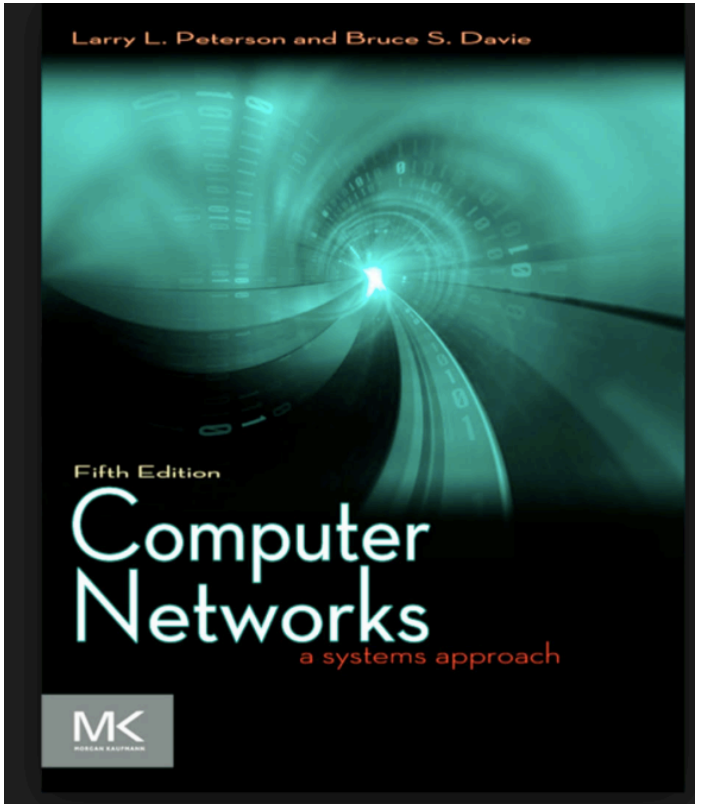
Computer Network

Introduction

Textbook

- Computer Networks, a systems approach
  - Larry L. Peterson and Bruce S. Davie
  - Fifth edition
  - A total of 9 chapters
    - In this course, we will NOT be covering some of the advanced networking topics covered in the text book.

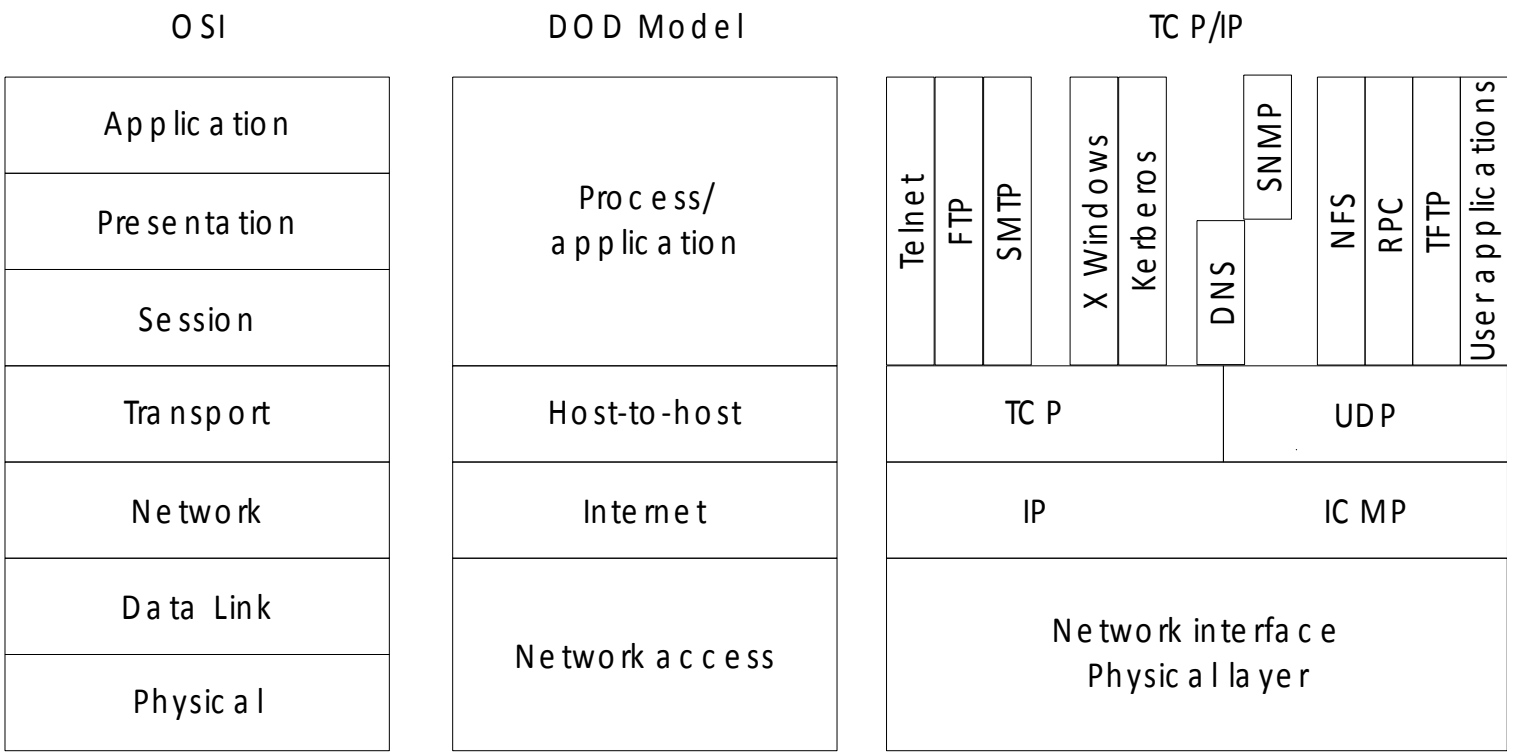


Textbook Chapters

- Chapter 1 - An Overview of network architecture

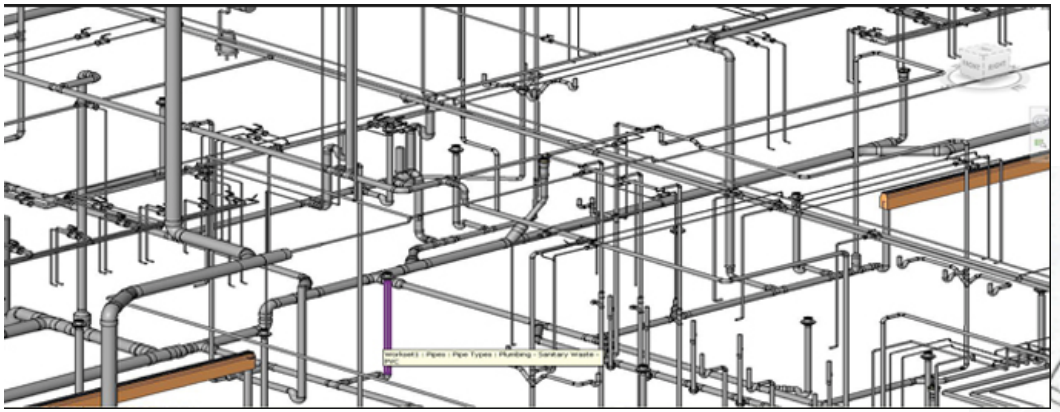
Textbook Chapters

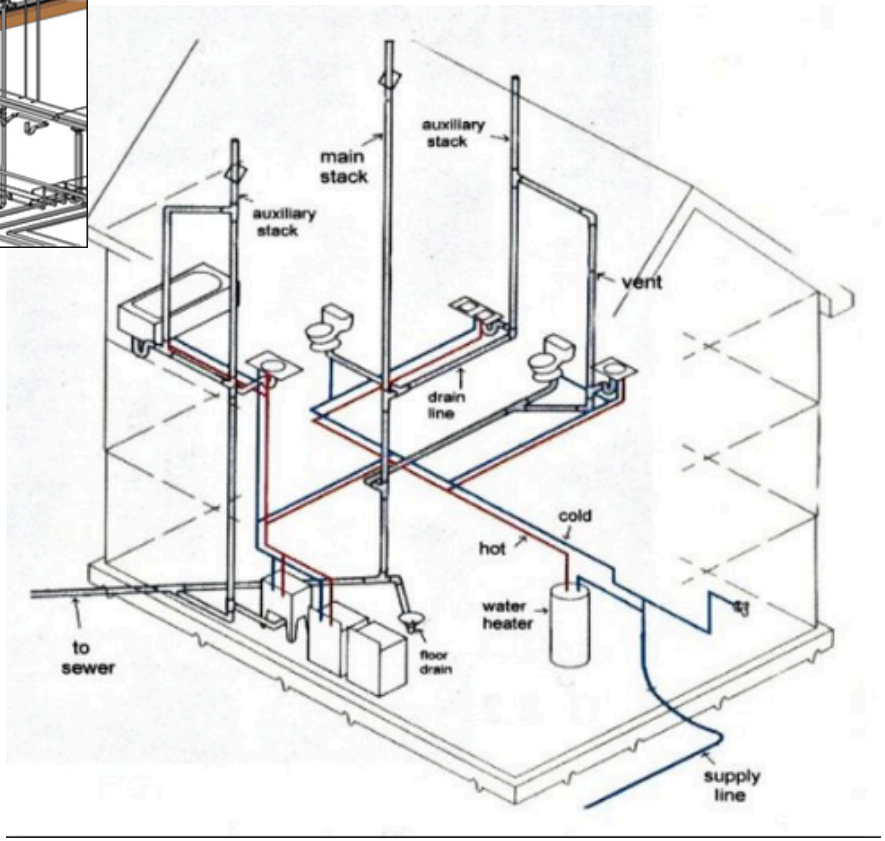
- Network architecture



Textbook Chapters

- Chapter 2 – Introduces the concepts of physical and data link.





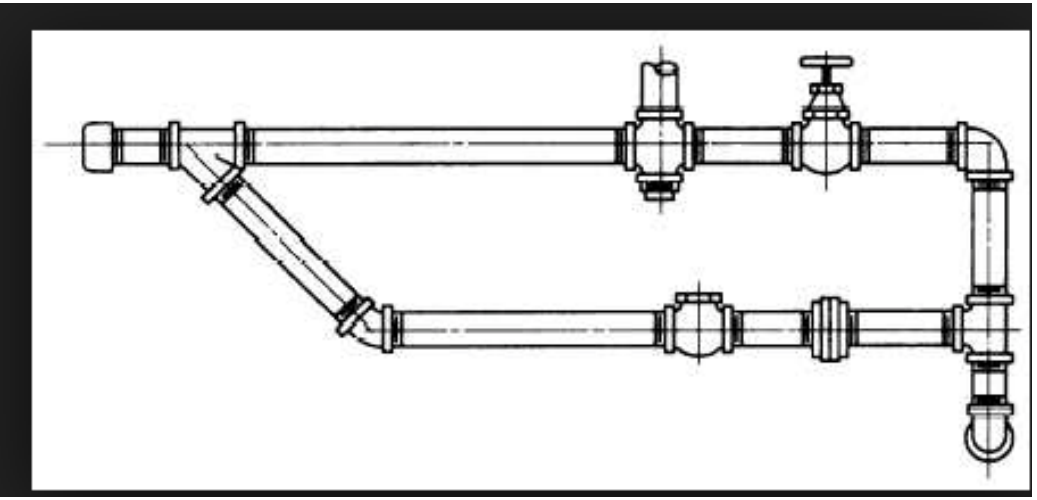
Textbook Chapters

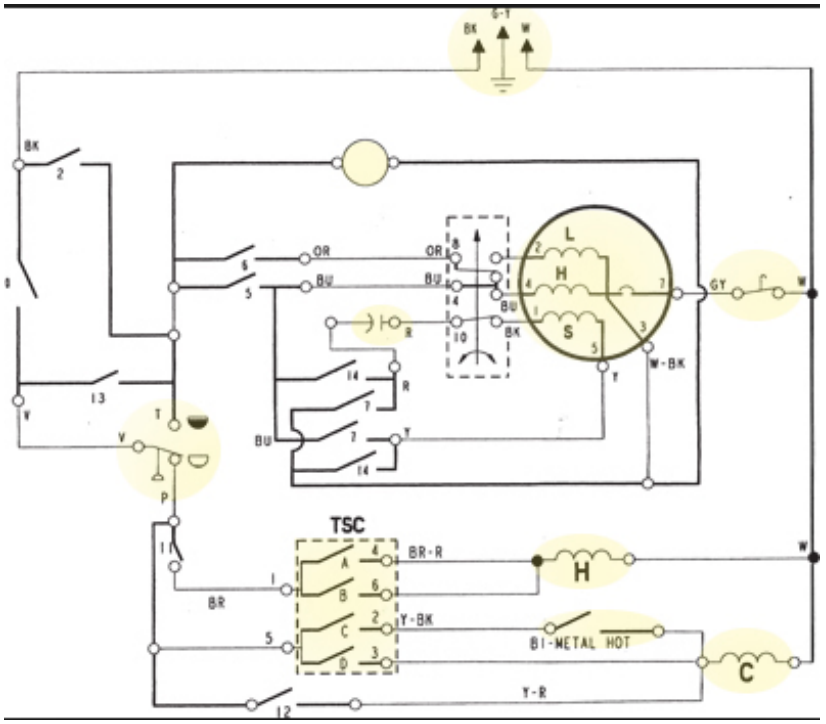
- Physical and data link.

OSI Layer	Function
Application	
Presentation	
Session	
Transport	
Network	
Data Link	Access control, data flow, framing, device ID, error detection
Physical	Electrical & physical interface, topology, encoding

Textbook Chapters

- Chapter 3 – Covers basic concepts of switching and routing





Textbook Chapters

- Switching and routing

OSI Layer	Function
Application	
Presentation	
Session	
Transport	
Network	Network ID, switching & routing
Data Link	Access control, data flow, framing, device ID, error detection
Physical	Electrical & physical interface, topology, encoding

Textbook Chapters (continue)

- Chapter 4 – Covers advanced Internetworking topics (still at layer 3 or the network layer)
  - Intra-domain routing protocols (chapter 3)
  - Multi-domain routing protocols
  - Interdomain routing
    - BGP (Border Domain Protocol)
  - IPv6

Textbook Chapters (continue)

- Chapter 5 – End-to-End Protocols (Transport layer)
  * Transmission Control Protocol (TCP)

OSI Layer	Function
Application	
Presentation	
Session	
Transport	Other (guaranteed delivery)
Network	Network ID, switching &routing
Data Link	Access control, data flow, framing, device ID, error detection
Physical	Electrical & physical interface, topology, encoding

Textbook Chapters

- Three highest layers in the OSI model

OSI Layer	Function
Application	User interface, communication apps
Presentation	Character sets, encryption, compression
Session	Maintain connection, login, upper layer errors
Transport	Other (guaranteed delivery)
Network	Network ID, routing
Data Link	Access control, data flow, framing, device ID, error detection
Physical	Electrical & physical interface, topology, encoding

Application Layer

Textbook Chapters (continue)

- Chapter 6 – Discusses congestion control and resource allocation (involves multiple layers)
  - How congestion control works in TCP
  - Congestion-Avoidance Mechanisms
  - Providing Quality of Service in IP (QoS)

Textbook Chapters (continue)

- Chapter 8 – Network Security, cryptographic tools, key distribution issues, authentication protocols (Diffie-Hellman key agreement)
  - Building of security systems using examples such as
    - Pretty Good Privacy (PGP)
    - Secure Shell (SSH)
    - IP security architecture (IPSEC).
  - Firewalls

Computer Network

Chapter 1

Foundation

Application

- Definition: An application is a computer program designed to help people perform an activity. (Wikipedia)
- Examples: World Wide Web, email, instant messaging, online social networking, streaming audio/video, etc.
- All the above examples of applications need a network to run

Addressing a Few Questions

- How to build a scalable network that will support different applications?
  - What is a (computer) network?
  - How is a computer network different from other types of networks?
  - What is a computer network architecture?

Network

- Definition:

net·work

/ˈnetˌwərk/

noun

1. an arrangement of intersecting horizontal and vertical lines.
   synonyms: web, lattice, net, matrix, mesh, crisscross, grid, reticulum, reticulation; plexus
   "a network of arteries"

2. a group or system of interconnected people or things.
   "a trade network"
   synonyms: system, complex, nexus, web, webwork
   "a network of friends"

Compute Network

- Definition:

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikimedia Shop

Interaction
Help

Article Talk Read Edit View

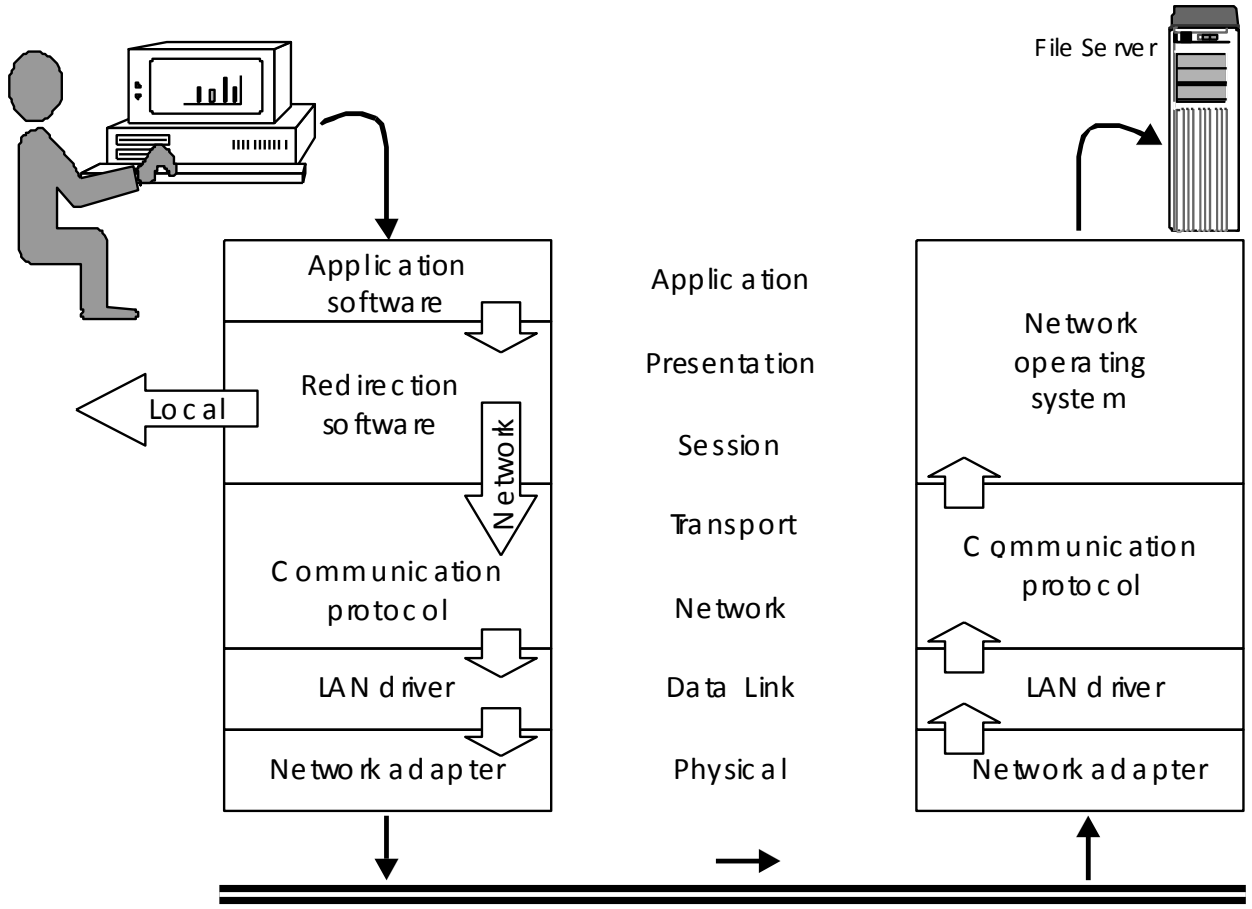
Computer network

From Wikipedia, the free encyclopedia

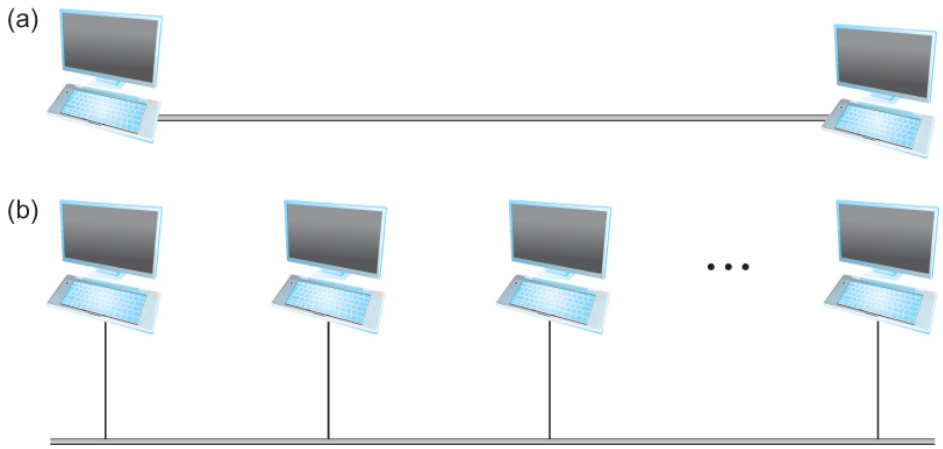
A **computer network** or **data network** is a telecommunications network that allows computers to exchange data. In computer networks, networked computing devices pass data to each other along data connections. Data is transferred in the form of packets. The connections (network links) between nodes are established using either cable media or wireless media. The best-known computer network is the Internet.

Network computer devices that originate, route and terminate the data are called network nodes.[1] Nodes can include hosts such as personal computers, phones, servers as well as networking hardware. Two such devices are said to be networked together when one device is able to exchange information with the other device, whether or not they have a direct connection to each other.

Network Connectivity



Connectivity

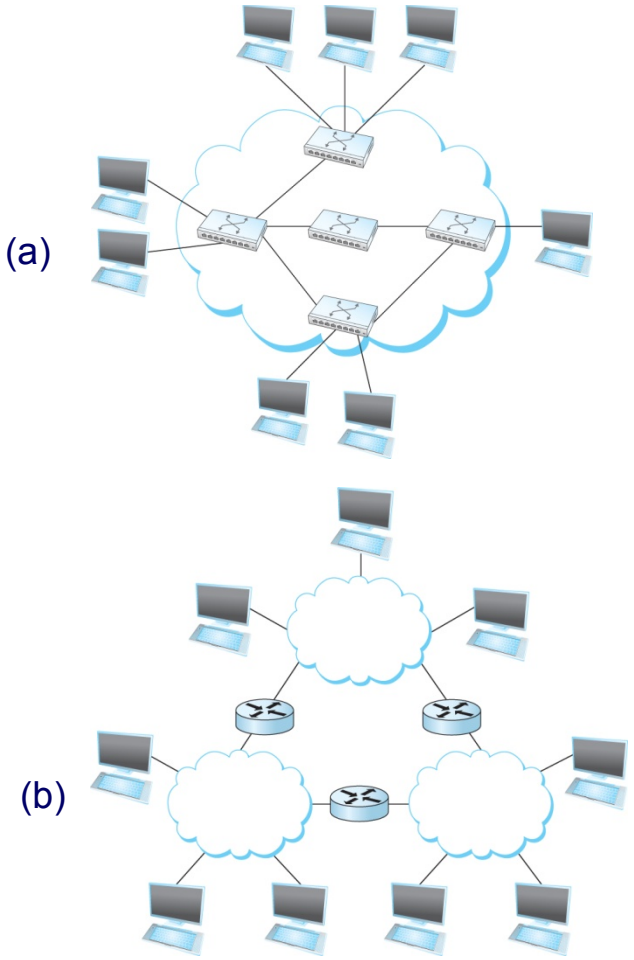


- Need to understand the following terminologies
  - Scale
  - Link
  - Nodes
  - Point-to-point
  - Multiple access
  - Switched Network
    - Circuit Switched
    - Packet Switched
  - Packet, message
  - Store-and-forward

(a) Point-to-point

(b) Multiple access

Connectivity

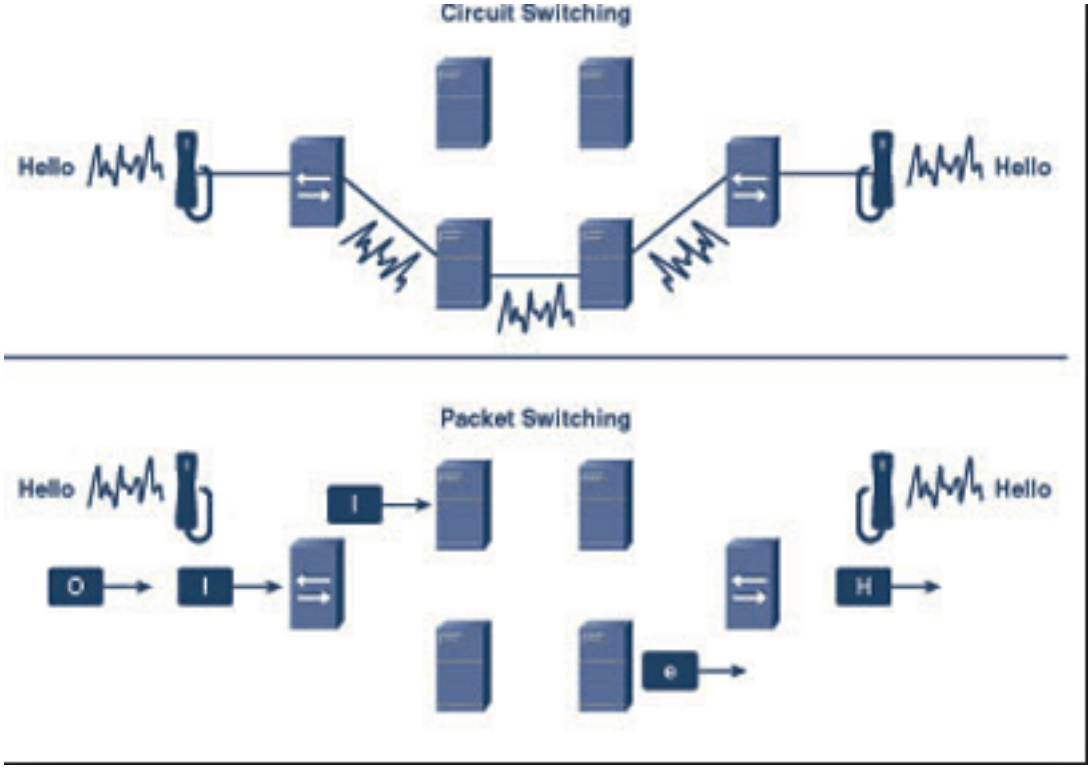


(a) A switched network

(b) Interconnection of networks

- Terminologies (contd.)
  - Cloud
  - Hosts
  - Switches
  - internetwork
  - Router/gateway
  - Host-to-host connectivity
  - Address
  - Routing
  - Unicast/broadcast/multicast

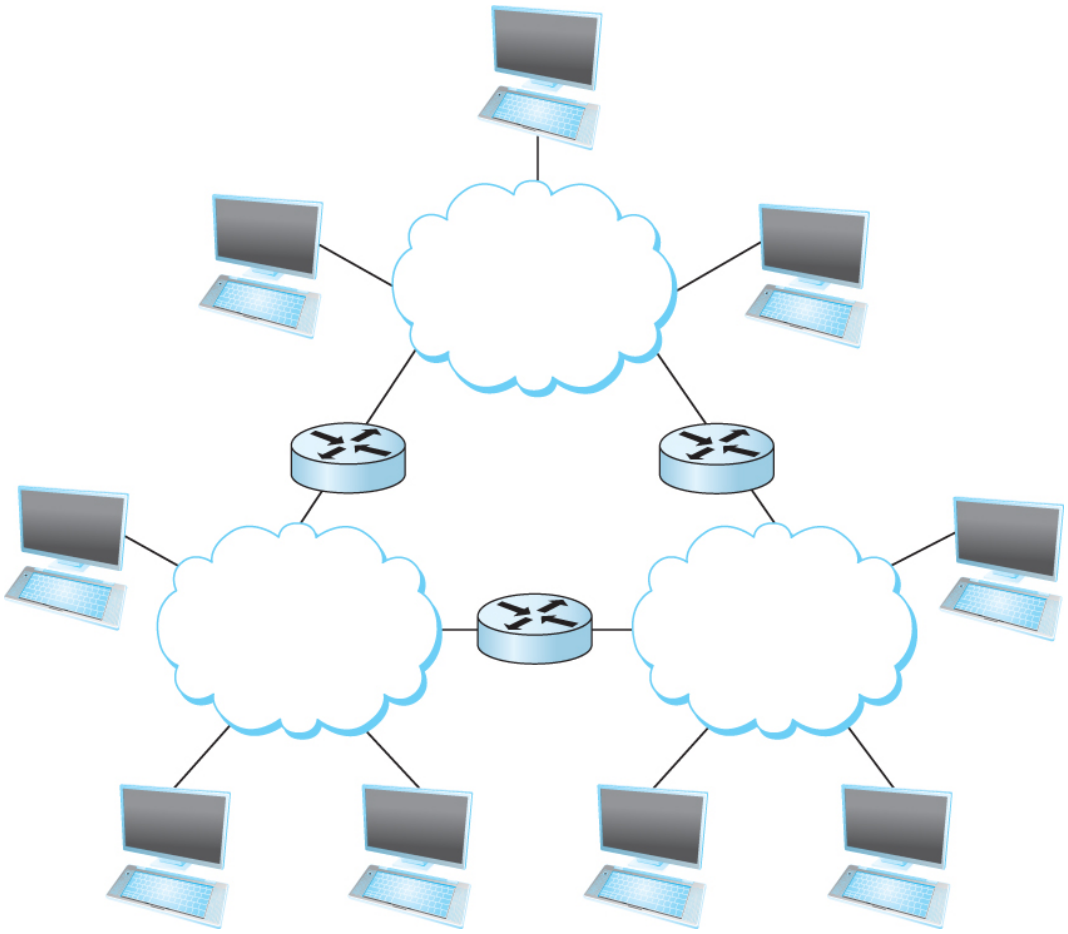
Circuit vs. Packet Switching



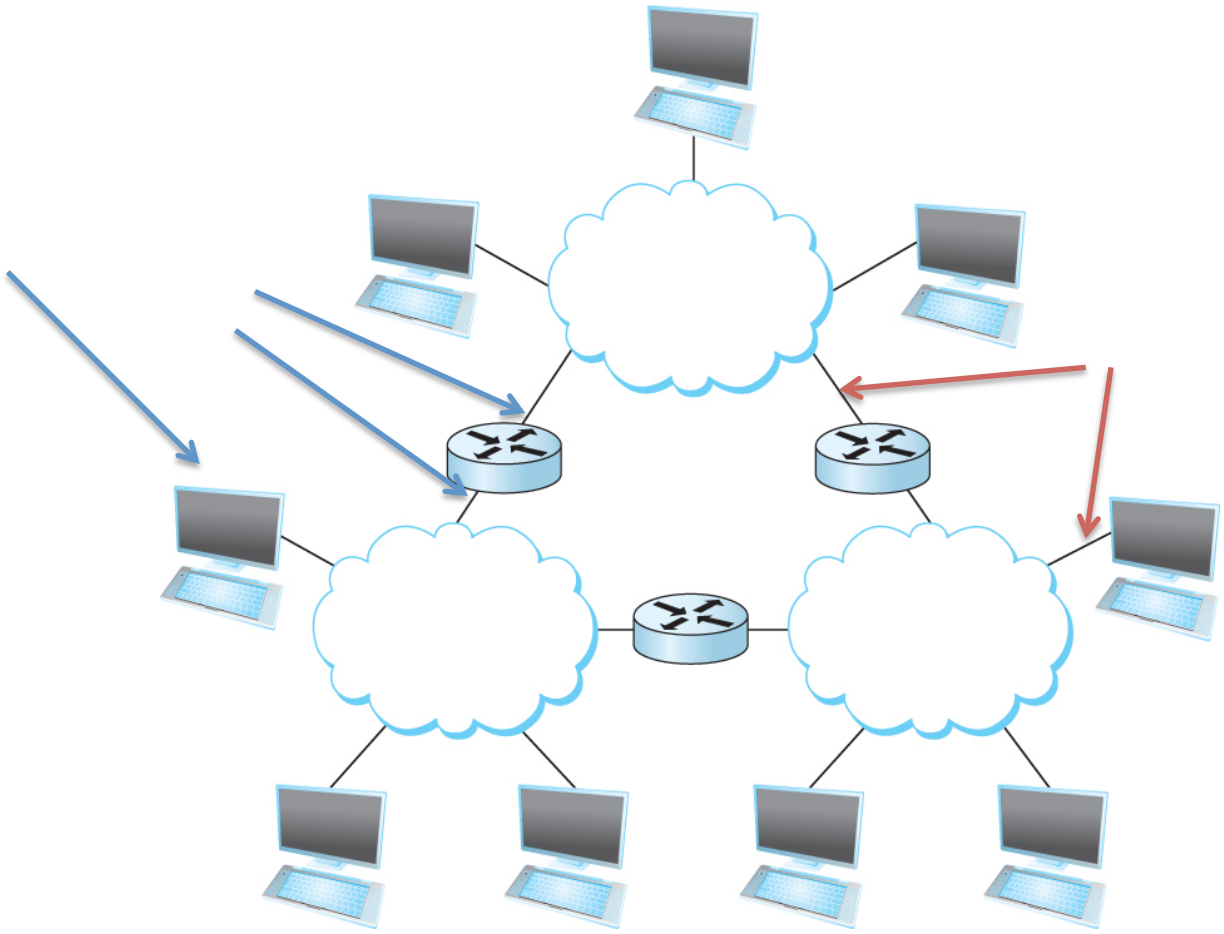
Circuit vs. Packet Switching

- Packet-switched network (more efficient)
  1. Each node receives a packet over some link
  2. Stores the packet in its internal memory
  3. Forwards it to the next node
- Circuit-switched network
  1. Establishes a dedicated circuit from source to destination node
  2. Sends stream of bits (or analog signal) across this circuit

Interconnection of Networks

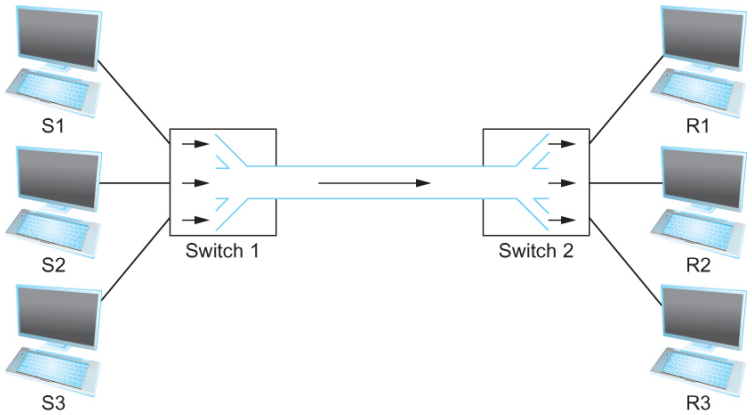


Interconnection of Networks



In communication **networks**, a **node** (Latin nodus, 'knot') is either a connection point, a redistribution point (e.g. data communications equipment), or a communication endpoint (e.g. data terminal equipment).

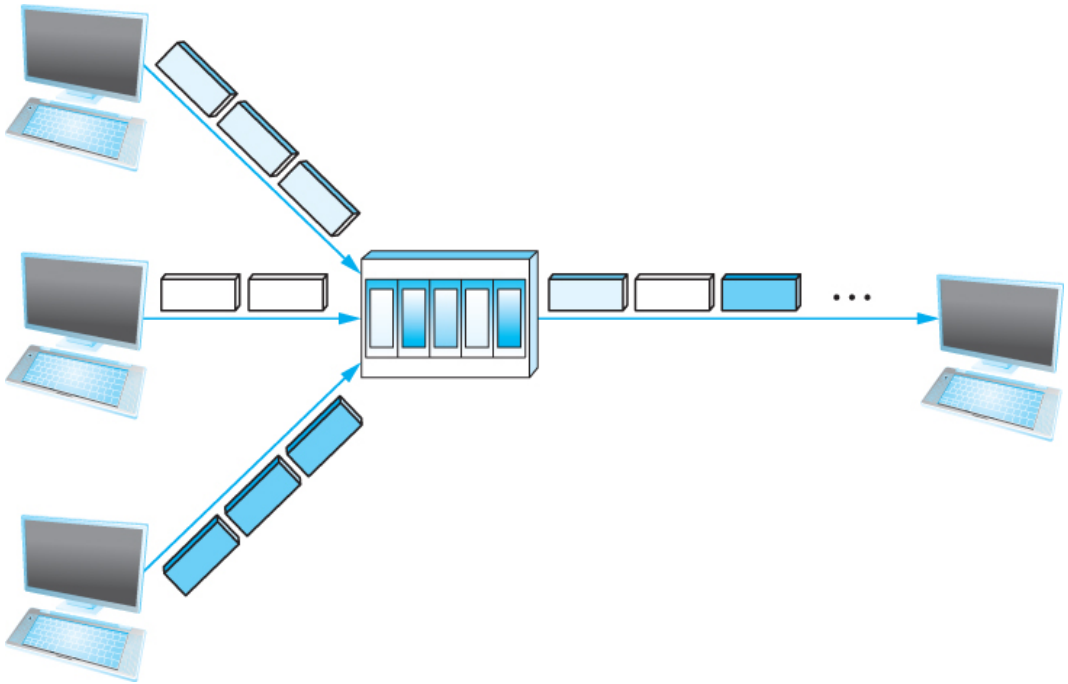
Sharing Resources: Links & Nodes



Multiplexing multiple logical flows over a single physical link

- How to share these resources?
  - Multiplexing
  - De-multiplexing
  - Time-division Multiplexing (TDM)
    - Time slots/data transmitted in predetermined slots
  - FDM: Frequency Division Multiplexing
  - Statistical Multiplexing
    - Modified TDM
    - Data is transmitted based on demand of each flow.

Statistical Multiplexing



Categorizing Networks Based on Size

- LAN
  - Local area network (extends less than 1 km)
  - School, home, office buikding
- MAN
  - Metropolitan area network (spans tens of kilometers), somewhere between LAN and WAN
  - University campus, a network of a few LAN networks
- WAN
  - Wide area network (can be worldwide)
  - Across metropolitan, regional, national or international boundaries, using leased telecommunication lines
- SAN
  - Storage area network (formerly known as system area network)
    - Confined to a single room
    - Fiber channel is a common SAN technology used to connect high-performance computing systems to storage servers and data vaults

Support for Common Services

- The goal of the network designer:
  - Support for common services used by the applications
  - Hide complexities without constraining the application designer
    - What functionalities the channels must provide to the application programs?
      - Guaranteed message delivery?
      - Message privacy important?
      - How about message delivery order at the receiving end?

Reliability

- Network designer also have to worry about three classes of failures:
  1. Bit errors
     - 1 is turned into 0 and vice versa
  2. Packet loss
     - Must be discarded due to uncorrectable bit error
     - An overloaded forwarding switch has no place to store the packet and therefore drops it (congestion)
     - Must distinguish between late and lost packets
  3. Node/link failure
     - Physical link is cut
     - Node crash

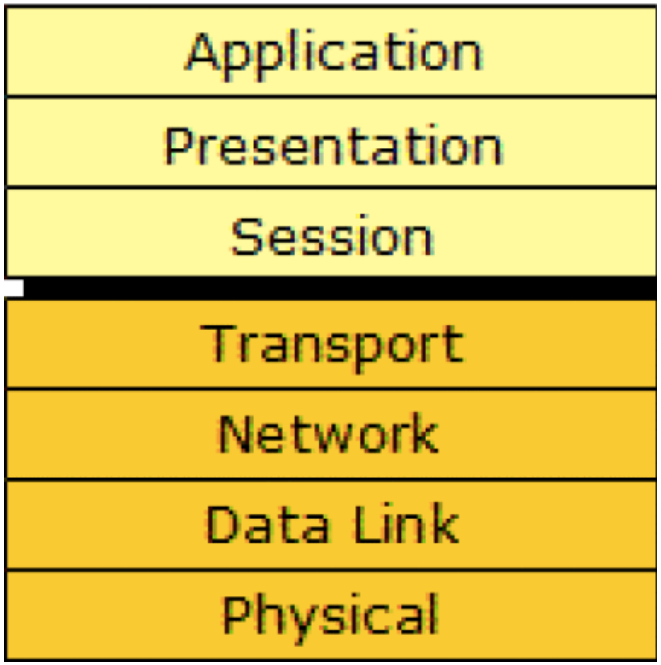
Manageability

- Networks need to be managed
  - Keeping the whole thing running correctly
  - Correctly configuring and adding new devices
  - Plug-and-play network devices (minimum user know-how needed)

don't keep plug in if not needed or not using, to keep users minimum

Network Architecture

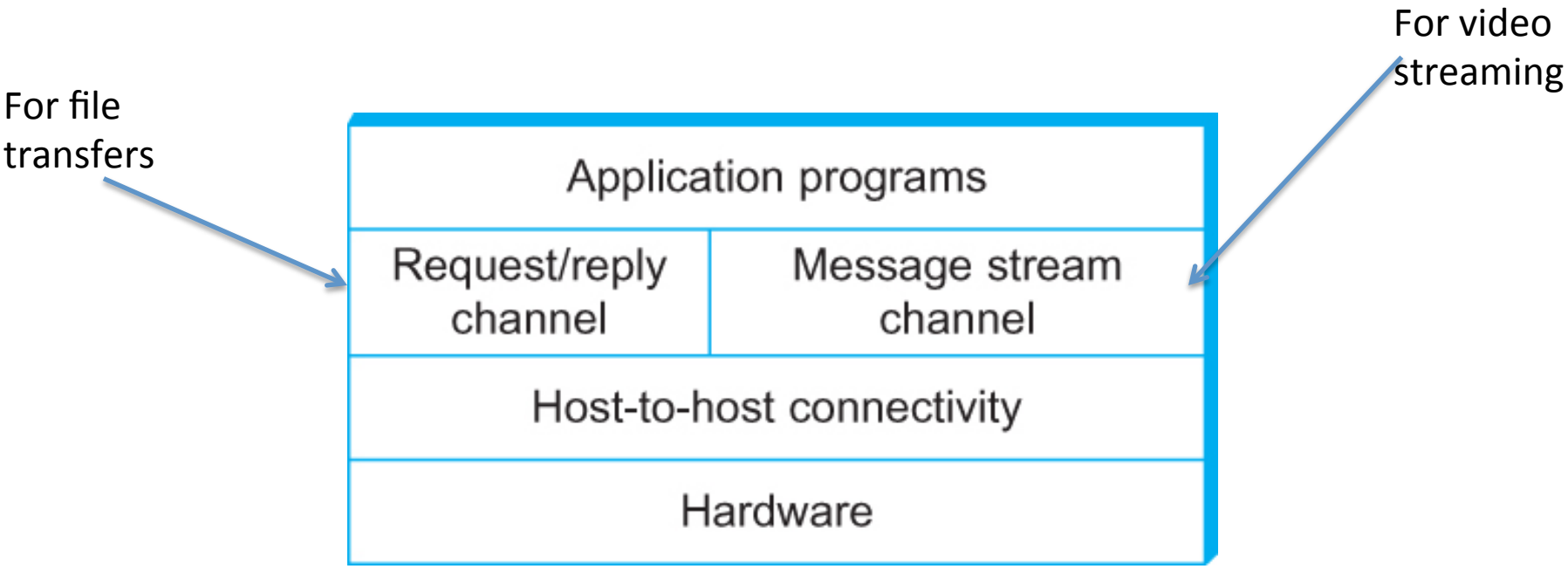
- Not an easy task
- Network architectures can help
  - The OSI architecture (7 layers)
    - Open System Interconnection
  - The Internet architecture (5 layers)
    - TCP/IP



Network Architecture

- Abstraction – hiding of details behind a well-defined interface
- Use layering to provide abstraction
- Layering also provides nice features;
  1. Decomposes the problem of building a network into more manageable components
     - Each component solves one part of the problem
  2. Provides a more modular design

Example Layered System

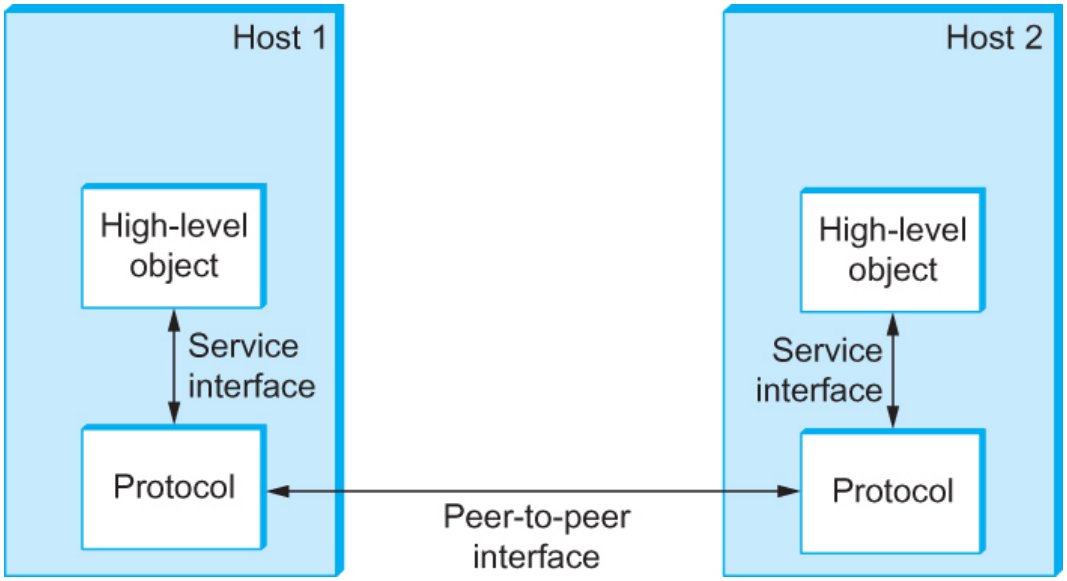


Protocol

- A set of rules and procedures used for communication
- Say you want to use USPS to send a letter to a friend:
  - Use an envelop
  - Write the destination address
    - Put it in a specific location on the envelop
    - Use a specific format
  - Put a return address
  - Put a stamp
  - Get it on to the USPS network
    - Post office
    - Mail man

Protocol

- A protocol in a layered architecture provides two interfaces:
  1. Between modules/layers on the same machine
  2. Between same modules/layers on peer machines



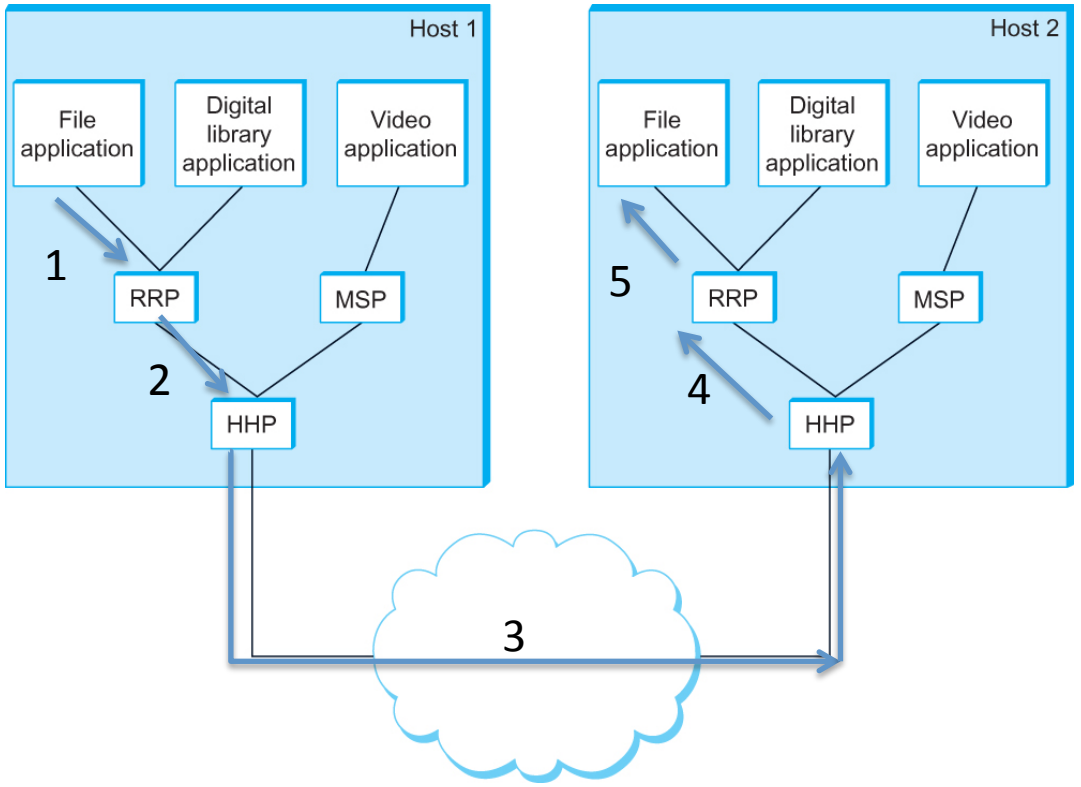
Application Protocol

- HTTP
  - Hyper Text Transfer Protocol
- TCP
  - Transmission Control Protocol
- URL
  - Uniform resource locater
  - http://www.scu.edu/sculogin/

- 17 messages for one URL request
  - 6 to find the IP (Internet Protocol) address
  - 3 for connection establishment of TCP
  - 4 for HTTP request and acknowledgement
    - » Client Request: I need this web site
    - » Server Reply: I got your request
    - » Server Reply: Here is the data you requested
    - » Client Ack: Thanks, I got it
  - 4 messages for tearing down TCP connection

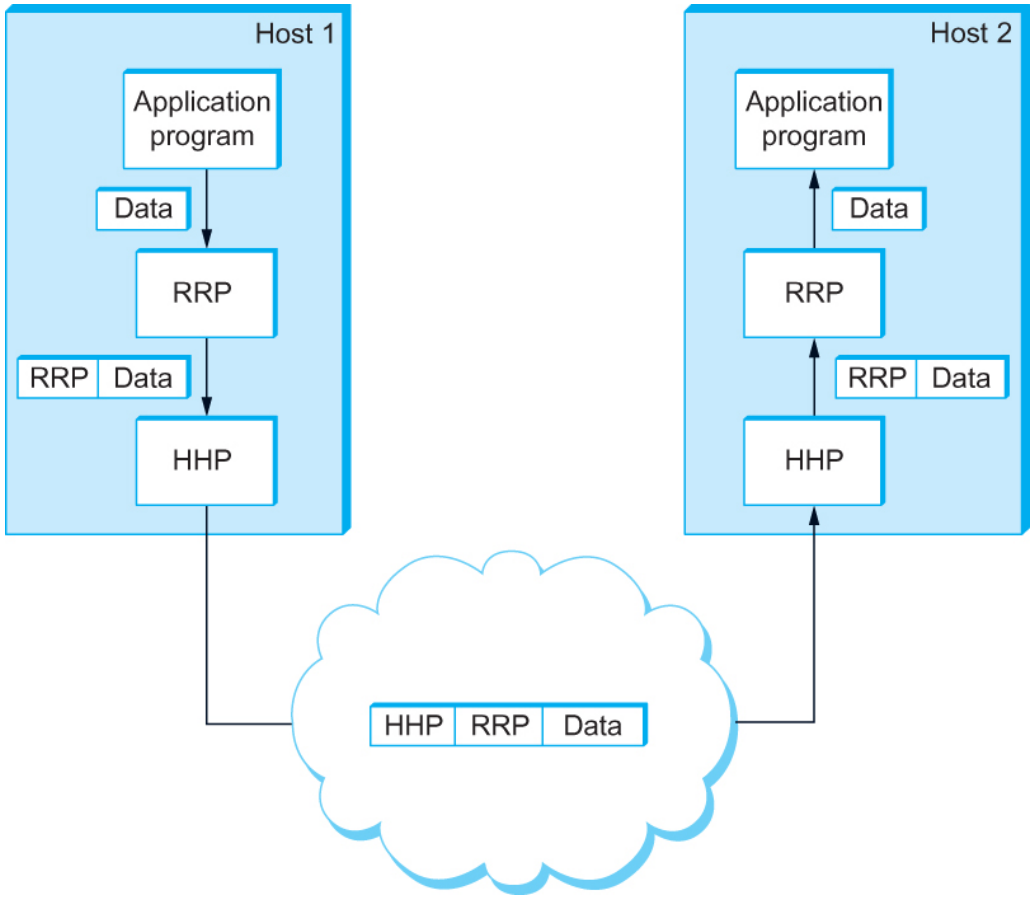
Example of a Protocol Graph

File access program on host 1 wants to send a message to its peer on host 2



Data Encapsulation

The process of data encapsulation is repeated at each level of the protocol graph



OSI Reference Model

- Open Systems Interconnect reference model provides general design guidance for data communication systems
  - to make sure all functions needed for communications are addressed
- Designed by International Standards Organization (ISO)
- 7 layer model of networking
- Each layer must be independent and only communicates to one above and below

OSI Reference Model (continue)

OSI Layer	Function
Application	User interface, communication apps
Presentation	Character sets, encryption, compression
Session	Maintain connection, login, upper layer errors
Transport	Other (guaranteed delivery) TCP/IP
Network	Network ID, routing
Data Link	Access control, data flow, framing, device ID, error detection
Physical	Electrical & physical interface, topology, encoding

OSI Reference Model (continue)

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

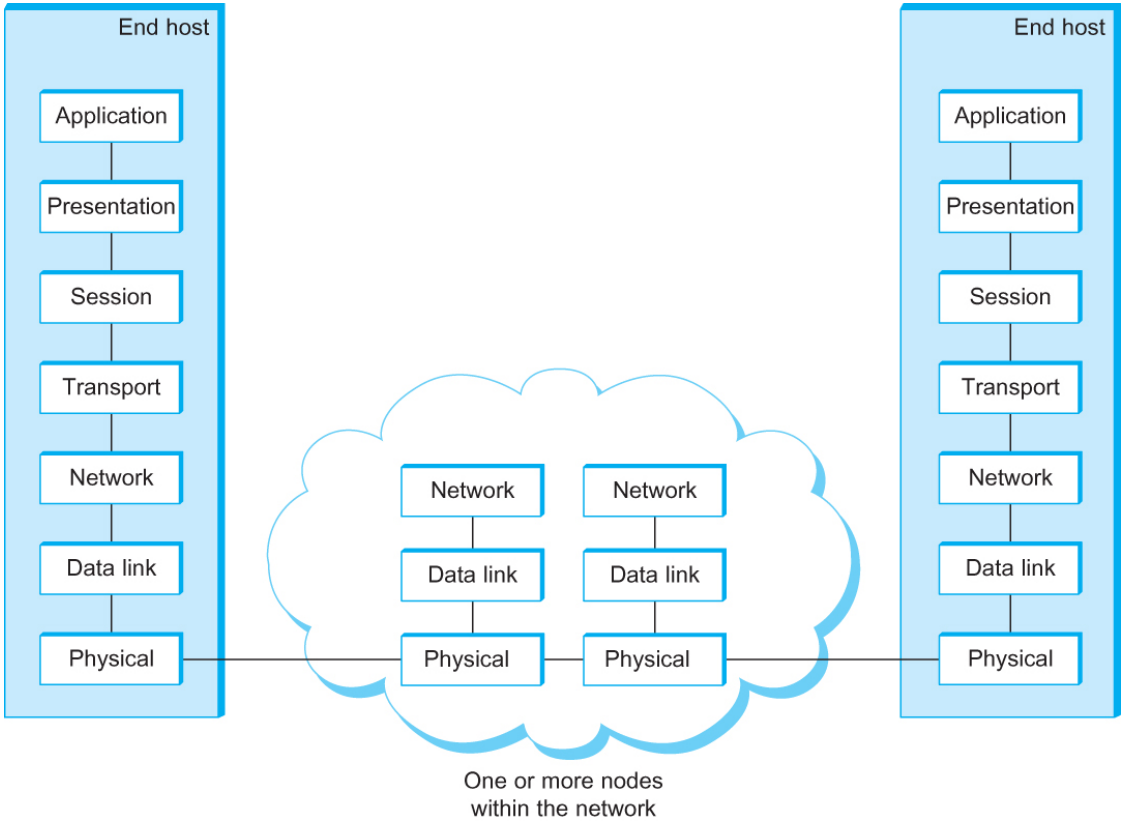
All	- Network APIs
People	- Formatting
Seem	- Synchronization
To	- Packet
Need	- Addressing/Routing
Data	- Data Frames
Processing	- Hardware

Description of Layers

- Physical Layer (layer 1)
  - Handles the transmission of raw *bits* over a communication link
- Data Link Layer (layer 2)
  - Collects a stream of bits into a larger aggregate called a *frame*
- Network Layer (layer 3)
  - Handles routing among nodes within a packet-switched network
  - Unit of data exchanged between nodes in this layer is called a *packet*

** The lower three layers are implemented on all network nodes

OSI Architecture



The OSI 7-layer Model

OSI – Open Systems Interconnection

Description of Layers

- Transport Layer (layer 4)
  - Implements a process-to-process channel
  - Unit of data exchanges in this layer is called a *message* 
- Session Layer (layer 5)
  - Provides a name space that is used to tie together the potentially different transport streams that are part of a single application
- Presentation Layer (layer 6)
  - Concerned about the format of data exchanged between peers
- Application Layer (layer 7)

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

OSI Reference Model

7 Application — Network APIs — (end hosts)
- Application/user wants to use network, either receive or send to network

6 Presentation — Formatting — (end hosts)
- What format is this data in?

5 Session — Synchronization — (end hosts)
- Sending and receiving nodes must be in synch, timing is important

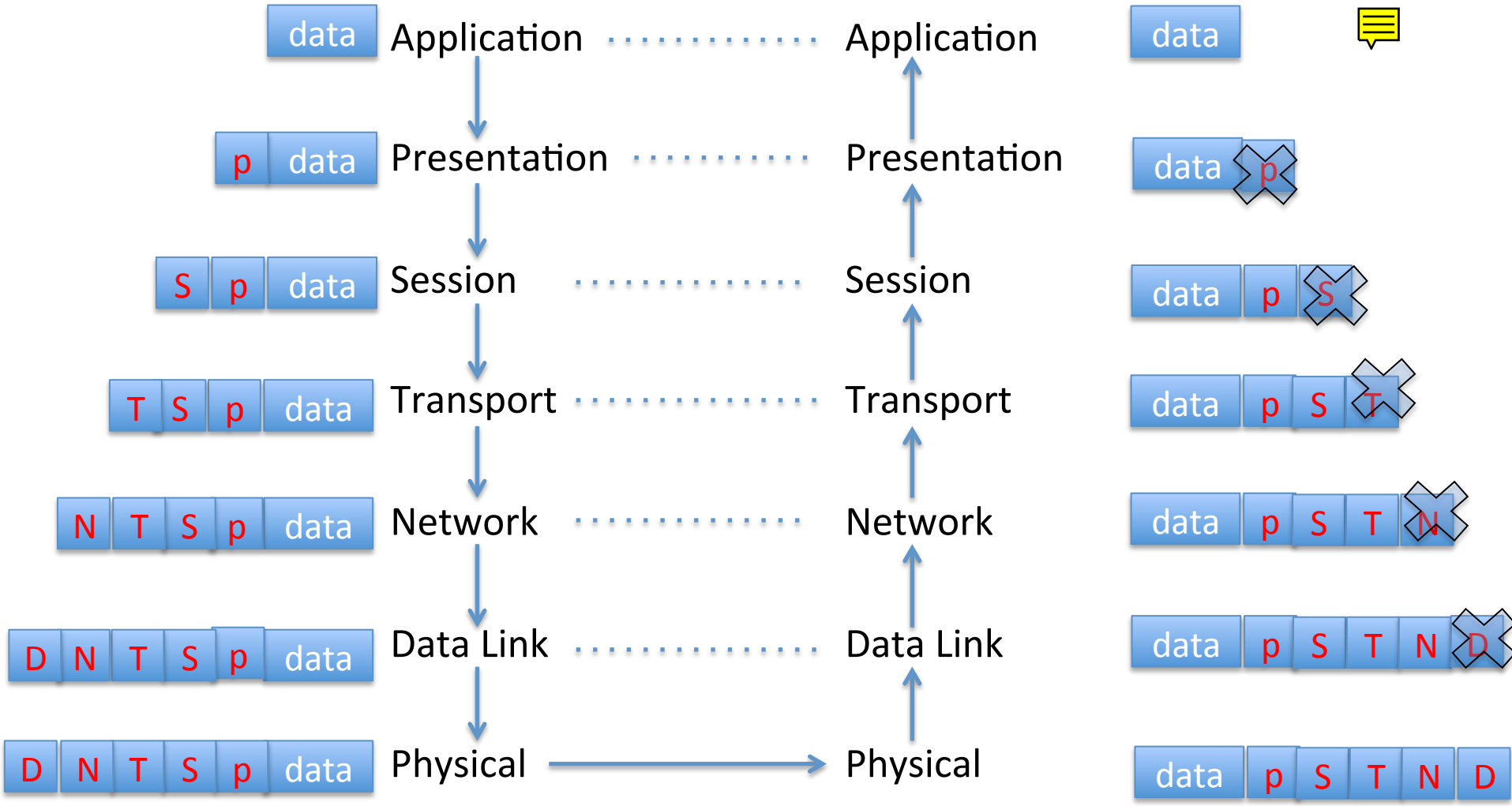
4 Transport — Packet — (end hosts)
- Packet management (how many are they, did they all get there?)

3 Network — Addressing/Routing — (routers, gateways)
- IP addressing and routing packets all the way to the destination

2 Data Link — Data Frames — (Bridges, switches)
- Data frame management (putting it into the appropriate envelope, Ethernet or token ring frame?)

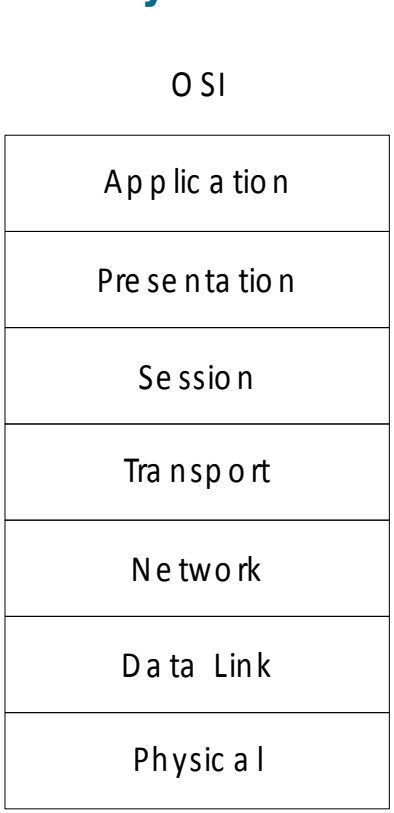
1 Physical — Hardware — (hubs & repeaters, no brains))
- How is the network card (NIC) is connected to the cable, how are we physically connected to the network? Puts the data on the network wire

Encapsulation in OSI Model

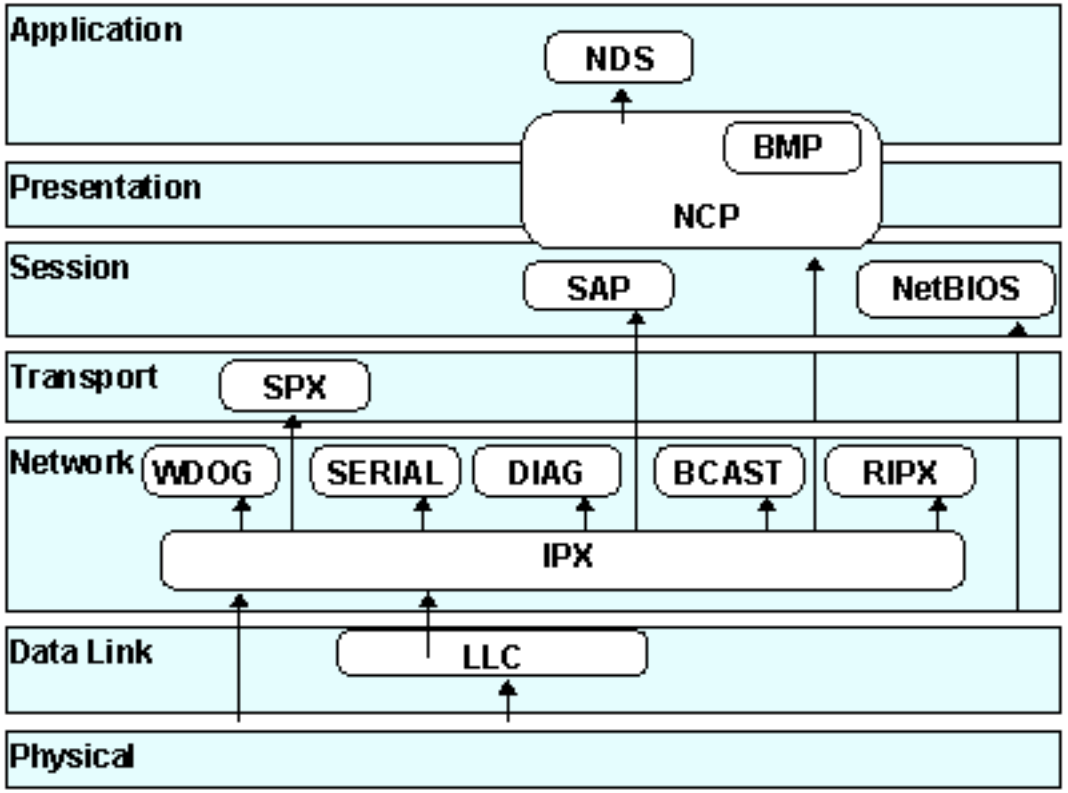


TCP/IP is not OSI

as opposed to Novell protocol suite. OSI 7 layers developed after TCP/IP that's why.



The Novell protocol suite



What is TCP/IP?

- TCP/IP stands for Transmission Control Protocol/Internet Protocol.
  - A collection of many protocols (a suite) that is used for communications between computing devices.
- What is a protocol as it relates to a communications network?
  - A formal set of conventions governing the format and relative timing of message exchange in the network.

What is TCP/IP? (continue)

What is its strength?

- Funded by the U.S. military
- Vendor independent
- Transmission medium and network hardware independent
- Built into UNIX and available Windows built in support.
- The only transport protocol used by the Internet

What is TCP/IP? (continue)

What is its weakness?

- Speed
- Management

What is TCP/IP? (continue)

nobody owns it , public

Who maintains it?

- It is an open protocol
- Controlled by Internet Engineering Task Force (IETF).
  - IETF
  - IESG

What is TCP/IP? (continue)

Main components:

- Internet Protocol (IP)
- Transmission Control Protocol (TCP)

What is TCP/IP? (continue)

More Protocols

- ICMP (Internet Communications Message Protocol)
- UDP (User Datagram Protocol)
- SMTP (Simple Mail Transfer Protocol)
- POP (Post Office Protocol)
- SNMP (management protocol)

What is TCP/IP? (continue)

End-user and Admin Applications

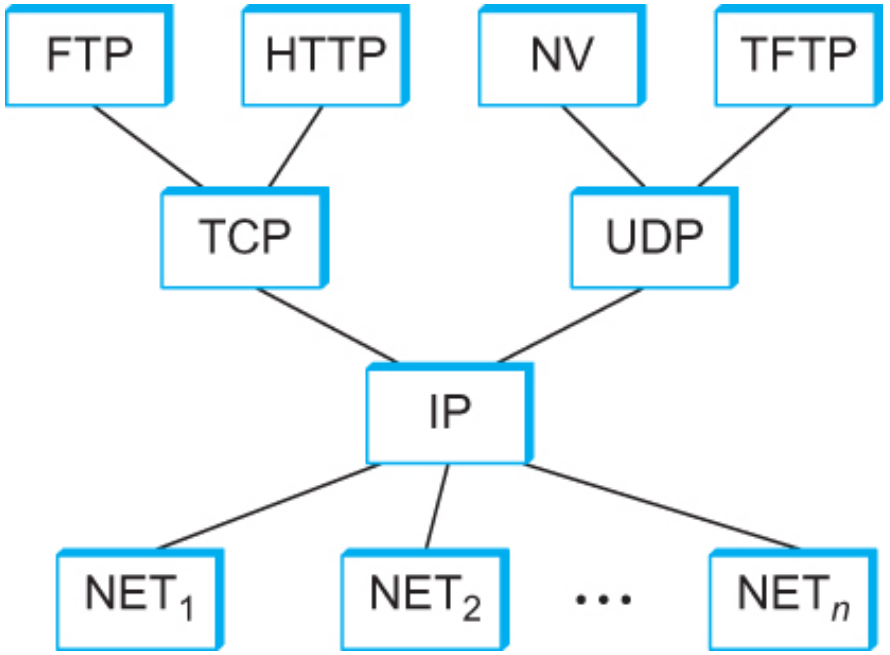
- FTP (File Transfer Protocol) - transfers files
- Telnet - terminal program
- PING, Ipconfig, Nslookup - troubleshooting utility
- E-mail - email
- World Wide Web - resource for information

System Applications

- Domain Name System (DNS) – resolves domain names to IP addresses
- DHCP (Dynamic Host Configuration Protocol) – automatically assigns a configuration to an IP host
- NTP (Network Time Protocol) – external clock sync

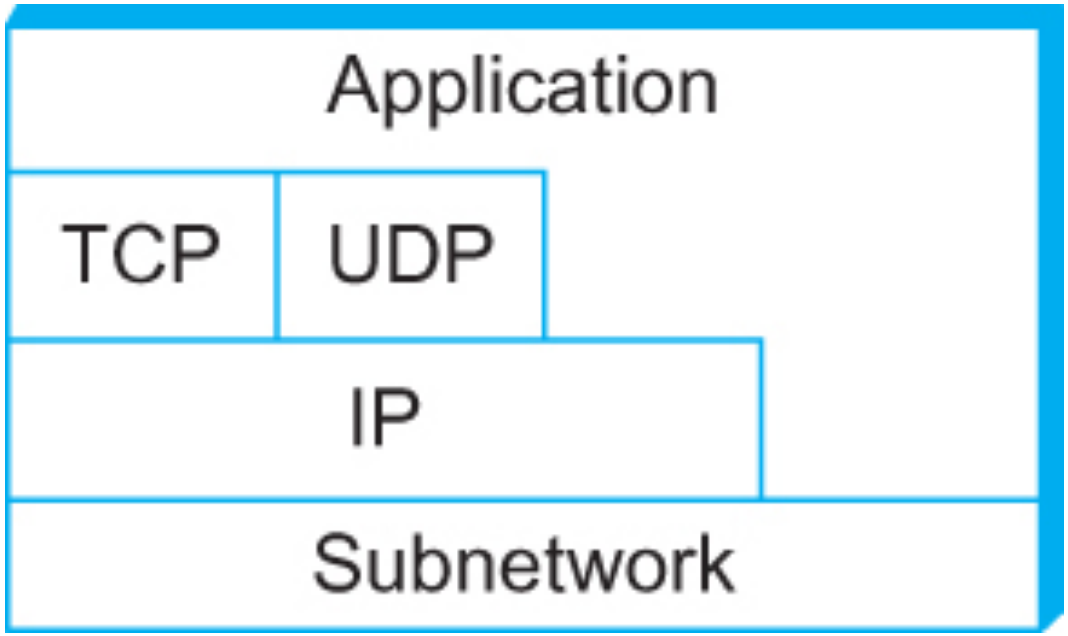
Internet Architecture

Internet Protocol graph
The four-layer model



Internet Architecture

The subnetwork layer was historically referred to as the network layer.



Application Programming Interface

- OS provides a network interface or network API
  - Socket interface

Telephony Analogy

- Say both Bob and Alice have installed home phones

1. A unique phone number is assigned to each phone
2. Both Bob and Alice have turned the ringer on to listen for an incoming call
3. Bob lifts up the phone
4. Bob dials Alice's number
5. Alice's phones starts ringing
6. Alice picks up the phone
7. Both Bob and Alice talk and exchange data
8. At the end of their conversation both Bob and Alice hang up the phone.

Computer Network Communication

- Say Bob wants to FTP (File Transfer Protocol) a few files to Alice's computer

1. Endpoints for communication is created on both ends
   - Telephone installation
2. An unique address is assigned to both ends
   - Assigning unique phone numbers
3. Bob initiates a connection to Alice's computer
   - Picks up the phone and dials Alice's phone number
4. Alice's computer has to be in ready state to receive and accept the connection
   - Alice's phone is idle with ringer on
5. Once connection established Bob can transfer the files
   - Bob and Alice can talk on the phone
6. Once files transferred the connection is then closed
   - Both Bob and Alice hang up the phone

Socket Interface

- Socket() – Establish the endpoints for the communication
  - Install the phones for both Bob and Alice
- Bind() – Assign unique addresses (both IP and port addresses are assigned here) for the endpoints
  - Assign unique phone numbers for Bob and Alice
- Listen() – Wait for a caller (listen on that port for any requests)
  - Alice's phone is idle, ringer on, and ready to receive phone calls
- Connect() – Dial a number
  - Bob dials Alice's phone number
- Accept() – Receive call
  - Alice's phone rings and she picks up
- Send(), Receive() – Exchange data
  - Bob and Alice are talking on the phone
- Close() – Connection is closed
  - Both Bob and Alice hang up the phone

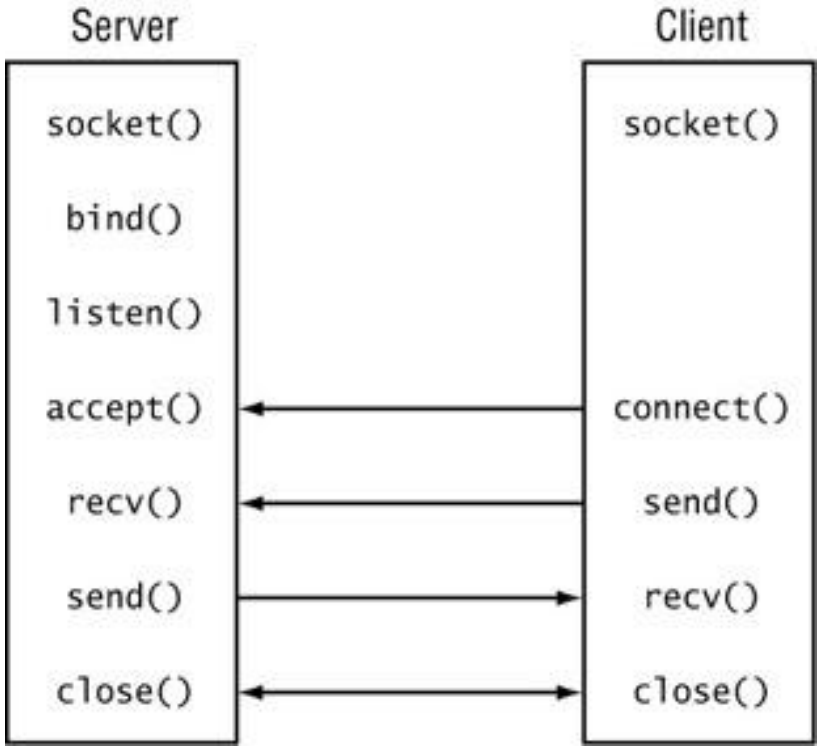
Socket Interface

- Socket Interface is used for Client-Server Programming (IPC over a network)
  - Server – An entity that provides information
  - Client - An entity that seeks information
- Sequence of events:
  1. The server starts up first and waits for a client to connect to it
  2. Client then connect
  3. The client requests for information
  4. The server sends the info to client
  5. Client disconnects
  6. Server waits for more clients to connect

Socket Interface

- A Typical TCP Server-Client Interaction Chart

Assign an IP address and port to listen to →



Socket Interface

- Programming TCP Sockets in C

```
-Server:
        => Create a socket with the socket()
        => Bind the socket to an address using the bind().
        => Listen for connections with the listen().
        => Accept a connection with the accept().
        => Send and receive data, use the read() and write() system calls.

-Client:
        => Create a socket with the socket()
        => Connect the socket to the address of the server using the connect() system call.
        => Send and receive data, use the read() and write() system calls.
```

1. Run the server program first (./server.out 2345)
2. Then run the client program
   ./client.out sever_ip_addr 2345

Socket Interface

- Programming UDP Sockets in C

```
-Server:
        => Create a socket with the socket()
        => Bind the socket to an address using the bind().
        => Send and receive data, use the recvfrom() and sendto() system calls.

-Client:
        => Create a socket with the socket()
        => Send and receive data, use the recvfrom() and sendto() system calls.
```

1. Run the server program first (./server.out 2345)
2. Then run the client program
   ./client.out server_ip_addr 2345