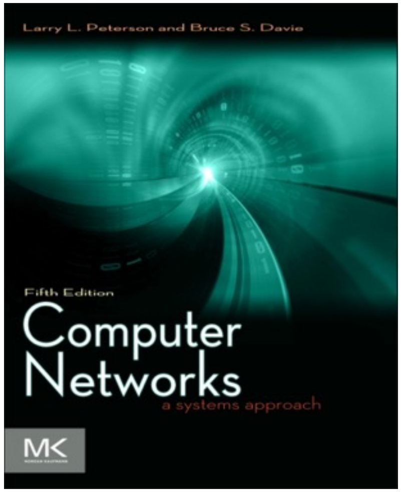
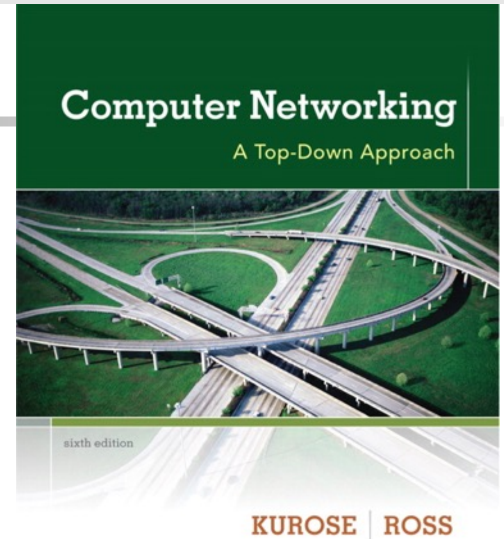
Slides are an edited mashup of two books





Computer Networks:
A Systems Approach, 5e
Larry L. Peterson and Bruce S. Davie

Computer Networking: A Top Down Approach
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

Why is Internet Architecture relevant to you

- The Internet is our economies most critical infrastructure.
- The Internet is our economies most enabling metaphor.
- Objectives for student learning:
  - What are the technical features that allowed the Internet to become so pervasive and diverse during the course of your lifetime.
  - Understand its basic structure, and distill from it lessons for (a) the use and governance of Internet technology in modern organizations, and
    (b) lessons for creation of other new technologies and products.

Internet history

1972-1980: Internetworking, new and proprietary nets

- 1970: ALOHAnet satellite network in Hawaii
- 1974: Cerf and Kahn - architecture for interconnecting networks
- 1976: Ethernet at Xerox PARC
- late70's: proprietary architectures: DECnet, SNA, XNA
- late 70's: switching fixed length packets (ATM precursor)
- 1979: ARPAnet has 200 nodes

Cerf and Kahn's internetworking principles:

- minimalism, autonomy - no internal changes required to interconnect networks
- best effort service model
- stateless routers
- decentralized control

define today's Internet architecture

Internet history

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: Csnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

Internet history

1990, 2000's: commercialization, the Web, new apps

- early 1990's: ARPAnet decommissioned
- 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- early 1990s: Web
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990's: commercialization of the Web

late 1990's – 2000's:

- more killer apps: instant messaging, P2P file sharing
- network security to forefront
- est. 50 million host, 100 million+ users
- backbone links running at Gbps

Internet history

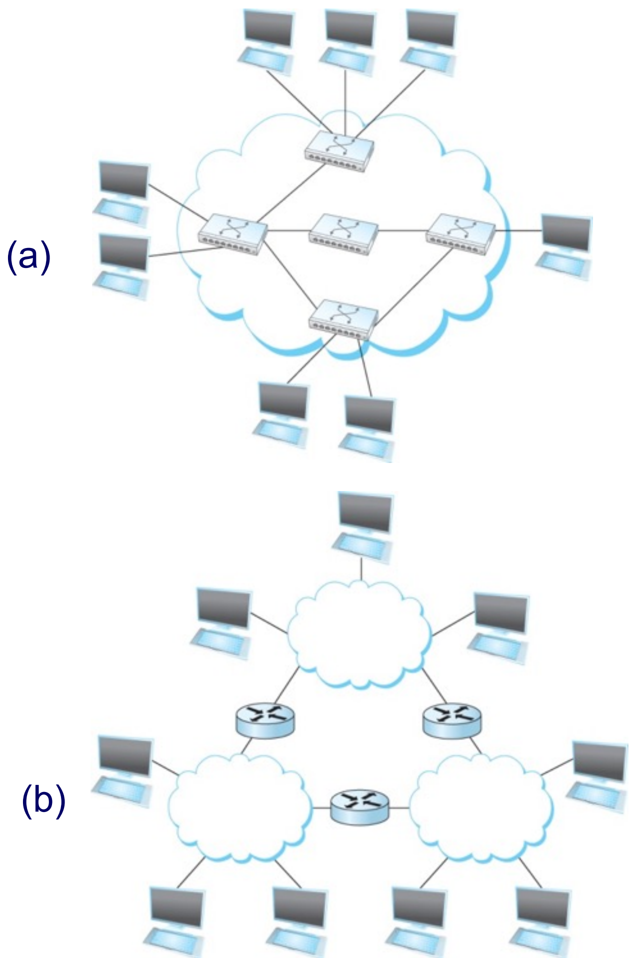
2005-present

- ~750 million hosts
  - Smartphones and tablets
- Aggressive deployment of broadband access
- Increasing ubiquity of high-speed wireless access
- Emergence of online social networks:
  - Facebook, Instagram, …
- Service providers (Google, Amazon) create their own networks
  - Bypass Internet, providing "instantaneous" access to search, email, etc.
- E-commerce, universities, enterprises running their services in "cloud" (eg, Amazon EC2)

What makes the Internet appear as single service

- Networks share common architecture and protocols that enable communication within and among them.
  - Architecture: how components of the networks interrelate
  - Protocols: standards governing the interchange of data
- The architecture and protocols were and to some extent are shaped by fundamental (and interrelated) design principles adopted by early builders of the Internet.
  - Hourglass
  - End-to-End
  - Distributed design and decentralized control
  - Heterogeneity, Scalability

Connectivity Terminologies

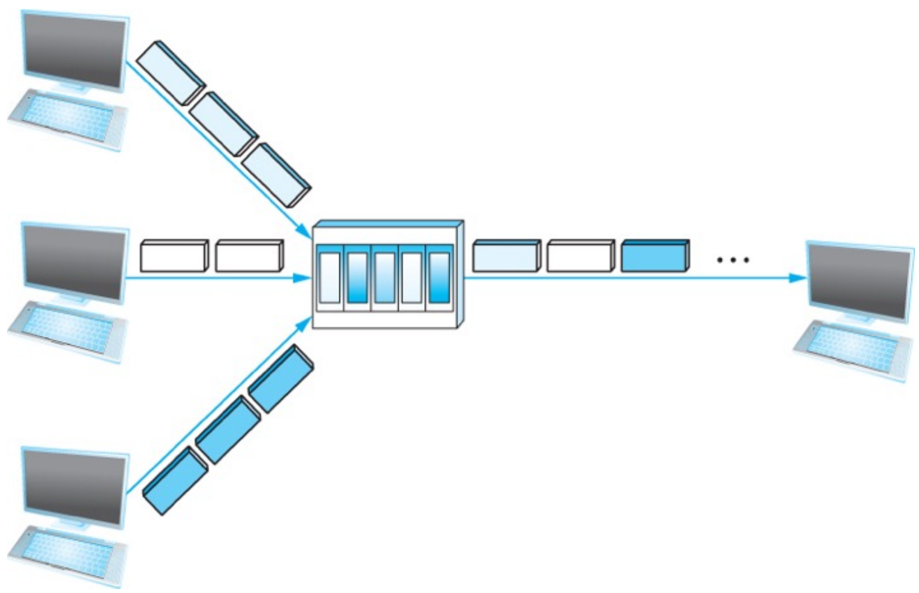


(a) A switched network

(b) Interconnection of networks

- Link, Nodes
- Point-to-point, Multiple access
- Switched Network
  - Circuit Switched
  - Packet Switched: Store-and-forward
- Cloud
- Hosts
- Switches
- internetwork
- Router/gateway
- Host-to-host connectivity
- Address
- Routing
- Unicast/broadcast/multicast

Cost-Effective Resource Sharing

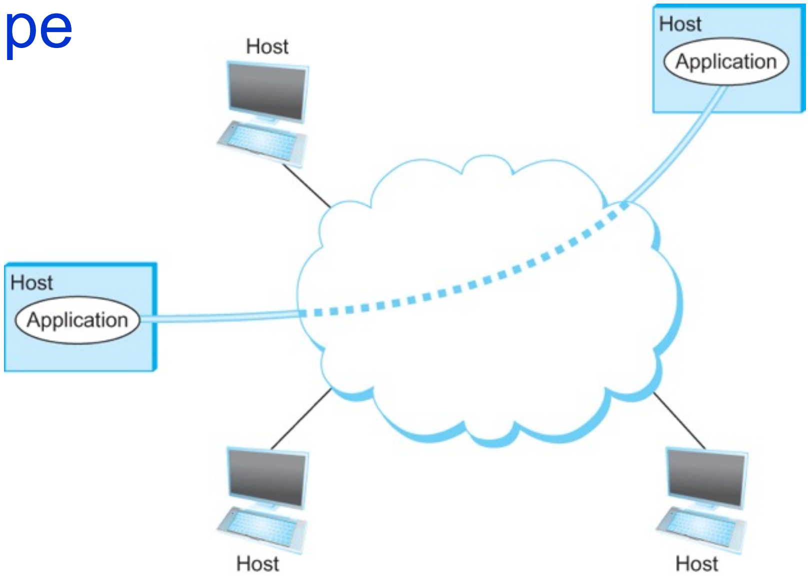


A switch multiplexing packets from multiple sources onto one shared link

- FDM: Frequency Division Multiplexing
- Statistical Multiplexing
  - Data is transmitted based on demand of each flow.
  - What is a flow?
  - Packets vs. Messages
  - FIFO, Round-Robin, Priorities (Quality-of-Service (QoS))
  - Congested?
- LAN, MAN, WAN
- SAN (System Area Networks

Support for Common Services

- Logical Channels
  - Application-to-Application communication path or a pipe



Process communicating over an abstract channel

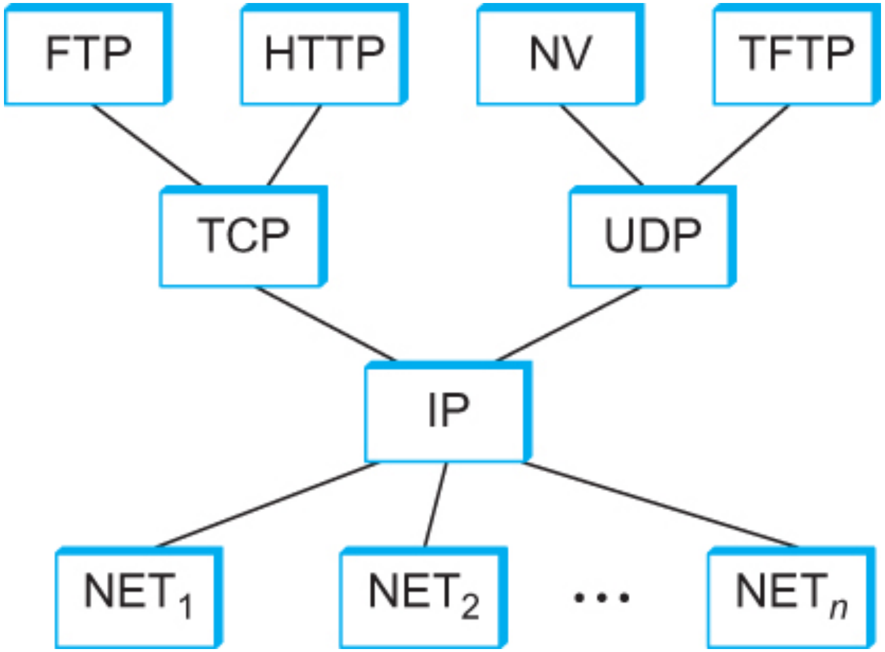
Reliability challenge

- Network software should hide (inevitable) errors from applications
- Bits are lost
  - Bit errors (1 to a 0, and vice versa)
  - Burst errors – several consecutive errors
- Packets are dropped (largely Congestion)
  - Links and Node failures
  - Messages are delayed
- Messages are delivered out-of-order
- Third parties eavesdrop

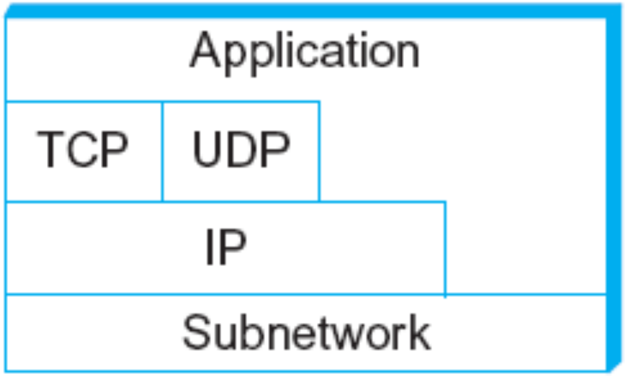
Protocols

- Protocol defines the interfaces between the layers in the same system and with the layers of peer system
- 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

Internet Architecture



Internet Protocol Graph



Alternative view of the Internet architecture. The "Network" layer shown here is sometimes referred to as the "sub-network" or "link" layer.

Description of (Lower) Layers

- Physical Layer
  - Handles the transmission of raw bits over a communication link
- Data Link Layer
  - Collects a stream of bits into a larger aggregate called a *frame*
  - Network adaptor along with device driver in OS implement the protocol in this layer
  - Frames are actually delivered to hosts
- Network Layer
  - Handles routing among nodes within a packet-switched network
  - Unit of data exchanged between nodes in this layer is called a *packet*

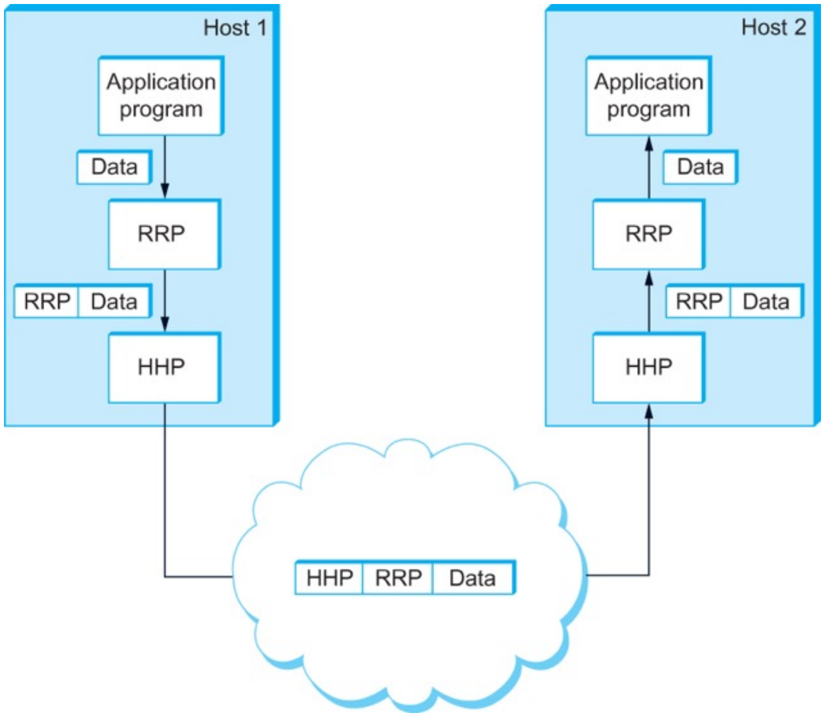
The lower three layers are implemented on all network nodes

Description of (Higher) Layers

- Transport Layer
  - Implements a process-to-process channel
  - Unit of data exchanges in this layer is called a *message*
- Session Layer
  - Provides a name space that is used to tie together the potentially different transport streams that are part of a single application
- Presentation Layer
  - Concerned about the format of data exchanged between peers
- Application Layer
  - Standardize common type of exchanges

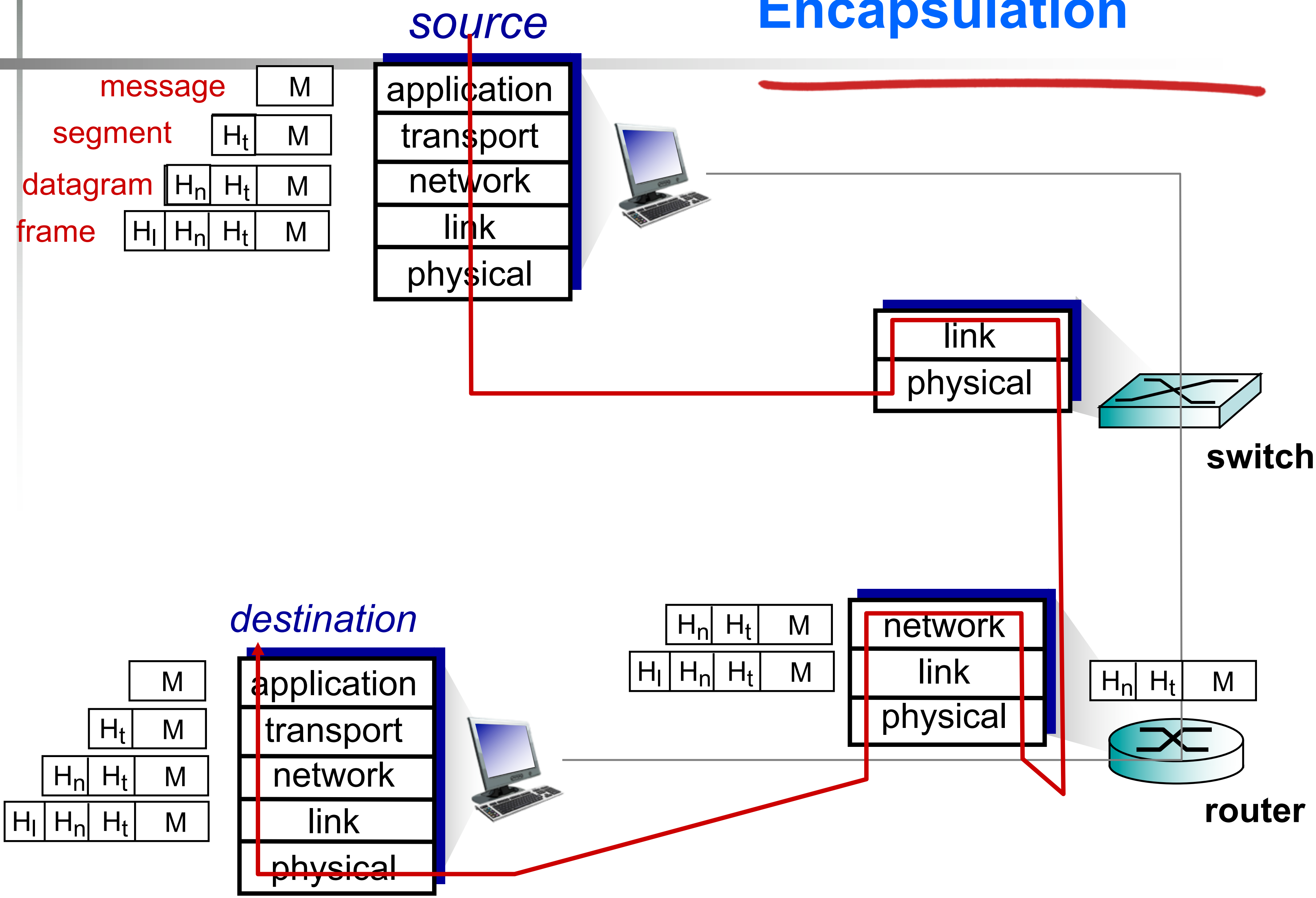
The transport layer and the higher layers typically run only on end-hosts and not on the intermediate switches and routers

Encapsulation



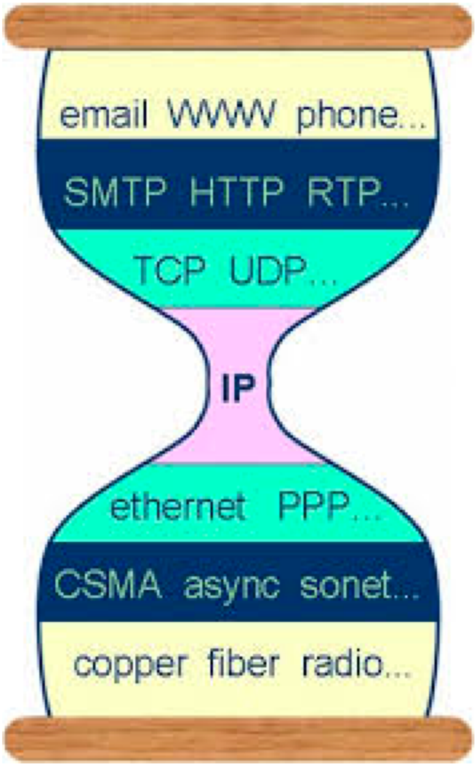
High-level messages are encapsulated inside of low-level messages

Encapsulation



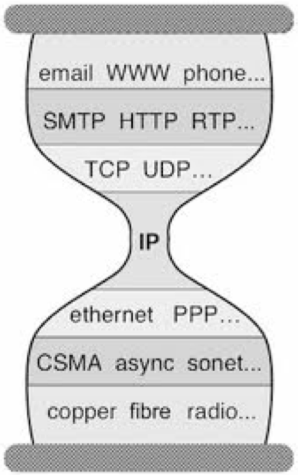
Internet Architecture

- Does not imply strict layering. The application is free to bypass the defined transport layers and to directly use IP or other underlying networks
- An hour-glass shape – wide at the top, narrow in the middle and wide at the bottom. IP serves as the focal point for the architecture
- In order for a new protocol to be officially included in the architecture, there needs to be both a protocol specification and at least one (and preferably two) representative implementations of the specification
- IETF Governance
  - "rough consensus and running code"



Benefits of Hourglass architecture

- Internet designed to operate over different underlying communications technologies, including those yet to be introduced, and to support multiple and evolving applications and services.
- Does not impede or restrict particular applications (although users, ISPs may make optimizations)
- Enables developers to write applications without knowing/adapting to details of underlying networks
- Enables users to adopt applications without involvement/approval from network operators
- Critical separation between network technology and higher-level services through which users actually interact with the Internet visualized as hourglass
- IP as minimal viable agreement/min common denominator maximizes flexibility



Why a narrow waist is important metaphor for new systems, products …

- Tim O'Reilly:
  - do as little as possible….the less you include the easier it will be to agree and you dont tie yourself down…because we dont know what will come [sic: in this case less is more]
  - Build a system and let it evolve
  - Create architecture for participation—iTUNES, App Store…[sic: it started with Internet, Includes maps mashups and APIs!!]
  - TBL didnt have to ask anyones permission to put up WWW on the net…they would have said no…'http is poorly designed protocol..will never scale'
  - Tolerate as much failure and participation as needed to introduce new systems/innovations rapidly/iteratively and innovate
- Naughton:
  - Allow innovation to be tried for free

End-to-end architecture

- Edge-based innovation derives from early design decision that the Internet should have an end-to- end architecture:
  - The network provides communications fabric connecting the many computers at its ends
  - Network offers very basic level of service, data transport
  - Beyond transporting data—locate special features needed to support specific applications in or close to applications/devices at network edge.
  - Only put feature lower down if performance improvement justifies it
- E2E design facilitates
  - designing for: failure, change, dynamics, decentralized control, rolling asynchronous adoption, of components
  - scalability and therefore longevity of architecture
- QUESTION – Reliable transport of data – Packet level hop by hop, Packet level end to end (Process), Message/File level end to end (Application).

Scalability

- Internet's design enabled it to support a growing amount of communications:
- Growth in number of users and attached devices
- Growth in volume of communications per device and total

- Scale implies… heterogeneity…designing for Heterogeneity is a good step in future proofing

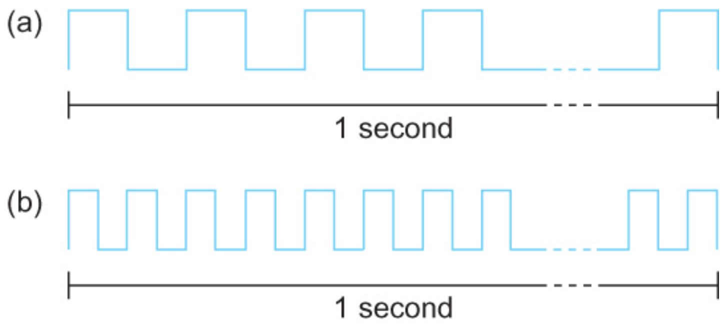
Performance

- Latency = Propagation + transmit + queue
- Propagation = distance/speed of light
- Transmit = size/bandwidth

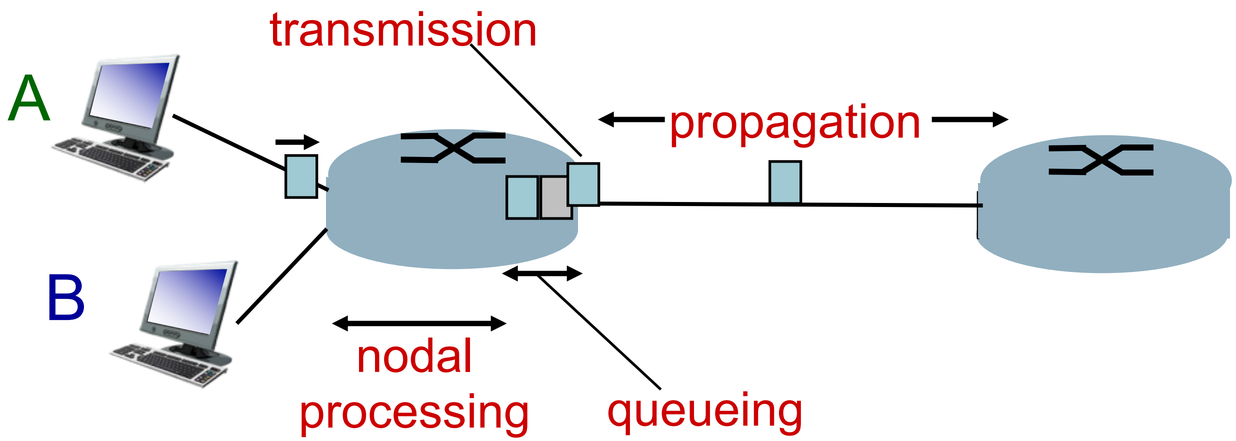
- One bit transmission => propagation is important
- Large bytes transmission => bandwidth is important

Bandwidth

- Width of the frequency band
  - Number of bits per second that can be transmitted over a communication link
- 1 Mbps: 1 x 10^6 bits/second = $1 \text{x} 2^{20}$ bits/sec
- 1 x 10^{-6} seconds to transmit each bit or imagine that a timeline, now each bit occupies 1 micro second space.
- On a 2 Mbps link the width is 0.5 micro second.
- Smaller the width more will be transmission per unit time.



Four sources of packet delay



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

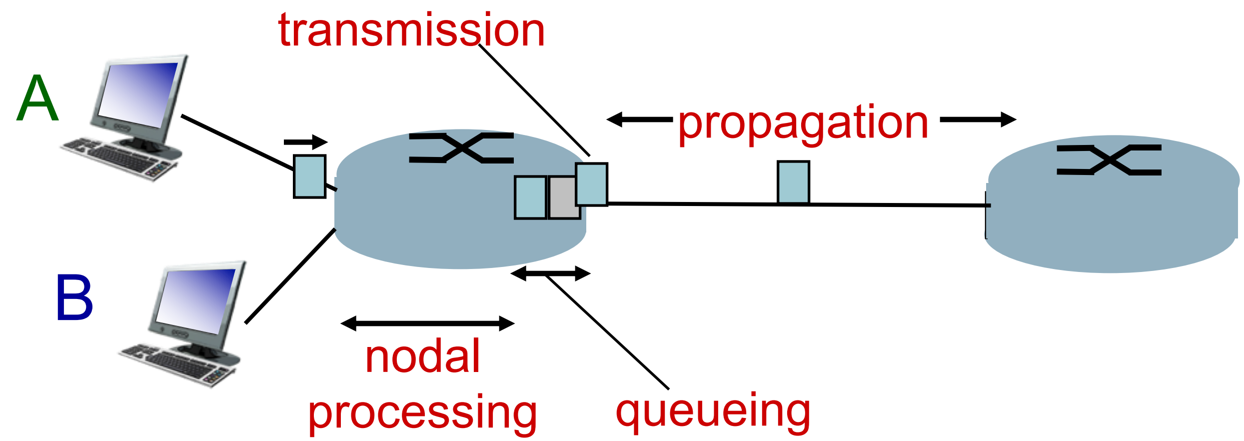
(1) d_{proc}: nodal processing

- check bit errors
- determine output link
- typically < msec

(2) d_{queue}: queueing delay

- time waiting at output link for transmission
- depends on congestion level of router

Four sources of packet delay



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

(3) d_{trans}: transmission delay:

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

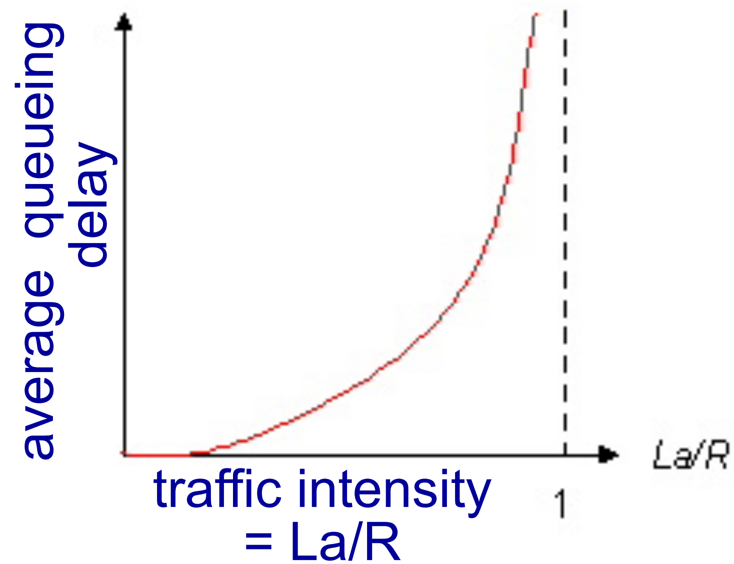
(4) d_{prop}: propagation delay:

- *d*: length of physical link
- *s*: propagation speed in medium (~2x10^8 m/sec)
- $d_{prop} = d/s$

d_{trans} and d_{prop} *very* different

Queueing delay -- Congestion

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



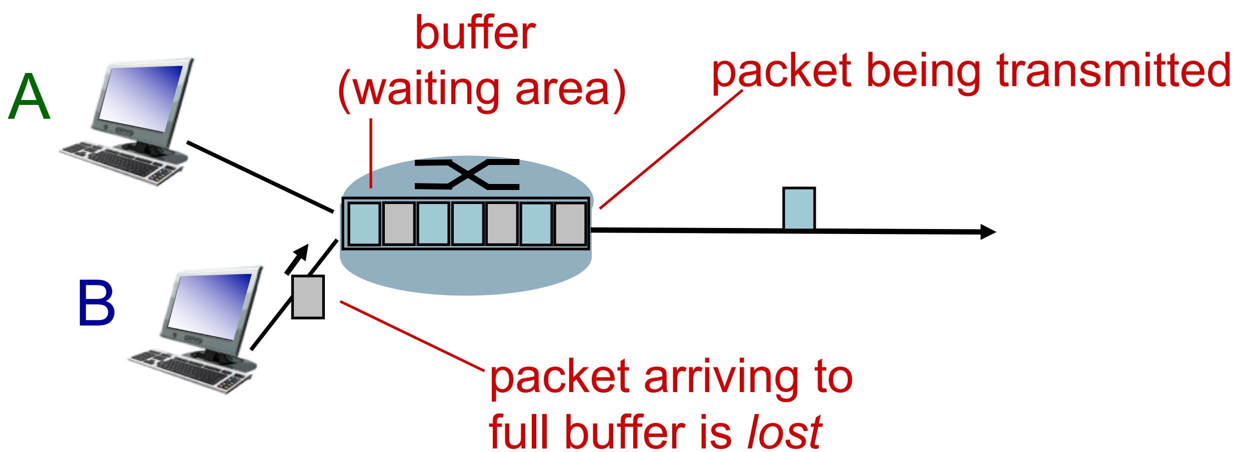
- *La/R* ~ 0: avg. queueing delay small
- *La/R* -> 1: avg. queueing delay large
- *La/R* > 1: more “work” arriving than can be serviced, average delay infinite!

La/R ~ 0

La/R -> 1

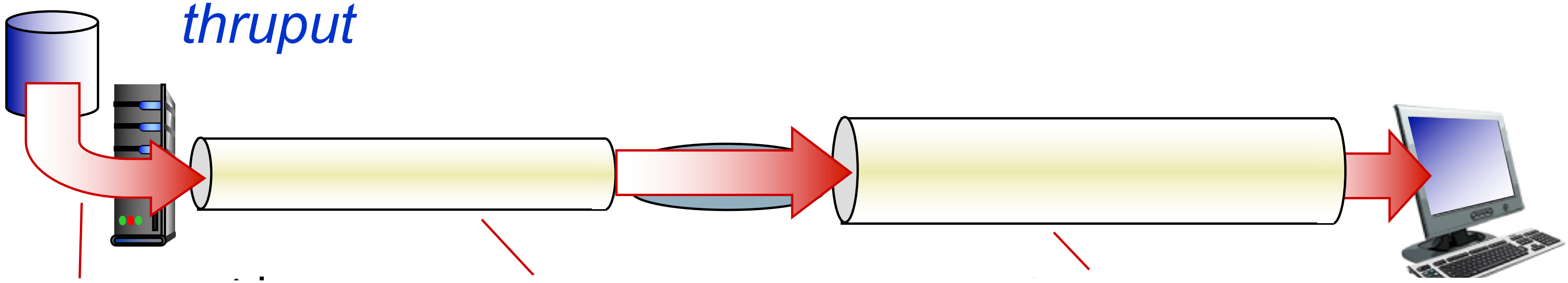
Packet loss

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

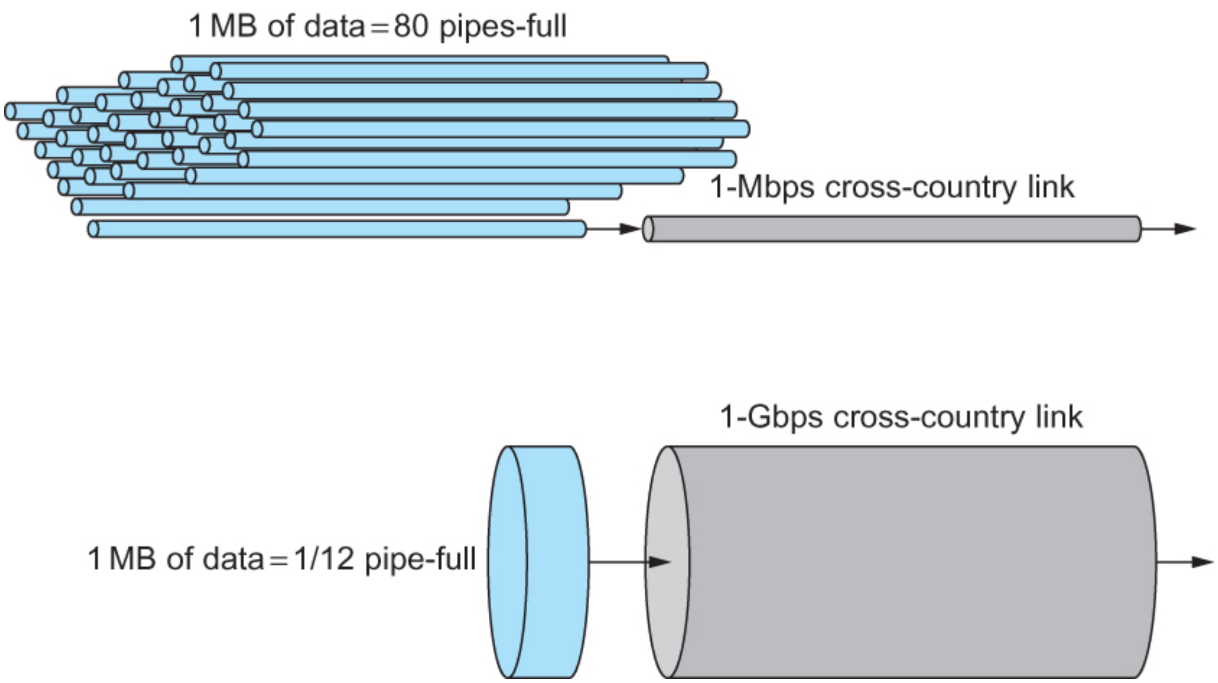


Throughput

- *throughput:* rate (bits/time unit) at which bits transferred between sender/receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time
  - *bottleneck link on end to end path constrains thruput*



Relationship between bandwidth and latency



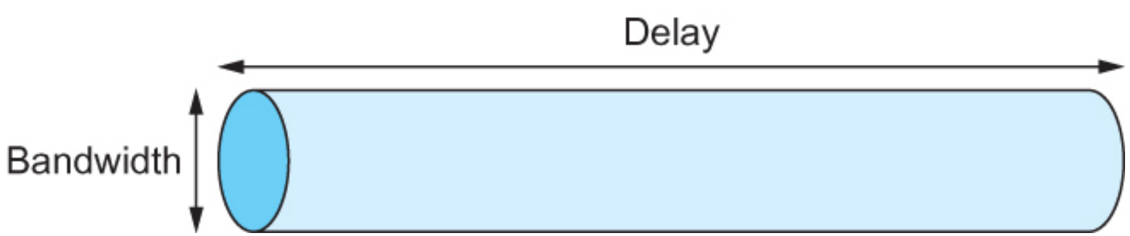
A 1-MB file would fill the 1-Mbps link 80 times, but only fill the 1-Gbps link 1/12 of one time

Delay X Bandwidth

- We think the channel between a pair of processes as a hollow pipe
- Latency (delay) length of the pipe and bandwidth the width of the pipe
- Delay of 50 ms and bandwidth of 45 Mbps

⇒ 50×10^{-3} seconds x 45×10^{6} bits/second

⇒ 2.25×10^{6} bits = 280 KB data.



Network as a pipe

Delay X Bandwidth

- Relative importance of bandwidth and latency depends on application
  - For large file transfer, bandwidth is critical
  - For small messages (HTTP, NFS, etc.), latency is critical
  - Variance in latency (jitter) can also affect some applications (*e.g.*, audio/video conferencing)
- How many bits the sender must/could transmit before the first bit arrives at the receiver
  - Takes another one-way latency to receive a response from the receiver
  - If the sender does not fill the pipe—send a whole delay × bandwidth product's worth of data before it stops to wait for a signal—the sender will not fully utilize the network
  - Control travels over same network as data – latency impairs feedback which impairs thruput