# Network Programming

Dr. Thaier Hayajneh

Computer Engineering Department

## Introduction

## Introduction

**Our goal:**

- get "feel" and terminology
- more depth, details are your responsibility
- approach:
  - use Internet as example

**Overview:**

- what's the Internet?
- what's a protocol?
- network edge; hosts, access net, physical media
- network core: packet/circuit switching, Internet structure
- performance: loss, delay, throughput
- protocol layers, service models
- history

## Introduction 1/2

- What is a network? Set of nodes connected by communication links
  - Components
    - ✓(Network edge) Computing devices (end hosts, PDAs, …) connected to the network
    - ✓(Network core) Routers/switches that move data through the network
    - ✓(Media) Physical links that carry information (fiber, copper, radio, and satellite)
    - ✓Applications that communicate with each other to provide services (Email, file transfer, and Web browsing).
- What is an internetwork? A network of networks (an internet)
  - Specific example is the ***Internet***

## Introduction 2/2

- Network physical topology
  - Geometric representation of the relationship of all the links and nodes to one another
  - Categories: *mesh*, *star*, *bus*, and *ring*
- Network Categories
  - Local-Area Network (LAN)
  - Metropolitan-Area Network (MAN)
  - Wide-Area Network (WAN)
  - Personal-Area Network (PAN)

# What's the Internet: "nuts and bolts" view

PC
server
wireless laptop
cellular handheld

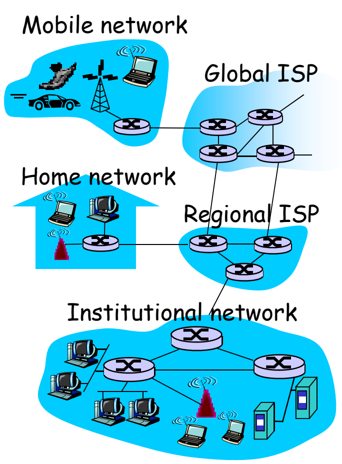
- millions of connected computing devices: *hosts = end systems*
  - running *network apps*

access points
wired links

- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate = *bandwidth*

router

- *routers:* forward packets (chunks of data)

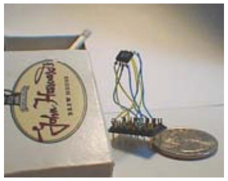


---

# "Cool" internet appliances

IP picture frame
http://www.ceiva.com/

Web-enabled toaster + weather forecaster



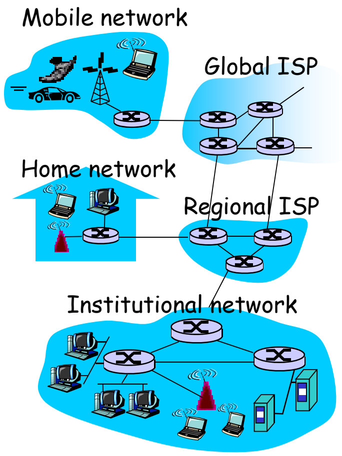
World's smallest web server
http://www-ccs.cs.umass.edu/~shri/iPic.html

Internet phones

---

# What's the Internet: "nuts and bolts" view

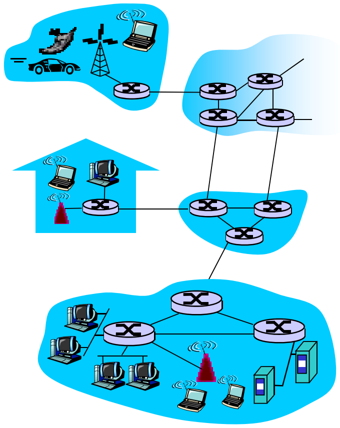
- *protocols* control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, Skype, Ethernet
- *Internet:* "network of networks"
  - loosely hierarchical
  - public Internet versus private intranet
- Internet standards
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



---

# What's the Internet: a service view

- communication *infrastructure* enables distributed applications:
  - Web, VoIP, email, games, e-commerce, file sharing
- communication services provided to apps:
  - reliable data delivery from source to destination
  - "best effort" (unreliable) data delivery



# What's a protocol?

**human protocols:**
- ❑ "what's the time?"
- ❑ "I have a question"
- ❑ introductions

… specific msgs sent

… specific actions taken when msgs received, or other events

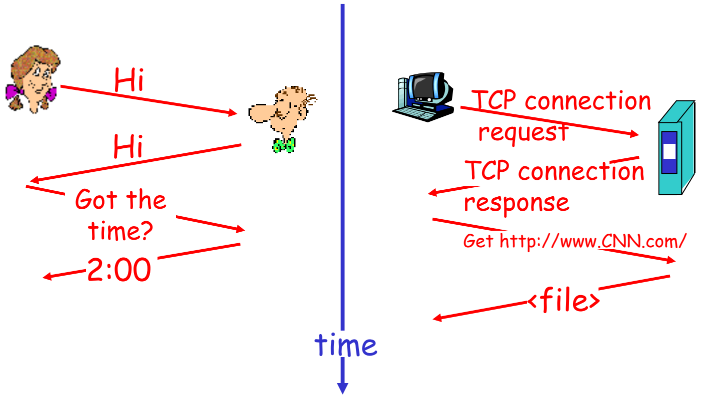
**network protocols:**
- ❑ machines rather than humans
- ❑ all communication activity in Internet governed by protocols

*protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt*

---

# What's a protocol?

a human protocol and a computer network protocol:



Q: Other human protocols?

---

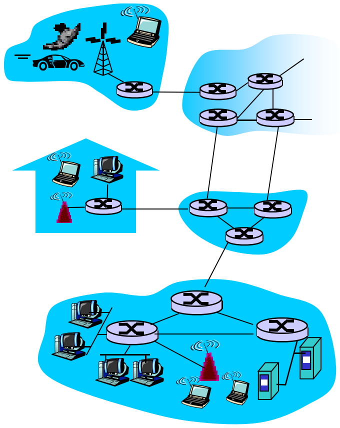
# Network Protocols

- Applications to communicate across a computer network
  - ➢Invent a *protocol* (an agreement how will communicate)
  - ➢Which application is expected to initiate communicate and when responses are expected
    - ✓Syntax: format of data
    - ✓Semantics: meaning of each section of bits (How it is interpreted and what action (s) to be taken)
    - ✓Timing: when data should be sent and how fast?
- Example: Web Server and Web client
- Other examples? Other modes of communication?

---

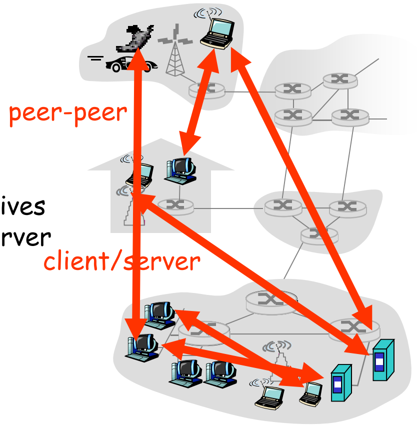
# A closer look at network structure:

- ❑ network edge: applications and hosts
- ❑ access networks, physical media: wired, wireless communication links
- ❑ network core:
  - ❖ interconnected routers
  - ❖ network of networks



# The network edge:

- end systems (hosts):
  - run application programs
  - e.g. Web, email
  - at “edge of network”
- client/server model
  - client host requests, receives service from always-on server
  - e.g. Web browser/server; email client/server
- peer-peer model:
  - minimal (or no) use of dedicated servers
  - e.g. Skype, BitTorrent



---

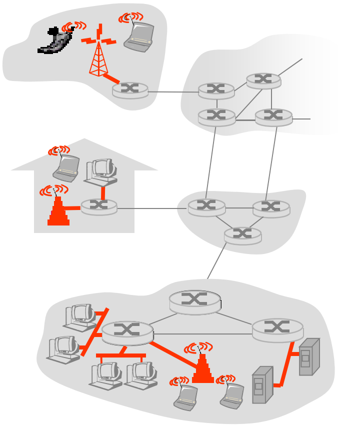
# Access networks and physical media

*Q: How to connect end systems to edge router?*

- residential access nets
- institutional access networks (school, company)
- mobile access networks

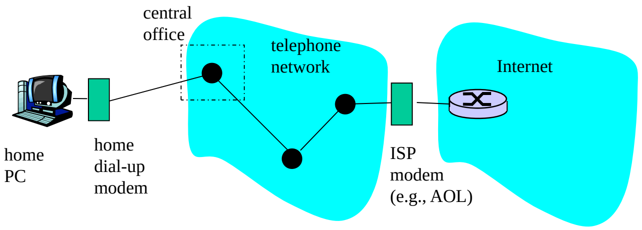
*Keep in mind:*

- bandwidth (bits per second) of access network?
- shared or dedicated?



---

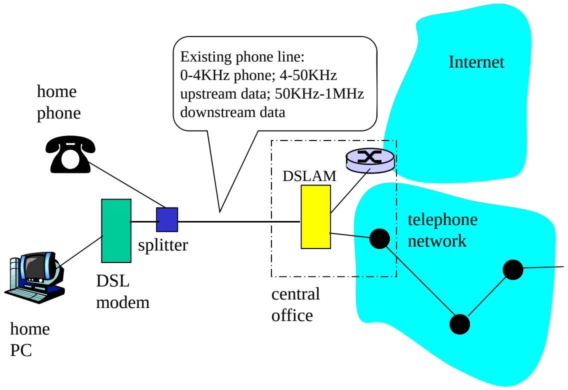
# Dial-up Modem



- Uses existing telephony infrastructure
  - Home is connected to central office
- up to 56Kbps direct access to router (often less)
- Can’t surf and phone at same time: not “always on”

---

# Digital Subscriber Line (DSL)



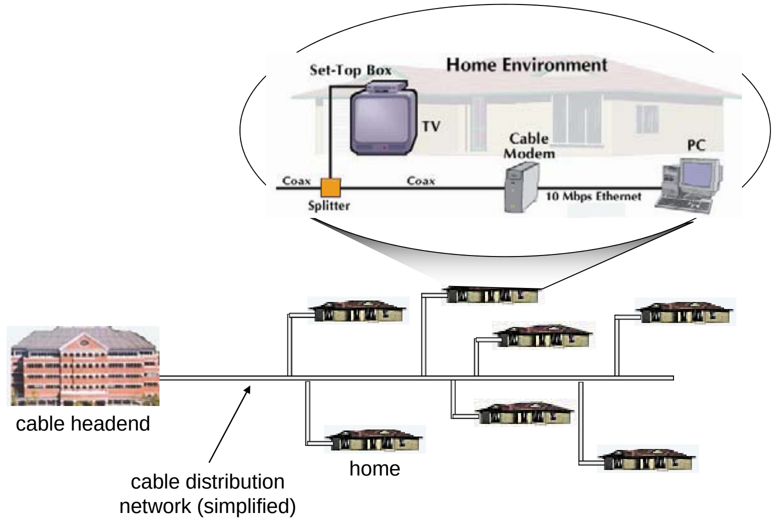
- Also uses existing telephone infrastruture
- up to 1 Mbps upstream (today typically < 256 kbps)
- up to 8 Mbps downstream (today typically < 1 Mbps)
- dedicated physical line to telephone central office

# Residential access: cable modems

- ❑ Does not use telephone infrastructure
  - ❖ Instead uses cable TV infrastructure
- ❑ HFC: hybrid fiber coax
  - ❖ asymmetric: up to 30Mbps downstream, 2 Mbps upstream
- ❑ network of cable and fiber attaches homes to ISP router
  - ❖ homes share access to router
  - ❖ unlike DSL, which has dedicated access

---

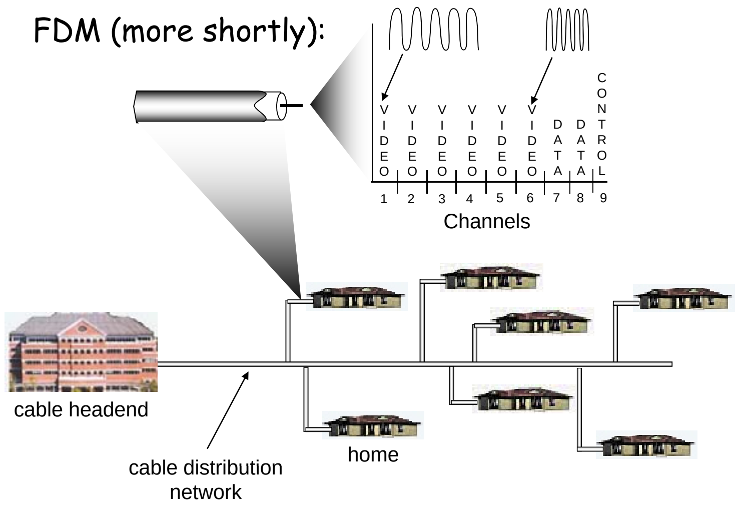
# Cable Network Architecture: Overview



---

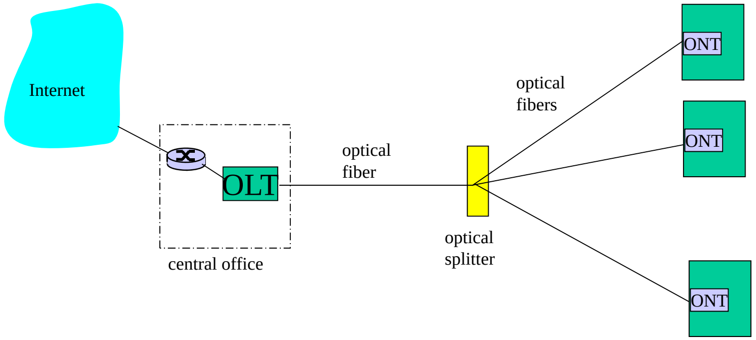
# Cable Network Architecture: Overview

FDM (more shortly):



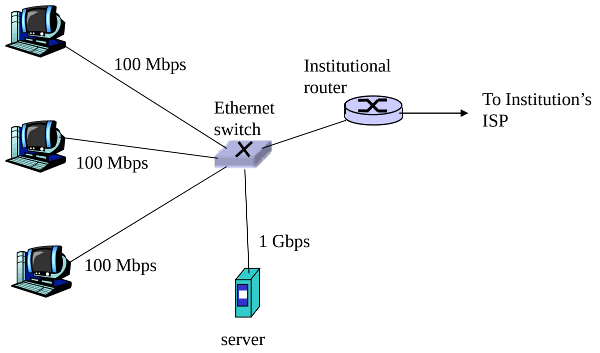
---

# Fiber to the Home



- ❑ Optical links from central office to the home
- ❑ Two competing optical technologies:
  - ❖ Passive Optical network (PON)
  - ❖ Active Optical Network (PAN)
- ❑ Much higher Internet rates; fiber also carries television and phone services

# Ethernet Internet access

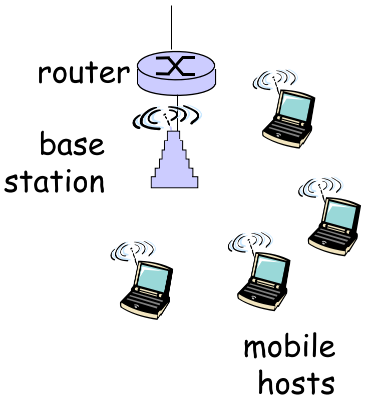


- Typically used in companies, universities, etc
- 10 Mbs, 100Mbps, 1Gbps, 10Gbps Ethernet
- Today, end systems typically connect into Ethernet switch

---

# Wireless access networks

- shared *wireless* access network connects end system to router
  - via base station aka "access point"
- wireless LANs:
  - 802.11b/g (WiFi): 11 or 54 Mbps
- wider-area wireless access
  - provided by telco operator
  - ~1Mbps over cellular system (EVDO, HSDPA)
  - next up (?): WiMAX (10's Mbps) over wide area

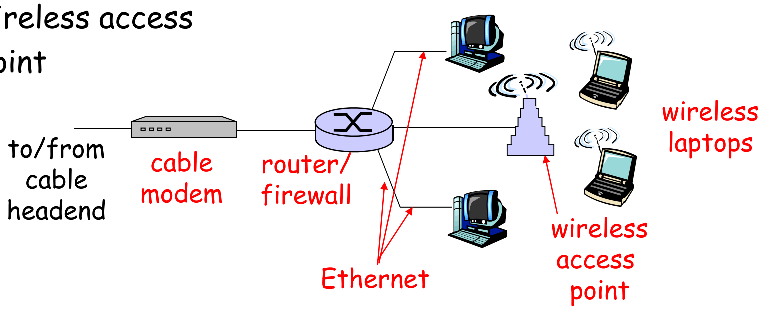


---

# Home networks

Typical home network components:

- DSL or cable modem
- router/firewall/NAT
- Ethernet
- wireless access point



---

# Physical Media

- Bit: propagates between transmitter/rcvr pairs
- physical link: what lies between transmitter & receiver
- guided media:
  - signals propagate in solid media: copper, fiber, coax
- unguided media:
  - signals propagate freely, e.g., radio

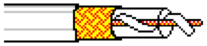
## Twisted Pair (TP)

- two insulated copper wires
  - Category 3: traditional phone wires, 10 Mbps Ethernet
  - Category 5: 100Mbps Ethernet

# Physical Media: coax, fiber

**Coaxial cable:**

- two concentric copper conductors
- bidirectional
- baseband:
  - single channel on cable
  - legacy Ethernet
- broadband:
  - multiple channels on cable
  - HFC



**Fiber optic cable:**

- glass fiber carrying light pulses, each pulse a bit
- high-speed operation:
  - high-speed point-to-point transmission (e.g., 10's-100's Gps)
- low error rate: repeaters spaced far apart ; immune to electromagnetic noise

---

# Physical media: radio

- signal carried in electromagnetic spectrum
- no physical "wire"
- bidirectional
- propagation environment effects:
  - reflection
  - obstruction by objects
  - interference

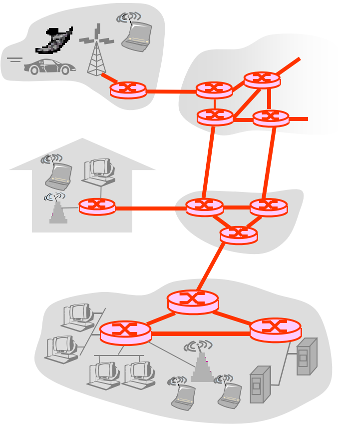
**Radio link types:**

- **terrestrial microwave**
  - e.g. up to 45 Mbps channels
- **LAN** (e.g., Wifi)
  - 11Mbps, 54 Mbps
- **wide-area** (e.g., cellular)
  - 3G cellular: ~ 1 Mbps
- **satellite**
  - Kbps to 45Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
  - geosynchronous versus low altitude

---

# The Network Core

- mesh of interconnected routers
- ***the* fundamental question:** how is data transferred through net?
  - **circuit switching:** dedicated circuit per call: telephone net
  - **packet-switching:** data sent thru net in discrete "chunks"

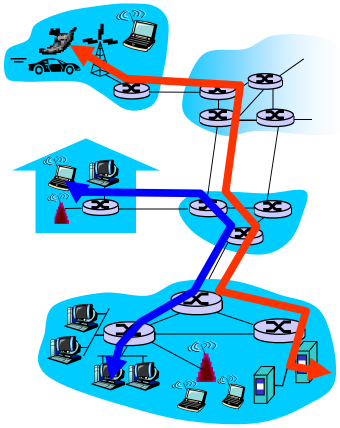


---

# Network Core: Circuit Switching

**End-end resources reserved for "call"**

- link bandwidth, switch capacity
- dedicated resources: no sharing
- circuit-like (guaranteed) performance
- call setup required



# Network Core: Circuit Switching

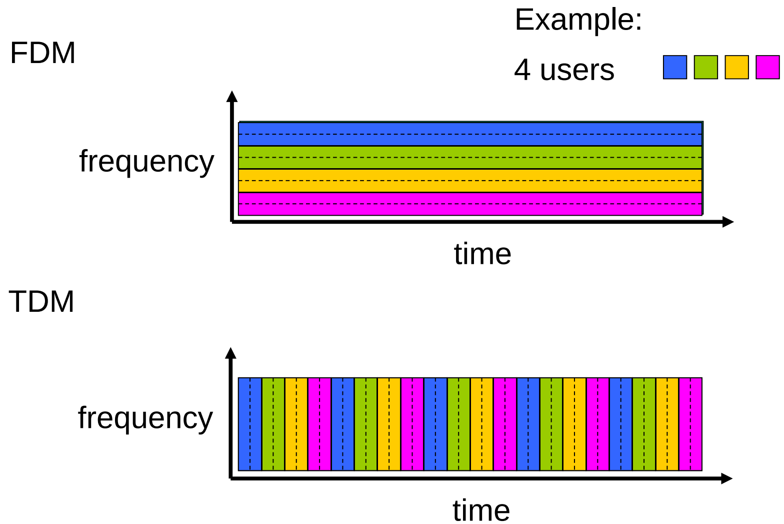
network resources (e.g., bandwidth) divided into "pieces"

- pieces allocated to calls
- resource piece *idle* if not used by owning call *(no sharing)*

- dividing link bandwidth into "pieces"
  - frequency division
  - time division

---

# Circuit Switching: FDM and TDM



---

# Data Transfer Through the Network

- Packet-Switching Data sent through network in discrete chunks
  - ➢Each end-to-end data stream divided into *packets*
    - ✓Users' packets *share* network resources
    - ✓Each packet uses full link bandwidth
  - ➢Resource contention
    - ✓aggregate resource demand can exceed amount available
    - ✓Congestion → packets queue, wait for link use
    - ✓store and forward → packets move one hop at a time
      - ❑transmit over link
      - ❑wait turn at next link

---

# Network Core: Packet Switching

each end-end data stream divided into *packets*

- user A, B packets *share* network resources
- each packet uses full link bandwidth
- resources used *as needed*

~~Bandwidth division into "pieces"~~
~~Dedicated allocation~~
~~Resource reservation~~

resource contention:

- aggregate resource demand can exceed amount available
- congestion: packets queue, wait for link use
- store and forward: packets move one hop at a time
  - Node receives complete packet before forwarding

# Data Transfer Through the Network

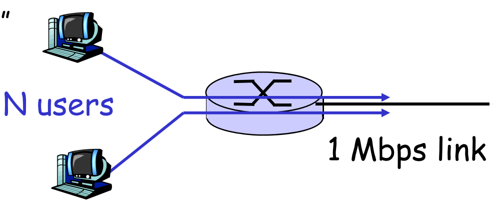
- Packet-Switching approaches
  - **datagram network**
    - ✓*destination address* determines next hop
    - ✓routes may change during session
    - ✓analogy: driving, asking directions
  - **virtual circuit network**
    - ✓Requires call setup
    - ✓each packet carries tag (virtual circuit ID), tag determines next hop
    - ✓*fixed path* determined at *call setup time*, remains fixed thru call
    - ✓routers maintain per-call state

---

# 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?

---

# Packet switching versus circuit switching

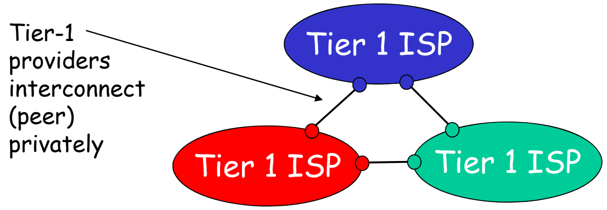
Is packet switching a "slam dunk winner?"

- great for bursty data
  - resource sharing
  - simpler, no call setup
- excessive congestion: packet delay and loss
  - protocols needed for reliable data transfer, congestion control
- Q: How to provide circuit-like behavior?
  - bandwidth guarantees needed for audio/video apps
  - still an unsolved problem

---

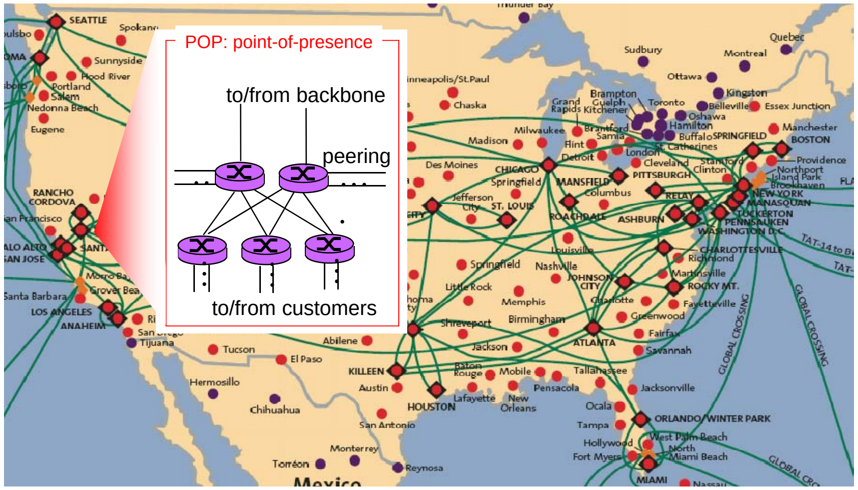
# Internet structure: network of networks

- roughly hierarchical
- at center: "tier-1" ISPs (e.g., Verizon, Sprint, AT&T, Cable and Wireless), national/international coverage
  - treat each other as equals



Tier-1 providers interconnect (peer) privately

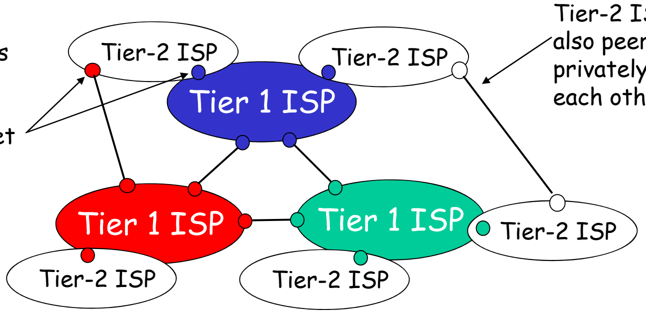
# Tier-1 ISP: e.g., Sprint



---

# Internet structure: network of networks

- "Tier-2" ISPs: smaller (often regional) ISPs
  - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs



Tier-2 ISP pays tier-1 ISP for connectivity to rest of Internet

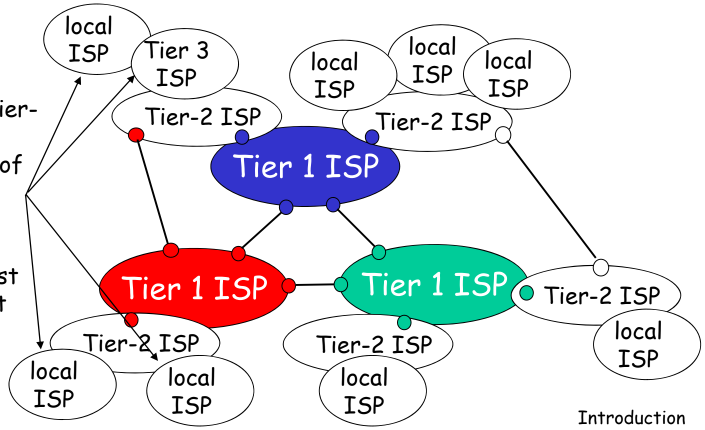
- tier-2 ISP is *customer* of tier-1 provider

Tier-2 ISPs also peer privately with each other.

---

# Internet structure: network of networks

- "Tier-3" ISPs and local ISPs
  - last hop ("access") network (closest to end systems)

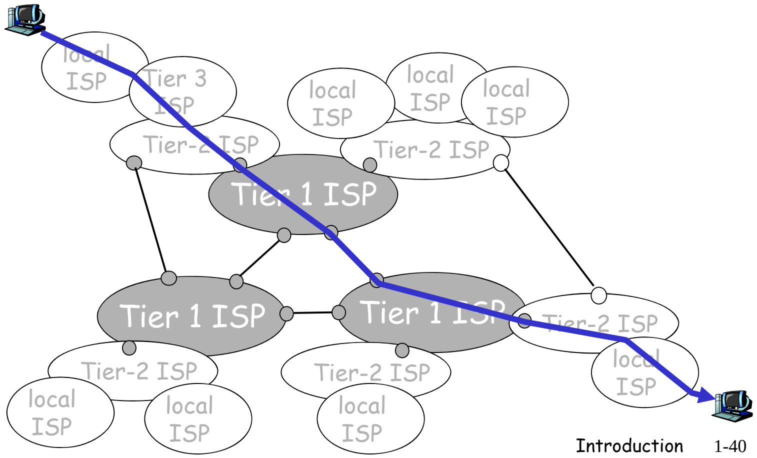


Local and tier-3 ISPs are *customers* of higher tier ISPs connecting them to rest of Internet

---

# Internet structure: network of networks

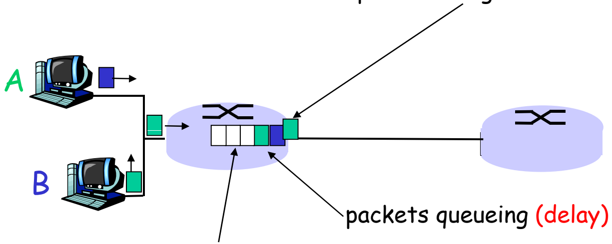
- a packet passes through many networks!



# How do loss and delay occur?

packets *queue* in router buffers

- ❑ packet arrival rate to link exceeds output link capacity
- ❑ packets queue, wait for turn



packet being transmitted (delay)

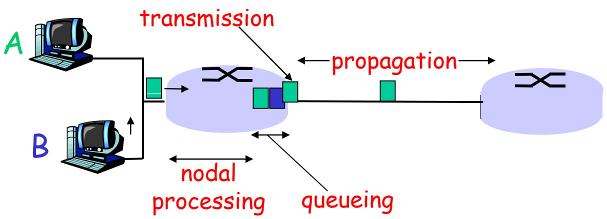
packets queueing (delay)

free (available) buffers: arriving packets dropped (loss) if no free buffers

---

# Four sources of packet delay

- ❑ 1. nodal processing:
  - ❖ check bit errors
  - ❖ determine output link
- ❑ 2. queueing
  - ❖ time waiting at output link for transmission
  - ❖ depends on congestion level of router



---

# Delay in packet-switched networks

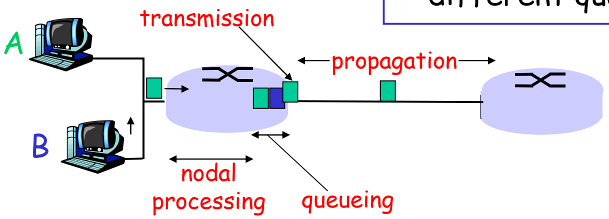
3. Transmission delay:

- ❑ R=link bandwidth (bps)
- ❑ L=packet length (bits)
- ❑ time to send bits into link = L/R

4. Propagation delay:

- ❑ d = length of physical link
- ❑ s = propagation speed in medium (~$2x10^8$ m/sec)
- ❑ propagation delay = d/s

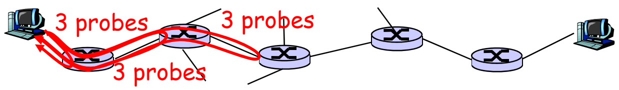
Note: s and R are *very* different quantities!



---

# “Real” Internet delays and routes

- ❑ What do “real” Internet delay & loss look like?
- ❑ Traceroute program: provides delay measurement from source to router along end-end Internet path towards destination. For all *i:*
  - ❖ sends three packets that will reach router *i* on path towards destination
  - ❖ router *i* will return packets to sender
  - ❖ sender times interval between transmission and reply.



# "Real" Internet delays and routes

traceroute: gaia.cs.umass.edu to www.eurecom.fr

Three delay measurements from gaia.cs.umass.edu to cs-gw.cs.umass.edu

```
1  cs-gw (128.119.240.254)  1 ms  1 ms  2 ms
2  border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)  1 ms  1 ms  2 ms
3  cht-vbns.gw.umass.edu (128.119.3.130)  6 ms 5 ms 5 ms
4  jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)  16 ms 11 ms 13 ms
5  jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)  21 ms 18 ms 18 ms
6  abilene-vbns.abilene.ucaid.edu (198.32.11.9)  22 ms  18 ms  22 ms
7  nycm-wash.abilene.ucaid.edu (198.32.8.46)  22 ms  22 ms  22 ms
8  62.40.103.253 (62.40.103.253)  104 ms 109 ms 106 ms
9  de2-1.de1.de.geant.net (62.40.96.129)  109 ms 102 ms 104 ms
10  de.fr1.fr.geant.net (62.40.96.50)  113 ms 121 ms 114 ms
11  renater-gw.fr1.fr.geant.net (62.40.103.54)  112 ms  114 ms  112 ms
12  nio-n2.cssi.renater.fr (193.51.206.13)  111 ms  114 ms  116 ms
13  nice.cssi.renater.fr (195.220.98.102)  123 ms  125 ms  124 ms
14  r3t2-nice.cssi.renater.fr (195.220.98.110)  126 ms  126 ms  124 ms
15  eurecom-valbonne.r3t2.ft.net (193.48.50.54)  135 ms  128 ms  133 ms
16  194.214.211.25 (194.214.211.25)  126 ms  128 ms  126 ms
17  * * *
18  * * *
19  fantasia.eurecom.fr (193.55.113.142)  132 ms  128 ms  136 ms
```

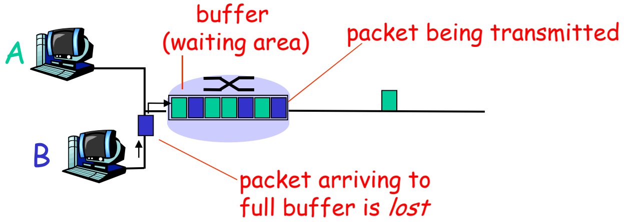
trans-oceanic link

* means no response (probe lost, router not replying)

---

# Packet loss

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



buffer (waiting area)

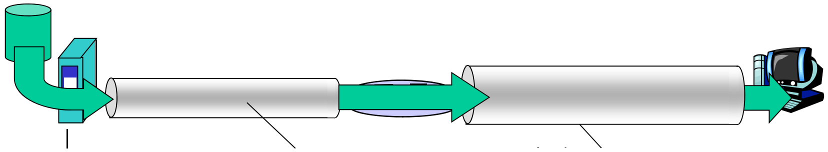
packet being transmitted

packet arriving to full buffer is *lost*

---

# Throughput

- ❑ *throughput:* rate (bits/time unit) at which bits transferred between sender/receiver
  - ❖ *instantaneous:* rate at given point in time
  - ❖ *average:* rate over longer period of time



server sends bits (fluid) into pipe

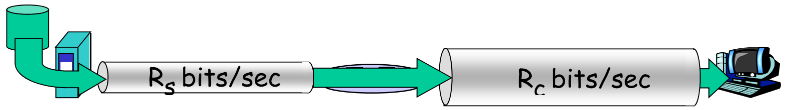
pipe that can carry fluid at rate $R_s$ bits/sec)

pipe that can carry fluid at rate $R_c$ bits/sec)

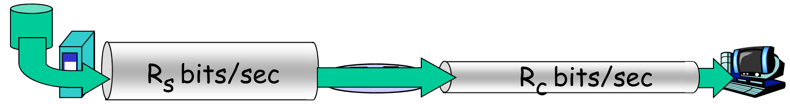
---

# Throughput (more)

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



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

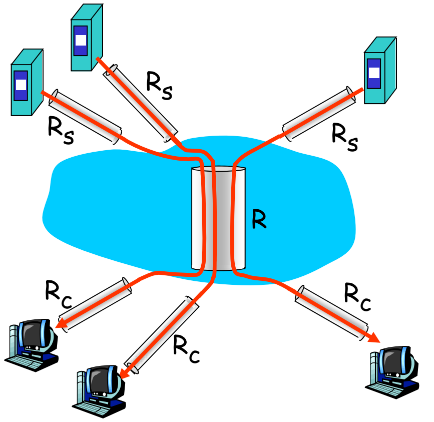


*bottleneck link*

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

# Throughput: Internet scenario

- per-connection end-end throughput: $\min(R_c, R_s, R/10)$
- in practice: $R_c$ or $R_s$ is often bottleneck



10 connections (fairly) share backbone bottleneck link R bits/sec

---

# Protocol "Layers"

Networks are complex!

- many "pieces":
  - hosts
  - routers
  - links of various media
  - applications
  - protocols
  - hardware, software

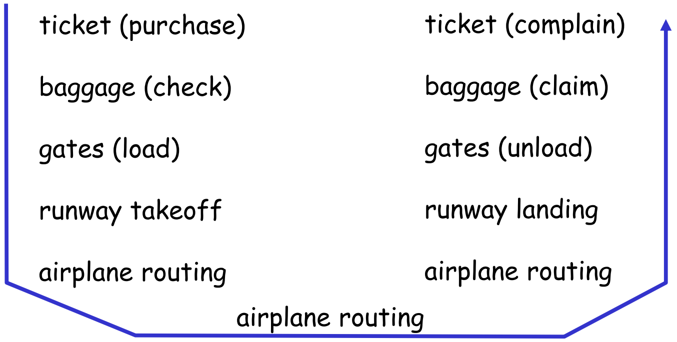
Question:

Is there any hope of *organizing* structure of network?

Or at least our discussion of networks?

---

# Organization of air travel



ticket (purchase)
baggage (check)
gates (load)
runway takeoff
airplane routing

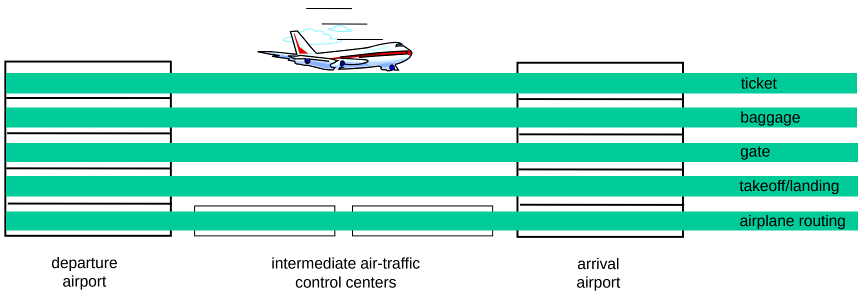
airplane routing

ticket (complain)
baggage (claim)
gates (unload)
runway landing
airplane routing

- a series of steps

---

# Layering of airline functionality



Layers: each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below

# Protocol Stack: ISO OSI Model

- Application
- Presentation
- Session
- Transport
- Network
- Data link
- Physical

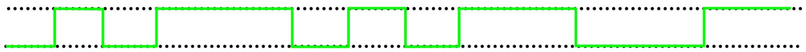
ISO: the International Standards Organization
OSI: Open Systems Interconnection Reference Model (1984)

---

# Layer 1: Physical Layer

- Application
- Presentation
- Session
- Transport
- Network
- Data link
- Physical

- ❑ Responsibilities:
  - ❖ Transmission of a raw bit stream
  - ❖ Forms the physical interface between devices
- ❑ Issues:
  - ❖ mechanical and electrical interfaces
  - ❖ time per bit
  - ❖ distances



---

# Layer 2: Data Link Layer

- Application
- Presentation
- Session
- Transport
- Network
- Data link
- Physical

- ❑ Responsibilities:
  - ❖ Provides reliable transfer of information between two adjacent nodes
  - ❖ Provides frame-level error control
  - ❖ Provides flow control
- ❑ Issues:
  - ❖ *framing* (dividing data into chunks)
    - header & trailer bits
  - ❖ addressing

**10110110101** → **01100010011** → **10110000001** →

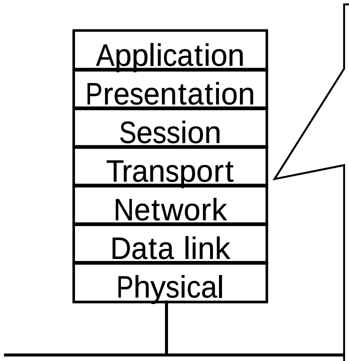
---

# Layer 3: Network Layer

- Application
- Presentation
- Session
- Transport
- Network
- Data link
- Physical

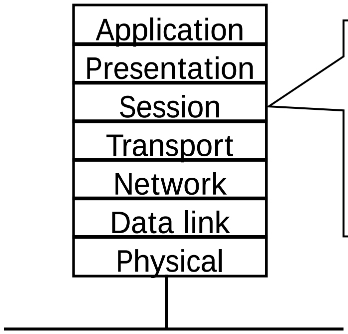
- ❑ Responsibilities:
  - ❖ path selection between end-systems (routing).
    - Dynamic routing
    - Fixed routing
  - ❖ fragmentation & reassembly
  - ❖ translation between different network types

# Layer 4: Transport Layer



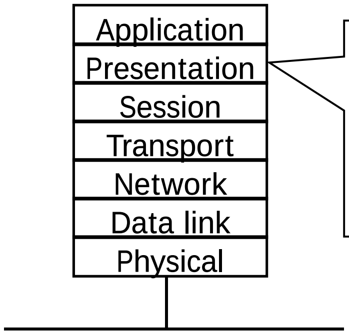
- Responsibilities:
  - provides virtual end-to-end links between peer processes.
  - end-to-end flow control
- Issues:
  - headers
  - error detection
  - reliable communication

# Layer 5: Session Layer



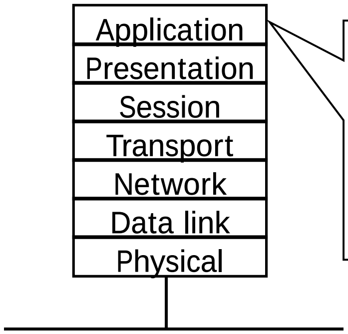
- Responsibilities:
  - Establishes, manages, and terminates a communication session with remote systems
  - Groups several user-level connections into a single "session"
- Many protocol suites do not include a Session Layer.

# Layer 6: Presentation Layer



- Responsibilities:
  - Represents data properly
  - Data encryption
  - Data compression
  - Data conversion
- Many protocol suites do not include a Presentation Layer.

# Layer 7: Application Layer

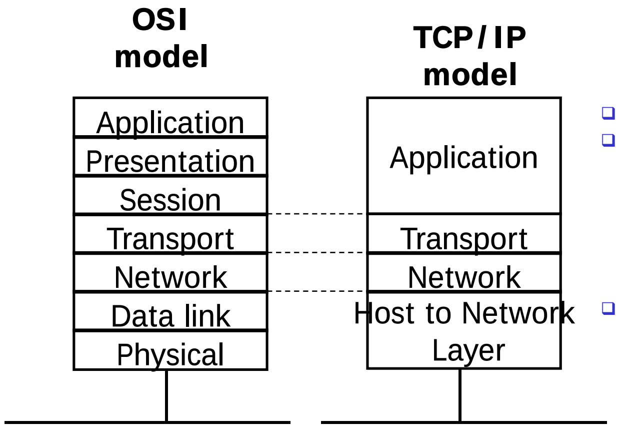


- Responsibilities:
  - Anything not provided by any of the other layers
  - Implements communication between two applications of the same type
    - Examples:
      - FTP
      - HTTP
      - SMTP/POP3/IMAP (email)

# Problems

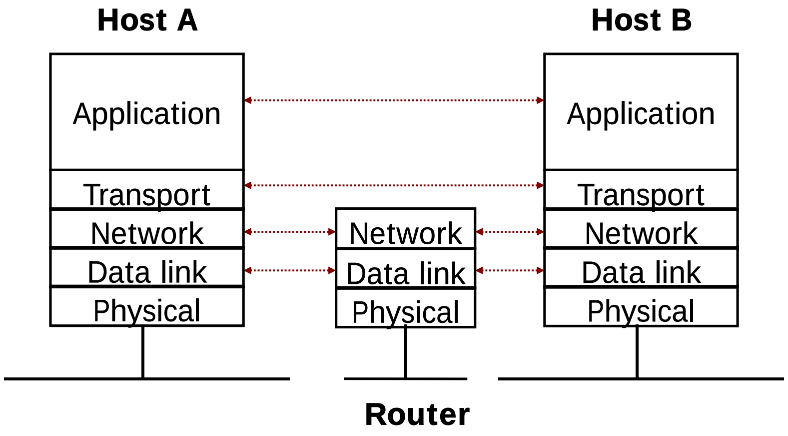
- Seven layers not widely accepted
- Standardized before implemented
- Top three layers fuzzy
- Internet or TCP/IP layering widespread

# TCP/IP Layering Architecture

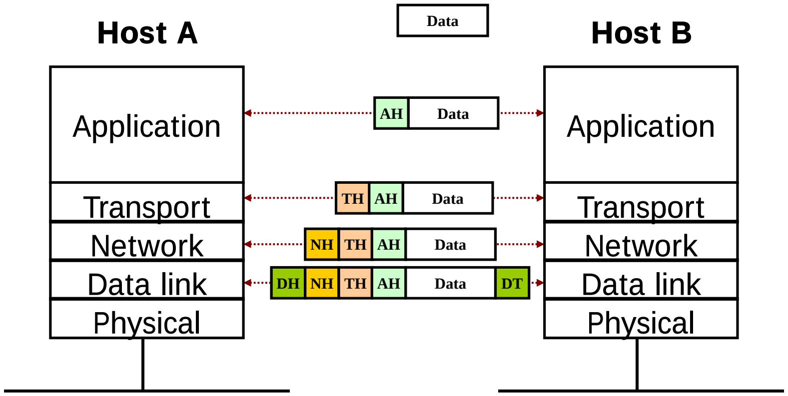


- A simplified model
- The network layer
  - Hosts drop packets into this layer, layer routes towards destination- only promise- try my best
- The transport layer
  - Reliable/unreliable byte oriented stream

# Hybrid Reference Model



# Header encapsulation and stripping



# Layering & Headers

- Each layer needs to add some control information to the data in order to do it's job.
- This information is typically pre-appended to the data before being given to the lower layer.
- Once the lower layers deliver the data and control information - the peer layer uses the control information.

---

# What are the headers?

Physical:

no header - just a bunch of bits.

Data Link:

- address of the receiving endpoints
- address of the sending endpoint
- length of the data
- checksum.

---

# Network layer header - examples

- protocol suite version
- type of service
- length of the data
- packet identifier
- fragment number
- time to live
- protocol
- header checksum
- source network address
- destination network address

---

# Important Summary

- Data-Link:
  - communication between machines on the same network.
- Network:
  - communication between machines on possibly different networks.
- Transport:
  - communication between processes (running on machines on possibly different networks).

# Addresses

- Each communication endpoint must have an address.
- Consider 2 processes communicating over an internet:
  - the network must be specified
  - the host (end-system) must be specified
  - the process must be specified.

# Addresses at Layers

- Physical Layer
  - no address necessary
- Data Link Layer
  - address must be able to select any host on the network.
- Network Layer
  - address must be able to provide information to enable routing.
- Transport Layer
  - address must identify the destination process.

# Repeater

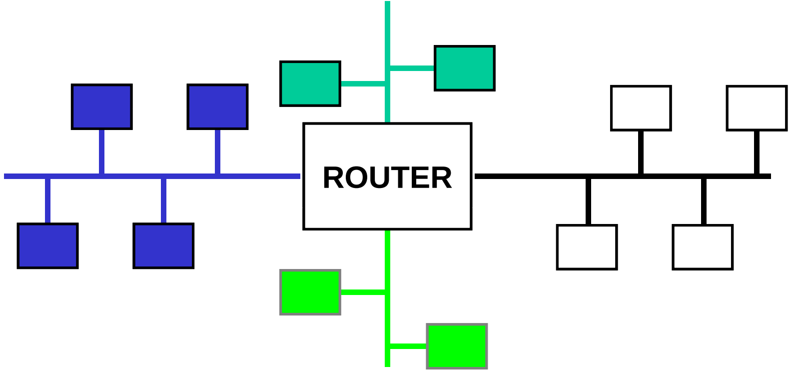
- Copies bits from one network to another
- Does not look at any bits
- Allows the extension of a network beyond physical length limitations

# Bridge

- Copies frames from one network to another
- Can operate selectively - does not copy all frames (must look at data-link headers).
- Extends the network beyond physical length limitations.

## Router

- Copies packets from one network to another.
- Makes decisions about what *route* a packet should take (looks at network headers).



## Gateway

- Operates as a router
- Data conversions above the network layer.
- Conversions:
  - encapsulation - use an intermediate network
  - translation - connect different application protocols
  - encryption - could be done by a gateway

## Which layer?

- Repeater & Hub
  - physical layer
- Bridge & Switch
  - data link layer
- Router
  - network layer
- Gateway
  - network layer and above.

## Hardware vs. Software

- Repeaters are typically hardware devices.
- Bridges can be implemented in hardware or software.
- Routers & Gateways are typically implemented in software so that they can be extended to handle new protocols.
- Many workstations can operate as routers or gateways.

## Internet History

### *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 public demonstration
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes

THE ARPA NETWORK

---

## Internet History

### *1972-1980: Internetworking, new and proprietary nets*

- 1970: ALOHAnet satellite network in Hawaii
- 1974: Cerf and Kahn - architecture for interconnecting networks
- 1976: Ethernet at Xerox PARC
- ate70's: proprietary architectures: DECnet, SNA, XNA
- late 70's: switching fixed length packets (ATM precursor)
- 1979: ARPAnet has 200 nodes

Cerf and Kahn's internetworking principles:
  - minimalism, autonomy - no internal changes required to interconnect networks
  - best effort service model
  - stateless routers
  - decentralized control

define today's Internet architecture

---

## Internet History

### *1980-1990: new protocols, a proliferation of networks*

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

---

## Internet History

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

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

Late 1990's – 2000's:

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

# Internet History

2007:

- ~500 million hosts
- Voice, Video over IP
- P2P applications: BitTorrent (file sharing) Skype (VoIP), PPLive (video)
- more applications: YouTube, gaming
- wireless, mobility