**Department of Electrical Engineering**
**University of Arkansas**

# ELEG 5693 Wireless Communications
# Ch. 9 Wireless Netowrks

**Dr. Jingxian Wu**
**wuj@uark.edu**

# OUTLINE

- **Ad hoc wireless networks**
- **Protocol layers**
- **Cross-layer design**

# AD HOC WIRELESS NETWORKS

- **Ad hoc wireless network**
  - A collection of wireless mobile nodes that self-configure to form a network without the aid of any established infrastructure.
  - Ad hoc: with little or no planning, fashioned from whatever is immediately available
  - Different from: infrastructure-based network (such as cellular network)
  - Allow people and devices to seamlessly internetwok in areas with no preexisting communication infrastructure.
    - No pre-installed basestations
  - Self-organizing, Rapidly deployed
    - Nodes cooperate to provide connectivity ➔ multihop relay.
    - Easily enable instantaneous person-to-person, person-to-machine, or machine-to-person communications.
  - Self-healing
    - Even we one or some nodes break down, the network can still operate.

# AD HOC WIRELESS NETWORKS

- **Examples**
  - Ad-hoc mode of IEEE 802.11
    - IEEE 802.11 (Wireless LAN) has two modes
      - Infrastructure mode: wireless router, laptops
      - Ad-hoc mode: no wireless router.
        - » Laptops can directly talk with one another without router.
  - Mesh extension of IEEE 802.16
    - Regular operation of IEEE 802.16 (WiMax) requires basestation.
    - Mesh extension can provide services to users that do not have good coverage from BS.
      - Signals are relayed from other users.
  - Soldiers equipped with multimode mobile communication devices on battle field
    - No fixed wireless base station or pre-installed infrastructrue are needed.
  - Small vehicular devices equipped with sensors in hostile environment to collect data.
    - Data are relayed by the sensor nodes.
  - Ship-to-ship ad-hoc communication
    - Doesn't need the assistance from satellite.
  - Police, rescue, etc.

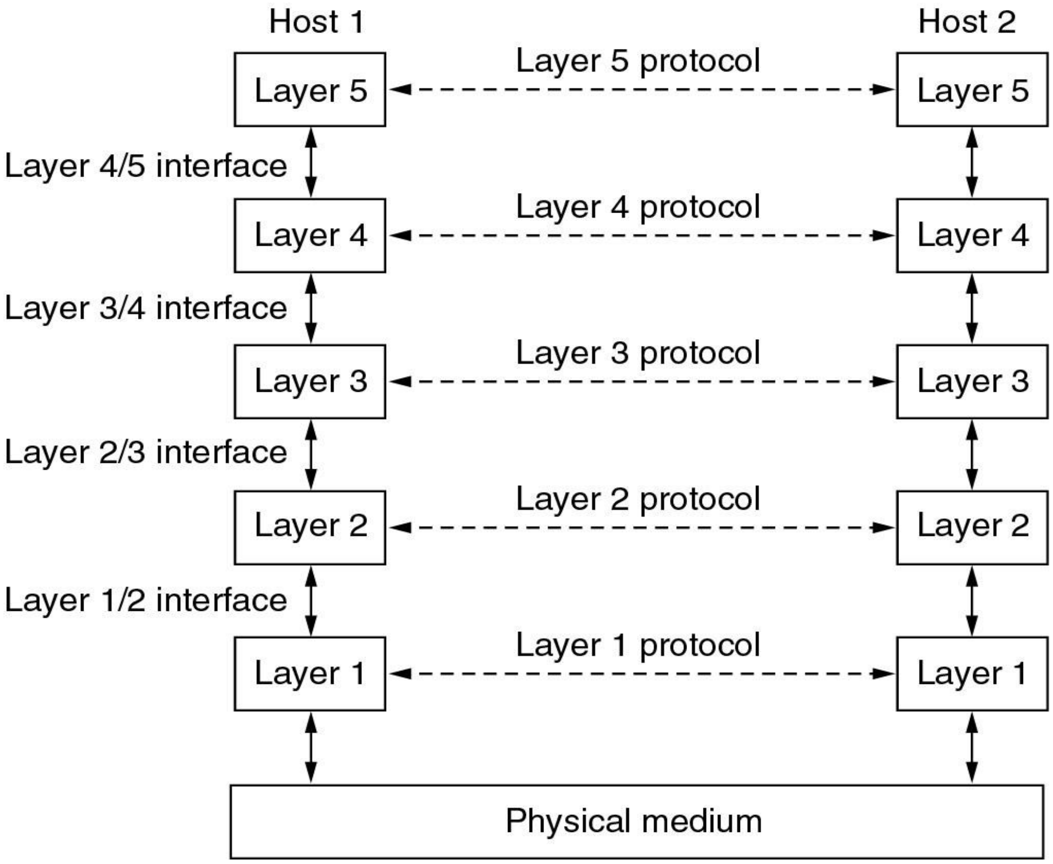
# AD HOC WIRELESS NETWORKS

- **Evolution**
  - PRNET (packet radio network) : the first MANET
    - Sponsored by DoD (Department of Defense), launchened in 1972
    - Evolved into survivable adaptive radio networks (SURAN) in 1980s
    - Goal: provide packet-switched networking to mobile battlefield elements in an infrastructureless, hostile environment.
    - MAC layer: combination of ALOHA and CSMA (carrier sensing multiple access)
    - Network layer: a variation of distance vector routing.
    - Handled datagram traffic reasonably well.
  - Ad-hoc mode of IEEE 802.11: 1990s
  - NTDR (Near-term digital radio): sponsored by DoD
    - Used by U.S. Army in late 2002
    - Clustering, link-state routing, self-organized two-tier structure.
  - Still largely an R&D activity
    - IETF MANET workgroup: two protocols will become “proposed standards”

# OUTLINE

- Ad hoc wireless networks
- **Protocol layers**
- Cross-layer design

- **Network software is organized as a stack of layers**
  - Modular architecture makes development easier and cheaper

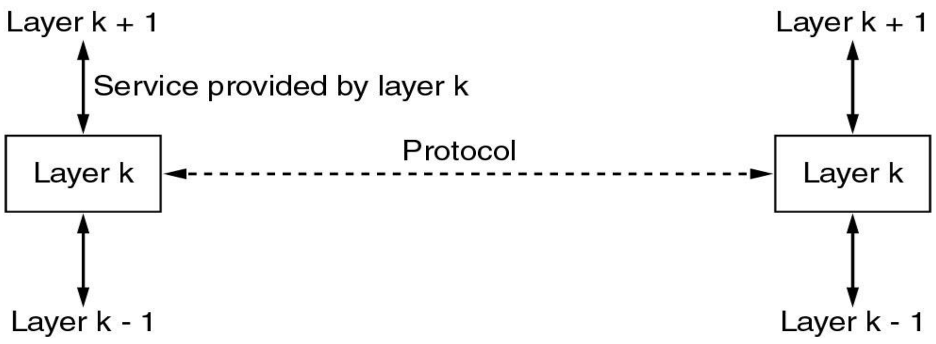


# INTRODUCTION: SOFTWARE AND PROTOCOLS

- **Layered structure: vertical**
  - Each layer provides services to the layers above
    - Via the interface between layers.
    - Details of the implementation of each layer is hidden to the higher layers → Data Encapsulation.
  - Between each adjacent pair of layers is an interface
    - Interface clearly defines the services the lower layer makes available to the upper layer.
    - Clear cut interface make it simpler to change layer implementation → different host can use different implementation.
  - A layer in a machine is represented by an entity in that machine.
    - Entity (SW or HW) does the function of that layer.

# INTRODUCTION: SOFTWARE AND PROTOCOLS

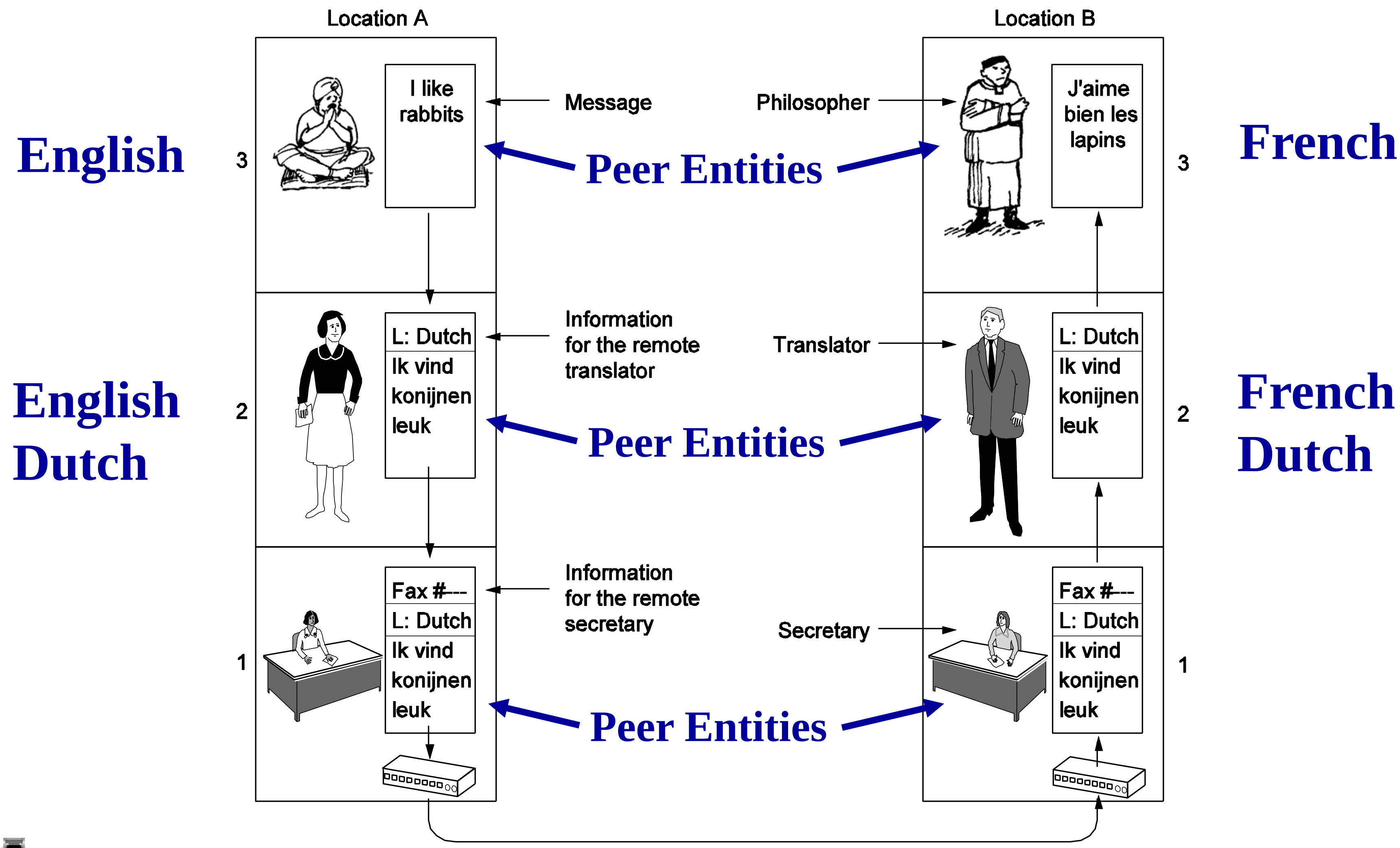
- **Layered structure: horizontal**
  - Peer Entities
    - Entities in the same layer, but in different machines.
    - Layer *n* on one machine carries on conversation (exchanges information) with layer *n* on another machine.
    - Peer entities communicate with each other using protocol.
  - Protocol: An agreement between communication parties on how communication is to proceed.
    - One of the most important concepts in computer network!
    - Each layer has its own protocol.
    - The protocol of a certain layer can only be understood by its peer entities.
  - Physically, peer entities do not talk with each other directly.
    - Using the layers below them until the physical medium.
    - Actual communication occurs through physical medium




UNIVERSITY OF ARKANSAS

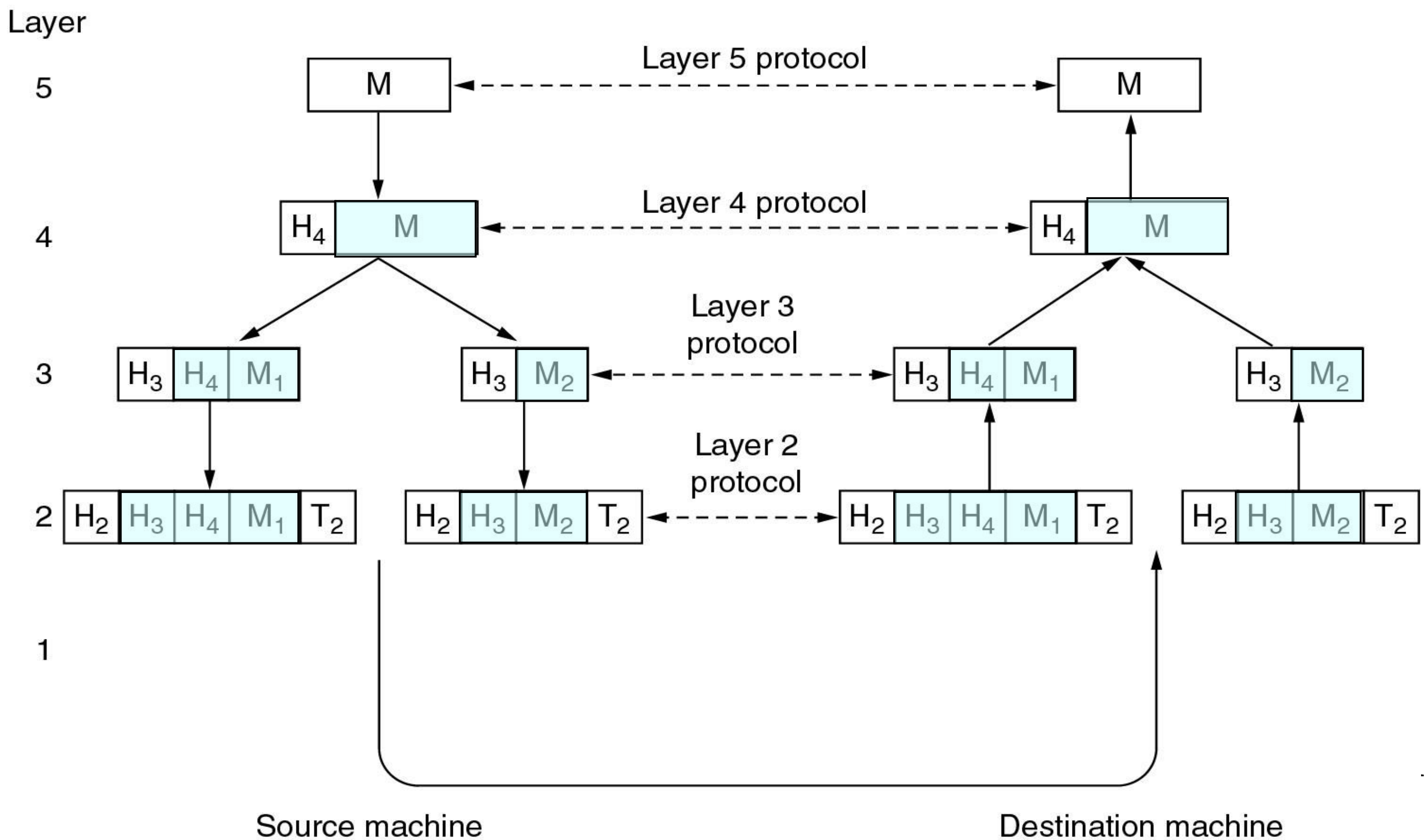
# INTRODUCTION: SOFTWARE AND PROTOCOLS

- The philosopher–translator–secretary architecture



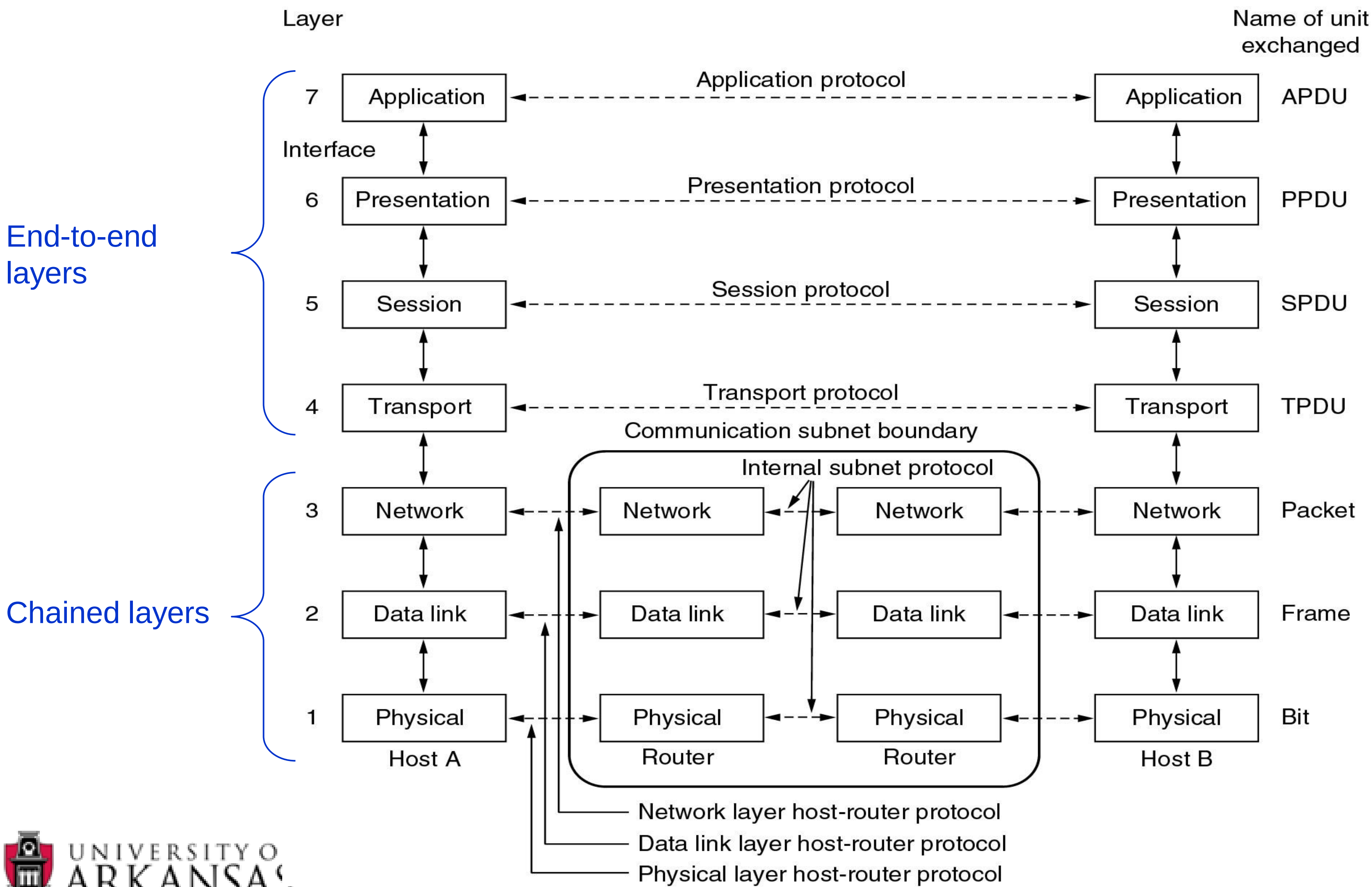
# INTRODUCTION: SOFTWARE AND PROTOCOLS

- **Protocol Stack: A list of protocols used by a certain system, one protocol per layer.**
- **Protocol Data Unit (PDU):**
  - A unit of data that is specified by a protocol of a given layer and delivered among peer entities.
  - Consisting protocol control information of given layer and possibly data of that layer.



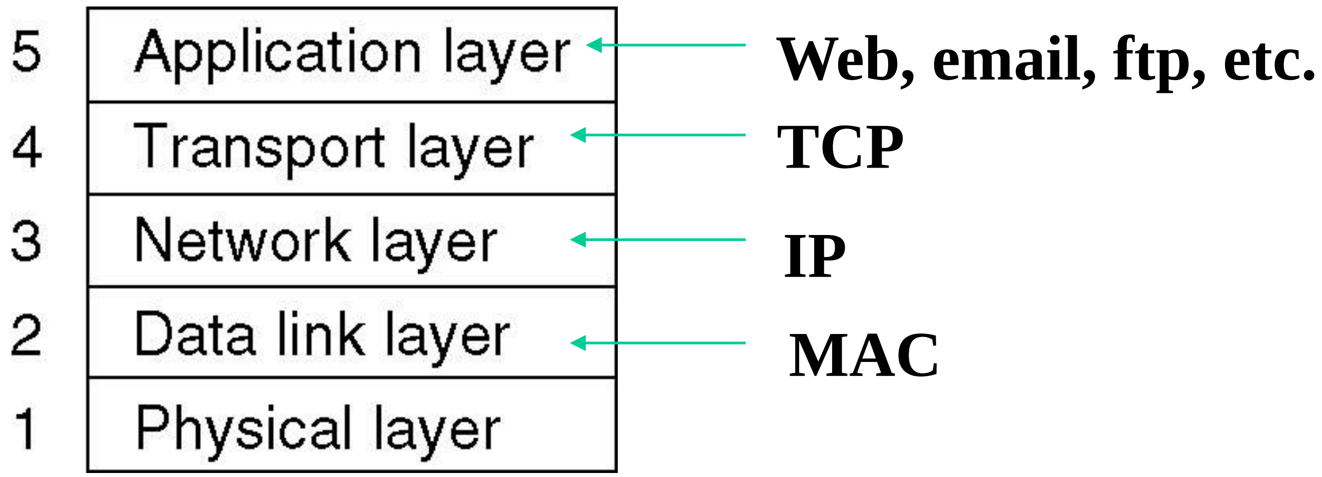
# INTRODUCTION: SOFTWARE AND PROTOCOLS

- **Open System Interconnect (OSI) model**



# INTRODUCTION: SOFTWARE AND PROTOCOLS

- **A simplified model**



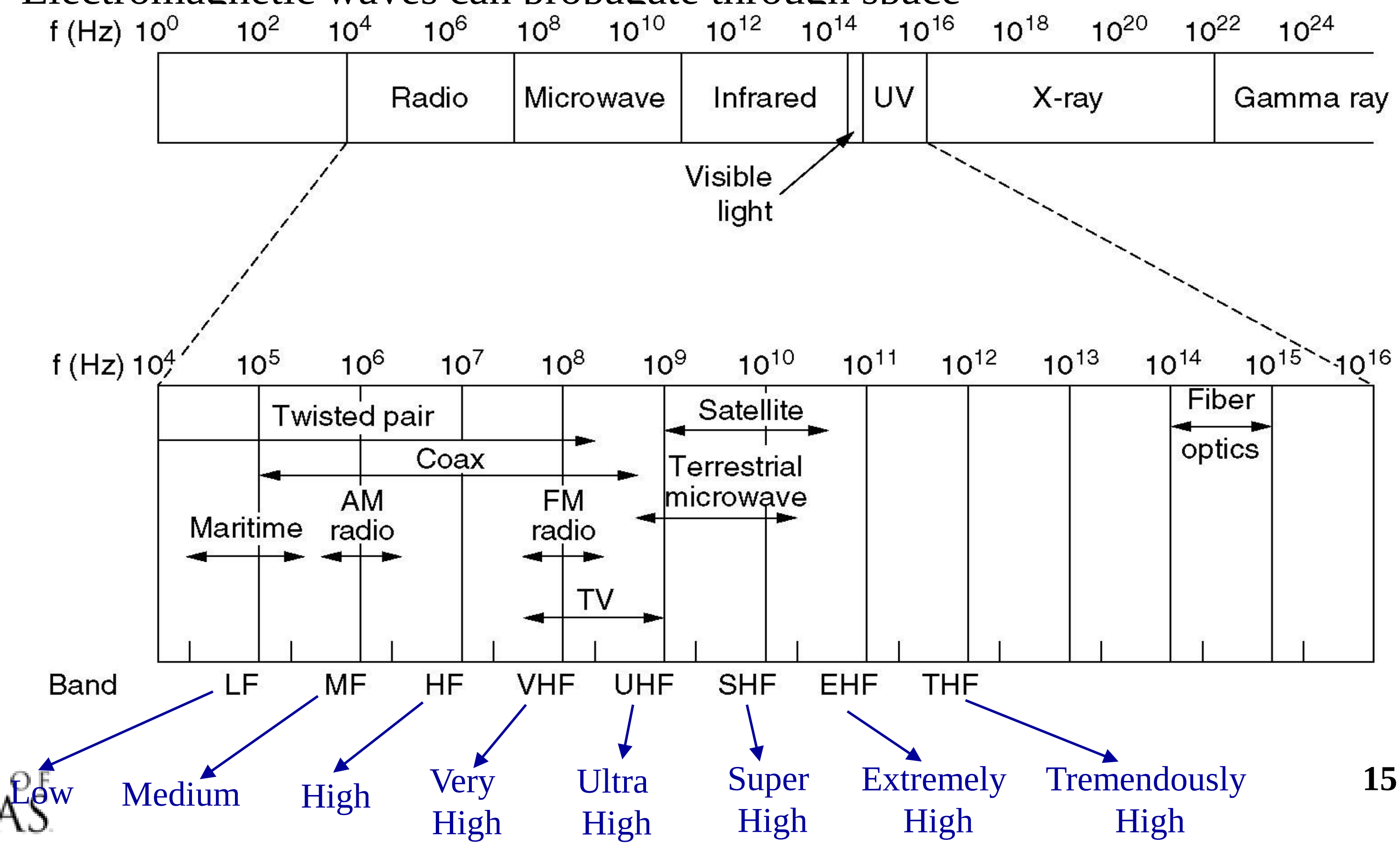

UNIVERSITY OF ARKANSAS

# PHYSICAL LAYER

- **Physical layer is responsible for**
  - Transmitting information over the physical medium
    - Make sure that when a ‘1’ is sent out at Tx, a ‘1’ is received at Rx.
  - Activation and deactivation of physical connections.
- **Physical layer specifies interface characteristics**
  - Mechanical interface.
    - E.g. number of pins of the connector, the shape of the connector, etc.
  - Electrical interface.
    - E.g. volts used to represent ‘1’, how long should a bit last, etc.
  - Procedural interface.
    - E.g. the sequence of events to activate/deactivate a physical medium.
- **Example physical layer standards**
  - RS-232, V.92, X.21

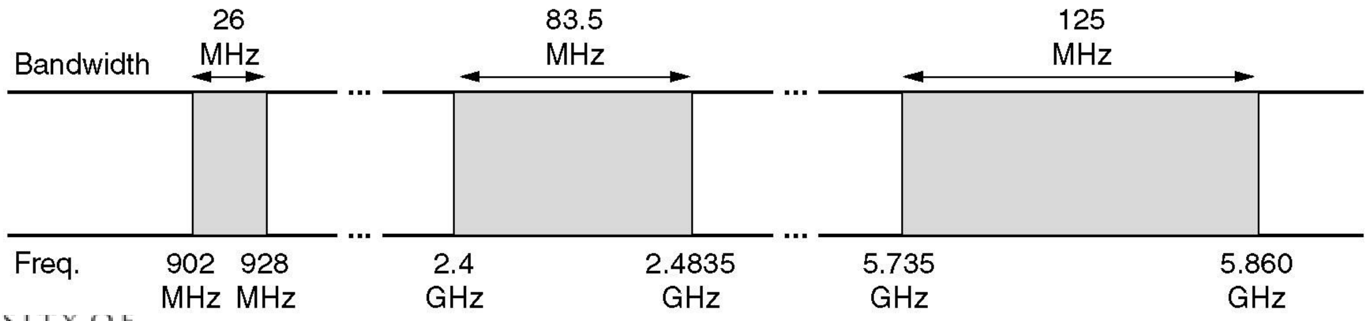
# PHYSICAL LAYER

- **Unguided transmission media**
  - Electromagnetic Waves
    - Change of electrical field causes change of magnetic field, and vice versa
    - Electromagnetic waves can propagate through space



# PHYSICAL LAYER

- **Spectrum Allocation**
  - Worldwide, ITU-R coordinates the allocation of spectrum
    - One device can be used in multiple countries
  - In U.S., FCC (Federal Communication Commission) is in charge of the spectrum allocation.
- **Spectrum Allocation Algorithms**
  - Beauty Contest: Each carrier explains why its proposal serves public best.
  - Lottery: Lottery is held among interested companies
  - Auctions: Certain bandwidth is given to the highest bidder.
- **ISM (Industrial, Scientific, Medical) bands**
  - Allocated by government for unlicensed use (device power must be under 1watt to avoid interference)
  - Bluetooth, WiFi, cordless phone, etc.

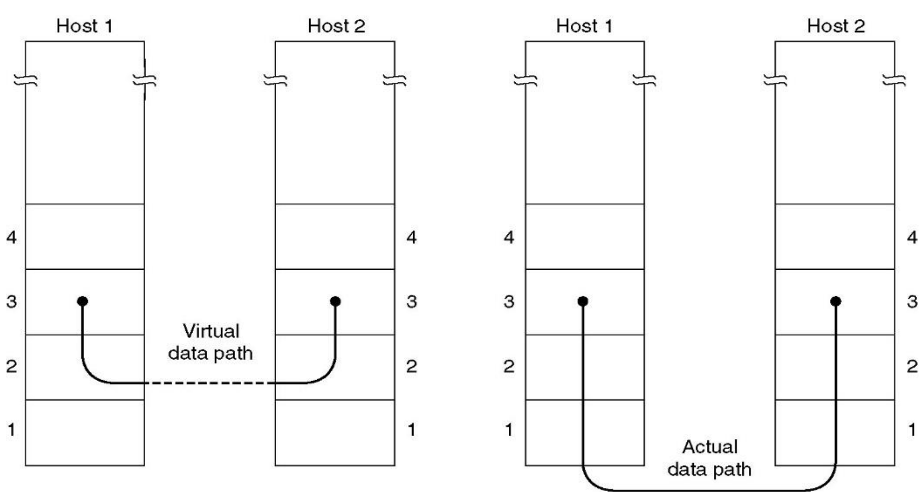


# LINK LAYER

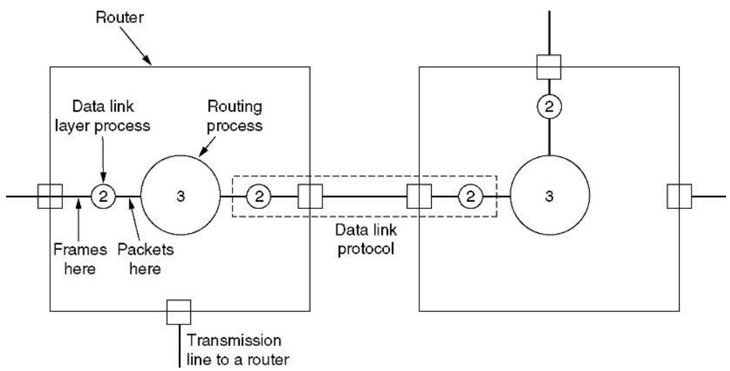
- **Link layer**
  - Deals with reliable communication between adjacent machines.
  - The channel acts as a wire ➔ no switching or routing
- **Functions of data link layer**
  - Providing well-defined service interface to the network layer
  - Framing
    - Encapsulates packets from network layer to frames.
  - Flow control
    - Keep fast transmitter from swamping slow receiver
  - Dealing with transmission errors
    - Error detection
    - Error correction
  - Media access control (MAC)
    - Determine how to allocate a single broadcast channel (multi access channel) among multiple competing users

# LINK LAYER

- **Services to upper layer**
  - The entity in data link layer is usually a software (might be embedded in a chip)
  - One router might have multiple different data link layers
    - E.g. connecting both wireless network and wired network

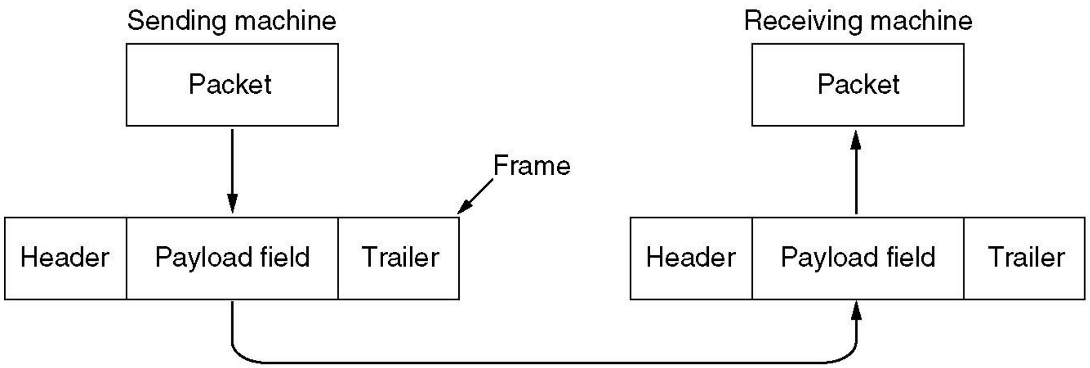



(a) Virtual communication (b) Actual communication



# LINK LAYER

- **Framing**
  - Encapsulate packet from network layer to frame
    - Breaks down larger packets
    - Add header and trailer




  - Break raw bit streams from physical layer into streams
    - Process each stream individually

University of Arkansas

# LINK LAYER

- **Flow Control**
  - Keep high speed Tx from swamping low speed Rx
  - Feedback-based flow control
    - Rx sends back information giving it permission to send more data.

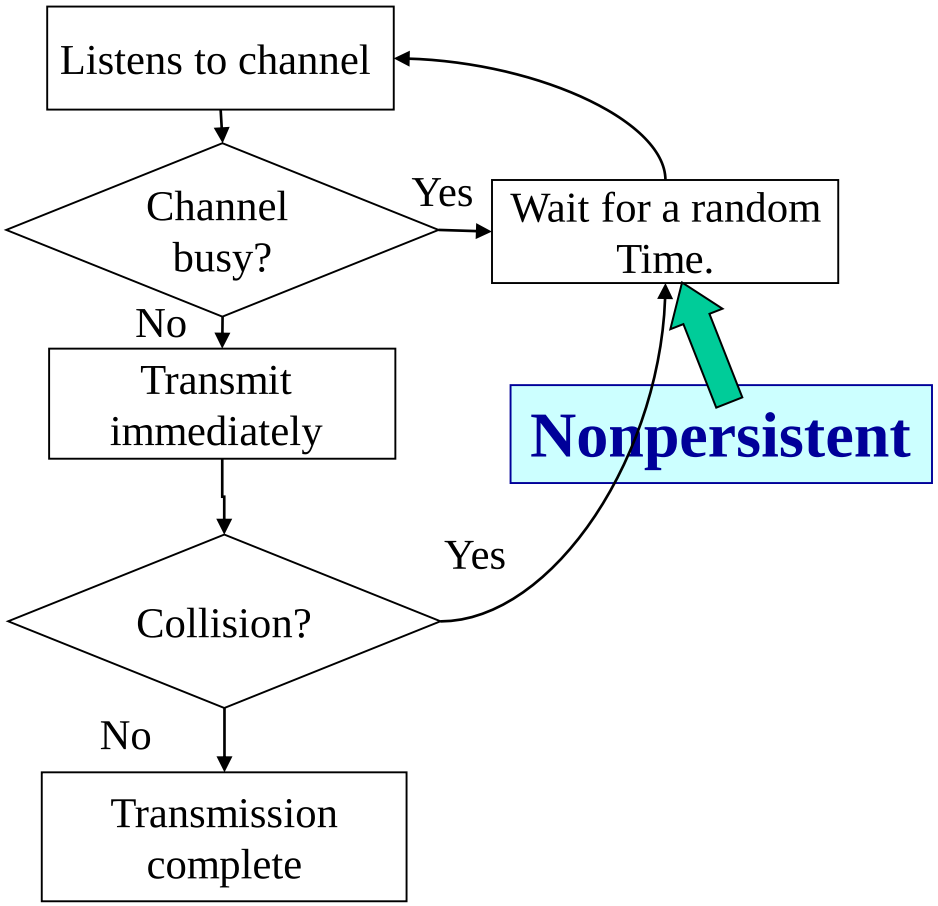
- **Error Control**
  - Acknowledgement (Ack)
    - Positive Ack: Tx successful. Negative Ack: Tx unsuccessful.
    - Timer used at Tx➔no Ack before timer expired = negative Ack.
      - If Ack is lost, Rx will have multiple copies of same frame
      - Sequence # is used to ensure no duplication and loss.
  - Error detection and correction code
    - Cyclic redundancy check (CRC)

# LINK LAYER

- **Media access control (MAC)**
  - Determine how to allocate a single broadcast channel (multiaccess channel) among multiple competing users
  - Collision: Two devices transmit simultaneously, they overlap in time and collision occurs
- **MAC classifications**
  - Static channel allocation
    - E.g. frequency division multiple access (FDMA): statically allocate a portion of the bandwidth to each user (broadcast TV)
    - E.g. time division multiple access (TDMA): divide time into slots, and allocate different slots to different users (cell phone systems)
  - dynamic channel allocation
    - Dynamically allocate channel to competing users.
    - Efficient for bursty traffics: data coming in irregularly.
      - Most widely used in LANs

# LINK LAYER

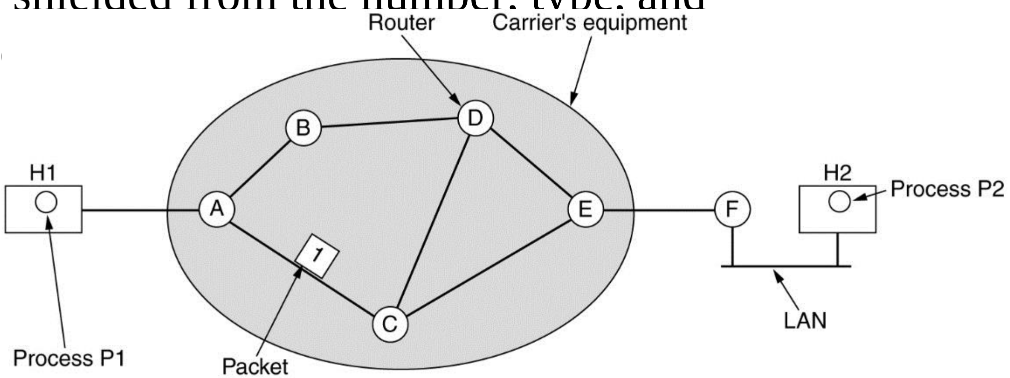
- **MAC example: CSMA**
  - Carrier sensing multiple access
  - Before sending out data, hosts listen to the carrier to see if there is transmission in the channel.
  - If there is transmission, host will back off.
  - If there is no transmission, host will act differently for different CSMA protocols



# NETWORK LAYER

- **Functions: getting packets from the source to the destination.**
  - Routing
    - choose appropriate path for a packet; knowledge of the topology.
  - Internetworking
    - connect different types of networks (cooper, fiber optic, wireless, etc.).
  - Quality of Service
  - Congestion Control
    - avoid overloading some of the lines.
  - Service to transport layer
    - Connectionless service
    - Connection oriented service (virtual circuit service)
    - Transport layer should be shielded from the number, type, and topology of the routers pr

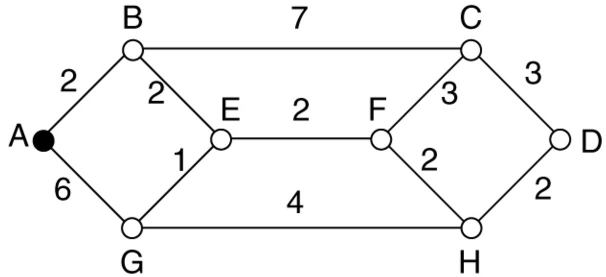
**Routers can be in either subnet (owned by operator) or LAN (owned by end users).**




UNIVERSITY OF ARKANSAS

# NETWORK LAYER

- **Routing**
  - Deciding which output line an incoming packet should be forwarded to.
  - Datagram routing (connectionless):
    - routing is performed for every packet.
    - A packet is called a datagram
  - Virtual circuit routing (connection oriented)
    - routing is only performed at connection setup ➔ session routing.
    - All packets of the same session following the same route
- **Shortest path: find out the route with the shortest "distance" between source and destination.**
  - Different measures can be used to measure "distance"
    - E.g. # of hops, physical distance (km), queuing delay …
  - In practical system, the "distance" measure is usually a function of the distance, bandwidth, average traffic, communication cost, mean queue length, delay, and other factors.

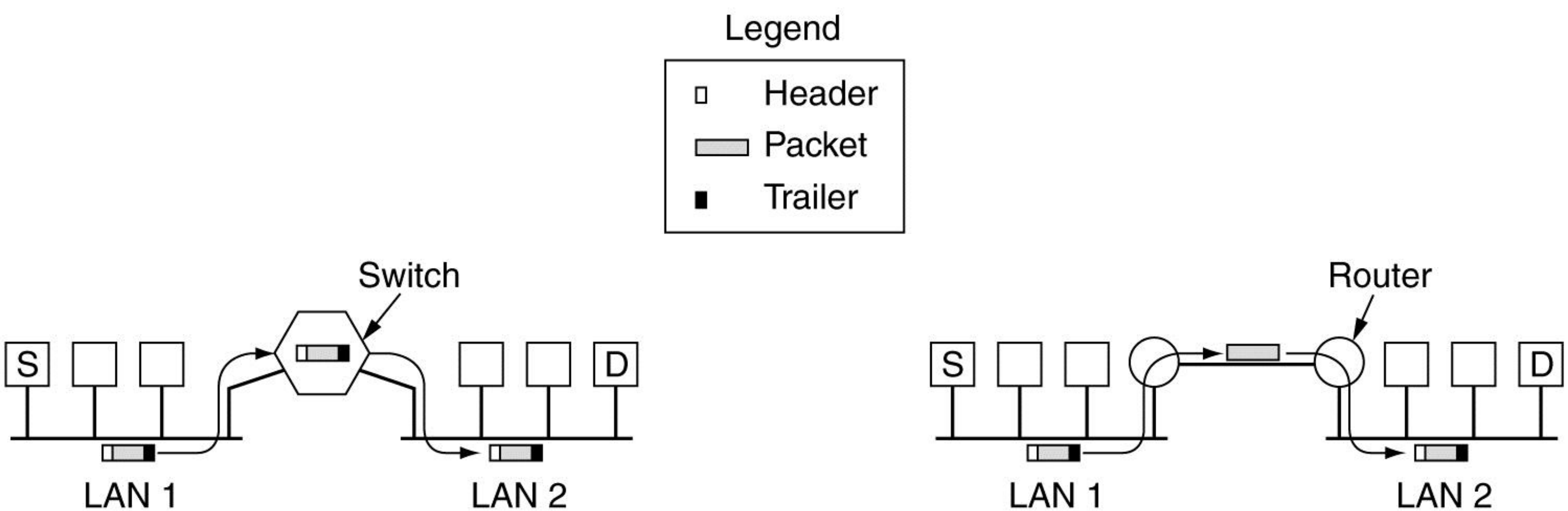



(a)

UNIVERSITY OF ARKANSAS

# INTERNETWORK: CONNECTION

- **Internetworking: how networks can be connected**
  - Physical layer: repeater, Hub
  - Data Link layer: switch, bridge
  - Network layer: router (gateway: multi-protocol router)
  - Transport layer: transport gateway
  - Application layer: application gateway

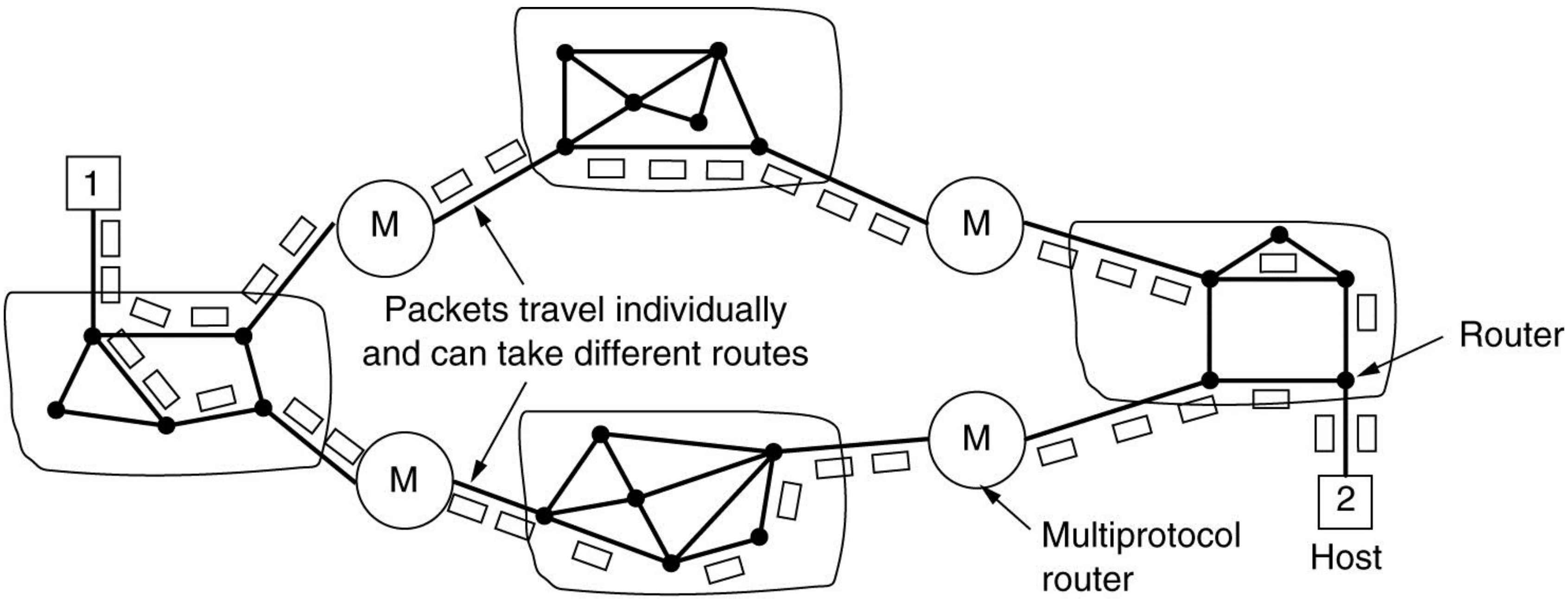



(a) Switch (the entire frame is transmitted)

(b) Router (only IP packet is transmitted)

# INTERNETWORK: CONNECTIONLESS INTERNETWORK

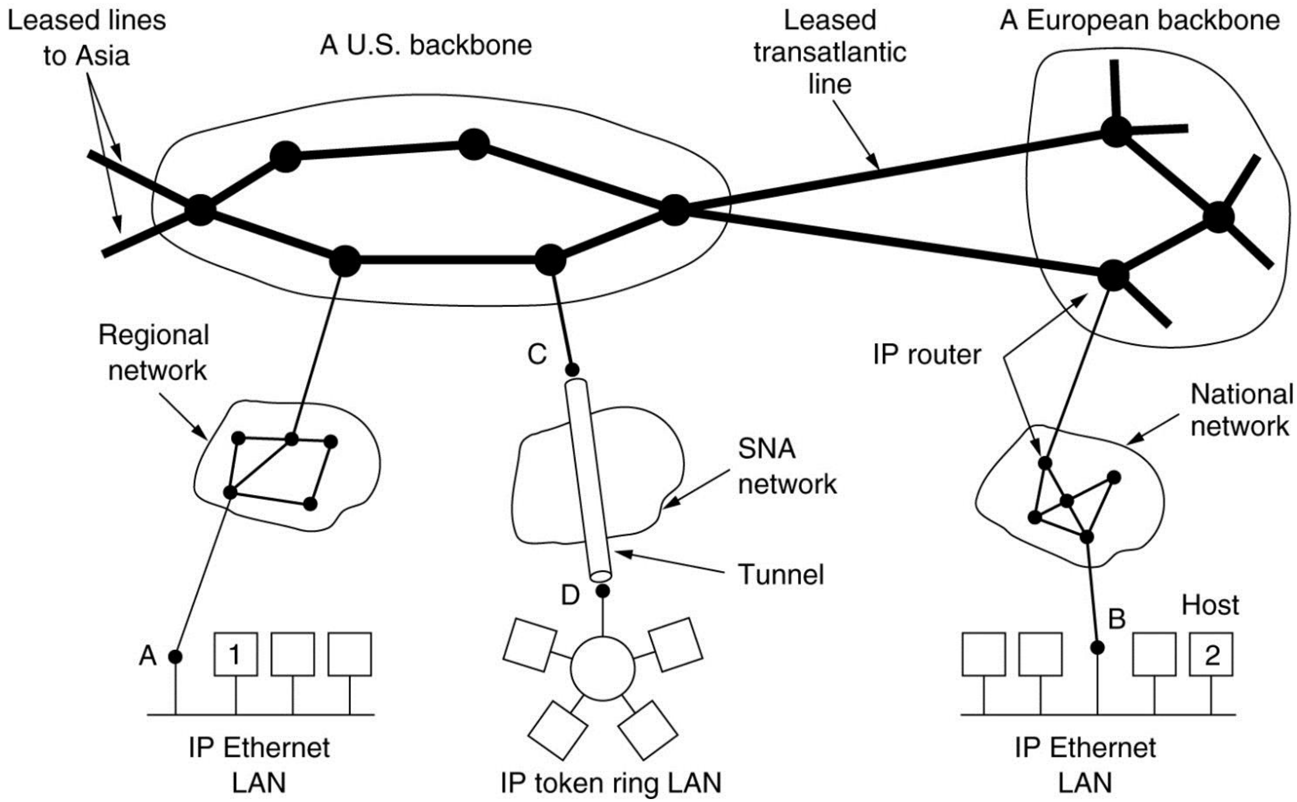
- **Connectionless Internetwork**
  - Packets might go through different paths.
  - Protocol translation.
  - Address mapping
  - IP packet is a “universal” packet recognized by most networks.




UNIVERSITY OF ARKANSAS

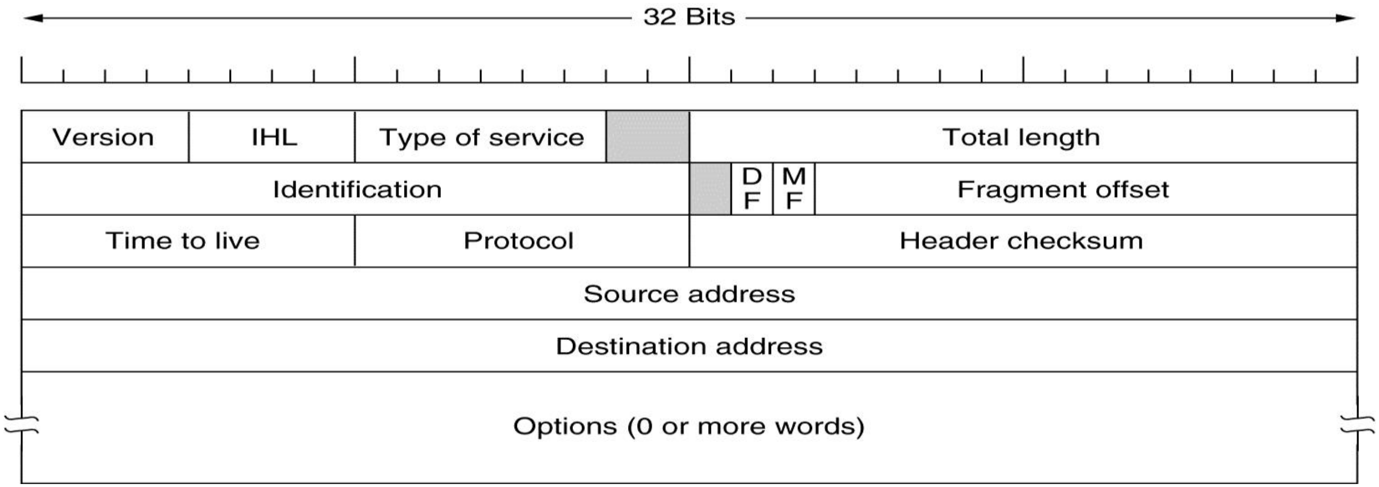
# NETWORK LAYER IN INTERNET

- **Internet Protocol (IP) is the glue for Internet**
  - IP was designed with internetworking in mind.
  - It provides best-efforts (not guaranteed) way to transmit datagram from source to destination.




UNIVERSITY OF ARKANSAS

# INTERNET: IP PROTOCOL




- Version: the version of the protocol (IPv4, IPv6)
- IHL (4 bits): header length (in the unit of 32-bit word, or 4-byte word).
  - Min value = 5 ➔ 5 x 4 = 20 bytes. Max value = 15 ➔ 15 x 4 = 60 bytes.
- Type of service (6 bits): used to distinguish different service classes (QoS).
  - Differentiated services: 4 queuing priorities, 3 discard probabilities, etc.
  - This field has been ignored by routers.
- Total length (16 bits): the total length of the IP packet (including header and data)
  - Maximum length: $2^{16}$ = 65536 bytes = 64KB

# INTERNET: IP PROTOCOL

- Identification (16 bits): identify which datagram a fragment belongs to.
  - Fragments belonging to the same datagram has the same ID field.
- DF: Don't fragment.
- MF: More fragment. More fragments are following.
  - All fragments except the last one have this bit set to 1.
- Fragment Offset (13 bits): the seq. # of the first elementary fragment in the current fragment. Elementary fragment: 8 bytes.
  - $2^{13}$ = 8192 elementary fragments. ➔ 8192 x 8 = 65536 bytes.
- Time to live (8 bits): decrease by 1 after each hop.
  - When hits 0, packet is discarded and a warning is sent back to source.
- Protocol: which protocol is on the transport layer (TCP, UDP, etc.)
- Header checksum: check header only. MUST be recomputed at each router! Why?
- Source address (32 bits): IP address of source.
- Destination address (32 bits): IP address of destination.

UNIVERSITY OF ARKANSAS

# INTERNET: IP PROTOCOL

- Option: variable length. Can contain many options
  - Each option starts with 1 byte word to identify the option
  - Some options

| Option | Description |
|---|---|
| Security | Specifies how secret the datagram is |
| Strict source routing | Gives the complete path to be followed |
| Loose source routing | Gives a list of routers not to be missed |
| Record route | Makes each router append its IP address |
| Timestamp | Makes each router append its address and timestamp |

  - More options can be found at www.iana.org/assignments/ip-parameters

# INTERNET: IP ADDRESS

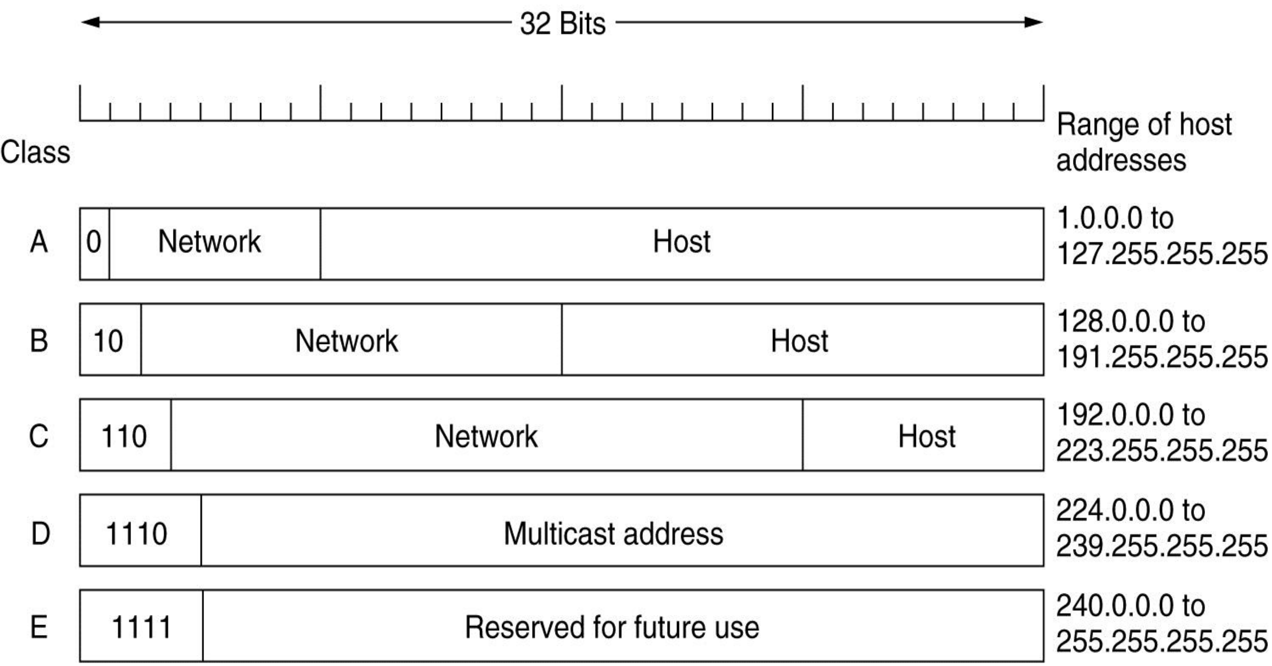
- **IP address: 32-bit**
  - $2^{32}$ = 4295 million addresses.
  - Format: dotted decimal.
    - Binary: 11000000.00101001.00000110.00010100
    - Hexadecimal: C0.29.06.14
      - Hexadecimal:Binary

| | | | |
|---|---|---|---|
| 0: 0000; | 1: 0001; | 2: 0010; | 3: 0011 |
| 4: 0100; | 5: 0101; | 6: 0110; | 7: 0111 |
| 8: 1000; | 9: 1001; | A: 1010; | B: 1011 |
| C: 1100; | D: 1101; | E: 1110; | F: 1111 |

    - Decimal: 192.41.6.20
      - 12x16+0 = 192; 2 x 16+9=41; 0x16+6=6; 1x16+4=20.
  - IP address assignment is managed by ICANN (Internet Corporation for Assigned Names and Numbers)

# INTERNET: IP ADDRESS

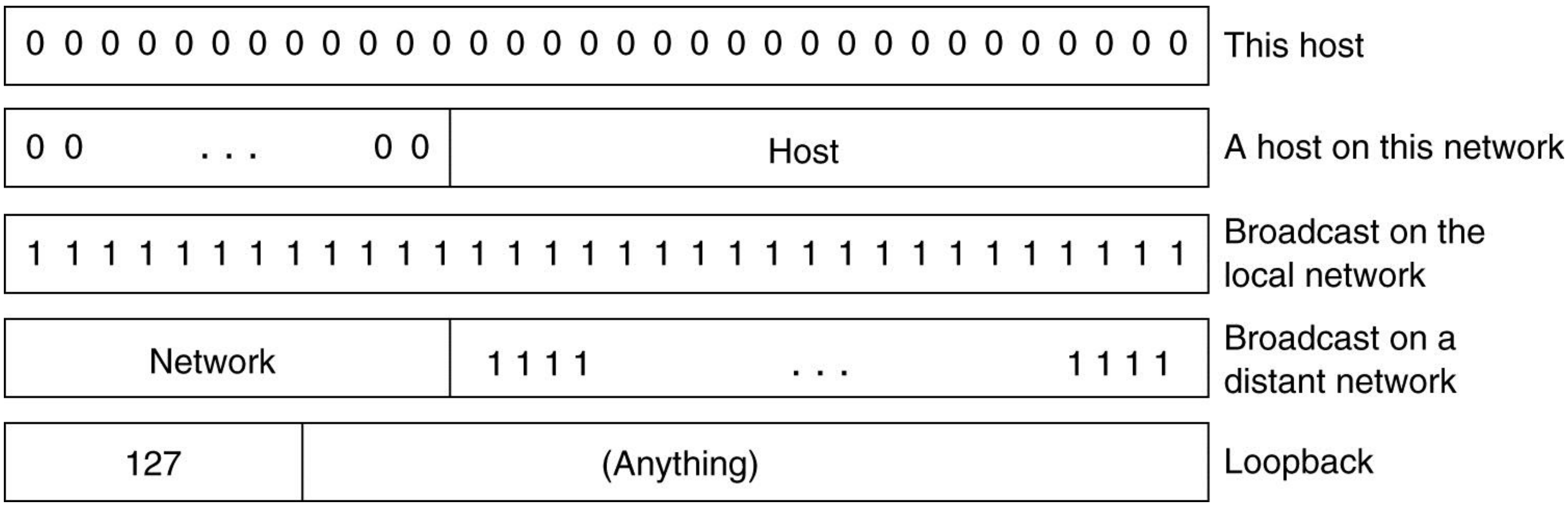
- **IP addresses are divided into 5 classes**
  - 0.0.0.0: this host
  - 255.255.255.255: broadcast




- 128 networks with 16 million hosts
- 16384 networks with 64K hosts.
- 2 million networks with 256 hosts.
- Multicast
- Reserved

# INTERNET: IP ADDRESS

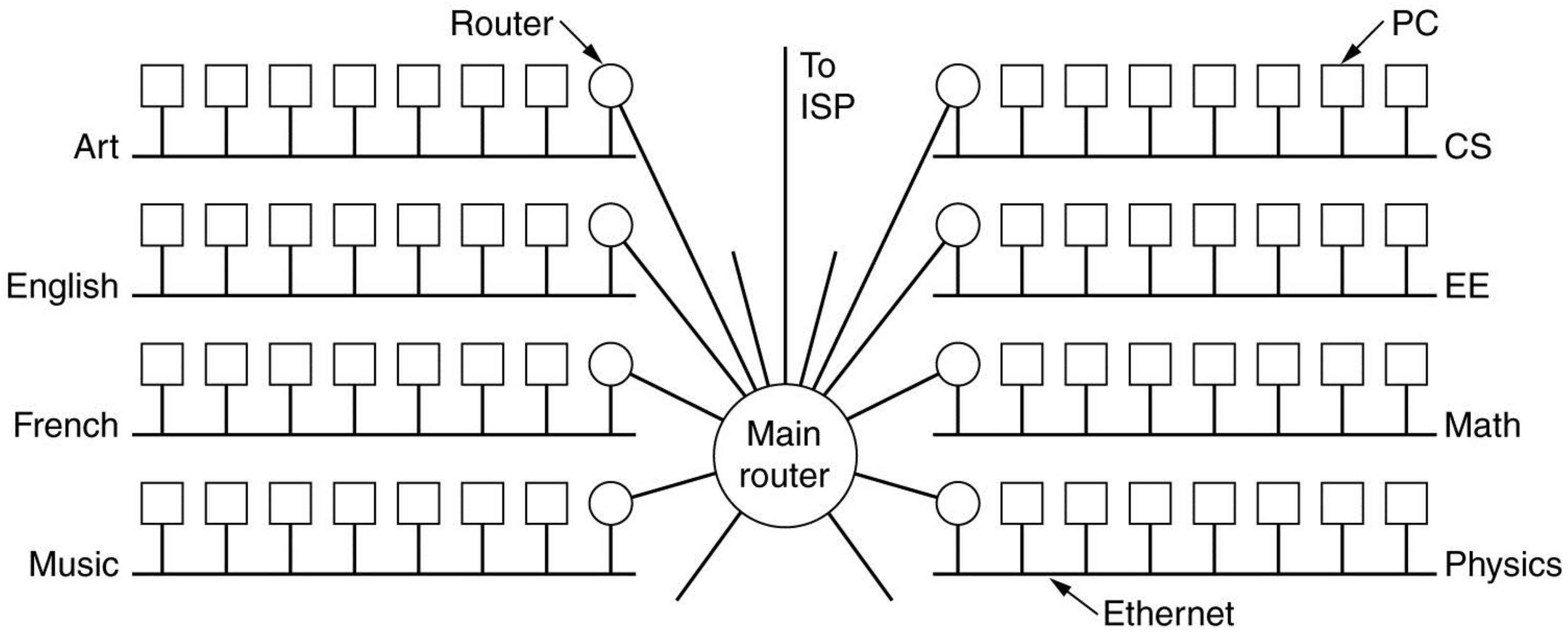
- **Special IP addresses**




– Loopback: for testing only.
  – Packets sent to this address will not be put in the wire.
  – They are going to be processed locally as incoming packets.

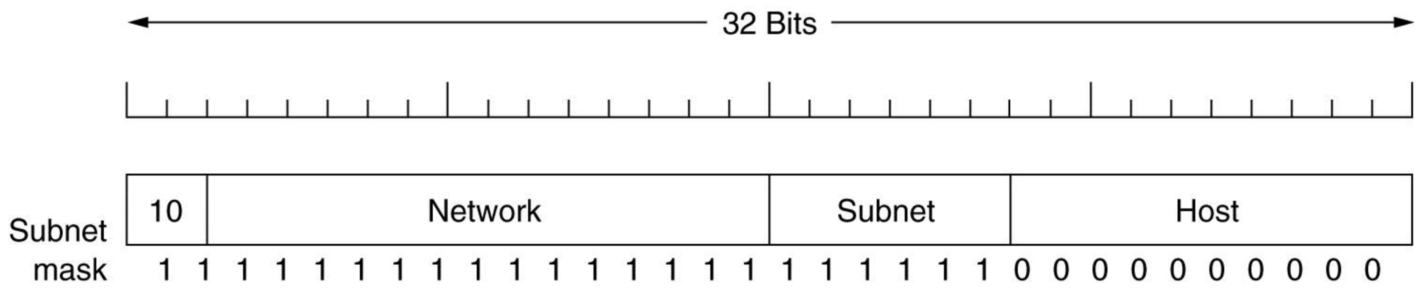
# INTERNET: IP ADDRESS

- **Subnet**
  - Allow a single network be split into small parts for internal use but still act like a single network to the outside.
    - E.g. Each LAN within the network is a subnet.



# INTERNET: IP ADDRESS

- **Subnet (Cont'd)**
  – Some bits of the host number are used to identify subnet.
    - E.g. in a class B network, 16-bits are used to identify hosts in the network. Among the 16-bit, 6 bits are used for subnet # (64 Ethernets), 10 bits are used for host # (1022 hosts in each subnet).
    - Subnet mask: 11111111.11111111.11111100.00000000 (255.255.252.0, or 130.157.23.16.0/22)
      – 22 bits are used for network # and subnet #; 10 bits are used for host #
  – Router in the network can route packet based on the subnet #.




**E.g: subnet 1: 10000010 00110010 000001|00 00000001**

**subnet 2: 10000010 00110010 000010|00 00000001**

**Each IP contains three fields: network address, subnet address, host address**

UNIVERSITY OF ARKANSAS

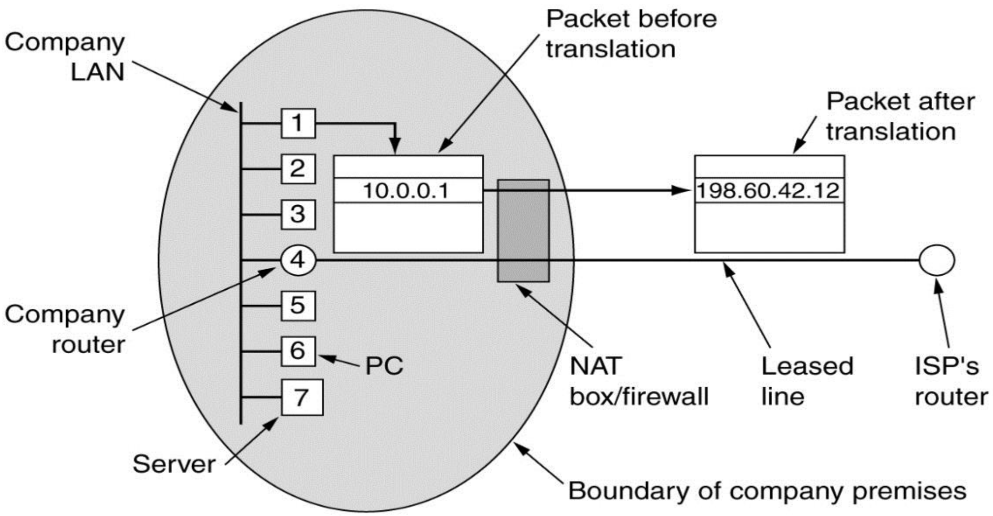
# INTERNET: IP ADDRESS

- **NAT (Network Address Translation)**
  - Motivation: Most users want static IP address ➔ We are running out of IP addresses!
  - Solution: Assign 1 or a few IP to company or ISP; Private IP are used inside company or ISP; NAT is used for address translation!
    - Private IP address ranges

      10.0.0.0 - 10.255.255.255/8 (16,777.216 hosts)

      172.16.0.0 - 172.31.255.255/12 (1,048,576 hosts)

      192.168.0.0 - 192.168.255.255/16 (65,536 hosts)
    - Packets leaving local network will be replaced with true IP address.

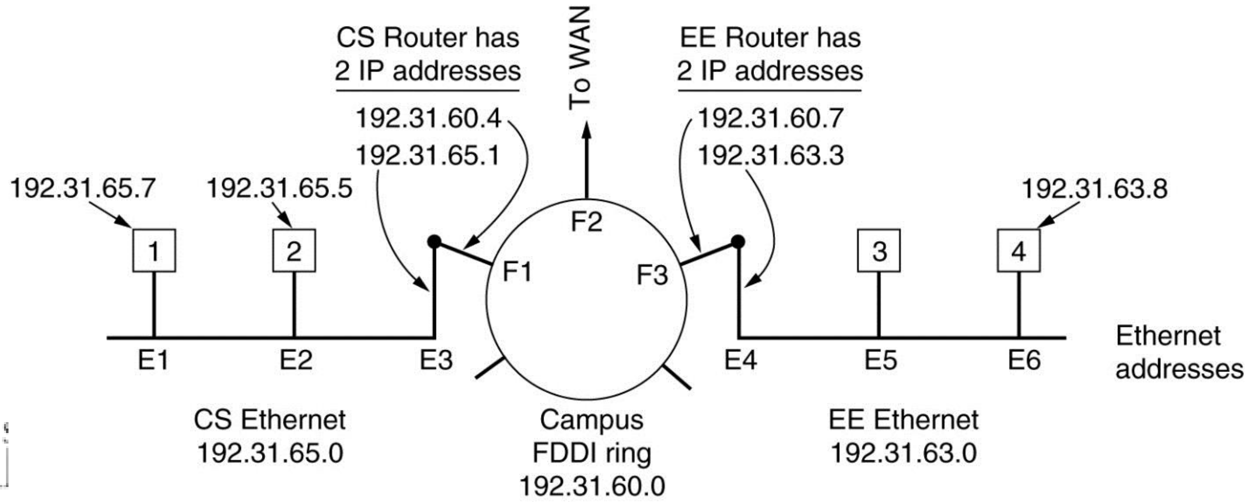


# INTERNET: IP ADDRESS

- **NAT (Cont'd)**
  - All incoming packets are addressed to the real IP of the network.
    - How to distinguish incoming packets for different hosts?
  - TCP (or UDP) has header fields for source port and dest. port.
    - Each port corresponding to one of the processes in host.
      - E.g. port 80: http; port 21: ftp;
  - NAT uses TCP source port to distinguish hosts!
    - Each host is assigned a unique TCP port # at NAT.
    - NAT maintains a table (host # and original port # v.s. NAT mapping port #)
    - When a packet is sent to NAT from the private network
      - its IP address is replaced by real IP address;
      - its TCP src port is changed to the NAT port based on ( host # and original TCP port #).

# INTERNET: INTERNET CONTROL PROTOCOLS

- **Address Resolution Protocol (ARP)**
  - How to find Ethernet address (MAC) by IP address?
  - E1 to E2: Host 1 broadcasts an ARP packet asking: what is the MAC address for 192.31.65.5?
    - Host 2 answers with its own MAC address.
    - All the hosts make an entry in their own (IP, MAC) mapping table for host 1 and 2.
  - E1 to E4: Host 1 sends the packet with dest. MAC address being F1 (or broadcast).
    - F1 builds a new MAC frame to F3
    - F3 builds a new MAC frame asking about the MAC address of E4



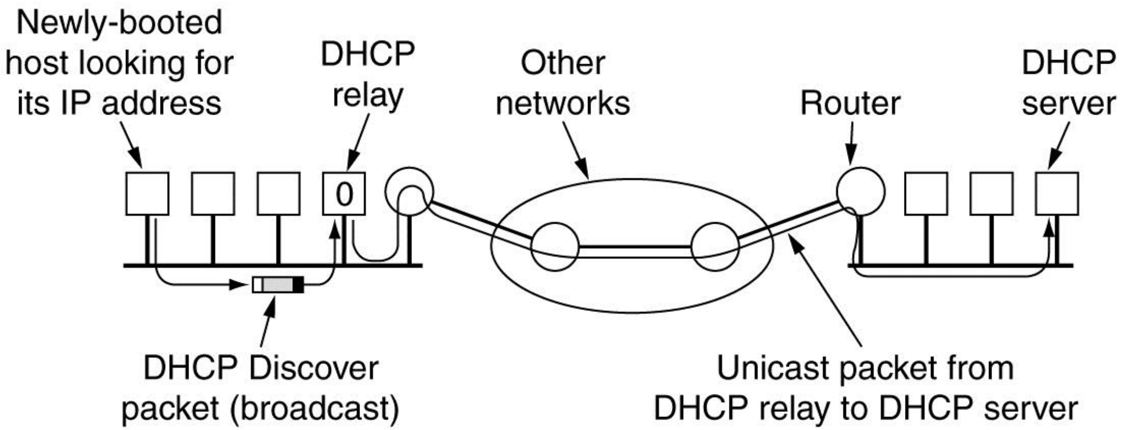
# INTERNET: INTERNET CONTROL PROTOCOLS

- **RARP (Reverse Address Resolution Protocol)**
  - A newly booted host broadcast its MAC address, and asking for IP address.
  - RARP server replies with the IP by looking the MAC in its table.
  - Router won't forward MAC broadcast message (dest: all '1's).
    - Each network needs an RARP server.
- **BOOTP**
  - A newly booted host broadcast its MAC address using UDP messages
    - UDP message can be forwarded by routers
  - BOOTP server will reply with IP and other information
    - E.g. the IP address of file server to download Operating System image.
  - The MAC/IP mapping table must be configured by hand.

UNIVERSITY OF ARKANSAS

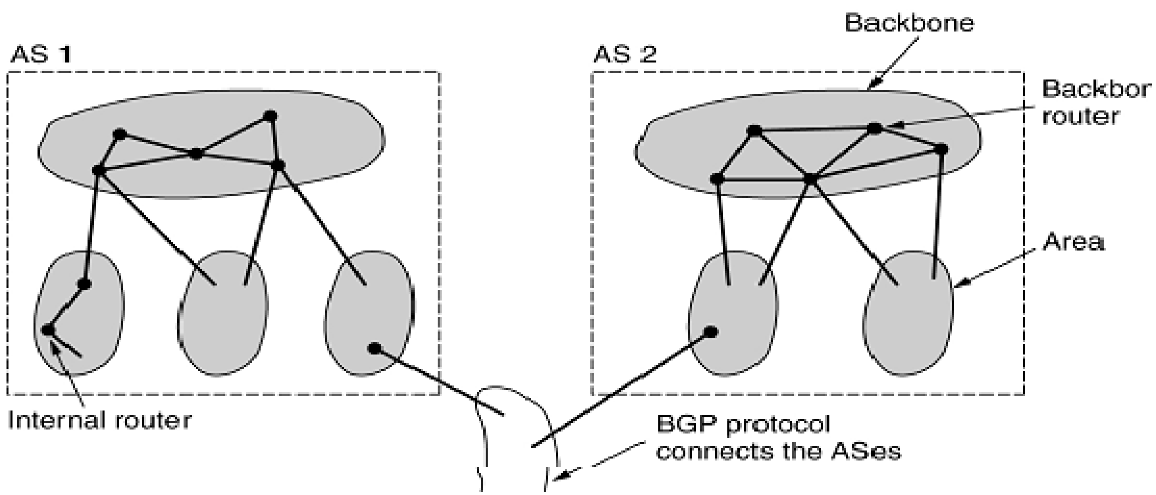
# INTERNET: INTERNET CONTROL PROTOCOLS

- **DHCP (Dynamic Host Configuration Protocol)**
  - A newly-booted host broadcasts a DHCP DISCOVER packet.
  - Each LAN has a DHCP relay agent
    - Forward the message to DHCP server through unicast.
    - The DHCP server might not be reachable through broadcasting.
  - DHCP server automatically assigns an IP address to the host.
  - Leasing: IP is assigned for that host for a period of time.
    - Host must renewal before expiration.



# INTERNET: OSPF

- **OSPF (Open Shortest Path First) – RFC 2328**
  - An interior gateway (multi-protocol router) routing protocol: routing inside an independent network (autonomous system, or AS).
  - AS is divided into areas: a network or a set of contiguous networks.
    - Each AS has a backbone area connected to all other areas.
    - Router connecting to two or more areas is part of backbone.
  - Four types of routers:
    - internal routers, area border router, backbone router, AS boundary router




UNIVERSITY OF ARKANSAS

# INTERNET: BGP

- **BGP (Border Gateway Protocol)**
  - An exterior gateway routing protocol: routing between ASes.
  - The major difference from interior gateway routing: need to consider politics
    - E.g. Do not use United States for traffic from British Columbia to Ontario; Traffic starting at IBM should not pass Microsoft; Transit traffic of only paid customers, etc.
  - The rules are manually configured in BGP routers.
  - BGP used an enhanced distance vector protocol
    - Exchanging information with neighbors about cost to all destinations.

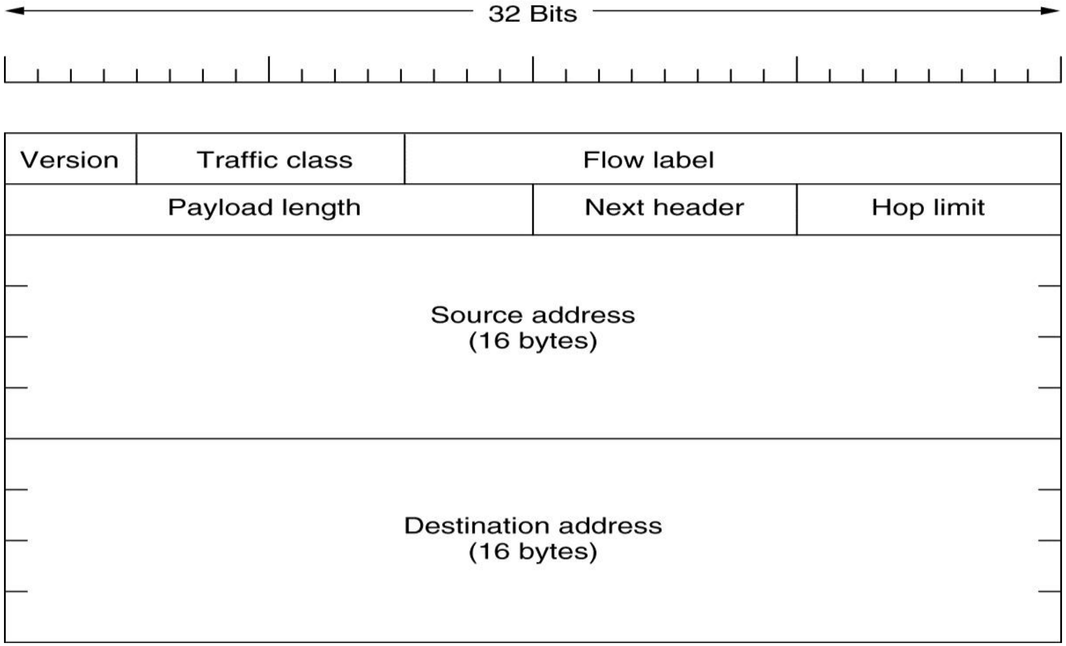
# INTERNET: IPV6

- **IPv6: the evolution of IP**
  - 16 bytes of addresses: support effectively unlimited addresses.

$$2^{16\times 8} = 3.4\times 10^{38}$$

    - For the entire earth (water & land), allow $7\times 10^{23}$ per square meter!
  - Header is simplifier compared with IPv4
    - Fixed length: 40 bytes
    - Packets can be processed faster in router ➔ lower delay, larger throughput.
  - Better support of options
    - Many required fields in IPv4 become options in IPv6
  - Better security
  - Better support of QoS.
  - Not compatible with IPv4, but compatible with most other protocols
    - TCP, UDP, ICMP, IGMP, OSPF, BGP, DNS

# INTERNET: IPV6 MAIN HEADER




- Version: 6
- Traffic class: different service classes for QoS
- Flow label: the label of a particular connection. Routers can use this field for QoS purpose. (e.g., this flow is delay sensitive).
  - Each flow is designated by source, destination, and flow #.

# INTERNET: IPV6 MAIN HEADER

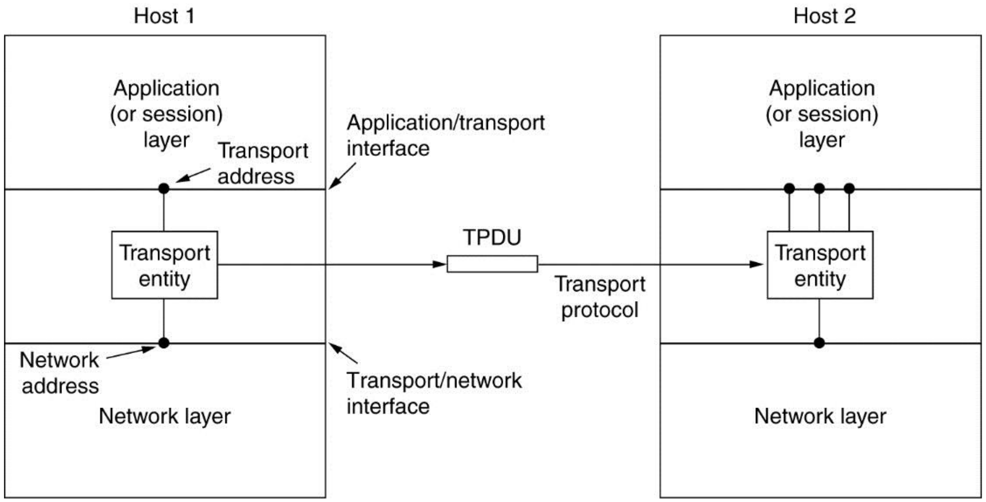
- **Main Header (Cont'd)**
  - Payload length: how many bytes following the 40-byte header
  - Next header: which of the 6 optional header is following the main header.
    - If no optional header, tells which transport protocol to pass the packet to.
  - Hop limit: same as Time to live in IPv4.
  - IPv6 address notation: eight groups of four hexadecimal digits

    8000:0000:0000:0000:0123:4567:89AB:CDEF
    - Can be simplified as

      8000::123:4567:89AB:CDEF
      - Consecutive zeros are replaced by ::
      - Leading 0s are omitted.

# TRANSPORT LAYER

- **Functions: provide reliable, cost-effective data transport from source to destination.**
  - Independent of the physical network
  - Independent of the network structure & topology
  - The first true end-to-end layer
    - Transport layer does not exist on routers.
    - Transport layer on the source is directly talking with transport layer on the destination.
    - On other layers, e.g. network layer, the conversation is usually between intermediate neighbors.




UNIVERSITY OF ARKANSAS

# TRANSPORT LAYER

- **Connection-oriented service**
  - Three phases:
    - Connection establishment
    - Data transfer
    - Connection release
  - Packets belonging to the same connection might take different routes!
    - If the underlying network layer is connectionless.
  - Example: TCP (Transport Control Protocol)

- **Connectionless service**
  - No connection required
  - E.g. UDP (User Datagram protocol)
  - Could be on top of connection-oriented network layer

# TRANSPORT LAYER: WHY TRANSPORT LAYER?

- **Network layer providing connectionless and connection oriented services as well.**
  - Why do we need another layer?
- **Network layer mainly resides on routers**
  - Routers are owned by operators
  - Users have no control about the operation of the network layer.
  - If something goes wrong in the network, the users could do nothing to stop it.
- **We need an addition layer on top of network layer to improve the QoS.**
  - If error happens in network, simply set up another connection.
  - Lost data can be detected and compensated by transport layer.
  - Provide a unified interface to users
    - Application programmers do not need to worry about the underlying network

# TRANSPORT LAYER: SERVICE PRIMITIVES

- **Service primitives**
  - A set of interfaces for the user to access the service
  - Usually in the form of function calls.
  - Analogy: printf( ) is the service primitive for the format printing service provided by C language.
    - Users do not need to know the implementation of the function.

- **Different transport layer protocols have different service primitives**

# TRANSPORT LAYER: SERVICE PRIMITIVES

- **A simple example**

| Primitive | Packet sent | Meaning |
|---|---|---|
| LISTEN | (none) | Block until some process tries to connect |
| CONNECT | CONNECTION REQ. | Actively attempt to establish a connection |
| SEND | DATA | Send information |
| RECEIVE | (none) | Block until a DATA packet arrives |
| DISCONNECT | DISCONNECTION REQ. | This side wants to release the connection |

- **The service primitives shield the underlying implementations from users**
  - Acknowledgement, lost packets, congestion, etc. are invisible to users.
  - Provide reliable service on top of unreliable networks
  - All users need to do is call SEND to send a packet at source, and call RECEIVE at destination to retrieve the packet.
- **Used directly by programmers or users**
  - Usually are convenient and easy to use.

# TRANSPORT LAYER: BERKELEY SOCKETS

- **Berleley sockets**
  - Socket: access point for the service in transport layer.
  - Servives primitives used in Berkeley Unix for TCP.
  - Part of the OS kernel.
  - The most widely used service primitives for TCP.

| Primitive | Meaning |
|---|---|
| SOCKET | Create a new communication end point |
| 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 | Send some data over the connection |
| RECEIVE | Receive some data from the connection |
| CLOSE | Release the connection |

In Linux or Unix system, use *man command_name* (*e.g. man socket*) to get more information.

# TRANSPORT LAYER: UDP

- **User datagram protocol (UDP) – RFC 768**
  - Connectionless protocol
  - Simple: IP with a short header
  - Fewer message required (no connection setup, acknowledgement).
  - Example: DNS (Domain Name System)
    - Client send a request contains a domain name
    - DNS server replies the IP of the domain name
    - Only two messages.
  - Good for short transaction, or delay sensitive applications.
- **Why UDP? Why not use IP directly?**
  - Each host has only one IP address
  - Multiple network applications are running simultaneously.
  - One UDP entity can provide services to multiple users.
    - Each user has it's own TCP port #.
  - We can easily identify the destination application by using
    - (IP, UDP port #)

**We want to use the addressing element of UDP.**

# UDP

32 Bits

| | |
|---|---|
| Source port | Destination port |
| UDP length | UDP checksum |

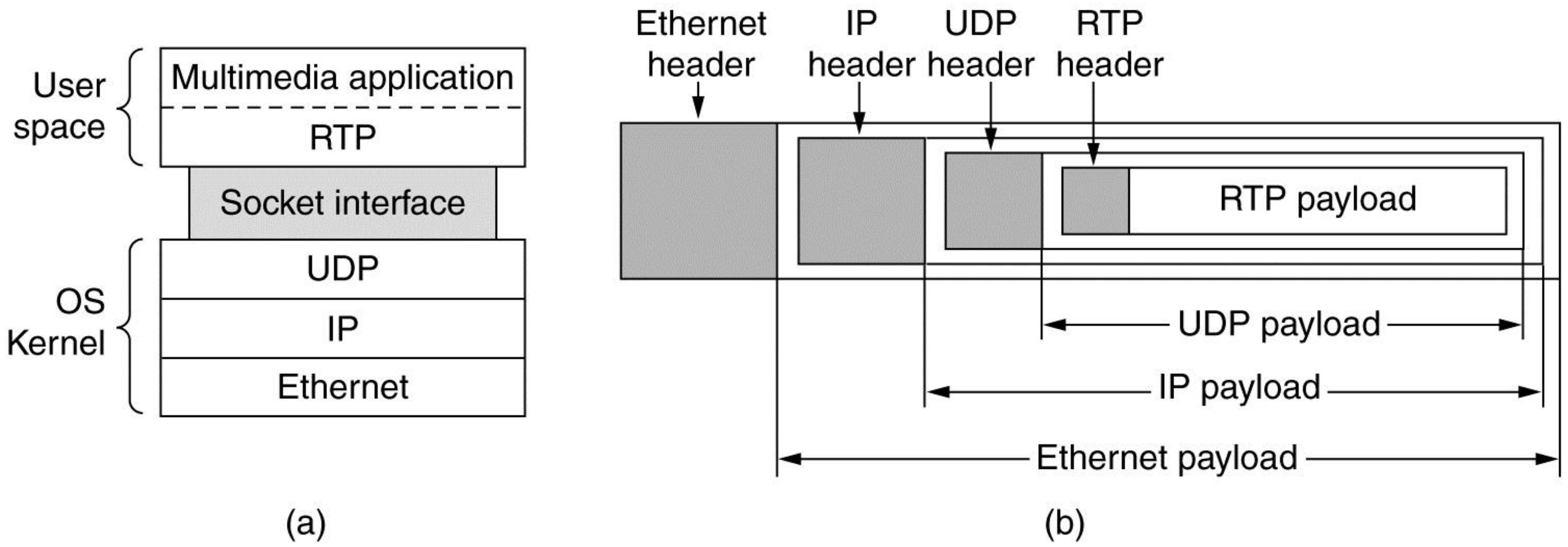
- **Header**
  - Source port: port # of source
  - Destination port: port # of destination
  - Length: the length of the entire UDP packet (head + payload).
  - Checksum: optional. All 0s if not computed.
- **What the UDP does not do**
  - Flow control
  - Acknowledgement
  - Retransmission (retransmission is meaningless for real-time application)

**These operations are left to the applications!**

UNIVERSITY OF ARKANSAS

# UDP: REAL-TIME TRANSPORT PROTOCOL

- **Real-time transport protocol (RTP)**
  - The engine for real-time applications
    - Internet telephony, videoconferencing, video-on-demand, etc.

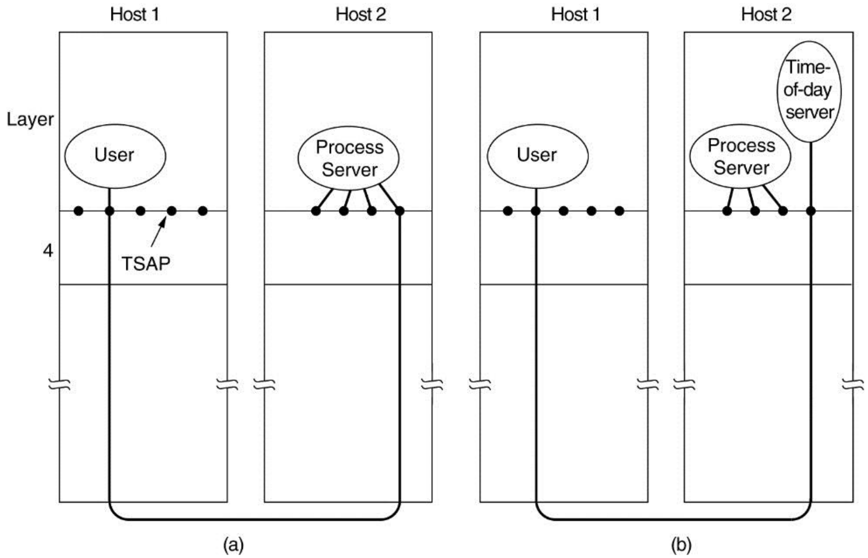


# TCP: SERVICES

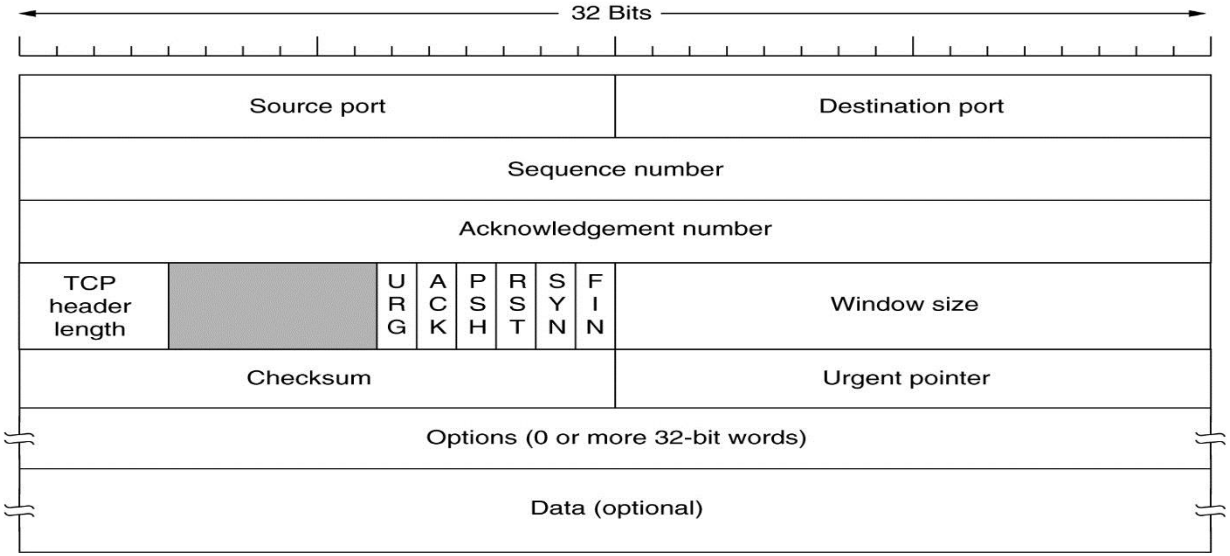
- **Transmission control protocol – RFC 793, 1122, 1323**
  - Reliable end-to-end byte stream over an unreliable internetwork
    - IP gives no guarantee of the delivery of datagram
    - It's up to TCP to guarantee the in-order and reliable delivery of the data to application layer.
- **TCP service model**
  - Application access the services provided by TCP by creating sockets at both communication parties.
    - Each socket is associated with a unique port number.
  - Connections are established between sockets.
    - One socket might be used for multiple connections.
      - E.g. several users can connect to the same FTP server simultaneously.
  - Byte stream: all bytes are treated equally.
  - Full duplex: data transmissions can occur at both directions simultaneously.

# TCP: SERVICES

- **Daemon**
  - Server programs runs at background
    - FTP daemon is associated with port 21
  - Internet daemon (inetd)
    - Attached to multiple ports and waiting for connection requests.
    - When a connection request for a particular application comes in, inetd will fork of a new process and wake up the corresponding service daemon.

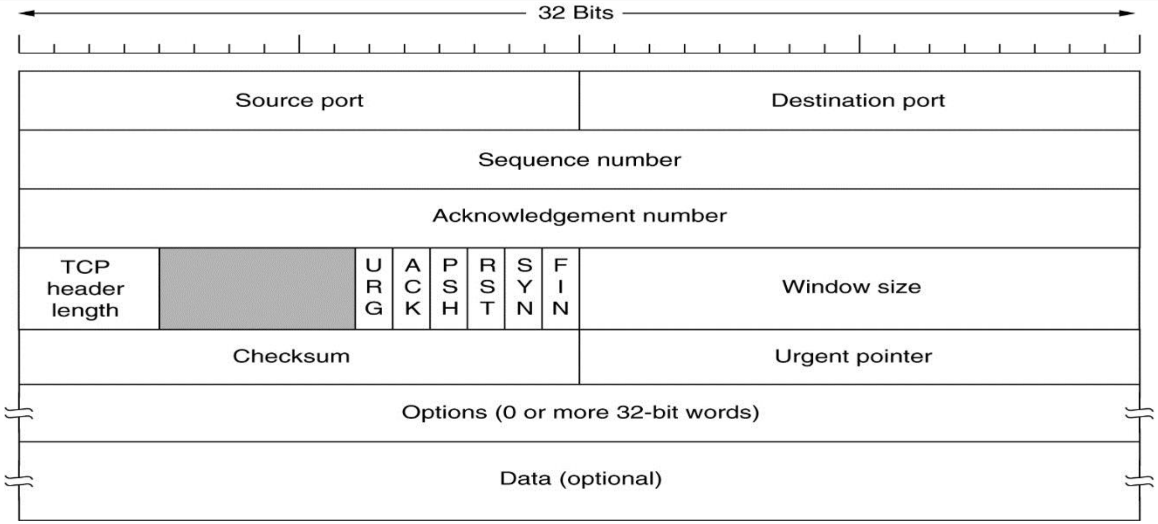


# TCP: PROTOCOL



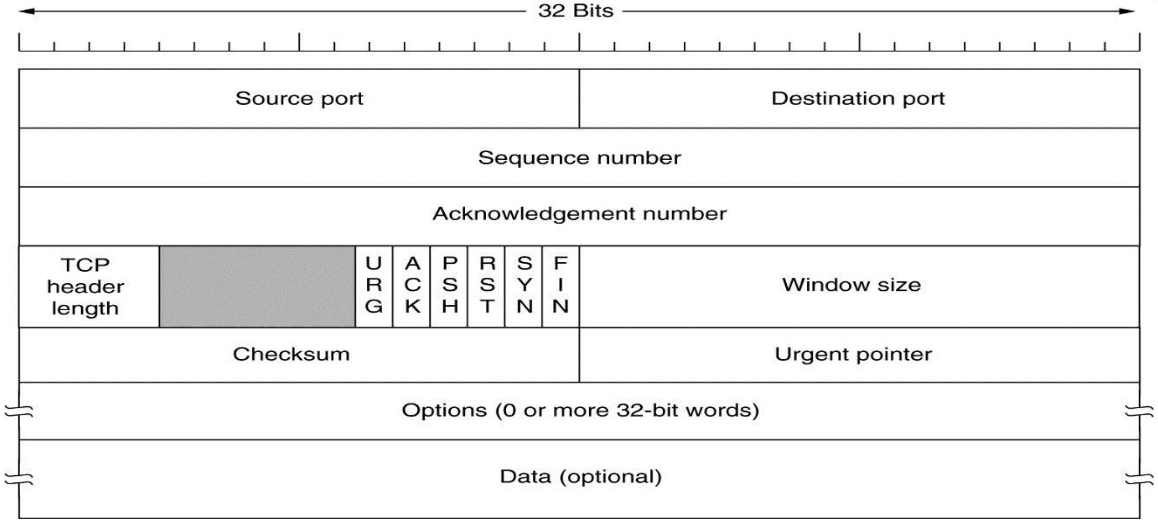

- **Source port, destination port**
  - **16-bit:**
  - **Specify the sockets on source and destination.**
- **Sequence number:**
  - **For a particular connection, each byte has its own sequence number**
  - **The sequence number of the first byte in the payload**
- **Acknowledge number: (Used in combination with the ACK flag)**
  - **All bytes (0, Ack# - 1) have been successfully received**
  - **The next expected byte is Ack#.**

# TCP: PROTOCOL



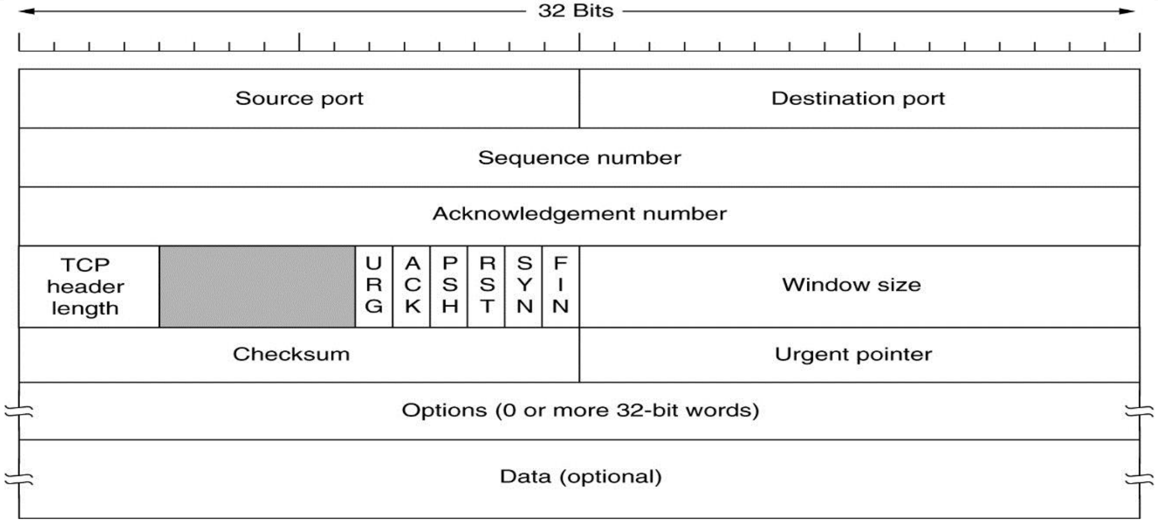

- **TCP header length (in the unit of 32-bit word)**
  - **How many 32-bit words are in the TCP header.**
  - **Fixed header: 20 bytes. Optional header: variable length.**
- **URG: urgent flag, used in combination with urgent flag**
  - **If set to 1, indicating the urgent pointer is in use**
- **Urgent pointer: (relative offset from the seq#)**
  - **pointing to a byte in the payload starting from which the data is urgent**
  - **TCP will be notified by application which data is urgent**
    - **E.g. telnet, ctrl+C to terminate an application.**
  - **If the data is urgent, TCP cannot hold it in buffer. Must deliver it immediately even if the sender window is 0.**

# TCP: PROTOCOL




- Ack: when set to 1, the Ack# is valid. Otherwise Ack# will be ignored.
- PSH: PUSH flag
  - When set to 1, notify TCP entity to send out data immediately.
  - E.g. in remote login, hit enter.
- RST:
  - when set to 1, reset a connection (due to crash or other reasons).
  - Also used to reject invalid TCP packets (e.g. delayed duplicate).
- SYN:  used for connection establishment
  - Connection request: SYN = 1, ACK = 0.
  - Connection accept: SYN = 1, ACK = 1.

# TCP: PROTOCOL




- **FIN: set to 1 used for connection release.**
  - **After sending out a packet with FIN = 1, the host will no longer transmit more data. (but will continue receive data)**
- **Window size: used for flow control**

# TCP: PROTOCOL

- Checksum
  - checks TCP header, data, and part of the IP header (source IP, dest. IP, protocol (6), TCP packet length) ➔ violate the layered structure
  - Other than TCP header, the following fields are checked.

| | | |
|---|---|---|
| Source address | | |
| Destination address | | |
| 0 0 0 0 0 0 0 0 | Protocol = 6 | TCP segment length |

# TCP: CONNECTION ESTABLISHMENT

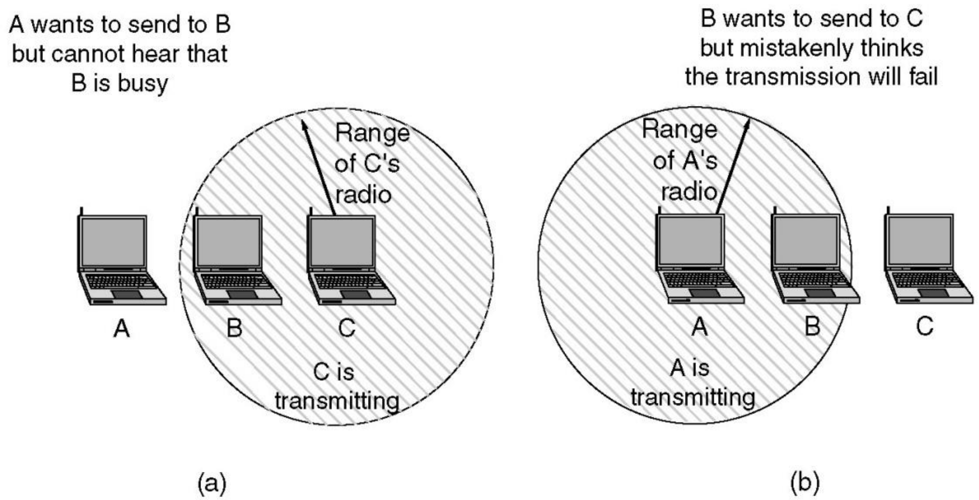
- **Service primitives**
  - Server: LISTEN, ACCEPT
  - Client: CONNECT
- **Three-way handshake**
  - After CONNECT is called, a connection request packet will be generated.
    - Connection request packet (SYN = 1, ACK =0, Seq# = x).
    - When the connection request packet arrives at the receiver, the receiver checks if LISTEN and ACCEPT have been executed for the port.
    - If server accepts request, send back Ack (SYN = 1, ACK=1, Ack# = x+1, Seq# = y)
      - Client sends back an Ack (SYN=0, ACK=1, Ack#=y+1) to finish the three way handshake.
    - If server rejects request, send back Rej (RST = 1, ACK = 1, Ack#=x+1)
    - The initial sequence# (x, y) are assigned based on the clock at the hosts.

# APPLICATION LAYER

- The applications directly used by the end users
- Examples
  - HTTP (Hyper Text Transfer Protocol)
  - FTP (File Transfer Protocol)
  - E-mail

# PROTOCOL: MAJOR CHALLENGES

- **Major challenges**
  - Network layer: dynamic routing
    - Nodes are mobile
    - Distributed multihop wireless network with time-varying topology.
  - MAC sublayer:
    - Hidden station problem
    - Exposed station problem
  - Physical layer
    - Power consumption.
    - Unreliable wireless link



# OUTLINE

- Ad hoc wireless networks
- Protocol layers
- **Cross-layer design**

# CROSS-LAYER DESIGN

- **Cross-layer design**
  - Perform joint optimization across protocol layers
  - Why?
    - Layered structure works well for conventional wired networks
    - The isolation of layers doesn't work very well for wireless ad hoc network
      - Can't perform optimization across layers
      - The physical channel and the network structure is constantly changing → the layered structure might not be able to respond to the change fast enough

# CROSS-LAYER DESIGN

- **Example 1**
  - Physical layer: adaptive modulation and coding (AMC)
    - Select the modulation and coding based on the channel condition
  - Network layer:
    - Adaptive routing: there might be multiple paths between source and destination. Select the path with the better physical layer performance.
- **Example 2**
  - Hybrid Automatic Retransmit Request (HARQ)
    - In the MAC layer, if there is collision during transmission, re-transmit
    - Perform maximal ratio combining (MRC) between the originally transmitted signal and the re-tx signal at the physical layer.
- **Example 3**
  - Cooperative diversity: multiple spatially distributed nodes to help the forward of a node
  - Network diversity: multiple routes through the network are used to send a single packet