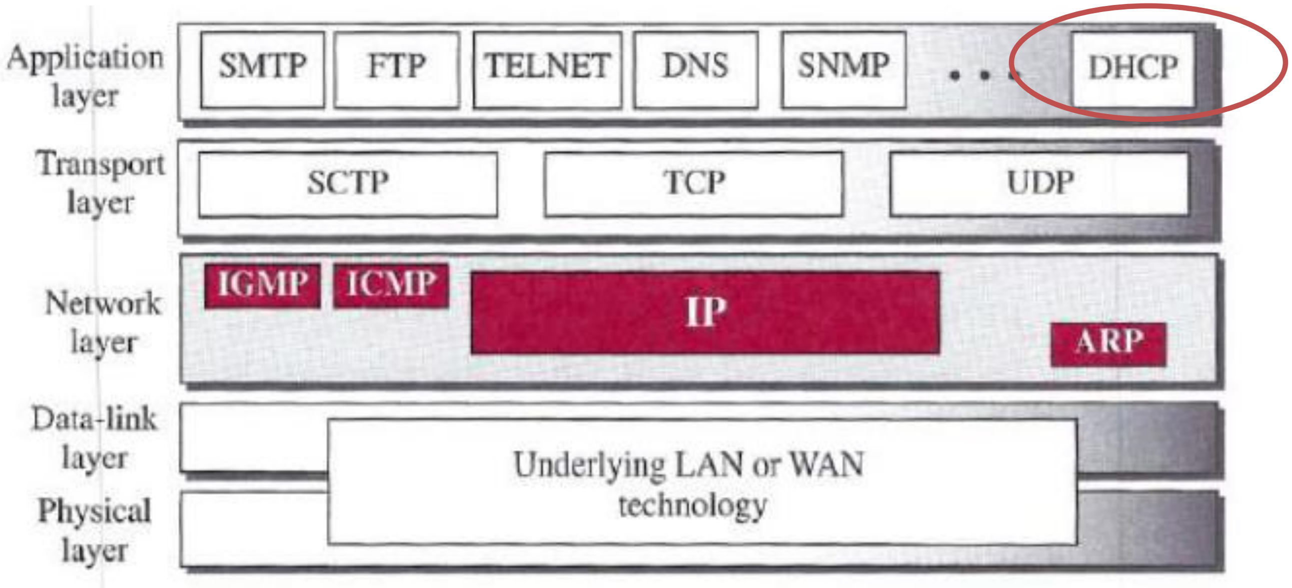
CS348: Computer Networks

DHCP, NAT, ICMP

Dr. Manas Khatua
Assistant Professor
Dept. of CSE, IIT Guwahati
E-mail: manaskhatua@iitg.ac.in

DHCP in TCP/IP Suite

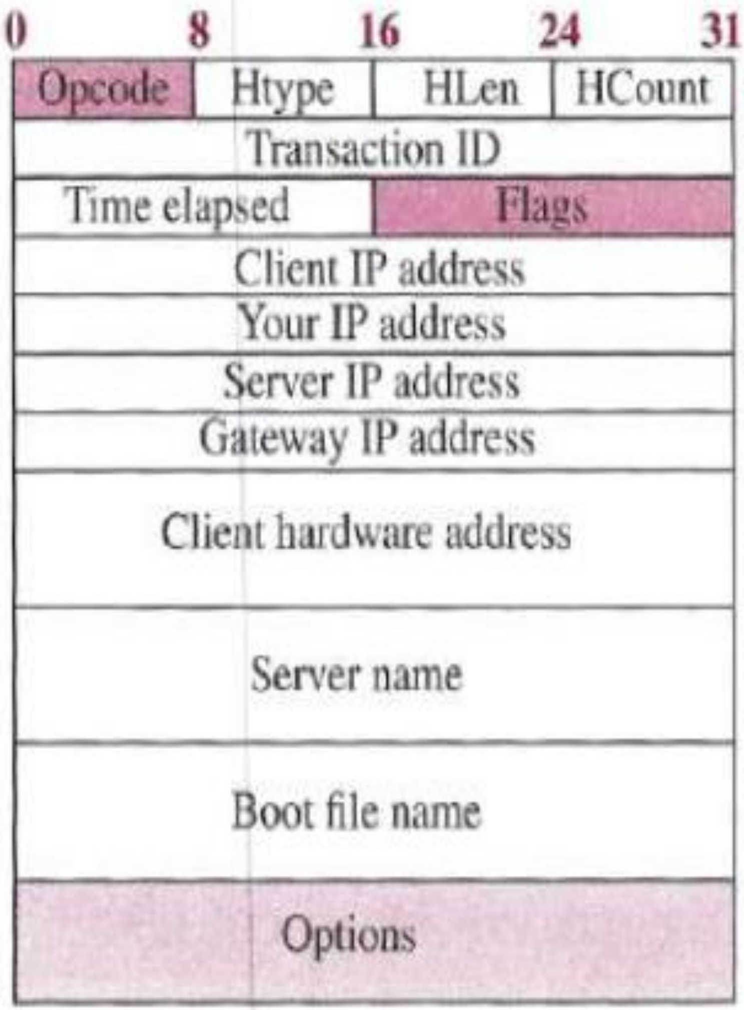


Dynamic Host Configuration Protocol (DHCP)

- is an application-layer program,
- using the client-server paradigm,
- actually helps TCP/IP at the network layer.
- Automatically assigns IP addresses to the host and routers.
- Ideally, every network should have at least one DHCP server

Earlier versions of DHCP was BOOTP (Bootstrap Protocol)

DHCP Frame Format



Fields:

Opcode: Operation code, request (1) or reply (2)

Htype: Hardware type (Ethernet, ...)

HLen: Length of hardware address

HCount: Maximum number of hops the packet can travel

Transaction ID: An integer set by the client and repeated by the server

Time elapsed: The number of seconds since the client started to boot

Flags: First bit defines unicast (0) or multicast (1); other 15 bits not used

Client IP address: Set to 0 if the client does not know it

Your IP address: The client IP address sent by the server

Server IP address: A broadcast IP address if client does not know it

Gateway IP address: The address of default router

Server name: A 64-byte domain name of the server

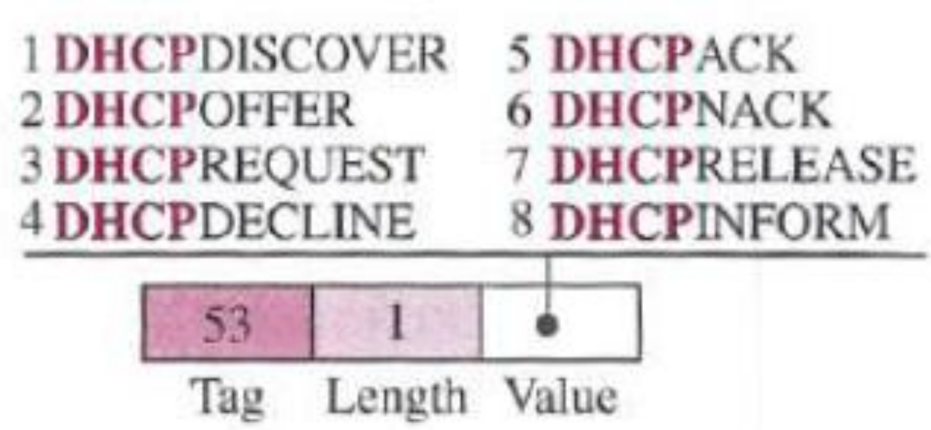
Boot file name: A 128-byte file name holding extra information

Options: A 64-byte field with dual purpose described in text

Options Field

Options: 64 Byte field with dual purpose

- 1 Byte Tag/ Code; specifies the option type.
- 1 Byte Length; specifies the number of bytes in this particular option
- 0-58 Byte value; specifies the data being sent
- 4 Byte magic cookie (99.130.83.99); to identify the information as vendor-independent option fields.



DHCP Scenario

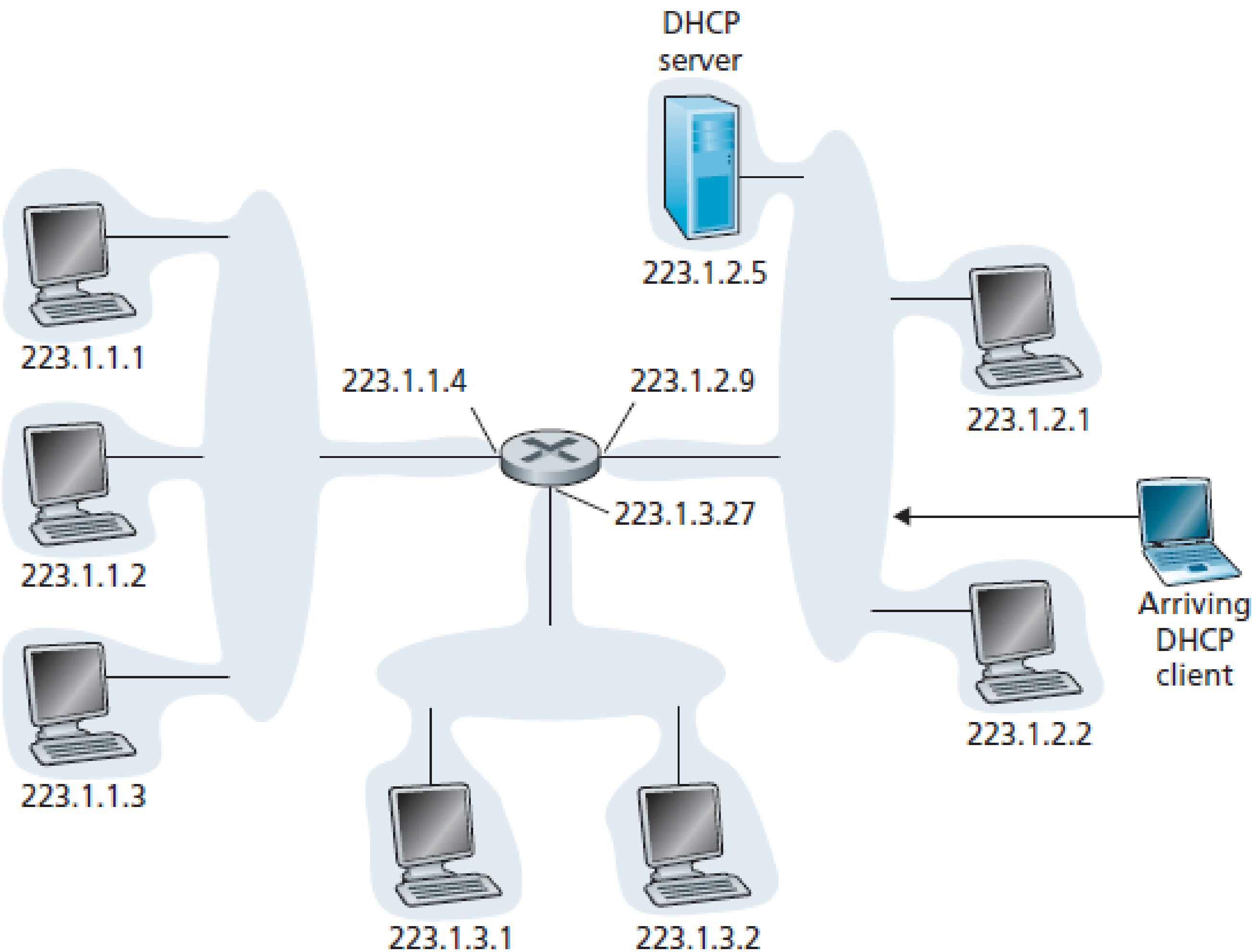


Figure 4.20 ♦ DHCP client-server scenario

DHCP Steps

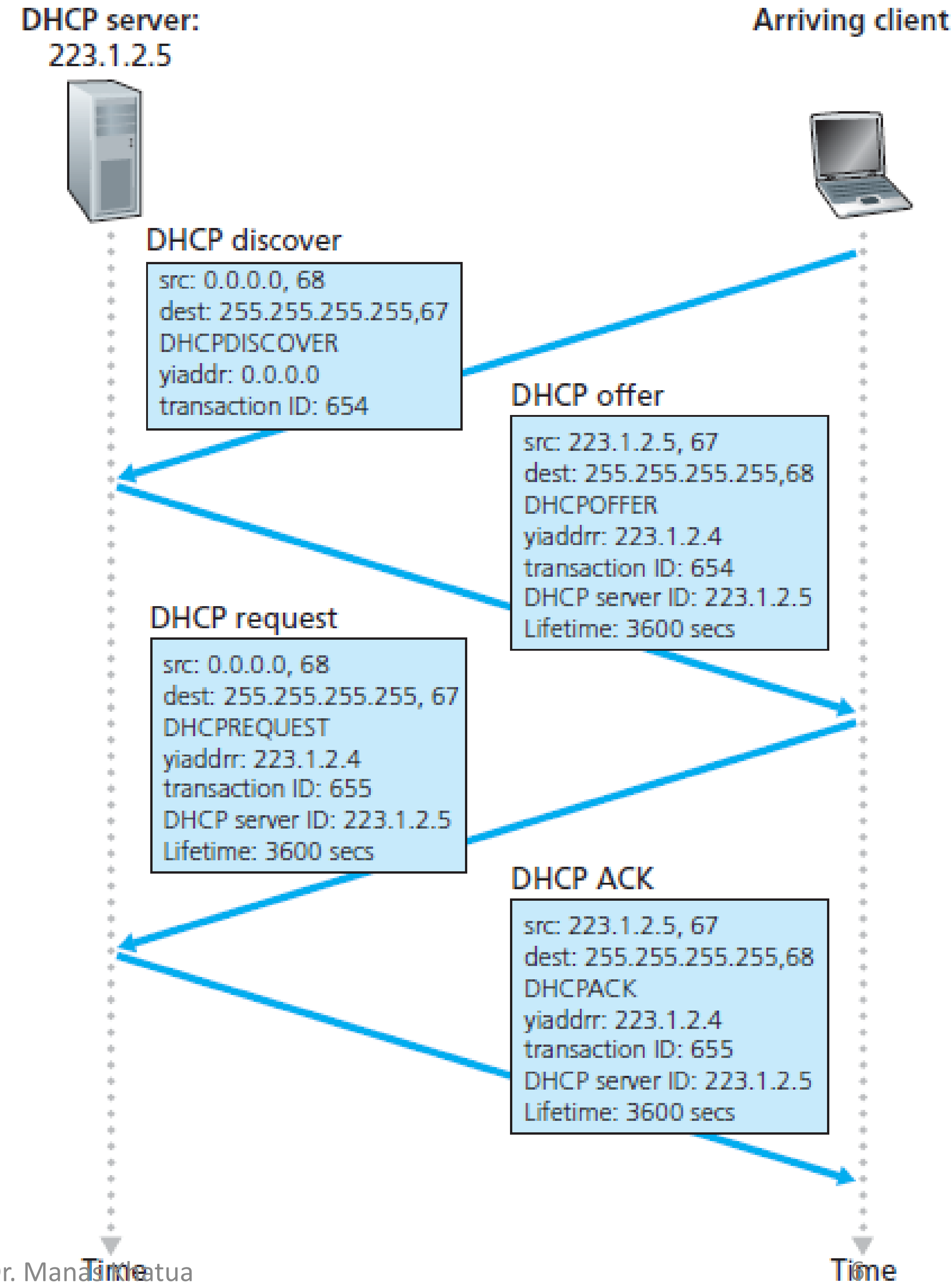
<u>4 step process</u>

1. *DHCP server discover*
   UDP packet to port 67.
   This host IP: 0.0.0.0, Port: 68
   Broadcast IP: 255.255.255.255
   Transaction ID: 654 (set by client)

2. *DHCP server offer(s)*
   Transaction ID: 654
   Your IP: 223.1.2.4
   Mask, DHCP server IP,
   Lifetime: 3600 sec

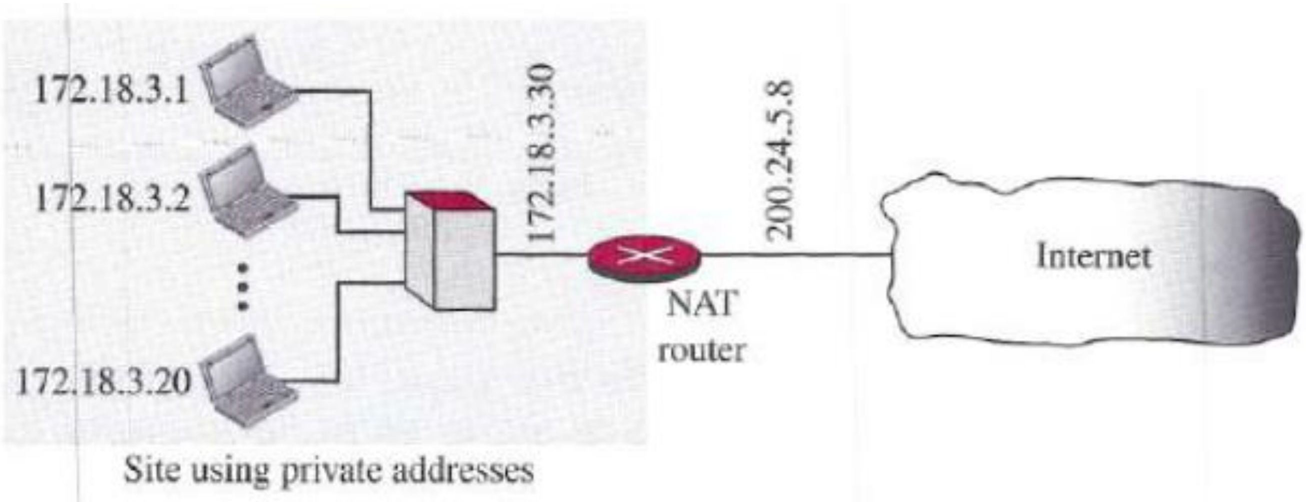
3. *DHCP request*
   Select one offer and request to grant

4. *DHCP ACK*
   Server confirms the request



Network Address Translation (NAT)

- Problem: after a period, business grows or the household needs a larger range of IP
- Expensive Naïve Solution: get more IP from the ISP
- Better Solution: NAT.
  - use a set of private addresses for internal communication, and
  - a set of global addresses (at least one) for communication with the world.



NAT Operations

Private IP Addresses: 10.0.0.0/8, *172.16.0.0/12, 192.168.0.0/16,* and 169.254.0.0/16

NAT translation table	
WAN side	**LAN side**
138.76.29.7, 5001	10.0.0.1, 3345
. . .	. . .

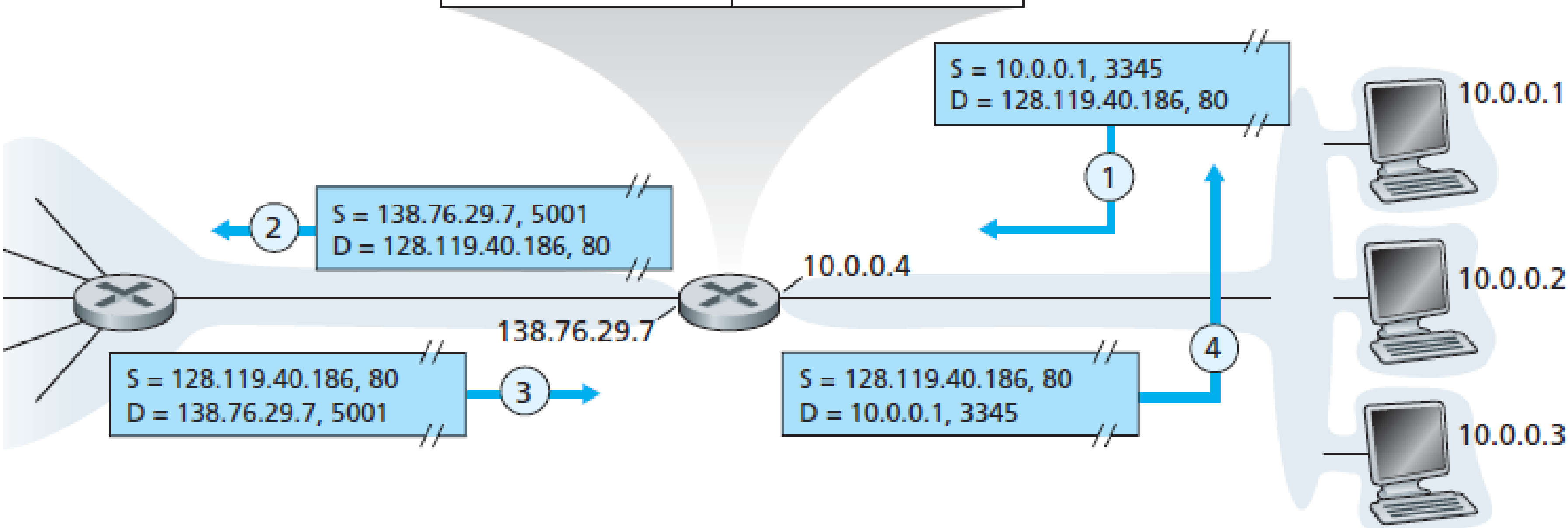


Figure 4.22 ♦ Network address translation

NAT

- The NAT-enabled router does not *look* like a router to the outside world
- Instead the NAT router behaves to the outside world as a *single* device with a *single* IP
- The NAT-enabled router is hiding the details of the home network from the outside world.
- The router runs a DHCP server to provide addresses to computers within the NAT-DHCP-router-controlled home network's address space.
- NAT has enjoyed widespread deployment. It has few objections:
  - port numbers are meant to be used for addressing processes, not for addressing hosts.
  - Routers are supposed to process packets only up to layer 3, not up to layer 4
  - the NAT protocol violates the so-called end-to-end argument; that is, hosts should be talking directly with each other, without interfering nodes modifying IP addresses and port numbers.
  - we should use IPv6 to solve the shortage of IP addresses, rather than NAT
  - another major problem with NAT is that it interferes with P2P applications
    - if Peer B is behind a NAT, it cannot act as a server and accept TCP connection from Peer A

ICMP

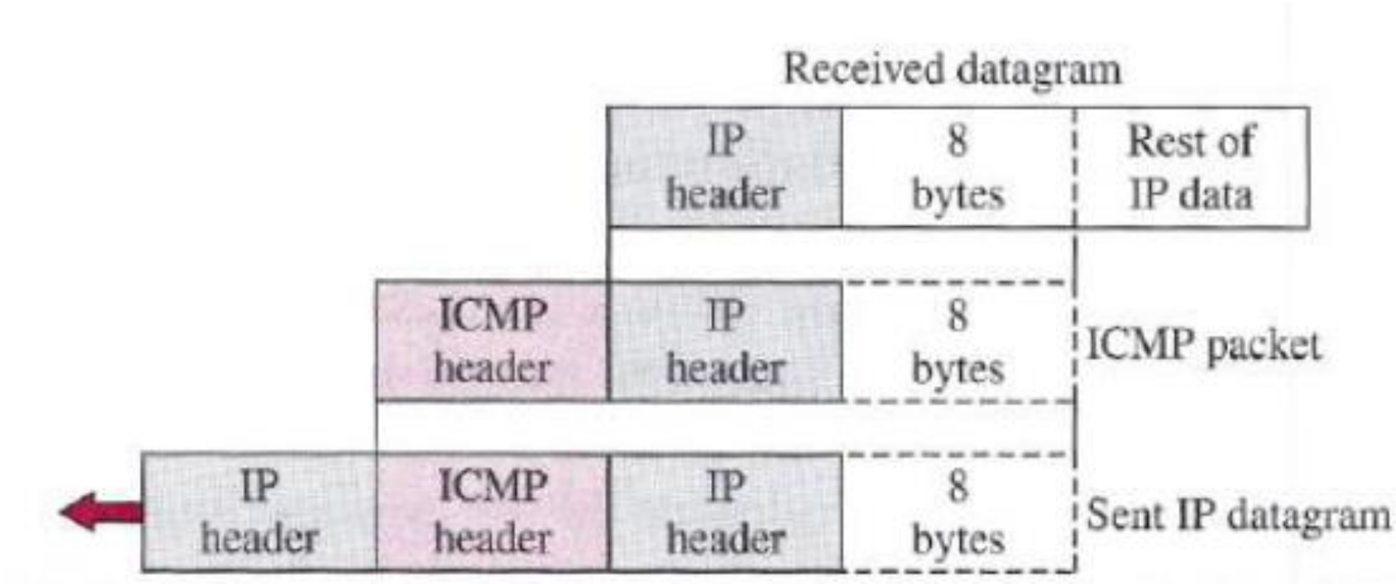
- ICMP: Internet Control Message Protocol

- What happens
  - if something goes wrong?
  - if router discards a datagram?
  - if TTL finishes?
  - if fragmentation is not permitted?

- Need a mechanism for network management

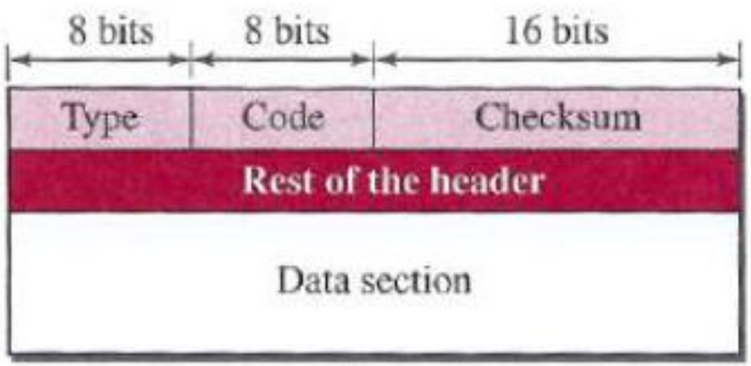
ICMP

- Its messages are not passed directly to the data-link layer as would be expected.
- Instead, the messages are first encapsulated inside IP datagrams before going to the lower layer.

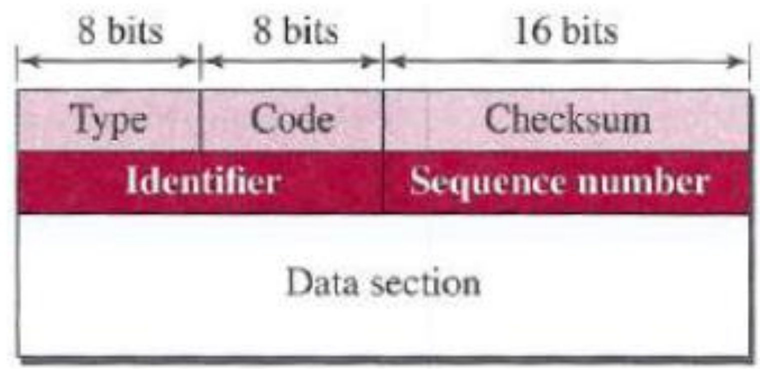


ICMP Messages

- ICMP Message size:
  - 8-byte header and
  - a variable-size data section



Error-reporting messages



Query messages

Type and code values

Error-reporting messages
03: Destination unreachable (codes 0 to 15)
04: Source quench (only code 0)
05: Redirection (codes 0 to 3)
11: Time exceeded (codes 0 and 1)
12: Parameter problem (codes 0 and 1)

Query messages
08 and 00: Echo request and reply (only code 0)
13 and 14: Timestamp request and reply (only code 0)

Error Reporting Messages

- Only error reporting; no error correction
- Messages are sent to original sources of the datagrams
- No error message for:
  - datagram carrying an ICMP error message
  - a fragmented datagram that is not the first fragment
  - a datagram having a multicast address
  - a datagram having a special address such as 127.0.0.0 or 0.0.0.0

Debugging Tools

- ***Ping:*** to find if a host is alive and responding
  - The source host sends ICMP echo-request messages;
  - the destination, if alive, responds with ICMP echo-reply messages.
  - It can calculate the round-trip time

```
$ ping auniversity.edu
PING auniversity.edu (152.181.8.3)   56 (84) bytes of data.
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=0   ttl=62   time=1.91 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=1   ttl=62   time=2.04 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=2   ttl=62   time=1.90 ms
```

Cont…

- The *traceroute* program in UNIX or *tracert* in Windows can be used to trace the path of a packet from a source to the destination.
  - It can find the IP addresses of all the routers that are visited along the path
  - It takes help of ICMP error reporting messages

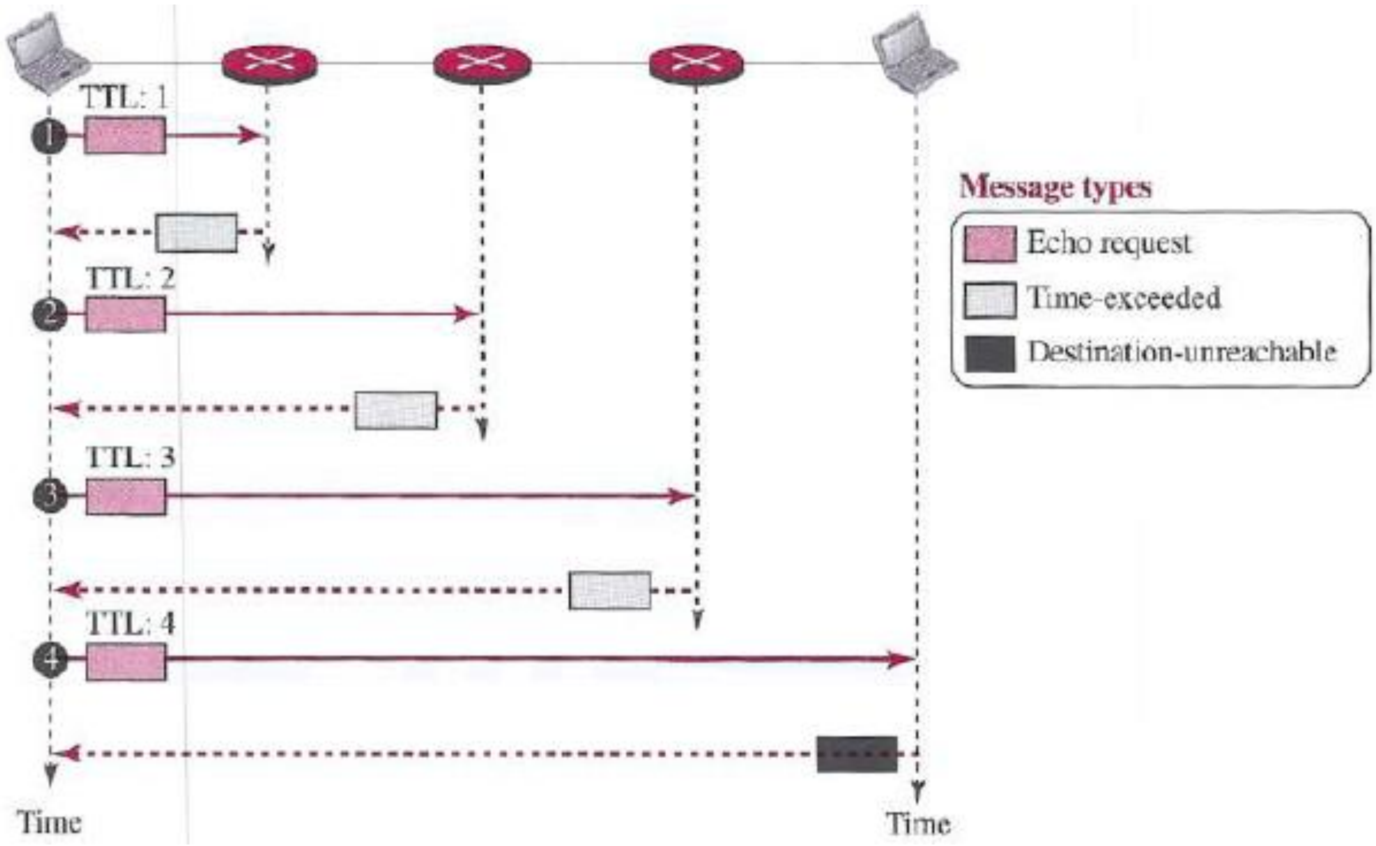
$ *traceroute* printers.com

traceroute to printers.com (13.1.69.93), 30 hops max, 38-byte packets

1 route.front.edu	(153.18.31.254)	0.622 ms	0.891 ms	0.875 ms
2 ceneric.net	(137.164.32.140)	3.069 ms	2.875 ms	2.930 ms
3 satire.net	(132.16.132.20)	3.071 ms	2.876 ms	2.929 ms
4 alpha.printers.com	(13.1.69.93)	5.922 ms	5.048 ms	4.922 ms

Cont…

- The *traceroute* application program is encapsulated in a UDP user datagram, but *traceroute* intentionally uses a port number that is not available at the destination.



Thanks!