TCP/IP

Hemant Kr Rath
Dept of Electrical Engg
IIT-Bombay
E-Mail: hemantr@ee.iitb.ac.in

TCP/IP

- What is it?
  - Basic communication language or protocol of Internet
  - Used as a communications protocol in a private network
    - Viz.: *Intranet or Extranet*
  - TCP/IP is a *two-layer* program
    - The higher layer, *Transmission Control Protocol* (TCP)
      - Manages the assembling of a message/file into smaller packets
    - The lower layer, *Internet Protocol* (IP)
      - Handles the address part of each packet so that it gets to the right destination

Different Types of Services

Connection Oriented	*Services*	*Example*
	Reliable messages stream	Sequence of pages
	Reliable byte stream	Remote Login
	Unreliable Connection	Digitized Voice
Connection Less	Unreliable datagram	Electronic Junk mail
	Acknowledged datagram	Registered mail
	Request-reply	Database query

Layered Architecture

Service/Layer	*Telephone*	*Postal*	*Computer*
Message Creation	Call Initiation	Composition	Application
Message Structuring		Placing in Cover	Transport
Path Allocation	Circuit Formation	Sorting	Network
Transmission	Voice Transmission	Mail Bags	Data Link + Physical Layer
Reception		Delivery	

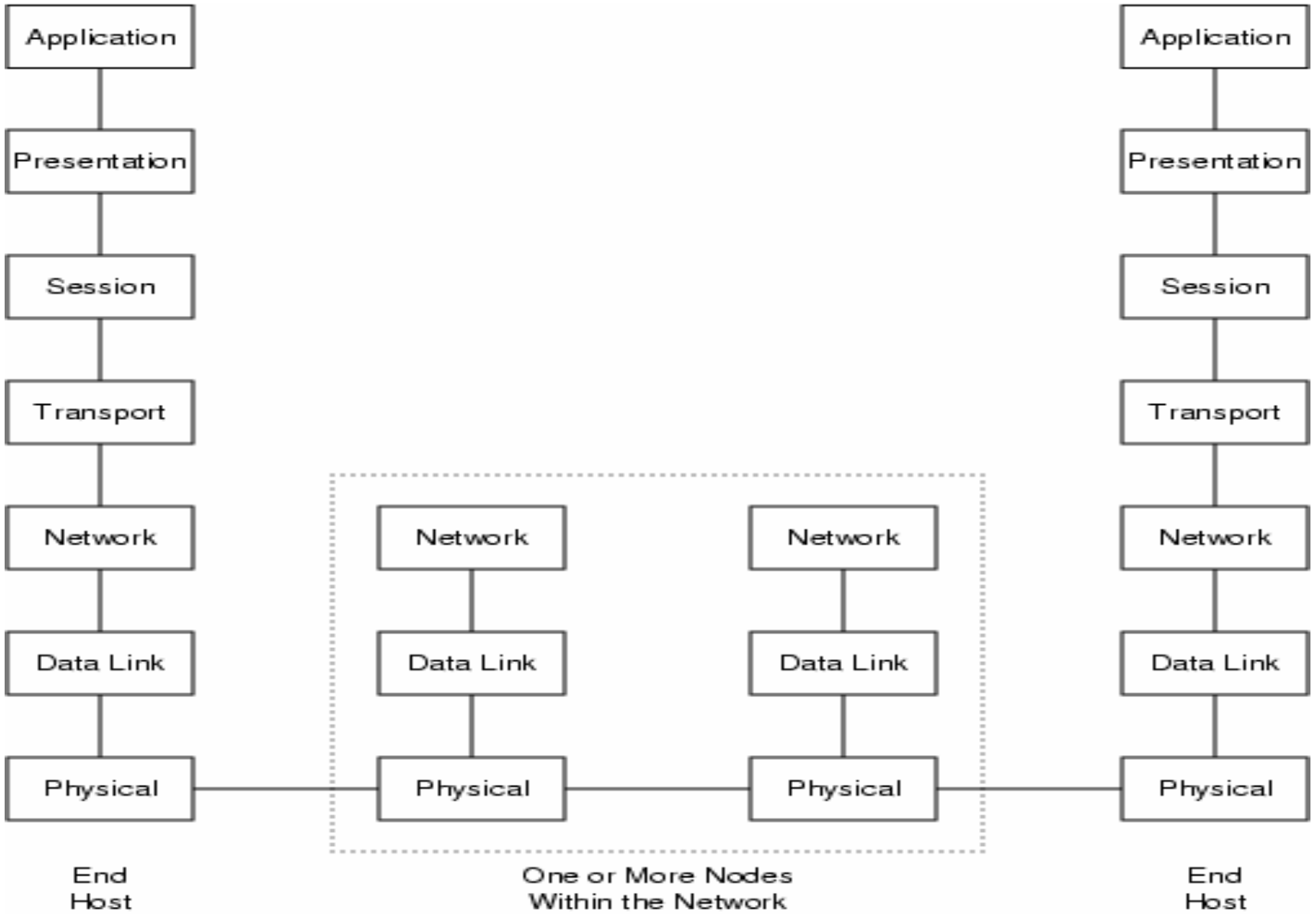
Why Layers?

- Allows a *divide-and-conquer* strategy
  - Easier to understand and implement
  - Implementation of one layer can be changed without affecting other layers
- Specification is separated from implementation
- Reuse of functionality
- Upper layers share the services of lower layers

Disadvantages of Layers

- Implementation may not be the most *efficient*
  - Major concern in low bandwidth networks and low power hosts
    - Viz.: Wireless ad-hoc network
- Cumbersome
  - Load on layers may not be balanced
  - Too-many layers

OSI Layers


Application
Presentation
Session
Transport
Network
Data Link
Physical
Network
Data Link
Physical
Network
Data Link
Physical
Application
Presentation
Session
Transport
Network
Data Link
Physical
End Host
One or More Nodes Within the Network
End Host

Concepts of OSI Model

- *Services* provided by different layers
  - Set of *operations* that a layer provides to the layer above it
  - Says nothing about how these operations are performed by
- *Interface* between adjacent layers
  - An interface tells processes in the higher layer *how to access* the underlying layer for getting the services
- *Protocols* obeyed by different layers
  - A protocol is a *set of rules* governing the format and meaning of the messages exchanged by peer entities

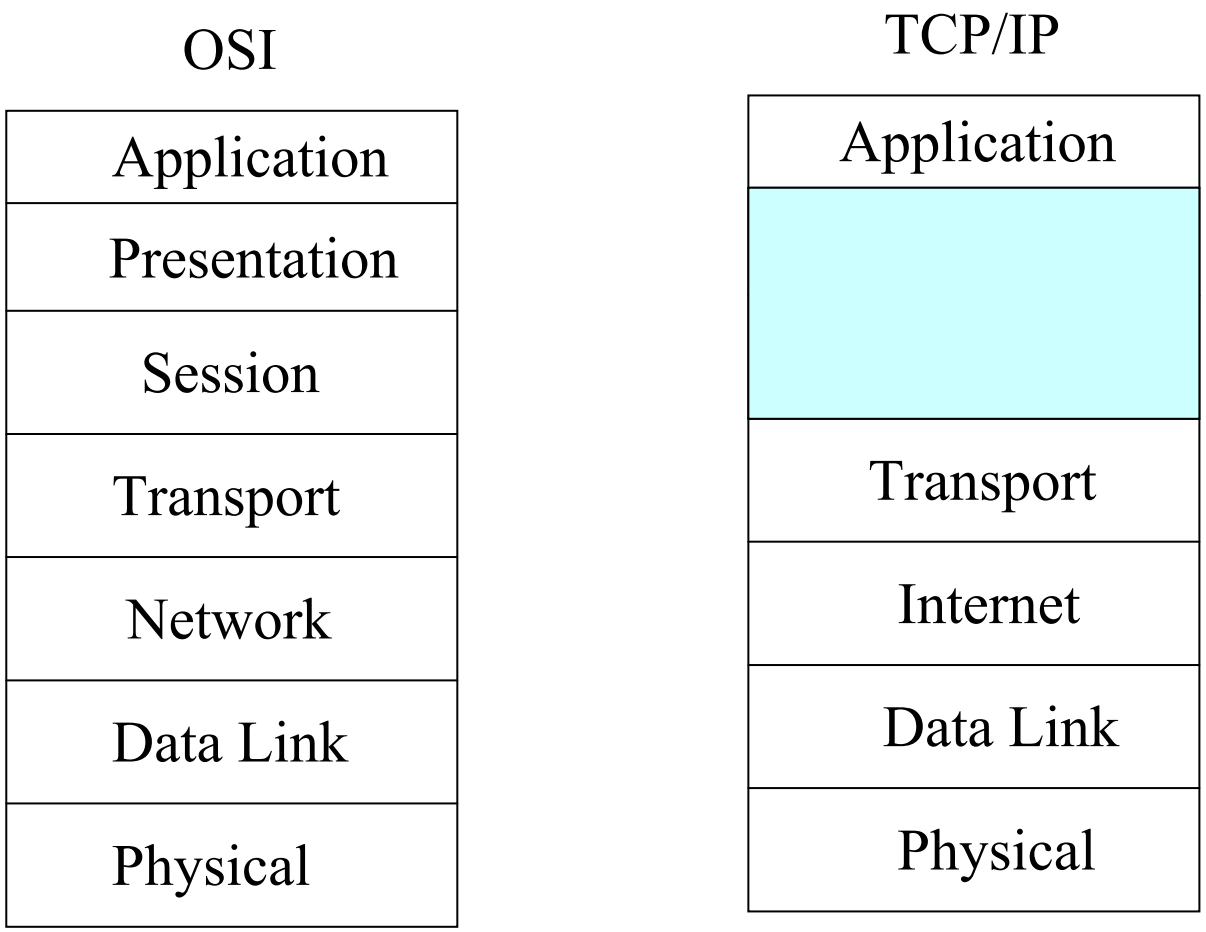
A Critique of OSI Layers

- The Reference Model came *before* the Protocols
  - Created problems in defining the *functionalities* to be incorporated in the different layers.
- *Imbalance* among the 7 layers with respect to their relative functionalities
  - *Separate sub-layers* in the Data Link (MAC) and the Network layer (IP)
- Session and Presentation layers have *little significance*

TCP/IP Reference Model

- The TCP/IP reference model evolved from *ARPANET*
  - *Multiple networks* to be interconnected in a seamless manner
- The crucial layer is the *Internet Layer*
  - Permits hosts to introduce packets into any network
- Uses *Internet Protocol (IP)* to perform packet routing
  - IP layer is supported by a *Host-to-network* layer
- The layer above IP layer enables peer entities on hosts to exchange messages
  - Similar to the OSI *Transport layer*

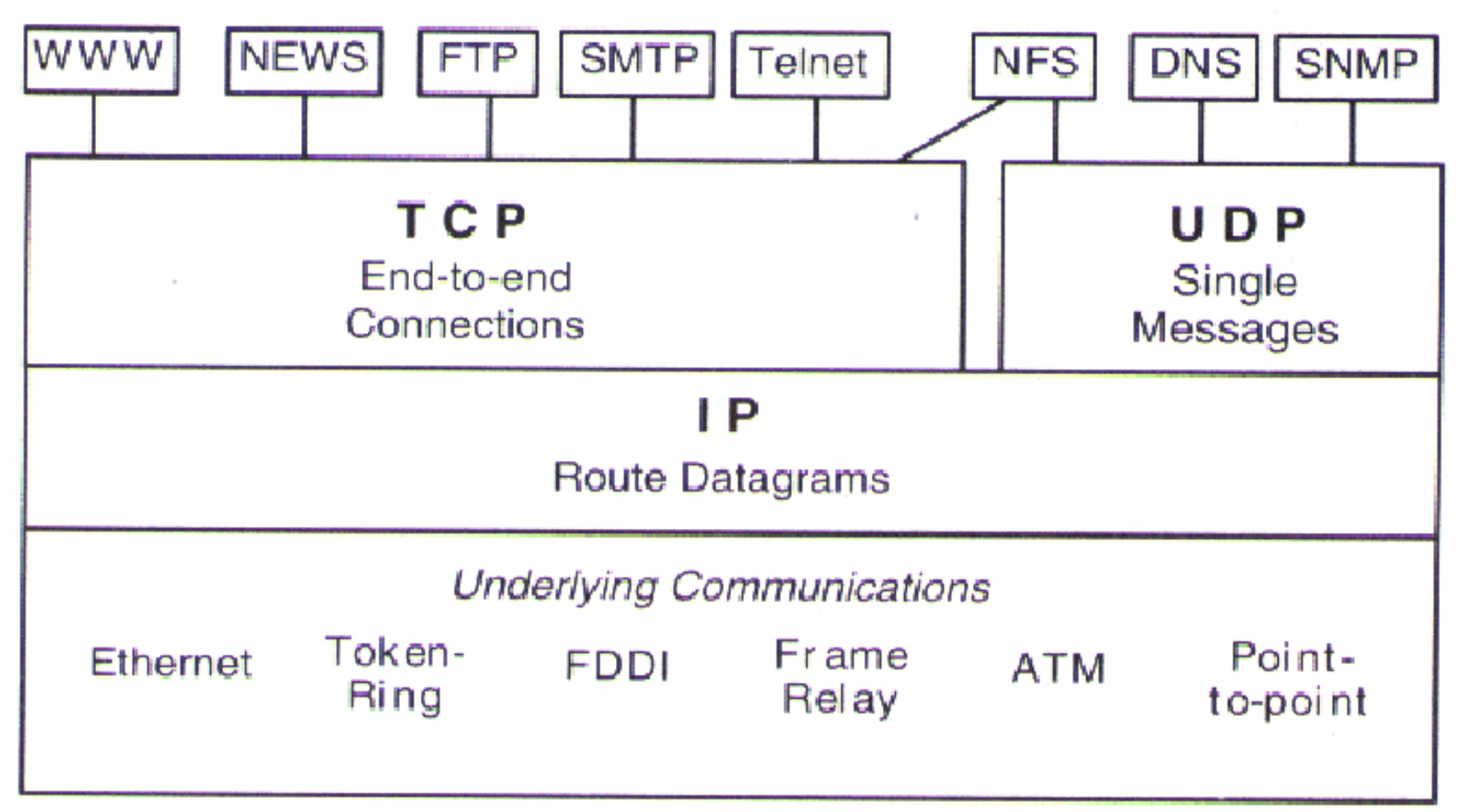
OSI vs. TCP/IP Layers



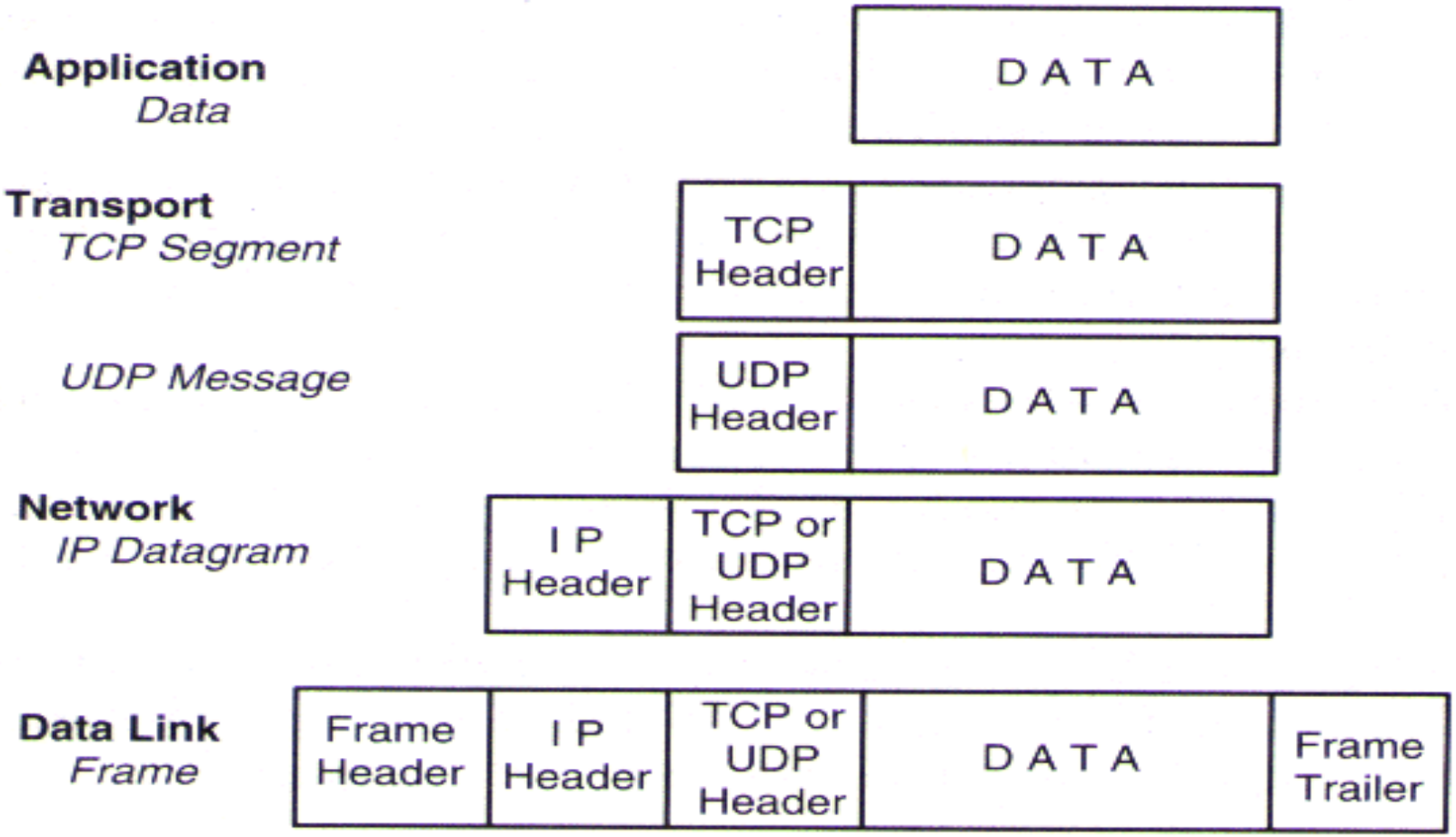
A Critique of TCP/IP

- Opposed to OSI
  - No differentiation between *specification and implementation*
- Does not give any effective model
  - Poorly suited to describing any other protocol stack
- *Host-to-network* layer is not really a layer
  - It is an interface between network and data link layers
- All protocols other than TCP and IP are ad-hoc
- TCP/IP protocols are widely used
  - OSI model is useful only for discussing computer networks

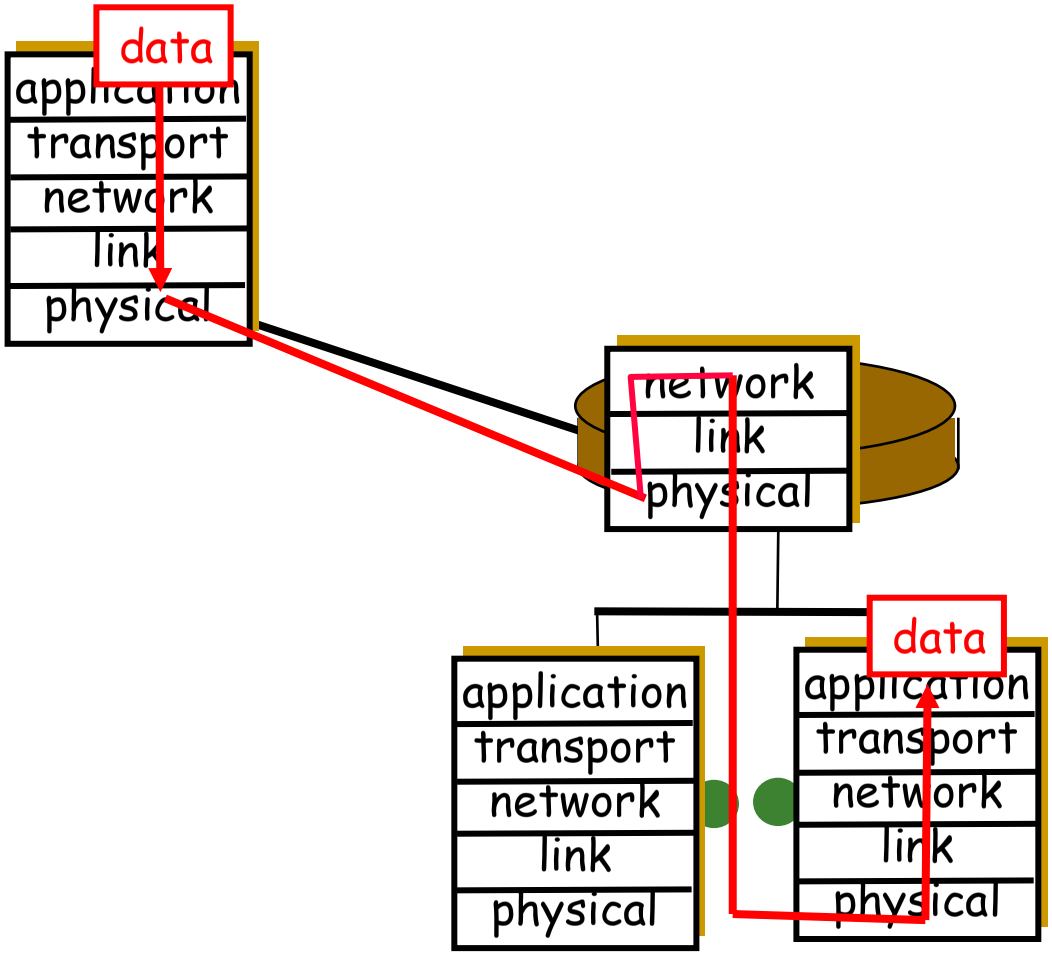
TCP/IP User Interface



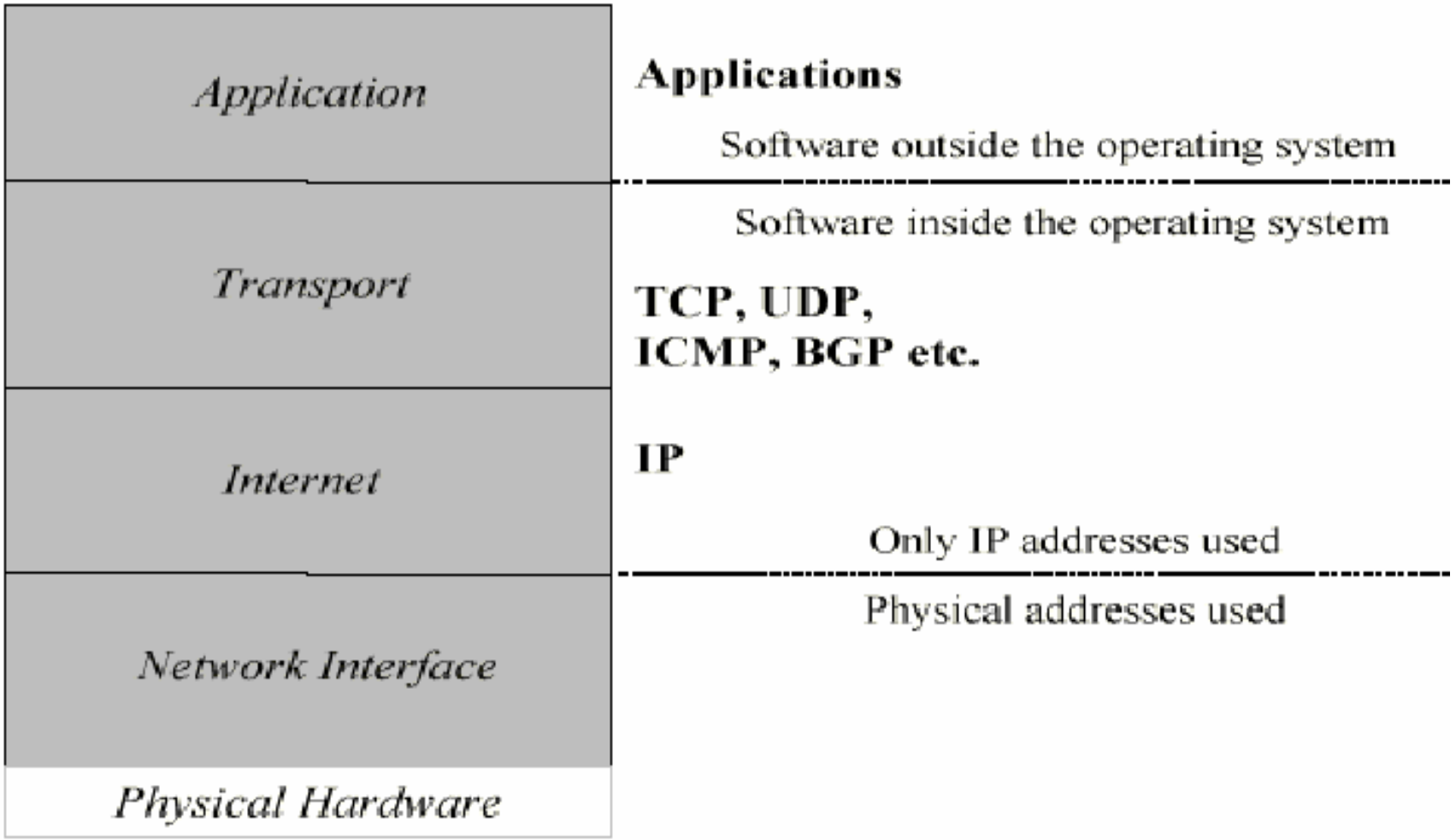
Entity Exchanged Between Layers



Communication in Layering



Conceptual Layering in TCP/IP Networks



IP Addressing

- Each host needs to have a unique *IP address*
  - Random addresses would make routing impossible
  - Neighbouring hosts should have similar addresses
  - Are hierarchical in nature
    - Aggregation is possible
- IPv4 Addressing
  - 32-bit addressing
- IPv6 Addressing
  - 128-bit addressing (RFC 3513)
- Address has two parts
  - *Net-id* and *Host-id*
  - Net-ids are assigned by a *central authority*

IPv4 Addressing

- IP address is given in the *dotted decimal notation*
  - 4 sets of 8-bit numbers: *A.B.C.D*
    - Viz: 144.16.160.2 , 127.0.0.1, 192,168.100.1
- Types of Addresses
  - Unicast, Multicast, Broadcast
- Special Addresses
  - All 0's -- host
  - All 1's -- limited broadcast
  - 255.255.255.255 (a string of 1's) refers to local broadcast
  - 127.x.y.z is used for loop-back address

IPv4 Addresses

- Class-A Address: 1.0.0.0 - 127.255.255.255
  - 0 + 7-bit net-id + 24-bit host-id
- Class-B Address:128.0.0.0 - 191.255.255.255
  - 10 + 14-bit net-id + 16-bit host-id
- Class-C Address: 192.0.0.0 - 223.255.255.255
  - 110 + 21-bit net-id + 8-bit host-id
- Class-D Address: 224.0.0.0 - 239.255.255.255
  - 1110 + 28-bit multicast group
- Reserved: 240.0.0.0 - 247.255.255.255

Private and Global IP Address

- Private Address
  - Used for private network : (viz: IIT-Bombay)
  - Unique only in the private network
  - Should not be advertised to the outside world
  - Class A - 10.0.0.0 to 10.255.255.255
  - Class B - 172.16.0.0 to 172.31.255.255
  - Class C - 192.168.0.0 to 192.168.255.255
- Global Address
  - Visible to outside world and unique globally
  - Outside connectivity for these addresses is through
    - Application level proxies
    - Network Address Translation (NAT)

IP Address Issues

- *Inefficient:* wasted addresses
- *Inflexible:* fixed interpretation
- Not scalable
  - Number of networks is growing
  - Not enough network numbers
- Solutions
  - *Sub-netting and Super-netting*
  - *CIDR:* Variable interpretations for the network number
  - *DHCP:* Dynamic host configuration
  - *IPv6:* 128-bit address space

Subnetting

- Breaks larger network to many smaller networks
- Reduces the broadcast domain
  - Hence network traffic within a subnet
- Simplifies network management
  - Smaller network => simpler management
- To create subnetwork bits from the *host portion* of the IP address is taken and used to define subnets
- For subnet scheme to work
  - Every machine on the network must know which part of the host address will be used as subnet mask.
    - Accomplished by assigning subnet mask to each machine

Subnetting

- When no subnetting : use default subnet mask e.g.
  - Class A : 255.0.0.0
  - Class B : 255.255.0.0
  - Class C : 255.255.255.0
- Only 8-bits available for hosts, possible subnet masks:
  - 10000000 = 128, 11000000 = 192, 11100000 = 224, 11110000 = 240
  - 11111000 = 248, 11111100 = 252, 11111110 = 254
- Cannot have only 1 bit for subnetting
- Also need at least 2 bits for hosts
  - With 1 bit, you cannot have a valid host address
    - The bit will be used up for network id and broadcast id
  - Hence only valid subnet masks are from 192 to 252

Subnetting

- Subnet id cannot be all zero or all ones
  - e.g. If subnet id is 192 (11000000)
    - Then only subnet id (01000000) and (10000000) are allowed
- Example subnet id: 01000000 (64)
- The subnet id : 01000000 = 64
- First valid host id : 01000001 = 65
- Last valid host id : 01111110 = 126
- Broadcast address : 01111111 = 127

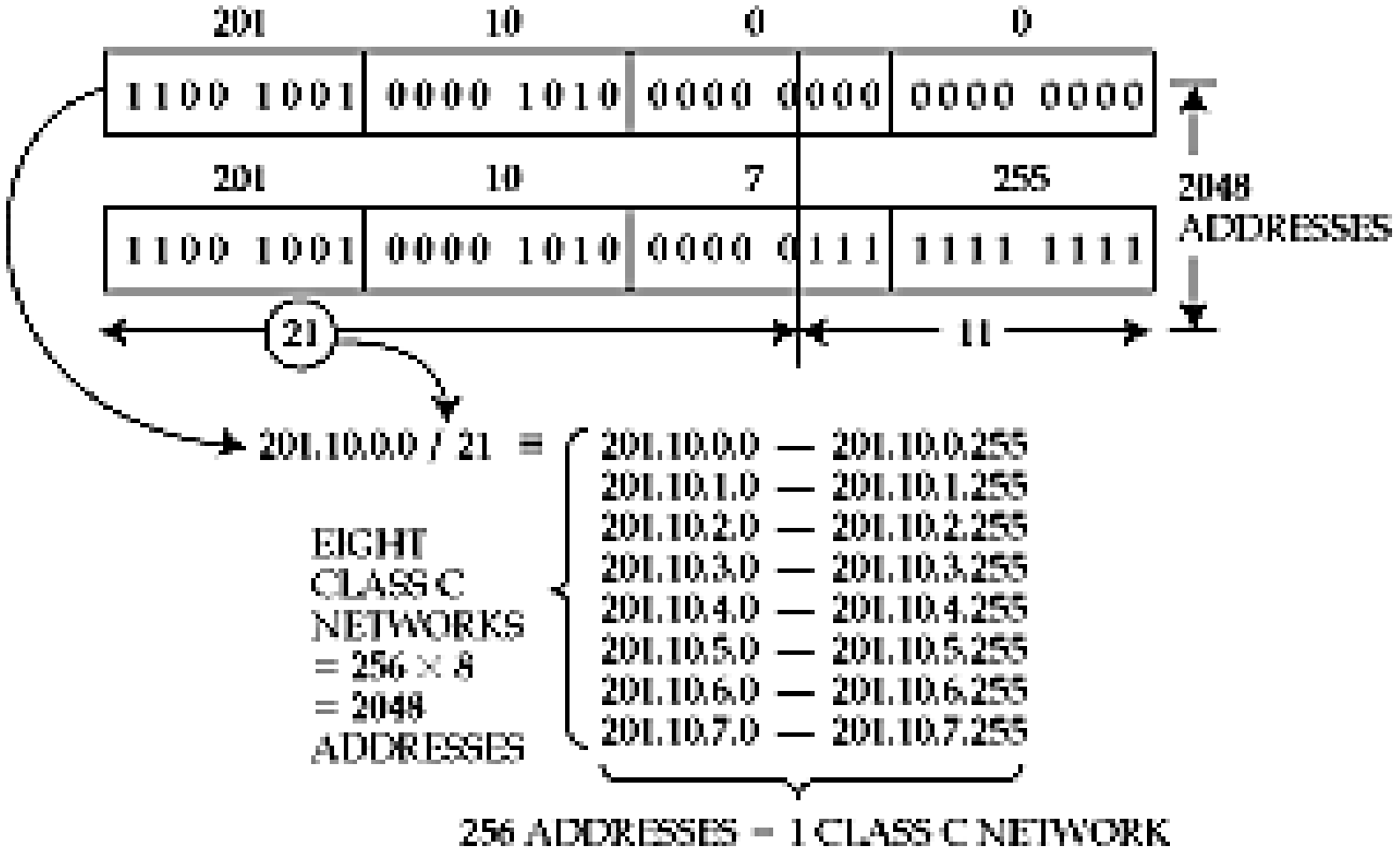
Supernetting

- Opposite of Subnetting
  - Merges several smaller blocks of continuous IP addresses (networks) into one larger block of addresses
- A site may get a continuous block of class C addresses
  - With a common prefix
- Instead of advertising all networks, *common prefix* is advertised
  - Reduces the sizes of routing tables and making the search hierarchical
    - Example: IP addresses 144.16.64.x, 144.16.65.x, 144.16.66.x and 144.16.67 may be clubbed together as 144.16.64.0
- Other option is to advertise a pair
  - consisting of the 1st network and the number of networks
- Only new routers implement this

Classless Inter Domain Routing (CIDR)

- New addressing scheme for the Internet
- Why CIDR?
  - Running out of IP addresses
  - Running out of capacity in the global routing tables
- Medium sized networks choose class B addresses, leading to wasted space
  - Allow ways to represent a set of class C addresses as a block, so that class C space can be used
  - Use a CIDR mask
- RFC 1518 and 1519

Classless Inter Domain Routing (CIDR)



Address Resolution Protocol: ARP

- *NIC* – is the interface for a PC to Internet
- Each NIC needs to be uniquely (globally) identified
  - Physical Address or *MAC Address*
  - 48-bit and unique
- The MAC Address can be of two types
  - Fixed and Configurable
- IP Address has to be mapped to the MAC Address
  - Needed for actual delivery of frames
  - This is called *Address Resolution*
- W.r.t. the MAC Address, the IP Address can be
  - Independent, as in Ethernet; mapping is dynamic
  - Dependent, as in proNET-10; mapping can be directly done

Address Resolution Protocol: ARP

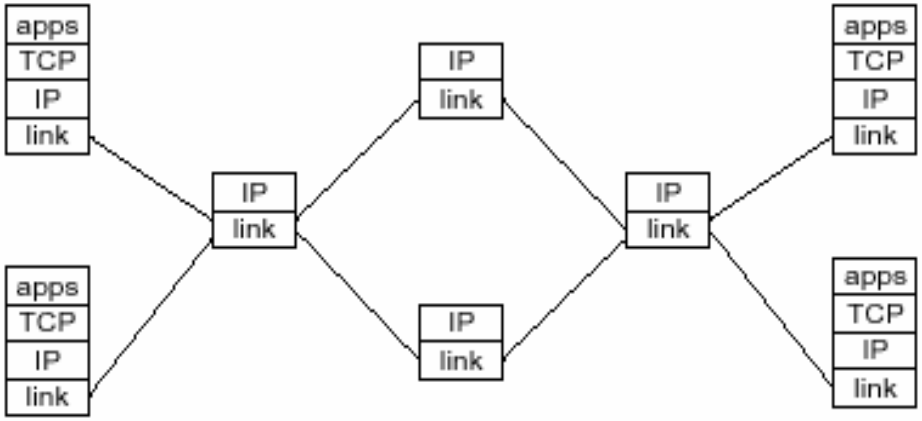
- Resolution through direct mapping is trivial
  - If MAC address is configured to be the host part of the IP address
- Resolution through *Dynamic Binding*
  - Required because NIC may be changed
  - Else, IP-MAC address mappings on all hosts will have to updated
  - TCP/IP uses ARP which consists of the following steps
    - Broadcast IP Address of the destination
    - Broadcast nature of the LAN is assumed
    - Destination replies with its MAC address
    - Source maintains a cache of IP-to-MAC Address bindings

Reverse ARP: RARP

- To obtain an IP address, given a MAC Address
  - Used by *diskless hosts* to obtain IP addresses
  - Third parties can also obtain mapping
- Client *broadcasts an RARP* Request with own MAC Addr
  - Same as ARP format
- RARP server looks up a table and responds
  - As routers do not forward broadcasts, an *RARP server is needed in every local network*
  - If there are multiple servers, only primary replies
  - If request is resent, secondary replies
    - Possibly the primary is failed (RFC 903)

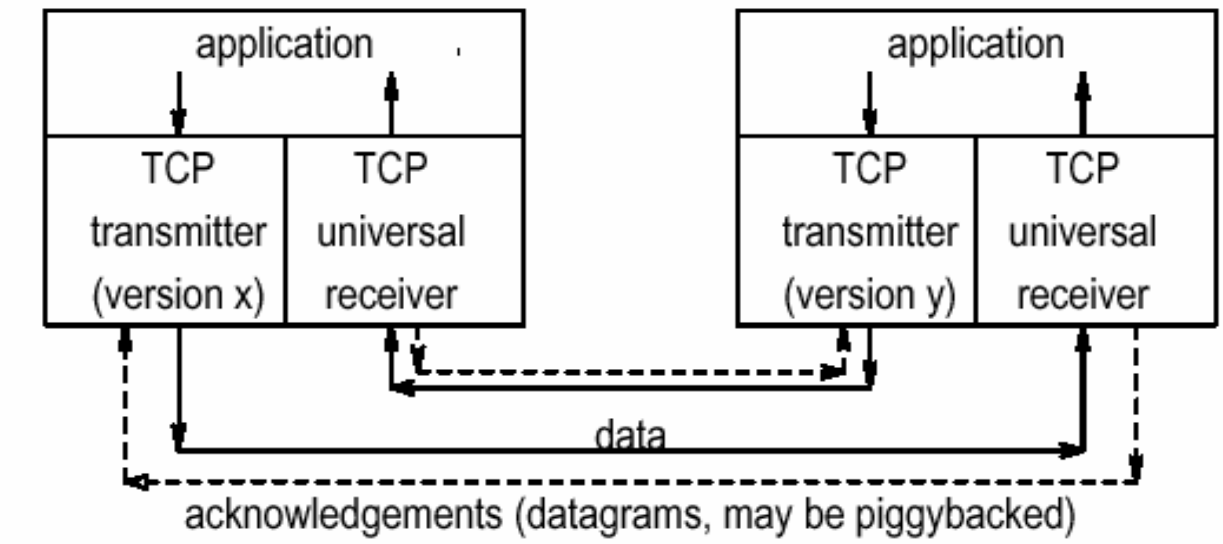
Functions of the TCP Protocol

- Provides sequenced, reliable data transport service
  - Using sequence numbers, and retransmission
- Sender-receiver flow control
  - Receiver advertises an acceptable window of packets
- Network congestion control
  - Each connection adapts transmission window



TCP is Full Duplex and Symmetric

- Txr and Rxr for each direction of communication
- Rxr protocol is universal
  - Same in all common implementations
- Different Txr and Rxr implementations can coexist
  - Even in the same connection – need sockets



Sockets

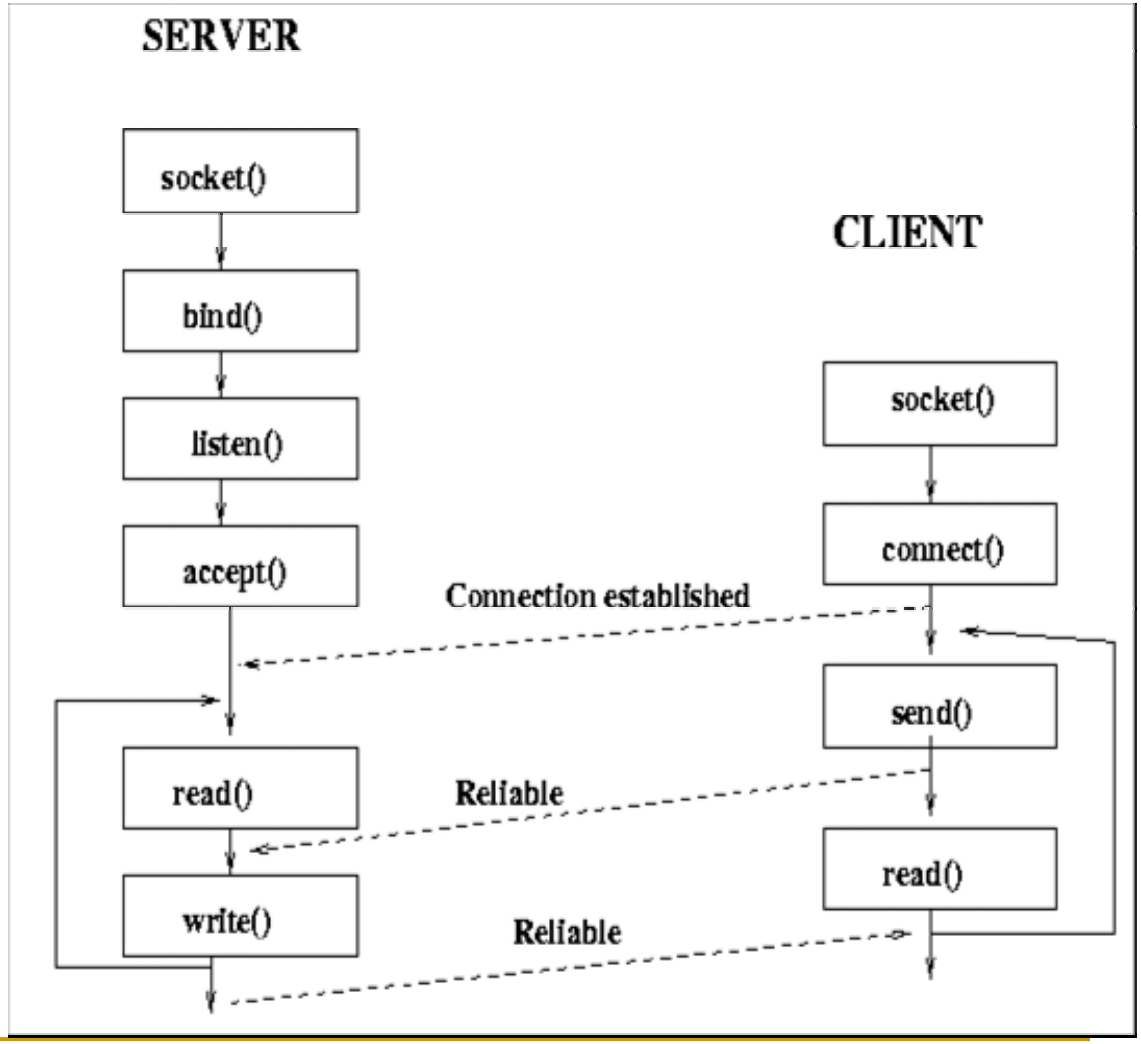
- What is it?
  - Combination of an IP address and a Port
  - Creates a new communication end-point
  - Connection is identified as *(socket1, socket2)*
- Why do we require?
  - For client-server communication
    - Viz.: Telnet, Ftp, HTTP etc.
- Types of Socket
  - *TCP Socket* - Stream Sockets and connection oriented
  - *UDP Socket* - Datagram Sockets and connection less
  - *RAW Socket*
    - Provide direct access to the lower-layer protocols,
    - Viz. IP and the Internet Control Message Protocol (ICMP)

Socket Primitives for TCP

- BIND
  - Attach a local address to a socket
- LISTEN
  - Announce willingness to accept connections; give queue size
- ACCEPT
  - Block the caller until a connection attempt arrives
- CONNECT
  - Actively attempt to establish a connection
- SEND
  - Sends some data over the connection
- RECEIVE
  - Receive some data over the connection
- CLOSE
  - Release the connection

Socket Programming

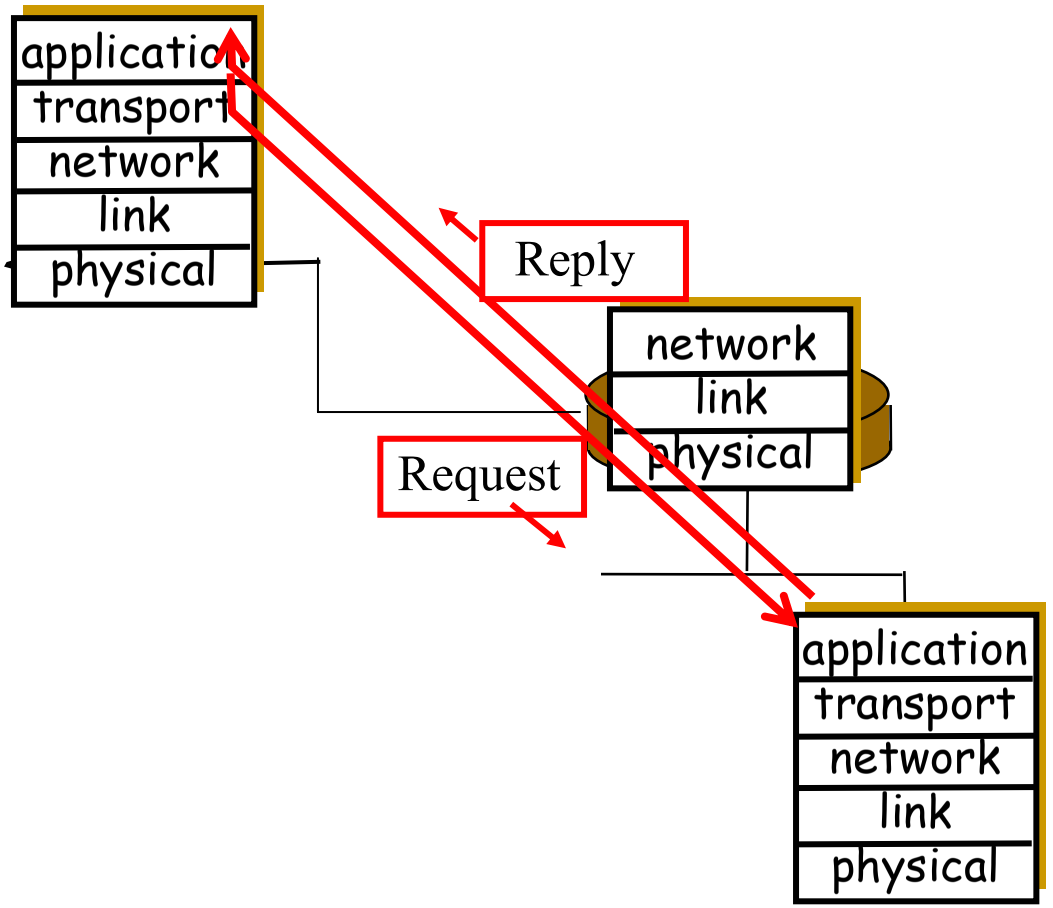
- Typical sequence
  - Allocate Local Resources
  - Specify communication End Points (ports)
  - Initiate/Listen for connections
  - Send/receive data
  - Terminate connections gracefully
  - Release resources
- Design Issues
  - For scalability, State full/Stateless servers, Concurrent, Iterative service



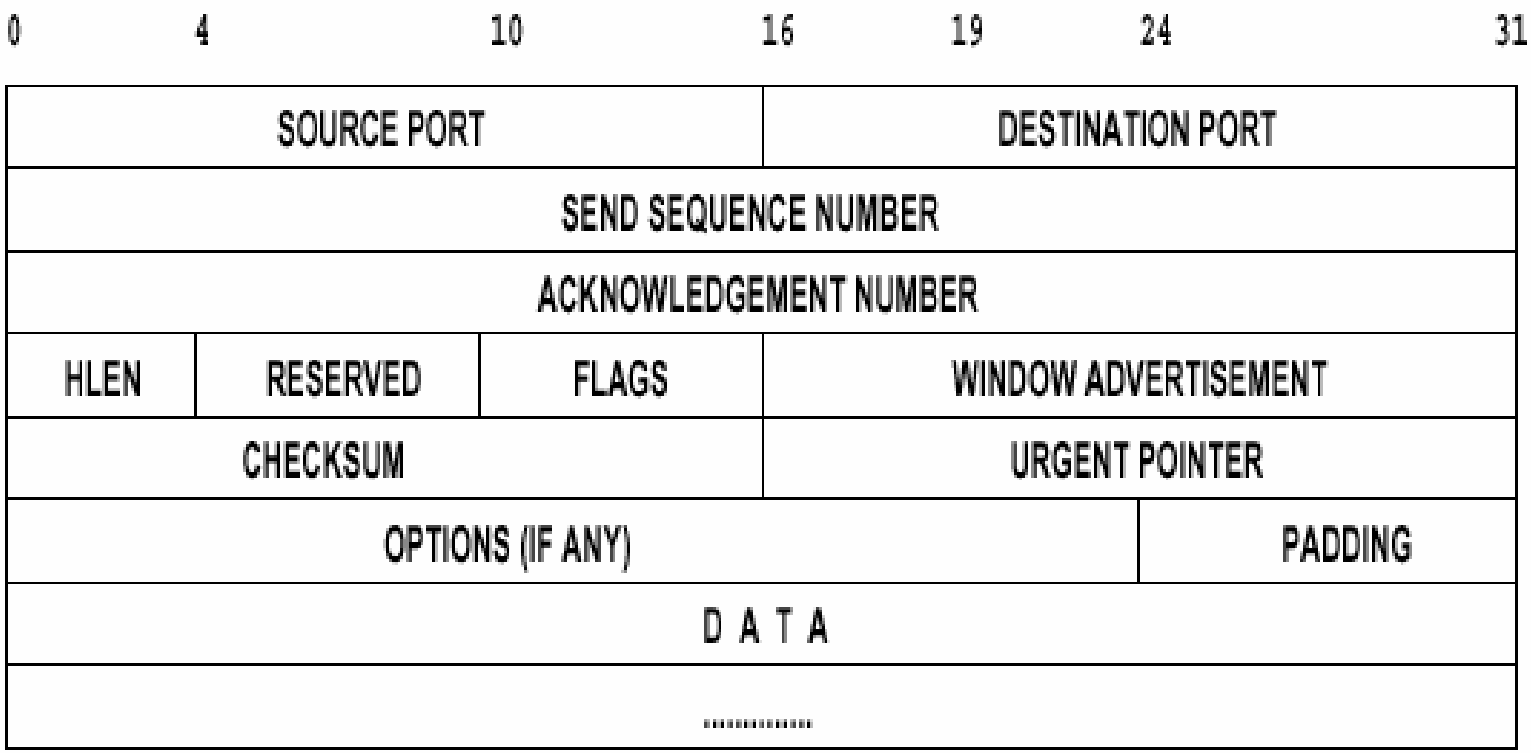
TCP Port Numbers

- *Well-known ports* (below 256) are reserved for standard services: (Port no. in decimal):
  - 20: File Transfer Protocol (data)
  - 21 File Transfer Protocol
  - 23 Telnet
  - 25 SMTP (Mail Transfer)
  - 37 Time
  - 42 Host Name Server
  - 53 Domain Name Server
  - 79 Finger
  - 119 NNTP (News Transfer Protocol)

Client Server Paradigm



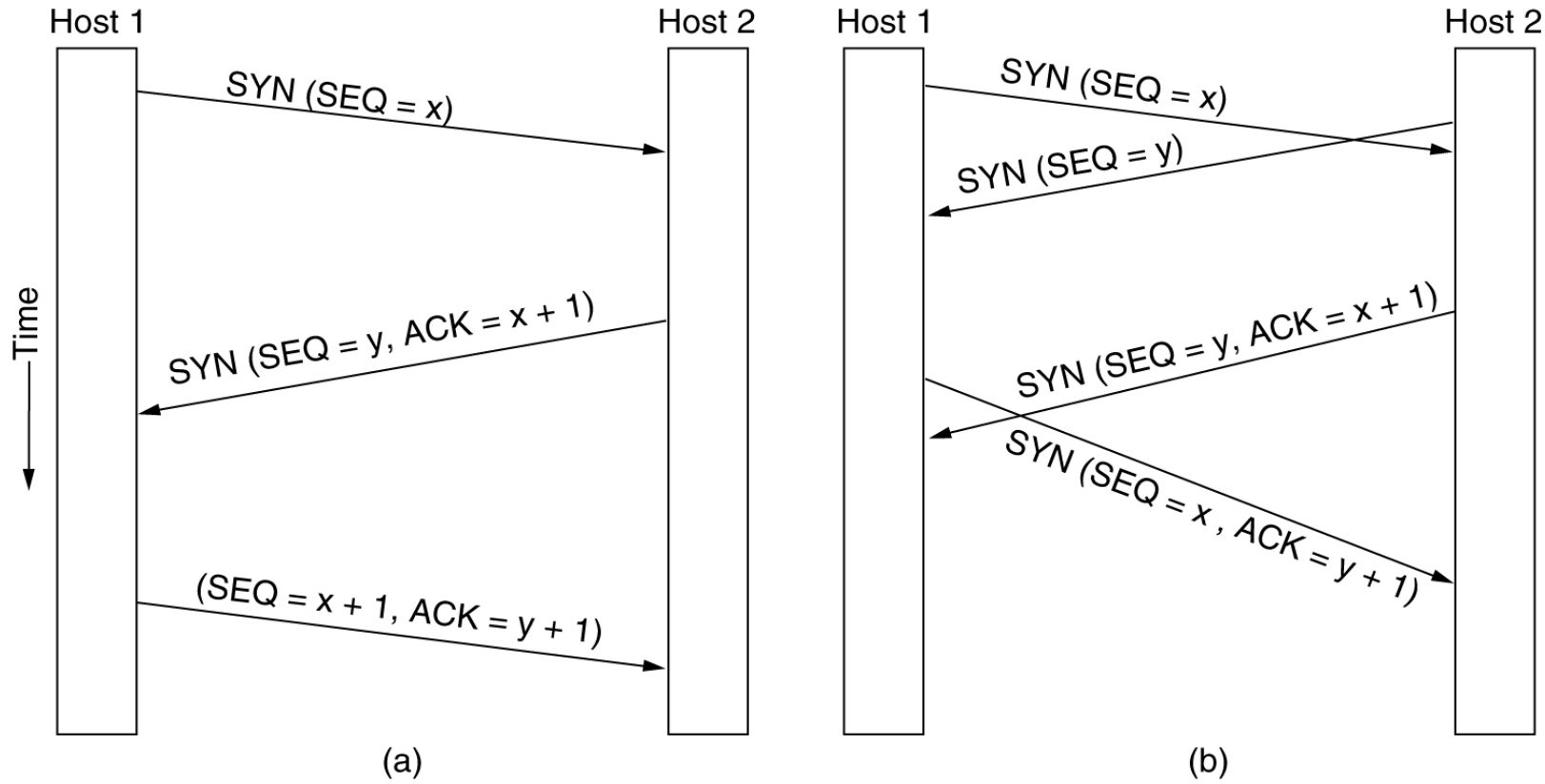
TCP Header



Each TCP connection is between Src Port No. + Src IP Address, and dstn Port No. + dstn IP Address

Port numbers permit multiplexing of several connections into one IP Addr

TCP Connection Establishment



- (a) TCP connection establishment in the normal case
- (b) Call collision

Three-way Handshaking in TCP/IP

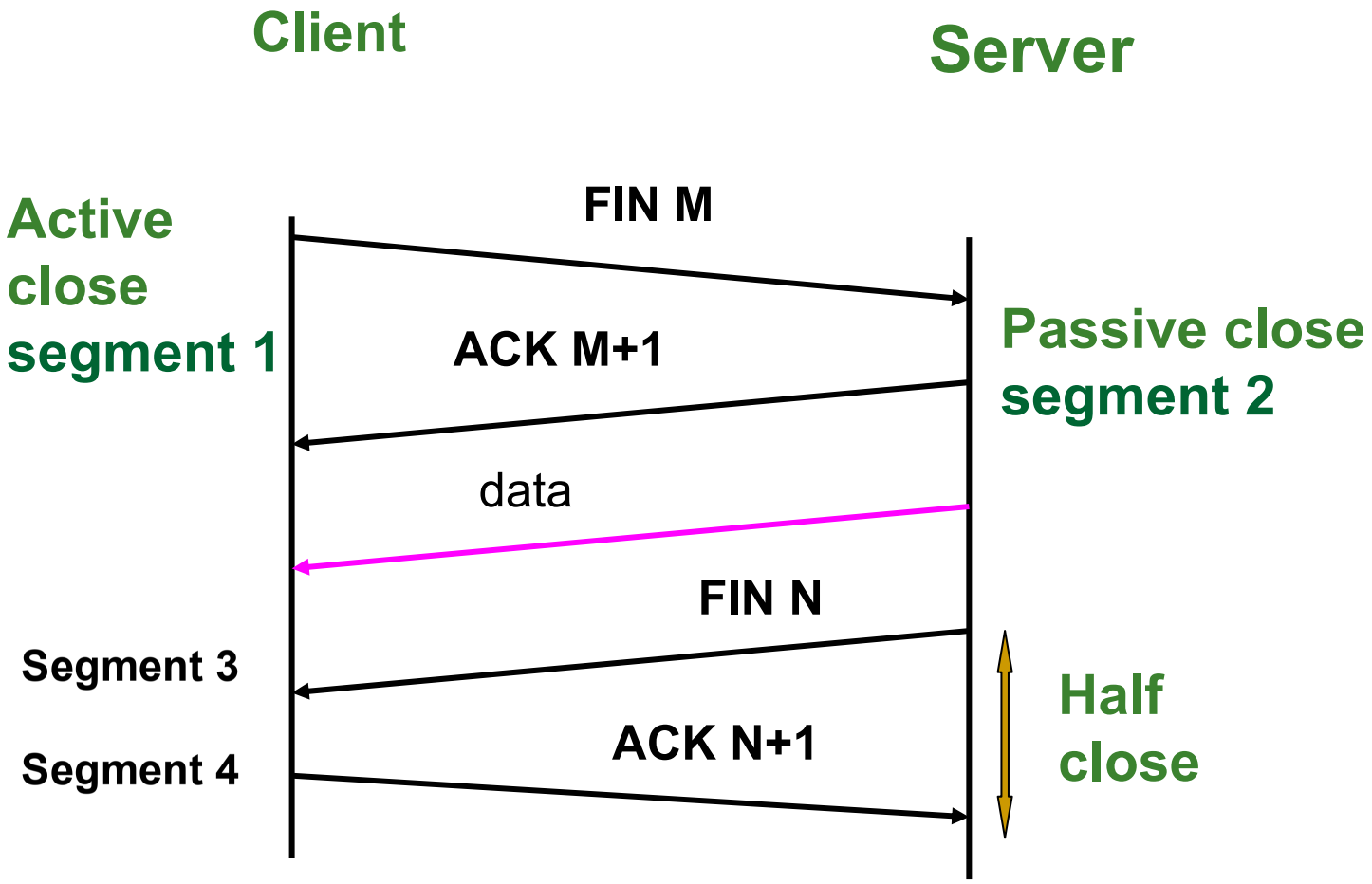
- Server does a *passive open*, and Client *active connect*
- Two party wants to agree on a set of parameters
  - New Initial Sequence number for both sides, MSS size
- Three Way Handshaking Steps
  - *Step 1: Client (active party) send a TCP segment with flag==SYN and sequence number = x, destination port number, MSS*
  - *Step 2: Server responds with flags = SYN + ACK, sequence number = y, Ack = x+1, MSS,*
  - *Step 3: Client responds with flag = ACK, Ack = y+1*

[Ack is set 1+ sequence number -- to tell what is the next sequence number expected and that all earlier sequence number is received]

Three-way Handshaking in TCP/IP

- Why not start with fixed sequence number?
  - Sequence Number (32 bit) starts randomly for each connection
- Takes care of problems of duplicate SYNs, ACKs
- *Comments*
  - Initial sequence number is not zero, a random 32 bit number
    - This is to avoid duplicate SYNs
  - After a crash, host is required to not reboot for the maximum packet lifetime
    - Wait for MSL time before generating or responding any TCP control message like SYN or ACK
    - This makes packets from previous connections die
  - The sequence number can be randomized based on clock
    - Clock is assumed to be running even if host crashes

TCP Connection Release



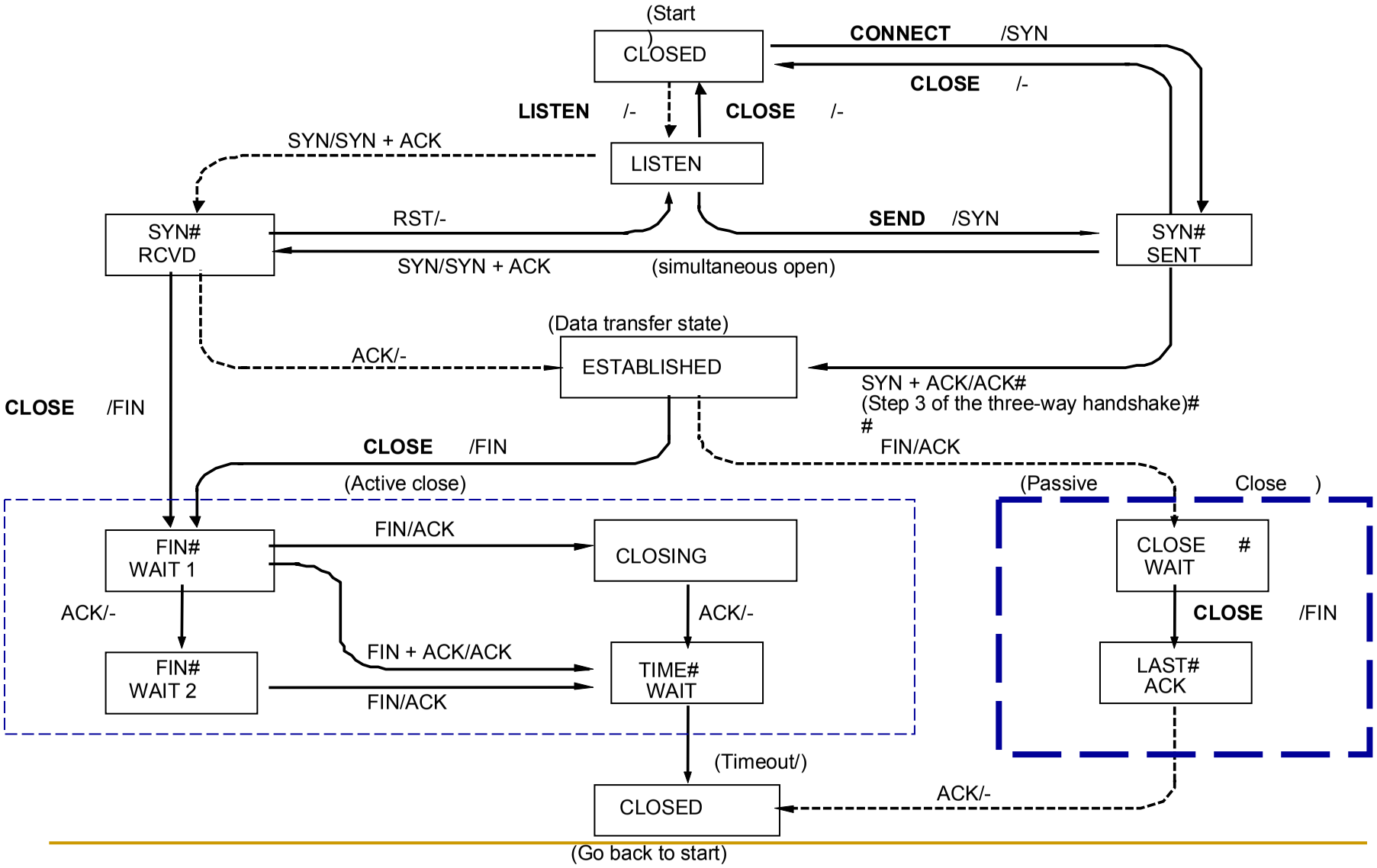
TCP Connection Release

- Two way release is necessary
  - Each end of the full-duplex connection must be closed *independently*
  - Either side can send *FIN* when it is done sending data
  - Both ends maintain state related to connection.
  - On end of data communication these states should be released
- Half Close (One way close)
  - Side A sends *FIN* (with sequence number)
  - Receive *FIN-ACK*, shut down --> Half close
  - Half close state
    - Side A cannot send data to Side B, but vice versa possible

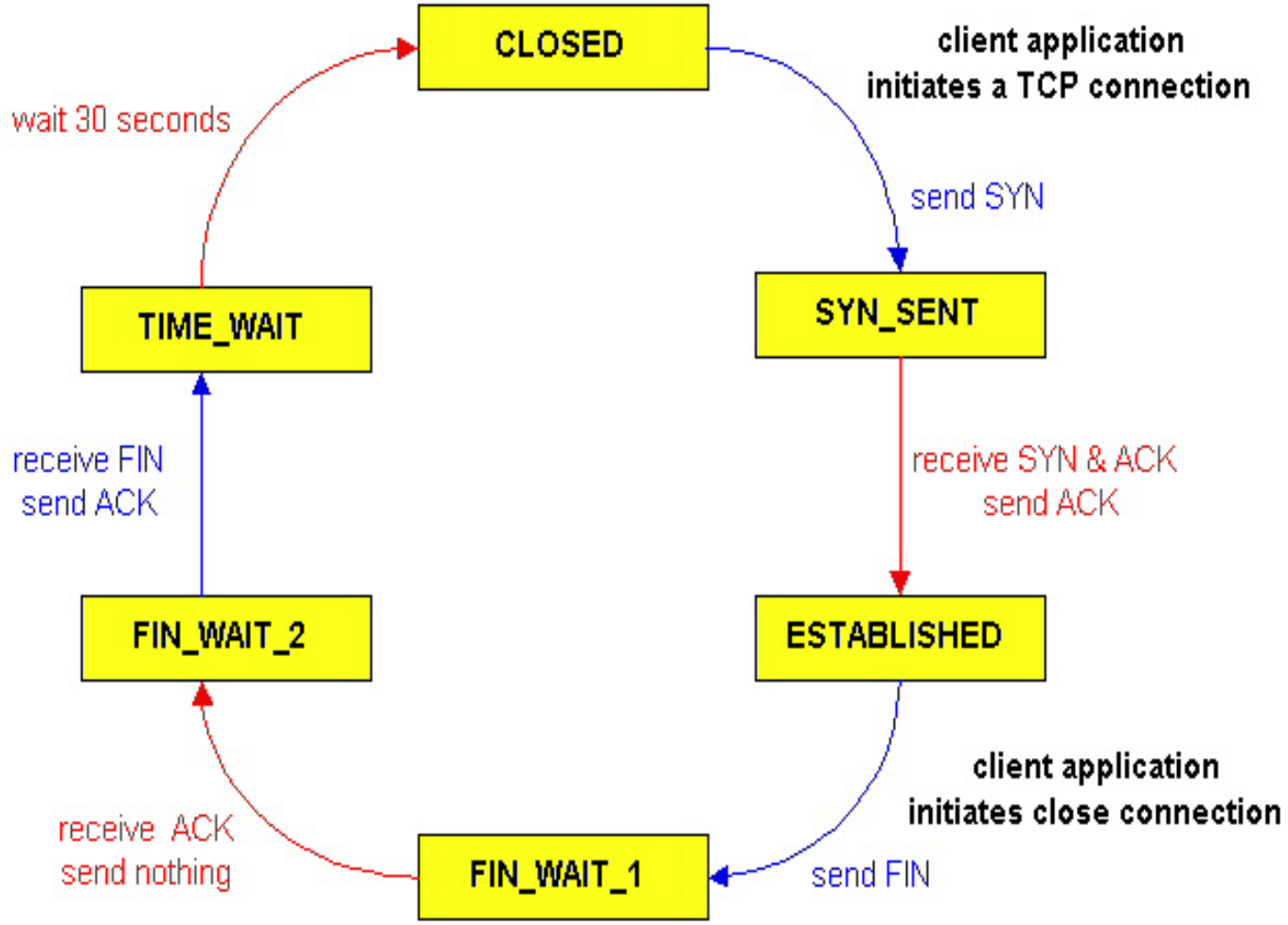
TCP Connection Release

- Full close (Both way close)
  - Side B also send *FIN* (Can piggy-back on FIN-ACK)
    - Side A Send FIN-ACK in the other direction
  - Lost FINS etc, taken care of by timers
    - If FIN/ FIN-ACK is lost after several attempts, sender (active closer) releases connection (other side will time-out and release)
  - *Wait for some time*
    - The active closer waits for a fixed time after sending FIN-ACK and before actually closing connection (twice max segment lifetime, MSL)
- TCP cannot reallocate the socket pair till 2MSL
  - 2MSL wait protects against delayed segments from the previous “incarnation” of the connection

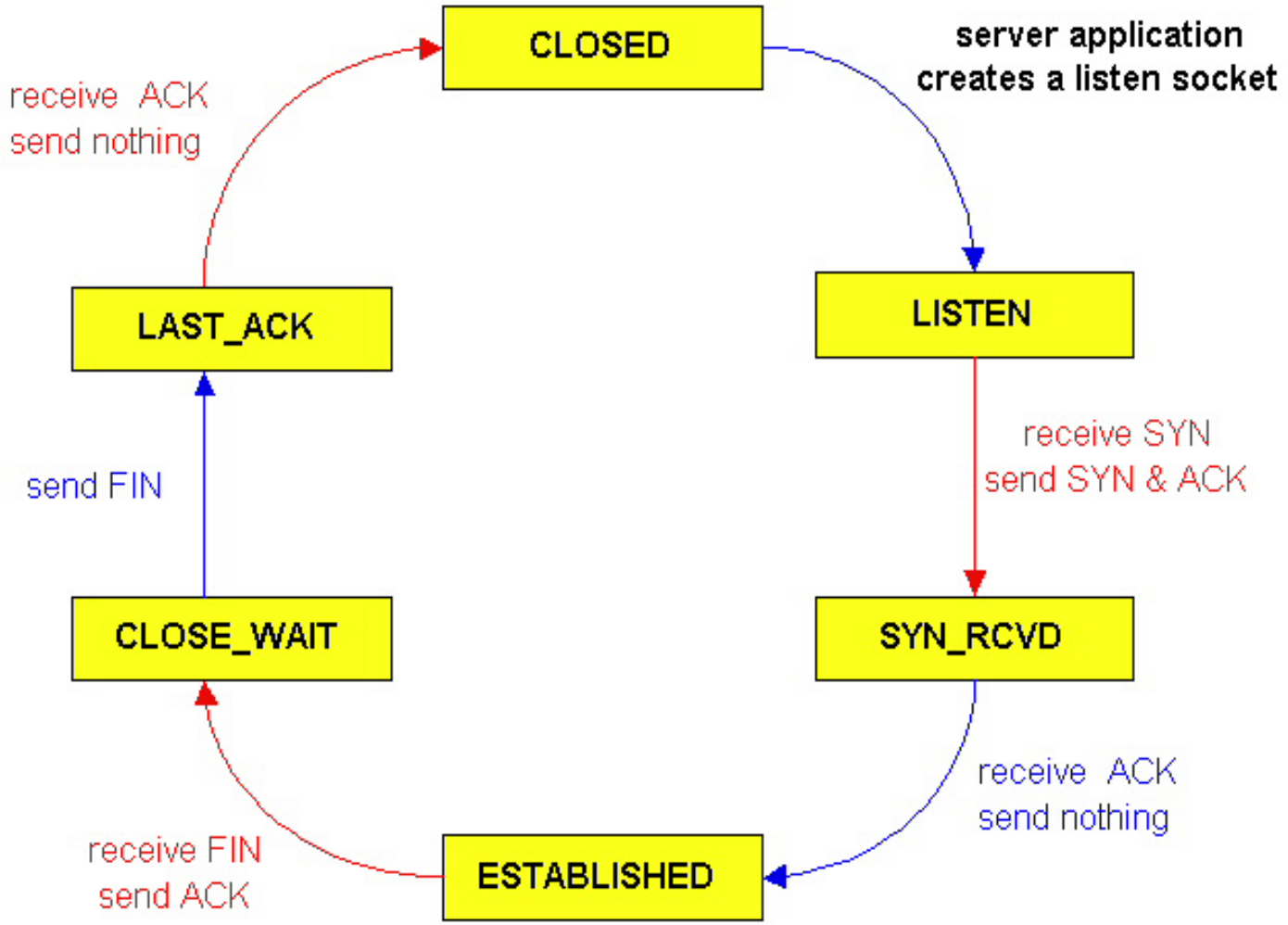
State Diagram for TCP



TCP Client States



TCP Server States



TCP Window Adaptation

- At transmitter, at time t, W(t) = transmitter's congestion window
  - If an acknowledgement is received, increase W(t)
  - If loss indication (i.e., timeout) drop W(t)
- Desirable round-trip window may be much larger than buffers at each hop
  - If start with the full window, will surely lose packets
- Slow Start
  - Initially increase is fast (exponential), then linear
- If there are many connections, then they should share
  - Thus windows should adapt when connections arrive or depart
- Adaptation will be forced by packet loss/duplicate acks

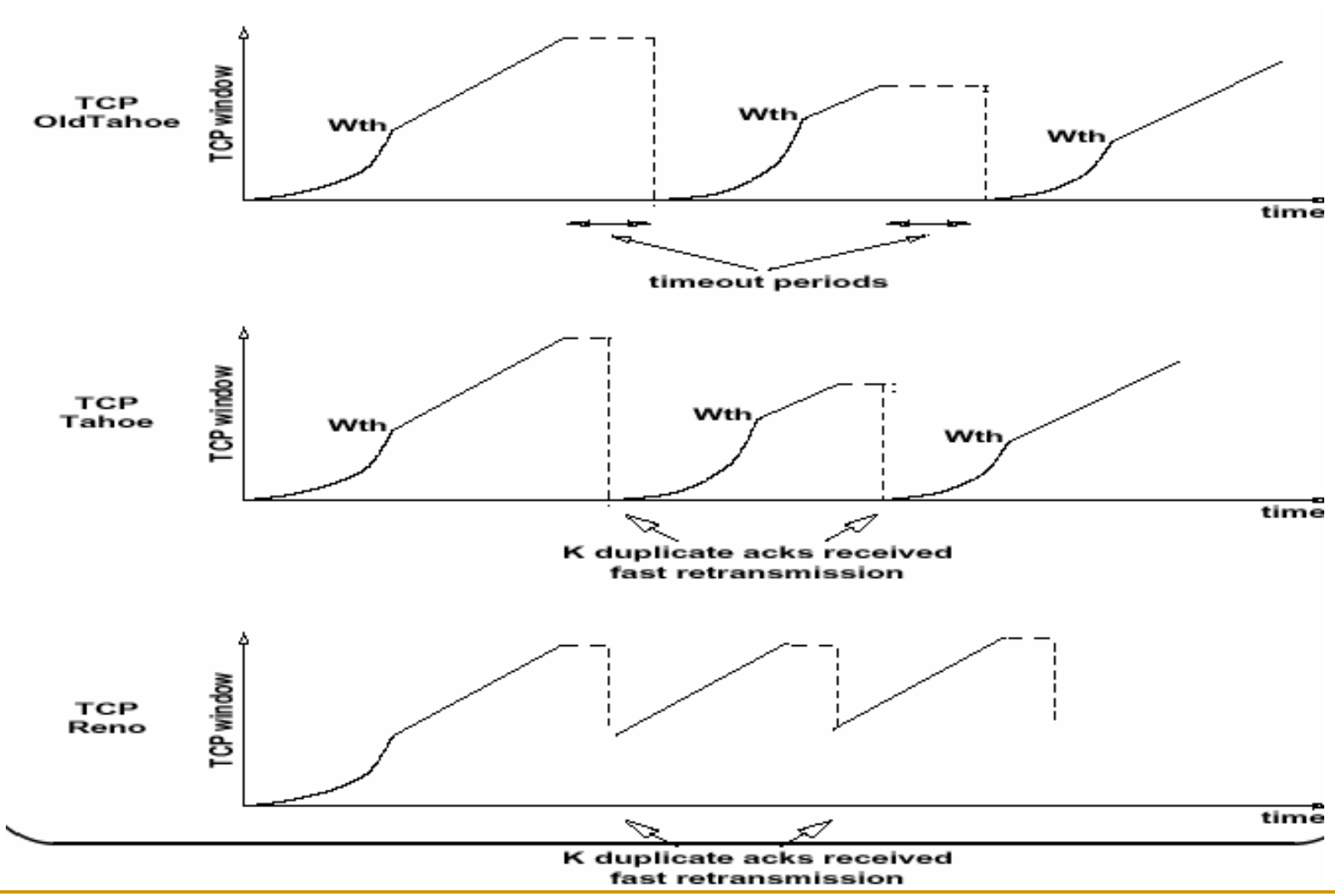
TCP Congestion Control

- Why Congestion?
  - Nodes do not have enough processing capability / incoming traffic is faster than outgoing traffic
- Effect of Congestion
  - Delay and Packet Loss in the network
- Solutions
  - Congestion Control
    - Control after congestion (Reactive), Viz.: TCP Tahoe, TCP Reno etc.
  - Congestion Avoidance
    - Control before congestion (Proactive), Viz.: TCP Vegas, RED etc.
- Congestion control strategies
  - Adaptive Window Management technique

Congestion Control in TCP

- Fast-retransmit
  - For packet loss (time-out) drop the *cwnd* window size to 1
  - For 3 duplicate acks, drop the *cwnd* window size as half of *cwnd* size
  - $W_{th}(t^{+}) = [W(t)/2]$
- Fast-recovery
  - In case of fast-retransmit, do not drop congestion window to 1
  - Set it to $W_{th}(t^{+}) = [W(t)/2]$
- TCP Reno and Reno-2
  - In Reno, *cwnd* decremented for each mark (*loss or 3 dupacks*)
  - In Reno-2 *cwnd* does not decremented for each mark
    - Decremented once per window or RTT
  - Reno-2 is good for wireless link

Congestion Control in TCP



Elements of Routing

- Performance Criteria
  - Number of hops, Cost, Delay, Throughput
- Decision Time
  - Packets (datagram), or Session (virtual circuit)
- Decision Making Node
  - Central (centralised), Each (distributed), or Originating node
- Network Information Source
  - None, Local, Adjacent node, Nodes along route, or All nodes
- Routing Strategy
  - Fixed, Flooding, Random or Adaptive
- Adaptive Routing Update Time
  - Continuous, Periodic, upon Load change, or upon Topology

Routing Algorithms

- Used to set up the proper routing paths from source nodes to destination nodes
  - Mapped as appropriate entries in ***Routing Tables*** maintained at the nodes of the network.
- Decides which output line an incoming packet should be transmitted on, packet by packet
- Should be fair, robust, correct, simple, optimal and efficient
- *Optimise metrics* such as delay, throughput, no. of hops
- Can be *static or dynamic*
  - Adapts to changes in the network

Routing Mechanism

- *Hop-by-hop Routing*
  - Each router selects its output link for the destination using its routing table
- *Source Routing*
  - Source node puts the complete route e.g. A – B – C – E for the path from A to E in the packet header
- *Hot–Potato Forwarding*
  - Forwards on *shortest output queue*
  - Simple, may not reach destination
- *Flooding*
  - Forwards packet on all links except the one it came on
  - Very reliable, *exponential number* of packets
- *Selective Flooding*
  - Router discards duplicates of a packet transmitted once by labelling every packet with source id and sequence number
  - Forwards only in the *right direction*

Interior & Exterior Routing

- *Interior Routing*
  - Interior Routing occurs *within an autonomous system*
    - Viz.: RIP and OSPF
  - The basic routable element in this case is the *IP network or subnetwork*
- *Exterior Routing*
  - Exterior routing occurs *between autonomous systems*
    - Viz.: BGP
  - The basic routable element here is the Autonomous System identified by an *Autonomous System Number (ASN)*
- While there may be many different interior routing schemes, a *single exterior routing system* manages the global Internet

Routing Table: Example

```
[hemantr@vani hemantr]$ /sbin/route -n
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
10.107.1.13	0.0.0.0	255.255.255.255	UH	0	0	0	eth0
10.107.1.12	0.0.0.0	255.255.255.255	UH	0	0	0	eth0
10.107.2.1	0.0.0.0	255.255.255.255	UH	0	0	0	eth0
10.107.1.5	0.0.0.0	255.255.255.255	UH	0	0	0	eth0
10.107.1.4	0.0.0.0	255.255.255.255	UH	0	0	0	eth0
10.107.1.7	0.0.0.0	255.255.255.255	UH	0	0	0	eth0
10.107.1.6	0.0.0.0	255.255.255.255	UH	0	0	0	eth0
10.107.1.3	0.0.0.0	255.255.255.255	UH	0	0	0	eth0
169.254.0.0	0.0.0.0	255.255.0.0	U	0	0	0	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo
0.0.0.0	10.107.1.11	0.0.0.0	UG	0	0	0	eth0

Devices for Interconnection

- *Repeater* - Connects two LANs at the Physical layer level
- *Hub* - Multi-port Repeater, i.e. connects multiple LANs
  - When a packet arrives at one port, it is copied to the other ports so that all segments of the LAN can see all packets
- *Bridge* - Connects LANs at the Data Link layer level
  - Connects two LANs or two segments of the same LAN
    - The two LANs may be alike or dissimilar
    - e.g., a bridge can connect an Ethernet with a Token-Ring LAN
  - May be needed due to the physical distance between two LANs
  - May be because providing so many hosts may not be possible on one LAN
  - Provides security against promiscuous mode

Devices for Interconnection

- *Router* - Connects networks at the Network layer
  - Connects at least two networks
    - Decides routes for packets, based on destination address and network topology
  - Exchanges information with other routers
    - To learn network topology
  - Maintains a *Routing Table*
    - Available routes and their conditions
  - Uses table along with path cost algorithms
    - Determine the best route for a given packet

Devices for Interconnection

- *Gateway* - Connects networks at Transport or higher
  - Gateway can be a Router
  - A network point that acts as an entrance to another network
  - Often acts as a proxy server and a firewall server
- *Switch* - Implementation of a Bridge/Router/Gateway in silicon - as a Layer 2 / Layer 3 / Layer 4….. Switch
  - When a packet arrives at one port, it is copied to the other ports so that all segments of the LAN can see all packets
  - Layer 3 switches (IP switches) also perform routing functions

Thanks