Computer Networks

Fall 2020

Prof./Dr. Shen Bichuan

Chongqing University of Technology

What is this Course All About

- Fundamental principles of Computer Networks
- First course – Broad coverage of topics (important topics in depth)
- Topics categorized to:
  - network architectures and technologies
  - protocols
  - applications
- We will not discuss specific implementations: e.g., how to configure the latest cisco routers

Why Learn about Networking?

- Indispensable part of modern society
  - Commercial – e-commerce, banking, inventorying, telecommunications, archiving, health
  - Social – critical infrastructure, homeland security, policing
  - Human interaction/communication – email, chat, videoconferencing, social networking, entertainment

Why Learn about Networking?

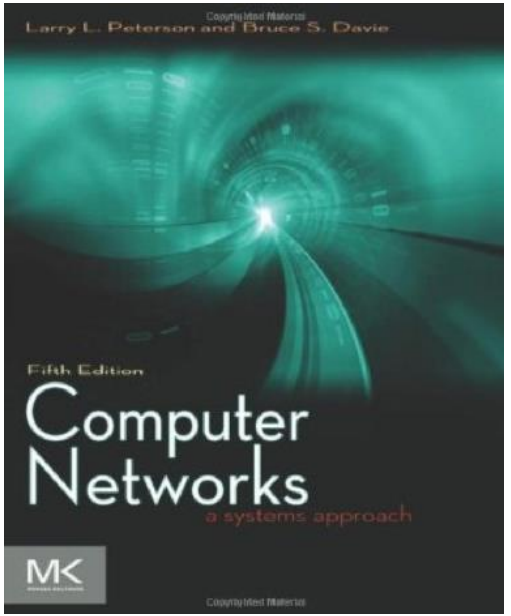
- Appears in every facet of engineering
  - Modern trend – Network every (electronic) device (computers, phones, sensors, planes, cars, TVs, appliances, heart monitors, …)
- Many fields to pursue graduate studies
  - Many problems remain unsolved
  - Research funding is still strong

Course Notes

- **Textbook**
- "Computer Networks: A Systems Approach"
- L. Peterson, and B. Davie, 5th edition.

- Additional References
- "Computer Networks"
- S. Tanenbaum and D. Wetherall,
- 5th edition,

- **Course Groupsite**
- Lectures, Homework, Useful links,
- Supplementary material, Announcements



Grading Scheme

Assignment	Points
Homework & Labs	30
Project	30
Final Exam	40
Total	**100**

Homework: TBD

Final Exam: TBD

Course Objectives

- Develop a fundamental understanding of the network architecture, design principles and performance metrics
- Become familiar with TCP/IP protocols and the mechanisms for reliable data communication via computer networks
- Be able to evaluate the performance of various network technologies and protocols
- Think as an engineer: Can specify, characterize, and design network, and understand what technologies should be employed to build a network with particular specifications?
- Foster knowledge, capabilities and interests in conducting research in the broad areas related to computer networks

Topics to be covered

- Network architectures, performance metrics, layering
- Medium access control
- Internetworking, routing
- TCP/IP protocols, flow control
- Congestion control and resource allocation
- Network security
- Application and Network Programming

Chapter 1

Introduction

Uses of Computer Networks

- Classification based on application domains, areas, contents, usages, and structures etc.
- Domains: personal, home, traffic, various businesses and industries, education, finance, medical, social, military
- Media: wired, wireless, mobile, fiber optics, laser, infrared, microwave, millimeter wave
- Contents: text, voice, music, graphic, images, multimedia, hypertext, video (conferencing and streaming), various data and formats

Applications Examples

- Resource Sharing
  - File sharing
  - Network Printer
  - Network storage
  - Network computing power
  - Backup Systems
- VPN (Virtual Private Networks)
  - Ending the limitation of geographic disperse working environments.
- Client - Server

Applications

- **Information and Data Sharing**
  - **Many web applications, including social and industries**
  - **Traffic**
  - **Online education, DingDing, QQ, Wechat, WeLink**
  - **Email**
  - **VoIP**
  - **Video**
  - **Tele-Conferencing**
  - **Desktop Sharing**
  - **Telemedicine**

Applications

- Audio + Video (Wechat, QQ)
- Message and Instant Messaging (Email, Wechat, QQ, DingDing, Twitter)
- Online Audio and Music (Radio Channels, Ximalaya, QQ Music, Kugou)
- Online Video (Tencent Video, Baidu Video, YouTube)
- Social Networking:
  - Wechat, QQ
  - Facebook, Linkedin
- E-commerce and Online Shopping (Taobao, JD, Suning, Amazon)
- Finances (Online Banking,支付宝, Wechat)
- Online auctions (eBay)

Applications

- **Entertainment:**
  - **MP3 and HD-quality movies**
  - **TV shows – IPTV (IP TeleVision)**
  - **Interactive Live Digital TV**
- **Game Playing**
  - **Multiperson real-time simulation games.**
- **Ubiquitous Computing**
  - **Smart Home Monitoring**
- **RDIF (Radio Frequency Identification)**
  - **Replacing Bar Codes with a smart deivices that my turn the real world inot the Internet of things.**

Mobile Applications

- Mobile computers (handheld and laptops)
  - Fastest growing segments in computer history.
  - Individuals are able to use their mobile devices to:
    - Read and send email,
    - Tweet,
    - Watch Movies,
    - Download Music,
    - Play Games,
    - Serf the Web
- Internet connectivity allows for those applications to be easily built
  - Wireless Networks (Cars, Boats, and Airplanes can not have wired Connections)
  - Cellular Networks
  - Wireless hotspots (802.11 Standard).
  - Wireless Networking vs. Mobile Wireless Networks

Mobile Uses

- Smart Phones – Integration of Internet with Telephony
  - Driving the wireless-mobile applications
  - 4G and 5G cellular networks provides fast data services
  - GPS is a standard feature
  - m-commerce (mobile commerce)
  - NFC (Near Field Communication) smart phones act as an DFID smartcard and interact with nearby reader for payment.
- Sensor Networks
  - Notes that Sense/gather data about state of the physical world.
  - It is revolutionizing science
- Wearable Computers
  - Implantable Devices
    - Pacemakers, Insulin pumps, …
    - Controllable wirelessly

Case Study 1: A “Simple” Task

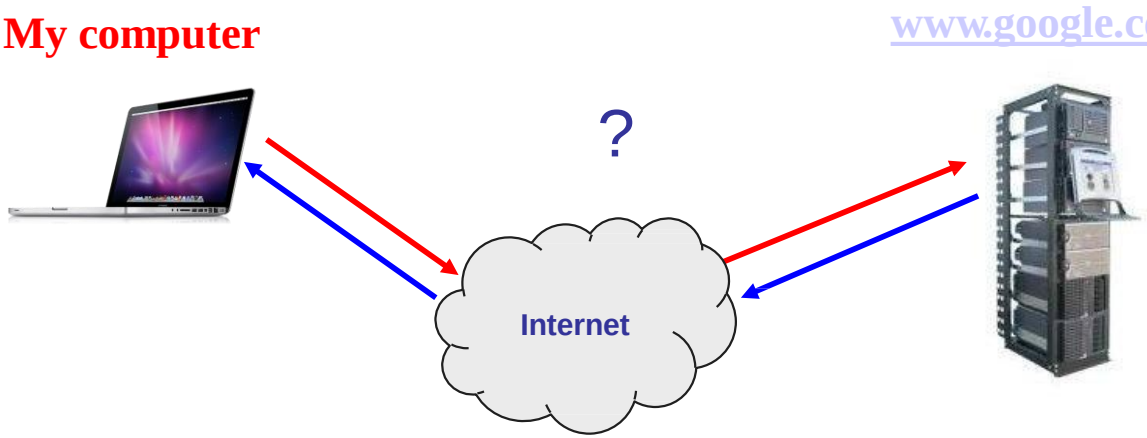
- Send information from one computer to another

Host

- Endpoints are called hosts
  - Could be computer, iPod, cel lphone, etc.
- The plumbing is called a link
  - We don’t care what the physical technology is: Ethernet, wireless, cellular, etc.

Host

Actually Quite Complicated



Web request (HTTP)

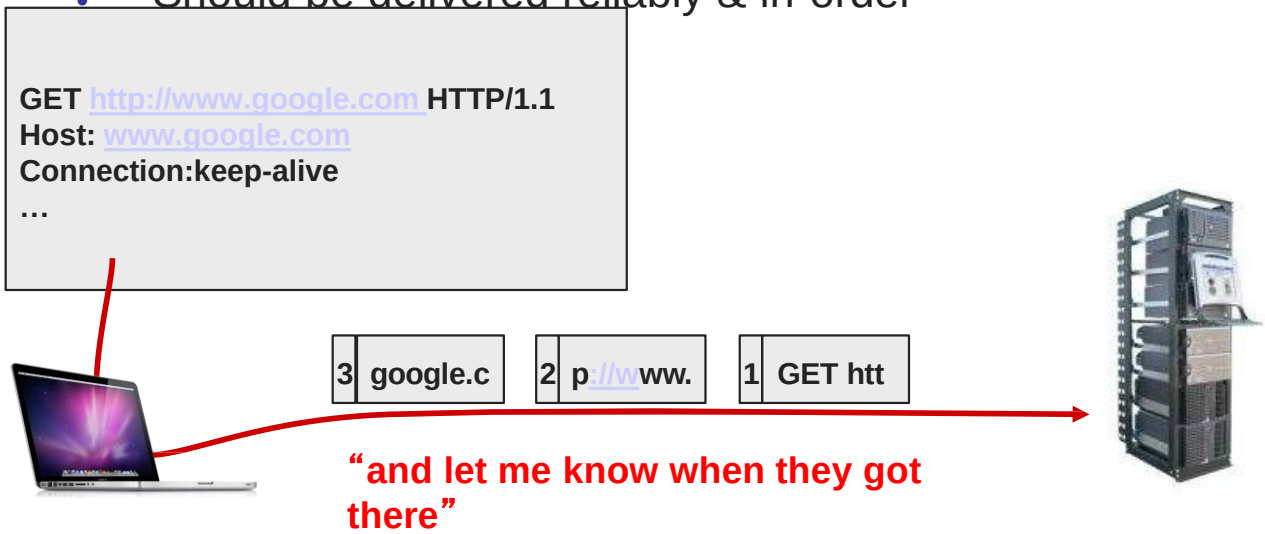
- Turn click into HTTP request

Name resolution (DNS)

- Where is www.google.com?

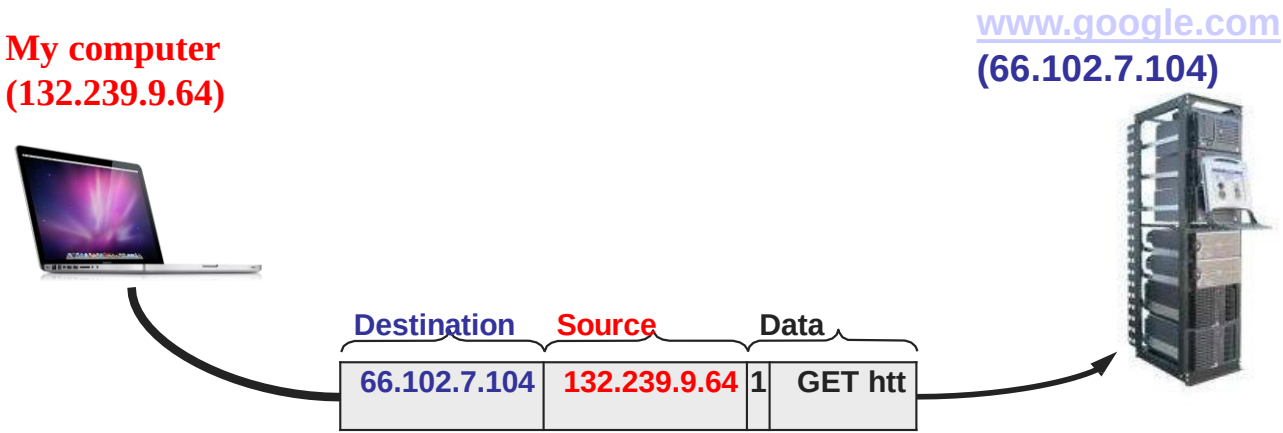
Data transport (TCP)

- Break message into packets (TCP segments)
- Should be delivered reliably & in-order



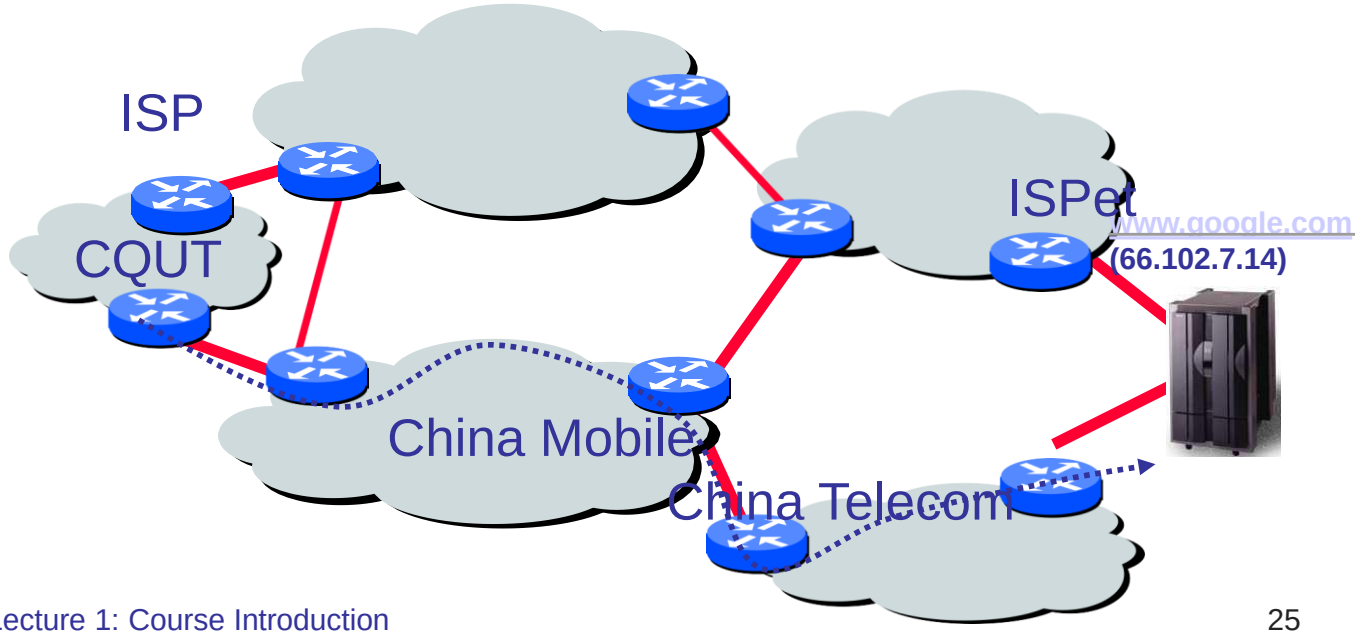
Global Network Addressing

- Address each packet so it can traverse network and arrive at host



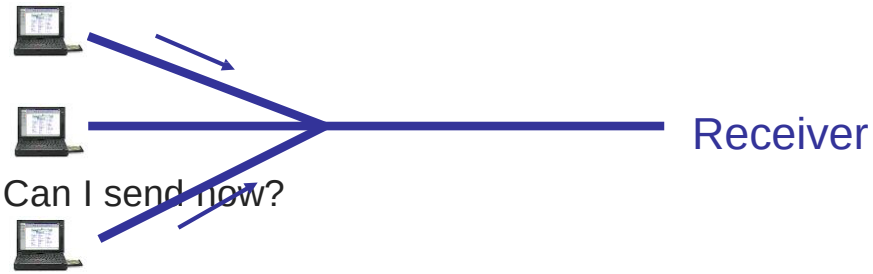
Network Routing

- Each router forwards packet towards destination



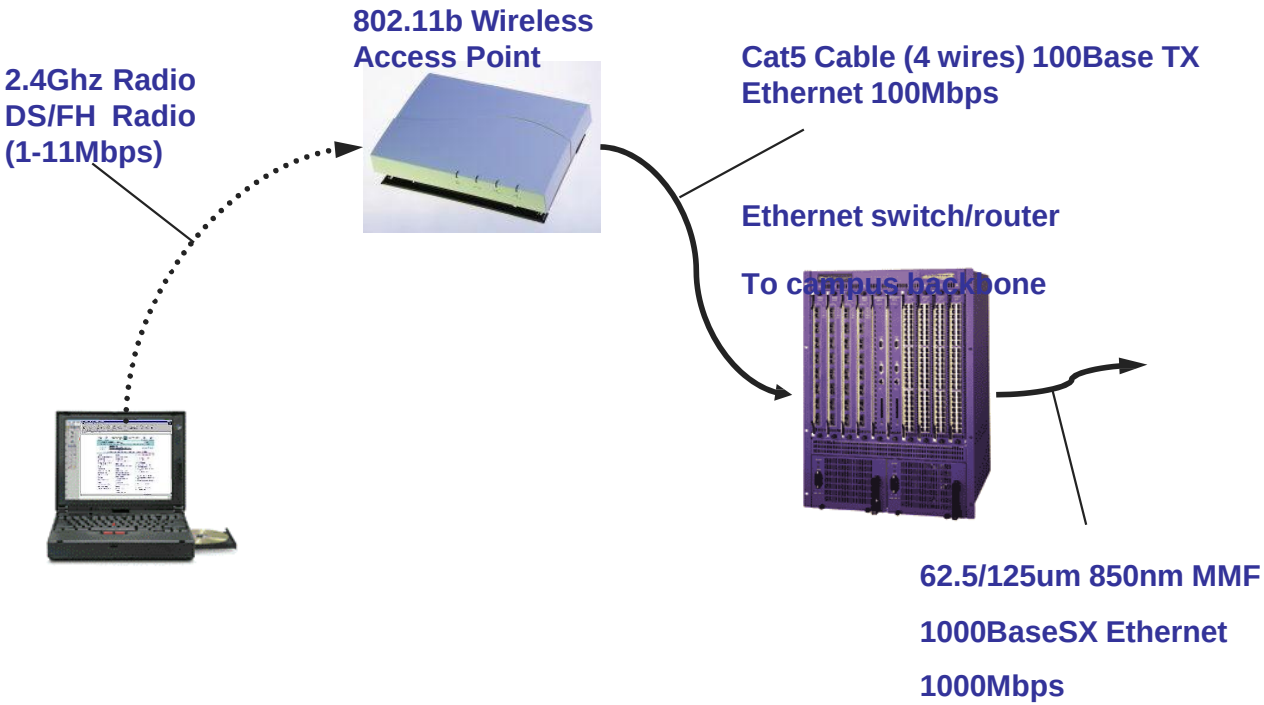
Link management (Ethernet)

- Break message into frames
- Media Access Control (MAC)
  - Can I send now?



  - Can I send now?
- Send frame

Physical layer



Functions Need to Achieve

- Addressing: finding destination, MAC/NET layer in OSI and TCP/IP model
- Connecting: establishing connection between source to destination, PHY/MAC/NET
- Transmission: from start to finish
- Negotiating: protocol
- Managing: large system, network function correctly, devices, nodes, links, servers, bandwidth allocation, managed centrally or by multiple entities
- Configuring: devices and services
- Arbitrating: conflicting access or collision

Requirements and Challenges

- Connectivity, Scale and Scalability, Efficiency, Reliability, Managibility
- Usability:
- Scalability: extension and expansion, growing rapidly and uncontrollably
- Flexibility:
- Quality: system metrics and individual QoS across network, many impacted factors

Requirements and Challenges

- Fairness: access and load fairness, schedule and priority
- Reliability: failures including bit errors, burst, block or packet errors, node or link cutoff, complicated by limited resources, heterogeneous devices and nodes, software and networks
- Complexity
- Security and threat: cybersecurity issues and solutions
- Coverage: broad and wide

Three Main Categories of Networks

Telecom Networks

Telephone, Voice, etc

Cable TV Networks

TV Programs

Computer Networks

Core, many apps and functions

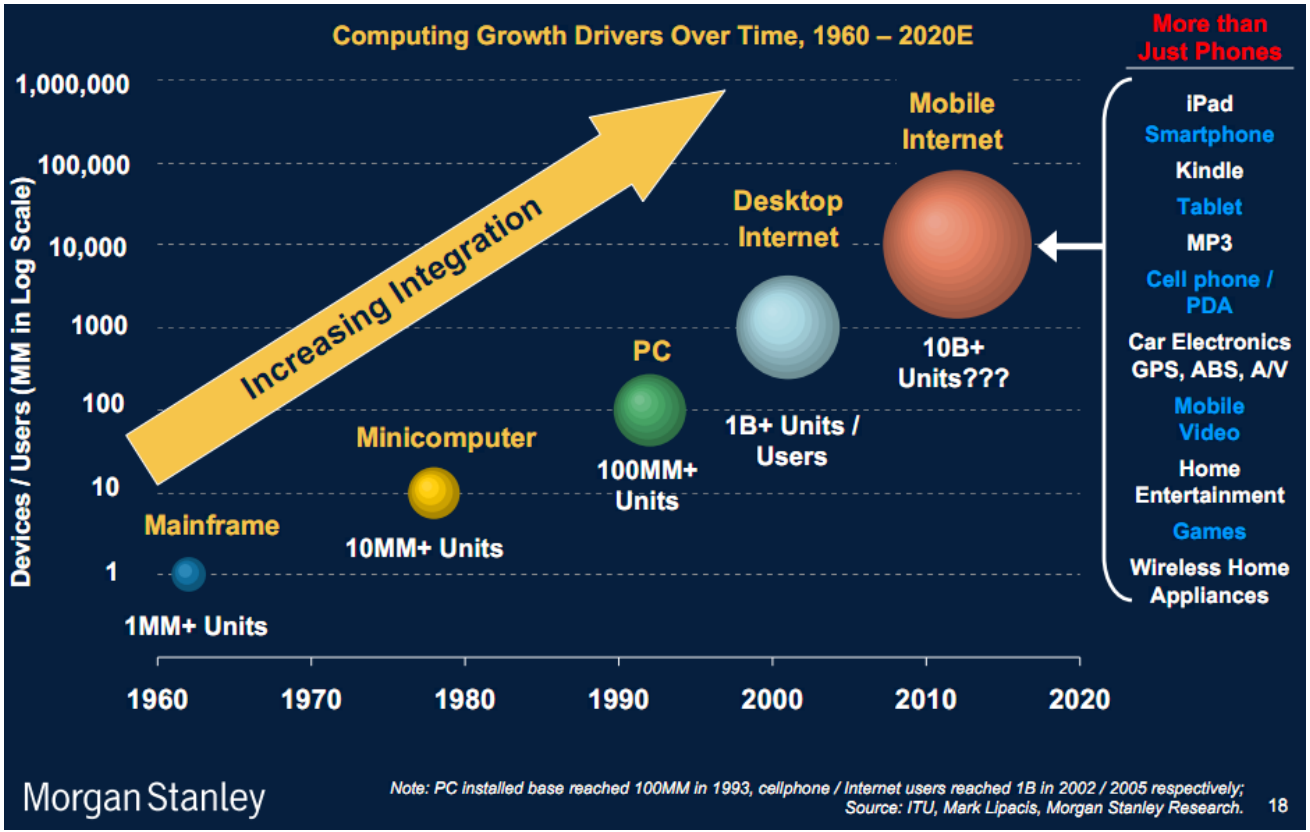
Computer Networks are the core and the fastest growing

Three Merge into One

Evolution and Features

- All digitized，IP
- Broadband, high speed
- Standardized TCP/IP
- Decentralized，distributed
- Transparent，logic separate from physical

Development and Evolution



Evolution and Features of Computer Networks

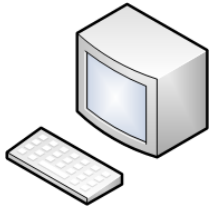
- Open architecture and ecology
- Internet of everything, human-computer interaction, cloud platform
- Cross-border integration, ubiquitous access and service, comprehensive and in-depth integration of traditional emerging industries, into all aspects of society
- More high-speed data transmission, fusion and open sharing
- Comprehensive intelligence, intelligent manufacturing, intelligent transportation, intelligent services, intelligent products, intelligent life
- Secure trusted computing and communications

Biggest Internet Challenges

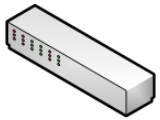
- **Scale**
  - How to manage such a large system,
  - growing rapidly and uncontrollably,
  - consisting of heterogeneous devices,
  - managed by multiple entities
  - having limited resources, such as IP bandwidth

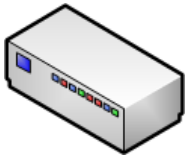
Network Elements

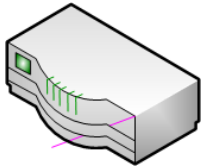
- **Nodes: Special purpose devices**


PC

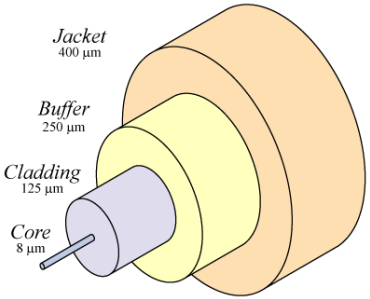

server


switch

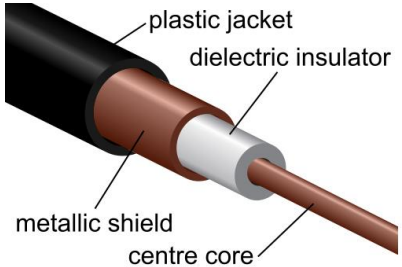

bridge


router

- **Links: Connections between nodes**



Optical fiber



Coaxial cable


wireless

Network Design

- The task of connecting nodes via links, so that nodes can exchange information, reliably, timely, efficiently, safely, privately, and with low cost.

- Need to define the network architecture, protocols, applications, interfaces, policies, usages.

- Let's start with the architecture
  - Directly connected networks
  - Circuit-switched networks
  - Packet-switched Networks

How do we Evaluate a Network

- Metrics (think again a transportation network)
  - Speed, data or bit speed, bit rate
  - Bandwidth, upper data speed limit and boundary, Shannon's theorem
  - How many end nodes or users can it service (throughput)?
  - How fast can it service them (delay latency, or jitter)?
  - How reliable can it service them (collisions, losses, outage probabilities, etc)?
  - Can it provide any service guarantees (QoS)?
  - Any other metrics you can think of?

Network Performance Metrics

Bandwidth

Amount of data transmitted per unit of time; per link, or end-to-end

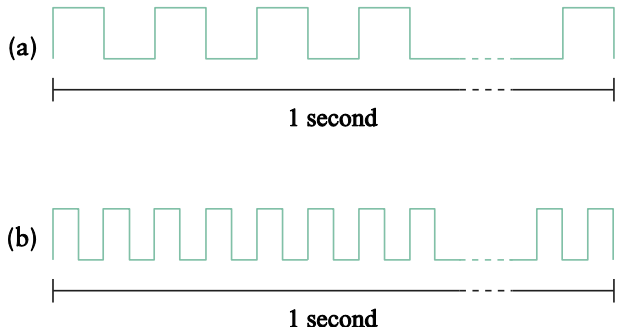
Units 1KB = 2^{10} bytes, 1Mbps = 10^6 bits per sec

How many KB/sec is a 1Mbps line? How many MB/sec?

Throughput

Data rate delivered by the a link, connection or network

Per link or end-to-end, same units as Bandwidth



Latency or Delay

Time for sending data from host A to B (in sec, msec, or μsec)

Per link or end-to-end

- Usually consists of
  - T_t: Transmission delay
  - T_p: Propagation delay
  - T_q: Queuing delay

Round Trip Time (RTT) : time to send a message from A to B and back

- Important for flow control mechanisms

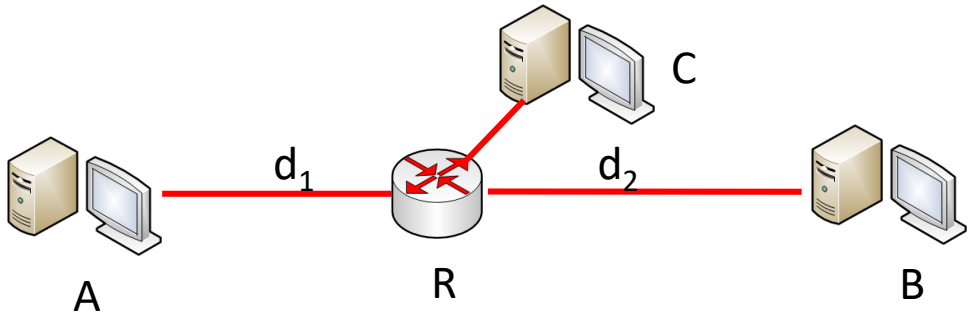
Delay Calculation

T_t : Transmission Delay: file size/bandwidth

T_p : Propagation Delay: time needed for signal to travel the medium, Distance / speed of medium

Tc : Processing Delay: time needed for processing and computing packet

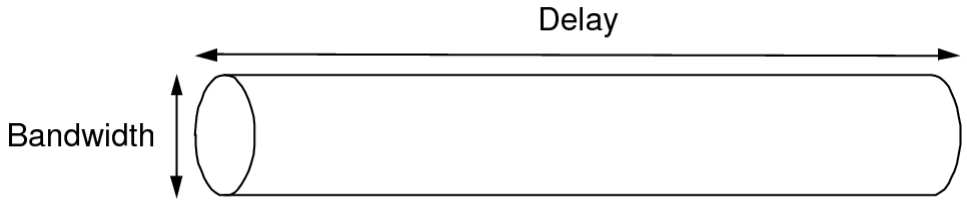
T_q: Queuing Delay: time waiting in router's buffer



Bandwidth x Delay Product

The amount of data (bits or bytes) “in the pipe”

Example: 100Mbps x 10ms = 1 Mbit



The amount of data sent before first bit arrives

Usually use RTT as delay: amount of data before a reply from a receiver arrives to the sender

Network Comparison of Delay Bandwidth Products

Link Type	Bandwidth	Distance	RTT	Delay x BW
Dial-up	56 kbps	10 km	87 µs	5 bits
Wireless LAN	54 Mbps	50 m	0.33 µs	18 bits
Satellite link	45 Mbps	35,000 km	230 ms	10 Mb
Cross-country fiber	10 Gbps	4,000 km	40 ms	400 Mb

Infinite bandwidth

Propagation delay dominates

Throughput = Transfer size/Transfer time

Transfer time = RTT + Transfer size/Bandwidth

1MB file across 1Gbps line with 100ms RTT, Throughput is 74.1 Mbps

Computing Application Bandwidth

FTP can utilize entire BW available

Video-on-demand may specify upper limit (only what's needed)

Example: res: 352x240 pixels, 24-bit color, 30 fps

- Each frame is (352 x 240 x 24)/8 =247.5 KB
- Total required BW = 352 x 240 x 24 x 30 = 60.8 Mbps

Network Jitter

Variability in the delay between packets

Video-on-demand application: If jitter is known, application can decide how much buffering is needed

Example: jitter is 50ms per frame and 10s video at 30fps must be transmitted.

If *Y* frames buffered, video can play uninterrupted for *Y* x 1/30s.

The last frame will arrive 50 x (10 x 30 – Y) ms after video start, worst case

$$Y/30 = 50 \times (300 - Y) \rightarrow Y = 180 \text{ frames}$$

- Delay or latency is the time required for data (a message or packet, or even bits) to travel from one end of the network (or link) to the other.
- This is called delay or delay, and is different from "jitter". For example, students' collective lateness is called delay, time variations in students' lateness is called jitter, the results and effects are different
- The delay in the network consists of the following different parts:
  ① Transmission delay: transmission and the sending queue dependent
  ② Propagation delay: routing path, single-hop, multi-hop, transmission medium, PHY layer dependence
  ③ Processing delay: computing and processing dependent
  ④ Queue delay: queue

Network Function Classification

There are three major categories:

- **Broadcast Network: one-to-all**
- **Point-to-point Network (P2P): unicast, one-to-one**
- **Multicast Network: one-to-many, group communication, business group, community, circle, including group, WeChat group, QQ group, linkedin, support various data formats, message text, speech and voice, image and video, live stream**

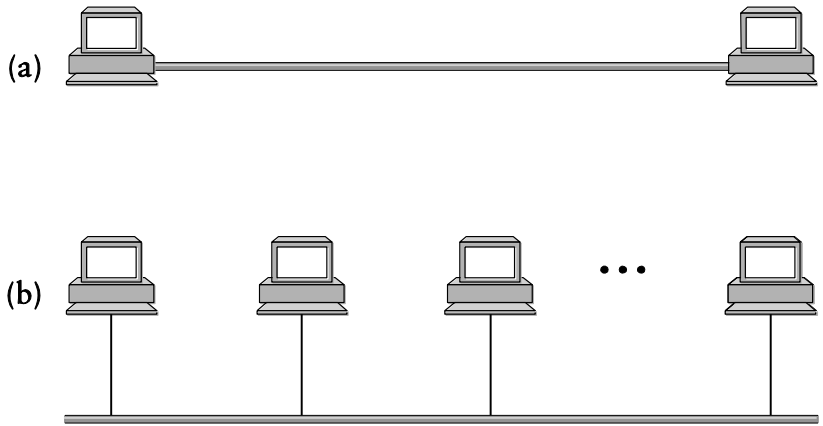
Summary: the logic network can be relatively independent of the physical network; Separation of functions and services from physical equipment and devices; Easy reconfiguration of functions, services and management; Embodies the scalability and flexibility; Good confidentiality and security

- **Three types of transmission technologies:**
  - **Broadcast**
    - Communication channel shared by all machines
    - Packets send by any machine are received by all the others.
      - An address field within each packed specifies the intended recipient.
      - If packed is intended for some other machine, it is just ignored
      - If packed is indented for the recipient machine then it is processed.
    - Wireless network is a common example of a broadcast link
      - Communication is shared over a coverage region that depends on the wireless channel and the transmitting machine.
    - Broadcast systems usually also allow the possibility of addressing a packet to all destinations.
  - **Multicast**
    - One to many
  - **Point-to-point**
    - Connect individual pairs of machines
    - Packets (short messages) may have to visit one or more intermediates machines.
    - Multiple routes of different lengths are possible.
    - Finding good ones is important.
    - Unicasting – transmission with exactly one sender and exactly one receiver.

Directly-Connected Networks

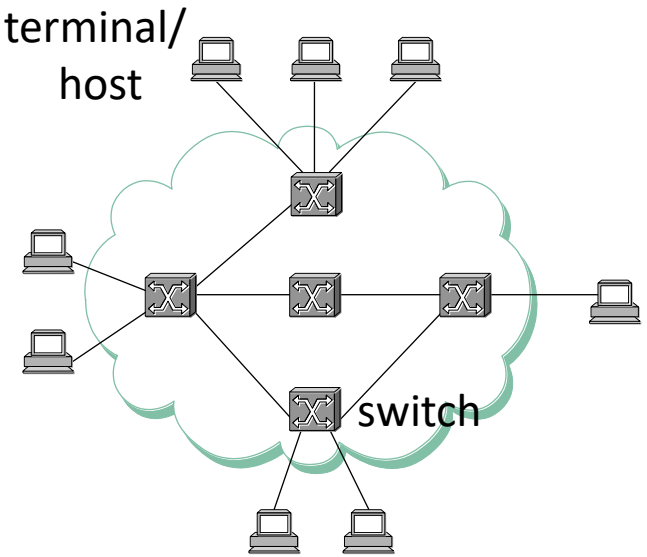
- **Point-to-point links: Each node is directly connected to all others via a link**

- **Multiple access: All nodes share the same physical medium**



Switched Networks

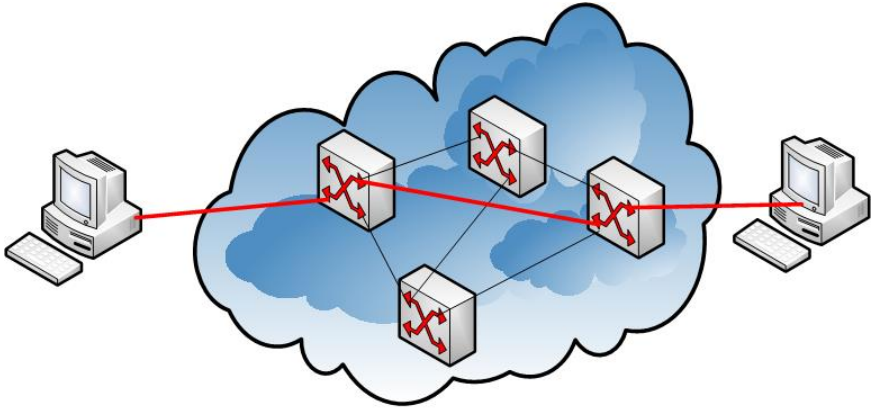
- **Circuit-Switched**
  - A dedicated circuit is established across a set of links
  - Example: Telephone network



- **Packet-Switched**
  - Data is split into blocks called packets or messages.
  - Store-and-forward strategy
  - Switches: Store and forward packets

Circuit-Switched Networks

- End-to-end permanent connection
  - Dedicated path for communication
  - No need for a destination address since a path is already established
- Once communication is complete, connection is ended and links are released.



Advantages of Circuit Switching

- Guaranteed bandwidth (Quality of Service)
  - Predictable bitrate and delay
  - Good for delay-sensitive applications
- Reliable communication
  - Rare packet loss
  - Information or packets are delivered in order
- Simple data routing
  - Forwarding based on time slot or frequency (multiplexing)
  - No need to inspect a packet header for address
- Low per-packet overhead
  - Forwarding based on time slot or frequency
  - No IP (and TCP/UDP) header on each packet

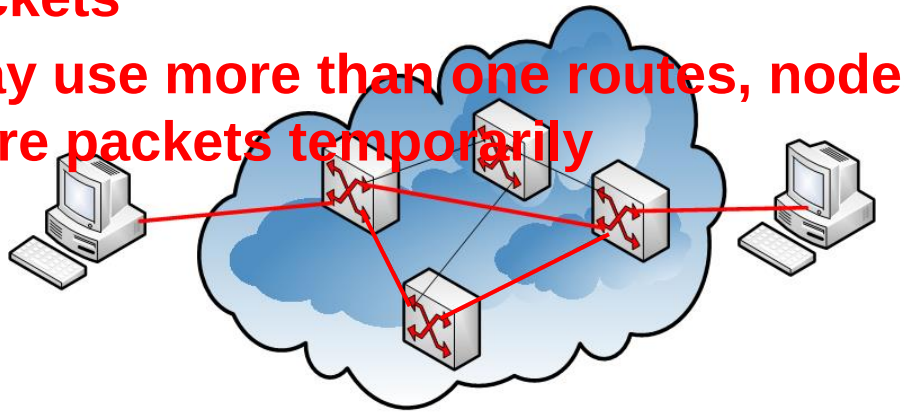
Disadvantages of Circuit Switching

- **Wasted bandwidth**
  - Connection and network resources always occupied per usage
  - Bursty traffic leads to idle connection during silent period
- **Blocked connections**
  - Connection refused when resources are not sufficient
  - Unable to offer “okay” service to everybody
- **Connection set-up delay**
  - No communication until the connection is set up
  - Unable to avoid extra latency for small data transfers
- **Network state**
  - Network nodes must store per-connection information
  - Unable to avoid per-connection storage and state

Packet Switched Networks

- Data is divided into packets (messages)
  - Each packet contains identification info (source/destination address seq. number, etc)

- Packets traverse the network individually
  - Use the destination address to forward packets
  - May use more than one routes, nodes may store packets temporarily



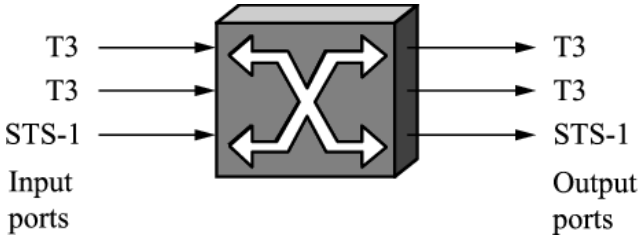
Features of Packet Switching
(mainly not disadvantages)

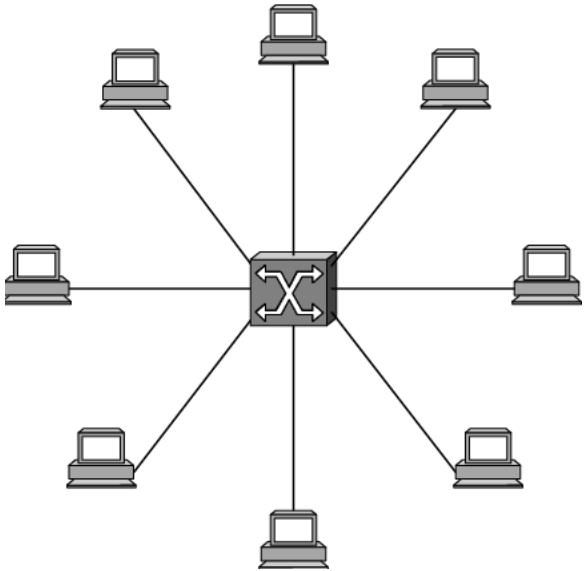
- **No guaranteed bandwidth**
  - Harder to build applications requiring QoS
- **Per packet overhead**
  - Need a header with source/dest. address, etc.
- **Complex end-to-end control**
  - Packets can be lost, corrupted or delivered out-of-order
- **Delay and Congestion**
  - No congestion control, can lead to arbitrary delays and packet drops

Switching Topology

A switch implements a star topology

Switches are MIMO devices





Ports are numbered

Connectionless Networks

No dedicated connection between communicating hosts

Packets are sent to the switch at any time (no contention)

Source is not aware of the state of the destination

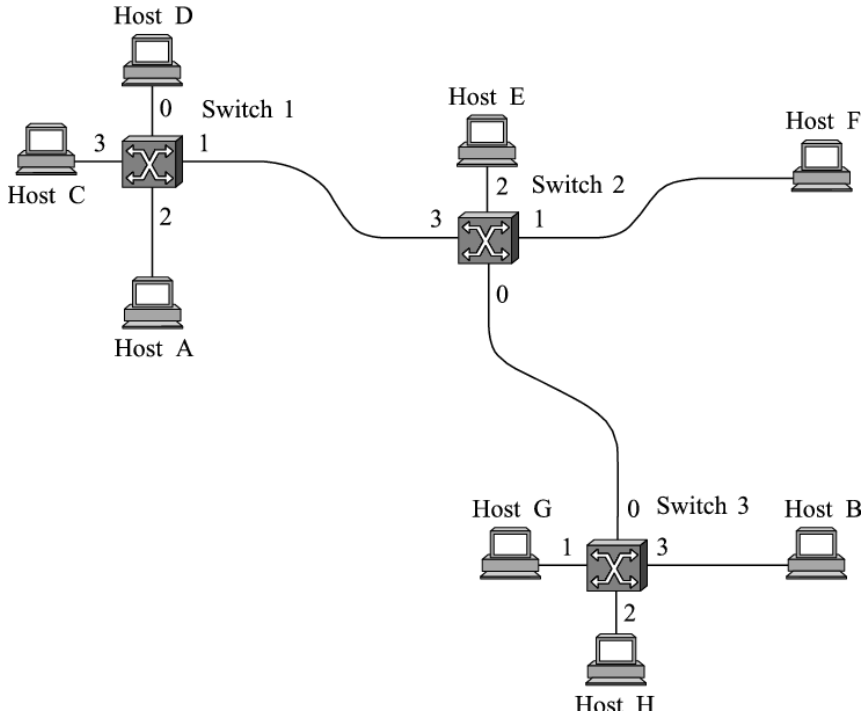
Packets may follow independent paths to the destination (out-of-order delivery, larger delays, etc.)

Less prone to switch failures if alternative paths exist

Datagrams

Packets sent to each switch containing the destination address

Destination	Port
A	3
B	0
C	3
D	3
E	2
F	1
G	0
H	0



Virtual Circuit Switching

Also referred to as connection-oriented

First a connection is setup, followed by a data transfer

VC table created for the connection setup

- Incoming VC identifier (identifies the connection per link)
- Outgoing VC identifier (possibly different that the incoming)
- Incoming interface (different than the VC Identifier, similar to a port)
- Outgoing interface a(different than the VC identifier, similar to a port)

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	5	1	11

VC Connection Setup

Permanent VC (PVC): Administrator sets up a permanent connection and configures VC table

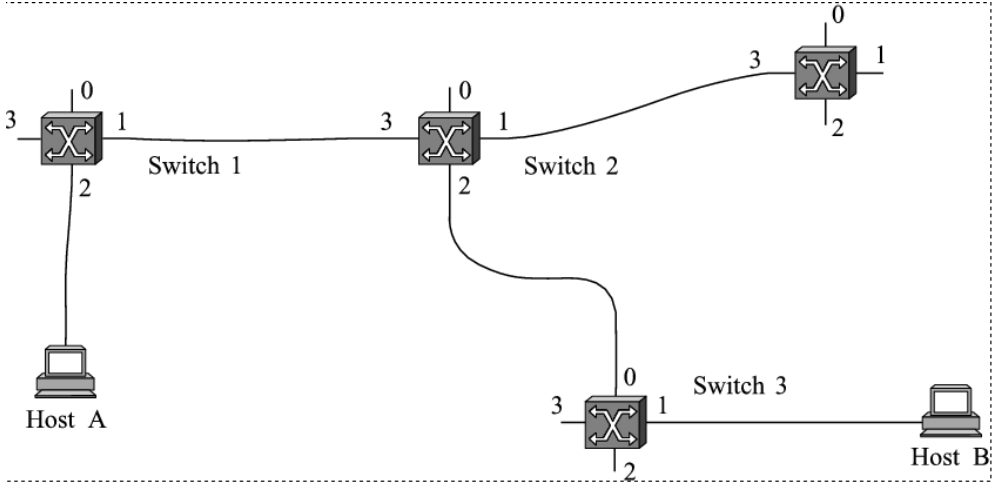
Switched VC (SVC): The host signals to the switches in order to establish a VC, dynamically.

Example: PVC from A to B

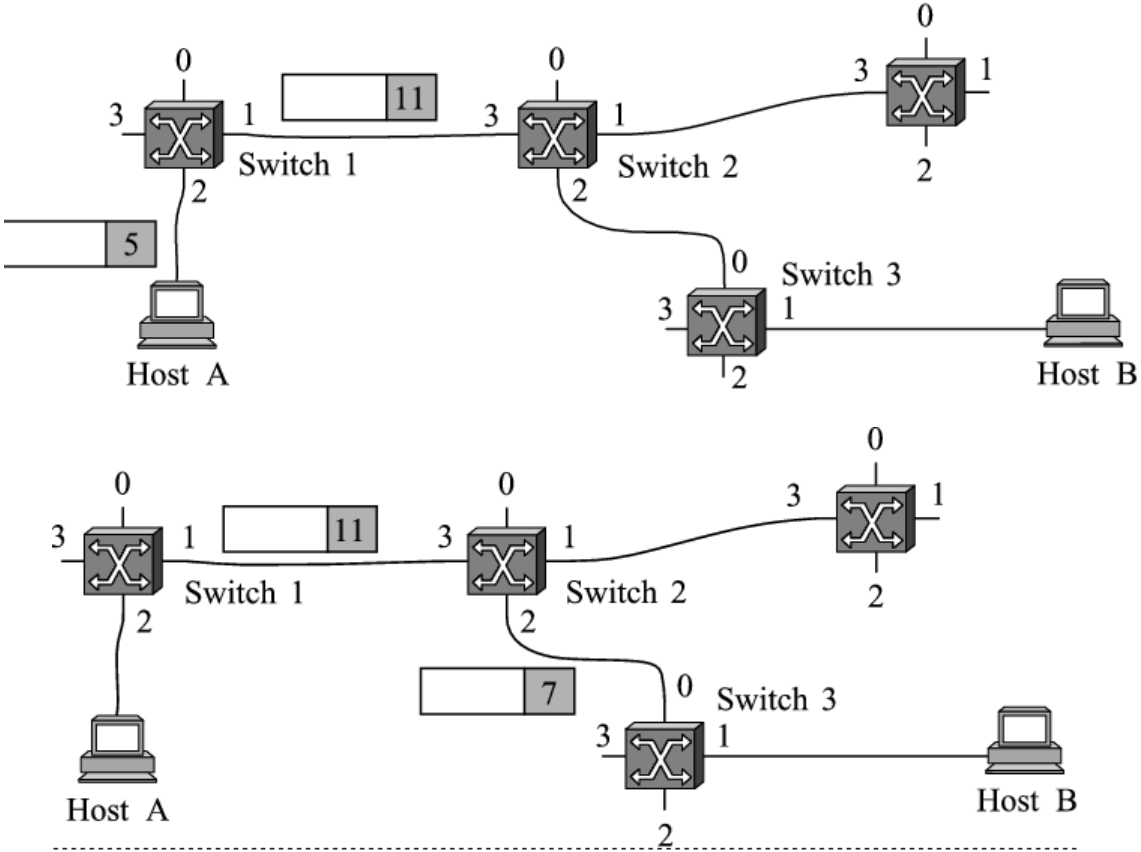
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	5	1	11

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	11	2	7

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	7	1	4



Data Transfer Stage



Example: SVC from A to B

A sends setup msg to switch 1 indicating the address of B

Switch 1 setups incoming/outgoing interfaces and VCIs

Connection setup msg is forwarded like a datagram to switch 2

Switch 2 repeats the setup process

Once data stage is over, connection is torn down

Some Observations on VC

Initial setup msg contains the address of destination, data packets just need the VCI #

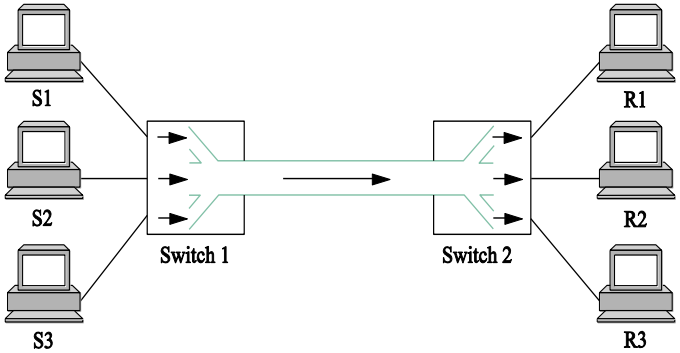
A switch/link failure leads to the repetition of the connection set up process. Old connection must also be terminated

Routing algorithm is needed for establishing the VC

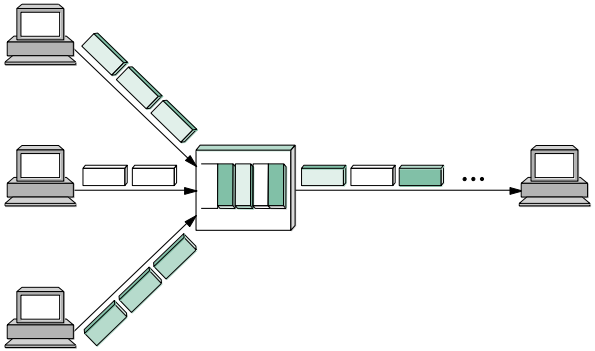
Advantages of Packet Switching

- No wasted bandwidth (not entirely true)
  - Links are not reserved during idle period
- Multiplexing (see next slides)
  - Frequency, time, statistical, and code multiplexing
- Service
  - More connections and services, multi-task
  - No blocking of users and connections
- Adaptation
  - More flexible and scalable
  - Can adapt to network congestion and failures

Multiplexing



Three pairs of senders/receivers share the same physical link to communicate

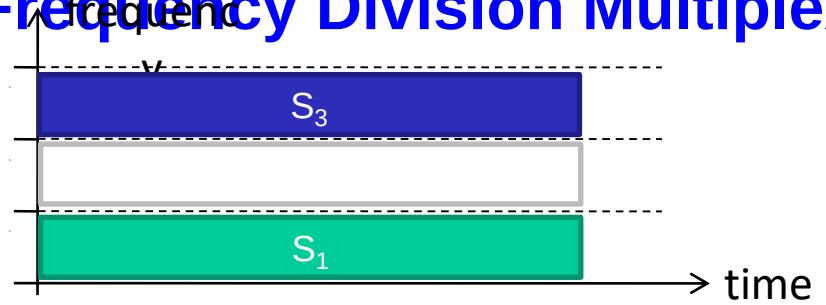


A switch is multiplexing packets from different senders into one packet stream

Multiplexing Methods

- **Time Division Multiplexing**

- **Frequency Division Multiplexing**



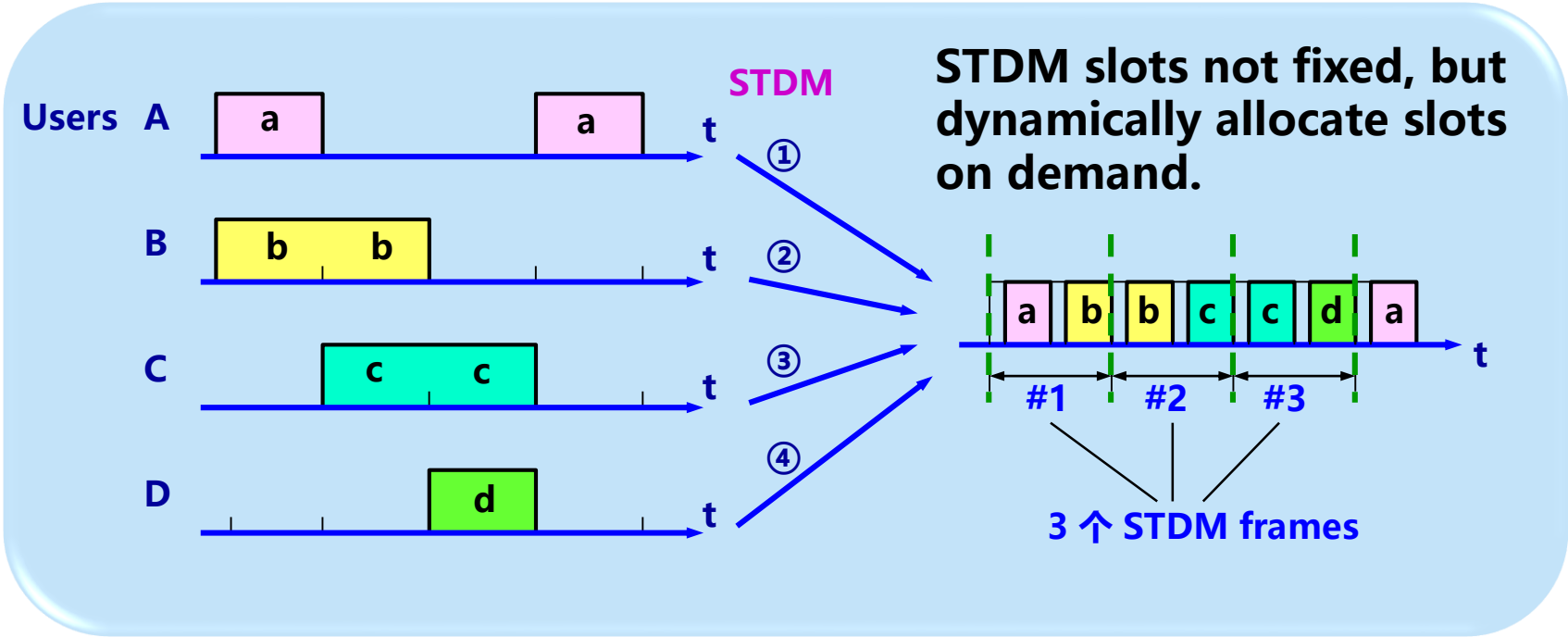
Multiplexing Methods

- **Statistical multiplexing**
  - **Division of the communication medium into a number of channels of variable bandwidth**
  - **STDM frames do not allocate slots fixed, but dynamically allocate slots on demand.**
  - **Therefore, statistical time-division multiplexing can improve the utilization of the line.**

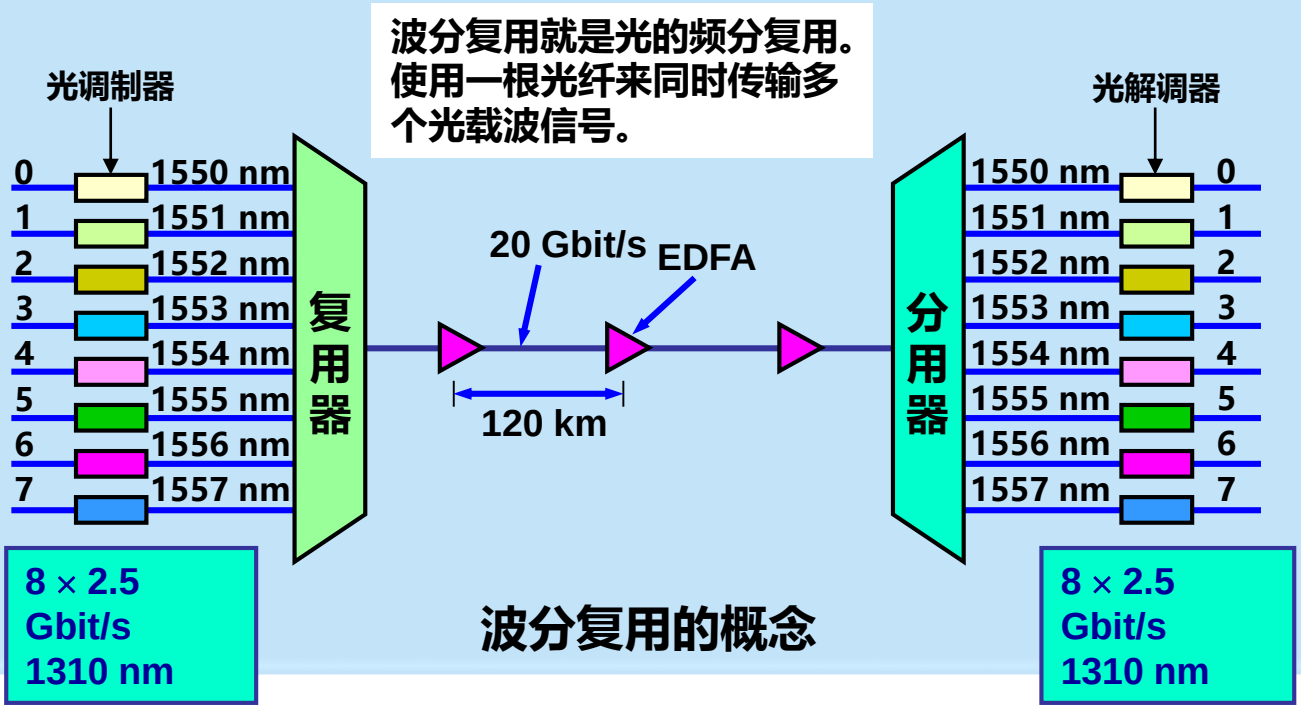
CDM Code Division Multiplexing

- **Code Division Multiple Access**
- **Each user USES different code types that have been specially selected, so there is no interference with each other.**
- **The signal sent by this system has a strong anti-interference ability, its spectrum is similar to white noise, not easy to be detected by the enemy.**

STDM (Statistic TDM)



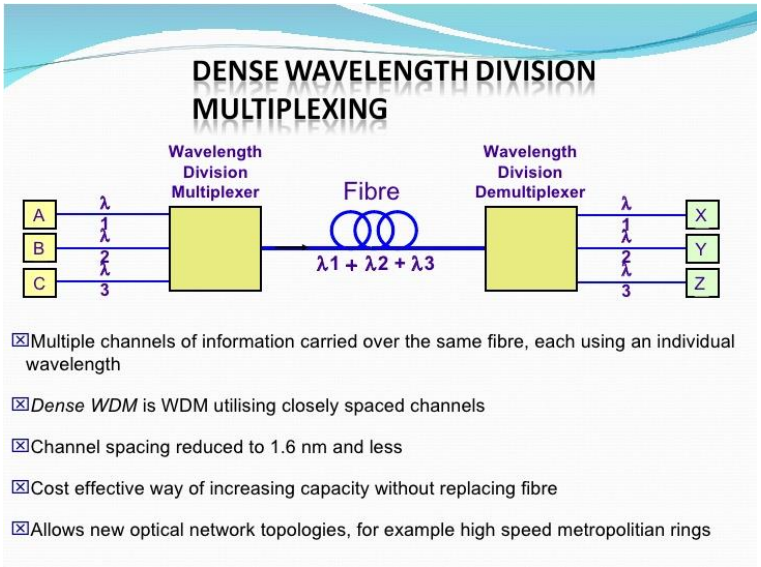
2.4.2 波分复用 WDM(Wavelength Division Multiplexing)



波分复用的概念

Optical WDM，Greatly Improve Total Channel Bandwidth

- **Dr. Tingye Li, Optical fiber WDM Poineer, AT&T and Bell Labs, OSA USA**



Network Hardware (1)

Alternative Criteria: Scale

Distance is important as a classification metric because different technologies are used at different scales.

- Personal area networks
- Local area networks
- Metropolitan area networks
- Wide area networks
- The internet

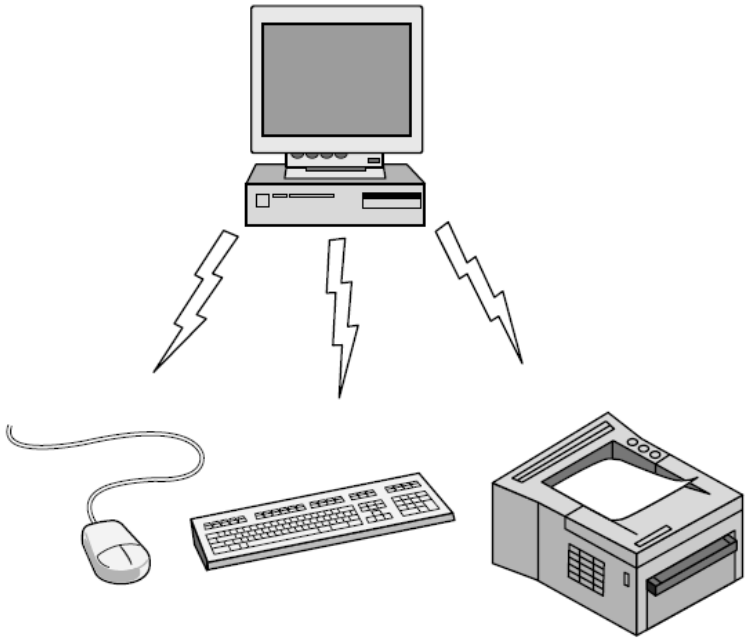
Network Hardware (2)

Classification of interconnected processors by scale.

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet

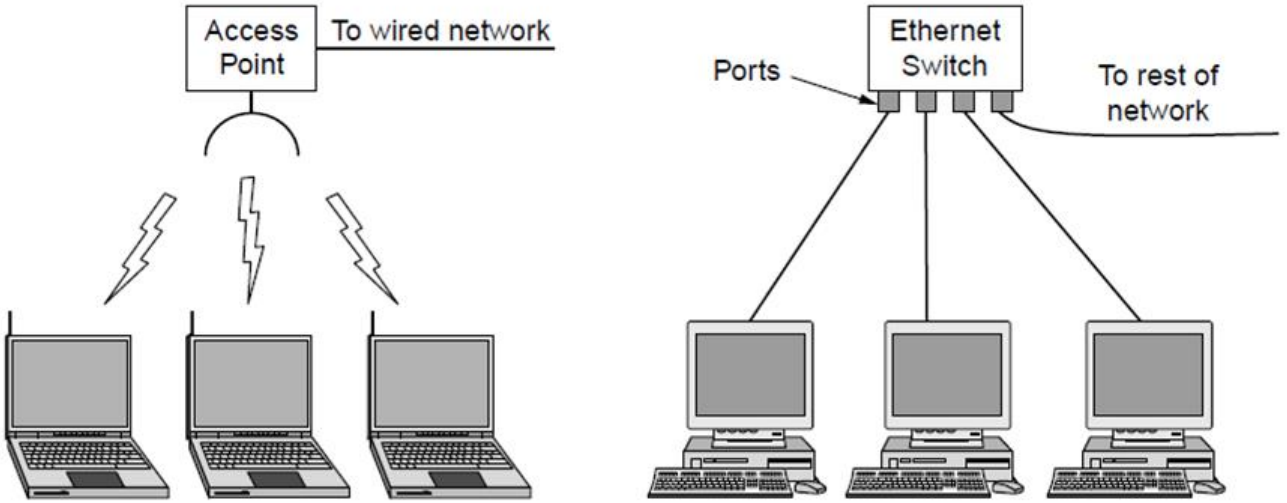
Personal Area Network

Bluetooth PAN (Personal Area Network) configuration



Local Area Networks

Wireless and wired LANs.
(a) IEEE 802.11 or WiFi.
(b) Switched Ethernet (802.3).

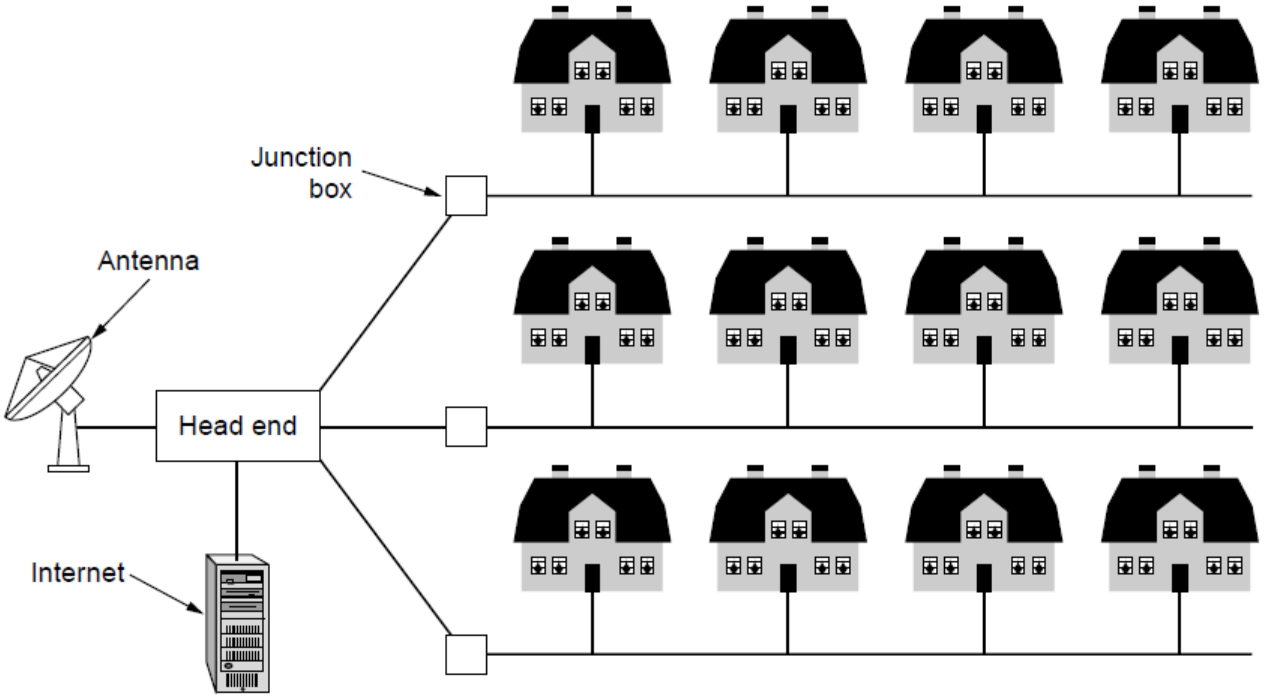


Local Area Networks (LAN)

- Switched Ethernet
  - Switch; Hardware that connects two devices point-to-point
  - A Switch has multiple ports
- Physical vs. Virtual LAN – VLAN

- Dynamic vs. Static Channel Allocation
  - Static Allocation: Each device is allocated its time slot weather or not it uses it.
  - Dynamic methods allow changing the time allocation scheme.
- Dynamic Allocation
  - Centralized
  - Decentralized

Metropolitan Area Networks

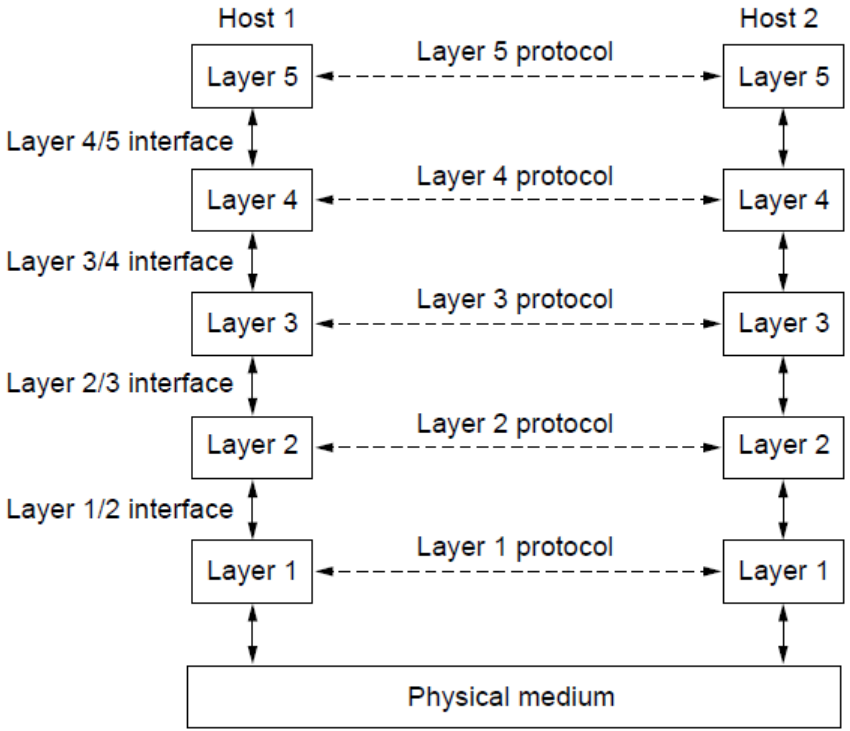


A metropolitan area network based on cable TV.

Network Software

- Protocol hierarchies
- Design issues for the layers
- Connection-oriented versus connectionless service
- Service primitives
- Relationship of services to protocols

Protocol Hierarchies (1)



Layers, protocols, and interfaces.

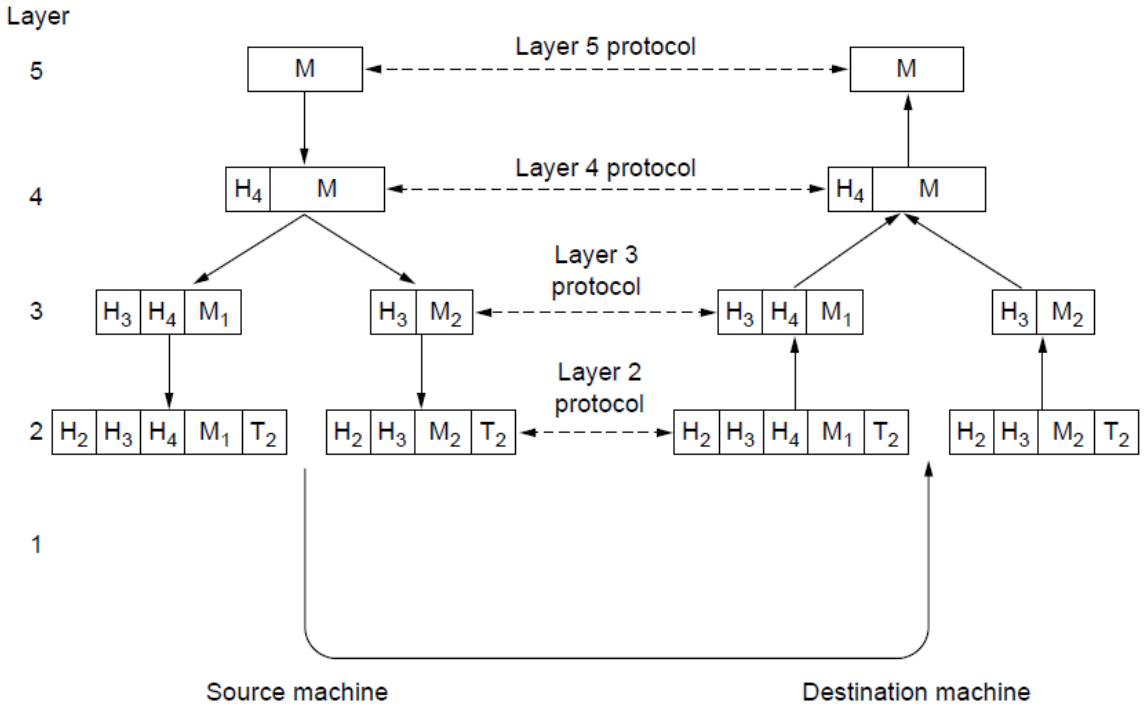
Definitions

- **Protocol – is an agreement between the communicating parties.**
- **Peers – the entities comprising corresponding layers on different machines.**
  - **Peers use the protocol to communicated with each other.**
- **No data is directly transferred from layer n on one machine to layer n on another machine.**
  - **Each Layer passed data and control information to the layer immediately below it until the lowest layer is reached.**
  - **Below layer 1 is the physical medium through which actual communication occurs.**
  - **Virtual communication is shown by dotted lines and physical communication by solid lines the previous figure.**

Definitions

- **Interface**
  - **It defines which primitive operations and services the lower layer makes available to the upper one.**
- **Network Architecture:**
  - **A set of layers and protocols.**
- **The specification of the network architecture must contain enough information to allow an implementation of the program or the hardware for each layer so that it will obey appropriately the protocol.**
- **Protocol Stack:**
  - **The list of protocols used by a certain system – one protocol per layer.**

Protocol Hierarchies (3)



Example information flow supporting virtual communication in layer 5.

Design Issues

- **Reliability:**
  - Network must operate correctly although it is made up of a collection of components that are themselves unreliable.
- **Error Detection:**
  - It typically uses codes to locate the erroneously transmitted bit(s) and request re-transmission.
- **Error Correction**
  - Correct messages is recovered from the possibly incorrect bit(s) that were originally received.
- **Routing:**
  - Finding a working path through a network.
- **Protocol Layering:**
  - Networks grow larger over time and new designs emerge that need to connected to the existing networks.

Design Issues (cont.)

- **Addressing and Naming:**
  - Every layer needs a mechanism for identifying the senders and receivers that are involved in a particular message.
- **Internetworking:**
  - Different network technologies often have different limitations:
    - Not all communication channels preserve the order of messages send on them.
    - Differences in the maximum size of a message that the networks can transmit.
- **Scalable:**
  - Designs that continue to work well when the network gets large.
- **Resource Allocation**
  - Networks work with their resources to provide services to various hosts. If they are not aware of limitations of the networks resources than the network is providing proper resource allocation.
- **Flow Control**
  - Feedback from the receiver to the sender is often used to alleviate the problem of the sender swamping the slow receiver with data.

Design Issues (cont.)

- **Congestion:**
  - The problem may occur when the network is oversubscribed because to many computers want to send too much traffic and the network will not be able to deliver them all.
  - Overloading problem of the network.
  - One strategy is for each computer to reduce its demand.
- **Quality of Service**
  - Additional Resources (other then Bandwidth),
  - Real-time delivery (for applications that require high throughput),
  - Live Video,
- **Network Security**
  - How good is the network against different kinds of threats
    - Eavesdropping,
    - Confidentiality,
    - Authentication,
    - Integrity, etc.

Connection-Oriented Versus Connectionless Service

- Layers can offer two different types of service to the layers above them:
  - Connection-oriented, and
  - Connectionless

Connection-Oriented Service

- Modeled after telephone system:
  - Pickup-the-phone
  - Dial the number
  - Talk
  - Hang-up
- Service User:
  - Establishes a connection,
  - Uses a connection (sender pushes objects in at one end and the receiver takes them out at the other end).
  - In some cases when connection is established, the sender, receiver, and a subnet conduct a negotiation about the parameters to be used:
    - Maximum message size,
    - Quality of service required,
    - Other issues (like …?)

Connection-Oriented Service

- **A circuit:**
  - **Another name for a connection with associated resources such as a fixed bandwidth.**

Connectionless Service

- **Modeled after a postal system:**
  - **Each message carries the full destination address, and**
  - **Each one is routed through the intermediate nodes inside the system independent of all the subsequent messages.**

- **Different Names for Messages:**
  - **Store-and-forward switching: Packet, a message, is processed in full before sending it on the next node.**
  - **Cut-through-switching: when the onward transmission of a message at a node start before it is completely received.**
- **Each kind of the Service can be further characterized by its reliability:**
  - **A reliable service is implemented by having the receiver acknowledge the receipt of each message.**
  - **Acknowledgment service introduces overhead and delays.**

Connection-Oriented Service

- **Example: File Transfer**
  - **The owner want to be sure that all the bits arrive correctly and in the same order they were sent.**
  - **Almost there are no instances were the consumers prefer service that occasionally scrambles or loses a few bits for the gained speed.**

Connection-Oriented Service

- **Reliable connection-oriented service:**
  - **Message Sequences, and**
  - **Byte Streams**
- **Message Sequences:**
  - **Message boundaries are preserved.**
  - **Example: Two 1024 byte messages are sent, the arrive as two distinct 1024-byte messages; Never as one 2048-byte message.**
- **Byte Streams:**
  - **Message is send as a stream of bytes with no concepts of message boundaries.**
  - **Example: When a 2048-byte message arrives at the receiver there is no way to tell if they were sent as**
    - **One 2048-byte message,**
    - **Two 1024-byte message, or**
    - **2048 1-byte messages.**

Example of Applications

- **The transit delays introduced by acknowledgments are unacceptable:**
  - **Digitized voice traffic for Voice-Over-IP (VoIP).**
  - **Digitized video conference**

- **Not all applications require connections. Spam:**
  - **Spammer does not want to go through the trouble of setting up and latter tearing down a connection to a recipient just to send them one more item.**
  - **100% reliability is not essential either.**

- **Datagram:**
  - **Unreliable (not acknowledged) connectionless service.**
  - **It is analogous to telegram service**

Example of Applications

- **Acknowledged Datagram:**
  - The convenience of not having to establish a connection, but
  - Reliability essential
  - Similar to “Return Receipt” for the letter.
  - Example: Text Messaging on mobile phones
- **Request-Reply Service:**
  - Sender transmits a single datagram containing a request;
  - The reply contains the answer.
  - Example: Mobile phone sending the query to a “map server” to retrieve the map data.

Connection-Oriented Versus Connectionless Service

	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages
	Reliable byte stream	Movie download
	Unreliable connection	Voice over IP
Connection-less	Unreliable datagram	Electronic junk mail
	Acknowledged datagram	Text messaging
	Request-reply	Database query

Six different types of service.

Reliable vs. Unreliable Communication

- **Why would one prefer unreliable communication vs. reliable one?**
  1. **Reliable communication may not be available: Ethernet.**
     - **Packets can be damaged.**
     - **It is up to higher levels of protocol to recover from this problem.**
     - **Many reliable services are built on top of the unreliable service.**
  2. **The delays for providing reliable service are not acceptable:**
     - **Real time applications such as multimedia.**

Service Primitives (1)

- A service is formally specified by a set of primitives (operations).
- Primitives are operations that are available to the user processes to access the service.
- The set of primitives available are different for connection-oriented services from those of connectionless service.
- Example in the next slide

Service Primitives (1)

Six service primitives that provide a simple connection-oriented service

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
ACCEPT	Accept an incoming connection from a peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

Service Primitives

- The primitives presented in the previous slide might be used for request-reply interaction in a client-server environment:

1. Server executes LISTEN to indicate that it is prepared to accept incoming connections.
   - Blocking system call.
   - The server process is blocked until a request for connection appears.
2. Client process executes CONNECT to establish a connection (1) with the server.
   - Specifies who to connect to (parameter giving the server's address).
   - OS sends a packed to the peer asking it to connect (See Figure next slide).
   - Client process is suspended until there is a response.

Service Primitives

3. The server process can establish the connection by executing ACCEPT primitive (2).
   - OS sees that the packet is requesting a connection upon reception of the packet.
   - OS checks to see if there is a listener and if so it unblocks it.
   - Sends a response back to the client process to accept the connection.
   - The arrival of this response then releases the client.
   - At this point both client and server a running and they have connection established.
4. The server will execute RECEIVE to prepare to accept the first request.
   - Server does this immediately upon being released from the LISTEN, before acknowledgment can get back to the client.
   - The RECEIVE is a blocking call.

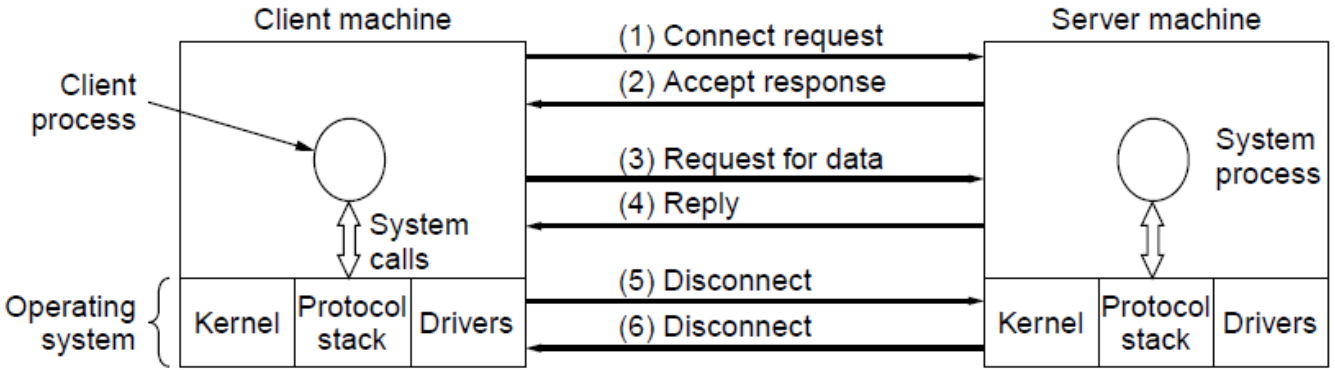
Service Primitives

5. The client will execute SEND to transmit its request (3) followed by RECEIVE to get the reply.
   - The arrival of the request packed at the Server unblocks it so it can handle the request.
   - After the server has done the work it will issue a SEND to return the answer to the client (4).
   - The arrival of the this packed unblocks the client which can now inspect the answer.
   - If further request are required it can make them now.

Service Primitives

6. When the client is done it executed DISCONNECT to terminate the connection (5).
   - Initial DICONNECT is a blocking call, suspending the client and sending a packet to the server saying that the connection is no longer needed.
   - When the server gets the packed it also issues a DISCONNECT of its own, acknowledging the client and releasing the connection (6).
   - When the server's packet gets back to the client machine, the client process is released and the connection is broken.

Service Primitives (2)



A simple client-server interaction using acknowledged datagrams.

Service Primitives

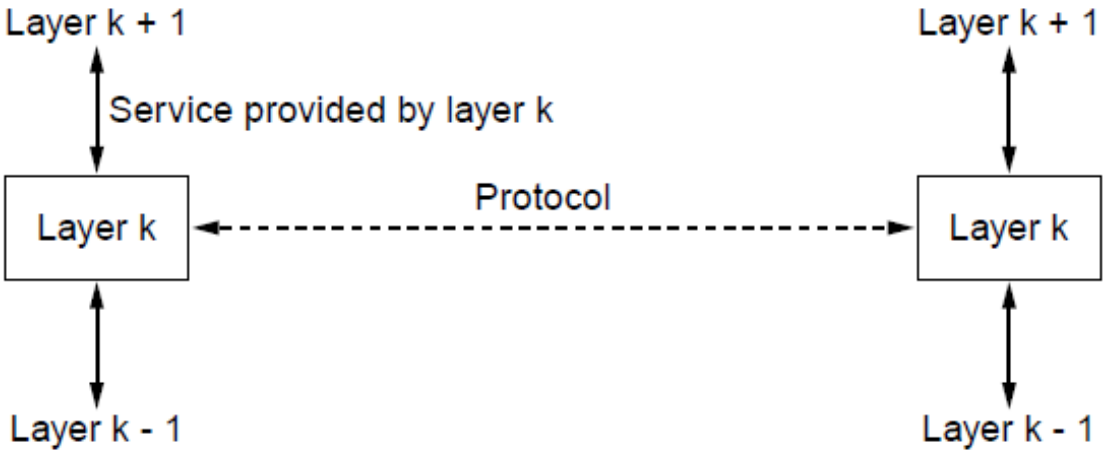
- **Many things can go wrong:**
  - **Timing (e.g., CONNECT is done before LISTEN)**
  - **Packets can get lost, …**

- **Why not using connectionless service:**
  - **Only two (2) packets would be needed vs. six (6), however,**
  - **Large messages**
  - **Transmission errors**
  - **Lost packets**
  - **Etc.**
- **Example:**
  - **How would the client know whether the last packet actually received was really the last packet sent?**

The Relationship of Services to Protocols

- A *service* is a set of primitives (operations) that a layer provides to the layer above it.
  - The service defines what operations the layer is prepared to perform on behalf of its users, but it does not say anything at all about how these operation are implemented.
- A *protocol* is a set of rules governing the format and meaning of the packets, or messages that are exchanged by the peer entities within a layer.
  - Entities use protocols to implement their service definitions.
  - They are free to change their protocols at will, provided they do not change the service visible to their users.
  - In this way the service and the protocol are completely decoupled.

The Relationship of Services to Protocols



The relationship between a service and a protocol.

The Relationship of Services to Protocols

- **Key Concept:**
  - Services relate to interfaces between layers
  - Protocols relate to the packets send between peer entities on different machines.

- **Programming Languages Analogy:**
  - Service is like an abstract data type or an object in an object-oriented language.
    - It defines operations that can be performed on an object bud does not specify how these operations are implemented.
  - Protocol relates to the *implementation* of the service and as such is not visible to the user of the service.

Reference Models

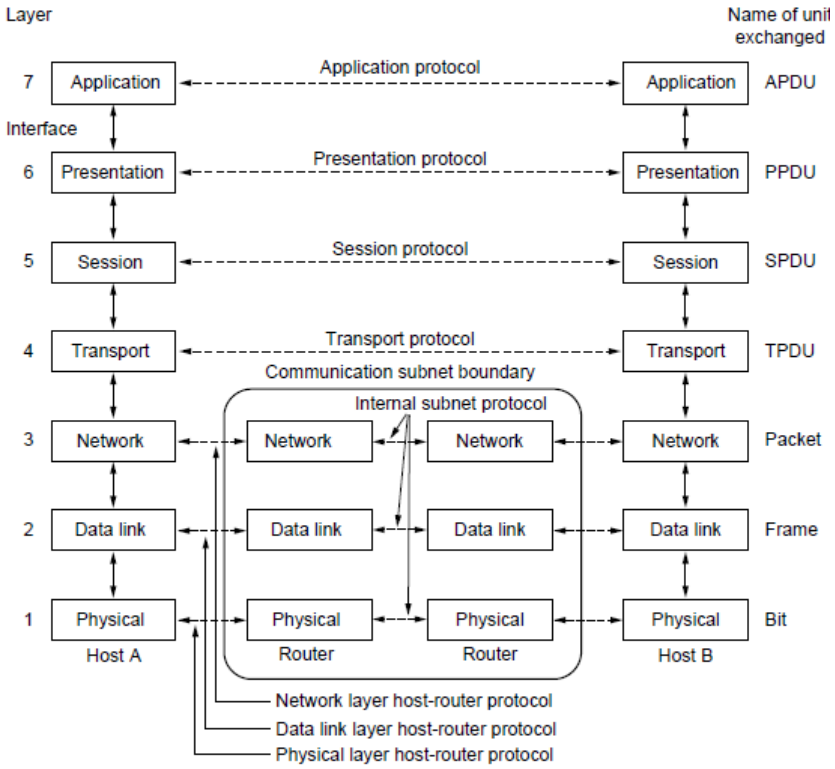
- OSI reference model
- TCP/IP reference model
- Model used for this text
- Comparison of OSI and TCP/IP
- Critique of OSI model and protocols
- Critique of TCP/IP model

The OSI Reference Model

Principles for the seven layers

- **Layers created for different abstractions**
- **Each layer performs well-defined function**
- **Function of layer chosen with definition of international standard protocols in mind**
- **Minimize information flow across interfaces between boundaries**
- **Number of layers optimum**

The OSI Reference Model



The OSI reference model

OSI Reference Model Layers

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

Physical Layer

- Is concerned with transmitting raw bits over a communication channel.
- Design Issues:
  - Ensuring that when one side sends a 1 – bit of information it is received as 1-bit (not as 0-bit or 2-or more- bits).
  - What type of signal should be used to represent “1” and “0”?
  - How many nano seconds a bit lasts?
  - Whether transmission can occur simultaneously in both direction?
  - How initial connection is being established?
  - How it is torn down when both sides are finished?
  - How many pins the network connector has?
  - What each pin is used for? Etc.

- Deals with mechanical, electrical, timing interfaces, and the physical transmission medium.

Data Link Layer

- Main task of the data link layer is to transform a raw transmission facility into a line that appears free of undetected transmission errors.
- It does this by:
  - Break up the input data into *data frames*.
  - Sequential transmission of each frame.
  - The receiver confirms correct receipt of each frame by sending back an *acknowledgment frame*.
  - How to keep a fast transmitter from drowning a slow receiver in data.
  - Some traffic mechanism may be needed to let the transmitter know when the receiver can accept more data.
  - Broadcast networks have an additional issue in the data link layer:
    - How to control access to the shared channel?
    - A special sublayer of the data link layer, called "*Medium Access Control*" sublayer, deals with this problem

Network Layer

- This layer controls the operation of the subnet.
- Key design issue is determining how packets are *routed* from source to destination.
  - Static tables are wired into the network and are rarely changed, or
  - They are changed more often dynamically to avoid failed components.
    - They can be determined at the start of each conversation (e.g., login session), or
    - They can be highly dynamic and for each packed the new routing can be established depending on the load.
- Congestion handling: If two many packets are present in the subnet at the same time, they will get in each other's way forming bottlenecks.
- Quality of Service:
  - Delay,
  - Transit time,
  - Jitter, Etc.

  are also a network layer issues.
- It is up to the network layer to overcome all the problems that occur in heterogeneous networks so that they may be interconnected.
- In broadcast networks the routing problem is simple so the network layer is often thin or even nonexistent.

Transport Layer

- The main function of Transport Layer is to:
  - Accept data from above it,
  - Split it up into smaller units if needed be,
  - Pass these to the network layer,
  - Ensure that the pieces all arrive correctly at the other end,
  - All this must be done efficiently and in a way that isolated the upper layers from the inevitable changes in the hardware technology over the course of time.
- In addition, it is charged for determining what type of service to provide to the session layer, and ultimately, to the user of the network.
- Example:
  1. Error-free point-to-point channel that delivers messages or bytes in the order in which they were send.
  2. Transporting of isolated messages with no guarantees about the order of delivery,
  3. Broadcasting of messages to multiple destination.
- Transport Layer is a true end-to-end layer; it carries data all the way form the source to the desitnation.

Session Layer

- The session layer allows users on different machines to establish *sessions* between them.
- Services:
  - *Dialog control* - Keeping track the whose turn is it to transmit,
  - *Token management* – Preventing tow parties from attempting the same critical operation simultaneously, and
  - *Synchronization* – Checkpointing long transmissions to allow them to pick up form where they left off in the event of a crash and subsequent recovery.

Presentation Layer

- This layer is concerned with the presentation of the message; that is syntax and semantics of the information transmitted.
- It deals with different internal data representations on different machines:
  - Abstract data structures,
  - Standard encoding to be used,

Application Layer

- This layer commonly contains a variety of protocols that are needed by the users.
- For example:
  - *HTTP* – Hyper Text Transfer Protocol,
  - *FTP* - File Transfer Protocol
  - *POP/SMTP* – E-mail Protocol,
  - *RSS* – Network News, etc.

The TCP/IP Reference Model Layers

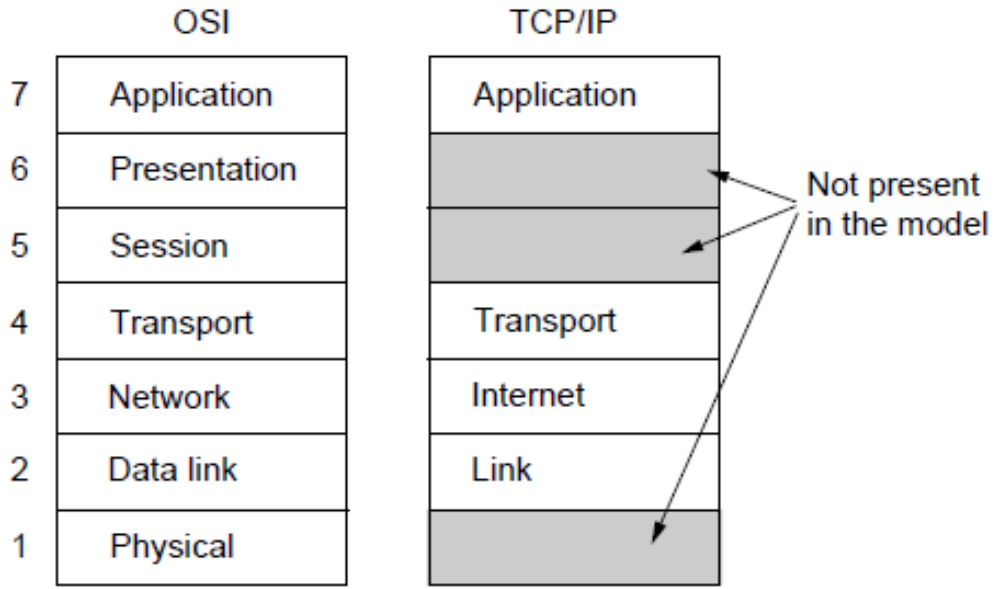
- Link layer
- Internet layer
- Transport layer
- Application layer

The TCP/IP Reference Model

- Grandparent of all wide area computer networks *ARPANET*
- It's successor Internet

- ARPANET research network sponsored by the DoD.
- Used initially leased telephone lines.
- When satellite and radio networks were included the new reference architecture was needed.
- Hence the ability to connect to multiple networks in a seamless way was one of the major design goals.
- This architecture latter became known as the *TCP/IP Reference Model*.
- Design criteria:
  - Network be able to survive loss of subnet hardware without existing conversations being broken off.
  - Applications with divergent requirements were supported ranging from file transfer to real-time speech transmission.

The TCP/IP Reference Model (1)



The TCP/IP reference model

Link Layer

- **Packet switched network**
- **Connectionless layer that runs across different networks.**
- **The lowest layer, the *link layer*, describes what links such as serial lines and classic Ethernet must do to meet the needs of this connectionless internet layer.**
- **It is not actual layer in the classical sense of the term rather is an interface between hosts and transmission links.**

Internet Layer

- The *Internet Layer* holds this architecture together.
- Its job is to permit hosts to inject packets into any network and have them travel independently to the destination (potentially on a different network).
- The packets may arrive in a completely random order from the original and the higher layer must rearrange them – if in-order of delivery is desired.

- The internet layer define san official packet format and protocol called IP (Internet Protocol).
- Packet routing is a major issue and IP has not proven effective at avoiding congestion.

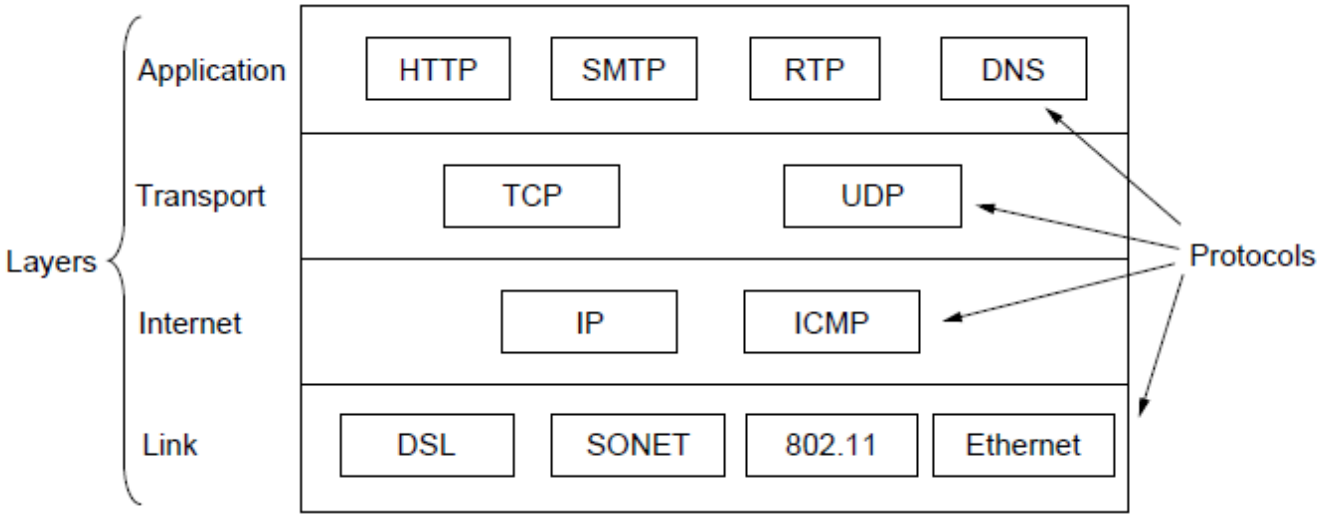
Transport Layer

- ***Transport Layer*** **is designed to allow peer entities on the source and destination hosts to carry on a conversation, similarly to the OSI transport layer.**
- **Two end-to-end transport protocols:**
  - **TCP (Transmission Control Protocol) – reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine in the internet.**
  - **UDP (User Datagram Protocol) – is unreliable connectionless protocol.**

Transport Layer

- **TCP (Transmission Control Protocol) – reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine in the internet.**
  - **It segments the incoming byte stream into discrete messages**
  - **Passes each one on to the internet layer.**
  - **At the receiver the TCP process reassembles the received messages.**
  - **Flow control is also managed by TCP to ensure that a fast sender cannot swamp a slow receiver.**
- **UDP (User Datagram Protocol) – is unreliable connectionless protocol.**
  - **For applications that do not want TCP's sequencing or flow control and they want to provide one of their own.**
  - **Widely used for one-shot, client-server-type request-reply queries and application in which prompt deliver is more important than accurate delivery.**
    - **Speech**
    - **Video**

The TCP/IP Reference Model (2)

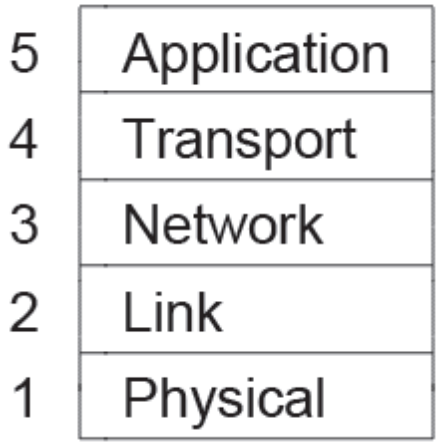


The TCP/IP reference model with some protocols we will study

Application Layer

- Applications must include any session or presentation functions that they require.
- Experience with the OSI model has proven this view to be correct: these layers are of little use to most applications.
- Application Layer contains all the higher-level protocols.
  - TELNET - Virtual Terminal
  - FTP – File Transfer Protocol
  - SMTP – electronic mail
- Many other protocols have been added (see figure in the previous slide):
  - DSN – Domain Name System
  - HTTP – Hyper Text Transfer Protocol
  - RTP – Real-time Transfer Protocol

The Model Used in this Book



The reference model used in this book.

The Model Used in this Book

- Using the 5 layers:
  - Physical
  - Link
  - Network
  - Transport, and
  - Application
- Value of OSI model is retained for understanding network architecture.
- In addition we concentrate primarily on protocols that are important in practice:
  - TCP/IP
  - 802.11
  - SONET
  - Bluetooth.

Comparison of the OSI and TCP/IP Reference Models

Concepts central to OSI model

- **Services**
- **Interfaces**
- **Protocols**

Service

- Each layer provides a service to the layer above it.
- The service definition tells what the layer does, not how entities above it access it or how the layer works.
- It defines the layer's semantics.

Interface

- A layer’s interface tells the processes above it how to access it.
- It specifies what the parameters are and what results to expect.
- This layer also says nothing about how the layer works inside.

Protocol

- A layer's protocol its is own business: it can use any protocols it wants to as long as it gets the job done (i.e. provides the offered services).
- A layer is allowed to change the protocol with the condition that it will not affect the software in higher layers.

Object Oriented Programming

- Those ideas fit very nicely with modern ideas about object-oriented programming.
- An object has:
  - A set of methods (operations) that processes outside the object can invoke.
  - A set of data (method's parameters) that defines the object.
  - The code internal to the object is its protocol and is not visible or of any of concern outside the object.
  - The object provides the set of services through object's interface.

The Properties

- TCP/IP model did not originally distinguish between:
  - Services
  - Interfaces, and
  - Protocols
- The model was retrofitted after the fact to make it more OSI-like.
- However, OSI model has a better hidden then in the TCP/IP model and can be replaced relatively easily as the technology changes.

The Properties

- **The OSI reference model was devised before the corresponding protocols were invented.**
  - **This ordering meant that the model was not biased toward one particular set of protocols: a fact that made it quite general.**
  - **The downside of this ordering was that the designers did not have much experience with the subject and did not have a good idea of which functionality to put in which layer.**

- **With TCP/IP the reverse was true: The protocols came first, and the model was really just a description of the existing protocols.**
  - **There was no problem with protocols fitting the model.**
  - **The trouble was that the model did not fit any other protocol stacks: It was not especially useful for describing other non-TCP/IP networks.**

Critique of the TCP/IP Model

- Does not distinguish clearly the concepts of services, interfaces, and protocols.
- Good Software Engineering practice requires differentiating between the specification and the implementation.
- The link layer is really not a layer at all: It is an interface between the network and data link layers. The distinction between in interface and a layer is crucial.
- TCP/IP model does not distinguish between the physical and data link layers.
  - Physical layer has to do with the transmission characteristics of the medium used (copper wire, fiber optics, wireless communication, etc.).
  - Data link layer job is to delimit the start and end of frames and get them from one side to the other with the desired degree of reliability.
- IP and TCP protocols were carefully thought out and well implemented, however, the other protocols were ad-hoc.
  - Example - TELNET designed for a ten-character-per second mechanical Teletype terminal and it does not know anything about graphical user interfaces and mice.

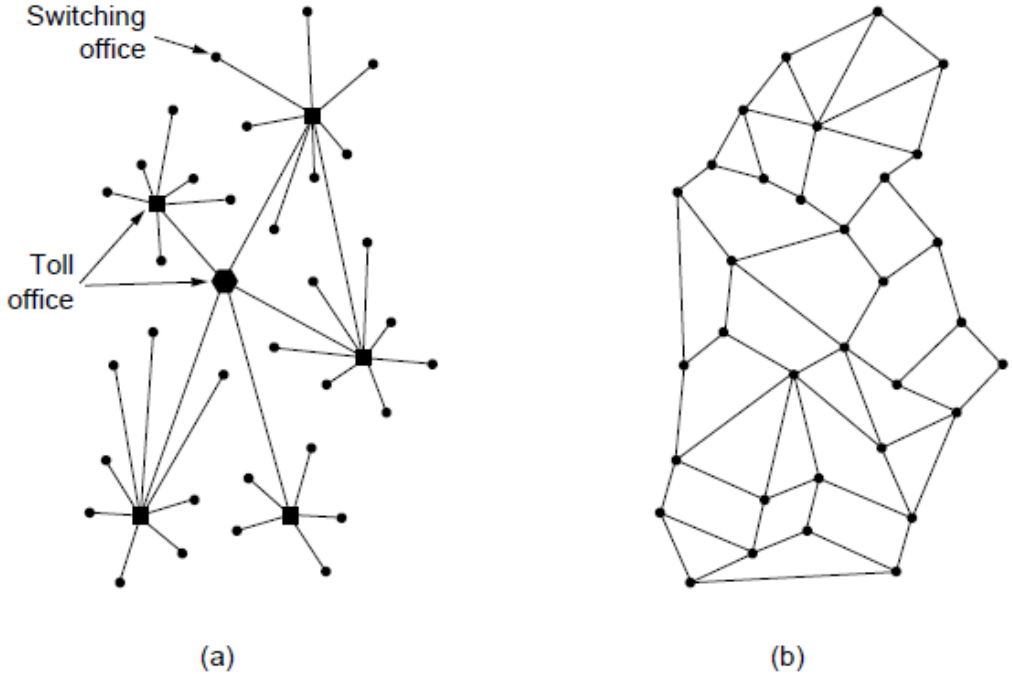
Example Networks

- Internet
- ARPANET
- NSFNET
- Third-generation mobile phone networks
- Wireless LANs: 802.11
- RFID and sensor networks

Internet

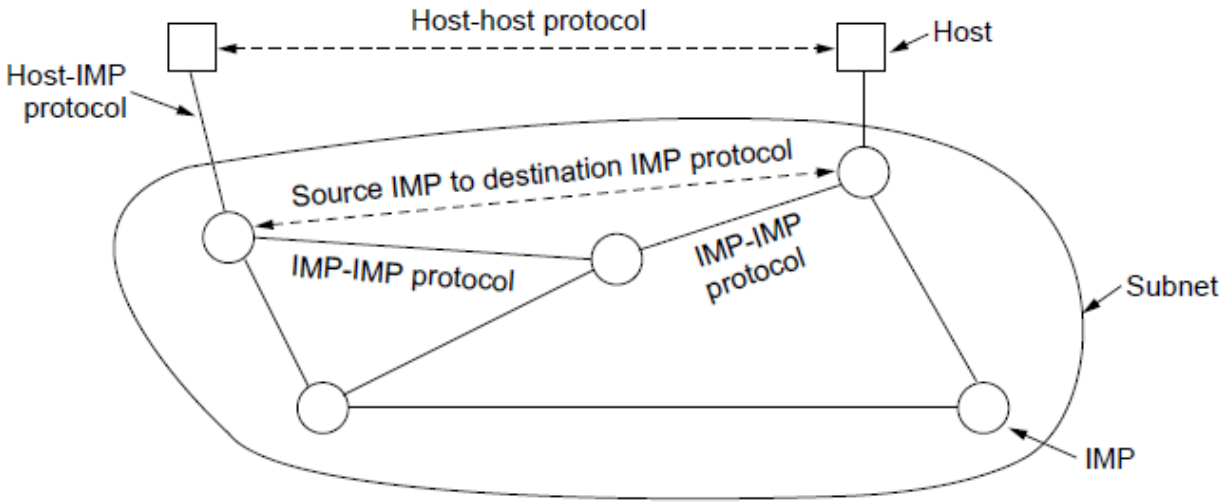
- **Is a vast collection different networks that use certain common protocols and provide certain common services.**

The ARPANET



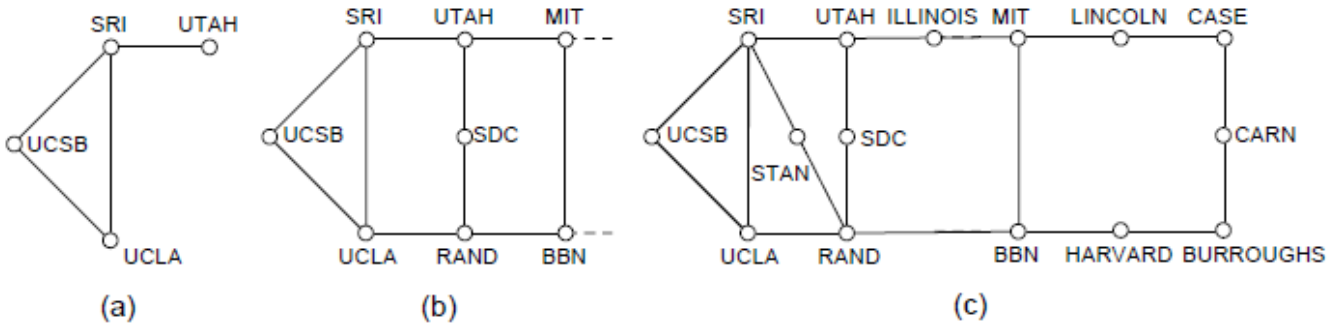
- Structure of the telephone system.
- Baran's proposed distributed switching system.

The ARPANET (2)



The original ARPANET design

The ARPANET (3)



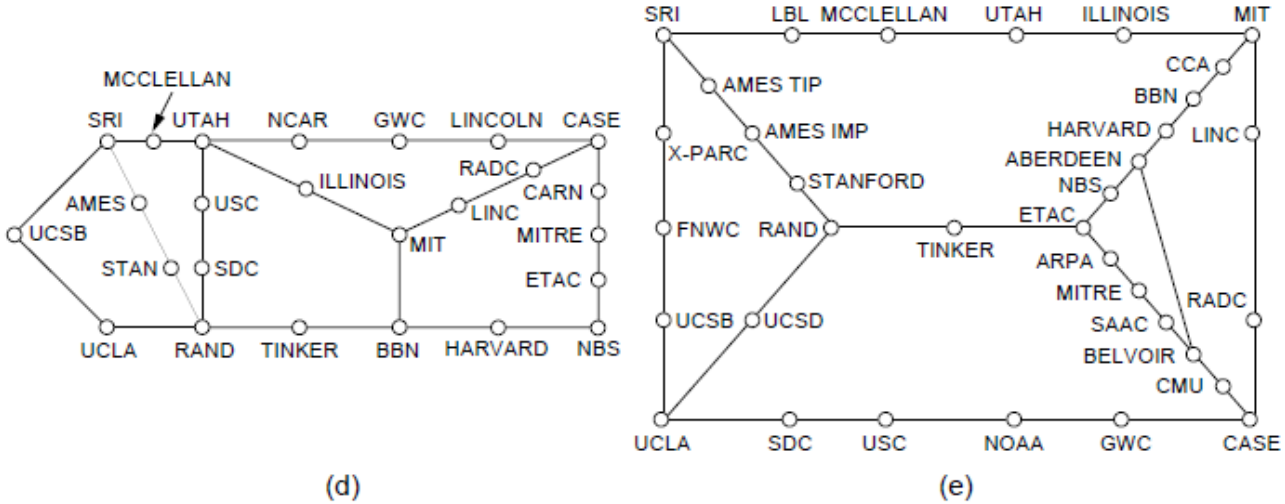
Growth of the ARPANET.

a)December 1969.

b)July 1970.

c)March 1971.

The ARPANET (4)

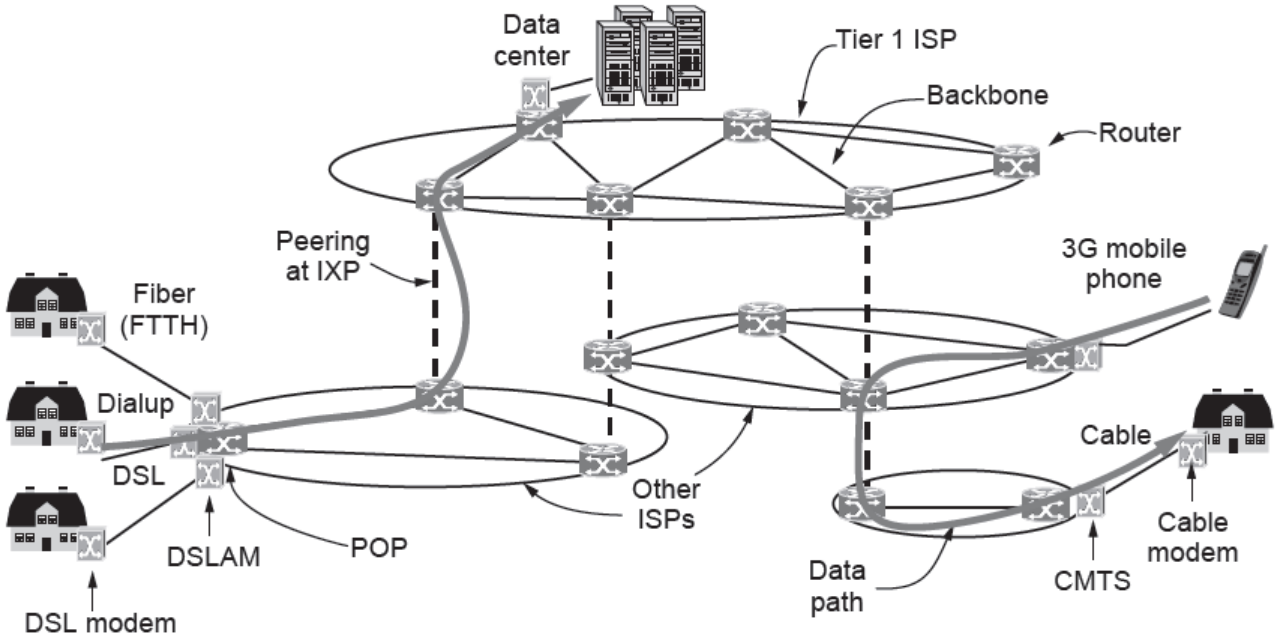


Growth of the ARPANET.

d)April 1972.

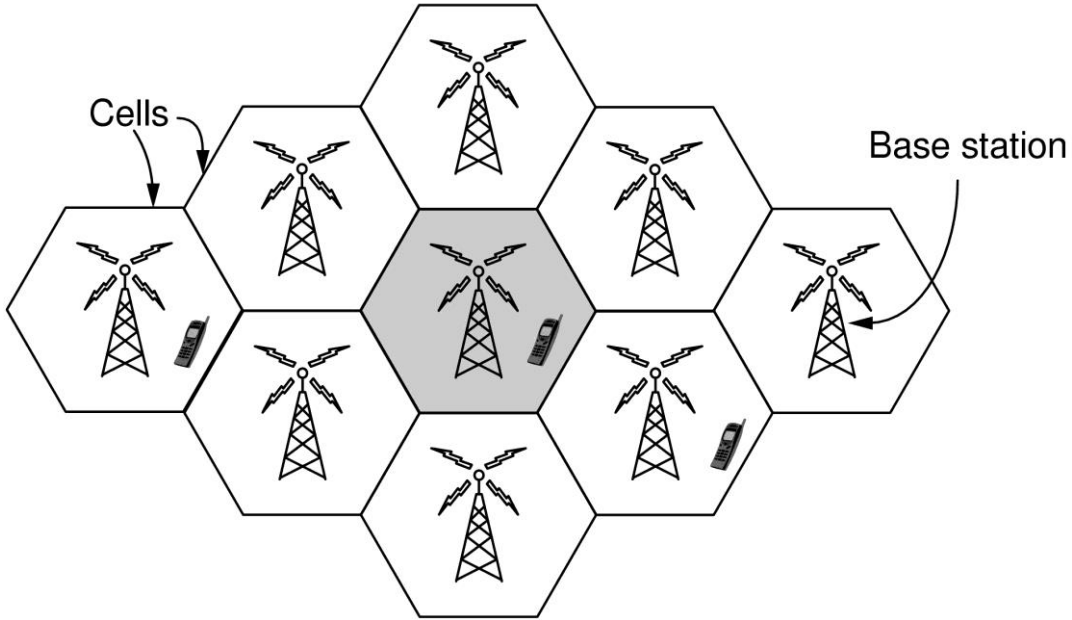
e)September 1972.

Architecture of the Internet



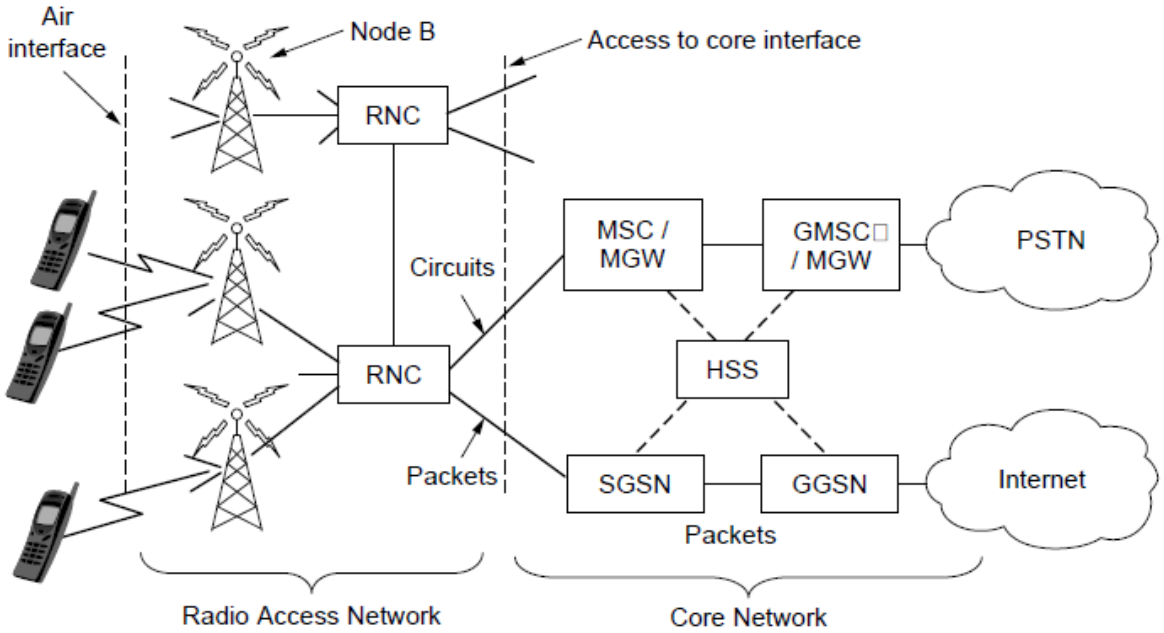
Overview of the Internet architecture

Third-Generation Mobile Phone Networks (1)



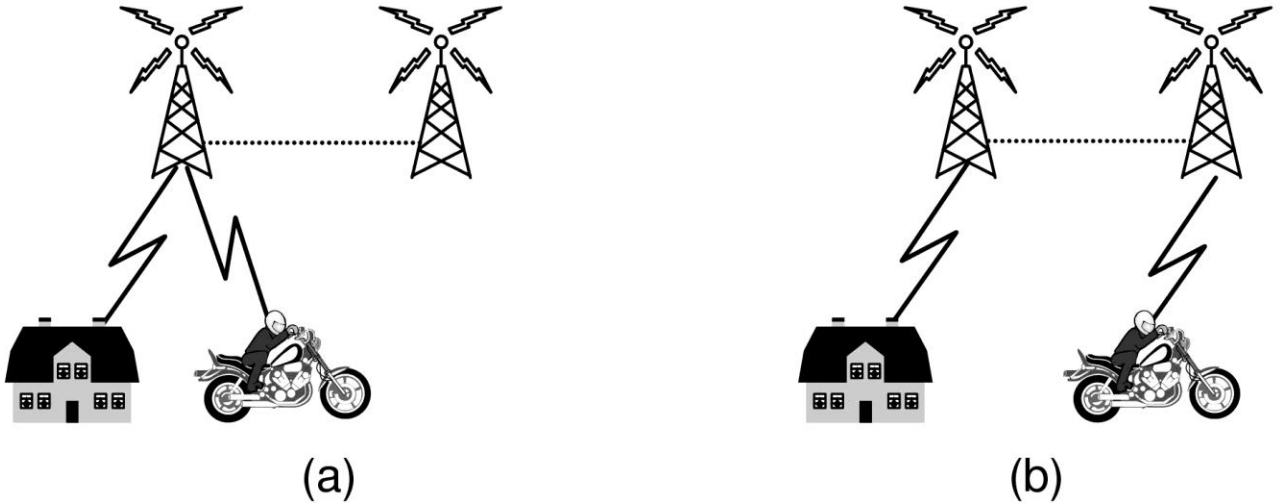
Cellular design of mobile phone networks

Third-Generation Mobile Phone Networks (2)



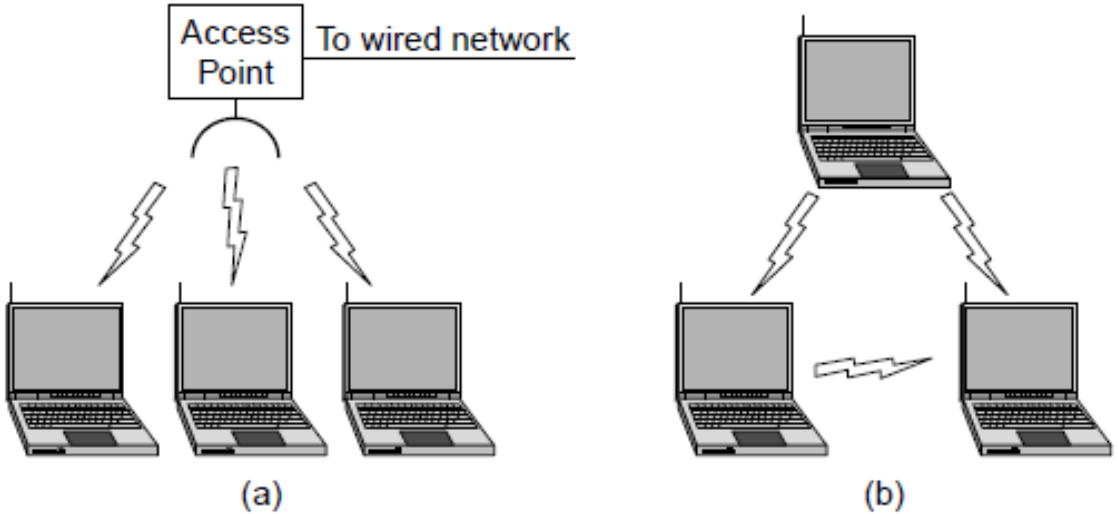
Architecture of the UMTS 3G mobile phone network.

Third-Generation Mobile Phone Networks (3)



Mobile phone handover (a) before, (b) after.

Wireless LANs: 802.11 (1)

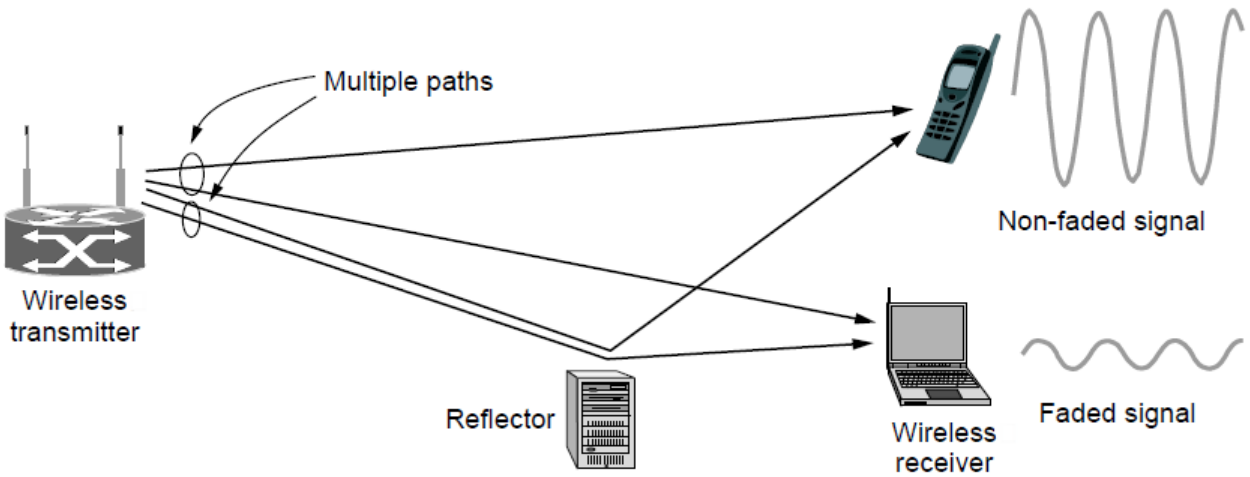


(a)Wireless network with an access point.

(b)Ad hoc network.

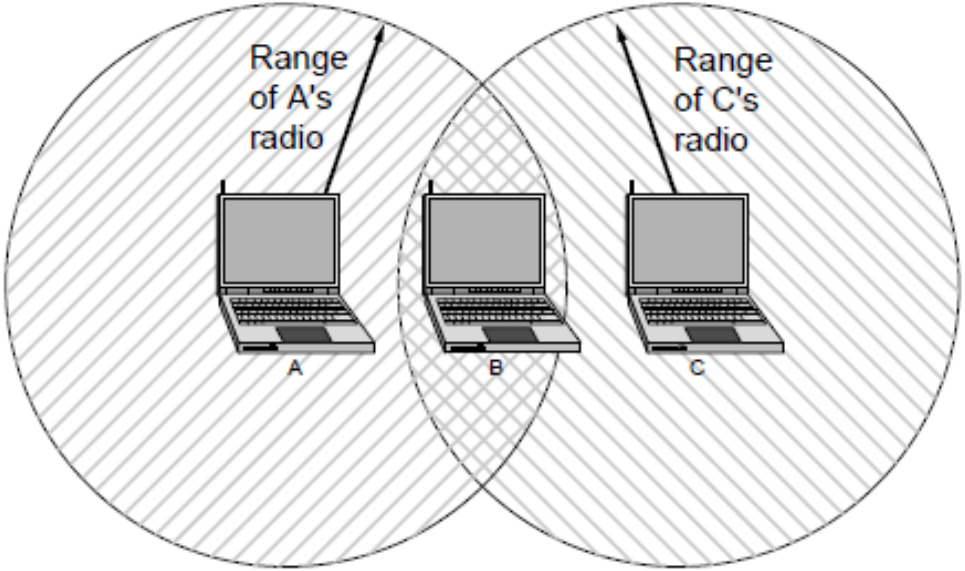
Wireless LANs: 802.11 (2)

Multipath fading

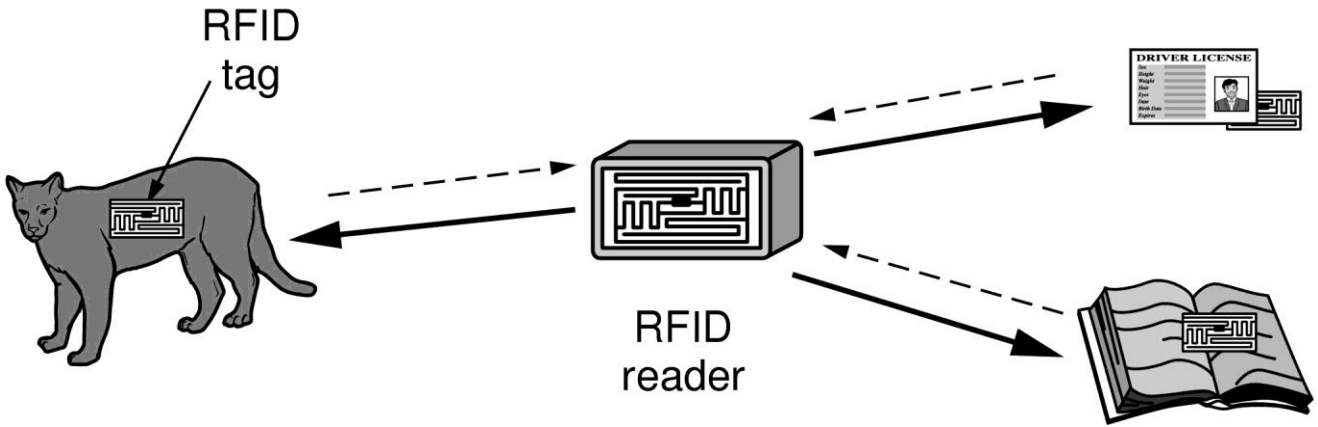


Wireless LANs: 802.11 (3)

The range of a single radio may not cover the entire

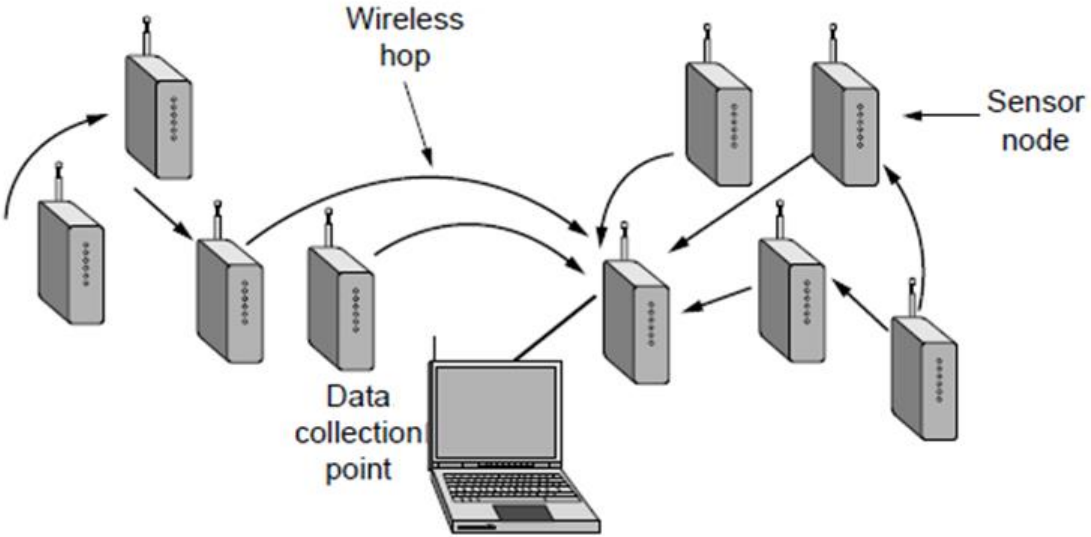


RFID and Sensor Networks (1)



RFID used to network everyday objects.

RFID and Sensor Networks (2)



Multihop topology of a sensor network

Network Standardization

- Who's Who in telecommunications
- Who's Who in international standards
- Who's Who in internet standards

International Standards (1)

Number	Topic
802.1	Overview and architecture of LANs
802.2 ↓	Logical link control
802.3 *	Ethernet
802.4 ↓	Token bus (was briefly used in manufacturing plants)
802.5	Token ring (IBM's entry into the LAN world)
802.6 ↓	Dual queue dual bus (early metropolitan area network)
802.7 ↓	Technical advisory group on broadband technologies
802.8 †	Technical advisory group on fiber optic technologies
802.9 ↓	Isochronous LANs (for real-time applications)
802.10 ↓	Virtual LANs and security
802.11 *	Wireless LANs (WiFi)
802.12 ↓	Demand priority (Hewlett-Packard's AnyLAN)

The 802 working groups. The important ones are marked with *.

The ones marked with ↓ are hibernating. The one marked with † gave up and disbanded itself.

International Standards (2)

802.13	Unlucky number; nobody wanted it
802.14 ↓	Cable modems (defunct: an industry consortium got there first)
802.15 *	Personal area networks (Bluetooth, Zigbee)
802.16 *	Broadband wireless (WiMAX)
802.17	Resilient packet ring
802.18	Technical advisory group on radio regulatory issues
802.19	Technical advisory group on coexistence of all these standards
802.20	Mobile broadband wireless (similar to 802.16e)
802.21	Media independent handoff (for roaming over technologies)
802.22	Wireless regional area network

The 802 working groups. The important ones are marked with *.

The ones marked with ↓ are hibernating. The one marked with † gave up and disbanded itself.

End

Chapter 1