CISC 7332X T6
C14a: Internetworks and The Internet

Hui Chen

Department of Computer & Information Science

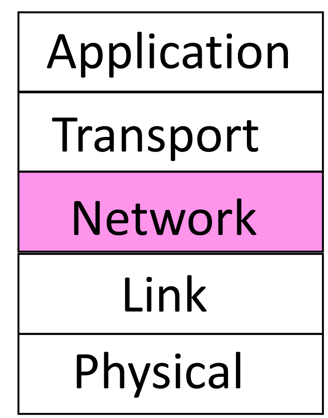
CUNY Brooklyn College

Acknowledgements

- Some pictures used in this presentation were obtained from the Internet
- The instructor used the following references
  - Larry L. Peterson and Bruce S. Davie, Computer Networks: A Systems Approach, 5th Edition, Elsevier, 2011
  - Andrew S. Tanenbaum, Computer Networks, 5th Edition, Prentice-Hall, 2010
  - James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach, 5th Ed., Addison Wesley, 2009
  - Larry L. Peterson's (http://www.cs.princeton.edu/~llp/) Computer Networks class web site

The Network Layer

- Responsible for delivering packets between endpoints over multiple links



Outline

- Motivation
- Design issues
- Routing algorithms
- Congestion Control
- Quality of Service
- Internetworking
- Network Layer of the Internet

Outline

- Discussed
  - Motivation
  - Design issues
  - Routing algorithms
- To discuss
  - Internetworking
  - Network Layer of the Internet
- Next week
  - Congestion Control
  - Quality of Service

Lecture Outline

- Motivation: two major problems
- Case study of internetworks
  - internet and the Internet
  - Global addressing scheme
  - Packet fragmentation and assembly
  - Best effort service model and datagram forwarding
  - Address translation
  - Host configuration
  - Error reporting

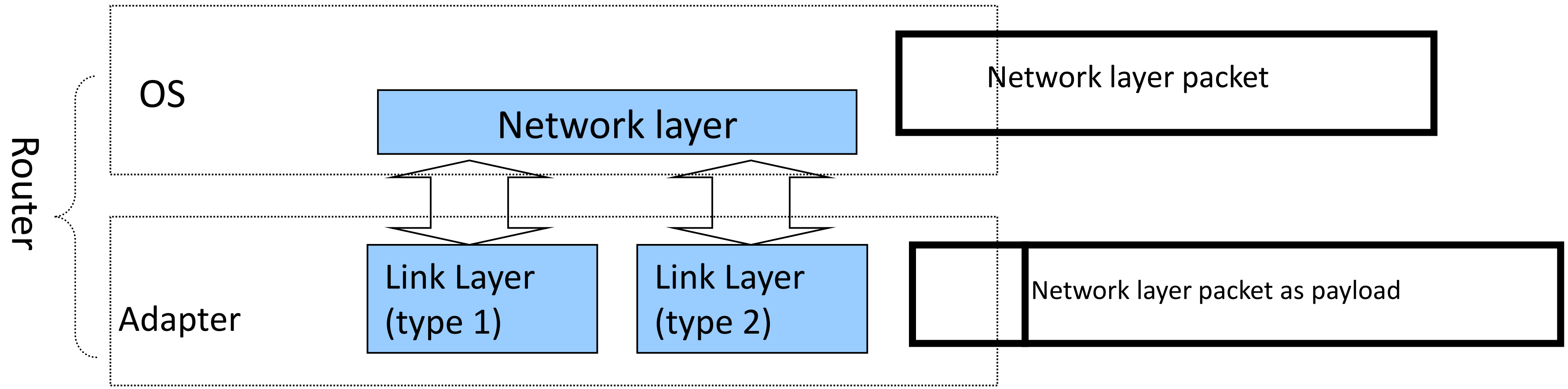
Why Network Layer?

- Responsible for delivering packets between endpoints over multiple links
- But, does an extended LAN deliver packets between endpoints over multiple links?

Heterogeneity and Scalability

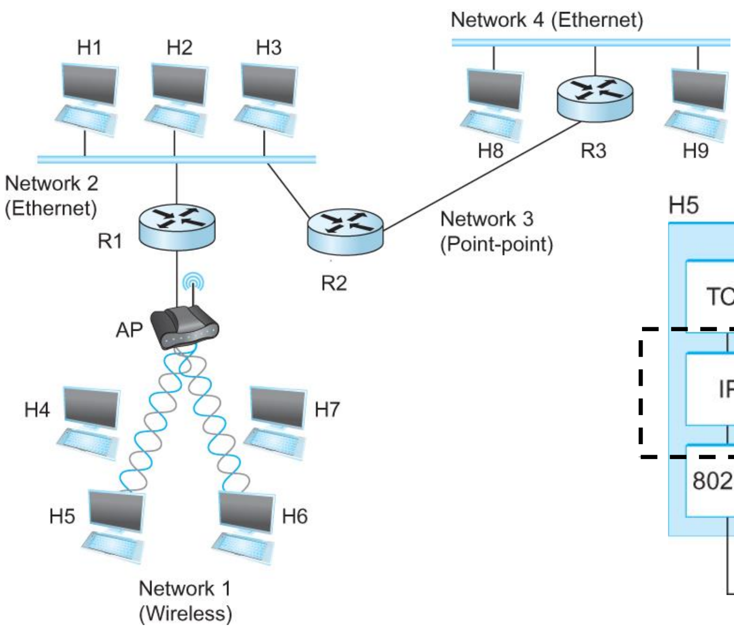
- LAN: small in size
- How to extend LAN?
  - Bridges and switches
  - Good for large networks?
- Problem 1
  - Scalability problem: spanning tree algorithms → very long path and huge forwarding tables
- Problem 2
  - Heterogeneity problem: bridges and switches: link level/layer 2 devices → networks must be using the same type of links

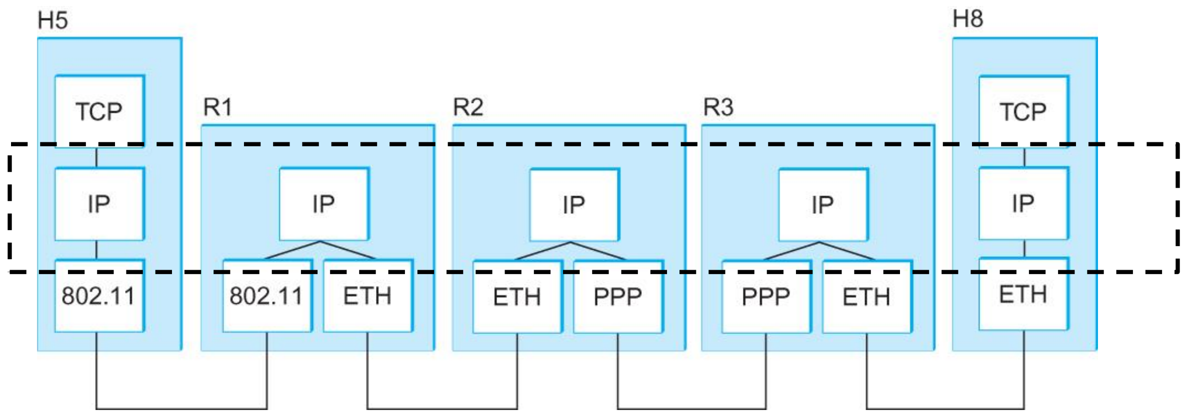
Solution: Network Layer



Example: Internet Protocol: Addressing Heterogeneity

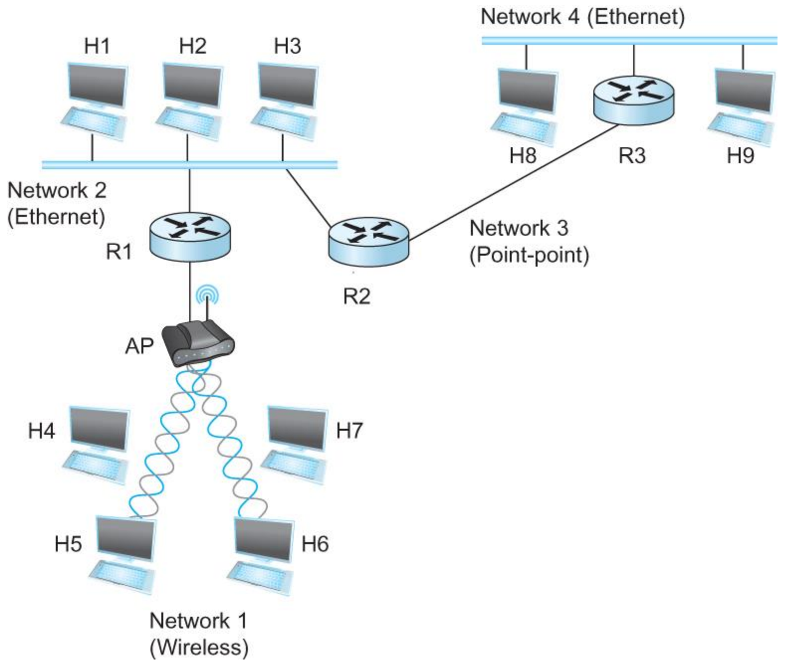
- A well-know network layer protocol is the Internet protocol
  - Provides uniform interface to upper layer
  - Can cope with different data link layer technologies





Example: Internet Protocol: Addressing Scalability

- A well-know network layer protocol is the Internet protocol
  - Forwarding data to a network instead of a host, reducing forwarding table size on the network layer and forwarding table size on the data link layer giving the same numbers of hosts



Questions?

- Motivating example
- Scalability and heterogeneity problems

Case Study: The Internet

- Global internetworks built on IP → The Internet $\neq$ internet
- Using Internet Protocol (IP) as a case study
  - Datagram forwarding and service model
  - IP packet format and global IP addressing scheme
  - Deal with Link layer and network layer interfacing
    - Packet fragmentation and assembly
    - Address translation
  - Other important issues
    - Host configuration
    - Error reporting

Versions of IP

- IPv4
- IPv6

IP Service Model

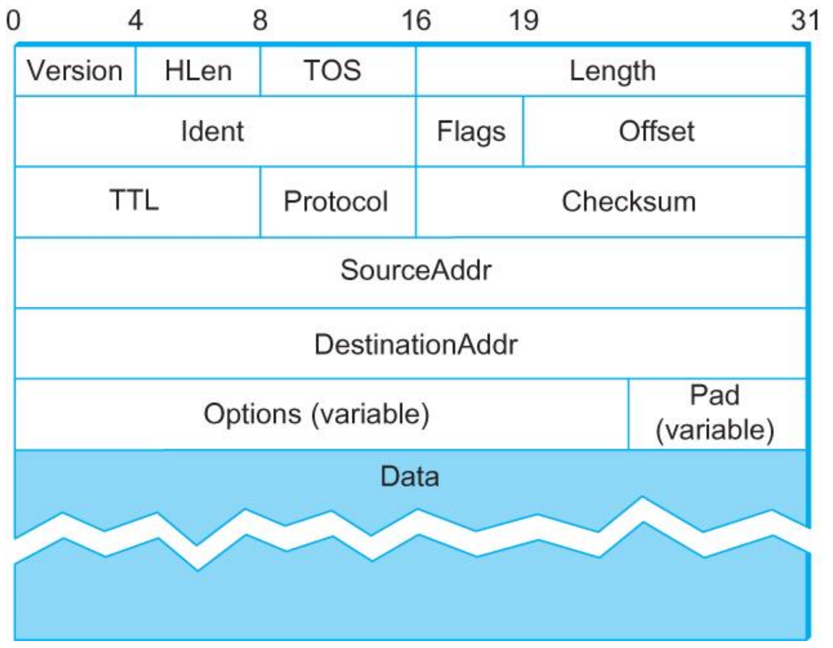
- Packet Delivery Model
  - Connectionless model for data delivery
  - Best-effort delivery (unreliable service)
    - packets may be lost
    - packets may be delivered out of order
    - duplicate copies of a packet may be delivered
    - packets may be delayed for a long time
- Global Addressing Scheme
  - Provides a way to identify all hosts in the network

Basic Data Structure: IP Packet

- Design Goals
  - Attributes and purposes
    - Support error detection and handling
    - Support networks as a forwarding source and destinations
    - Support different networking technologies
    - Support multiplexing
    - Support extensibility

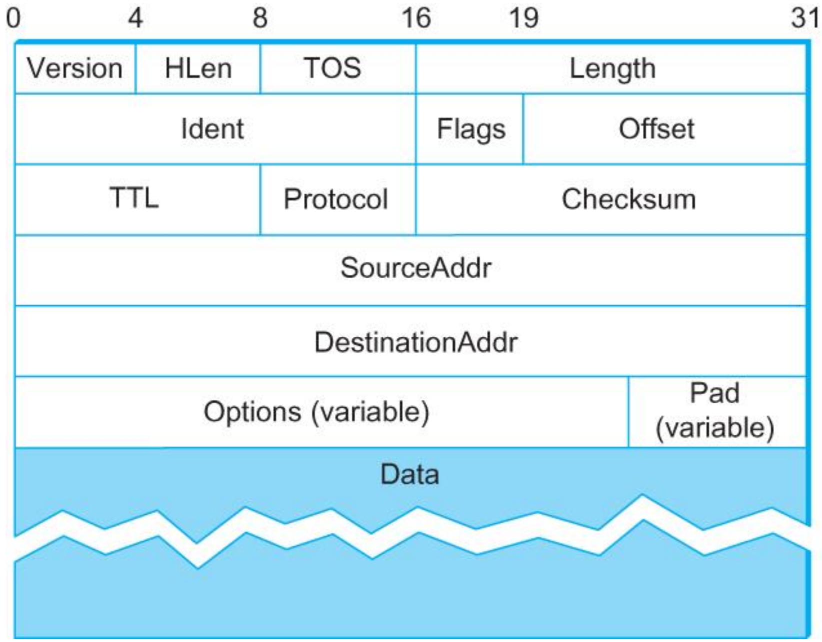
IPv4 Packet Format (1)

- Convention used to illustrate IP packet
  - 32 bit words
  - Top word transmit first
  - Left-most byte transmit first



IPv4 Packet Format

- Version (4): 4 or 6. The rest is for version 4. Discussing 6 in later lessons
- Hlen (4): number of 32-bit words in header
- TOS (8): type of service (not widely used)
- Length (16): number of bytes in this datagram
- Ident (16): used by fragmentation
- Flags/Offset (16): used by fragmentation
- TTL (8): number of hops this datagram has traveled
- Protocol (8): demux key (TCP=6, UDP=17)
- Checksum (16): of the header only
- DestAddr & SrcAddr (32)



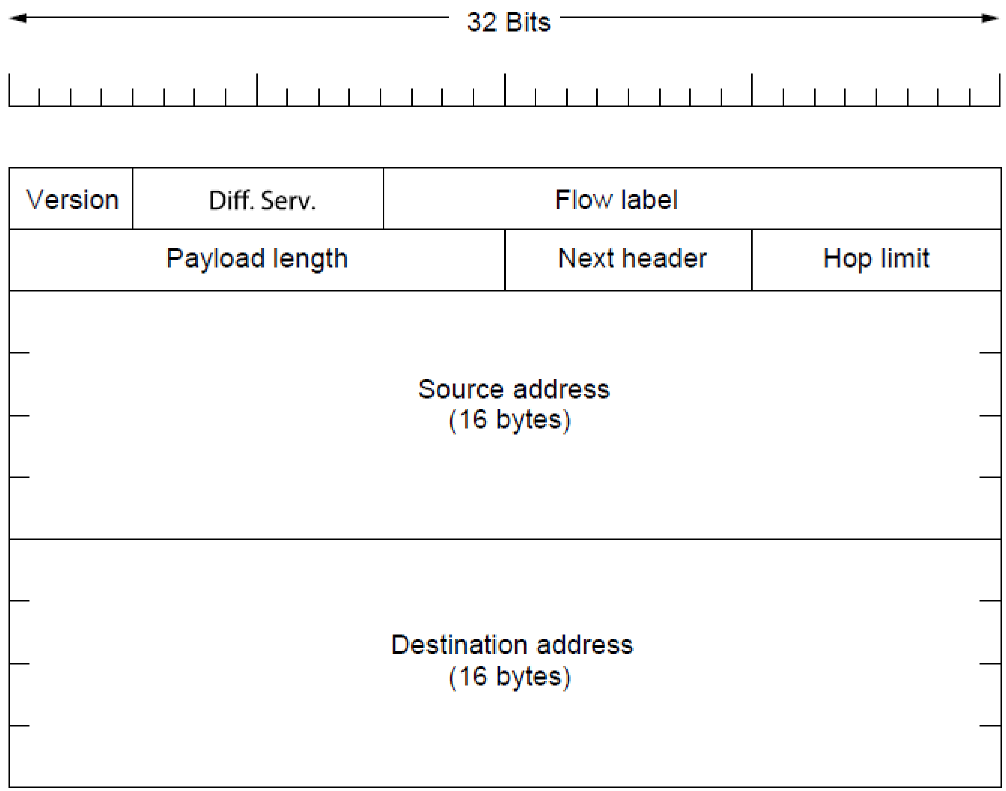
IPv6

- Major upgrade in the 1990s due to impending address exhaustion, with various other goals
    - Support billions of hosts
    - Reduce routing table size
    - Simplify protocol
    - Better security
    - Attention to type of service
    - Aid multicasting
    - Roaming host without changing address
    - Allow future protocol evolution
    - Permit coexistence of old, new protocols, …

IPv6 Packet

- IPv6 protocol header has much longer addresses (128 vs. 32 bits) and is simpler (by using extension headers)

IPv6 Packet Format



IPv6 Header

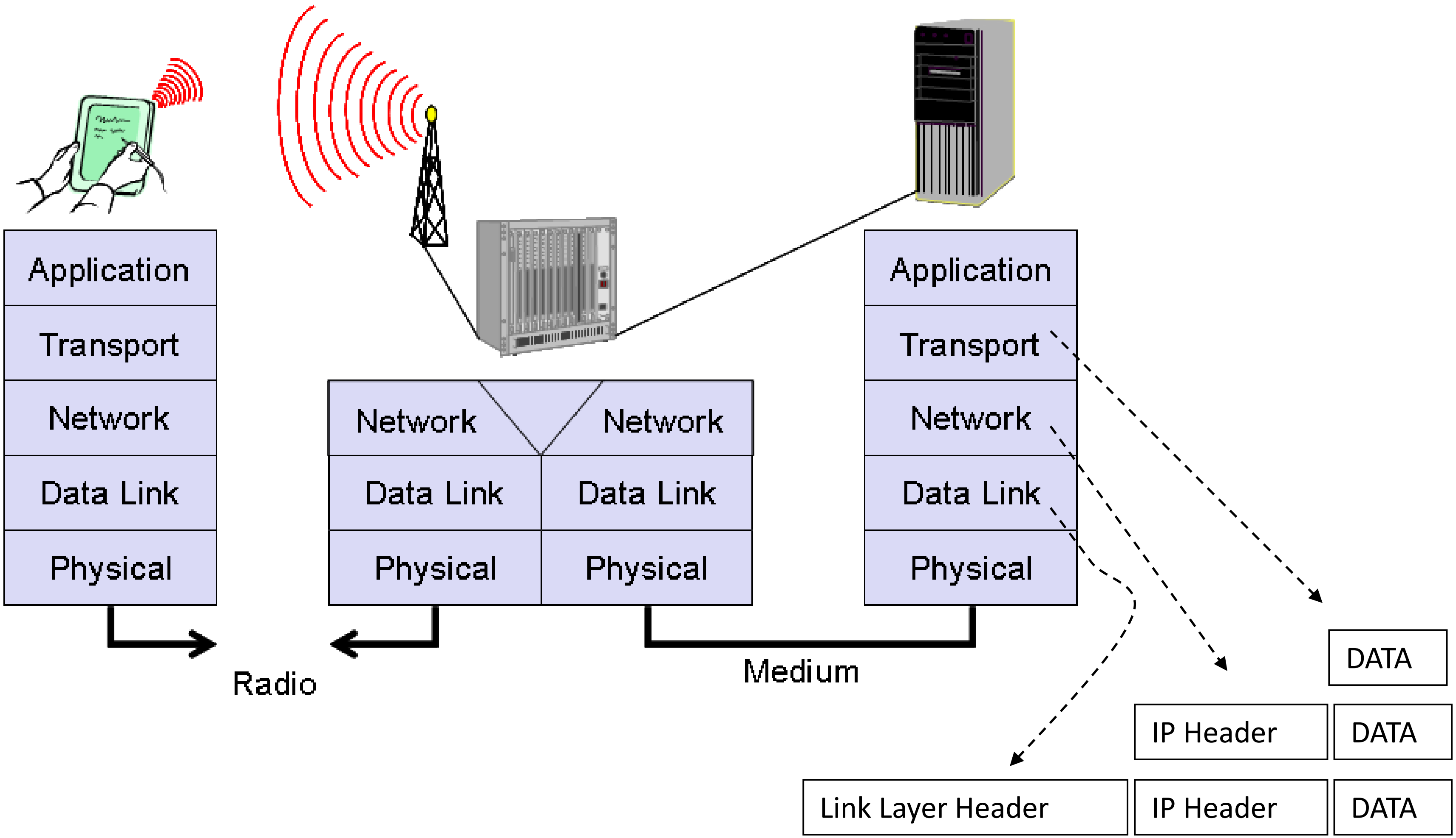
- IPv6 extension headers handles other functionality

Extension header	Description
Hop-by-hop options	Miscellaneous information for routers
Destination options	Additional information for the destination
Routing	Loose list of routers to visit
Fragmentation	Management of datagram fragments
Authentication	Verification of the sender's identity
Encrypted security payload	Information about the encrypted contents

Capturing an IP Packet

- And examining it …
- Use the *ethercap* application (a part of project 1)
- Use Wireshark
- Use Microsoft Network Monitor (Message Analyzer)
- Use tcpdump
- Use *libpcap* in your own application
- ……

IP Packet



Application
Transport
Network
Data Link
Physical
Network
Network
Data Link
Data Link
Physical
Physical
Application
Transport
Network
Data Link
Physical
Radio
Medium
DATA
IP Header
DATA
Link Layer Header
IP Header
DATA

hchen@ubuntu: ~/work/edu/csed/networks/ethernet

```
03f0  d1 7f 7d 98 c2 7c f9 16 90 27 2e fd 3f 1f 17 7f    ..}..|...'..?...
0400  fc ff 75 2c 41 3d 36 ea 9a 62 e8 20 02 38 0b 85    ..u,A=6..b. .8..
0410  26 e3 7e 97 b9 8b c6 42 ab 99 93 0d 20 19 e9 32    &.~....B.... ..2
0420  11 07 d3 f5 88 35 ed b8 29 8d 38 06 e1 0a 0c be    .....5..).8.....
0430  ba 5c ca 23 47 92                                  .\.#G.
Captured at interface: ens33 frame from 00:0c:29:0e:bb:7c
0000  00 50 56 c0 00 08 00 0c 29 0e bb 7c 08 00 45 10    .PV.....)..|..E.
0010  0c a8 1a c5 40 00 40 06 7b aa c0 a8 0b 7f c0 a8    ....@.@.{.......
0020  0b 01 00 16 53 9e 7f cc 0d bf 82 2d 14 a4 50 18    ....S......-..P.
0030  01 0f a4 6b 00 00 dd 3d 84 83 44 b2 95 e2 6f 44    ...k...=..D...oD
0040  a5 f9 a6 3b 84 79 98 2c 9b 56 f1 7a 5d 75 4b 63    ...;.y.,.V.z]uKc
0050  74 42 6d b4 db 91 aa 12 53 3f e0 c7 be 38 a8 75    tBm.....S?...8.u
0060  22 e9 cd fc dd ab 8f ea 5a 7f e5 49 e7 52 a7 e9    ".......Z..I.R..
0070  00 ba ce a6 09 fc 2e 3f 8c 50 2f 3e 5b 43 15 fa    .......?.P/>[C..
0080  54 01 a6 e8 f8 fa 56 6c 23 b0 71 40 b5 de c5 64    T.....Vl#.q@...d
0090  14 c7 93 7e e0 87 f6 83 39 35 4e 8f a7 c1 57 59    ...~....95N...WY
00a0  ea e0 54 c9 42 b6 4f 1b 74 6c 8a 82 b3 43 20 a6    ..T.B.O.tl...C .
00b0  ba a4 84 40 0b 6a ee 28 11 7c ea 40 f7 43 c3 c3    ...@.j.(.|.@.C..
00c0  45 42 0c 3c 70 17 15 95 be c8 f0 55 b0 1e f3 8d    EB.<p......U....
00d0  ab 89 d6 b6 f1 55 7c 69 39 e1 02 52 55 40 d2 29    .....U|i9..RU@.)
00e0  4a b9 57 06 db 67 59 2f 84 b2 21 e1 f1 2e e1 e3    J.W..gY/..!.....
00f0  03 95 23 d5 b9 87 02 67 b2 b3 ca 3d 64 87 32 9b    ..#....g...=d.2.
0100  83 4d 92 a7 97 d2 57 b3 51 f5 09 aa 76 71 bd 98    .M....W.Q...vq..
0110  b9 b0 00 4d 4f 41 c1 d5 fc da 8b d5 83 42 81 1c    ...MOA.......B..
0120  dc 0e 81 7f e6 46 9e 06 9f 1f 08 2a c8 04 00 3d    .....F.....*...=
0130  69 bc 79 6b ab 73 91 cc d7 36 74 7a cb 44 16 32    i.yk.s...6tz.D.2
0140  5b 26 b1 75 a5 4c a8 93 64 f0 f7 c2 dd c4 07 21    [&.u.L..d......!
0150  3f 5d b3 96 56 0b e4 57 d4 6c 0c 27 6b ed 4c 7b    ?]..V..W.l.'k.L{
```

Ethernet Protocol Numbers

VIM - /usr/include/net/ethernet.h

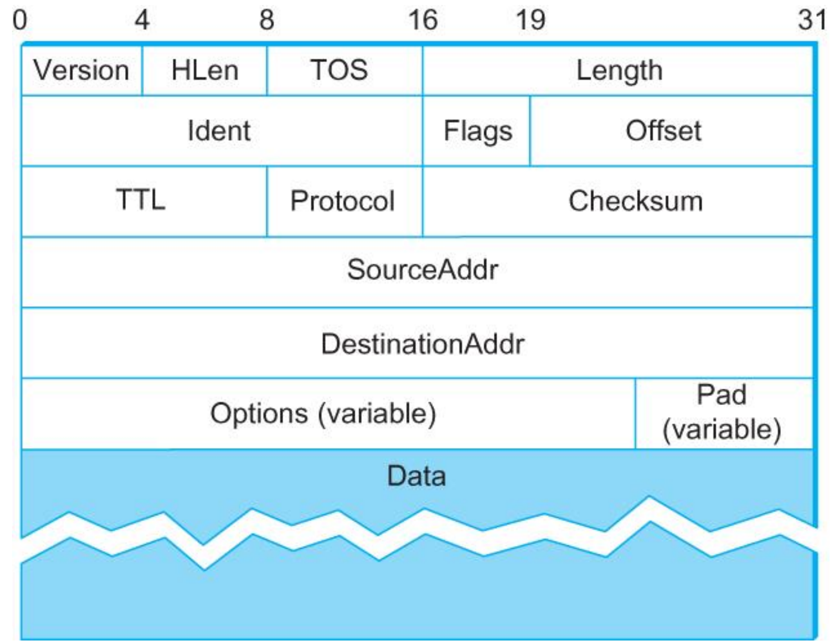
```
/* 10Mb/s ethernet header */
struct ether_header
{
  u_int8_t  ether_dhost[ETH_ALEN];  /* destination eth addr */
  u_int8_t  ether_shost[ETH_ALEN];  /* source ether addr    */
  u_int16_t ether_type;             /* packet type ID field */
} __attribute__ ((__packed__));

/* Ethernet protocol ID's */
#define ETHERTYPE_PUP           0x0200          /* Xerox PUP */
#define ETHERTYPE_SPRITE        0x0500      /* Sprite */
#define ETHERTYPE_IP            0x0800      /* IP */
#define ETHERTYPE_ARP           0x0806      /* Address resolution */
#define ETHERTYPE_REVARP        0x8035      /* Reverse ARP */
#define ETHERTYPE_AT            0x809B      /* AppleTalk protocol */
#define ETHERTYPE_AARP          0x80F3      /* AppleTalk ARP */
#define ETHERTYPE_VLAN          0x8100      /* IEEE 802.1Q VLAN tagging */
#define ETHERTYPE_IPX           0x8137      /* IPX */
#define ETHERTYPE_IPV6          0x86dd      /* IP protocol version 6 */
#define ETHERTYPE_LOOPBACK      0x9000      /* used to test interfaces */


#define ETHER_ADDR_LEN  ETH_ALEN                 /* size of ethernet addr */
"/usr/include/net/ethernet.h" [readonly] 84L, 3221C          49,1          60%
```

Quick Exercise C14a-1

- Below shows a captured Ethernet frame
  - Q1: what is the length in bytes of the largest IP packet?
  - Q2: what is the length in bytes of the smallest IP packet?
  - Q3: consider the Ethernet frame below, what are the values (bits) of each field in the IP packet below (underlined). Which byte is the last byte of this IP packet?



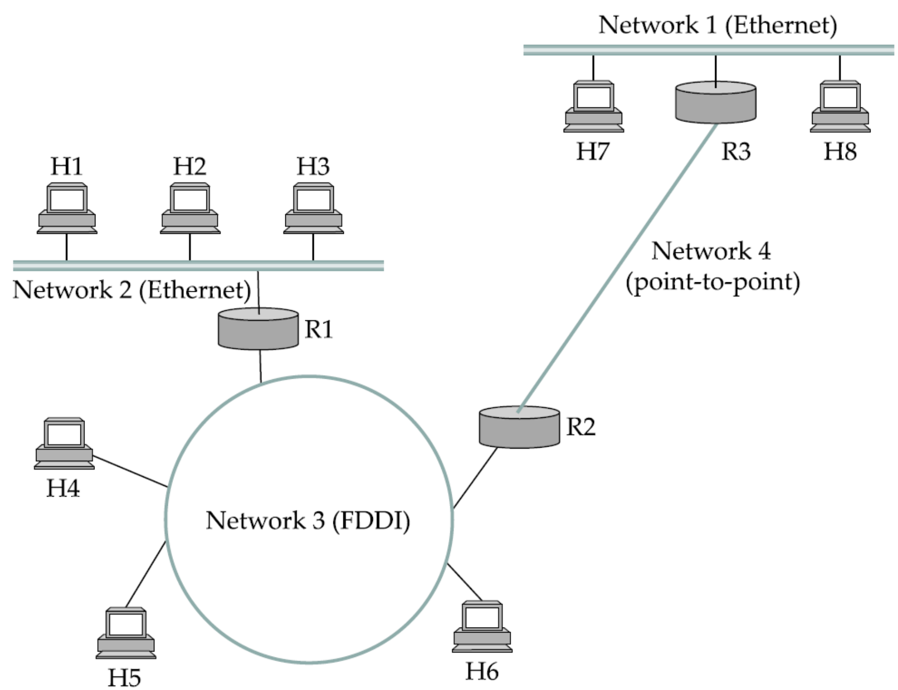

```
0000   00 23 ae 7b 49 11 00 13 72 8f ba 11 08 00 45 00
0010   00 28 78 41 40 00 80 06 fe d4 c0 a8 01 34 c0 a8
0020   01 35 07 e3 00 16 b6 c0 0a da b6 1e 1a b7 50 10
0030   f1 80 a0 30 00 00 00 00 00 00 00 00
```

IP Fragmentation and Reassembly

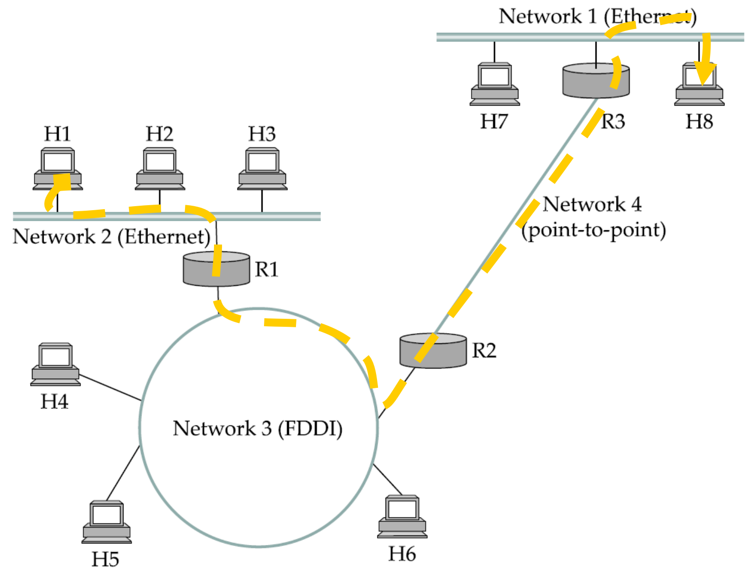
- Each network has some MTU (Maximum Transmission Unit)
  - Ethernet (1500 bytes), FDDI (4500 bytes)
- Strategy
  - Fragmentation occurs in a router when it receives a datagram that it wants to forward over a network which has (MTU < datagram)
  - Reassembly is done at the receiving host
  - All the fragments carry the same identifier in the *Ident* field
  - Fragments are self-contained datagrams
  - IP does not recover from missing fragments

IP Fragmentation and Reassembly: Example

- IP packet
  - Data: 1400 bytes
  - IP header: 20 bytes
- MTU
  - Ethernet=1500
  - FDDI=4500
  - PPP=532

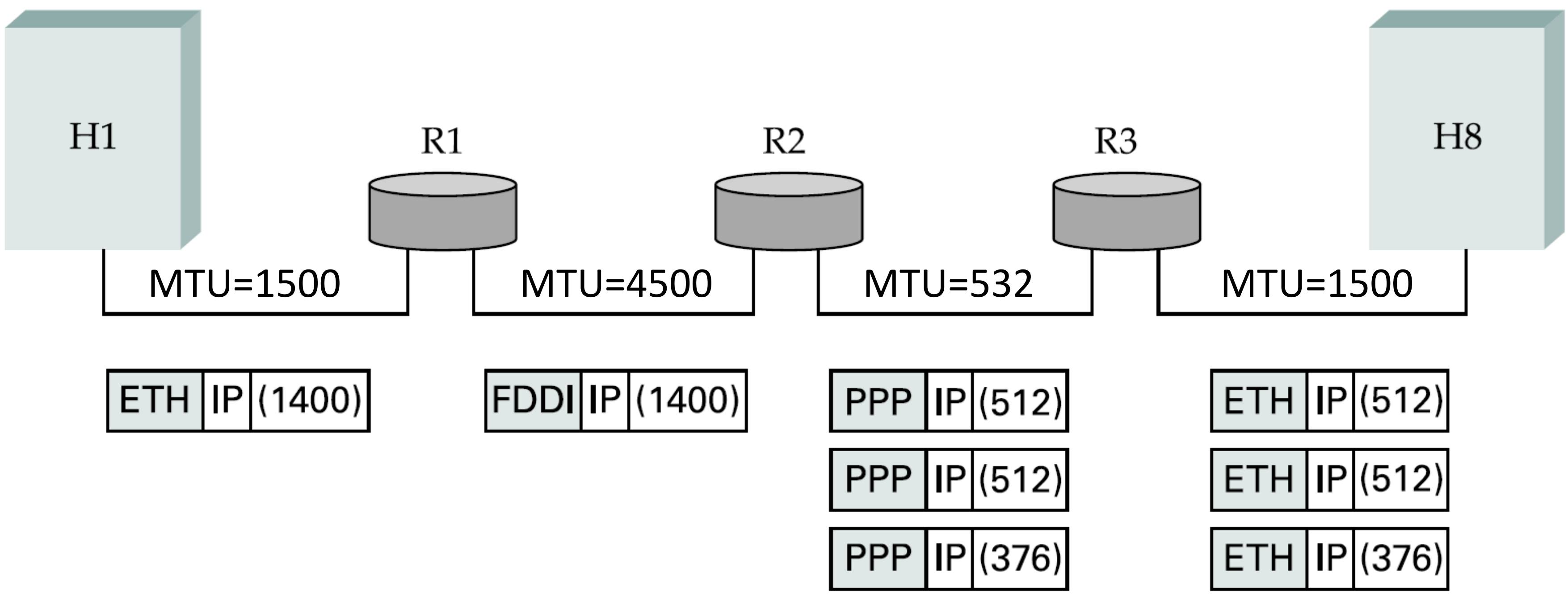


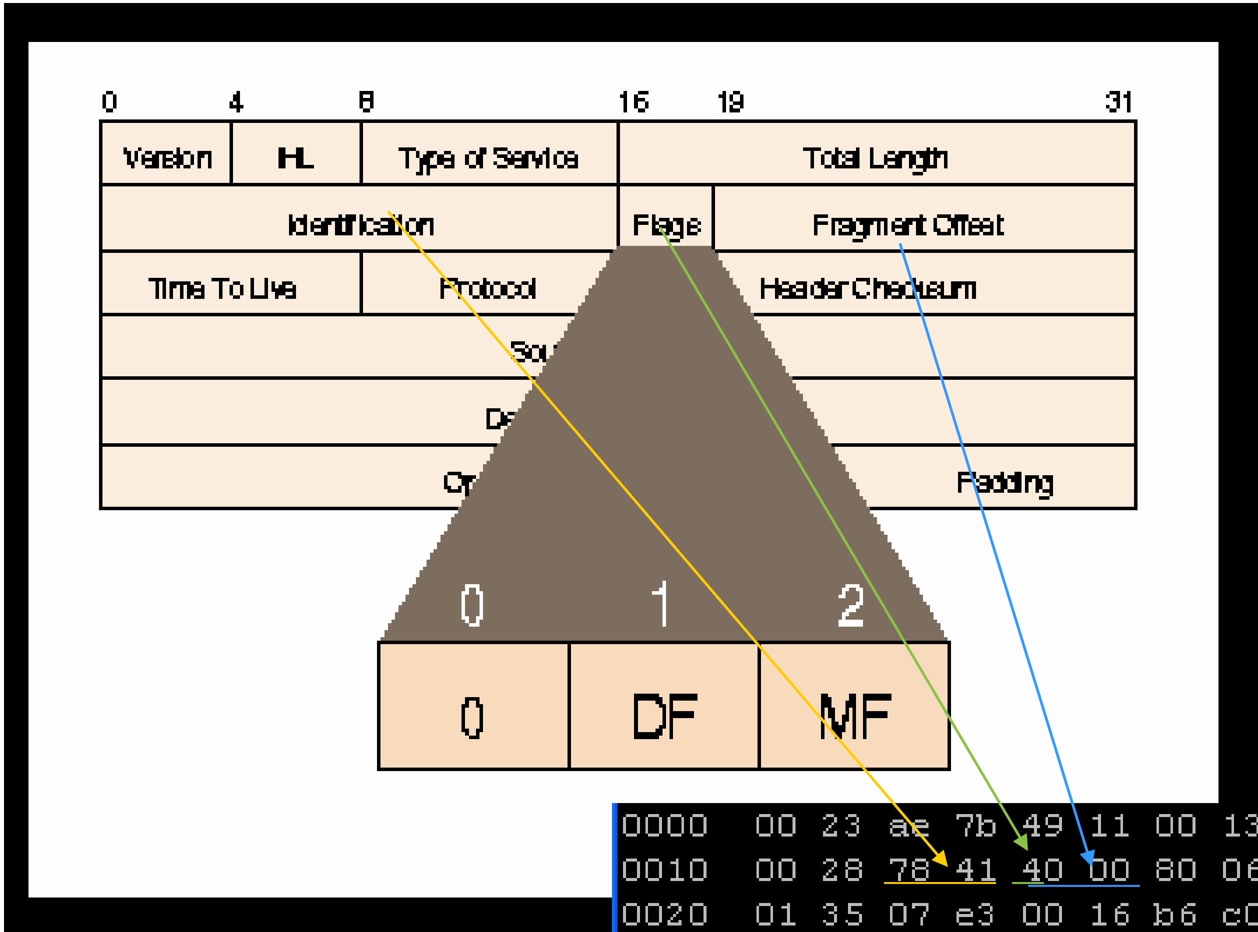
Example



IP packet at H1

- Data: 1400 bytes
- IP header: 20 bytes





IP packet begins

```
0000   00 23 ae 7b 49 11 00 13 72 8f ba 11 08 00 45 00
0010   00 28 78 41 40 00 80 06 fe d4 c0 a8 01 34 c0 a8
0020   01 35 07 e3 00 16 b6 c0 0a da b6 1e 1a b7 50 10
0030   f1 80 a0 30 00 00 00 00 00 00 00 00
```

IP packet ends

Bit 0: reserved, must be zero
Bit 1: (DF) 0 = May Fragment, 1 = **D***on't* **F***ragment*.
Bit 2: (MF) 0 = Last Fragment, 1 = **M***ore* **F***ragments*.
Source: http://www.freesoft.org/CIE/Course/Section3/7.htm

Example

Ident:

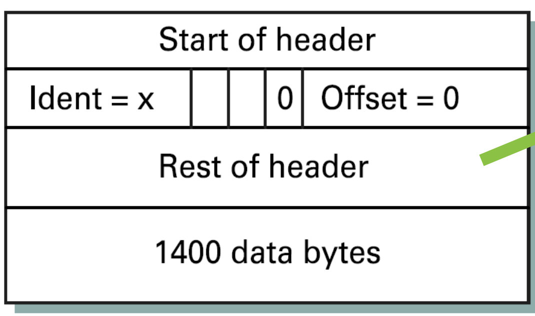
- Same across all fragments
- Unique for each packet

MF (M_{ore} $F_{ragments}$) bit in Flags:

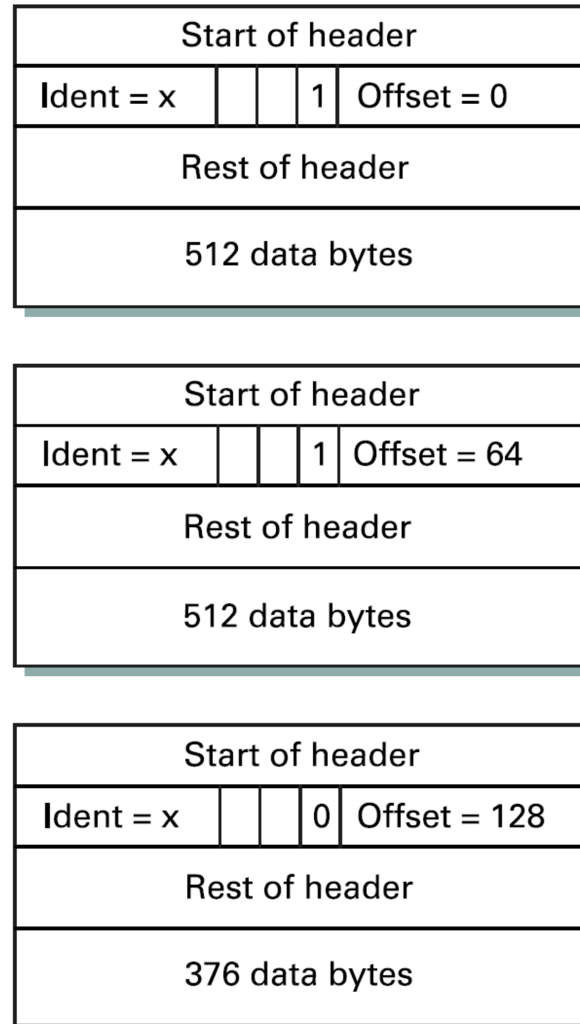
- set → more fragments to follow
- 0 → last fragment

Offset

- in 8-byte chunks

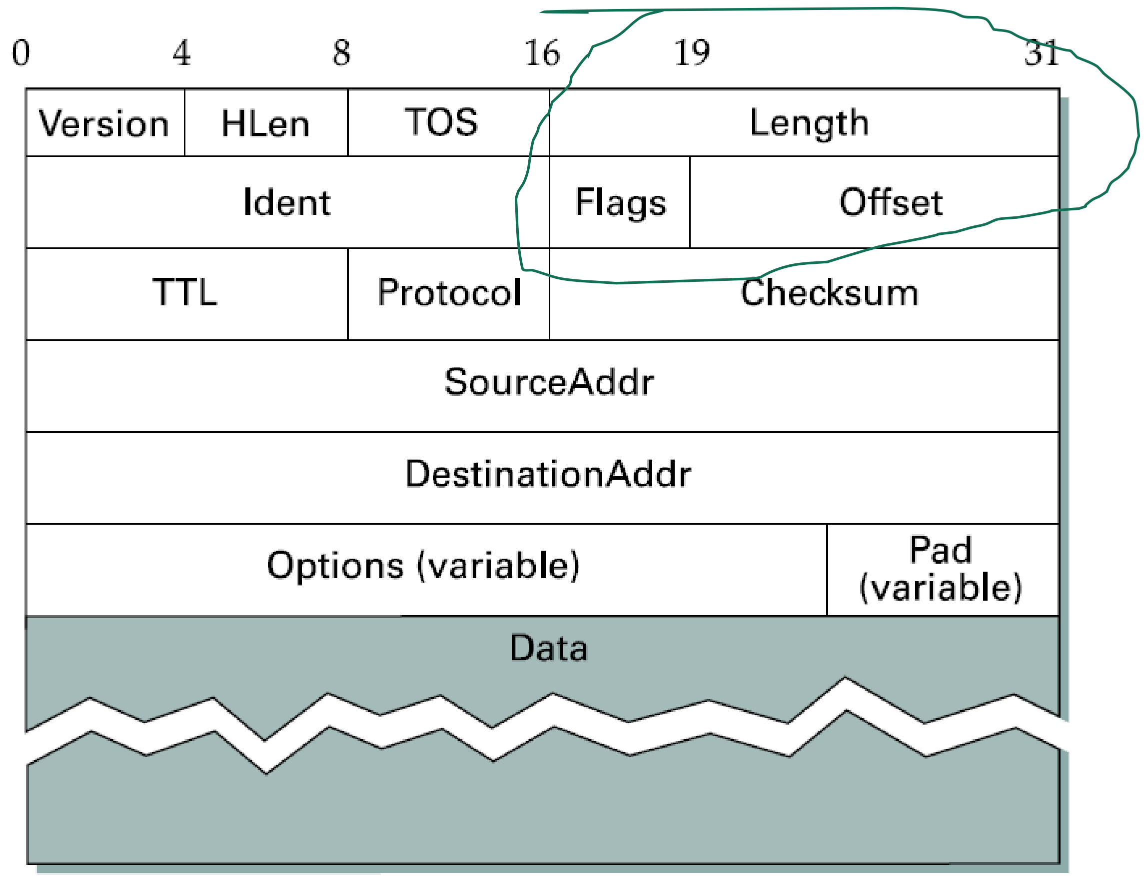


Fragmented into three fragments



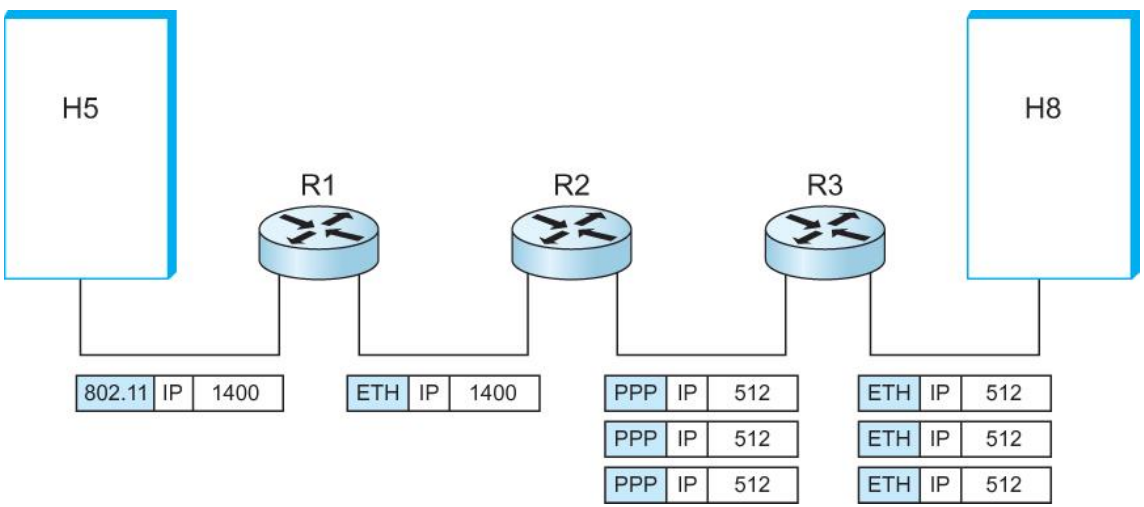
Q: why 8-byte chunks?

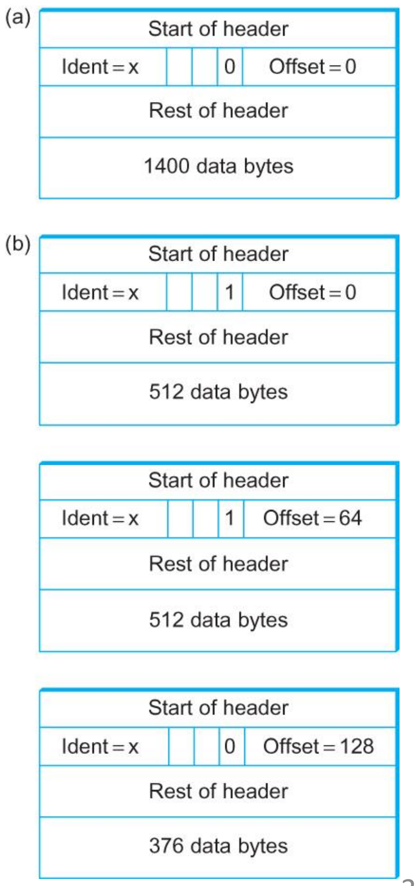
Hint for "*Why 8-byte Chunk?*"



IP Fragmentation and Reassembly

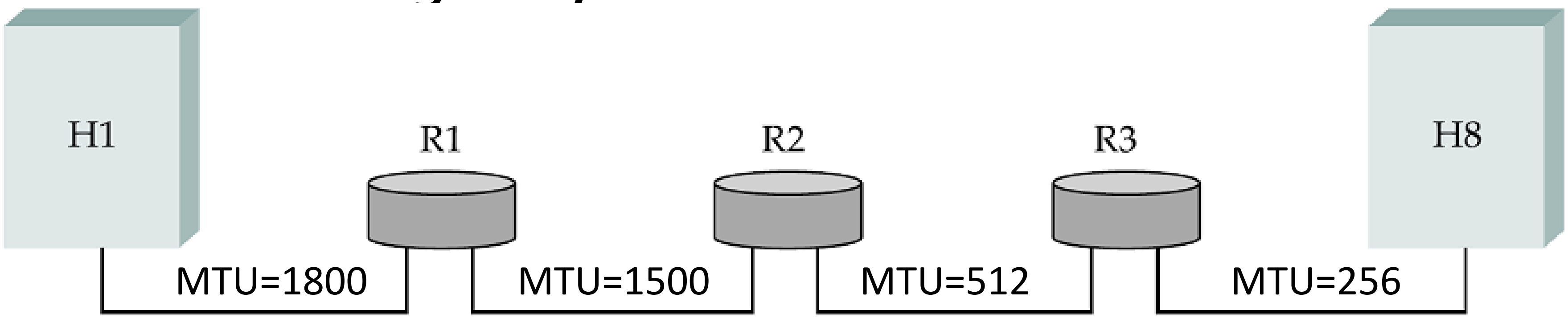
- IP datagrams traversing the sequence of physical networks





Quick Exercise C14a-2

- For an imaginary network below



- Q: H1 sends an IP packet of 1800 bytes including IP header to H8. Please show
  1. IP datagrams traversing the sequence of physical networks graphed above
  2. Header fields of IP datagrams before entering and after leaving each router and hosts

Questions?

- IPv4 packet
- Examining IP packet
- IP fragmentation and assembly

IP Address

- To identify a host on the Internet, use an IP address
- Currently deployed Internet Protocols
  - IP version 4 (IPv4)
  - IP version 6 (IPv6)
  - The very first field in an IP packet indicates the version of IP protocol
  - Globally unique except local networks & private networks
  - Hierarchical (network number + host number)

IPv4 Address

- 32 bit integer
  - Divided into two parts
    - Network number and host number (using prefix or network mask)
- Human-readable form
  - IPv4 numbers-and-dots notation, each number corresponds to a byte in the address
  - Example: 146.245.201.50
- Facing exhaustion of address space, moving to IPv6

IPv4 Private Networks

- Private networks
  - Not routable in a public network
  - 24-bit block 10.0.0.0–10.255.255.255
  - 20-bit block 172.16.0.0–172.31.255.255
  - 16-bit block 192.168.0.0–192.168.255.255

IPv4 Link Local and Loopback Address

- Link local address
  - Not routable
  - For configuration purpose
  - 169.254.0.0/16 (16 bit block: 169.254.0.0 – 169.254.255.255)
- Loopback address
  - Only stay within the host
  - 127.0.0.0/8 (24 bit block: 127.0.0.0 – 127.255.255.255)

Broadcast, Multicast, and Unicast

- The addresses are divided into broadcast, multicast, and unicast address
  - Broadcast address: all 1's in the host number for the network
  - IPv4 Multicast: 224.0.0.0/4 (224.0.0.0 – 239.255.255.255)

A Few IPv4 Address Types

Address Type	Binary Prefix	IPv4 CIDR Notation
Private Network	1100 0000 1010 1000	192.168.0.0/16
	1010 1100 0001	172.16.0.0/12
	1010 0000	10.0.0.0/8
Loopback	0111 1111	127.0.0.0/8
Link-local Unicast	1111 1110 10	169.254.0.0/16
Documentation (TEST-NET-1)	1100 0000 0000 0000 0000 0010	192.0.2.0/24
Documentation (TEST-NET-2)	1100 0110 0011 0011 0110 0100	198.51.100.0/24
Documentation (TEST-NET-3)	1100 1011 0000 0000 0111 0001	203.0.113.0/24
Multicast	1110	224.0.0.0/4
Global Unicast	Everything else (with exceptions)	

IPv6 Address

- 128 bits/16 bytes in length
- IPv6 Notation: a human friendly text representation
- x:x:x:x:x:x:x:x where x is a 16-bit (or 2-byte) hexadecimal number, e.g.,
  - `47CD:1234:4422:ACO2:0022:0022:1234:A456`
- Contiguous 0s can be compressed, e.g.,
  - `47CD:0000:0000:0000:0000:0000:A456:0124`
  - can be written as
  - `47CD::A456:0124`

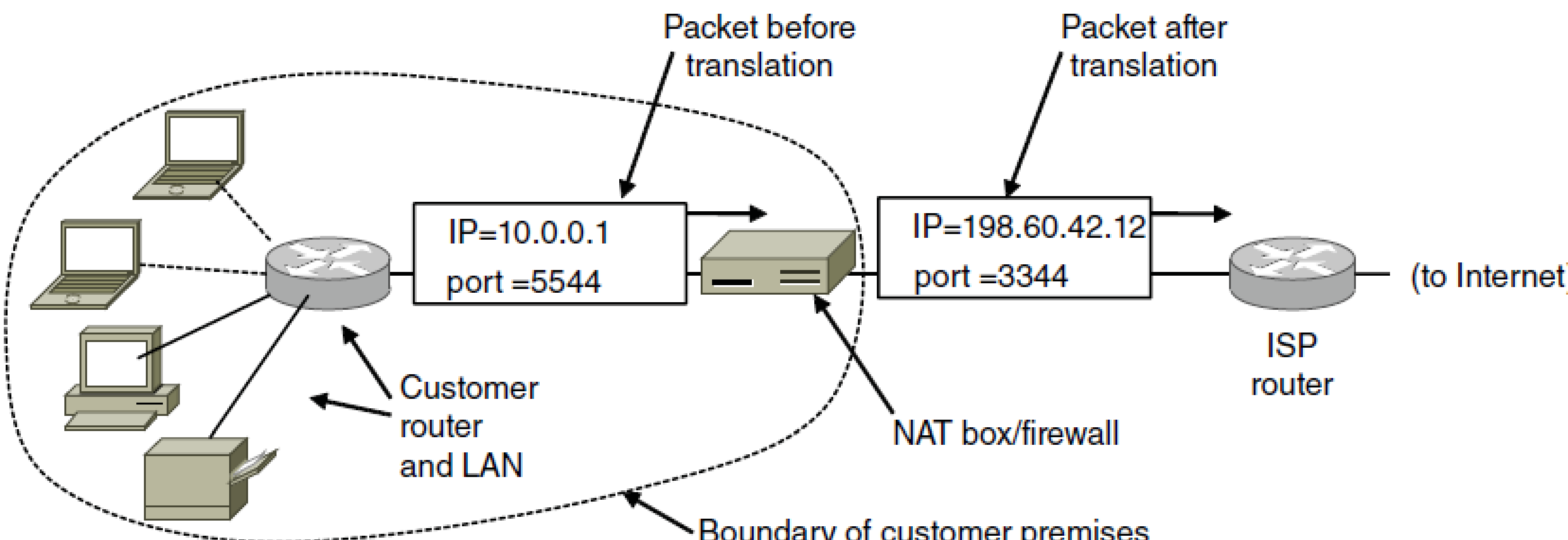
A Few IPv6 Address Types

Address Type	Binary Prefix	IPv6 Notation
Unspecified	00…0 (128 bits)	::/128
Loopback	00…1 (128 bits)	::1/128
Multicast	1111 1111	FF00::/8
Link-local Unicast	1111 1110 10	FE80::/10
Private Network	1111 110	FC00::/7
Documentation	0010 0000 0000 0001 0000 1101 1011 1000	2001:0DB8::/32
Global Unicast	Everything else (with exceptions)	

Network Address Translation

- Network Address Translation and Network Address and Port Translation (NAT or NAPT)
- NAT box maps one external IP address to many internal IP addresses
- Uses additional information to tell connections apart
  - TCP/UDP port
- Violates layering; very common in homes, etc.

NAT: Example


Packet before translation
Packet after translation
IP=10.0.0.1
port =5544
IP=198.60.42.12
port =3344
(to Internet)
ISP router
Customer router and LAN
NAT box/firewall
Boundary of customer premises

Questions?

- IP addresses
- IPv4 and IPv6 addresses
- Different types of IP addresses
- Network address translation

Host Name

- A host may be identified by its name
  - Example: the Domain Name Service (DNS)
- Domain Name Service (DNS)
  - A global name database, and an application on the Internet that does the translation
    - (host name/DNS resolution) Host name → IP address
    - (reverse host name/DNS resolution) IP address → host name
  - Example
    - www.brooklyn.cuny.edu
    - www.google.com
  - Communications are done using IP addresses
    - DNS provides the translation

Look Up Host IP Address

- While on a host, you can look up its IP addresses
- Be aware that a host may have multiple IP addresses
  - an IP address is assigned to a network interface on a host, and a host can have multiple network interfaces
  - a network interface can be assigned multiple IP addresses
- Windows
  - ipconfig
- Mac OS X
  - ifconfig
- Linux
  - ip address or ifconfig

Look Up IP addresses for Host Names

- Use nslookup, available on many operating systems (Windows, Mac OS X, Linux …)
- Use dig on Linux
- Example
  - nslookup www.google.com
  - nslookup www.brooklyn.cuny.edu
  - dig www.google.com

Quick Exercise C14a-3

- Find out IPv4 addresses of following hosts and indicate the class to which the IP addresses belong
  - www.sci.brooklyn.cuny.edu
  - www.drsr.sk
  - www.google.com
- Remark
  - There are many ways to find out the IP address of a host given a domain name
    - Example: nslookup www.sci.brooklyn.cuny.edu
    - (which works on most platforms including Windows, Unix/Linux, and Mac OS X)
  - Convert the first number (from left) to a binary number, then take a look at the 1st, and/or 2nd, and/or 3rd bit

Questions

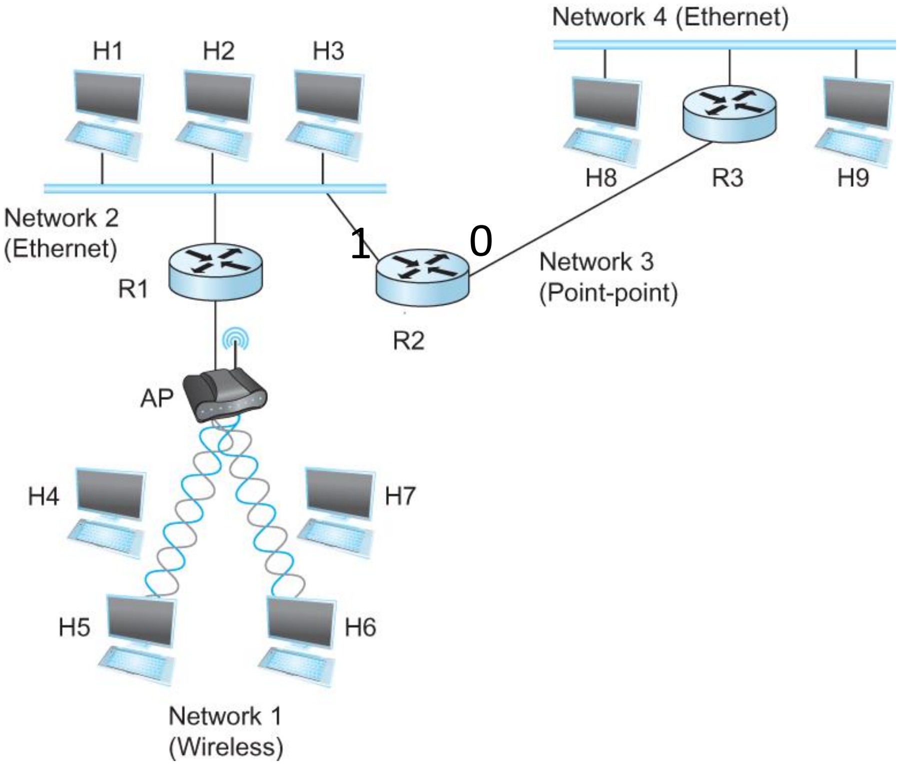
- Hostname and IP addresses

IP Datagram Forwarding

- Strategy
  - every datagram contains destination's address
  - if directly connected to destination network, then forward to host
  - if not directly connected to destination network, then forward to some router
  - forwarding table maps network number into next hop
  - each host has a default router
  - each router maintains a forwarding table

Forwarding Table: Example

- Forwarding table at router R2 that has two interfaces 0 and 1



NetworkNum	NextHop
1	R1
2	Interface 1
3	Interface 0
4	R3

Forwarding Algorithm

- Algorithm

```
if (NetworkNum of destination = NetworkNum of one of
  my interfaces) then
  deliver packet to destination over that interface
else
  if (NetworkNum of destination is in my forwarding
  table) then
       deliver packet to NextHop router
  else
       deliver packet to default router
```

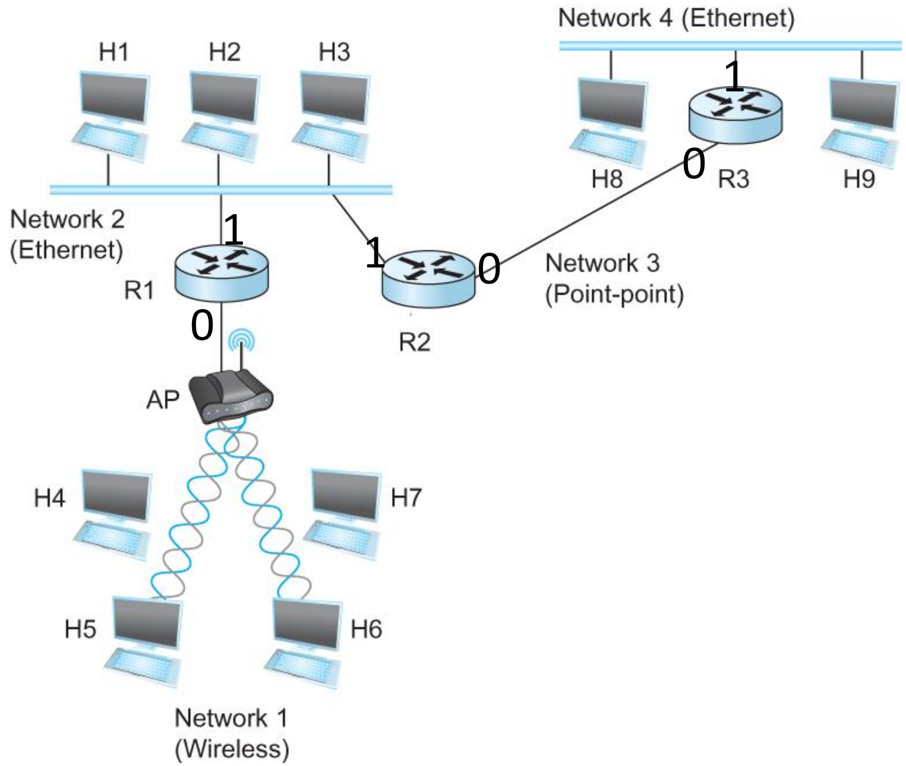
Forwarding Algorithm

- For a host with only one interface and only a default router in its forwarding table, this simplifies to

```
if (NetworkNum of destination = my NetworkNum) then
  deliver packet to destination directly
else
  deliver packet to default router
```

Quick Exercise C14a-4

- Construct forwarding tables for routers R1 and R3. Interfaces of routers are marked

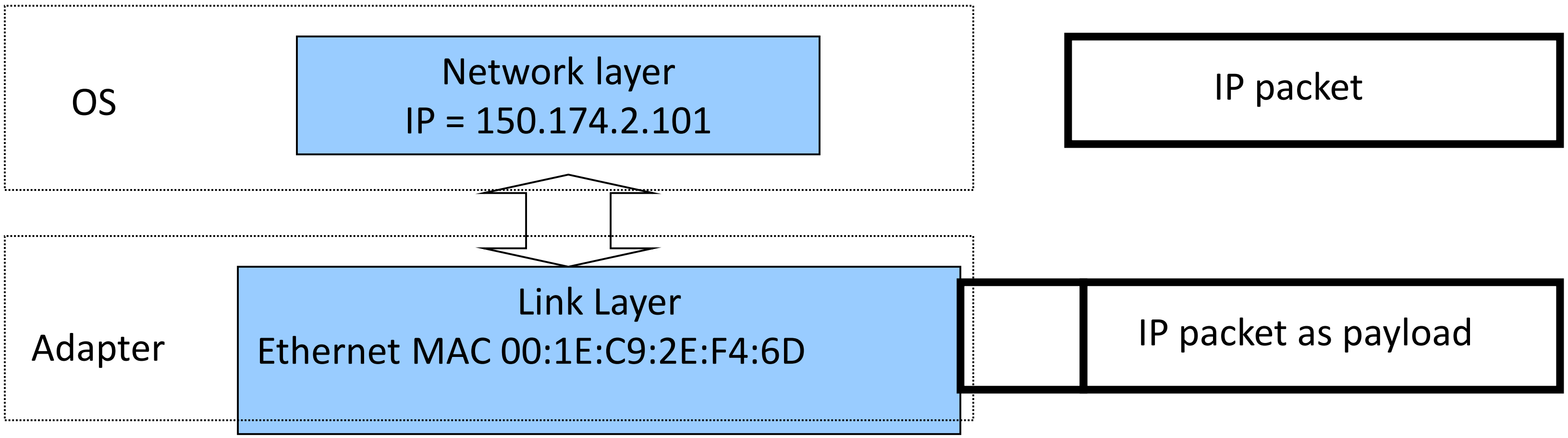


Questions?

- IP datagram forwarding
- Forwarding tables

IP address to Physical Address Translation

- Questions:
  - In an IP network, an IP packet is the payload of one or more Ethernet frames. Who prepares the Ethernet frame headers which contain destination Ethernet/physical address of the destination node?
  - How does the source node know the Ethernet/physical address of the destination node?

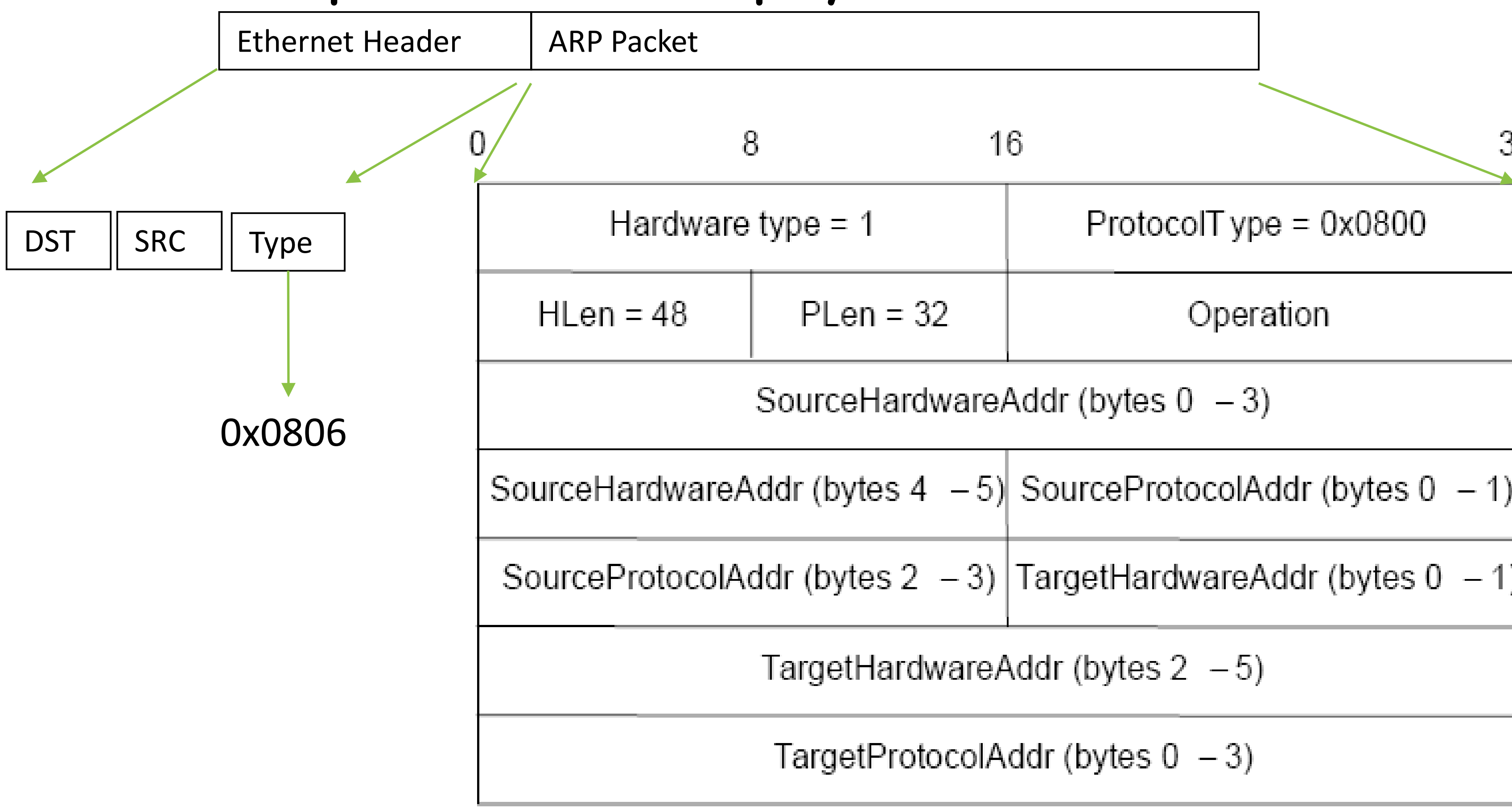


Map IP Addresses into Physical Addresses

- Map IP addresses into physical addresses
  - destination host
  - next hop router
- Techniques
  - encode physical address in host part of IP address
  - table-based
- ARP (Address Resolution Protocol)
  - table of IP to physical address bindings
  - broadcast request if IP address not in table
  - target machine responds with its physical address
  - table entries are discarded if not refreshed

ARP Packet Format

- An ARP packet is the payload of a frame



ARP Packet Format

- HardwareType: type of physical network (e.g., Ethernet)
- ProtocolType: type of higher layer protocol (e.g., IP)
- HLEN & PLEN: length of physical and protocol addresses
- Operation: request or response
- Source/Target Physical/Protocol addresses

0	8	16	31
Hardware type = 1		ProtocolType = 0x0800	
HLen = 48	PLen = 32	Operation	
SourceHardwareAddr (bytes 0–3)			
SourceHardwareAddr (bytes 4–5)		SourceProtocolAddr (bytes 0–1)	
SourceProtocolAddr (bytes 2–3)		TargetHardwareAddr (bytes 0–1)	
TargetHardwareAddr (bytes 2–5)			
TargetProtocolAddr (bytes 0–3)			

ARP Packet: Examples

Ethernet Header	ARP Packet

DST	SRC	Type

Type: 0x0806

Operation (opcode)

- 1 request
- 2 reply
- 3 request reverse
- 4 reply reverse

0	8	16	31
Hardware type = 1		ProtocolT ype = 0x0800	
HLen = 48	PLen = 32	Operation	
SourceHardwareAddr (bytes 0 – 3)			
SourceHardwareAddr (bytes 4 – 5)		SourceProtocolAddr (bytes 0 – 1)	
SourceProtocolAddr (bytes 2 – 3)		TargetHardwareAddr (bytes 0 – 1)	
TargetHardwareAddr (bytes 2 – 5)			
TargetProtocolAddr (bytes 0 – 3)			

```
Interface: eth0
0000  ff ff ff ff ff ff 00 0c 29 a0 51 6d 08 06 00 01   ........).Qm....
0010  08 00 06 04 00 01 00 0c 29 a0 51 6d c0 a8 01 48   ........).Qm...H
0020  00 00 00 00 00 00 c0 a8 01 33                     ..........3
Interface: eth0
0000  ff ff ff ff ff ff 00 0c 29 a0 51 6d 08 06 00 01   ........).Qm....
0010  08 00 06 04 00 01 00 0c 29 a0 51 6d c0 a8 01 48   ........).Qm...H
0020  00 00 00 00 00 00 c0 a8 01 33                     ..........3
```

ARP: Discussion

- Prevent stalled entries
  - Table entries will timeout (~15 minutes)
  - Do not refresh table entries upon reference
- Fresh entries (reset timer)
  - Update table if already have an entry
- Reduce ARP messages
  - Update table with source when you are the target in ARP request messages

ARP Table in Practice (1)

Command Prompt

```
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\hui>arp -a

Interface: 192.168.6.1 --- 0x5
  Internet Address      Physical Address      Type
  192.168.6.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 169.254.69.79 --- 0xa
  Internet Address      Physical Address      Type
  169.254.255.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 169.254.117.23 --- 0xd
  Internet Address      Physical Address      Type
  169.254.255.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
```

ARP Table in Practice (2)

hchen@ubuntu: ~

```
hchen@ubuntu:~$ arp -a
? (192.168.11.254) at 00:50:56:e5:7d:64 [ether] on ens33
? (192.168.11.1) at 00:50:56:c0:00:08 [ether] on ens33
_gateway (192.168.11.2) at 00:50:56:ef:1a:09 [ether] on ens33
hchen@ubuntu:~$
```

Questions?

- Address translation

Host Configuration

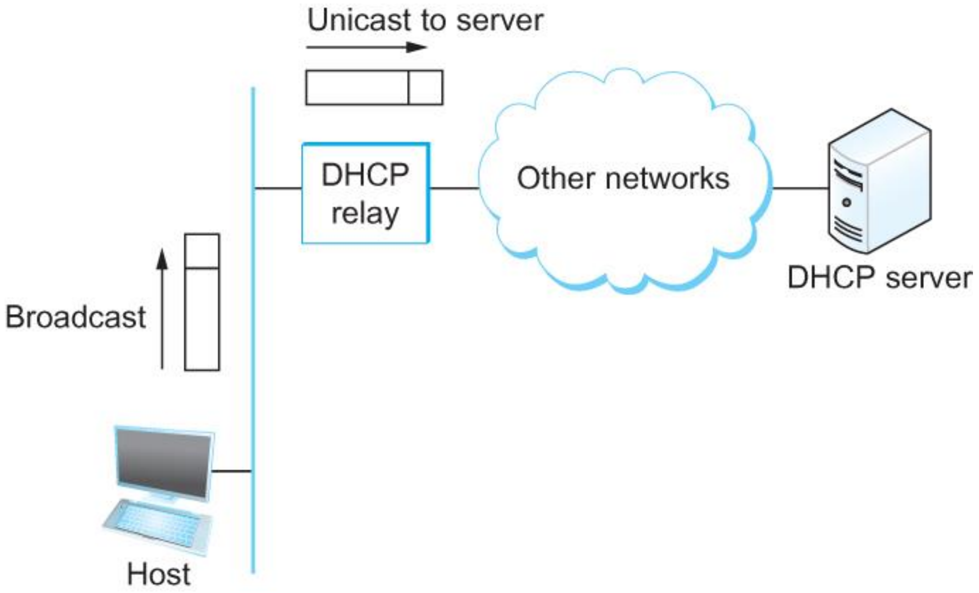
- Ethernet addresses are configured into network by manufacturer and they are unique
- IP addresses must be unique on a given internetwork but also must reflect the structure of the internetwork
- Most host Operating Systems provide a way to manually configure the IP information for the host
- Drawbacks of manual configuration
    - A lot of work to configure all the hosts in a large network
    - Configuration process is error-prune
- Automated Configuration Process is required

Dynamic Host Configuration Protocol (DHCP)

- DHCP server is responsible for providing configuration information to hosts
- There is at least one DHCP server for an administrative domain
- DHCP server maintains a pool of available addresses

DHCP

- Newly booted or attached host sends DHCPDISCOVER message to a special IP address (255.255.255.255)
- DHCP relay agent unicasts the message to DHCP server and waits for the response



IPv4 Host Configuration

- DHCP?

Internet Control Message Protocol (ICMP)

- Defines a collection of error messages that are sent back to the source host whenever a router or host is unable to process an IP datagram successfully
  - Destination host unreachable due to link /node failure
  - Reassembly process failed
  - TTL had reached 0 (so datagrams don't cycle forever)
  - IP header checksum failed
- ICMP-Redirect
  - From router to a source host
  - With a better route information

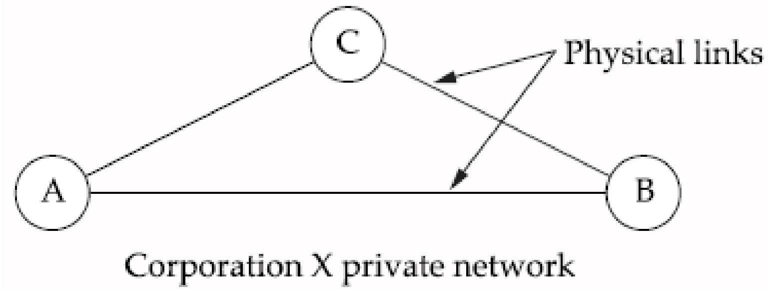
Error Reporting

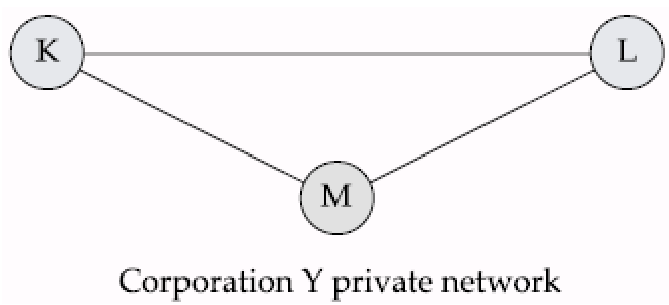
- ICMP?

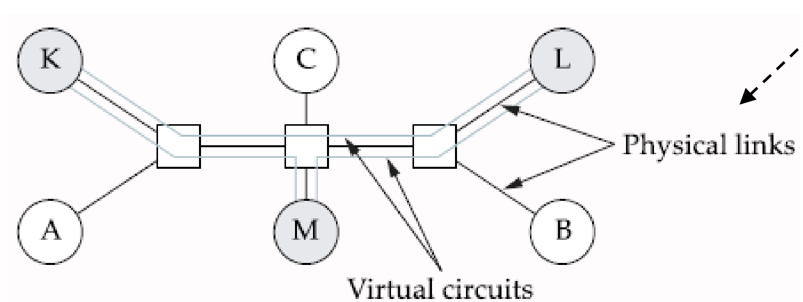
Virtual Networks and Tunnels

- Internetworks often have shared infrastructure networks
- Data packets may not be forwarded without restriction
- Virtual Private Networks (VPN)
  - VPN is a heavily overused and definitions vary
  - An "private" network utilizing an shared network infrastructure

Virtual Private Networks: Example

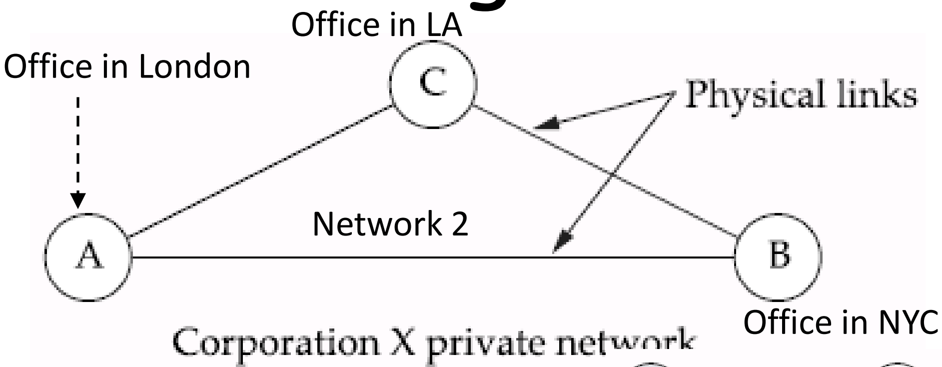


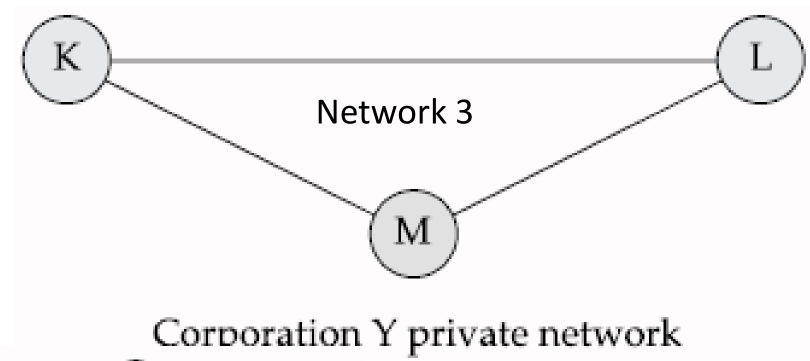


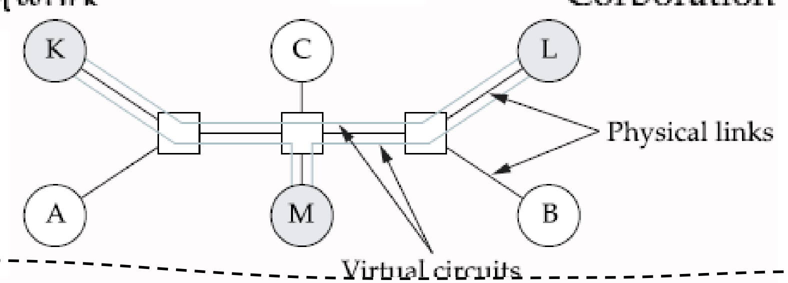


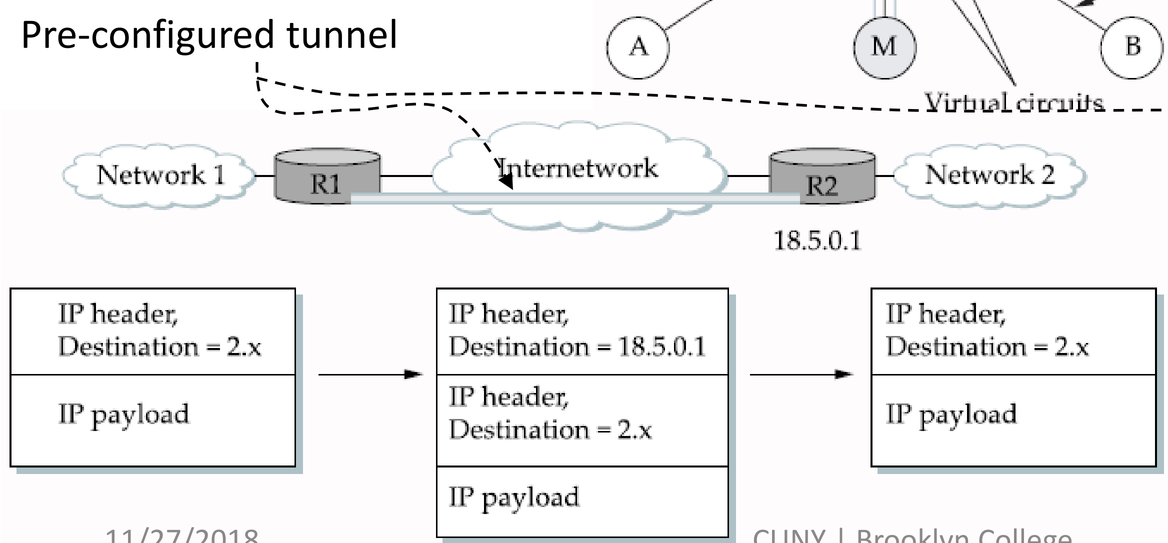
- Corporations X and Y want their own networks via “leased lines” belonging to other networks
- X wants to keep their data private
- So does Y
- X and Y have “virtual” private networks
- “virtualization” can be done on different layers
  - Layer 2 VPN
  - Layer 3 VPN

Virtual Private Networks via IP Tunneling









Forwarding Table at R1

NetworkNum	NextHop
1	Interface 0
2	Virtual interface 0
Default	Interface 1

Questions?

- internet and the Internet
- Global addressing scheme
- Packet fragmentation and assembly
- Best effort service model and datagram forwarding
- Address translation
- Host configuration
- Error reporting
- Tunneling and virtual private networks