



Binding Protocol Address

รศ. ดร. อนันต์ พลเพิ่ม

Asso. Prof. Anan Phonphoem, Ph.D.

anan.p@ku.ac.th

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

Computer Engineering Department

Kasetsart University, Bangkok, Thailand



Outline

- About IP address
- Address resolution algorithms
- Address Resolution Protocol (ARP)



About IP Address

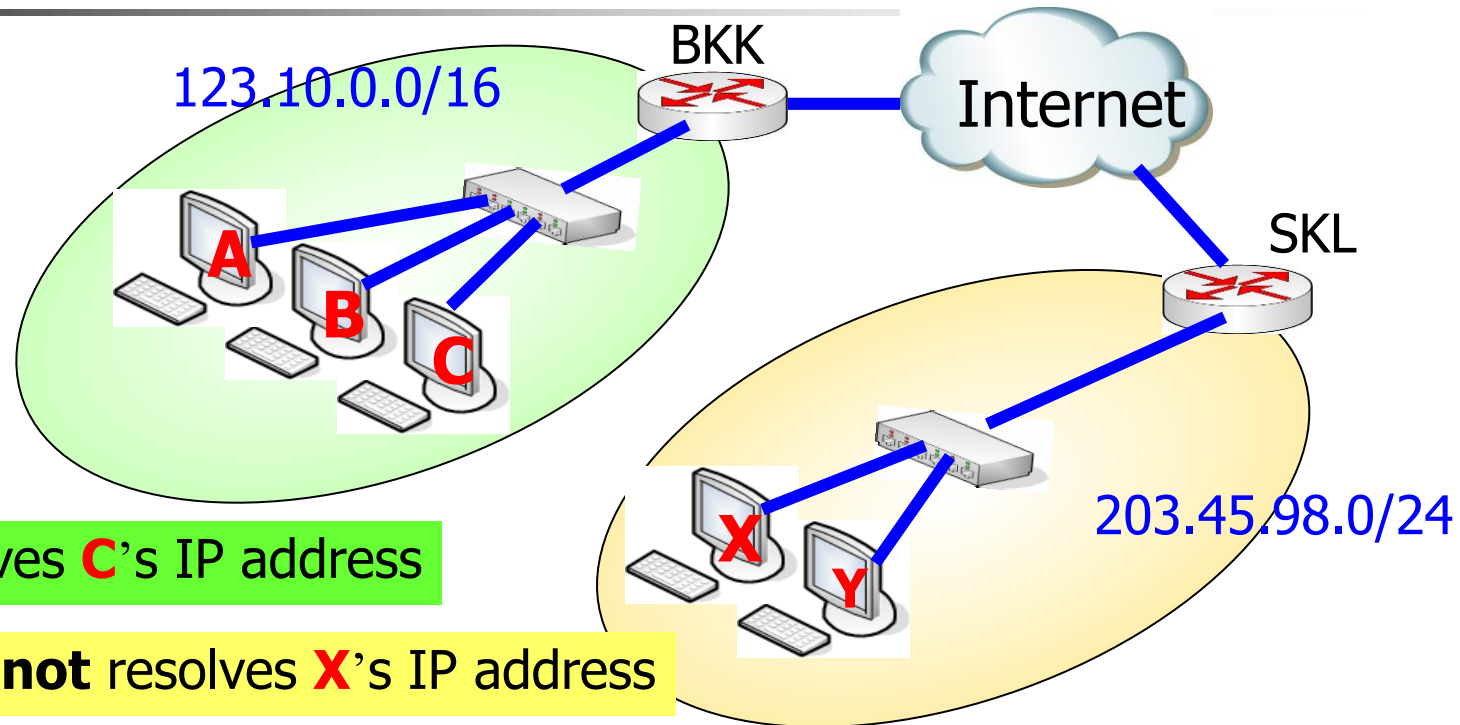
- Virtual address
 - maintained by software
 - Address for network layer
- Sending frame occurs in data link layer
 - Ethernet – MAC address (hardware add)
- Relationship between IP and MAC add.
 - need mapping process



Address Resolution

- Translation process
 - from protocol address (IP add)
 - to equivalent hardware address (MAC add)
- IP is resolved to MAC address
- Process occurs locally
 - same physical network (such as LAN)

Locally Resolve Address



A → C A resolves C's IP address

A → X A does **not** resolve X's IP address

A resolves **BKK**'s IP address → sends data to BKK
BKK resolves **next hop** IP address → sends data to ...

...

SKL resolves **X**'s IP address → sends data to X



Address Resolution Algorithms

- Table lookup
 - used in WAN
- Closed-form computation
 - used in configurable network
- Message exchange
 - used in LAN with static address



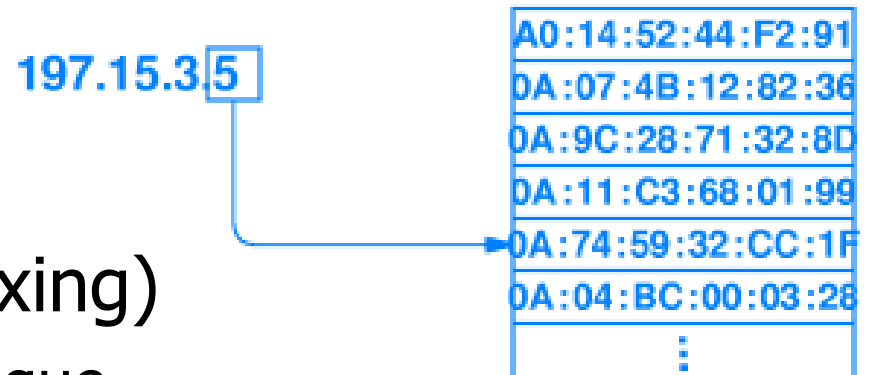
Table Lookup

- An array (table) stores in memory
- each entry contains (P,H)

IP Address		Hardware Address
197.15.3	2	0A:07:4B:12:82:36
197.15.3	3	0A:9C:28:71:32:8D
197.15.3	4	0A:11:C3:68:01:99
197.15.3	5	0A:74:59:32:CC:1F
197.15.3	6	0A:04:BC:00:03:28
197.15.3	7	0A:77:81:0E:52:FA

Table Lookup

- Easy
- For small network
 - fast searching
- For large network
 - Hashing (direct indexing)
 - data structure technique
 - more efficient



Hashing



Closed-Form Computation

- For configurable addressing
 - both hardware and IP add can be changed
- Compute **mathematical function** that maps IP to hardware address
- Choose value that optimizes translation

IP add	HW add
204.110.38.1	1
204.110.38.2	2
204.110.38.250	250



Message Exchange

- Resolve add by sending message and wait for reply
- Two possible designs
 - centralized – server
 - distributed – each node response



Message Exchange

- Centralized
 - easy to configure and maintain
 - need server(s) – cost
 - can become bottleneck
- Distributed
 - cheap (no server)
 - need communication protocol
 - communication overhead



Outline

- About IP address
- Address resolution algorithms
- **Address Resolution Protocol (ARP)**



Address Resolution Protocol (ARP)

- Message exchange
- TCP/IP protocol suite
- Two basic message types
 - request – IP that want to resolve
 - response – IP and requested MAC

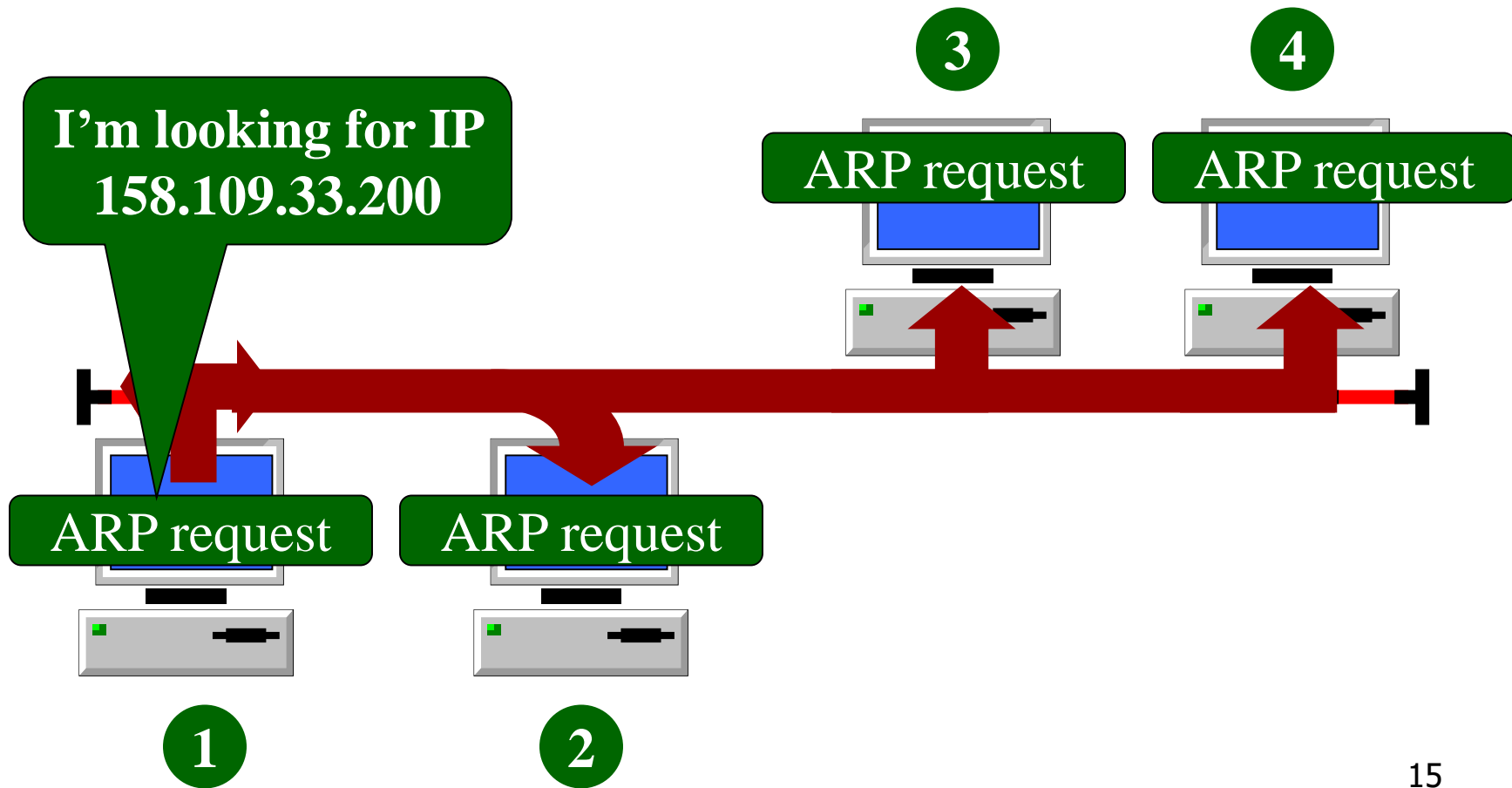


ARP protocol

- 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 data link layers, not data link layer protocol

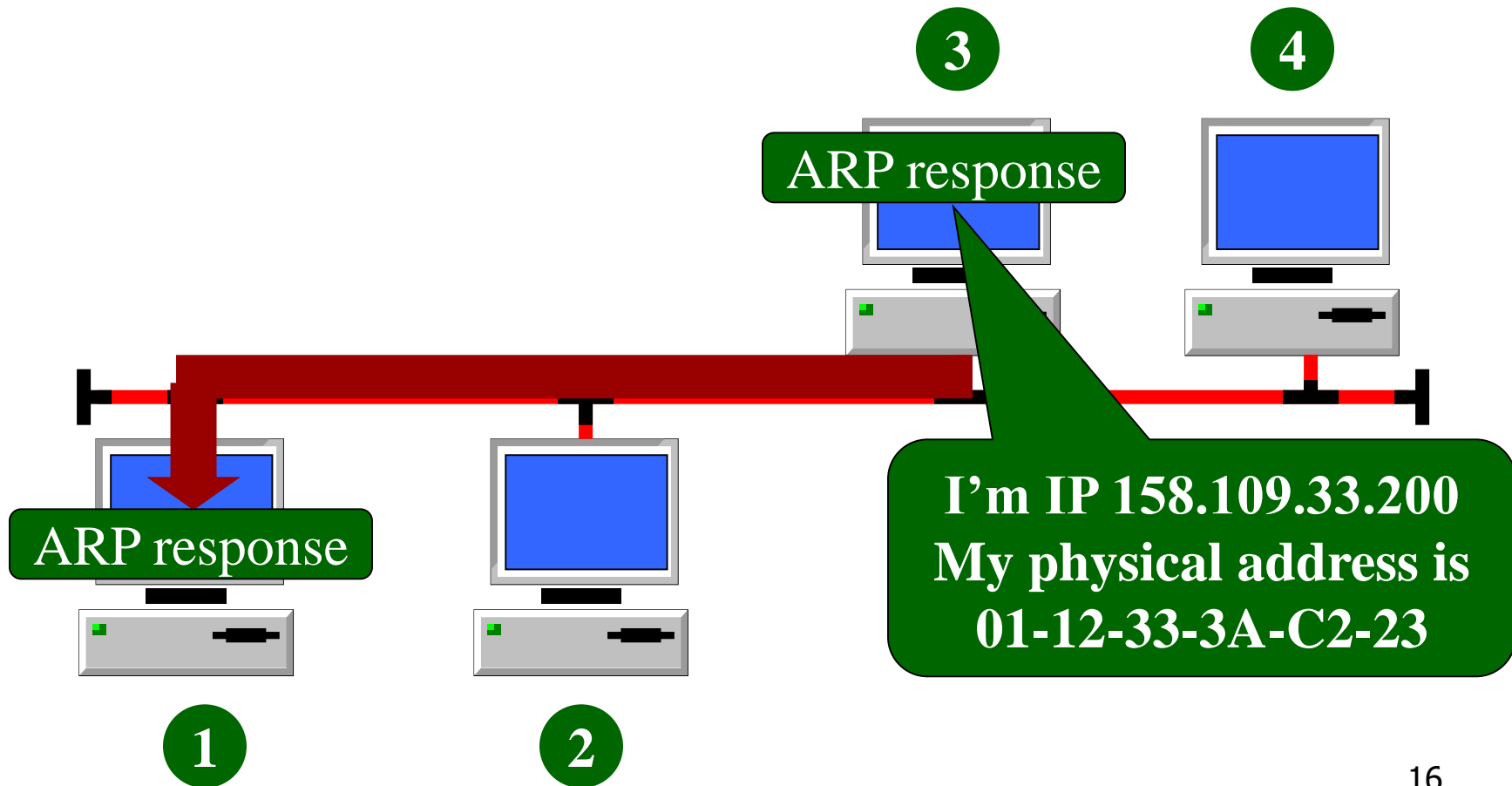
ARP Protocol

Station 1 is looking for MAC add of IP 158.109.33.200



ARP Protocol

Station 3 (IP 158.109.33.200) responses



ARP as a command line

entry in ARP table

```
% 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
```

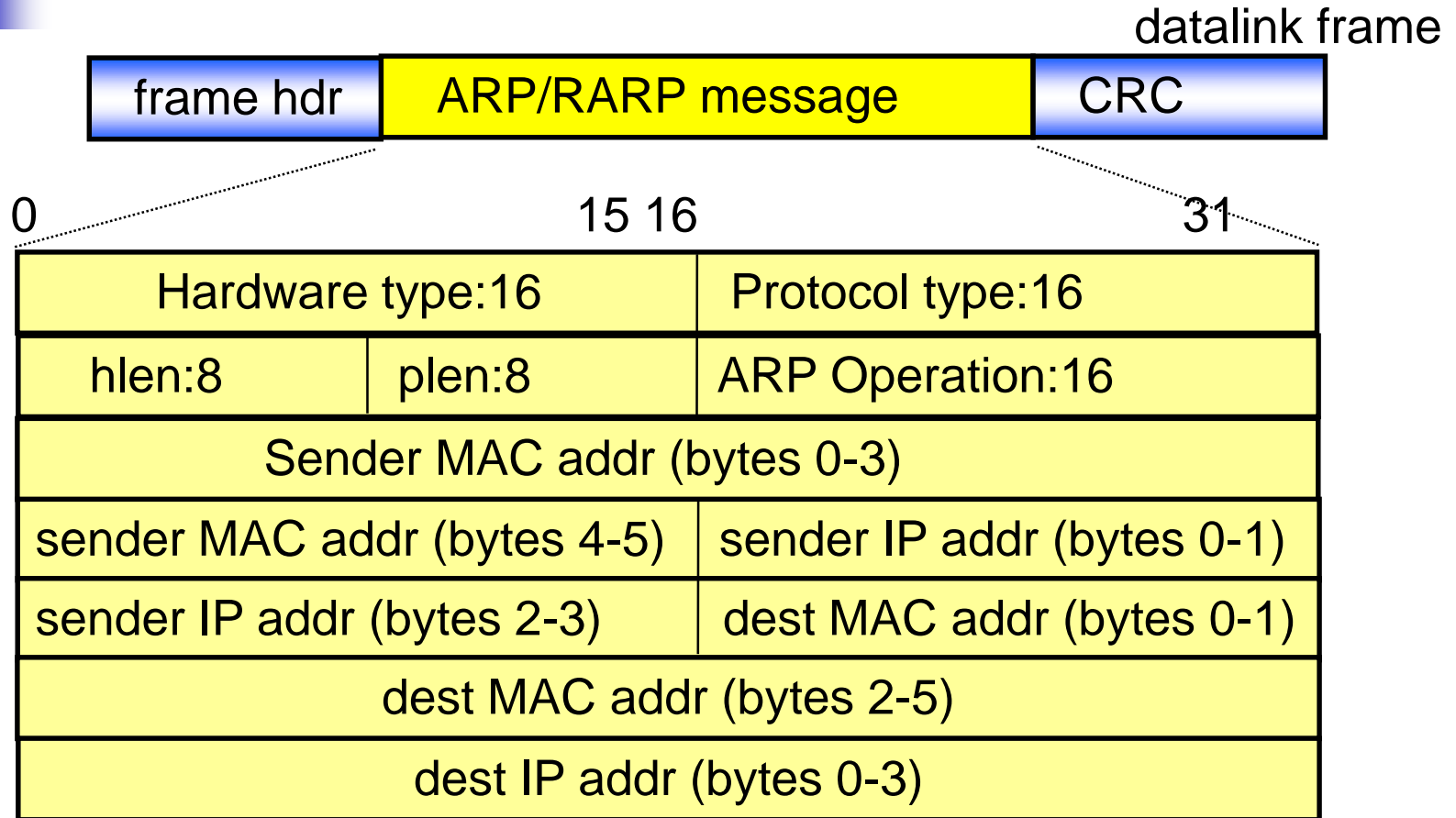
```
www.cpe.ku.ac.th (158.108.33.5) at 0:0:e8:15:cc:c
```



ARP in action

To display arp table : `arp -a`
To delete arp table : `arp -d`

ARP datagrams



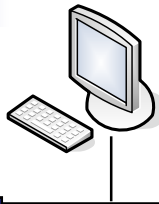


Header details

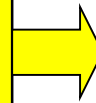
Hardware type:16		Protocol type:16	
hlen:8	plen:8	ARP Operation:16	
Sender MAC addr (bytes 0-3)			
sender MAC addr (bytes 4-5)		sender IP addr (bytes 0-1)	
sender IP addr (bytes 2-3)		dest MAC addr (bytes 0-1)	
dest MAC addr (bytes 2-5)			
dest IP addr (bytes 0-3)			

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

ARP request packet



IP: 158.108.33.2 MAC: 02:60:8c:2e:b5:8b
 IP: 158.108.33.5 MAC: ??



Sample ARP request Ethernet packet

FF:FF:FF:FF:FF		
02:60:8c:2e:b5:8b		
0x0806		
0x01		0x800
0x06	0x04	0x001
02:60:8c:2e:b5:8b		
158.108.33.2		
00:00:00:00:00:00		
158.108.33.5		
checksum		

dest MAC (broadcast)

source MAC

ARP frame type

Ethernet / IP

MAC=6/ IP=4 /request

source MAC

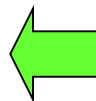
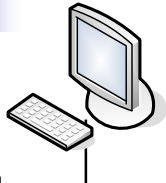
source IP

dest MAC (unknown)

dest IP

Ethernet checksum

ARP reply packet



IP: 158.108.33.5 MAC:00:00:e8:15:cc:0c
 IP: 158.108.33.2 MAC:02:60:8c:2e:b5:8b



Sample ARP reply Ethernet packet

02:60:8c:2e:b5:8b		
00:00:e8:15:cc:0c		
0x0806		
0x01	0x800	
0x06	0x04	0x002
00:00:e8:15:cc:0c		
158.108.33.5		
02:60:8c:2e:b5:8b		
158.108.33.2		
checksum		

dest MAC (unicast)
 source MAC
 ARP frame type
 Ethernet / IP
 MAC=6/ IP=4 /reply
 source MAC
 source IP
 dest MAC
 dest IP
 Ethernet checksum



ARP mechanisms

- Each node maintains ARP cache
 - first looks in cache to find entry
 - if entry is not used for a while (~15 min), delete it
- Receive node can add MAC address entry for source 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



Summary

- About IP address
- Address resolution algorithms
 - Table lookup
 - Closed-form computation
 - Message exchange
- Address Resolution Protocol (ARP)