Introduction to Computer Networks

Outline

- Networking History
- Statistical Multiplexing
- Performance Metrics

A Brief History of Networking: early years

- Roots traced to public telephone network of the 60’s
  - How can computers be connected together?
- Three groups were working on packet switching as an efficient alternative to circuit switching
- L. Kleinrock had first published work in ‘61
  - Showed packet switching was effective for bursty traffic
- P. Baran had been developing packet switching at Rand Institute and plan was published in ‘67
  - Basis for ARPAnet
- First contract to build network switches awarded to BBN
- First network had four nodes in ‘69

History of the Internet contd.

- By ’72 network had grown to 15 nodes
  - Network Control Protocol - first end-to-end protocol (RFC001)
  - Email was first application – R. Tomlinson, ’72
- In ’73 R. Metcalfe invented Ethernet
- In ’74 V. Cerf and R. Kahn developed open architecture for Internet
  - TCP and IP

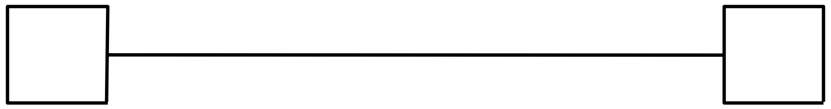
History of the Internet contd.

- By ' 79 the Internet had grown to 200 nodes and by the end of ' 89 it had grown to over 100K!
  - Much growth fueled by connecting universities
  - L. Landweber from UW was an important part of this!
- Major developments
  - TCP/IP as standard
  - DNS
- In ' 89 V. Jacobson made MAJOR improvements to TCP
- In ' 91 T. Berners-Lee invented the Web
- In ' 93 M. Andreesen invented Mosaic
- The rest should be pretty familiar…

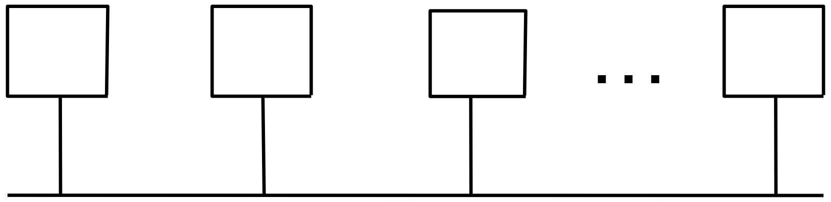
Building Blocks

- Nodes: PC, special-purpose hardware…
  - hosts
  - switches

- Links: coax cable, optical fiber…
  - point-to-point

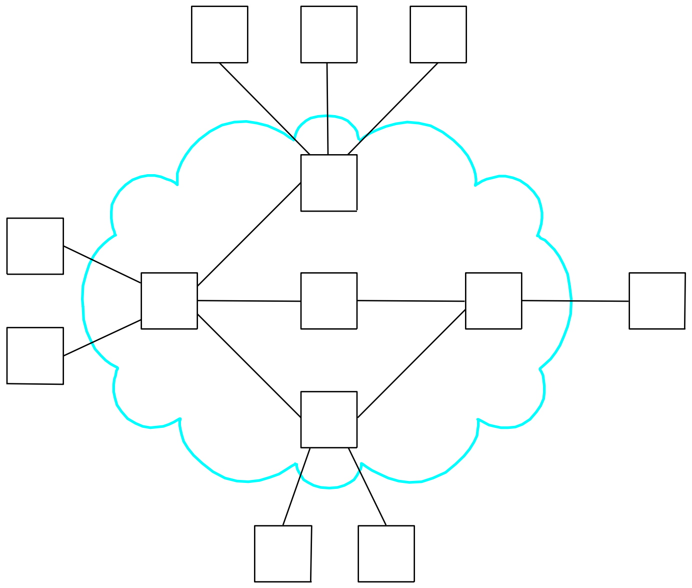


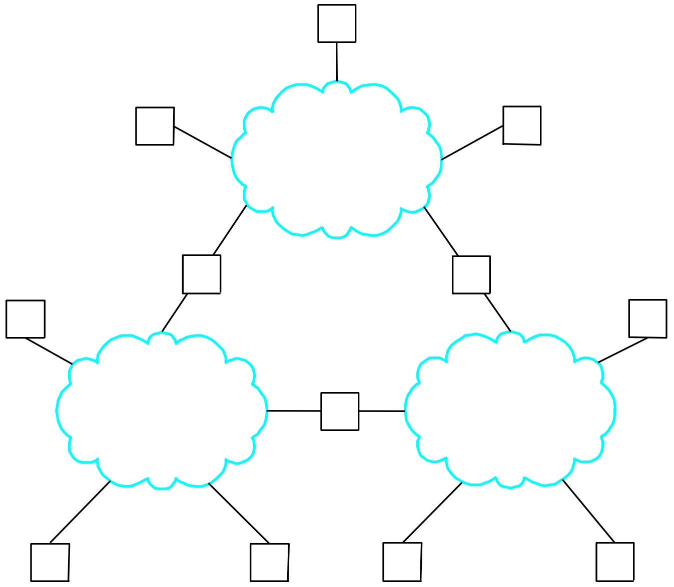
  - multiple access



Switched Networks

- A network can be defined recursively as...
  - two or more nodes connected by a link, or
  - two or more networks connected by two or more nodes





Strategies

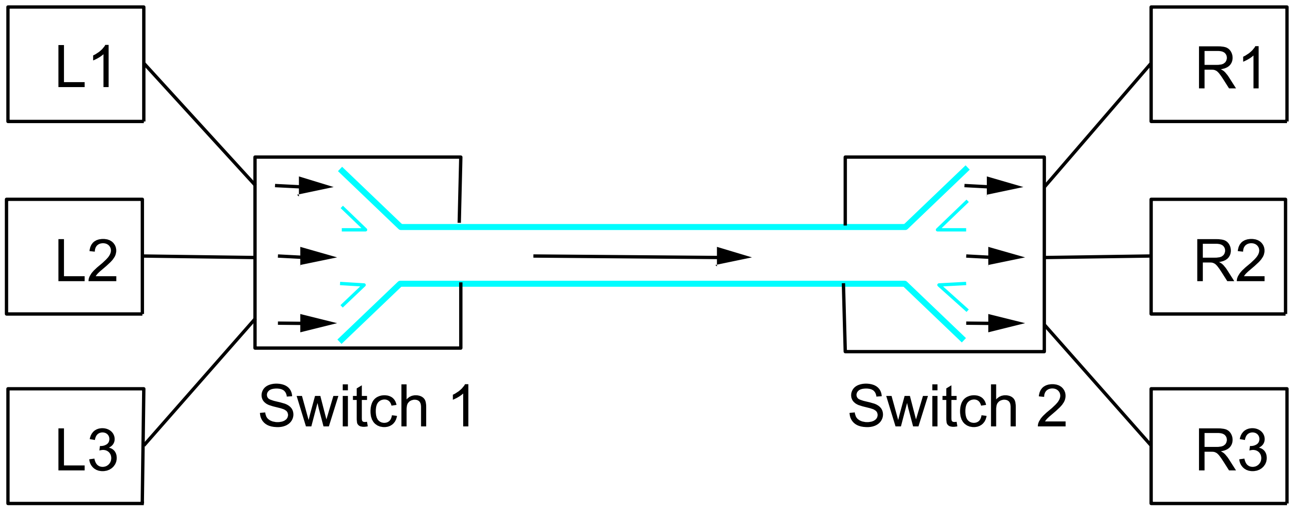
- Circuit switching: carry bit streams
  - original telephone network
- Packet switching: store-and-forward messages
  - Internet

Addressing and Routing

- Address: byte-string that identifies a node
  - usually unique
- Routing: process of forwarding messages to the destination node based on its address
- Types of addresses
  - unicast: node-specific
  - broadcast: all nodes on the network
  - multicast: some subset of nodes on the network

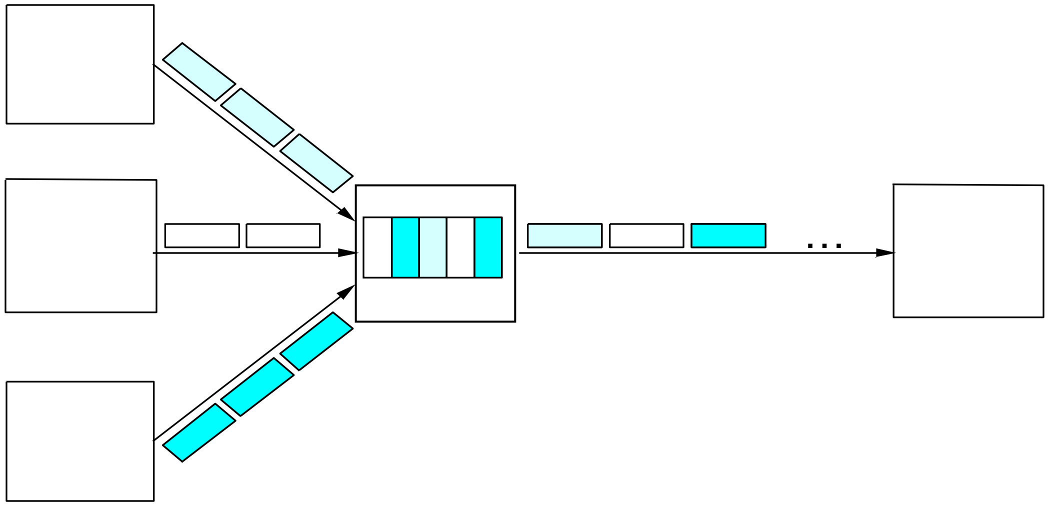
Multiplexing

- Time-Division Multiplexing (TDM)
- Frequency-Division Multiplexing (FDM)



Statistical Multiplexing

- On-demand time-division
- Schedule link on a per-packet basis
- Packets from different sources interleaved on link
- Buffer packets that are *contending* for the link
- Buffer (queue) overflow is called *congestion*

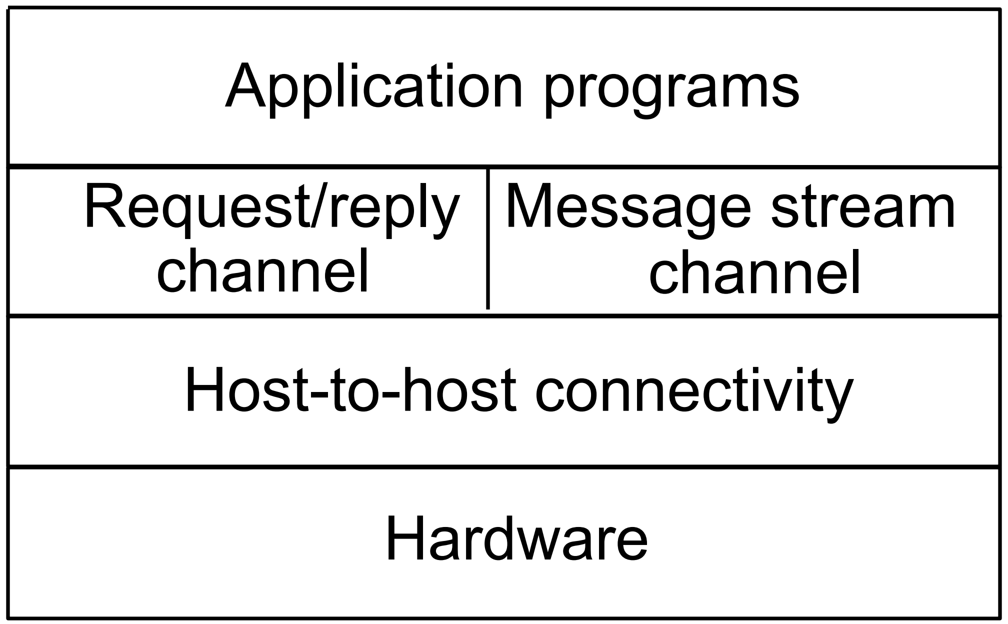


Example: Circuit vs. Packet Switching

- Suppose host A sends data to host B in a bursty manner such that 1/10th of the time A actively generates 100Kbps and 9/10th of the time A sleeps
  - Under circuit switching, given a 1Mbps link, how many users can be supported?
    - Answer: 10 with no delays for any user
  - Under packet switching given a 1Mbps links how many users can be supported?
    - Answer: about 30 with low probability of delay
  - **Point: 3 times more users can be supported!**

Layering

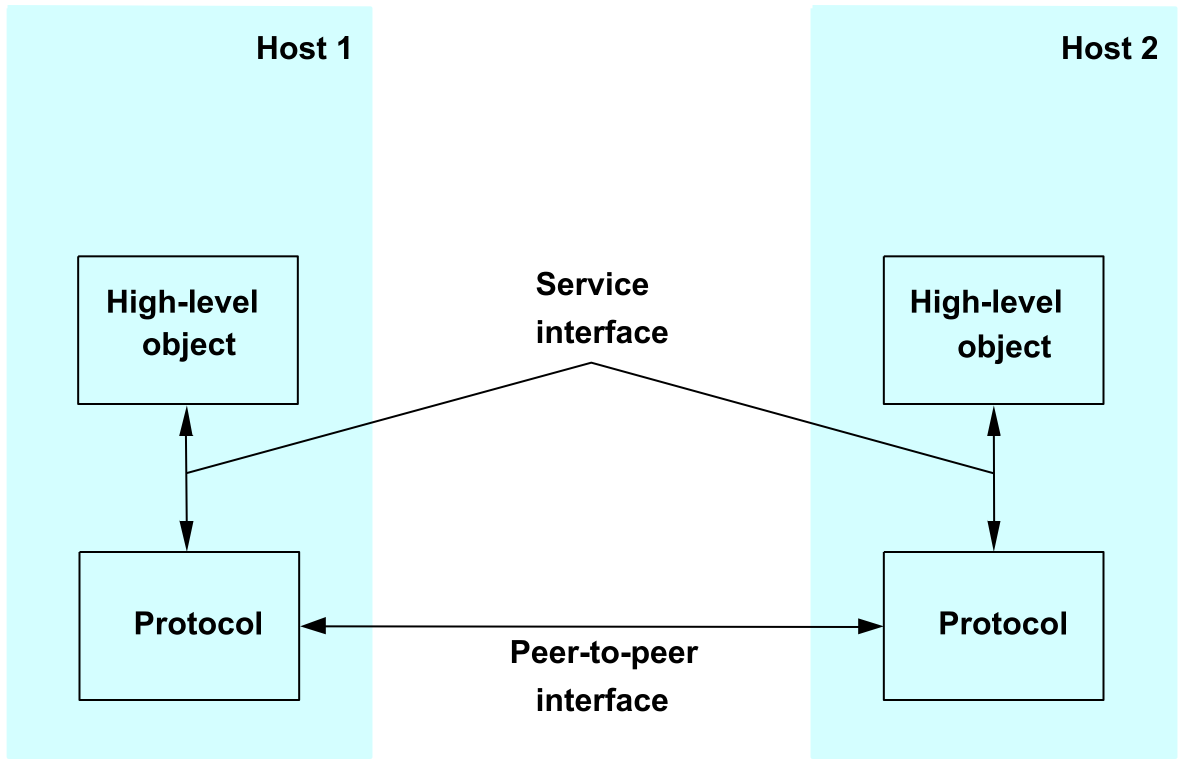
- Use abstractions to hide complexity
- Abstraction naturally lead to layering
- Alternative abstractions at each layer



Protocols

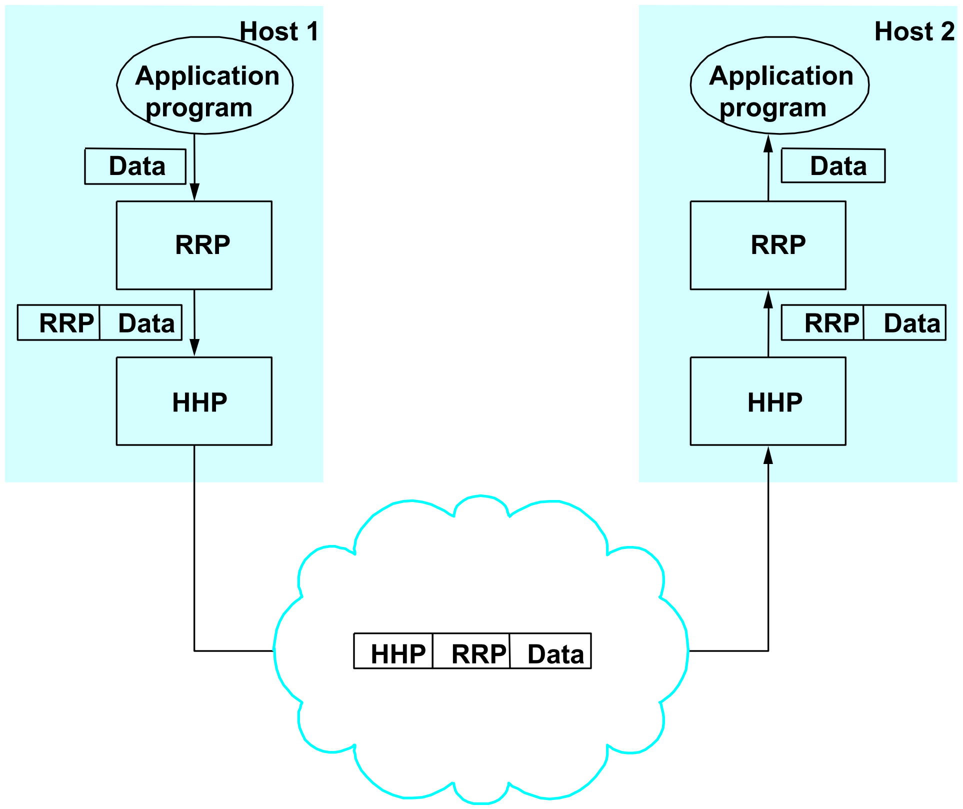
- Building blocks of a network architecture
- Each protocol object has two different interfaces
  - *service interface*: operations on this protocol
  - *peer-to-peer interface*: messages exchanged with peer
- Term “protocol” is overloaded
  - specification of peer-to-peer interface
  - module that implements this interface

Interfaces


Host 1
Host 2
High-level object
High-level object
Service interface
Protocol
Protocol
Peer-to-peer interface

Machinery

- Multiplexing and Demultiplexing (demux key)
- Encapsulation (header/body)



Performance Metrics

- Data rate or Throughput
  - data transmitted per time unit
  - link versus end-to-end
  - notation
    - KB = 2^{10} bytes
    - Mbps = 10^6 bits per second
- Latency (delay)
  - time to send message from point A to point B
  - one-way versus round-trip time (RTT)
  - components

    Transfer time = Propagation + Transmit + Queue

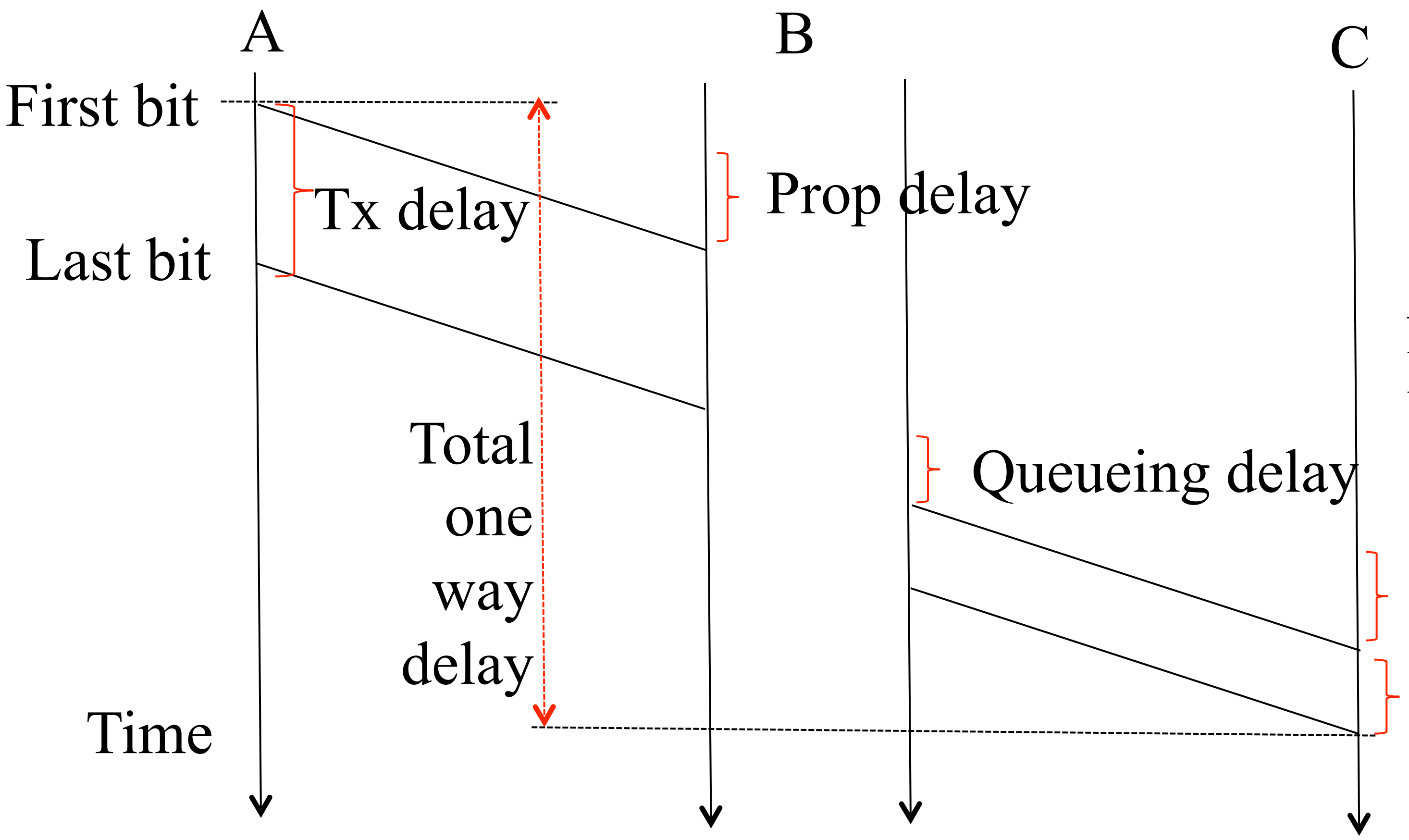
    Propagation = Distance / c

    Transmit = Transfer Size / Data rate

A note about terminology

- Bandwidth in the strict sense measures width of the operating band (units Hz, KHz, MHz)
  - A WiFi channel operates on a 20 MHz band
- Terms that represents rate at which data is sent are: data rate and throughput
  - How many bits did we get through per unit time (units bps, Kbps, Mbps)
  - Capacity --- what is the max rate at which you can send data through a link
- However, in networking parlance, bandwidth is also the term often used to mean data rate
  - Should be clear from context what bandwidth means

Understanding Latency



1 Kb data

A-B @ 1 Mbps

B-C @ 2 Mbps

What matters more in transfer time

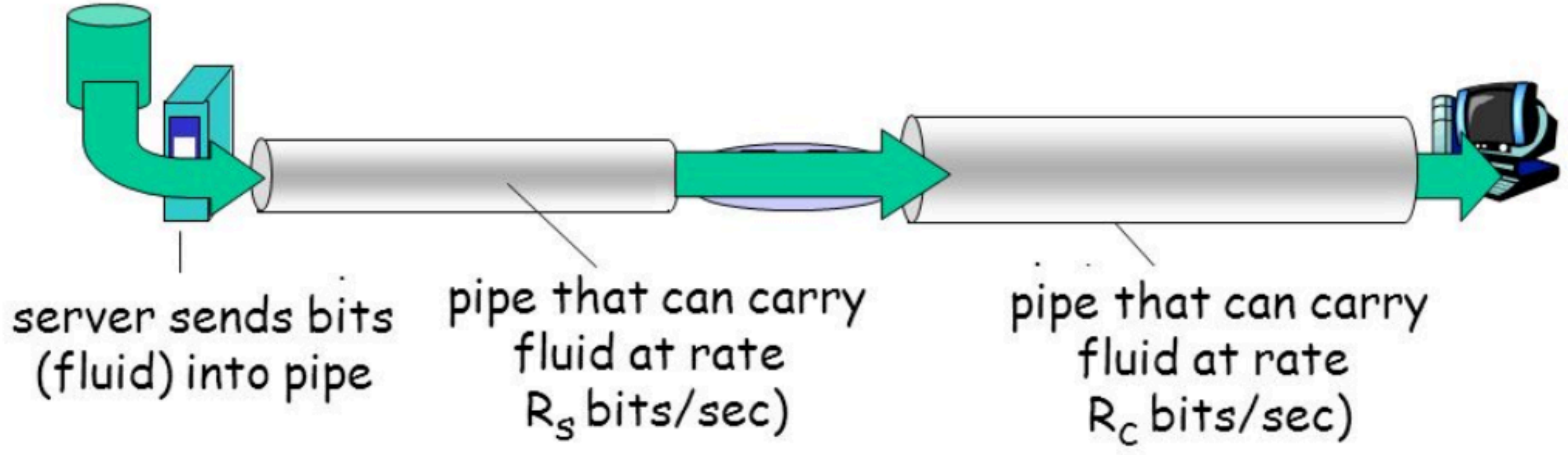
- The link data rate or propagation delay?

- Take an example:
  - Link is 6000 km long =20 ms (prop delay at speed of light)
  - Data rate is 1 Mbps
- When sending little data (say 1 Kb)
  - Transfer time = 1 ms txdelay + 20 ms prop delay + q' delay
- When sending a large amount of data (say, 100 Mb)
  - Transfer time = 100s txdelay + 20 ms prop delay + q' delay

One way delay vs Round Trip Time

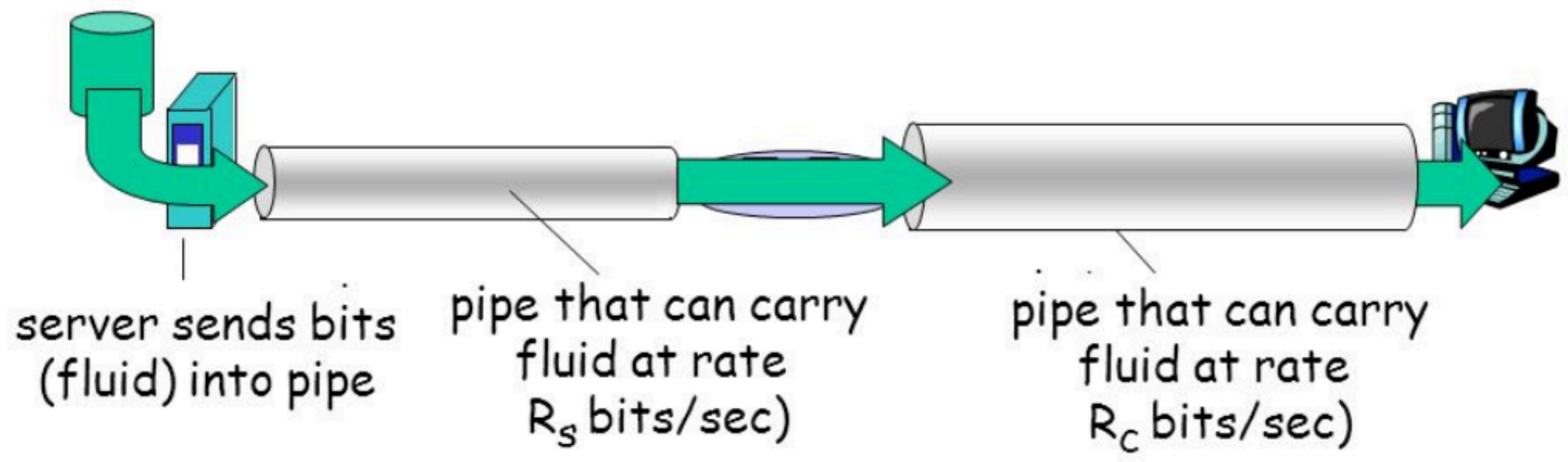
- One way delay is the time to send data from a source (S) to a destination (D)
- Round trip time is the sum of one way delays in both directions, i.e., S to D and D to S

Analogy of Capacity, Throughput, Propagation delay


server sends bits (fluid) into pipe
pipe that can carry fluid at rate R_s bits/sec)
pipe that can carry fluid at rate R_c bits/sec)

Propagation delay ~ lengths of the pipe
Capacity ~ width of the pipe(s)
Throughput ~ actual amount of content flowing through the pipe

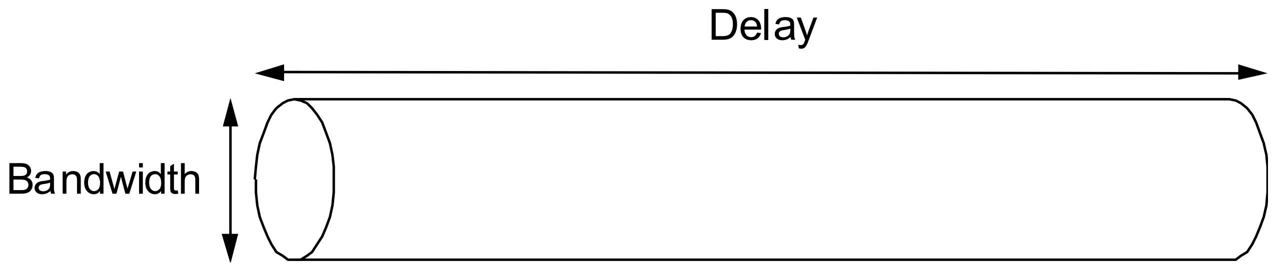
End-to-end capacity


server sends bits (fluid) into pipe
pipe that can carry fluid at rate R_s bits/sec)
pipe that can carry fluid at rate R_c bits/sec)

End-to-end capacity is determined by the bottleneck link

Delay x Bandwidth Product

- Amount of data “in flight” or “in the pipe”
- Example: 100ms x 45Mbps = 560KB



Sample problem

- Calculate the total time required to transfer a 1.5 MB file in the following cases, assuming a RTT of 80 ms, a packet size of 1 KB data, and an initial 2×RTT of "handshaking" before data is sent.

(a) The bandwidth is 10 Mbps, and data packets can be sent continuously.

(b) The bandwidth is 10 Mbps, but after we finish sending each data packet we must wait one RTT before sending the next.

(c) The link allows infinitely fast transmit, but limits bandwidth such that only 20 packets can be sent per RTT.

(d) Zero transmit time as in (c), but during the first RTT we can send one packet, during the second RTT we can send two packets, during the third we can send four = 23−1, and so on.

Solution

- We will count the transfer as completed when the last data bit arrives at its destination.

(a) 1.5 MB = 12,582,912 bits. 2 initial RTT's (160 ms) + 12,582,912/10,000,000 bps (transmit) + RTT/2 (propagation) ≈ 1.458 seconds.

(b) Number of packets required = 1.5 MB/1 KB = 1,536. To the above we add the time for 1,535 RTTs (the number of RTTs between when packet 1 arrives and packet 1,536 arrives), for a total of 1.458+122.8 = 124.258 seconds.

(c) Dividing the 1,536 packets by 20 gives 76.8. This will take 76.5 RTTs (half an RTT for the first batch to arrive, plus 76 RTTs between the first batch and the 77th partial batch), plus the initial 2 RTTs, for 6.28 seconds.

(d) Right after the handshaking is done we send one packet. One RTT after the handshaking we send two packets. At n RTTs past the initial handshaking we have sent 1 + 2 + 4 +···+ 2^n = 2^(n+1) − 1 packets. At n = 10 we have thus been able to send all 1,536 packets; the last batch arrives 0.5 RTT later. Total time is 2 + 10.5 RTTs, or 1 second.