Advanced Communication Networks

Muhammad Taha Jilani

Lecture - 12

Transport layer Protocols

Most Common Transport Protocols

- **Transmission Control Protocol (TCP)**
  - TCP is as complex as UDP is simple, but with same concept as both are end-to-end protocols.
  - But the major difference between UDP and TCP is that TCP is connection oriented.
  - TCP provides a one-to-one, connection-oriented, reliable communications service. TCP handles the establishment of a TCP connection, the sequencing and acknowledgment of packets sent, and the recovery of packets lost during transmission.

Transport layer Protocols

Most Common Transport Protocols

- **Transmission Control Protocol (TCP)**
  - Major Internet applications such as the World Wide Web, email, remote administration, and file transfer rely on TCP.
  - TCP protocol operations may be divided into three phases. Connections must be properly established in a multi-step handshake process (*connection establishment*) before entering the *data transfer* phase. After data transmission is completed, the *connection termination* closes established virtual circuits and releases all allocated resources

Transport layer Protocols

Most Common Transport Protocols

- **TCP Implementation**
  - Connections are established between client to server and back, using a *three-way handshake,* within these connection
    - Data is divided up into packets by the operating system
    - Packets are numbered, and received packets are acknowledged
    - Connections are explicitly closed
      - (or may abnormally terminate)

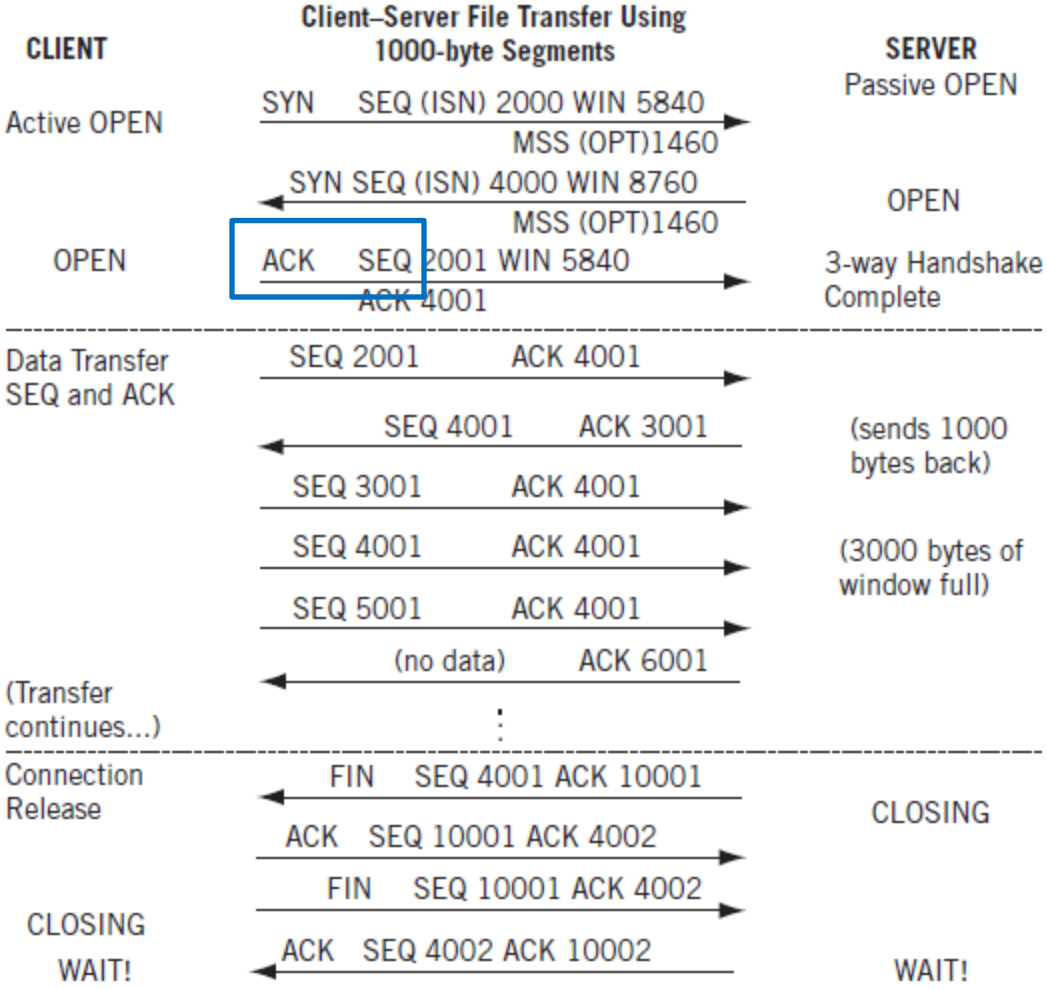
Transport layer Protocols

- **Client–server interaction with TCP three-way handshake**
  - TCP uses unique terminology for the connection process, a single bit called the SYN (synchronization) bit is used to indicate a connection request.
    - This single bit is embedded in a complete 20-byte (usually) TCP header,.
  - Connections and data segments are acknowledged with the ACK bit,
  - A request to terminate a connection is made with the FIN (final) bit.

Transport layer Protocols

- **Client–server interaction with TCP three-way handshake**

How TCP is reliable?



Transport layer Protocols

- **How TCP is reliable?**
  - **Reliability**
    - Error detection and correction
    - Each segment has a sequence number used to put the message back together at the destination.
    - Flow control prevents a sender from overflowing a receiver with more data than it can handle.

Transport layer Protocols

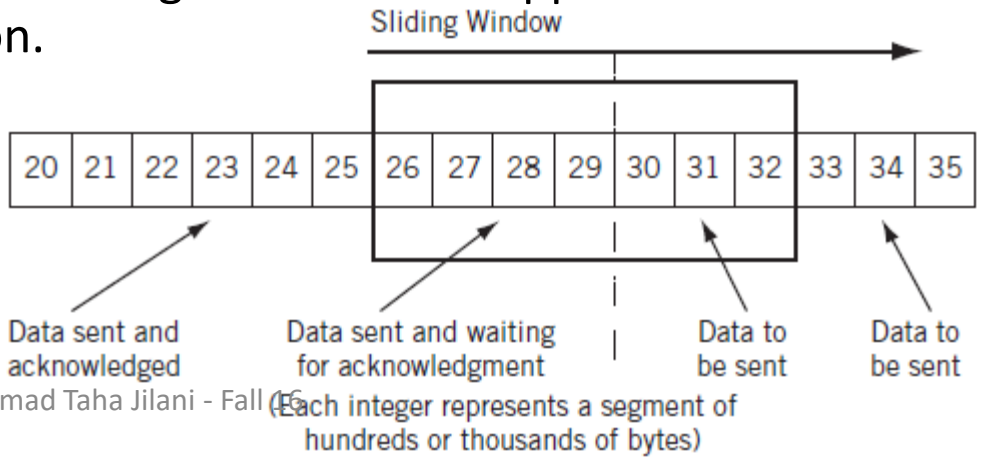
- **TCP Implementation**
  - **Reliability**
    - **Error control** — error detection *and* error correction
    - It provides end-to-end error control
      - Error control includes mechanisms for detecting corrupted segments, lost segments, out-of-order segments, and duplicated segments. Error control also includes a mechanism for correcting errors after they are detected. Error detection and correction in TCP is achieved through the use of three simple tools: **checksum**, **acknowledgment**, and **time-out**.
      - Checksum: 16-bit fro every TCP segment
      - Acknowledgement: Control segments that carry no data but consume a sequence number that are also acknowledged. ACK segments are never acknowledged
      - Time-out: Segment retransmission: when a timer expires (based on round-trip time (RTT) of segments) or when the sender receives three duplicate ACKs.
    - **Flow control** — Transport layer can include flow control mechanisms to prevent senders from overwhelming receivers.
    - Datagrams do not require these services.

Transport layer Protocols

- **TCP Implementation**
  - **Flow Control**
    - Flow control can be implemented at any protocol level or even every protocol layer.
    - In practice, flow control is most often a function of the transport layer (end to end).
    - TCP is a “byte-sequencing protocol” in which every byte is numbered.
    - Although each segment must be acknowledged, one acknowledgment can apply to multiple segments. Senders can keep sending until the data in all unacknowledged segments equals the window size of the receiver.

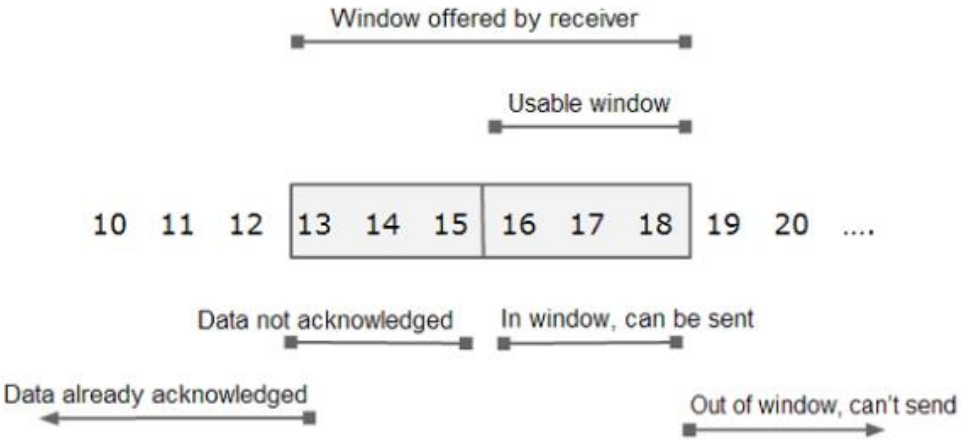
Transport layer Protocols

- **TCP Implementation**
  - **Flow Control**
    - A conceptual “window” overlays the set of data, and two moveable boundaries are established in this series of segments to form three types of data.
      - segments that have been sent and acknowledged,
      - segments sent and waiting for an acknowledgment,
      - segments waiting to be transmitted.
    - The flow control mechanism in TCP is a sliding window procedure that prevents senders from overwhelming receivers and applies in both directions of a TCP connection.



Transport layer Protocols

- **TCP Implementation**
  - **Flow Control**



The available window advertised by the receiver is 6. This means that receiver can accept 6 bytes as of now.
The window at sender side covers bytes ranging from 13 to 18 (I.e. 6 bytes in total).
Out of this range, 13-15 are the bytes which have been sent but no acknowledgement is yet received for them.
Bytes 16-18 are the bytes that sender can send as soon as possible.
If sender starts receiving acknowledgement for bytes 13 to 15, the left end of the window starts closing in.
The right end starts opening up as more and more window size is advertised by the receiver.
This window slides towards right depending upon how fast receiver consumes data and sends acknowledgement and hence known as sliding window.

Transport layer Protocols

- **TCP Implementation**
  - **Congestion Control**
  - TCP is a virtual circuit service that adds reliability to the IP layer, reliability that is lacking in UDP.
    - By providing sequencing and flow control to the host-to-host interaction, which in turn provides a congestion control mechanism to the routing network as a whole (as long as TCP, normally an end-to-end concern, is aware of the congested condition).

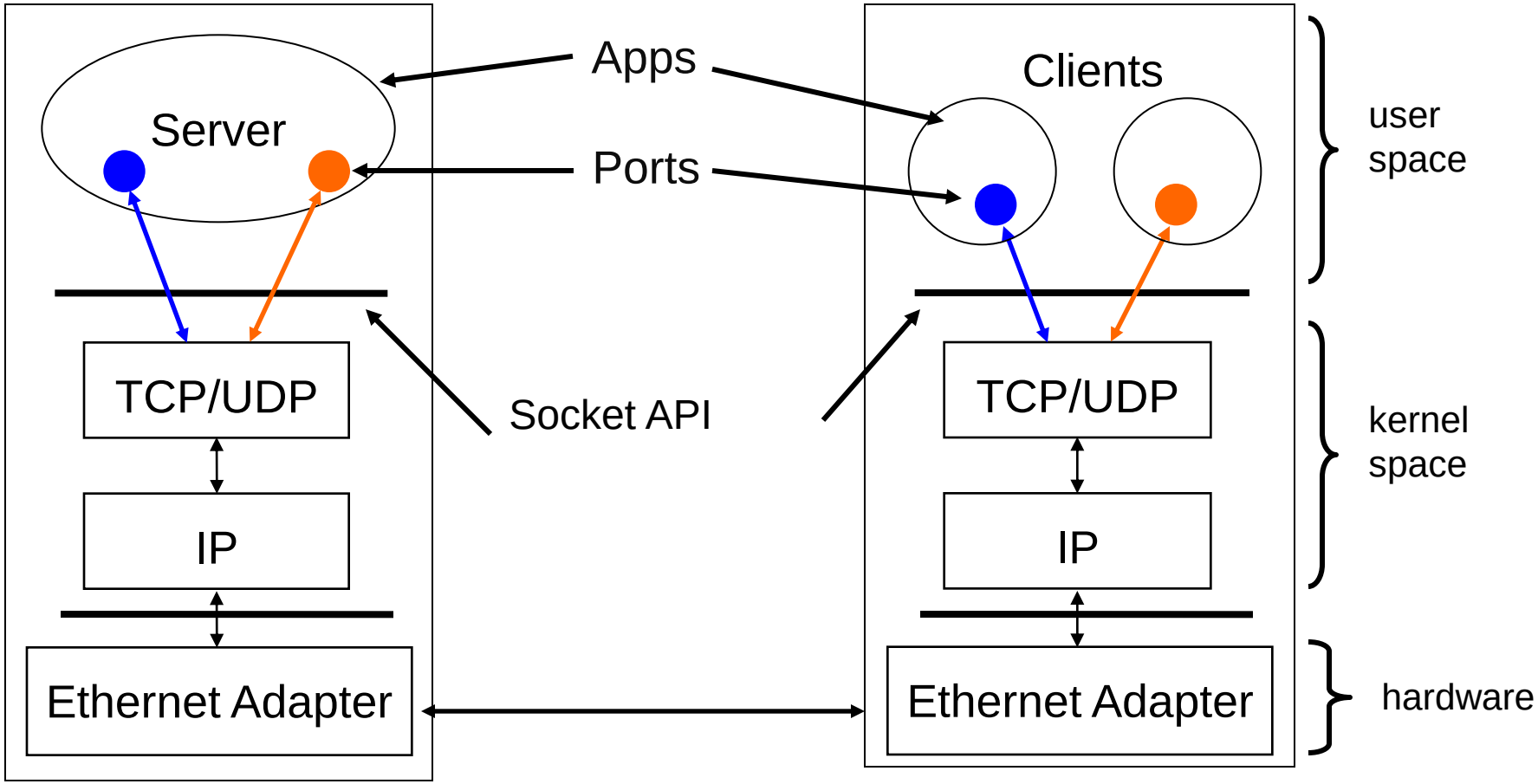
Transport layer Protocols

- **TCP Implementation**
  - **Congestion Control**

  How can routers tell the hosts using TCP (which is an end-to-end protocol) that there is congestion on the network?
  - Routers are not supposed to check TCP headers in transit packets, they just allowed to oversee IP headers.
    - Routers know when a network is congested (they are the first to know), so they can easily flip some bits in the IPv4 and IPv6 headers of the packets they route.
    - These bits are in the TOS (IPv4) and Flow (IPv6) fields, and the hosts can read these bits and react to them by adjusting windows when necessary.

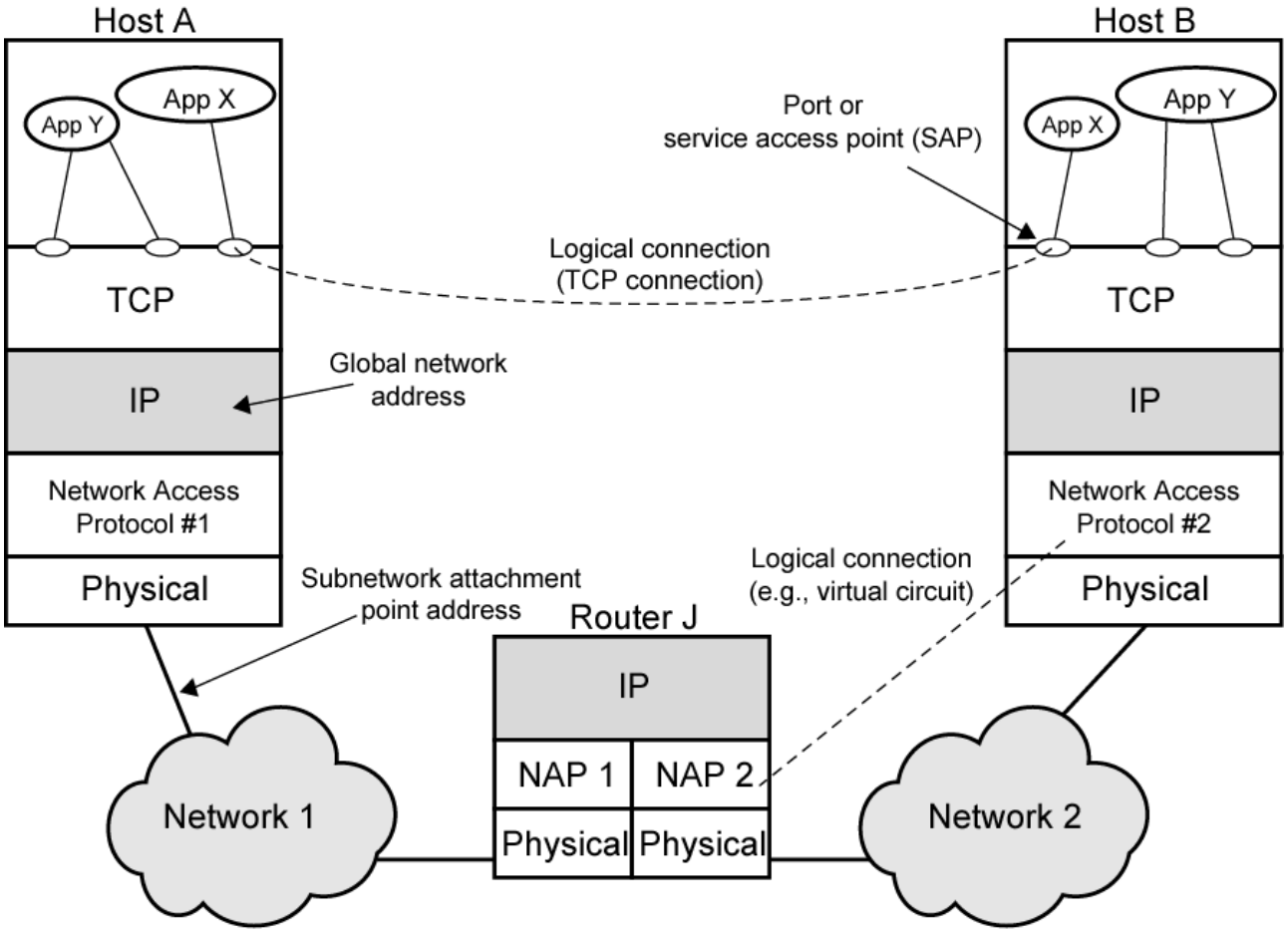
Communication Thru Transport Protocols

Server and Client exchange messages over the network through a common Socket API



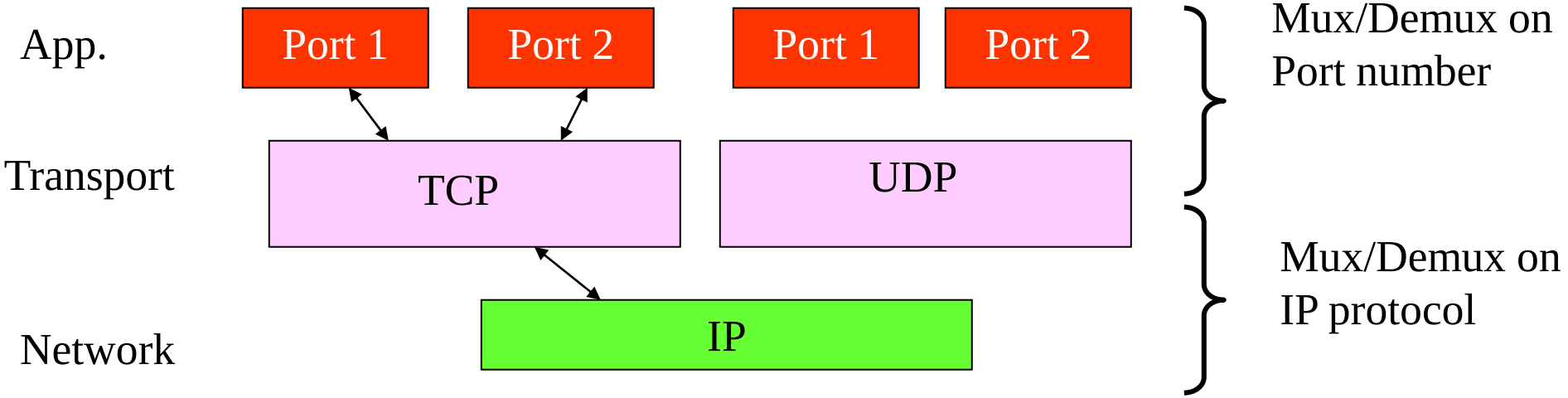
Transport layer Protocols

- **TCP Implementation**



Transport layer Protocols

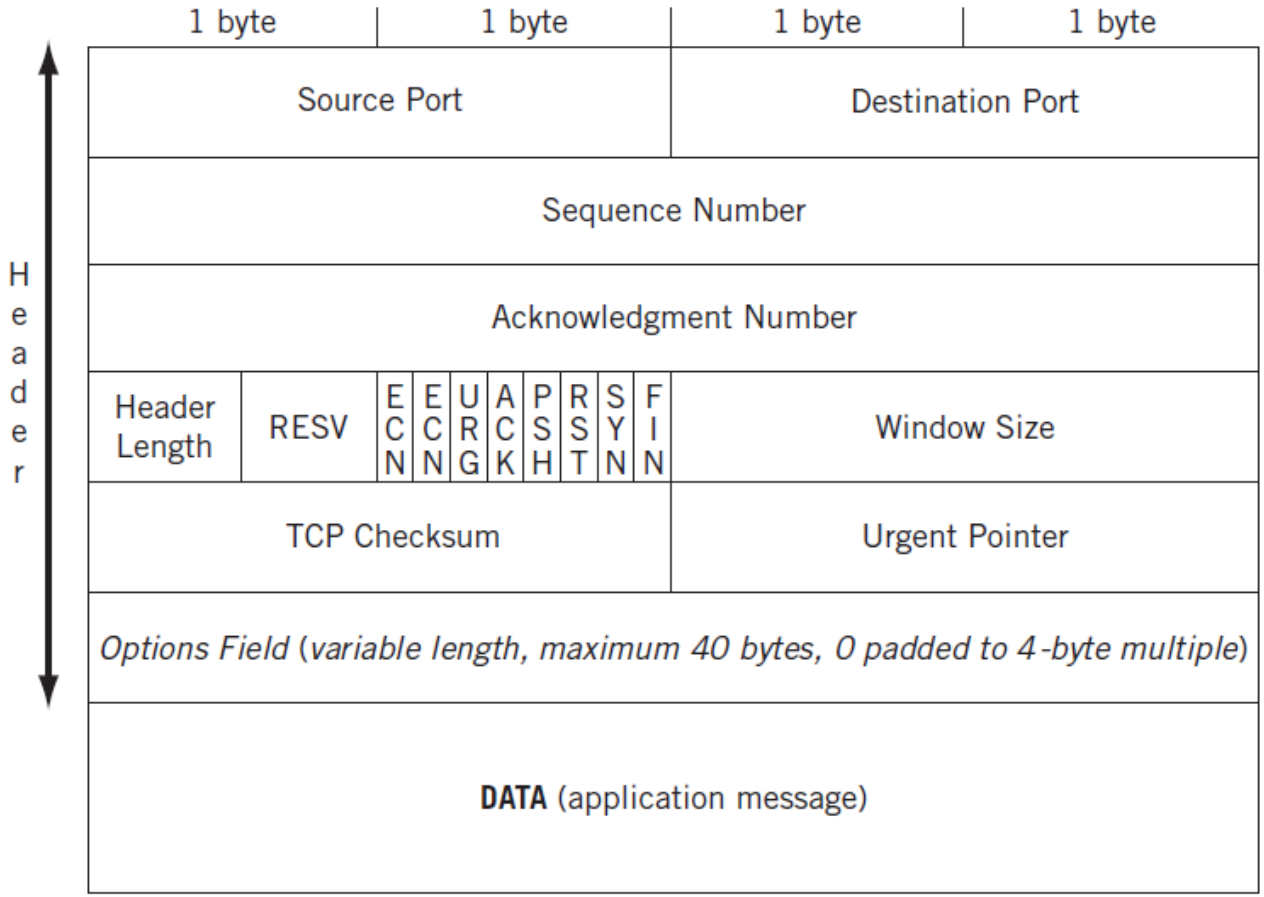
- **Transmission Control Protocol**



- For connection, needs:
  - Source: (address, port) AND Destination: (address, port)
  - Only need **one port on host to allow multiple connections, since each connection will have different (host, port) at other end**
- Passive open: application contacts OS & indicates will accept outgoing/incoming connection, OS assigns port and listens
- Active open: application requests OS to connect to an (host, port)

Transport layer Protocols

- **TCP Header**



Transport layer Protocols

- **TCP Header**

TCP header is the same for IPv4 and IPv6

TCP Segment Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Sequence Number			
64	Acknowledgment Number			
96	Data Offset	Res	Flags	Window Size
128	Header and Data Checksum		Urgent Pointer	
160...	Options			

Field	**Purpose**
Source Port	Identifies originating application
Destination Port	Identifies destination application
Sequence Number	Sequence number of first octet in the segment
Acknowledgment #	Sequence number of the next expected octet (if ACK flag set)
Len	**Length of TCP header in 4 octet units**
Flags	TCP flags: SYN, FIN, RST, PSH, ACK, URG
Window	Number of octets from ACK that sender will accept
Checksum	Checksum of IP pseudo-header + TCP header + data
Urgent Pointer	Pointer to end of “urgent data”
Options	Special TCP options such as MSS and Window Scale

Each new connection (re-tries of failed connections do not count)

The size of the fixed TCP header is 20 bytes , the size of the fixed IPv4 header is 20 bytes, and the size of the fixed IPv6 header is 40 bytes

Transport layer Protocols

TCP – Header

- The maximum segment size (MSS) is a parameter of the options field of the TCP header that specifies the largest amount of data, specified in bytes, that a computer or communications device can receive in a single TCP segment (in PCs usually set by operating system).
- To avoid fragmentation in the IP layer, a host must specify the maximum segment size as equal to the largest IP datagram that the host can handle excluding IP and TCP header sizes.
  - IPv4 hosts can handle an MSS of 536 bytes
  - IPv6 hosts can handle an MSS of 1220 bytes
- Small MSS values will reduce or eliminate IP fragmentation, but will result in higher overhead.
- Each direction of data flow can use a different MSS.

Transport layer Protocols

TCP vs UDP

- Some applications require reliable ordered delivery of packets.
- The TCP protocol provides this capability. It uses error detection, retransmissions and acknowledgements. This protocol cares about your data.
- Other applications don’t care if every packet is received. These applications can take advantage UDP’s lower overhead to enable faster transmissions.
- TCP is strictly used for point to point or unicast transmissions while UDP can also be used for multicast and broadcast transmissions.

Transport layer Protocols

Comparing Transport Protocols

TCP

- Reliable – guarantee delivery
- Byte stream – in-order delivery
- Connection-oriented – single socket per connection
- Slow transfer
- Setup connection followed by data transfer
- Unicast communication only
- Example TCP applications
  - Web, Email, Telnet

UDP

- No guarantee of delivery
- Not necessarily in-order delivery
- Connection-less - single socket to receive messages
- Fast transfer
- Datagram – independent packets
- Must address each packet
- Unicast, multicast or broadcast
- Example UDP applications
  - Multimedia, voice over IP, streaming traffic

Most Common Transport Protocols

Stream Control Transmission Protocol (SCTP)

- TCP has provided the primary means to transfer data reliably across the Internet. However, TCP has imposed limitations on several applications.
- Therefore, to remove this limitation SCTP is developed by IETF in 2000.
- SCTP is similar to TCP in many ways.
  - They are both unicast connection-oriented protocols that provide reliable transport, in-sequence packet delivery and rate-adaptive congestion control.
  - TCP has an additive 16-bit checksum and SCTP has a 32-bit CRC
- SCTP does provide some functions not found in TCP. SCTP is message-oriented whereas TCP is stream-oriented.
- SCTP can handle multiple simultaneous streams and multiplexed streams where TCP can handle only a single stream of data per connection.

Explanation in Next third slide

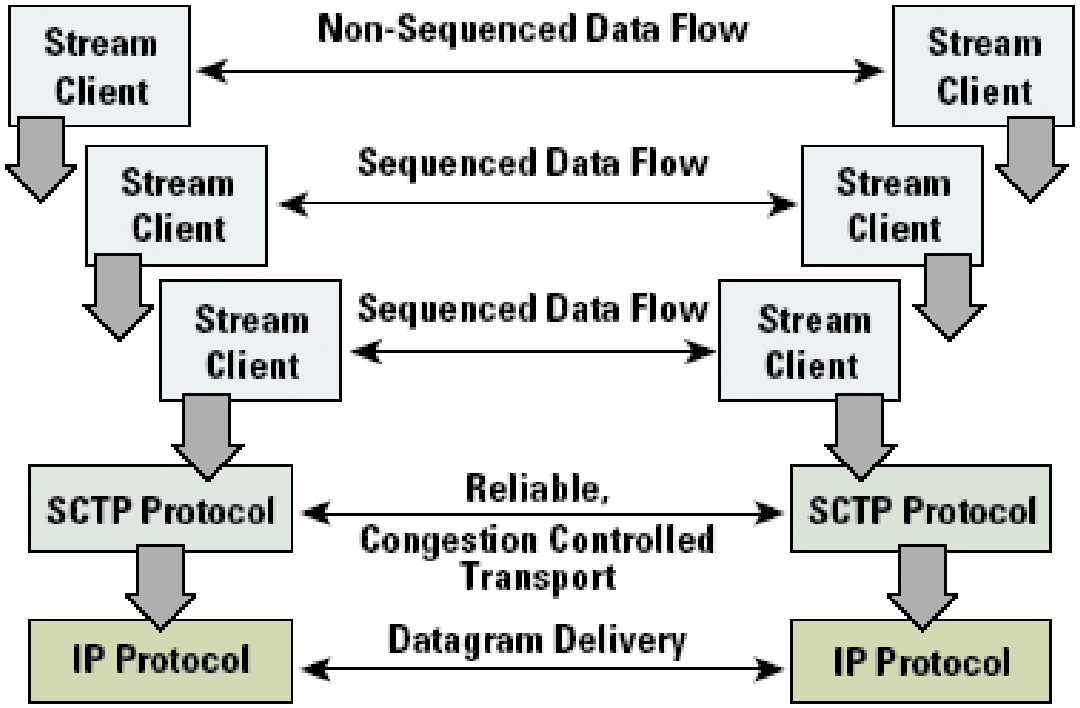
Most Common Transport Protocols

Stream Control Transmission Protocol (SCTP)

- SCTP also has many features that are similar to UDP. Both support unreliable transport and out-of-order packet delivery
- SCTP does have a 12-byte header compared to UDP's 8-byte header, but that is negligible when comparing performance between the protocols.
- **Therefore, not every application is well suited to either TCP or UDP and SCTP can provide the best of both TCP and UDP capabilities.**

Most Common Transport Protocols

Stream Control Transmission Protocol (SCTP)



Most Common Transport Protocols

Stream Control Transmission Protocol (SCTP)

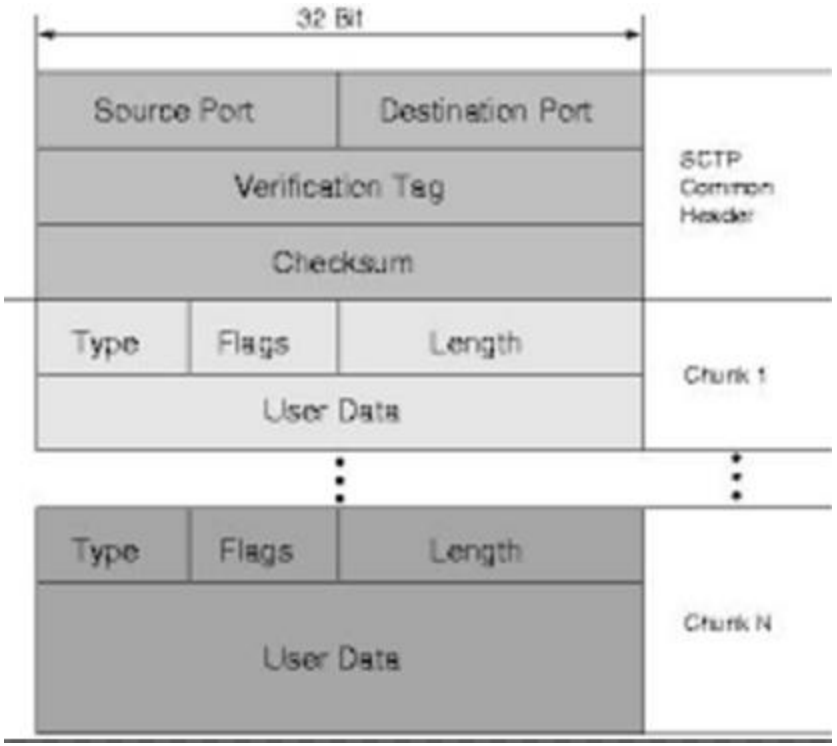
- SCTP applications submit their data to be transmitted in messages (groups of bytes) to the SCTP transport layer.
- SCTP places messages and control information into separate chunks (data chunks and control chunks), each identified by a chunk header.
- The protocol can fragment a message into a number of data chunks, but each data chunk contains data from only one user message.
- SCTP bundles the chunks into SCTP packets. The SCTP packet, which is submitted to the Internet Protocol, consists of a packet header, SCTP control chunks (when necessary), followed by SCTP data chunks (when available).

Most Common Transport Protocols

Stream Control Transmission Protocol (SCTP)

Bits	0–7	8–15	16–23	24–31
+0	Source port		Destination port	
32	Verification tag			
64	Checksum			
96	Chunk 1 type	Chunk 1 flags	Chunk 1 length	
128	Chunk 1 data			
...	...			
...	Chunk N type	Chunk N flags	Chunk N length	
...	Chunk N data			

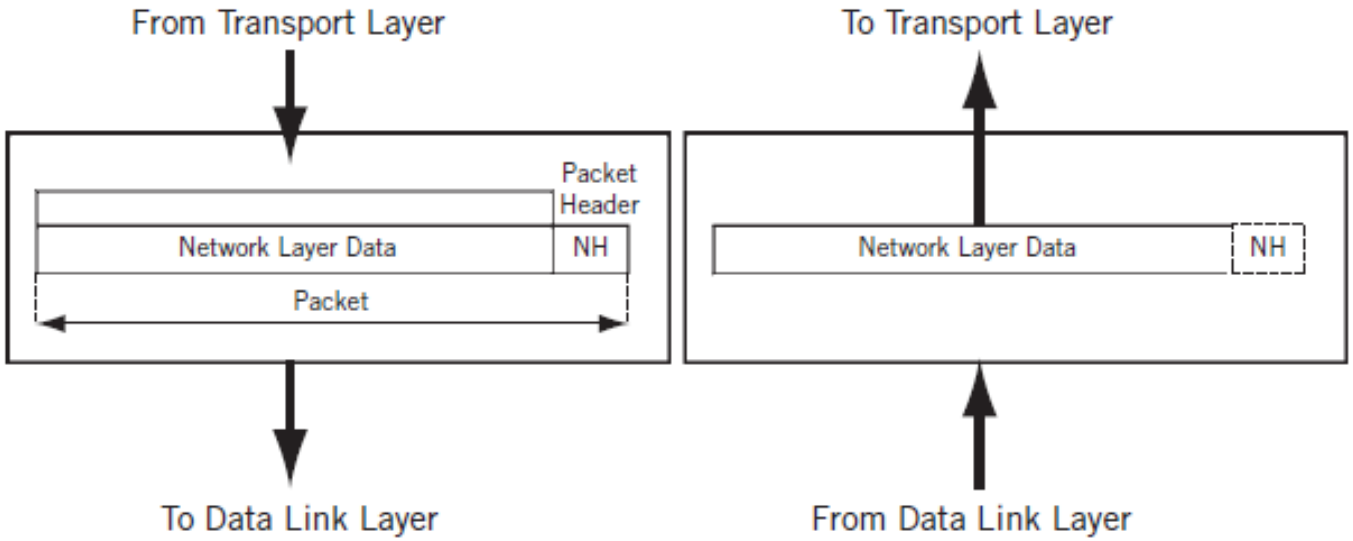
Another illustration



TCP/IP Protocols Suite

Internet / Network Layer

- It delivers data from transport to network interface layer (vice-versa), in the from of packets



TCP/IP Protocols Suite

Internet / Network Layer Protocols

- The Internet Protocol (IP) is a network layer (Layer 3) protocol that contains addressing information and some control information that enables packets to be routed.
- Most prominent feature of IP are the
  - Best-effort packet delivery service
  - IP addresses and IP prefixes
- The most popular IP are the
  - IPv4
  - IPv6

TCP/IP Protocols Suite

Internet / Network Layer

- The Internet/network layer delivers *packet* from source to destination, across as many links as necessary.
- There can be many different types of data link and physical layers on the network, depending on the variety of the link types, but the network layer is essentially the same on all systems, end systems, and intermediate systems alike.

TCP/IP Protocols Suite

Internet / Network Layer

How to Addresses?

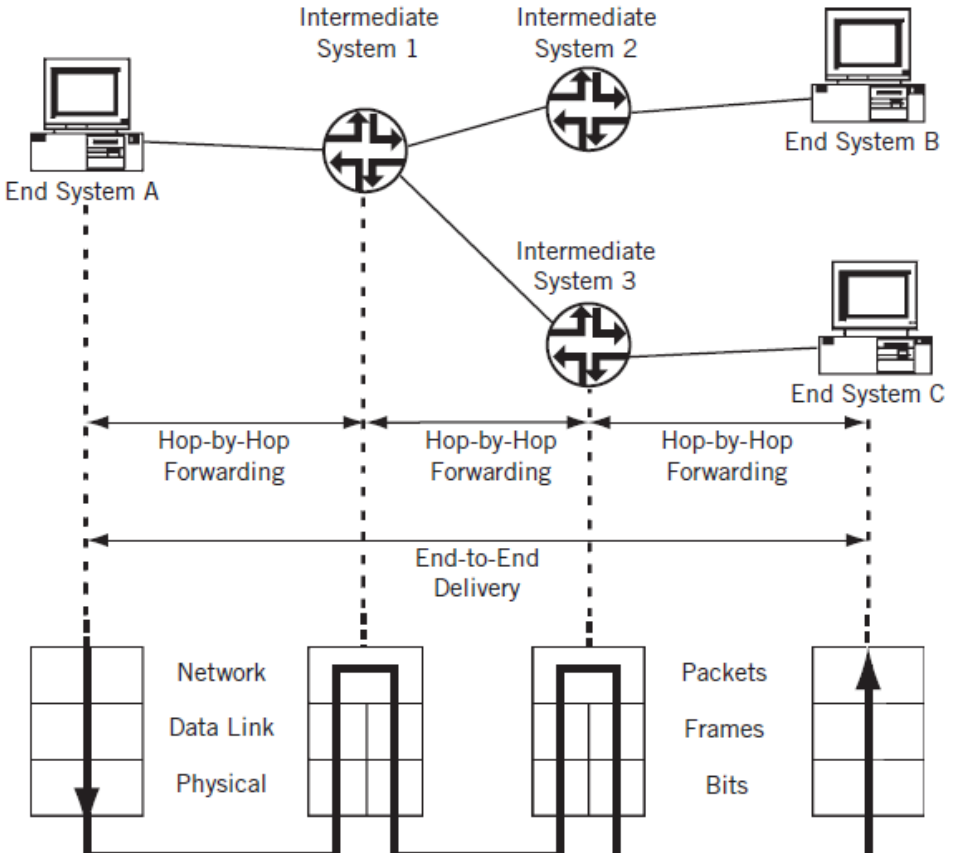
- MAC address are under ***IEEE*** guideline - no systematic assignment of physical addresses (and many addresses on WANs can be duplicates and so have "local significance only")
- IP addresses are under ***Internet Corporation for Assigned Names and Numbers (ICANN)*** - they are globally administered, unique, and have a portion under which many devices are grouped
  - only the network portion of the IP address is (like area codes) are globally administered, while the rest is (like telephone number) is locally administered, often independently.
  - ARIN, American Registry for Internet Numbers (US, Canada, and numerous North Atlantic and Caribbean islands)

The price for blocks of IPv4 addresses of 65,536 addresses or smaller is about $7 to $8 per address in the ARIN region

TCP/IP Protocols Suite

Source-to-Destination Delivery at the Network Layer

Building tables to pass data from source to destination is called *routing*, and the use of these tables for packet delivery is called *forwarding*. The forwarding of packets inside frames always takes place hop by hop.



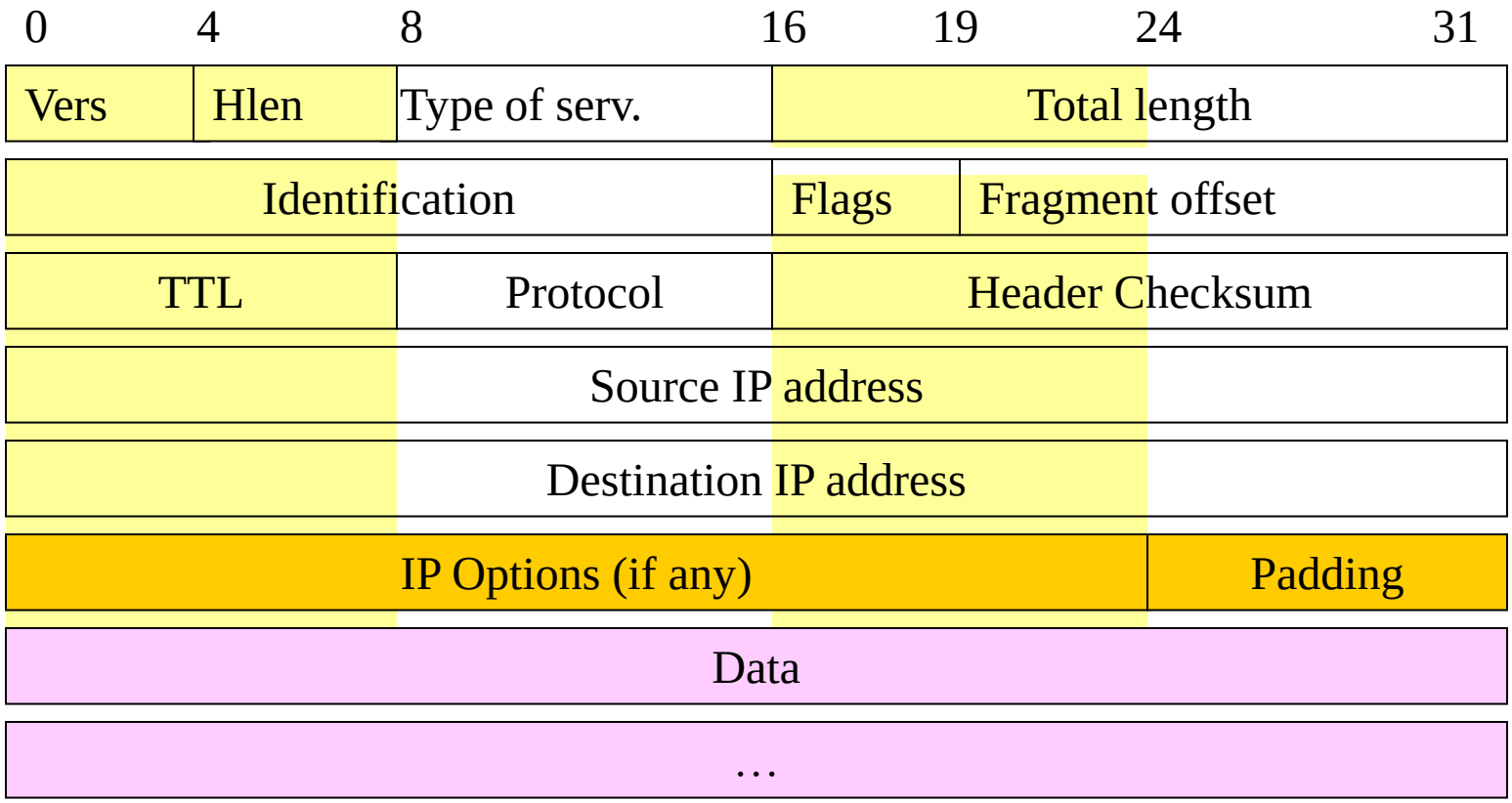
On the Internet, the intermediate systems that act at the packet level (Layer 3) are called *routers* and the devices that act on frames (Layer 2) are called *switches*

TCP/IP Protocols Suite

- **Internet / Network Layer Protocols**
  - **Maximum Transmission Unit (MTU)**
  - It is the maximum size of an IP packet that can be transmitted without fragmentation
    - IPv4 At least 68,max of 64KB
    - IPv6 At least 1280,max of 64KB, (up to 4GB with optional jumbogram)
    - Ethernet (v2) 1500 bytes
  - "Path MTU Discovery", a technique for determining the path MTU between two IP hosts. It works by setting the DF (Don't Fragment) option in the IP headers of outgoing packets. Any device along the path whose MTU is smaller than the packet will drop such packets and send back an ICMP "*Fragmentation Needed / Datagram Too Big*" message containing its MTU. This information allows the source host to reduce its assumed path MTU appropriately. The process repeats until the MTU becomes small enough to traverse the entire path without fragmentation.

TCP/IP Protocols Suite

- **IP Header Format**

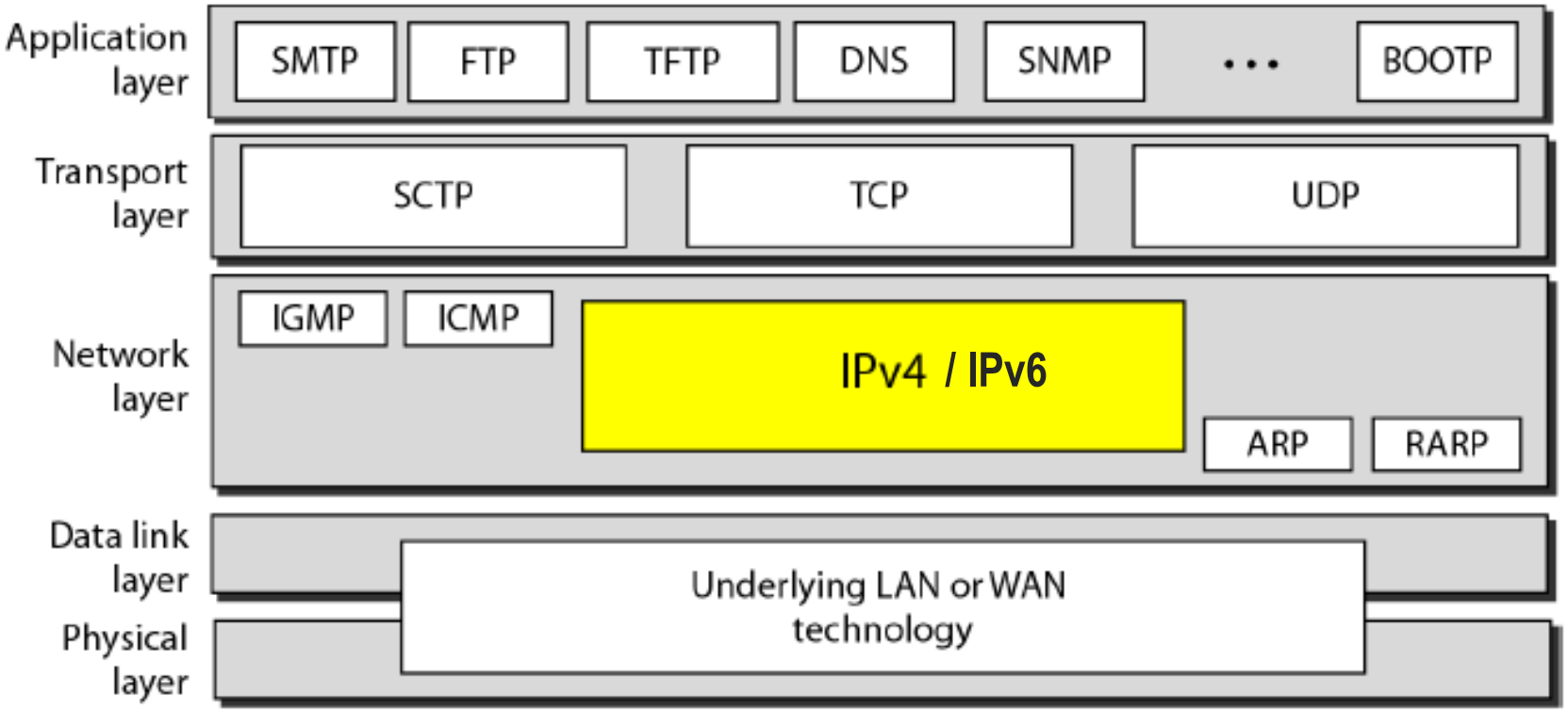


TCP/IP Protocols Suite

- **Vers** (4 bits): version of IP protocol (IPv4=4 or =6)
- **Hlen (**4 bits): Header length in 32 bit words, without options (usual case) = 20
- **Type of Service – TOS** (8 bits): Being used for QoS
- **Total length** (16 bits): length in bytes, includes header and data
- **Time to live – TTL** (8bits): specifies how long packet is allowed to remain in internet
  - Routers decrement by 1
  - When TTL = 0 router discards packets
- **Protocol** (8 bits): specifies the format of the data area
  - Protocol numbers administered by central authority to guarantee agreement, e.g. TCP=6, UDP=17 etc.

TCP/IP Protocols Suite

- **Source-to-Destination Delivery**



TCP/IP Protocols Suite

Internet layer Protocols

- The goal is to create low-level functionality that hides physical addresses and allows higher-level programs to work only with internet addresses.
  - Two machines on a given physical network can communicate only if they know each other's physical address.

TCP/IP Protocols Suite

Internet layer Protocols

- A has only B's internet address, now how to map?
  - First, while final delivering a packet, the packet must be sent across one physical network to its final destination. The computer sending the packet must map the final destination's Internet address to the destination's physical address.
  - Second, at any point along the path from the source to the destination other than the final step, the packet must be sent to an intermediate router. Thus, the sender must map the intermediate router's Internet address to a physical address.

TCP/IP Protocols Suite

Address Resolution Protocol (ARP)

- When host *A* wants to resolve IP address I_B it broadcasts a special packet that asks the host with IP address I_B to respond with its physical address, P_B. *All* hosts, including B, receive the request, but only host B recognizes its IP address and sends a reply that contains its physical address.
- When *A* receives the reply, it uses the physical address to send the internet packet directly to B.
- Summarize:
  - Knows IP address only
  - Asked for MAC

TCP/IP Protocols Suite

Address Resolution Protocol (ARP)

The Address Resolution Protocol, ARP, allows a host to find the physical address of a target host on the same physical network, given only the target's IP address.

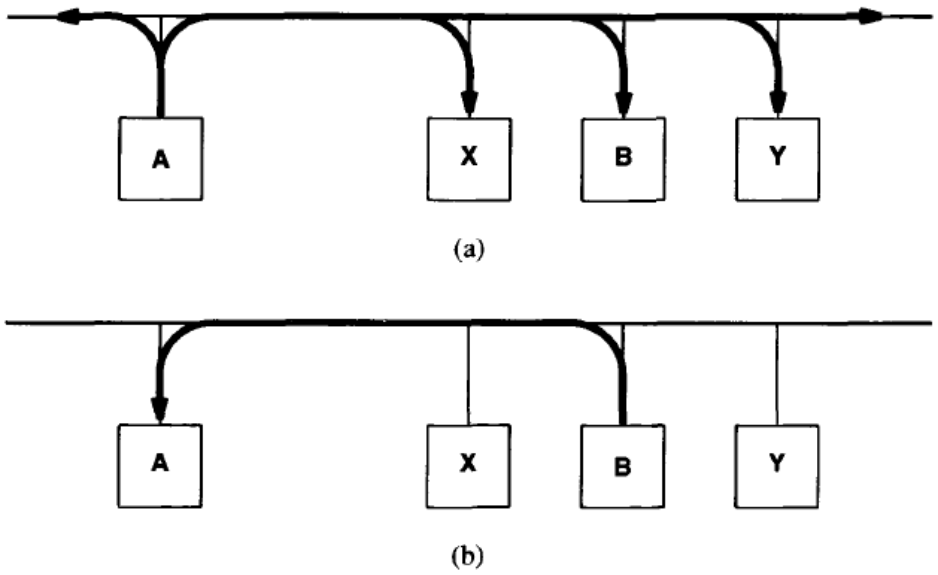


Figure 5.1 The ARP protocol. To determine P_B, B's physical address, from I_B, its IP address, (a) host A broadcasts an ARP request containing I_B to all machines on the net, and (b) host B responds with an ARP reply that contains the pair (I_B, P_B).

TCP/IP Protocols Suite

Address Resolution Protocol (ARP)

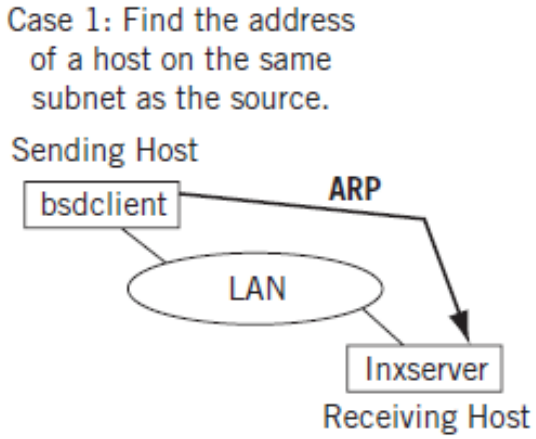
- Broadcasting is far too expensive to be used every time one machine needs to transmit a packet to another because every machine on the network must receive and process the broadcast packet.
- Therefore, ARP results are cached.
  - To reduce communication costs, computers that use ARP maintain a cache of recently acquired IP-to-MAC mappings
    - The devices that send the ARP requests cache the results, and the device that receives the ARP usually also caches the MAC address in the arriving ARP request.
    - The idea is that if one device in a pair sends in one direction, the other device in the pair will probably send in the opposite direction as well.
    - That is, whenever a computer sends an ARP request and receives an ARP reply, it saves the IP address and corresponding hardware address information in its cache for successive lookups.

TCP/IP Protocols Suite

Address Resolution Protocol (ARP)

- ***Host to host***—The ARP sender is a host and wants to send a packet to another host on the same LAN. In this case, the IP address of the destination is known and the MAC address of the destination must be found.

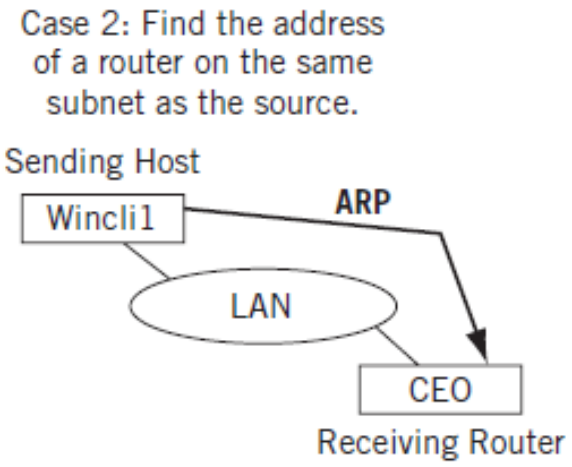
ARP is used when a host wants to send to another host on the same network and the MAC address of the destination is not already known. LAN host broadcast an ARP request on LAN and the sender waits for a reply.



TCP/IP Protocols Suite

Address Resolution Protocol (ARP)

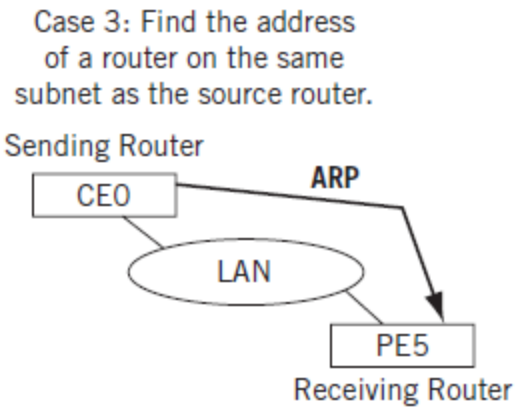
- ***Host to router***—The ARP sender is a host and wants to send a packet to another host on a different LAN. A forwarding (routing) table is used to find the IP address of the router. In this case, the IP address of the router is known and the MAC address of the router must be found.



TCP/IP Protocols Suite

Address Resolution Protocol (ARP)

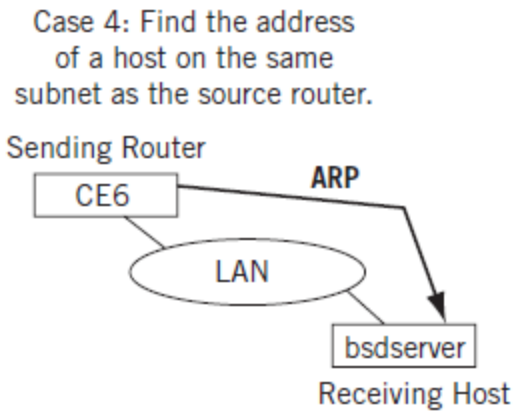
- ***Router to router***—The ARP sender is a router and wants to forward a packet to another router on the same LAN. A forwarding (routing) table is used to find the IP address of the router. In this case, the IP address of the router is known and the MAC address of the destination router must be found.



TCP/IP Protocols Suite

Address Resolution Protocol (ARP)

- ***Router to host***—The ARP sender is a router and wants to forward a packet to a host on the same LAN. In this case, the IP address of the host is known (from the IP destination address on the packet) and the MAC address of the host must be found.



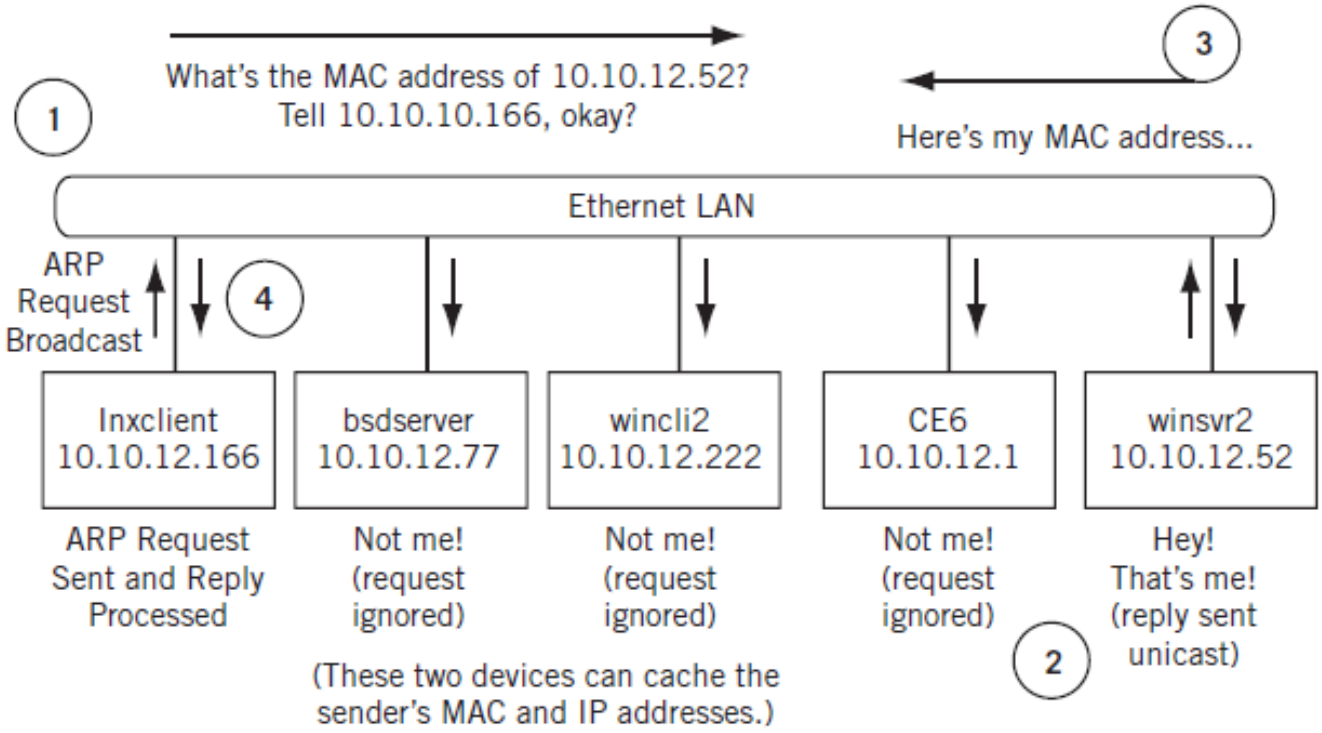
TCP/IP Protocols Suite

ARP Operation

- ARP process adds to TCP/IP is a mechanism for a source device to ask, “Who has IP address 10.10.12.52 and what is the physical (MAC) address associated with it?”
- ARP messages are broadcast frames sent to all stations. The proper destination IP layer realizes that the destination IP address in the packet matches its own and replies directly to the sender.
- The target device replies by simply reversing the source and destination IP address in the ARP packet. The target also uses its own MAC address as the source address in the frame and message.

TCP/IP Protocols Suite

ARP Operation



TCP/IP Protocols Suite

Address Resolution Protocol (ARP)

- Functionally, ARP is divided into two parts:
  - The first part maps an IP address to a physical address when sending a packet, and
  - the second part answers requests from other machines.
- ARP operation is completely transparent to the user. ARP operation is usually triggered when a user runs some TCP/IP application, such as FTP, and the frame's destination MAC address is not in the ARP cache.

TCP/IP Protocols Suite

ARP Variations

- The main address resolution protocol is the Address Resolution Protocol (ARP) itself, but there are also
  - Reverse ARP (RARP),
  - Proxy ARP,
  - Inverse ARP (InARP),
  - ARP for ATM networks (ATMARP).
- There are more network types than LANs and there are more "addresses" that need to be associated with IP addresses than "hardware" addresses.

TCP/IP Protocols Suite

Reverse Address Resolution Protocol (RARP)

- Reverse of ARP
- When host knows only MAC address
- Reverse ARP (RARP) is used in cases where a device on a TCP/IP network knows its physical (MAC) address but must determine the IP address associated with it.
- A RARP request ("I have MAC address X.X.X.X. What's my IP address?") is sent to a device running the RARP server process. The RARP server replies with the IP address of the device.
- Dynamic Host Configuration Protocol (DHCP) has mostly replaced RARP
- Summarize:
  - Knows MAC address only
  - Asked for IP address

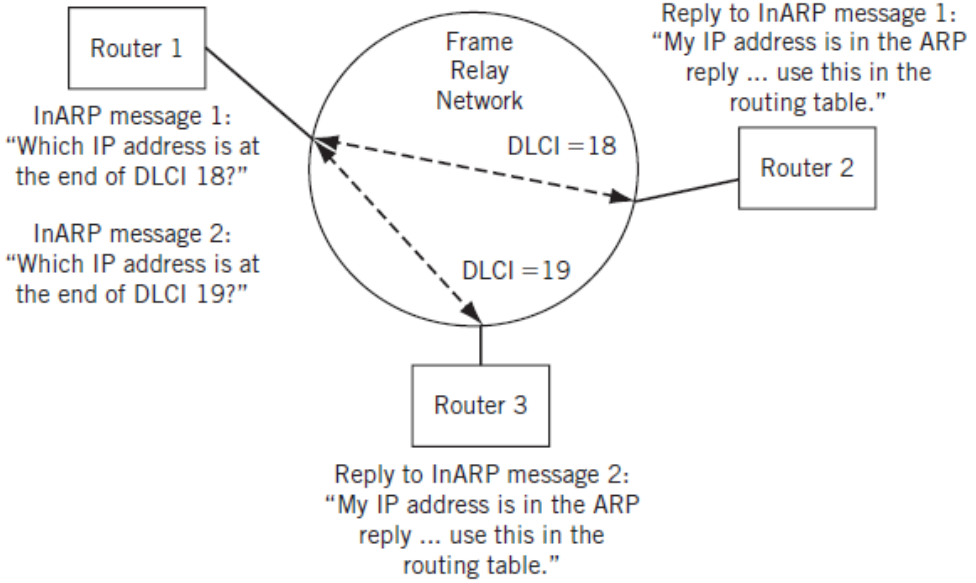
TCP/IP Protocols Suite

ARP Variations

- On most WANs, ARP is still used, but as a limited multicast rather than a broadcast. ARP has a couple of variations used to address WAN environments such as frame relay and ATM networks.
- These public network technologies use *virtual circuits* (a type of logical connection) at the frame (frame relay) or cell (ATM) level instead of MAC addresses.
- Frame relay uses InARP
- ATM uses ATMARP, it can used to find the ATM virtual path identifier (VPI) and/or virtual channel identifier (VCI) over an ATM network
- The issue in frame relay and ATM is to find the virtual circuit number, associated with a particular IP address.

TCP/IP Protocols Suite

Inverse Resolution Protocol (InARP)



Inverse ARP (InARP) exchange over a frame relay network. In this case, the hardware address (DLCI) is known and the sender needs to determine the IP address.

- InARP (Inverse ARP) was developed for use on frame relay networks.
- Instead of using ARP to determine MAC-layer LAN addresses, TCP/IP networks linked by frame relay networks use InARP to determine the IP address at the other end of a frame relay Data Link Connection Identifier (DLCI) number to use when sending IP packets

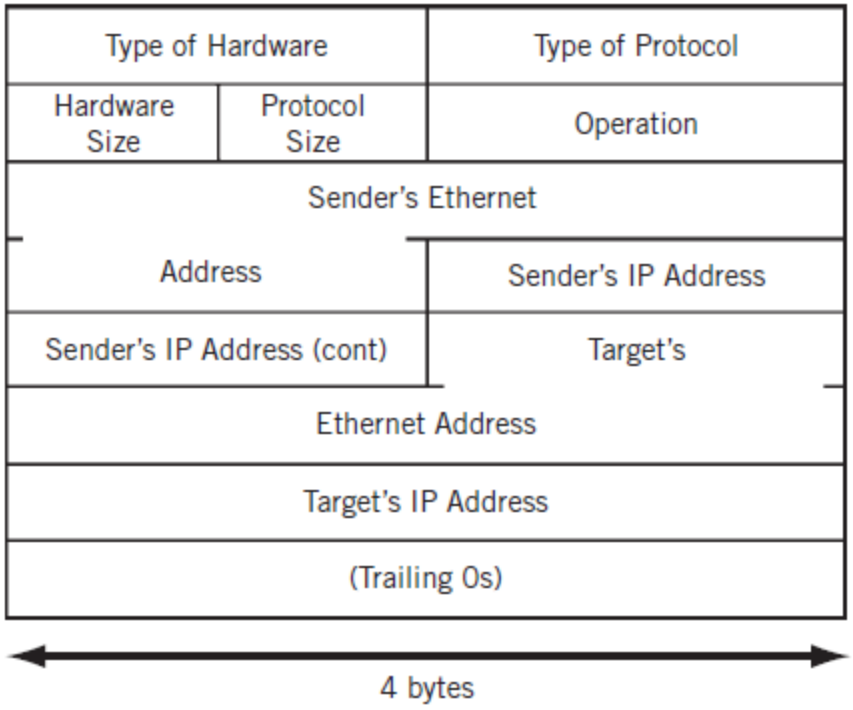
TCP/IP Protocols Suite

Address Resolution Protocol (ARP) Packet Format

- ARP uses packets, but these are not IP packets. ARP messages ride inside Ethernet frames, or any LAN frame, in exactly the same way as IP packets.
- There is no need to use an IP address here anyway: ARP frames are valid only for a particular LAN segment and never leave the local LAN (i.e., ARP messages cannot be routed).

TCP/IP Protocols Suite

Address Resolution Protocol (ARP) Packet Format



The ARP message's fi elds. The message is placed directly inside a frame, such as an Ethernet Frame.

Hardware Size—This byte identifies the size, in bytes, of the hardware address. The Ethernet MAC address is 6 bytes long.
Protocol Size—This byte identifies the size, in bytes, of the Layer 3 protocols. IPv4 addresses are 4 bytes long.
Operation—This 2-byte field identifies the ARP message's intent. For example, an ARP request ("Who has this IPv4 address?") has the operation value of 1 and a reply value of 2.

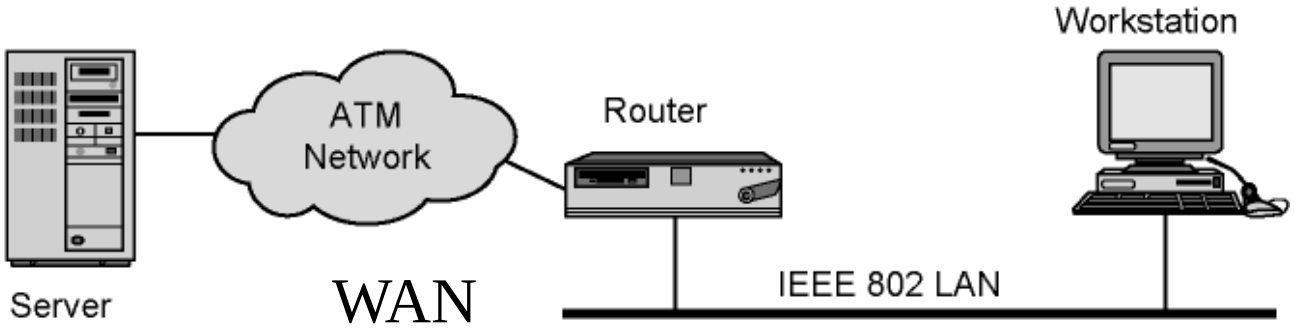
Communication with TCP/IP Protocols

(Sending Packet over Internet)

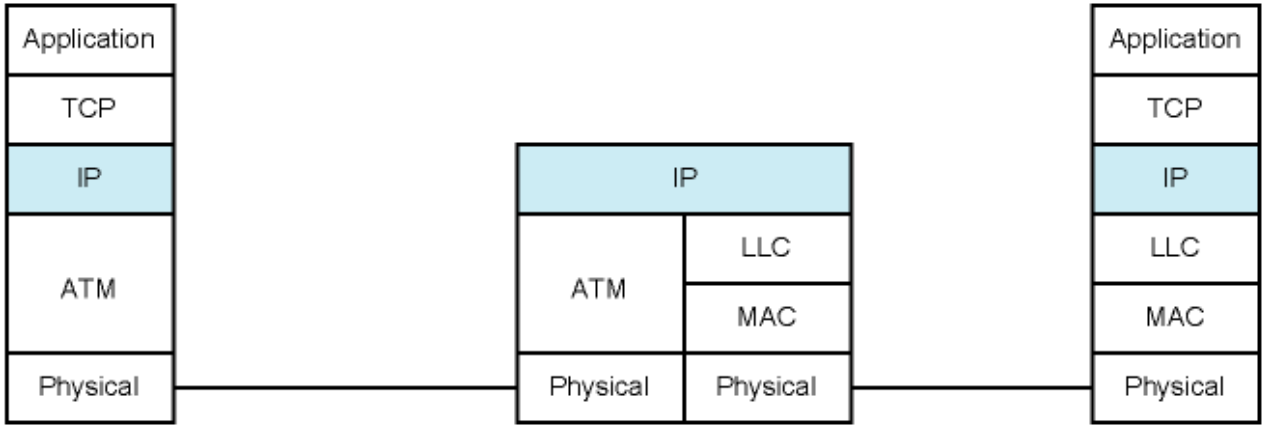
TCP/IP PROTOCOLS SUITE

Communication with TCP/IP Protocols

Scenario of end-to-end communication between **Workstation** and **Server** over the network



Operation of each layer participating in communication



Action of Sender

1. Preparing the data. The application protocol prepares a block of data for transmission. For example, an email message (SMTP), a file (FTP), or a block of user input (Telnet).

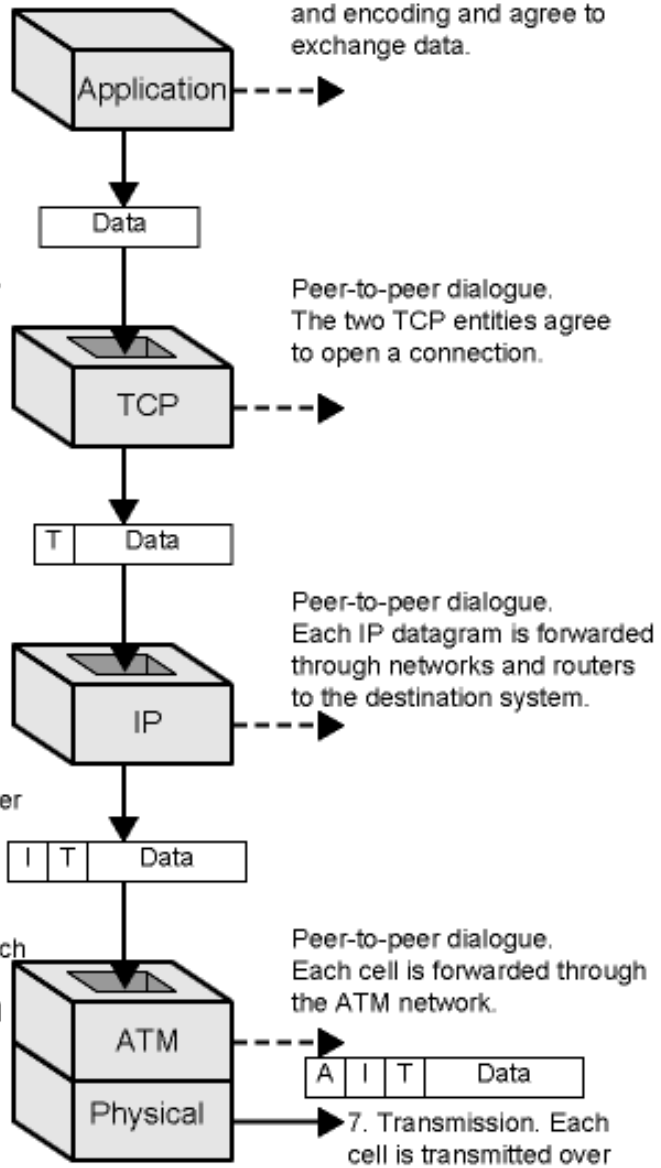
2. Using a common syntax. If necessary, the data are converted to a form expected by the destination. This may include a different character code, the use of encryption, and/or compression.

3. Segmenting the data. TCP may break the data block into a number of segments, keeping track of their sequence. Each TCP segment includes a header containing a sequence number and a frame check sequence to detect errors.

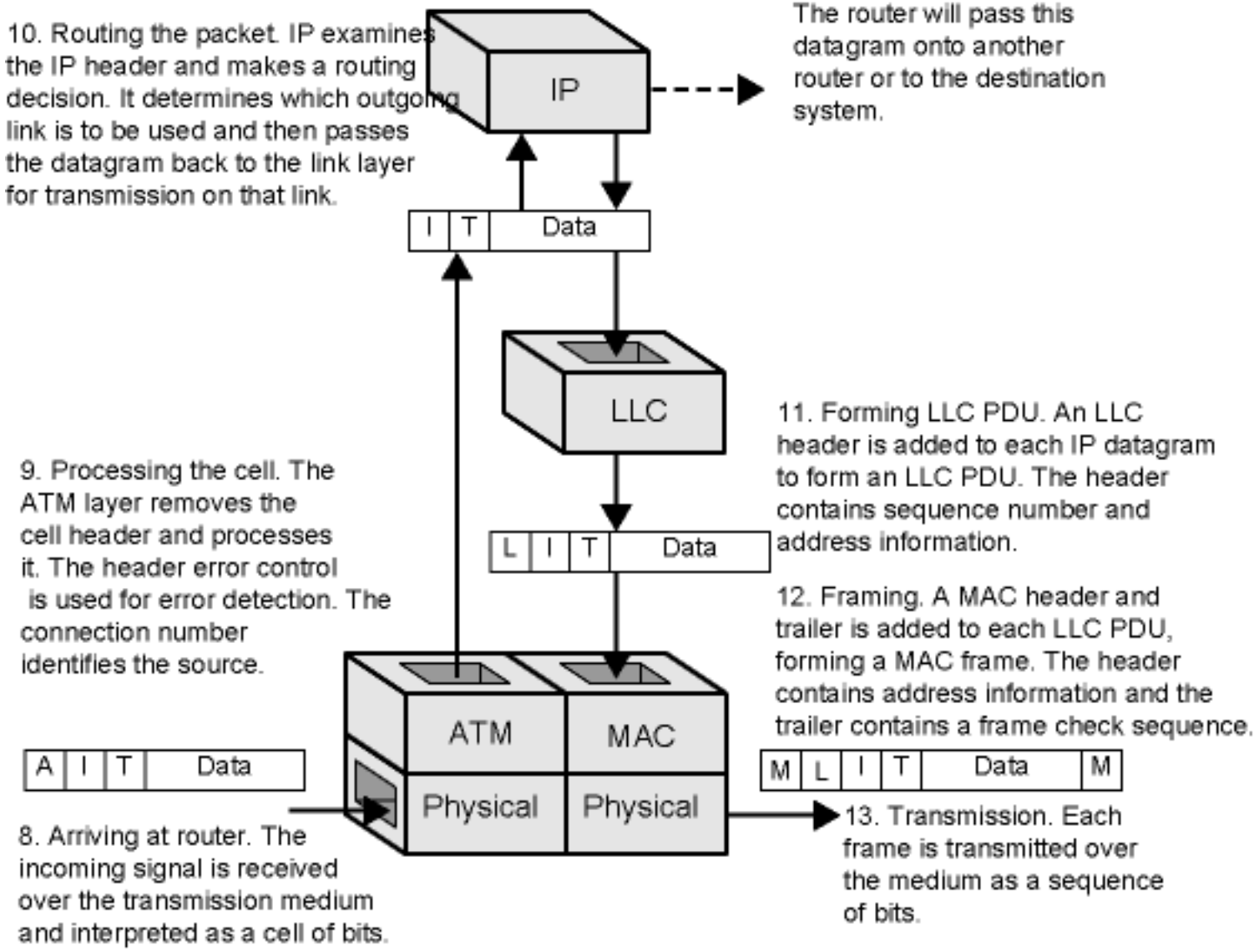
4. Duplicating segments. A copy is made of each TCP segment, in case the loss or damage of a segment necessitates retransmission. When an acknowledgment is received from the other TCP entity, a segment is erased.

5. Fragmenting the segments. IP may break a TCP segment into a number of datagrams to meet size requirements of the intervening networks. Each datagram includes a header containing a destination address, a frame check sequence, and other control information.

6. Framing. An ATM header is added to each IP datagram to form an ATM cell. The header contains a connection identifier and a header error control field



Action of Router



Action of Receiver