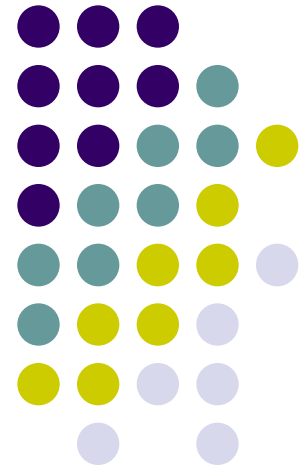
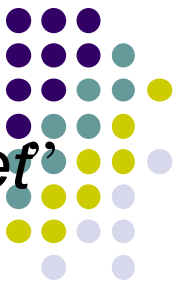


# TCP/IP Networks

## A Brief Introduction

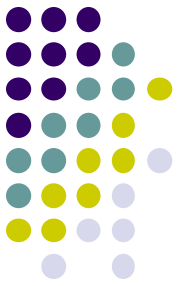


# Internet Evolution (I)



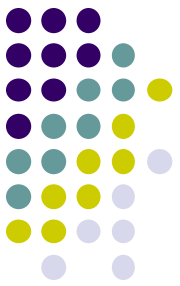
- In 1969, ARPA funded and created the “ARPAnet” network
  - Robust, reliable, vendor-independent data communications
- In 1975
  - Convert from experimental to operational network
  - TCP/IP begun to be developed
- In 1983
  - The TCP/IP is adopted as Military Standards
  - ARPnet → MILNET + ARPnet = Internet
- In 1985
  - The NSF created the NSFnet to connect to Internet
- In 1990
  - ARPA passed out of existence, and in 1995, the NSFnet became the primary Internet backbone network

# Internet Evolution (II)



- In 1995
  - IETF issued a specification for a next-generation IP
- In 1996
  - IPv6 is standardized
- In 1998
  - RFC2373 IP Version 6 Addressing Architecture
  - RFC2463 ICMPv6 for IPv6
- In 2003
  - RFC3513 IPv6 Addressing Architecture (replace RFC2373)
- In 2006
  - RFC4295 Mobile IPv6 Management Information Base
  - RFC4487 Mobile IPv6 and Firewalls: Problem Statement
- In 2008
  - RFC5380 Hierarchical Mobile IPv6(HMIPv6) Mobility management
- In 2009
  - RFC5568 Mobile IPv6 Fast Handovers

# General Introduction (1)



- TCP/IP

- Used to provide data communication between hosts
  - How to delivery data reliably
  - How to address remote host on the network
  - How to handle different type of hardware devices
- 4 layers architecture
  - Each layer perform certain tasks
  - Each layer only need to know how to pass data to adjacent layers

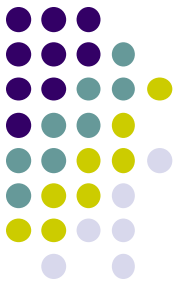
Application	Telnet, FTP, e-mail, etc.
Transport	TCP, UDP
Network	IP, ICMP, IGMP
Link	device driver and interface card

# General Introduction (2)



- Four layer architecture
  - Physical + Network Access Layer (or Interface)
    - Network Interface Card + Driver
    - Handle all the hardware details of whatever type of media
  - Network Layer (or say, Internet Layer)
    - Handle the movement of packets on the network
  - Transport Layer
    - Provide end-to-end data delivery services
  - Application Layer
    - Handle details of the particular application

# General Introduction (3)

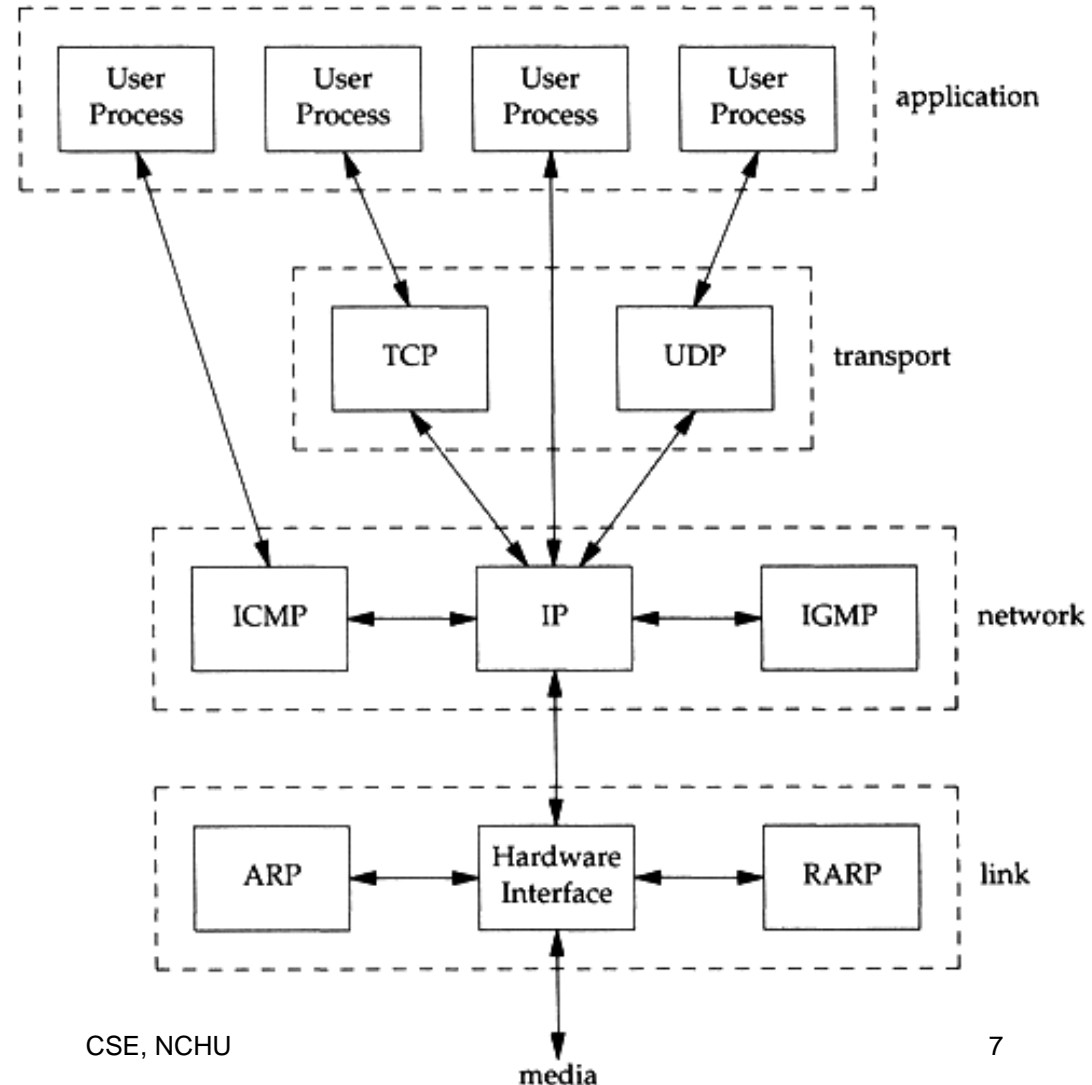


- Various Addresses
  - MAC Address
    - Media Access Control Address
    - 48-bit Network Interface Card Hardware Address
      - 24bit manufacture ID
      - 24bit serial number
    - Ex: 00:07:e9:10:e6:6b
  - IP Address
    - 32-bit Internet Address
    - Ex:140.120.13.254
  - Port
    - 16-bit uniquely application identification
    - Ex:FTP port 21, ssh port 22, telnet port 23

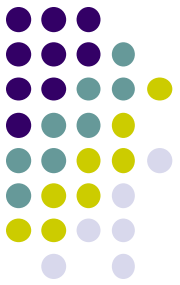
# General Introduction (4)



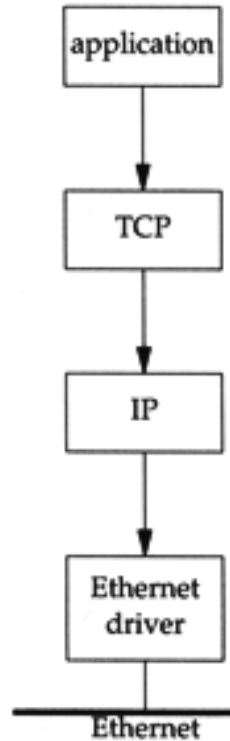
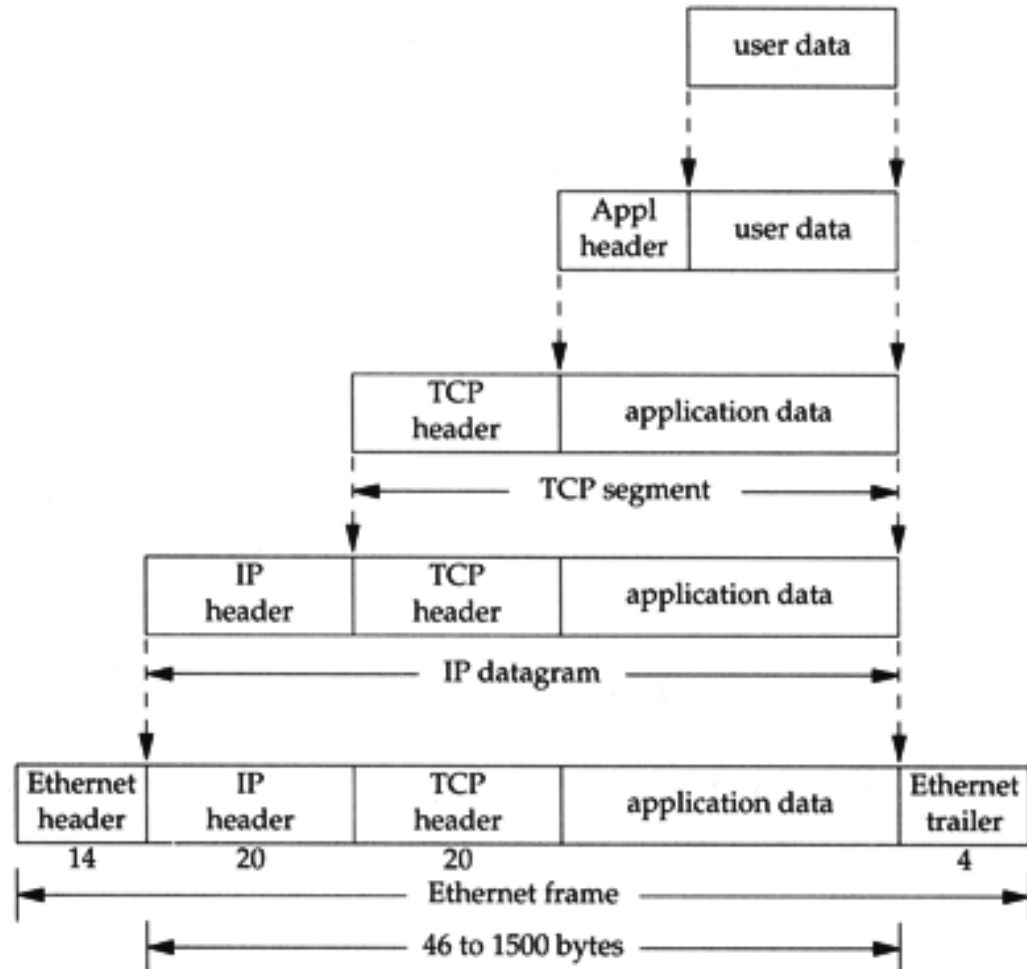
- Each layer has several protocols
  - A layer defines a data comm. function that may be performed by certain protocols
  - A protocol provides a service suitable to the function of that layer.



# General Introduction (5)

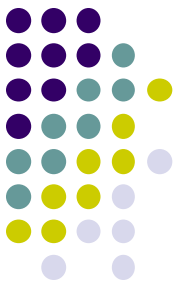


- Send data (encapsulation)

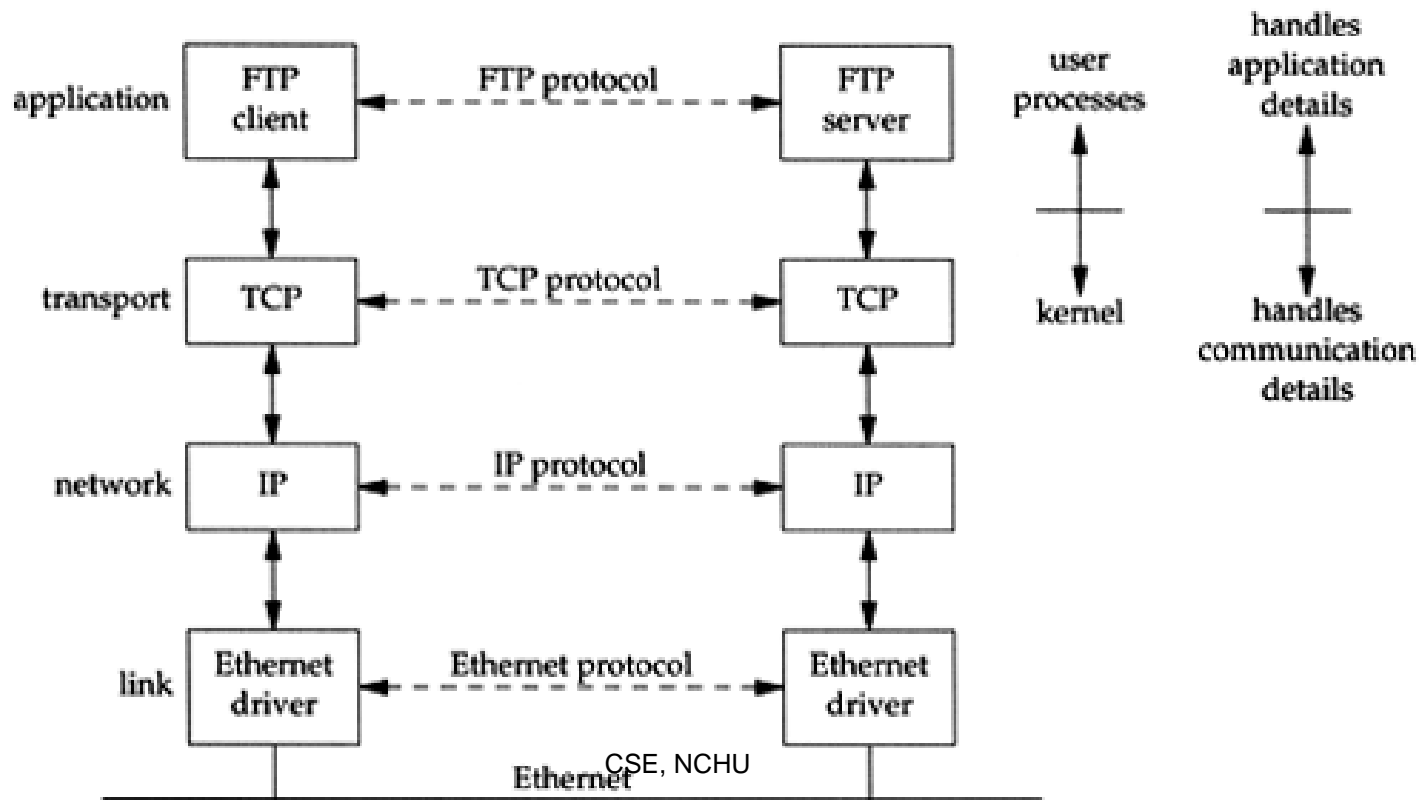




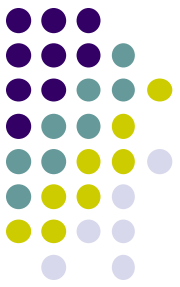
# General Introduction (6)



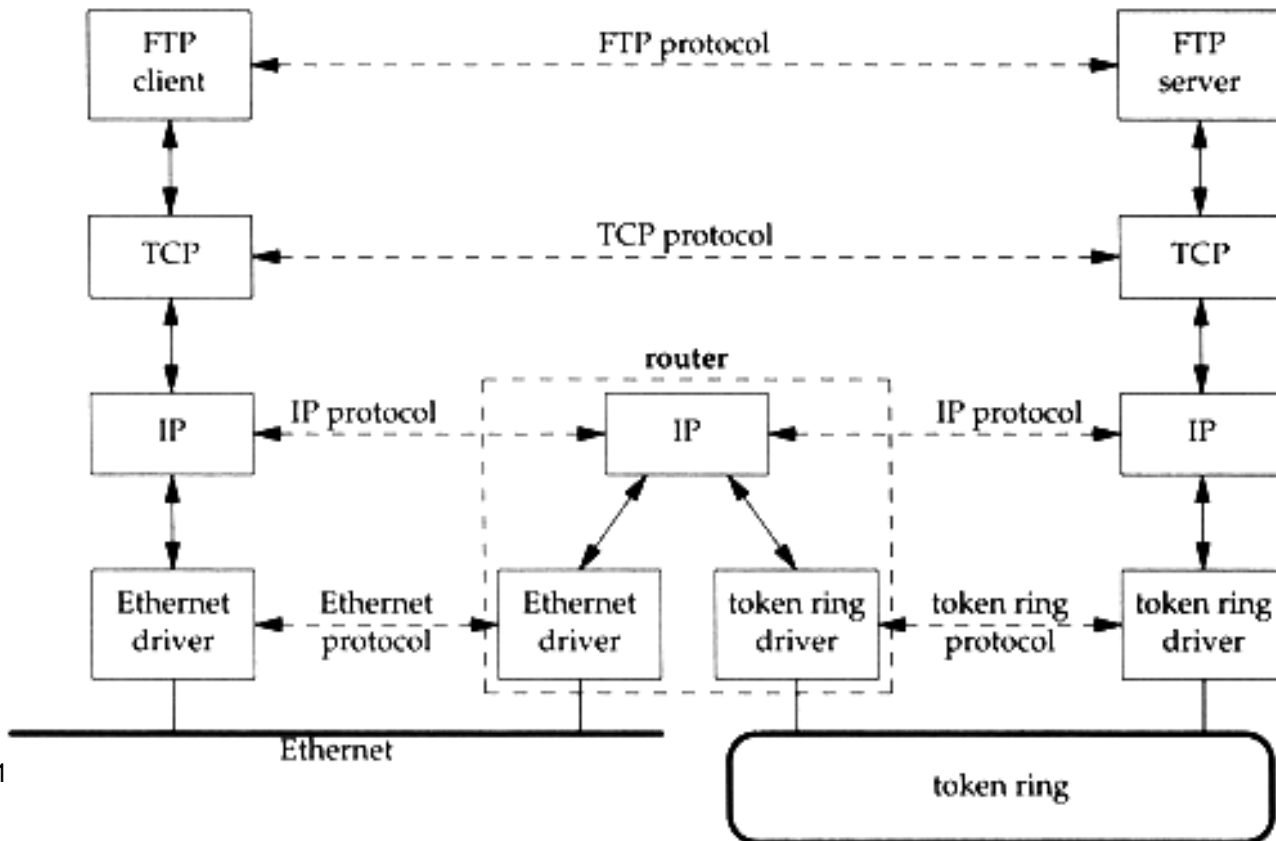
- Addressing --- nearby (same network)



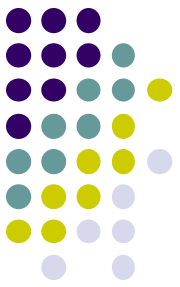
# General Introduction (7)



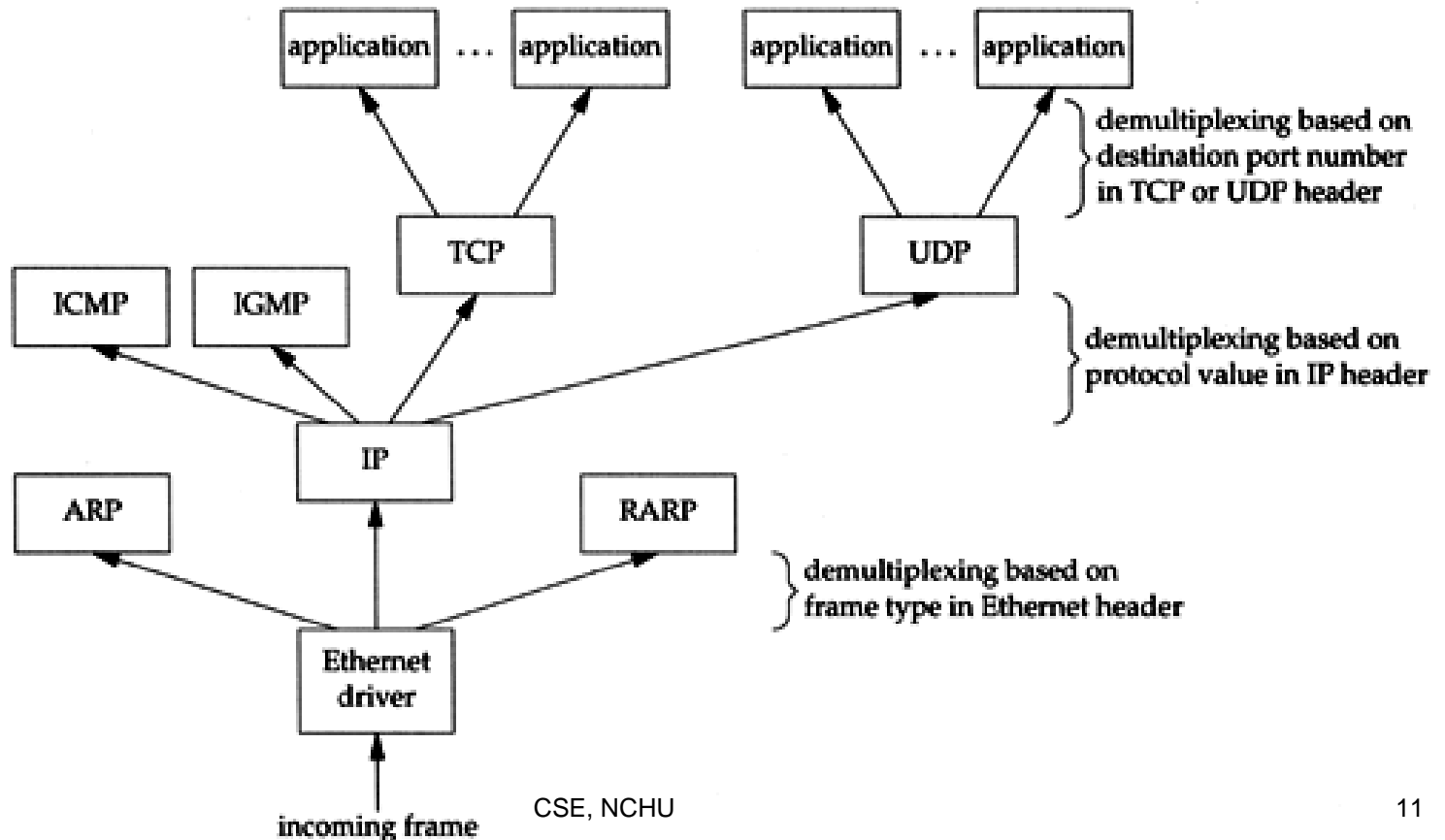
- Addressing --- faraway  
(across network)



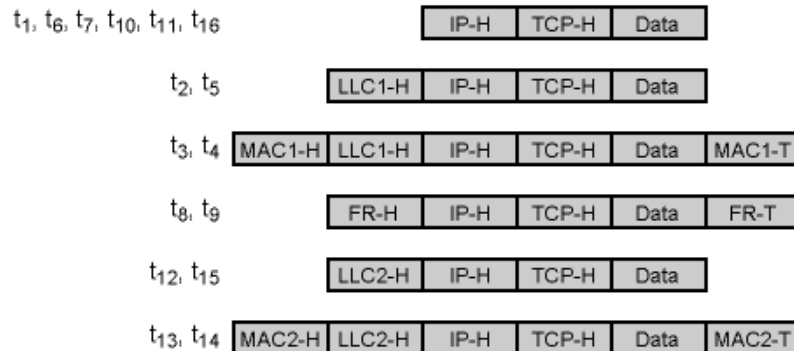
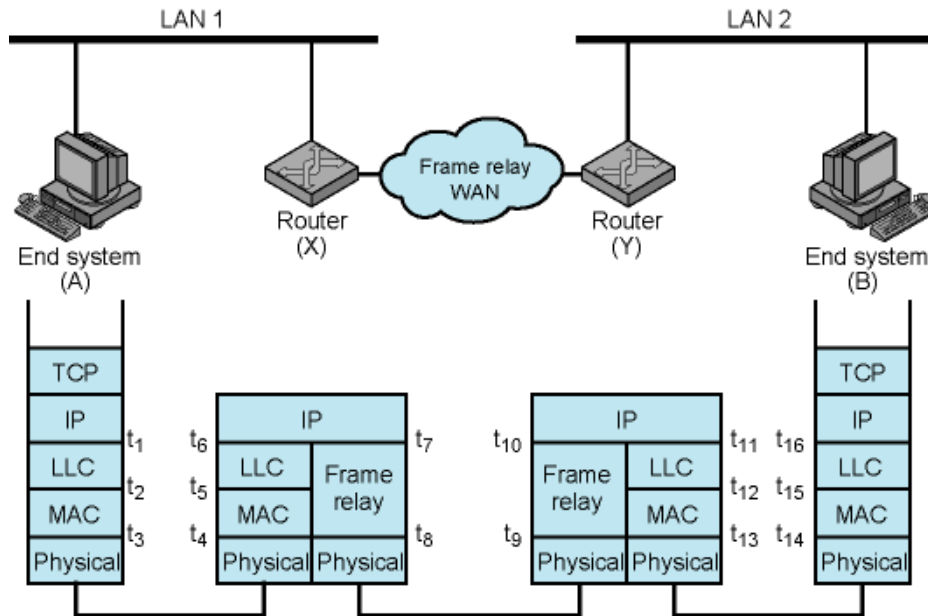
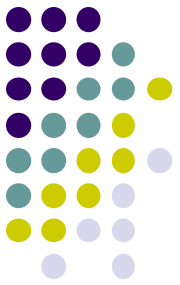
# General Introduction (8)



- Receive Data --- demultiplexing

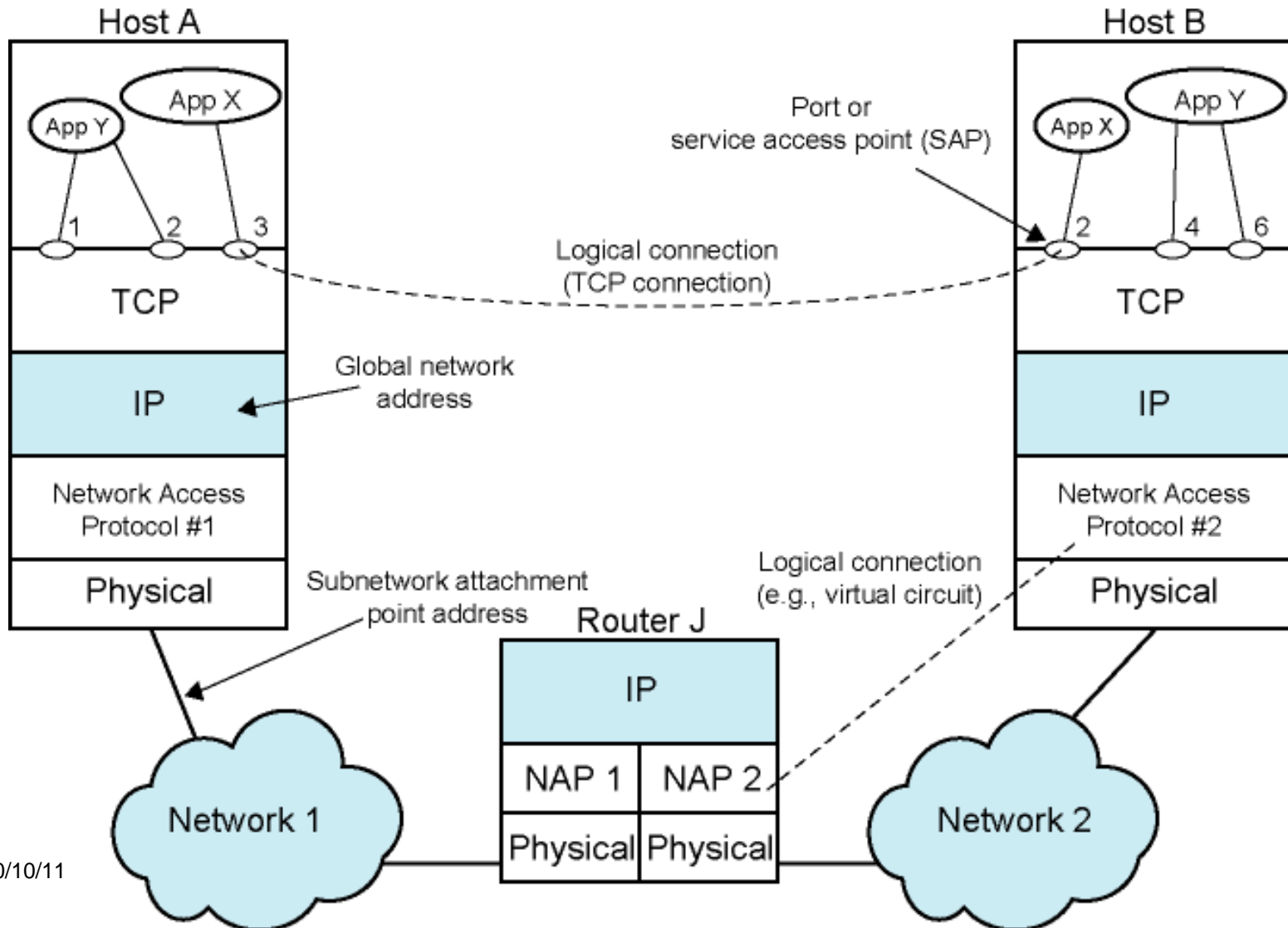
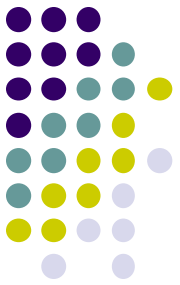


# General Introduction (9)



TCP-H = TCP header      MACi-T = MAC trailer  
 IP-H = IP header        FR-H = Frame relay header  
 LLCi-H = LLC header    FR-T = Frame relay trailer  
 MACi-H = MAC header

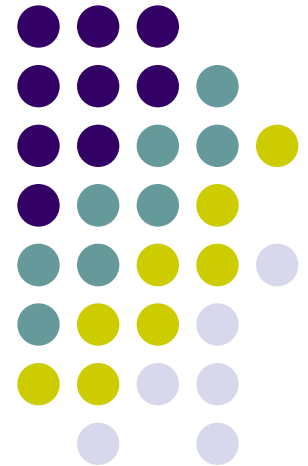
# General Introduction (10)

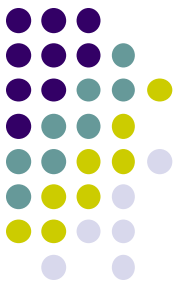


# Interface Layer

Provides 2 services for IP:

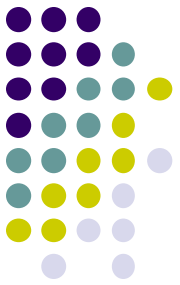
- Mapping IP addresses to Interface specific addresses
- Encapsulating IP datagrams for transmission over a specific medium





# Introduction of Interface Layer

- Tasks of Interface layer
  - Send and receive IP datagrams for IP module
  - ARP request and reply
  - RARP request and reply
- TCP/IP support various link layers, depending on the type of hardware used:
  - Ethernet
  - Token Ring
  - FDDI (Fiber Distributed Data Interface)

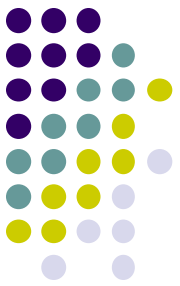


# Ethernet

- Features

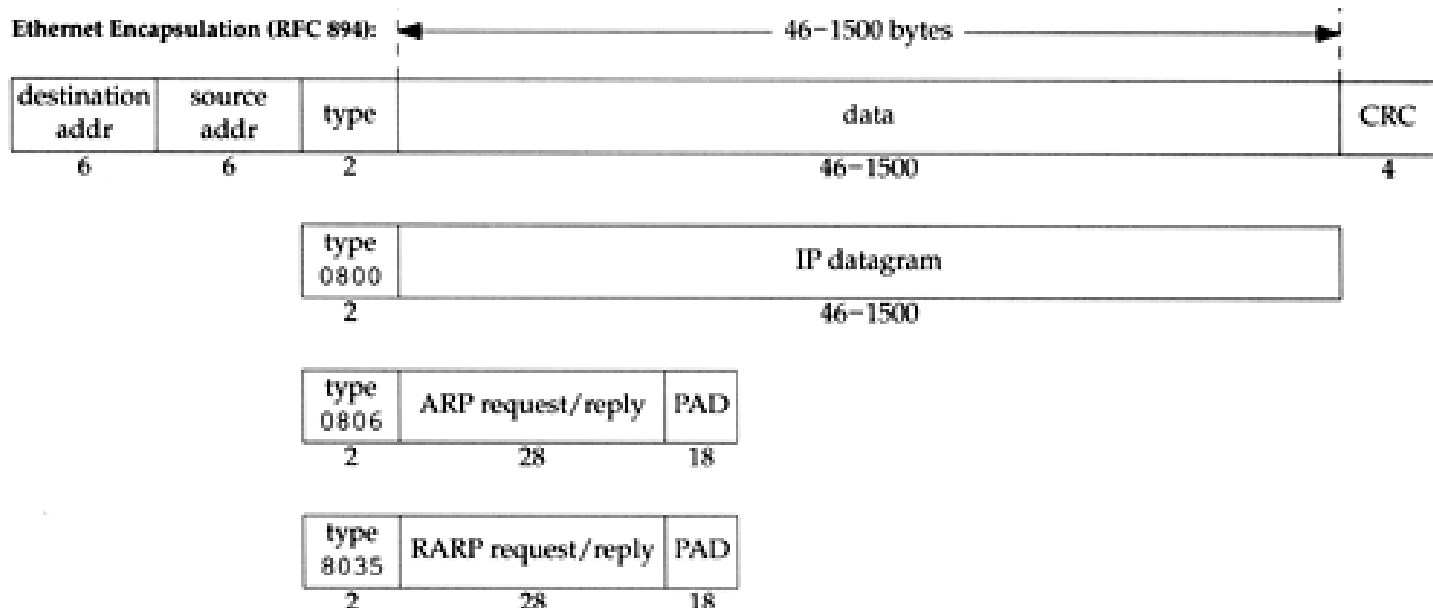
- Predominant form of local LAN technology used today
- Use CSMA/CD
  - Carrier Sense, Multiple Access with Collision Detection
- Use 48bit MAC address
- Operate at 10/100/1000 Mbps
- Ethernet frame format is defined in RFC894
  - This is the actually used format in reality



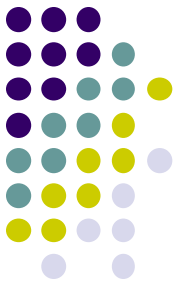


# Ethernet Frame Format

- 48 bits hardware address
  - For both destination and source address
- 16-bit type is used to specify the type of data
  - 0800 → IP datagram
  - 0806 → ARP, 8035 → RARP



# Loopback Interface



- Pseudo NIC

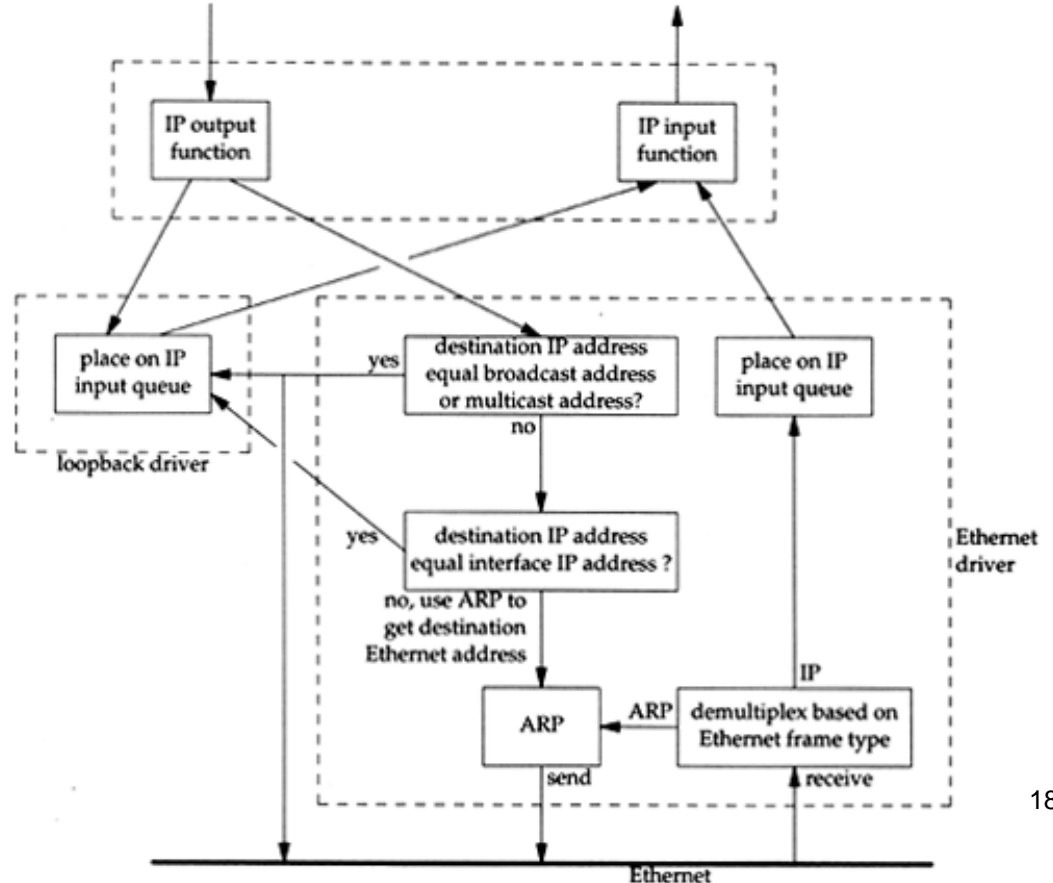
- Allow client and server on the same host to communicate with each other using TCP/IP

- IP

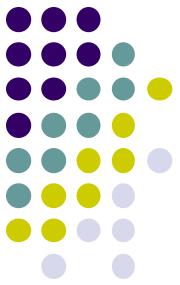
- 127.0.0.1

- Hostname

- localhost



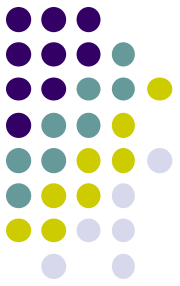
# MTU



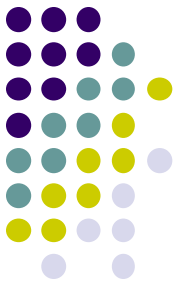
- Maximum Transmission Unit
  - Limit size of payload part of Ethernet frame
    - 1500 bytes
  - If the IP datagram is larger than MTU,
    - IP performs “fragmentation”
- MTU of various physical device, **576B** by default
- Path MTU
  - Smallest MTU of any data link MTU between the two hosts
  - Depend on route

Network	MTU (bytes)
Hyperchannel	65535
16 Mbits/sec token ring (IBM)	17914
4 Mbits/sec token ring (IEEE 802.5)	4464
FDDI	4352
Ethernet	1500
IEEE 802.3/802.2	1492
X.25	576
Point-to-point (low delay)	296

# Address translation Mechanisms

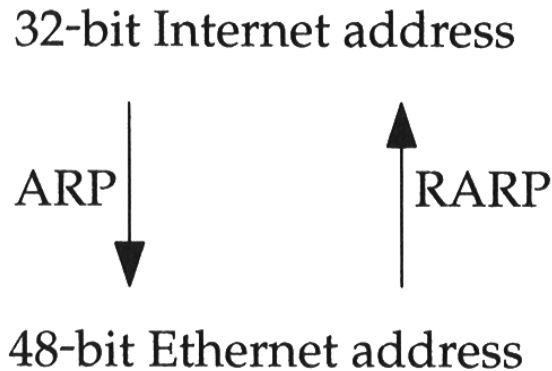


1. **Algorithmically** --- deterministic method for achieving one-to-one mapping
2. **Statically** --- a static table is built during system configuration
3. **Dynamically** --- a decentralized fashion using a protocol to determine the mapping



# ARP and RARP

- Mapping between IP and Ethernet address



- When an Ethernet frame is sent on LAN from one host to another,
  - It is the 48bit Ethernet address that determines for which interface the frame is destined

# Address Resolution Protocol



- **Is a dynamic mechanism and use the broadcasting facility**

When the local IP wants to send a datagram to a device on an attached Ethernet, the interface layer performs the following:

- (1) ARP cache is first consulted
- (2) If the IP-MAC address is not in cache, give up the IP datagram and send out an ARP frame
- (3) Upon receiving an ARP frame,
  - a) check the sender protocol address and update its corresponding media address
  - b) check the target protocol address, if not the local IP then discards, otherwise construct ARP response frame to the original sender

# ARP Example

% ftp ccbsd5

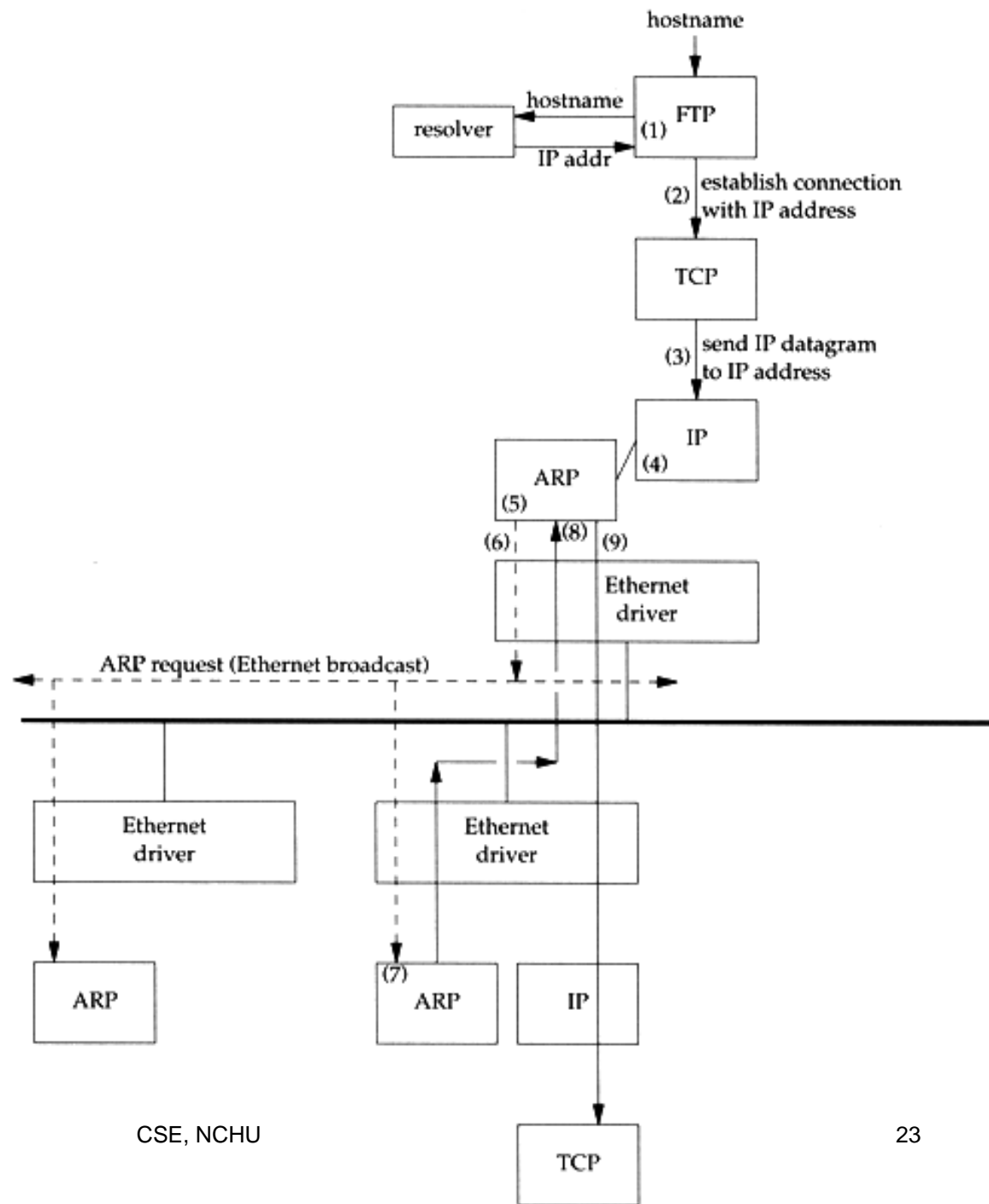
(4) next-hop or direct host

(5) Search ARP cache

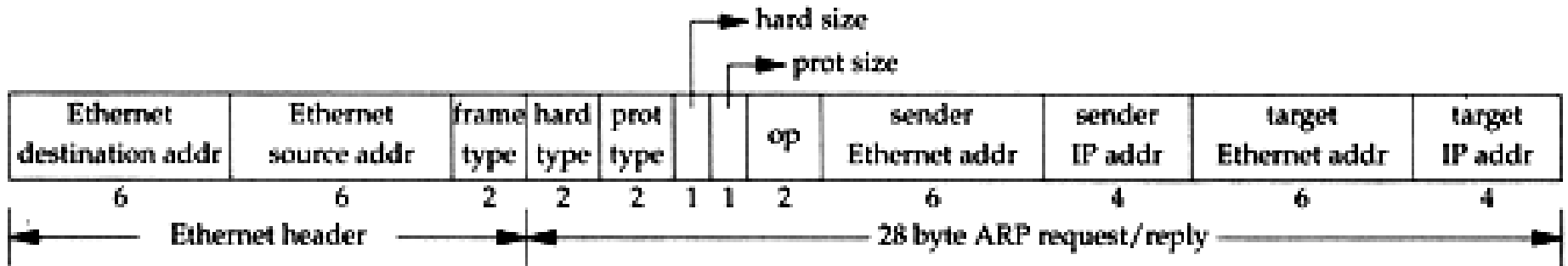
(6) Broadcast ARP request

(7) ccbsd5 response ARP reply

(9) Send original IP datagram



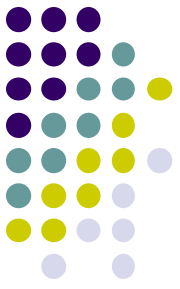
# ARP/RARP Packet Format



- Ethernet dest. addr: all 1's (broadcast)
- Known value for IP  $\leftrightarrow$  Ethernet
  - Frame type: 0x0806 (ARP), 0x8035 (RARP)

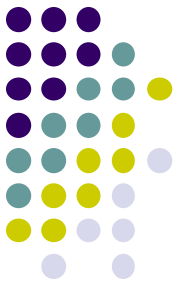


# ARP Frame



- **Frame type:** 0x0806 (ARP), 0x8035 (RARP)
- **Hardware type:** type of hardware address (1 for Ethernet)
- **Protocol type:** type of upper layer address (0x0800 for IP)
- **Hard size:** size in bytes of hardware address (6 for Ethernet)
- **Protocol size:** size in bytes of upper layer address (4 for IP)
- **Op:** 1, 2, 3, 4 for ARP request, reply, RARP request, reply

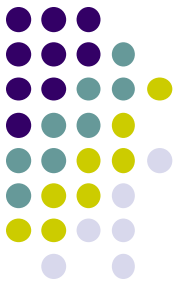
# ARP Cache



- Maintain recent ARP results
  - come from both ARP request and reply
  - expiration time
    - Complete entry = 20 minutes
    - Incomplete entry = 3 minutes
  - Use arp command to see the cache
  - Ex: % arp -a
    - % arp -da
    - % arp -S ccbsd5.csie.nctu.edu.tw
    - 00:07:e9:39:66:77

```
tytsai@tybsd: ~ > arp -a
ccamd.csie.nctu.edu.tw (140.113.235.1) at 00:0f:ea:48:92:85 on fxp0 [ethernet]
tybsd.csie.nctu.edu.tw (140.113.235.4) at 00:09:6b:7a:25:f7 on fxp0 permanent [ethernet]
e3rtn-235.csie.nctu.edu.tw (140.113.235.254) at 00:0e:38:a4:c2:00 on fxp0 [ethernet]
? (192.168.1.30) at (incomplete) on fxp1 [ethernet]
```

# Ex: Use tcpdump to see ARP



- Host 140.113.214.22 → 140.113.214.49
  - Clear ARP cache of 140.113.214.22 (0:20:ed:6d:eb:c )
  - Run tcpdump on 140.113.214.49 (0:2:a5:6e:8d:42 )
    - % sudo tcpdump -i fxp0 -e arp
    - % sudo tcpdump -i fxp0 -n -e arp
    - % sudo tcpdump -i fxp0 -n -t -e arp
  - On 140.113.214.22, ssh to 140.113.214.49

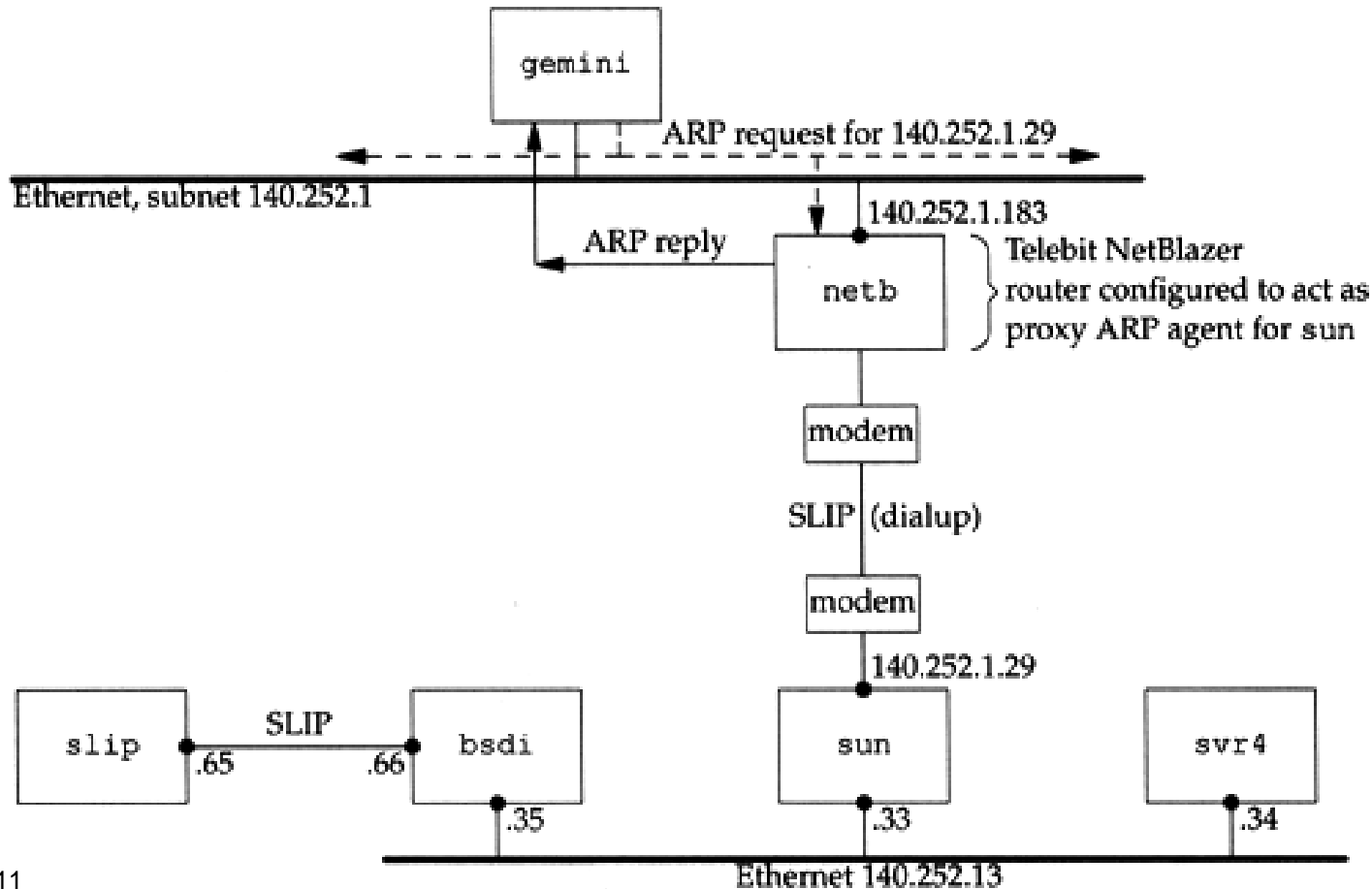
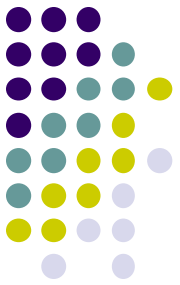
```
23:35:04.971913 0:20:ed:6d:eb:c Broadcast arp 60: arp who-has r21619.csie.nctu.edu.tw tel  
l u214.csie.nctu.edu.tw  
23:35:04.971921 0:2:a5:6e:8d:42 0:20:ed:6d:eb:c arp 60: arp reply r21619.csie.nctu.edu.tw  
is-at 0:2:a5:6e:8d:42
```

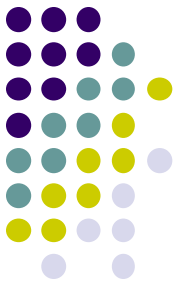
```
23:44:12.407720 0:20:ed:6d:eb:c ff:ff:ff:ff:ff:ff 0806 60: arp who-has 140.113.214.49 tel  
l 140.113.214.22  
23:44:12.407730 0:2:a5:6e:8d:42 0:20:ed:6d:eb:c 0806 60: arp reply 140.113.214.49 is-at 0  
:2:a5:6e:8d:42
```

```
0:20:ed:6d:eb:c ff:ff:ff:ff:ff:ff 0806 60: arp who-has 140.113.214.49 tell 140.113.214.22  
0:2:a5:6e:8d:42 0:20:ed:6d:eb:c 0806 60: arp reply 140.113.214.49 is-at 0:2:a5:6e:8d:42
```

# Proxy ARP

- Let router answer ARP request on one of its networks for a host on another network

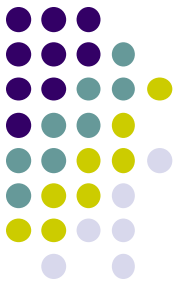




# Gratuitous ARP

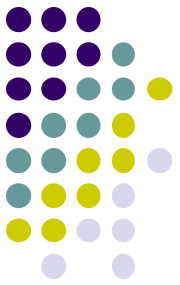
- Gratuitous ARP
  - The host sends an ARP request looking for its own IP
  - Provide two features
    - Used to determine whether there is another host configured with the same IP
    - Used to cause other host to update ARP cache when changing its hardware address

# Examples of ARP Spoofing & ARP Proxy Agent



- Provided by the instructor as shown in the classroom.

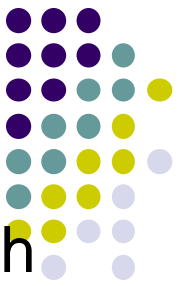
(refer to **CH09 ARP協定.ppt**, p41- p55)



# RARP

- Principle
  - Used for the diskless system to read its hardware address from the NIC and send an RARP request to gain its IP
- RARP Server Design
  - RARP server must maintain the map from hardware address to an IP address for many hosts
  - Link-layer broadcast
    - This prevents most routers from forwarding an RARP request

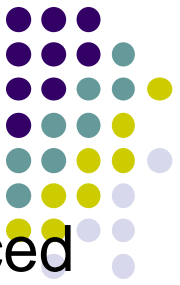
# RARP, BOOTP, and DHCP (1/2)



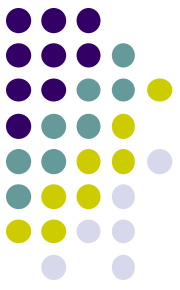
- RARP server sends the broadcasts query and back with an IP address
- RARP broadcast message can't get through routers, an RARP server needs on each network
- BOOTP uses UDP and forwards over routers, but requires manual configuration of table mapping
- DHCP server and client use **DHCPDISCOVER**, **DHCPOFFER**, **DHCPREQUEST**, and **DHCPACK** to build the communication. (only the later 2 messages if the client has been admitted.)



# RARP, BOOTP, and DHCP(2/2)

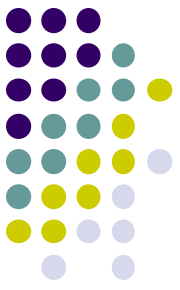


- RARP is replaced by BOOTP, and DHCP is an enhanced version of BOOTP.
- Rather than **static** allocation, DHCP uses **lease-based** IP allocation, either **automatic** allocation or **dynamic** allocation, depending upon the duration of the lease.
- DHCP server may offer TCP/IP setups for its clients, for instances, router address, DNS address, netmask, etc.
- DHCP server may exist over distance, newly booted host requires **DHCP relay** agent (by broadcasting DHCP recovery packet) to unicast message to DHCP server.



# IP Assignment (1/2)

- IP assignment is a process where a subnets are created for each subgroup or department.
- The IP assignment is tracked by the network operations center (NOC).
- For example, the subnet 192.168.12.64 has broadcast address 192.168.12.127. Any computer in the subnet is assigned in the range of 65 to 126.
- IP assignment can be done either manually or dynamically, BOOTP and DHCP are two well-known approaches for IP assignment.



# IP Assignment (2/2)

- DHCP is a superset of BOOTP and runs on the same port number.
- DHCP requests an IP address from the DHCP server. The DHCP server retrieves an available IP address from a pool dedicated to the subnet of the requesting client, and the DHCP server also specifies a lease time together with the IP address.
- Router doesn't pass broadcast addresses, so a DHCP relay server is required for over distance (out of a LAN).
- DHCP requires UDP and uses port 68 for client, 67 for server. Message Type (MT) include **discover**, **offer**, **request**, and **ack** (later two MTs are unicasts between relay and server).