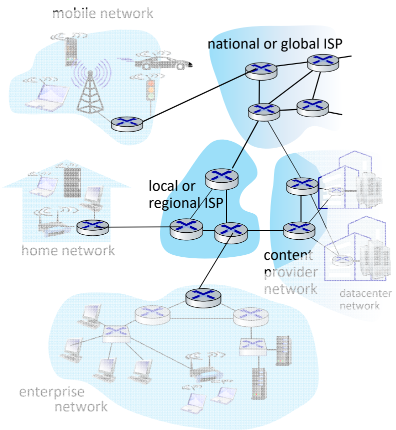
Chapter 1: roadmap

- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- **Network core:** packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History

The network core

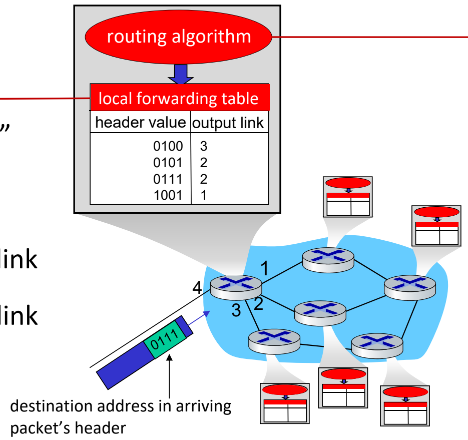
- composed of mesh of interconnected routers
- main goal: transfer data (for multiple sources-destination pairs) through network
  - circuit switching: dedicated circuit per call
  - packet switching: network forwards packets from one router to the next router, across *shared links* on path from source to destination



Two key network-core functions

Forwarding:

- aka "switching"
- *local* action: move arriving packets from router's input link to appropriate router output link

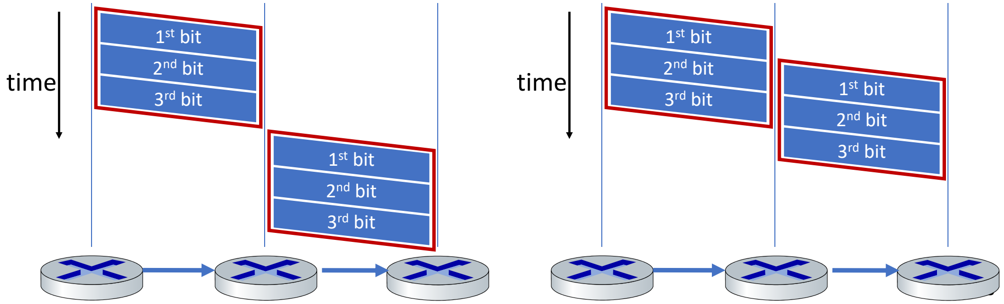


Routing:

- *global* action: determine source-destination paths taken by packets
- routing algorithms

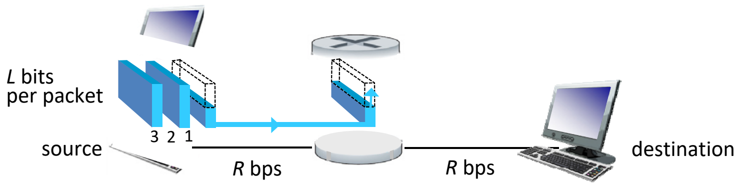
Packet switching: store-and-forward (mostly)

- store-and-forward
  - *entire* packet must arrive at router (and integrity is verified) before it can be transmitted on next link.
- cut-through
  - starts forwarding a packet before the whole packet has been received

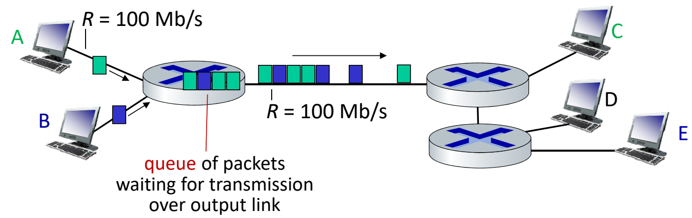


Packet switching: transmission delay

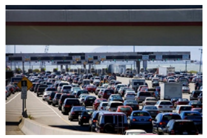
- transmission delay of a L-bit packet over a R-bps link
  - It takes L/R seconds
- transmission delay **caused by store-and-forward**
  - example: L = 10 Kbits and R = 100 Mbps
    - 1-hop transmission delay = 0.1 msec
    - 2-hop transmission delay = 0.2 msec



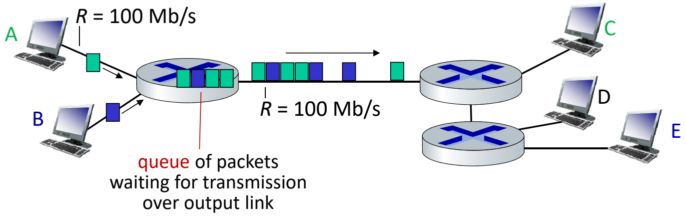
Packet switching: queueing



Queueing occurs when work arrives faster than it can be serviced:



Packet switching: queueing



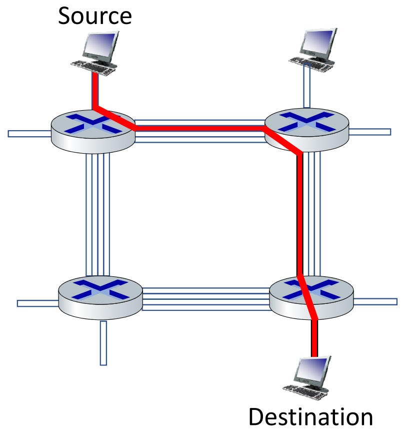
Packet queuing and loss: if arrival rate (in bps) to link exceeds transmission rate (bps) of link for some period of time:

- packets will queue, waiting to be transmitted on output link
- packets can be dropped (lost) if memory (buffer) in router fills up

Alternative to packet switching: circuit switching

end-to-end resources allocated to and reserved for a "call" between source and destination

- in diagram, each link has four circuits.
  - The call gets 2nd circuit in top link and 1st circuit in right link.
- dedicated resources: no sharing
  - circuit-like (guaranteed) performance
- circuit segment idles if not used by call
- commonly used in traditional telephone networks



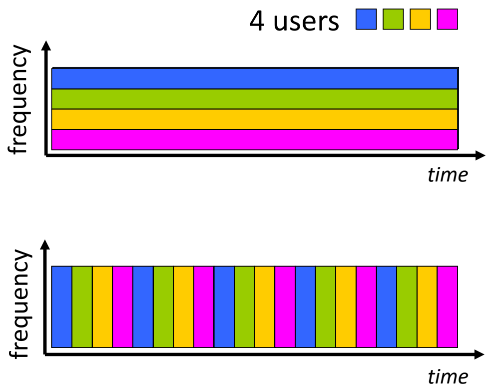
Circuit switching: FDM and TDM

Frequency Division Multiplexing (FDM)

- optical, electromagnetic frequencies divided into (narrow) frequency bands
- each call is allocated its own band and thus can transmit at max rate of that narrow band

Time Division Multiplexing (TDM)

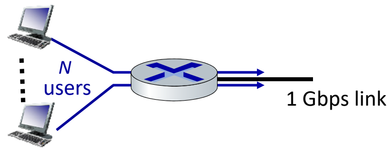
- time divided into slots
- each call is allocated periodic slot(s) and can transmit at maximum rate of (wider) frequency band (only) during its time slot(s)



Packet switching versus circuit switching

example:

- 1 Gb/s link
- each user:
  - 100 Mb/s when "active"
  - active 10% of time



Q: how many users can use this network under circuit-switching and packet switching?

- *circuit switching:*
  - at most ___ users
- *packet switching:*
  - at most ___ users

- *packet switching:*
  - with 35 users, the probability of >10 active users at same time is roughly 0.0004

Packet switching versus circuit switching

Is packet switching a “slam dunk winner”?

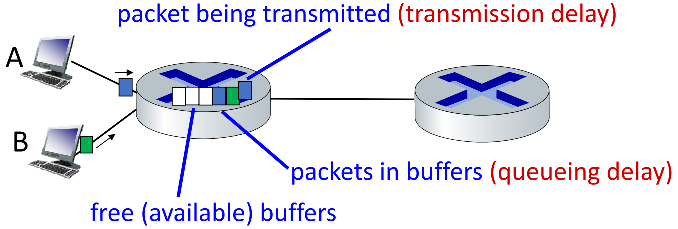
- great for “bursty” data – sometimes has data to send, but at other times not
  - allow more concurrent users
  - better resource sharing
  - simpler, no call setup
- excessive congestion is possible: packet delay and loss due to buffer overflow
  - protocols needed for reliable data transfer and congestion control
- *Q:* How to provide circuit-like behavior with packet switching?
  - “It’s complicated.” We’ll study various techniques that try to make packet switching as “circuit-like” as possible.

Chapter 1: roadmap

- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History

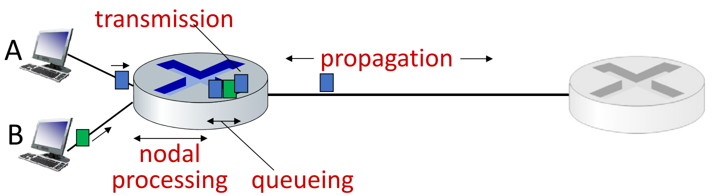
How do packet delay and loss occur?

- packets *queue* in router buffers, waiting for turn for transmission
  - queue length grows when arrival rate to link (temporarily) exceeds output link capacity
- packet loss occurs when buffer (aka. queue) fills up



- arriving packets are dropped (loss) if no free buffers

Packet delay: four sources

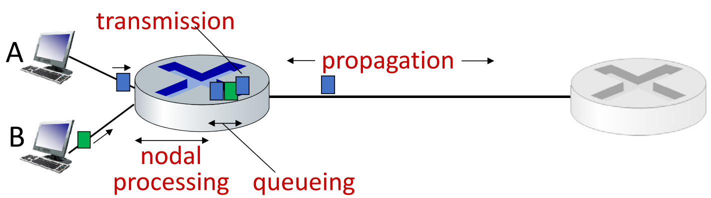


$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{proc}: nodal processing
- check bit errors
- determine output link
- typically < microsecs

d_{queue}: queueing delay
- time waiting at output link for transmission
- depends on congestion level of router

Packet delay: four sources



$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

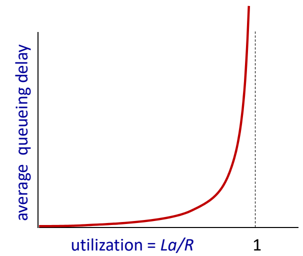
d_{trans}: transmission delay
- *L*: packet length (bits)
- *R*: link *transmission rate (bps)*
- $d_{trans} = L/R$

d_{prop}: propagation delay
- *d*: length of physical link
- *s*: propagation speed (~2×10^8 m/sec)
- $d_{\text{prop}} = d/s$

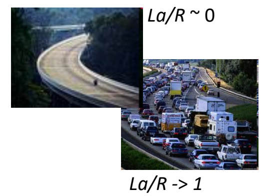
Queueing delay (revisited)

- *a:* average packet arrival rate (packets/s)
- *L:* packet length (bits/packet)
- *R:* link bandwidth [aka. transmission rate] (bits/s)

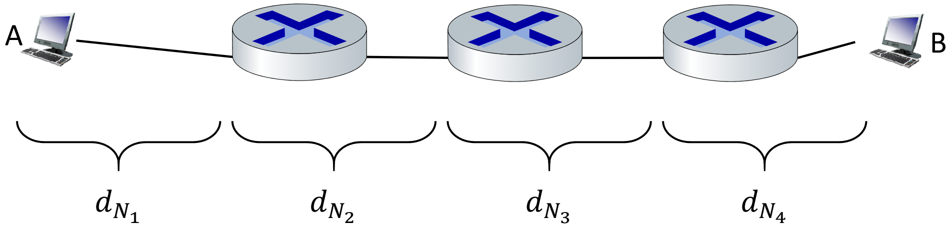
$$\frac{L \cdot a}{R} : \frac{\text{arrival rate in bits/s}}{\text{service rate in bits/s}}$$ *"utilization"*



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



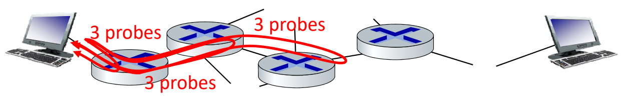
End-to-end delay



End-to-end delay from A to B $= d_{N_1} + d_{N_2} + d_{N_3} + d_{N_4}$

"Real" Internet delays and routes

- what do "real" Internet delay & loss look like?
- **traceroute** (or tracert) program: provides delay measurement from source to each router along the path towards destination. For each router *i:*
  - sends three packets that will reach router *i* on path towards destination (with time-to-live field value of *i*)
  - router *i* will return packets to sender
  - sender measures time interval between transmission and reply



Real Internet delays and routes

traceroute: gaia.cs.umass.edu to www.eurecom.fr

3 delay measurements from gaia.cs.umass.edu to cs-gw.cs.umass.edu

```
1  cs-gw (128.119.240.254)  1 ms  1 ms  2 ms
2  border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)  1 ms  1 ms  2 ms
3  cht-vbns.gw.umass.edu (128.119.3.130)  6 ms 5 ms 5 ms
4  jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)  16 ms 11 ms 13 ms
5  jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)  21 ms 18 ms 18 ms
6  abilene-vbns.abilene.ucaid.edu (198.32.11.9)  22 ms  18 ms  22 ms
7  nycm-wash.abilene.ucaid.edu (198.32.8.46)  22 ms  22 ms  22 ms
8  62.40.103.253 (62.40.103.253)  104 ms 109 ms 106 ms
9  de2-1.de1.de.geant.net (62.40.96.129)  109 ms 102 ms 104 ms
10  de.fr1.fr.geant.net (62.40.96.50)  113 ms 121 ms 114 ms
11  renater-gw.fr1.fr.geant.net (62.40.103.54)  112 ms  114 ms  112 ms
12  nio-n2.cssi.renater.fr (193.51.206.13)  111 ms  114 ms  116 ms
13  nice.cssi.renater.fr (195.220.98.102)  123 ms  125 ms  124 ms
14  r3t2-nice.cssi.renater.fr (195.220.98.110)  126 ms  126 ms  124 ms
15  eurecom-valbonne.r3t2.ft.net (193.48.50.54)  135 ms  128 ms  133 ms
16  194.214.211.25 (194.214.211.25)  126 ms  128 ms  126 ms
17  * * *
18  * * *
19  fantasia.eurecom.fr (193.55.113.142)  132 ms  128 ms  136 ms
```

3 delay measurements to border1-rt-fa5-1-0.gw.umass.edu

trans-oceanic link

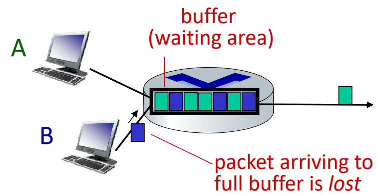
looks like delays *decrease*! Why?

* means no response (probe lost, router not replying)

* Do some traceroutes from exotic countries at www.traceroute.org

Packet loss

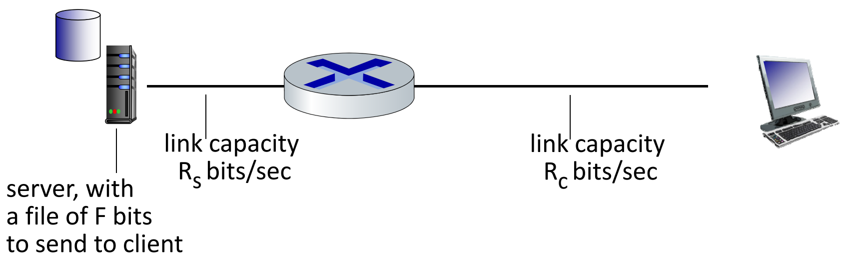
- As aforementioned,
  - queue (aka buffer) has finite capacity
  - packet that arrives to a full buffer is dropped



- lost packet may be retransmitted
  - by previous node,
  - by source end system,
  - or not at all

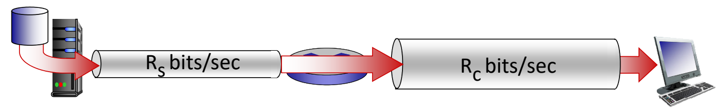
Throughput

- *throughput:* rate (bits/s) at which bits are being received at the destination
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time

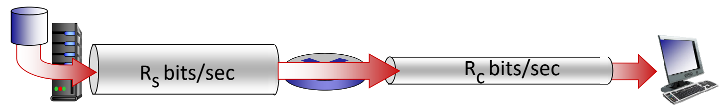


Throughput

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



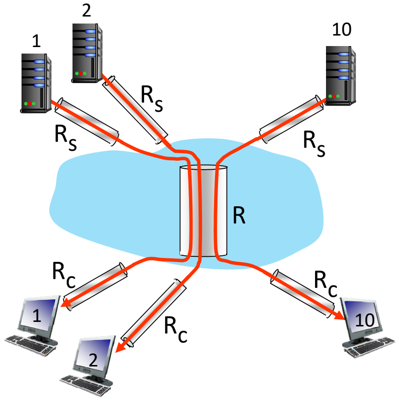
$R_s > R_c$ What is average end-to-end throughput?



bottleneck link

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

Throughput: network scenario



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

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