Computer Networks

(SCC.203)

Week 16-1

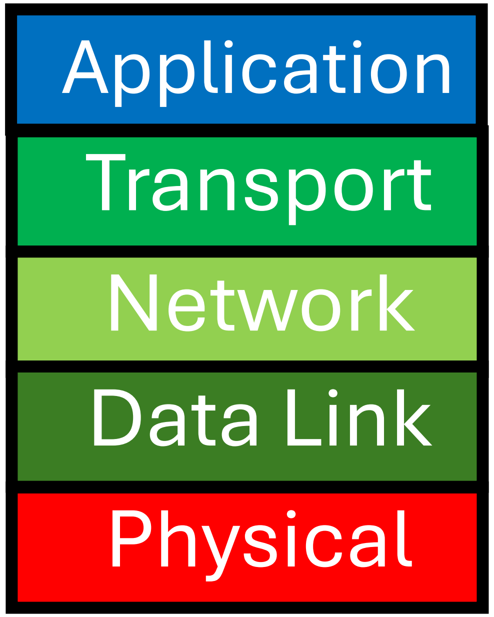
Muhammad Bilal

Week	Topic	Video Recording	Lecture Slides
11	What is the Internet?	Lecture 01	part1 part2
11	Edge & Core Networking	Lecture 02	lecture_slides_02
12	Delay Loss & Throughput	Lecture 03(No audio until minute 32 - apologies!)	lecture_slides_03
12	Protocol Layers & Encapsulation	Lecture 04	lecture_slides_04
13	Network Applications	Lecture 05	lecture_slides_05
13	Web & HTTP	Lecture 06	lecture_slides_06
14	Email	Lecture 07	email_slides
14	DNS	Lecture 08	dns_slides
15	Network Transport & UDP	Lecture 09	udp_slides
15	TCP	Lecture 10	tcp_slides
16	Buffering, Forwarding, IPv4 & Addressing		
16	NAT & DHCP		
17	Switching & Routing		
17	BGP & OSPF		
18	Multiple Access & LANs		
18	Error Detection & Correction		
19	Congestion Control		
19	Advanced Topics in Networking I		
20	Advanced Topics in Networking II		
20	Revision Lecture		

Announcement!

- The course work 2 assessment will be based on a quiz scheduled for week 20, which will cover the tasks outlined in the lab document.
    1. **Quiz Date:** The quiz will be conducted in week 20.
    2. **Assessment Criteria:** Your performance in the quiz will be based on your implementation of the tasks provided in the lab document.
    3. **Task Implementation:** It is essential for all students to implement all the tasks mentioned in the lab document before the quiz date. Please ensure that you bring your implemented models or solutions to the quiz.
    4. **Quiz Structure:** The quiz questions will be designed to assess your understanding and implementation of the lab tasks. If you have successfully implemented all the tasks as per the document, you should be able to answer the quiz questions confidently.

Network Layer



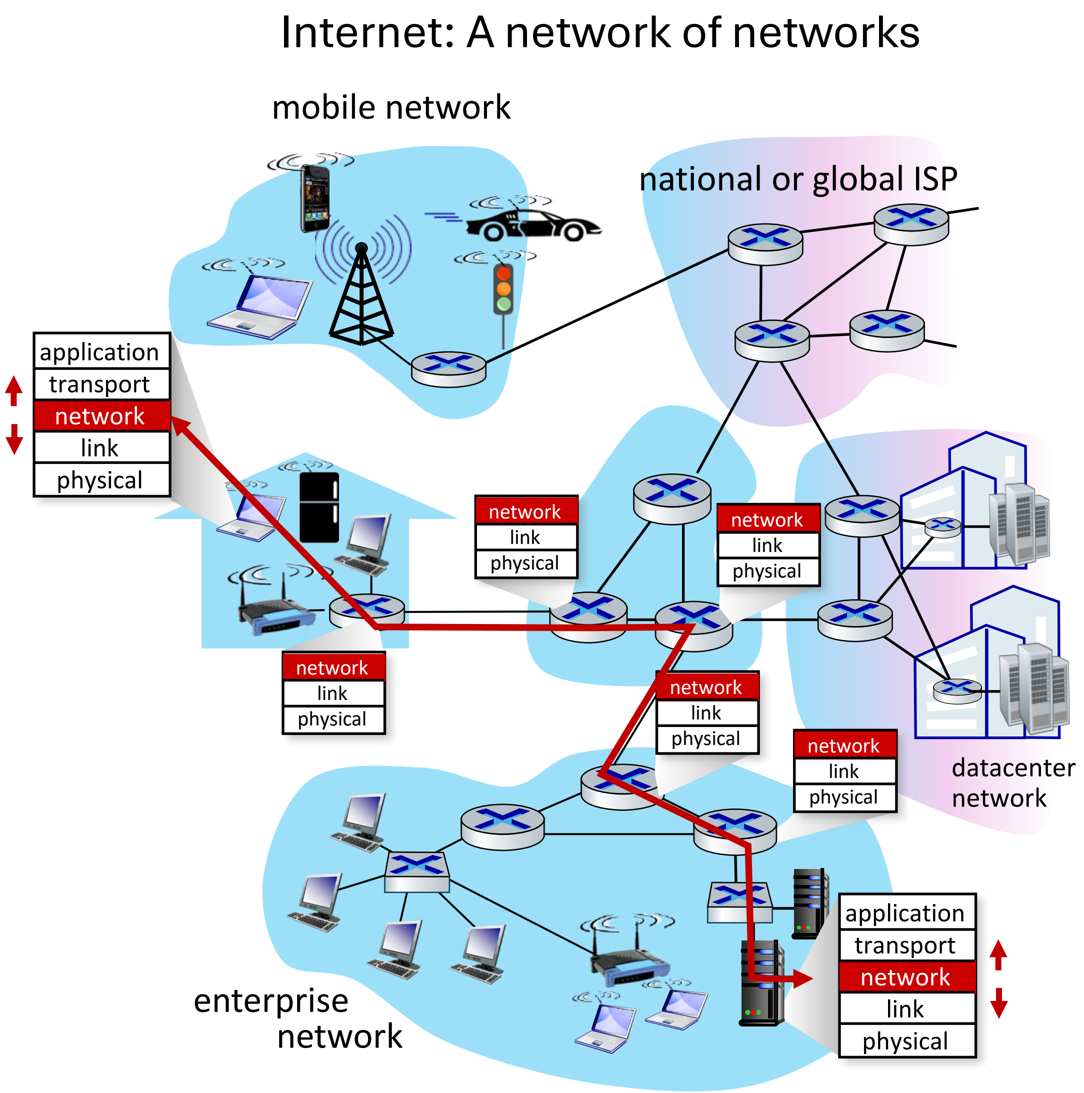
- Function:
  - Route packets end-to-end on a network, through multiple hops
  - Ties the entire protocol stack together!
  - Only one protocol: **Internet Protocol (IP)**
- Key challenge:
  - How to represent addresses
  - How to route packets
    - Scalability
    - Convergence

Network layer: overview

Data plane and control plane

Network-layer services and protocols

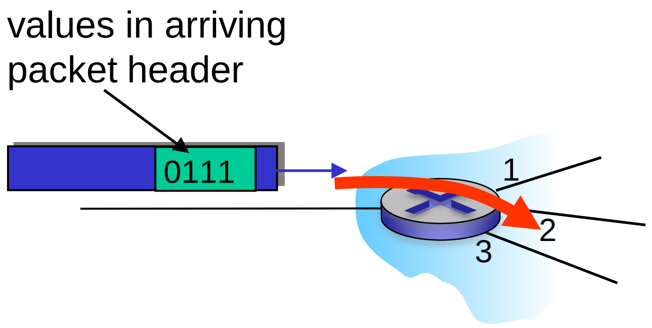
- transport segment from sending to receiving host
  - sender: encapsulates segments into datagrams, passes to link layer
  - receiver: delivers segments to transport layer protocol
- routers:
  - *forwarding:* move packets from a router's input link to appropriate router output link
  - *routing:* determine route taken by packets from source to destination
    - *routing algorithms*



Network layer: data plane, control plane

Data plane:

- *local*, per-router function
- determines how datagram arriving on router input port is forwarded to router output port
- Forwards packets based on the built logic of the Control Plane



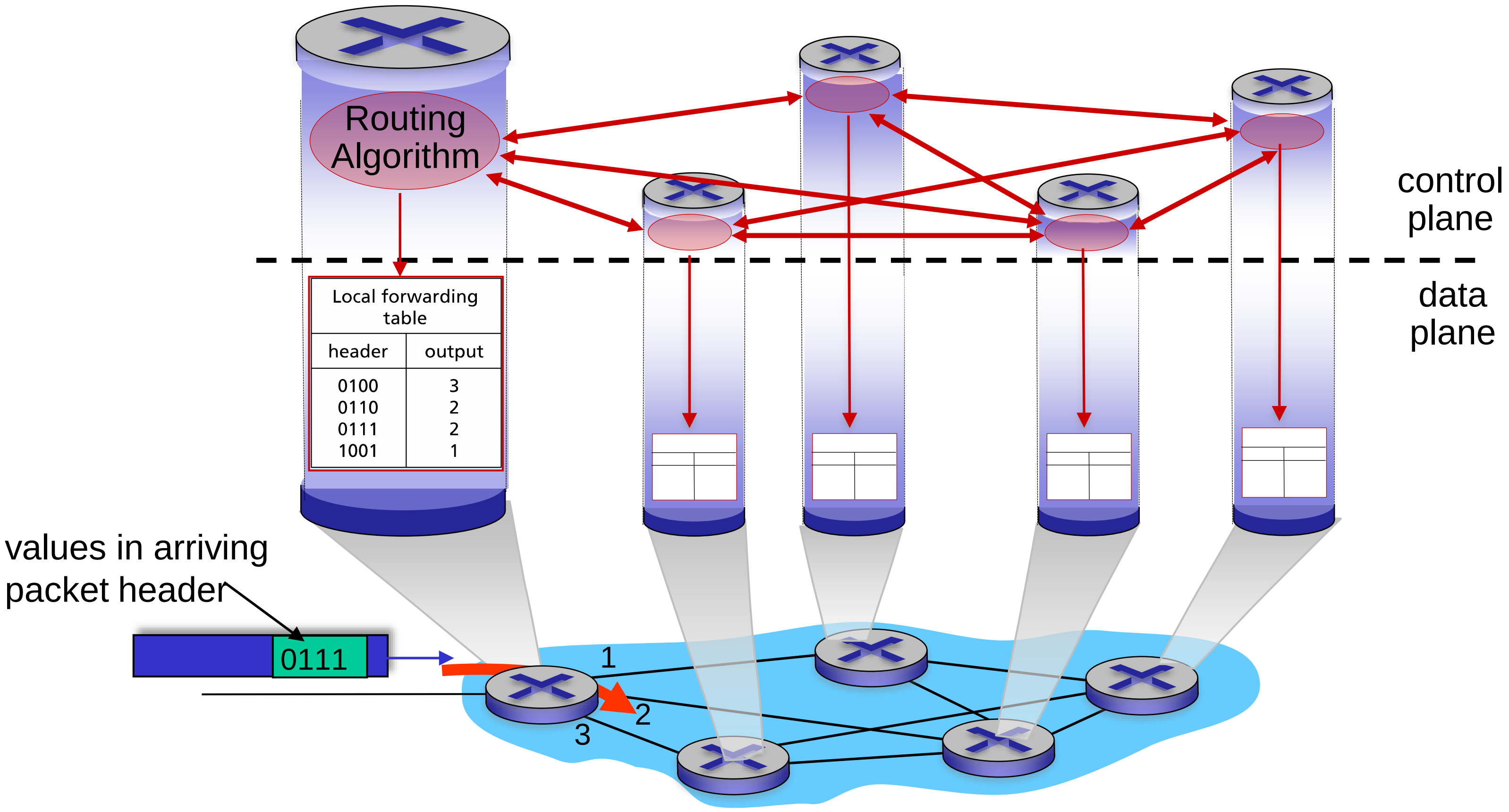
Control plane

- *network-wide* logic
  - determines how datagram is routed among routers along end-end path.
  - Packets are processed by the router to update the routing table

- two control-plane approaches:
  - *traditional routing algorithms:* implemented in routers
  - *software-defined networking (SDN)*: implemented in (remote) servers

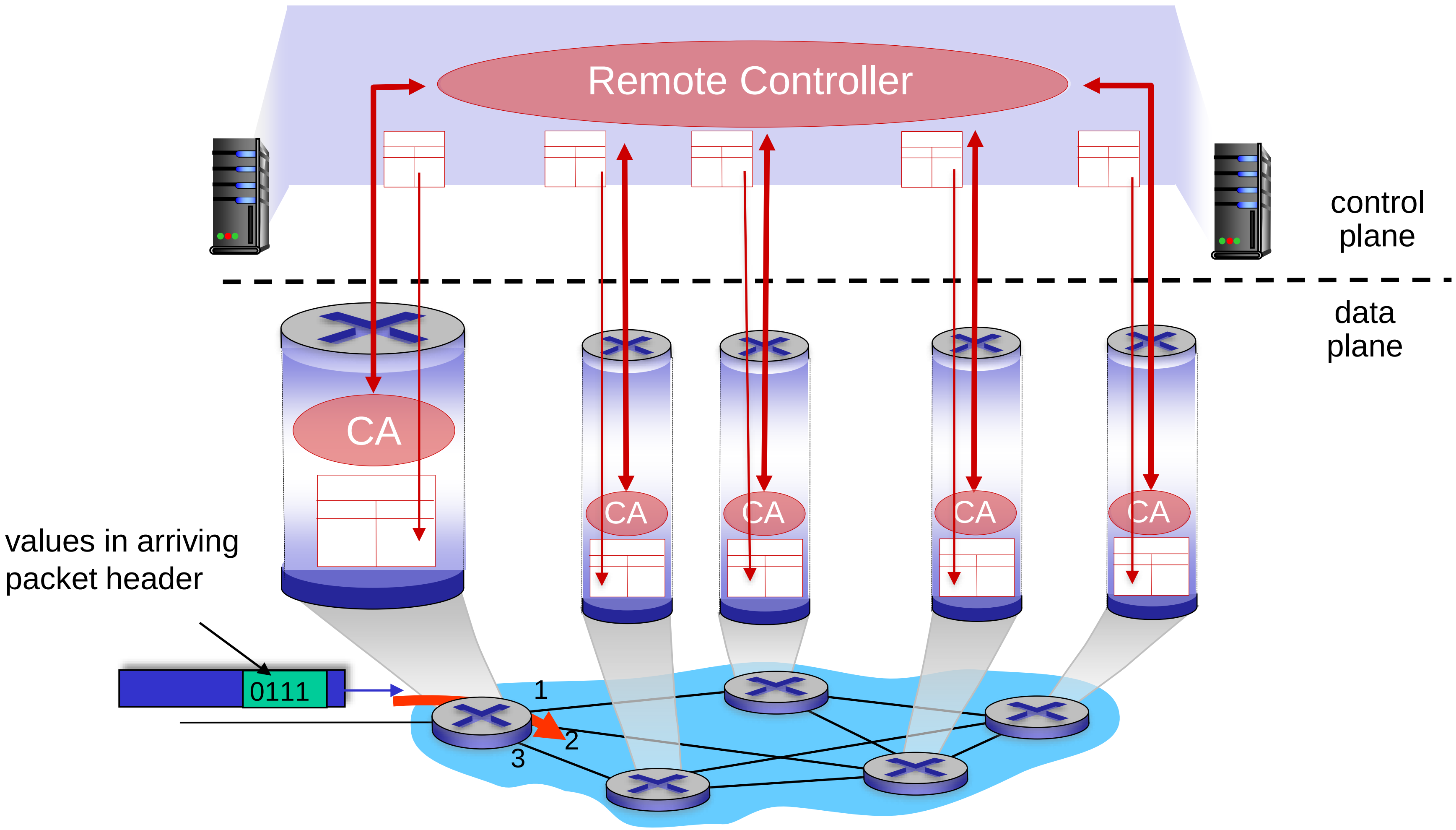
Per-router control plane

Individual routing algorithm components *in each and every router* interact in the control plane to establish a forwarding table.

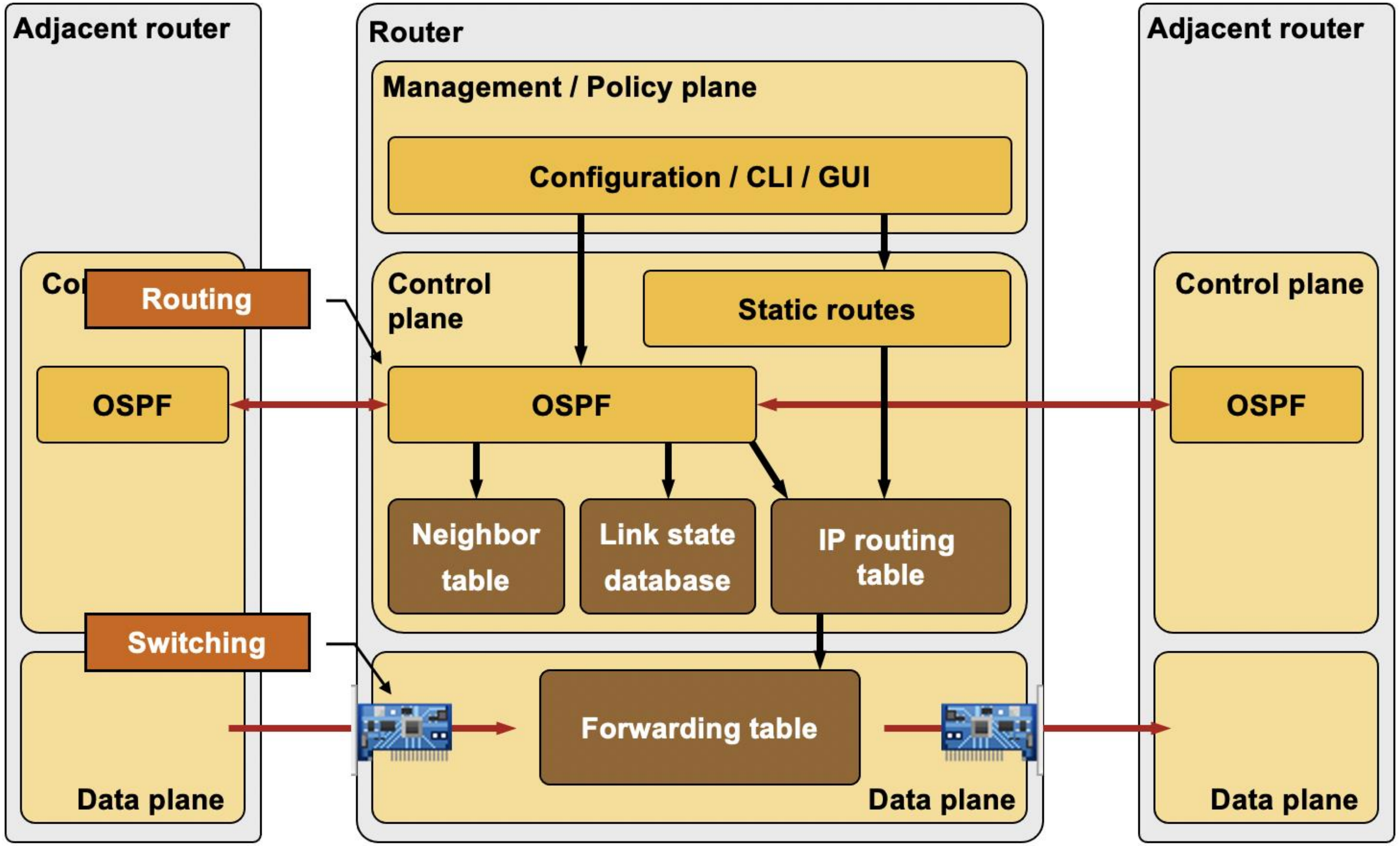


Software-Defined Networking (SDN) control plane

Remote controller computes, installs forwarding tables in routers



Three Planes, Different Altitudes

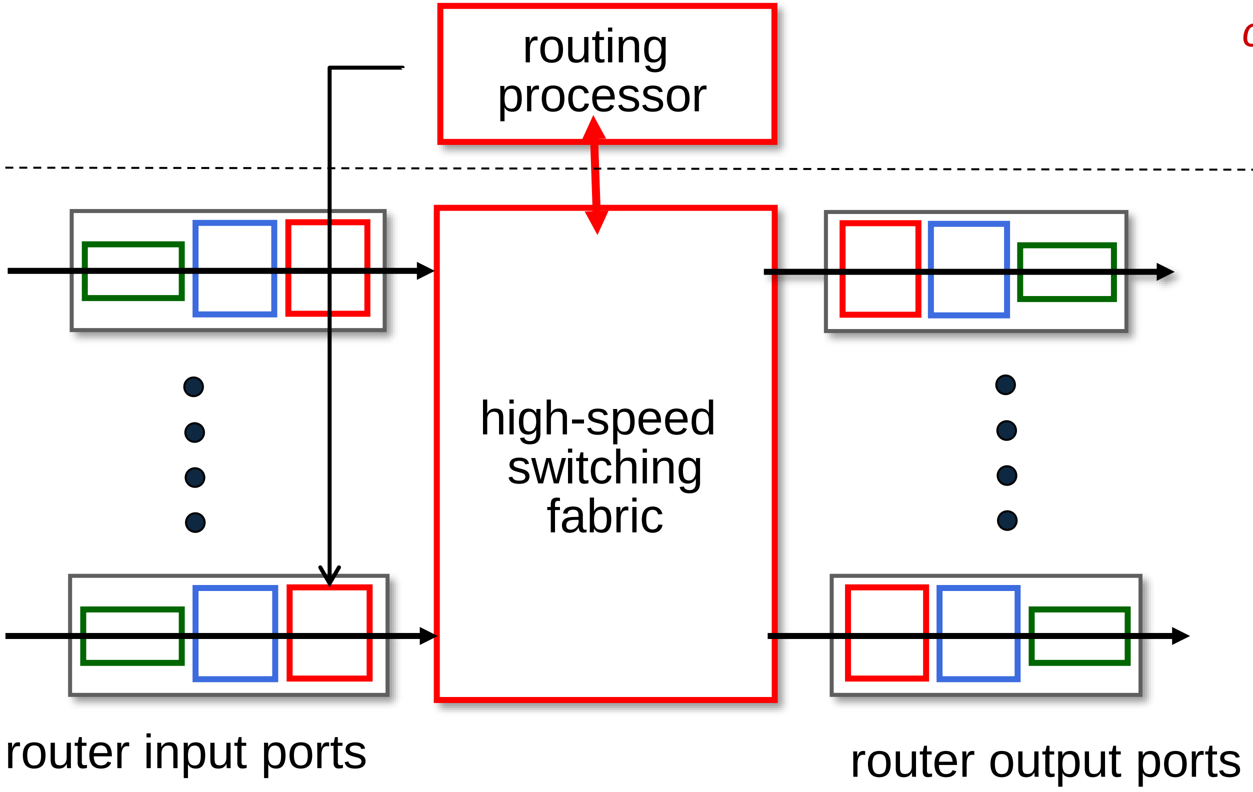


What’s inside a router

Input/Output ports, switching, buffer management, scheduling

Router architecture overview

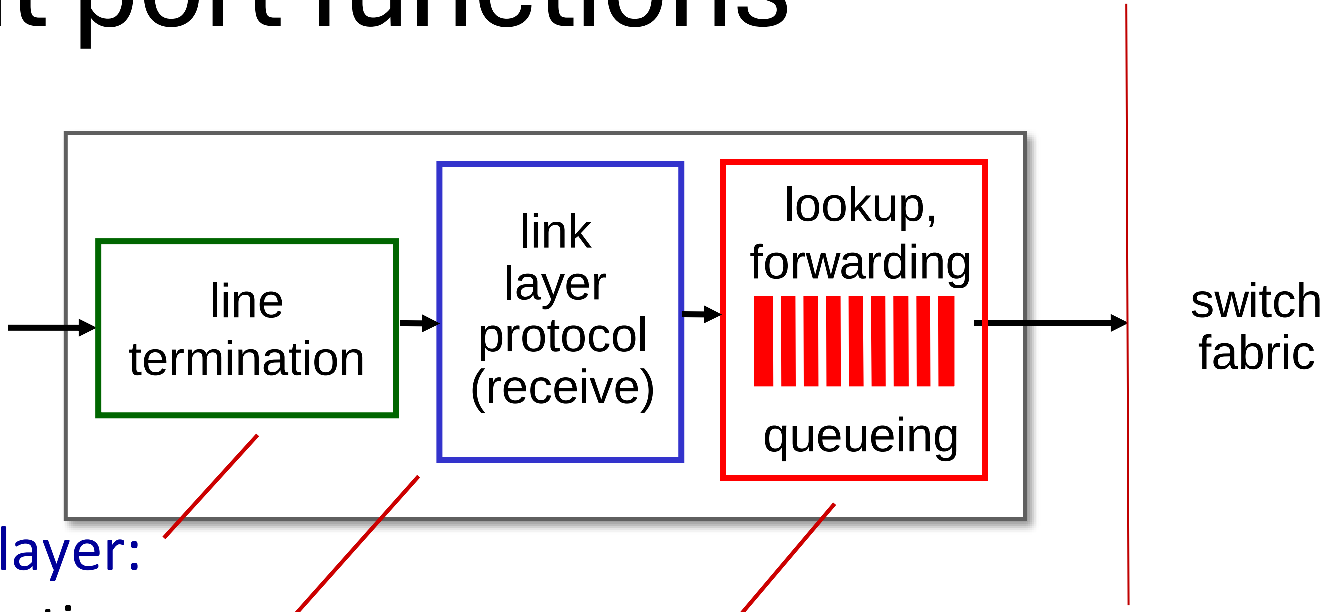
high-level view of generic router architecture:



routing, management
control plane (software)
operates in millisecond
time frame

forwarding data plane
(hardware) operates
in nanosecond
timeframe

Input port functions



physical layer:
bit-level reception

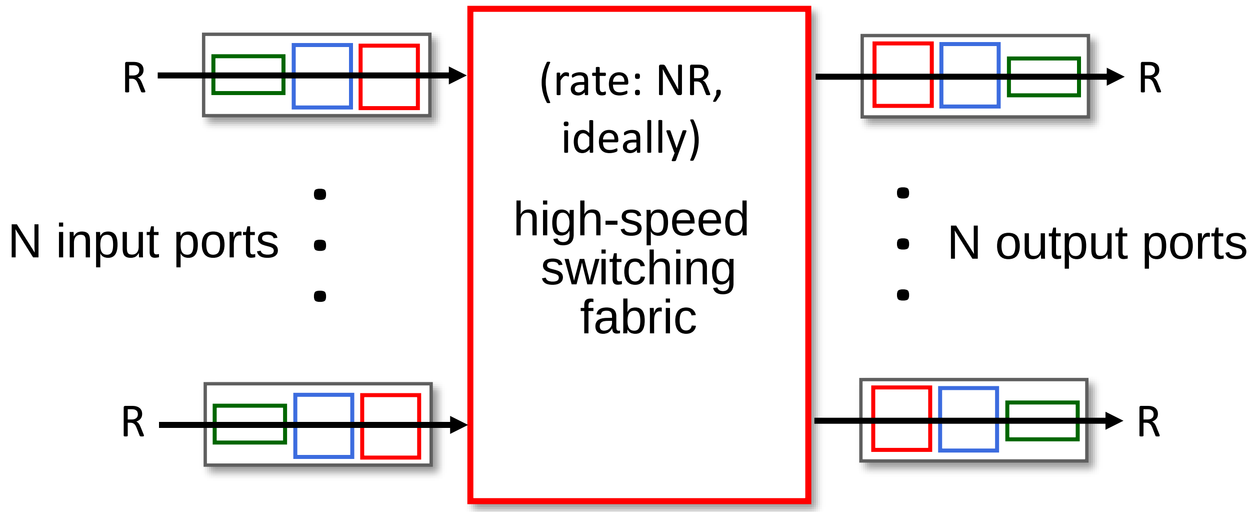
link layer:
e.g., Ethernet
(chapter 6)

decentralized switching:

- using header field values, lookup output port using forwarding table in input port memory *("match plus action")*
- goal: complete input port processing at 'line speed'
- input port queuing: if datagrams arrive faster than forwarding rate into switch fabric

Switching fabrics

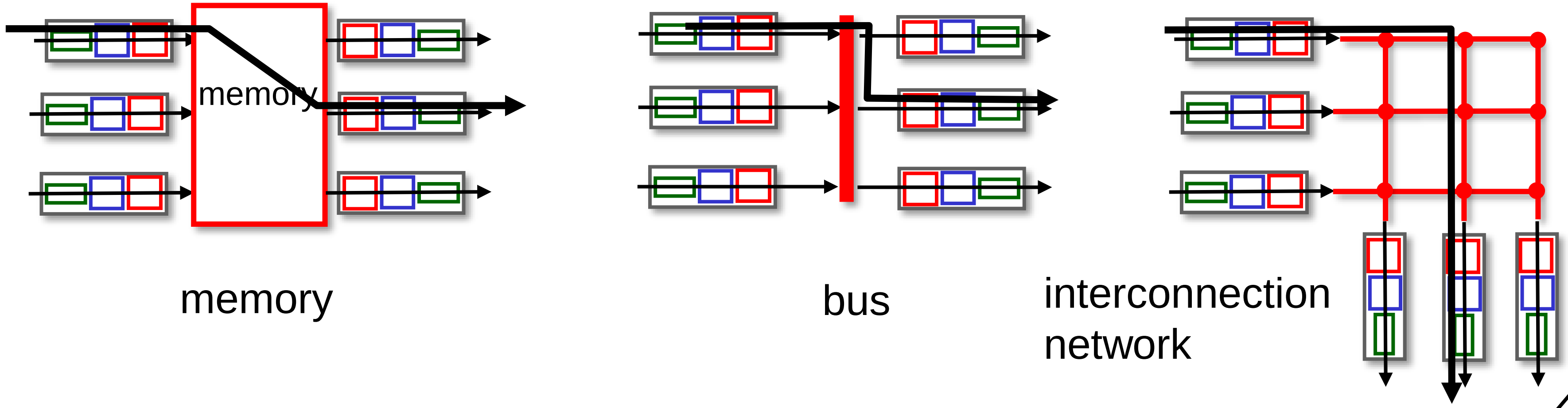
- transfer packet from input link to appropriate output link
- switching rate: rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable



Switching fabrics

- datagram from input port memory to output port memory via a shared bus

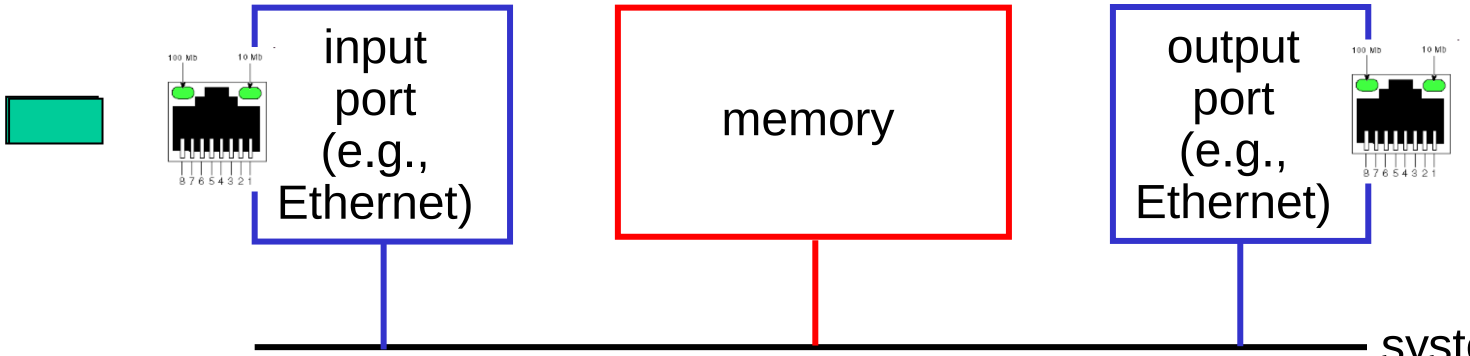
- three major types of switching fabrics:



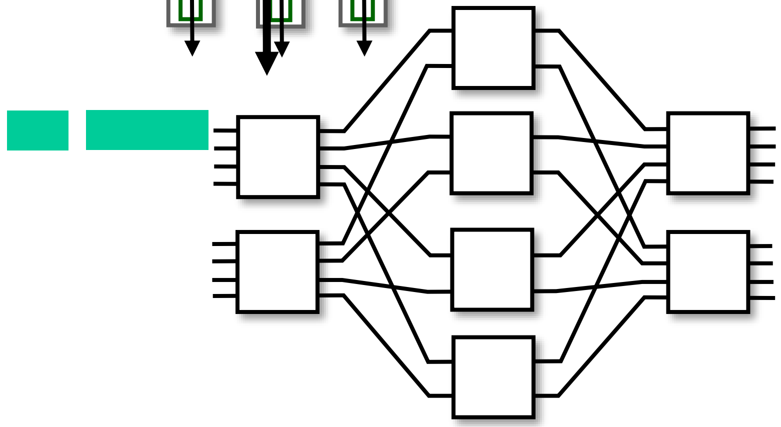
memory

bus

interconnection network



system bus

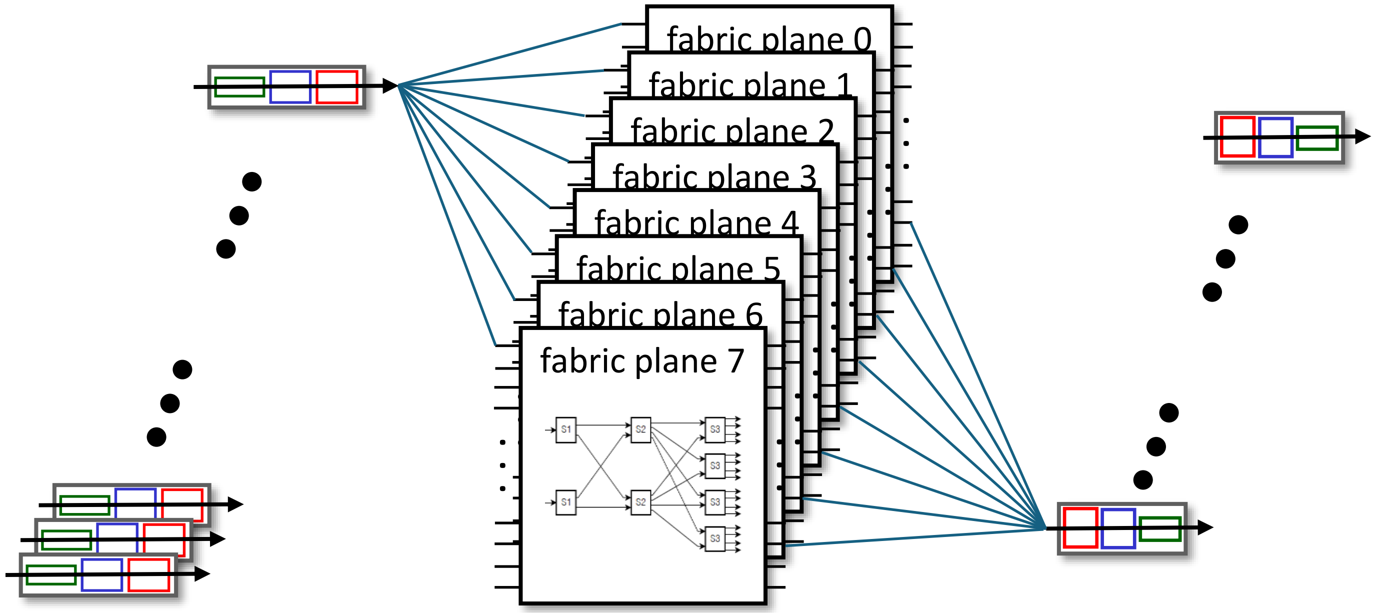


8x8 multistage switch
built from smaller-sized switches

Switching via interconnection network

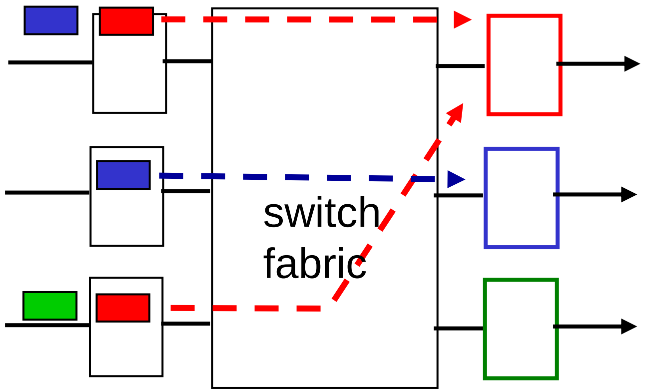
- scaling, using multiple switching “planes” in parallel:
  - speedup, scaleup via parallelism

- Cisco CRS router:
  - basic unit: 8 switching planes
  - each plane: 3-stage interconnection network
  - up to 100’s Tbps switching capacity

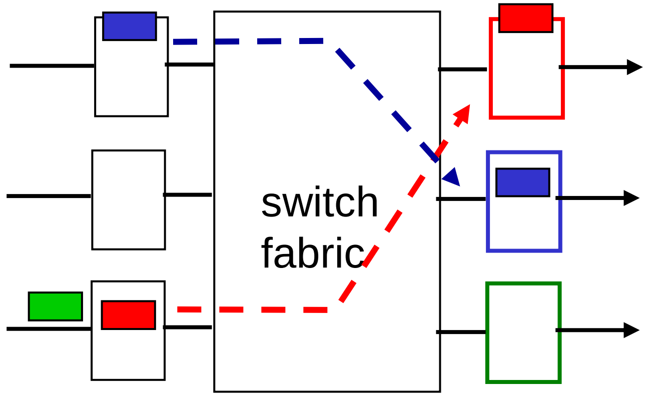


Input port queuing

- If switch fabric slower than input ports combined -> queueing may occur at input queues
  - queueing delay and loss due to input buffer overflow!
- Head-of-the-Line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward



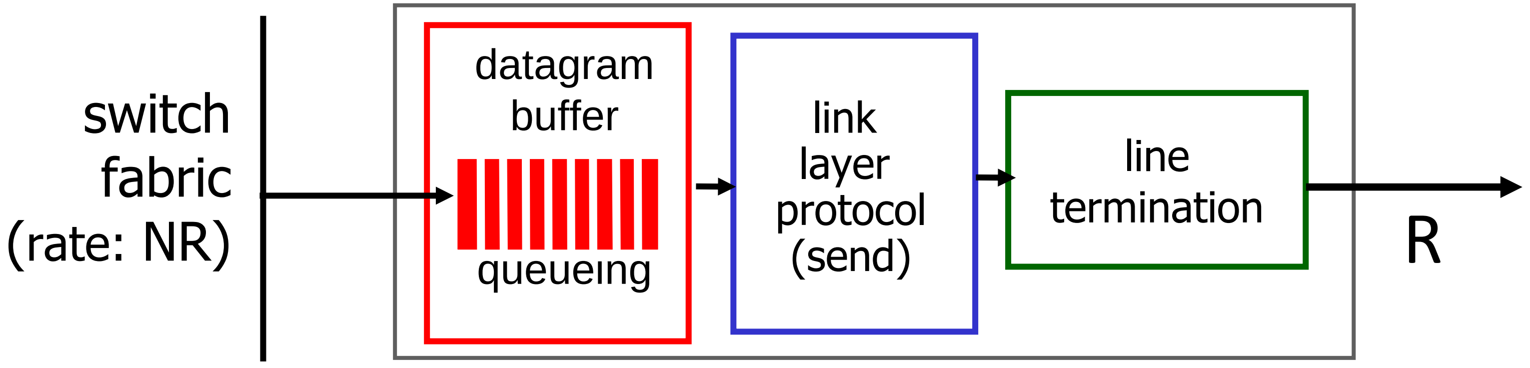
output port contention: only one red datagram can be transferred. lower red packet is *blocked*



one packet time later: green packet experiences HOL blocking

Output port queuing

This is a really important slide



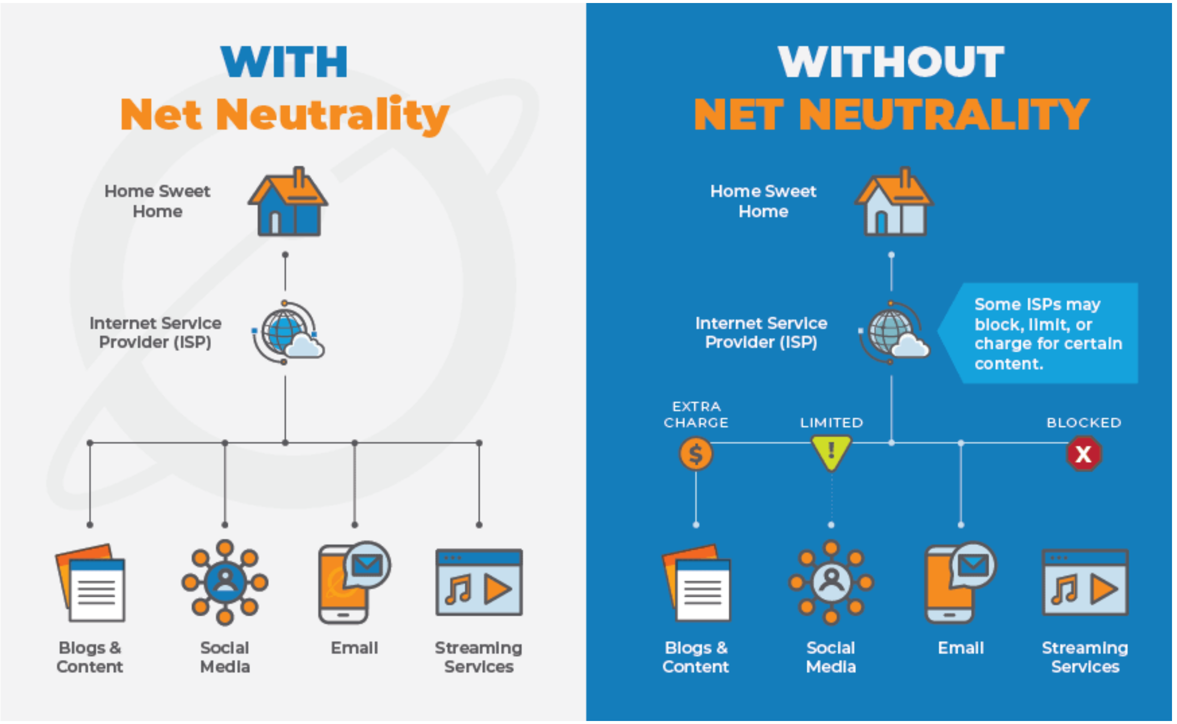
- *Buffering* required when datagrams arrive from fabric faster than link transmission rate. *Drop policy:* which datagrams to drop if no free buffers?

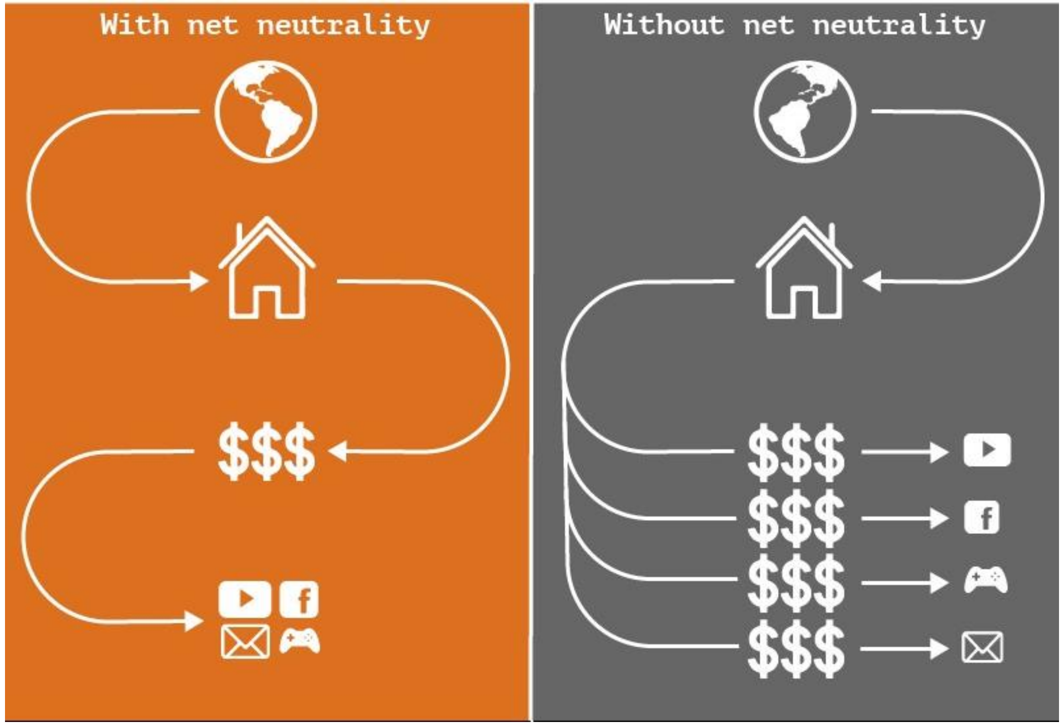
Datagrams can be lost due to congestion, lack of buffers

- *Scheduling discipline* chooses among queued datagrams for transmission

Priority scheduling – who gets best performance, **network neutrality**

Network Neutrality





Sidebar: Network Neutrality

2015 US FCC *Order on Protecting and Promoting an Open Internet:* three "clear, bright line" rules:

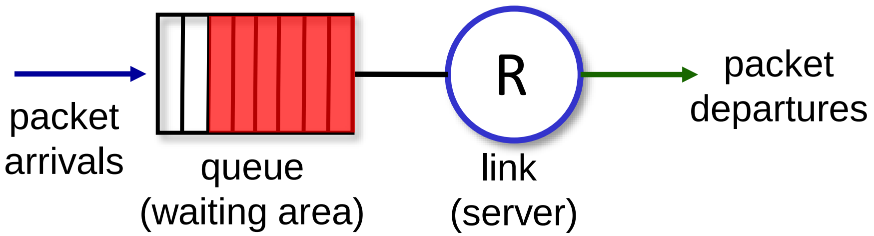
- no blocking … "shall not block lawful content, applications, services, or non-harmful devices, subject to reasonable network management."
- no throttling … "shall not impair or degrade lawful Internet traffic on the basis of Internet content, application, or service, or use of a non-harmful device, subject to reasonable network management."
- no paid prioritization. … "shall not engage in paid prioritization"

Packet Scheduling: FCFS

packet scheduling: deciding which packet to send next on link

- first come, first served
- priority
- round robin
- weighted fair queueing

Abstraction: queue



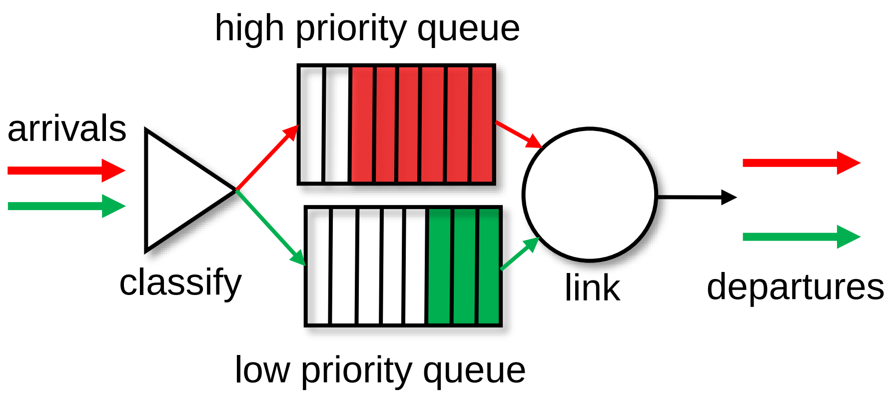
FCFS: packets transmitted in order of arrival to output port

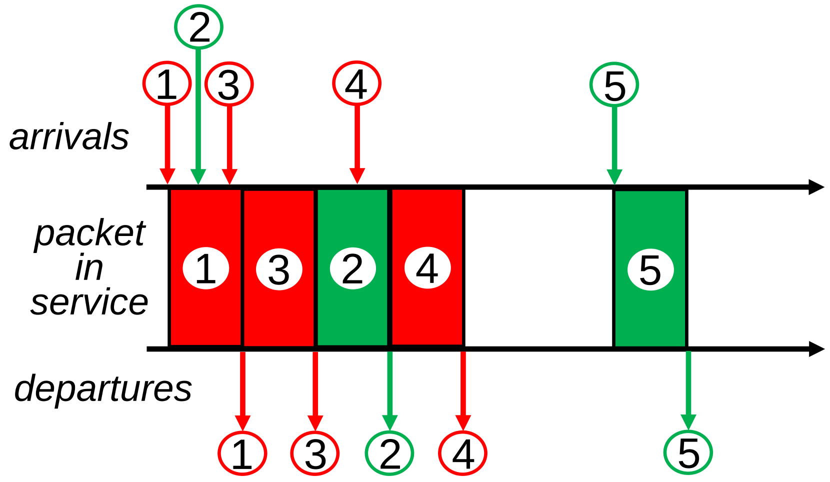
- also known as: First-in-first-out (FIFO)

Scheduling policies: priority

Priority scheduling:

- arriving traffic classified, queued by class
  - any header fields can be used for classification
- send packet from highest priority queue that has buffered packets
  - FCFS within priority class



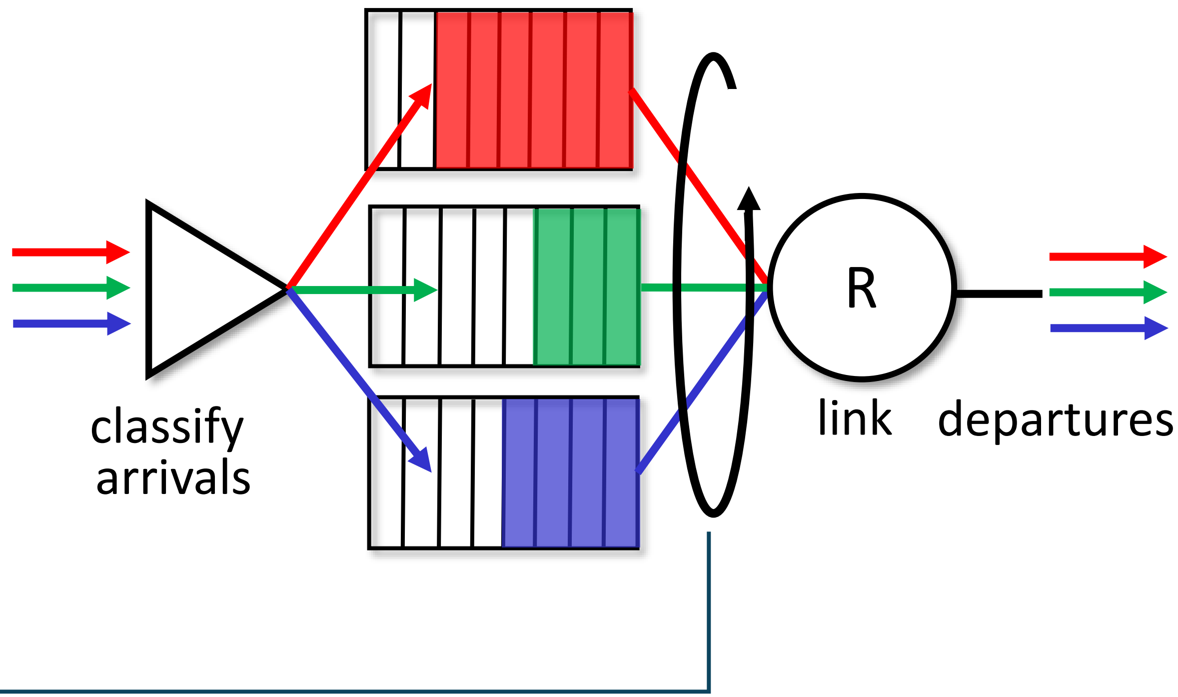


Scheduling policies: round robin

Round Robin (RR) scheduling:

- arriving traffic classified, queued by class
  - any header fields can be used for classification

- server cyclically, repeatedly scans class queues, sending one complete packet from each class (if available) in turn



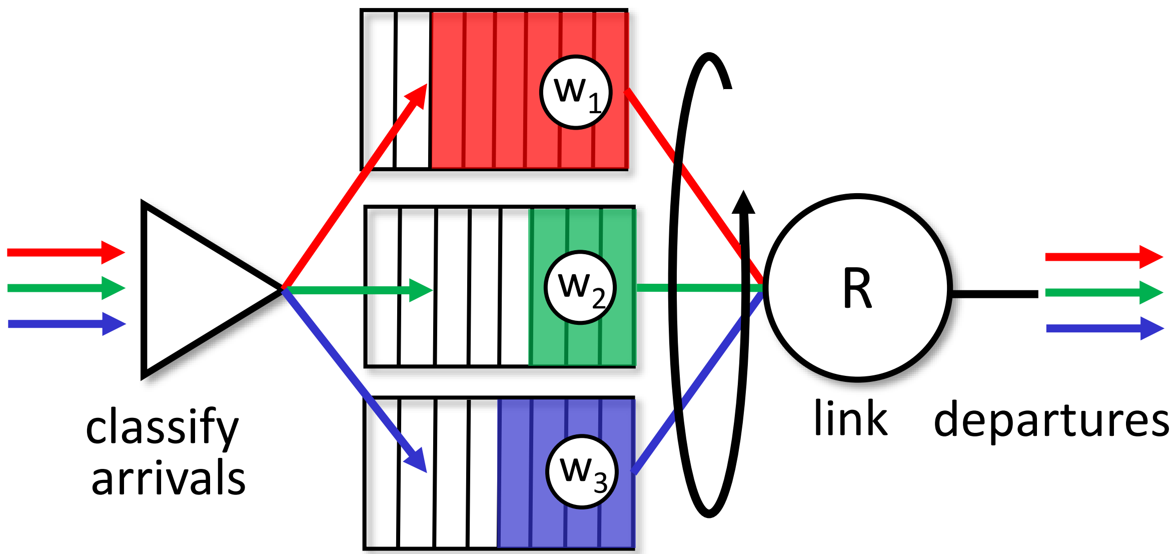
Scheduling policies: weighted fair queueing

Weighted Fair Queuing (WFQ):

- generalized Round Robin
  - each class, i, has weight, w_i, and gets weighted amount of service in each cycle:

$$\frac{w_i}{\Sigma_j w_j}$$

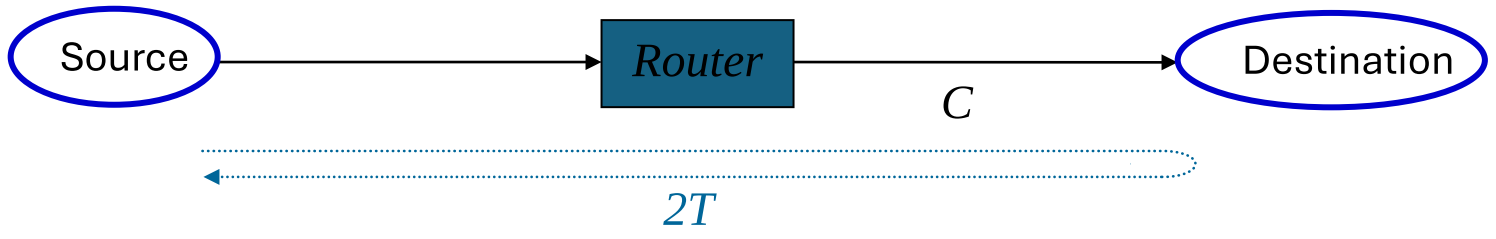
  - minimum bandwidth guarantee (per-traffic-class)



How much Buffering?

Rule of thumb, large or small?

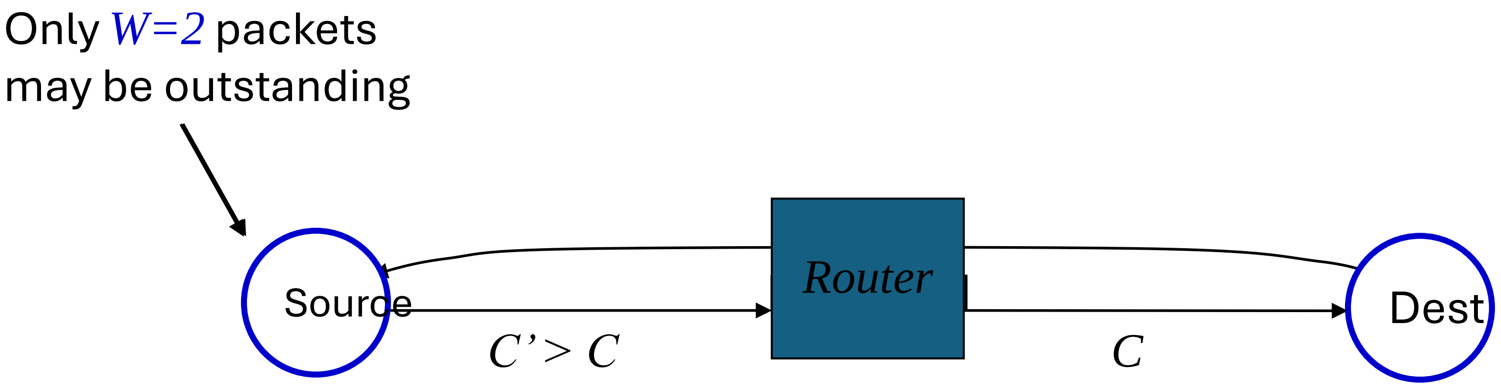
How much Buffer does a Router need?



- Use to be Universally applied rule-of-thumb:
  - A router needs a buffer size: $\boxed{B = 2T \times C}$
    - 2T is the two-way propagation delay
    - C is capacity of bottleneck link
- Where does the rule of thumb comes from? (Answer: TCP)

- Context
  - Appears in IETF architectural guidelines.
  - Usually referenced to Villamizar and Song: “High Performance TCP in ANSNET”, CCR, 1994.
  - Already known by inventors of TCP [Van Jacobson, 1988]
  - Has major consequences for router design

TCP

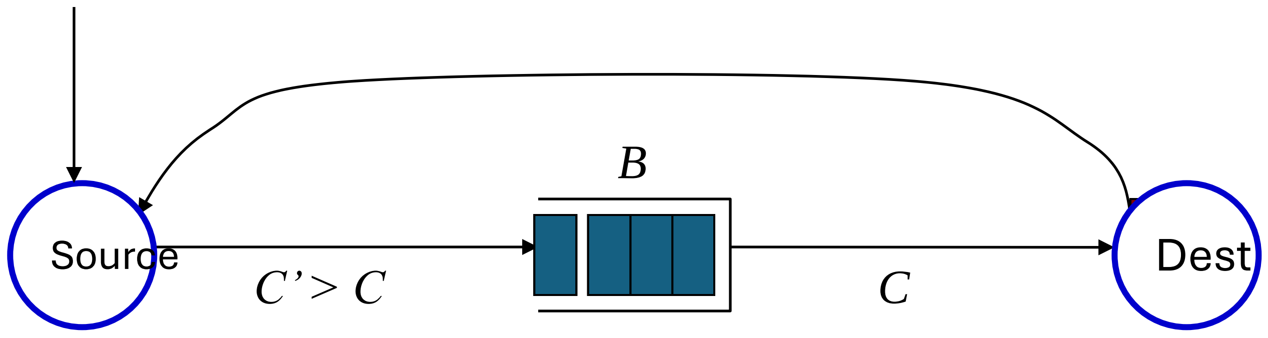


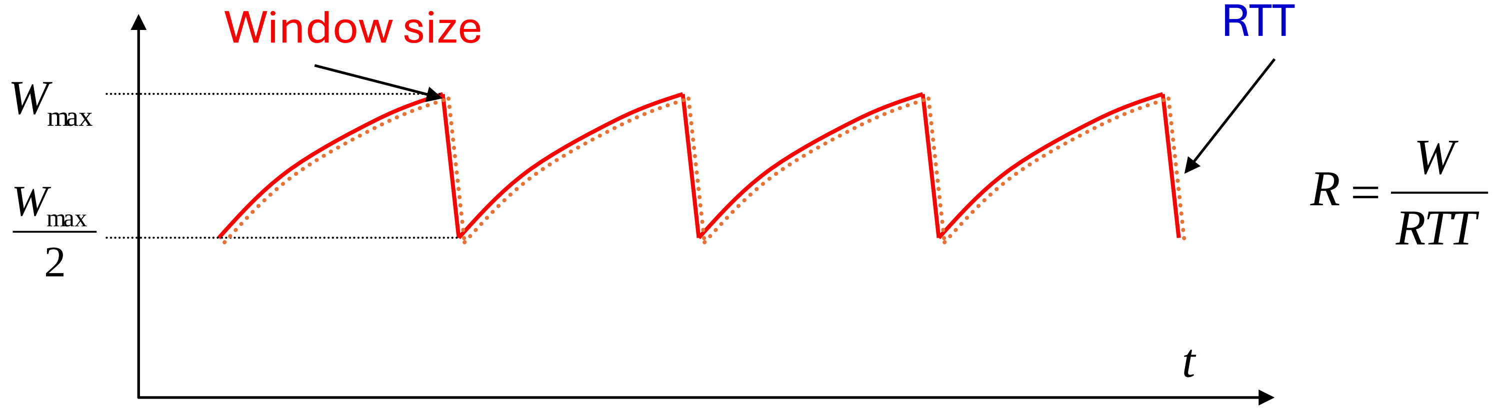
- TCP Congestion Window controls the sending rate
  - Sender sends packets, receiver sends ACKs
  - Sending rate is controlled by Window W,
  - At any time, only W unacknowledged packets may be outstanding
  - The sending rate of TCP is $\boxed{R = \frac{W}{RTT}}$

Single TCP Flow

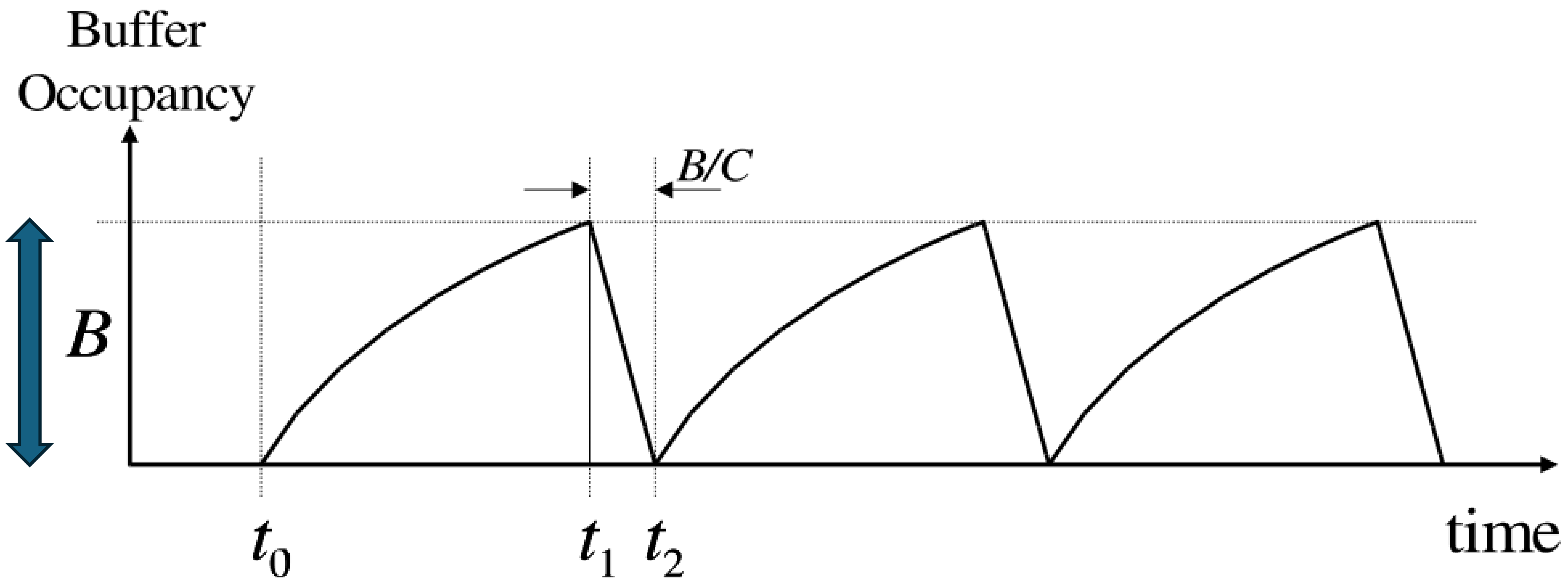
Router with large enough buffers for full link utilization

For every W ACKs received,
send W+1 packets





Required buffer is height of sawtooth



When the sender first pauses at t_1, the buffer is full, and so it drains over a period B/C until t_2

Origin of rule-of-thumb

- Before and after reducing window size, the sending rate of the TCP sender is the same

$$R_{old} = R_{new}$$

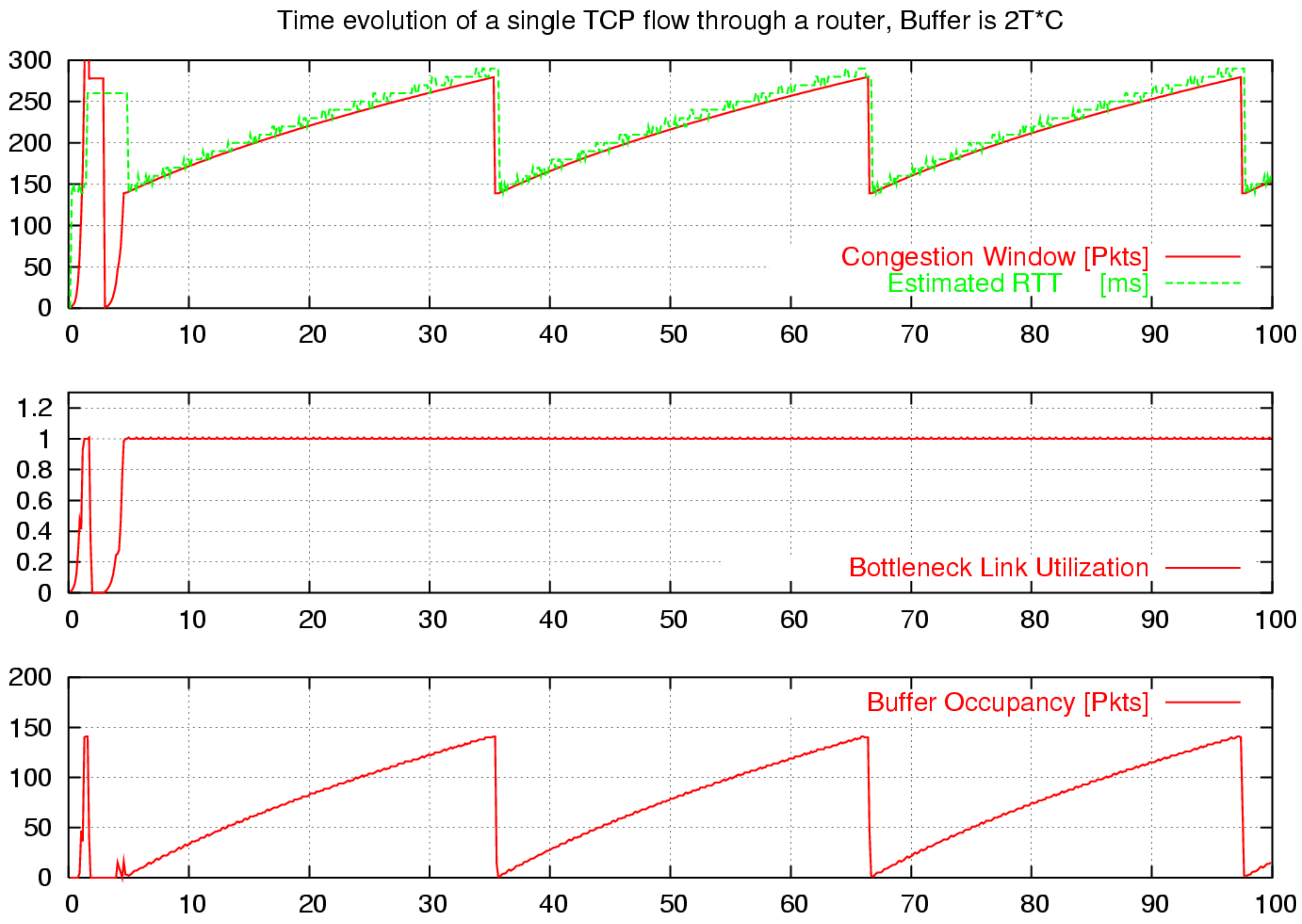
- Inserting the rate equation we get

$$\frac{W_{old}}{RTT_{old}} = \frac{W_{new}}{RTT_{new}}$$

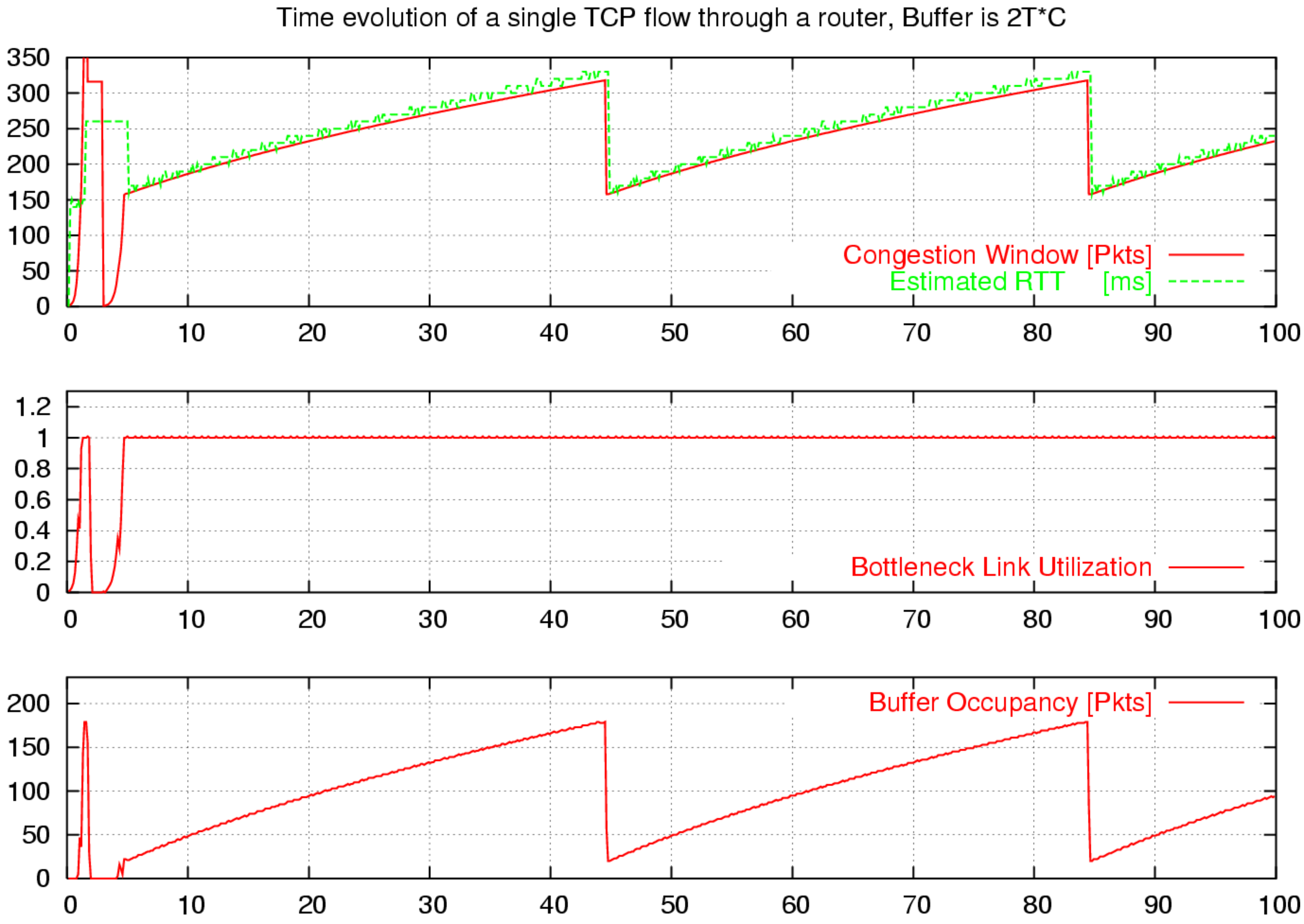
- The RTT is part transmission delay T and part queueing delay B/C . We know that after reducing the window, the queueing delay is zero.

$$\frac{W_{old}}{2T + B/C} = \frac{W_{old}/2}{2T} \quad \Leftrightarrow \quad \boxed{B = 2T \times C}$$

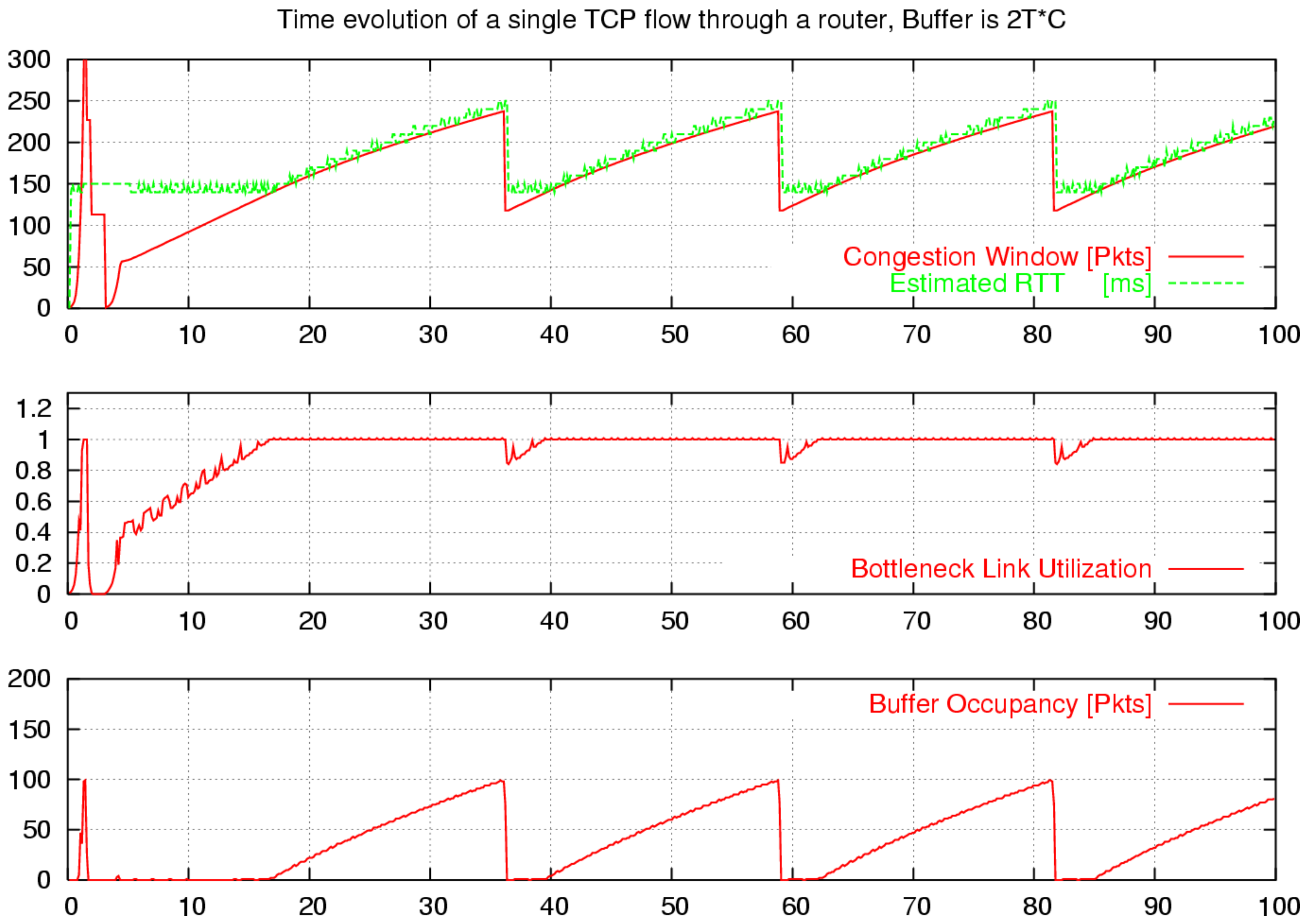
Buffer = rule of thumb



Over-buffered Link



Under-buffered Link

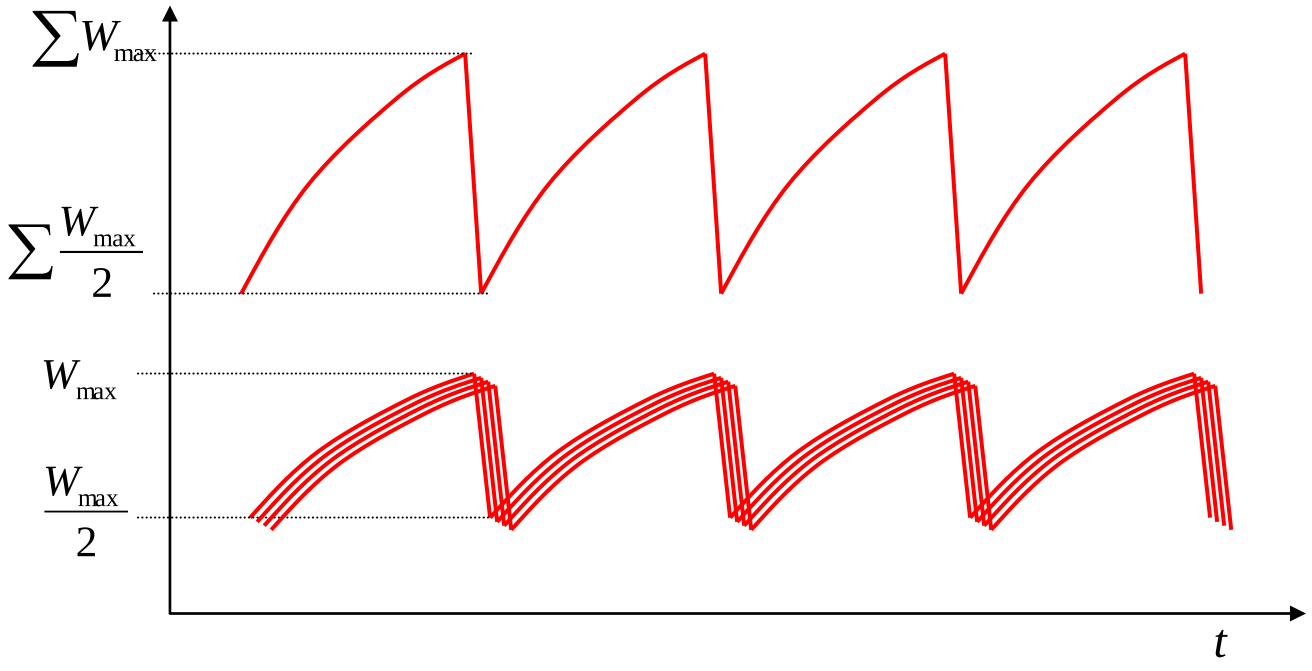


Rule-of-thumb

- Rule-of-thumb makes sense for one flow
- Typical backbone link has > 20,000 flows
- Does the rule-of-thumb still hold?

- Answer:
  - If flows are perfectly synchronized, then Yes.
  - If flows are desynchronized then No.

If flows are synchronized

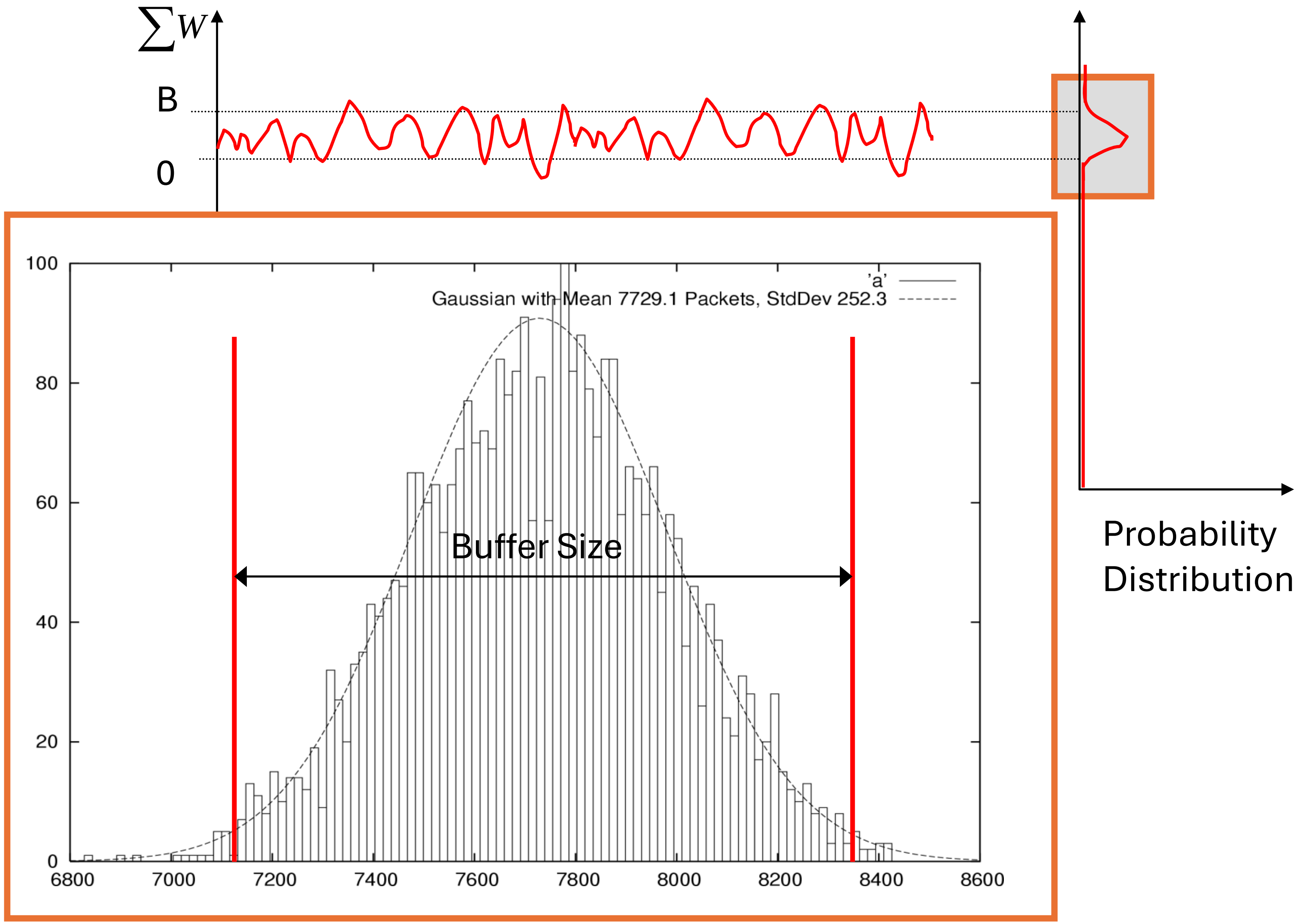


- Aggregate window has same dynamics
- Therefore buffer occupancy has same dynamics
- Rule-of-thumb still holds.

When are Flows Synchronized?

- Small numbers of flows tend to synchronize
- Large aggregates of flows are not synchronized
  - For > 200 flows, synchronization disappears
  - Measurements in the core give no indication of synchronization

If flows are not synchronized



Quantitative Model

- Model congestion window of a flow as random variable

$$W_i(t) \quad \text{model as} \quad W_i \quad \text{where} \quad P[W_i = x] = f(x)$$

- For many de-synchronized flows
  - We know congestion windows are independent
  - All congestion windows have the same probability distribution

$$E[W_i] = \mu_W \qquad \qquad \mathrm{var}[W_i] = \sigma_W^2$$

- Now central limit theorem gives us the distribution of the sum of the window sizes

$$\sum_n W_i(t) \;\rightarrow\; n\mu_W + \sqrt{n}\sigma_W N(0,1)$$

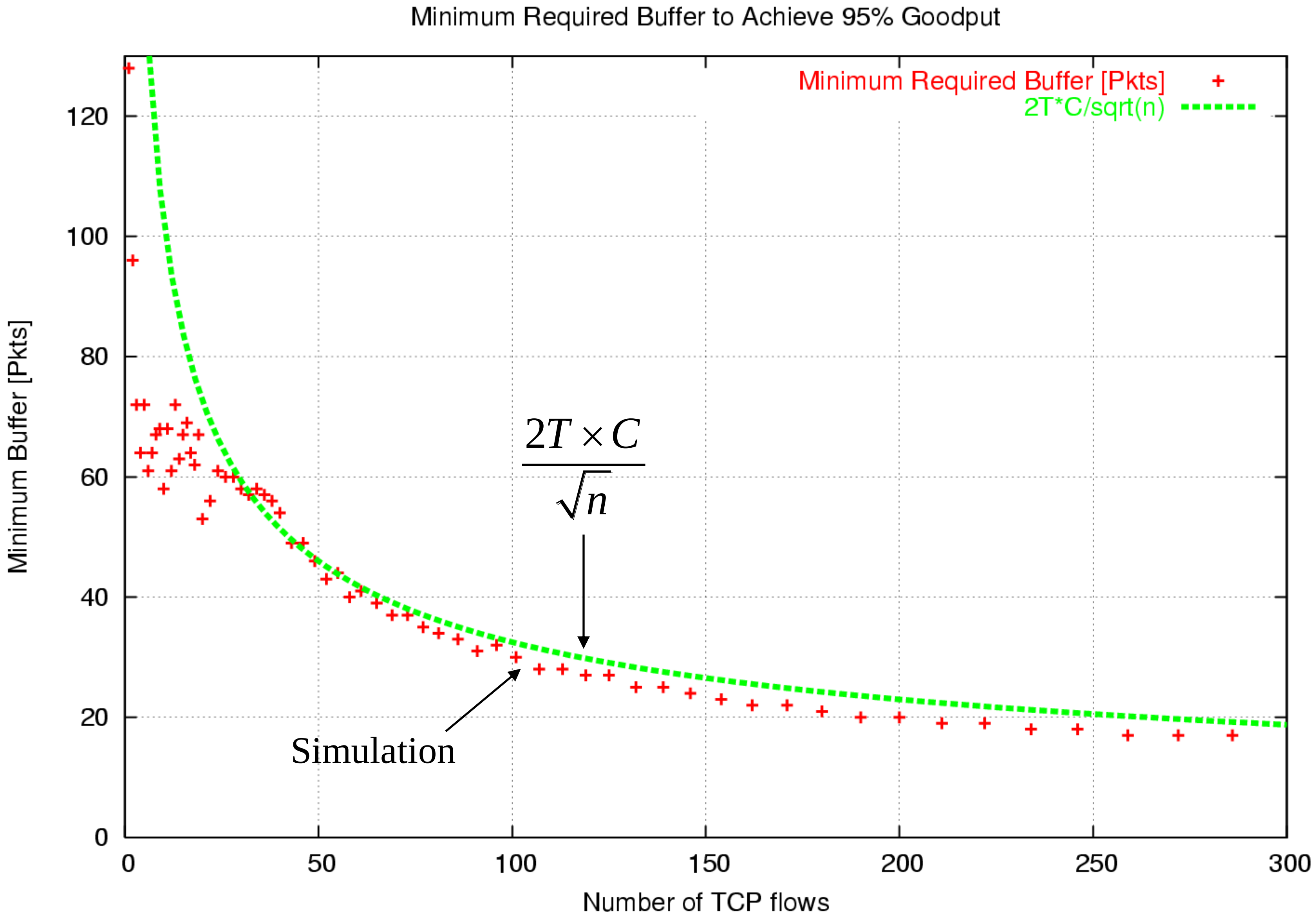
Central Limit Theorem

- CLT tells us that the more variables (Congestion Windows of Flows) we have, the narrower the Gaussian (Fluctuation of sum of windows)
  - Width of Gaussian decreases with $\frac{1}{\sqrt{n}}$
  - Buffer size should also decreases with $\frac{1}{\sqrt{n}}$

$$B \rightarrow \frac{B_{n=1}}{\sqrt{n}} = \frac{2T \times C}{\sqrt{n}}$$

https://dl.acm.org/doi/10.1145/1030194.1015499

Required buffer size



In summary

- Flows in the core are desynchronized

- For desynchronized flows, congested routers need only buffers of

$$\boxed{B = \frac{2T \times C}{\sqrt{n}}}$$

Buffer requirements for short flows

- So far we were assuming a congested router with long flows in congestion avoidance mode.
  - What about flows in slow start?
  - Do buffer requirements differ?

- Answer: Yes, however:
  - Required buffer in such cases is independent of line speed and RTT (same for 1Mbit/s or 40 Gbit/s)
  - In mixes of flows, long flows drive buffer requirements
    - Short flow result relevant for uncongested routers

Long Flows – Utilization (II)
Model vs. ns2 vs. Physical Router

GSR 12000, OC3 Line Card

TCP Flows	Router Buffer			Link Utilization		
	$\frac{2T \times C}{\sqrt{n}}$	Pkts	RAM	Model	Sim	Exp
100	0.5 x	64	1Mb	96.9%	94.7%	94.9%
	1 x	129	2Mb	99.9%	99.3%	98.1%
	2 x	258	4Mb	100%	99.9%	99.8%
	3 x	387	8Mb	100%	99.8%	99.7%
400	0.5 x	32	512kb	99.7%	99.2%	99.5%
	1 x	64	1Mb	100%	99.8%	100%
	2 x	128	2Mb	100%	100%	100%
	3 x	192	4Mb	100%	100%	99.9%

Impact on Router Design

- 10Gb/s linecard with 200,000 x 56kb/s flows
  - Rule-of-thumb: Buffer = 2.5Gbits
    - Requires external, slow DRAM
  - Becomes: Buffer = 6Mbits
    - Can use on-chip, fast SRAM
    - Completion time halved for short-flows
- 40Gb/s linecard with 40,000 x 1Mb/s flows
  - Rule-of-thumb: Buffer = 10Gbits
  - Becomes: Buffer = 50Mbits

- For more details…
  - "Sizing Router Buffers – Guido Appenzeller, Isaac Keslassy and Nick McKeown, to appear at SIGCOMM 2004 https://dl.acm.org/doi/10.1145/1030194.1015499

IP: the Internet Protocol

datagram format, addressing, fragmentation

IP Datagram format

IP protocol version number

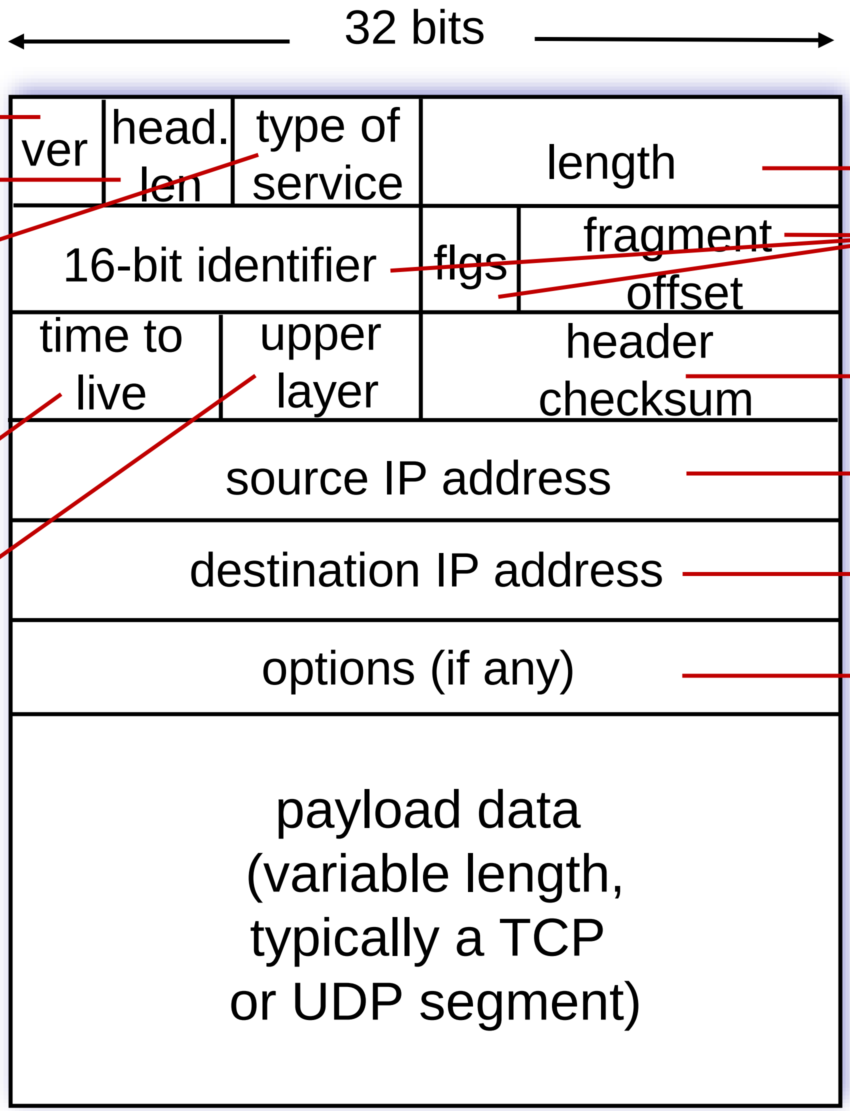
header length(bytes)

“type” of service:

- diffserv (0:5)
- ECN (6:7)

TTL: remaining max hops
(decremented at each router, trace route)

upper layer protocol (e.g., TCP or UDP)



total datagram length (bytes)

fragmentation/ reassembly

header checksum

32-bit source IP address

Maximum length: 64K bytes
Typically: 1500 bytes or less

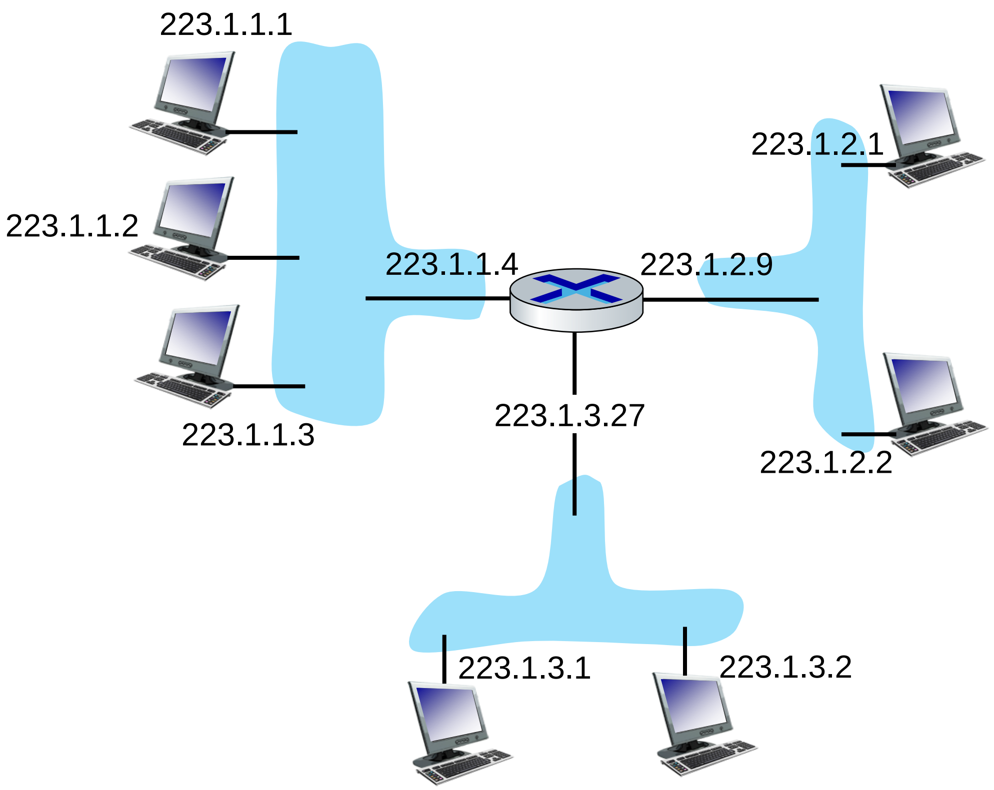
e.g., timestamp, record route taken

overhead

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead for TCP+IP

IP addressing: introduction

- IP address: 32-bit identifier associated with each host or router *interface*
- interface: connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)



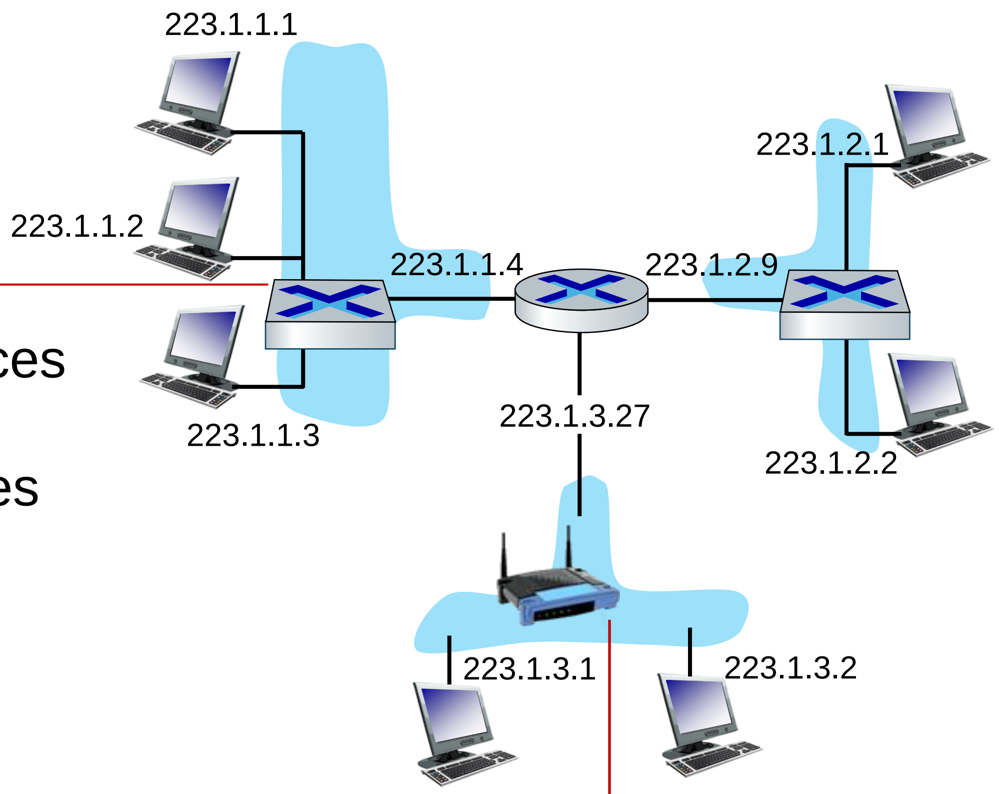
dotted-decimal IP address notation:

$$223.1.1.1 = \underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}$$

IP addressing: introduction

Q: how are interfaces actually connected?

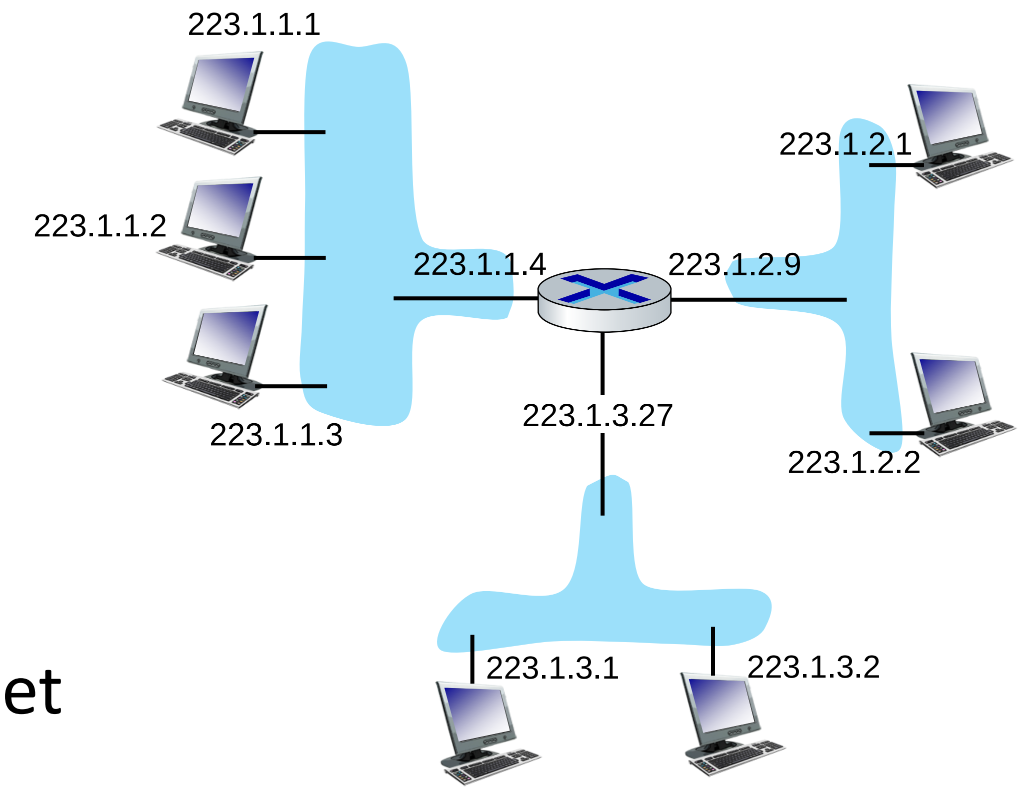
A: wired Ethernet interfaces connected by Ethernet switches



A: wireless WiFi interfaces connected by WiFi base station

Subnets

- *What's a subnet ?*
  - device interfaces that can physically reach each other without passing through an intervening router
- IP addresses have structure:
  - subnet part: devices in same subnet have common high order bits
  - host part: remaining low order bits

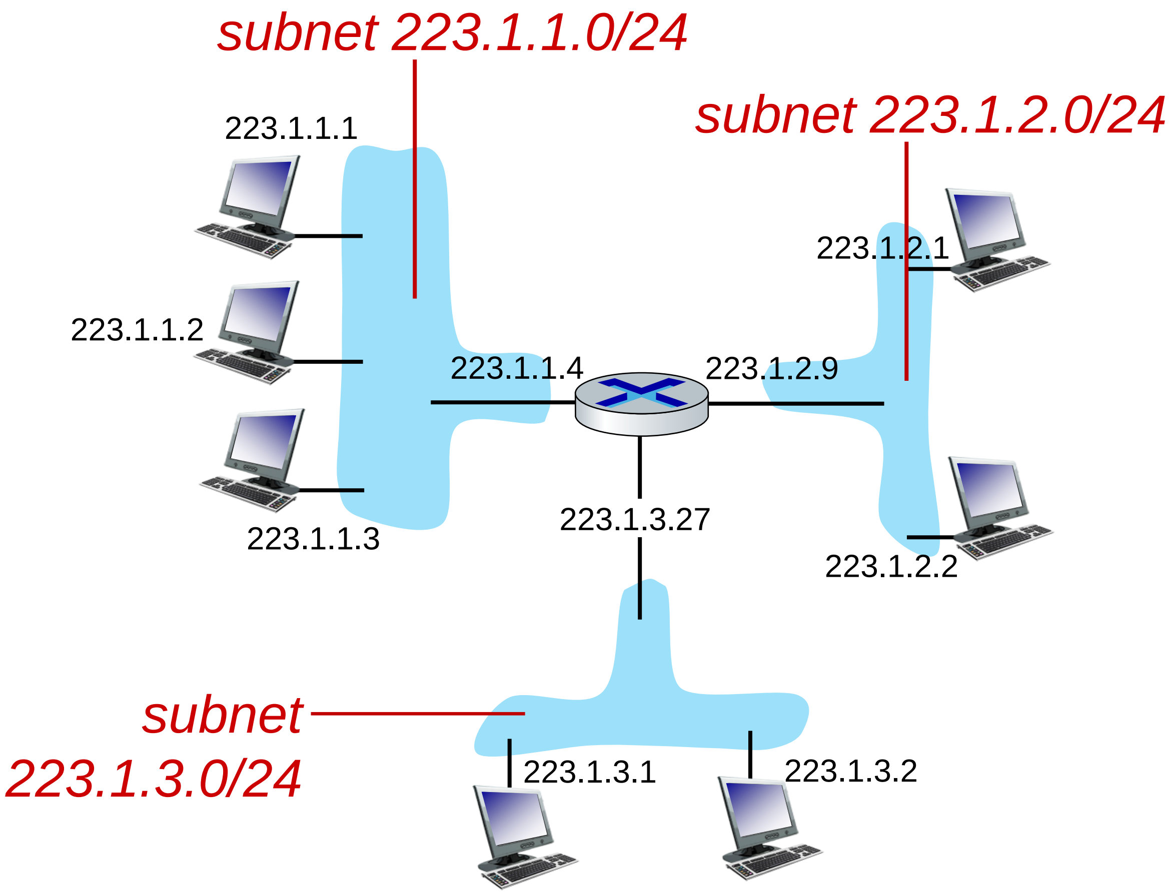


network consisting of 3 subnets

Subnets

Recipe for defining subnets:

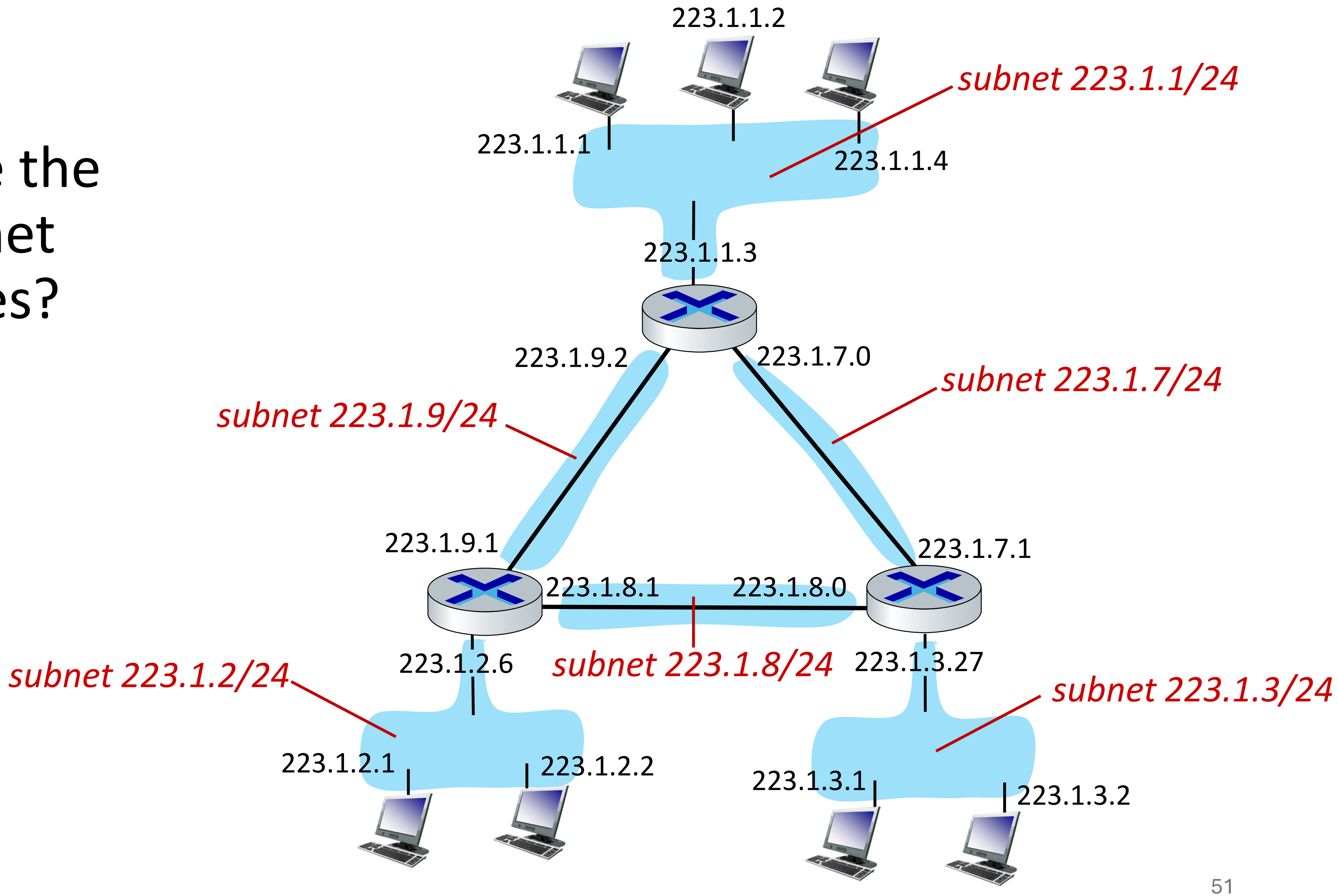
- detach each interface from its host or router, creating "islands" of isolated networks
- each isolated network is called a *subnet*



subnet mask: /24
(high-order 24 bits: subnet part of IP address)

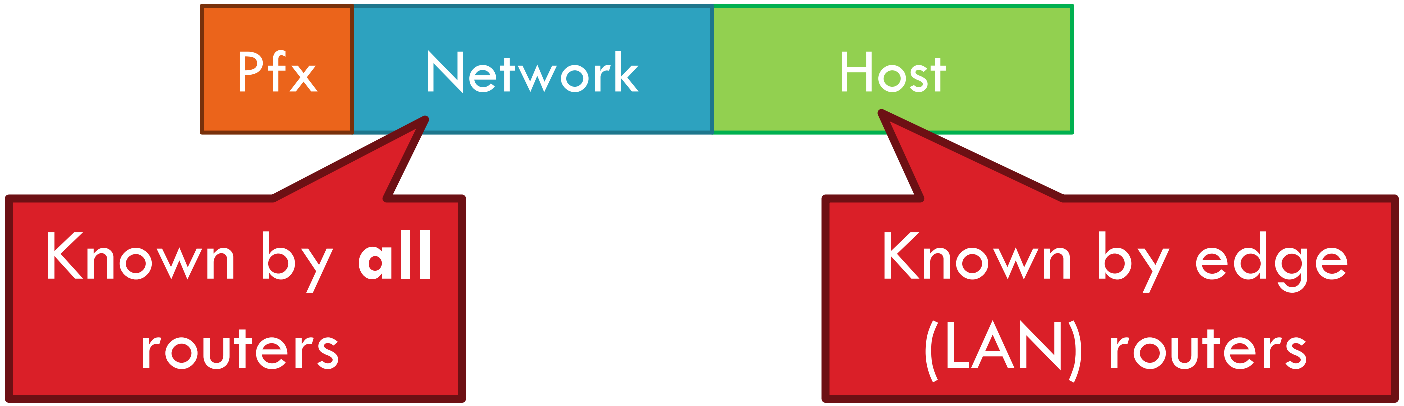
Subnets

- what are the /24 subnet addresses?



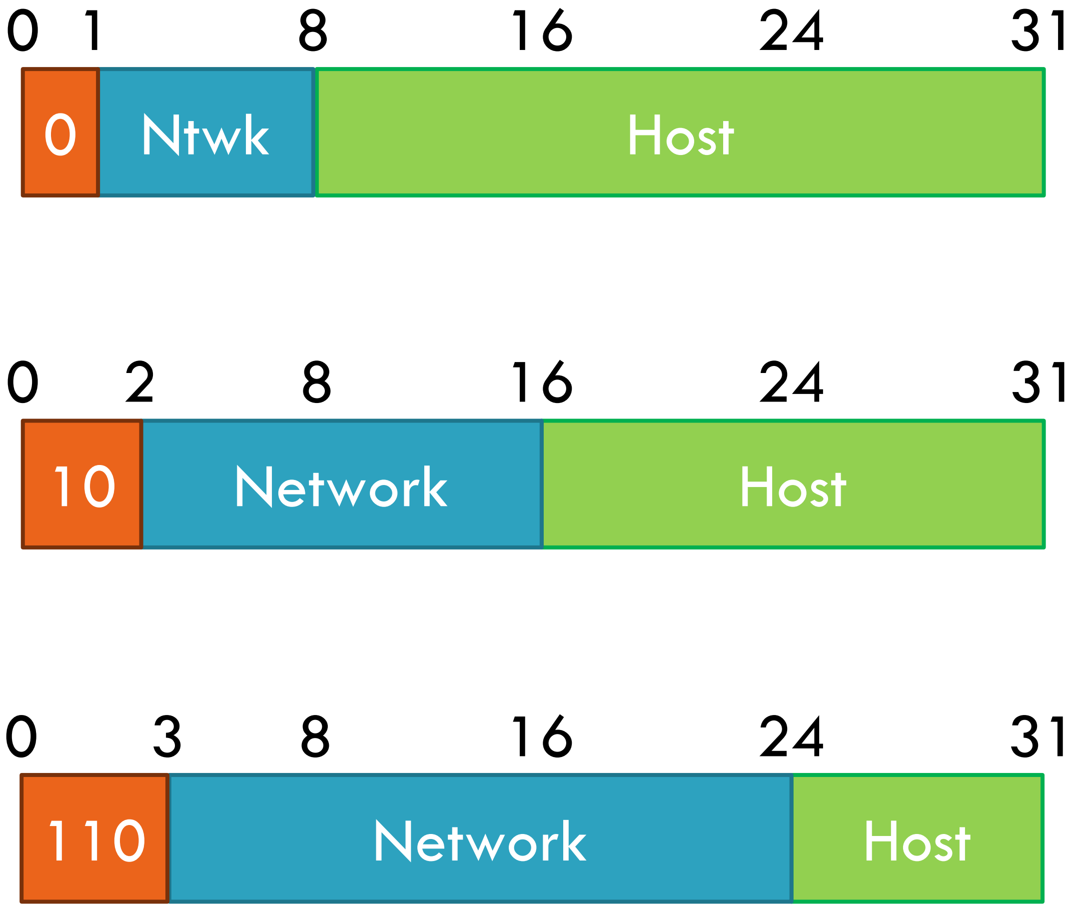
Flat IP Addressing does not scale well

- Routing Table Requirements
  - For every possible IP, give the next hop
  - But for 32-bit addresses, 2^{32} possibilities (4,294,967,296) !
  - **Too slow**
- Hierarchical address scheme
  - Separate the address into a network and a host



Classes of IP Addresses

- Class A
- Class B
- Class C



Example: MIT
18.*.*.*

Example: NEU
129.10.*.*

Example:
216.63.78.*

Class Sizes

Way too big

Class	Prefix Bits	Network Bits	Number of Classes	Hosts per Class
A	1	7	$2^7 - 2 = 126$ (0 and 127 are reserved)	$2^{24} - 2 = 16{,}777{,}214$ (All 0 and all 1 are reserved)
B	2	14	$2^{14} = 16{,}398$	$2^{16} - 2 = 65{,}534$ (All 0 and all 1 are reserved)
C	3	21	$2^{21} = 2{,}097{,}512$	$2^8 - 2 = 254$ (All 0 and all 1 are reserved)
			Total: 2,114,036	

Too many network IDs

Too small to be useful

Classless Inter-Domain Routing (CIDR) (pronounced "cider")

- Key ideas: Flexible division between network and host addresses
  - Get rid of IP classes
  - Network prefix can be any size
  - A mask is a 32-bit number that determines the network part and the host part

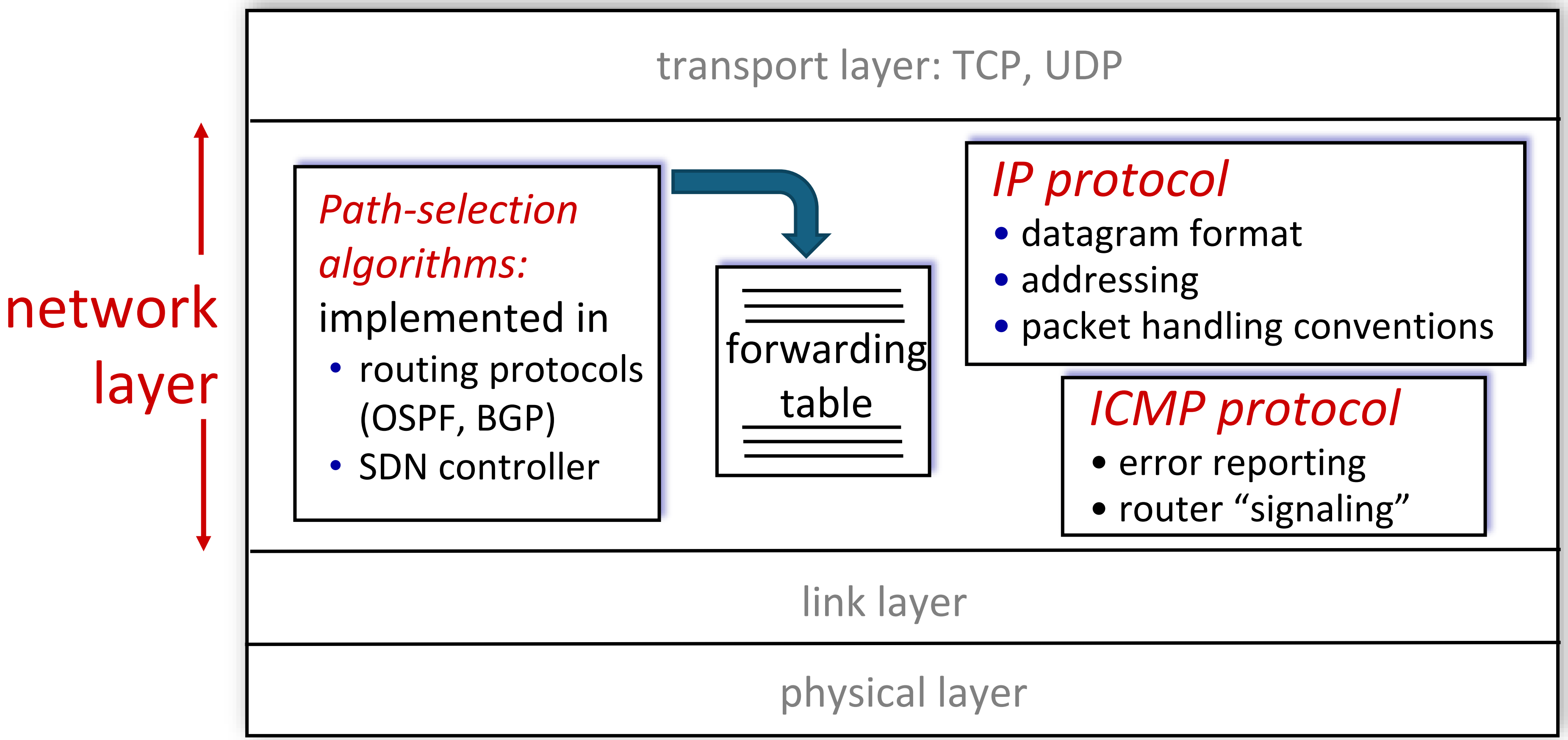
subnet part ← → host part

11001000 00010111 00010000 00000000

200.23.16.0/23

Network Layer: Internet

host, router network layer functions:



Destination-based forwarding

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through	0
11001000 00010111 00010000 00000100 through 11001000 00010111 00010000 00000111	3
11001000 00010111 00011000 11111111	
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Q: but what happens if ranges don't divide up so nicely?

Longest prefix matching

longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** ********	0
11001000 00010111 00011000 ********	1
11001000 00010111 00011*** ********	2
otherwise	3

examples:

11001000 00010111 00010110 10100001 which interface?

11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range				Link interface
11001000	00010111	00010***	********	0
11001000	00010111	00011000	********	1
11001000	00010111	00011***	********	2
otherwise				3

match!

examples:

11001000 00010111 00010110 10100001 which interface?

11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** ********	0
11001000 00010111 00011000 ********	1
11001000 00010111 00011*** ********	2
otherwise	3

match!

examples:

11001000 00010111 00010110 10100001 which interface?

11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** ********	0
11001000 00010111 00011000 ********	1
11001000 00010111 00011*** ********	2
otherwise	3

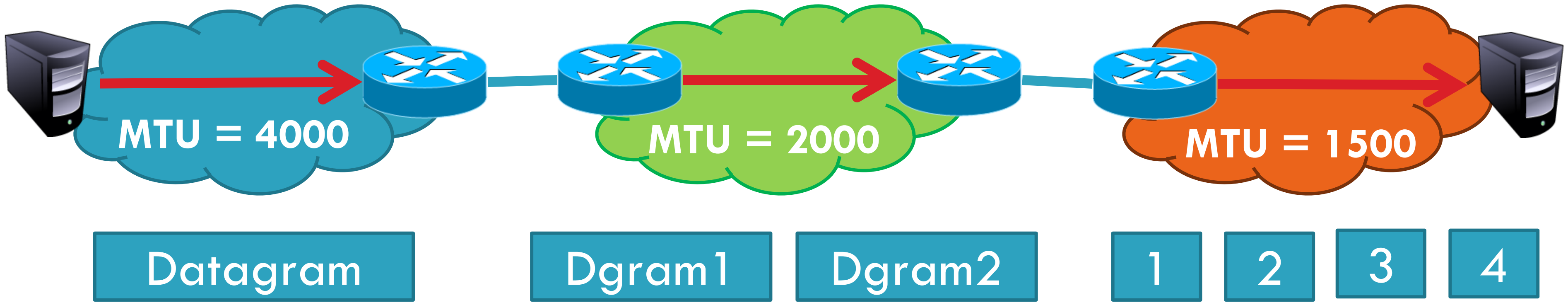
match!

examples:

11001000 00010111 00010110 10100001 which interface?

11001000 00010111 00011000 10101010 which interface?

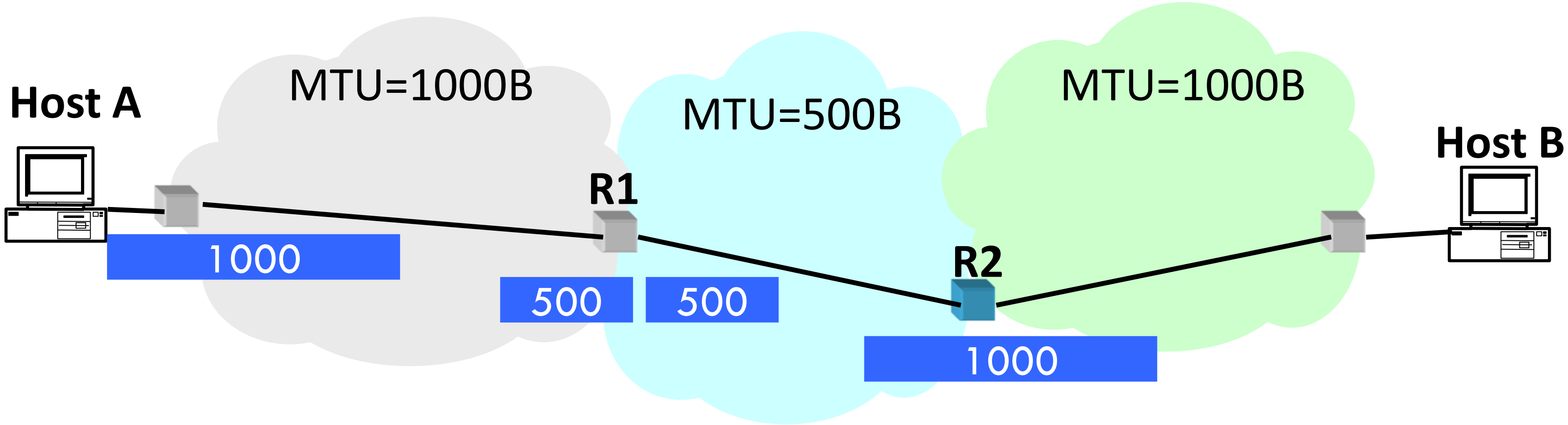
Problem: How to cope with different MTUs?



- Problem: each network has its own MTU
  - Maximum datagram size / Maximum Transmission Unit (MTU)
  - Minimum MTU may not be known for a given path
- IP Solution: fragmentation
  - Split datagrams into pieces when MTU is reduced

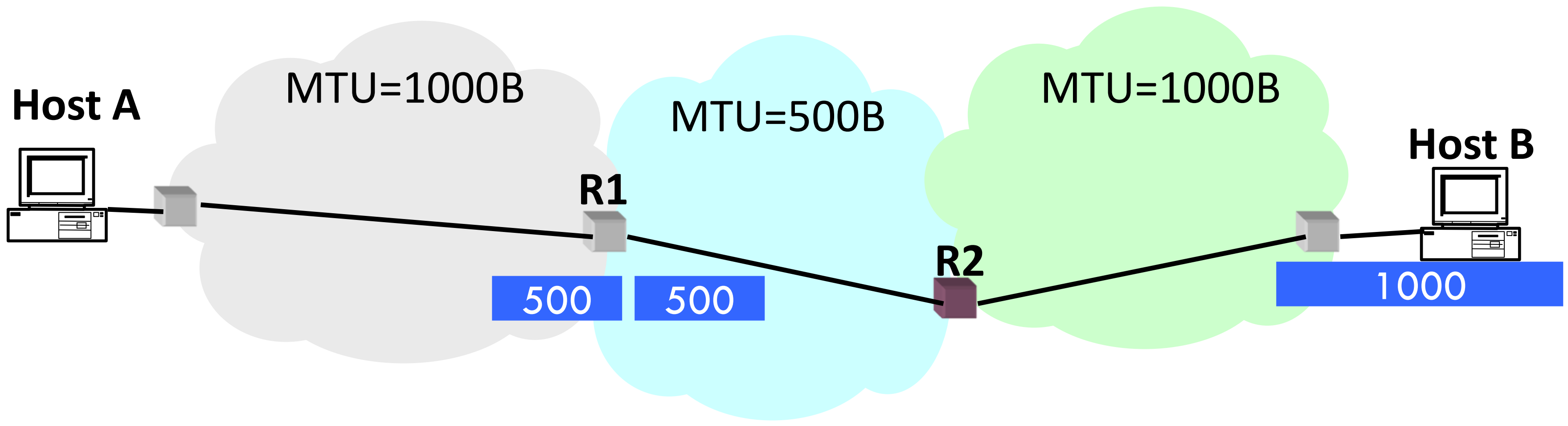
Where should reassembly happen?

- **Answer #1:** within the network, with no help from end-host *B* (receiver)



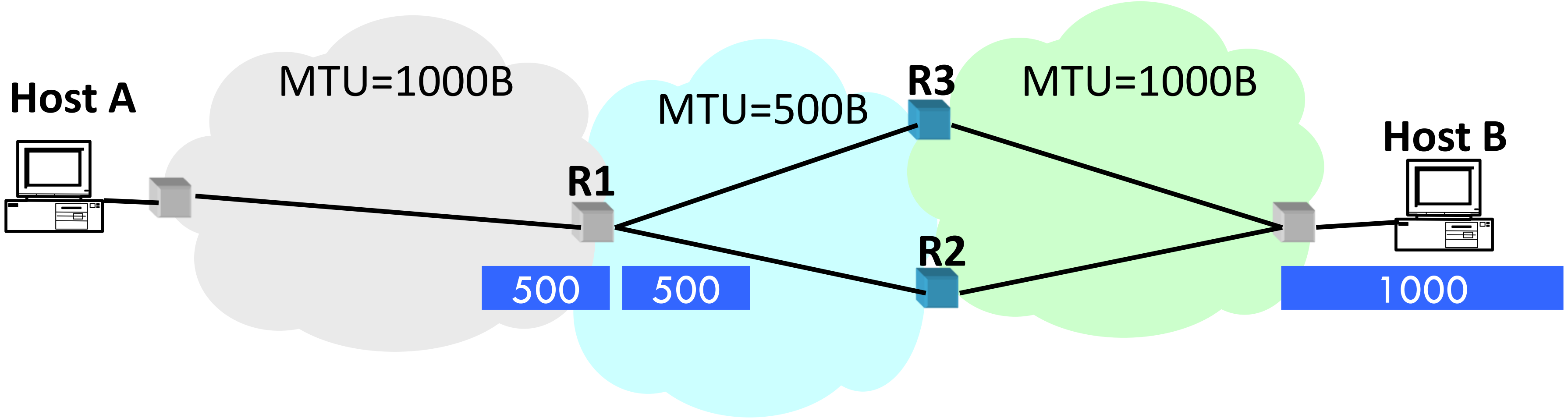
Where should reassembly happen?

- **Answer #1:** within the network, with no help from end-host *B* (receiver)
- OR
- **Answer #2:** at end-host *B* (receiver) with no help from the network



Where should reassembly happen?

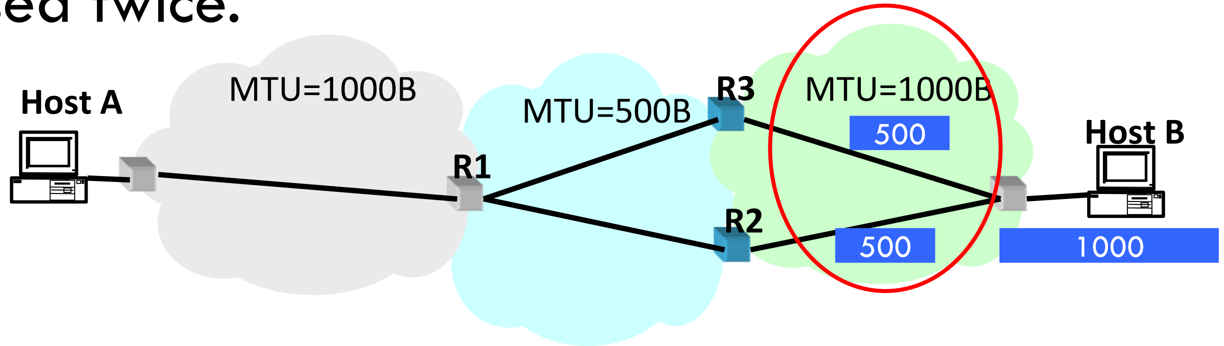
- **Answer #1:** within the network, with no help from end-host *B* (receiver) ✗
- **Answer #2:** at end-host *B* (receiver) with no help from the network ✓
- Fragments can travel across different paths!



Fragmentation is Considered Harmful

- Although IP's fragmentation is in keeping with the end-to-end principle, fragmentation is generally considered harmful for two performance-related reasons:

1. Fragmentation causes inefficient use of resources, same packet processed twice.



2. Loss of fragments leads to degraded performance
   - Loss of any fragment requires retransmit of entire datagram

Administrator: Command Prompt

```
C:\Windows\System32>ping mit.edu -f -l 1473

Pinging mit.edu [23.43.64.242] with 1473 bytes of data:
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.

Ping statistics for 23.43.64.242:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Windows\System32>ping mit.edu -l 1473

Pinging mit.edu [23.43.64.242] with 1473 bytes of data:
Reply from 23.43.64.242: bytes=1473 time=11ms TTL=50
Reply from 23.43.64.242: bytes=1473 time=11ms TTL=50
Reply from 23.43.64.242: bytes=1473 time=11ms TTL=50
Reply from 23.43.64.242: bytes=1473 time=11ms TTL=50

Ping statistics for 23.43.64.242:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 11ms, Maximum = 11ms, Average = 11ms

C:\Windows\System32>ping mit.edu -f -l 1470

Pinging mit.edu [23.43.64.242] with 1470 bytes of data:
Reply from 23.43.64.242: bytes=1470 time=10ms TTL=50
Reply from 23.43.64.242: bytes=1470 time=11ms TTL=50
Reply from 23.43.64.242: bytes=1470 time=11ms TTL=50
Reply from 23.43.64.242: bytes=1470 time=10ms TTL=50

Ping statistics for 23.43.64.242:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 10ms, Maximum = 11ms, Average = 10ms
```

Thanks for listening!
Any questions?

Try to answer these:

Why was the Internet Protocol designed this way?

Why connectionless, datagram, best-effort?

Why fragmentation in the network?

Why the Internet address be hierarchical?

What are the implications of buffer size on network performance?

What address does a host have?

Are there other ways to design networks?

Acknowledgment

- James F. Kurose University of Massachusetts, Amherst
- Keith W. Ross NYU and NYU Shanghai
- Guido Appenzeller, Stanford University