Lecture 2
Plan: finish 1.4-1.7

Review:
Computer Networks: Definition (interconnected computers with a single technology. Ask about

- Shared memory multiprocessors
- A central computer with time-sharing terminals

Network Hardware: very diverse → leads to problems in the software phase.

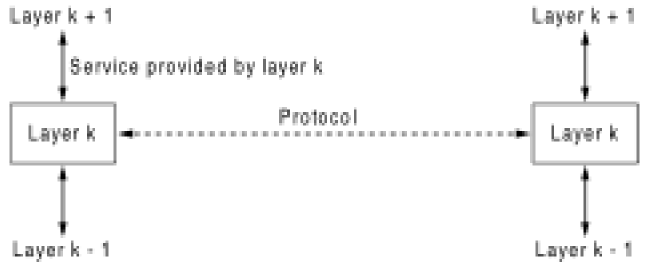
Network software: Due to complexity in hardware and data communication, protocol stacks are proposed.
e.g. transfer a file between two hosts.
Q:/ What is needed?
A:/ a data path between the two computers.

Q:/ What else do we have to be careful?
A:/
the source system must be able to tell the path that he is ready to send info;
the source system must be certain that the destination system is ready;
the application on the source system who will send the file has to be certain that the receiving application is ready
the destination application has to be sure that it’s got the correct data.
’’’’

Thus instead of single module that handles all this, the whole task is broken into small subtasks → layered

One note about the difference between service and protocol
Each layer has its own protocol (how the layer performs its task), a service to the layer above it (the operations that the layer can do for the layer above it), and an interface (how the layer above can access this layer)

Similar to objects in OO
Protocol: the implementation of the member functions
Services: the job the member functions can do
Interface: how to call those member functions



Two reference models: OSI and TCP/IP

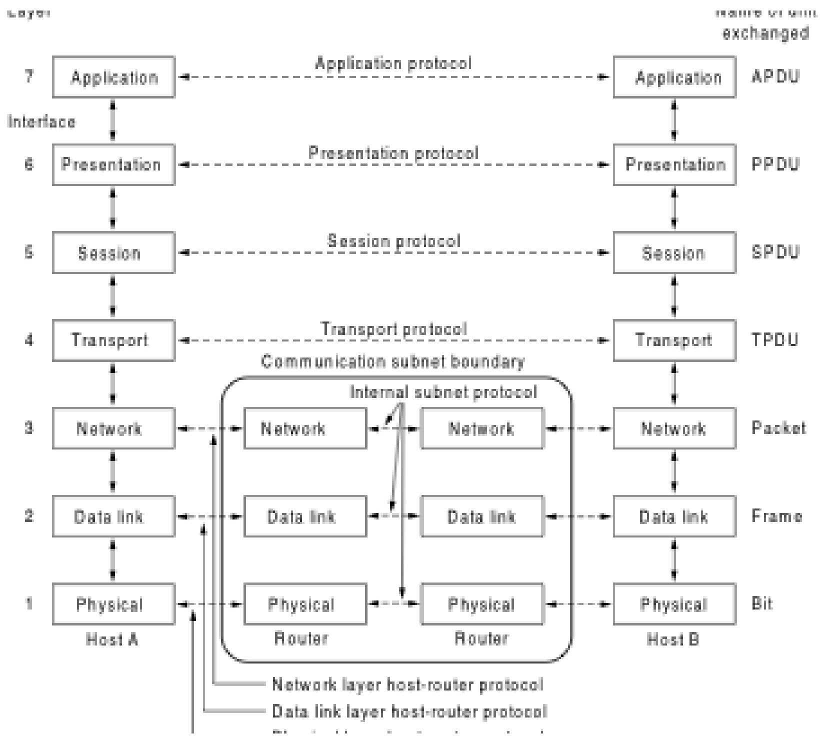
OSI (open system interconnectio

- Later than TCP/IP, bad timing
- not strictly followed by any system

- useful as a conceptual model
- its terminology is still useful

The OSI model prescribes seven layers:

1. Application
2. Presentation
3. Session
4. Transport
5. Network
6. Data link
7. Physical



We'll talk briefly about each one.

Physical layer: deals with *bits*

- How to transmit a sequence of bits along some physical medium (e.g., wire, fiber, microwave)
- A different standard is needed for each type of media.
- For example, are 0's and 1's represented by differences in voltages, frequencies, or what?
- How long (in time) is each bit? duplex or one way

Data link layer: deals with *frames*

- Give some structure to a bit stream
- Some error checking may be done here
- Synchronization between sender and receiver may be done here

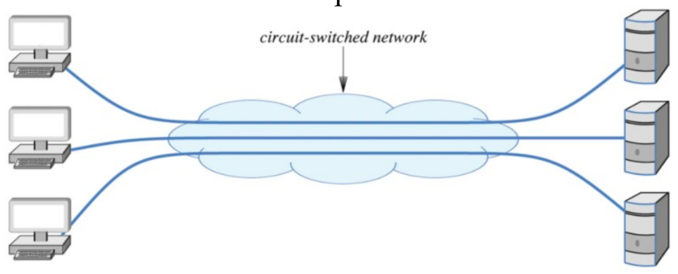
Q:/ Is it good to send all information in a bit stream all in one big stream?
A:/ No. It may occupy the network for a long time and may have to be re-transmitted in whole if the data is damaged.

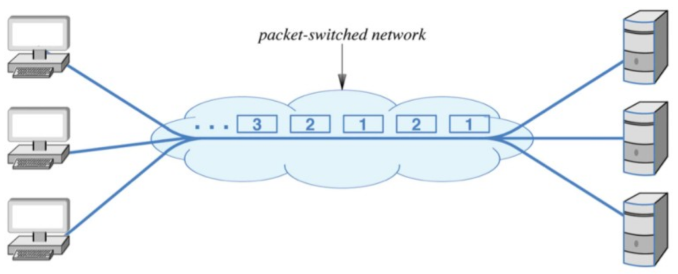
To do this, divide the bit stream into frames of say, 1000 bytes each. If there is a problem (an error or lost frame) the damage is limited, and we can resynchronized.
Q:/ What is involved?

- Coding schemes (with redundancy) so errors can be detected
- Acknowledgements from receiver to sender so that the sender knows if a frame arrived correctly
- Flow control -- so a slow receiver can tell a fast sender to slow down
- Access control for broadcast networks (MAC)

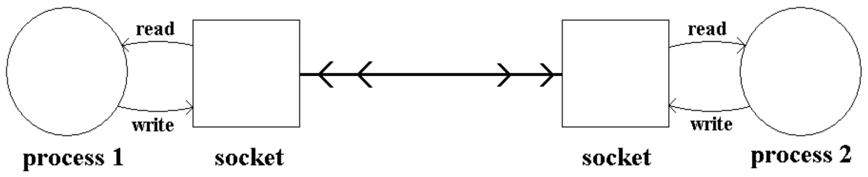
These issues involved in both point-to-point and broadcast networks
Network layer: deals with *packet*
- A packet is a short message.
- The biggest issue here is routing
- A packet may have to go through many intermediate nodes before arriving at its destination
- "store and forward" – Packet switching (to avoid the monopoly of the transmission network by a large block of data, talk about circuit switching, frame switching, pp222 comer)

circuit switching originates from telephone company and it establishes a path between sender and receiver with guaranteed isolation from paths used by other transmissions. (i.e. isolated real or virtual paths). The path is setup before transmission starts and is destroyed when transmission is over.
Packet switching uses a different idea. Senders will break their information into packets and sent them to the network. The packets from different senders will compete against each other.





ransport layer: deals with *datagrams*, *streams*
The objective is to provide a more transparent level of service, for use by applications. This is the true end-end layer where sender and receiver talk to each other directly. The lower layers talk to the neighbors in the transmission route.
example Socket is an abstraction which forms the interface from transport layer to applications. Sockets may support TCP (stream) or UDP (datagram) service.
A stream socket allows an application to write a sequence of bytes to the socket, which can be read by an application on the other side.



Issues

- connection-oriented vs. connectionless → connection means a logic association between two entities.
  - In a connection-oriented service, once a connection is made, each communication is separate; it can be used
  - Connectionless is faster (a small header) and the connection-oriented is more reliable.
- reliable vs. unreliable
  - reliable ==> error-checking
  - data is guaranteed to arrive in correct order without errors or duplication
  - examples
    - file transfer, telnet, email, ecommerce (reliable)
    - streaming video, video conferencing (unreliable)

Session layer: deals with sessions
A session between processes on different machines
May provide basic transport + enhanced services
example: When a session is terminated, all processes during the session are terminated.
example: Establish checkpoints, so that in case of a crash, an operation can be resumed where it left off instead of starting over
Presentation layer
example: Data formatting for basic data types, data structures
Application layer
Application-specific protocols, such as telnet, ftp, smtp, http, dns

TCP/IP reference model
The basis for the Internet
Evolved from ARPANET -- Advanced Research Projects Agency Network in 1969. A way for various Department of Defense installations to communicate, which could survive a calamity like a nuclear war.

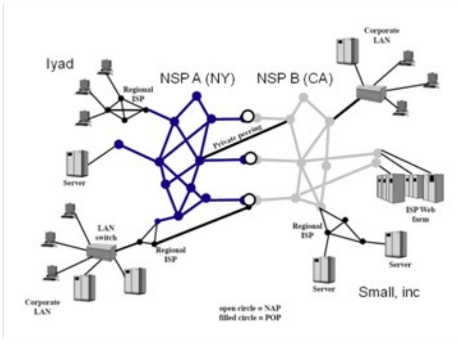
ARPANET was so successful that SATNET, RADNET are proposed. Due to the different technologies used in those networks, TCP/IP was proposed to handle internetworks.
Critical design ideas:
- Packet switching
- Host computers + subnet structure
  ==> Provide alternative routes between hosts through the subnet. Rerouting possible if some points were damaged

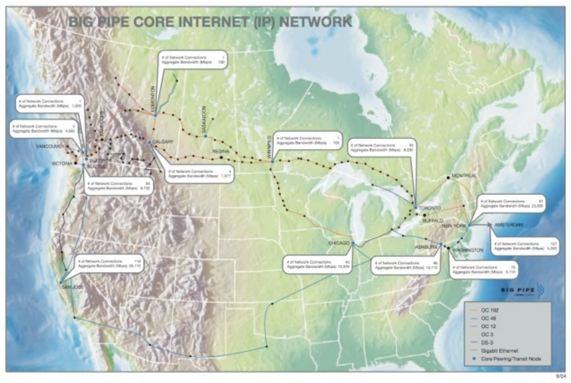
Q:/ How do we go online?

A:/

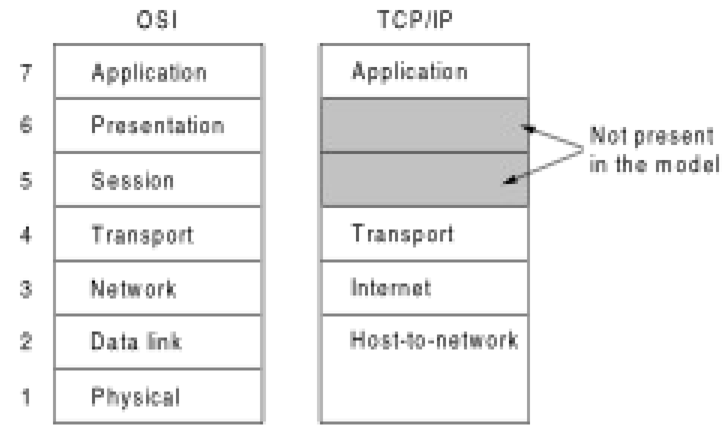
Terminologies:
POP: point of presence of the ISP
CPE: consumer premises equipment
NAP: Network Access Point
NSP: Network Service provider (provide backbone services to ISP)

Example: How does Iyad access the host of small inc.





TCP/IP Protocols



Host-to-network layer
Includes physical layer and the network access layer
The physical layer is similar to the OSI physical layer
The network access layer deals with issues inside the network that the host resides in.

Internet layer

- packet-switched
- connectionless -- each packet is handled independently (may come out of order at the destination)
- IP (Internet protocol) defines the packet format

The job of the internet layer is to deliver packets to the correct address on internetworks. It is very similar to regularly snail mail in real life. (think about how to send an international letter using snail mail)

Major concern: routing

- IPv4 uses 32-bit IP addresses
- IPv6 uses 128-bit IP addresses

Transport layer
Goal: allow a "conversation" between peer processes at each end (source and destination)

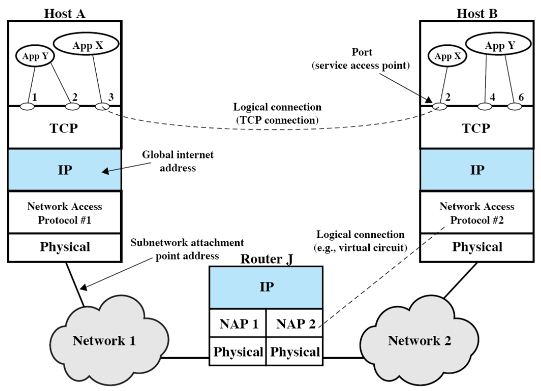
TCP -- Transmission Control Protocol

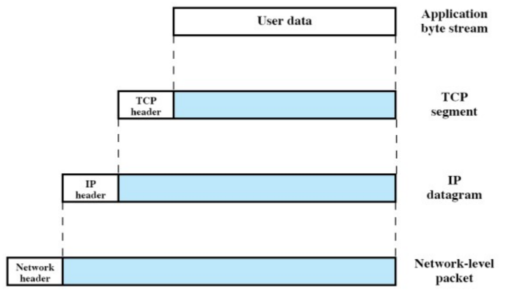
- reliable, connection-oriented
- interface to TCP is a byte stream
- TCP breaks the stream into chunks and sends each one through the internet layer, which takes it and constructs an IP packet

UDP -- User Datagram Protocol

- unreliable, connectionless
- each datagram corresponds to a single IP packet
- used when prompt delivery is more important than accurate delivery
    - o games
    - o video conferencing
    - o streaming audio/video

TCP connection example
pp36, Stalling
How does App X in host A send info to App X in host B?

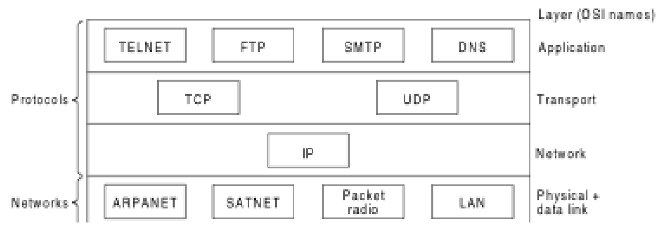

Host A
App Y
App X
1
2
3
TCP
IP
Network Access Protocol #1
Physical
Global internet address
Subnetwork attachment point address
Logical connection (TCP connection)
Port (service access point)
Host B
App X
App Y
2
4
6
TCP
IP
Network Access Protocol #2
Physical
Logical connection (e.g., virtual circuit)
Router J
IP
NAP 1
NAP 2
Physical
Physical
Network 1
Network 2



Ports: unique address associated with a process on a host. IP doesn't need to know the port number
Header: at each layer, something extra is added (error control, address, flow control, sequence number, etc)
Transport header includes destination port, sequence number, and checksum
IP datagram header includes at least host IP address

Application layer
All higher level protocols

ftp
telnet
dns -- Domain Name Service
http -- the web



OSI vs. TCP/IP
Common: layered protocol structure

OSI
The OSI model provides a good way of abstractly viewing network protocols, their structure and functions.

- service
- interface
- protocol

The model was proposed first before protocols were designed.

TCP/IP
TCP/IP is intended as a heterogeneous *inter*networking protocol, so it assumes that lower levels are handled by a local network (i.e., somebody else's problem)
TCP/IP describes what is really in use.

Tanenbaum suggests using a hybrid model, consisting of 5 layers:

Application
Transport
Network
Data link -- LAN protocols
Physical -- communication media

If time allows. (skipped)

A word about middleware
Middleware is software lying between network software and application software, used to support applications. Some examples:

COM -- Component object model (Microsoft)
CORBA -- Common Object Request Broker Architecture
RPC -- Remote Procedure Call
RMI -- Remote Method Invocation (Java)

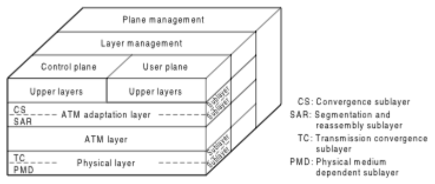
Other network technologies
TCP/IP is not the only important network technology. Many others exist. How did network services evolve?

The early days of networking saw many vendor-specific technologies, such as
- IBM Token Ring
- Novell Netware
- AppleTalk
- Banyan vines
- Ethernet
- FDDI
- ATM

Good ones became industry standards.

ATM
One example of an alternative high-speed network is ATM (Asynchronous Transfer Mode)
- Fast cell-switching (cell = packet but very small (53 bytes))
- Connection-oriented
- 155 - 622 - 2488 - 9953 Mbps (155Mbps is good for HDTV signals)
- Developed by a consortium of telephone companies and computer vendors
- Objective: Support new services like video on demand

ATM Reference Model



3-d model
- 3 planes:
  - User plane - user data
  - Control plane - connection control
  - Management plane
- 3 layers
  - AAL - ATM Adaptation Layer
    - CS - Convergence sublayer
    - SAR - Segmentation and reassembly
  - ATM - handles

        - Flow control
        - Cell header operation/extraction
        - cell multiplex/demultiplex
        - virtual circuit management
    - Physical - two sublayers
        - TC - transmission convergence
        - PM - physical medium