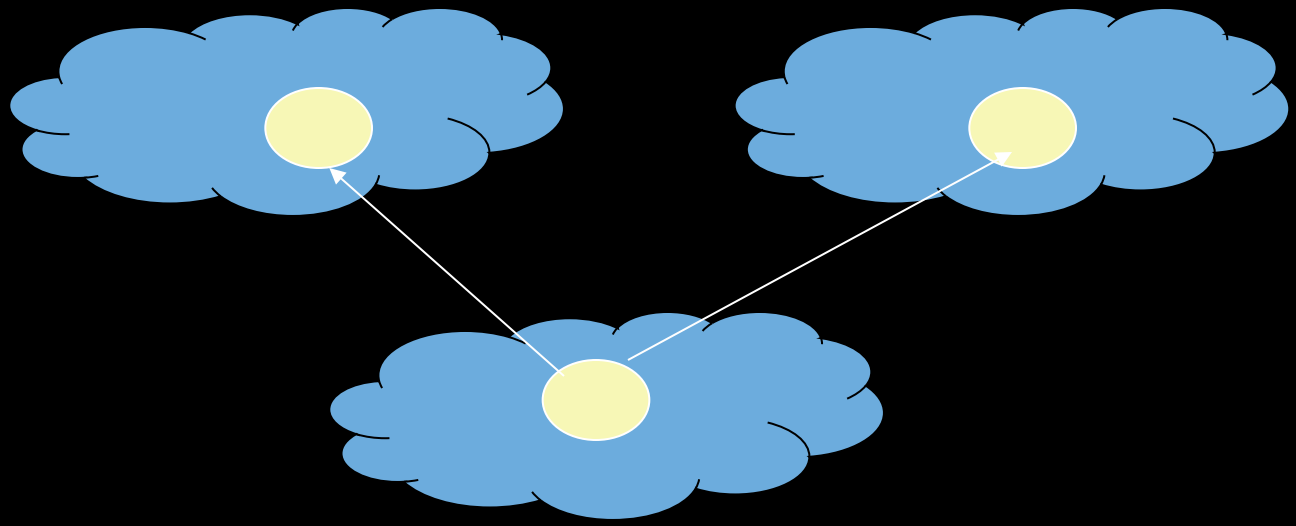
- The use of redundant network links for the purposes of *external* connectivity

- Can be achieved at many layers of the protocol stack and many places in the network
  - Multiple network interfaces in a PC
  - An ISP with multiple upstream interfaces

- Can refer to having multiple connections to
  - The same ISP
  - Multiple ISPs

Why Multihome?

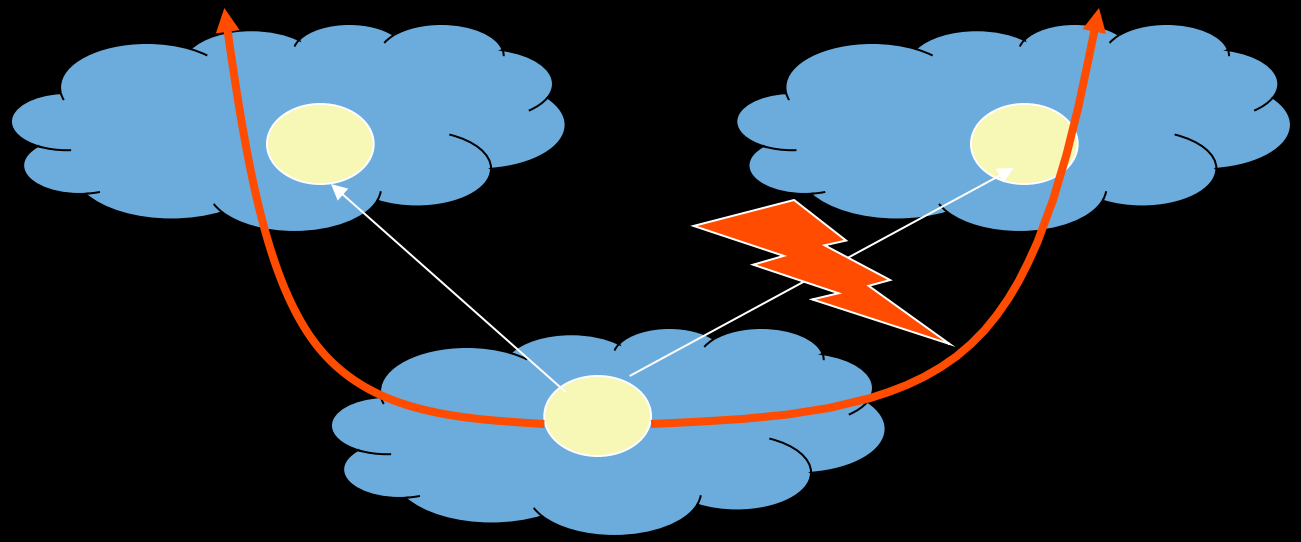
- Availability
- Performance
- Cost



Interdomain traffic engineering: the process by which a multihomed network configures its network to achieve these goals

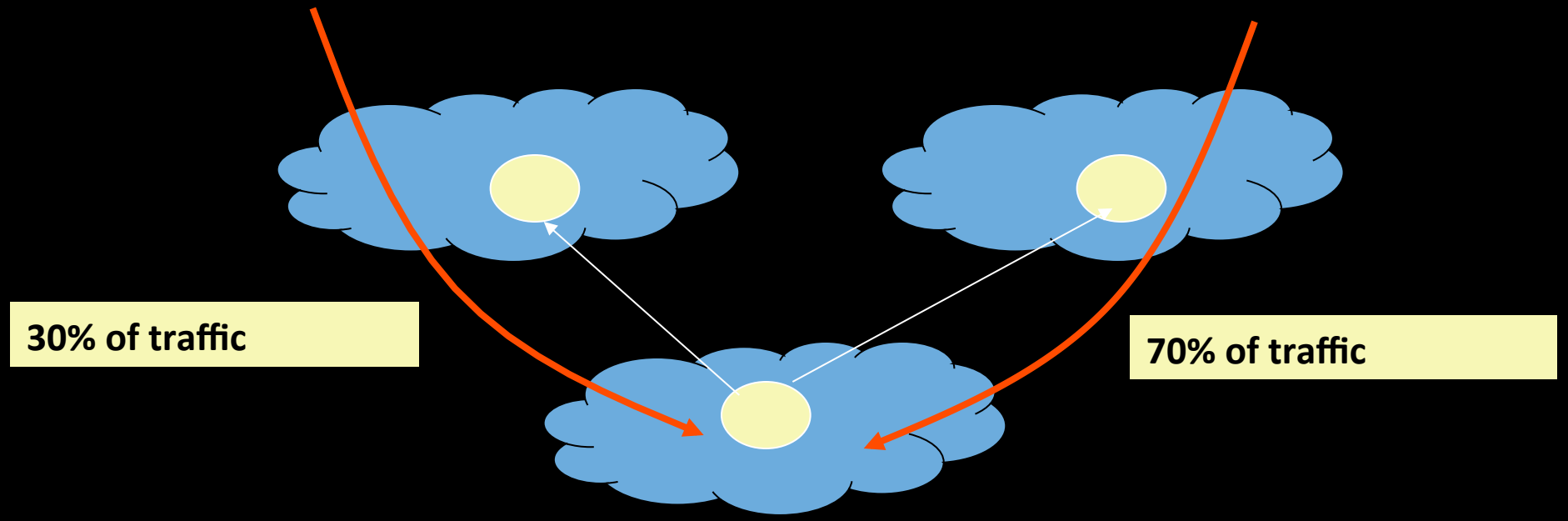
Availability

- Maintain connectivity in the face of:
  - Physical connectivity problems (fiber cut, device failures, etc.)
  - Failures in upstream ISP



Performance

- Use multiple network links at once to achieve **higher throughput** than just over a single link.
- Allows incoming traffic to be **load-balanced**.

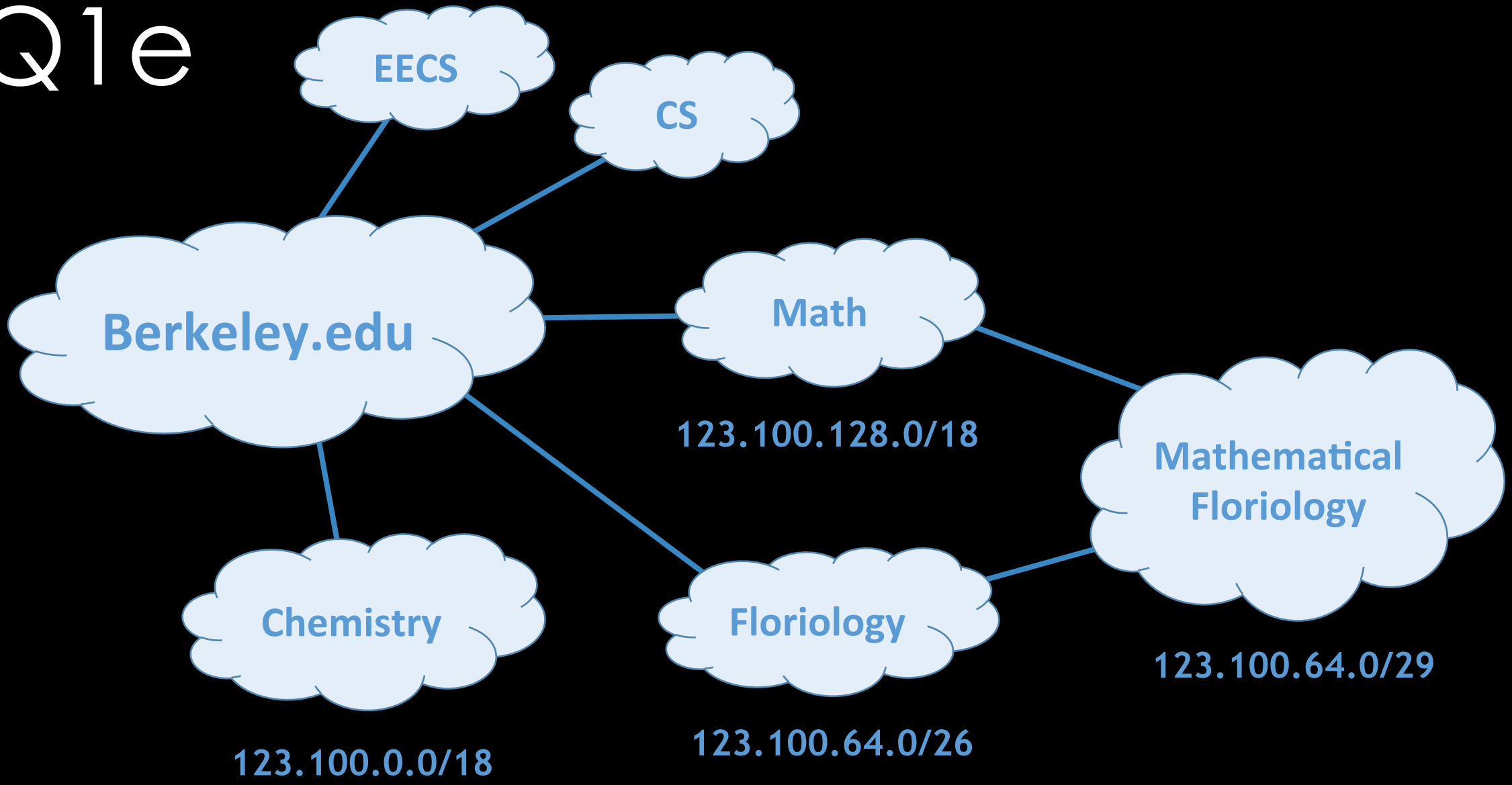


Problem with Multihoming

- Routing table growth
  - Advertising prefix out multiple ISPs – can't aggregate

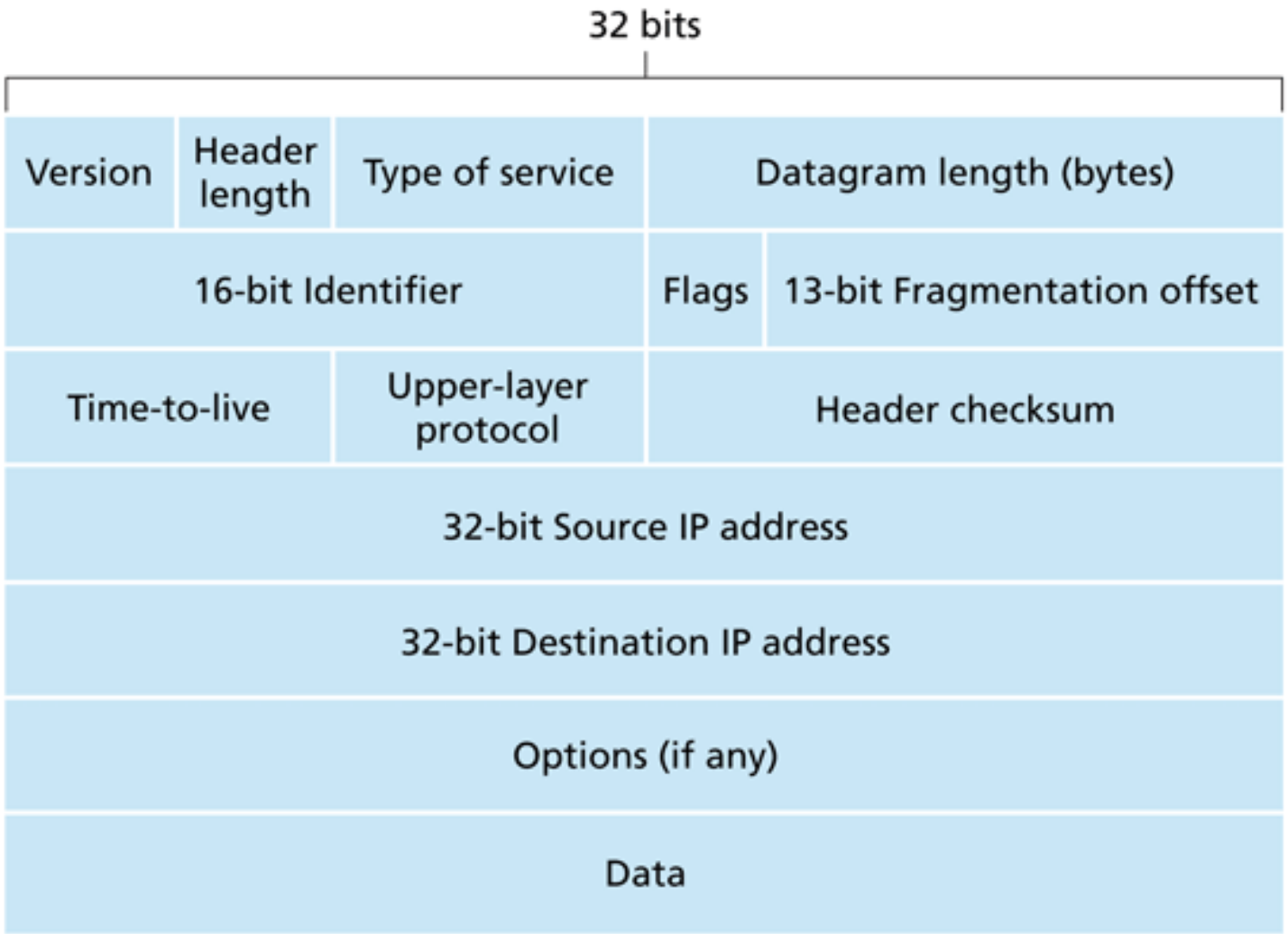
Let's take Q1e as an example!

Q1e


EECS
CS
Berkeley.edu
Math
123.100.128.0/18
Mathematical
Floriology
123.100.64.0/29
Chemistry
Floriology
123.100.0.0/18
123.100.64.0/26

IP Header

Q2


32 bits
Version
Header length
Type of service
Datagram length (bytes)
16-bit Identifier
Flags
13-bit Fragmentation offset
Time-to-live
Upper-layer protocol
Header checksum
32-bit Source IP address
32-bit Destination IP address
Options (if any)
Data

Q2

a) TTL, checksum

b) Packet will loop forever (TTL never decremented)

c) Routers by other vendors will drop your packets!

d) If there are options, your router will read a bogus address

Fragmentation

Fragmentation Fields

Identifier: which fragments belong together

Flags:

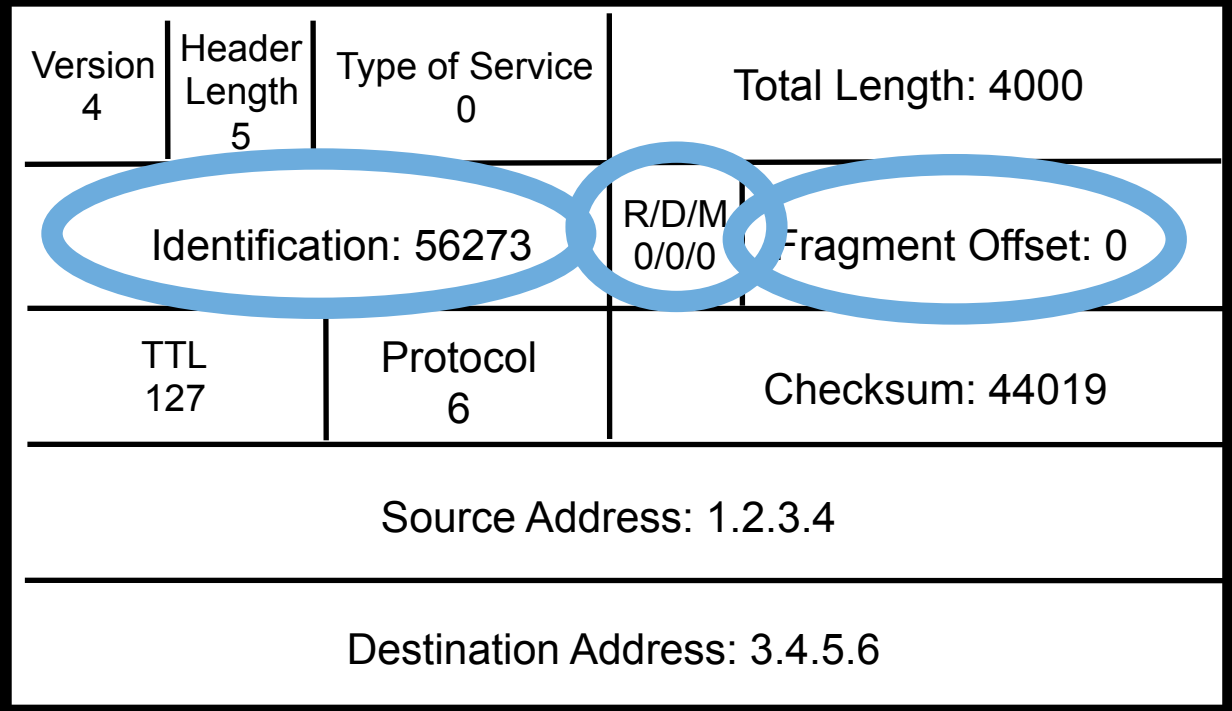
- **R**eserved: ignore
- **D**F: don't fragment
- **M**F: more fragments coming

Offset: portion of datagram this fragment contains

- **in 8-byte units**

What if fragments arrive out of order?

- Isn't MF meaningless?
- Doesn't the data get out of order?



IP Header

Why This Works

Fragment without MF set (last fragment)

- Tells host which are the last bits in datagram

All other fragments fill in holes in datagram

Can tell when holes are filled, regardless of order

Example of Fragmentation

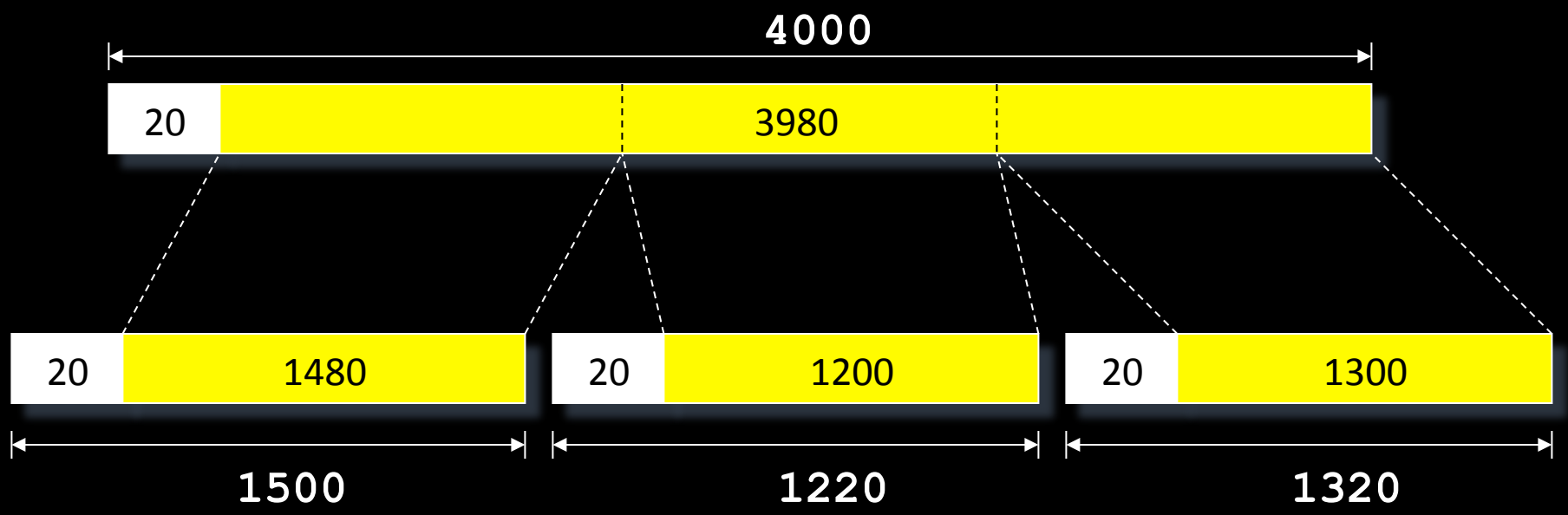
4000 byte packet, but Maximum Transmission Unit (MTU) is 1500 bytes

Version 4	Header Length 5	Type of Service 0	Total Length: 4000
Identification: 56273		R/D/M 0/0/0	Fragment Offset: 0
TTL 127	Protocol 6	Checksum: 44019	
Source Address: 1.2.3.4			
Destination Address: 3.4.5.6			

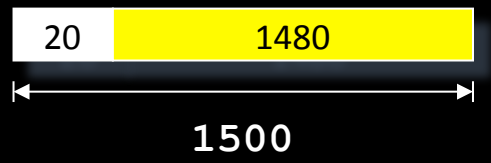
(3980 more bytes of payload here)

Example of Fragmentation

Datagram split into 3 pieces



Example of Fragmentation



First piece (many possibilities!)

Version 4	Header Length 5	Type of Service 0	Total Length: 1500
Identification: 56273		R/D/M 0/0/1	Fragment Offset: 0
TTL 127	Protocol 6		Checksum: xxx
Source Address: 1.2.3.4			
Destination Address: 3.4.5.6			

Example of Fragmentation

Second piece. First 1480 bytes already covered, so offset = 1480 / 8 = 185

Version 4	Header Length 5	Type of Service 0	Total Length: 1220	
Identification: 56273			R/D/M 0/0/1	Fragment Offset: 185
TTL 127		Protocol 6	Checksum: xxx	
Source Address: 1.2.3.4				
Destination Address: 3.4.5.6				

Example of Fragmentation

Third piece. First (1480+1200) = 2680 bytes already covered, so

Offset = 2680/ 8 = 335, and MF flag = 0

Version 4	Header Length 5	Type of Service 0	Total Length: 1320	
Identification: 56273			R/D/M 0/0/0	Fragment Offset: 335
TTL 127	Protocol 6		Checksum: xxx	
Source Address: 1.2.3.4				
Destination Address: 3.4.5.6				

Where Should Reassembly Occur?

Classic case of E2E principle

Must be done at ends

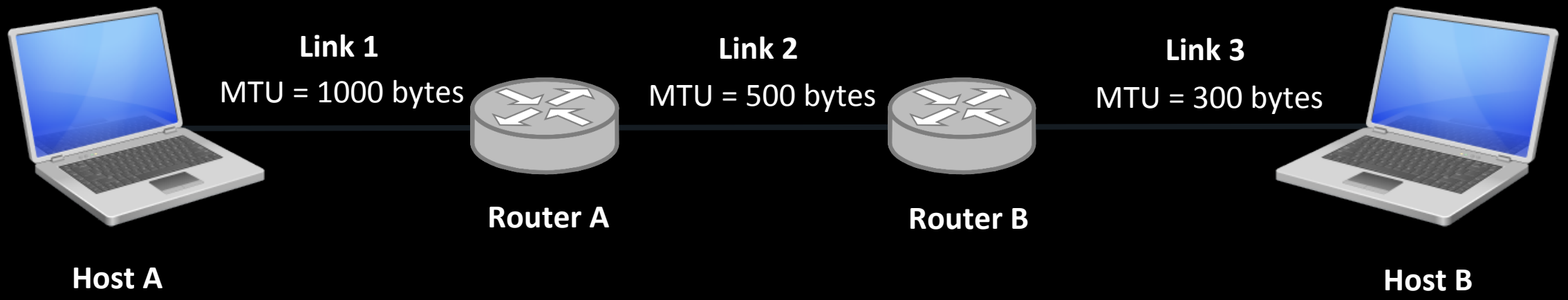
- Fragments may take different paths

Imposes burden on network

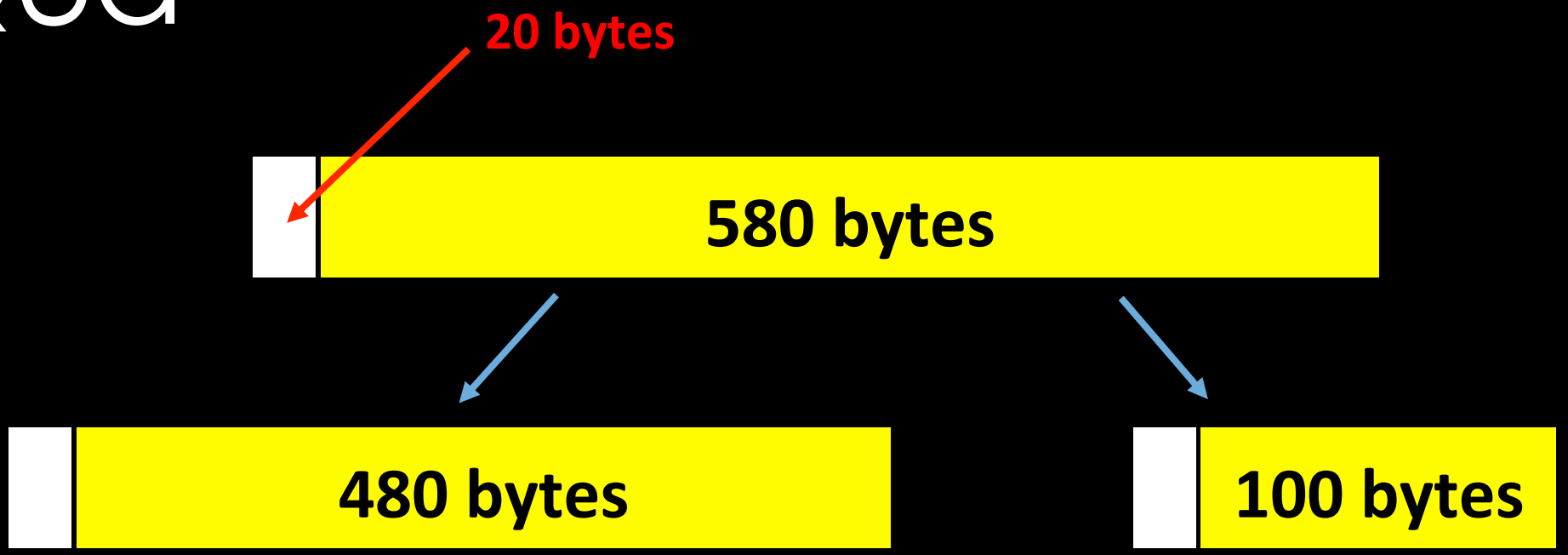
- Complicated reassembly algorithm
- Must hold onto state

Little benefit, large cost for network reassembly

Q3


Link 1
MTU = 1000 bytes
Link 2
MTU = 500 bytes
Link 3
MTU = 300 bytes
Router A
Router B
Host A
Host B

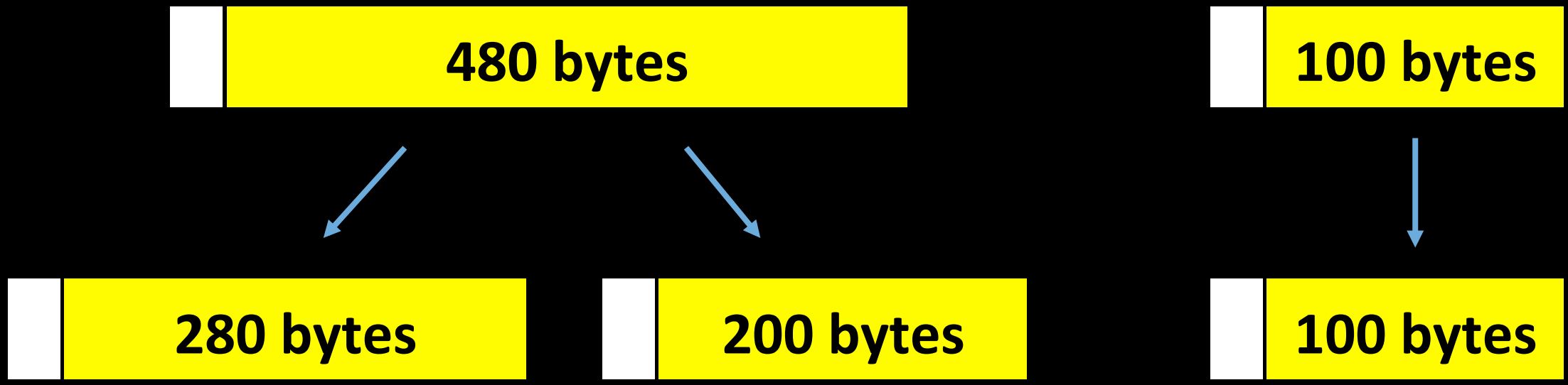
Q3a



Total length = 500 bytes
Flags = 001
Offset = 0

Total length = 120 bytes
Flags = 000
Offset = 480/8 = 60

Q3b



Total length = 300 bytes
Flags = 001
Offset = 0

Total length = 220 bytes
Flags = 001
Offset = 280/8 = 35

Total length = 120 bytes
Flags = 000
Offset = 60

Q3

c) Packets can arrive out of order

d) Offsets allow simple recursive fragmentation

f) End-to-end argument! Fragmentation is time-consuming; doing it in the network slows down forwarding