

Computer Networks

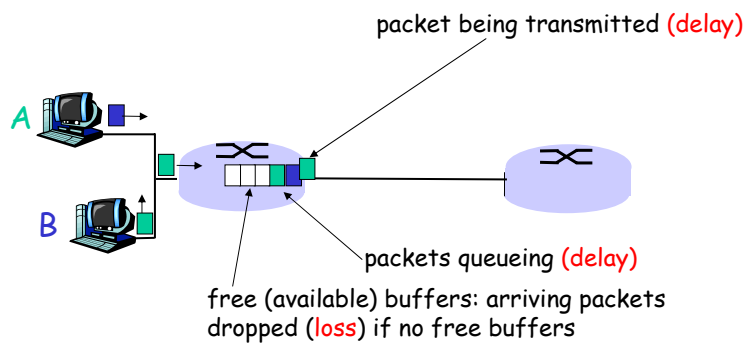
Delay, Loss and Throughput Layered architectures

Based on Computer Networking, 4th Edition by Kurose and Ross

Stan Kurkovsky

How do loss and delay occur?

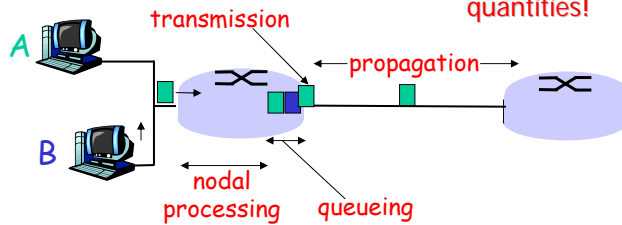
- packets queue in router buffers
- packet arrival rate to link exceeds output link capacity
- packets queue, wait for turn



Stan Kurkovsky

Four sources of packet delay

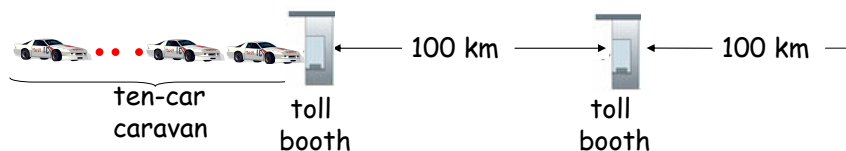
- Processing delay**
 - check bit errors
 - determine output link
 - $\sim 1 \times 10^{-6}$ sec
- Queueing**
 - time waiting at output link for transmission
 - depends on congestion level of router
- Transmission delay**
 - R = link bandwidth (bps)
 - L = packet length (bits)
 - time to send bits into link = L/R
 - $\sim 1 \times 10^{-6}$ to 1×10^{-3} sec
- Propagation delay**
 - d = length of physical link
 - s = propagation speed in medium $\sim 2\text{-}3 \times 10^8$ m/sec
 - propagation delay = d/s
 - **Note: s and R are very different quantities!**



Stan Kurkovsky

Caravan analogy

- Cars "propagate" at 100 km/hr
- Toll booth takes 12 sec to service a car (transmission time)
- Car ~ bit; caravan ~ packet
- **Q: How long until caravan is lined up before 2nd toll booth?**
- Time to "push" entire caravan through toll booth onto highway = $12 \times 10 = 120$ sec
- Time for last car to propagate from 1st to 2nd toll booth: $100\text{km}/(100\text{km/hr}) = 1$ hr
- **A: 62 minutes**
- Cars now "propagate" at 1000 km/hr
- Toll booth now takes 1 min to service a car
- **Q: Will cars arrive to 2nd toll booth before all cars serviced at 1st toll booth?**
- **A: Yes! After 7 min, 1st car at 2nd toll booth and 3 cars still at 1st toll booth.**
- 1st bit of packet can arrive at 2nd router before packet is fully transmitted at 1st router!
- [See animation](#)



Stan Kurkovsky

Nodal delay

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

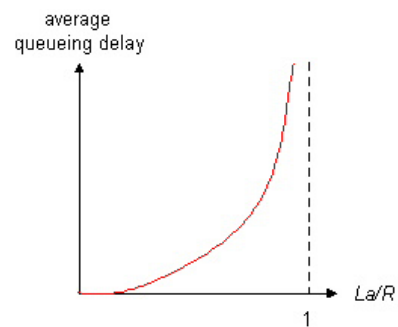
- d_{proc} = processing delay
 - typically a few microseconds or less
- d_{queue} = queuing delay
 - depends on congestion
- d_{trans} = transmission delay
 - = L/R , significant for low-speed links
- d_{prop} = propagation delay
 - a few microseconds to hundreds of milliseconds

Stan Kurkovsky

Queueing delay (revisited)

- R =link bandwidth (bps)
- L =packet length (bits)
- a =average packet arrival rate

- **traffic intensity = La/R**

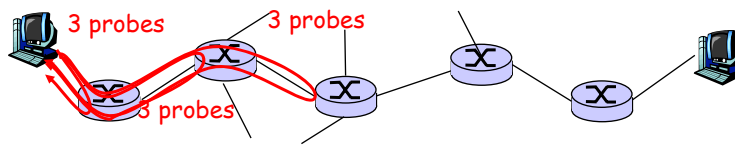


- $La/R \sim 0$: average queueing delay small
- $La/R \rightarrow 1$: delays become large
- $La/R > 1$: more "work" arriving than can be serviced, average delay infinite!

Stan Kurkovsky

"Real" Internet delays and routes

- What do "real" Internet delay & loss look like?
- **Traceroute program:** provides delay measurement from source to router along end-end Internet path towards destination.
- For all *i*:
 - sends three packets that will reach router *i* on path towards destination
 - router *i* will return packets to sender
 - sender times interval between transmission and reply.



Stan Kurkovsky

"Real" Internet delays and routes

Traceroute www.oxford.ac.uk
(from a home computer with SBCGlobal ADSL service)

Tracing route to www.oxford.ac.uk [163.1.0.90]
over a maximum of 30 hops:

```

1    2 ms    2 ms    2 ms  192.168.0.1
2    16 ms   17 ms   17 ms  bras7-10.mrdnct.sbcglobal.net [204.60.4.43]
3    15 ms   15 ms   14 ms  dist2-vlan60.mrdnct.sbcglobal.net [66.159.184.227]
4    14 ms   15 ms   14 ms  bbl-g1-0-1.mrdnct.sbcglobal.net [204.60.203.113]
5    17 ms   17 ms   17 ms  bbl-p4-0.nycmny.sbcglobal.net [151.164.42.99]
6    17 ms   17 ms   17 ms  core1-p4-0.crnyny.sbcglobal.net [151.164.240.34]
7    17 ms   17 ms   18 ms  core2-p1-0.crnyny.sbcglobal.net [151.164.188.82]
8    22 ms   23 ms   23 ms  core2-p3-0.crhva.sbcglobal.net [151.164.188.198]
9    23 ms   23 ms   23 ms  core1-p8-0.crhva.sbcglobal.net [151.164.188.21]
10   23 ms   22 ms   23 ms  bbl-p4-1.hrndva.sbcglobal.net [151.164.242.70]
11   23 ms   23 ms   22 ms  ex1-p5-0.egabva.sbcglobal.net [151.164.191.134]
12   25 ms   24 ms   23 ms  ash-bbl-geth6-3-6-13.telia.net [213.248.88.17]
13   29 ms   30 ms   29 ms  nyk-bbl-pos7-1-0.telia.net [213.248.80.70]
14  123 ms   96 ms   99 ms  ldn-bb2-pos7-1-0.telia.net [213.248.65.93]
15   98 ms   99 ms   98 ms  ldn-b3-pos9-0.telia.net [213.248.64.50]
16   99 ms   98 ms   99 ms  jnt-110793-ldn-b3.c.telia.net [213.248.100.238]
17   97 ms   99 ms   98 ms  po0-1.lond-scr4.ja.net [146.97.35.133]
18  107 ms  109 ms  106 ms  po1-0.read-scr.ja.net [146.97.33.26]
19  107 ms  109 ms  107 ms  po2-0.read-bar2.ja.net [146.97.35.106]
20  109 ms  110 ms  108 ms  pos2-0.oxford-bar.ja.net [193.63.108.74]
21  110 ms  108 ms  109 ms  146.97.40.82
22  109 ms  108 ms  113 ms  www.oxford.ox.ac.uk [163.1.0.90]

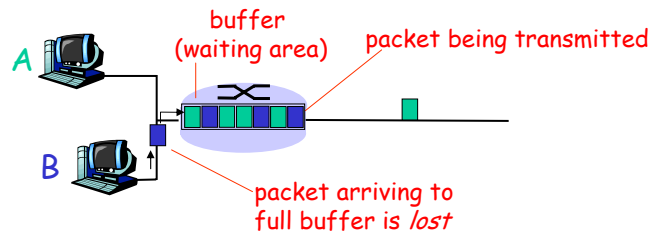
```

Trace complete.

Stan Kurkovsky

Packet loss

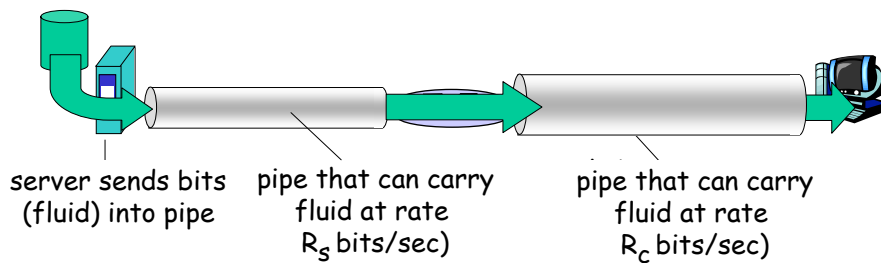
- queue (aka buffer) preceding link in buffer has finite capacity
- when packet arrives to full queue, packet is dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not retransmitted at all



Stan Kurkovsky

Throughput

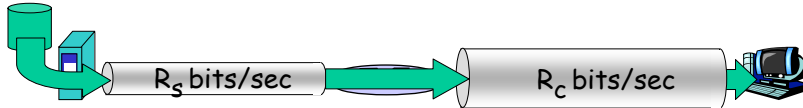
- *throughput*: rate (bits/time unit) at which bits transferred between sender/receiver
 - *instantaneous*: rate at given point in time
 - *average*: rate over long(er) period of time



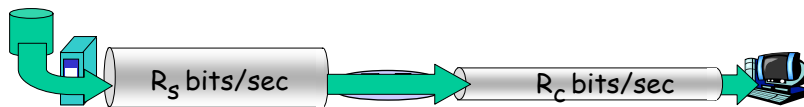
Stan Kurkovsky

Throughput (more)

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



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

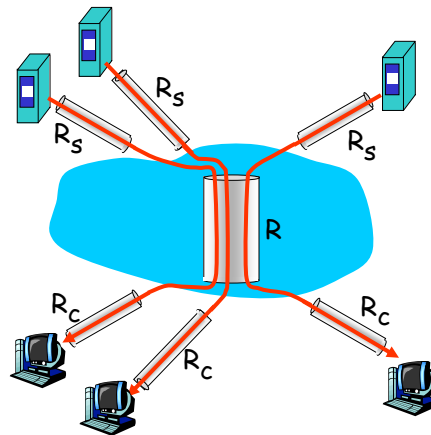


- **Bottleneck link:** link on end-end path that constrains end-end throughput

Stan Kurkovsky

Throughput: Internet scenario

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



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

Stan Kurkovsky

Protocol "Layers"

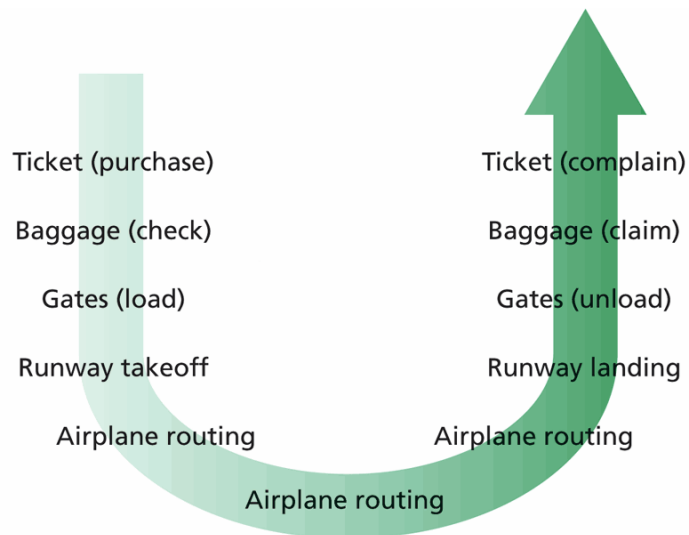
Networks are complex!

- many "pieces":
 - hosts
 - routers
 - links of various media
 - applications
 - protocols
 - hardware, software
- Question: Is there any hope of *organizing* structure of network?
 - Or at least our discussion of networks?

Stan Kurkovsky

Organization of air travel

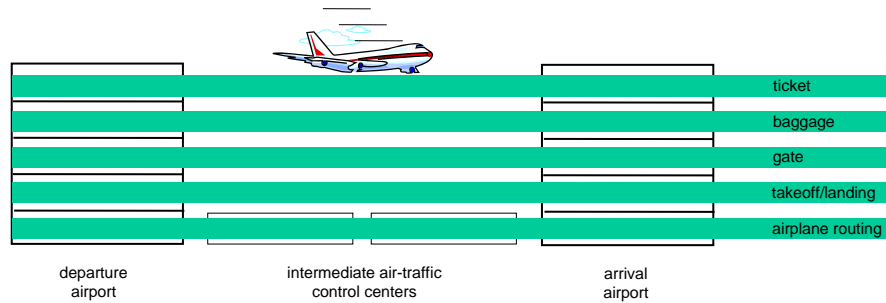
- a series of steps



Stan Kurkovsky

Layering of airline functionality

- **Layers:** each layer implements a service
 - via its own internal-layer actions
 - relying on services provided by layer below



Stan Kurkovsky

Why layering?

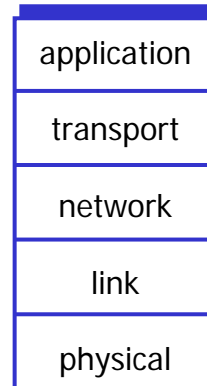
Dealing with complex systems:

- explicit structure allows identification, relationship of complex system's pieces
 - layered **reference model** for discussion
- modularization eases maintenance, updating of system
 - change of implementation of layer's service transparent to rest of system
 - e.g., change in gate procedure doesn't affect rest of system
- Service model: each layer provides a service by
 - Performing actions within that layer
 - Using the services of the layer directly below

Stan Kurkovsky

Internet protocol stack

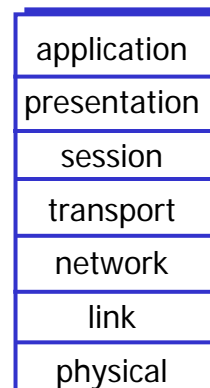
- **application:** supporting network applications
 - FTP, SMTP, HTTP
- **transport:** host-host data transfer
 - TCP, UDP
- **network:** routing of datagrams from source to destination
 - IP (defines structure of datagrams), routing protocols
- **link:** data transfer between neighboring network elements
 - PPP, Ethernet
- **physical:** bits "on the wire"



Stan Kurkovsky

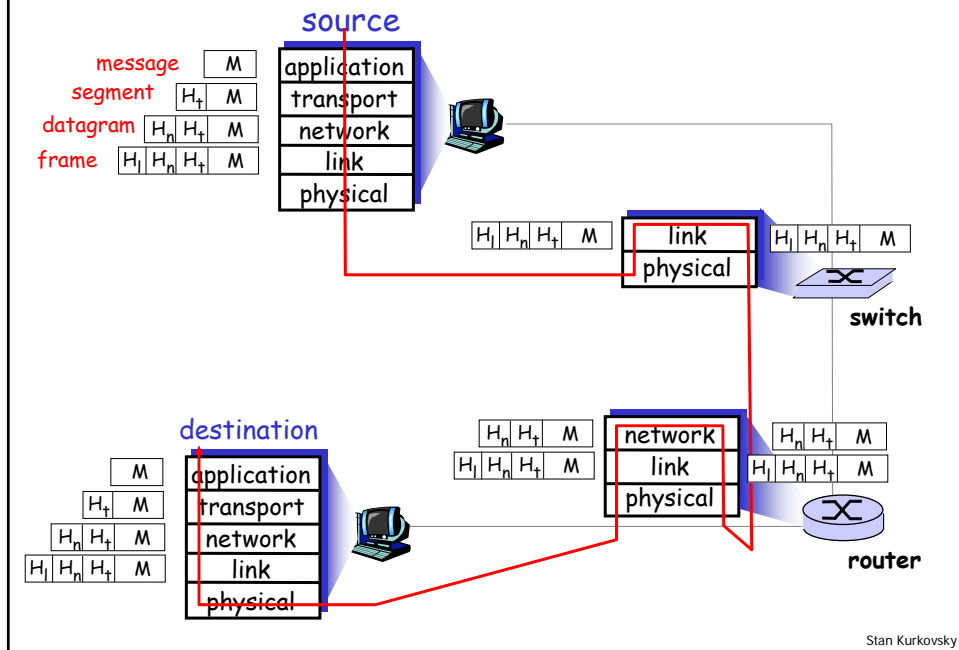
ISO/OSI reference model

- **presentation:** allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- **session:** synchronization, checkpointing, recovery of data exchange
- Internet stack "missing" these layers!
 - these services, *if needed*, must be implemented in application
 - needed?



Stan Kurkovsky

Encapsulation



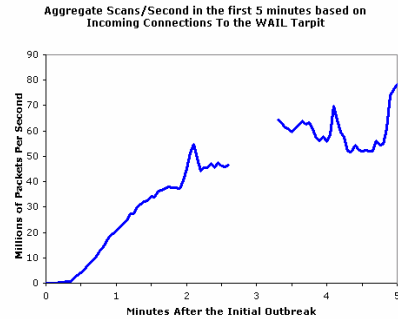
Network Security

- **attacks on Internet infrastructure:**
 - infecting/attacking hosts: malware, spyware, worms, unauthorized access (data stealing, user accounts)
 - denial of service: deny access to resources (servers, link bandwidth)
- **Internet not originally designed with (much) security in mind**
 - *original vision*: "a group of mutually trusting users attached to a transparent network"
 - Internet protocol designers playing "catch-up"
 - Security considerations in all layers!

Stan Kurkovsky

What can bad guys do: malware?

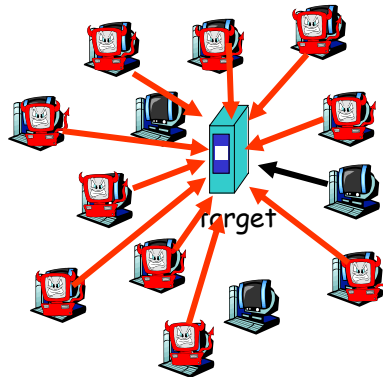
- **Spyware:**
 - infection by downloading web page with spyware
 - records keystrokes, web sites visited, upload info to collection site
- **Virus**
 - infection by receiving object (e.g., e-mail attachment), actively executing
 - self-replicating: propagate itself to other hosts, users
- **Worm:**
 - infection by passively receiving object that gets itself executed
 - self-replicating: propagates to other hosts, users



Stan Kurkovsky

Denial of service attacks

- attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic
1. select target
 2. break into hosts around the network (see malware)
 3. target from compromised hosts

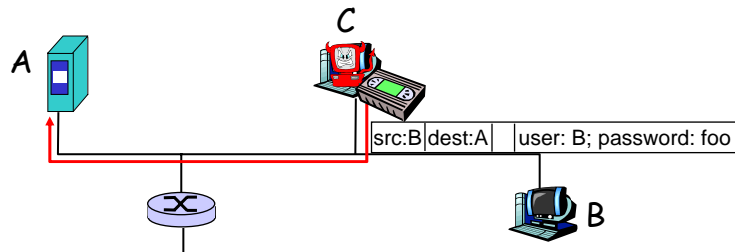


Stan Kurkovsky

Sniff, modify, delete your packets

Packet sniffing:

- broadcast media (shared Ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by
- *IP spoofing:* send packet with false source address
- *record-and-playback:* sniff sensitive info (e.g., password), and use later
 - password holder *is* that user from system point of view



Stan Kurkovsky