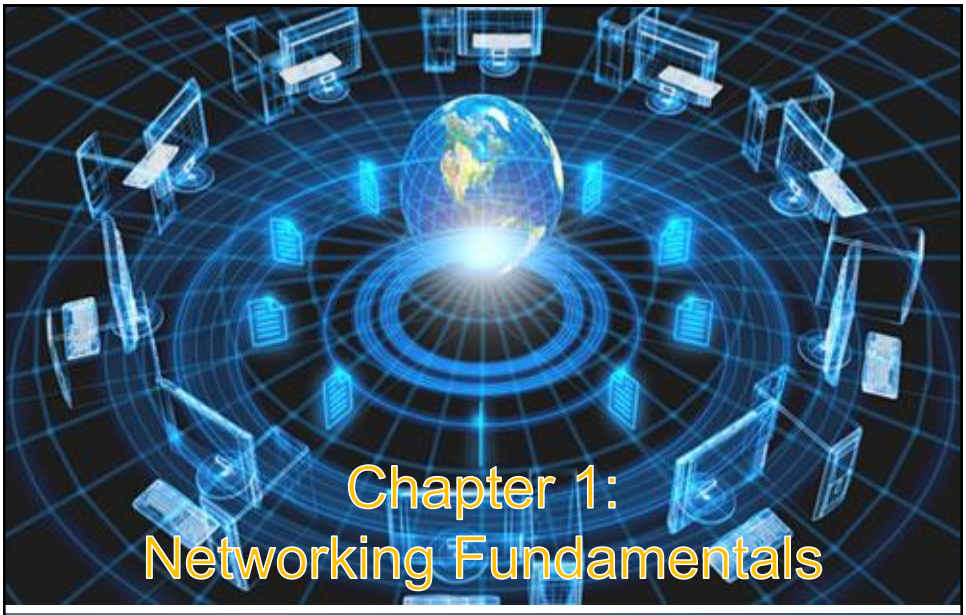


Chapter 1: Networking Fundamentals

Lecturer: Nguyễn Thị Thanh Vân – FIT - HCMUTE

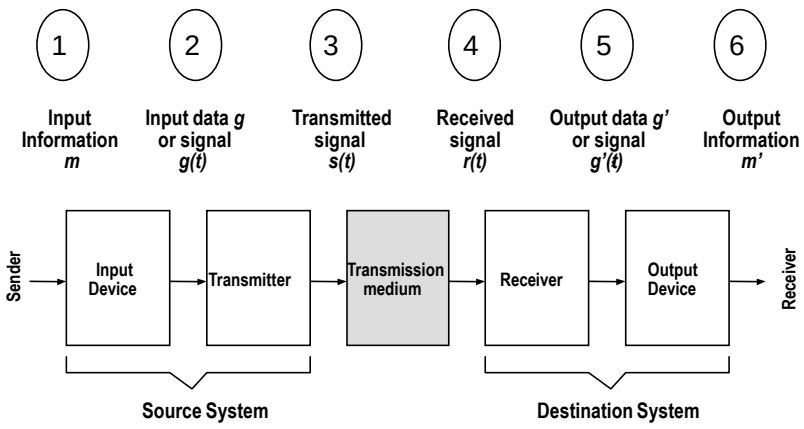
Contents

- History of computer network
- Computer network
- Network topology
- Network protocol
- Network Components
- Internet
- Packet-Switched Networks problems:
  - Delay, Loss, and Throughput in
- Protocol Layers and Their Service Models
- OSI model
- TCP/IP model

History of computer network

- 1960's – "How can we transmit bits across a communication medium efficiently and reliably?"
- 1970's – "How can we transmit packets across a communication medium efficiently and reliably?"
- 1980's – "How can we provide communication services across a series of interconnected networks?
- 1990's – "How can we provide high-speed, broadband communication services to support high-performance computing and multimedia applications across the globe?"
- 2000's – What do you think will dominate in the next 10 years?

A Communication Model



Common Communication Tasks

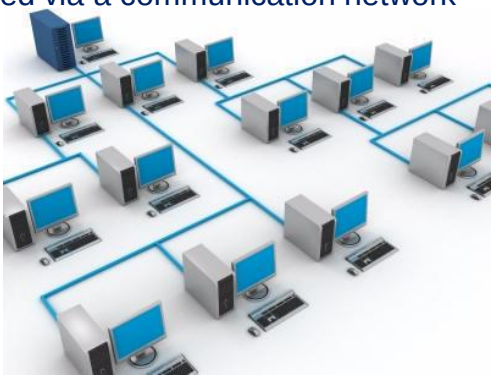
- **Data encoding**
- **Signal generation:** electro-magnetic signals to be transmitted over a transmission medium
- **Synchronization:** timing of signals between the transmitter and receiver
- **Error detection and correction:** ensuring that transmission errors are detected and corrected
- **Flow control:** ensuring that the source does not overwhelm the destination by sending data faster than the receiver can handle
- **Multiplexing:** make more efficient use of a transmission facility. This technique is used at different levels of communication
- **Addressing:** indicating the identity of the intended destination
- **Routing:** selecting appropriate paths for data being transmitted
- **Message formatting:** conforming to the appropriate format
- **Security:** ensuring secure message transmission
- **Systems management**

Communication network

- A communication network is a collection of devices connected by some communications media and Network Architecture (topology and protocol)
  - Example devices are:
    - mainframes, minicomputers, supercomputers
    - workstations, personal computers
    - printers, disk servers, robots
    - X-terminals
    - Gateways, switches, routers, bridges
    - Cellular phone, Pager, TRS
    - Refrigerator, Television, Video Tape Recorder
  - Communications Media
    - twisted pairs, coaxial cables, fiber optics
    - line-of-sight transmission: lasers, infra-red, microwave, radio
    - satellite links
    - Power line

Computer network

- Computer Communication – the exchange of information between computers for the purpose of cooperative action
- Computer Network – a collection of computers interconnected via a communication network



Transportation vs. Computer Networks

Transportation Network	Computer Network
Vehicles/People	Packets/Payload
Street address	IP address
Intersection	Bridge/router
Street, highway, path	Link/broadband/path
Traffic jam	Network congestion
Stop and go traffic light	Flow control
Taking alternative path	Alternative route
Collision	Collision of packets
HOV lane	Flow Priority
Following a route to school	Routing algorithm
…	…

Applications of Networks

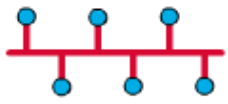
- **Resource Sharing**
  - Hardware (computing resources, disks, printers)
  - Software (application software)
- **Information Sharing**
  - Easy accessibility from anywhere (files, databases)
  - Search Capability (WWW)
- **Communication**
  - Email
  - Message broadcast
- **Remote computing**
- **Distributed processing (GRID Computing)**

Contents

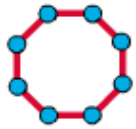
- History of computer network
- Computer network
- Network topology
- Network protocol
- Network Components
- Internet
- Packet-Switched Networks problems:
  - Delay, Loss, and Throughput in
- Protocol Layers and Their Service Models
- OSI model
- TCP/IP model

Network topology

- The network topology defines the way in which devices are connected.

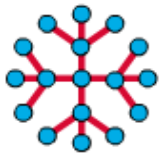


Bus Topology



Ring Topology

Star Topology



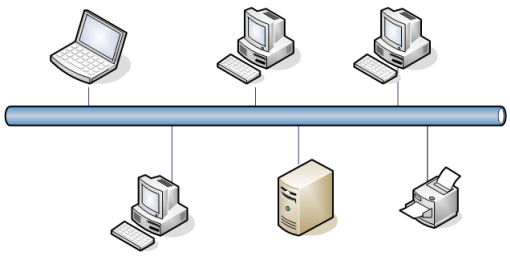
Extended Star Topology

Mesh Topology

Bus topology

- All networked nodes are interconnected, peer to peer, using a single, open-ended cable
- Both ends of the bus must be terminated with a terminating resistor to prevent signal bounce

BUS Topology

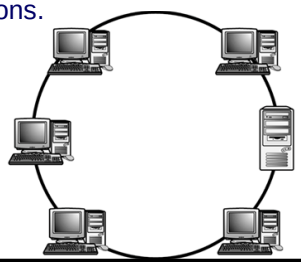


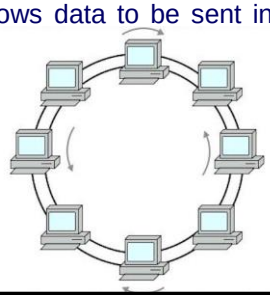
Bus topology

- Advantage:
  - Easy to implement and extend
  - Well suited for temporary networks that must be set up in a hurry
  - Typically the least cheapest topology to implement
  - Failure of one station does not affect others
- Disadvantage
  - Difficult to administer/troubleshoot
  - Limited cable length and number of stations
  - A cable break can disable the entire network; no redundancy
  - Maintenance costs may be higher in the long run
  - Performance degrades as additional computers are added

Ring Topology

- A frame travels around the ring, stopping at each node. If a node wants to transmit data, it adds the data as well as the destination address to the frame.
- The frame then continues around the ring until it finds the destination node, which takes the data out of the frame.
  - Single ring – All the devices on the network share a single cable
  - Dual ring – The dual ring topology allows data to be sent in both directions.



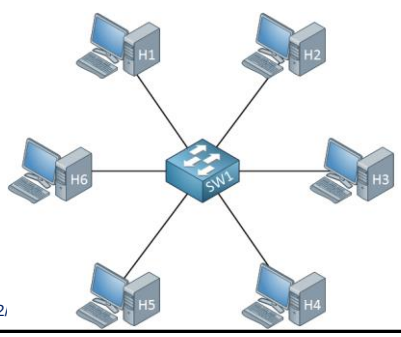


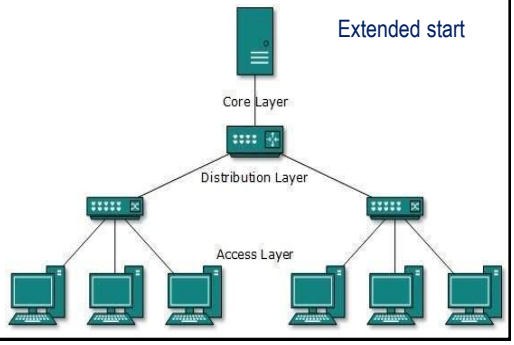
Ring Topology

- Advantage
  - This type of network topology is very organized
  - Performance is better than that of Bus topology
  - No need for network server to control the connectivity between workstations
  - Additional components do not affect the performance of network
  - Each computer has equal access to resources
- Disadvantage:
  - Each packet of data must pass through all the computers between source and destination, slower than star topology
  - If one workstation or port goes down, the entire network gets affected
  - Network is highly dependent on the wire which connects different components

Start topology

- Have connections to networked devices that “radiate” out form a common point
- Each device can access the media independently
- Have become the dominant topology type in contemporary LANs (replace buses and rings)



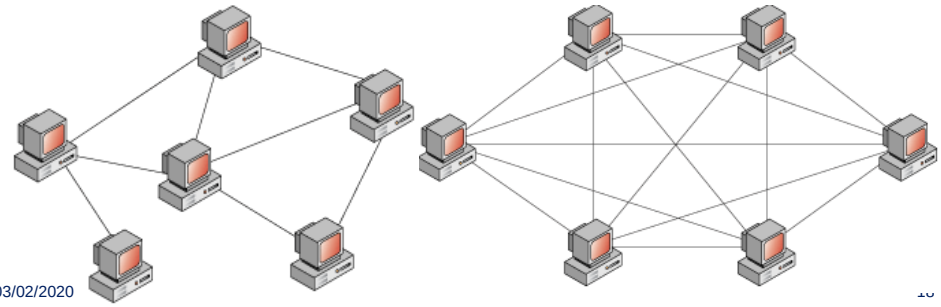


Start topology

- Advantage:
  - Compare to bus: gives far much better performance
  - Easy to connect new nodes or devices
  - Centralized management.
  - Failure of one node or link doesn't affect the rest of network
- Disadvantage:
  - If central device fails whole network goes down
  - The use of hub, a router or a switch as central device increases the overall cost of the network
  - Performance and as well number of nodes which can be added in such topology is depended on capacity of central device

Mesh topology

- Partial Mesh Topology :
  - In this topology some of the systems are connected in the same fashion as mesh topology but some devices are only connected to two or three devices.
- Full Mesh Topology :
  - Each and every nodes or devices are connected to each

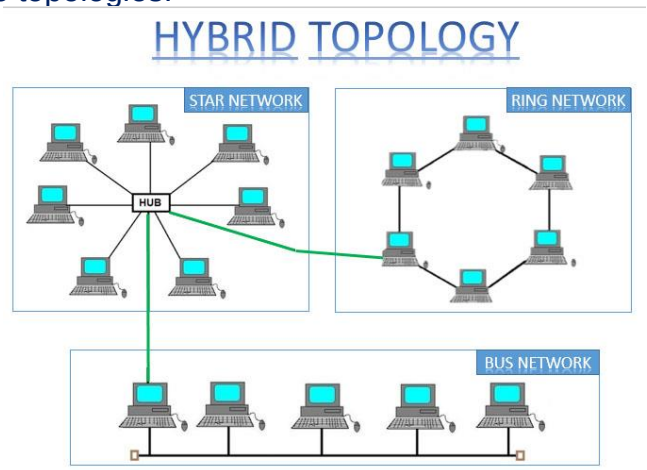


Mesh topology

- Advantages
  - Each connection can carry its own data load.
  - It is robust.
  - Fault is diagnosed easily.
  - Provides security and privacy.
- Disadvantages:
  - Installation and configuration is difficult.
  - Cabling cost is more.
  - Bulk wiring is required.

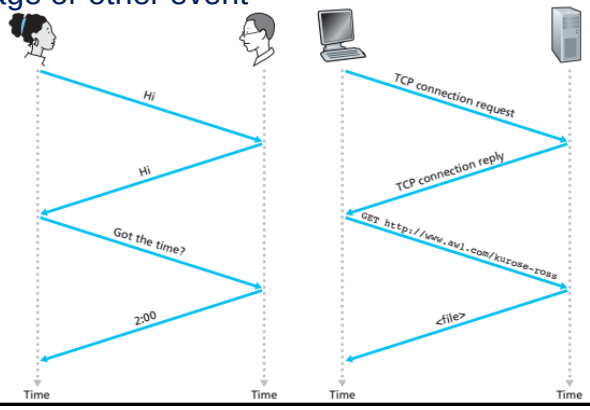
Hybrid topology

- Many different types of topologies which is a mixture of two or more topologies.



Protocol

- A **protocol** defines the format and the order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other event



Network Components

- Physical Media
  - Cable,
- Interconnecting Devices
  - Router, switch, hub…
- Computers
  - Server, client
- Networking Software
  - Protocol, Network OS
- Applications
  - Mail, web….

Network types

- Depending on the size and range of the computer network, you can differentiate between **different network dimensions**.
- The most important network types include:
  - Personal Area Networks (PAN): modern devices are integrated into a network
  - Local Area Networks (LAN): more than 1 computer is to be connected
  - Metropolitan Area Networks (MAN): connects several LAN
  - Wide Area Networks (WAN): extend MAN across large geographic areas, such as countries or continents
  - Global Area Networks (GAN): Internet

Contents

- History of computer network
- Computer network
- Network topology
- Network protocol
- Network Components
- Internet
- Packet-Switched Networks problems:
  - Delay, Loss, and Throughput in
- Protocol Layers and Their Service Models
- OSI model
- TCP/IP model

Internet

- Introduction to Internet
- The Network Edge
- The Network core
  - Switching Techniques: Circuit and Packet

How does the Internet Look Like?

Internet History (1)

1961–1972: Early packet-switching principles

- **1961:** Kleinrock – queueing 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

27

Internet History (2)

1972–1980: Internetworking, new and proprietary nets

- **1970:** ALOHAnet satellite network in Hawaii
- **1973:** Metcalfe's PhD thesis proposes Ethernet
- **1974:** Cerf and Kahn - architecture for interconnecting networks
- **late 70s:** Proprietary architectures: DECnet, SNA, XNA
- **late 70s:** 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

28

Internet History (3)

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

29

Internet History (4)

1990s: 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 1960s]
  - HTML, http: Berners-Lee
  - 1994: Mosaic, later Netscape
  - Late 1990s: commercialization of the WWW

Late 1990's:

- Est. 50 million computers on Internet
- Est. 100 million+ users
- Backbone links running at 1 Gbps

30

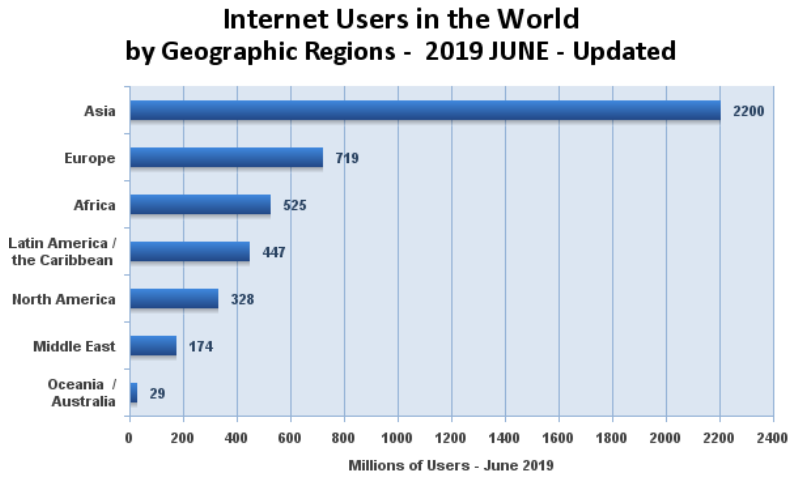
Biggest Internet Challenge

- Scale
  - How to manage such a large system,
  - growing rapidly and uncontrollably,
  - consisting of heterogeneous devices,
  - managed by multiple entities
  - having limited resources

- Let's take things one at a time

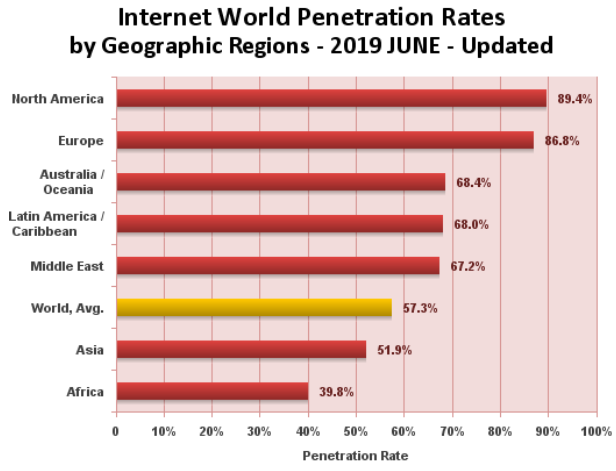
31

How Many Users?



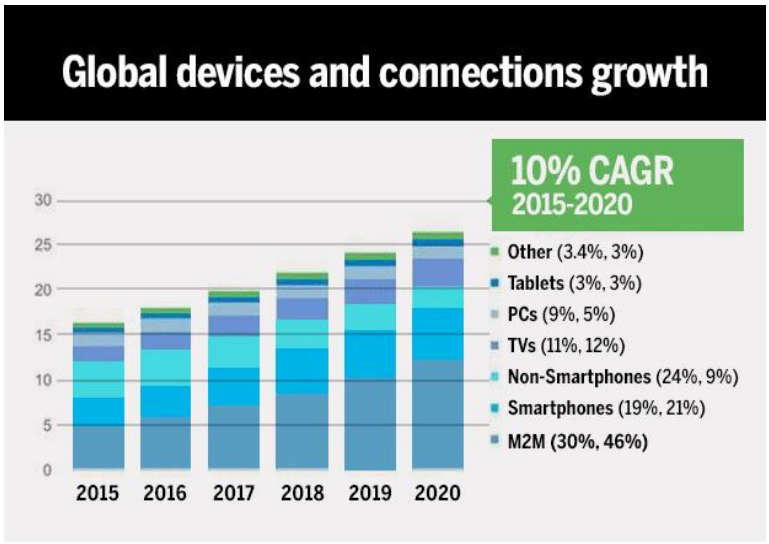
Source: Internet World Stats - www.internetworldstats.com/stats.htm
Basis: 4,422,494,622 Internet users estimated in June 30, 2019

How many more Users?



Source: Internet World Stats - www.internetworldstats.com/stats.htm
Penetration Rates are based on a world population of 7,716,223,209 and 4,422,494,622 estimated Internet users in June 30, 2019.

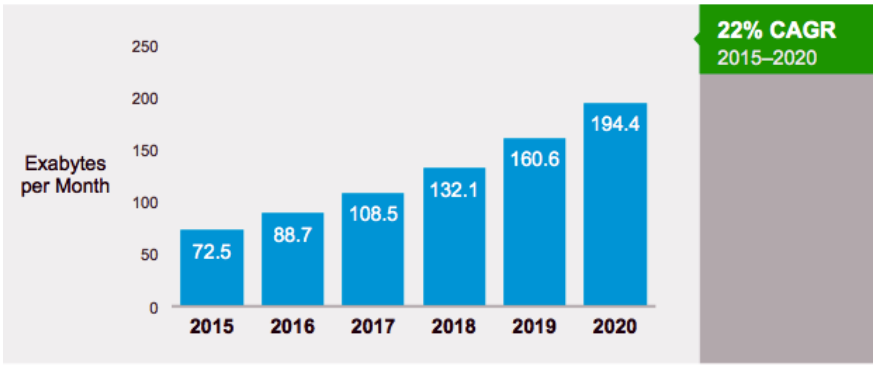
Global IP traffic



Global IP traffic

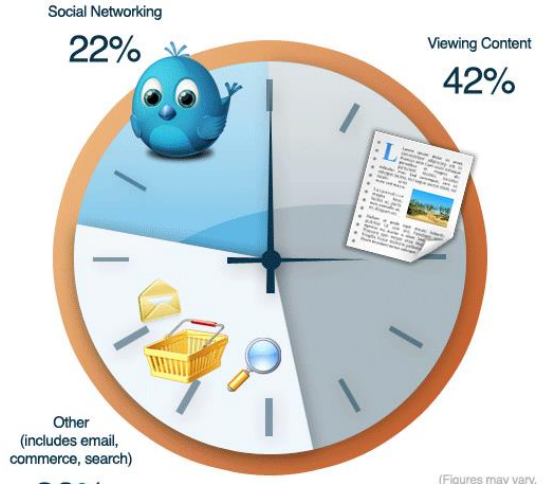
Global IP Traffic Growth / Top-Line

Global IP traffic will increase 3-fold from 2015–2020



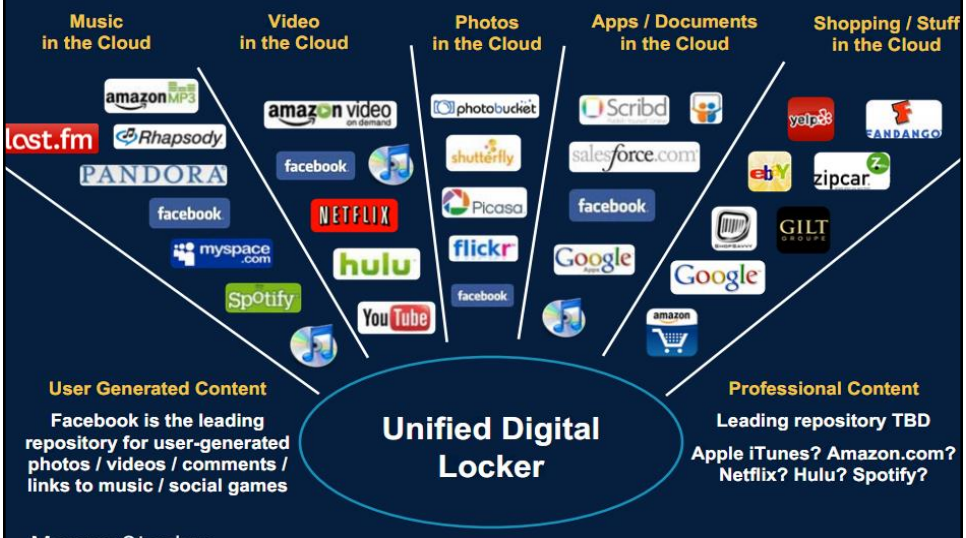
Source: Cisco VNI Global IP Traffic Forecast, 2015–2020

How is Time Spent?

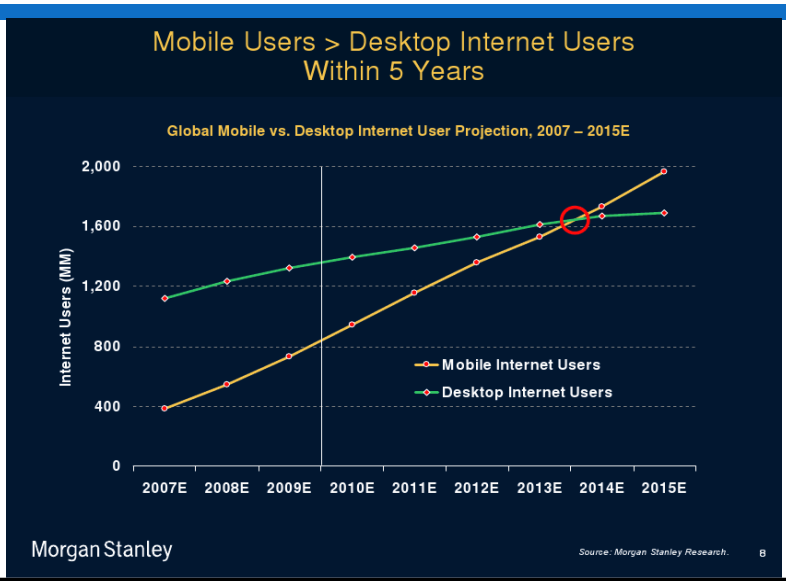


(Figures may vary, slightly as categories tend to overlap)
sources: Pew, Nielsen

What Do Users Expect?



How do they get it?



38

What's the Internet: a view

- Millions of connected devices:
  - ***Hosts*** = ***end systems***
  - Running ***network apps***

PC

Server

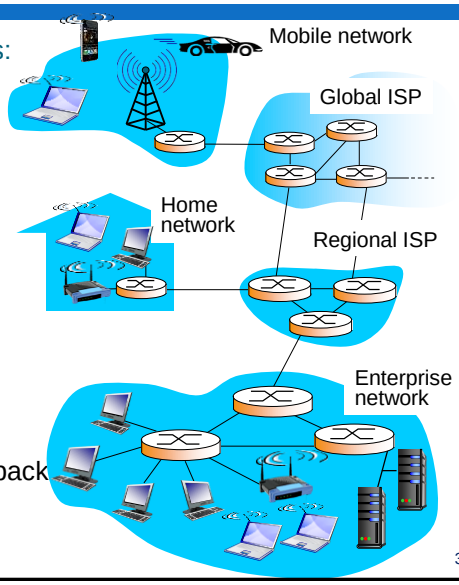
Wireless laptop

Smartphone

- ***Communication links***
  - Fiber, copper, radio, satellite
  - Transmission rate: ***bandwidth***

Wireless links

Wired links



Router

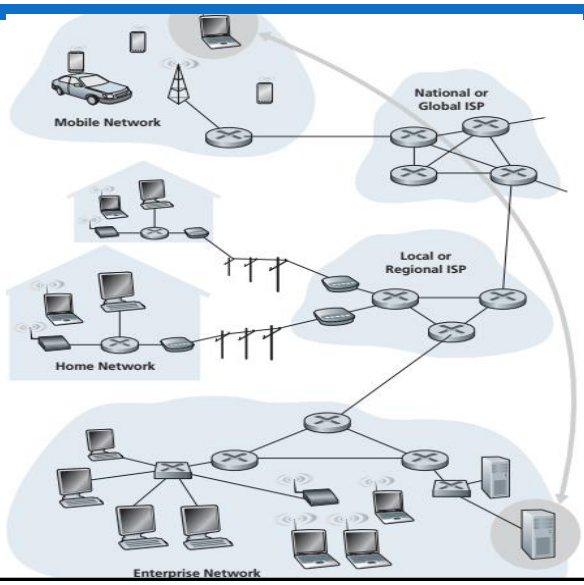
- ***Packet switches****:* forward packets (chunks of data)
  - ***Routers*** and ***switches***

The Network Edge

- Devices connected to the Internet are often referred to as end systems.
- End systems are also referred to as *hosts* because they host (that is, run) application programs
- They are referred to as end systems because they sit at the edge of the Internet,

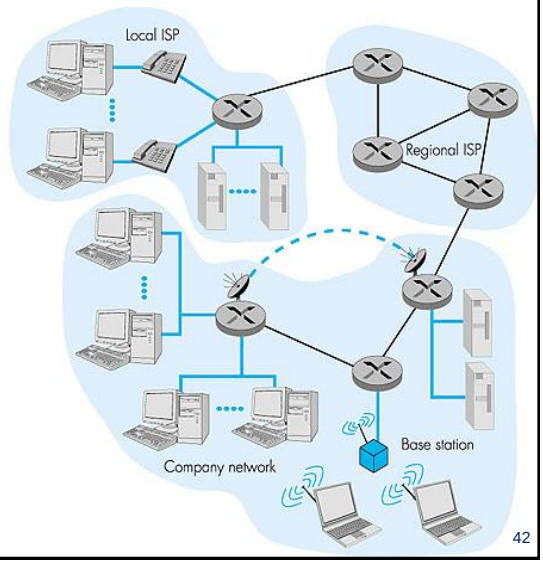
The Network Edge

- Network:
  - Mobile
  - Home
  - Enterprise



The access network

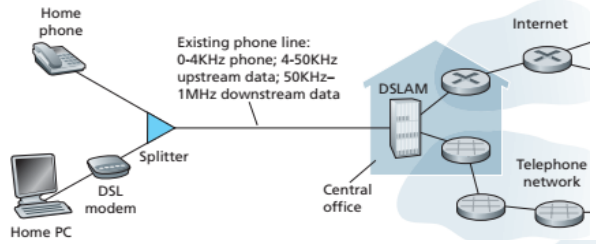
- The network that physically connects an end system to the first (edge router) on a path from the end system to any other distant end system



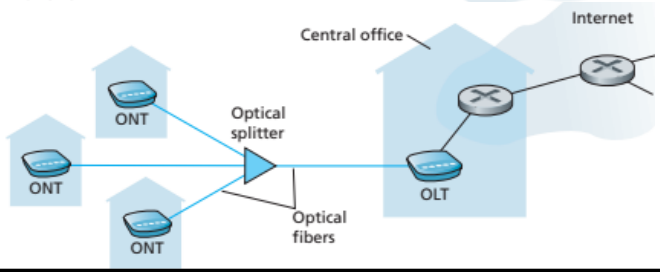
The access network

- Home Access: DSL, Cable, FTTH, Dial-Up, Satellite

DSL



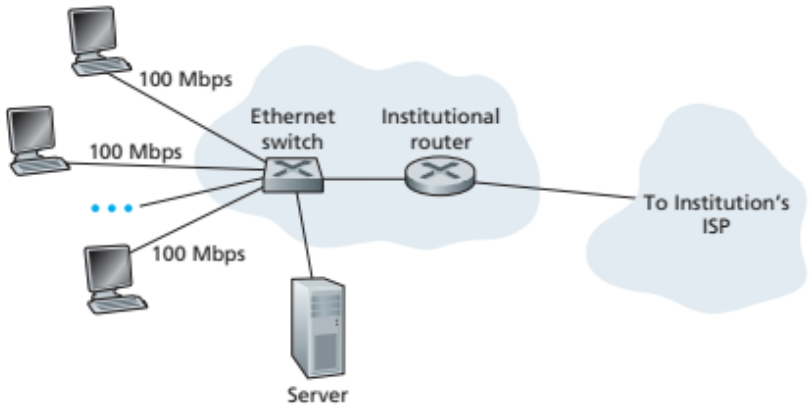
FTTH



The access network

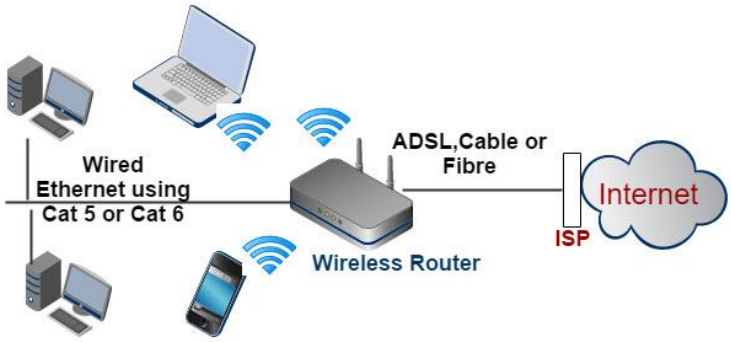
- Access in the Enterprise (and the Home): Ethernet and WiFi

Ethernet



The access network

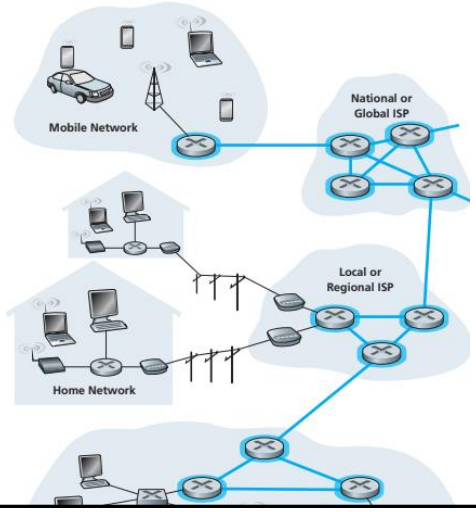
- Access in the Enterprise (and the Home): Ethernet and WiFi



Network Diagram-Typical Simple Home Network

The Network Core

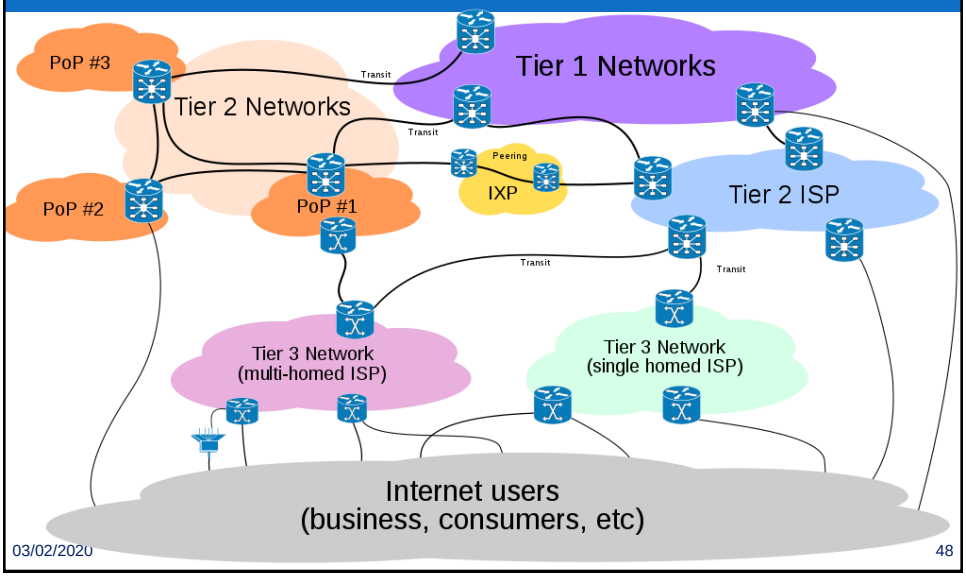
- the mesh of packet switches and links that interconnects the Internet's end systems



Internet structure: network of networks

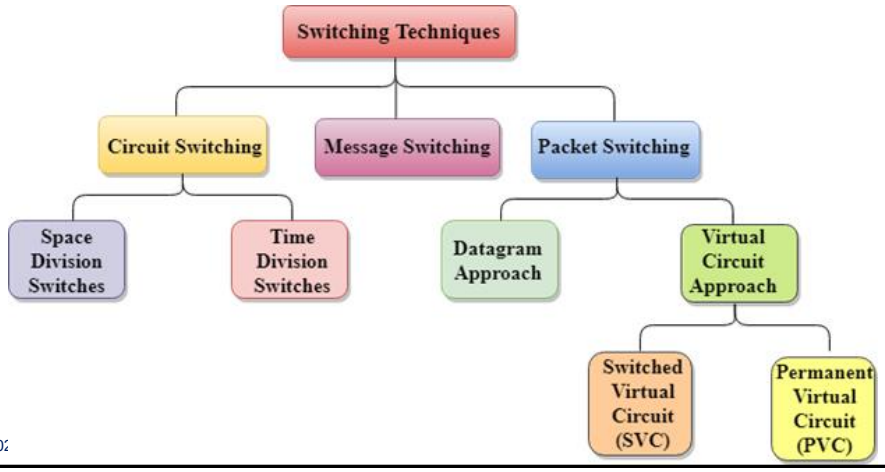
- Network Structure 1
  - interconnects all of the access ISPs with a single global transit ISP - a network of routers and communication links that not only spans the globe, but also has at least one router near each of the hundreds of thousands of access ISPs
- Network Structure 2,
  - consists of the hundreds of thousands of access ISPs and multiple global transit ISPs (the top tier and access ISPs at the bottom tier)
- Network Structure 3
  - multi-tier hierarchy – Internet
  - Add more points of presence (PoPs) - group of routers in the provider's network
- Network Structure 4
  - ISPs, regional ISPs, tier-1 ISPs, PoPs, multi-homing, peering, and IXPs
- Network Structure 5
  - Network Structure 4 by adding **content provider networks**

A Network of Networks



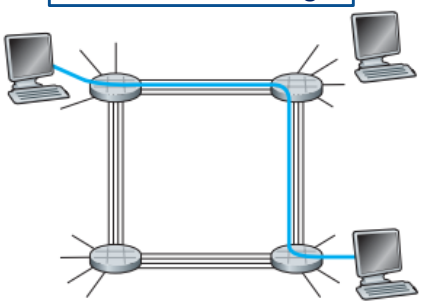
Data transmission: Switching Techniques

- The switching technique will decide the best route for data transmission.

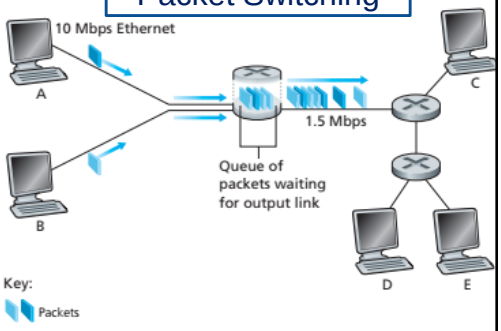


Switching Techniques

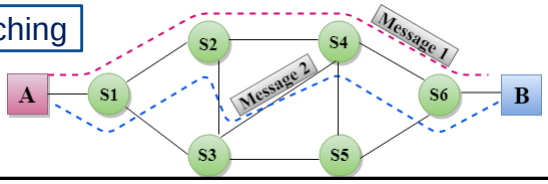
Circuit Switching



Packet Switching



Message Switching



Switching Techniques

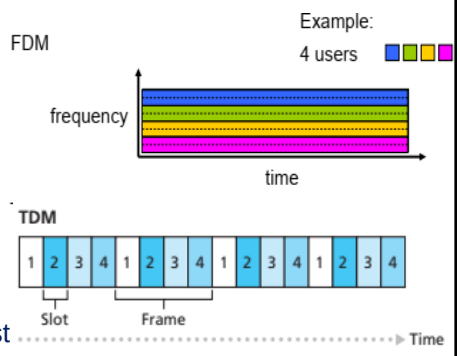
Circuit Switching Vs Packet Switching

Circuit Switching	Packet Switching
Physical path between source and destination	No physical path
All packets use same path	Packets travel independently
Reserve the entire bandwidth in advance	Does not reserve
Bandwidth Wastage	No Bandwidth wastage
No store and forward transmission	Supports store and forward transmission

3

Circuit Switching: FDM and TDM

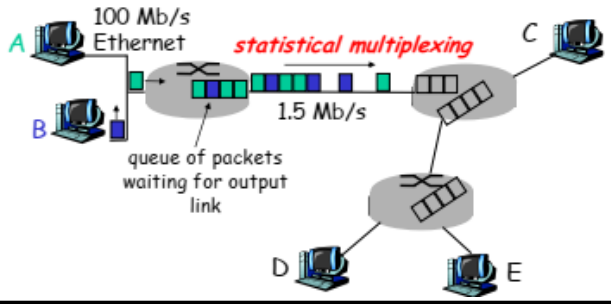
- FDM: the frequency spectrum of a link is divided up among the connections established across the link
- TDM: time is divided into frames of fixed duration, and each frame is divided into a fixed number of time slots
- Ex, How long does it take to send a file of 640,000 bits from host A to host B over a circuit- switched network?
  - All links are 1.536 Mbps
  - Each link uses TDM with 24 slots/sec
  - 500 msec to establish end-to-end circuit



1 circuit: (1.536 Mbps)/24 = 64 kbps
so it takes:
(640,000 bits)/(64 kbps) = 10 seconds
to transmit the file
add the circuit establishment time,
=> 10.5 seconds to send the file

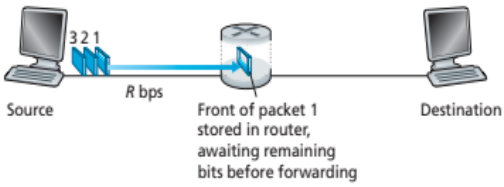
Packet Switching: Statistical Multiplexing

- The message splits into packets that are given a unique number to identify their order at the receiving end.
- Every packet contains some information in its headers such as source address, destination address and sequence numbe
- Sequence of A & B packets does not have fixed pattern, bandwidth shared on demand => **statistical multiplexing**.
- TDM: each host gets same slot in revolving TDM frame



Packet-switching: store-and-forward

- store and forward: entire packet must arrive at router before it can be transmitted on next link
- takes L/R seconds to transmit (push out) packet of L bits on to link at R bps
- Example:
  - L = 7.5 Mbits
  - R = 1.5 Mbps
  - transmission delay = 15sec



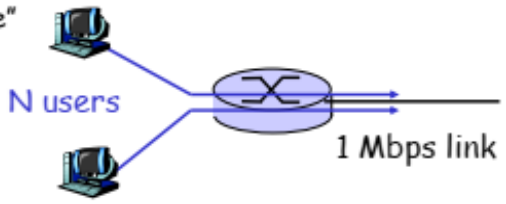
- the general case of sending one packet from source to destination over a path consisting of *N* links each of rate *R* (N-1 router) between source and destination

$$d_{end-to-end} = N \frac{L}{R}$$

Packet switching versus circuit switching

Packet switching allows more users to use network!

- 1 Mb/s link
- each user:
  - 100 kb/s when "active"
  - active 10% of time
- *circuit-switching:*
  - 10 users
- *packet switching:*
  - with 35 users, probability > 10 active at same time is less than .0004



Q: how did we get value 0.0004?

Contents

Packet-Switched Networks

- Store and forward:
  - entire packet must arrive at router before it can be transmitted on next link
- Ideally,
  - Internet services need to move as much data as we want between any two end systems, without any loss of data.
  - computer networks necessarily constrain throughput (the amount of data per second that can be transferred) between end systems,

⇒ Problems: Delay, Loss, and Throughput

Delay in Packet-Switched Networks

- Packets experience **delay** on end-to-end path
- **Four** sources of delay at each hop

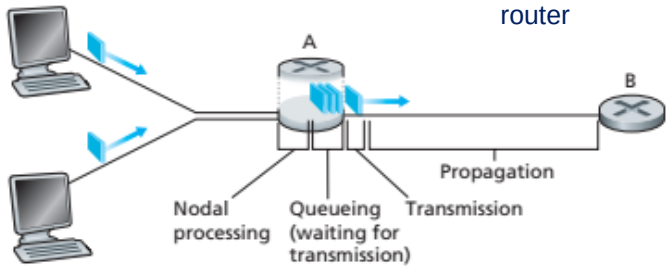
1. Processing delay:

– The time to examine the packet's header and determine where to direct the packet

2. Queueing delay

– Time waiting at output link for transmission

– Depends on congestion level of router



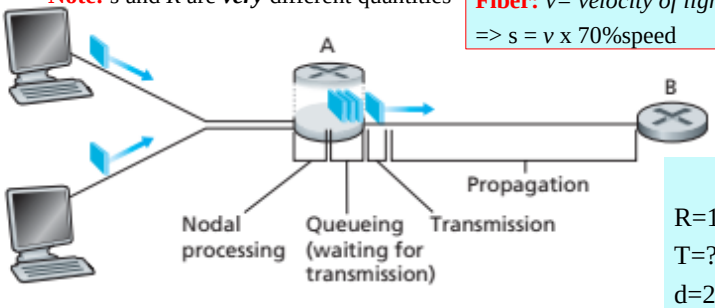
Delay in Packet-Switched Networks

3. Transmission Delay:
R = Link bandwidth (bps)
L = Packet length (bits)
Time to send bits into link:
T = *L/R*

4. Propagation Delay:
d = Length of physical link
s = propagation speed in medium (~2×10^8 m/sec)
propagation delay: $\mathbf{d_{prop} = d/s}$

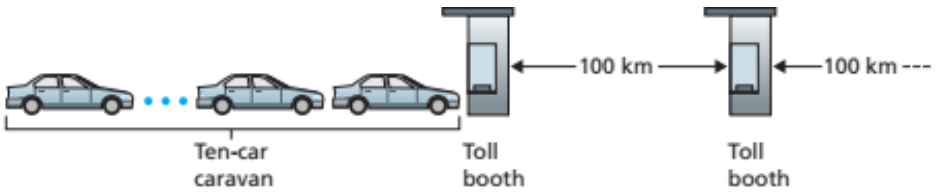
Fiber: *v= velocity of light* =$3x10^8$m/s
=> s = *v* x 70%speed

Note: *s* and *R* are ***very*** different quantities



Ex:
R=1kbps, L=1Kb
T=?
d=20km, d_{prop} ?

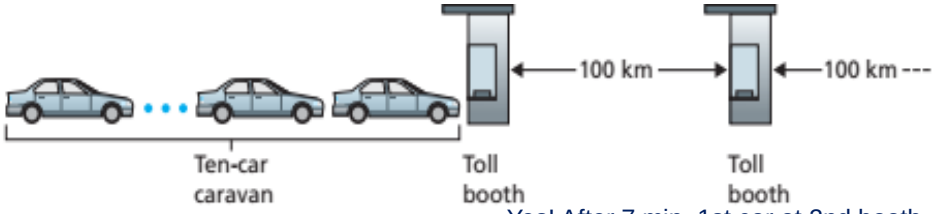
Caravan analogy



- cars "propagate" at 100 km/hr
- toll booth takes 12 sec to service 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*10 = 120 sec = 2minutes
- Time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr)= 1 hr
- A:

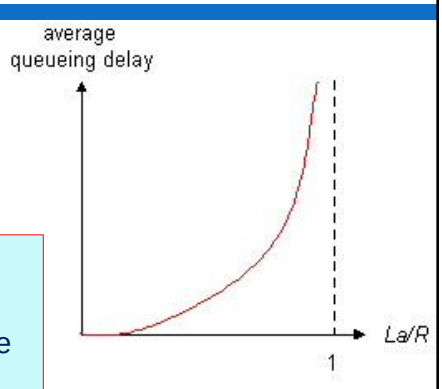
Caravan analogy



- Cars now "propagate" at 1000 km/hr
- Toll booth now takes 1 min to service a car
- Q: Will cars arrive to 2nd booth before all cars serviced at 1st booth?
  - Yes! After 7 min, 1st car at 2nd booth and 3 cars still at 1st booth.
  - 1st bit of packet can arrive at 2nd router before packet is fully transmitted at 1st router!
  - See Ethernet applet at AWL Web site
  - the first bits in a packet can arrive at a router while many of the remaining bits in the packet are still waiting to be transmitted by the preceding router

Queueing delay

- d_{queue} is the time it takes for the packet to be transmitted onto the link
- the length of this time is defined by the number of packets that was added to the queue prior to this packet



R = Link bandwidth (bps)
L = Packet length (bits)
a = Average packet arrival rate
Traffic intensity = *La/R*

- *La/R* ~ 0: Average queueing delay small
- *La/R* → 1: Delays become large
- *La/R* > 1: More "work" arriving than can be serviced, average delay infinite!

Nodal delay

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

- d_{proc} = processing delay: typically a few microsecs or less
  - The time that Packet will be taken by receiver and then it will be processed
- d_{queue} = queuing delay: depends on congestion
  - d_{queue} is the time it takes for the packet to be transmitted onto the link

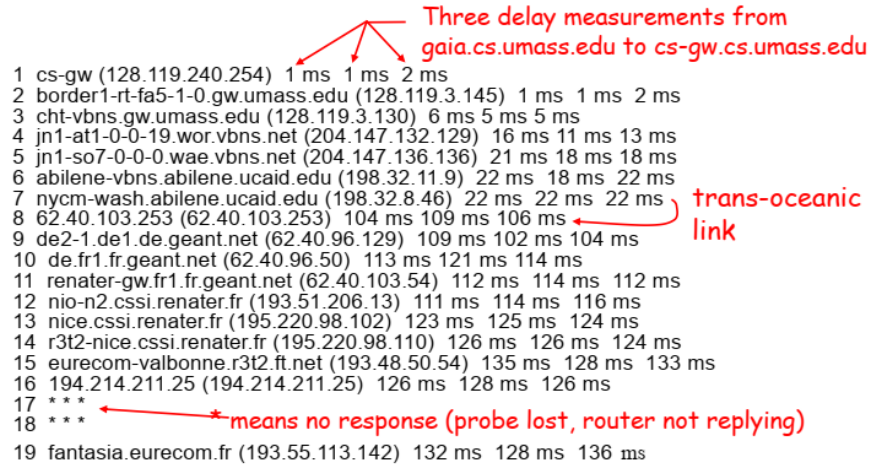
d_{proc} and d_{queue} depend on the speed of processor

- d_{trans} = transmission delay: significant for low-speed links
- d_{prop} = propagation delay: a few microsecs to hundreds of msecs

- Note: speed of processor is very high, d_{queue} and d_{proc} are less

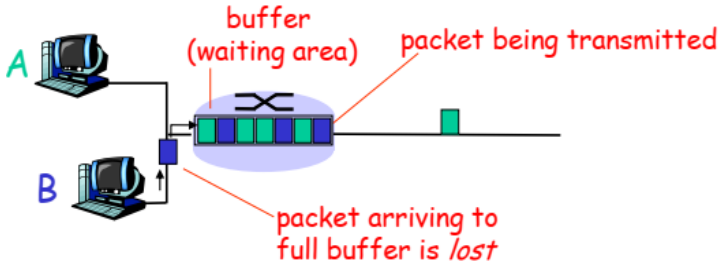
"Real" Internet Delays and Routes

`traceroute` (or `tracert`): Routers, round-trip delays on source-dest path
Also: `pingplotter`, various Windows programs



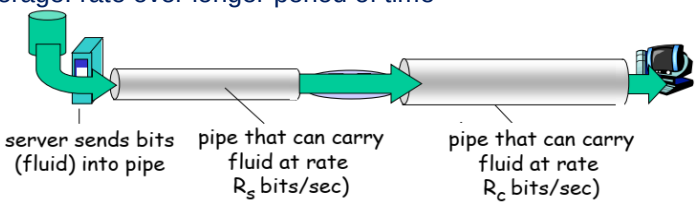
Packet loss

- queue (aka buffer) preceding link in buffer has finite capacity
  - a packet can arrive to find a full queue.
  - With no place to store such a packet, a router will drop that packet; that is, the packet will be lost
- lost packet may be retransmitted by previous node, by source end system, or not at all

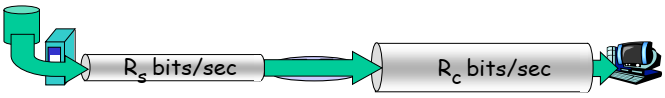


Throughput

- throughput: rate bits transferred between sender/receiver
  - instantaneous: rate at given point in time
  - average: rate over longer period of time



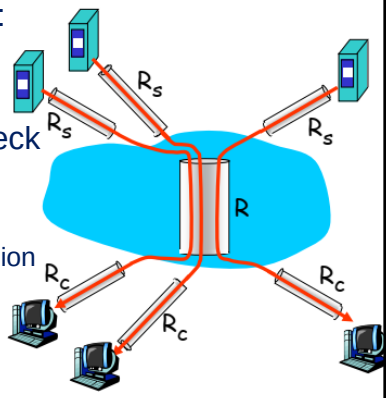
- What is average end-end throughput?
  - $R_s < R_c$
  - $R_s > R_c$



- link on end-end path that constrains end-end throughput

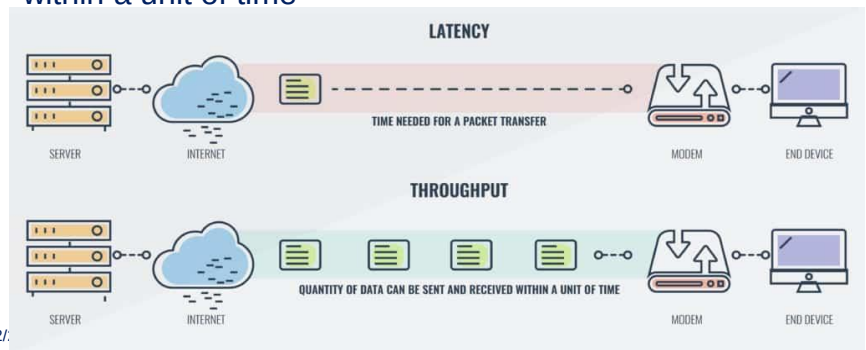
Throughput: Internet scenario

- 10 connections (fairly) share backbone bottleneck link R bits/sec
- per-connection end-end throughput: **min(R_c,R_s,R/10)**
- in practice: R_c or R_s is often bottleneck
- Ex:
  - *Rs* = 2 Mbps, *Rc* = 1 Mbps, *R* = 5 Mbps, and the common link divides its transmission rate equally among the 10 downloads
  - the end-to-end throughput for each download is now reduced to…?



Latency vs Throughput

- Latency – The time taken for a packet to be transferred across a network. You can measure this as one-way to its destination or as a round trip.
- Throughput – The quantity of data being sent and received within a unit of time



Latency vs Throughput

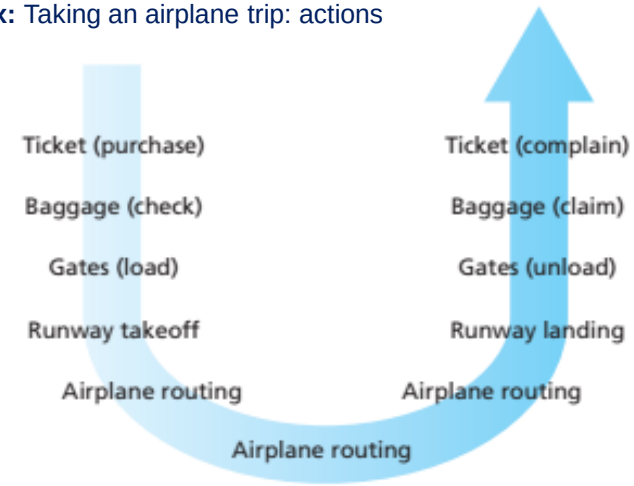
- The more routers a packet has to travel through the more latency there is because each router has to process the packet
- Throughput is a good way to measure the performance of the network connection because it tells you how many messages are arriving at their destination successfully
- Both network latency and throughput are important because they have an effect on how well your network is performing.
- The bandwidth of your network is limited to the standard of your internet connection and the capabilities of your network devices
- Tools for Measuring Network Throughput, BW
  - **SolarWinds Flow Tool Bundle**
  - speedtest

Excersices

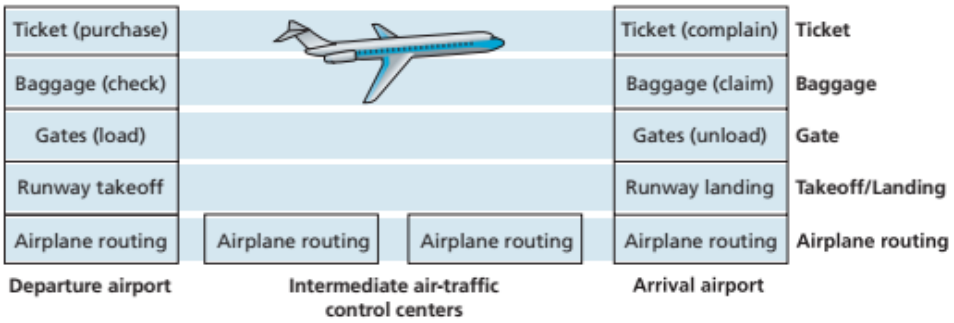
- How long does it take a **packet of length** 1,000 bytes to propagate over a **link of distance** 2,500 km, **propagation speed** 2.5 x 10^8 m/s, and **transmission rate 2 Mbps**? More generally, how long does it take a packet of length L to propagate over a link of distance d, propagation speed s, and transmission rate R bps? Does this delay depend on packet length? Does this delay depend on transmission rate?
- Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has **three** links, of rates R_1 = 500 kbps, R_2 = 2 Mbps, and R_3 = 1 Mbps.
  - a. Assuming no other traffic in the network, what is the throughput for the file transfer?
  - b. Suppose **the file is 4 million bytes**. Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B?
  - c. Repeat (a) and (b), but now with R2 reduced to 100 kbps.

Protocol Layers and Their Service Models

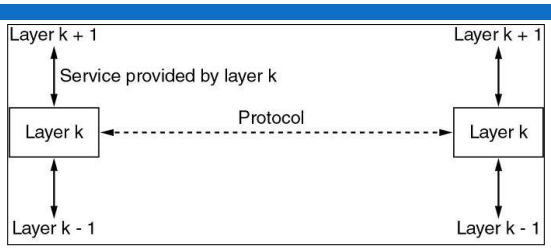
- **Layered Architecture**
  - **Ex:** Taking an airplane trip: actions



Protocol Layers



Protocol Layers

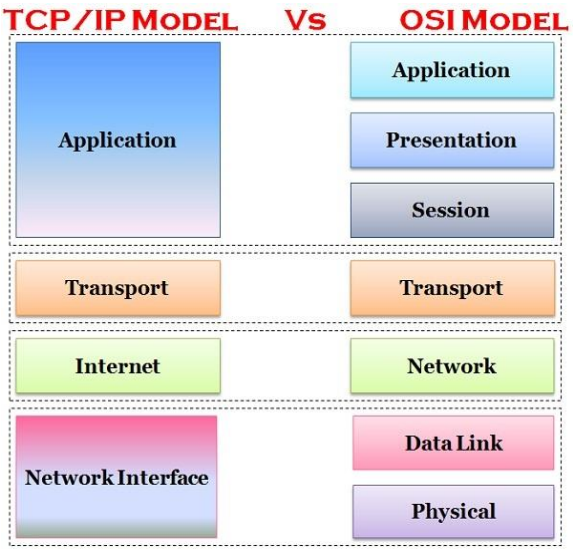


- **Less complex:** network models break the concepts into smaller parts.
- **Standard interfaces:** allow multiple vendors to create products that fill a particular role, with all the benefits of open competition.
- **Easier to learn:** more easily discuss and learn about the many details of a protocol specification.
- **Easier to develop:** Reduced complexity allows easier program changes and faster product development.
- **Multivendor interoperability:** meet the same networking standards means that computers and networking gear from multiple vendors can work in the same network.
- **Modular engineering:** implements higher layers, another vendor can write software that implements the lower layers

Contents

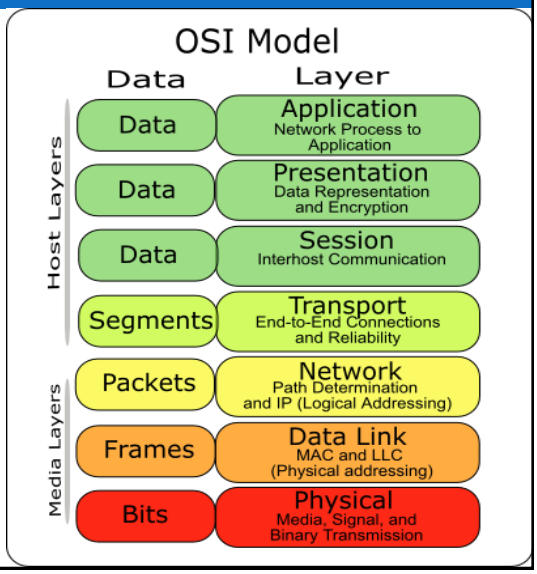
OSI & TCP/IP Models

The Internet protocol stack and OSI reference model



OSI model

- the OSI model can be used as a standard of comparison to other networking models
- OSI did have a well-defined set of functions associated with each of its seven layers,



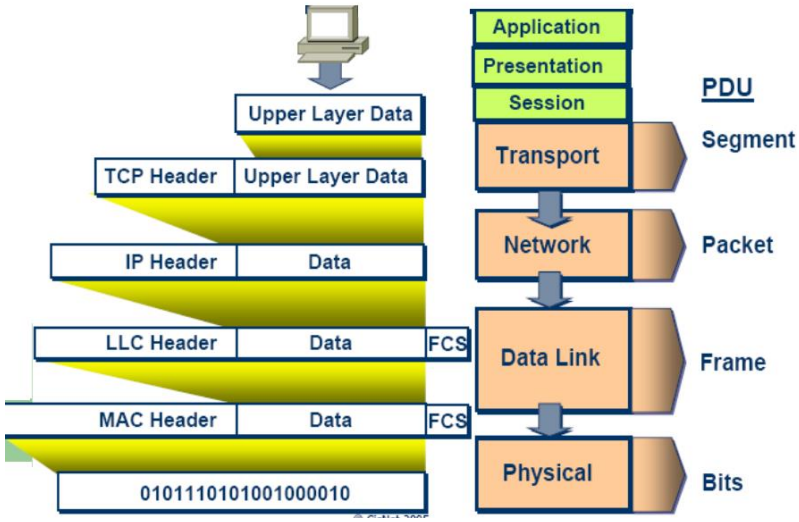
OSI model - functions

Layer	Functional Description
7	**Application.** Provides an interface from the application to the network by supplying a protocol with actions meaningful to the application
6	**Presentation**. negotiates data formats, such as ASCII text, JPEG.
5	**Session**. provides methods to group multiple bidirectional messages into a workflow for easier management and easier backout of work that happened if the entire workflow fails.
4	**Transport**. focuses on data delivery between the two endpoint hosts
3	**Network**. defines logical addressing, routing, and the routing protocols
2	**Data link**. defines the protocols for delivering data over a particular single type of physical network
1	**Physical**. defines the physical characteristics of the transmission medium

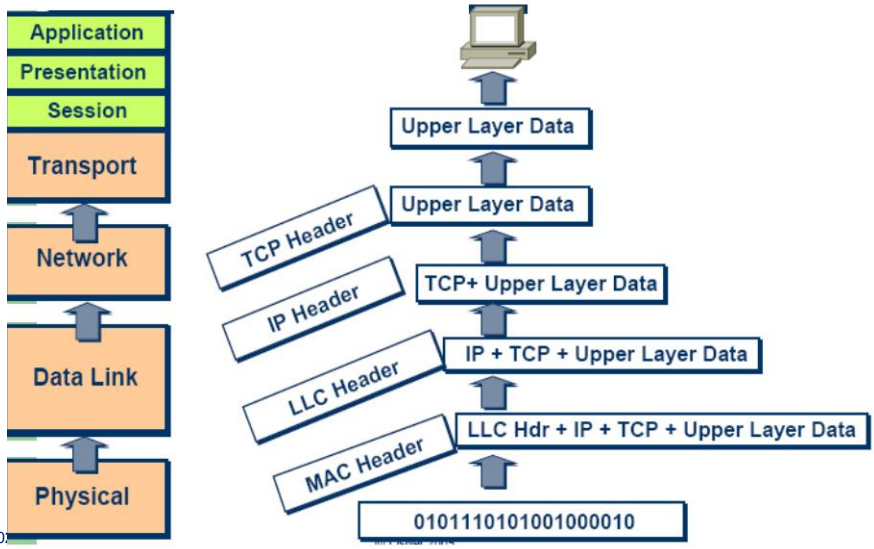
OSI model - protocols

OSI Model	Protocols
Application Layer	DNS, DHCP, FTP, HTTPS, IMAP, LDAP, NTP,POP3, RTP, RTSP, SSH, SIP, SMTP, SNMP, Telnet, TFTP
Presentation Layer	JPEG, MIDI, MPEG, PICT, TIFF
Session Layer	NetBIOS, NFS, PAP, SCP, SQL, ZIP
Transport Layer	TCP, UDP
Network Layer	ICMP, IGMP, IPsec, IPv4, IPv6, IPX, RIP
Data Link Layer	ARP, ATM, CDP, FDDI, Frame Relay, HDLC, MPLS, PPP, STP, Token Ring
Physical Layer	Bluetooth, Ethernet, DSL, ISDN, 802.11 Wi-Fi

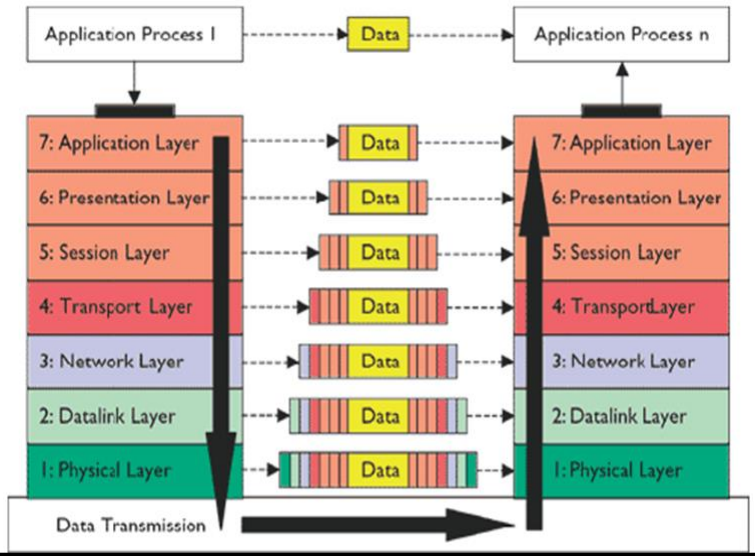
OSI Encapsulation



OSI De-encapsulation



OSI Transmission



ex

a. Physical Provides ---------- error-free packet delivery

b. Data Link ---------- Establishes responses between applications

c Networking ---------- Deals with mechanical and electrical communications

d. Transport ---------- Formats, encrypts, and compresses data

e. Session ---------- Determines routes and addressing

f. Presentation ---------- Provides user access to the environment

g. Application ---------- Packages bits into data

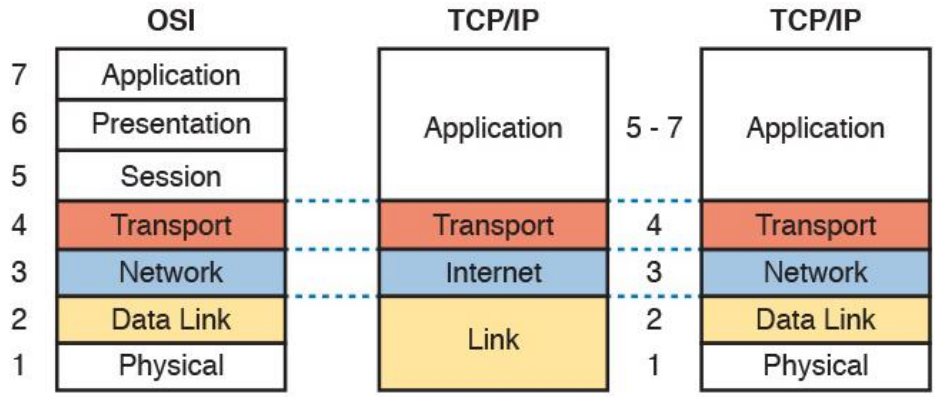
Contents

- History of computer network
- Computer network
- Network topology
- Network protocol
- Network Components
- Internet
- Packet-Switched Networks problems:
  - Delay, Loss, and Throughput in
- Protocol Layers and Their Service Models
- OSI model
- TCP/IP model

TCP/IP

- Introduction
- TCP/IP Encapsulation
- Internet protocol stack
- Application layer
- Transport layer
- Network layer
- Link layer

OSI, TCP/IP, TCP/IP updated

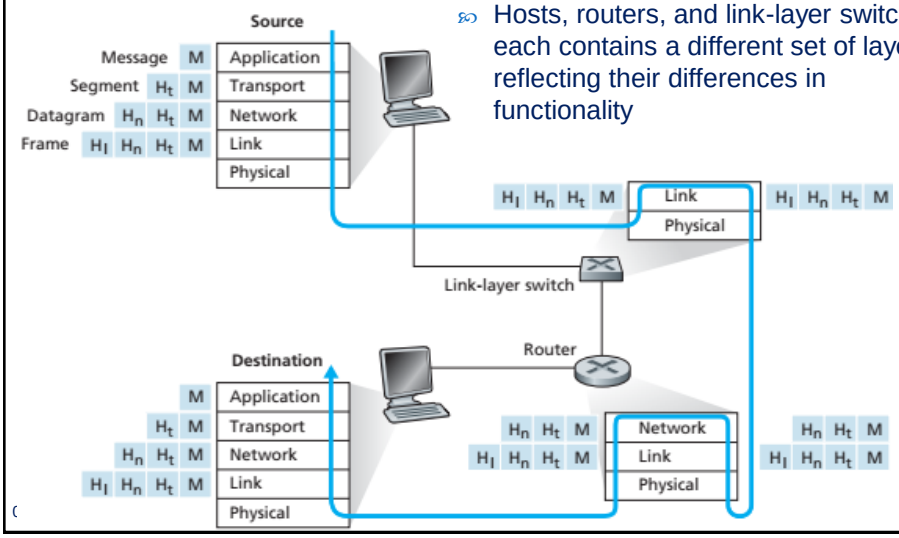


Introduction to TCP/IP

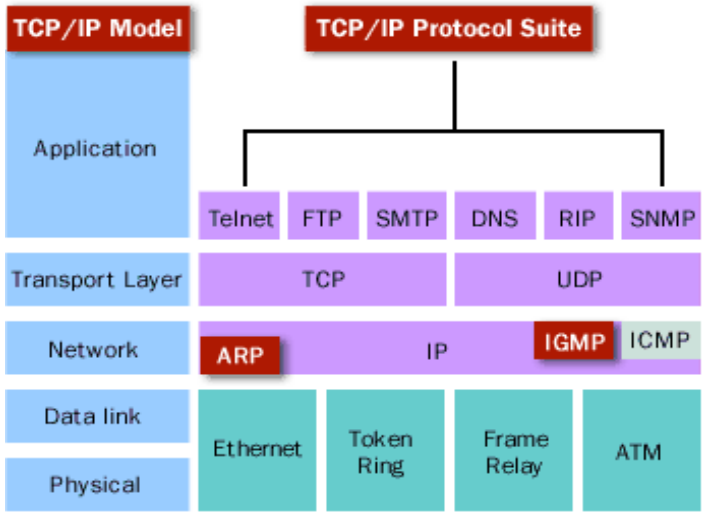
- The TCP/IP stands for Transmission Control Protocol / Internet Protocol suite
  - was designed and implemented by Department of Defense(DoD) to maintain communication.
  - TCP first came on the scene way back in 1973, It was divided into two distinct protocols TCP & IP.
  - Later in 1983, TCP/IP replaced the Network Control Protocol (NCP).

TCP/IP - Encapsulation

- Hosts, routers, and link-layer switches; each contains a different set of layers, reflecting their differences in functionality



TCP/IP - The Internet protocol stack



Application layer

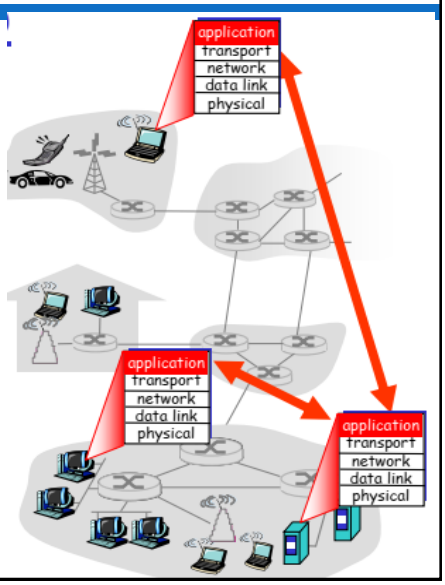
- Introduction
- Application architectures
- Socket programming

Some network apps

- e-mail
- web
- text messaging
- remote login
- P2P file sharing
- multi-user network games
- streaming stored video (YouTube, Hulu, Netflix)
- voice over IP (e.g., Skype)
- real-time video conferencing
- social networking
- search
- …
- …

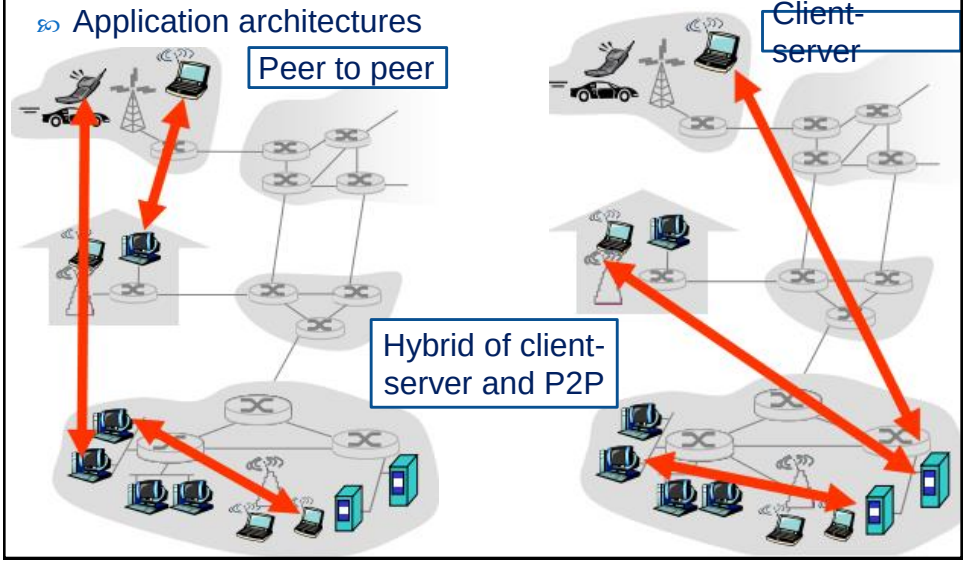
TCP/IP Application layer

- provide services to the application software running on a computer.
- write programs that
  - run on (different) end systems
  - communicate over network e.g., web server software communicates with browser software
- No need to write software for network-core devices
  - Network-core devices do not run user applications
  - applications on end systems allows for rapid app development, propagation



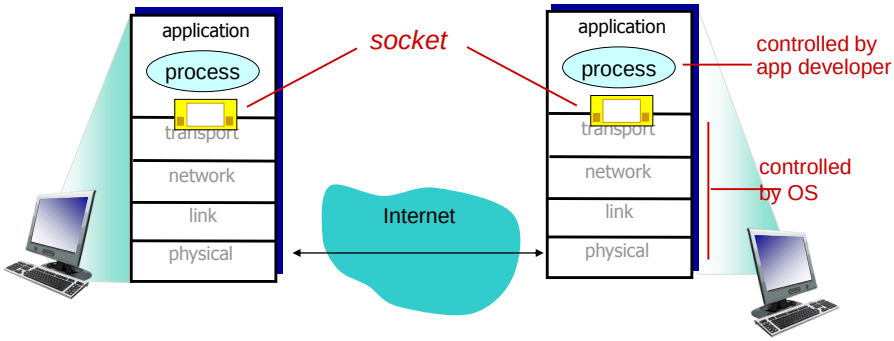
TCP/IP Application layer

- Application architectures



TCP/IP Application layer

- **Socket**
  - process sends/receives messages to/from its socket
  - socket analogous to door
    - sending process shoves message out door
    - sending process relies on transport infrastructure on other side of door to deliver message to socket at receiving process



Addressing processes

- to receive messages, process must have *identifier*
- host device has unique 32-bit IP address
- *Q:* does IP address of host on which process runs suffice for identifying the process?
  - *A:* no, *many* processes can be running on same host
- *identifier* includes both IP address and port numbers associated with process on host.
- example port numbers:
  - HTTP server: 80
  - mail server: 25
- to send HTTP message to gaia.cs.umass.edu web server:
  - IP address: 128.119.245.12
  - port number: 80
- more shortly…

2-95

App-layer protocol defines

- types of messages exchanged,
  - e.g., request, response
- message syntax:
  - what fields in messages & how fields are delineated
- message semantics
  - meaning of information in fields
- rules for when and how processes send & respond to messages

open protocols:

- defined in RFCs
- allows for interoperability
- e.g., HTTP, SMTP

proprietary protocols:

- e.g., Skype

2-96

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

timing
- some apps (e.g., Internet telephony, interactive games) require low delay to be "effective"

throughput
- some apps (e.g., multimedia) require minimum amount of throughput to be "effective"
- other apps ("elastic apps") make use of whatever throughput they get

security
- encryption, data integrity, …

Transport service requirements: common apps

application	data loss	throughput	time sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	no loss	elastic	no
real-time audio/video	loss-tolerant	audio: 5kbps-1Mbps video:10kbps-5Mbps	yes, 100's msec
stored audio/video	loss-tolerant	same as above	
interactive games	loss-tolerant	few kbps up	yes, few secs
text messaging	no loss	elastic	yes, 100's msec yes and no

Internet transport protocols services

TCP service:

- *reliable transport* between sending and receiving process
- *flow control:* sender won't overwhelm receiver
- *congestion control:* throttle sender when network overloaded
- *does not provide:* timing, minimum throughput guarantee, security
- *connection-oriented:* setup required between client and server processes

UDP service:

- *unreliable data transfer* between sending and receiving process
- *does not provide:* reliability, flow control, congestion control, timing, throughput guarantee, security, orconnection setup,

Q: why bother? Why is there a UDP?

Internet apps: application, transport protocols

application	application layer protocol	underlying transport protocol
e-mail	SMTP [RFC 2821]	TCP
remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	HTTP (e.g., YouTube), RTP [RFC 1889]	TCP or UDP
Internet telephony	SIP, RTP, proprietary (e.g., Skype)	TCP or UDP

TCP/IP

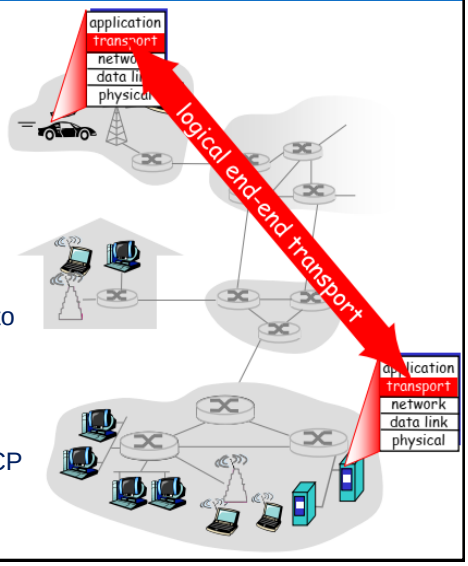
- TCP/IP Encapsulation
- Internet protocol stack
- Application layer
- Transport layer
- Network layer
- Link layer

Transport layer

- Introduction
- TCP and UDP
- TCP - reliable data transfer
- TCP Connection Management
- TCP flow Control
- TCP Congestion Control

TCP/IP transport layer

- provide logical communication between app processes running on different hosts
- transport protocols run in end systems
  - send side: breaks app messages into segments, passes to network layer
  - rcv side: reassembles segments into messages, passes to app layer
- more than one transport protocol available to apps
  - Transmission Control Protocol - TCP
  - User Datagram Protocol - UDP

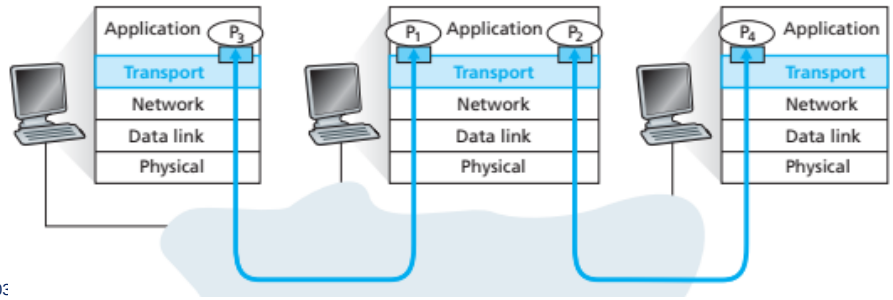


TCP/IP transport layer

- a multiplexing/demultiplexing service is needed for all computer networks

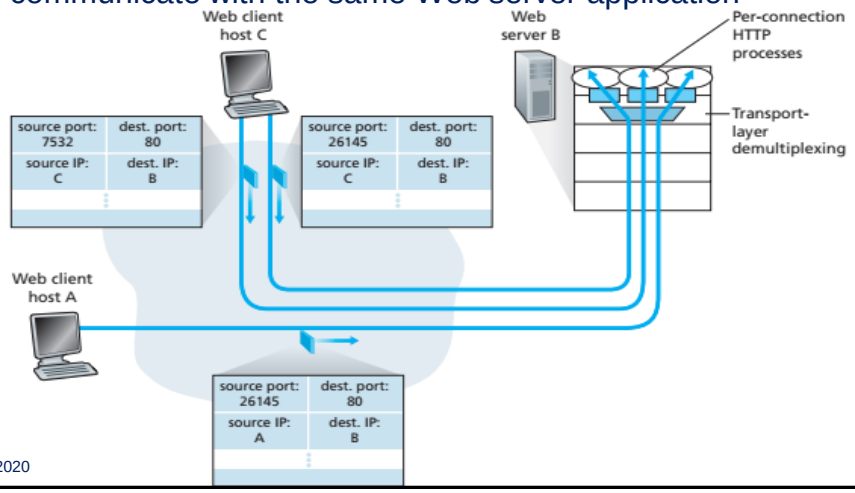
multiplexing at sender:
handle data from multiple sockets, add transport header (later used for demultiplexing)

demultiplexing at receiver:
use header info to deliver received segments to correct socket



TCP/IP transport layer

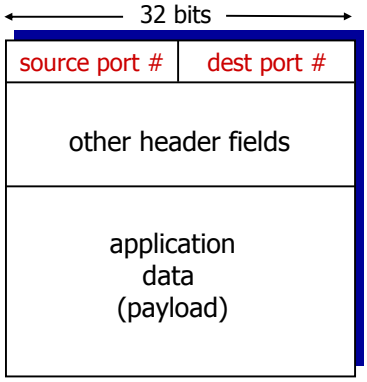
- Two clients, using the same destination port number (80) to communicate with the same Web server application



TCP/IP transport layer

How demultiplexing works

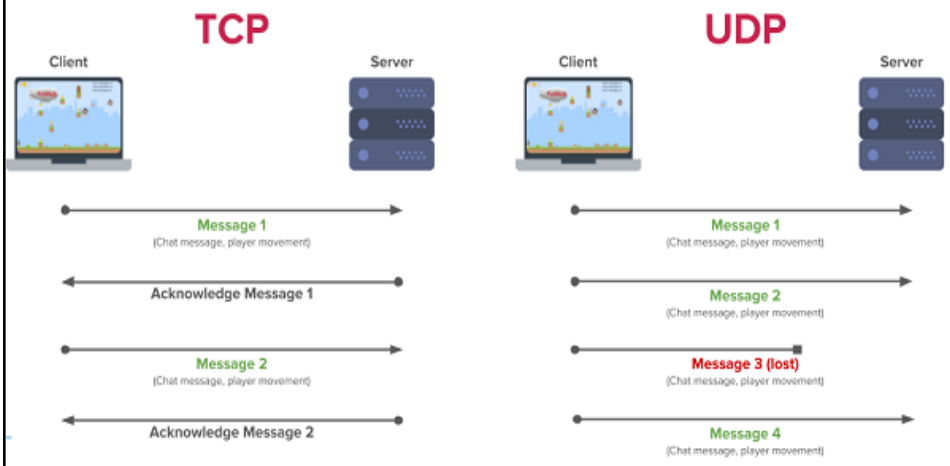
- host receives IP datagrams
  - each datagram has source IP address, destination IP address
  - each datagram carries one transport-layer segment
  - each segment has source, destination port number
- host uses *IP addresses & port numbers* to direct segment to appropriate socket



TCP/UDP segment format

TCP/IP transport layer

- TCP and UDP



TCP/IP transport layer

TCP

- *reliable transport* between sending and receiving process
- *flow control:* sender won't overwhelm receiver
- *congestion control:* throttle sender when network overloaded
- *does not provide:* timing, minimum throughput guarantee, security
- *connection-oriented:* setup required between client and server processes

UDP

- *unreliable data transfer* between sending and receiving process
- *does not provide:* reliability, flow control, congestion control, timing, throughput guarantee, security, orconnection setup,

Q: why bother? Why is there a UDP?

TCP/IP transport layer

- Header format TCP & UDP

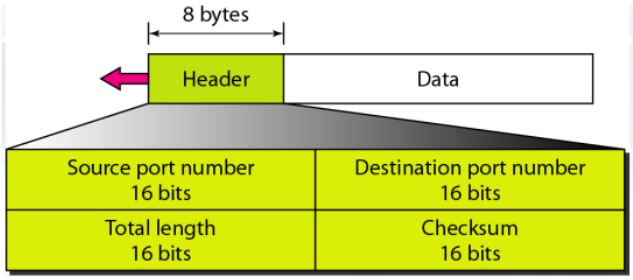
TCP Segment Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Sequence Number			
64	Acknowledgment Number			
96	Data Offset	Res	Flags	Window Size
128	Header and Data Checksum		Urgent Pointer	
160...	Options			

UDP Datagram Header Format

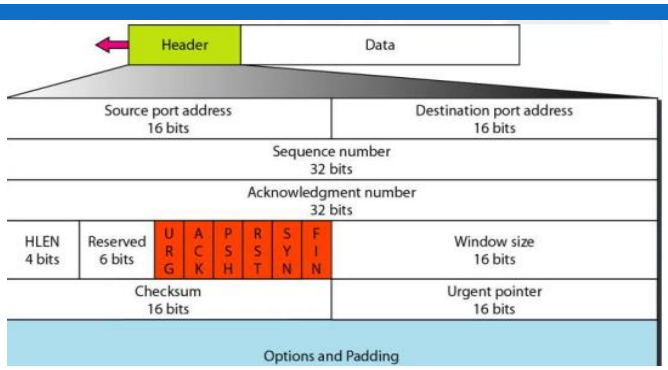
Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Length		Header and Data Checksum	

UDP



1. Source Port - is used to identify the source port of the packet.
2. Destination Port - is used identify application level service on destination machine.
3. Length - specifies the entire length of UDP packet (including header).
It is 16-bits field and minimum value is 8-byte, i.e. the size of UDP header itself
4. Checksum - stores the checksum value generated by the sender before sending.
IPv4 has this field as optional so when checksum field does not contain any value it is made 0 and all its bits are set to zero.

TCP



1. Source Port (16-bits) - of the application process on the sending device.
2. Destination Port (16-bits) - of the app process on the receiving device.
3. Sequence Number (32-bits) - data bytes of a segment in a session.
4. Acknowledgement Number (32-bits) - When ACK flag is set, this number contains the next sequence number of the data byte expected and works as acknowledgement of the previous data received.

TCP

5. Data Offset (4-bits) - implies both, the size of TCP header (32-bit words) and the offset of data in current packet in the whole TCP segment.
6. Reserved (3-bits) - for future use and all are set zero by default
7. Flags(1-bit each):
- URG - Urgent Pointer field has significant data and should be processed.
- ACK - If ACK is cleared to 0, it indicates that packet does not contain ack.
- PSH - When set, it is a request to the receiving station to PUSH data (as soon as it comes) to the receiving application without buffering it.
- RST - It is used to restart a connection.
- SYN - This flag is used to set up a connection between hosts.
- FIN - release a connection and no more data is exchanged thereafter.

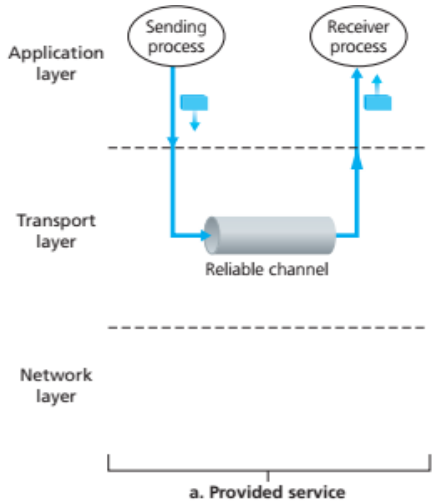
8. Windows Size - is used for flow control between two stations and indicates the amount of buffer (in bytes) the receiver has allocated for a segment, i.e. how much data is the receiver expecting.
9. Checksum - contains the checksum of Header, Data and Pseudo Headers.
10. Urgent Pointer - It points to the urgent data byte if URG flag is set to 1.
11. Options - It facilitates additional options which are not covered by the regular header.

TCP & UDP

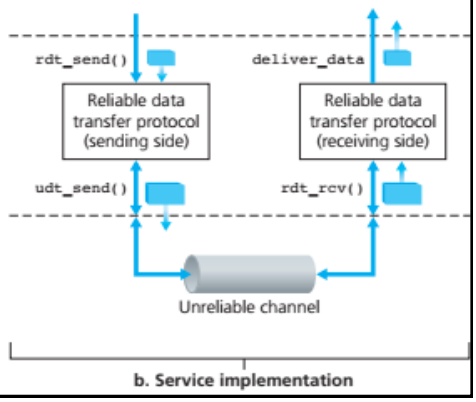
No.	TCP	UDP
1.	Connection Oriented Protocol	Connection-less Protocol
2.	Connection in byte stream	Connection in message stream
3.	It does't support multicasting and broadcasting	It supports broadcasting
4.	It provides error control and flow control	Error Control and Flow control is not provided
5.	Supports full Duplex	Does not support full Duplex
6.	TCP packet is called as Segment	UDP packet is called as User Datagram

Reliable data transfer

- implementing reliable data transfer at layer: link, transport, app



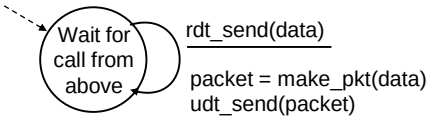
- the layer *below* the reliable data transfer protocol may be unreliable



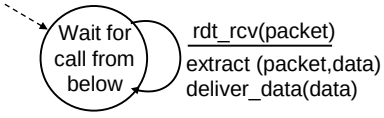
TCP – a reliable data transfer protocol

- underlying channel perfectly reliable
  - No Bit Errors
  - No Packets Loss

Sender

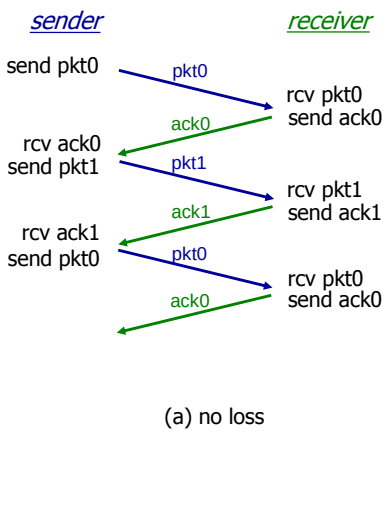


Receiver

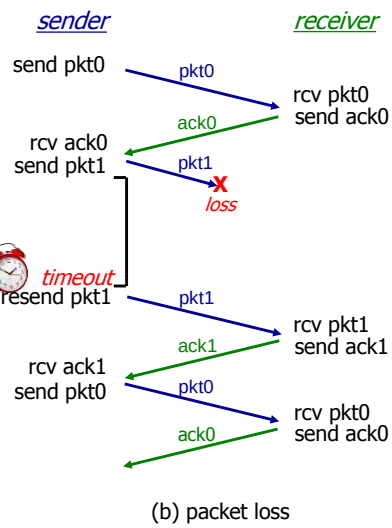


- Rdt over a Perfectly Reliable Channel: rdt1.0.
  - No need feedback, but bits in a packet may be corrupted.
- Rdt over a Channel with Bit Errors: rdt2.0: control mess "OK" or "NOT."
  - retransmission (ARQ (Automatic Repeat reQuest): detect error, feedback, retransmits
- Rdt over a Lossy Channel with Bit Errors: rdt3.0
  - corrupting bits, the underlying channel can lose packets as well,
  - checksumming, sequence numbers, ACK packets, and retransmissions

TCP - reliable data transfer (rdt3.0)

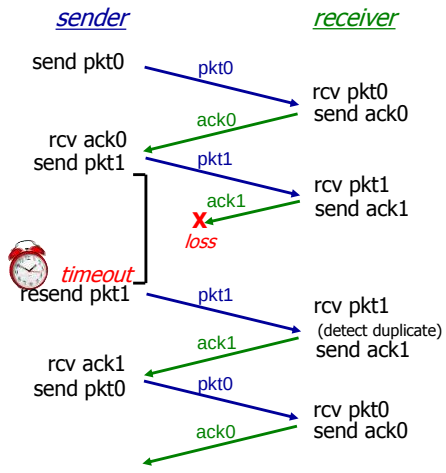


(a) no loss

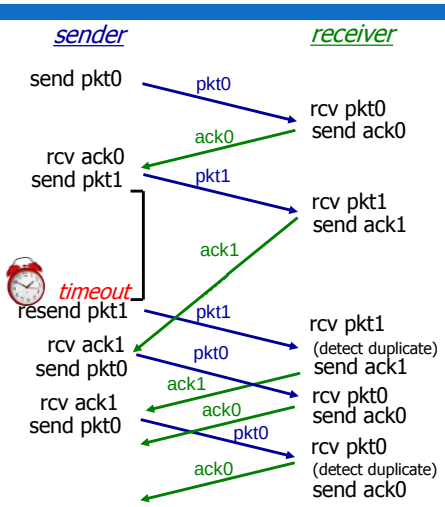


(b) packet loss

TCP - reliable data transfer (rdt3.0)



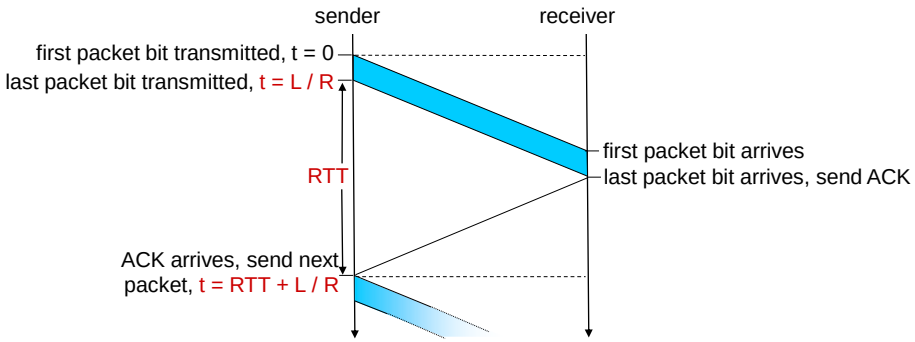
(c) ACK loss



(d) premature timeout/ delayed ACK

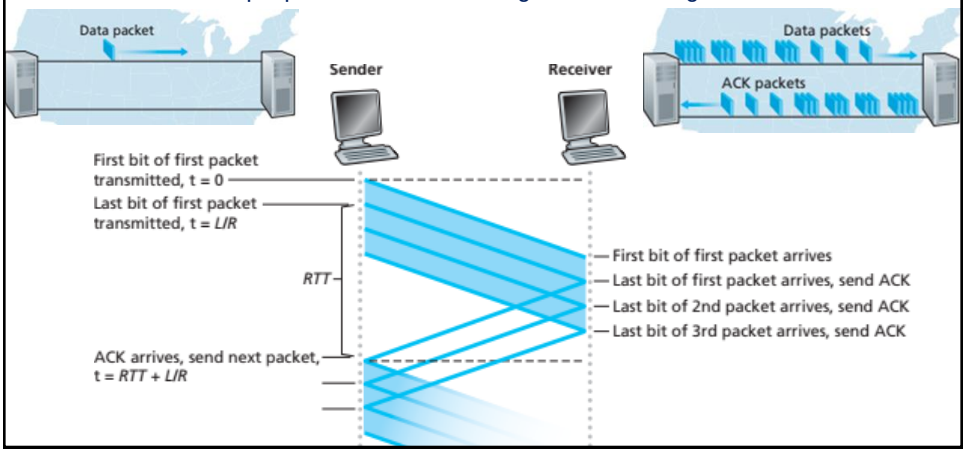
TCP - reliable data transfer (rdt)

- stop-and-wait operation



TCP - reliable data transfer (rdt)

- **Pipelined Reliable Data Transfer Protocols**
  - stop-and-wait protocol: delay, the poor performance
  - => send multiple packets without waiting for acknowledgments



TCP - reliable data transfer (rdt)

- **Pipelined protocols:**

Go-back-N:

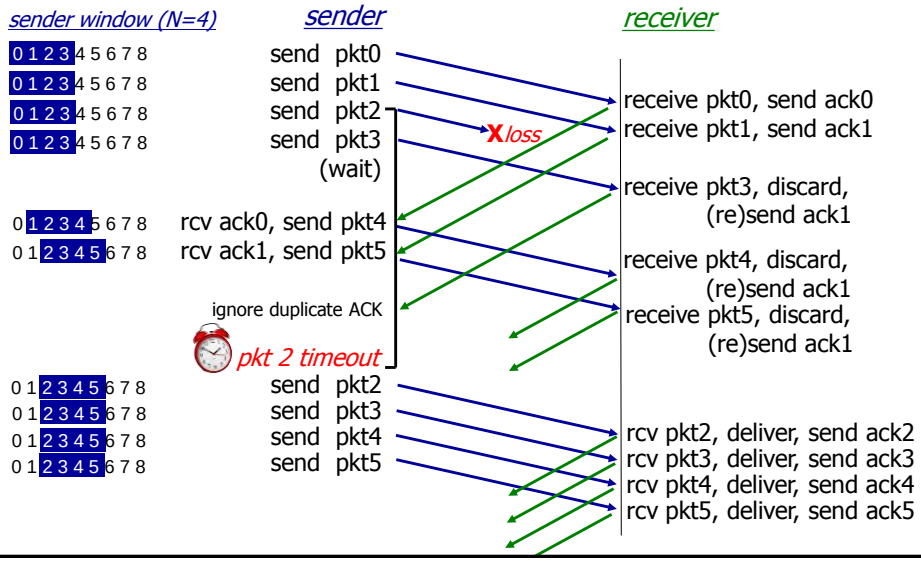
- sender
  - can have up to N unACKed packets in pipeline
- receiver
  - only sends *cumulative ACK*
  - doesn't ACK packet if there's a gap
- sender
  - has timer for oldest unACKed packet
  - when timer expires, retransmit *all* unACKed packets

Selective Repeat:

- sender
  - can have up to N unACKed packets in pipeline
- receiver
  - sends *individual ACK* for each packet
- sender
  - maintains timer for each unACKed packet
  - when timer expires, retransmit only that unACKed packet

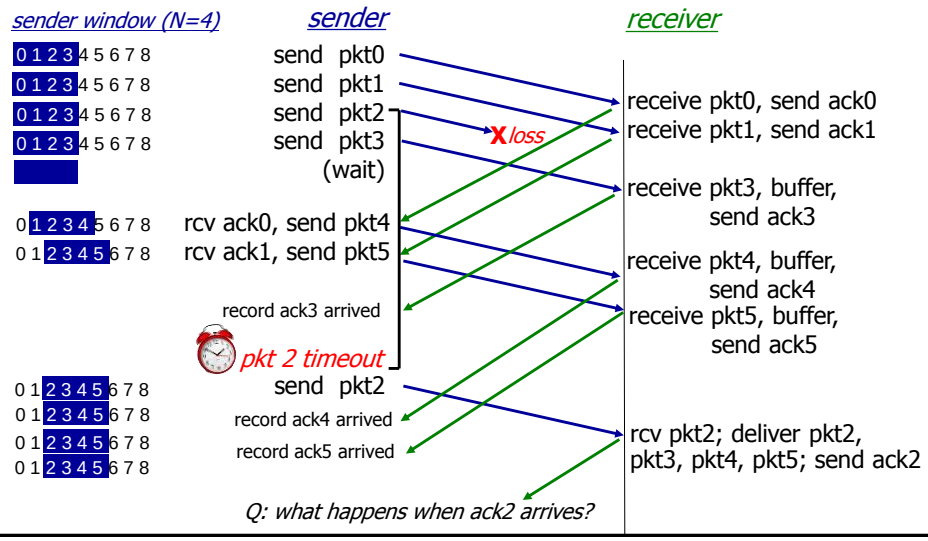
TCP - reliable data transfer (rdt)

GBN in action



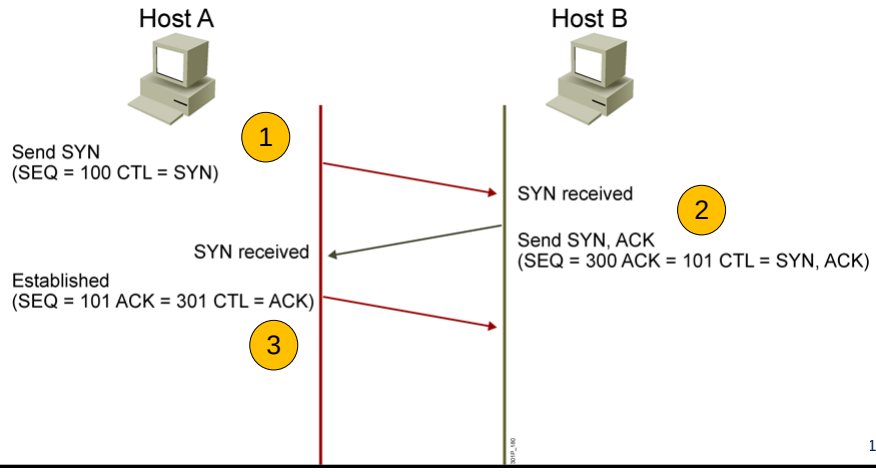
TCP - reliable data transfer (rdt)

Selective repeat in action



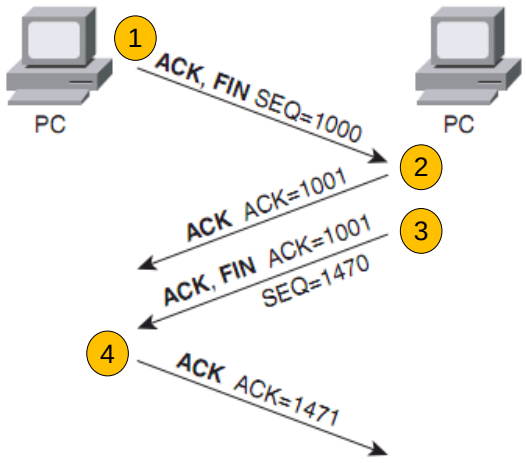
TCP Connection Management
Three-Way Handshake

- TCP Connection Establishing: agree on connection parameters

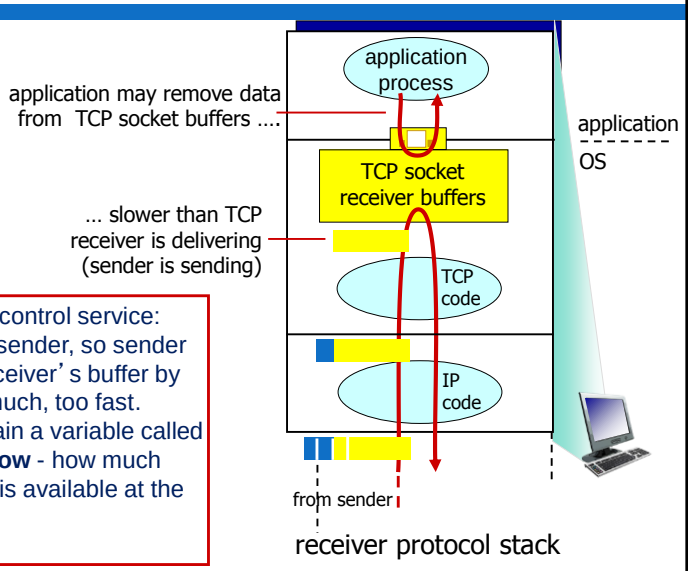


TCP Connection Management
Connection Termination

- TCP Connection Termination



TCP flow control

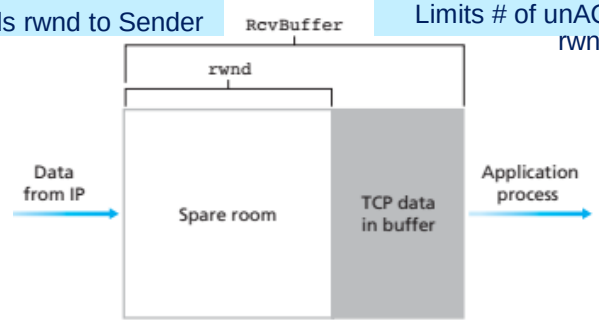


TCP provides a flow-control service:
- receiver controls sender, so sender won't overflow receiver's buffer by transmitting too much, too fast.
- the sender maintain a variable called the **receive window** - how much free buffer space is available at the receiver

Congestion Control

Receiver
Sends rwnd to Sender

Sender
Limits # of unACKed bytes to rwnd



- Unused Buffer Space: spare room in the buffer

```
rwnd = RcvBuffer – [LastByteRcvd – LastByteRead]
```

- TCP is not permitted to overflow the allocated buffer

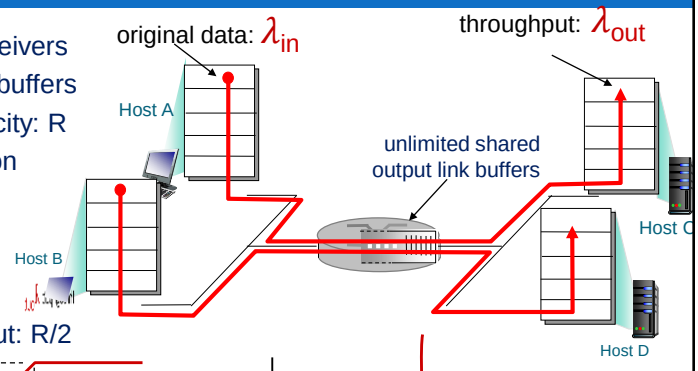
```
LastByteRcvd – LastByteRead ≤ RcvBuffer
```

TCP Congestion Control

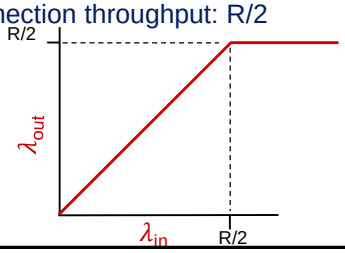
- Congestion informally: "too many sources sending too much data too fast for *network* to handle"
- different from flow control!
- manifestations:
  - lost packets (buffer overflow at routers)
  - long delays (queueing in router buffers)
- TCP Congestion Control
  - why congestion occurs in the first place
  - at the cost of congestion
    - more work (retrans) for given "goodput"
    - unneeded retransmissions: link carries multiple copies of pkt
      - decreasing goodput

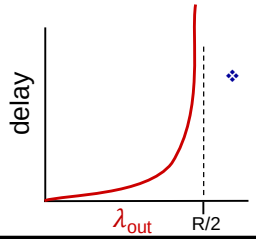
TCP Congestion Control

- 2 senders, 2 receivers
- 1 router, infinite buffers
- output link capacity: R
- no retransmission



- maximum per-connection throughput: R/2

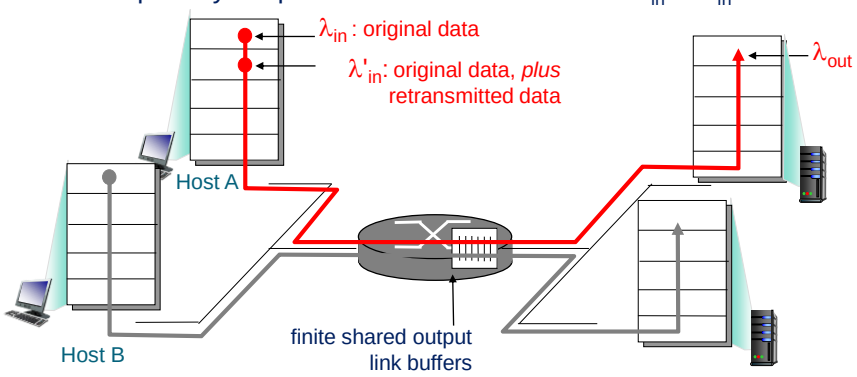




- large delays as arrival rate, λ_{in}, approaches capacity

TCP Congestion Control

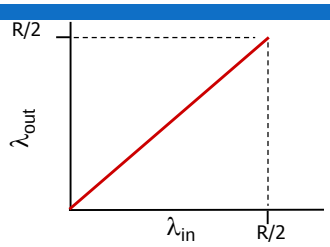
- one router, *finite* buffers
- sender retransmission of timed-out packet
  - application-layer input = application-layer output: $\lambda_{in} = \lambda_{out}$
  - transport-layer input includes *retransmissions* : $\lambda_{in} \geq \lambda_{in}$

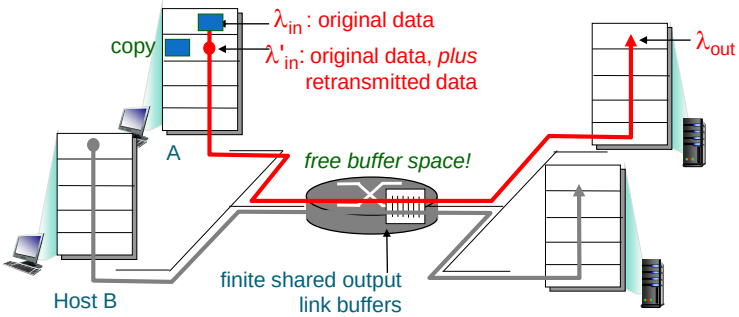


TCP Congestion Control

idealization: perfect knowledge

- sender sends only when router buffers available



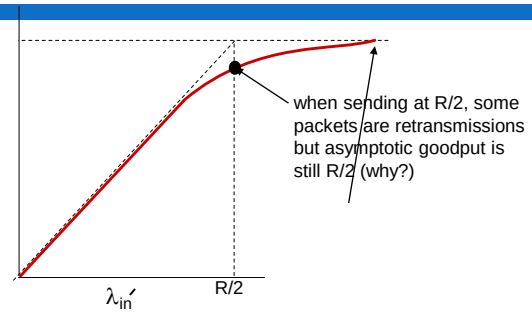


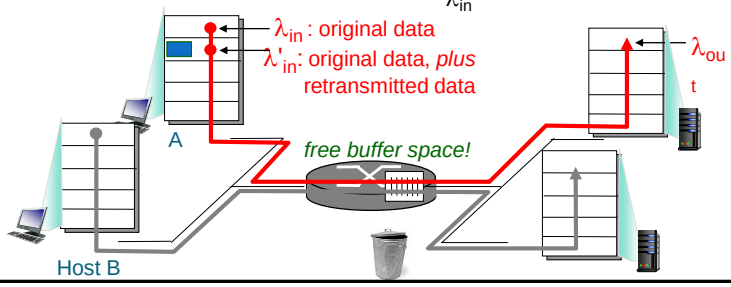
TCP Congestion Control

Idealization: known loss

packets can be lost, dropped at router due to full buffers

- sender only resends if packet *known* to be lost

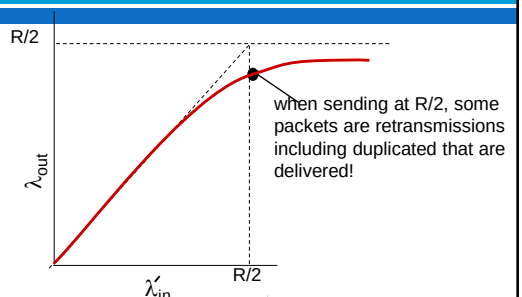


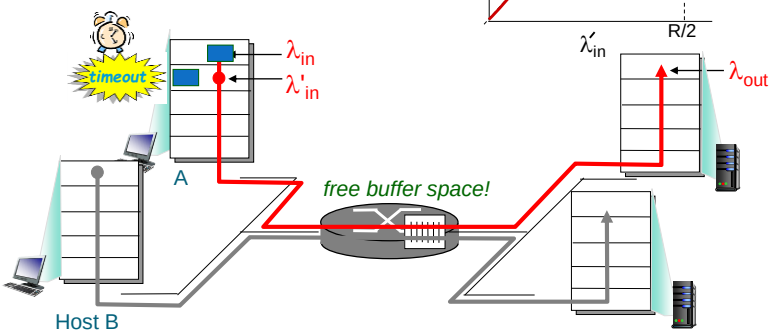


TCP Congestion Control

Realistic: duplicates

- packets can be lost, dropped at router due to full buffers
- sender times out prematurely, sending *two* copies, both of which are delivered





TCP & UDP Applications

Port Number	Protocol	Application
20	TCP	FTP Data
21	TCP	FTP Control
22	TCP	SSH
23	TCP	Telnet
25	TCP	SMTP
53	UDP,TCP	DNS
67,68	UDP	DHCP
69	UDP	TFTP
80	TCP	HTTP
110	TCP	POP3
161	UDP	SNMP
443	TCP	SSL
16,384-32,767	UDP	RTP-based Voice and Video

TCP/IP

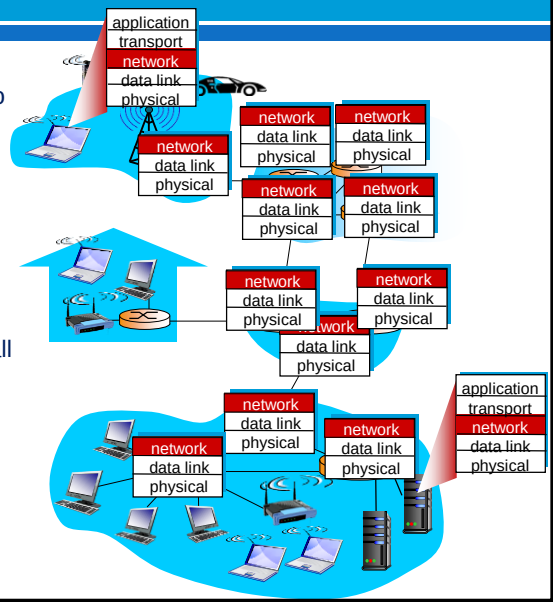
- TCP/IP Encapsulation
- Internet protocol stack
- Application layer
- Transport layer
- Network layer
- Link layer

Network layer

- Introduction
- Functions
- Internet protocol
- IP fragmentation and reassembly
- IP Address
- ICMP
- Tools: Traceroute, Ping, pingpath

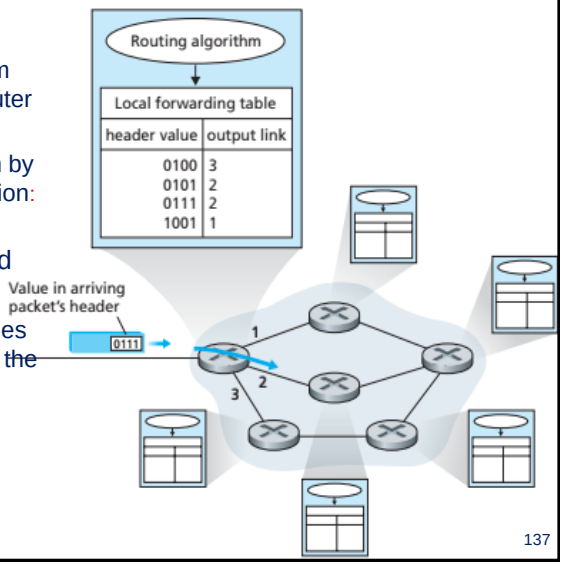
Network layer

- transport segment from sending to receiving host
- on sending side encapsulates segments into datagrams
- on receiving side, delivers segments to transport layer
- network layer protocols in *every* host, router
- router examines header fields in all IP datagrams passing through it



Network layer - functions

- *F*unctions
  - *forwarding:* move packets from router's input to appropriate router output
  - *routing:* determine route taken by packets from source to destination: *routing algorithms*
- Interplay between routing and forwarding
  - the routing algorithm determines the values that are inserted into the routers' forwarding tables



Network layer service models

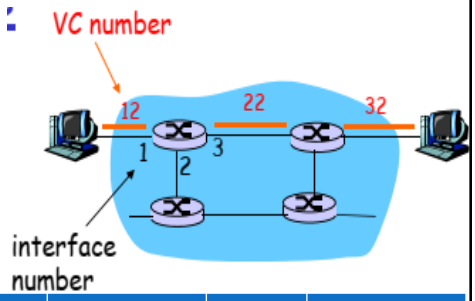
- **virtual-circuit (VC) network:** provide a connection service:
  - ATM and frame relay
- **datagram networks :** provide a connectionless service:
  - Internet
- VC: "source-to-dest path behaves much like telephone circuit"
  - call setup, teardown for each call before data can flow
  - each packet carries VC identifier (not destination host address)
  - every router on source-dest path maintains "state" for
  - link, router resources (bandwidth, buffers) may be allocated to VC (dedicated resources = predictable service)

Virtual circuits

- a VC consists of:
  - path from source to destination
  - VC numbers, one number for each link along path
  - entries in forwarding tables in routers along path
- packet belonging to VC carries VC number (rather than dest address)
- VC number can be changed **on each link**.
  - New VC number comes from forwarding table

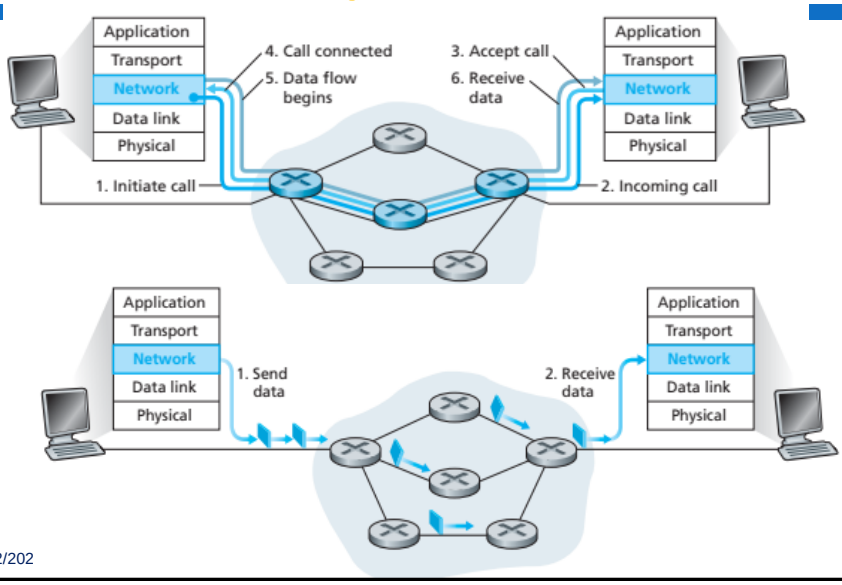
Virtual circuits

- A simple VC
  - Host A -> Host B
  - Path: A-R_1-R_2-B
  - VC nums: 12, 22, 32
- when a packet in this VC
  - leaves Host A, the value in the VC number field **is 12**;
  - leaves R1, the value is 22;
  - leaves R2, the value is 32
- the forwarding table in **R1** might l
  something
  like this



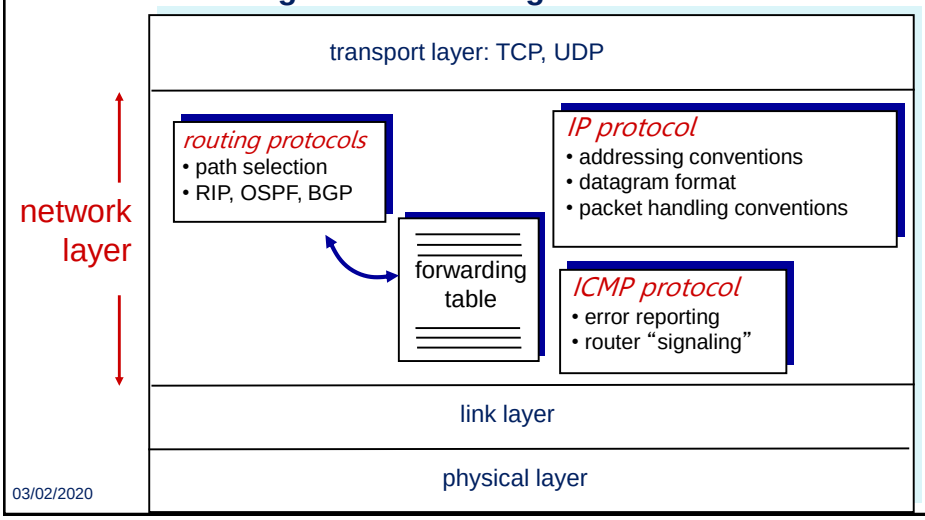
In Int	Incoming VC	Out int	Outgoing VC
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
…	…	…	…

virtual-circuit (VC) networks and datagram networks

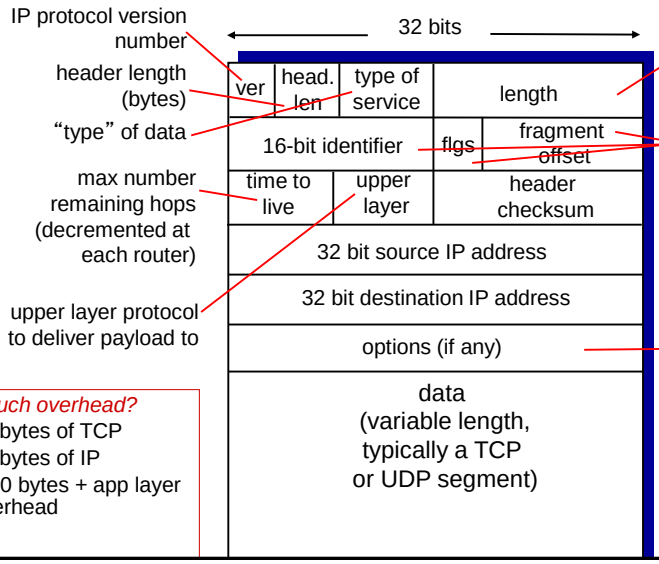


Internet Protocol (IP)

- **IP: Forwarding and Addressing in the Internet**



IP datagram format



how much overhead?

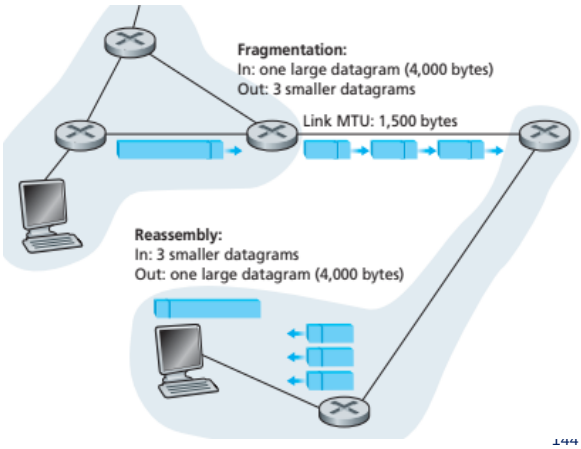
- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

143

IP fragmentation and reassembly

- For example,
  - Ethernet frames can carry up to 1,500 bytes of data,
  - some wide-area links can carry no more than 576 bytes.
- The maximum amount of data that a link-layer frame can carry:

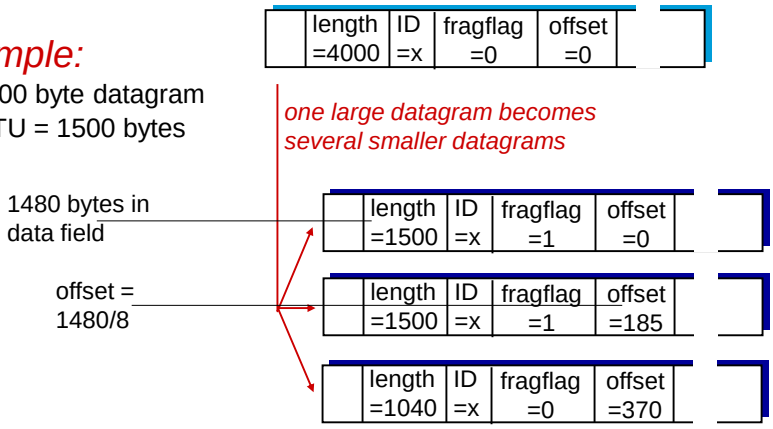
maximum transmission unit (MTU).



IP fragmentation, reassembly

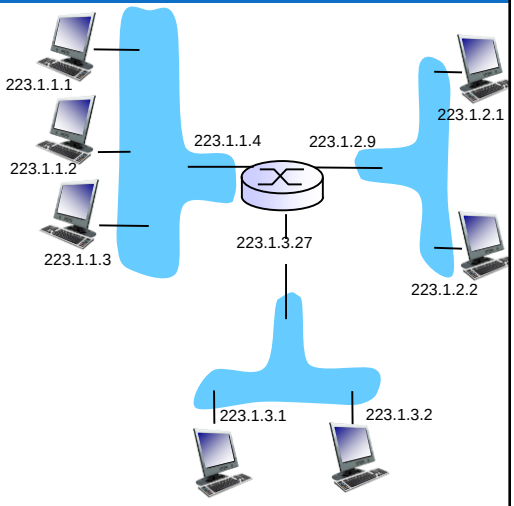
example:

- 4000 byte datagram
- MTU = 1500 bytes



IP addressing

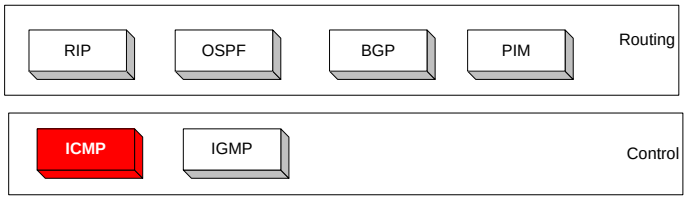
- *IP address:* 32-bit identifier for host, router *interface*
- *interface:* connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)
- *IP addresses associated with each interface*



146

Internet Control Message Protocol (ICMP)

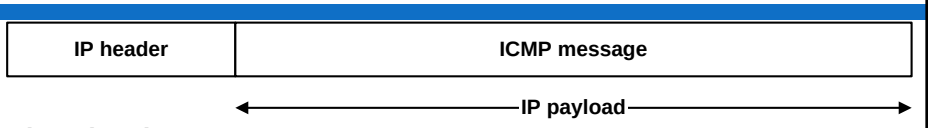
- The IP (Internet Protocol) relies on several other protocols to perform necessary control and routing functions:
  - Control functions (ICMP)
  - Multicast signaling (IGMP)
  - Setting up routing tables (RIP, OSPF, BGP, PIM, …)



- The **ICMP** is a helper protocol that supports IP with facility for
  - Error reporting
  - Simple queries

147

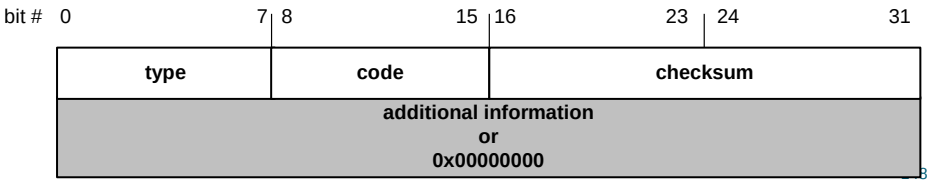
ICMP message format



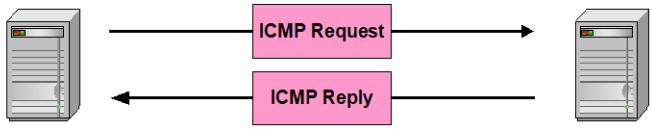
4 byte header:

- Type (1 byte): type of ICMP message
- Code (1 byte): subtype of ICMP message
- Checksum (2 bytes): similar to IP header checksum. Checksum is calculated over entire ICMP message

If there is no additional data, there are 4 bytes set to zero.
→ each ICMP messages is at least 8 bytes long



ICMP Query message



- **ICMP query:**
  - Request sent by host to a router or host
  - Reply sent back to querying host
- **Ex, ICMP queries:**

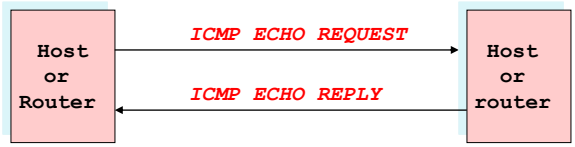
Type/Code:	Description
8/0	Echo Request
0/0	Echo Reply
13/0	Timestamp Request
14/0	Timestamp Reply
10/0	Router Solicitation
9/0	Router Advertisement

- The **ping** command uses Echo Request/ Echo Reply

149

Example of a Query:
Echo Request and Reply

- Ping's are handled directly by the kernel
- Each Ping is translated into an ICMP Echo Request
- The Ping'ed host responds with an ICMP Echo Reply

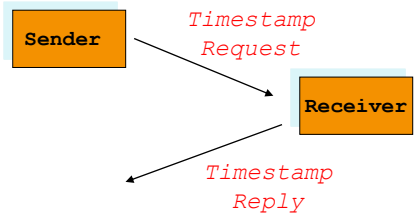



```
Command Prompt

C:\Users\test01>ping tuoitre.vn

Pinging tuoitre.vn [222.255.239.80] with 32 bytes of data:
Reply from 222.255.239.80: bytes=32 time=3ms TTL=56
Reply from 222.255.239.80: bytes=32 time=2ms TTL=56
Reply from 222.255.239.80: bytes=32 time=8ms TTL=56
Reply from 222.255.239.80: bytes=32 time=3ms TTL=56

Ping statistics for 222.255.239.80:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 8ms, Average = 4ms

C:\Users\test01>
```

150

Example of a Query: ICMP Timestamp

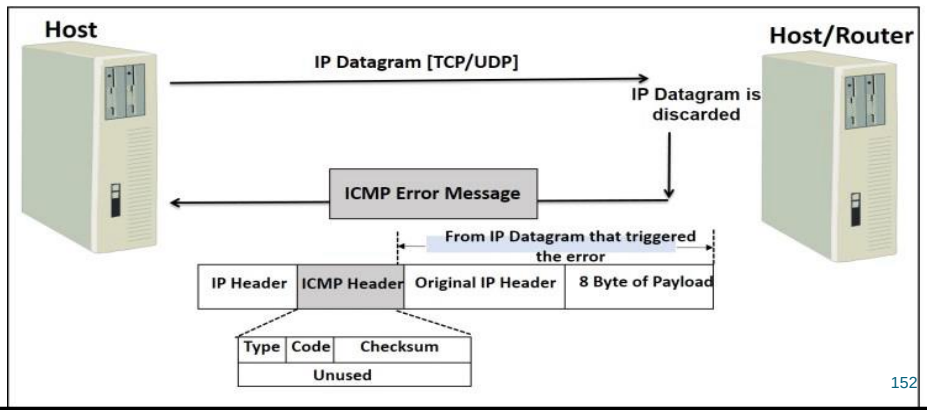
- A system (host or router) asks another system for the current time.
- Time is measured in milliseconds after midnight UTC (Universal Coordinated Time) of the current day
- Sender sends a request, receiver responds with reply



Type (= 17 or 18)	Code (=0)	Checksum
identifier		sequence number
32-bit sender timestamp		
32-bit receive timestamp		
32-bit transmit timestamp		

ICMP Error message

- report error conditions
- Typically sent when a datagram is discarded
- is often passed from ICMP to the application program
- include the complete IP header (the first 8 bytes of the payload UDP, TCP)



152

Frequent ICMP Error message

Type	Code	Description	
3	0–15	Destination unreachable	Notification that an IP datagram could not be forwarded and was dropped. The code field contains an explanation.
5	0–3	Redirect	Informs about an alternative route for the datagram and should result in a routing table update. The code field explains the reason for the route change.
11	0, 1	Time exceeded	Sent when the TTL field has reached zero (Code 0) or when there is a timeout for the reassembly of segments (Code 1)
12	0, 1	Parameter problem	Sent when the IP header is invalid (Code 0) or when an IP header option is missing (Code 1)

153

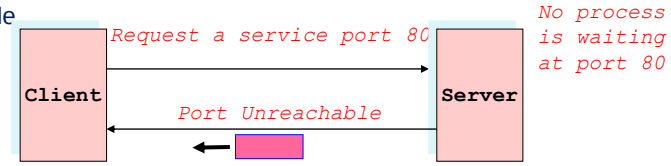
Some subtypes of the "Destination Unreachable"

Code	Description	Reason for Sending
0	Network Unreachable	No routing table entry is available for the destination network.
1	Host Unreachable	Destination host should be directly reachable, but does not respond to ARP Requests.
2	Protocol Unreachable	The protocol in the protocol field of the IP header is not supported at the destination.
3	Port Unreachable	The transport protocol at the destination host cannot pass the datagram to an application.
4	Fragmentation Needed and DF Bit Set	IP datagram must be fragmented, but the DF bit in the IP header is set.

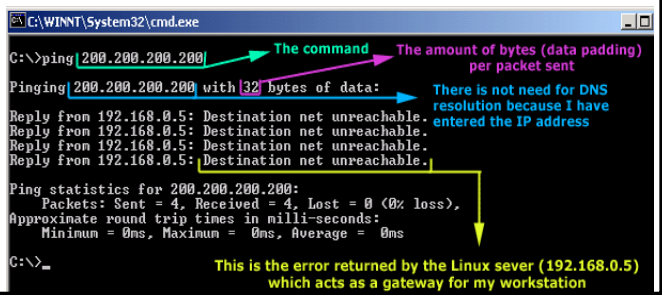
154

Example: ICMP Unreachable

- RFC 792: If, in the destination host, the IP module cannot deliver the datagram because the indicated protocol module or process port is not active, the destination host may send a destination unreachable message to the source host.
- Ex1: Port Unreachable



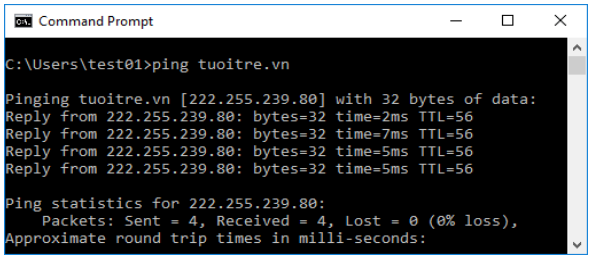
- Ex2: net Unreachable



Tools: ping, tracert

- Ping: Packet InterNet Groper (by Mike Muuss of the Army Research Laboratory in 12,1983)
  - check if a particular IP address is accessible or not.
  - checking if the computers on a local network are activeIt also measures round trip time and reports errors
  - Operate based on ICMP

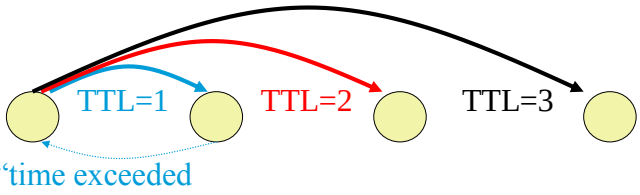
Ping <host> [-t/-n/-l] -> time, count, size



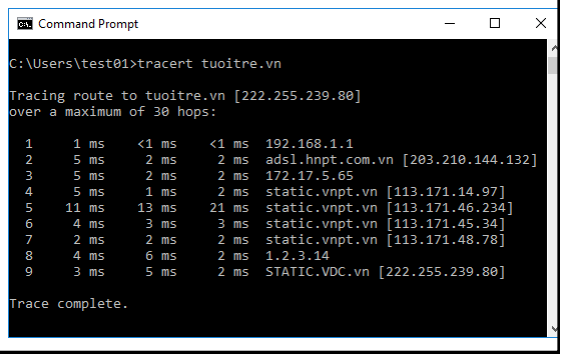
Tools: ping, tracert

- Tracert/Traceroute: (windows/Linux)
  - traces a packet from your computer to the host,
  - show the number of steps (hops) required to reach there and time
  - works by:
    - sending the packets of data with low survival time TTL - specifies how many steps (hops) can the packet survive before it is returned.
    - When a packet can't reach the final destination and expires at an intermediate step, that node returns the packet and identifies itself.
    - So, by increasing the TTL gradually,
  - identify the intermediate hosts.
    - If any of the hops come back with "Request timed out", it denotes network congestion and a reason for slow loading Web pages and dropped connections.

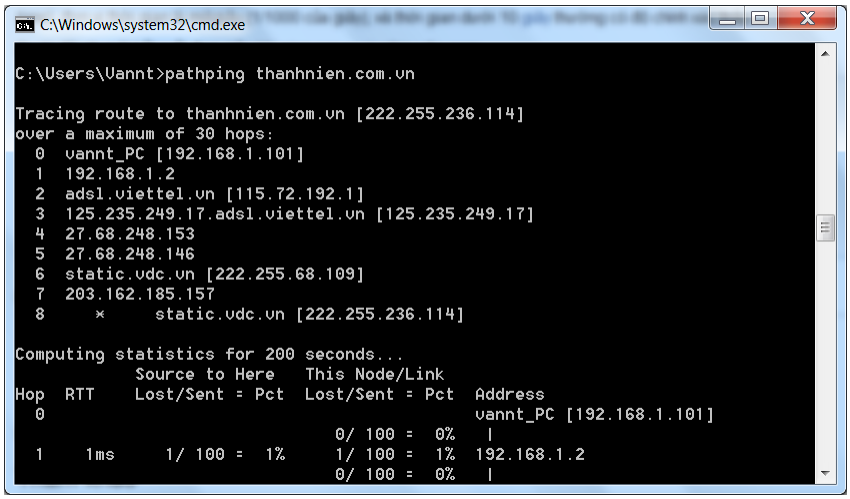
How Traceroute Works



- Send packets with increasing TTL values (+1)
  - Nodes along IP layer path decrement TTL
  - When TTL=0, nodes return "time exceeded" message and hop's IP is return.
  - When TTL<>0, TTL decrease and go next hop



Pathping, ex



TCP/IP

- TCP/IP Encapsulation
- Internet protocol stack
- Application layer
- Transport layer
- Network layer
- Link layer

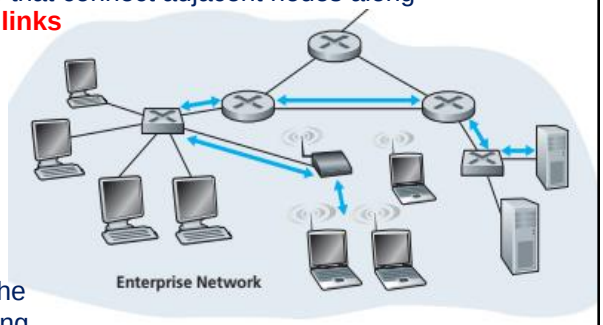
Link layer

- Introduction
- services
- Error detection and correction
- Transmission Modes
  - Simplex,
  - Half-Duplex and
  - Full-Duplex
- LAN addresses and ARP
- PPP

Link layer

Some terminology:

- hosts and routers are **nodes** (bridges and switches too)
- communication channels that connect adjacent nodes along communication path are **links**
  - wired links
  - wireless links
  - LANs
- PDU is a **frame**, encapsulates a network-layer datagram



- **Link-layer** protocol has the responsibility of transferring datagram from one node
- to adjacent node over a link
- different links:

e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link

Link Layer Services

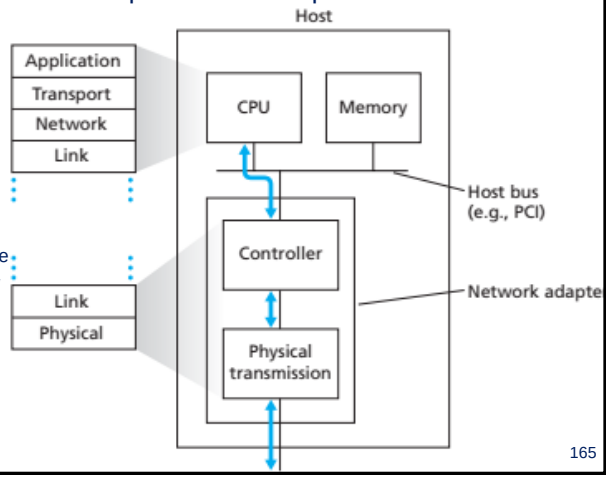
- Framing:
  - encapsulate datagram into frame, adding header, trailer
  - 'physical addresses' used in frame headers to identify source, destination
    - different from IP address!
- Link access
  - Media access control (MAC) protocol
  - Coordinate the frame transmissions of many nodes if multiple nodes share a medium
- Reliable delivery between adjacent nodes
  - we learned how to do this already (chapter 3)!
  - seldom used on low bit error link (fiber, some twisted pair)
  - Used on wireless links: high error rates
    - Correct an error locally at link level

Link Layer Services (more)

- *Flow Control:*
  - pacing between adjacent sending and receiving nodes
- *Error Detection*:
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- Error Correction:
  - receiver identifies *and corrects* bit error(s) without resorting to retransmission
- *Half-duplex and full-duplex*
  - with half duplex, nodes at both ends of link can transmit, but not at same time

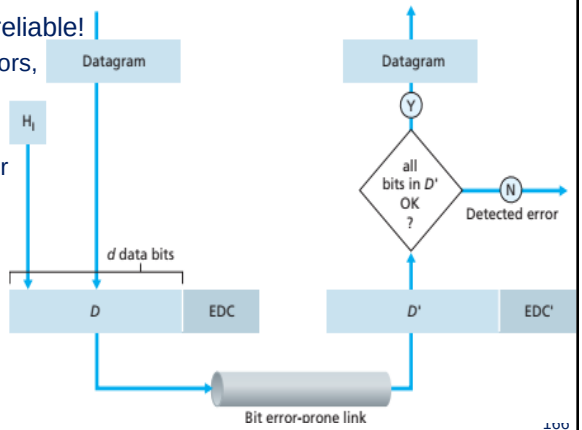
Link layer implementation

- the link layer is implemented in a **network adapter**
  - its relationship to other host components and to protocol stack functionality
- most of the link layer is implemented in hardware
- part of the link layer is implemented in software
- sending side:
  - encapsulates datagram in a frame
  - adds error checking bits, rdt, flow control, etc.
- receiving side
  - looks for errors, rdt, flow control,
  - extracts datagram, passes to receiving node



Error-Detection and -Correction Techniques

- EDC= Error Detection and Correction bits (redundancy)
- D = Data protected by error checking, may include header fields
- Error detection not 100% reliable!
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction

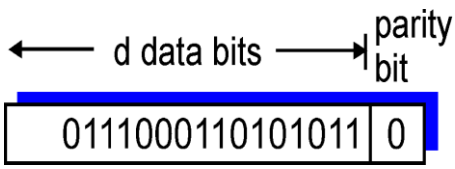


Error-Detection and -Correction Techniques

- Techniques for Error Detection
  - Parity checks
  - Checksumming methods
  - Cyclic redundancy checks

Parity Checks

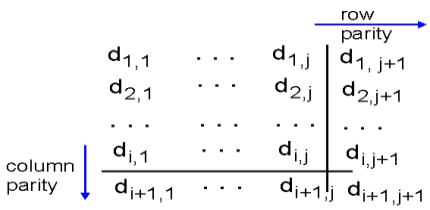
Single Bit Parity: **Detect single bit errors**



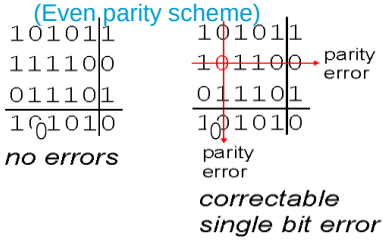
- Even parity scheme: choose the value of the parity bit such that the total number of 1s in the d+1 bits is even
- Odd parity scheme: choose the value of the parity bit such that the total number of 1s in the d+1 bits is odd

Parity Checks (Cont.)

Two Dimensional Bit Parity: **Detect *and correct* single bit errors**



(Even parity scheme)



Checksumming Methods

Goal: detect "errors" (e.g., flipped bits) in transmitted segment (note: used at transport layer *only*)

Internet checksum:

Sender:

- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into segment header

Receiver:

- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected. *But maybe errors nonetheless?* More later ….

- Checksum is easy and fast to compute
- Typically used in software implemented protocols (e.g. ,TCP and UDP)

170

Cyclic Redundancy Check

- view data bits, D, as a binary number
- choose r+1 bit pattern (generator), G (both sender and receiver know G)
- sender chooses r CRC bits, R, such that
  - <D,R> exactly divisible by G (modulo 2)
- receiver knows G, divides <D,R> by G.
  - If non-zero remainder: error detected!
  - can detect all burst errors less than r+1 bits
- widely used in practice (ATM, HDLC)

$D * 2^r$ XOR R — *mathematical formula*

Left shifts r bits

CRC Example

Want to find R such that:

$D \cdot 2^r$ XOR R = nG

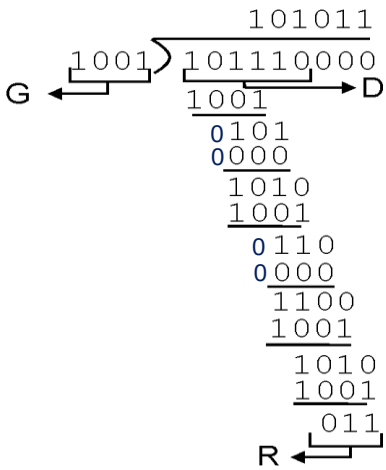
XOR R to the right of both sides :

$D \cdot 2^r$ = nG XOR R

equivalently:

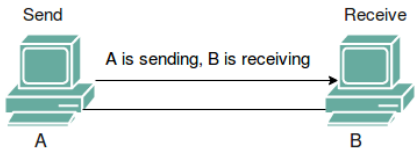
if we divide $D \cdot 2^r$ by G, the remainder is R

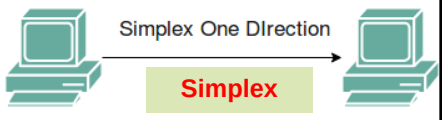
$$R = \text{remainder}\left[\frac{D \cdot 2^r}{G}\right]$$

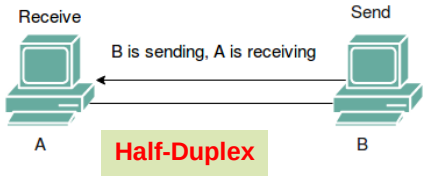


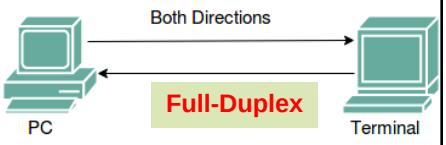
Transmission mode

- Transferring of data between two devices









Transmission mode

Simplex	Half Duplex	Full Duplex
Unidirectional	Two-directional, one at a time	Two-directional, simultaneously
Sender can only send data	Sender can send and receive data, but one a time	Sender can send and receive data simultaneously
Worst performing mode of transmission	Better than Simplex	Best performing mode of transmission
Keyboard and monitor	Walkie-talkie	Telephone

Link layer

- Introduction
- services
- Error detection and correction
- Transmission Modes
  - Simplex,
  - Half-Duplex and
  - Full-Duplex
- LAN addresses
- ARP
- PPP

LAN Addresses and ARP

32-bit IP address:

- *network-layer* address
- used to get datagram to destination IP network (recall IP network definition)

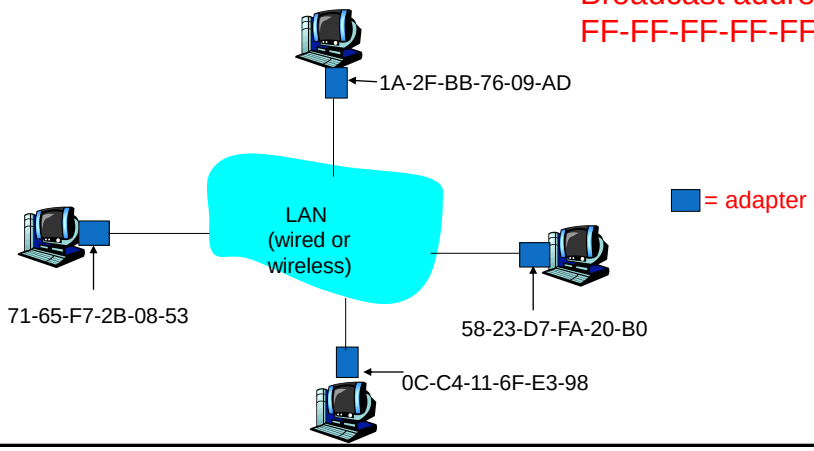
LAN (or MAC or physical or Ethernet) address:

- used to get datagram from one interface to another physically-connected interface (same network)
- 48 bit MAC address (for most LANs) burned in the adapter ROM

LAN Addresses and ARP

- Each adapter on LAN has unique LAN address
- Six bytes
- Expressed in hexadecimal notation

Broadcast address:
FF-FF-FF-FF-FF-FF



5a-177

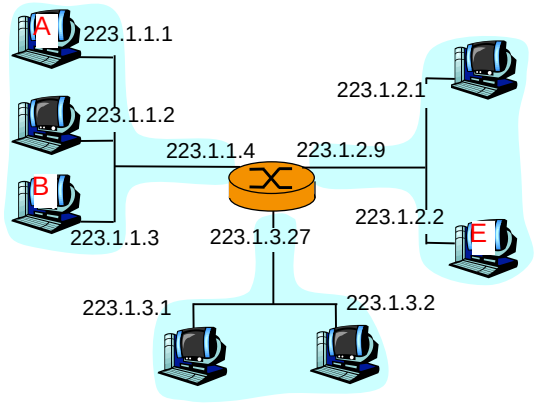
LAN Address (more)

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:
  - (a) MAC address: like Social Security Number
  - (b) IP address: like postal address
- MAC flat address => portability
  - MAC address of an adapter card does not change when it is moved from one LAN to another
- IP hierarchical address NOT portable
  - depends on IP network to which node is attached

Recall earlier routing discussion

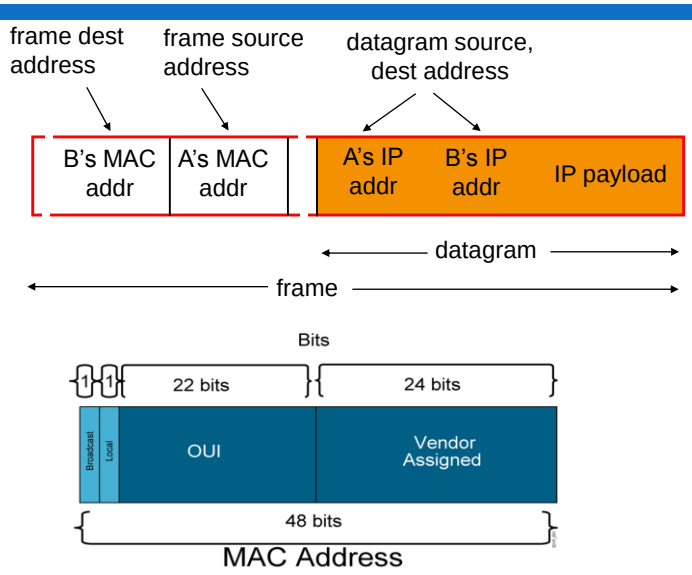
Starting at A, given IP datagram addressed to B:

- look up network address of B, find B on same network as A
- link layer send datagram to B inside link-layer frame



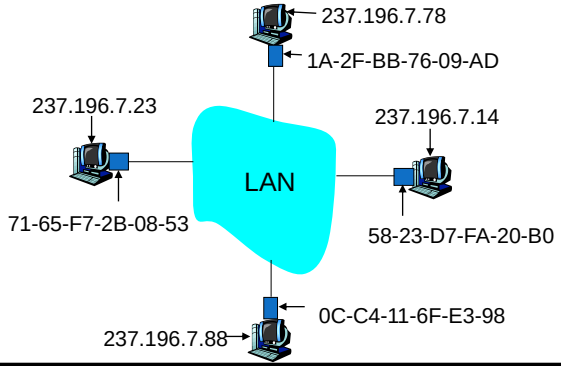
5a-179

MAC address



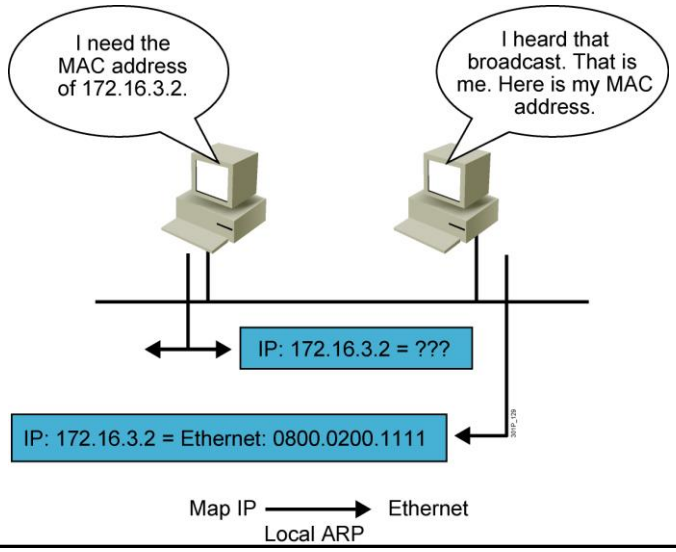
ARP: Address Resolution Protocol

- ARP: a communication protocol used for discovering the link layer address, such as a MAC address,
- It associated with a given internet layer address, typically an IPv4 address. This mapping is a critical function in the Internet protocol suite.
- ARP was defined in 1982 by RFC 826

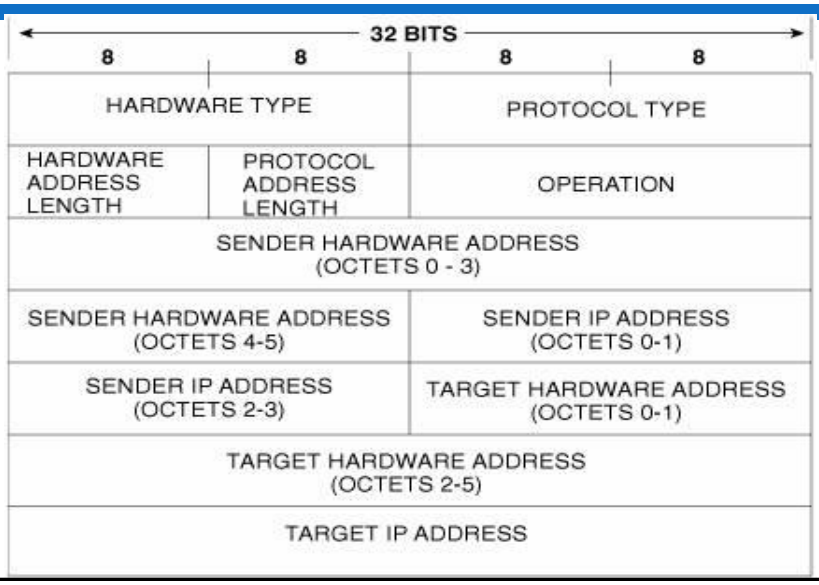


5a-181

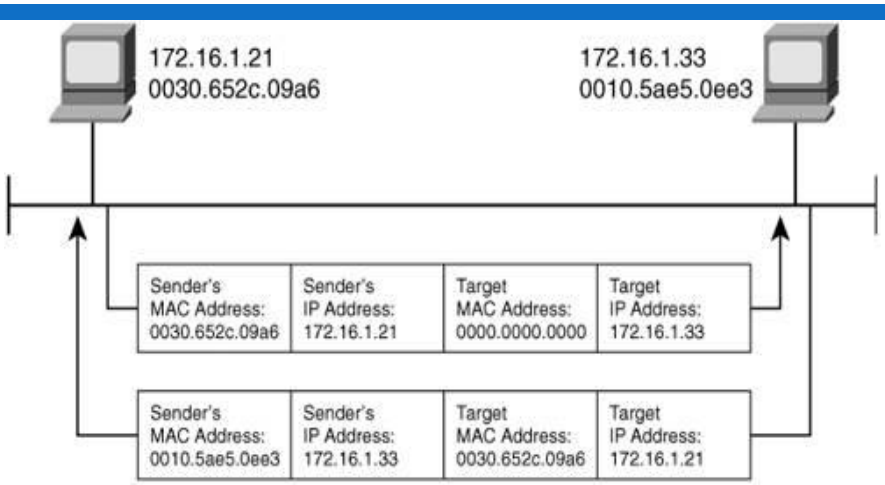
ARP



ARP packet format



How ARP works



An analyzer capture of the ARP Request

Ethernet II, Src: 00:30:65:2c:09:a6, Dst: ff:ff:ff:ff:ff:ff
- Destination: ff:ff:ff:ff:ff:ff (Broadcast)
- Source: 00:30:65:2c:09:a6 (AppleCom_2c:09:a6)
- Type: ARP (0x0806)

Address Resolution Protocol (request)
- Hardware type: Ethernet (0x0001)
- Protocol type: IP (0x0800)
- Hardware size: 6
- Protocol size: 4
- Opcode: request (0x0001)
- Sender MAC address: 00:30:65:2c:09:a6 (AppleCom_2c:09:a6)
  Sender IP address: 172.16.1.21 (172.16.1.21)
- Target MAC address: 00:00:00:00:00:00 (00:00:00_00:00:00) Target IP address: 172.16.1.33 (172.16.1.33)

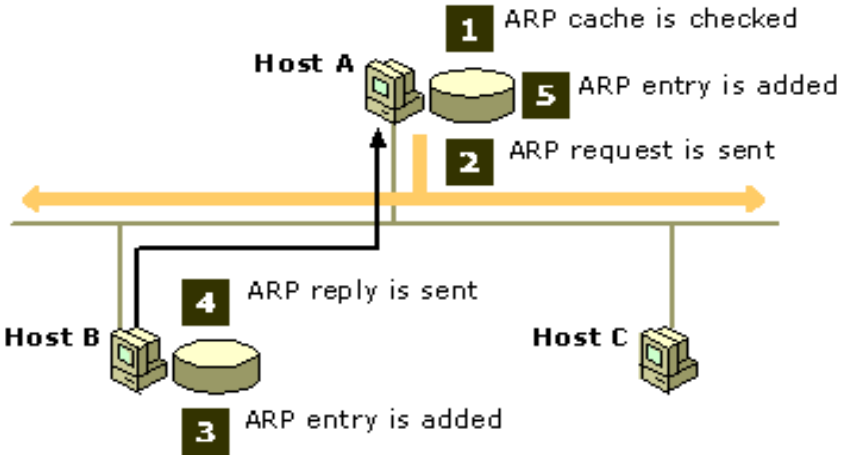
An analyzer capture of the ARP Reply

Ethernet II, Src: 00:10:5a:e5:0e:e3, Dst: 00:30:65:2c:09:a6
- Destination: 00:30:65:2c:09:a6 (AppleCom_2c:09:a6)
- Source: 00:10:5a:e5:0e:e3 (3com_e5:0e:e3)
- Type: ARP (0x0806)
- Trailer: 151515151515151515151515151515...

Address Resolution Protocol (reply)
- Hardware type: Ethernet (0x0001)
- Protocol type: IP (0x0800)
- Hardware size: 6
- Protocol size: 4
- Opcode: reply (0x0002)
- Sender MAC address: 00:10:5a:e5:0e:e3 (3com_e5:0e:e3)
- Sender IP address: 172.16.1.33 (172.16.1.33)
- Target MAC address: 00:30:65:2c:09:a6 (AppleCom_2c:09:a6)
- Target IP address: 172.16.1.21 (172.16.1.21)

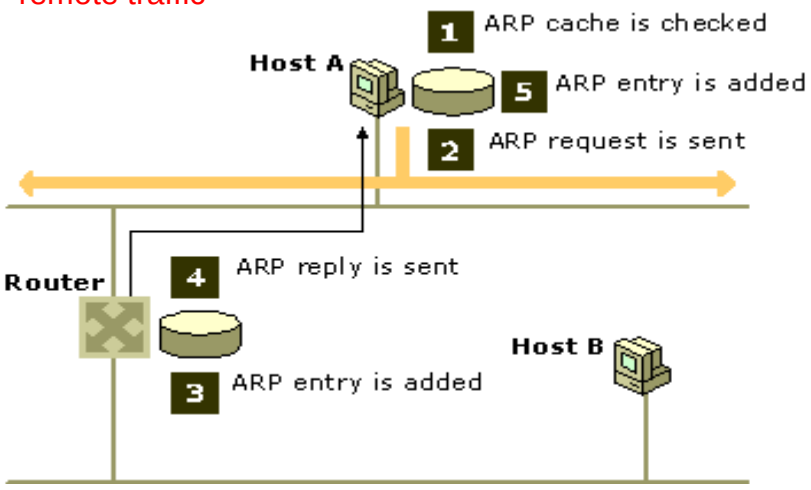
How ARP resolves

- How ARP resolves media access control addresses for local traffic

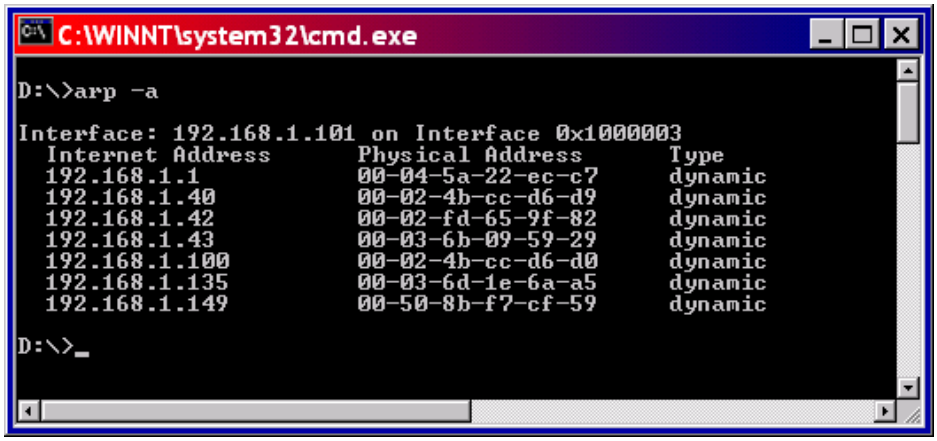


How ARP resolves

- How ARP resolves media access control addresses for remote traffic

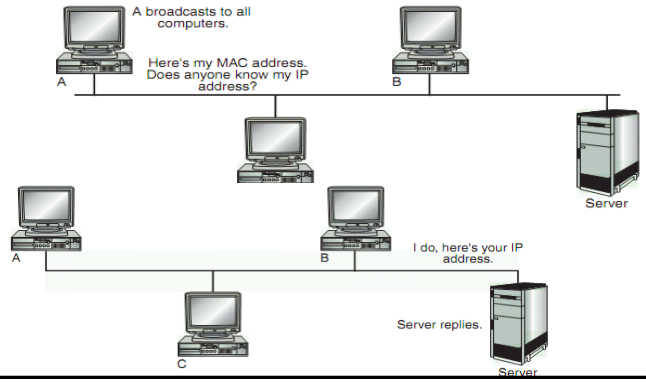


ARP Table



RARP

- **RARP – Reserve Address Resolution Protocol**
  - an obsolete computer networking protocol used by a client computer to request its Internet Protocol (IPv4) address from a computer network, when all it has available is its link layer or hardware address, such as a MAC address



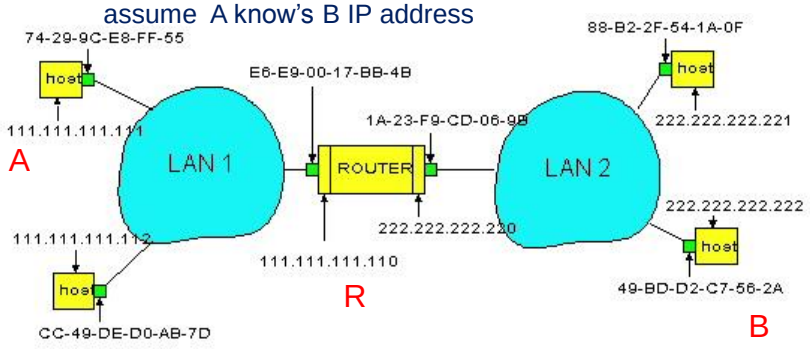
ARP protocol: Same LAN (network)

- A wants to send datagram to B, and B's MAC address not in A's ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - Dest MAC address = FF-FF-FF-FF-FF-FF
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC address (unicast)
- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed
- ARP is "plug-and-play":
  - nodes create their ARP tables without intervention from net administrator

5a-191

Routing to another LAN

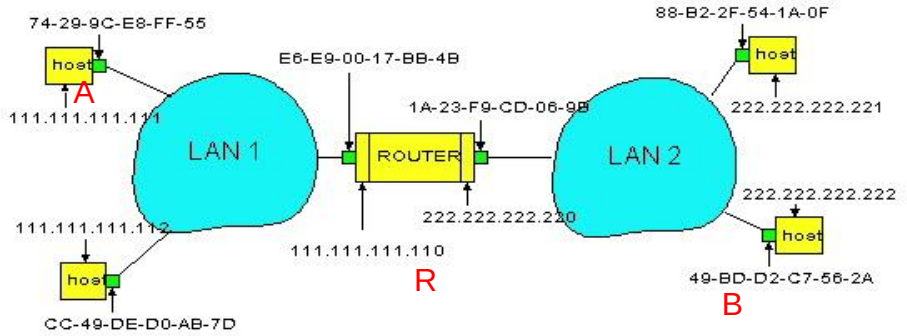
walkthrough: send datagram from A to B via R
assume A know's B IP address



- Two ARP tables in router R, one for each IP network (LAN)
- In routing table at source Host, find router 111.111.111.110
- In ARP table at source, find MAC address E6-E9-00-17-BB-4B, etc

5a-192

Routing to another LAN



- A creates datagram with source A, destination B
- A uses ARP to get R's MAC address for 111.111.111.110
- A creates link-layer frame with R's MAC address as destination, frame contains A-to-B IP datagram
- A's data link layer sends frame
- R's data link layer receives frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B's physical layer address, and creates frame containing A-to-B IP datagram sends to B

193

Link layer

- Introduction
- services
- Error detection and correction
- Transmission Modes
  - Simplex,
  - Half-Duplex and
  - Full-Duplex
- LAN addresses
- ARP
- PPP

Point to Point Data Link Control

- one sender, one receiver, one link: easier than broadcast link:
  - no Media Access Control
  - no need for explicit MAC addressing
  - e.g., dialup link, ISDN line
- popular point-to-point Data Link Control (DLC) protocols:
  - PPP (point-to-point protocol)
  - HDLC: High level data link control (Data link used to be considered “high layer” in protocol stack!)

PPP Design Requirements [RFC 1557]

- packet framing: encapsulation of network-layer datagram in data link frame
  - carry network layer data of any network layer protocol (not just IP) *at same time*
  - ability to demultiplex upwards
- bit transparency: must carry any bit pattern in the data field
- error detection (no correction)
- connection liveness: detect a link failure, signal link failure to network layer
- network layer address negotiation: endpoint can learn/configure each other’s network address

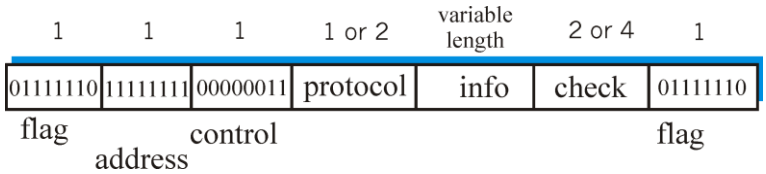
PPP non-requirements

- no error correction/recovery
- no flow control
- out of order delivery OK
- no need to support multipoint links (e.g., polling)

Error recovery, flow control, data re-ordering
all relegated to higher layers!

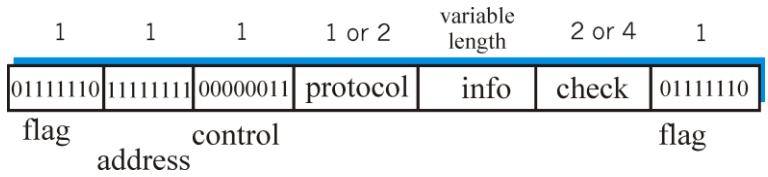
PPP Data Frame

- Flag: delimiter (framing)
- Address: does nothing (only one option)
- Control: does nothing; in the future possible multiple control fields
- Protocol: upper layer protocol to which frame delivered (eg, PPP-LCP, IP, IPCP, etc)



PPP Data Frame

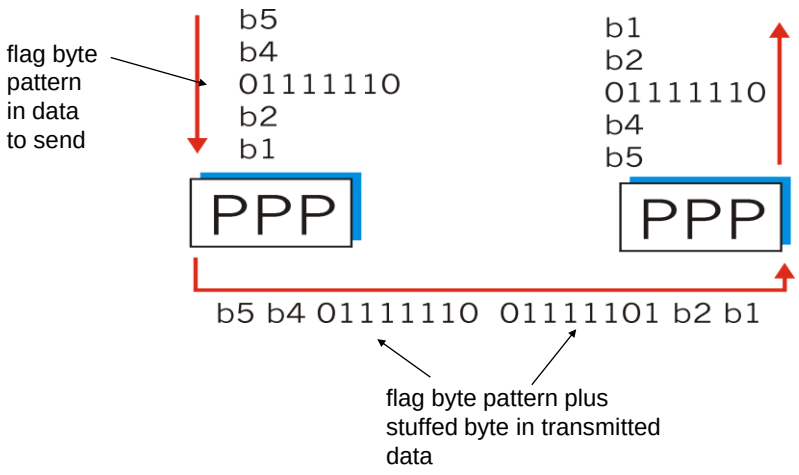
- info: upper layer data being carried, default maximum length = 1500 bytes
- check: cyclic redundancy check for error detection



Byte Stuffing

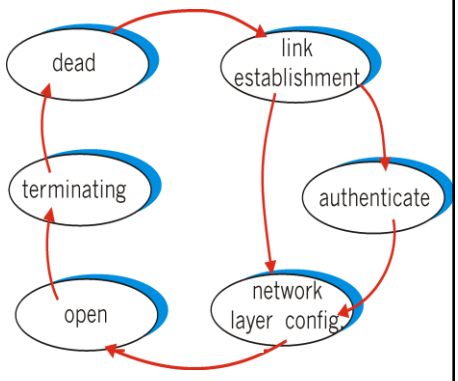
- "data transparency" requirement: data field must be allowed to include flag pattern <01111110>
  - Q: is received <01111110> data or flag?
- Sender:
  - adds ("stuffs") extra < 01111101> byte before each < 01111110> *data* byte
  - adds ("stuffs") extra < 01111101> byte before each < 01111101> *data* byte
- Receiver:
  - single 01111101 byte: discard 01111101
  - two 01111101 bytes in a row: discard first byte, continue data reception
  - single 01111110: flag byte

Byte Stuffing



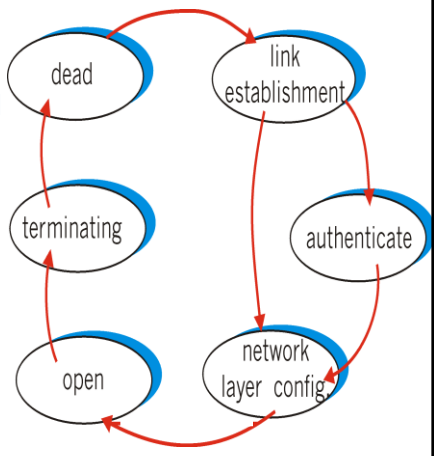
PPP Control Protocol

- Begins and ends in the dead state
- Enters link establishment state when the physical layer is present and ready to be used
- In the link establishment state, PPP link-control protocol (LCP) is used to negotiate link configuration options such as maximum frame size, authentication protocol (if any) to be used, etc.



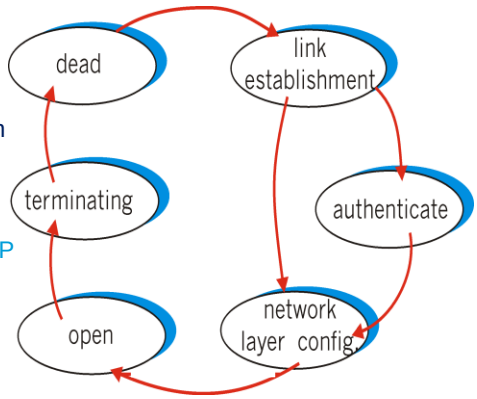
PPP Control Protocol

- Then, the end points enter the network layer configuration state to learn/configure network layer information using a network-control protocol
- The network-control protocol to be used depends on the specific network layer protocol
  - for IP: IP Control Protocol (IPCP) (protocol field: 8021) is used to configure/learn IP address
- Once the network layer has been configured, PPP enters the open state and may begin sending network layer datagrams



PPP Control Protocol

- The LCP echo-request frame and echo reply frame can be exchanged between Two PPP endpoints in order to check the status of the link
- To terminate the link, one end of the PPP link sends a terminate-request LCP frame and the other end replies with a terminate-ack LCP frame
- The link enter the dead state



Q/A

- History of computer network
- Computer network
- Network topology
- Network protocol
- Network Components
- Internet
- Packet-Switched Networks problems:
  - Delay, Loss, and Throughput in
- Protocol Layers and Their Service Models
- OSI model
- TCP/IP model