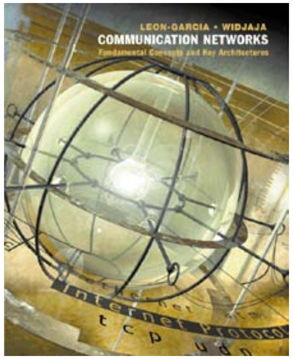
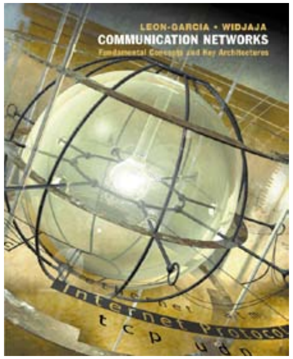
Chapter 8
Communication Networks and Services



The TCP/IP Architecture
The Internet Protocol
Internet Addressing
Address Resolution protocol
Internet Control Message Prototocol

Chapter 8

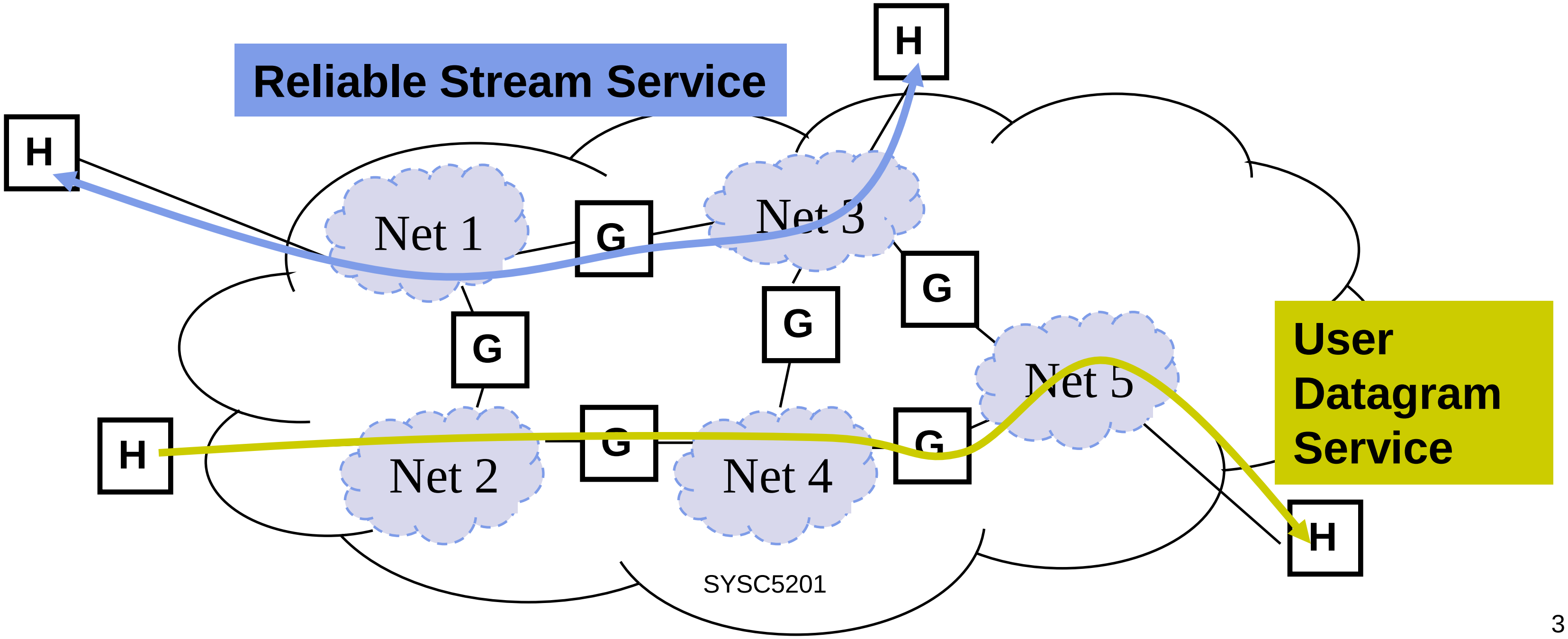
Communication Networks and Services



The TCP/IP Architecture

Why Internetworking?

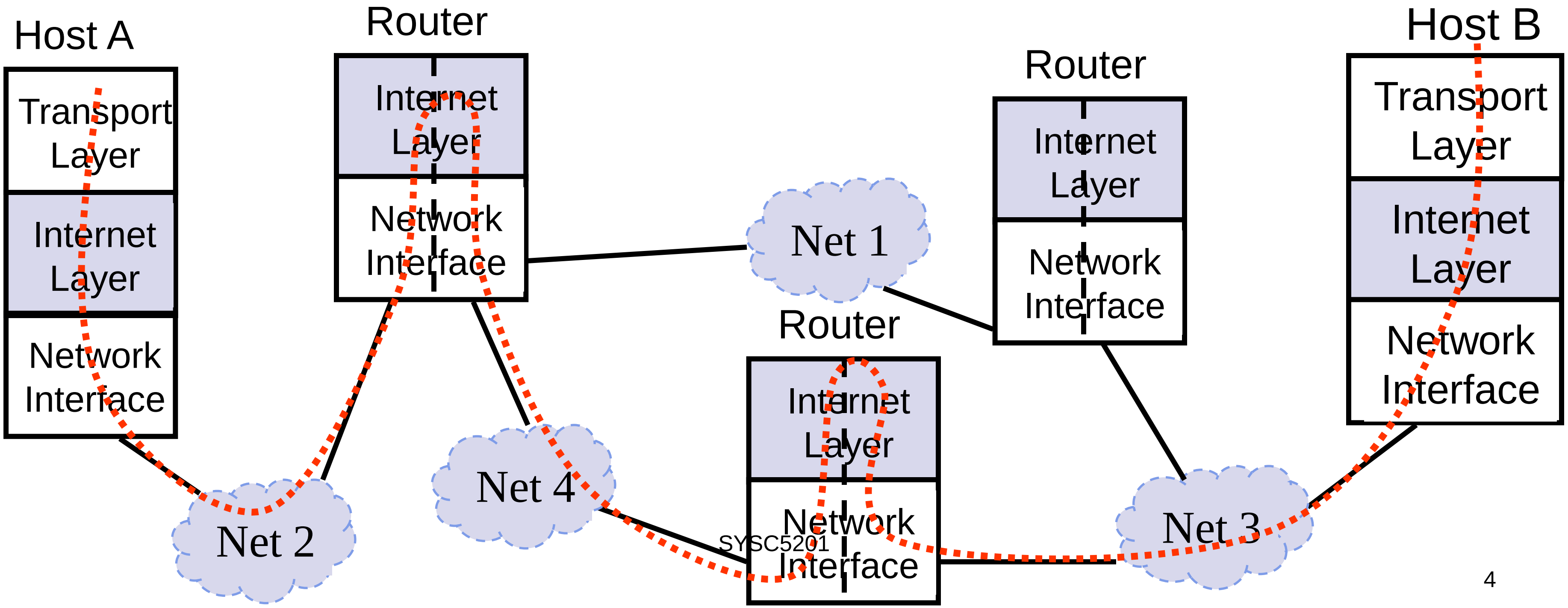
- To build a "*network of networks*" or Internet
  - operating over multiple, coexisting, different network technologies
  - providing ubiquitous connectivity through IP packet transfer
  - achieving huge economies of scale
- To provide *universal communication services*, support *distributed and diverse applications*
  - independent of underlying network technologies
  - providing common interface to user applications



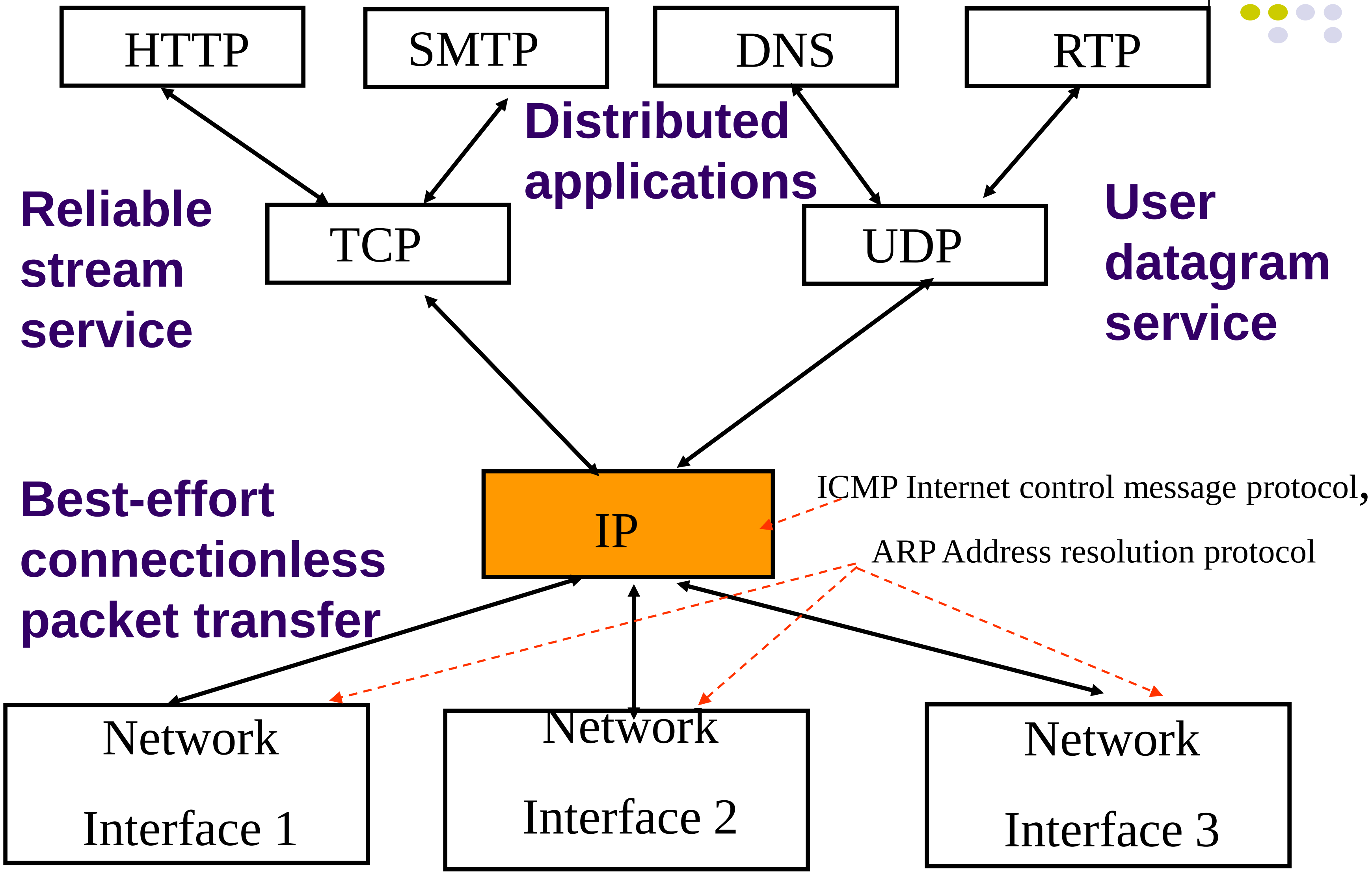
Internet Protocol Approach

- IP packets transfer information across Internet

  Host A IP → router→ router…→ router→ Host B IP
- IP layer in each router determines next hop (router)
- Network interfaces transfer IP packets across networks



TCP/IP Protocol Suite



HTTP

SMTP

DNS

RTP

Distributed applications

Reliable stream service

TCP

UDP

User datagram service

Best-effort connectionless packet transfer

IP

ICMP Internet control message protocol,

ARP Address resolution protocol

Network Interface 1

Network Interface 2

Network Interface 3

Diverse network technologies

Internet Names & Addresses

Internet Names

- Each host has a unique name
  - Independent of physical location
  - Facilitate memorization by humans
  - Depends on Domain Name
  - Domain: Network under single administrative unit
  - DNS: resolves domain name to IP address

- Host IP Name
  - Name given to host computer

- User Name
  - Name assigned to user

Internet Addresses

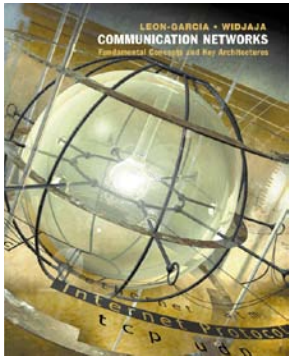
- Each host interface has globally unique *logical* 32 bit IP address
- Separate address for each physical interface to a network
- Routing decision is done based on destination IP address
- **IP address has two parts:**
  - ***netid*** and ***hostid***
  - *netid* unique (depends on Domain name)
  - *netid* facilitates routing
- Dotted Decimal Notation: byte1.byte2.byte3.byte4, e.g.,
  128.100.10.13

Physical Addresses

- LANs (and other networks) assign physical, i.e., NIC addresses to the physical interfaces to the network
- The network uses its own address to transfer packets or frames to the appropriate destination
- IP address needs to be resolved to physical address at each IP network interface to talk to data link layer
  - Q: In Ethernet LAN, how can A talk to B if A only knows B's IP address, e.g., using socket programming? What layer is IP? Ethernet?
- Translation **from IP address to physical (MAC) address** is done by the **address resolution protocol (ARP)**
- Example: Ethernet uses 48-bit addresses
  - Each Ethernet network interface card (NIC) has globally unique Medium Access Control (MAC) or physical address
  - First 24 bits identify NIC manufacturer; second 24 bits are serial number
  - $\underbrace{00:90:27}_{\text{Intel}}:96:68:07$ 12 hex numbers

Chapter 8

Communication Networks and Services



LEON-GARCIA • WIDJAJA
COMMUNICATION NETWORKS
Fundamental Concepts and Key Architectures

The Internet Protocol

Internet Protocol

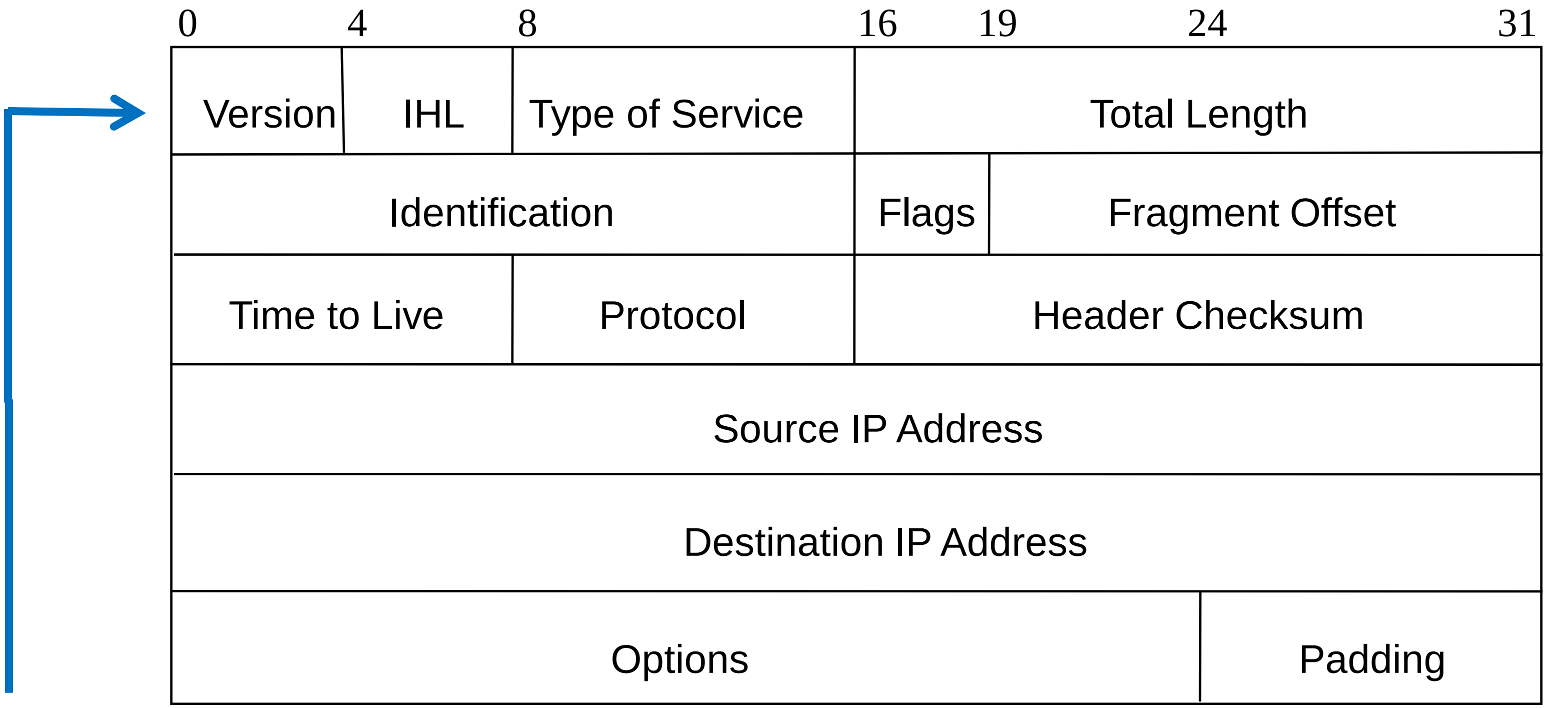
- Provides best effort, connectionless packet delivery
  - motivated by the need to keep routers simple and by adaptability to failure of network elements
  - packets may be lost, out of order, or even duplicated
  - higher layer protocols must deal with these, if necessary
- RFCs 791, 950, 919, 922, and 2474.
- IP is part of Internet STD number 5, which also includes:
  - Internet Control Message Protocol (ICMP), RFC 792
  - Internet Group Management Protocol (IGMP), RFC 1112

IP Packet Header

<table>
<tr><th>Bit # → 0</th><th>4</th><th>8</th><th>16</th><th>19</th><th>24</th><th>31</th></tr>
<tr><td>Version</td><td>IHL</td><td>Type of Service</td><td colspan="4">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td colspan="3">Fragment Offset</td></tr>
<tr><td colspan="2">Time to Live</td><td>Protocol</td><td colspan="4">Header Checksum</td></tr>
<tr><td colspan="7">Source IP Address</td></tr>
<tr><td colspan="7">Destination IP Address</td></tr>
<tr><td colspan="5">Options</td><td colspan="2">Padding</td></tr>
</table>

- **Minimum 20 bytes** (first 5 logical rows, 4 bytes/row in the figure)
- Packet security options, specification of a particular route for the packet, timestamps etc. (read RFC 2113). Not often used. Reserved for future extensions (for example RSVP etc.)

IP Packet Header



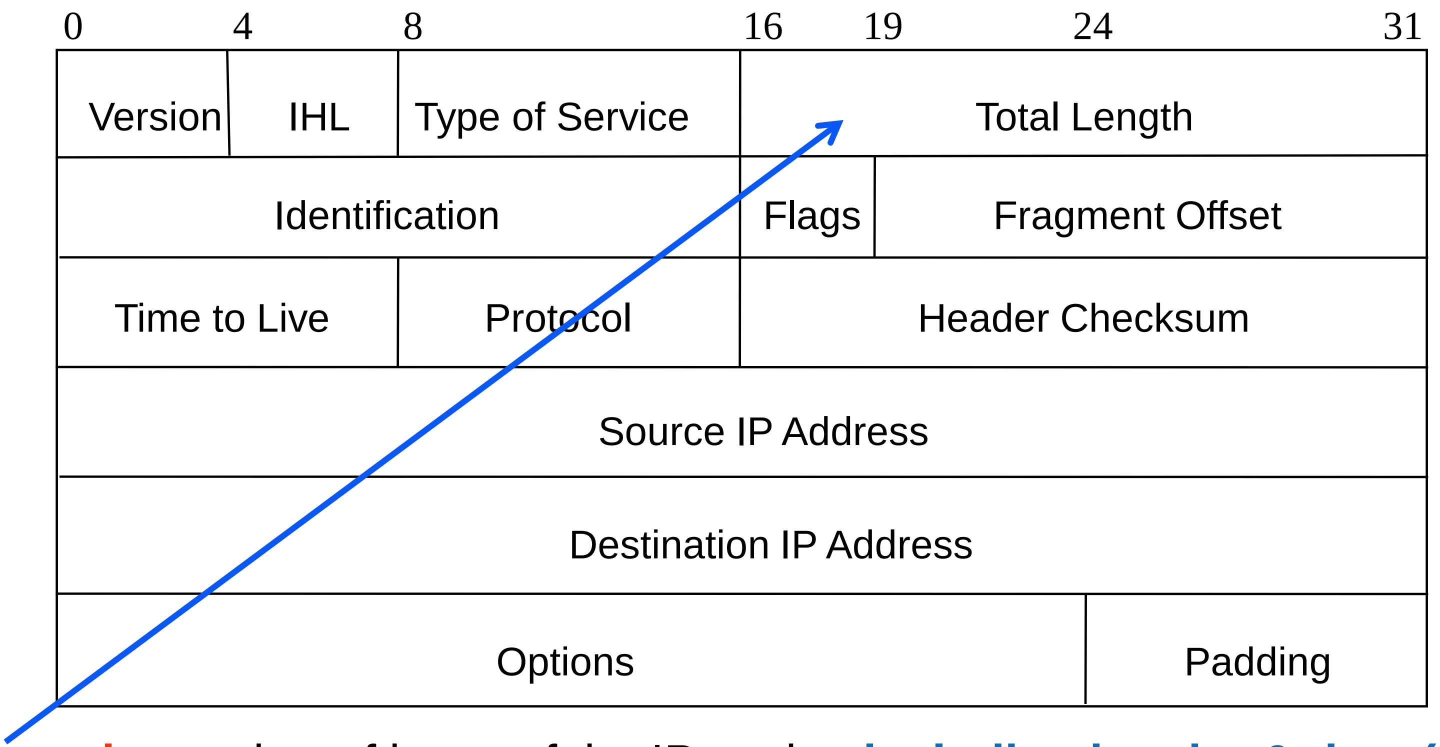
0	4	8	16	19	24	31
Version	IHL	Type of Service	Total Length			
Identification			Flags	Fragment Offset		
Time to Live		Protocol	Header Checksum			
Source IP Address						
Destination IP Address						
Options					Padding	

Version: current IP version is 4.

Internet header length (IHL): length of the **header** in 32-bit words or 4-byte length, e.g., 5 -> 20 bytes.

Type of service (TOS): priority of packet at each router. Differentiated Services (DiffServ) extends TOS field to include other services besides best effort, 6-bit used or 64 QoS levels.

IP Packet Header



Total length: number of bytes of the IP packet **including header & data (payload)**, maximum length is 65535 bytes.

Identification, Flags, and Fragment Offset: used for fragmentation and reassembly (more on this shortly).

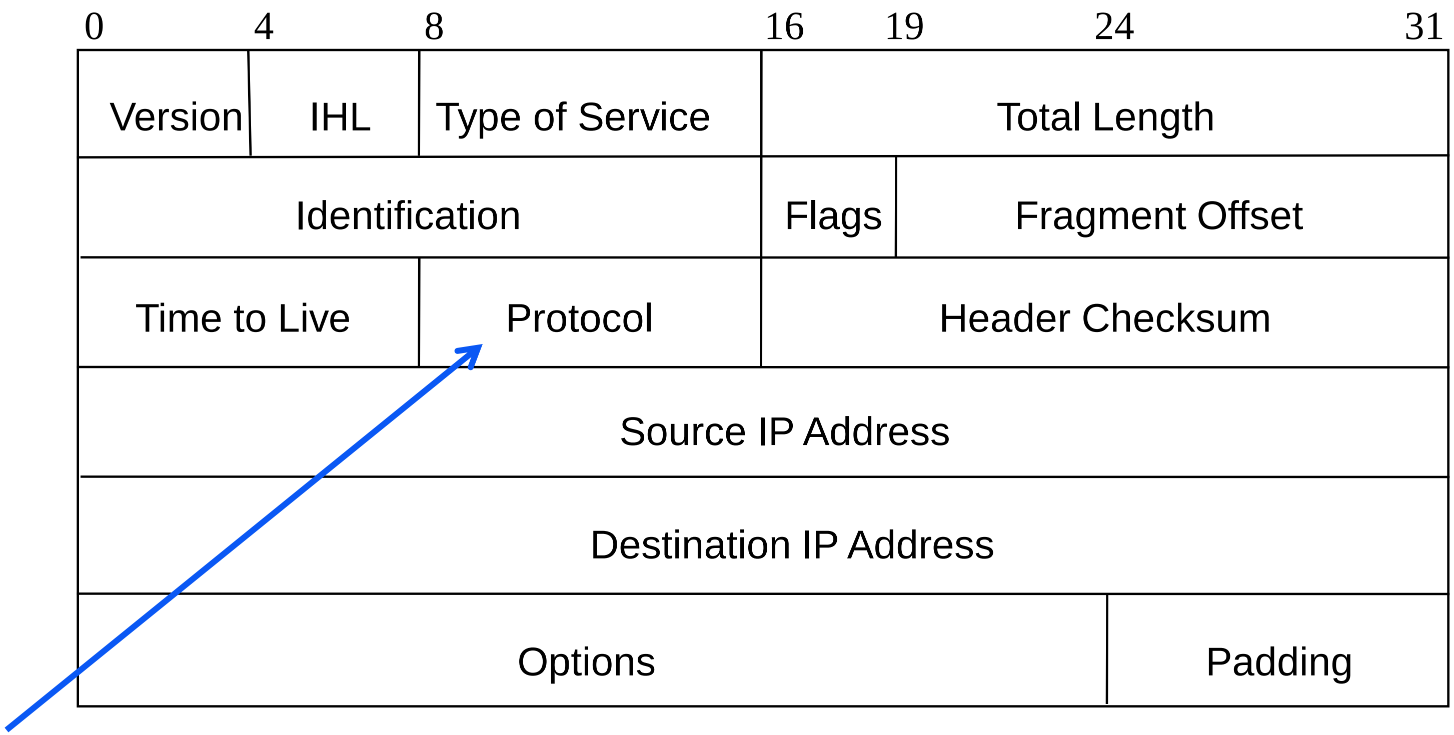
IP Packet Header

0	4	8	16	19	24 31
Version	IHL	Type of Service	Total Length		
Identification			Flags	Fragment Offset	
Time to Live		Protocol	Header Checksum		
Source IP Address					
Destination IP Address					
Options					Padding

Time to live (TTL): number of hops a packet is allowed to traverse in the network.

- Each router along the path to the destination decrements this value by one.
- If the value reaches zero before the packet reaches the destination, the router discards the packet and sends an error message back to the source.
- Q: Why TTL?
- Why loops may happen using SPF?

IP Packet Header

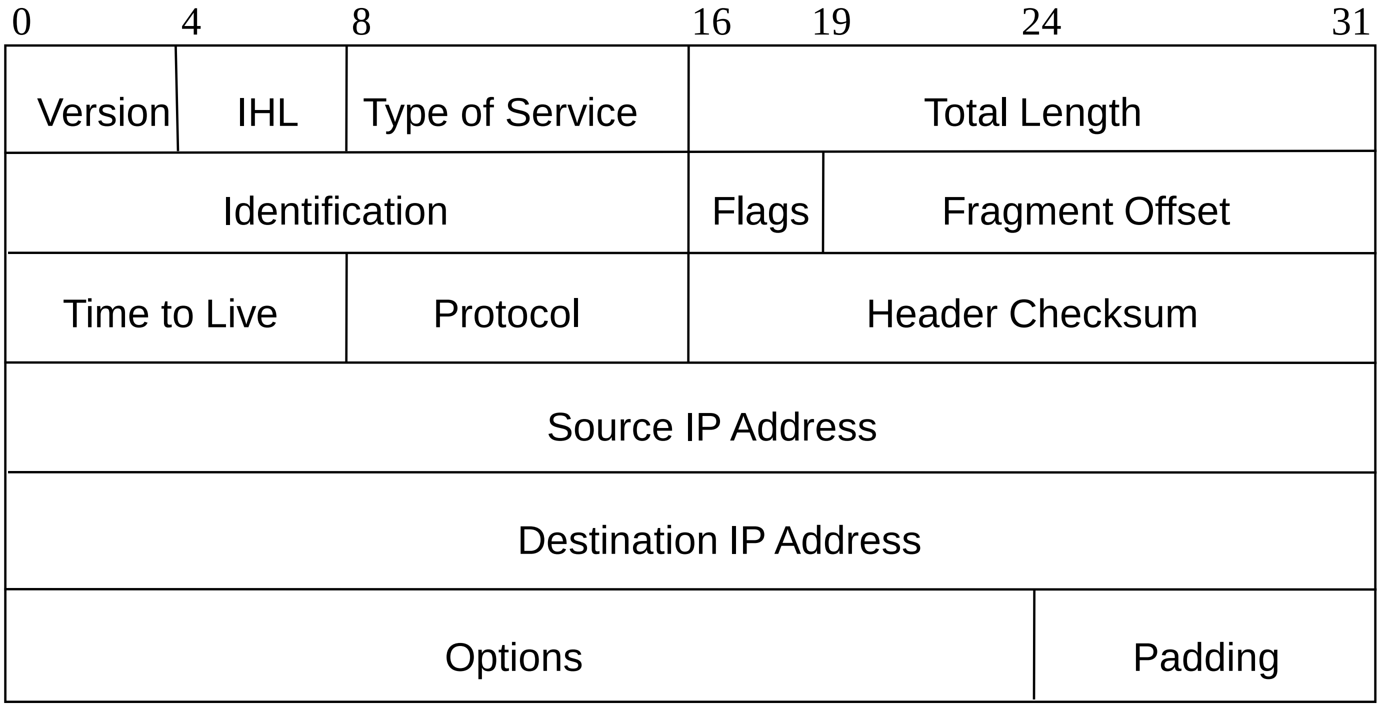


Protocol: specifies **upper-layer protocol** that is to receive IP data at the destination. Examples include TCP (prot. = 6), UDP (prot. = 17), and OSPF (prot. = 89).

Header checksum (CRC-16): verifies the integrity of the IP **header**.

Source IP address and **destination IP address:** contain the addresses of the source and destination hosts.

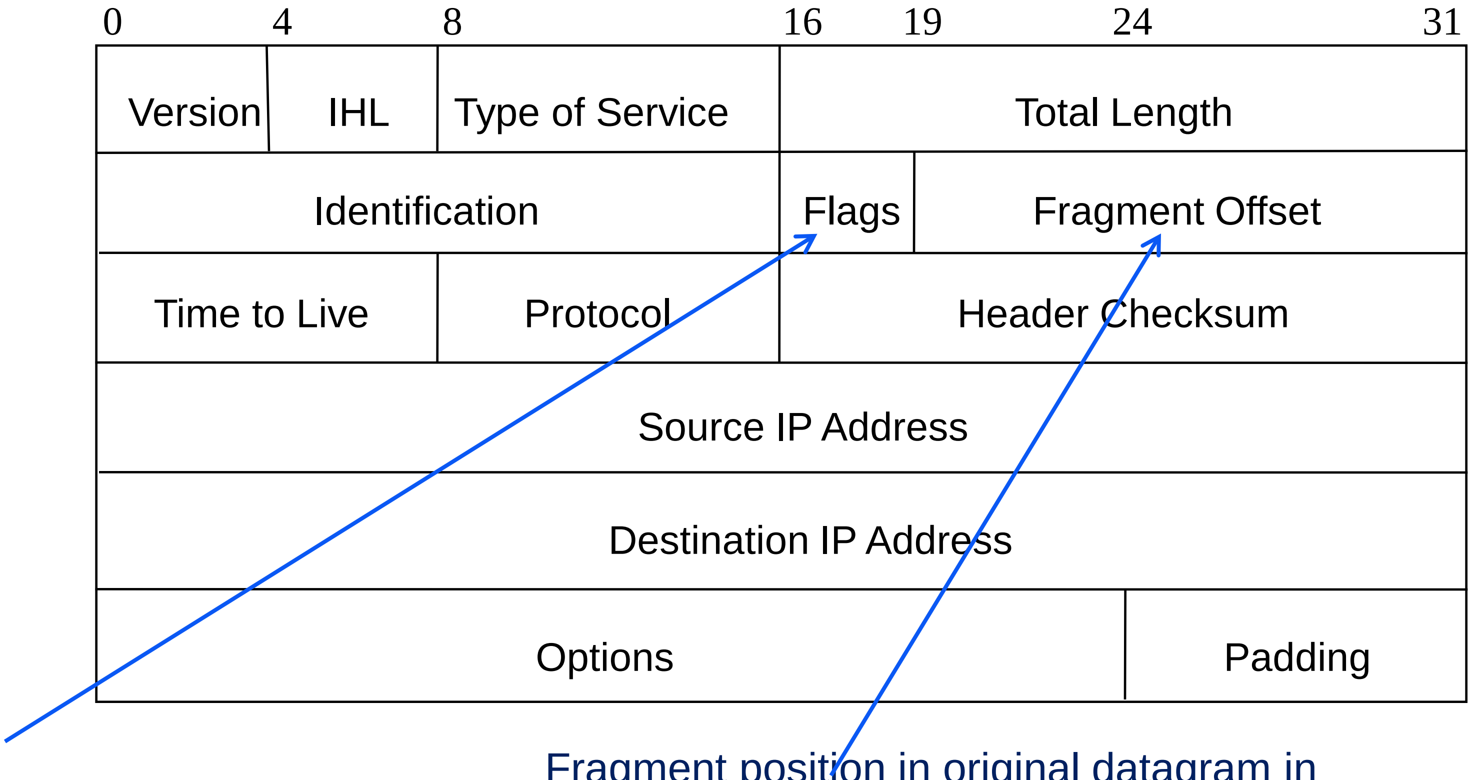
IP Packet Header



Options: Variable length field, allows packet to request special features such as security level, route to be taken by the packet, and timestamp at each router. Detailed descriptions of these options can be found in [RFC 791].

Padding: This field is used to make the header a multiple of 32-bit words.

IP Header – Flags & Fragmentation

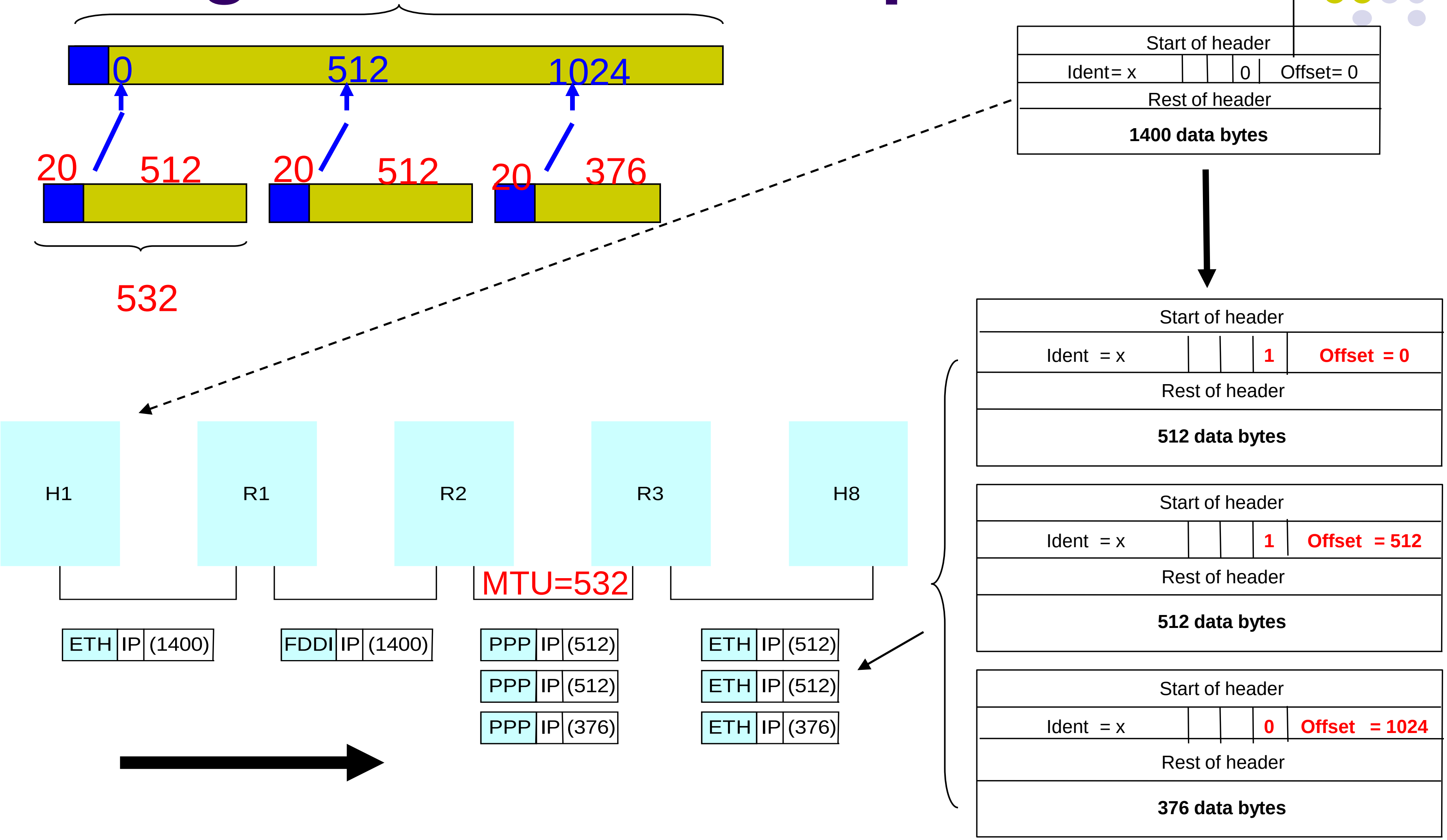


Flags
3bits: x, DF, MF
x
DF: Don’t fragment me
MF: More fragment to come

Fragment position in original datagram in multiple of 8 octets/bytes

MTU: max layer 3 packet that can be transmitted over a layer 2

Fragmentation Example



Start of header
Ident= x | 0 | Offset= 0
Rest of header
1400 data bytes

Start of header
Ident = x | **1** | **Offset = 0**
Rest of header
512 data bytes

Start of header
Ident = x | **1** | **Offset = 512**
Rest of header
512 data bytes

Start of header
Ident = x | **0** | **Offset = 1024**
Rest of header
376 data bytes

IP Header Processing

What steps need to be done?

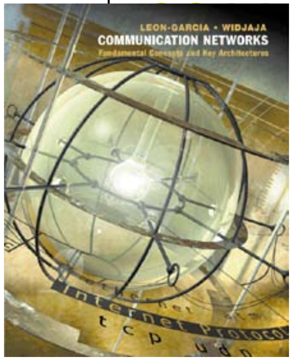
1. Error checking: Compute **header checksum** for correctness and check that fields in header (e.g. version and total length) contain valid values
2. Routing Table lookup: Determine next hop

   Q: Which field to check?

   - Destination IP address (and ToS if needed)
3. Update the header: Change fields that require updating (TTL, header checksum)

   Q: Why the checksum needs to be updated?

   - TTL has been changed and checksum is for the entire header



Chapter 8 Communication Networks and Services

- ***Internet Addressing***

IP Addressing

- RFC 1166
- Each host on Internet has unique 32 bit IP address
- Each address has two parts: ***netid and hostid***
  - ***Q: Why two parts instead of one?***
    - ***Think about area code for phone numbers, e.g., 613***
- ***netid*** unique & administered by
  - American Registry for Internet Numbers (ARIN)
  - Reseaux IP Europeens (RIPE)
  - Asia Pacific Network Information Centre (APNIC)
- Facilitates routing and increase scalability
- A separate address is required for each physical interface of a host to a network;
- Dotted-Decimal Notation:

IP address of 10000000 10000111 01000100 00000101

is 128.135.68.5 in dotted-decimal notation

Classful Addresses

Class A

	7 bits	**24 bits**
0	netid	hostid

- 126 (2^7-2) networks with up to ~16 million (2^{24}) hosts

1.0.0.0 to 127.255.255.255

Class B

		14 bits	**16 bits**
1	**0**	netid	hostid

- 16,382 networks with up to ~ 64,000 (2^{16}) hosts

128.0.0.0 to 191.255.255.255

Class C

			21 bits	**8 bits**
1	**1**	**0**	netid	hostid

- 2 million networks with up to 254 (2^8-2) hosts

192.0.0.0 to 223.255.255.255

Class D

				28 bits
1	1	1	0	multicast address

224.0.0.0 to
239.255.255.255

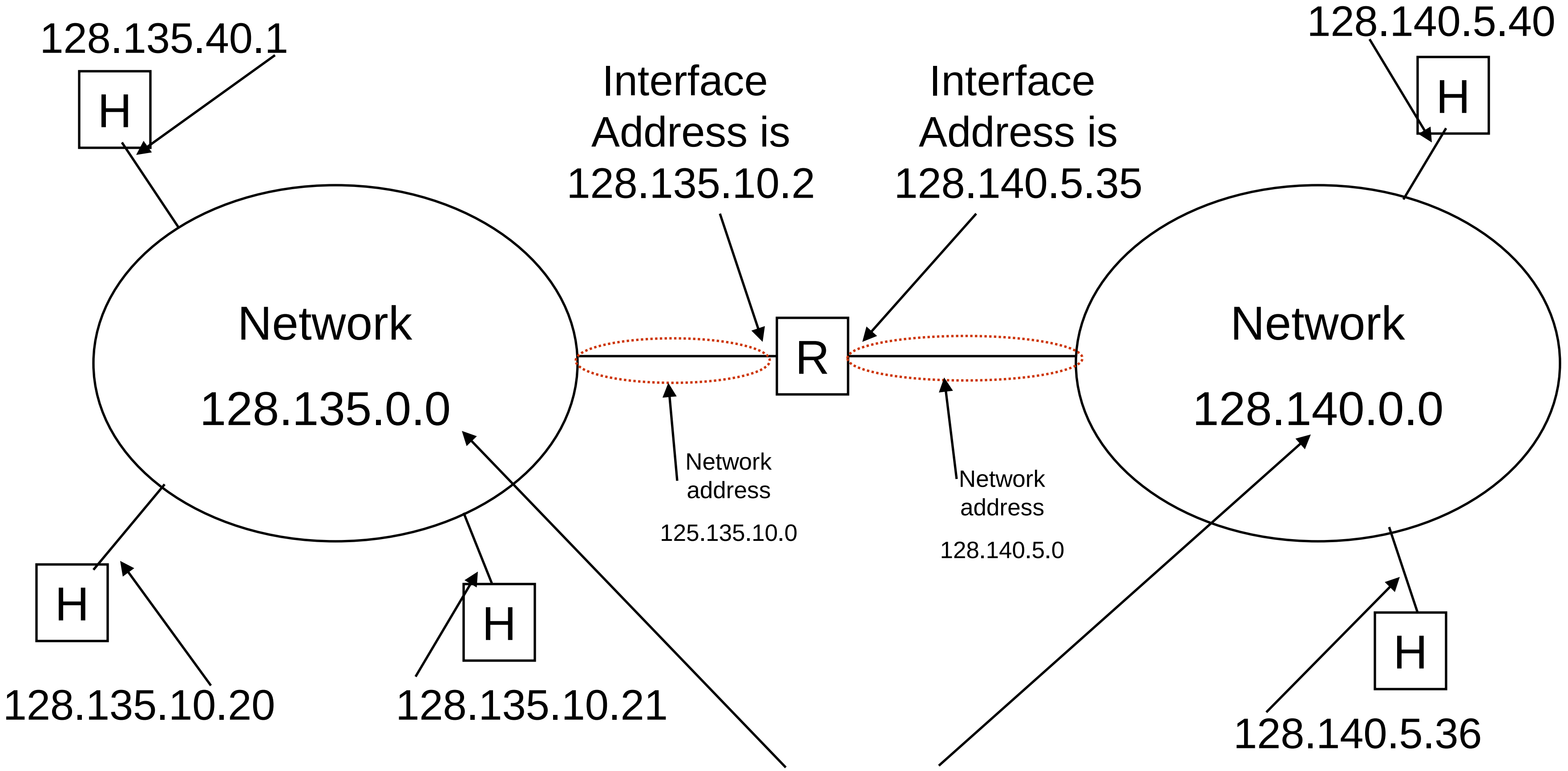
- Up to 250 million multicast groups at the same time
- Permanent group addresses
  - All systems in LAN; All routers in LAN;
  - All OSPF routers on LAN; All designated OSPF routers on a LAN, etc.
- Temporary groups addresses created as needed
- Special multicast routers

Class E (**1111**) is reserved for experiments

Private IP Addresses

- Specific ranges of IP addresses set aside for use in private networks (RFC 1918), considered *unregistered*.
- Use restricted to private internets, e.g., home or enterprise networks; routers in public Internet discard packets with these addresses
- Range 1: 10.0.0.0 to 10.255.255.255
- Range 2: 172.16.0.0 to 172.31.255.255
- Range 3: 192.168.0.0 to 192.168.255.255
- Q: How to covert private IP addresses to global address?
  - **Network Address Translation (NAT)**

Example of IP Addressing



128.135.40.1

Interface Address is 128.135.10.2

Interface Address is 128.140.5.35

128.140.5.40

Network 128.135.0.0

Network 128.140.0.0

R

H

Network address 125.135.10.0

Network address 128.140.5.0

128.135.10.20

128.135.10.21

128.140.5.36

Address with host ID=**all 0s** refers to the network

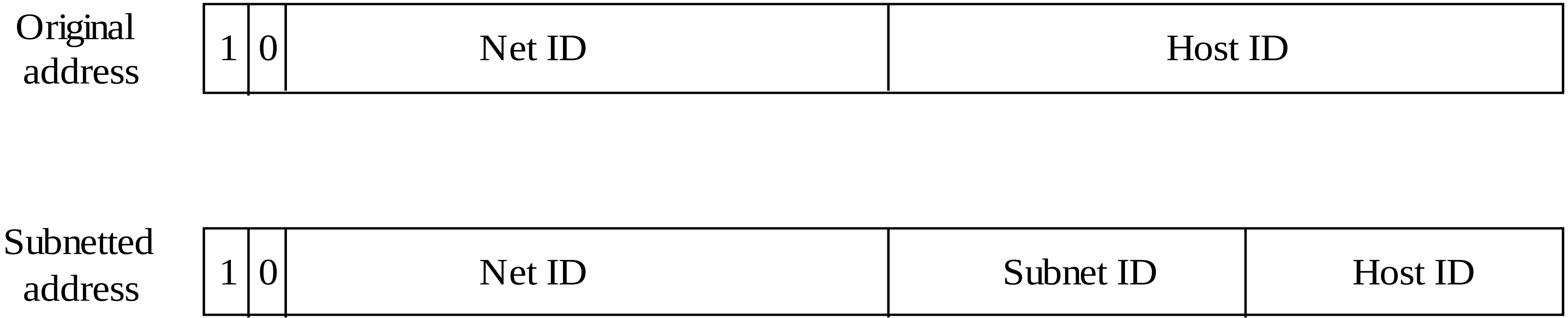
Address with host ID=**all 1s** refers to a broadcast packet

R = router

H = host

Subnet Addressing

- Subnet addressing introduces **another hierarchical level (on top of Classes A, B, C)**
- Transparent to remote networks
- Simplifies management of **multiplicity of LANs**
  - Isolation of subnets for privacy/security
  - Reduce broadcast domain
- **Q: How do we know the size of subnet?**
  - **Masking** used to find subnet number (**boundary**)



Original address

1 | 0 | Net ID | Host ID

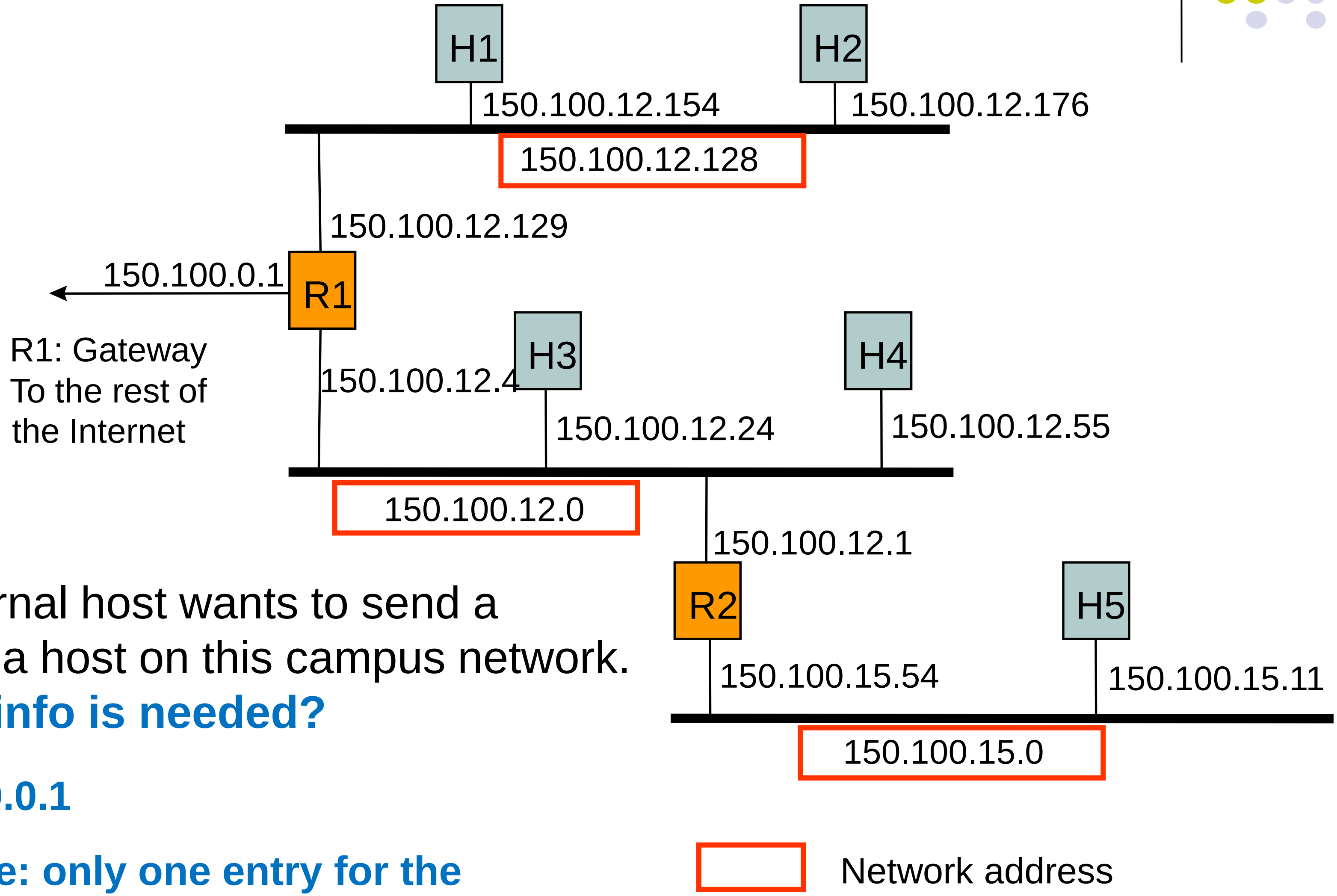
Subnetted address

1 | 0 | Net ID | Subnet ID | Host ID

Subnetting Example

- Organization has Class B address with network ID: 150.100.0.0
  - Q: Is it class B? How many bits are use for net/host IDs for class B?
    - Class B: (150) **10**…. (14 bits) + 16 host IDs
- Need to create subnets with up to 100 hosts each
  - Q: how many bits are needed for 100 hosts?
    - 7 bits sufficient for each subnet (2^7=128 hosts)
    - 16-7 = 9 bits for subnet ID
- Q: what is the subnet for an IP address, e.g.,150.100.12.176?
- **Apply subnet mask** to IP addresses to find corresponding subnet
  - Example: Find **subnet** for 150.100.12.176
  - IP add = 10010110 01100100 00001100 10110000
  - Mask = 11111111 11111111 11111111 10000000
  - **AND** = 10010110 01100100 00001100 10000000
  - Subnet = 150.100.12.128 **/25** ← specifies no of **consecutive 1's** in the mask (boundary)
  - Subnet address used by routers within an organization

Subnet Example



If an external host wants to send a packet to a host on this campus network.

Q: What info is needed?

150.100.0.1

Scalable: only one entry for the entire subnet

Routing with Subnetworks

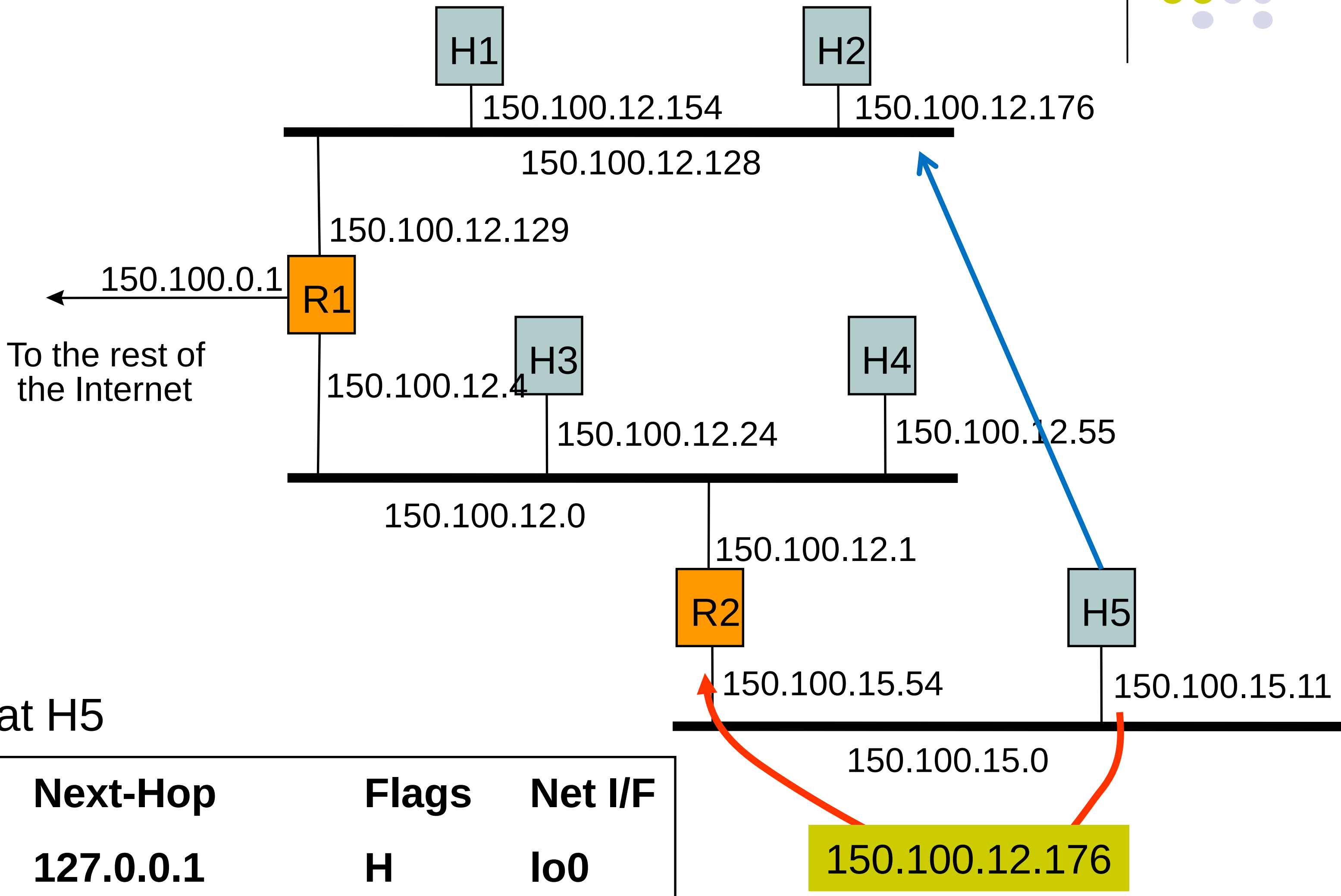
- IP layer in hosts and routers maintain a routing table
- **Originating host:** To send an IP packet, consult routing table
  - If destination host is in same network, send packet *directly* using appropriate network interface
  - Otherwise, send packet indirectly; typically, routing table indicates a default router
- **Router:** Examine IP destination address in arriving packet
  - If dest IP address not it's own, router consults routing table to determine next-hop and associated network interface & forwards packet

Routing Table

- Each row in routing table contains (and many more):
  - Destination IP address
  - IP address of next-hop router
  - Physical address
  - Statistics information
  - Flags
    - H=1 (0) indicates route is to a host (network)
    - G=1 (0) indicates route is to a router (directly connected destination)

- Routing table search order & action
  - Complete destination address; send as per next-hop & G flag
  - Destination network ID; send as per next-hop & G flag
  - Default router entry; send as per next-hop
  - Declare packet undeliverable; send ICMP "host unreachable error" packet to originating host

Example: Host H5 sends packet to host H2

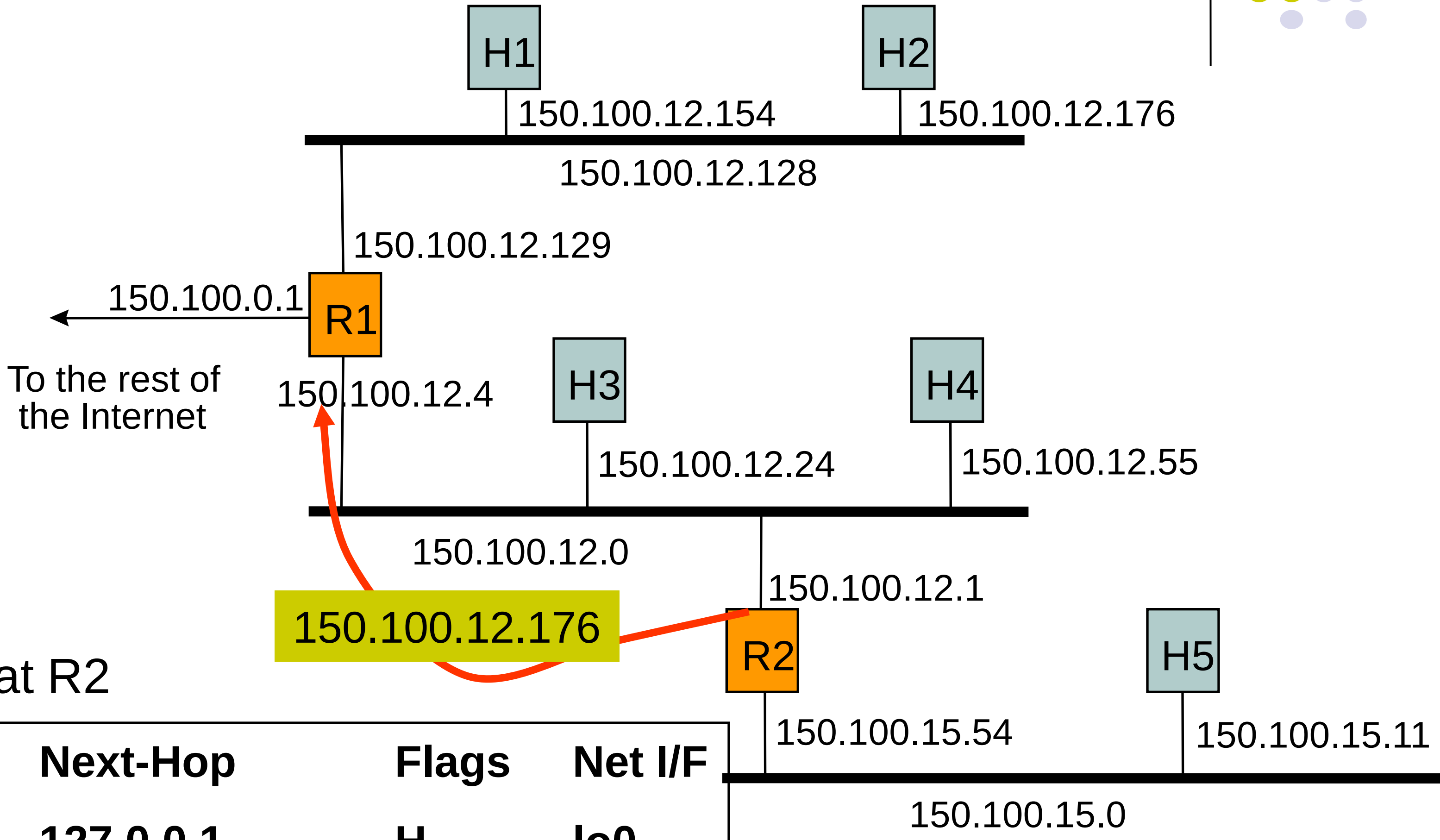


Routing Table at H5

Destination	Next-Hop	Flags	Net I/F
127.0.0.1	**127.0.0.1**	**H**	**lo0**
default	**150.100.15.54**	**G**	**emd0**
150.100.15.0	**150.100.15.11**		**emd0**

127.0.0.1 is for loopback (mostly for testing purpose)

Example: Host H5 sends packet to host H2



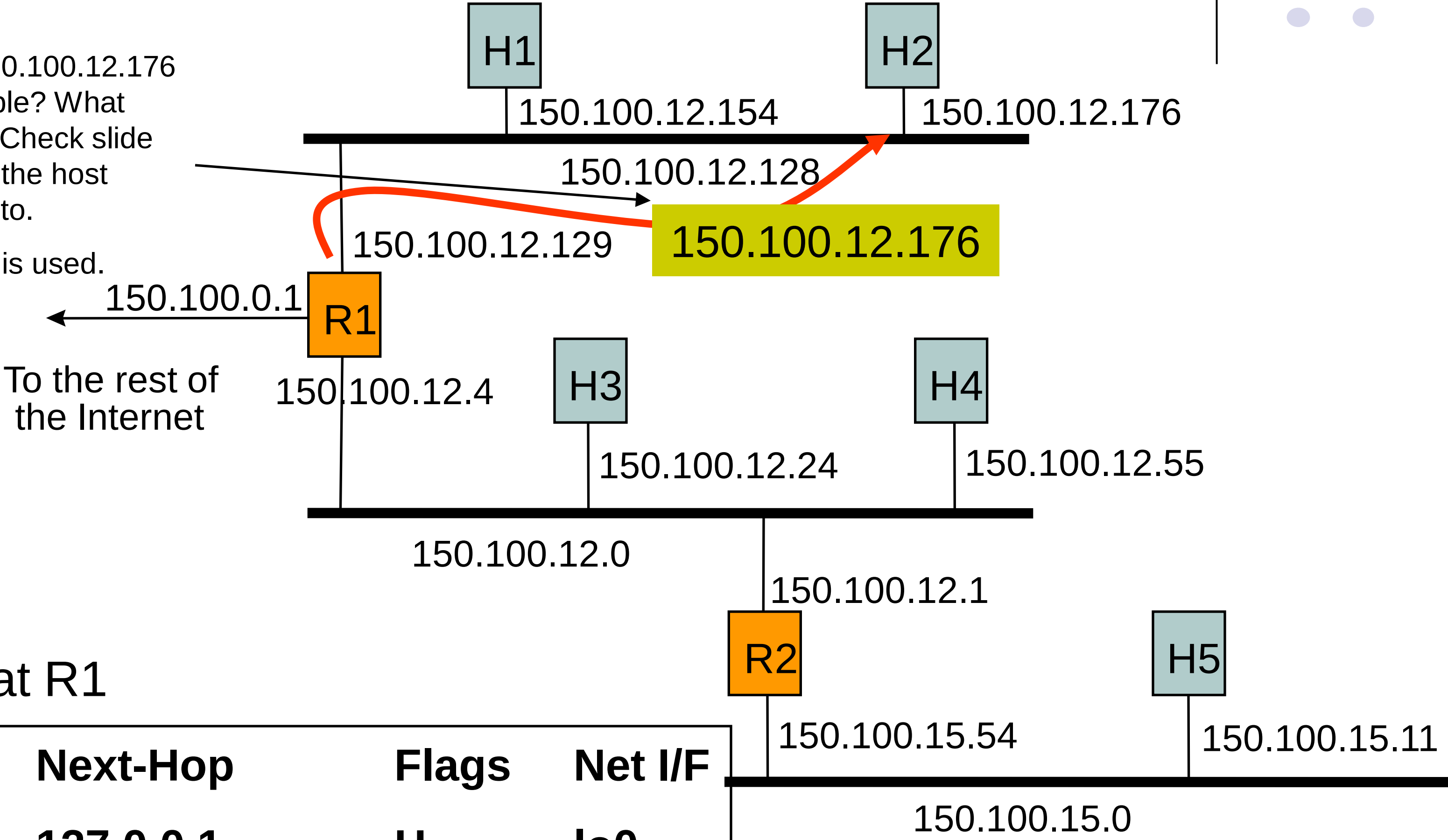
Routing Table at R2

Destination	Next-Hop	Flags	Net I/F
127.0.0.1	127.0.0.1	H	lo0
default	150.100.12.4	G	emd0
150.100.15.0	150.100.15.54		emd1
150.100.12.0	150.100.12.1		emd0

Example: Host H5 sends packet to host H2

What would happen if 150.100.12.176 was not in the routing table? What options would we have? Check slide 26 to see which network the host 150.100.12.176 belongs to.

Suppose a **9-bit subnet** is used.



Routing Table at R1

Destination	Next-Hop	Flags	Net I/F
127.0.0.1	127.0.0.1	H	lo0
150.100.12.176	150.100.12.176		emd0
150.100.12.0	150.100.12.4		emd1
150.100.15.0	150.100.12.1	G	emd1

IP Address Problems

- In the 1990, two problems became apparent
  - IP addresses were being exhausted
  - IP routing tables were growing very large
- IP Address Exhaustion
  - Class A, B, and C address structure inefficient
    - Class B too large for most organizations, but future proof
    - Class C too small
    - Rate of class B allocation implied exhaustion by 1994
- IP routing table size
  - Growth in number of networks in Internet reflected in # of table entries
    - From 1991 to 1995, routing tables doubled in size every 10 months
    - Stress on router processing power and memory allocation
- Short-term solution:
  - Classless Interdomain Routing (CIDR), RFC 1518
  - New allocation policy (RFC 2050)
  - Private IP Addresses set aside for intranets
- Long-term solution: IPv6 with much bigger address space

Supernetting

- Summarize a contiguous group of class C addresses using **variable-length mask.** **Reason?**
  - Not enough class B, get a range of class C IDs
- Example: 192.158.16.0/20
  - IP Address (192.158.16.0) & mask length (20)
  - IP add = 11000000 10011110 00010000 00000000
  - Mask = 11111111 11111111 11110000 00000000
  - Contains 16 Class C blocks, corresponding to 16 subnetworks:
    - From 11000000 10011110 00010000 00000000
      i.e. 192.158.16.0 (no. 1 subnetwork)
    - Up to 11000000 10011110 00011111 00000000
      i.e. 192.158.31.0 (no. 16 subnetwork)

Classless Inter-Domain Routing (CIDR)

- CIDR deals with Routing Table explosion problem
  - Networks represented by prefix and mask
  - Pre-CIDR: Network with a range of 16 continuous class C blocks requires 16 entries
  - Post-CIDR: Network with a range of 16 continuous class C blocks requires 1 entry
- Solution: Route according to prefix of address, not class
  - Routing table entry has <IP address, network mask>
  - Example: 192.32.136.0/21
  - 11000000 00100000 10001000 00000001 : min address
  - 11111111 11111111 11111000 00000000 : mask
  - 11000000 00100000 10001--- ------------- : IP prefix
  - 11000000 00100000 10001111 11111110 : max address

Longest Prefix Match-Classless Interdomain Routing (p.557)

- CIDR impacts routing & forwarding
- Routing tables and routing protocols must carry IP address and mask
- **Multiple entries may match** a given IP destination address
- Example: Routing table may contain
  - 205.100.0.0/22 which corresponds to a given supernet
  - 205.100.0.0/20 which results from aggregation of a larger number of destinations into a different supernet
  - Packet must be routed using the *more specific route*, that is, the **longest prefix match**
- Several fast longest-prefix matching algorithms are available

Routing table lookup: Longest Prefix Match

- **Longest Prefix Match:** Search for the routing table entry that has the longest match with the prefix of the destination IP address

1. Search for a match on all 32 bits
2. Search for a match for 31 bits

   …..

32. Search for a mach on 0 bits

Host route, loopback entry
→ 32-bit prefix match

Default route is represented as 0.0.0.0/0
→ 0-bit prefix match

128.143.71.21

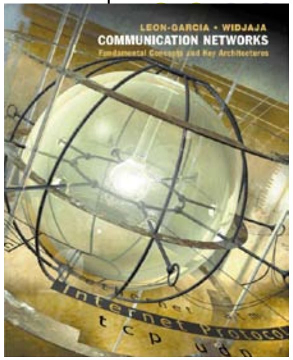
↓

Destination address	Next hop
10.0.0.0/8	R1
128.143.0.0/16	R2
128.143.64.0/20	R3
128.143.192.0/20	R3
128.143.71.0/24	R4
128.143.71.55/32	R3
default	R5

↓

The longest prefix match for 128.143.71.21 is for 24 bits with entry 128.143.71.0/24

Datagram will be sent to R4



Chapter 8 Communication Networks and Services

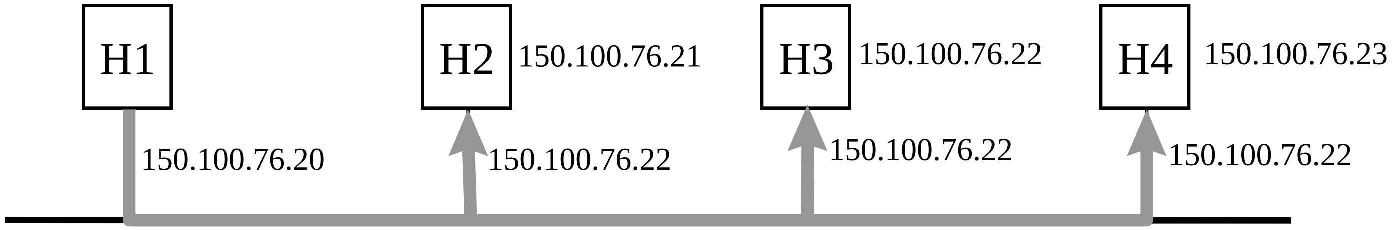
- ***ARP***
- ***Fragmentation and Reassembly***
- ***ICMP***

Address Resolution Protocol

Although IP address identifies a host, the packet is physically delivered by an underlying network (e.g., Ethernet) which uses its own *physical address* (MAC address in Ethernet).

How to map an IP address to a physical address?

H1 wants to learn physical address of H3 -> broadcasts an ARP request

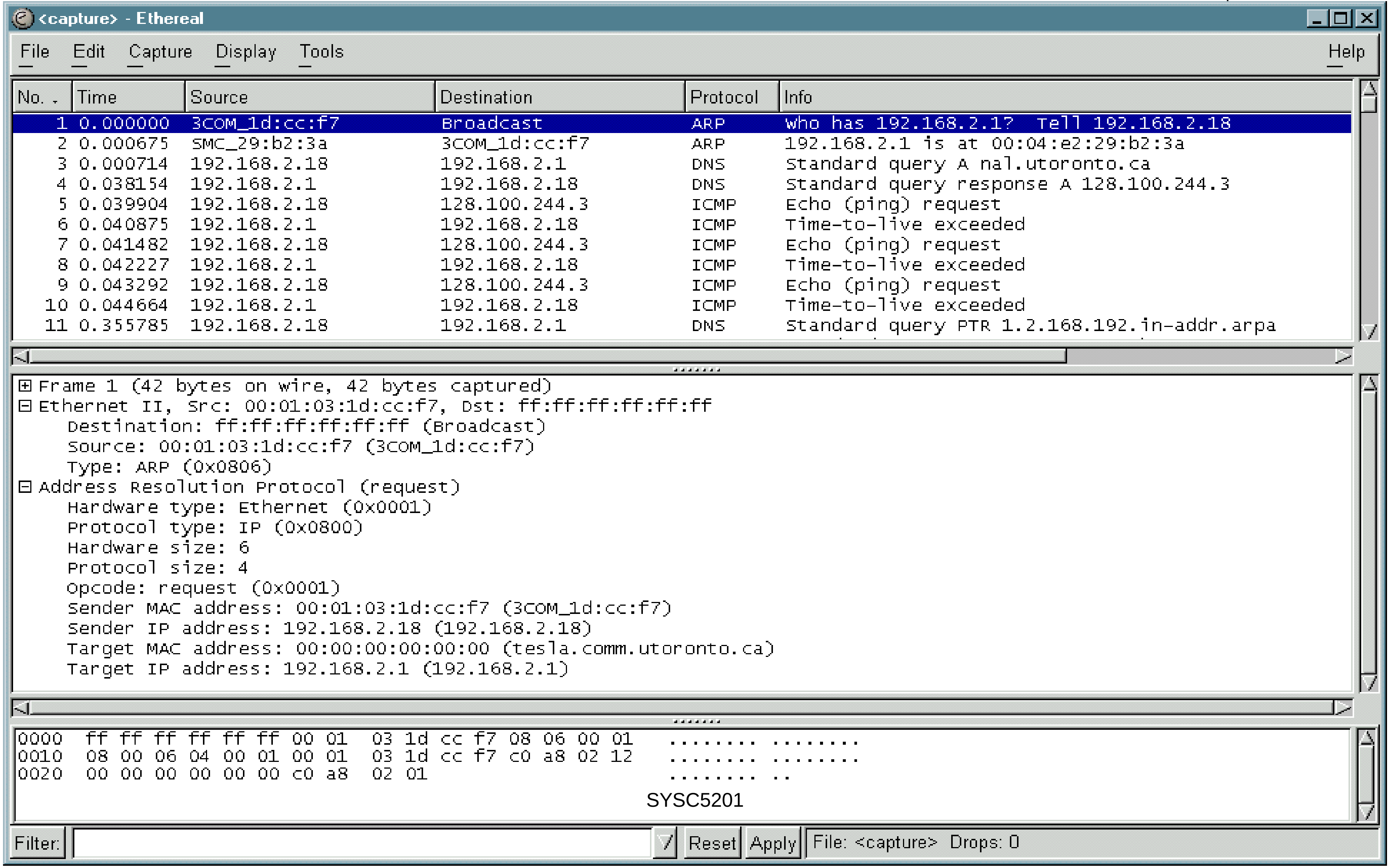


ARP request (what is the MAC address of 150.100.76.22?)

Every host receives the request, but only H3 reply with its physical address

ARP response (my MAC address is 08:00:5a:3b:94)

Example of ARP



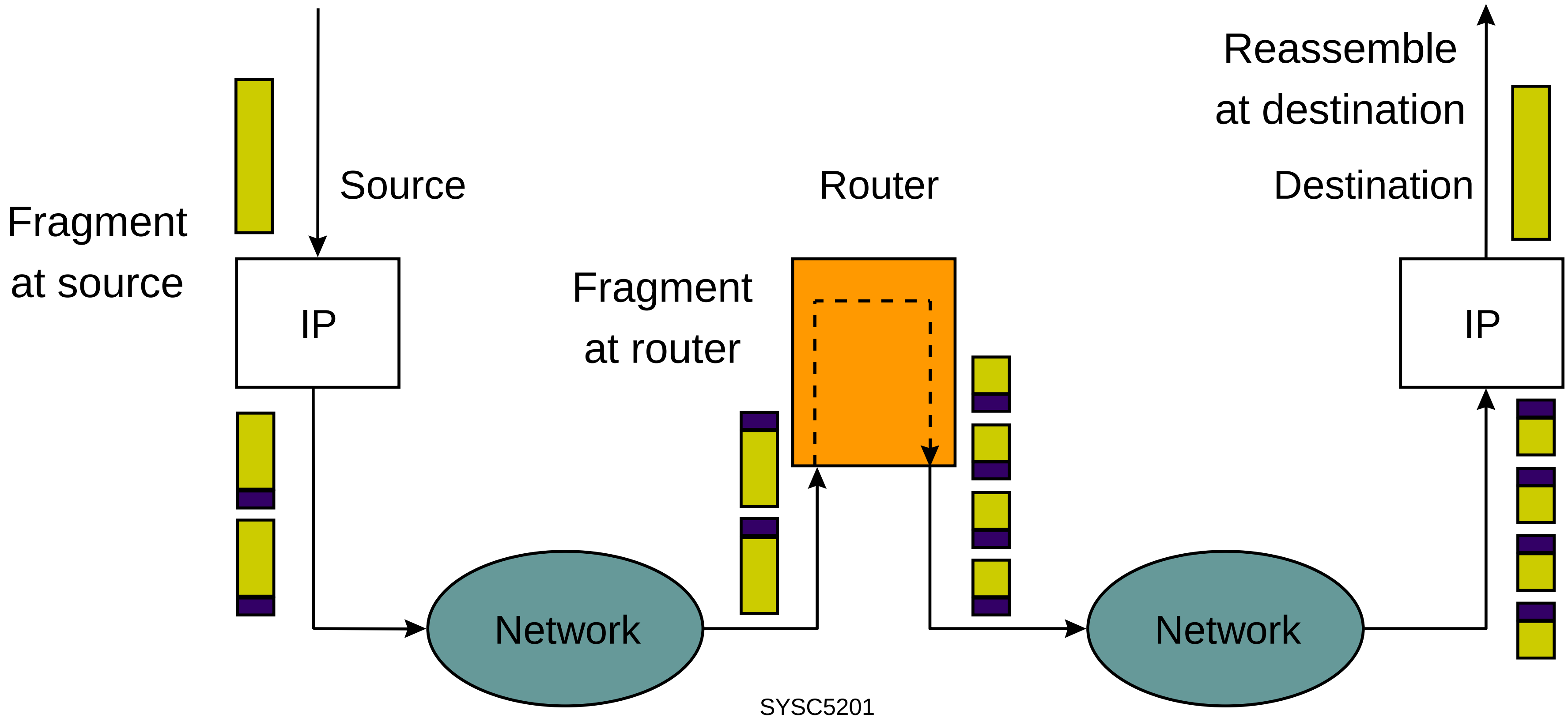
```
No. .  Time      Source             Destination        Protocol  Info
  1 0.000000  3COM_1d:cc:f7      Broadcast          ARP       Who has 192.168.2.1?  Tell 192.168.2.18
  2 0.000675  SMC_29:b2:3a       3COM_1d:cc:f7      ARP       192.168.2.1 is at 00:04:e2:29:b2:3a
  3 0.000714  192.168.2.18       192.168.2.1        DNS       Standard query A nal.utoronto.ca
  4 0.038154  192.168.2.1        192.168.2.18       DNS       Standard query response A 128.100.244.3
  5 0.039904  192.168.2.18       128.100.244.3      ICMP      Echo (ping) request
  6 0.040875  192.168.2.1        192.168.2.18       ICMP      Time-to-live exceeded
  7 0.041482  192.168.2.18       128.100.244.3      ICMP      Echo (ping) request
  8 0.042227  192.168.2.1        192.168.2.18       ICMP      Time-to-live exceeded
  9 0.043292  192.168.2.18       128.100.244.3      ICMP      Echo (ping) request
 10 0.044664  192.168.2.1        192.168.2.18       ICMP      Time-to-live exceeded
 11 0.355785  192.168.2.18       192.168.2.1        DNS       Standard query PTR 1.2.168.192.in-addr.arpa
```

```
⊞ Frame 1 (42 bytes on wire, 42 bytes captured)
⊟ Ethernet II, Src: 00:01:03:1d:cc:f7, Dst: ff:ff:ff:ff:ff:ff
    Destination: ff:ff:ff:ff:ff:ff (Broadcast)
    Source: 00:01:03:1d:cc:f7 (3COM_1d:cc:f7)
    Type: ARP (0x0806)
⊟ Address Resolution Protocol (request)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (0x0001)
    Sender MAC address: 00:01:03:1d:cc:f7 (3COM_1d:cc:f7)
    Sender IP address: 192.168.2.18 (192.168.2.18)
    Target MAC address: 00:00:00:00:00:00 (tesla.comm.utoronto.ca)
    Target IP address: 192.168.2.1 (192.168.2.1)
```

```
0000   ff ff ff ff ff ff 00 01  03 1d cc f7 08 06 00 01   ........ ........
0010   08 00 06 04 00 01 00 01  03 1d cc f7 c0 a8 02 12   ........ ........
0020   00 00 00 00 00 00 c0 a8  02 01                     ........ ..
```

Fragmentation and Reassembly

- **Identification** identifies a particular packet
- **Flags** = (unused, don’t fragment/DF, more fragment/MF)
- **Fragment offset** identifies the location of a fragment within a packet



Example: Fragmenting a Packet

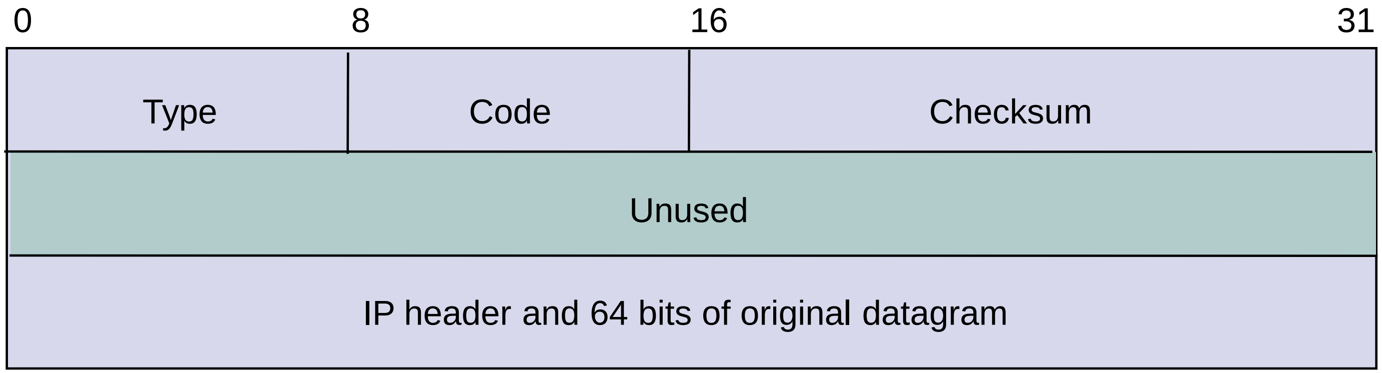
- A packet is to be forwarded to a network with MTU of 576 bytes. The packet has an IP header of 20 bytes and a data part of 1484 bytes.
- Maximum data length per fragment = 576 - 20 = 556 bytes.
- We set maximum data length to 552(=69X8) bytes to get **multiple of 8**. Note 552+552+380=1484

	Total Length	Id	MF	Fragment Offset
Original packet	1504	x	0	0
Fragment 1	572	x	1	0
Fragment 2	572	x	1	69
Fragment 3	400	x	0	138

Internet Control Message Protocol (ICMP)

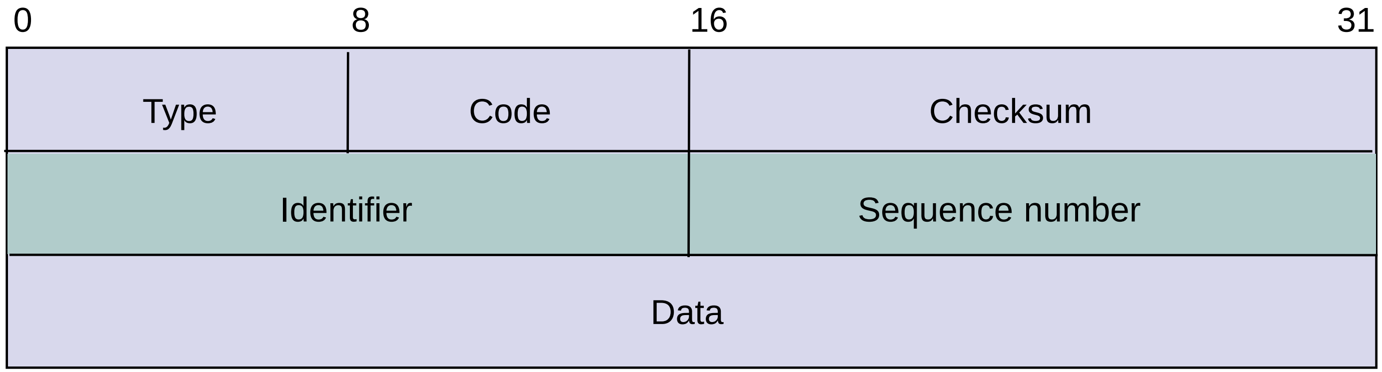
- RFC 792; Encapsulated in IP packet (prot. type = 1)
- Handles **error and control messages**
- If router cannot deliver or forward a packet, it sends an ICMP “host unreachable” message to the source
- If router receives packet that should have been sent to another router, it sends an ICMP “redirect” message to the sender; Sender modifies its routing table
- ICMP “router discovery” messages allow host to learn about routers in its network and to initialize and update its routing tables
- ICMP echo request and reply facilitate diagnostics and used in “**ping**”

ICMP Basic Error Message Format



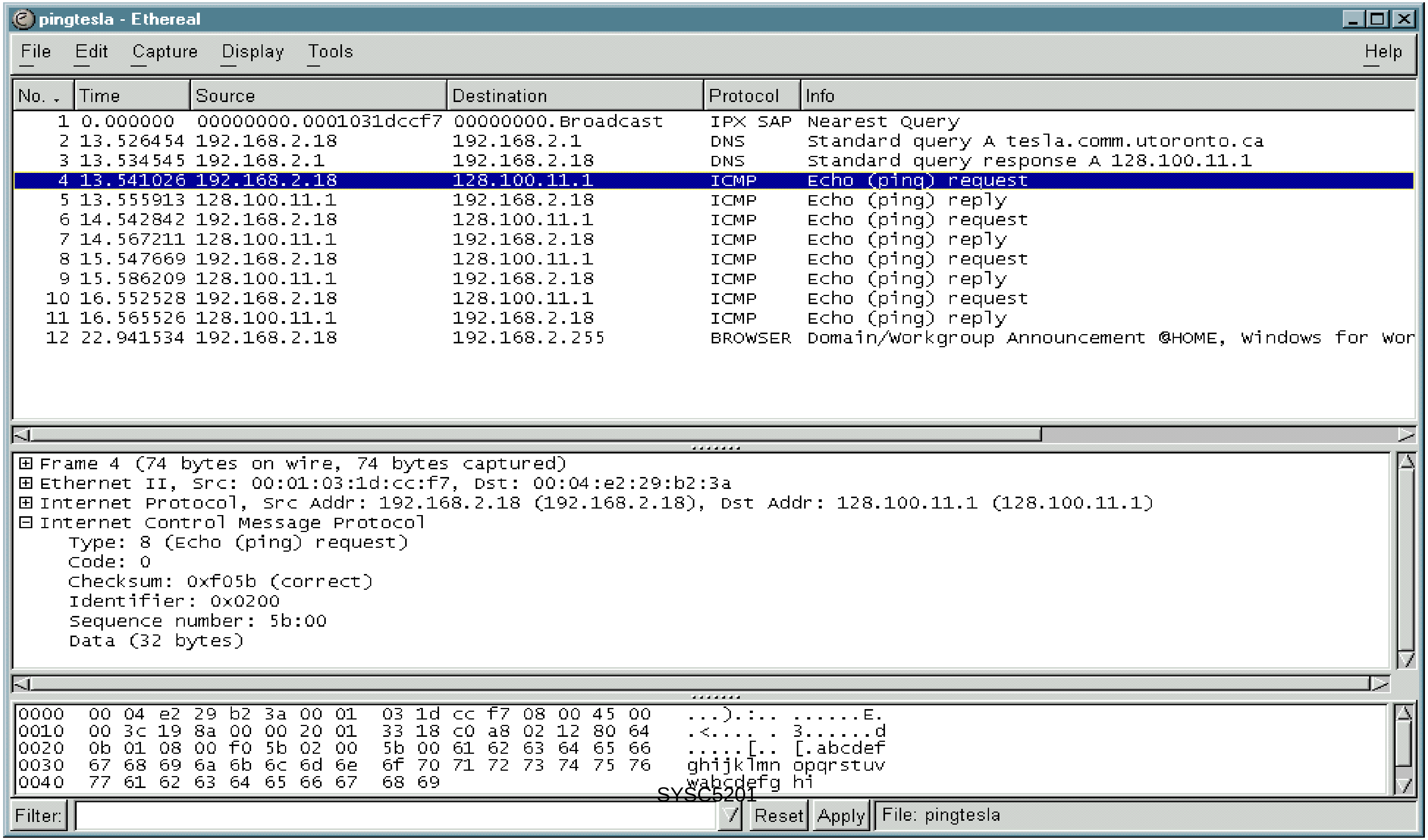
- *Type* of message: some examples
  - 0 Network Unreachable;   3 Port Unreachable
  - 1 Host Unreachable   4 Fragmentation needed
  - 2 Protocol Unreachable   5 Source route failed
  - 11 Time-exceeded, code=0 if TTL exceeded
- Code: purpose of message
- IP header & 64 bits of original datagram
  - To match ICMP message with original data in IP packet

Echo Request & Echo Reply Message Format



- Echo request: type=8; Echo reply: type=0
    - Destination replies with echo reply by copying data in request onto reply message
- Sequence number to match reply to request
- ID to distinguish between different sessions using echo services
- Used in PING

Example – Echo request



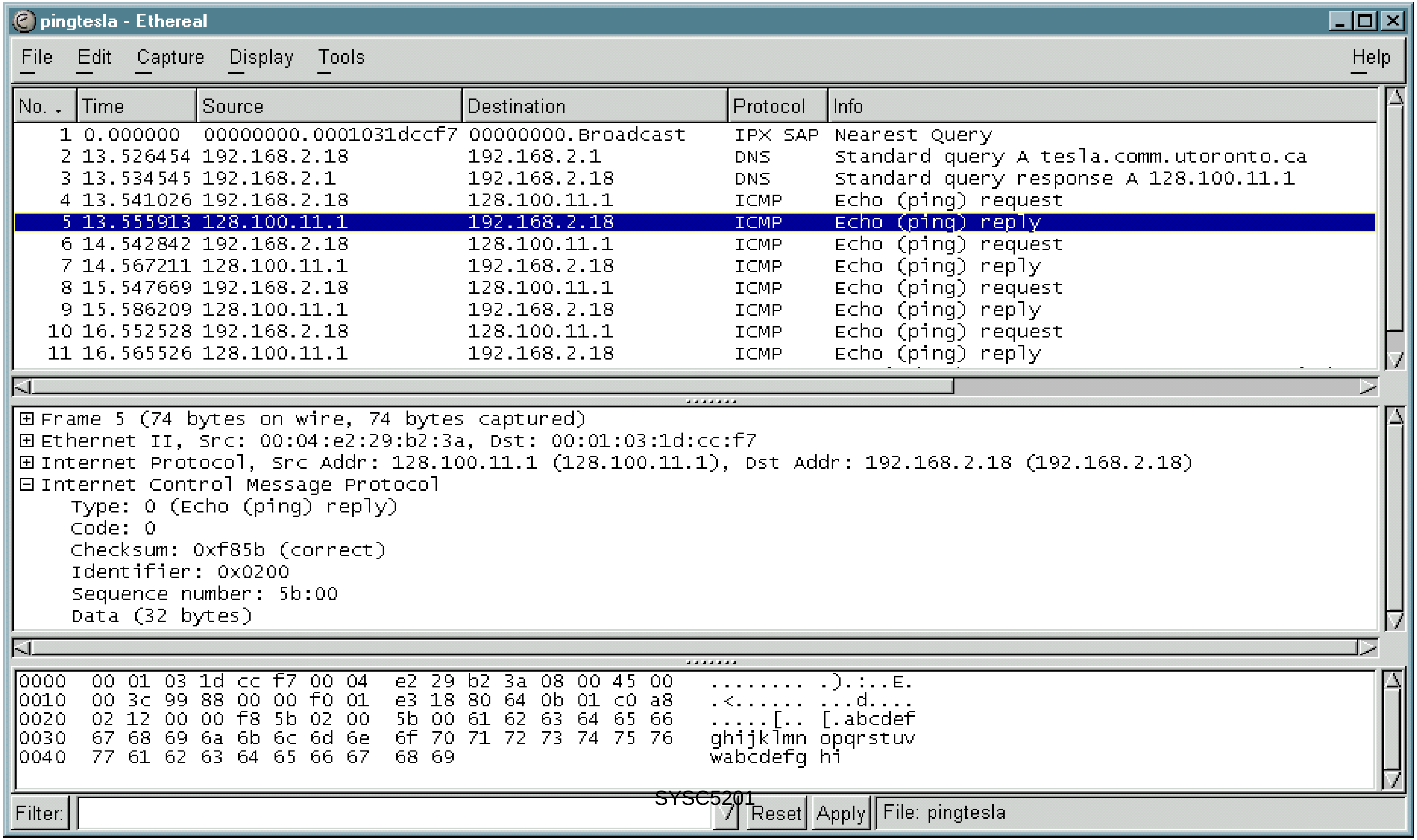
```
No. . Time      Source                  Destination         Protocol  Info
    1 0.000000  00000000.0001031dccf7   00000000.Broadcast  IPX SAP   Nearest Query
    2 13.526454 192.168.2.18            192.168.2.1         DNS       Standard query A tesla.comm.utoronto.ca
    3 13.534545 192.168.2.1             192.168.2.18        DNS       Standard query response A 128.100.11.1
    4 13.541026 192.168.2.18            128.100.11.1        ICMP      Echo (ping) request
    5 13.555913 128.100.11.1            192.168.2.18        ICMP      Echo (ping) reply
    6 14.542842 192.168.2.18            128.100.11.1        ICMP      Echo (ping) request
    7 14.567211 128.100.11.1            192.168.2.18        ICMP      Echo (ping) reply
    8 15.547669 192.168.2.18            128.100.11.1        ICMP      Echo (ping) request
    9 15.586209 128.100.11.1            192.168.2.18        ICMP      Echo (ping) reply
   10 16.552528 192.168.2.18            128.100.11.1        ICMP      Echo (ping) request
   11 16.565526 128.100.11.1            192.168.2.18        ICMP      Echo (ping) reply
   12 22.941534 192.168.2.18            192.168.2.255       BROWSER   Domain/Workgroup Announcement @HOME, Windows for Wor
```

```
⊞ Frame 4 (74 bytes on wire, 74 bytes captured)
⊞ Ethernet II, Src: 00:01:03:1d:cc:f7, Dst: 00:04:e2:29:b2:3a
⊞ Internet Protocol, Src Addr: 192.168.2.18 (192.168.2.18), Dst Addr: 128.100.11.1 (128.100.11.1)
⊟ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xf05b (correct)
    Identifier: 0x0200
    Sequence number: 5b:00
    Data (32 bytes)
```

```
0000  00 04 e2 29 b2 3a 00 01  03 1d cc f7 08 00 45 00   ...).:.. ......E.
0010  00 3c 19 8a 00 00 20 01  33 18 c0 a8 02 12 80 64   .<.... . 3......d
0020  0b 01 08 00 f0 5b 02 00  5b 00 61 62 63 64 65 66   .....[.. [.abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

]

Example – Echo Reply



```
No. . Time      Source                    Destination         Protocol Info
    1 0.000000  00000000.0001031dccf7     00000000.Broadcast  IPX SAP  Nearest Query
    2 13.526454 192.168.2.18              192.168.2.1         DNS      Standard query A tesla.comm.utoronto.ca
    3 13.534545 192.168.2.1               192.168.2.18        DNS      Standard query response A 128.100.11.1
    4 13.541026 192.168.2.18              128.100.11.1        ICMP     Echo (ping) request
    5 13.555913 128.100.11.1              192.168.2.18        ICMP     Echo (ping) reply
    6 14.542842 192.168.2.18              128.100.11.1        ICMP     Echo (ping) request
    7 14.567211 128.100.11.1              192.168.2.18        ICMP     Echo (ping) reply
    8 15.547669 192.168.2.18              128.100.11.1        ICMP     Echo (ping) request
    9 15.586209 128.100.11.1              192.168.2.18        ICMP     Echo (ping) reply
   10 16.552528 192.168.2.18              128.100.11.1        ICMP     Echo (ping) request
   11 16.565526 128.100.11.1              192.168.2.18        ICMP     Echo (ping) reply
```

```
⊞ Frame 5 (74 bytes on wire, 74 bytes captured)
⊞ Ethernet II, Src: 00:04:e2:29:b2:3a, Dst: 00:01:03:1d:cc:f7
⊞ Internet Protocol, Src Addr: 128.100.11.1 (128.100.11.1), Dst Addr: 192.168.2.18 (192.168.2.18)
⊟ Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0xf85b (correct)
    Identifier: 0x0200
    Sequence number: 5b:00
    Data (32 bytes)
```

```
0000  00 01 03 1d cc f7 00 04  e2 29 b2 3a 08 00 45 00   ........ .).:..E.
0010  00 3c 99 88 00 00 f0 01  e3 18 80 64 0b 01 c0 a8   .<...... ...d....
0020  02 12 00 00 f8 5b 02 00  5b 00 61 62 63 64 65 66   .....[.. [.abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```