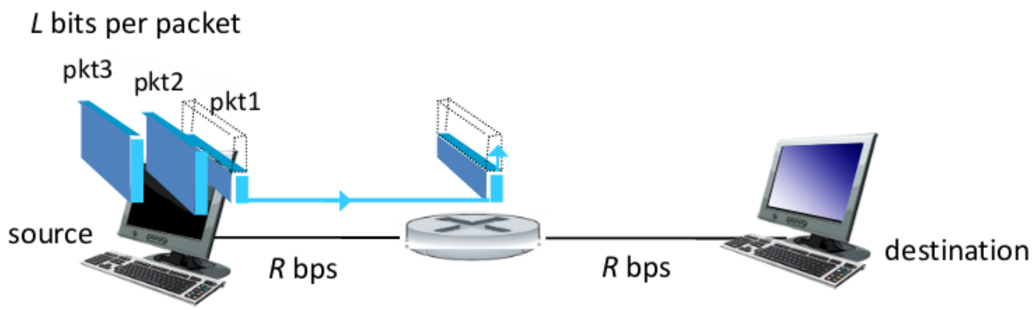
Network Core

1 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

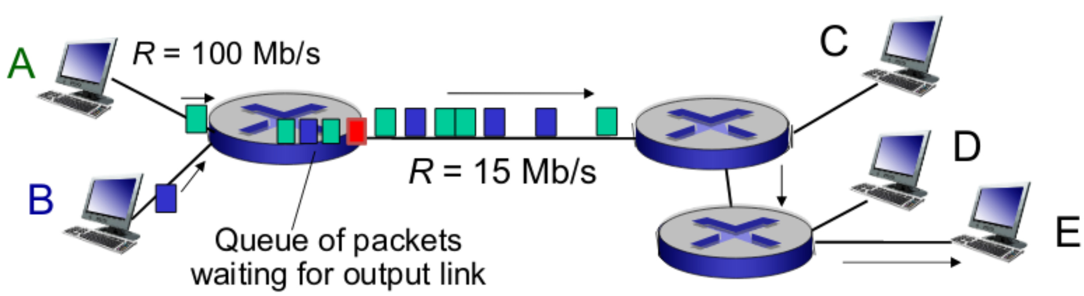
2 Packet-switching

2.1 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
- End-end delay = 2L/R (assuming zero propagation delay)

2.2 Queuing delay, loss



If arrival rate (in bits) to link exceeds transmission rate of link for a period of time

- Packets will queue, wait to be transmitted on link
- Packets can be dropped (lost) if memory (buffer) fills up

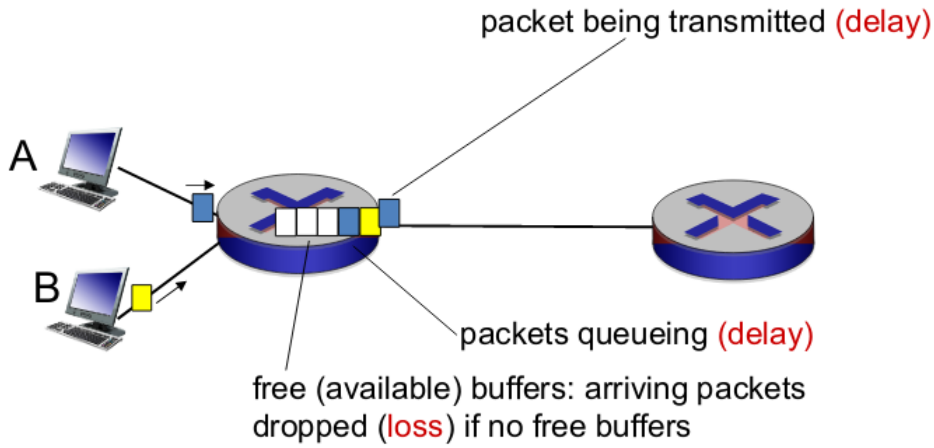
3 Two key network-core functions

Routing: Determines source-destination route taken by packets
Forwarding: Move packets from router's input to appropriate router output

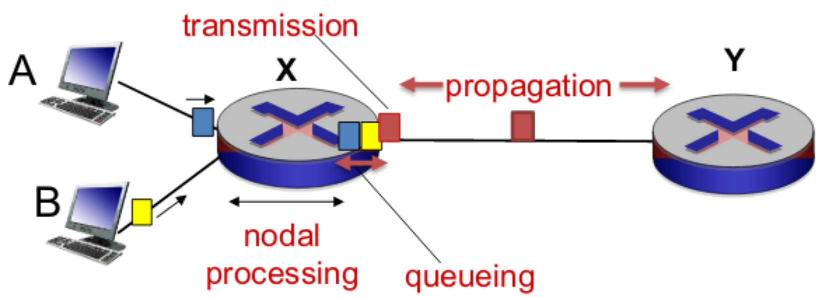
4 How do loss and delay occur?

Packets queue in router buffers

- Packet arrival rate to link (temporarily) exceeds output link capacity
- Packets queue, wait for turn



5 Four sources of packet delay



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

d_{proc}: nodal processing (delay in one router to process the packet)

- Check bit errors
- Find information to determine where to send packet
- Determine output link
- Typically < msec

d_{queue}: queuing delay

- Time waiting at output link for transmission
- Depends on congestion level of router

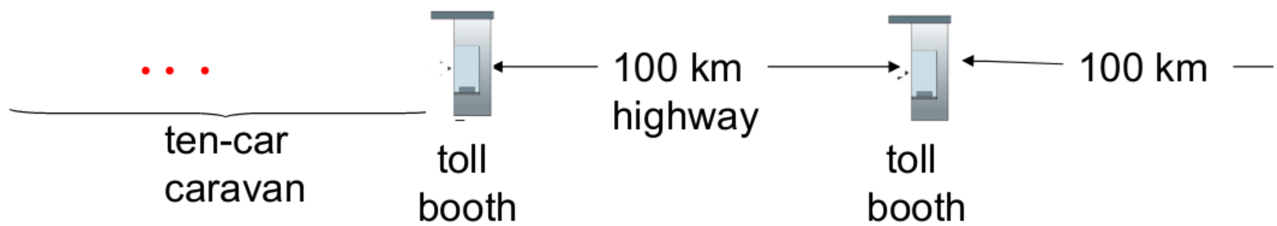
d_{trans}: Transmission delay

- How long it takes the packet to get out of the router
- L: Packet length (bits)
- R: Link bandwidth (bps)
- d_{trans}=L/R

d_{prop}: Propagation delay:

- Time for transmission of data between the routers
- d: length of physical link
- s: propagation speed ($\sim 2 \times 10^8 m/s$)
- $d_{prop} = d/s$

5.1 Caravan Analogy



Cars "propagate" at

- 100 km/hr
- Toll booth takes 12 seconds to service car (bit transmission)
- car ~ bit; caravan ~ packet

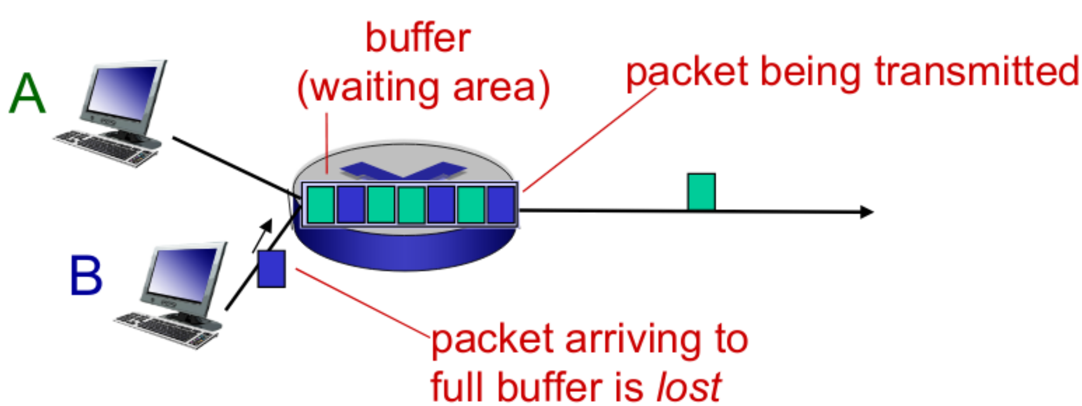
These caravans aren't actually caravans, instead a group of cars

Time to "push" entire caravan through toll booth onto highway=$12 \times 10 = 120$ seconds

Time for last car to propagate from 1st to 2nd toll booth =1hr

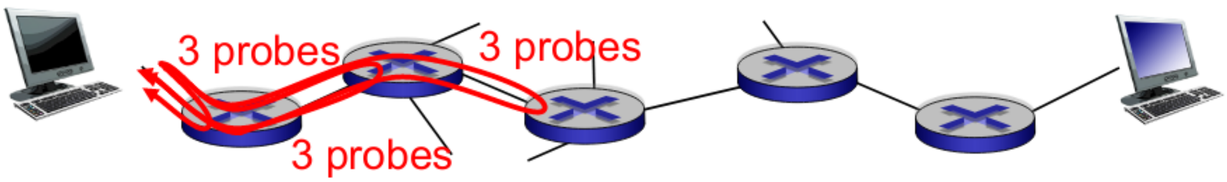
6 Packet Loss

- Queue (aka buffer) preceding link in buffer has finite capacity
- Packet arriving to a full queue dropped (aka lost)
- Lost packet may be retransmitted by previous node, bu source end system, or not at all

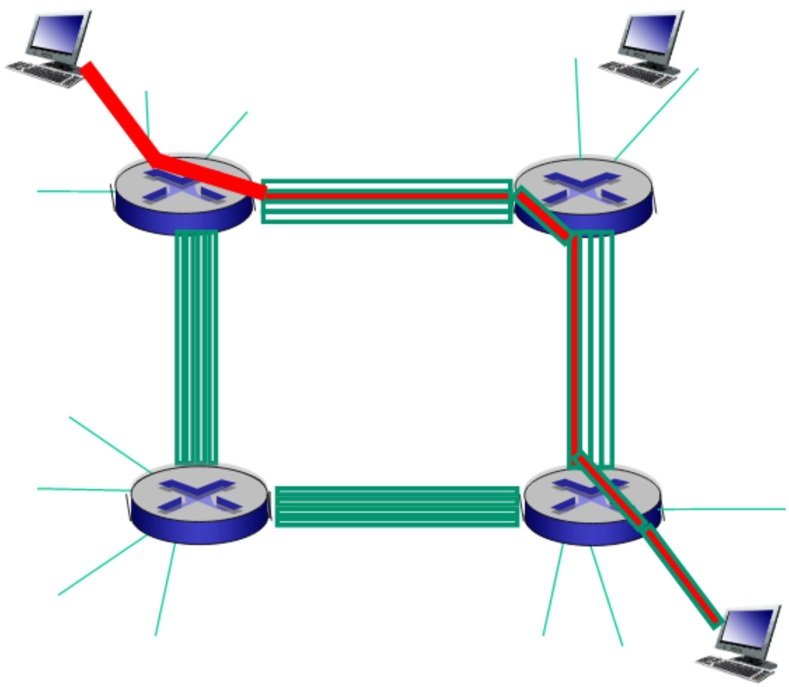


7 "Real" internet delays and routes

- 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 to router i on path towards destination
  - Router i will return packets to sender
  - Sender times interval between transmission and reply



8 Alternative core: circuit switching



End resources allocated to, reserved for "call" between source and destination

- In diagram, each link has four circuits. Call gets 2nd circuit in top link and 1st circuit in right link
- Dedicated resources: no sharing. Circuit-like (guaranteed) performance
- Circuit segment idle if not being used by call (no sharing)
- Commonly used in traditional telephone networks

9 Protocol "layers"

Protocols determine the format and order of messages between devices. Protocol layering has conceptual and structural advantages.
Protocol Stack: Protocols of the various layers

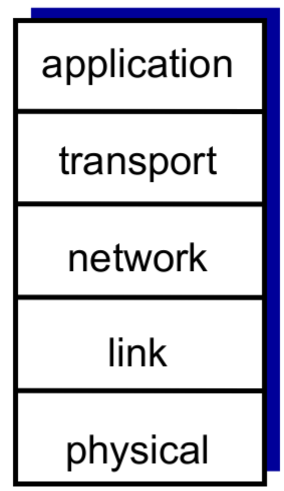
9.1 Why layering?

Dealing with complex systems:

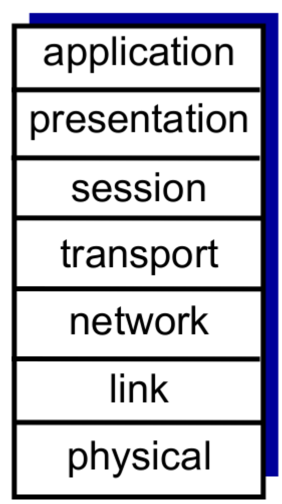
- Explicit structure allows identification, relationship of complex system's pieces
- Modularization eases maintenance, updating of system
  - Change of implementation of layer's service transparent to rest of system

9.2 Internet Protocol Stack

- **Application**: Supporting network applications
  - FTP, SMTP, HTTP
- **Transport**: process-process data transfer
  - TCP, UDP
- **Network**: Routing of datagrams from source to destination
  - IP, Routing Protocols
- **Link**: Data transfer between neighbouring network elements
  - Ethernet, 802.11, PPP
- **Physical**: Bits "on the wire"



9.3 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

10 Application Layer

Network Architecture:

- Client-server architecture
- P2P Architecture

Processes and Socket programming

- TCP
- UDP

11 Creating a network app

Write programs that:

- Run on (different) end systems
- Communicate over network

No need to write software for network-core devices

- Network-core devices do not run user applications
- Applications on end systems allow for rapid app development, propagation

12 Application Architectures

12.1 Client-Server Architecture

Server:

- Always-on host
- Fixed (static) IP address
- Data centres for scaling

Clients

- Communicate with server
- May be intermittently connected
- May have dynamic IP addresses
- Do not communicate directly with each other

12.2 P2P Architecture

- No always on server
- Arbitrary end systems directly communicate
- Peers request service from other peers, provide service in return to other peers
- Self scalability - new peers bring service capacity, as well as new service demands
- Peers are intermittently connected and change IP addresses

13 Processes communicating

Definition: Process

def:Process A program running within a host

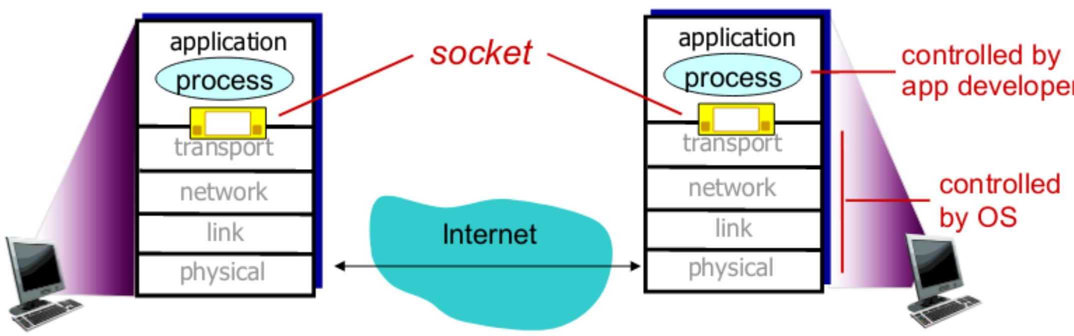
Definition: Socket

def:socket A software mechanism that allows a process to create and send messages into, and receive messages from the network

- Within same host, two processes communicate using inter-process communication (defined by OS)
- Processes in different hosts communicate by exchanging messages
- A process is analogous to a house, and its socket is analogous to its door
- Interface between application layer and transport layer

14 Sockets

- Process sends/receives messages to/from its socket
- A socket shoves message out of the door
- Sending process relies on transport infrastructure on the other side of the door to deliver message to a socket at the receiving process



15 What transport service does an app need?

Data integrity

- Some apps (e.g. file transfer, web transactions) require 100% reliable data transfer
- Other apps (e.g. audio) can tolerate some loss

Security:

- Encryption, data integrity, ...

Timing:

- Some apps (e.g. internet telephony, interactive games) require low delay to be ”effective”