# Internet Protocols

## Supporting Protocols and Framing

Updated: 9/30/14

# Supporting Protocols

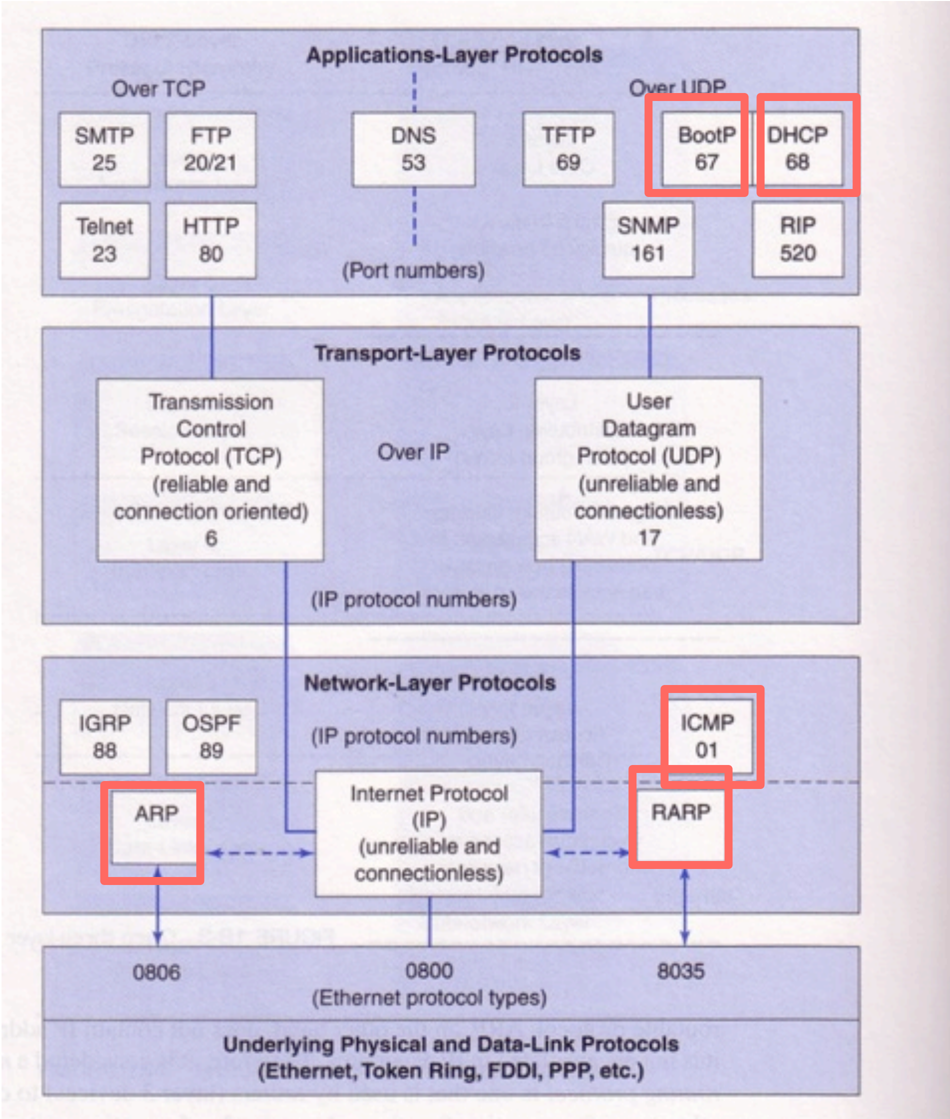
- ARP / RARP
- BOOTP
- ICMP
- DHCP
- NAT

# IP Supporting Protocols

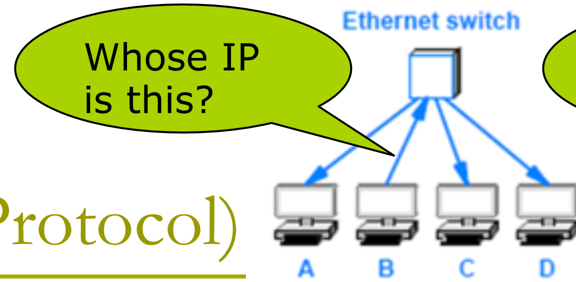
- IP protocol only deals with the data transfer (best-effort)
  - Possible Errors that can happen and not detected by IP: Data lost, duplication, out-of-order
  - However there are some error checking mechanisms:
    - CRC, TTL

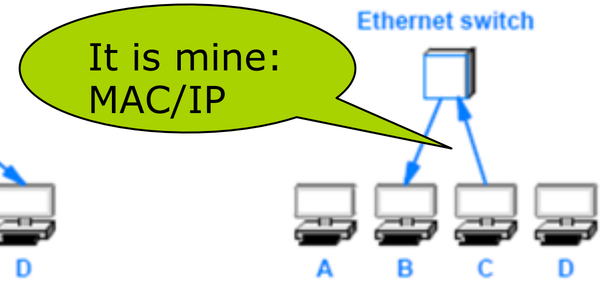
# IP Supporting Protocols

We focus on the following Protocols:
ICMP, ARP, RARP, BOOTP, DHCP



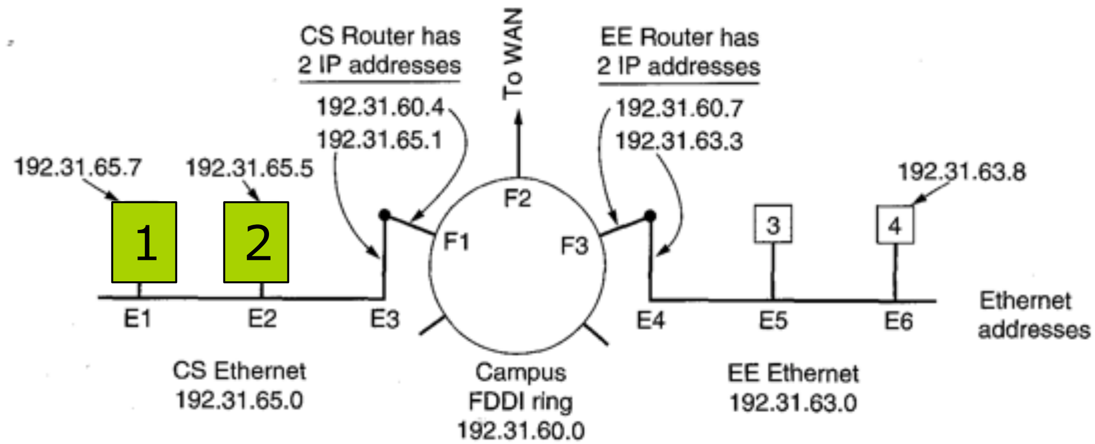
# ARP
# (Address Resolution Protocol)



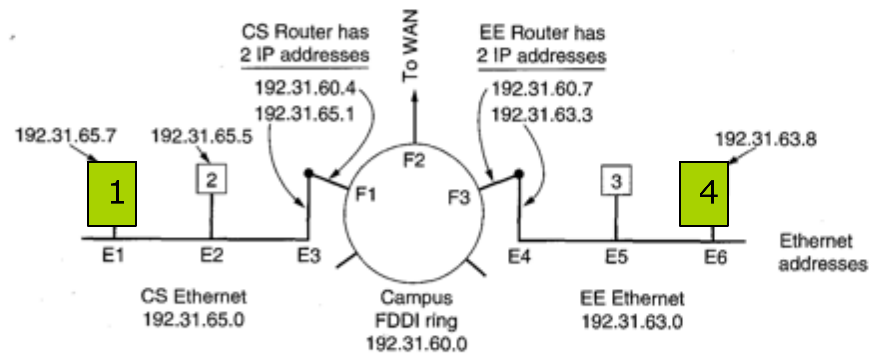

- Used to resolve network layer addresses into link layer addresses
- Exploits broadcast property of a LAN
- Each host on LAN maintains a table of IP subnetwork addresses
- If the address can not be found ARP broadcasts a request
  - Shouting: Who knows about this IP address?
- Other hosts listen and reply
  - The reply includes IP address and MAC (unicast)
  - Any interested host can learn about the new information

# ARP Example



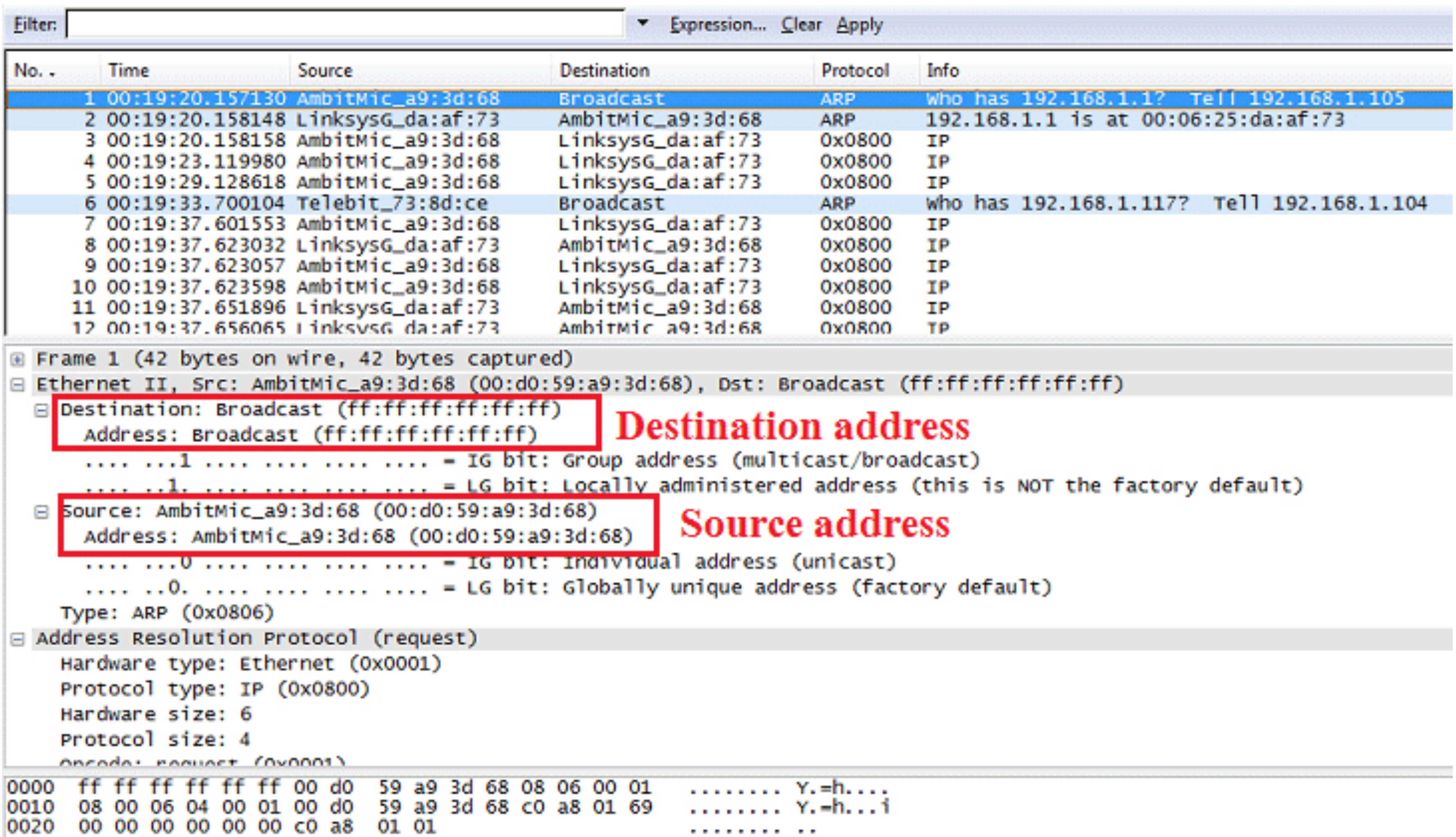

- Assume **1** is sending a message to **2** (192.31.65.5)
  - What is the MAC address for 192.31.65.5? Use ARP broadcast!
    - Host 2 responds to Host 1: it is E2
  - Host 1 maps IP and MAC;
    - Encapsulate the IP message in the Ethernet frame and sends it
  - Cashing can enhance ARP operation (Node 1 can cash the result)

# ARP Example

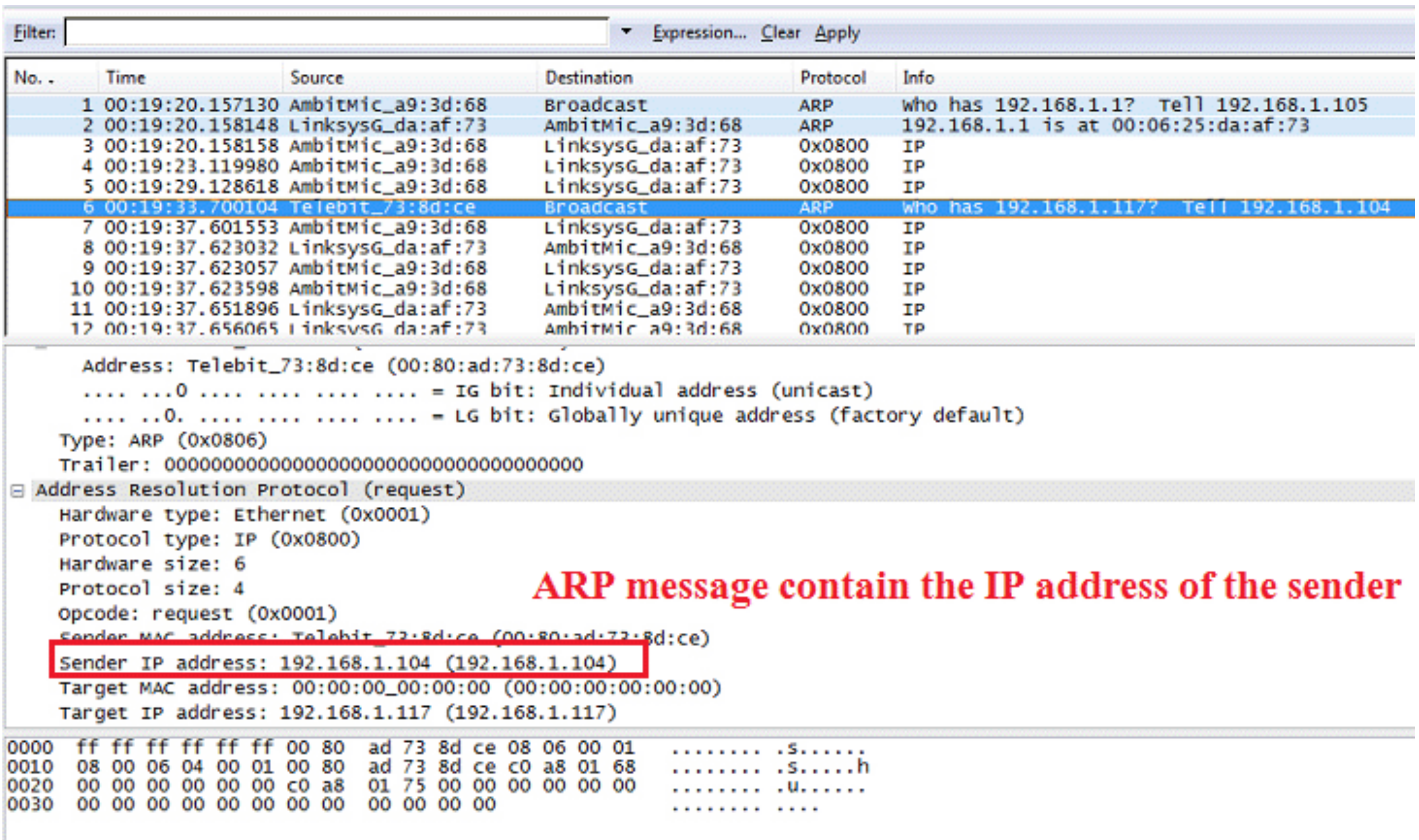



- Assume **1** is sending a message to **4** (rose@ee.sonoma.edu)
  - ee.sonoma.edu is the destination
  - Host **1** sends a message to Domain Name System (DNS): what is the IP address for ee.sonoma.edu? → 192.31.63.8
  - What is the MAC address for 192.31.63.8? ARP cannot pass through the router!
- Two choices:
  1. Reconfigure routers to respond to ARP (Proxy ARP)
     - The ARP Proxy is aware of the location of the destination
     - Proxy offers its own MAC address
     - Thus, it acts on behalf of the node: "send it to me, and I'll get it to where it needs to go."
     - In this example the Proxy can be E4
  2. Send the message to the LAN router
     - Note that ARP is limited to a single network
     - In the example above, the address binding or resolution is done between Node 1 and E3; then between E3 and E4; then E4 and node 4 (via broadcast).
     - Node 4 will send back its MAC to node 1 (not found in ARP cache)
     - Each router looks at the IP address and passes it to the next node using the routing table

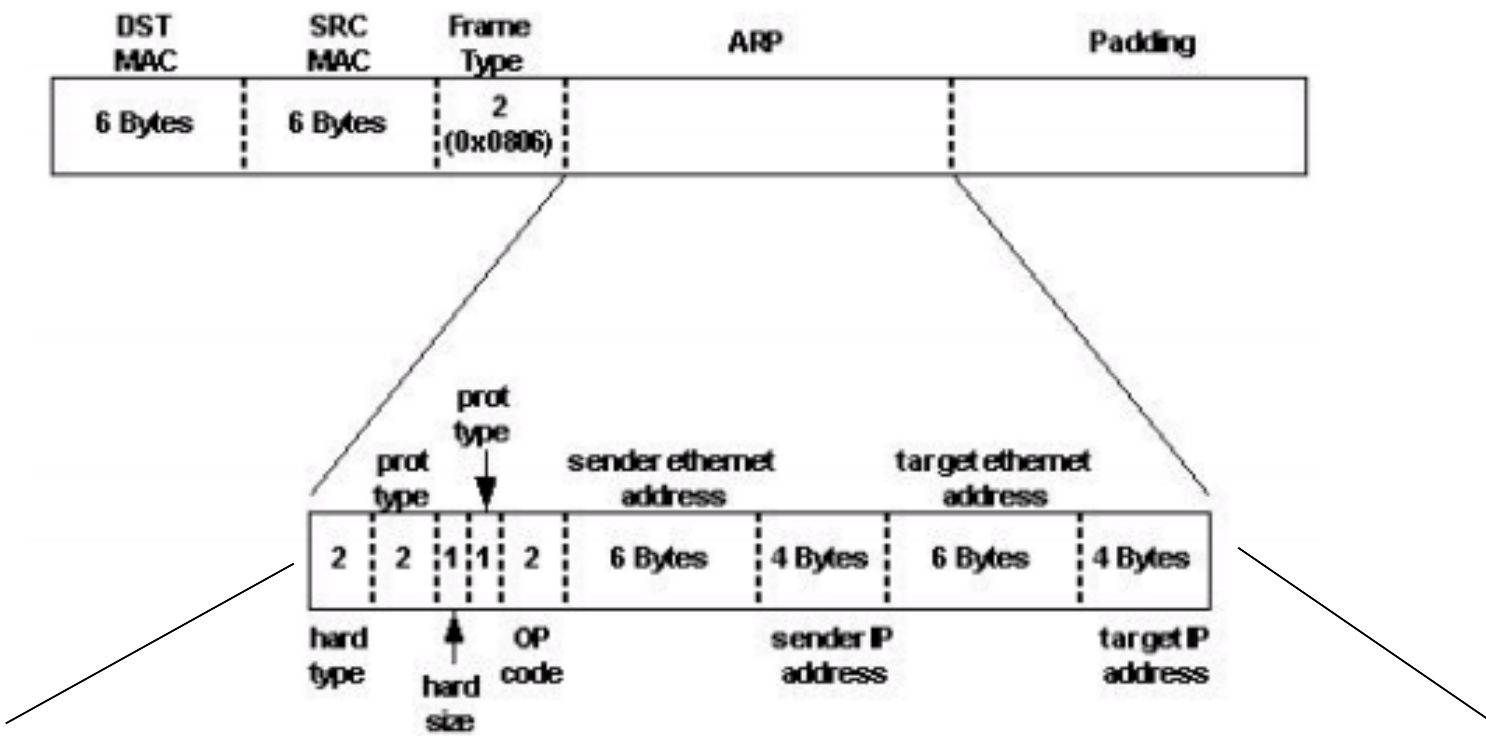
# ARP Request Content - Broadcast



# ARP Request Content –
## Contains IP Address



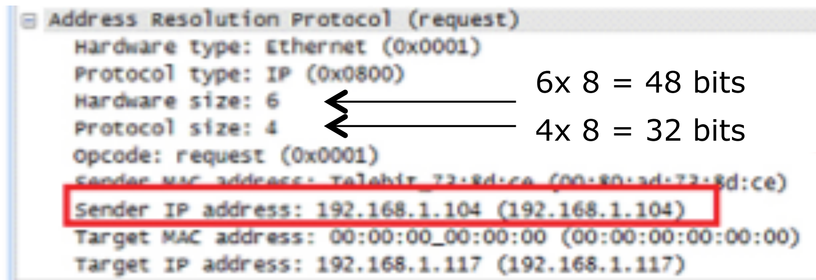
# ARP Message Format




| 0 | 8 | 16 | 24 – 31 |
|---|---|---|---|
| HARDWARE ADDRESS TYPE | | PROTOCOL ADDRESS TYPE | |
| HADDR LEN | PADDR LEN | OPERATION | |
| SENDER HADDR (first 4 octets) | | | |
| SENDER HADDR (last 2 octets) | | SENDER PADDR (first 2 octets) | |
| SENDER PADDR (last 2 octets) | | TARGET HADDR (first 2 octets) | |
| TARGET HADDR (last 4 octets) | | | |
| TARGET PADDR (all 4 octets) | | | |

# ARP Message Format

| 0 | 8 | 16 | 24 31 |
|---|---|---|---|
| HARDWARE ADDRESS TYPE | | PROTOCOL ADDRESS TYPE | |
| HADDR LEN | PADDR LEN | OPERATION | |
| SENDER HADDR (first 4 octets) | | | |
| SENDER HADDR (last 2 octets) | | SENDER PADDR (first 2 octets) | |
| SENDER PADDR (last 2 octets) | | TARGET HADDR (first 2 octets) | |
| TARGET HADDR (last 4 octets) | | | |
| TARGET PADDR (all 4 octets) | | | |

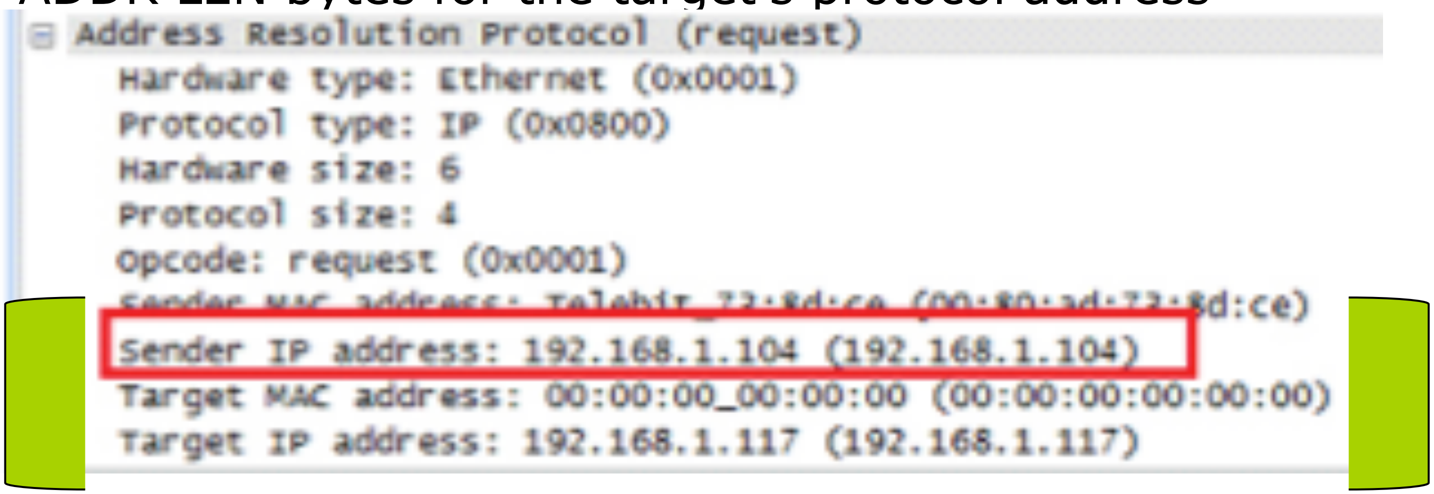



- HARDWARE ADDRESS TYPE
  - 16-bit field that specifies the type of hardware address being used
  - the value is 1 for Ethernet
- PROTOCOL ADDRESS TYPE
  - 16-bit field that specifies the type of protocol address being used
  - the value is 0x0800 for IPv4
- HADDR LEN
  - 8-bit integer that specifies the size of a hardware address in bytes
- PADDR LEN
  - 8-bit integer that specifies the size of a protocol address in bytes
- OPERATION
  - 16-bit field that specifies whether the message
    - request (the field contains 1) or
    - response (the field contains 2)

# ARP
# Message Format

| 0 | 8 | 16 | 24 31 |
|---|---|---|---|
| HARDWARE ADDRESS TYPE | | PROTOCOL ADDRESS TYPE | |
| HADDR LEN | PADDR LEN | OPERATION | |
| SENDER HADDR (first 4 octets) | | | |
| SENDER HADDR (last 2 octets) | | SENDER PADDR (first 2 octets) | |
| SENDER PADDR (last 2 octets) | | TARGET HADDR (first 2 octets) | |
| TARGET HADDR (last 4 octets) | | | |
| TARGET PADDR (all 4 octets) | | | |

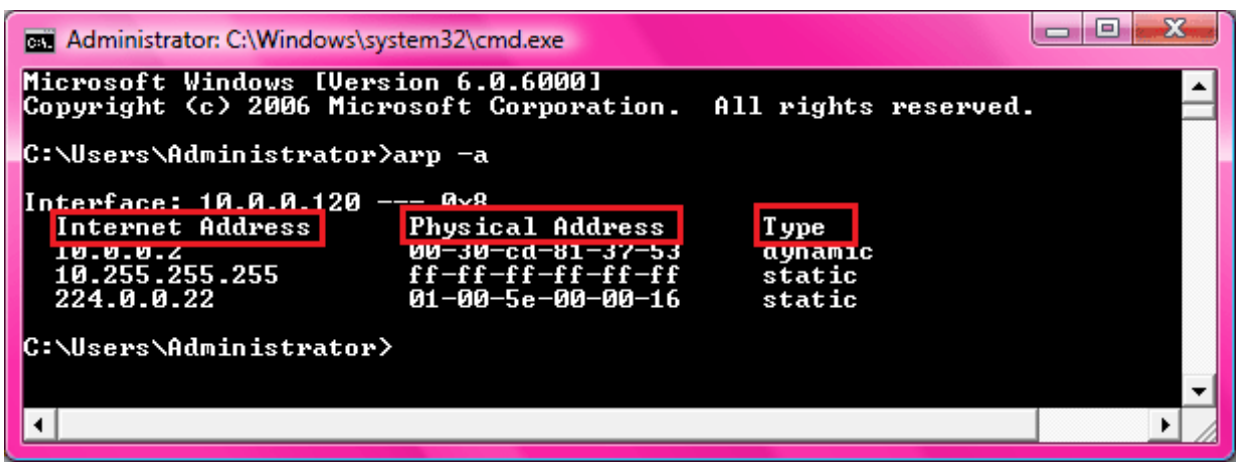
- SENDER HADDR
  - HADDR LEN bytes for the sender's hardware address
- SENDER PADDR
  - PADDR LEN bytes for the sender's protocol address
- TARGET HADDR
  - HADDR LEN bytes for the target's hardware address
- TARGET PADDR
  - PADDR LEN bytes for the target's protocol address



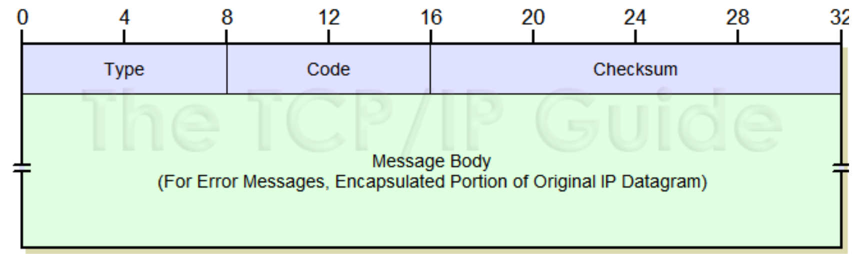
# Notes

- ARP is encapsulated in Ethernet frame
  - In this case Ethernet type will be ARP
- Sending ARP for each message is not efficient
  - Thus, cache is used (create a small local table)
  - The cache is checked before broadcasting the request

Cashed Results:



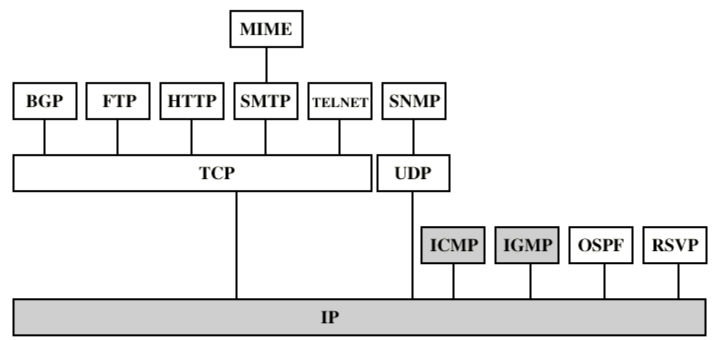
# Internet Control Message Protocol (ICMP)

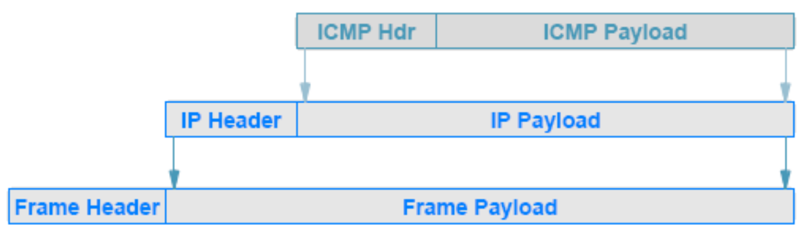



- ICMP error messages are used by routers and hosts to tell a device that sent a datagram about problems encountered in delivering it
  - It is used to test the network
  - ICMP messages are encapsulated in the IP packet
  - ICMP has many message types
    - Two basic categories: Report Error or Obtain Information

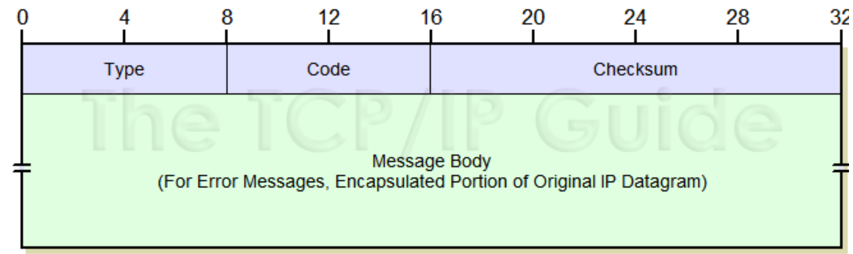
| Number | Type | Purpose |
|---|---|---|
| 0 | Echo Reply | Used by the ping program |
| 3 | Dest. Unreachable | Datagram could not be delivered |
| 5 | Redirect | Host must change a route |
| 8 | Echo | Used by the ping program |
| 11 | Time Exceeded | TTL expired or fragments timed out |
| 12 | Parameter Problem | IP header is incorrect |
| 30 | Traceroute | Used by the traceroute program |

Code field is used for subtypes



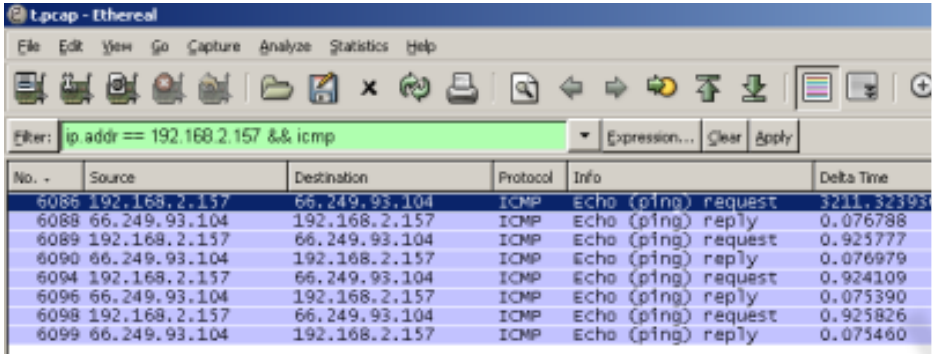



# ICMP Protocol Details




- Type: 8 bits
- Code: 8 bits
- ICMP Header Checksum: 16 bits
  - The 16-bit one's complement of the one's complement sum of the ICMP message (6 → -6)
- Data: Variable length
  - Contains the data specific to the message type indicated by the Type and Code fields.

# Encapsulated ICMP – Type 8



# Encapsulated ICMP – Type 8

```
Internet Protocol, Src: 192.168.2.157 (192.168.2.157), Dst: 66.249.93.104 (66.249.93.104)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 60
    Identification: 0x0fd3 (4051)
    Flags: 0x00
        0... = Reserved bit: Not set
        .0.. = Don't fragment: Not set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: ICMP (0x01)
    Header checksum: 0xc747 [correct]
        Good: True
        Bad : False
    Source: 192.168.2.157 (192.168.2.157)
    Destination: 66.249.93.104 (66.249.93.104)
```

```
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x475c [correct]
    Identifier: 0x0300
    Sequence number: 0x0300
    Data (32 bytes)
```

8 Bytes

Type. 8 bits. Set to 8.
Code. 8 bits. Cleared to 0.
ICMP Header Checksum. 16 bits.
Identifier. 16 bits. This field is used to help match echo requests to the associated reply. It may be cleared to zero.
Sequence number. 16 bits. This field is used to help match echo requests to the associated reply. It may be cleared to zero.
Data. Variable length. Implementation specific data (depends on the type)

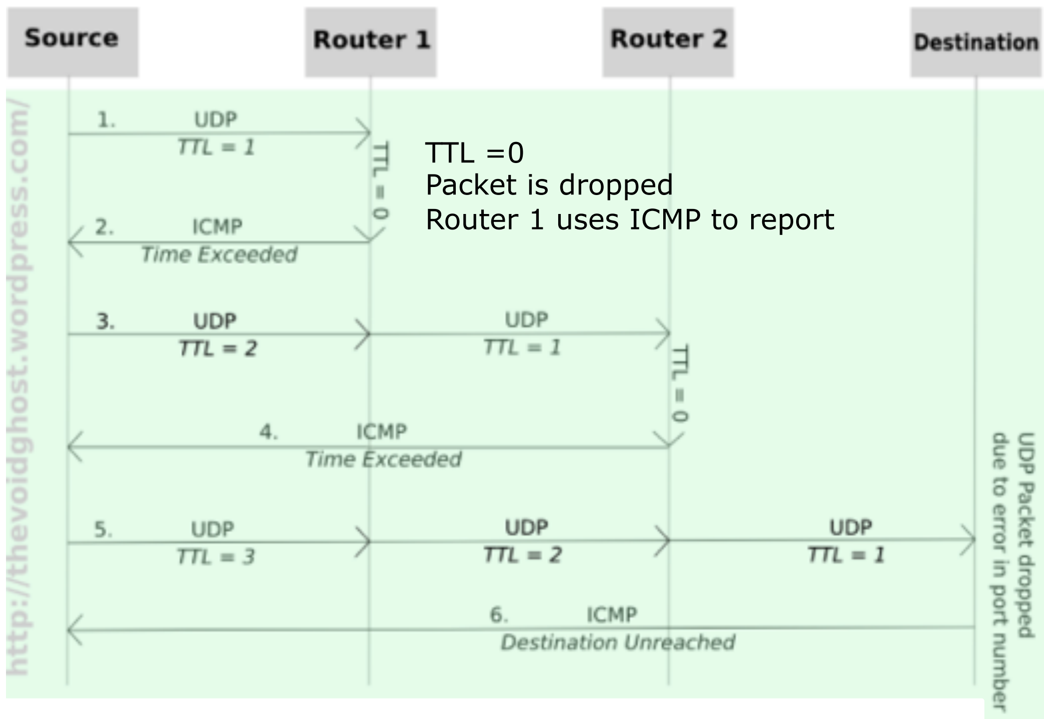
# ICMP

| Number | Type | Purpose |
| --- | --- | --- |
| 0 | Echo Reply | Used by the ping program |
| 3 | Dest. Unreachable | Datagram could not be delivered |
| 5 | Redirect | Host must change a route |
| 8 | Echo | Used by the ping program |
| 11 | Time Exceeded | TTL expired or fragments timed out |
| 12 | Parameter Problem | IP header is incorrect |
| 30 | Traceroute | Used by the traceroute program |

- ICMP messages do not have special priority
  - They are forwarded like any other datagram, with one minor exception:
    - If an ICMP error message causes an error no error message is sent
- The reason should be clear:
  - the designers wanted to avoid the Internet becoming congested carrying error messages about error messages

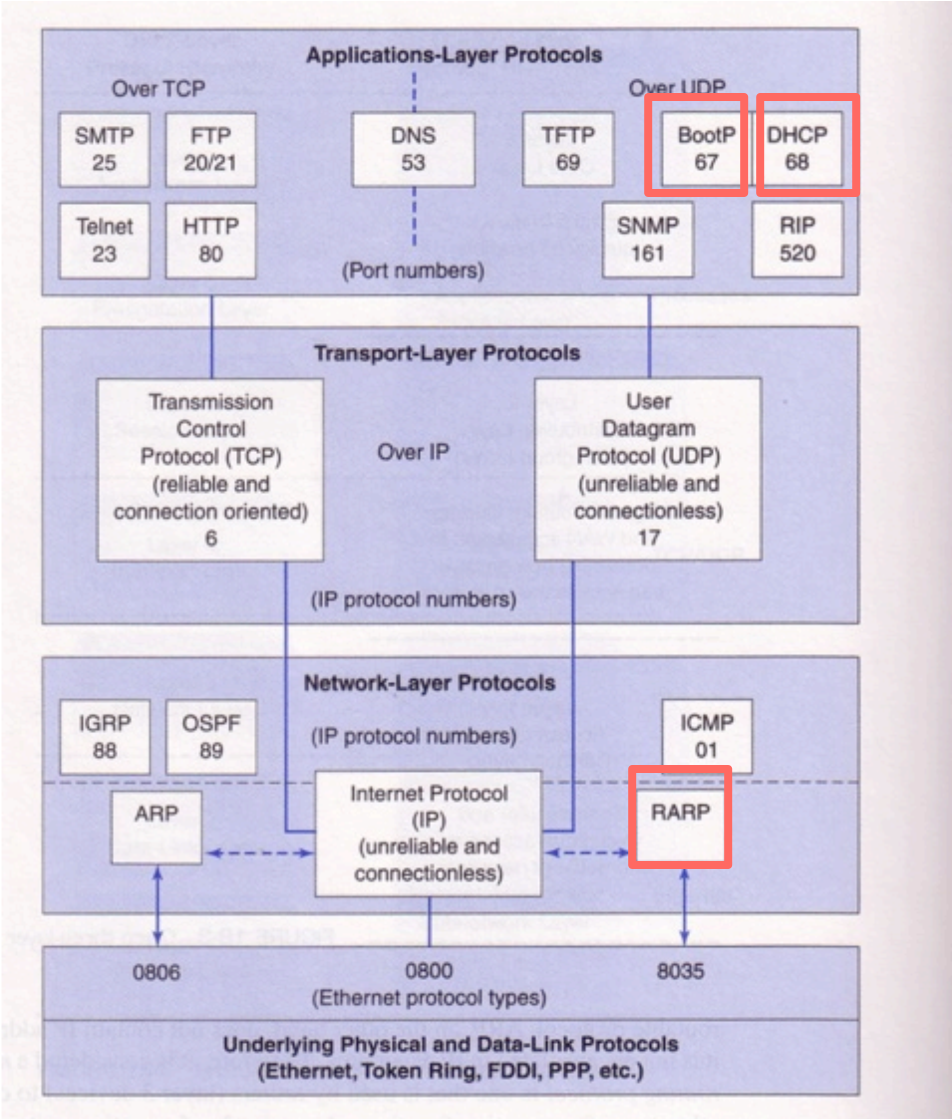
# ICMP & Traceroute

- Traceroute is a program that shows you route taken by packets through a network
- sends a UDP packet to the destination taking advantage of ICMP's messages




http://2buntu.com/articles/1203/traceroute-how-does-it-work/

# Protocol Software and Configuration



# Protocol Software and Configuration

- Once a host or router has been powered on, OS is started and the protocol software is initialized How does the protocol software in a host or router begin operation?
- For a router, the configuration manager must specify initial values for items such as
  - the IP address for each network connection
  - the protocol software to run
  - initial values for a forwarding table
  - the configuration is saved, and a router loads the values during startup
- Host configuration usually uses a two-step process, known as bootstrapping
  - A protocol was invented to allow a host to obtain multiple parameters with a single request, known as the Bootstrap Protocol (BOOTP)
    - Examples of such parameters: IP address; MASK; Local DNS
  - Currently, DHCP is used to take care of most configuration needed

# RARP and BOOTP

- Reverse ARP translates the Ethernet address to IP address
  - A diskless machine when it is booting can ask: My MAC is 12.03.23.43.23.23; what is my IP?
- RARP broadcasts the question (destination address is all one)
  - Not passed through the router!
- Major issue: Each LAN needs a RARP server!
- Bootstrap protocol uses UDP and forwards over routers
  - BOOTP is usually used during the bootstrap process - when a computer is starting up
  - Mapping must be done manually in each router!

# Dynamic Host Configuration Protocol

- DHCP allows a computer to join a new network and obtain an IP address automatically
  - The concept has been termed plug-and-play networking
- Replaces BOOTP and RARP
  - Extension of BOOTP data format
- DHCP uses UDP
  - UDP port 67 for sending data to the server
  - UDP port 68 for data to the client
- DHCP communications are connectionless in nature

# Dynamic Host Configuration Protocol

- DHCP has four basic phases:
  - IP discovery, IP lease offer, IP request, and IP lease acknowledgement
- First DHCP server must be discovered
  - The client broadcasts messages on the physical subnet to discover available DHCP servers
- IP Lease Offer
  - When a DHCP server receives an IP lease request from a client, it reserves an IP address for the client and extends an IP lease offer by sending a DHCP OFFER message to the client

| No. | Len | Time | Source | Destination | Protocol | Info |
|---|---|---|---|---|---|---|
| 1 | 314 | 0.000000 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Discover - Transaction ID ( |
| 2 | 342 | 0.000295 | 192.168.0.1 | 192.168.0.10 | DHCP | DHCP Offer - Transaction ID ( |
| 3 | 314 | 0.070031 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Request - Transaction ID ( |
| 4 | 342 | 0.070345 | 192.168.0.1 | 192.168.0.10 | DHCP | DHCP ACK - Transaction ID ( |

http://wiki.wireshark.org/DHCP

# Dynamic Host Configuration Protocol

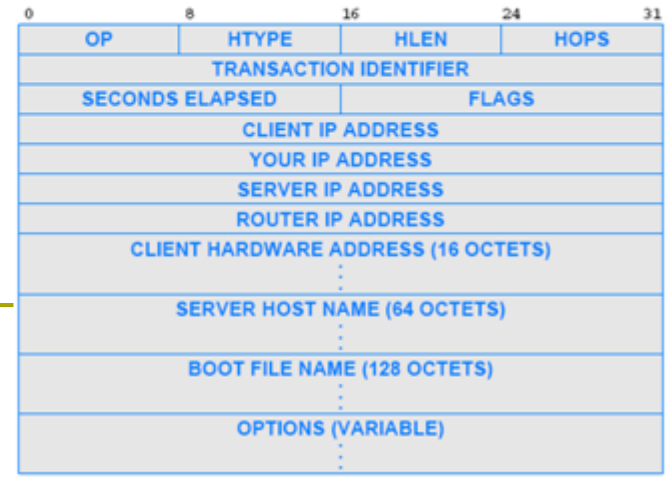
- A client can receive multiple offers from difference servers
  - Thus, it must request an IP address
- DHCP sends a Request packet to the DHCP server and receives a DHCP Reply
  - What is the IP address for this MAC?
  - It can also request its previous IP address!
- Even when an IP address is assigned, how long is it good for?
  - Before the IP address is removed find another IP address….called Leasing
- When the DHCP server receives the Request from the client, the configuration process enters its final phase
  - a DHCPACK (ACK) packet is sent to the client

| No. | Len | Time | Source | Destination | Protocol | Info |
|---|---|---|---|---|---|---|
| 1 | 314 | 0.000000 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Discover - Transaction ID ( |
| 2 | 342 | 0.000295 | 192.168.0.1 | 192.168.0.10 | DHCP | DHCP Offer - Transaction ID ( |
| 3 | 314 | 0.070031 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Request - Transaction ID ( |
| 4 | 342 | 0.070345 | 192.168.0.1 | 192.168.0.10 | DHCP | DHCP ACK - Transaction ID ( |

# DHCP

- DHCP includes several important details that optimize performance, such as the following:
- Recovery from loss or duplication
  - DHCP is designed to insure that missing or duplicate packets do not result in misconfiguration
  - If no response is received
    - a host retransmits its request (remember DHCP uses UDP!)
  - If a duplicate response arrives
    - a host ignores the extra copy
- Caching of a server address
  - once a host finds a DHCP server
    - the host caches the server's address
- Avoidance of synchronized flooding
  - DCHP takes steps to prevent synchronized requests
  - Synchronization can occur when all computers boot up at the same time!

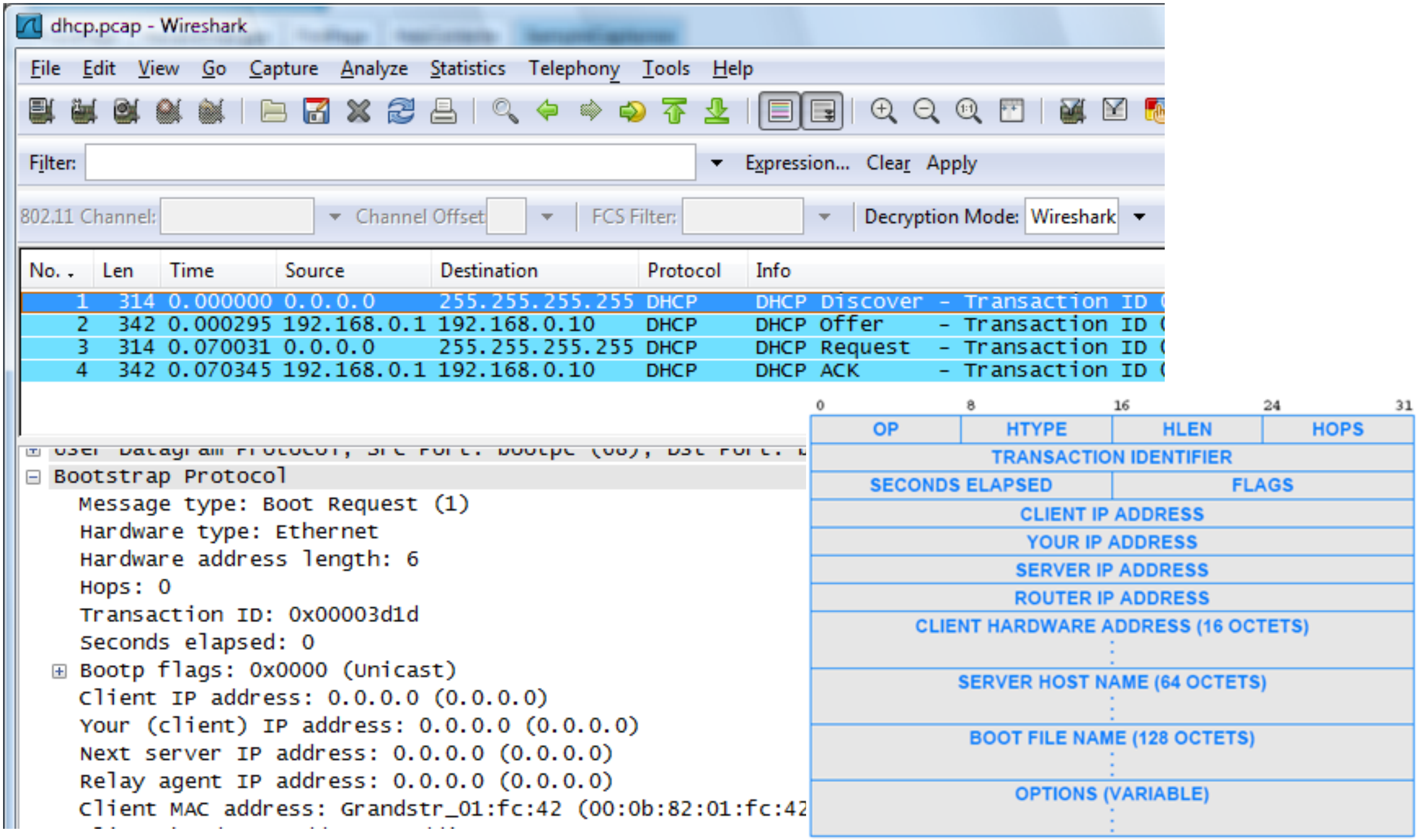
# DHCP Format




- DHCP adopted a slightly modified version of the BOOTP message format
- DHCP message format
  - OP specifies whether the message is a Request or a Response
  - HTYPE and HLEN fields specify the network hardware type and the length of a hardware address
  - FLAGS specifies whether it can receive broadcast or directed replies
  - HOPS specifies how many hops to the server
  - TRANSACTION IDENTIFIER provides a value that a client can use to determine if an incoming response matches its request
  - SECONDS ELAPSED specifies how many seconds have elapsed since the host began to boot

  - See next slide for Example….→

# DHCP Phases


dhcp.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply
802.11 Channel: Channel Offset: FCS Filter: Decryption Mode: Wireshark
No. Len Time Source Destination Protocol Info
1 314 0.000000 0.0.0.0 255.255.255.255 DHCP DHCP Discover - Transaction ID
2 342 0.000295 192.168.0.1 192.168.0.10 DHCP DHCP Offer - Transaction ID
3 314 0.070031 0.0.0.0 255.255.255.255 DHCP DHCP Request - Transaction ID
4 342 0.070345 192.168.0.1 192.168.0.10 DHCP DHCP ACK - Transaction ID
Bootstrap Protocol
Message type: Boot Request (1)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x00003d1d
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Grandstr_01:fc:42 (00:0b:82:01:fc:42
0
8
16
24
31
OP
HTYPE
HLEN
HOPS
TRANSACTION IDENTIFIER
SECONDS ELAPSED
FLAGS
CLIENT IP ADDRESS
YOUR IP ADDRESS
SERVER IP ADDRESS
ROUTER IP ADDRESS
CLIENT HARDWARE ADDRESS (16 OCTETS)
SERVER HOST NAME (64 OCTETS)
BOOT FILE NAME (128 OCTETS)
OPTIONS (VARIABLE)

| No. | Len | Time | Source | Destination | Protocol | Info |
|---|---|---|---|---|---|---|
| 1 | 314 | 0.000000 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Discover - Transaction ID |
| 2 | 342 | 0.000295 | 192.168.0.1 | 192.168.0.10 | DHCP | DHCP Offer - Transaction ID |
| 3 | 314 | 0.070031 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Request - Transaction ID |
| 4 | 342 | 0.070345 | 192.168.0.1 | 192.168.0.10 | DHCP | DHCP ACK - Transaction ID |

# Request and ACK

**DHCPREQUEST**

| UDP Src=0.0.0.0 sPort=68 Dest=255.255.255.255 dPort=67 | | | |
|---|---|---|---|
| **OP** | **HTYPE** | **HLEN** | **HOPS** |
| 0x01 | 0x01 | 0x06 | 0x00 |
| **XID** | | | |
| 0x3903F326 | | | |
| **SECS** | | **FLAGS** | |
| 0x0000 | | 0x0000 | |
| **CIADDR (Client IP Address)** | | | |
| 0x00000000 | | | |
| **YIADDR (Your IP Address)** | | | |
| 0x00000000 | | | |
| **SIADDR (Server IP Address)** | | | |
| 0xC0A80101 | | | |
| **GIADDR (Gateway IP Address)** | | | |
| 0x00000000 | | | |
| **CHADDR (Client Hardware Address)** | | | |
| 0x00053C04 | | | |
| 0x8D590000 | | | |

Client

Client uses Port 68

Assuming the client is choosing the offered IP address form the server

Server IP Address

**DHCPACK**

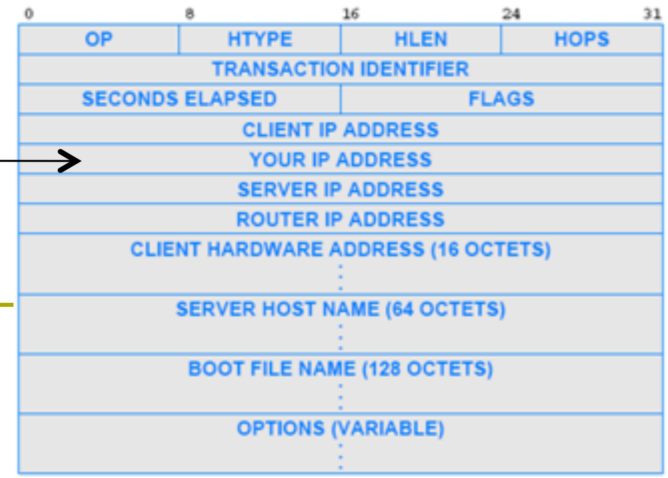
| UDP Src=192.168.1.1 sPort=67 Dest=255.255.255.255 dPort=68 | | | |
|---|---|---|---|
| **OP** | **HTYPE** | **HLEN** | **HOPS** |
| 0x02 | 0x01 | 0x06 | 0x00 |
| **XID** | | | |
| 0x3903F326 | | | |
| **SECS** | | | **FLAGS** |
| 0x0000 | | 0x0000 | |
| **CIADDR (Client IP Address)** | | | |
| 0x00000000 | | | |
| **YIADDR (Your IP Address)** | | | |
| 0xC0A80164 Client's NEW IP Address | | | |
| **SIADDR (Server IP Address)** | | | |
| 0xC0A80101 | | | |
| **GIADDR (Gateway IP Address switched by relay)** | | | |
| 0x00000000 | | | |
| **CHADDR (Client Hardware Address)** | | | |
| 0x00053C04 | | | |
| 0x8D590000 | | | |

Server

When the DHCP server receives the DHCPREQUEST message from the client, the configuration process enters its final phase. The acknowledgement phase involves sending a DHCPACK packet to the client.

# DHCP

Server fills it

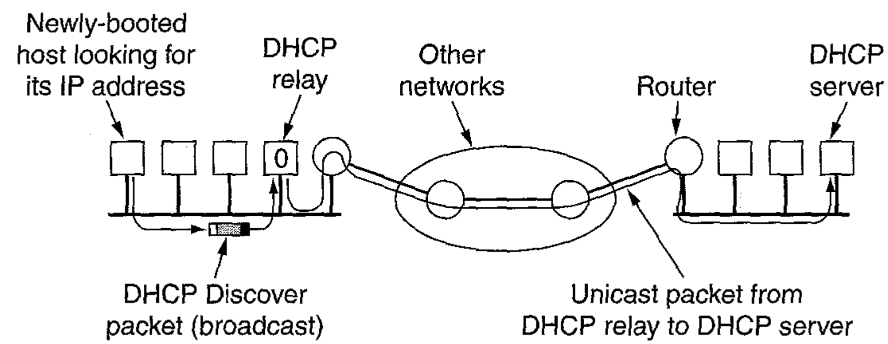



- Later fields in the message are used in a response to carry information back to the host that sent a request
  - if a host does not know its IP address, the server uses field YOUR IP ADDRESS to supply the value
  - server uses fields SERVER IP ADDRESS and SERVER HOST NAME to give the host information about the location of a server
  - ROUTER IP ADDRESS contains the IP address of a default router
- DHCP allows a computer to negotiate to find a boot image
  - The computer is boot up, request and OS
  - the host fills in field BOOT FILE NAME with a request
  - The DHCP server does not send an image
    - The host uses TFTP

# Early Release

- The user can end the lease through a process called early lease termination or lease release
- This is a very simple, unidirectional communication
  - The client sends a special DHCPRELEASE message unicast to the server that holds its current lease
  - The server then records the lease as having been ended
  - It does not need to reply back to the client (no ACK)
- Client can just assume that the lease termination has been successful
- Having clients send DHCPRELEASE to end a lease is considered a *courtesy*, rather than a requirement
- DHCP servers are designed to handle the case where a client "disappears" without formally ending an existing lease
  - Sending a DHCPRELEASE is clearly more efficient, however!

# Indirect DHCP Server Access Through a Relay



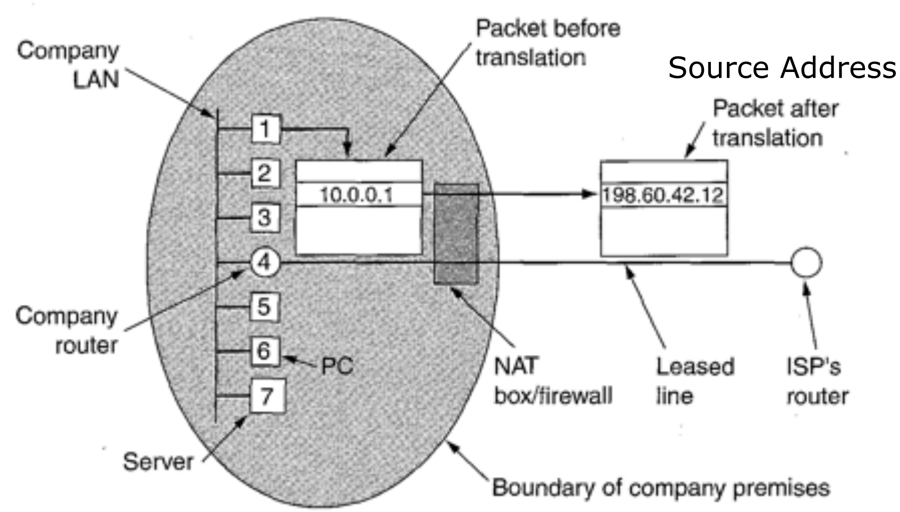

- DHCP broadcasts on the local network to find a server
- DHCP does not require each individual network to have a server
  - Instead, a DHCP relay agent forwards requests and responses between a client and the server
- At least one relay agent must be present on each network
  - The relay agent must be configured with the address of the appropriate DHCP server
- When the DHCP server responds
  - The relay agent forwards the response to the client

# Network Address Translation (NAT)

- Addresses are growing! What is the solution?
  - Use IPV6
  - Use NAT
- NAT:
  - Allows using one IP address per company
  - Internally new addresses can be added!
- How?

# NAT Operation

- IP reserved addresses
  - 10.x.y.z
  - 172.16.x.y
  - 192.168.x.y
- Receiving a packet from the Internet
  - Sender
    - Add IP address
    - TCP will have the destination port (0-1023) – standard port group
    - The port determines which server on the remote (destination) side to process the packet
  - NAT box:
    - Using the PORT address in TCP, change the IP address to a designated address (10.0.01)
- Sending a packet into the Internet
  - NAT box:
    - Changes  the source address from 10.0.0.1 to 198.60.42.12



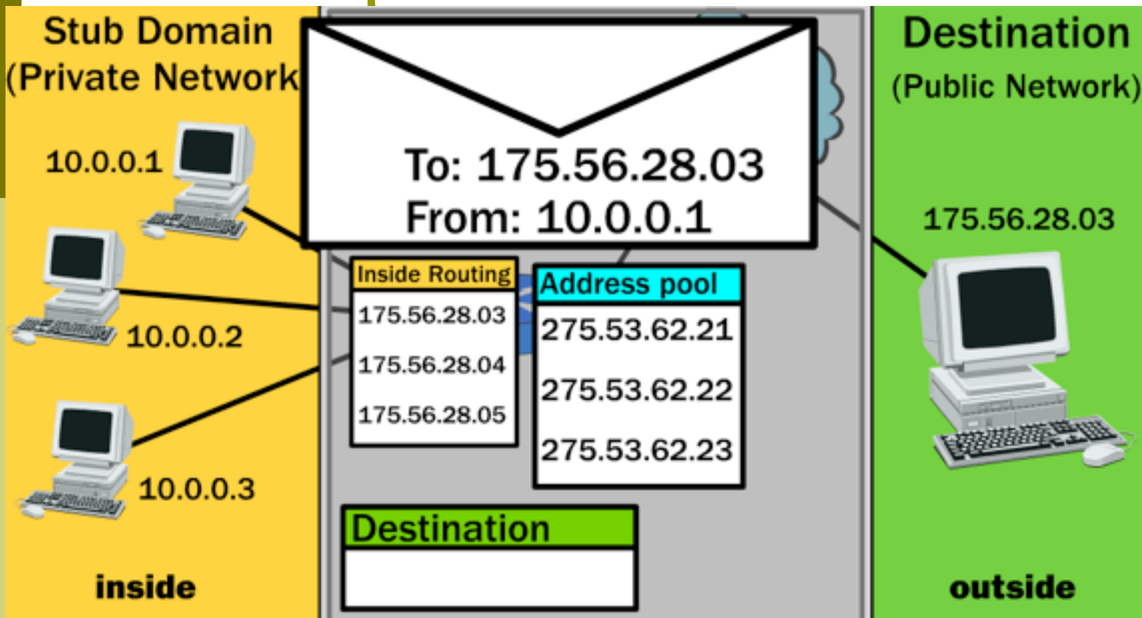

- 198.60.42.12

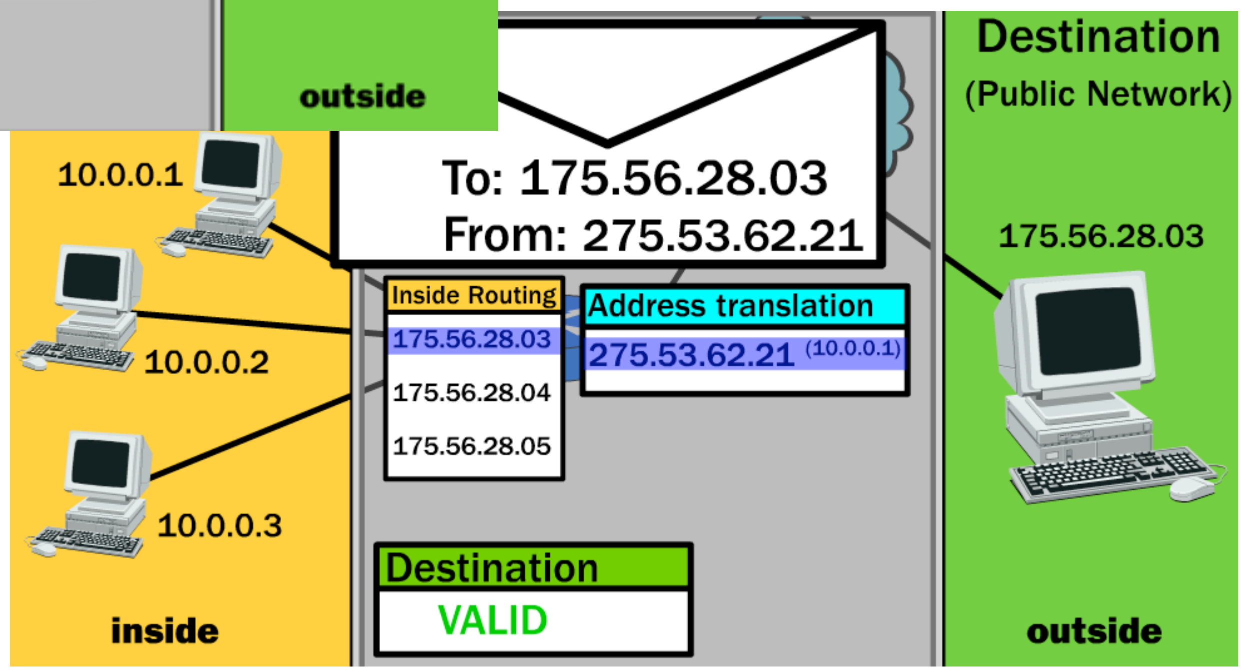
# NAT Issues…

- Addresses are not unique: many 10.0.0.1!
- NAT controls the incoming and outgoing packets – reliability!
- NAT accesses TCP and IP layers – layers should work independent of one another
- NAT only allows TCP/IP or UDP/IP
- NAT does not support applications which insert the IP address in the body (FTP or H.323)

DEMO
http://www3.rad.com/networks/2005/prvt-nat/main2.htm

# Example:


Stub Domain
(Private Network
10.0.0.1
10.0.0.2
10.0.0.3
inside
To: 175.56.28.03
From: 10.0.0.1
Inside Routing
175.56.28.03
175.56.28.04
175.56.28.05
Address pool
275.53.62.21
275.53.62.22
275.53.62.23
Destination
Destination
(Public Network)
175.56.28.03
outside


10.0.0.1
10.0.0.2
10.0.0.3
inside
To: 175.56.28.03
From: 275.53.62.21
Inside Routing
175.56.28.03
175.56.28.04
175.56.28.05
Address translation
275.53.62.21 (10.0.0.1)
Destination
VALID
Destination
(Public Network)
175.56.28.03
outside

# Remember…

- This is My MAC; what is my IP address? RARP / DHCP
- This is the destination host name, what it is IP address? DNS Server
- This is the IP address, what is your MAC address? ARP

# References

- Tanenbaum
- Tomasi Text Book
- Comer Text book