

# ARP: Address Resolution Protocol

*Surasak Sanguanpong*

*nguan@ku.ac.th*

*<http://www.cpe.ku.ac.th/~nguan>*

Last updated: May 24, 1999

# Agenda

- **Why need ARP?**
- **ARP functionality**
- **Proxy ARP**
- **RARP**

---

# IP and MAC address

- **Stations need to know MAC address to communicate**
- **Hardware MAC address**
  - Ethernet 6 bytes
  - Token ring 2 or 6 bytes
  - FDDI 2 or 6 bytes
- **HOW DOES IP ADDRESS GET MAPPED TO MAC ADDRESS?**
  - manual configuration by hand is tedious
  - automatic process by ARP

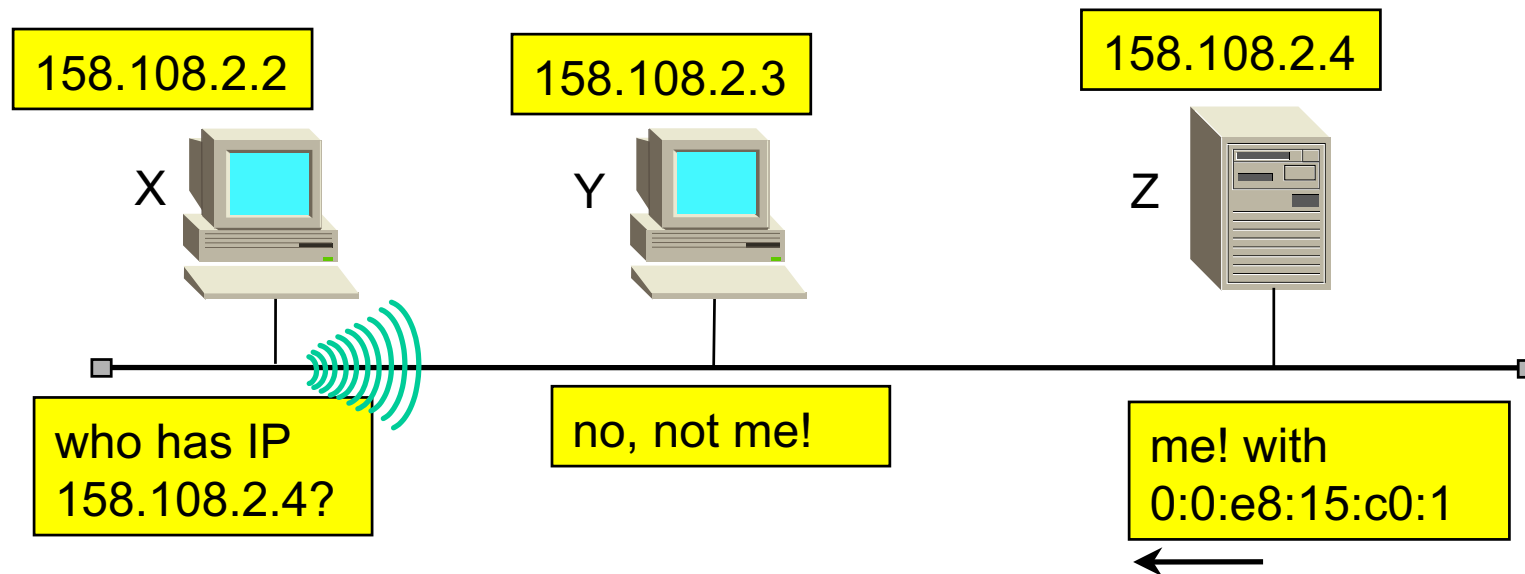
---

# ARP protocol

- **RFC 826 - Address Resolution Protocol**
- **ARP maps any network level address (such as IP) to its corresponding data link address (such as Ethernet)**
- **supported protocol in datalink layers, not data link layer protocol**

# ARP in operation

- Host X want to resolve MAC address of Z
  - X sends broadcast ARP request
  - X gets unicast ARP reply from Z




# ARP as a command line

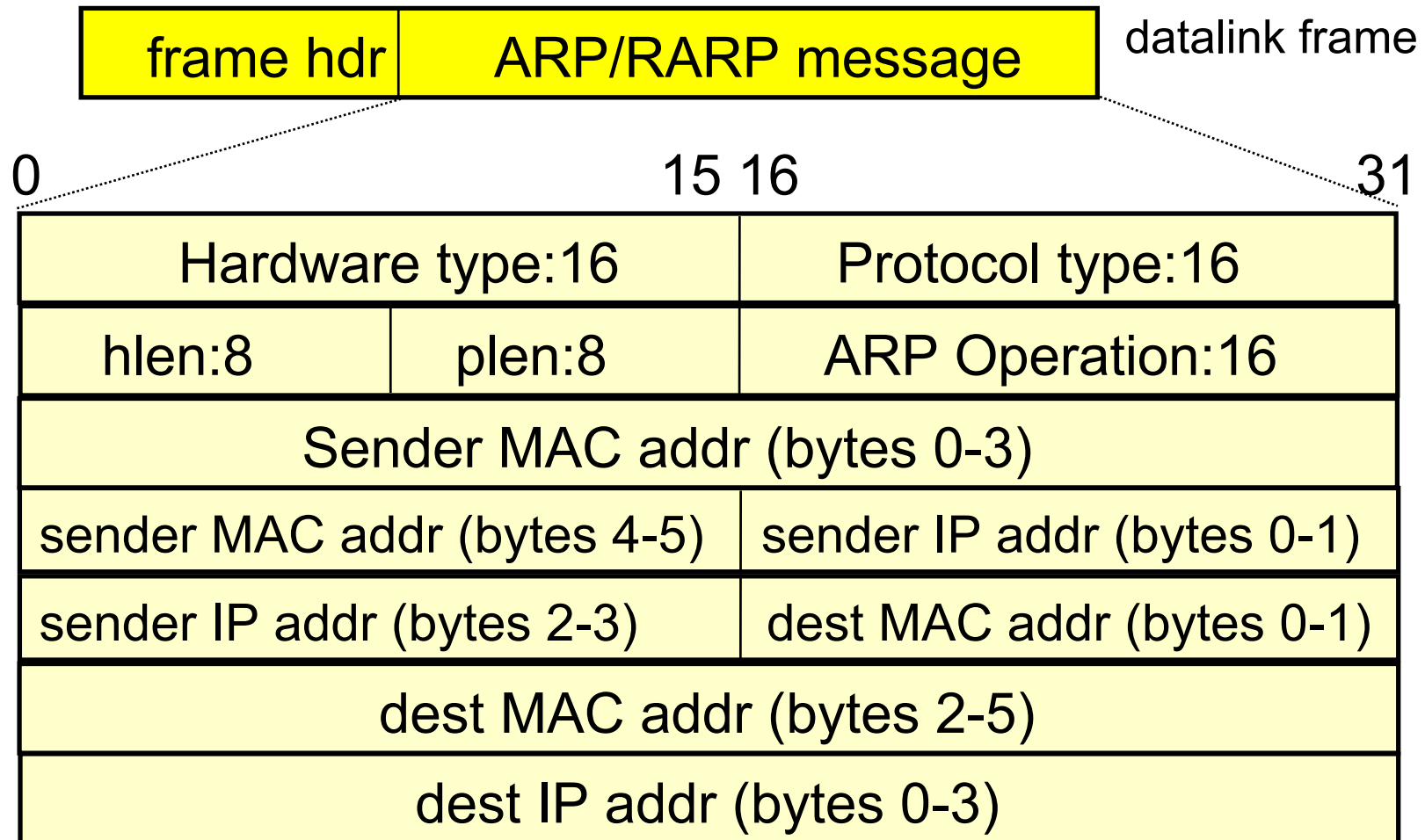
```
% arp -a
www.cpe.ku.ac.th (158.108.33.5) at 0:0:e8:15:cc:c
% telnet cc
:
% arp -a
router.cpe.ku.ac.th (158.108.33.1) at 0:0:c:6:13:4a
cc.cpe.ku.ac.th (158.108.33.2) at 2:60:8c:2e:b5:8b
www.cpe.ku.ac.th (158.108.33.5) at 0:0:e8:15:cc:c
```

entry in ARP table

more entries added



# ARP datagrams

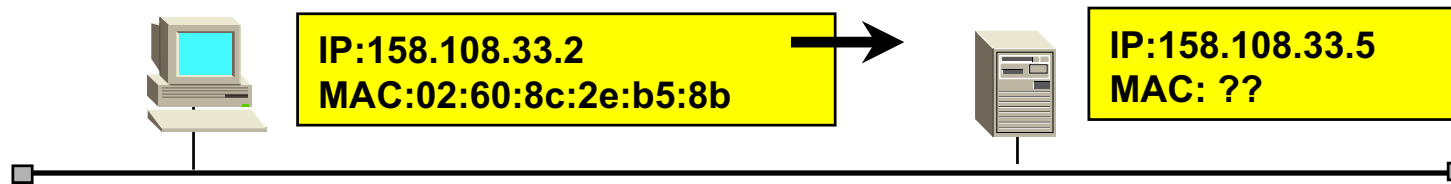


---

# Header details

- **hardware type** : Ethernet=1 ARCNET=7, localtalk=11
- **protocol type** : IP=0x800
- **hlen** : length of hardware address, Ethernet=6 bytes
- **plen** : length of protocol address, IP=4 bytes
- **ARP operation** : ARP request = 1, ARP reply = 2  
RARP request = 3, RARP reply = 4

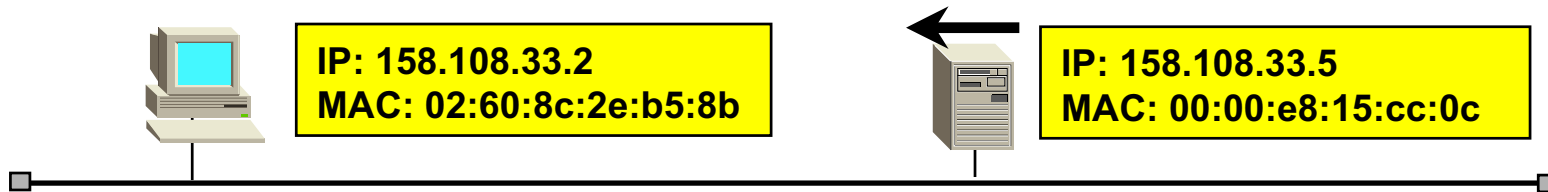
# ARP request packet



Sample ARP request Ethernet packet

FF:FF:FF:FF:FF			dest MAC (broadcast)
02:60:8c:2e:b5:8b			source MAC
0x0806			ARP frame type
0x01		0x800	Ethernet / IP
0x06	0x04	0x001	MAC=6/ IP=4 /request
02:60:8c:2e:b5:8b			source MAC
158.108.33.2			source IP
00:00:00:00:00:00			dest MAC (unknown)
158.108.33.5			dest IP
checksum			Ethernet checksum

# ARP reply packet



Sample ARP reply Ethernet packet

02:60:8c:2e:b5:8b			dest MAC (unicast)
00:00:e8:15:cc:0c			source MAC
0x0806			ARP frame type
0x01		0x800	Ethernet / IP
0x06	0x04	0x002	MAC=6/ IP=4 /reply
00:00:e8:15:cc:0c			source MAC
158.108.33.5			source IP
02:60:8c:2e:b5:8b			dest MAC
158.108.33.2			dest IP
checksum			Ethernet checksum

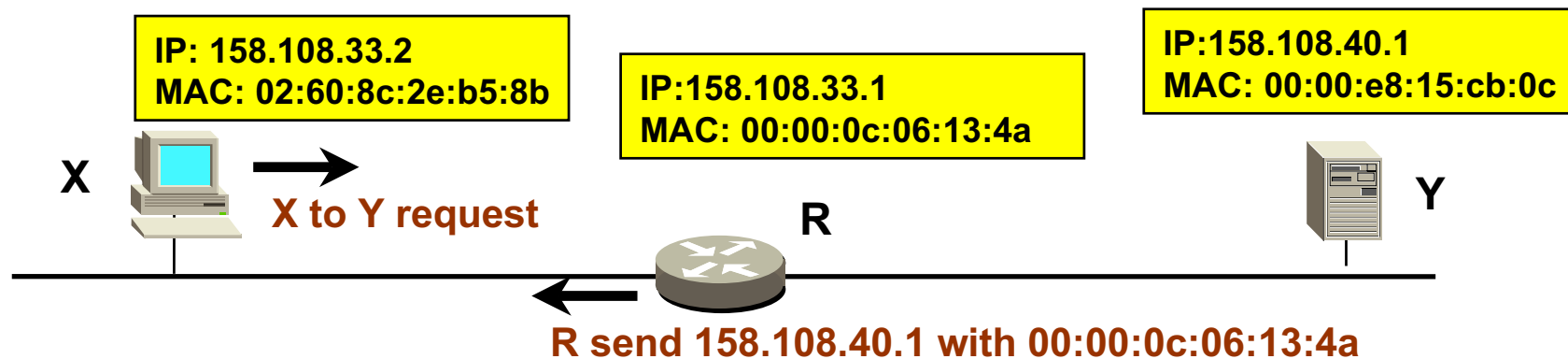
---

# ARP mechanisms

- **Each node maintains the ARP cache**
  - it first looks in the cache to find entry first
  - if the entry is not used for a period (~15 minutes), it is deleted.
- **Receive node can adds an MAC addr entry for source station in its own cache.**
- **ARP traffic load**
  - hosts quickly add cache entries.
  - If all hosts on a subnet are booted at the same time? => flurry of ARP requests and reply.

# Proxy ARP

- One node answers ARP request for another: Router R answers for Y



- Useful when some nodes on a network cannot support subnet
  - X do not understand subnet, so it thinks that Y is on the same subnet
- Router must be configured to be a proxy ARP

# RARP

- **Reverse ARP : map MAC addr to IP addr**
- **For device that can not store IP, usually diskless workstations**
- **Need to setup server wit RARP table**
- **Use the same frame format**
  - **0x0835 for Ethernet RARP request**
  - **operation 0x003 = RARP request**  
**0x004 = RARP reply**
- **RARP can not operate across router, BOOTP is more spread**