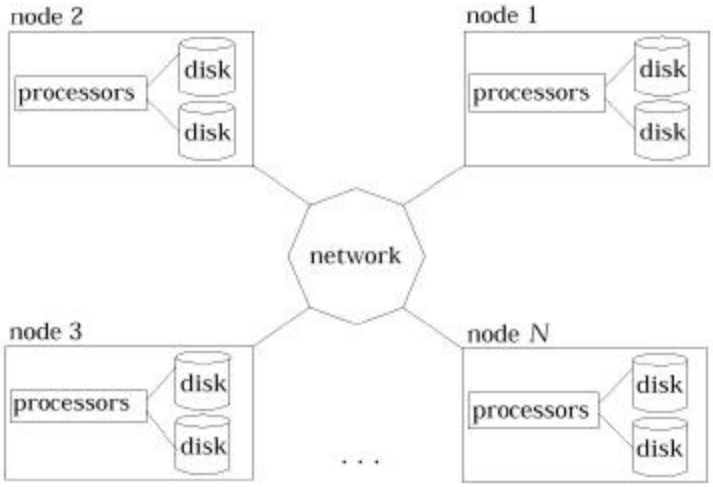
Silberschatz, et al.
Topics based on Chapter 14

Network Structures

Chapter Topics

- Background and motivation
- Network topologies
- Network types
- Communication issues
- Network design strategies

Distributed System



- Loosely-coupled processors inter-connected by communication network
  - Processors: also called sites, nodes, computers, machines, hosts …
  - *Site*: a location
  - *Host*: a particular system at the site
  - *Server*: resource provider
  - *Client*: resource user
  - *Resource*: hardware and software resources
- *Network operating system*: multiplicity of machines visible to users; logging in to remote systems; transferring data
- *Distributed operating system*: multiplicity is hidden; remote resources accessed in same way as local resources

Examples of node types

- Mainframes (IBM 3090, etc.)
  - Example applications: airline reservations; banking systems
  - Many large attached disks
- Workstations (Sun, RISC6000, etc.)
  - Example applications: computer-aided design; office information systems; private databases
  - Zero, one or two medium size disks
- Personal computers
  - Example applications: office information systems; small private databases
  - Zero or one small disk

Motivations for distributed systems

- Resource sharing
  - Examples: sharing and printing files; processing distributed database; using remote specialized hardware devices
- Computation speedup
  - Concurrent processing
  - Load sharing
- Reliability
  - Detect and recover from site failure; function transfer; reintegrate failed site on repair
- Communication
  - At the low level, message passing
  - Higher level functionality implemented on this, including file transfer, login, mail, remote procedure calls

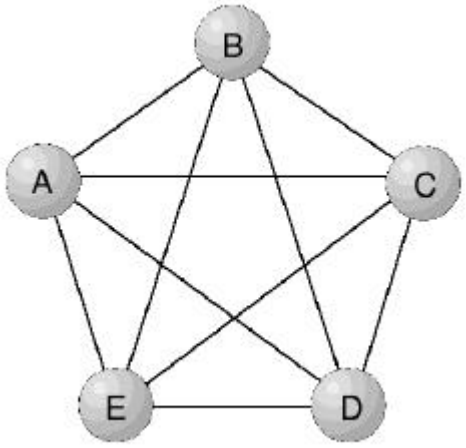
Distributed system topologies

- Sites in system can be physically connected in a variety of ways
- Comparison criteria
  - Basic cost
    - How expensive is it to link the various sites in the system?
  - Communication cost
    - How long does it take to send a message from site A to site B?
  - Reliability
    - If a link or site in the system fails, can the remaining sites still communicate with each other?

Distributed system topologies

- Topologies depicted as graphs; nodes correspond to sites
- Edge from node A to node B corresponds to a direct connection between the two sites
- Canonical network topologies
  - Fully connected
  - Partially connected
  - Hierarchical
  - Star
  - Ring
  - Multiaccess bus
  - Hybrid

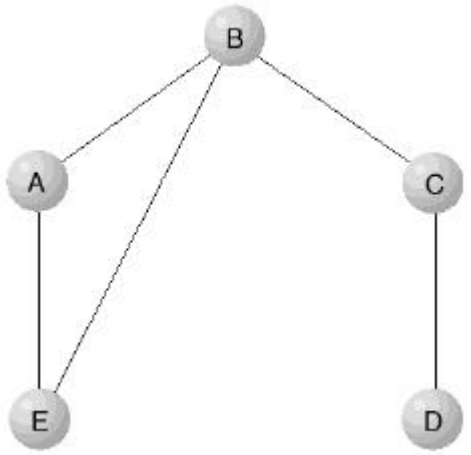
Fully connected network



Fully connected network

- Each site directly linked to all other sites
- Cost high: number of links grows as the square of the number of sites
- Fast communication
- Reliable system--many links must fail for the network to become partitioned
- *Partitioned*: split into two (or more) subsystems that lack any connection between them

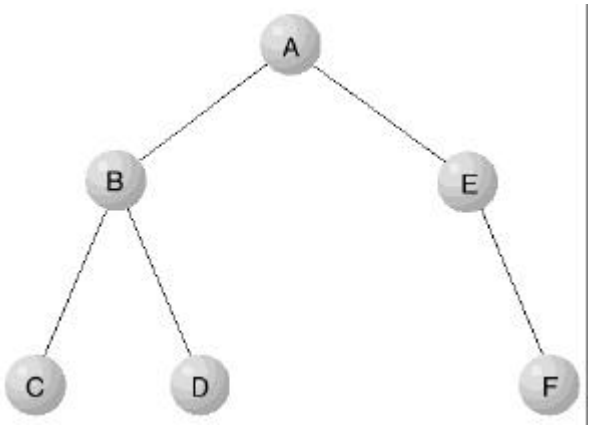
Partially connected network



Partially connected network

- Direct links between some, but not all, pairs of sites
- Lower cost than fully connected network
- Slower communication, since message may have to be sent through intermediaries
- Not as reliable as fully connected. Cutting link between B and C partitions network, for example.
- Minimize possibility of partitioning by requiring that each site connect to at least two others; eliminates possibility that single link failure will partition network

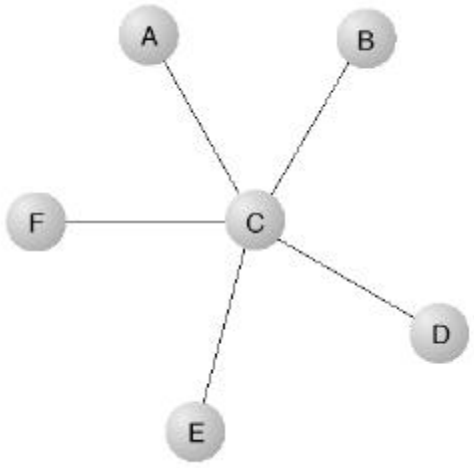
Hierarchical (tree) network



Hierarchical (tree) network

- Often mirrors corporate structure
- Siblings communicate through parent
- Mirrors observation that local systems more likely to communicate more than distant systems
- Loss of single non-leaf node partitions network

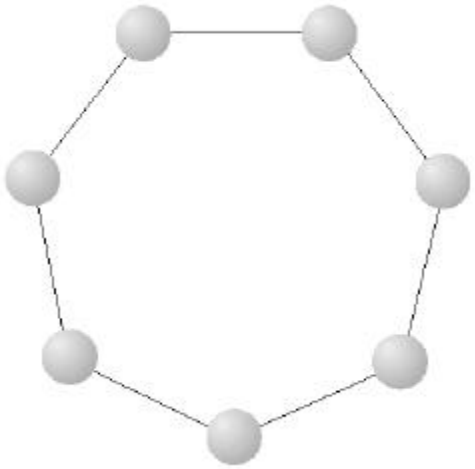
Star network



Star network

- One of sites serves as hub; connected to all others; no other connections
- Cost is linear in number of sites
- Communications cost low: at most two transfers required
- Speed may be an issue: central site can be a bottleneck; may completely dedicate central site to message-switching
- Failure of central site completely partitions network

Ring network

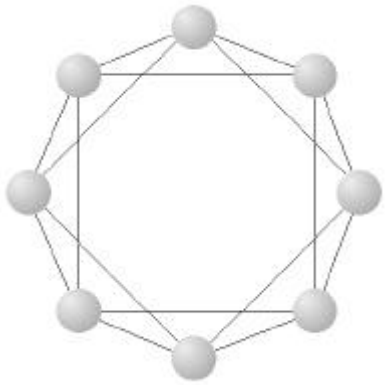


Ring network

- Each site connected to exactly two other sites
- Either unidirectional or bidirectional
  - Unidirectional: transmit to only one neighbor; all sites send information in same direction
  - Bidirectional: transmit to either neighbor
- Basic cost: linear in number of sites
- Communication cost may be high: n-1 hops maximum for unidirectional, n/2 maximum for bidirectional
- One failure partitions unidirectional ring; two failures partitions bidirectional ring
- Example: token ring

Ring network

- Improve characteristics by providing double links

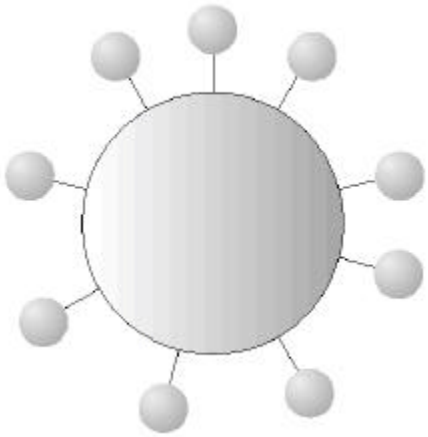


Multiaccess bus network (linear bus)

Multiaccess bus network (linear bus)

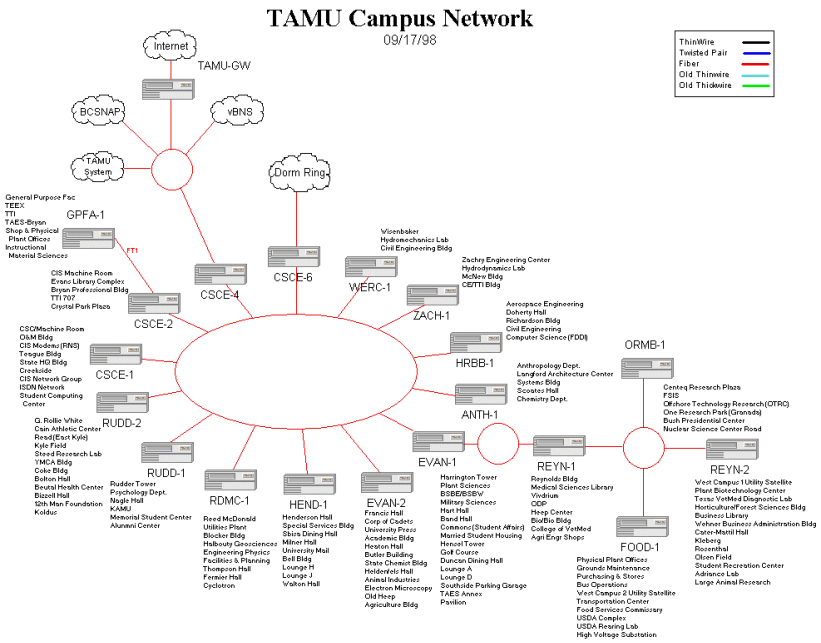
- Single, shared link
- Basic cost of network linear in number of sites
- Communication cost low, unless link becomes a bottleneck
- Unaffected by site failure, but link failure completely partitions network
- Example: Ethernet

Multiaccess bus network (ring bus)



Hybrid Networks

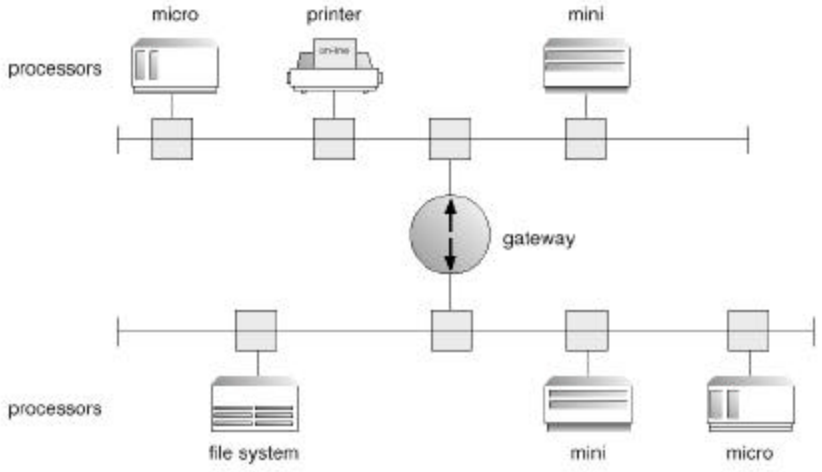
- Connecting together networks of differing types
- Example: Ethernet within a building; token ring on site; partially-connected (or hierarchic) network between sites



Network Types

- Local-area networks (LAN)
  - Designed to cover small geographical area
  - Multiaccess bus, ring, or star network
  - Speed around 10 megabits/second or higher
  - Broadcast is fast and cheap
- Wide-area networks (WAN)
  - Links geographically separated sites
  - Point-to-point connections over long-haul lines (often leased from a phone company)
  - Speed around 100 kilobits/second
  - Broadcast usually requires multiple messages

Local-Area network (LAN)



Communication issues

- Naming and name resolution
  - How do two processes locate each other to communicate?
- Routing strategies
  - How are messages sent through the network?
- Connection strategies
  - How do two processes send a sequence of messages?
- Contention
  - How are conflicting demands for use of the network resolved?

Naming and name resolution

- Processes on remote systems represented by tuple: <host-name, identifier>
  - Host-name: systems on network are named
  - Identifier: designates process on host; for example, process-id
- Host-name (human-readable) to host-id (unique; numeric) translation: *resolve* mechanism
- *Domain name service* (*DNS*) specifies naming structure of the hosts as well as name to address resolution on the Internet
  - Replaces Arpanet's system of complete host tables residing on each host

Domain name service (DNS)

- Logical addresses of Internet hosts in multiple parts: dilbert.cs.tamu.edu
- Each component has an associated name server
- Queries made in reverse order:
  - edu server asked for address for tamu.edu
  - tamu.edu server asked for address for cs.tamu.edu
  - cs.tamu.edu server asked for address for dilbert.cs.tamu.edu
  - IP address returned
- Caches allow steps to be skipped

Routing strategies

- Fixed routing
  - Path from A to B is specified in advance; the path changes only if a hardware failure disables it.
  - Shortest path usually chosen. Minimizes communication costs.
  - Cannot adapt to load changes
  - Ensures messages delivered in the order sent
- Virtual circuit
  - A path from A to B is fixed for the duration of one *session*. Different sessions involving messages from A to B may have different paths.
  - Session as short as file transfer; as long as remote login period
  - Partial remedy to adapting load changes
  - Ensures messages delivered in order sent.
- Dynamic routing

Routing strategies

- Dynamic routing
  - The path used to send a message from site A to site B is chosen only when a message is sent
  - Separate messages may be assigned different paths
    - Usually select the link that is least used at the particular time
    - Hence can adapt to load changes by avoiding using heavily used paths
  - Messages may arrive out of order
    - Include sequence number with each message
    - Question: what do we do if out of sequence?

Routing strategies

- Mixing static and dynamic routing
  - Static route to *gateway*
  - Gateway dynamically routes messages to other locations on the network
- *Router*: entity responsible for routing messages; either host computer with appropriate software or special-purpose device
  - Determines if message needs to be passed from network on which it is received to another network connected to the router
  - Routing protocol used between routers to inform them of network changes; allow them to update routing tables

Packet strategies

- Messages generally are of variable length
- Simpler design is fixed-length messages
  - Called packets, frames, datagrams
- Transfers can be *reliable* or *unreliable*
  - TCP is an example of a reliable protocol (implies ACK)
    - Question: how to limit effects of network latency on reliable transfer?
  - UDP is an example of an unreliable protocol
- Message within single packet: connectionless
- Message larger than a single packet
  - Packetized (i.e., split up into packet-sized pieces)
  - Connection established and the pieces are sent reliably

Connection strategies

- Circuit switching
  - Permanent physical link established for the duration of the communication
  - Unavailable to other processes, even if no active communication
  - Example: telephone system
- Message switching
  - Temporary link established for duration of one message transfer
  - Message is block of data with system information (e.g., source, destination, error correction codes)
  - Example: post-office mailing system
- Packet switching
  - Messages of variable length divided into fixed length packets
  - Each packet may take a different path to destination
  - Packets must be reassembled into messages as they arrive

Connection strategies Tradeoffs

- Circuit switching
  - Requires setup time
  - May waste network bandwidth
  - But… incurs less overhead for shipping each message
- Message and packet switching
  - Less setup time
  - More overhead per message
- Packet switching most common method on data networks
  - Makes best use of network bandwidth
  - Not harmful to data to break it up/reassemble it (compare to video or audio stream)

Network contention

- Several sites may want to transmit information over a link simultaneously. Techniques to avoid repeated collisions include
  - CSMA/CD
  - Token passing
  - Message slots

CSMA/CD

- Carrier sense with multiple access (CSMA); Collision detection (CD)
  - Before transmitting, listen to determine if another message is being transmitted (CSMA)
  - If link is free, can begin transmitting
  - If two or more sites begin transmitting at the same time, then they will register a collision detection (CD) and stop transmitting
  - On CD, try again after a random time interval
- When network is busy, many collisions occur, and thus performance may be degraded
- CSMA/CD is the basic idea behind Ethernet

Token passing

- *Token*: a unique message type
- Token continuously circulates in the network (ring structure)
- Site wishing to transmit waits for token
- On token's arrival, site removes token and begins transmitting
- After transmitting, retransmits token
- Issue: what happens if token lost?
  - Election to pick site to regenerate token
- Characteristic: constant performance
  - Advantage for heavily loaded networks
  - Disadvantage, vs Ethernet for lightly loaded networks

Message slots

- A number of fixed-length message slots circulate around the network (ring structure)
- Site ready to transmit waits for an empty message slot to arrive, inserts its (fixed-length) information into the slot
- Receiving slot removes message, reuses the message slot for its own message or recirculates empty message slot onto the network

Network design strategies
ISO 7-layer network model

- Communication network partitioned into multiple layers
- Each layer (logically) communicates with corresponding layer on remote system
- Systems agree on protocols for each of the layers
- Physically, a message starts at (or above) the top-level layer and is passed through each lower level in turn

ISO 7-layer network model

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

ISO 7-layer network model

- Physical layer
  - Handle mechanical and electrical details of the physical transmission of a bit stream
- Data-link layer
  - Handles frames (fixed-length parts of packets), including any error detection and recovery that occurred in the physical layer
- Network layer
  - Provides connections and routes packets in the communication network, including handling the address of outgoing packets, decoding the address of incoming packets, and maintaining routing information for proper response to changing load levels

ISO 7-layer network model

- Transport layer
  - Responsible for low-level network access and for message transfer between clients, including partitioning messages into packets, maintaining packet order (or not), controlling flow, and generating physical addresses

ISO 7-layer network model

- Session layer
  - Implements sessions, or process-to-process communication protocols
- Presentation layer
  - Resolves the difference in formats among the various sites in the network, including character conversions, and half duplex/full duplex (echoing)
- Application layer
  - Interacts directly with users; deals with file transfer, remote-login protocols and electronic mail, as well as schemas for distributed databases

CSMA/CD (Ethernet)

- Commonly, coaxial cable or twisted-pair at 10 Mbps
- Standard media
  - 10 Base 2
    - Thin wire coaxial cable (0.25 inch) with maximum segment length of 200 m
  - 10 Base 5
    - Thick wire coaxial cable (0.5 inch diameter) with maximum segment length of 500 m
  - 10 Base T
    - Hub (star) topology with twisted-pair drop cables
  - 10 Base F
    - Hub (star) topology with optical fiber drop cables

CSMA/CD

- Thick-wire connections made with a *tap; uses transceiver*
- Transceiver functions
  - Send and receive data to and from the cable
  - Detect collisions on the cable medium
  - Provide electrical isolation between the coaxial cable and cable interface electronics
  - Protect the cable from any malfunctions in either the transceiver or the attached device (*jabber control*)

CSMA/CD

- Controller card
  - Encapsulation and de-encapsulation of frames for transmission and reception on the cable
  - Error detection
  - DMA

CSMA/CD

- Frame format
  - Preamble (7 octets, each equal to 10101010)
    - Used for bit synchronization
  - Start-of-frame delimiter (1 octet, 10101011)
  - Destination and source network addresses
    - 2 or 6 octets
    - Individual address or group address specified by first bit
  - Length indicator (2 octets)
  - Data (<= 1500 octets)
  - Pad (optional), if needed to make minimum length requirements
  - Frame check sequence (i.e., CRC); 4 octets

CSMA/CD

- Frame transmission
  - Monitor link until empty. If not-empty, wait until empty and also for **interframe gap** time before transmitting (to allow the passing frame to be received)
  - During transmission, monitor to detect collision
  - If collision detected, stop transmission and turn on "jam signal" to guarantee that everyone detects the collision
  - Schedule retransmission after delaying for a short, randomly selected, time interval

CSMA/CD

- Collision
  - Retransmission of frame attempted up a defined maximum number of tries: **attempt limit**
  - Repeated collisions indicate a busy medium, so progressively increases time delay between repeated retransmission attempts. **Truncated binary exponential backoff**
    - After transmission of jam sequence, delay for random integral number of slot times before attempting to retransmit the affected frame
    - **Collision window**: effectively twice the time for the first bit of the preamble to propagate to all parts of the cable medium (corrupted signal may need to propagate back)
    - **Slot time** defines worst-case time delay must wait
    - Slot time = 2 x (transmission path delay) + safety margin
    - Number of slot times to wait is a uniformly distributed random integer R in the range 0 <= R < 2^K, where K = min(N, backoff limit)

TCP/IP

- Internet's protocol; developed in 1980's
- Supports communication across heterogeneous networks (i.e., *internets*)--note small "i"
- No official protocol model, but can arrange tasks into five relatively independent layers
  - Application layer
  - Host-to-host, or transport layer
  - Internet layer
  - Network access layer
  - Physical layer

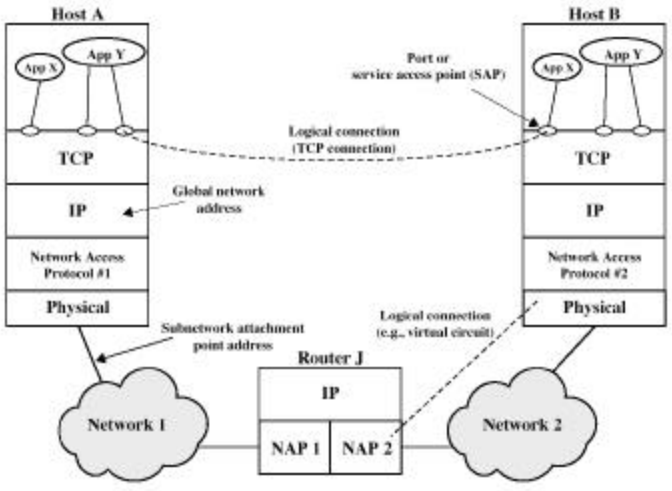
TCP/IP protocol layers

- Application layer
  - Logic to support user applications (ISO session, presentation, and application layers)
- Host-to-host, or transport layer
  - Message transfer between clients; packetizing; maintaining packet order, etc. (ISO transport layer)
  - TCP (also UDP)
- Internet layer
  - Procedures to allow data to traverse multiple, interconnected networks (ISO network layer, in part)
  - IP: internet protocol

TCP/IP protocol layers

- Network access layer
  - Exchange of data between an end system and the network to which it is attached (ISO link layer and network layer, in part)
  - Examples: X.25 (packet switching), Ethernet, etc.
- Physical layer
  - Physical interface between a data transmission device and a transmission medium or network (ISO Physical layer)

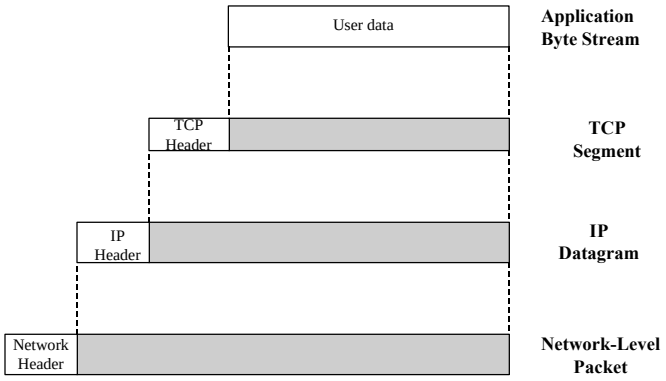
TCP/IP concepts



Protocol stack

- Logically, each level communicates with its peer
- Physically, message begins at application level and passes through each lower-level layer in turn
  - Each layer adds a header to the message on transmission, strips the header off on receipt
  - More information about header contents later
    - Example information in TCP header includes destination port, sequence number, checksum
    - Example information in IP header includes destination subnetwork address, facilities requests (e.g., priority in the subnetwork)

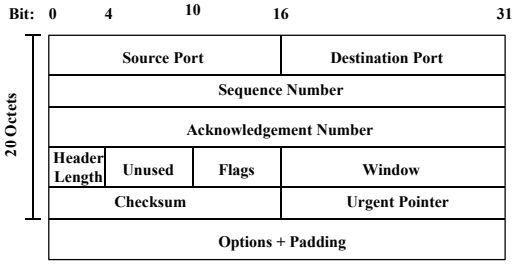
Protocol data units in the TCP/IP architecture



TCP and UDP

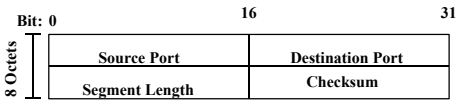
- Transport layer protocols
- TCP: reliable connection for the transfer of data between applications
- UDP: connectionless service for application-level procedures; does not guarantee delivery, preservation of sequence, or protection against duplication; enables messages to be sent with only a minimum of protocol overhead
- Protocol goals reflected in headers (follow)

TCP and UDP headers



TCP header

UDP header



TCP/IP applications

- Simple Mail Transfer Protocol (SMTP) [TCP]
- File Transfer Protocol (FTP) [TCP]
- Telnet [TCP]
- Name Server Protocol (NSP)
- Simple Network Management Protocol (SNMP) [UDP]


```
#ident    "@(#)services       1.16      97/05/12 SMI"       /* SVr4.0 1.8       */
#
# Network services, Internet style
#
tcpmux              1/tcp
echo                7/tcp
echo                7/udp
discard             9/tcp               sink null
discard             9/udp               sink null
systat              11/tcp              users
daytime             13/tcp
daytime             13/udp
netstat             15/tcp
chargen             19/tcp              ttytst source
chargen             19/udp              ttytst source
ftp-data  20/tcp
ftp                 21/tcp
telnet              23/tcp
smtp                25/tcp              mail
time                37/tcp              timserver
time                37/udp              timserver
name                42/udp              nameserver
whois               43/tcp              nicname             # usually to sri-nic
domain              53/udp
domain              53/tcp
bootps              67/udp                                  # BOOTP/DHCP server
bootpc              68/udp                                  # BOOTP/DHCP client
hostnames 101/tcp             hostname  # usually to sri-nic
sunrpc              111/udp             rpcbind
sunrpc              111/tcp             rpcbind
```

```
#
# Host specific functions
#
tftp               69/udp
rje                77/tcp
finger             79/tcp
link               87/tcp             ttylink
supdup             95/tcp
iso-tsap  102/tcp
x400               103/tcp                              # ISO Mail
x400-snd  104/tcp
csnet-ns  105/tcp
pop-2              109/tcp                              # Post Office
uucp-path 117/tcp
nntp            119/tcp         usenet           # Network News Transfer
ntp                123/tcp                              # Network Time Protocol
ntp                123/udp                              # Network Time Protocol
NeWS               144/tcp            news              # Window System
```



```
#
# UNIX specific services
#
# these are NOT officially assigned
#
exec               512/tcp
login              513/tcp
shell              514/tcp            cmd               # no passwords used
printer            515/tcp            spooler           # line printer spooler
courier            530/tcp            rpc               # experimental
uucp               540/tcp            uucpd             # uucp daemon
biff               512/udp            comsat
who                513/udp            whod
syslog             514/udp
talk               517/udp
route              520/udp            router routed
new-rwho  550/udp            new-who           # experimental
rmonitor  560/udp            rmonitord # experimental
monitor            561/udp                              # experimental
pcserver  600/tcp                                  # ECD Integrated PC board srvr
kerberos  750/udp        kdc           # Kerberos key server
kerberos  750/tcp            kdc               # Kerberos key server
ufsd               1008/tcp  ufsd              # UFS-aware server
ufsd               1008/udp  ufsd
ingreslock      1524/tcp
listen          2766/tcp                        # System V listener port
nfsd               2049/udp  nfs                 # NFS server daemon (clts)
nfsd               2049/tcp  nfs                 # NFS server daemon (cots)
lockd              4045/udp                      # NFS lock daemon/manager
lockd              4045/tcp
dtspc              6112/tcp                      # CDE subprocess control
fs                 7100/tcp                      # Font server
xaudio             1103/tcp  Xaserver  # X Audio Server
```