Computer Networking Basics

Prof. Chien-Chao Tseng

曾 建 超 教授

Department of Computer Science
National Yang Ming Chiao Tung University
cctseng@cs.nctu.edu.tw

References:

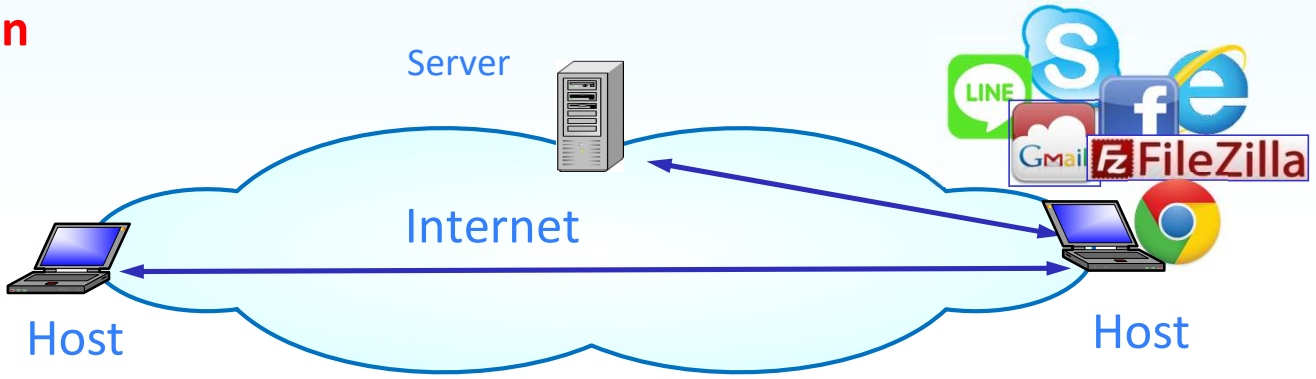
National Chiao Tung University

Outline

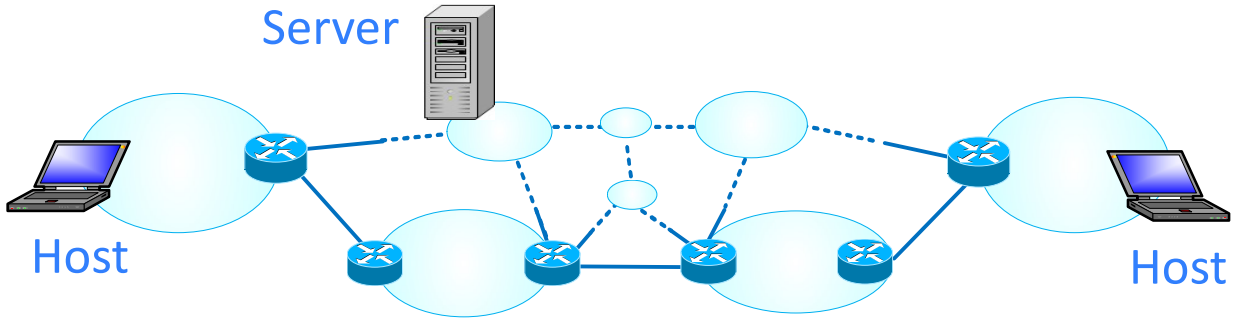
- Internet Protocol and Packet Multiplexing
- Layer 2 Switching
- IP Addressing and Forwarding
- ARP, DHCP, and ICMP

Internet: Interconnected networks

- **Abstraction**

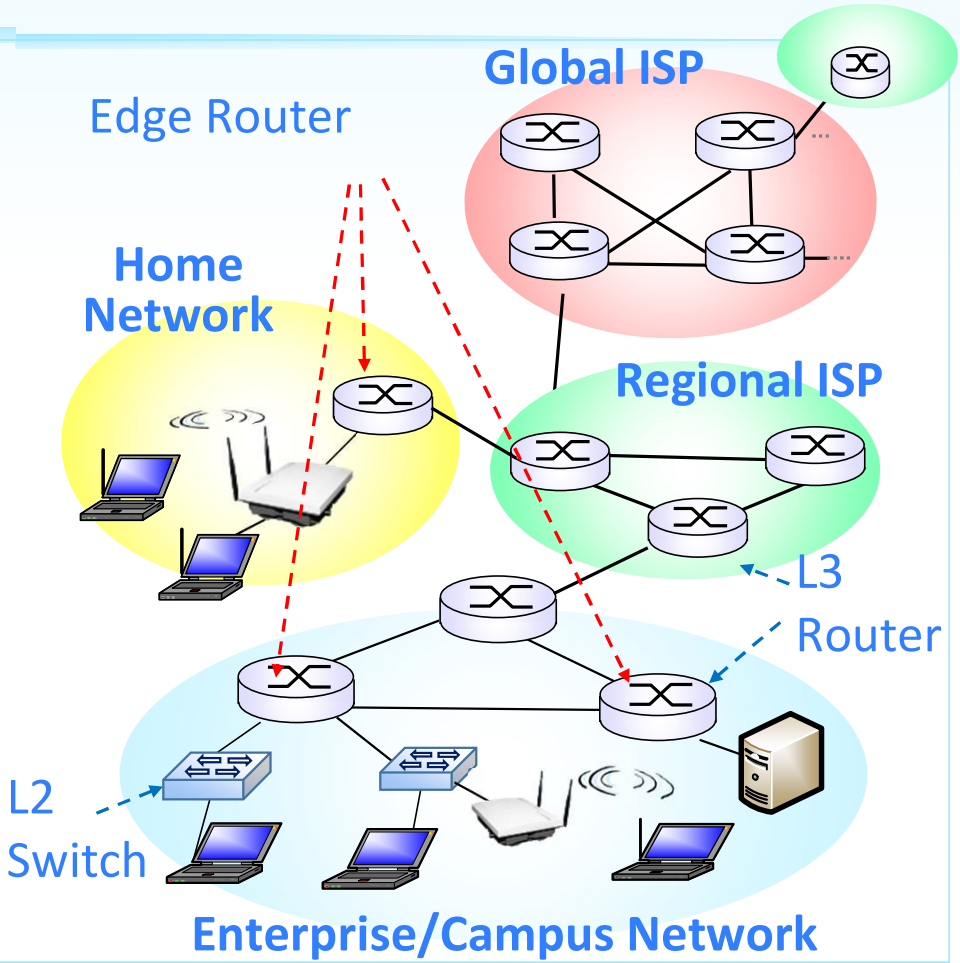


- ***Internet:*** "network of networks"



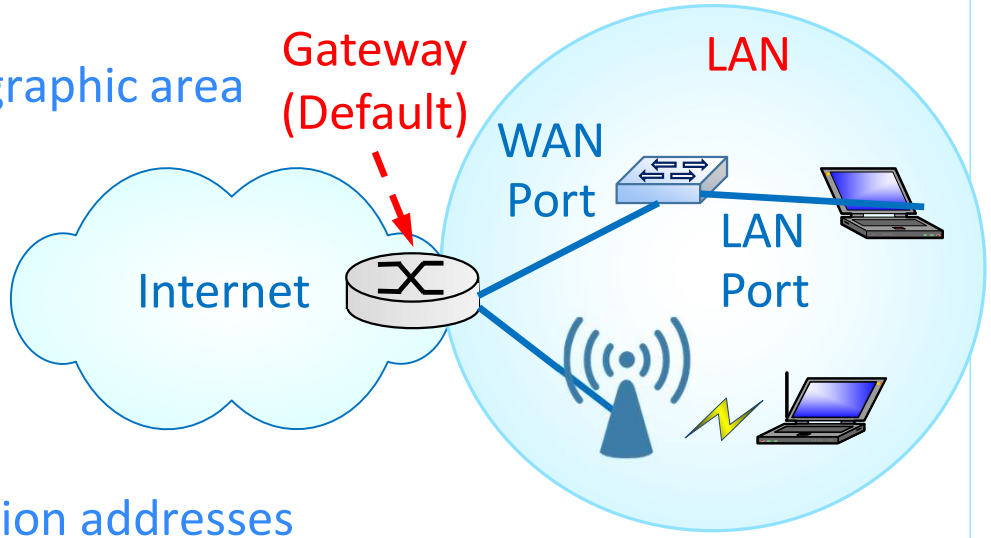
A Closer Look at Internet Structure

1) **Network Edge**
   - Hosts: clients and servers
     - May in data centers
2) **Access Networks**
   - Networks that connect **end systems** to **edge routers**
   - Edge router: **first router** on a path from a host to another distance host
   - Physical media: wired, wireless communication links
3) **Network Core**
   - Mesh of interconnected routers
   - **Network of networks**



Connecting to Networks

- **Local Area Network (LAN)**
  Network that interconnects computers within a limited area – a residence, laboratory, office building, or campus
- **Wide Area Network (WAN)**
  Network that extends over a large geographic area
- **Gateway**
  Network node that passes traffic from a local network to other networks or Internet
- **Default Gateway**
  Forwarding node for unknown destination addresses



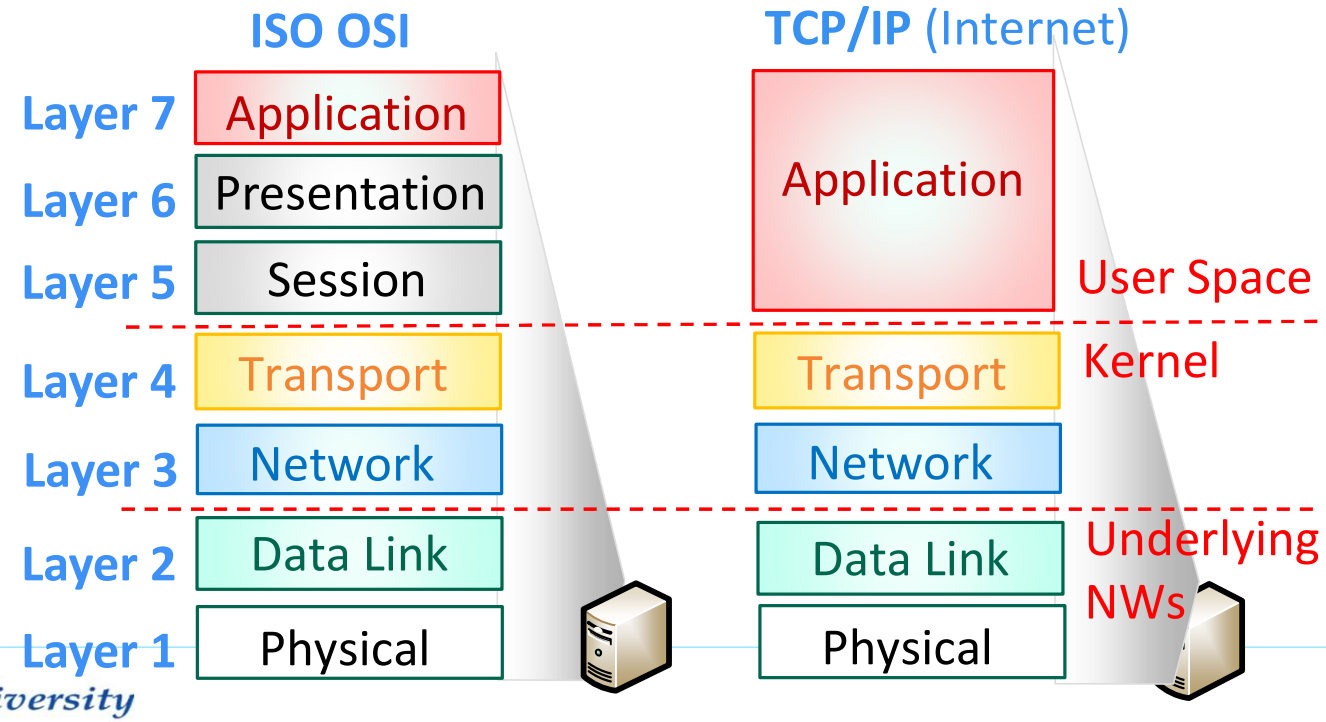
Your Network Configuration

- **Two Approaches**
  - **Automatic Configuration** (via **Dynamic Host Configuration Protocol;** DHCP)
  - **Manual Configuration**

1. **Internet Protocol (IP) Address**
   - E.g., 140.113.24.00/24
2. **Subnet Mask**
   - Mask for extracting network address (NWID)
     - E.g., 255.255.255.0
3. **Default Gateway**
   - Forwarding node for unknown destinations
4. **Domain Name System** Servers
   - Map domain names to IP addresses

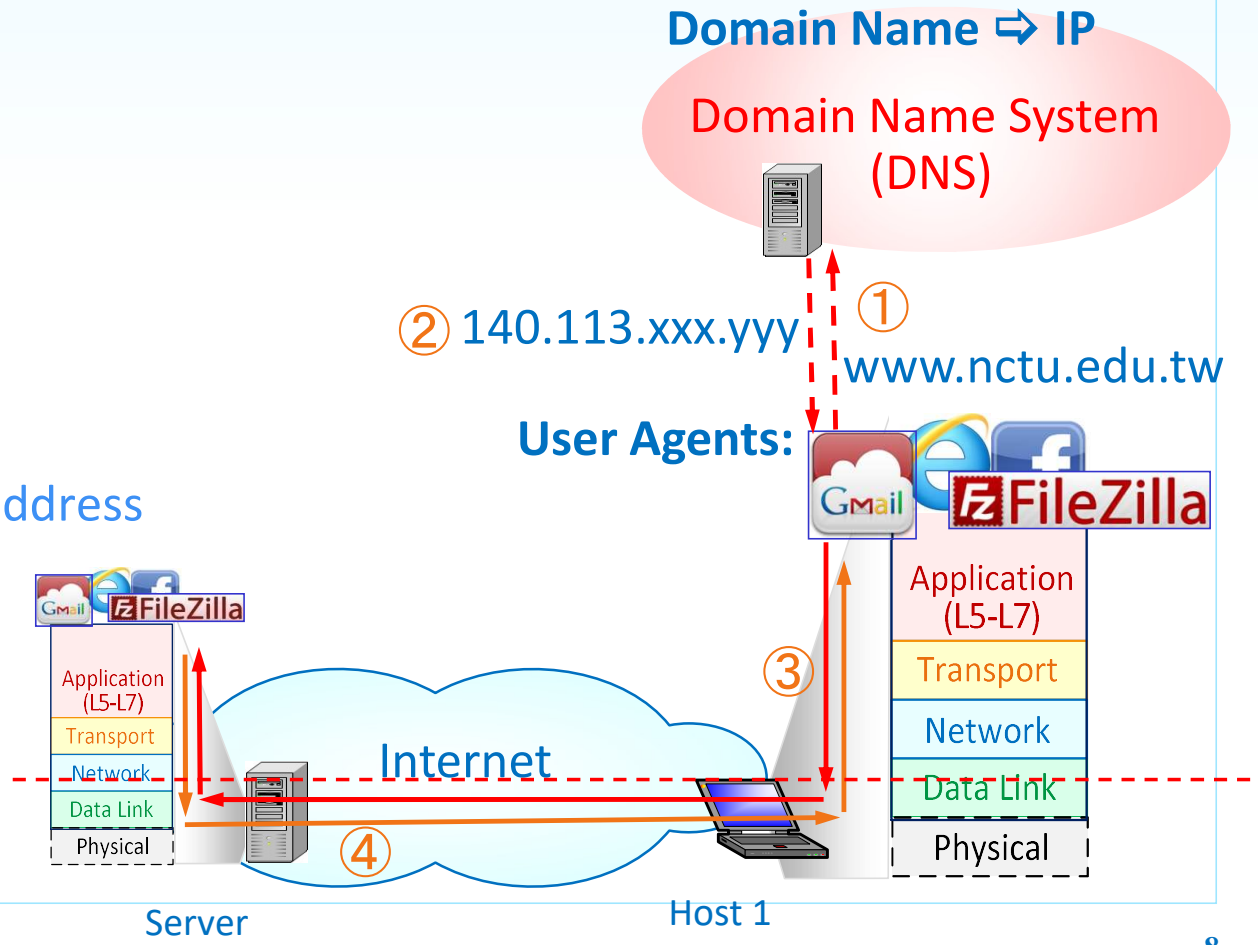
TCP/IP and OSI ISO Model

- ISO: International Standards Organization
  - OSI: Open Systems Interconnection (OSI)
- TCP/IP: Transport Control Protocol/Internet Protocol
  - four software layers built upon hardware



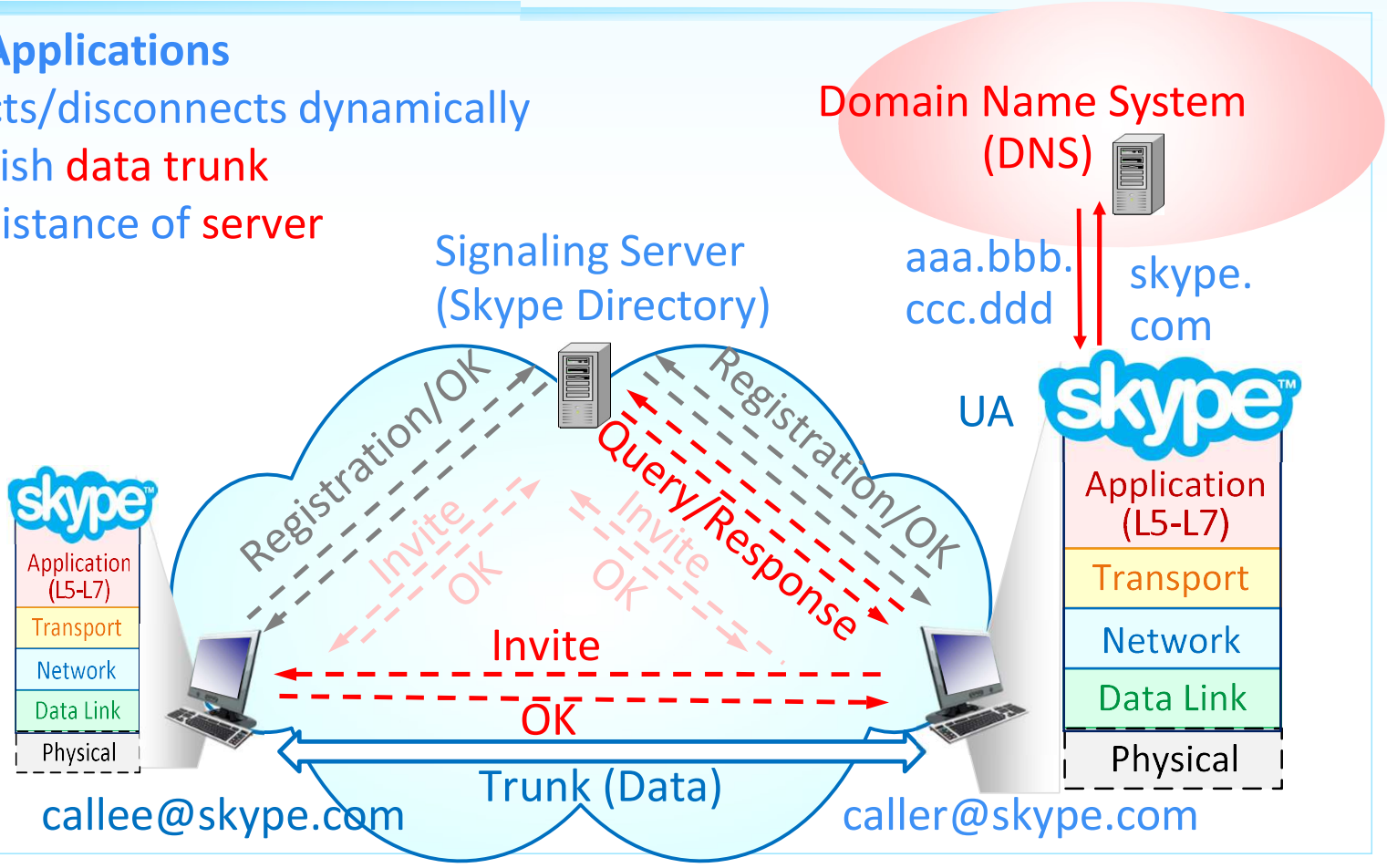
How Application Works – Client-Serve

- Host IDs in Internet
  - IPs: 140.113.xxx.yyy
- **Mnemonics Names**
  - Domain Names
    - www.cs.nctu.edu.tw
    - ftp.cs.nctu.edu.tw
- **Domain Name System**
  - Map domain name to IP address
- Client-Server Applications

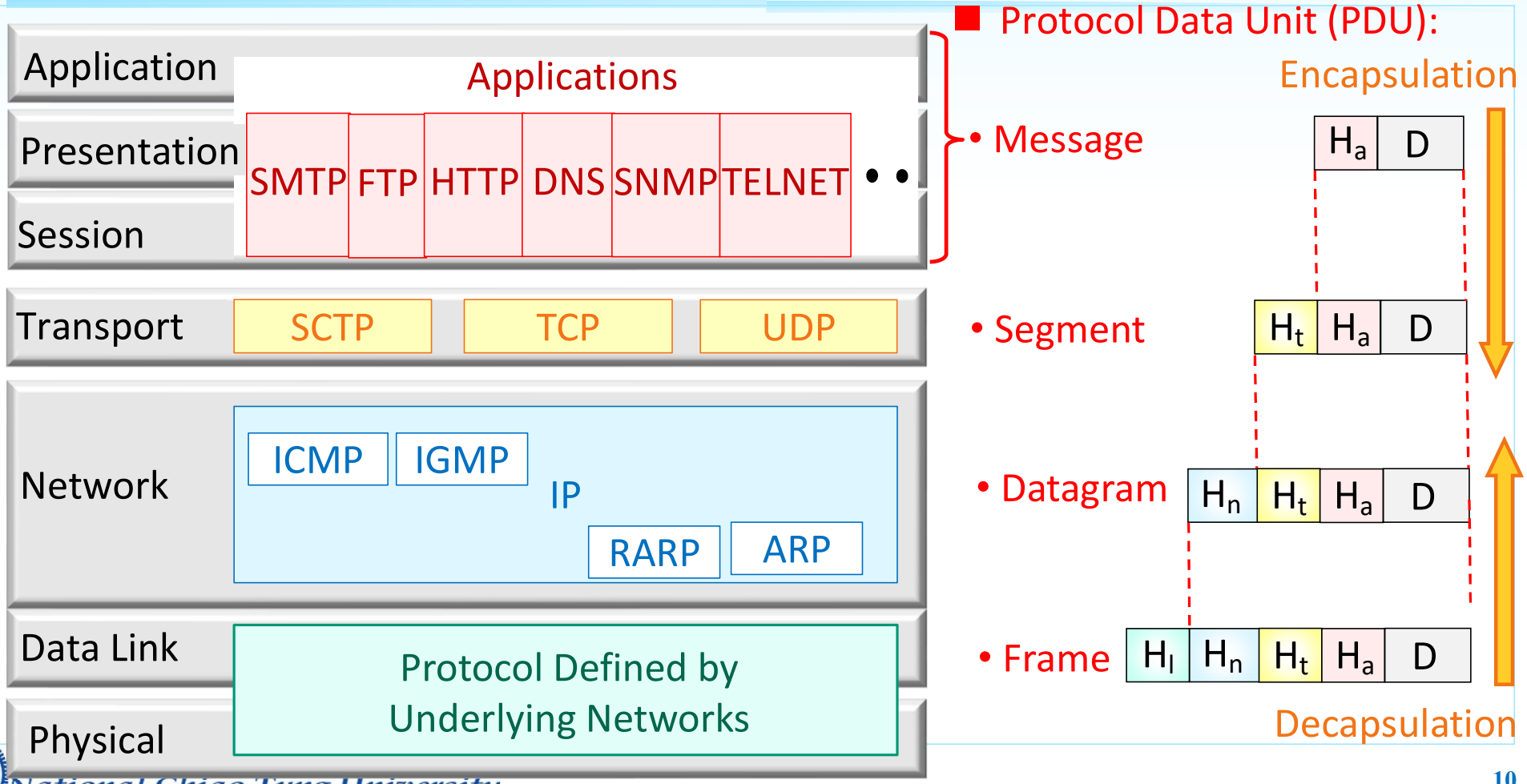


How Application Works – Peer to Peer

- **Peer-to-Peer Applications**
  - Host connects/disconnects dynamically
  - Hosts establish data trunk with the assistance of server



TCP/IP Protocol Stack and PDU

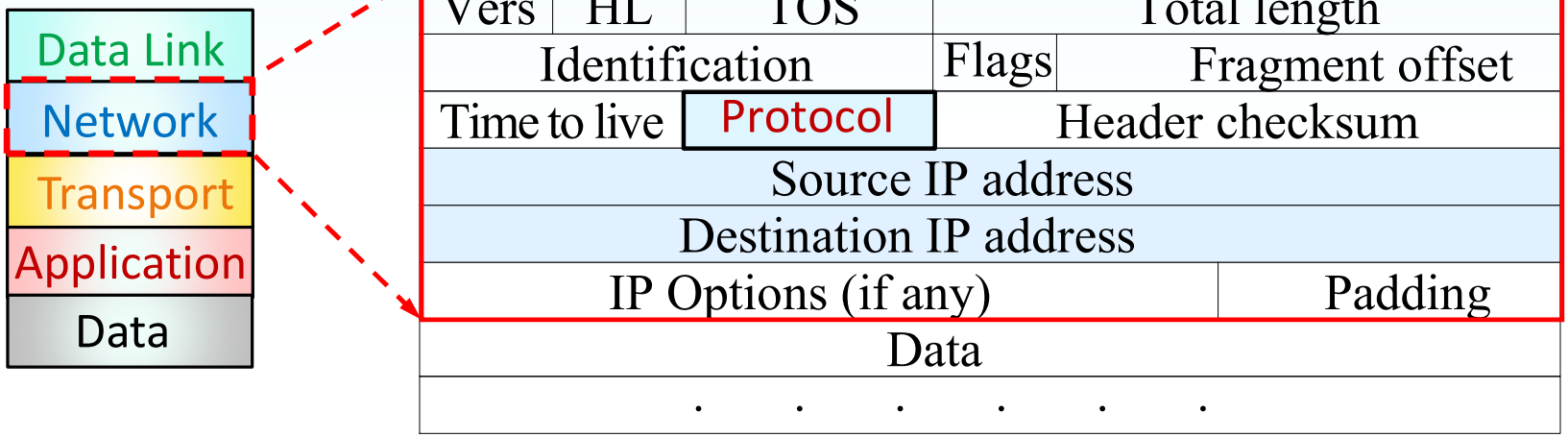


Example Protocols of TCP/IP

- SMTP: Simple Mail Transfer Protocol
- FTP: File Transfer Protocol
- HTTP: Hypertext Transfer Protocol
- DNS: Domain Name System
- SNMP: Simple Network Management Protocol
- SCTP: Stream Control Transmission Protocol
- TCP: Transmission Control Protocol
- UDP: User Datagram Protocol
- ICMP: Internet Control Message Protocol
- IGMP: Interne Group Management Protocol
- IP: Internet Protocol
- ARP: Address Resolution Protocol
- RARP: Reversed Address Resolution Protocol

IP Dtagram Header

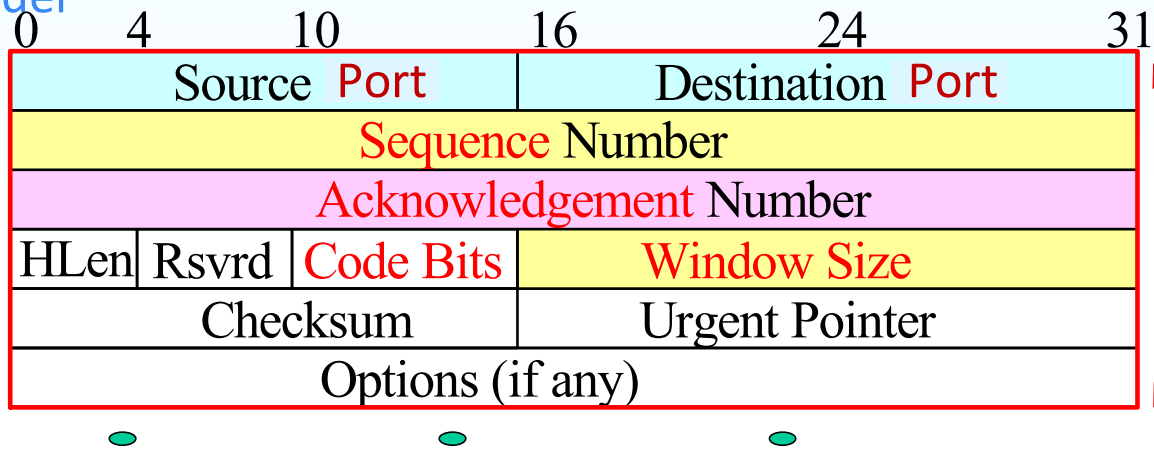
❖ IP Datagram Format

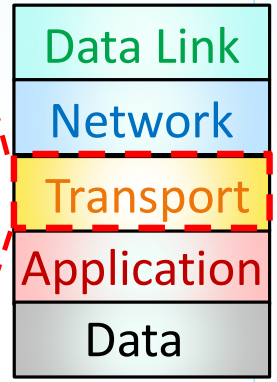


- Total Length: 16 bits
  - limits datagram to at most 65535 bytes
- Protocol:
  - 1 Internet Control Message Protocol (ICMP)
  - 6 Transmission Control Protocol (TCP)
  - 17 User Datagram Protocol (UDP)

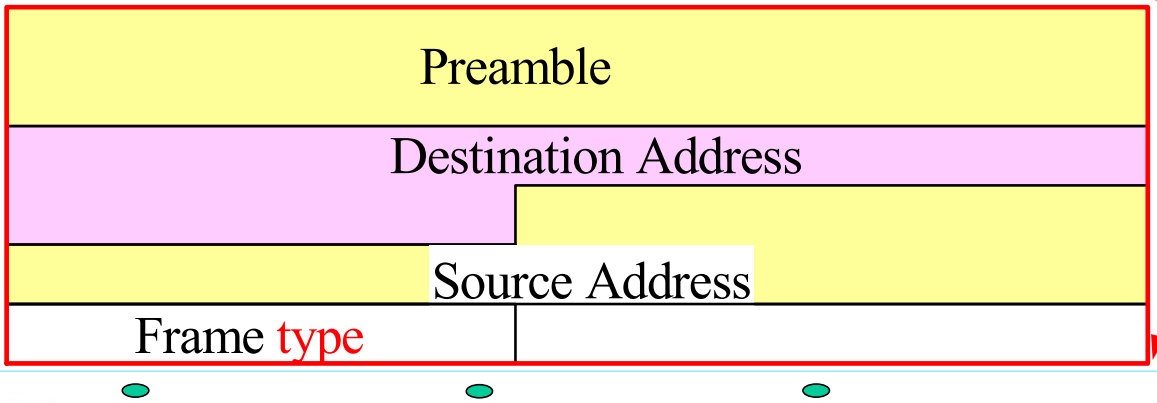
Ethernet Frame and TCP Segment Headers

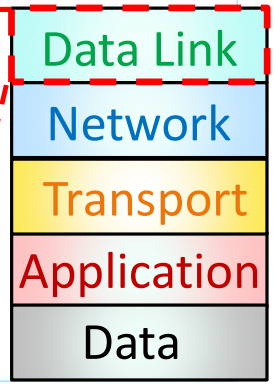
- TCP Segment Header
  - 80: HTTP
  - 443: HTTPS
  - 21: FTP Control
  - 20: FTP Data





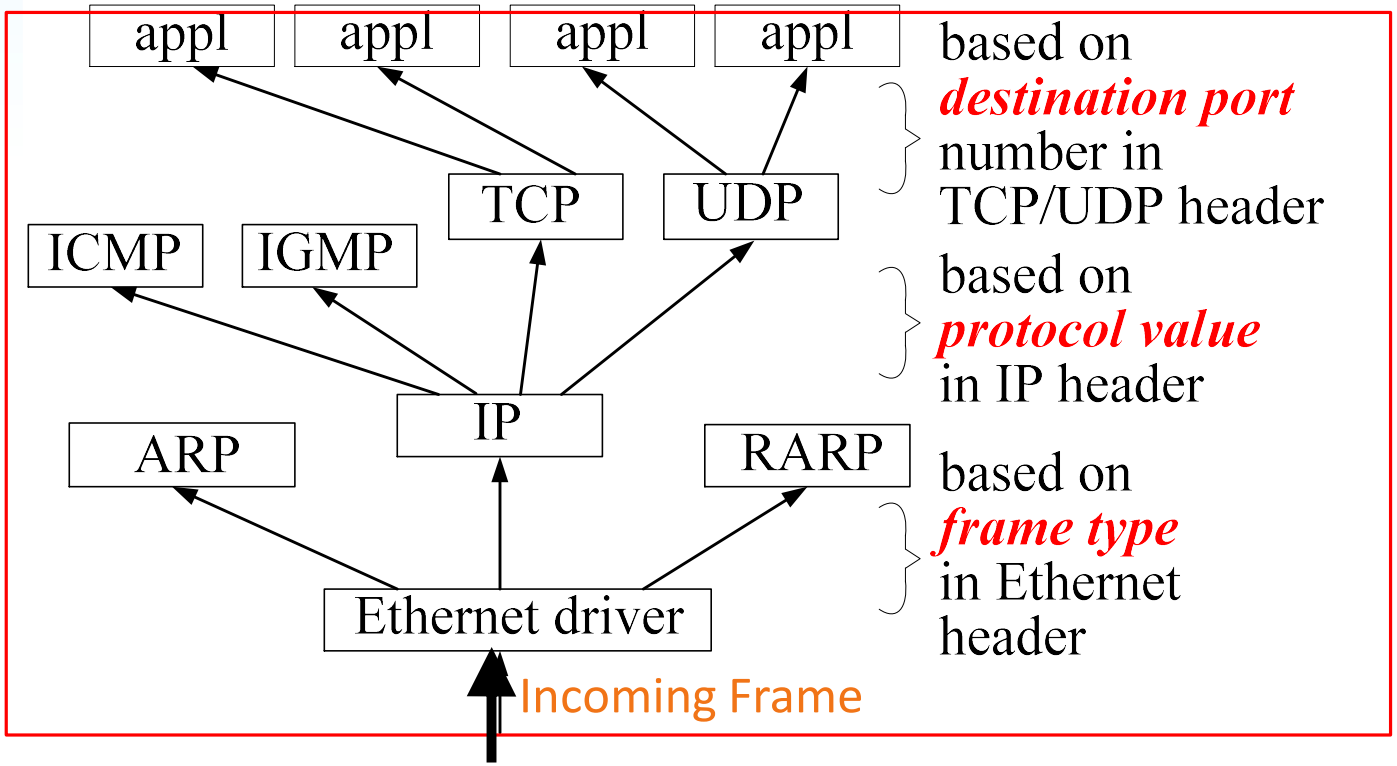
- Ethernet Frame Header
  - 0806_{16}: ARP
  - 0800_{16}: IP





Incoming Packet Demultiplexing

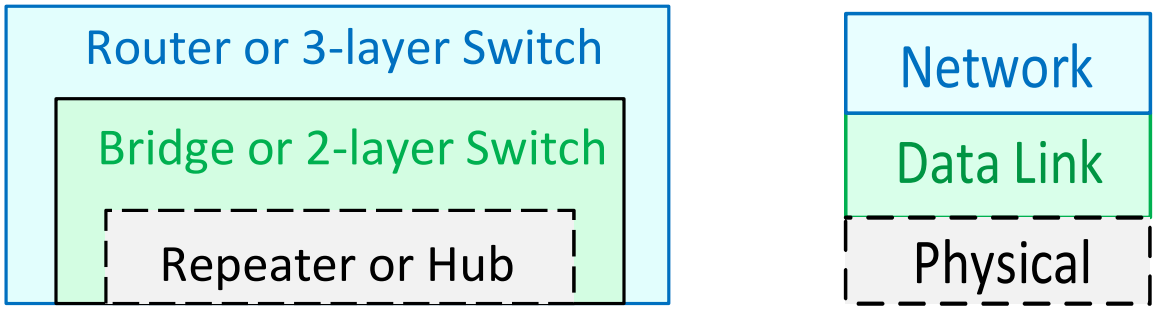
- Demultiplexing a received Ethernet frame (packet)



- Decapsulate packet as it goes up the protocol stack

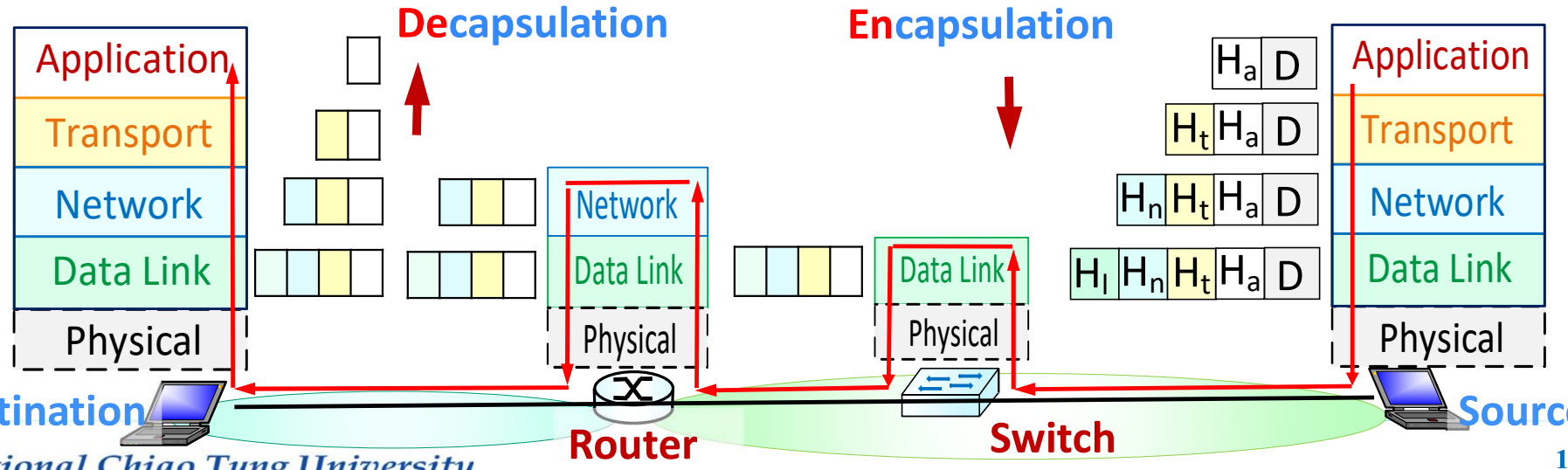
Connecting Devices

- Networks connect with one another via connecting devices.
- Connecting devices can operate in different layers.
- Three kinds of connecting devices:
  - Repeaters (or hubs; one-layer switches),
  - Bridges (or two-layer switches), and
  - Routers (or three-layer switches).



Switches vs. Routers

- Both are store-and-forward:
  - *Routers:* NW-layer devices, examine NW-layer headers
  - *Switches:* Link-layer devices, examine link-layer headers
- Both have forwarding tables:
  - *Routers:* compute forwarding tables, using routing algorithms, IP addresses
  - *Switches:* learn forwarding table, using flooding, learning, MAC addresses

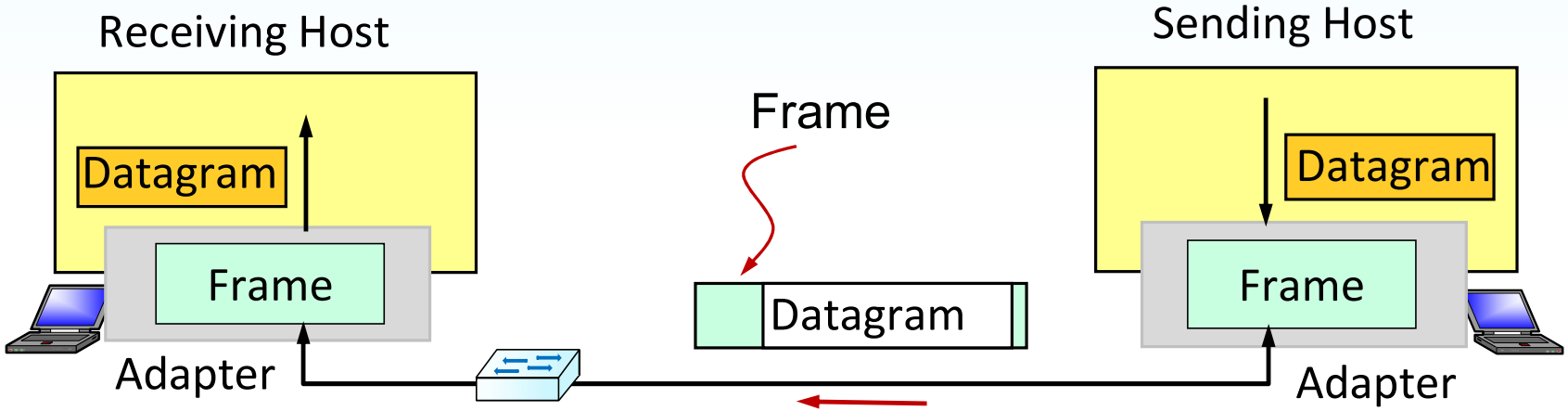


Outline

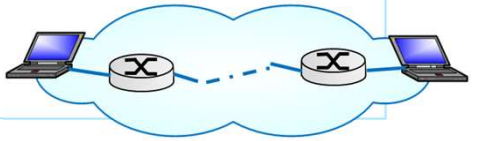
- Internet Protocol and Packet Multiplexing
- Layer 2 Switching
- IP Addressing and Forwarding
- ARP, DHCP, and ICMP

Link Layer Adaptors Communication

- Two hosts on same LAN: can physically communicate with each other directly



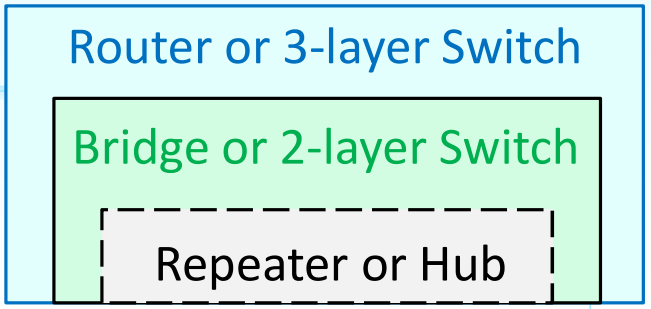
- Receiving Side
  - Looks for errors, flow control, etc
  - Extracts datagram, passes to upper layer
- Sending Side:
  - Encapsulates datagram in frame
  - Adds error checking bits, flow control, etc.
- Two hosts on different LANs: need IP layer routing and forwarding



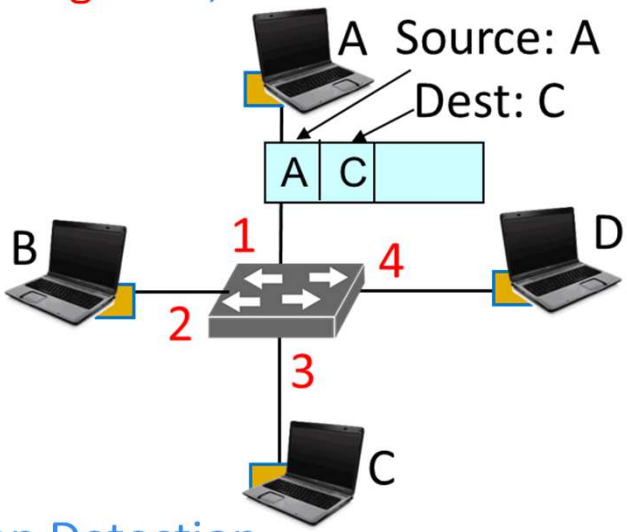
MAC Addresses

- Network Interface Card (NIC) (aka Network/LAN Adapter, or Physical NW Interface)
  - A computer hardware component that connects a computer to a computer NW.
- Each NIC has a Media Access Control (MAC) Address
- MAC (aka LAN, Physical, or Ethernet) address:
  - Used locally to deliver frame *from one interface to another physically-connected interface* (same network)
  - 48 bits MAC address (for most LANs)
    - Administered by IEEE
      - Manufacturer buys portion of MAC address space (to assure uniqueness)
    - Burned in NIC ROM, also sometimes software settable
  - Locally unique on a LAN
  - Flat address
    - Portable: can move NIC from one LAN to another

Ethernet Switch

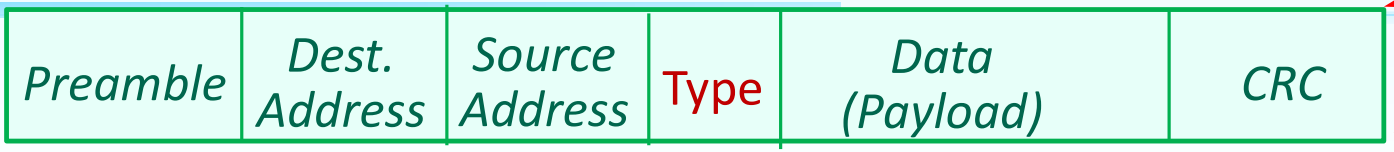


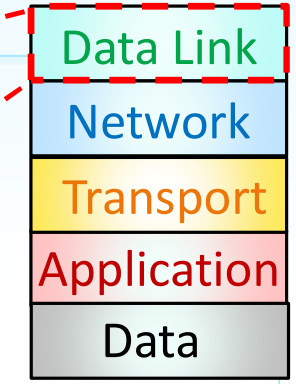
- A Link-layer Device:
  takes an *active* role to Store, forward Ethernet frames
  – Examine incoming frame MAC address,
    - Selectively forward frame to one-or-more outgoing links
    - When attempt to forward frame on an Ethernet segment, uses CSMA/CD to access the segment
- *Transparent*
  hosts are unaware of presence of switches
- *Plug-and-play, Self-learning*
  switches do not need to be configured



- CSMA/CD: Carrier Sense Multiple Access with Collision Detection

Ethernet Frame Structure





- Preamble: 7 bytes, to synchronize receiver, sender clock rates
- Addresses: 6 bytes
  - If adapter receives frame with
    - Matching Destination Address,
    - Broadcast Address (e.g., ARP packet), or
    - Participating Multicast Address

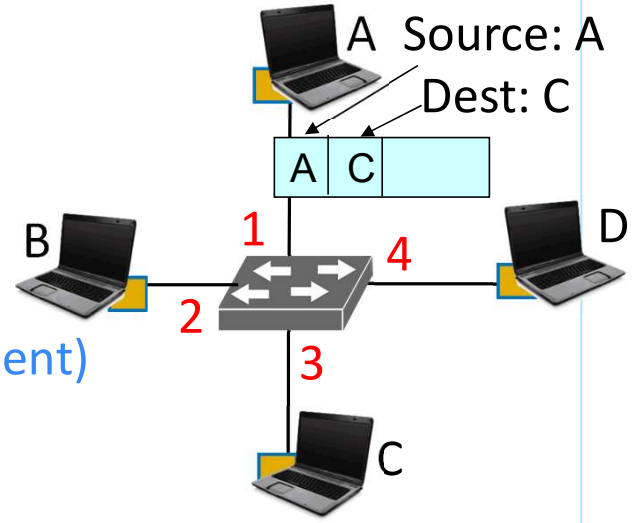
    it passes data in frame to network layer protocol
  - Otherwise, adapter discards frame
- Type: indicates higher layer protocol
  - Mostly IP but others possible (e.g., Novell IPX, AppleTalk)
    - 0x0800: IPv4, 0x0806: ARP
- CRC: checked at receiver, frame dropped if error detected

Switch Table and Self-Learning

MAC Addr	Interface	TTL

- Each switch has a switch table
- **Table entry:**
  - MAC Address of Host, Interface to Reach Host, Time Stamp
  - Looks like a routing table
- How does switch create and maintain table entries?
- **MAC Learning**

  Self-learning which hosts can be reached through which interfaces
  - When switch receives a frame, it
    - Learns location (port) of sender (incoming LAN segment)
    - Records Sender-MAC/Port pair in switch table

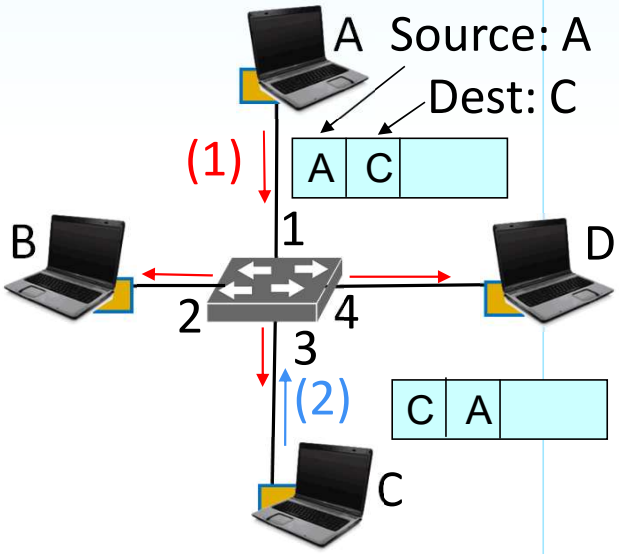


Flooding, Forwarding and Self-learning (MAC Learning)

- Assume switch table is initially empty

1) A→C:
Destination C location unknown
   - Flood
   - MAC Learning
     - Update switch table

MAC Addr	Interface	TTL
A	*1*	*60*



2) C → A:
Destination A location known
   - Unicast (Selectively send on just one link)
   - MAC Learning
     - Update switch table

MAC Addr	Interface	TTL
A *C*	*1* *3*	*60* *60*

Frame Filtering/Forwarding Algorithm

Skip

- Algorithm

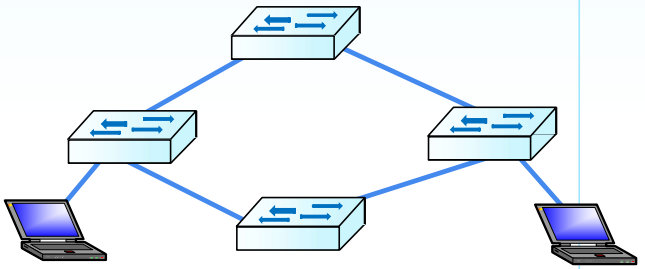
```
when  frame received at switch:
1. record incoming link, MAC address of sending host
2. index switch table using MAC destination address
3. if entry found for destination
    then {
        if destination on segment from which frame arrived
            then drop frame
            else forward frame on interface indicated by entry
    }
    else flood  /* forward on all interfaces except arriving
                   interface */
```

Redundancy at OSI Layers 1 and 2

- Multiple cabled paths provide physical redundancy
- Alternate physical paths to traverse network
- Improves reliability and availability
  - Possible to access network resources, despite path disruption

Primary Issues of Layer 2 Loops

- **Physical redundant paths** may cause **logical Layer 2 loops**
  - Due to flooding and MAC learning



1. **Broadcast storm**—
   Each switch may flood broadcasts endlessly
2. **MAC Database (Address Table) Instability**—
   Switch receives copies of same frame on **different ports**.
   - Update switch tables continuously
3. **Multiple-frame Transmission**—
   Destination station may receive multiple copies of unicast frames

➢ Solution: Spanning Tree Protocol (STP)

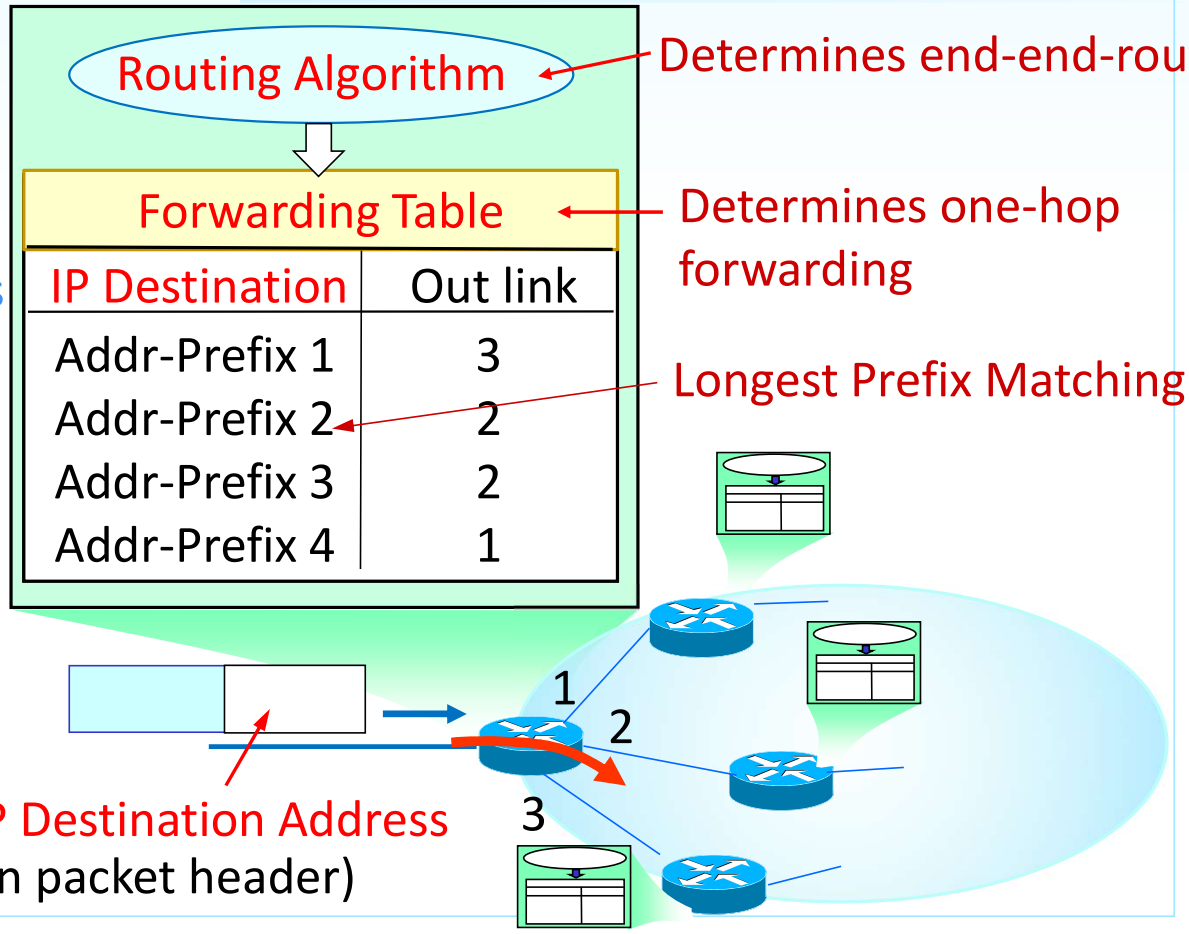
- Reference: https://www.youtube.com/watch?v=japdEY1UKe4
  - Spanning Tree Protocol Explained | Step by Step, by CertBros

Outline

- Internet Protocol and Packet Multiplexing
- Layer 2 Switching
- IP Addressing and Forwarding
- ARP, DHCP, and ICMP

Two Main Functions of IP Layer

- Routing and Forwarding
  - *Routing:* determines source-destination route for packets
  - *Forwarding:* moves packets from input to appropriate output
    - Based on IP Destination Addresses



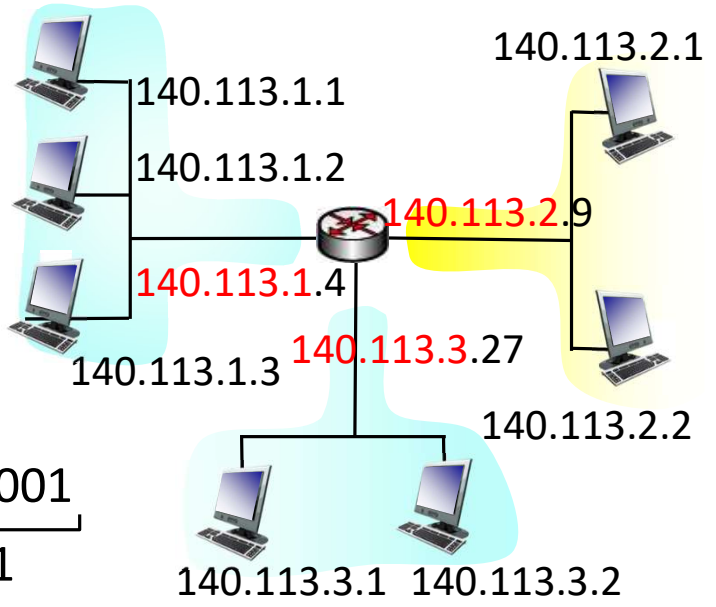
Internet Protocol and Addressing

- Internet Protocol (IPI):

  Connectionless, no FIFO ordering, and no error, loss or duplication detection
- IP address: 32-bit identifier for host, router interface
- Interface: connection between host/router and physical link
  - Host typically has one or two interfaces
  - Routers typically have multiple interfaces
    - One interface for each subnet (at least)
    - One IP for each interface (at least)
- IP addresses associated with each interface
- Dotted Decimal Notation:

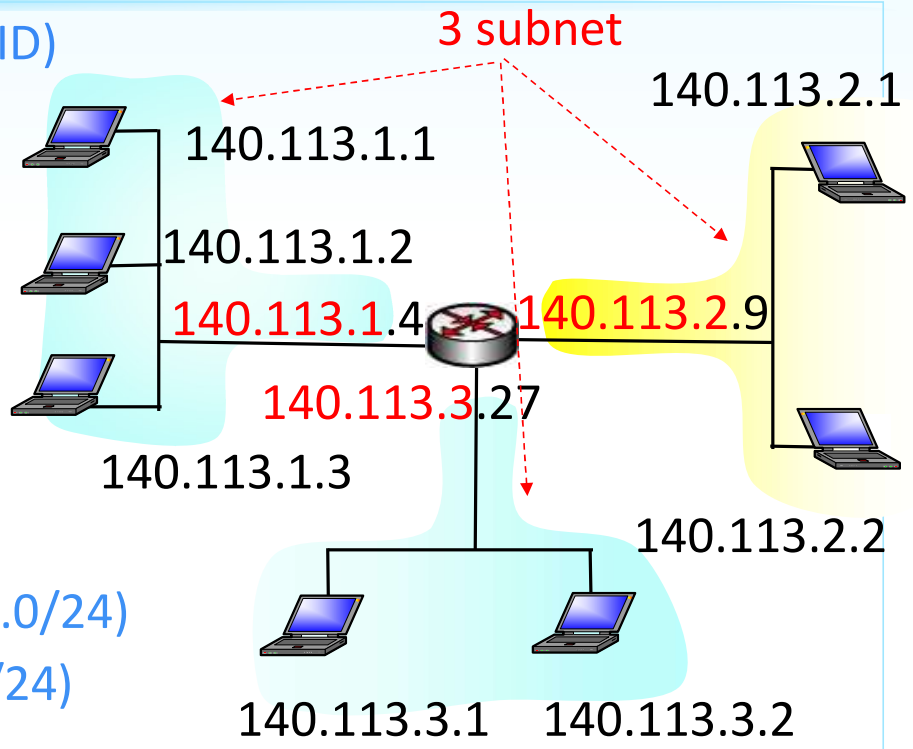
140.113.36.1 = 10001100 01110001 00101000 00000001

140 113 36 1



Sub-Networks (Subnets)

- IP address has two parts (Network ID, Host ID)
  - High order bits: Network (or Subnet) ID
  - Low order bits: Host ID
- Characteristics of Subnets
  - Device interfaces with same subnet ID
    - Can physically reach each other
      - without intervening router
- Two special IP addresses (in each network)
  - First IP: Network number (e.g., 140.113.1.0/24)
  - Last IP: Broadcast IP (e.g., 140.113.1.255/24)

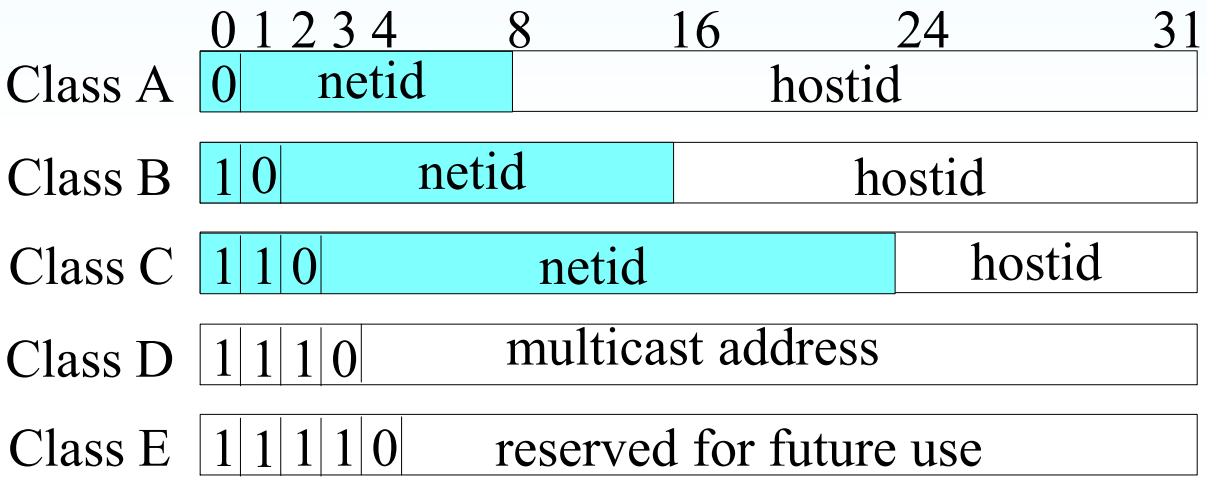


✓ Subnet Directed Broadcast Address: 140.113.255.255 (for 140.113.0.0/16)

✓ Broadcast Address: 255.255.255.255

Classful IP Addressing

- 32 bit IP address is divided into five sub-classes
- Five Classes



- Uses Network ID for packet forwarding
  - Class determines the length of Network ID
- Number of Class A addresses: 224-2
- Number of Class B addresses: 2^{16}-2
- ➢ Waste IP addresses, difficult to manage

Classless Inter-Domain Routing (CIRD)

- Subnetting
  partitioning a single network into more than one smaller sub-networks (subnets).

* E.g., 140.113.36.0: network with 24 bits NWID, May partition into
  - Two networks: with 25 bits NWID

NW ID	Host ID
140.113.36.0	**0000000** (140.113.36.0)
140.113.36.1	0000000 (140.113.36.128)

  - Four networks: with 26 bits NWID

NW ID	Host ID
140.113.36.00	**000000 (0)**
140.113.36.01	000000 (64)
140.113.36.10	000000 (128)
140.113.36.11	000000 (192)

➢ Classless Inter-Domain Routing (CIDR)
  – Arbitrary length of subnet portion
  – Format of addresses: a.b.c.d/x (x: number of bits in subnet portion)

* E.g., NW ID of 140.113.36.84?
  – /24: 140.113.36.0 (10001100 01110001 00010100 01010100)
  – /26: 140.113.36.64 (10001100 01110001 00010100 01010100)

Subnet Mask

- Subnet Mask (or netmask):
  bitmask used to determine NW ID of an IP address
- 140.113.36.84:

10001100 01110001 00010100 01010100

  - 140.113.36.84/24:
    - netmask *255.255.255.0*

NW ID | Host ID
11111111 11111111 11111111 | 00000000

  - 140.113.36.84/26:
    - netmask *255.255.255.192*

NW ID | Host ID
11111111 11111111 11111111 11 | 000000

- **IP** bitwise-ANDed with netmask
  - network address (NW ID) of 140.113.36.84
  - /24: 10001100 01110001 00010100 01010100 (140.113.36.0)
  - /26: 10001100 01110001 00010100 01010100 (140.113.36.64)

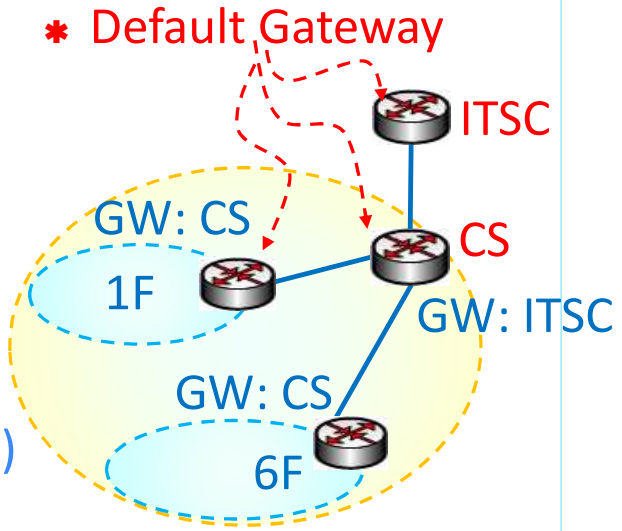
Techniques for Routing Table Entries

- Every host or router has Routing Tables (RTs)
- Techniques that make the size of RT **manageable**,

1) Next-hop Routing: RT holds only addr of next hop,
   - instead of complete route
2) Network-Specific Routing: entry for Dest. NW (NW ID)
   - Instead of Dest. Host addresses (Host ID)
3) Default Routing: one entry for Rest of Internet
   - Default Gateway (for Rest of Internet)

- Techniques that handles other issues such as security

4) Host-Specific Routing: entry for Dest. Host address
   - **Efficiency is sacrificed** for other advantages
     - E.g., **more control** over routing (for security or others)
   - An extreme case of **more specific route**



Longest Prefix Matching– More Specific Routes

- **Longest prefix matching:**
  use the entry with the *longest* address prefix that matches the destination address if more than one entry matches

Destination Address Range	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

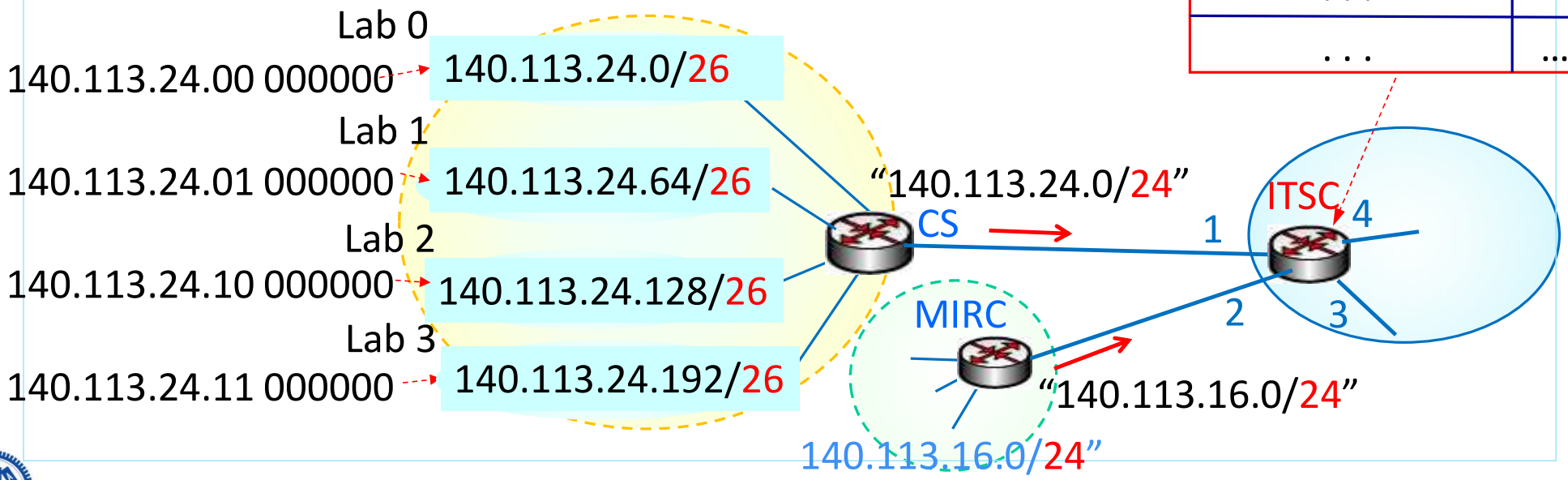
- Examples: (Logest Prefix Maching)
  1) DA: 11001000 00010111 00010110 10100001
     - ✓ Matches 1st entry, forwarded via interface 0
  2) DA: 11001000 00010111 00011000 10101010
     - ✓ Matches 2nd and 3rd entries, forwarded via interface 1

Scenario

Why More Specific Route: Route Aggregation

- Hierarchical Addressing allows efficient advertisement of routing information:
  - **Route Aggregation**
- E.g., Given a network of 140.113.24.0/24
  - Can divide it into four subnet of /26
  - But router CS advise 140.113.24.0/24, instead

Dst. Addr. Range	IF
140.113.24.0/24	1
140.113.16.0/23	2
. . .	…
. . .	…



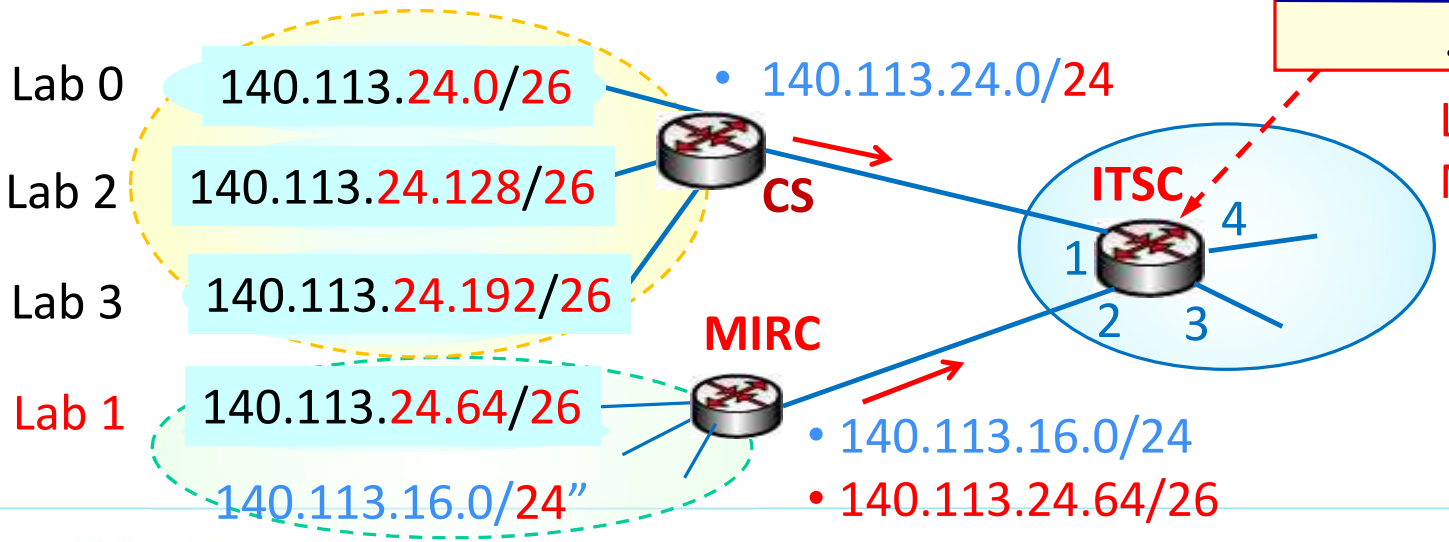
Example of Longest Prefix Matching

★ Ex., Lab 1 moves to MIRC, but retains original addresses
- MIRC advises more specific route to 140.113.24.64/26
  - ➢ITSC RT entries: 140.113.24.0/24, 140.113.24.64/26
- Longest Prefix Matching: 140.113.24.100 matches both
  - ➢ITSC forwards packet to MIRC (via interface 2)

ITSC RT Entries

Dst. Addr. Range	IF
140.113.24.64/26	1
140.113.24.0/24	2
140.113.16.0/24	…
. . .	…

Longest Prefix Matching



Back

IP Datagram Forwarding

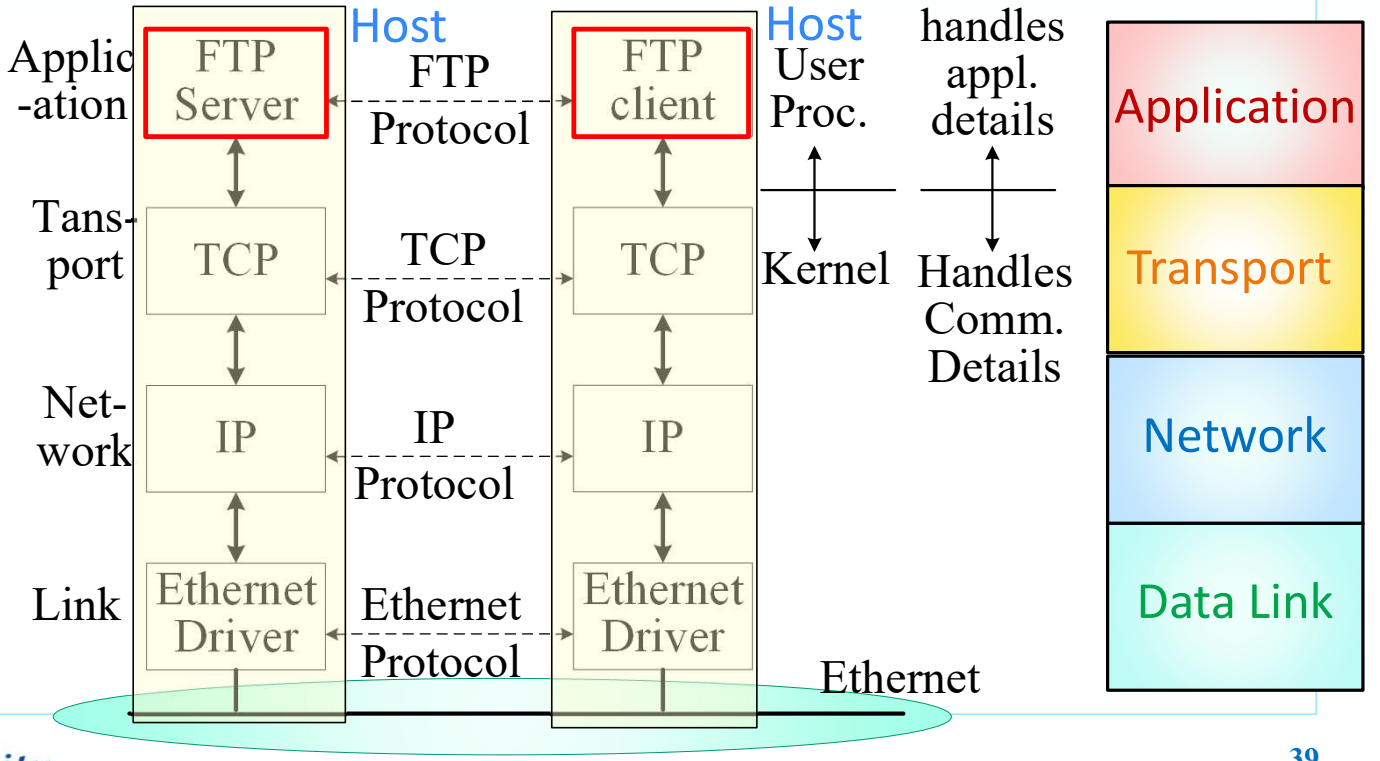
- Consult forwarding (routing) table to determine how to route an IP datagram
- **Direct Forwarding:** destination on the same physical network
  1. Finds *physical* address of destination host (ARP)
  2. Encapsulates *datagram* in a physical *frame* and
  3. Sends frame directly to ***physical*** (***MAC***) address of destination host
- **Indirect Forwarding:** destination not on a directly attached network
  1. Finds *physical* address of **next-hop router** (ARP)
  2. Encapsulates datagram in physical *frame* and
  3. Sends frame to ***physical*** (***MAC***) address of **next-hop router**
  - Forwards IP datagrams hop-by-hop until can deliver datagrams directly.

Note:

- ARP: address resolution protocol
- MAC: Media Access Control
- MAC Address:
  - Address of Network Interface Card
  - Ethernet: 48 bits MAC address

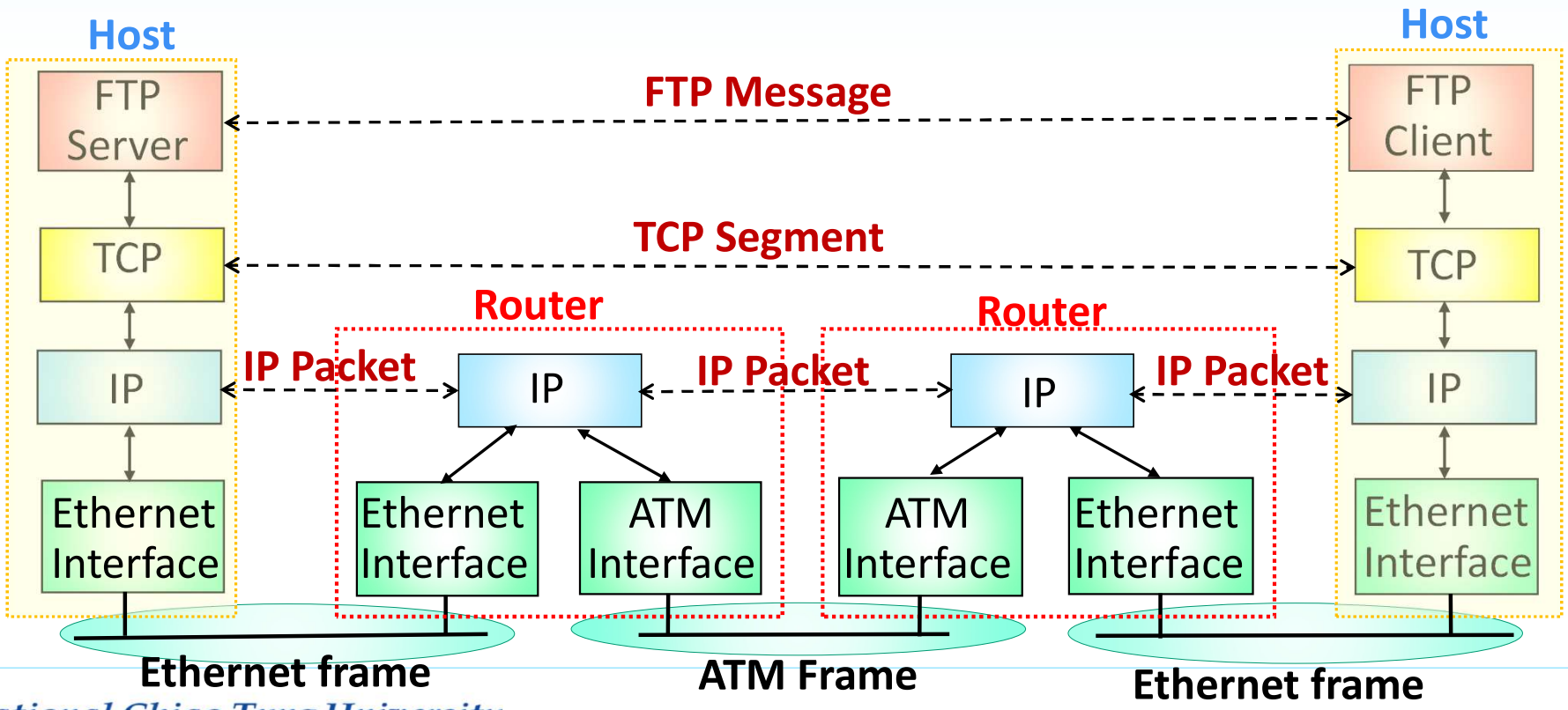
Data Forwarding within a LAN

- Two hosts on a LAN running FTP
- E.g., FTP client initiates FTP request
  - Client host deliver frames directly to server host



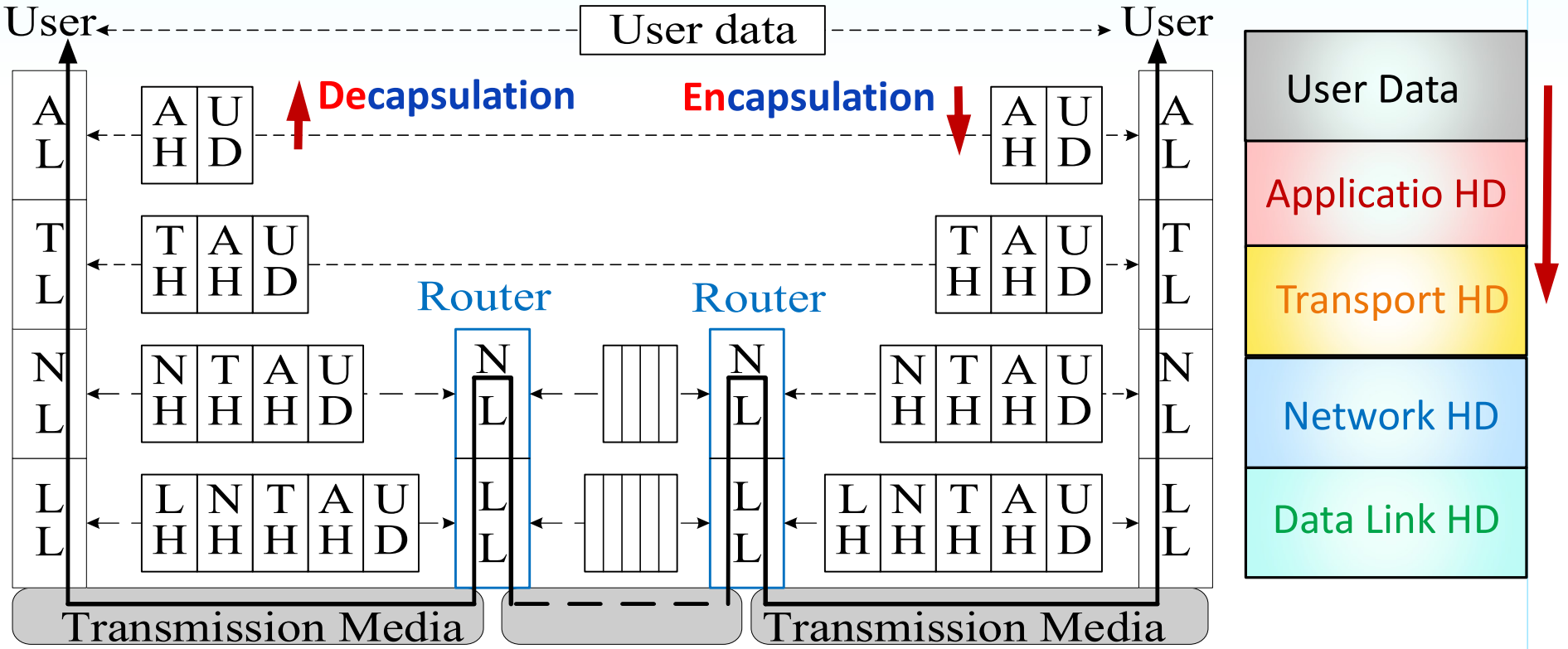
Data Forwarding across Networks

- Networks connected with Routers
- Router has one interface for each networks ➢ Hop-by-hop data delivery



Hop-by-hop Packet Forwarding

- Hop-by-Hop Forwarding and Encapsulation/Decapsulation



Encapsulation and Decapsulation

- Encapsulate packet when packet goes down the protocol stack
- De-capsulate packet when packet goes up the protocol stack

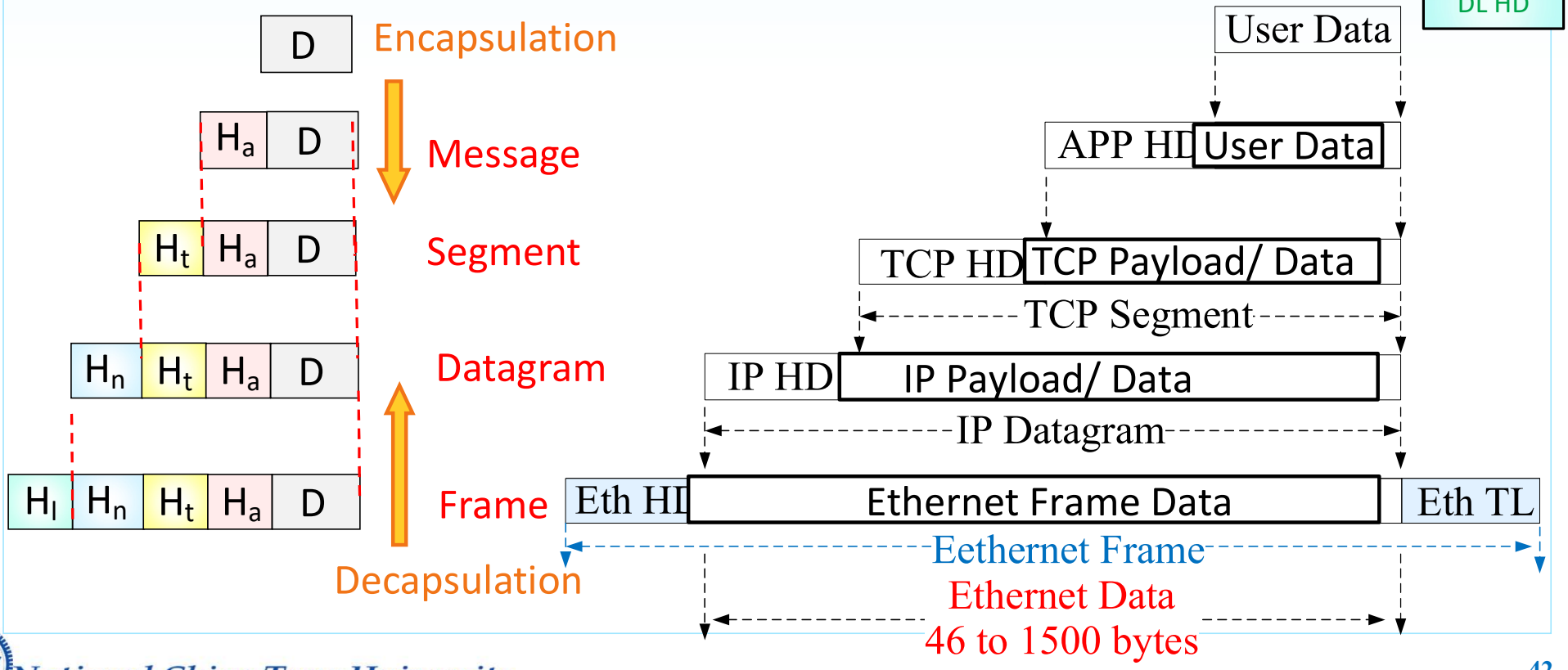
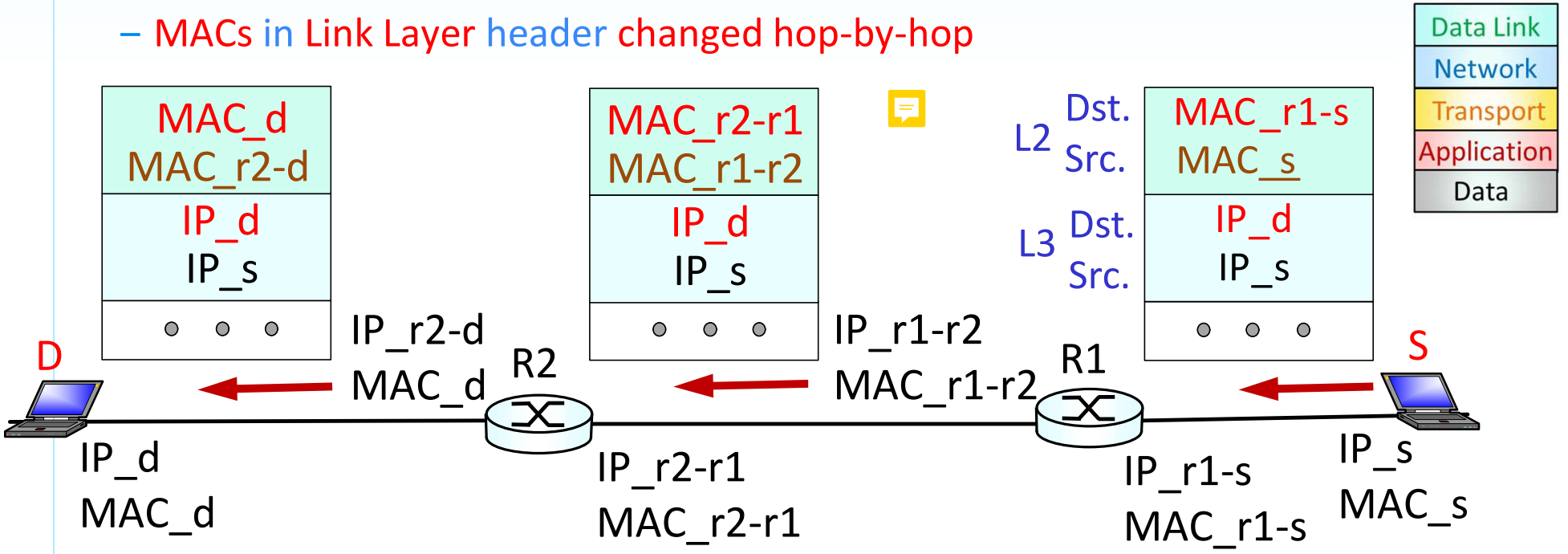


Illustration of Hop-by-Hop Packet Forwarding

- Host S sends packet to host D
  - IPs in Network Layer header not changed
  - MACs in Link Layer header changed hop-by-hop

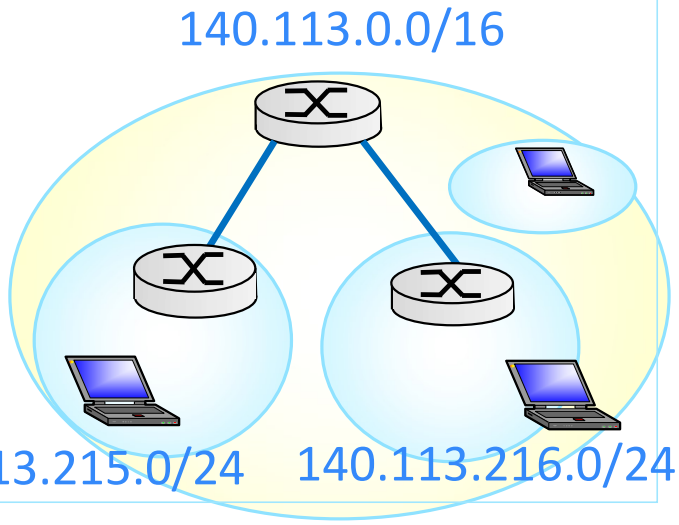
Convention Presentation of Internet Packet



✓Upper layer (Transport/Application) headers remain intact

Summary of IP Addresses

- IP Address (aka *network-layer* address):
  - Used to deliver datagram to destination host across Internet
  - 32-bit IP address
    - Associated with an adapter, set via DHCP or manually configure
    - Administered by Internet Corporation for Assigned Names and Numbers (ICANN)
  - Globally unique for public IPs,
  - Locally unique for private IPs
  - Hierarchical addresses
    - E.g., 140.113.0.0/16,
      - » 140.113.215.0/24, 140.113.216.0/24
    - ➢ Address of node depends on IP subnet attached
  - NOT Portable

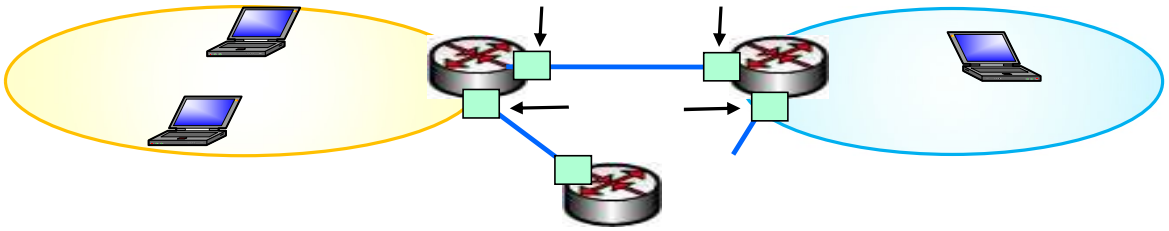


Outline

- Internet Protocol and Packet Multiplexing
- Layer 2 Switching
- IP Addressing and Forwarding
- ARP, DHCP, and ICMP

Address Resolution Protocol

- Address Resolution Protocol (ARP):
  mapping Internet addresses to physical (MAC) addresses
- Each IP node (host, router) has ARP table(s)
  - One table for each interface associated with a network (LAN)

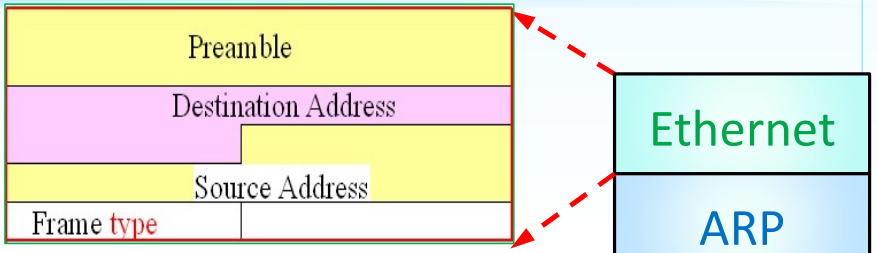


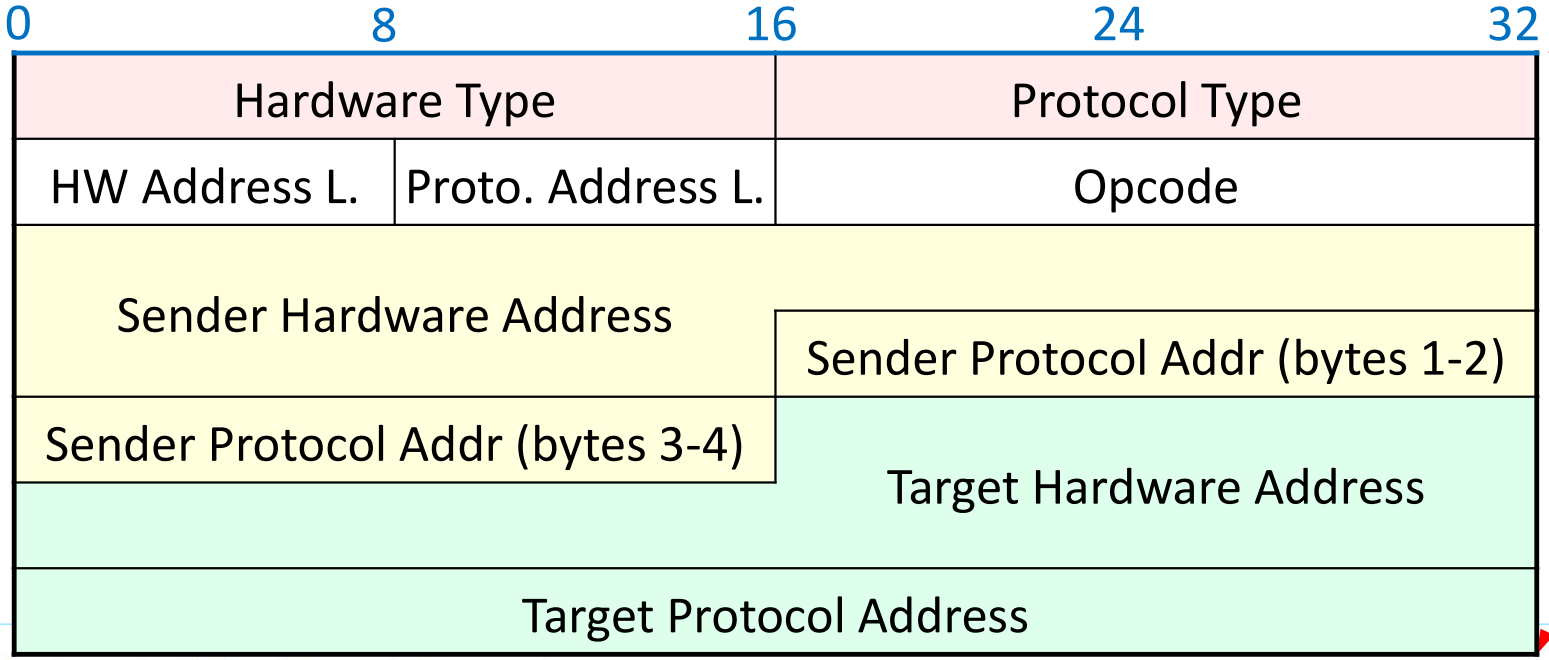
- ARP table: IP/MAC mappings (for some nodes on the LAN)

  < IP address; MAC address; TTL>
  - TTL (Time To Live):
    time after which address mapping will be forgotten (typically 20 min)
- Soft State:
  information that times out (goes away) unless refreshed

ARP Message Format

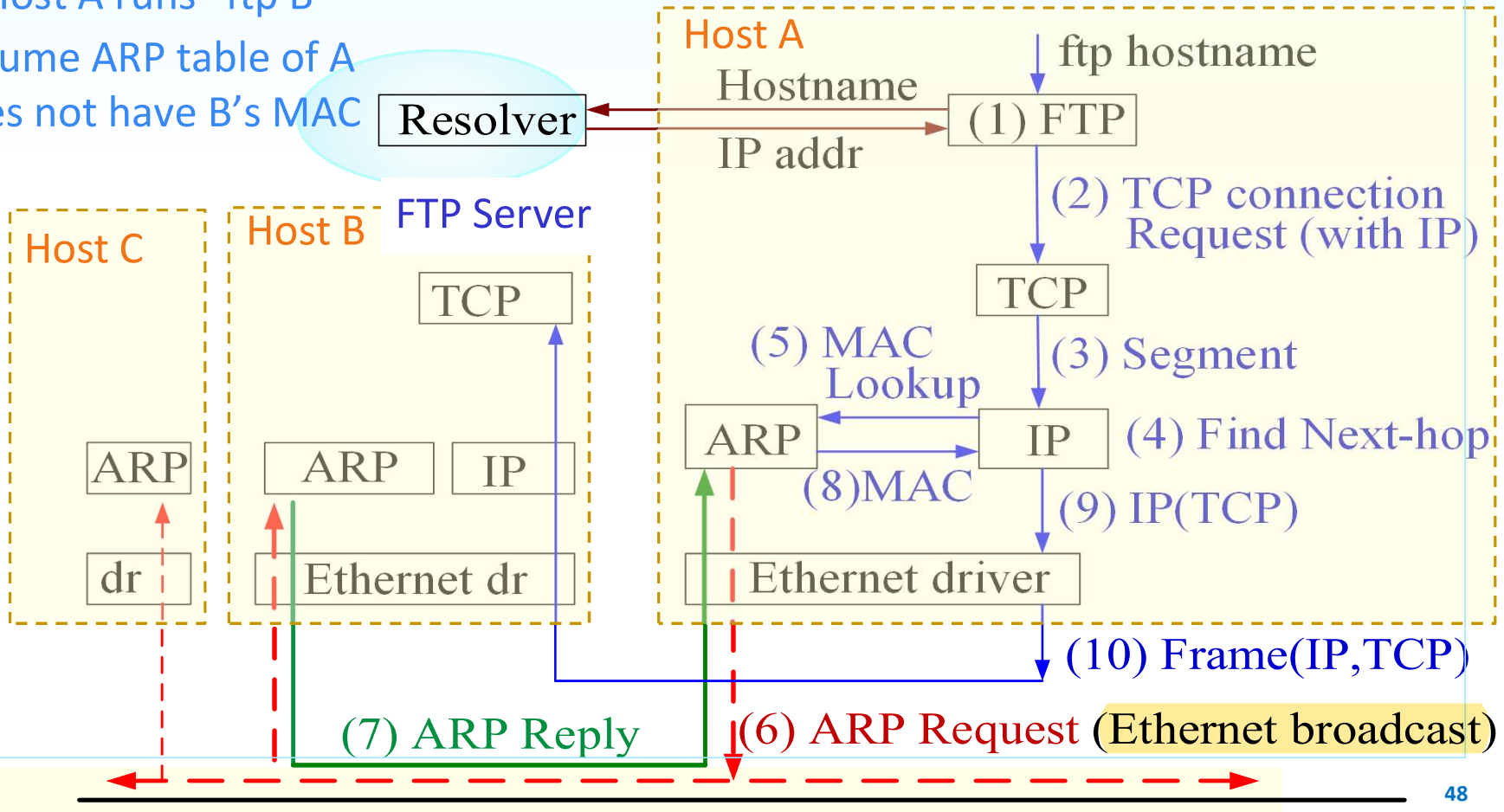
- Used to resolve IP to MAC address mapping
- Ethernet frame type:
  - 0806_{16} for ARP messages
- ARP Message Format:





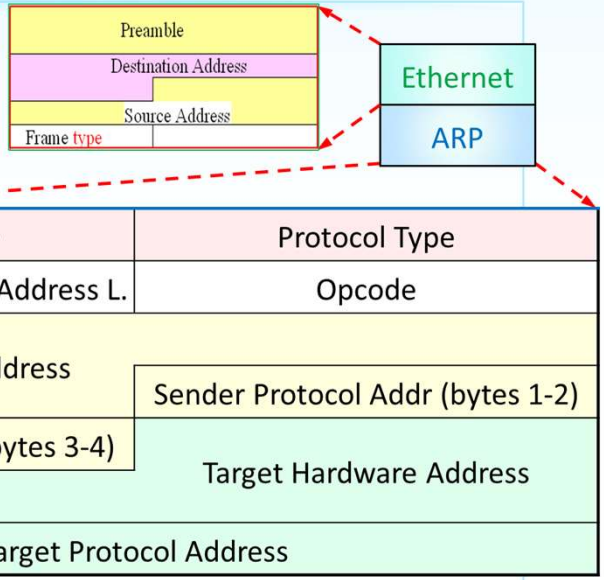
Procedure of ARP: Host A runs "ftp B"

- E.g., Host A runs "ftp B"
  - Assume ARP table of A does not have B's MAC



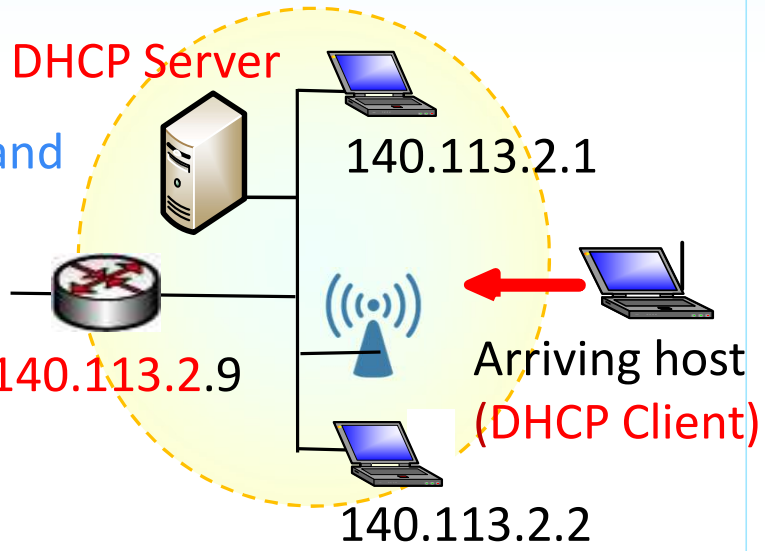
Procedure of ARP: Host A runs "ftp B" (cont.)

- ✓ Assume A's ARP table does not have B's MAC address
- A broadcasts ARP request, encapsulated in Ethernet frame
  - Ethernet Header: containing
    - Source MAC: A's MAC
    - Destination MAC: FF-FF-FF-FF-FF-FF
  - ARP Request message: containing
    - Sender A's MAC, A's IP address
    - Target B's MAC all 0s, B's IP address,
- All machines on LAN receive ARP Request
  - B Replies (unicast) to A with B's MAC address
  - ➢A caches (saves) IP-to-MAC address pair in ARP Table
- ✓ ARP is "plug-and-play":
  Create table automatically without intervention from administrator

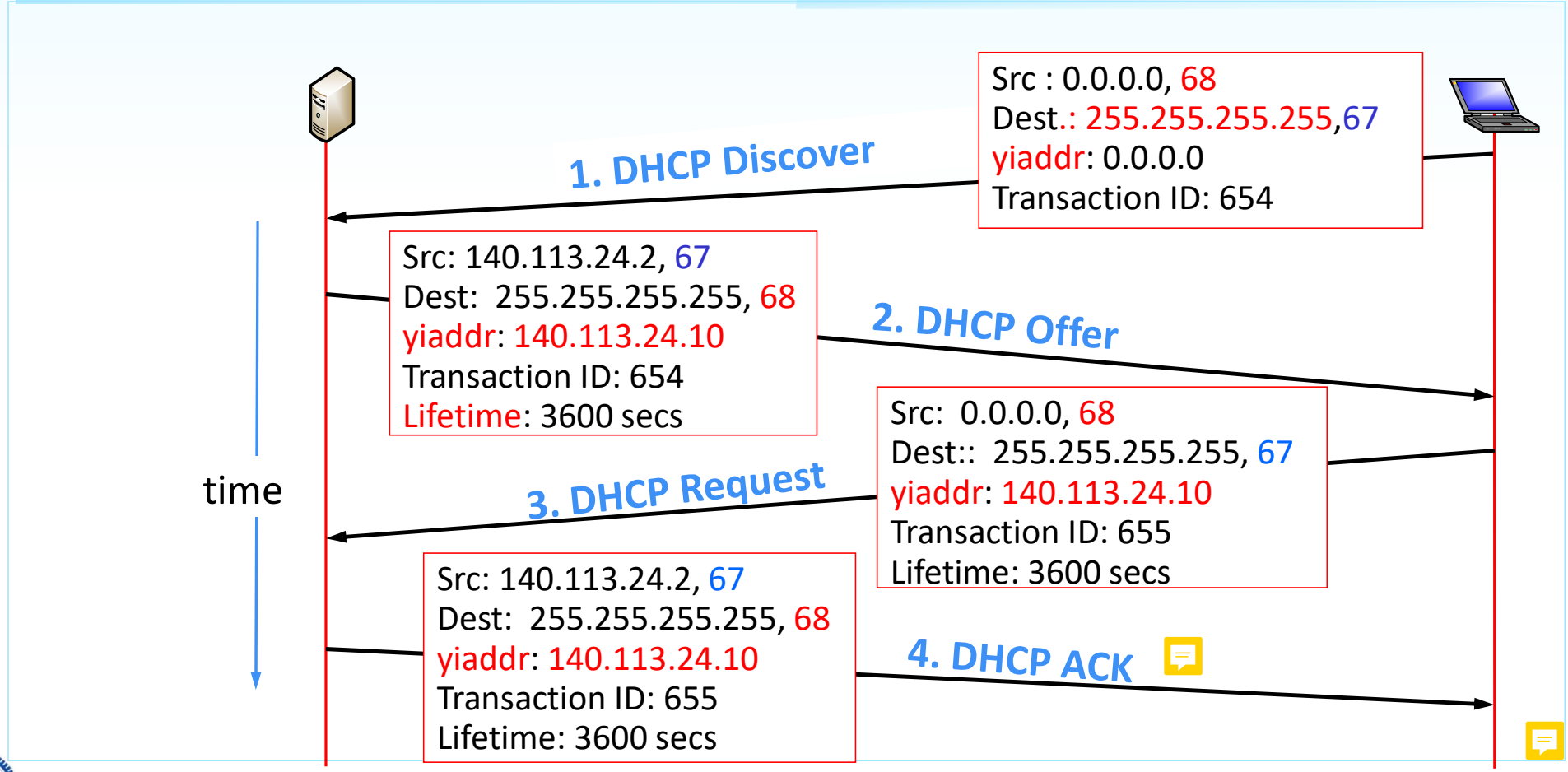


Dynamic Host Configuration Protocol (DHCP)

- Allow host to *dynamically* obtain IP address from DHCP server when joins network
- Host Can renew its lease on address in use
  - only hold address while "connected"
  - ➢Allows reuse of addresses
- Uses UDP with destination port 67 for the server and 68 for the client
- DHCP overview:
  1. Host broadcasts "DHCP Discover" msg
  2. DHCP server responds with "DHCP Offer" msg
  3. Host requests IP address: "DHCP Request" msg
  4. DHCP server sends address: "DHCP Ack" msg
- Recall: Local Broadcast Address: 255.255.255.255
  Subnet Directed Broadcast Address: 140.113.255.255 (for 140.113.0.0/16)

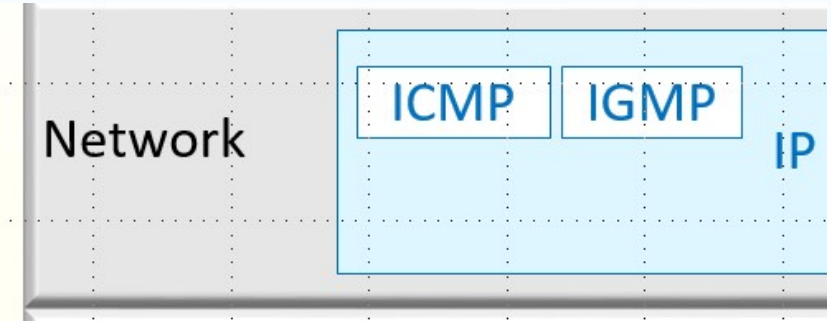


DHCP server: 140.113.24.2 DHCP Messages



Internet Control Message Protocol (ICMP)

- A TCP/IP network layer protocol companion to IP protocol



  - Defied in RFC 792
  - Provides troubleshooting, control and error message services.
  - Used by network devices to communicate network-level information
    - error reporting: unreachable host, network, port, protocol
    - echo request/reply (used by ping)

ICMP Message Format

- Encapsulated in an IP datagram

  - Protocol field =1

- ICMP Message Format

0 7	8 15	16 31
8-bit Type	8-bit Code	16-bit Checksum
Rest of Headers (4-bytes, contents depends on type and code)		
Data Section (varying, E.g., Timestamp, or IP header plus first 8 bytes of data)		

Some Types and Codes of ICMP Message

Type	Code	Description
0	0	echo reply (ping)
3	0	dest network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
5	0	redirect for Network
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery/selection/solicitation
11	0	TTL expired
12	0	bad IP header

Type 3: Destination Unreachable

Type 11: Time Exceeded

ICMP – Ping Program

- Use ICMP messages to find if a host is reachable (alive and responding)
  - Type 8, ICMP echo request
  - Type 0, ICMP echo reply
- **Format of ICMP** echo request/reply: Type: 0 or 8
  - **Code**: Additional context information
  - **Identifier:** process ID of the sending process
  - **Optional data:** any optional data sent must be echoed
- Sender inserts departure time in data section
  - ➢**Round-trip time = Returned time – Departure time**

Type (0 or 8)	**Code** (0)	**Checksum**
Identifier		Sequence number
Optional data		

Example of Ping

- Ping 8.8.8.8

```
C:\Users\Chien-Chao Tseng>ping 8.8.8.8

Ping 8.8.8.8 (使用 32 位元組的資料):
回覆自 8.8.8.8: 位元組=32 時間=10ms TTL=60
回覆自 8.8.8.8: 位元組=32 時間=10ms TTL=60
回覆自 8.8.8.8: 位元組=32 時間=9ms TTL=60
回覆自 8.8.8.8: 位元組=32 時間=10ms TTL=60

8.8.8.8 的 Ping 統計資料:
    封包: 已傳送 = 4，已收到 = 4，已遺失 = 0 (0% 遺失)，
大約的來回時間 (毫秒):
    最小值 = 9ms，最大值 = 10ms，平均 = 9ms
```

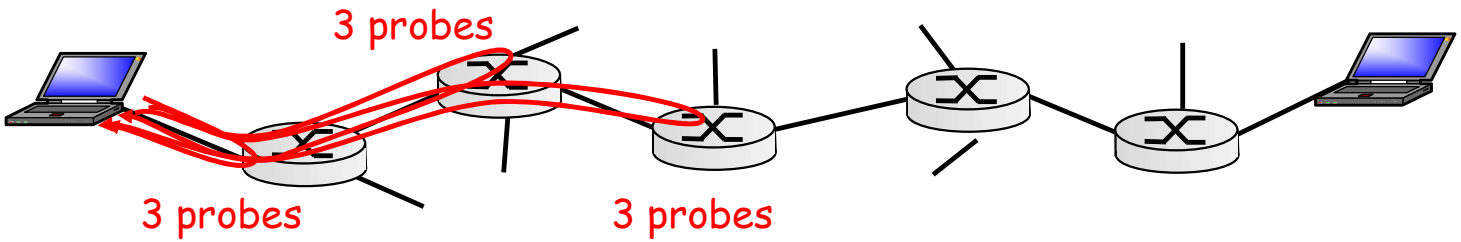
ICMP – Traceroute Program

- Used to trace the route packets take to the destination host
  - traceroute in Unix, tracert in Windows
- How **traceroute** works: (Leverage normal packet process of routers and host)
  - Routers: When receive a datagram
    - Decrement TTL by one
    - If TTL ≤ 0,
      - throws away the datagram and
      - sends back a "*Time exceeded*" ICMP message
    - Otherwise, **forward datagram** to the next-hop router or the destination host
  - Hosts: When receives a datagram,
    - If destination address matches,
      - demux datagram via *port*
      - If the *port* is not in-used, replies with a "*Port unreachable*" ICMP message

How traceroute works

- traceroute Program:
  - Start with TTL = *1*
  - Repeat {
    - Repeat 3 times
      - {Send a packet and wait for response}
    - Stop if receive responses from final destination
    - Otherwise, increase TTL}

- time-exceed or (from routers)
- destination-unreachable (from destination host)



➢ *N*-1 intermediate routers ⇨ 3*N* packets total

Time Exceed and Destination-Unreachable message

- Type 11: ICMP time-exceed Intermediate Routers:
  - Code 0: TTL expired

Type (11)	**Code** (0)	**Checksum**
Identifier		Sequence number
IP Header + First 8 bytes of datagram (UDP header)		

- Type 3: Destination-Unreachable
  - Code 3: Destination port unreachable

Type (3)	**Code** (3)	**Checksum**
Identifier		Sequence number
IP Header + First 8 bytes of datagram (UDP header)		

Router IPs in Traceroute

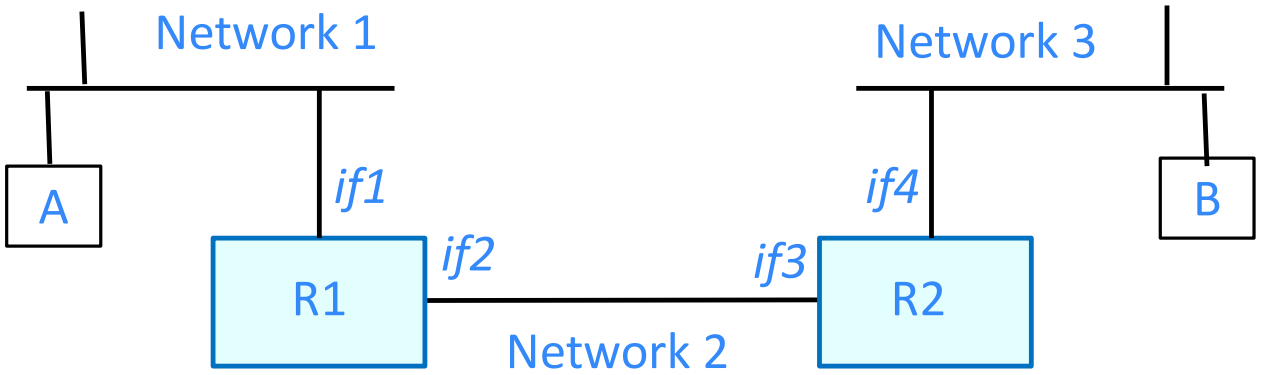
- Router IPs found in traceroute

  IPs of interfaces that receive the datagram.

  – IPs of input interfaces
- Traceroute from host A to host B

  – Result: if1, if3
- Traceroute from host B to host A

  – Result: if4, if2



traceroute: from NCTU to 8.8.8.8

- tracert in Windows

```
C:\Users\Chien-Chao Tseng>tracert 8.8.8.8

在上限 30 個躍點上
追蹤 dns.google [8.8.8.8] 的路由:

  1     8 ms     7 ms     7 ms  f45hc254.RAS.nctu.edu.tw [140.113.45.254]
  2     9 ms     8 ms     8 ms  192.168.211.2
  3     9 ms     8 ms     8 ms  not-a-legal-address [140.113.0.78]
  4    13 ms    10 ms    11 ms  140-113-254-10.Dorm-F2.NCTU.edu.tw [140.113.254.10]
  5    10 ms    11 ms    11 ms  216.239.48.209
  6    10 ms     9 ms     9 ms  72.14.238.17
  7    11 ms    10 ms    10 ms  dns.google [8.8.8.8]

追蹤完成。
```

Session Identification

- TCP Sessions on Host: Identified by Five Tuples

Protocol	Source IP	Source Port	Destination IP	Destination Port
TCP	192.168.0.193	53375	140.113.43.18	443
TCP	192.168.0.193	53509	140.113.43.18	443
TCP	192.168.0.193	55466	180.222.102.158	443
UDP	…	…	…	

- Application (Browser) provides Destination IP, Port
- OS provides Source IP, Port
  - assigns unique source port for each session