Chapter 1: roadmap

1.1 What is the Internet?

1.2 Network edge

- end systems, access networks, links

1.3 Network core

- circuit switching, packet switching
- hierarchical Internet structure

1.4 Performance:

- delay, loss and throughput

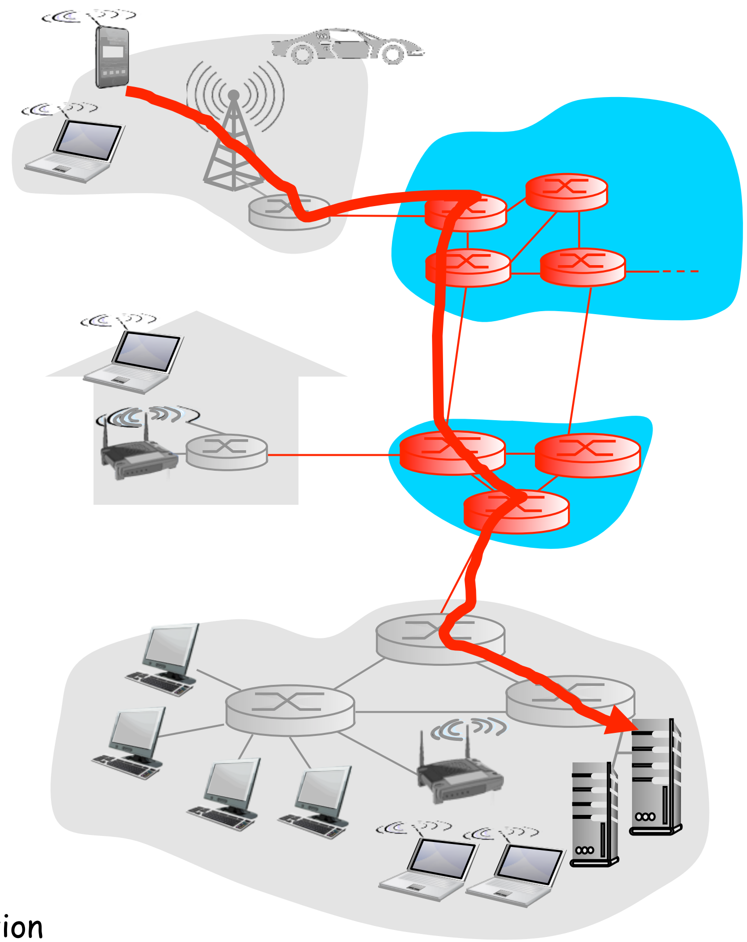
1.5 Protocol layers, service models

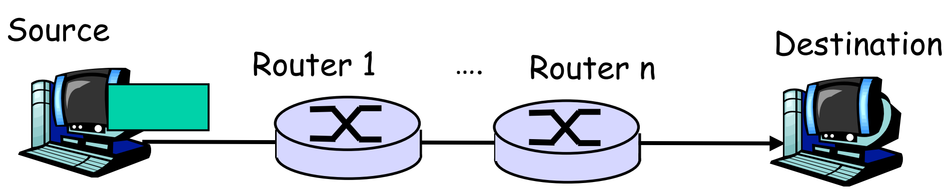
1.6 Networks under attack: security

1.7 History

The network core

- mesh of interconnected routers





Internet Design Principles

1. Packet vs Circuit Switching (BC 6 or KR 1.3)
2. Hierarchy: Network of Networks (BC 6 or KR 1.3)
3. Layered Architecture (BC 7 or KR 1.5)

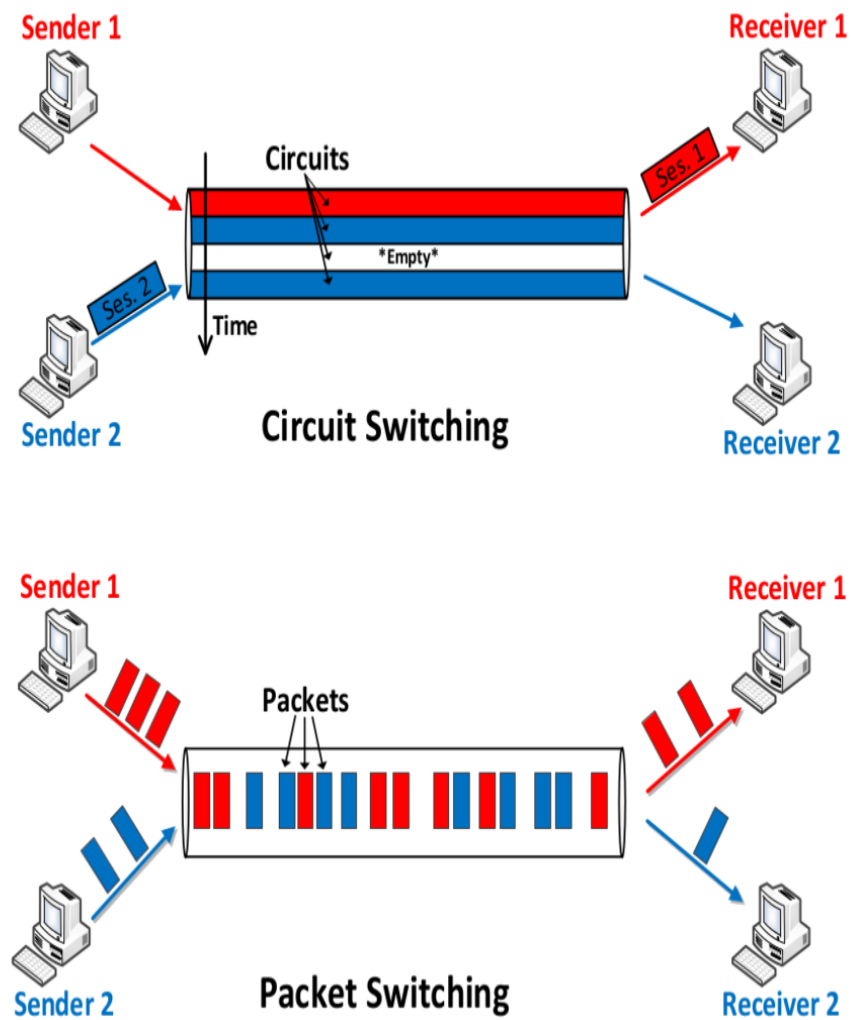
Through a Historical Perspective (or KR 1.7)

"Networks Illustrated: Principles without Calculus" by Brinton & Chiang

Book: http://www.amazon.com/gp/product/B00DIIKDJO
Coursera: https://www.coursera.org/course/ni

Sharing Resources

- FDMA, TDMA, CDMA
  - Circuit-switched: Dedicate / assign network resources
  - Negative – Not all resources used constantly!
- Paradigm shift
  - Packet-switched: Let everyone share the resources
- Session
  - Application-level
  - Unicast: One source, one dest.



Circuit Switching Advantage

- Debate runs far and deep

1. Guarantee of Quality

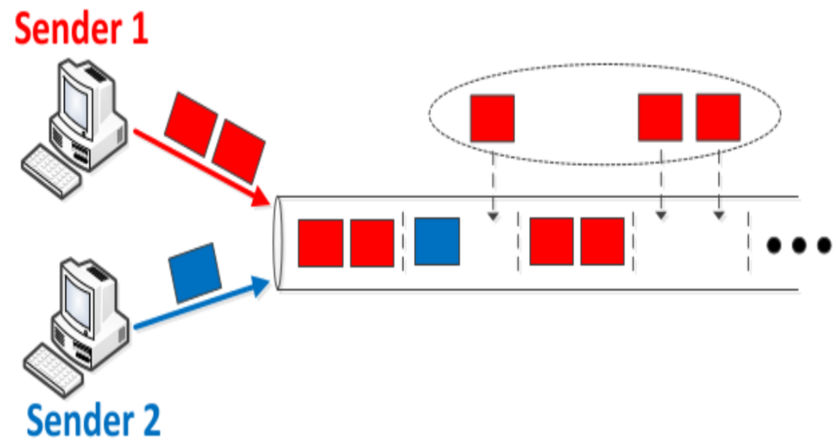
- Circuit Switching
  - Each session has a dedicated circuit
  - Throughput and delay performance will not change!
- Packet Switching
  - Best-effort service: no guarantees (“NO effort)
  - Links get congested, messages arrive out of order, …

Packet Switching Advantages

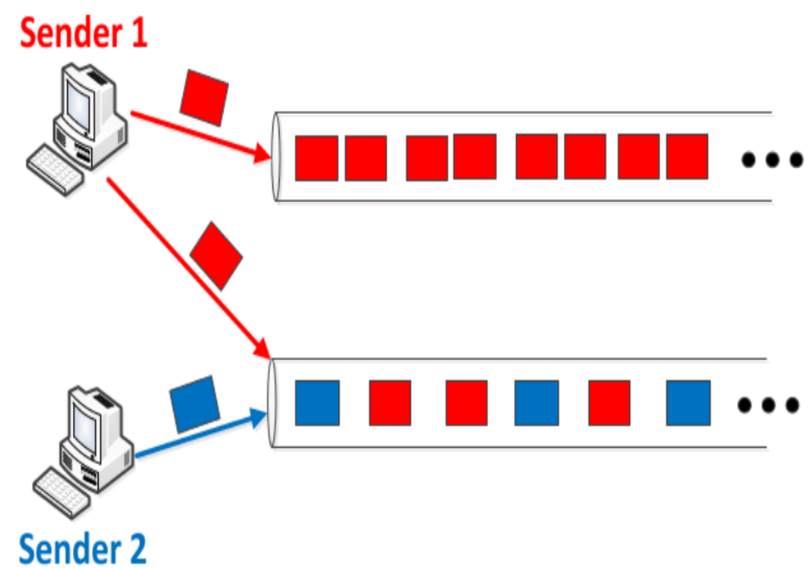
2. Ease of Connectivity
   - No need to allocate resources first
   - Transmit at will, as long as protocols are followed
3. Scalability
   - Large number of diverse sessions
   - Obtained through high efficiency
   - (1) Statistical multiplexing
   - (2) Resource pooling

Packet Switching Advantages

- Statistical multiplexing
  - Multiple sessions can share one path



- Resource pooling
  - One session can use multiple paths



- No wasted/idle network resources through reservations

Packet vs. Circuit Switching

Property	Circuit Switching	Packet Switching
Guarantee of Quality	✓	X
Ease of Connectivity	X	✓
Scalability	X	✓

- In the end, packet switching won the day
- Not clear until early 2000s
- Scalability first, then search for other quality control solutions

Resource Allocation

Human Analogies:

- reservation at a restaurant
- a reserved lane on the highway
- *Q:* other human analogies of reserved resources vs. on-demand allocation?

Networks:

- Resources (bandwidth, buffer) divided into pieces and allocated to calls or packets
- Reservations: circuit-switching
- No reservations: packet-switching

Packet Switching vs. Circuit Switching

Is packet switching a clear winner?"

+ great for bursty data
  - network's pov: resource sharing ("statistical multiplexing")
  - user's pov: simpler (no call setup), less delay to start service

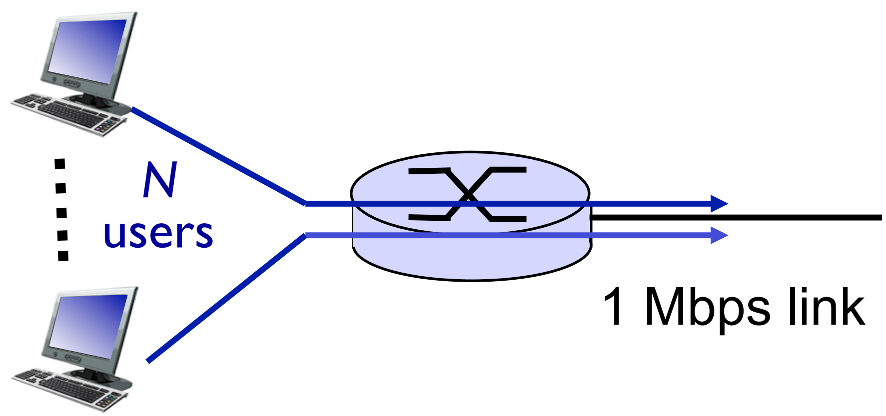
- excessive congestion possible
  - packet delay and loss
  - protocols needed for reliable data transfer, congestion control
  - no bandwidth guarantees needed for audio/video apps

Packet Switching vs. Circuit Switching

+ Packet switching allows more users ("statistical multiplexing)!

example:

- 1 Mb/s link
- each user:
  - 100 kb/s when "active"
  - active 10% of time
- *circuit-switching:*
  - 10 users
- *packet switching:*
  - with 35 users, probability {> 10 active at same time} is <0.0004 *

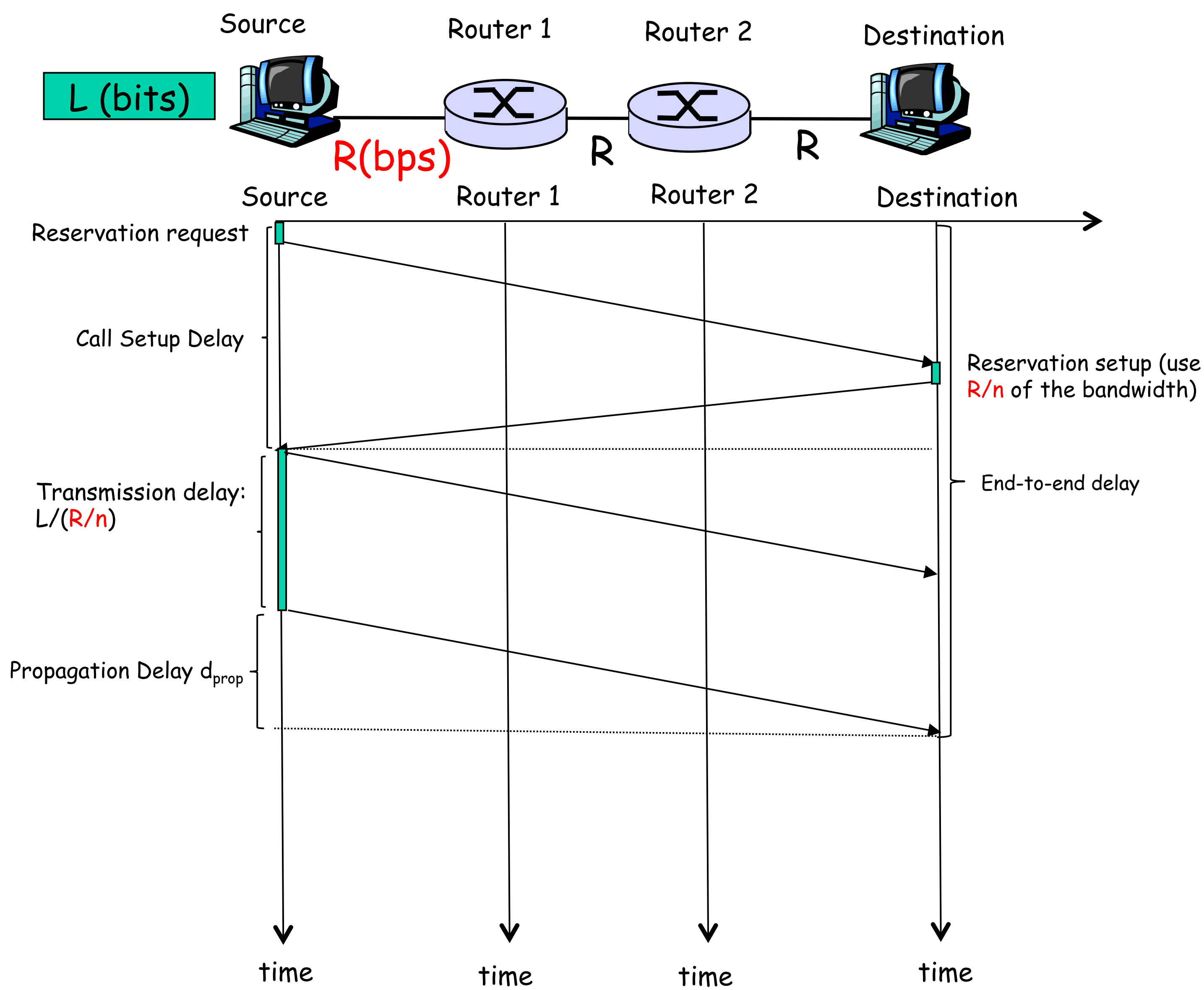


Q: how did we get value 0.0004?

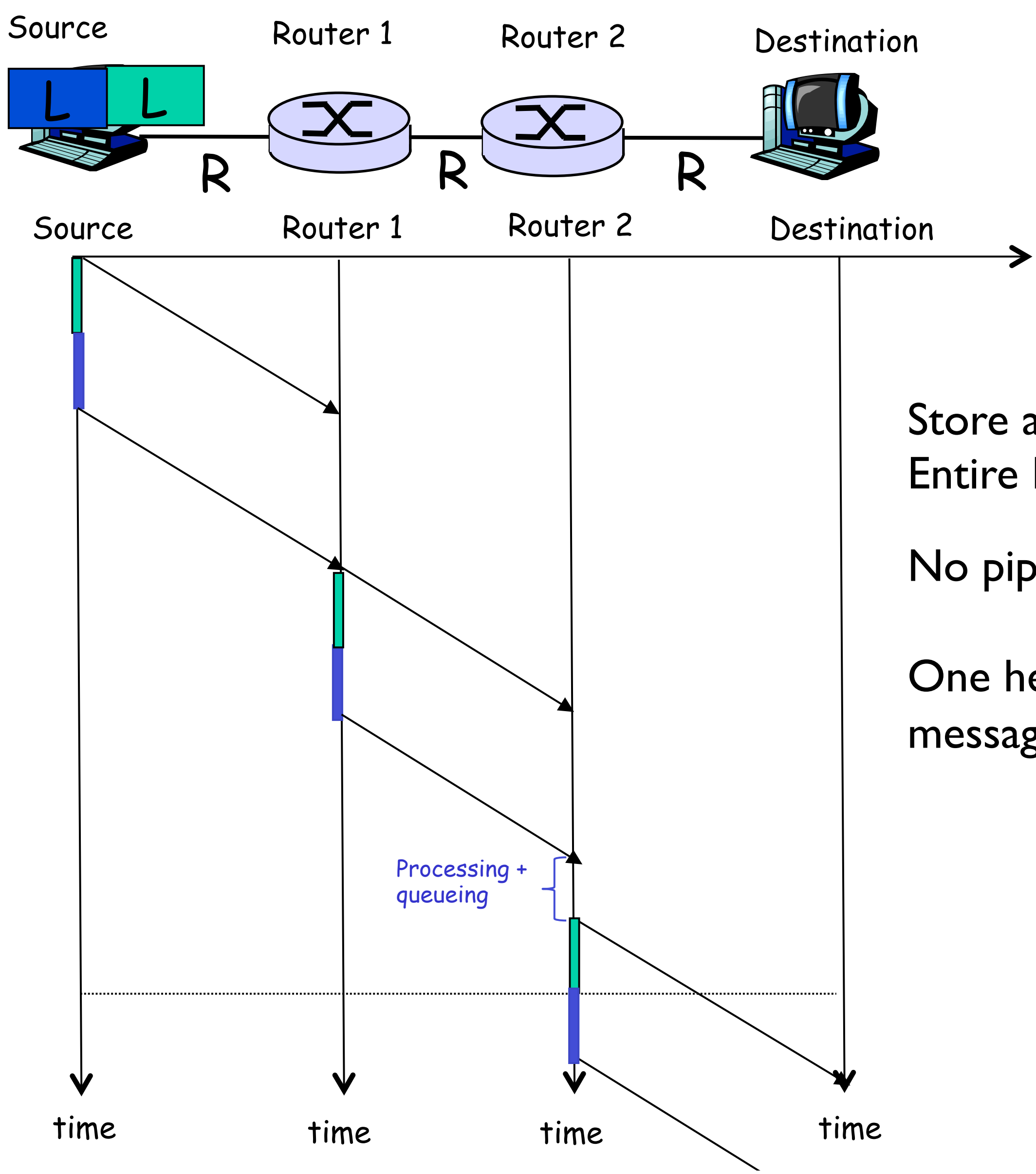
Q: what happens if > 35 users ?

Check out the online interactive exercises for more examples
http://wps.pearsoned.com/ecs_kurose_compnetw_6/216/55463/14198700.cw/index.html

Circuit-switching



Message Switching: no pipelining

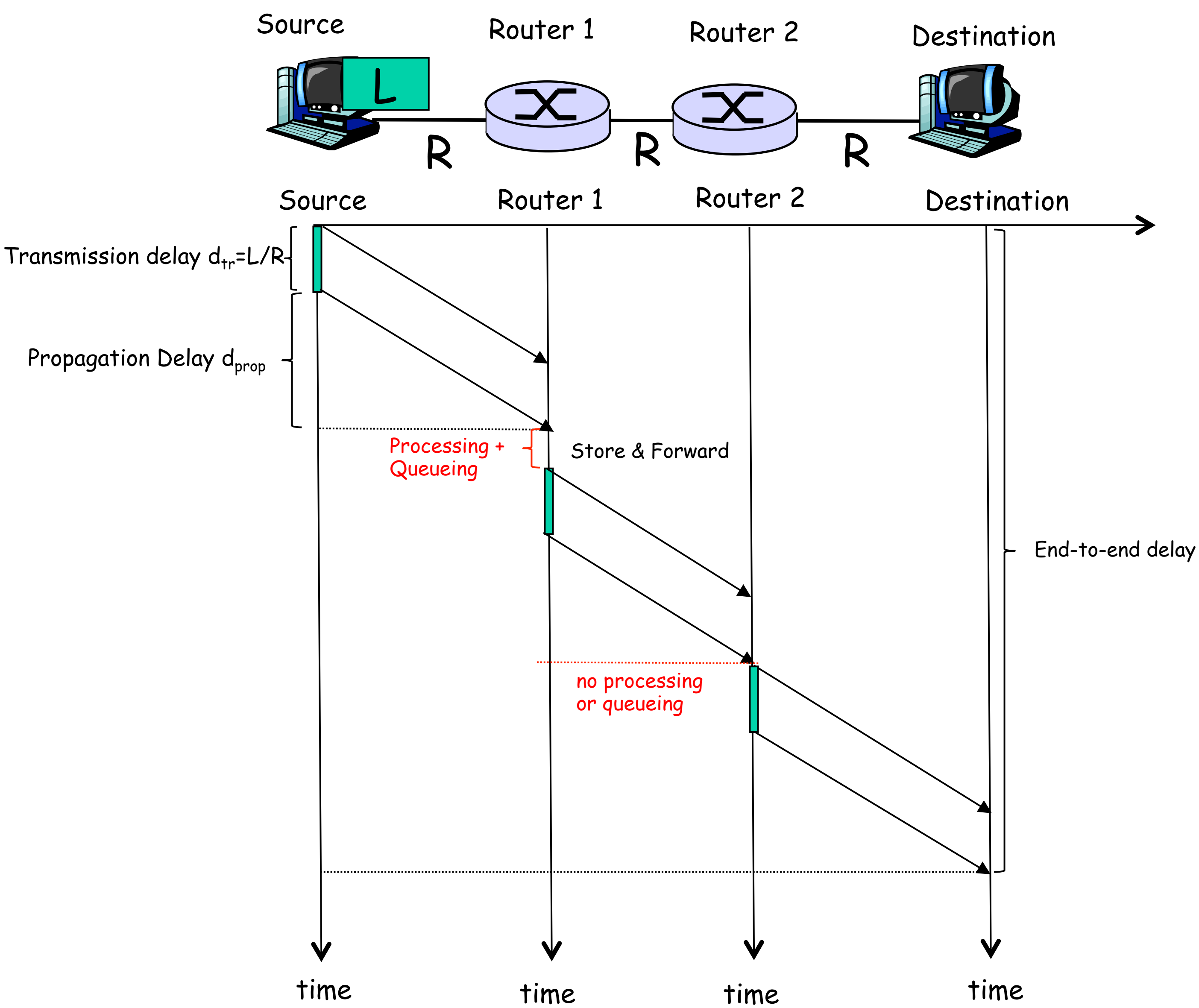


Store and Forward Entire Message

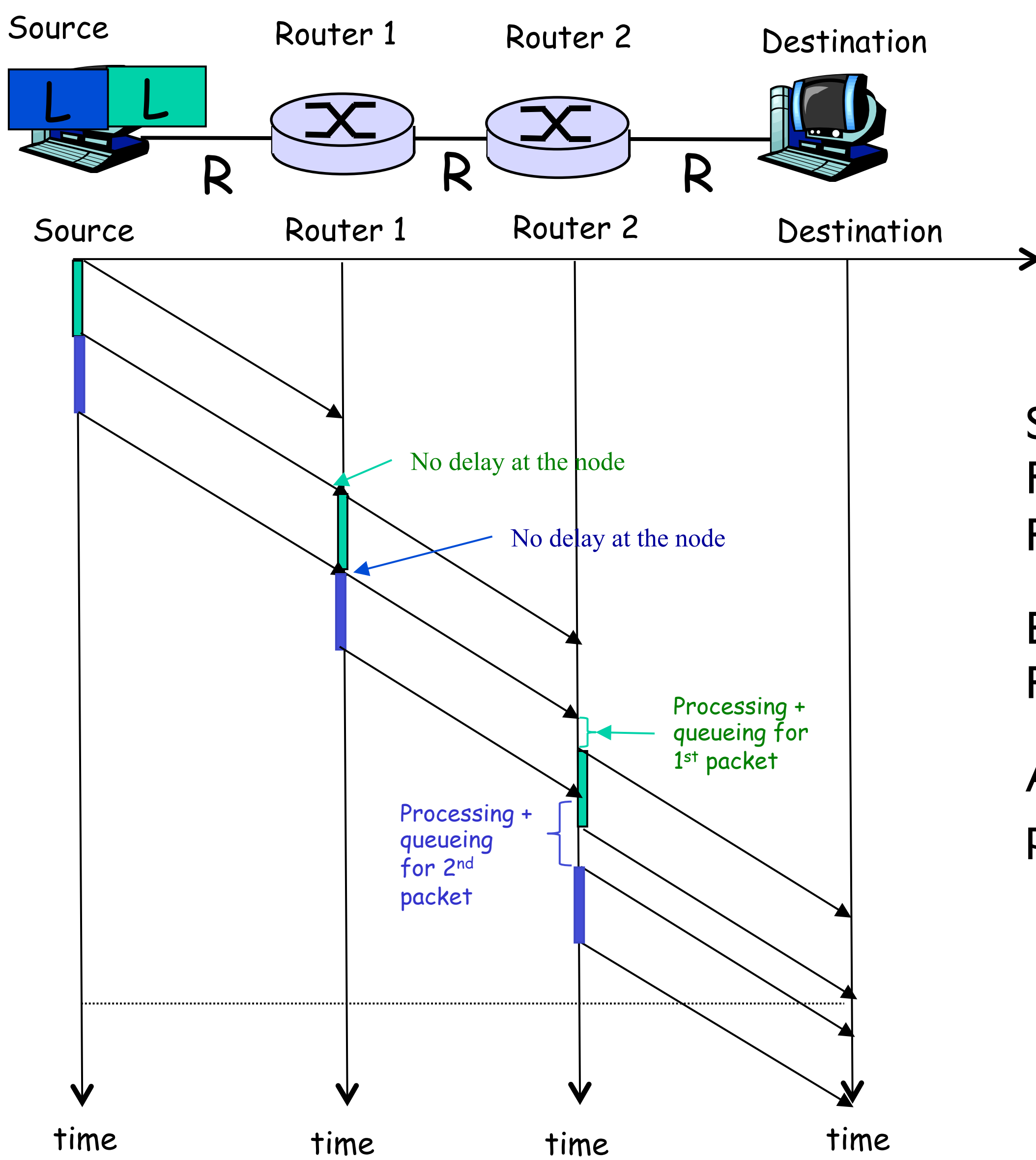
No pipelining

One header per message

Packet Switching: one packet



Packet Switching: multiple packets



Store and Forward each Packet

Benefit of Pipelining

Add a header per packet

Chapter 1: roadmap

1.1 What is the Internet?

1.2 Network edge

- end systems, access networks, links

1.3 Network core

- circuit switching, packet switching, network structure

1.4 Performance:

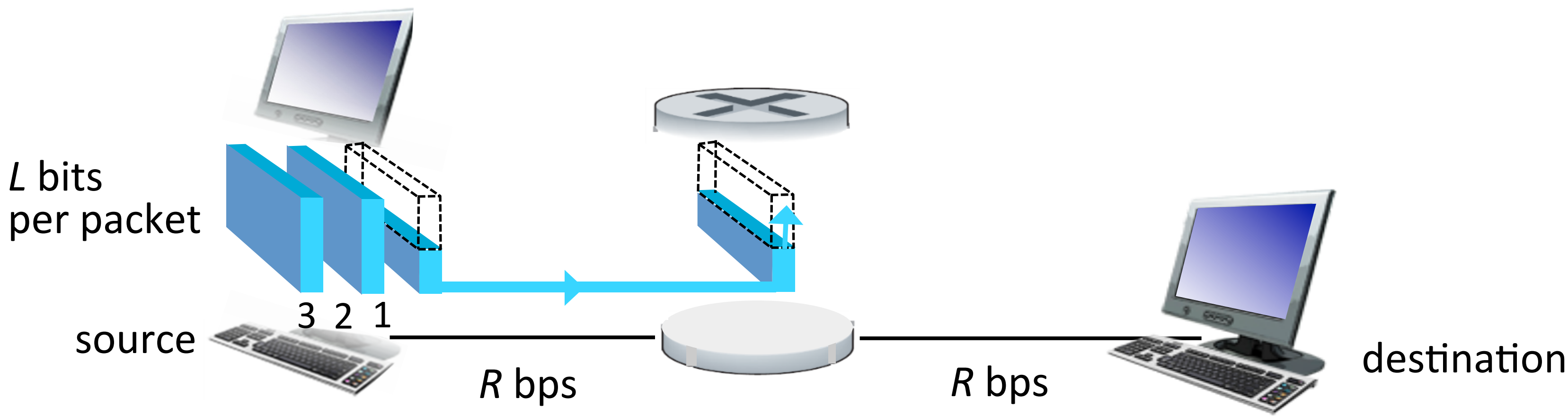
- delay, loss and throughput

1.5 Protocol layers, service models

1.6 Networks under attack: security

1.7 History

Packet-switching: store-and-forward

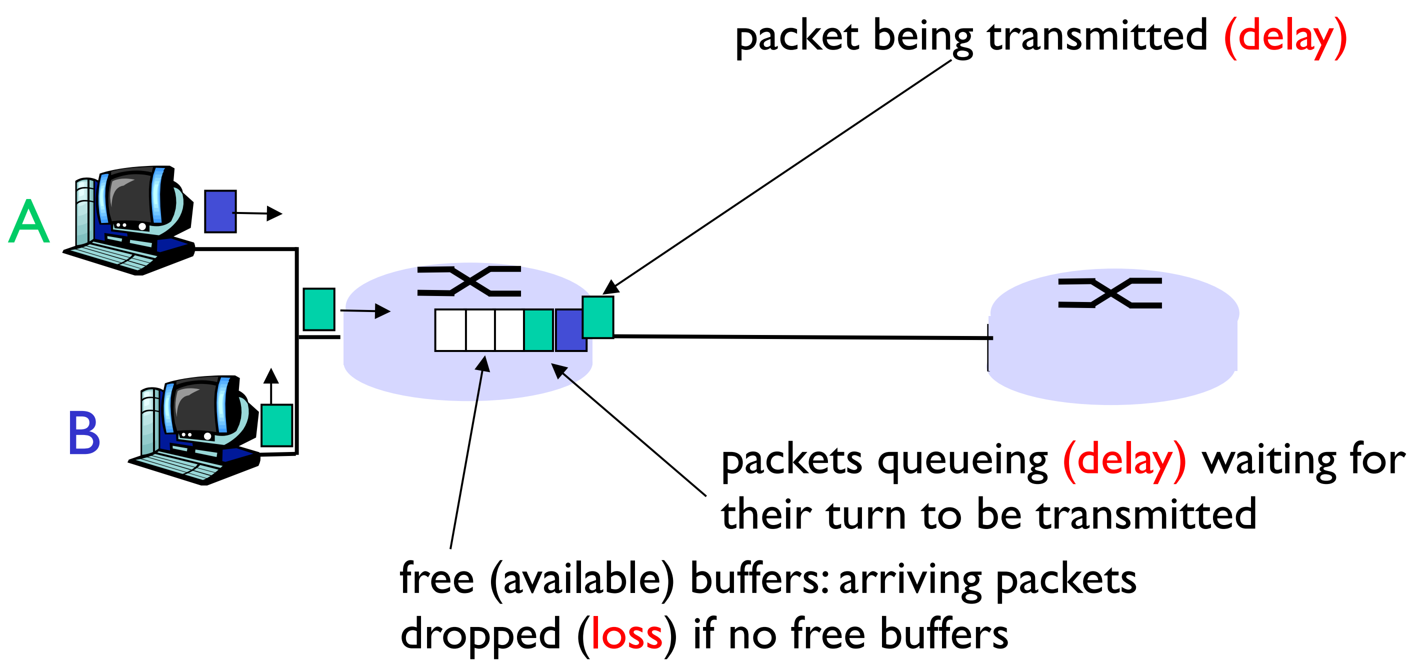


- takes *L*/*R* seconds to transmit (push out) *L*-bit packet into link at *R* bps
- *store and forward:* entire packet must arrive at router before it can be transmitted on next link

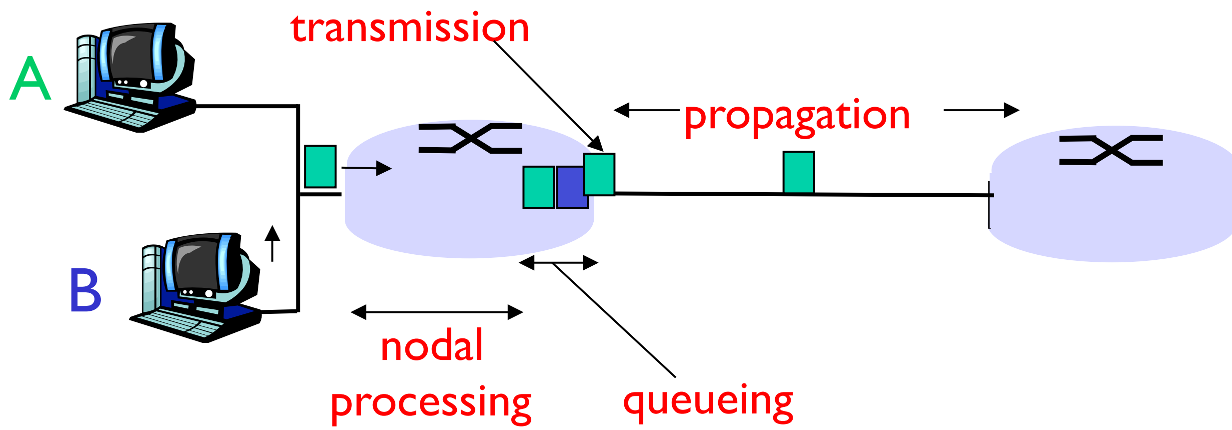
At Intermediate Nodes (routers)

Packets *queue* in router buffers

- if packet arrival rate to link exceeds output link capacity



Four sources of packet delay



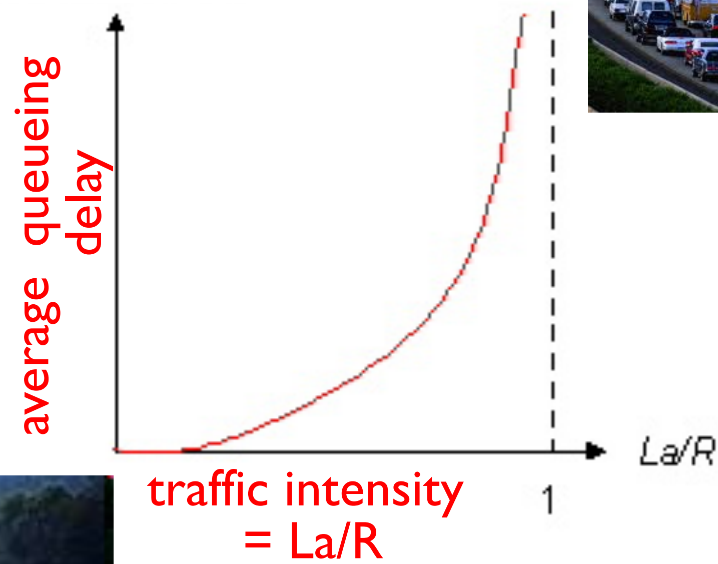
$$d = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

- d_{proc}: nodal processing
  - e.g. check but errors, determine output link, … Typically < 1ms.
- d_{queue}: queueing delay
  - time waiting at output buffer for transmission, depends on congestion level at router)
- d_{trans}: transmission delay =L/R
- d_{prop}: propagation delay = d/s,
  - d: length of physical link, s: propagation speed in medium (~$2x10^8$ m/sec)

Queueing delay

La/R → 1

- R: link bandwidth (bps)
- L: packet length (bits)
- a: average packet arrival rate (packets/sec)

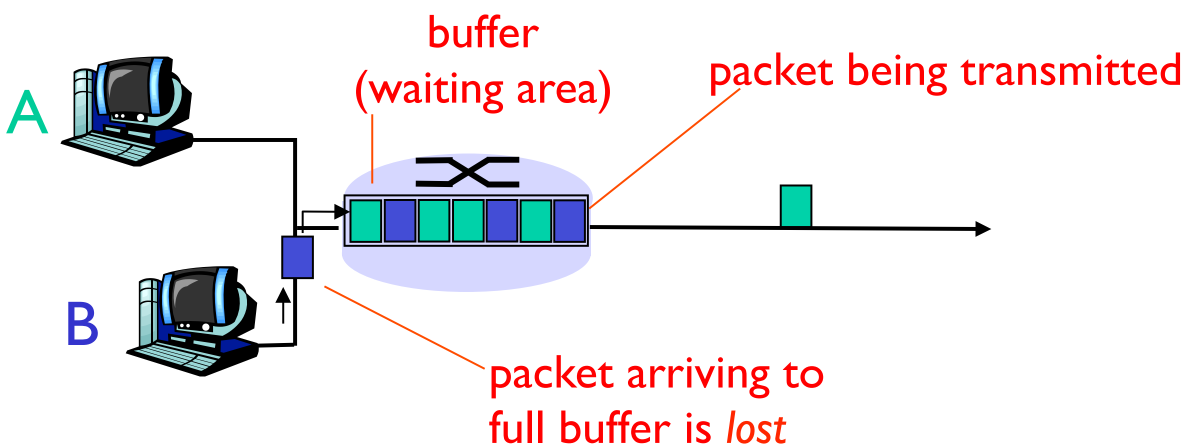


La/R ~ 0

- La/R ~ 0: avg. queueing delay small
- La/R → 1: avg. queueing delay large
- La/R > 1: more work arriving than can be serviced, average delay infinite!

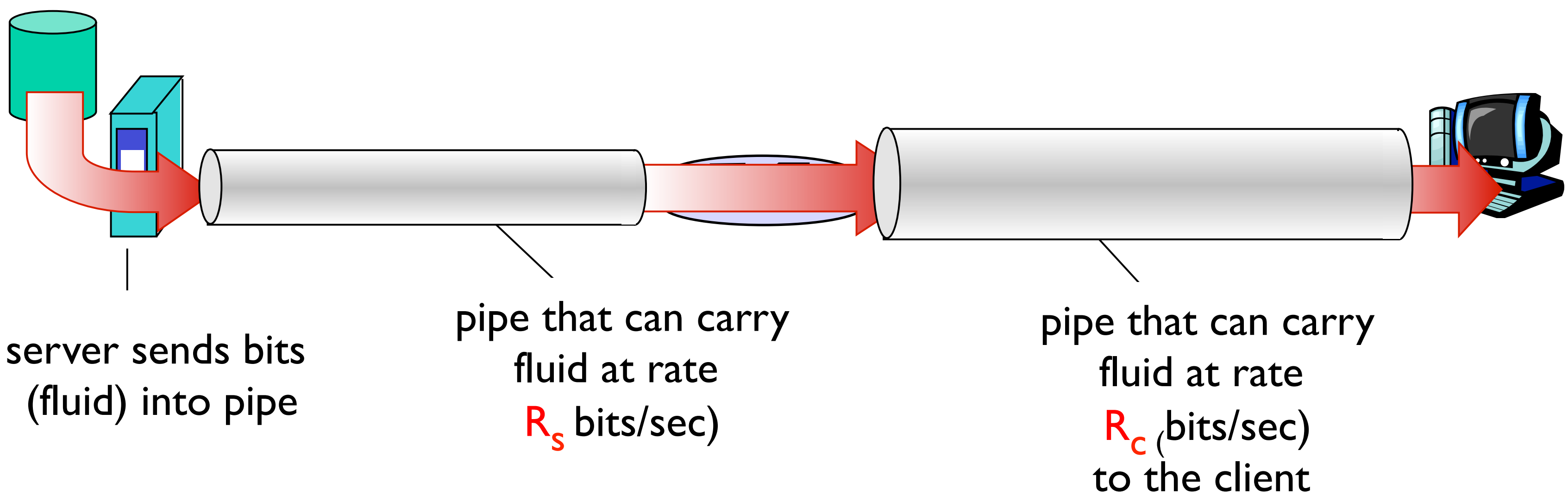
Packet loss

- Packets get dropped due to congestion at queues:
  - queue (aka buffer) preceding link in buffer has finite capacity
  - packet arriving to full queue dropped (aka lost)
  - lost packet may be retransmitted by previous node, by source end system, or not at all
- Packets can also be lost due to link failure, misconfiguration etc.



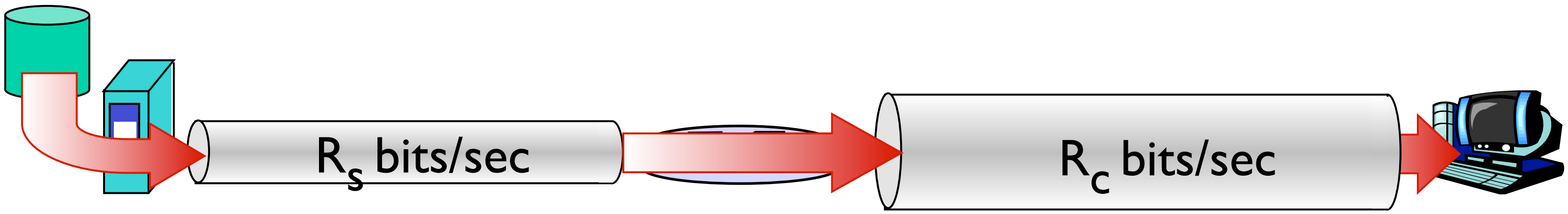
Throughput

- throughput: rate (bits/time unit) at which bits transferred between sender/receiver

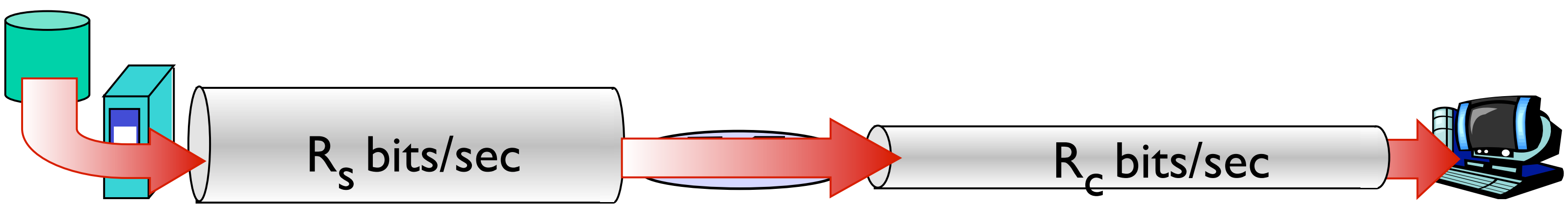


Throughput (cont'd)

- $R_s < R_c$ What is average end-end throughput?



- $R_s > R_c$ What is average end-end throughput?

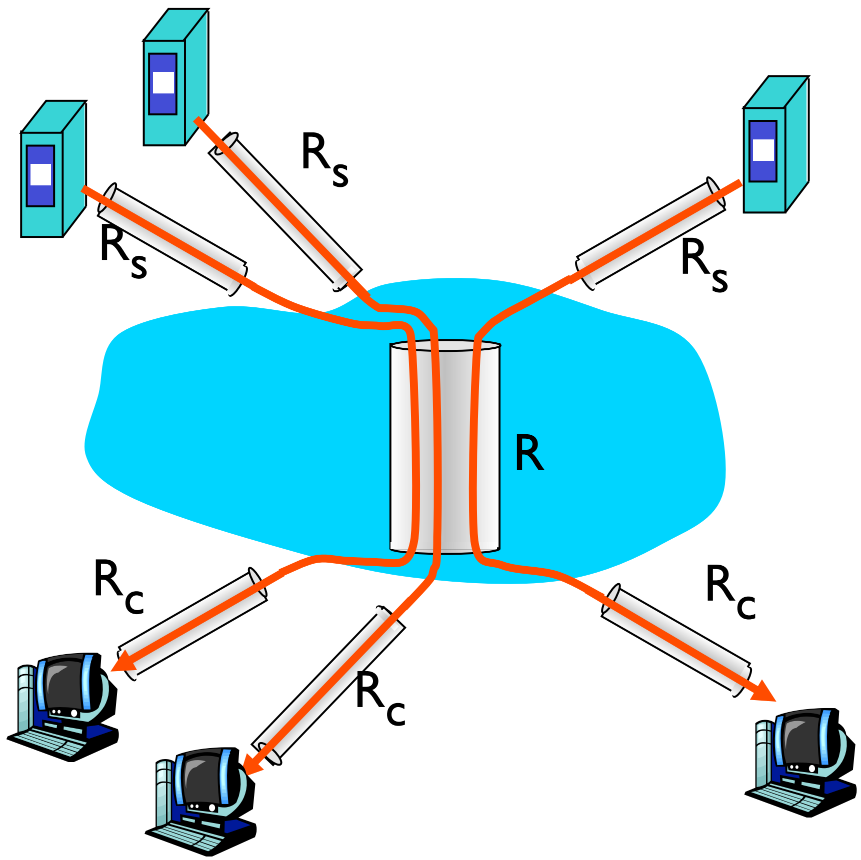


bottleneck link

link on end-end path that constrains end-end throughput

Throughput: Internet scenario

- per-connection end-end throughput: $\min(R_c, R_s, R/10)$
- in practice: R_c or R_s is often the bottleneck



10 connections (fairly) share backbone bottleneck link R bits/sec

Numerical example

- Question: How long does it take to send a file of F=640Kbits from Source to Destination over a packet switched network?
  - there are 3 links on the path
  - every link have speed: R=1.5 Mbps and propagation delay d_{prop}=10ms
  - processing and queuing delays are negligible
  - the file is broken into packets of P=1000 bits each (ignore added headers)

Interactive Exercise:

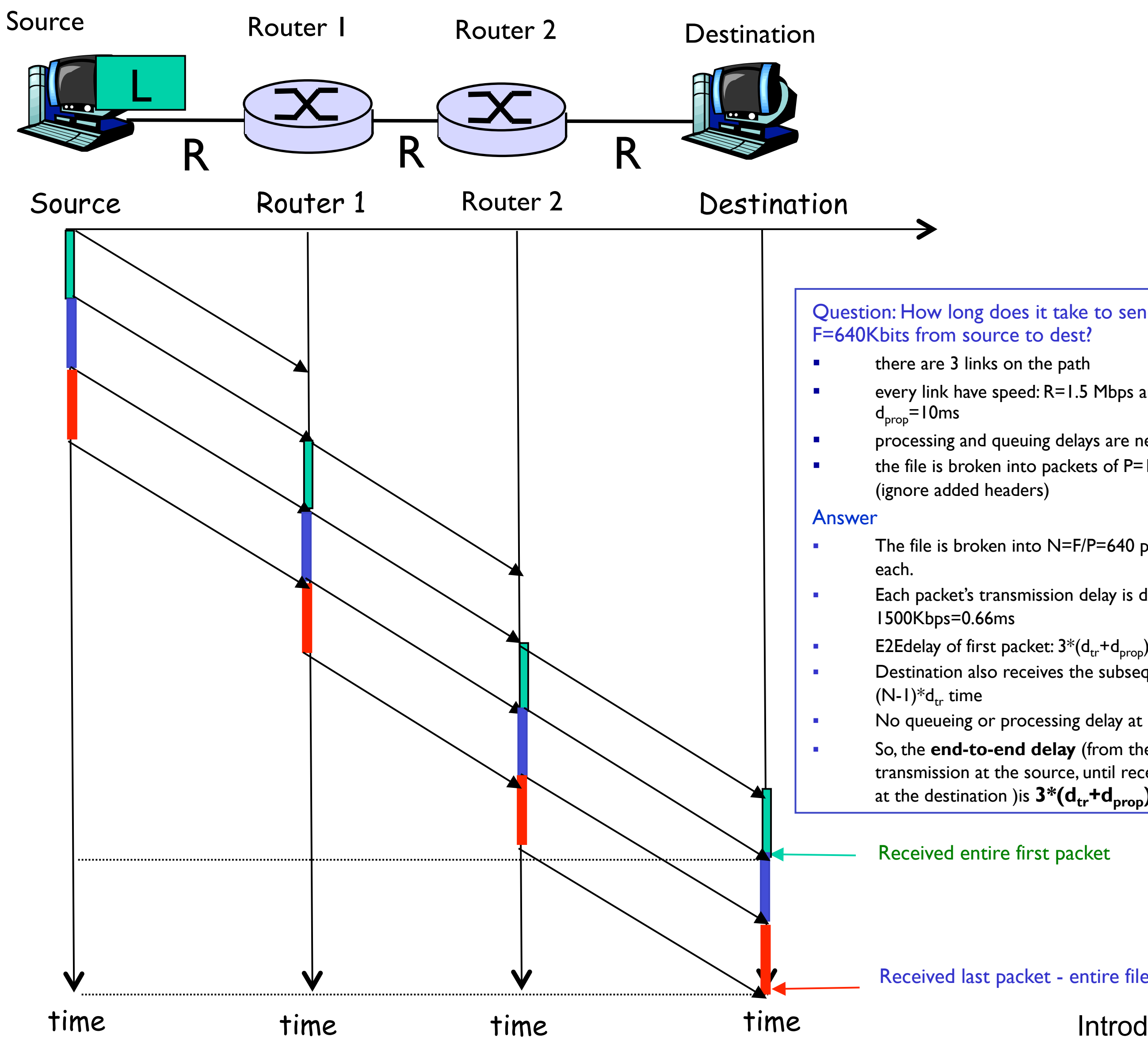
http://wps.pearsoned.com/ecs_kurose_compnetw_6/216/55463/14198700.cw/index.html

Packet-switching

E2E delay

on every hop:

$$d = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$



Question: How long does it take to send a file of F=640Kbits from source to dest?

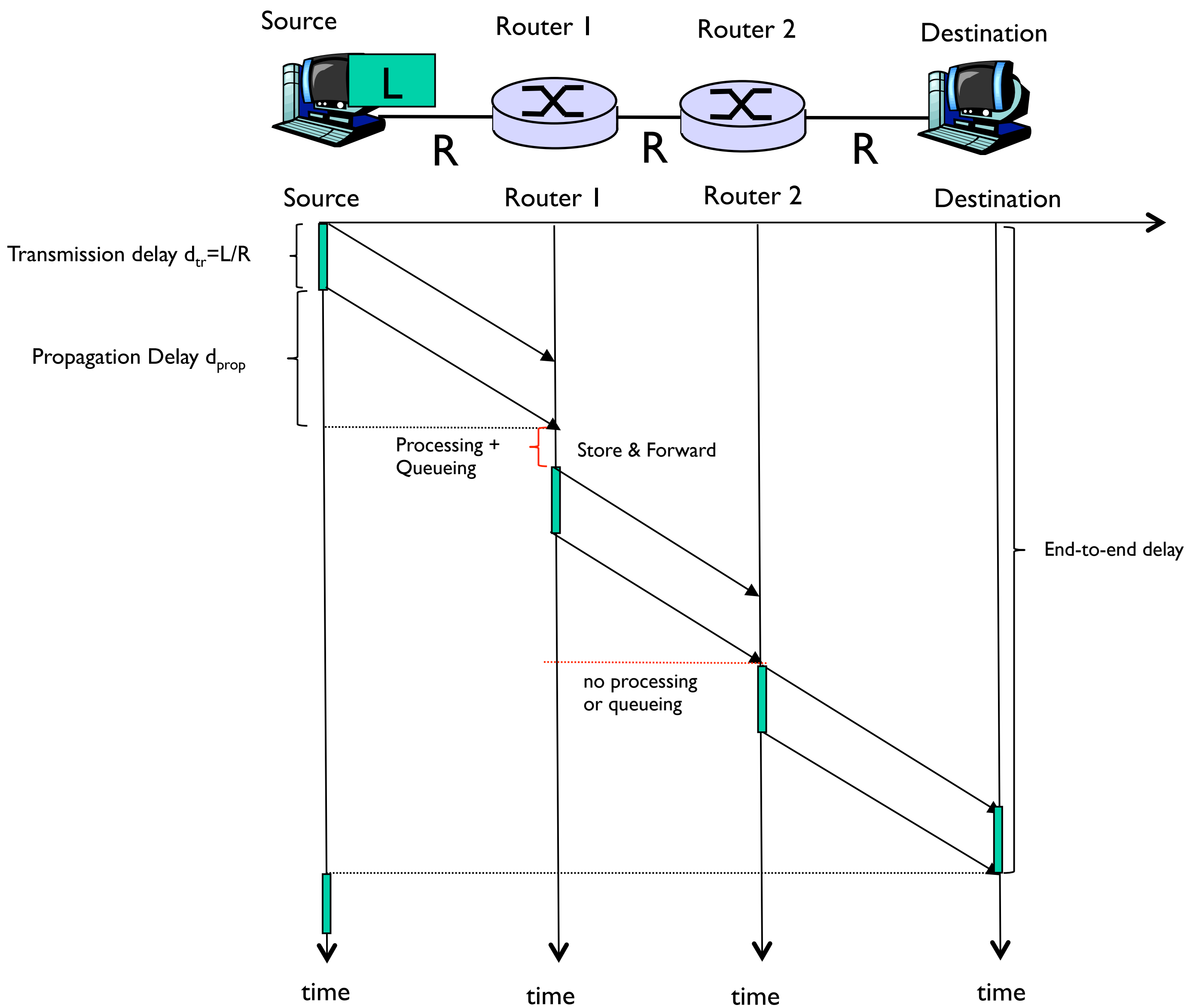
- there are 3 links on the path
- every link have speed: R=1.5 Mbps and propagation delay d_{prop}=10ms
- processing and queuing delays are negligible
- the file is broken into packets of P=1000 bits each (ignore added headers)

Answer

- The file is broken into N=F/P=640 packets of 1000 bits each.
- Each packet's transmission delay is d_{tr}=1000bits/1500Kbps=0.66ms
- E2Edelay of first packet: $3*(d_{tr}+d_{prop})$
- Destination also receives the subsequent packets in $(N-1)*d_{tr}$ time
- No queueing or processing delay at intermediate notes
- So, the **end-to-end delay** (from the beginning of transmission at the source, until receiving the entire file at the destination)is $\mathbf{3*(d_{tr}+d_{prop})+(N-1)*d_{tr}}$

Packet Switching

E2E throughput=L(bits)/e2e delay(sec)



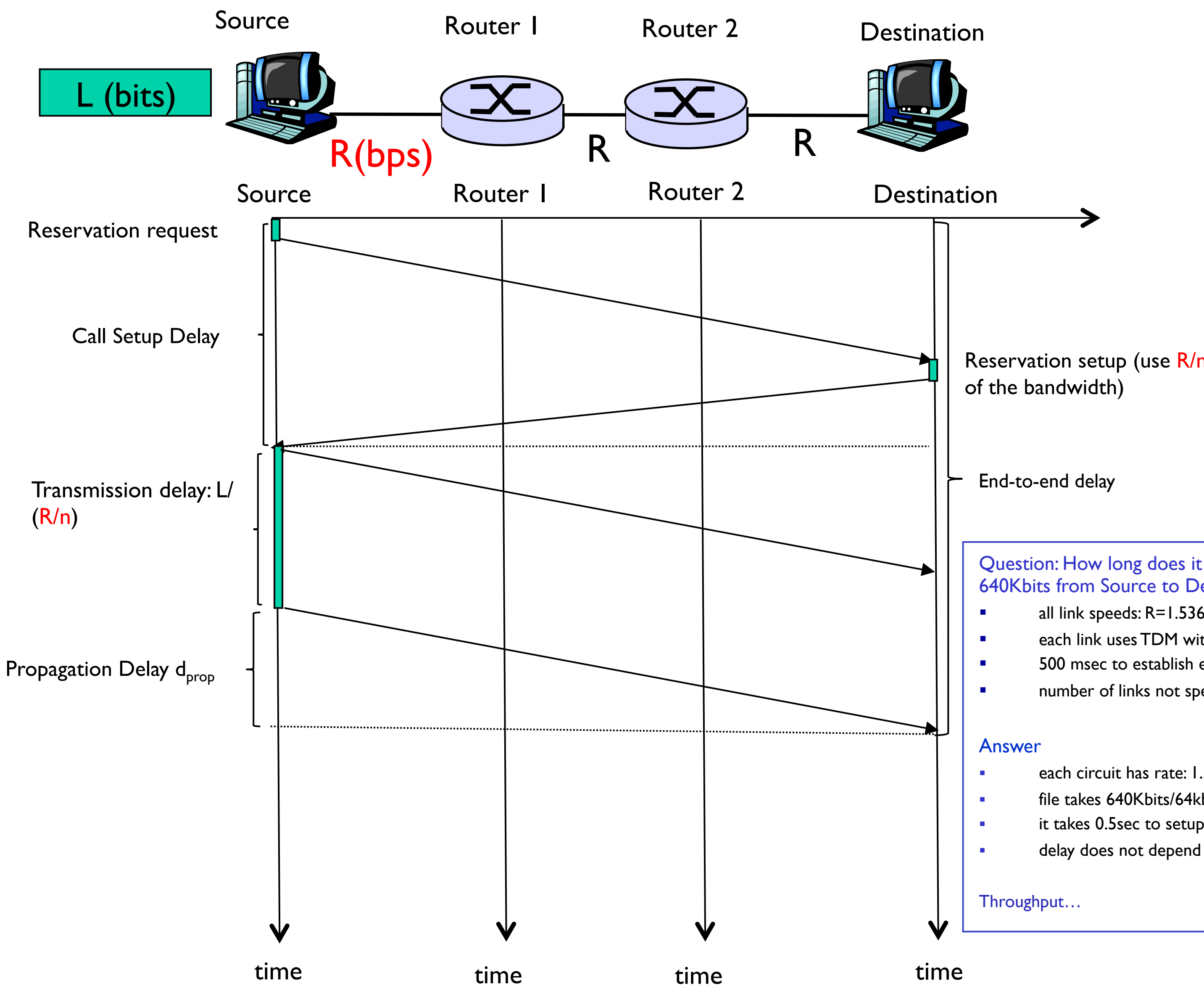
Notes on Throughput

- Alternative definition at source/receiver/router
- Goodput: only useful message counts, no headers or control messages
- Related but not the same with delay
- Interactive exercise: http://wps.pearsoned.com/ecs_kurose_compnetw_6/216/55463/14198700.cw/index.html

Numerical example

- Question: How long does it take to send a file of 640Kbits from Source to Destination over a circuit-switched network?
  - all link speeds: R=1.536 Mbps
  - each link uses TDM with n=24 slots/sec
  - 500 msec to establish end-to-end circuit

Circuit Switching: delay



Question: How long does it take to send a file of 640Kbits from Source to Destination?

- all link speeds: R=1.536 Mbps
- each link uses TDM with n=24 slots
- 500 msec to establish end-to-end circuit
- number of links not specified

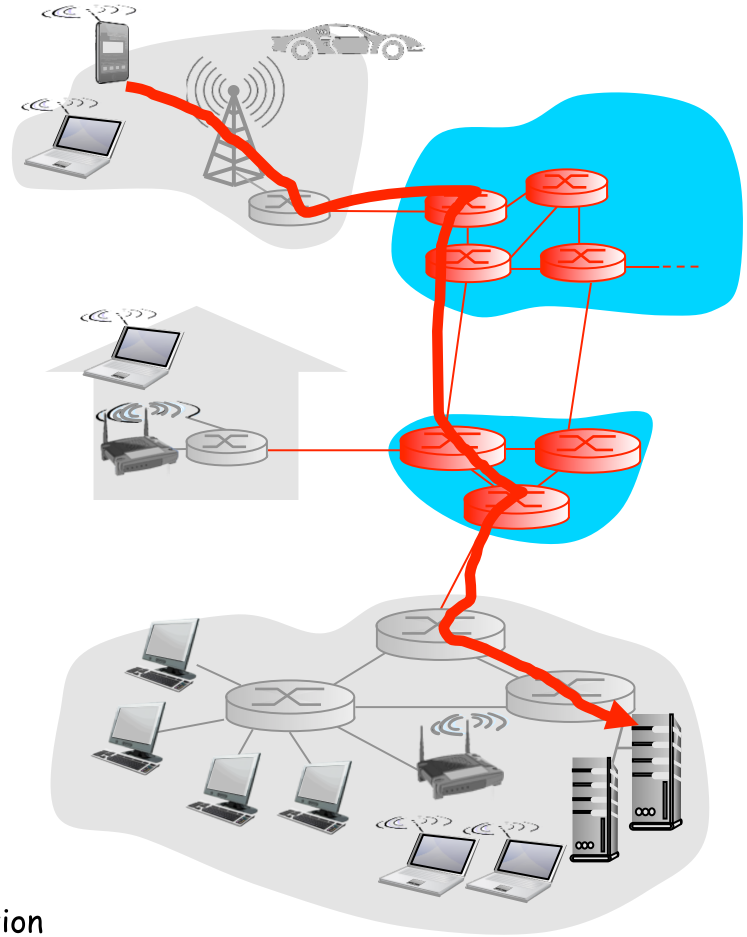
Answer

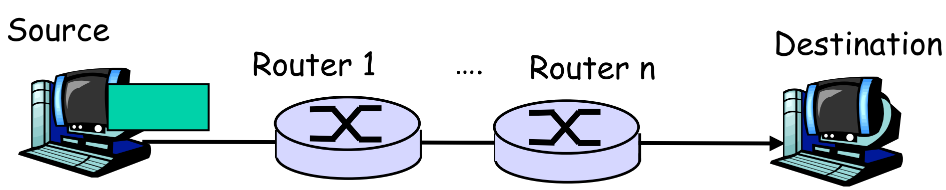
- each circuit has rate: 1.536Mbps/24=64kbps
- file takes 640Kbits/64kbps =10sec to transmit
- it takes 0.5sec to setup the circuit
- delay does not depend on the number of hops.

Throughput…

The network core

- mesh of interconnected routers
- packet-switching: hosts break application-layer messages into *packets*
  - forward packets from one router to the next, across links on path from source to destination
  - each packet transmitted at full link capacity



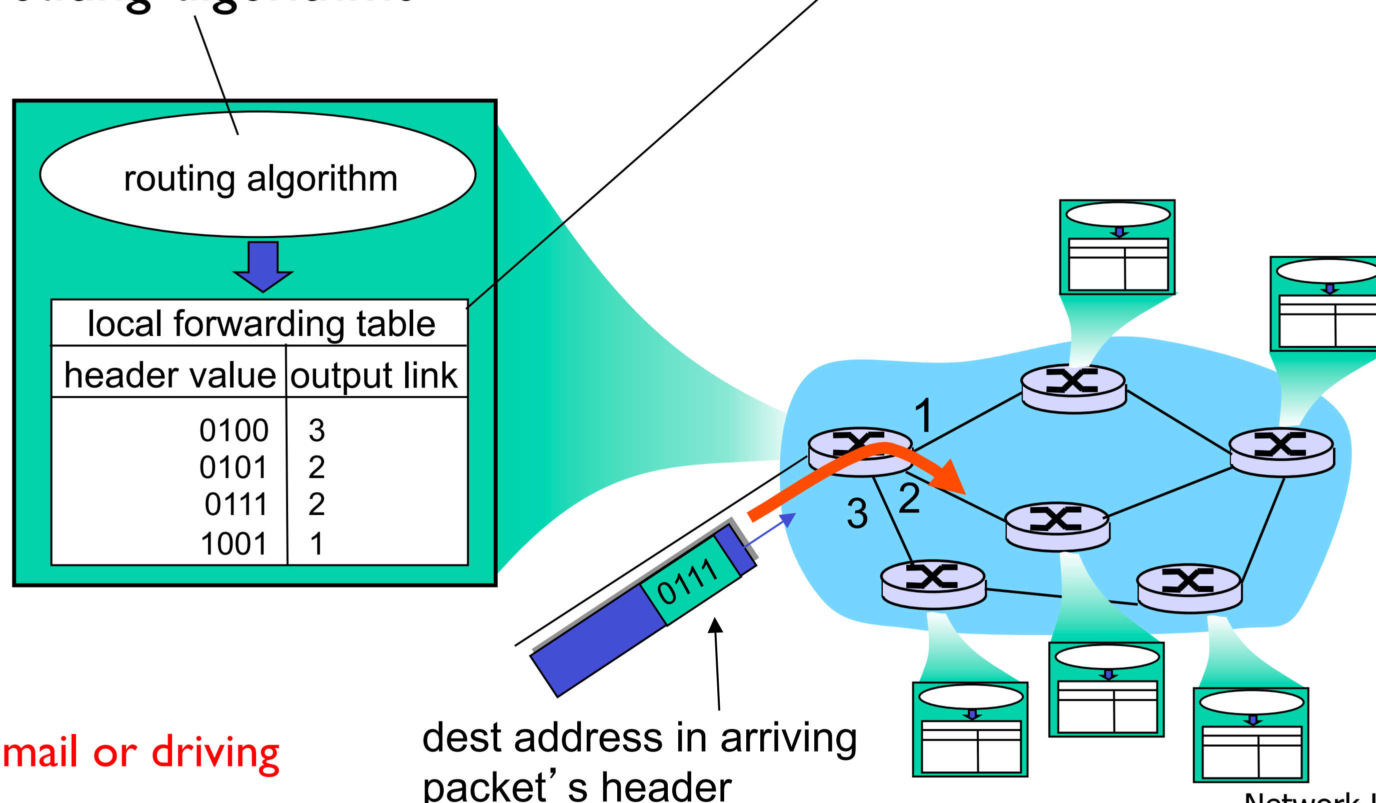


Two key network-core functions

routing: determines source-destination route taken by packets

- *routing algorithms*

forwarding: move packets from router's input to appropriate router output



Analogies: mail or driving

Chapter 1: roadmap

1.1 What is the Internet?

1.2 Network edge

- end systems, access networks, links

1.3 Network core

- circuit switching vs packet switching,
- network structure

1.4 Performance:

- delay, loss and throughput

1.5 Protocol layers, service models

1.6 Networks under attack: security

1.7 History