



Introduction

S-38.188 - Computer Networks - Spring 2005

General requirements

- So, you want to build a network ...
 - First you need to know the **requirements** the network must satisfy
 - The requirements vary depending on who you ask (different views)
- Requirements from different views:
 - Application programmer: service specific needs, e.g., packets sent should not get lost and should arrive in the same order
 - Network designer: cost effective design, efficient and fair usage of network resources
 - Network provider: easy management, reliable, fault isolation
 - Users expect services: e-mail, tele- and videoconferencing, e-commerce, video-on-demand, ...

Computer network characteristics

- Typically communications networks optimized for some service
 - telephone network
 - television/radio broadcast network
 - user terminals special purpose devices
- Modern computer networks are more general:
 - terminals general purpose PCs/workstations
 - networks able to carry essentially any kind of data
 - support many different applications
- Topics in this lecture
 - How computer networks provide connectivity (Requirement 1)
 - How efficient resource sharing is achieved (Requirement 2)
 - How applications “talk” to each other (Requirement 3)
 - How network performance affects the system (Requirement 4)
 - Requirements are reflected in network architectures
 - Basically, we get a “snap shot” of the issues covered in this course

3

Outline

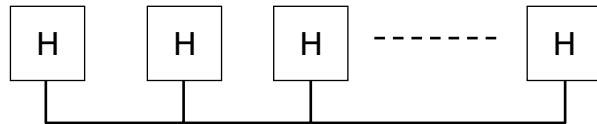
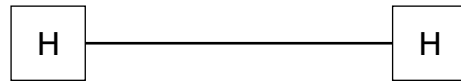
- Achieving connectivity
- Methods for resource sharing
- Enabling application level communication
- Performance issues
- Network architecture
- Historical background

4

Basic building blocks

- A network, in principle, consists of **nodes** and **links** connecting the nodes.
- Network nodes: PCs, servers, special purpose hardware
 - Internet terminology
 - hosts, end-systems: PCs and servers running network applications
 - routers (switches): store and forward packets through the network
- Links: optical fiber, coaxial cable, twisted pair copper, radio, etc.

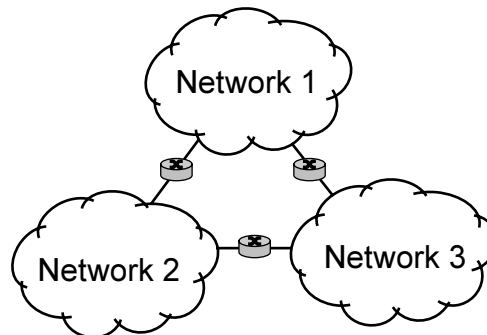
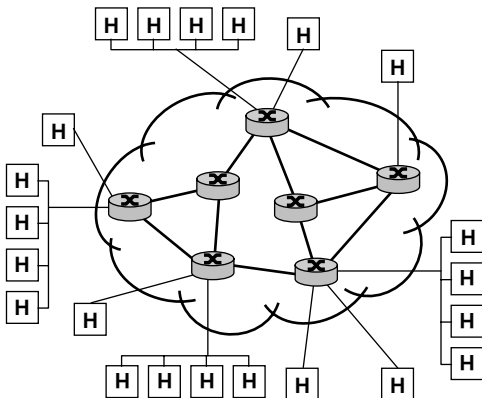
- point-to-point
 - hosts directly connected
- multiple access (LANs, etc.)
 - hosts share the common transmission medium



5

Building larger networks

- Large networks can not be built based on point-to-point connectivity
 - ⇒ use routers (switches) to interconnect hosts to each other
- Nodes connected together through switches to form connected networks
- Networks connected together through gateway routers to form bigger entities



6

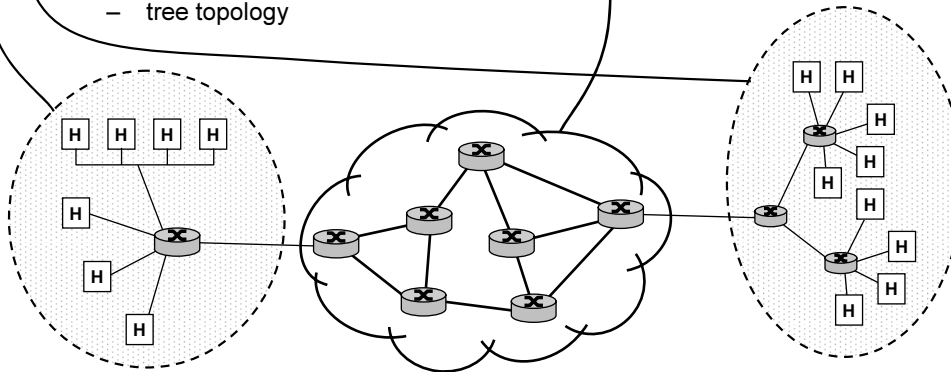
Network edge vs. core network in the Internet

- Access network

- customers are connected to the core network by the access network
- link speeds comparably low
 - access technologies: dial up (modem over twisted pair), xDSL, cable modem, ...
- may contain billing functionality, traffic management for each access
- tree topology

- Core network

- no end users directly connected to the core
- high link speeds
 - SDH/SONET over fiber based technologies
- simple functionality (forwards packets)
- mesh topology

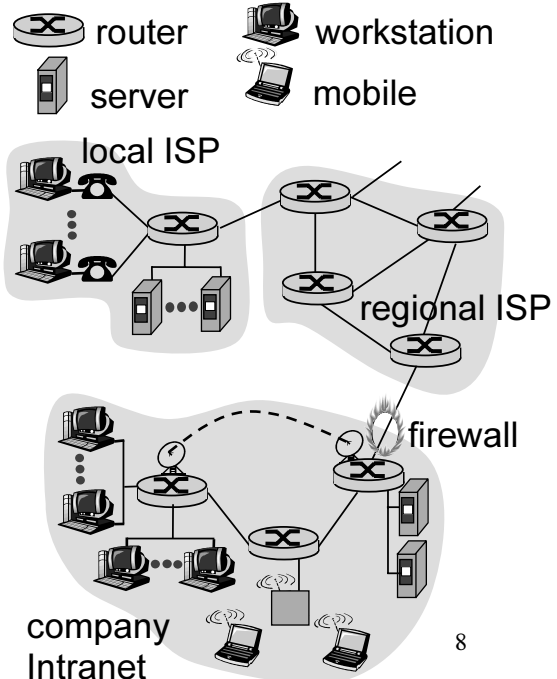


7

Internet¹

¹Jim Kurose, lecture notes for the course MPSCI 591E Computer Networking, <http://www-net.cs.umass.edu/cs591/>

- Consists of millions of hosts (end systems) connected by links and routers
- Hosts exchange messages by using **protocols** offering e.g.
 - reliable transfer
 - packet sequence integrity
- Routers forward data
 - based on best effort service
 - no guarantees on loss or timeliness
- “Network of networks”
 - loosely hierarchical
 - public Internet vs. private Intranets
 - Internet access provided by ISPs (Internet Service Providers)



8

Issues of scale

- Easy to build and manage a network supporting 100 users, but what if the number of users is 100 million ...
- A system allowing unlimited growth in size is said to **scale**.
 - Scalability a very desirable property for networking technologies
- Scalability of networks is often influenced very much by
 - the nature of the guarantees regarding service quality
 - the amount of information that the network has about the users
- One reason for the success of Internet technology is its scalability
 - The networking paradigm is based on best-effort service (no guarantees are made about the service quality) and the network is connectionless
 - The nodes of the network do not store any state information of the users/connections
 - New nodes and users can be added to the network (almost) without any complexity increases
 - Only the routing is affected by the increase in the number of nodes (route computation complexity grows with the number of nodes)

9

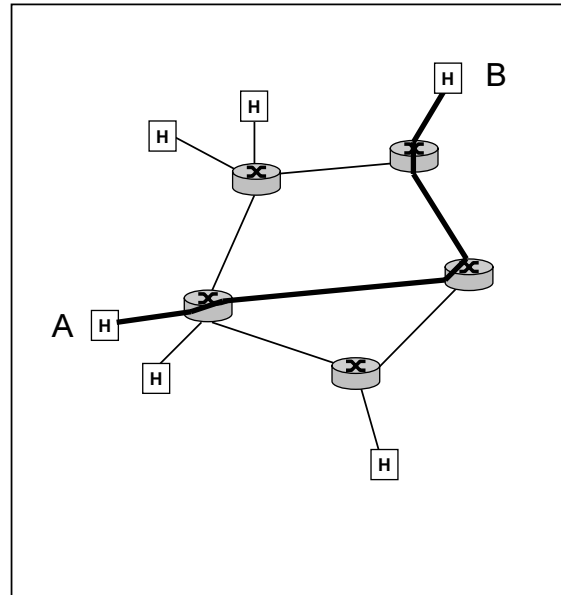
Switching modes

- **Circuit switching**
 - telephone networks
 - mobile telephone networks
 - ATM technology is also based on circuit switching
 - fast packet switching with fixed length packets (cells): ATM
 - integration of different traffic types (voice, data, video)
 - ⇒ multiservice networks
- **Packet switching**
 - data networks
 - two possibilities
 - **connectionless**: e.g. Internet (IP), SS7 (MTP)
 - **(connection oriented)**: e.g. X.25, Frame Relay

10

Circuit switching

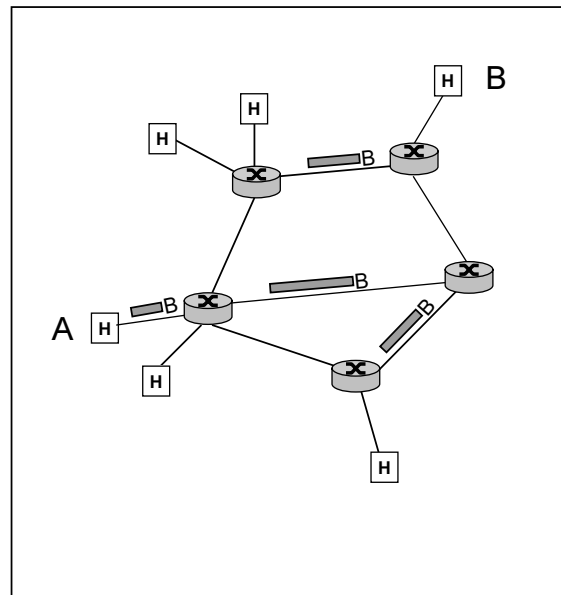
- **Connection oriented:**
 - connections **set up** end-to-end before information transfer
 - resources **reserved** for the whole duration of connection
- Information transfer as a **continuous stream**
- Before information transfer
 - delay (to set up the connection)
- During information transfer
 - no overhead
 - no extra delays



11

(Connectionless) packet switching

- **Connectionless:**
 - no connection set-up
 - no resource reservation
- Information transfer by using **discrete packets**
 - varying length
 - global address (of the destination)
- Before information transfer
 - no delays
- During information transfer
 - overhead (header bytes)
 - packet processing delays
 - queuing delays (since packets compete for shared resources)
 - routers “store-and-forward”



12

Addressing and routing

- Hosts need to distinguish each other when wishing to communicate
- Each host is assigned a unique byte-string known as **address**
- When a sender communicates with some destination B, in a packet switched network
 - the address of the destination (B) is attached to each packet, and
 - each router determines how to forward the packet based on the destination address
 - **routing** is the systematic process of determining where a packet is sent (which output port) based on the destination address
- Different addressing and routing scenarios
 - **unicast**: between a single sender and destination pair
 - **broadcast**: from a single user to all other users (e.g. network control messages)
 - **multicast**: from a single user to a subset of all users (e.g. distribution of files)

13

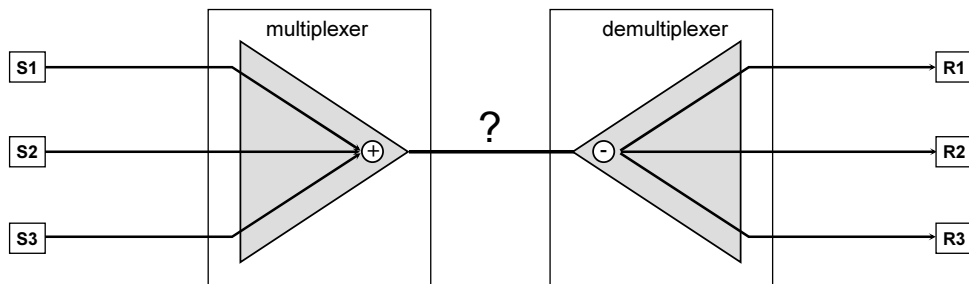
Outline

- Achieving connectivity
- **Methods for resource sharing**
- Enabling application level communication
- Performance issues
- Network architecture
- Historical background

14

Multiplexing

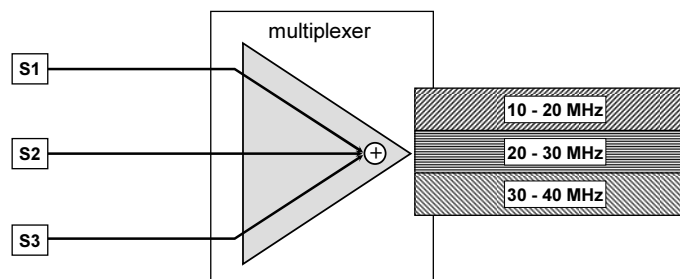
- Multiplexing
 - mechanism for achieving **resource sharing**, i.e., sharing of **link bandwidth**
- Problem:
 - How can the link bandwidth be shared among n different senders
- 1st approach: partition the bandwidth **strictly** for all users
 - FDM and TDM



15

Frequency Division Multiplexing (FDM)

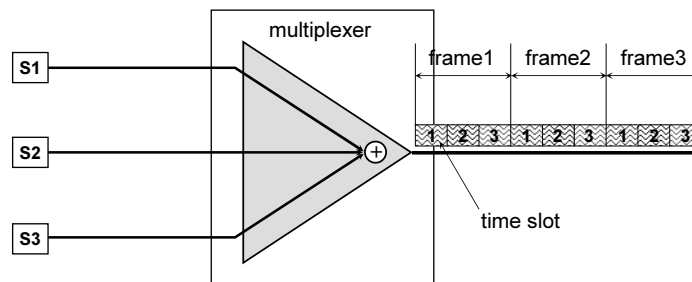
- FDM
 - oldest multiplexing technique
 - used e.g. in analogue circuit switched systems
 - fixed portion (frequency band) of the link bandwidth reserved for each channel
- FDM multiplexer is lossless
 - input: n 1-channel physical connections
 - output: 1 n -channel physical connection



16

Time Division Multiplexing (TDM)

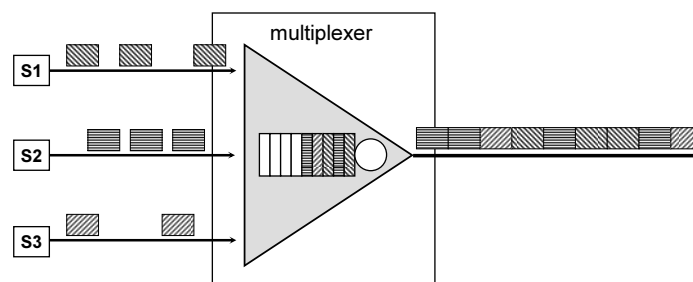
- TDM
 - used in digital circuit switched systems and digital transmission systems
 - information conveyed on a link transferred in **frames** of fixed length
 - fixed portion (time slot) of each frame reserved for each channel
- TDM multiplexer is lossless
 - input: n 1-channel physical connections
 - output: 1 n -channel physical connection



17

Statistical multiplexing (1)

- FDM and TDM are inefficient
 - If a sender has no data to transmit, the bandwidth allocated to the sender can not be used by others \Rightarrow **statistical multiplexing**
- In statistical multiplexing
 - basic transmission unit is called a packet
 - physical link is shared over time (cf. TDM) but **on-demand** (per each packet)
 - simultaneous packet arrivals are buffered (contention)
 - as a result, packets from multiple senders are *interleaved* at the output
 - buffer space is finite, thus buffer overflow is possible (congestion)



18

Statistical multiplexing (2)

- Statistical multiplexer is (typically) lossy
 - input: n physical connections with link speeds $R_i (i = 1, \dots, n)$
 - output: 1 physical connection with link speed $C \leq R_1 + \dots + R_n$
- However, the loss probability can be decreased by enlarging the buffer
 - with an “infinite” buffer enough that C exceeds **average aggregated** input rate
 - possible to dimension the size of the buffer such that a given loss probability is achieved (under some assumptions regarding the traffic)
- Statistical multiplexer and QoS (Quality of Service)
 - determining which packet to transmit from the buffer is called **scheduling**
 - FIFO: packets are served in the arrival order
 - Round robin: each connection (class) has own queue and they are served cyclically according to some weights
 - Many more exist...
 - by using different scheduling mechanisms, some connections can be given “preferential” treatment (e.g., weighted round-robin) \Rightarrow QoS enabled networks

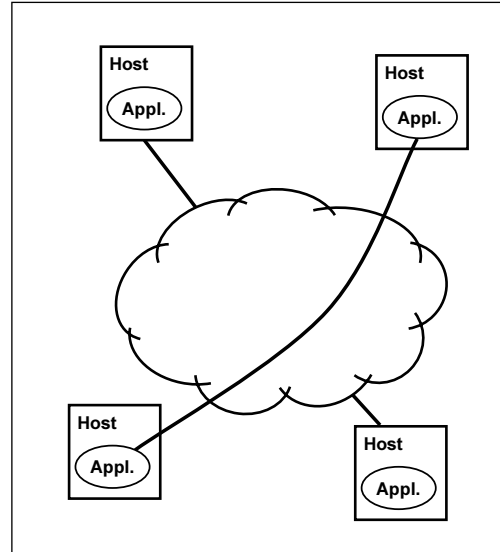
19

Outline

- Achieving connectivity
- Methods for resource sharing
- Enabling application level communication
- Performance issues
- Network architecture
- Historical background

Communication needs of applications

- Applications (processes) running on hosts need to communicate
 - different applications have different needs
- Typical application considerations
 - reliability?
 - packet sequence order?
 - security?
- Network design challenge
 - identify the set of common services that the applications need
 - hide the complexity of the network without imposing too many constraints on the applications
- Network provides “logical channels”
 - IPC = Inter Process Communication
 - fills in the “logical gap”



21

Application requirement classification

- 1 Client/server applications (request/reply applications)
 - **client** process makes a request and the **server** process replies
 - strict requirements on packet loss (no loss), may have security requirements
 - Examples: file transfer (FTP), file systems (NFS), HTML documents on the web, digital libraries
 - 2 Streaming applications
 - sender generates a **continuous** stream of packets
 - the stream can correspond to, e.g., digitized audio or video
 - applications have relatively tight requirements on the **timeliness** of packet delivery, but they can tolerate packet loss to some degree
 - videoconferencing has tighter demands than video on-demand
 - Security? Conferencing may require, e.g., encrypted transmission...
- Question 1: Are only 2 categories enough?
 - Question 2: Where is the functionality of each service implemented?

22

Reliable transfer - what can go wrong?

- Reliable transfer: one of the most important service properties
 - “network hides certain failures to make the network seem more reliable”
- Error types
 - **Bit errors:** bit or burst of bits is corrupted
 - Error correction detection may be able to fix the problem
 - **Packet errors:** complete packet is lost
 - Due to unrecoverable bit errors, congestion (most likely reason), software errors (misplaced packets, relatively rare)
 - Problem: Not easy to distinguish between packets that are excessively late (due to e.g. severe overload) and actually lost packets.
 - **Node/link failures:**
 - A physical link is damaged/cut, router crashes ...
 - Can cause massive service disruptions
 - In Internet routing protocols can recover from link failures
 - Problem: Not easy to determine if a router is e.g. completely down or just congested.

23

Outline

- Achieving connectivity
- Methods for resource sharing
- Enabling application level communication
- Performance issues
- Network architecture
- Historical background

24

Performance measures (1): bandwidth

- Bandwidth = throughput
 - no. of bits that can be transmitted over the network in a given time
 - unit: bits per second (bps), e.g. 10 Mbps (cf. MB = megabytes = 8 Mb)
- Link bandwidth and end-to-end bandwidth
 - bandwidth of a physical link has a deterministic value, e.g. 155 Mbps
 - link bandwidths are constantly improving: link bandwidths in the backbone
 - 1980's: 2 Mbps, 1990's: 155 Mbps, 2000: 1 Gbps
 - end-to-end the received bandwidth of an application depends on
 - other traffic in the network (congestion)
 - application limitations (CPU speed of the computer)
 - protocol overhead (each bit sent by the application is “wrapped” in possibly several “envelopes” until the bit is transmitted on a physical link)

25

Performance measures (2): latency

- Latency = delay
 - How long it takes a message to travel from one end of the nw to another
 - Measured in units of time, e.g., latency across US continent 24 ms
 - RTT (round trip time): time it takes a message to reach its destination and come back to the sender
- Components: propagation delay, transmission delay, queuing delay

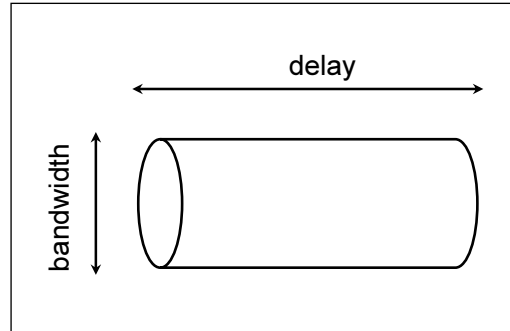
Latency	=	Propagation + Transmit + Queue
Propagation	=	Distance/SpeedOfLight
Transmit	=	Size/Bandwidth

- Speed of Light: 2.3×10^8 m/s in cable, 2.0×10^8 m/s in fiber
- Applications can be either bandwidth or latency bound
 - Telnet sessions are latency bound but large FTP transfers are bw bound

26

Delay x bandwidth product

- The product of RTT and bandwidth determines
 - the amount of information transmitted by the user before any feed-back from the destination can be received
- In broadband wide-area-networks (WAN) this product can be very large
 - the sender can overload the receiver
 - if the sender does not “fill in the pipe”, the network utilization may be low
- Example:
 - Assume that
 - distance is 1500 km
 - transmission rate $C = 100$ Mbps
 - The two-way propagation delay is
 - $2 \cdot 1500 / 300,000 \text{ s} = 0.01 \text{ s}$
 - Thus, the product of RTT and C is
 - $0.01 \cdot 100,000,000 \text{ bits} = 1,000,000 \text{ bits} = 1 \text{ Mbit}$



27

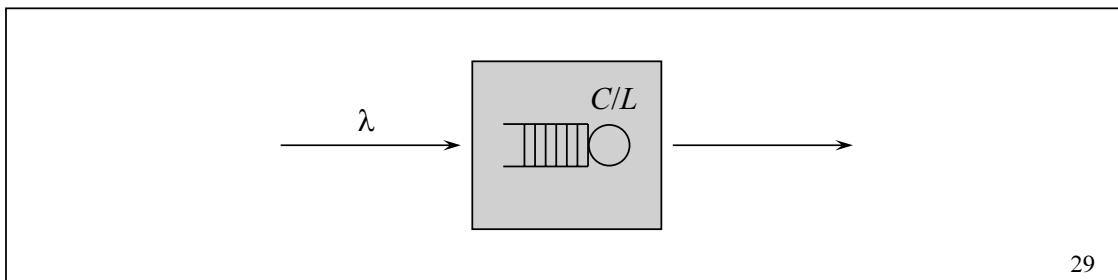
Delay x bandwidth product in high speed networks

- Assume $RTT = 100 \text{ ms}$, we aim to transmit a file of size 1 MB
 - 1 Mbps network: time to transmit = $80 \times RTT$
 - 80 pipes full of data (stream of data to send)
 - clearly, the network design problem would be to increase the bandwidth
 - 1 Gbps network: time to transmit = $0.08 \times RTT$
 - only 8 % of the pipe is filled (the file has become a single “packet”)
 - now, the latency dominates the network design
- Thus, coping with the delay seems like the main design issue in future high speed networks
- Applications have other performance requirements than delay and bandwidth
 - Applications may have an upper bound on required bandwidth
 - Real time applications have requirements on delay variation (jitter) caused by queuing in the network routers

28

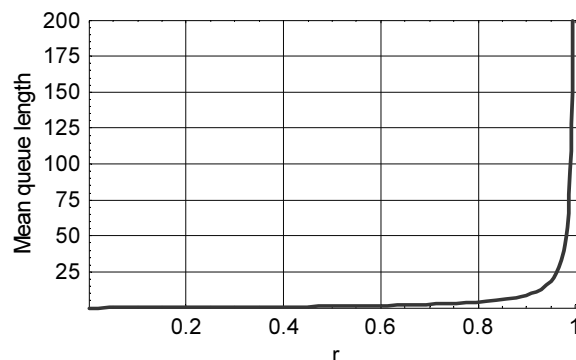
Performance of a statistical multiplexer (1)

- Internet is based on the use of statistical multiplexing
 - the output port of a router operates as a statistical multiplexer
- A statistical multiplexer can be modeled as a waiting system (= queue)
- Traffic consists of packets
 - each packet is transmitted with the full link speed C
 - packets arrive at a rate λ and let L denote the average packet length
 - packet service rate μ will be $\mu = C/L$
 - let $\rho = \lambda / \mu$, stability requirement: packet arrival rate $\lambda < \mu \Rightarrow \rho < 1$



Performance of a statistical multiplexer (2)

- Assume Poisson packet arrivals with exponentially distributed sizes
 - M/M/1 queuing system
- Load vs. mean queue length
 - mean queue length (and delay) rises sharply as load approaches 1
- Reasonable to design the network s.t. load < 0.9
 - link utilization always $< 100\%$
 - congestion control needed



- The results are qualitatively the same regardless of the assumptions of the traffic

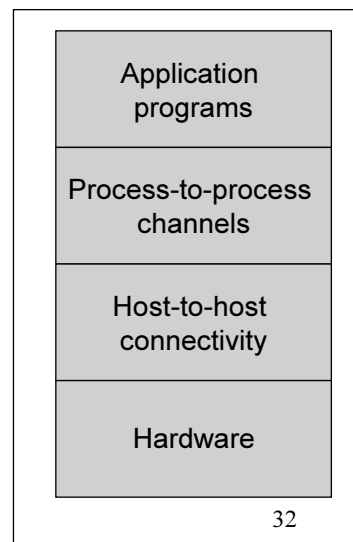
Outline

- Achieving connectivity
- Methods for resource sharing
- Enabling application level communication
- Performance issues
- **Network architecture**
- Historical background

31

Layered architectures

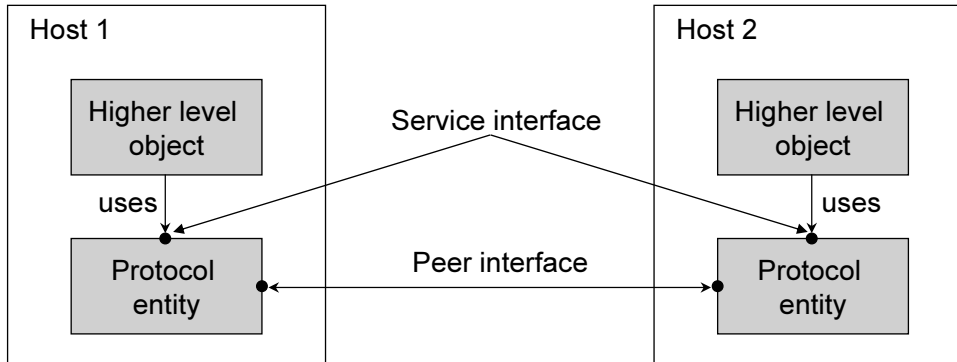
- A computer network must provide for a large number of hosts
 - cost effective, fair, robust and high performance connectivity, and
 - it must be easily able to accommodate new network technologies
- Network architecture
 - to guide the design and implementation of networks
 - abstractions used to hide complexities
- In networks, abstractions lead to layered designs
 - services offered at higher layers are implemented in terms of services provided by lower layers
 - often multiple abstractions (services) are provided to serve the varying requirements of above layers (multiplexing of upper layer protocols)
- Benefits of layering
 - decomposes the implementation problem into manageable components
 - modular design (adding new functionality may only affect one layer)



32

Protocols

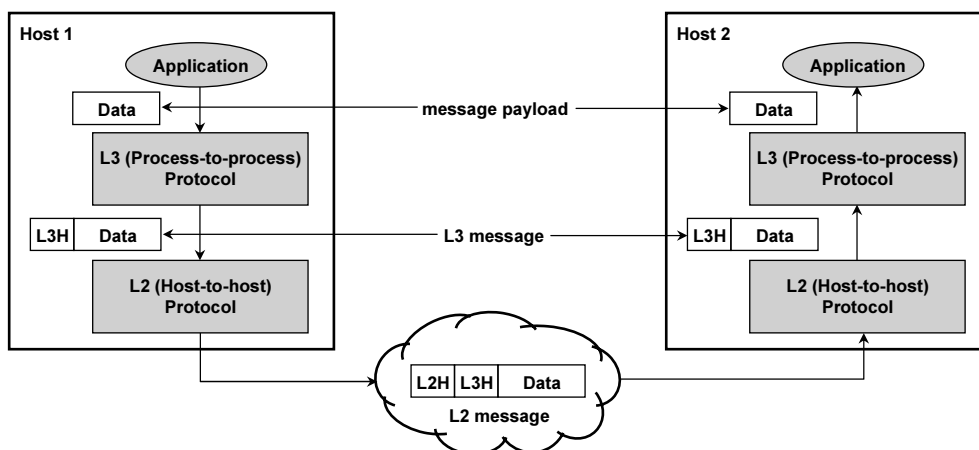
- Each layer implemented by a **protocol**
 - protocols offer communication services to higher level objects
- A protocol offers two interfaces:
 - **Service interface**: offered to higher level objects on the same host
 - **Peer interface**: offered to peer protocol objects existing on other hosts



33

Encapsulation

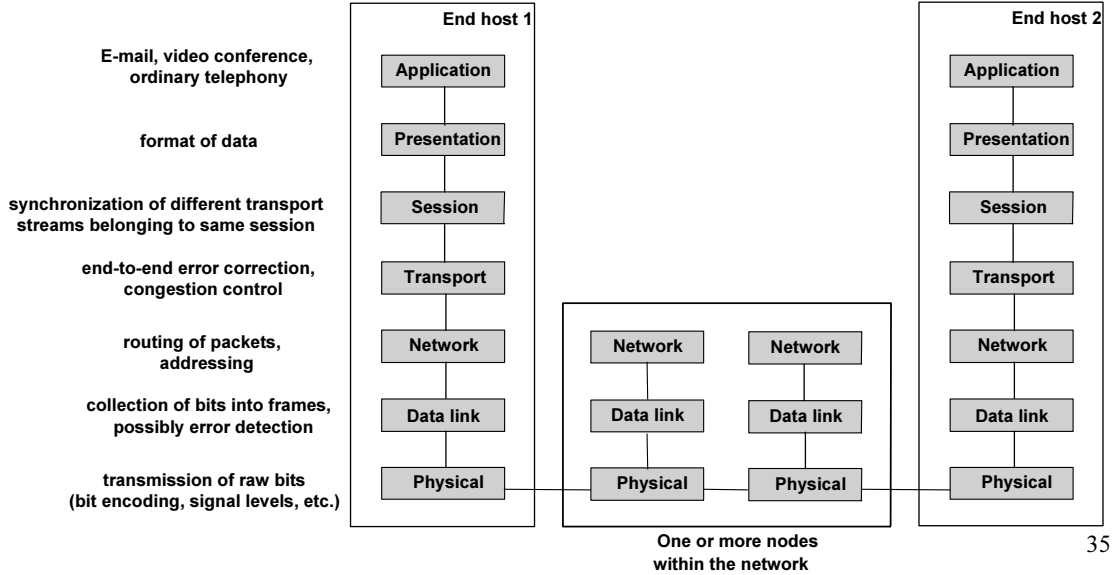
- At the sender side, each lower layer protocol adds a header (L3H, L2H) thus encapsulating the upper layer packet
 - simple transformations (compression, encryption) of the packet possible
- At the receiver side, each layer removes the corresponding header and forwards the packet to the higher layer protocol entity



34

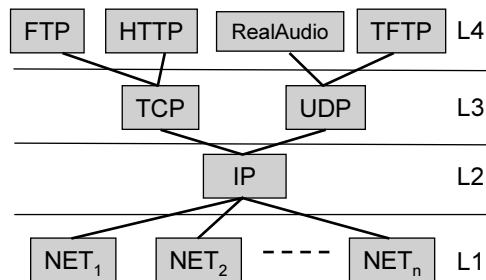
OSI (Open Systems Interconnect) architecture

- The “classic” 7-layer reference model (late 70’s)
 - protocols following the model defined in conjunction with ISO and ITU-T



Internet architecture

- Internet architecture has only 4 layers
 - L4: range of application protocols (FTP, ...)
 - L3: TCP (reliable byte transfer) and UDP (unreliable datagram delivery) provide logical channels to applications
 - L2: IP protocol interconnects multiple networks into a single logical network
 - L1: wide variety of network protocols
 - “hour glass” shape



Outline

- Achieving connectivity
- Methods for resource sharing
- Enabling application level communication
- Performance issues
- Network architecture
- Historical background

37

Brief Internet history (1)

¹Jim Kurose, lecture notes for the course MPSCI 591E Computer Networking, <http://www-net.cs.umass.edu/cs591/>

1961-1972: Early packet-switching principles

- 1961: Kleinrock - queuing theory shows effectiveness of packet-switching
- 1964: Baran - packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency
- 1969: first ARPAnet node operational
- 1972:
 - ARPAnet demonstrated publicly
 - NCP (Network Control Protocol) first host-host protocol
 - first e-mail program
 - ARPAnet has 15 nodes

38

Brief Internet history (2)

¹Jim Kurose, lecture notes for the course MPSCI 591E Computer Networking, <http://www-net.cs.umass.edu/cs591/>

1972-1980: Internetworking, new and proprietary nets

- 1970: ALOHAnet satellite network in Hawaii
- 1973: Metcalfe's Ph.D. thesis proposes Ethernet
- 1974: Cerf and Kahn - architecture for interconnecting networks
 - still determine largely the development of today's Internet
- late70's: proprietary architectures: DECnet, SNA, XNA
- late 70's: switching fixed length packets (ATM precursor)
- 1979: ARPAnet has 200 nodes

39

Brief Internet history (3)

¹Jim Kurose, lecture notes for the course MPSCI 591E Computer Networking, <http://www-net.cs.umass.edu/cs591/>

1980-1990: new protocols, a proliferation of networks

- 1983: deployment of TCP/IP
- 1982: smtp e-mail protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: ftp protocol defined
- 1988: TCP congestion control
- new national networks: Csnnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

40

Brief Internet history (4)

¹Jim Kurose, lecture notes for the course MPSCI 591E Computer Networking, <http://www-net.cs.umass.edu/cs591/>

1990's: commercialization, the WWW

- Early 1990's: ARPAnet decommissioned
 - 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
 - early 1990s: WWW
 - hypertext [Bush 1945, Nelson 1960's]
 - HTML, http: Berners-Lee
 - 1994: Mosaic, later Netscape
 - late 1990's: commercialization of the WWW
- Late 1990's:
 - estimated 50 million computers on Internet
 - estimated 100 million+ users
 - backbone links running at 1 Gbps