

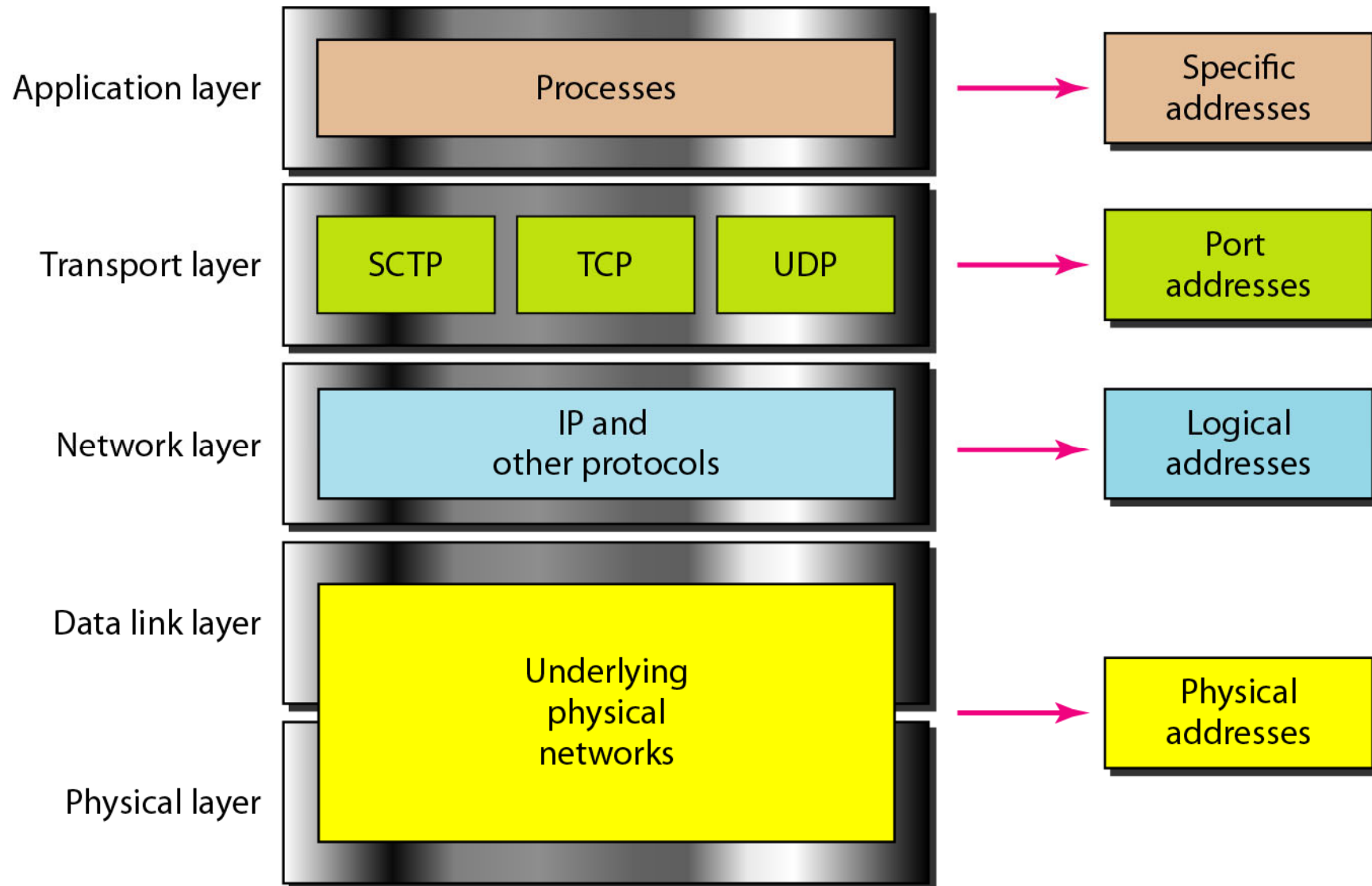
Lecture 5

Network Layer – Data Transfer

ELEC 3506/9506
Communication Networks

Dr Wibowo Hardjawana
School of Electrical and Information Engineering

TCP/IP Protocol & Addressing



Topics for the Day

- Internetworking
- IP service model
- IPv4 datagram format
- IPv4 addressing – Classful Addressing
- Subnetting and Classless Addressing
- Private Addressing
- NAT and Address Translation
- Routing a datagram from Source to Destination
- ICMP
- IPv6

Internetworking

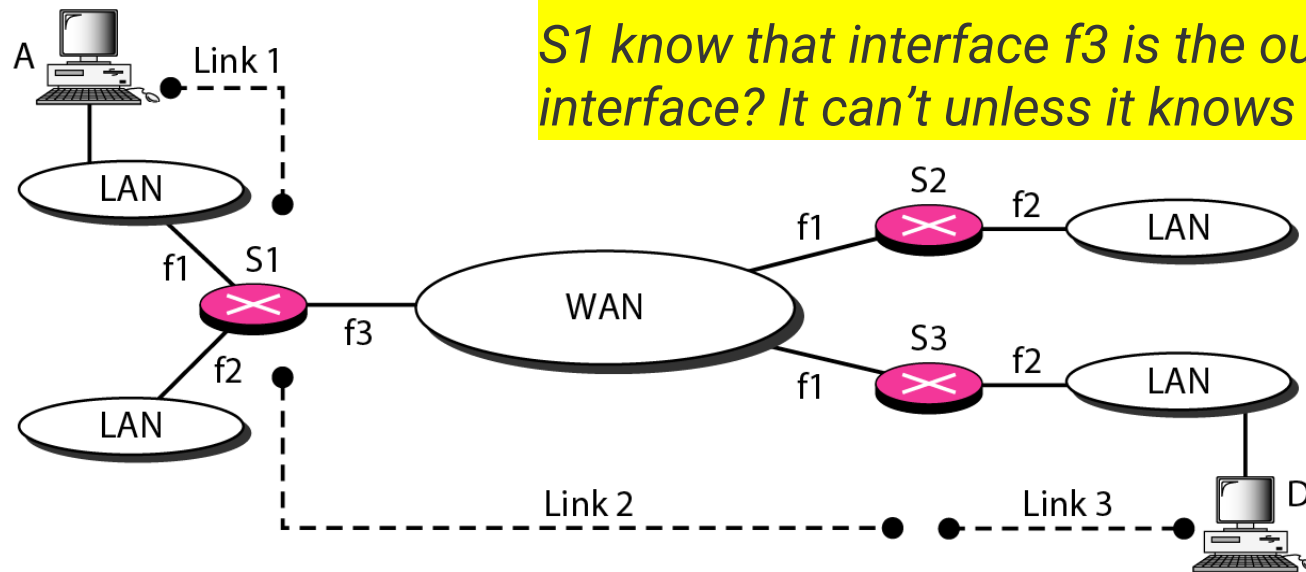
- The term “internetwork” refers to an arbitrary collection of networks interconnected to provide some sort of **host-to-host packet delivery service**
- Network Layer designed to solve the problem of
 - host-to-host delivery
 - Routing packets over networks
- Internetwork example
 - An internetwork is often referred to as a “network of networks” because it is made up of lots of smaller networks

Link Layer vs. Network Layer Delivery

- Hop-to-hop delivery

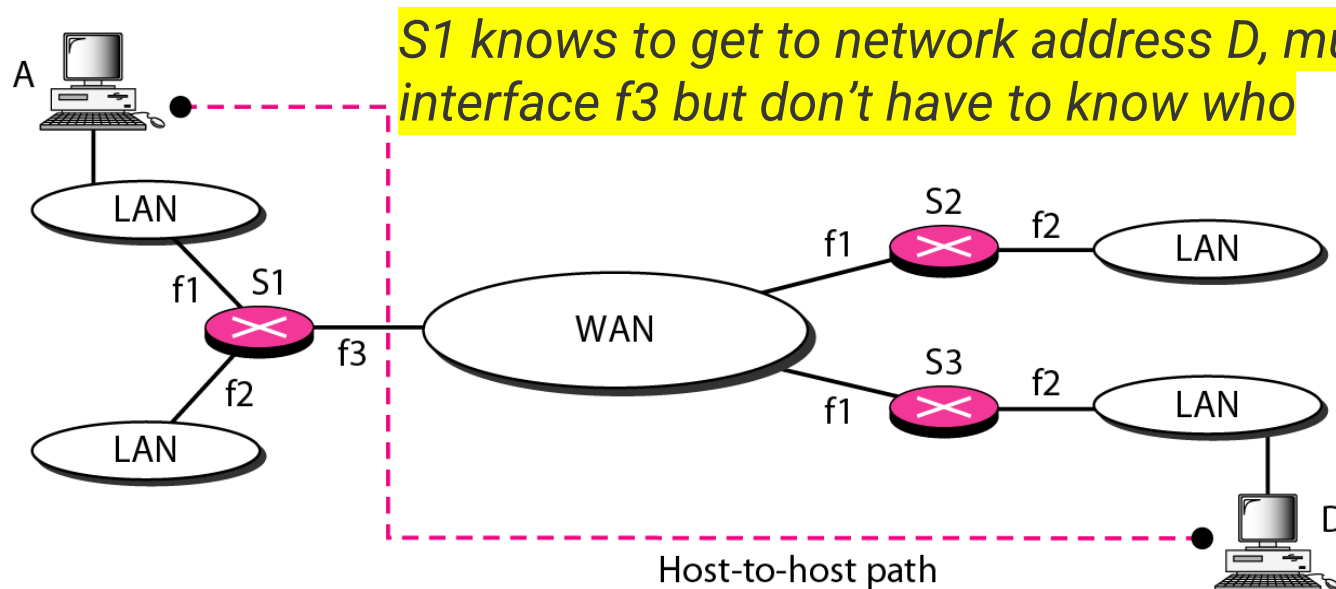


When the data arrives at interface $f1$ of $S1$, how does $S1$ know that interface $f3$ is the outgoing interface? It can't unless it knows MAC address of D .



Link Layer vs. Network Layer Delivery

- End-to-end delivery

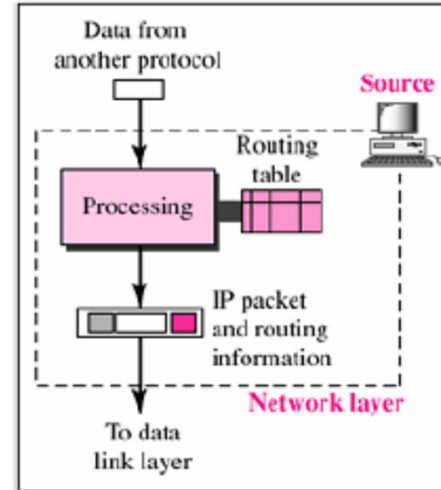


The network layer at source, destination and routers are responsible for host-to-host delivery and for routing the packets through the routers or switches.

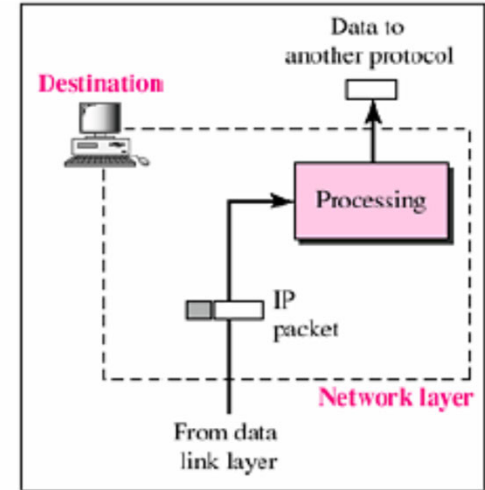
Function of Network Layer

At the source:

- Create a packet for the transport layer packet
- Embed logical addresses of the source and destinations
- check table for routing/outgoing info (interface or next MAC address)



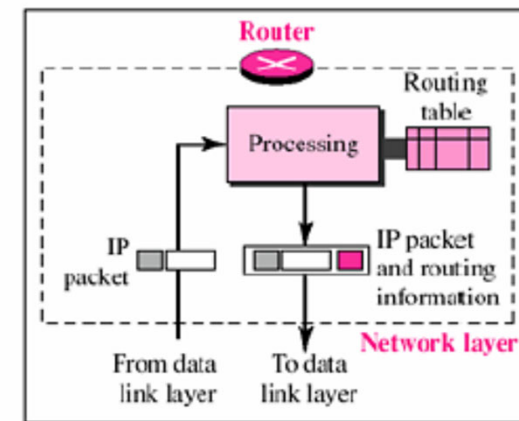
a. Network layer at source



b. Network layer at destination

At the router:

- Check routing table and finds the interface for sending
- Append this info to the packet

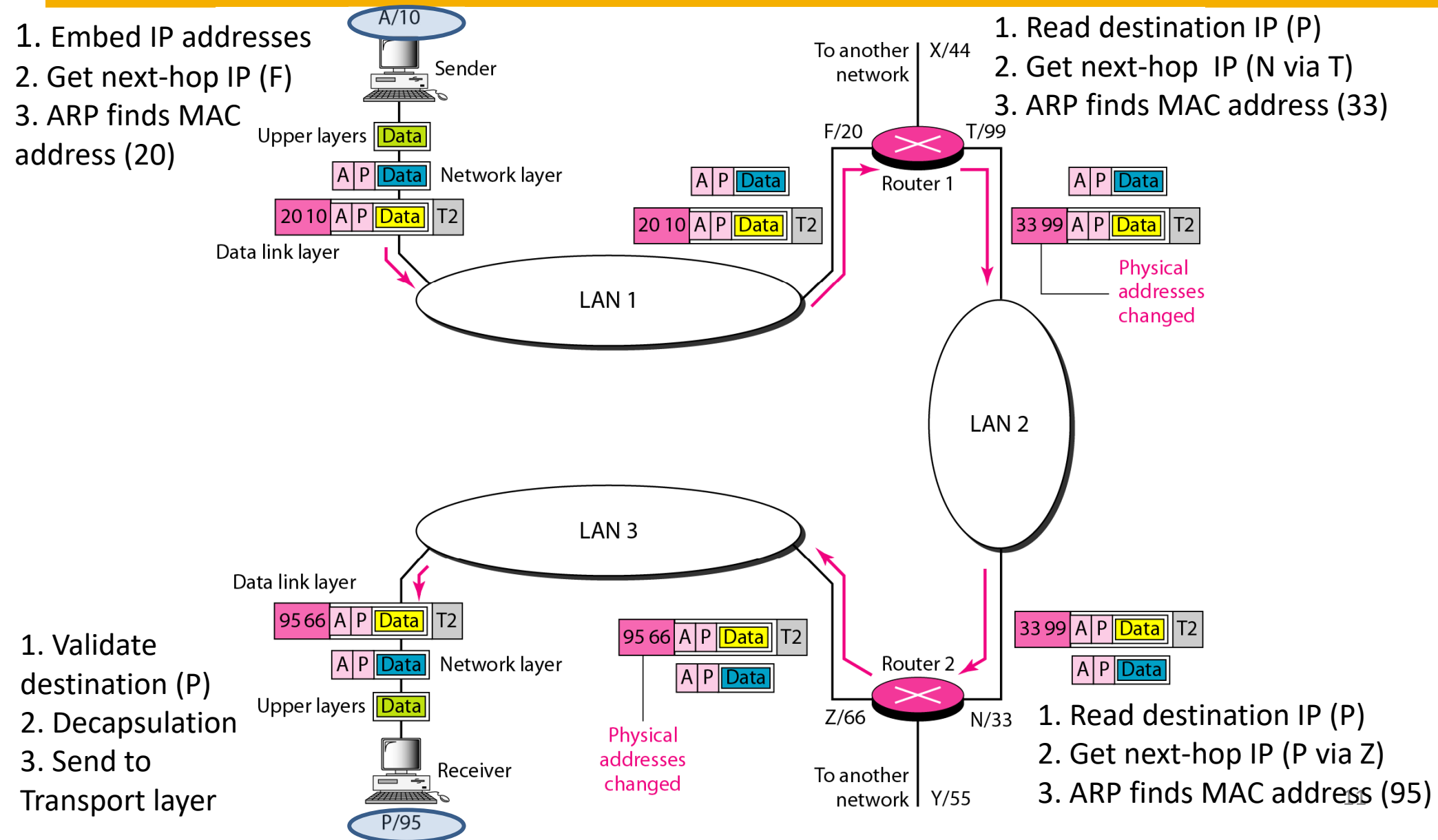


c. Network layer at a router

At the destination:

- Destination address verification
- Wait until all packets fragment is arrived if needed
- Passed to the transport layer

Logical/IP Address and Physical/MAC address

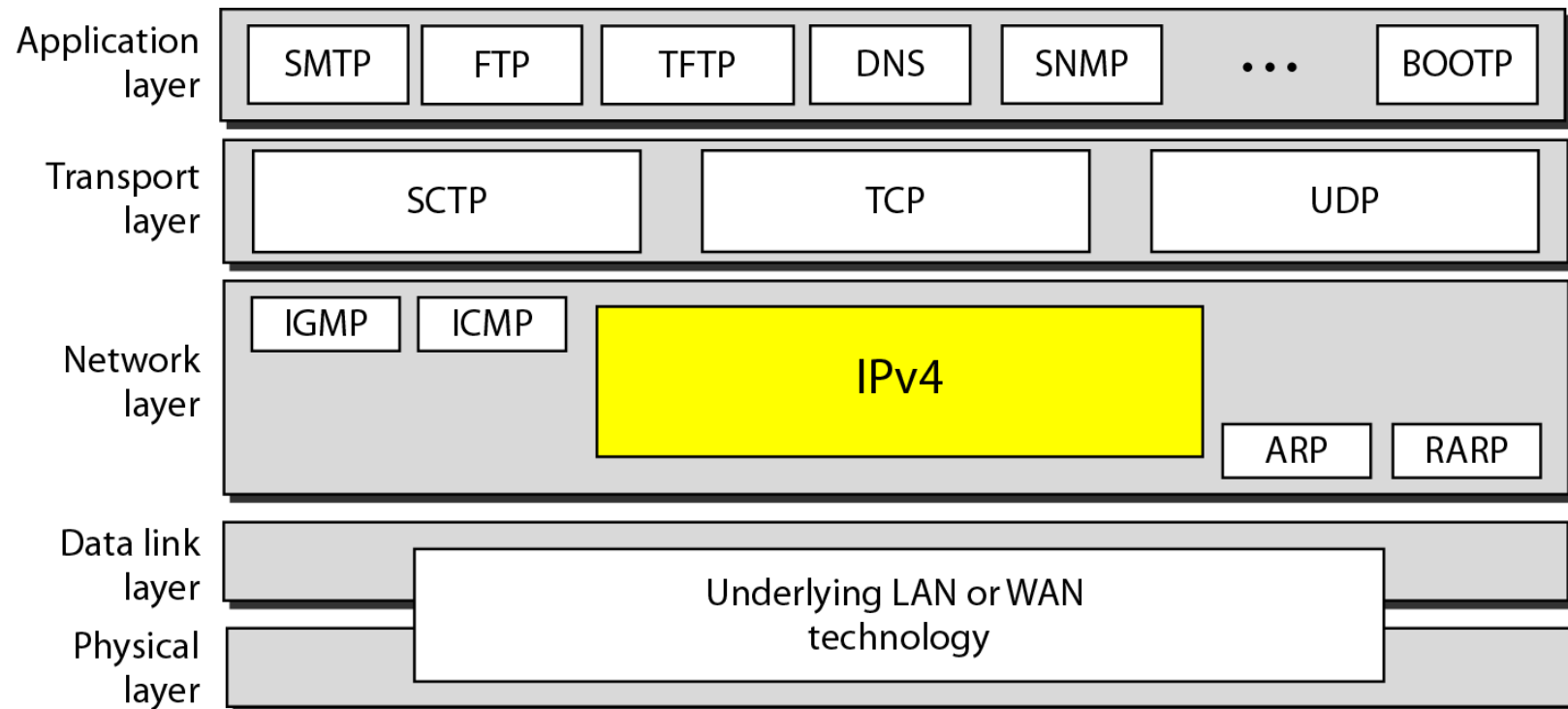


IP Service Model

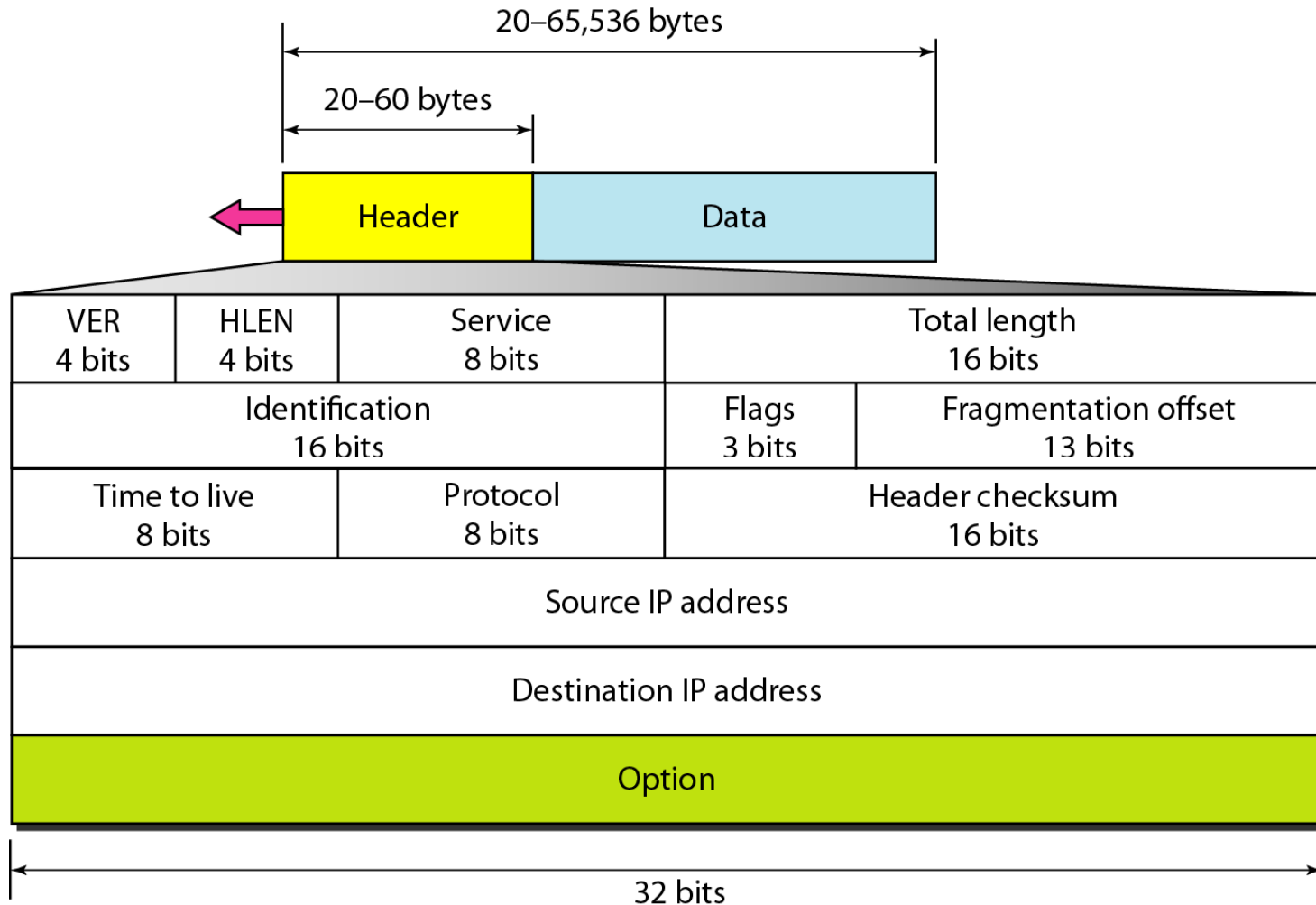
- A **unique addressing scheme** providing a way to identify all hosts in the internetwork
- A **datagram approach** for data delivery
 - **Connectionless service**
 - **Unreliable** - If something goes wrong and the packet gets lost, corrupted, miss-delivered, or in any way fails to reach its intended destination, the network layer does nothing
 - The network layer does **best effort** delivery, **no guarantee**
 - No Flow Control
 - No Error Control
 - It does not make any attempt to recover from the failure

Internet Protocol version 4 (IPv4)

- The **Internet Protocol version 4 (IPv4)** is the delivery mechanism used by the TCP/IP protocols.

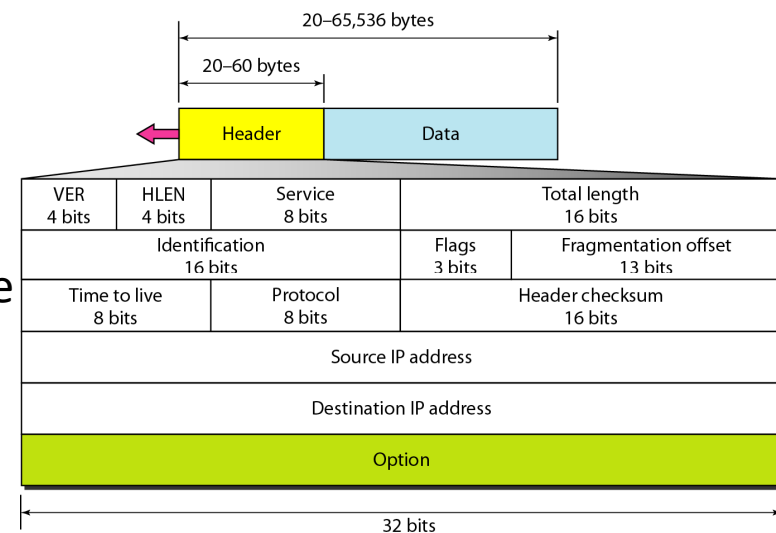


IPv4 Datagram Format



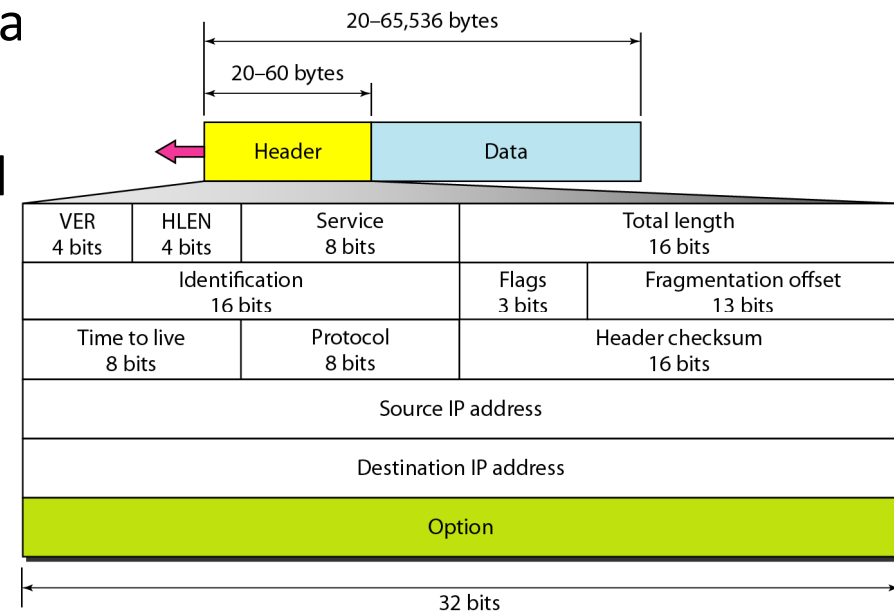
IPv4 Datagram Format

- IP header has a **20-byte fixed part**, followed by an optional part
- IP header fields
 - Version**: IP version
 - Header Length (HLEN)**: find when the header stop, and data starts (per 4 bytes)
 - Type of service**: allows packets to be treated differently based on application needs
 - Total Length**: total datagram length (including header) in bytes
 - Identification**: used in fragmentation. A datagram may be divided into several segments during transmission. When this happens, each fragment is identified as belonging to the same datagram by having the same number in this field
 - Flags**
 - Identify whether the datagram can be fragmented
 - Most left bits is reserved
 - 2nd bit: data can (1) or cant be (0) fragmented
 - 3rd bit: 1 if not the last fragment and more to come
 - 3rd bit: 0 if last and only fragment



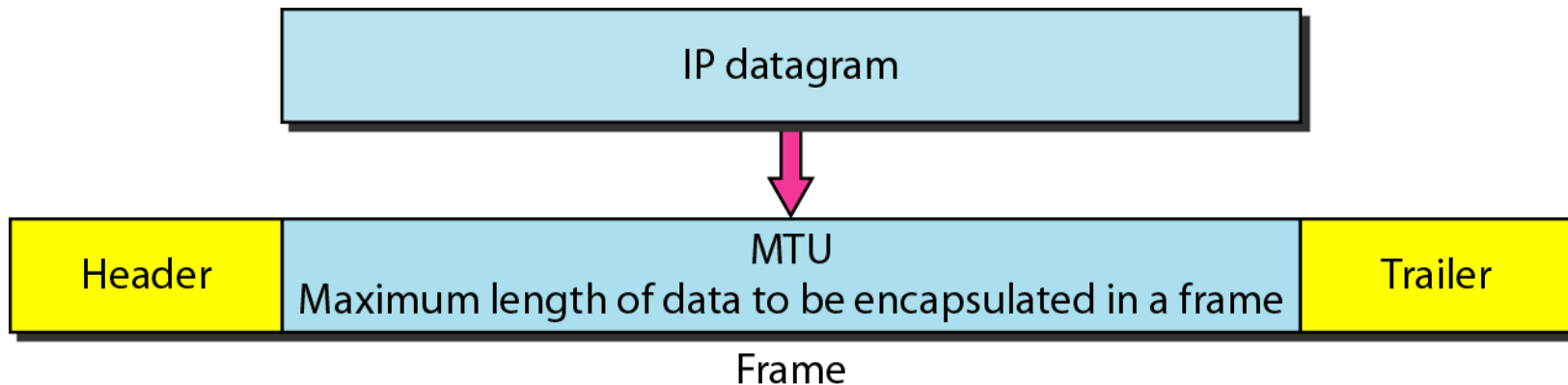
IPv4 Datagram Format

- **Fragmentation Offset**: used in fragmentation. Identify the position of the fragment in the datagram
- **Time-to-live**: defines the number of hops a datagram can travel before it is discarded
- **Protocol**: defines the upper-layer protocol encapsulated (TCP, UDP)
- **Header checksum**: header error check
- **Source/destination address**: 32-bit IP address of source and destination
- **Options**: carry optional field for routing, timing, management, etc.



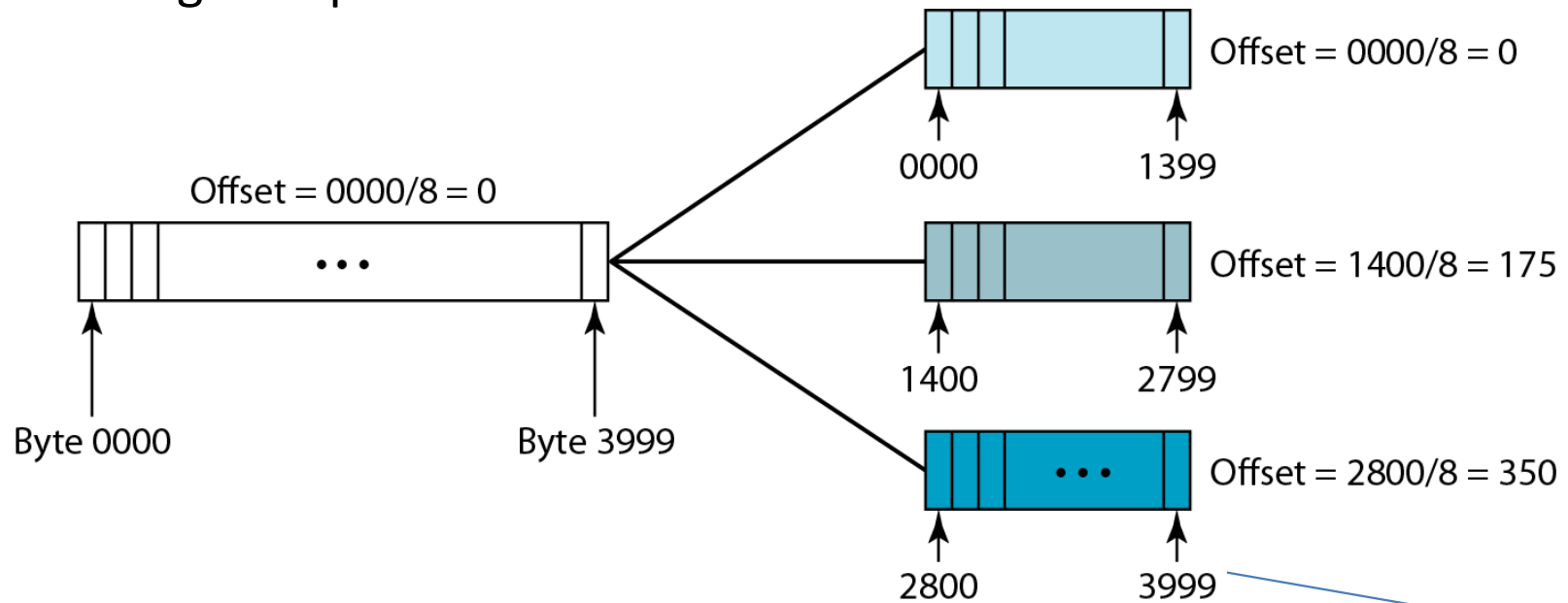
Maximum Transfer Unit (MTU)

- The **maximum transfer unit (MTU)** defines the largest Layer 3 packet that can be forwarded out an interface
- The **MTU allowed** is based on the **data link layer protocol**
 - The **maximum size** of the data portion of the data link frame is the MTU
 - The default MTU is **1500 bytes for Ethernet**



Fragmentation

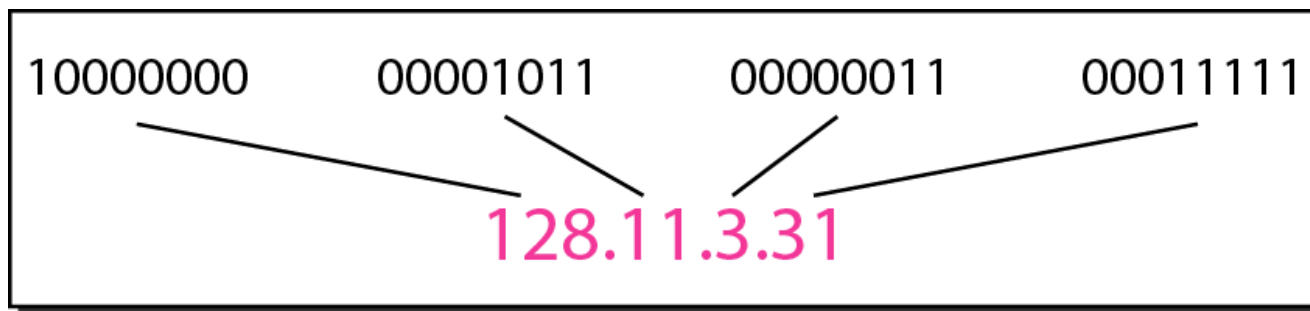
- **Fragmentation** is required if an interface's MTU is smaller than a packet that must be forwarded
 - **Fragmentation** breaks the packet into smaller packets
 - The fragmentation offset shows the relative position of the fragments with respect to the **whole datagram (in units of 8 bytes)**
 - 13 bits long to capture offset values



- Use total length, Flag, HLEN and fragmentation offset to find bytes start and end

IP Addressing

- Each node using the TCP/IP protocol suite has a **unique and universal 32-bit logical IP address**
- The total **address space** of IPv4 is 2^{32} or **4295 Million (Usable 3706 M)**.
- IP address has **two parts**: the **network number** and the **host number**
- Each network listed on the IP internetwork is seen as a single network that must be reached before an individual host within that network can be reached
- The IP address format is known as **dotted-decimal notation**



Classful Addressing

| | First byte | Second byte | Third byte | Fourth byte |
|---------|---------------|----------------|---------------|----------------|
| Class A | 0 | | | |
| Class B | 10 | | | |
| Class C | 110 | | | |
| Class D | 1110 | | | |
| Class E | 1111 | | | |

a. Binary notation

| | First byte | Second byte | Third byte | Fourth byte |
|---------|---------------|----------------|---------------|----------------|
| Class A | 0–127 | | | |
| Class B | 128–191 | | | |
| Class C | 192–223 | | | |
| Class D | 224–239 | | | |
| Class E | 240–255 | | | |

b. Dotted-decimal notation

network number

Classes and Blocks (Networks)

- Number of blocks (networks) and block size in classful IPv4 addressing

| Class | Number of Blocks (Networks) | Block (Network) Size |
|-------|--------------------------------|----------------------|
| A | 128 | 16 777 216 |
| B | 16 384 | 65 536 |
| C | 2 097 152 | 256 |
| D | 1 | 268 435 456 |
| E | 1 | 268 435 456 |

Reserved Addresses

| Address Block | Present Use |
|--------------------------------|------------------------------------|
| 0.0.0.0 to 0.255.255.255 | "This" Network |
| 10.0.0.0 to 10.255.255.255 | Private Use Networks |
| 14.0.0.0 to 14.255.255.255 | Public Data Networks |
| 24.0.0.0 to 24.255.255.255 | Cable TV Networks |
| 39.0.0.0 to 39.255.255.255 | Reserved but subject to allocation |
| 127.0.0.0 to 127.255.255.255 | Loopback |
| 128.0.0.0 to 128.0.255.255 | Reserved but subject to allocation |
| 169.254.0.0 to 169.254.255.255 | Link Local |
| 172.16.0.0 to 172.31.255.255 | Private use networks |

Reserved Addresses

| Address Block | Present Use |
|----------------------------------|---|
| 191.255.0.0 to 191.255.255.255 | Reserved but subject to allocation |
| 192.0.0.0 to 192.0.0.255 | Reserved but subject to allocation |
| 192.0.2.0 to 192.0.2.255 | Test Net |
| 192.88.99.0 to 192.88.99.255 | Relay anycast |
| 192.168.0.0 to 192.168.255.255 | Private Use Networks |
| 198.18.0.0 to 198.18.0.127 | Network interconnect device benchmark testing |
| 223.255.255.0 to 223.255.255.255 | Reserved but subject to allocation |
| 224.0.0.0 to 239.255.255.255 | Multicast |
| 240.0.0.0 to 255.255.255.255 | Reserved for future use |

Examples

Find the class of each address:

a. 227.12.14.87

The first byte is 227 (between 224 and 239); the class is D. $227 = 11100011_2$

b. 252.5.15.111

The first byte is 252 (between 240 and 255); the class is E. $252 = 11111100_2$

c. 134.11.78.56

The first byte is 134 (between 128 and 191); the class is B. $134 = 10000110_2$

Network and Broadcast Addresses

| | |
|-------------------|---|
| Network | The concept of a group of hosts. |
| Network number | A 32-bit number, usually written in dotted decimal form, that represents a network. This number cannot be assigned as an IP address to an interface of some computer. The host portion of the network number has a value of all binary 0s. |
| Network address | Another name for network number. |
| Broadcast address | A 32-bit number, usually written in dotted decimal form, that is used to address all hosts in the network. The host portion of the broadcast address has a value of all binary 1s. Broadcast addresses cannot be assigned as an IP address. |

Example Dissections of IP Addresses, No Subnetting

| | | | Broadcast Address |
|---------------------|-----------------------|----------------|-------------------|
| IP Address (Step 1) | Network Part (Step 2) | Network Number | 8.255.255.255 |
| 8.1.4.5 | 8 | 8.0.0.0 | 130.4.255.255 |
| 130.4.100.1 | 130.4 | 130.4.0.0 | |
| 199.1.1.4 | 199.1.1 | 199.1.1.0 | |
| 172.100.2.2 | 172.100 | 172.100.0.0 | |
| | | | |

A class
B class
C class
Private

Default Masks for Classful Addressing

- A **mask** can help identify the **netid** and **hostid**.
- Performing a bitwise logical **AND** operation between the **IP address** and the **subnet mask** results in the **netid**.

| <i>Class</i> | <i>Binary</i> | <i>Dotted-Decimal</i> |
|--------------|--|-----------------------|
| A | 11111111 00000000 00000000 00000000 | 255.0.0.0 |
| B | 11111111 11111111 00000000 00000000 | 255.255.0.0 |
| C | 11111111 11111111 11111111 00000000 | 255.255.255.0 |

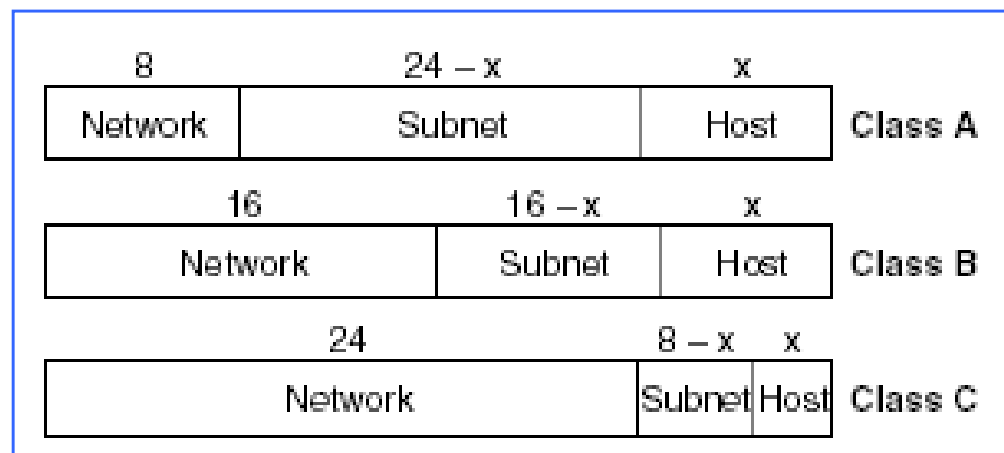
IP address AND subnet mask = Net ID (Network number)

IP Subnetting and Classless Addressing

- **Subnetting** is simply the process of treating **subdivisions of a single Class A, B, or C network** as if it were a network itself
- **Problem with IP address classes**
 - Inflexible, results in waste of IP address
 - Example 1: A network with four hosts will be assigned a Class C network number, the rest address space is wasted
 - Example 2: A network with 257 hosts will be assigned a Class B network number, the rest address space is wasted
- **Without subnetting**, the smallest group is a single, entire Class A, B, or C network number; the IP address space will soon be exhausted

IP Subnetting

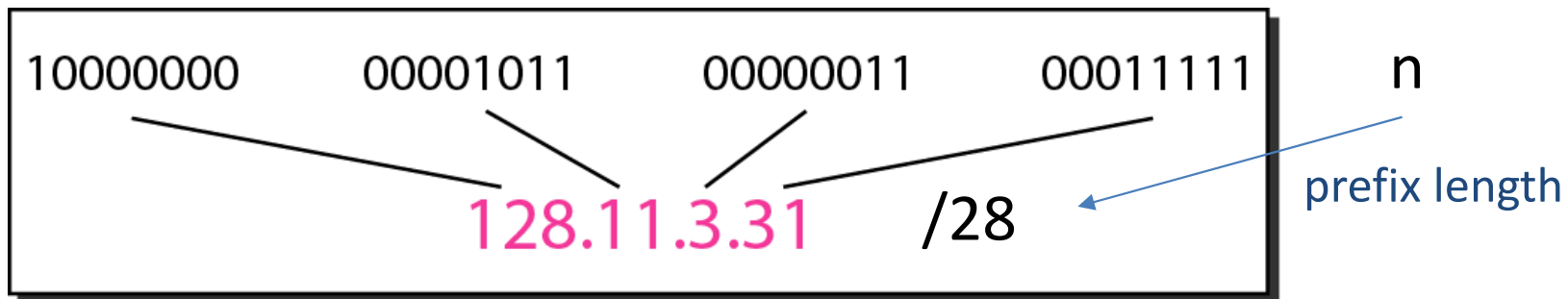
- IP addresses are designed with **two levels** of hierarchy
- With subnetting, **three portions** of the address now exists: **network, subnet and host**



- The number of hosts per subnet: $2^x - 2$
- **Subnet Mask:** defined with a value of n preceded by a slash notation (/n)

Classless Subnetting & Addressing

- Commonly used now to deal with address shortage
- No class definition
- Prefix length “n” to classify IP address belong to a specific domain



- n=28 indicates first 28 bits for network address and thus 4 bits will denote start and end IP address for that domain
- IP addresses from 128.11.3.16 to 128.11.3.31 belong this prefix length/class

Example

- A block of addresses is granted to a small organization. We know that one of the addresses is **205.16.37.36/29**. What is the first, the last, and the total number of addresses of this block? 29 bits for network address
- IP Address : **11001101.00100000.00100101.00100100** (205.16.37.36)
Mask : 11111111.11111111.11111111.1111**000** (255.255.255.248)
Subnet Address : 11001101.00100000.00100101.00100**000** (205.16.37.32)

| | | | | |
|-------------------------|---------------|--------------|-------------|-----------------------|
| SubNet. Add | 11001101 =205 | 00001000 =16 | 00100101=37 | 00100 000 = 32 |
| 1 st Address | 11001101 =205 | 00001000 =16 | 00100101=37 | 00100 001 = 33 |
| 2 nd Address | 11001101 =205 | 00001000 =16 | 00100101=37 | 00100 010 = 34 |
| 3 rd Address | 11001101 =205 | 00001000 =16 | 00100101=37 | 00100 011 = 35 |
| 4 th Address | 11001101 =205 | 00001000 =16 | 00100101=37 | 00100 100 = 36 |
| 5 th Address | 11001101 =205 | 00001000 =16 | 00100101=37 | 00100 101 = 37 |
| 6 th Address | 11001101 =205 | 00001000 =16 | 00100101=37 | 00100 110 = 38 |
| Broadcast Add | 11001101 =205 | 00001000 =16 | 00100101=37 | 00100 111 = 39 |

Example

- A block of addresses is granted to a small organization. We know that one of the addresses is **205.16.37.36/30**. What is the first, the last, and the total number of useable addresses of this block?
- A block of addresses is granted to a small organization. We know that one of the addresses is **192.168.10.64/28**.
- What is the first, the last, and the number of useable addresses of this block?

Example

- A block of addresses is granted to a small organization. We know that one of the addresses is **205.16.37.36/30**. What is the first, the last, and the total number of useable addresses of this block?

205.16.37.36/30 implies mask of 255.255.255.252 with two last bits that can be used
Start from 36=00100100; network address 205.16.37.36; 1st address 205.16.37.37; 2nd (last) address 205.16.37.38; 205.16.37.39 is broadcast address; in total this block has $2^2-2=2$ useable addresses

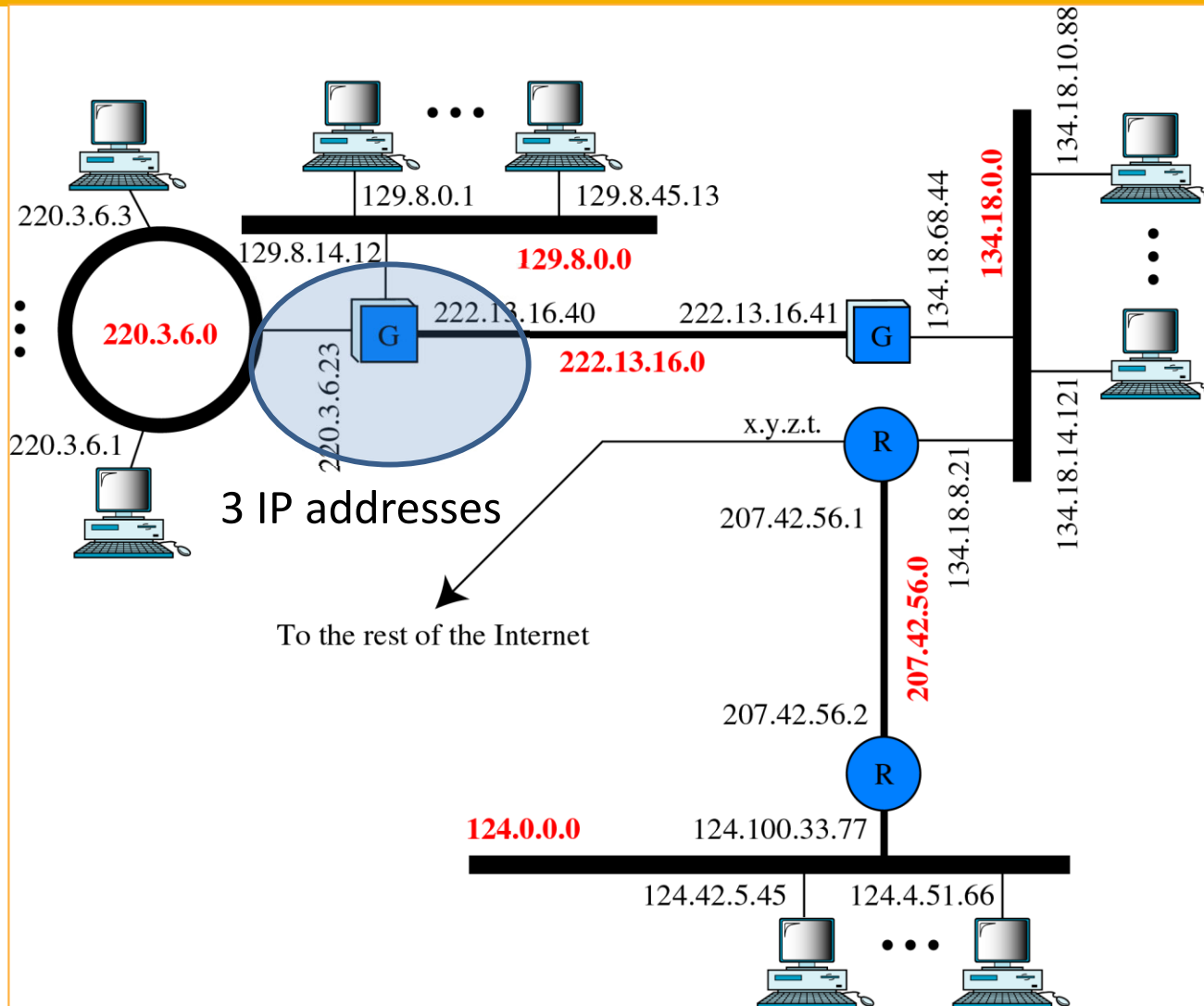
- A block of addresses is granted to a small organization. We know that one of the addresses is **192.168.10.64/28**.
- What is the first, the last, and the number of useable addresses of this block?

192.168.10.64/255.255.255.240; last 6 bits that can be allocated, 64=01000000;
network address 192.168.10.64; 1st address 192.168.10.65; 14th (last) address 192.168.10.78;
192.168.10.79 is broadcast address; in total this block has $2^4-2=14$ useable addresses

Nodes with More Than One Address

- An Internet address defines the node's connection to its network
- It specifies both the network to which a host belongs (netid) and the host itself (hostid)
- Therefore, any device connected to more than one network must have more than one internet address
 - *For example which device?? Router!!!*
- In fact, the device has a different address for each network connected to it

Example



Private Addressing and NAT

- Some organization may only want IP addresses to identify hosts **within networks of the organization**
- These hosts do not need to communicate with the Internet outside the organization
- When designing the IP addressing convention for such a network, any network number can be used
- This kind of network is called **private internet**
- RFC 1918 defines a set of networks that will never be assigned to any organization as a **registered** network number
 - Any organization can use these network numbers
 - No organization is allowed to advertise these networks as route into the Internet

Addresses for Private Networks

- **Advantages** of private addressing
 - The organization saves by applying for a registered IP address segment
 - Saves IP address space
- **Disadvantages**
 - All hosts in the organization must stay away from the outside Internet
- **Network address translation (NAT)** is proposed to address the problem
 - NAT allows a host that do not have a valid registered IP address to communicate with other hosts through the Internet

| <i>Range</i> | | | <i>Total</i> |
|--------------|----|-----------------|--------------|
| 10.0.0.0 | to | 10.255.255.255 | 2^{24} |
| 172.16.0.0 | to | 172.31.255.255 | 2^{20} |
| 192.168.0.0 | to | 192.168.255.255 | 2^{16} |

NAT

- NAT can significantly save IP address space while satisfying the requirement of hosts to be connected to the Internet
- It is widely used by small organizations and/or ISP
- NAT achieves its goal by obtaining several valid registered IP addresses
- These valid registered IP addresses are only assigned to those hosts requiring to communicate through the Internet
- They are assigned on demand

NAT Address Translation

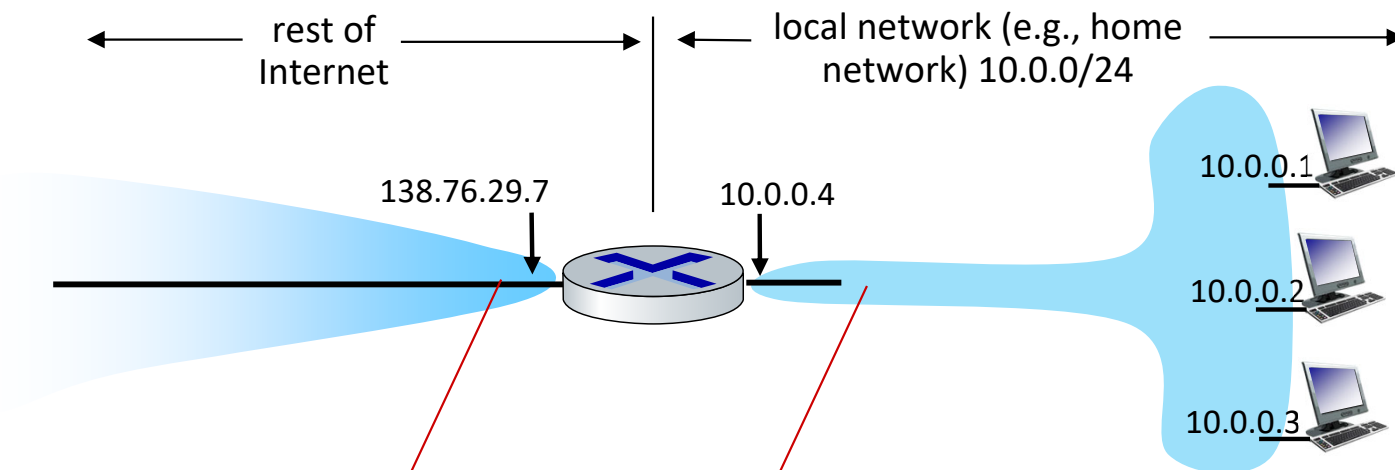
- Using a Pool of IP Addresses:
 - Since the NAT router has only one global address, only one private network host can access the same external host.
 - Using a pool of IP addresses at the router solves this problem
- Using both IP Address and Port Numbers:
 - To allow many-to-many relationship between private network hosts and external server programs, more information is included in the translation tables

socket

| <i>Private Address</i> | <i>Private Port</i> | <i>External Address</i> | <i>External Port</i> | <i>Transport Protocol</i> |
|------------------------|---------------------|-------------------------|----------------------|---------------------------|
| 172.18.3.1 | 1400 | 25.8.3.2 | 80 | TCP |
| 172.18.3.2 | 1401 | 25.8.3.2 | 80 | TCP |
| ... | ... | ... | ... | ... |

NAT: network address translation

NAT: all devices in local network share just **one** IPv4 address as far as outside world is concerned



all datagrams *leaving* local network have *same* source NAT IP address: 138.76.29.7, but *different* source port numbers

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT: network address translation

- all devices in local network have 32-bit addresses in a “private” IP address space (10/8, 172.16/12, 192.168/16 prefixes) that can only be used in local network
- advantages:
 - just **one** IP address needed from provider ISP for *all* devices
 - can change addresses of host in local network without notifying outside world
 - can change ISP without changing addresses of devices in local network
 - security: devices inside local net not directly addressable, visible by outside world

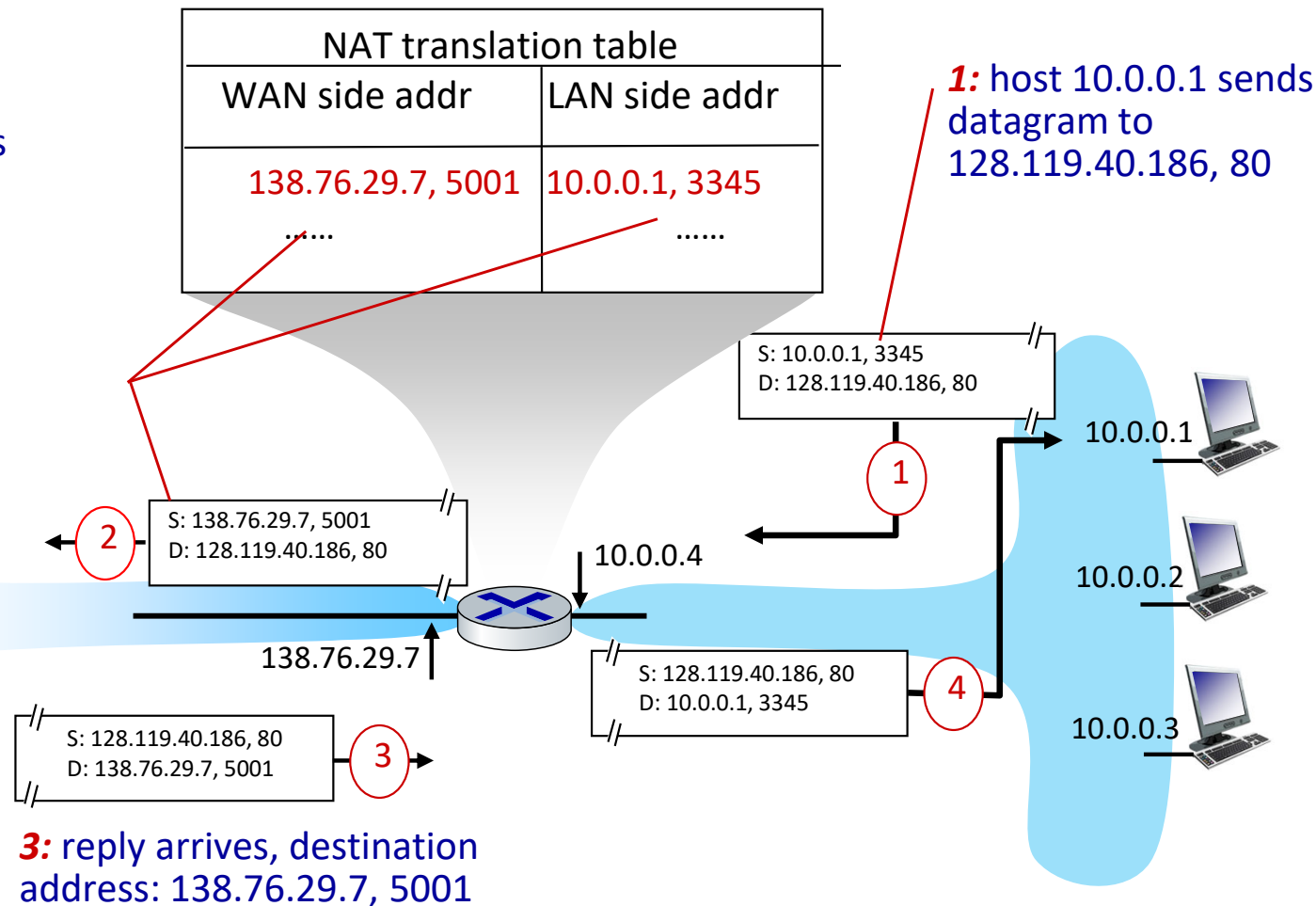
NAT: network address translation

implementation: NAT router must (transparently):

- **outgoing datagrams: replace** (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
 - remote clients/servers will respond using (NAT IP address, new port #) as destination address
- **remember (in NAT translation table)** every (source IP address, port #) to (NAT IP address, new port #) translation pair
- **incoming datagrams: replace** (NAT IP address, new port #) in destination fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: network address translation

2: NAT router changes datagram source address from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

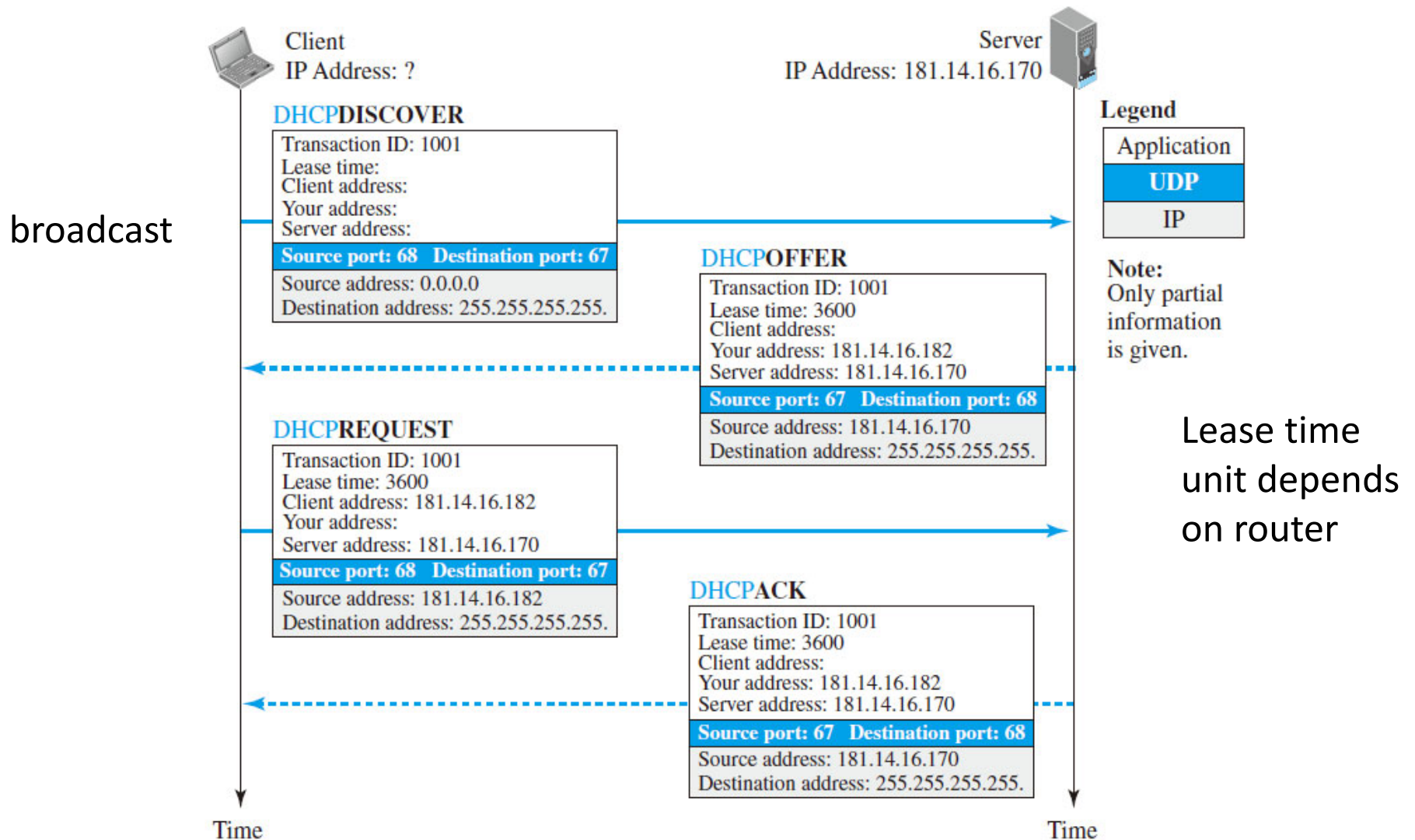


NAT: How to get an IP Address

Hosts (host portion):

- hard-coded by system admin in a file
- **DHCP**: Dynamic Host Configuration Protocol: dynamically get address: “plug-and-play”
 - Assign permanent or temporary IP addresses
 - host broadcasts “**DHCP discover**” msg
 - DHCP server responds with “**DHCP offer**” msg
 - host requests IP address: “**DHCP request**” msg
 - DHCP server sends address: “**DHCP ack**” msg

NAT: How to get an IP Address



Routing a IP datagram from source to destination (same subnet)

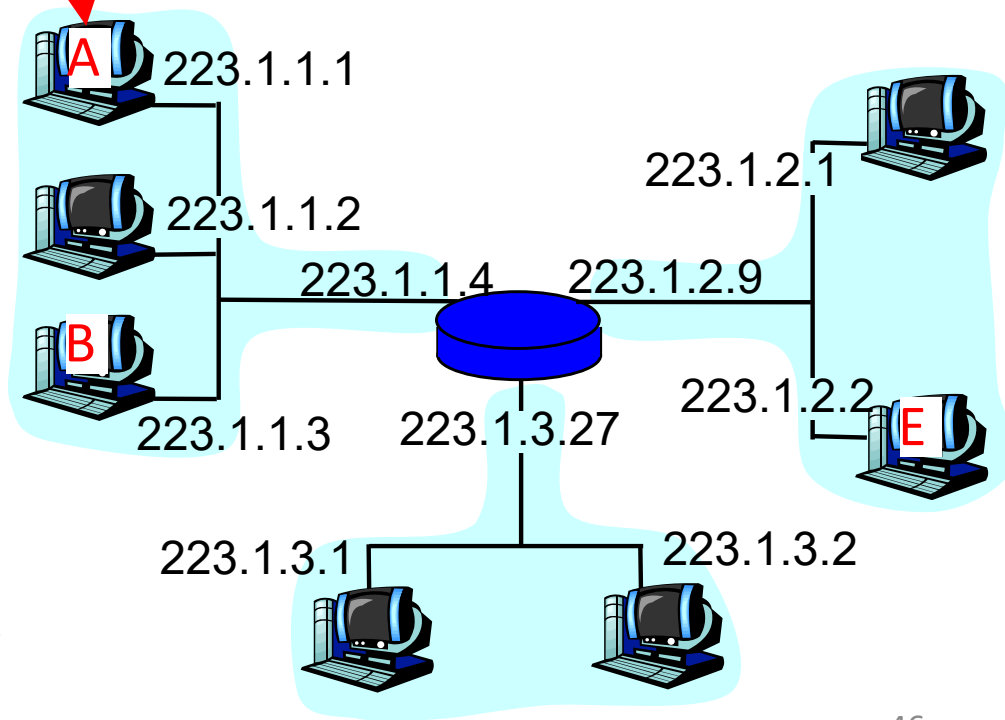
| | | | |
|-------------|-----------|-----------|------|
| misc fields | 223.1.1.1 | 223.1.1.3 | data |
|-------------|-----------|-----------|------|

Starting at A, given IP datagram addressed to B:

- ❑ look up net. IP address of B
- ❑ find B is on same subnet as A
- ❑ link layer will send datagram directly to B inside link-layer frame
 - B and A are directly connected
 - **MAC addresses table** is used

assume the subnet mask is 255.255.255.0

| Dest. Net. | next router | Nhops |
|------------|-------------|-------|
| 223.1.1.0 | | 1 |
| 223.1.2.0 | 223.1.1.4 | 2 |
| 223.1.3.0 | 223.1.1.4 | 2 |



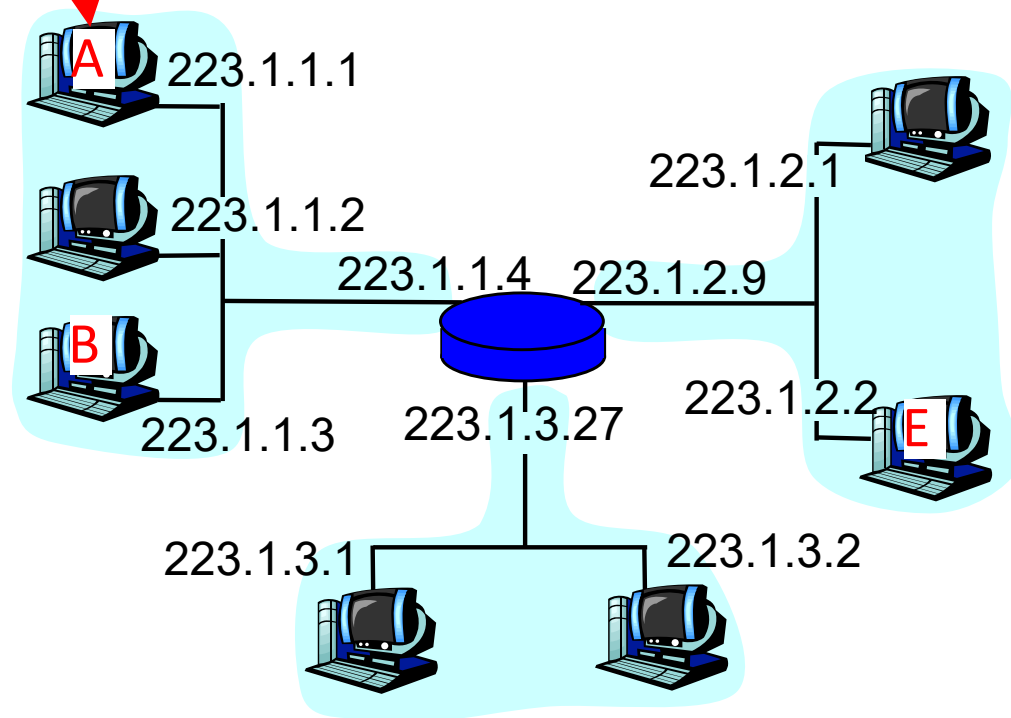
Routing a IP datagram from source to destination (different subnet)

| | | | |
|-------------|-----------|-----------|------|
| misc fields | 223.1.1.1 | 223.1.2.2 | data |
|-------------|-----------|-----------|------|

Starting at A, dest. E:

- ❑ look up network IP address of E
- ❑ E on *different* subnet network
 - A, E not directly attached
- ❑ **routing table**: next hop router to E is 223.1.1.4
- ❑ link layer sends datagram to router 223.1.1.4 inside link-layer frame
- ❑ datagram arrives at IP 223.1.1.4 router interface
- ❑ continued.....

| Dest. Net. | next router | Nhops |
|------------|-------------|-------|
| 223.1.1.0 | | 1 |
| 223.1.2.0 | 223.1.1.4 | 2 |
| 223.1.3.0 | 223.1.1.4 | 2 |



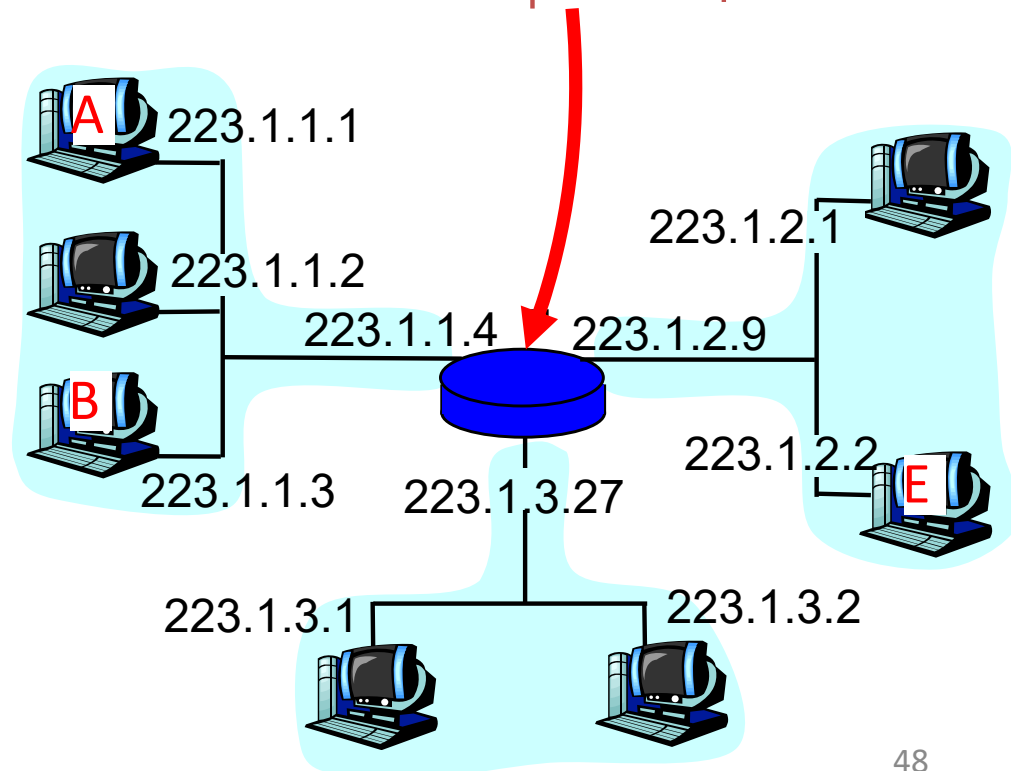
Routing a IP datagram from source to destination (different subnet)

| | | | |
|-------------|-----------|-----------|------|
| misc fields | 223.1.1.1 | 223.1.2.2 | data |
|-------------|-----------|-----------|------|

Arriving at 223.1.4, destined for 223.1.2.2

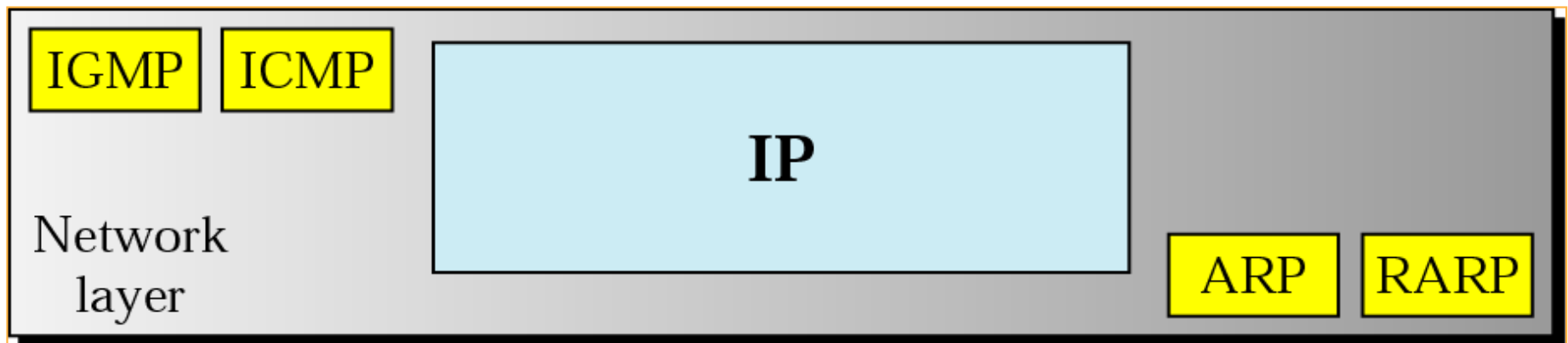
- ❑ look up network IP address of E
- ❑ E on *same* network as router's interface 223.1.2.9
 - router, E directly attached
- ❑ link layer sends datagram to 223.1.2.2 inside link-layer frame via interface 223.1.2.9
- ❑ datagram arrives at 223.1.2.2 by E's MAC address !!! (hooray!)

| Dest. network | next router | Nhops | interface |
|---------------|-------------|-------|------------|
| 223.1.1.0 | - | 1 | 223.1.1.4 |
| 223.1.2.0 | - | 1 | 223.1.2.9 |
| 223.1.3.0 | - | 1 | 223.1.3.27 |



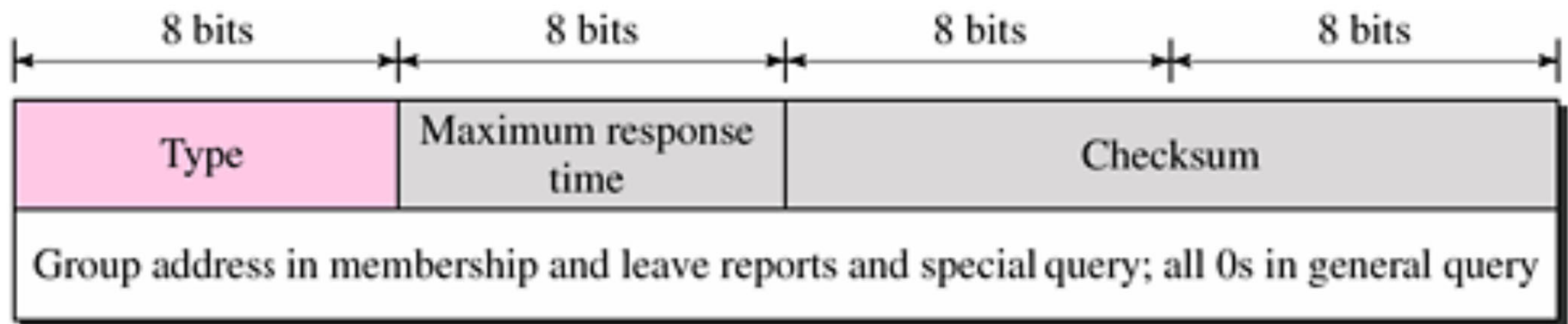
Internet Control Protocols

- In addition to IP, the Internet has several **control** protocols (if the transmission goes wrong or to find addresses) used in the network layer as follows:
 - ICMP
 - IGMP
 - ARP
 - RARP



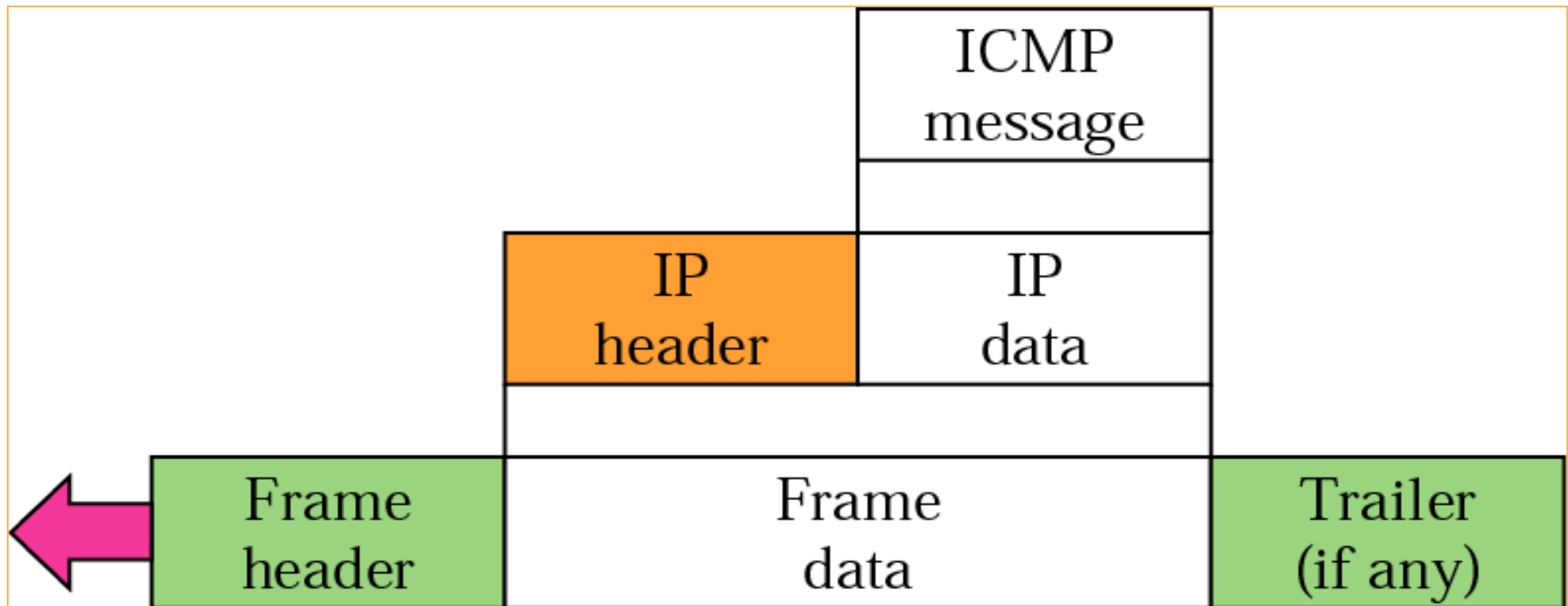
Internet Group Management Protocol (IGMP)

- It is used to manage multicasting (one-to-many communication)
- Query: check membership continuation and interest on multicast group
- Membership report: request to join multicast group
- Leave Report: request to leave multicast group
- Max response time: max time to answer a query
- Checksum: Error detection
- Group Address: group id (multicast address of the group)



Internet Control Message Protocol (ICMP)

- IPv4 has no error reporting or correcting mechanism
- ICMP is implemented by all TCP/IP hosts
- When something unexpected occurs, the event is reported



ICMP Message Types

ICMP Message Types

| Message | Purpose |
|------------------------------------|---|
| *Destination Unreachable | This tells the source host that there is a problem delivering a packet. |
| *Time Exceeded | The time it takes a packet to be delivered has become too long; the packet has been discarded. |
| Source Quench | The source is sending data faster than it can be forwarded; this message requests that the sender slow down. |
| *Redirect | The router sending this message has received some packet for which another router would have had a better route; the message tells the sender to use the better router. |
| *Echo | This is used by the ping command to verify connectivity. |
| Parameter Problem | This is used to identify a parameter that is incorrect. |
| Timestamp | This is used to measure roundtrip time to particular hosts. |
| Address Mask Request/Reply | This is used to inquire about and learn the correct subnet mask to be used. |
| Router Advertisement and Selection | This is used to allow hosts to dynamically learn the IP addresses of the routers attached to the subnet. |

Applications of ICMP

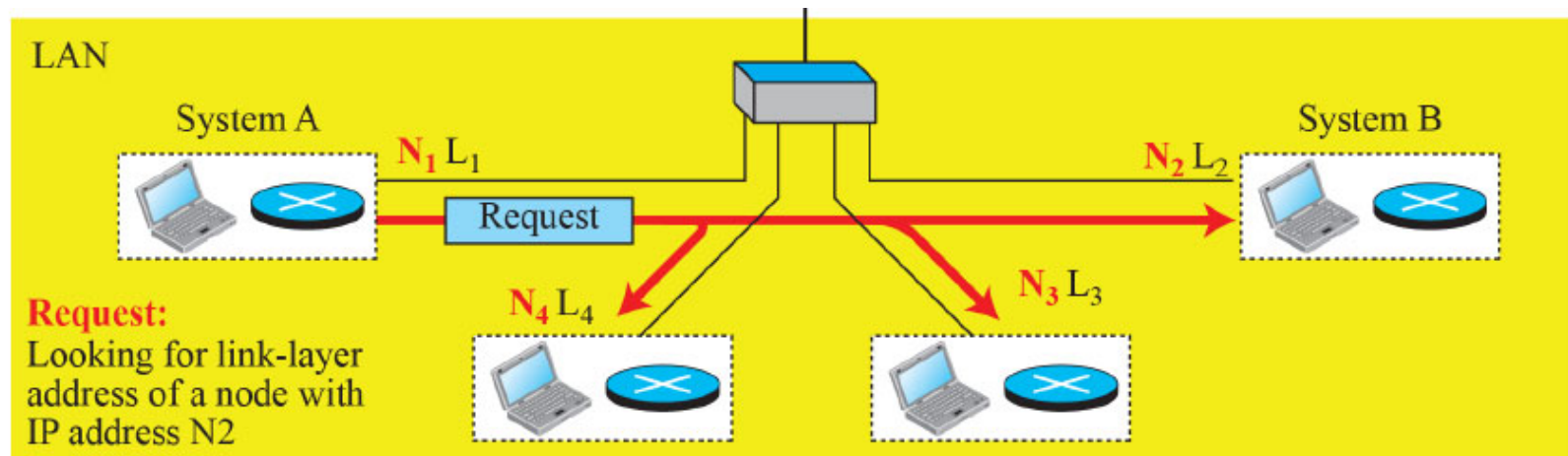
- “trace” command (variants: **tracert**, **traceroute**, etc.)
 - Is used to discover the **route from source to destination**
 - It uses “Time exceeded” message. By purposefully sending IP packets with TTL set to 1, an “ICMP Time Exceeded” message is returned by the first router in the route
 - It learns the IP address of the first router by receiving the “ICMP Time Exceeded” message from that router
 - It then sends IP packets with TTL set to 2 and learns the IP address of the second router.
 - Eventually, a set of packets is delivered to the destination and the addresses of all routers along the route is learned
- Exercises:
 - Try the command “tracert www.yahoo.com” in your home computer (internet enabled) to find the name of the routers along the route from your computer to www.yahoo.com

Note: In case tracert does not work in your computer, try other form “trace” or “traceroute”

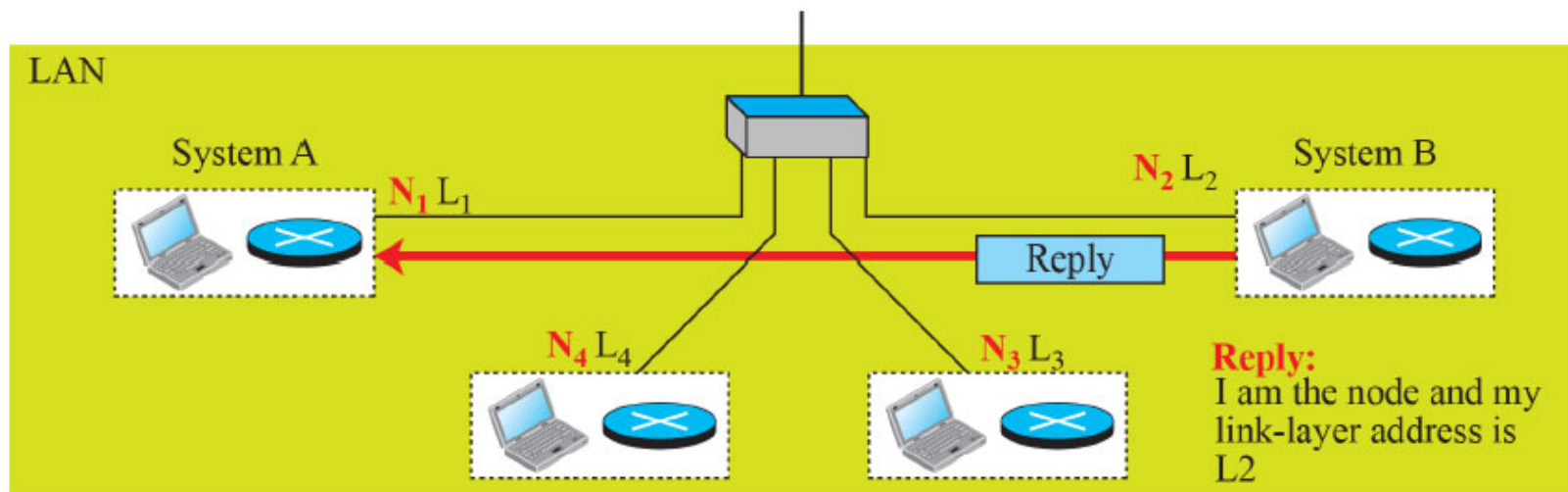
Applications of ICMP

- “ping” command
 - Is widely used to test connectivity between your host and a destination host
 - An “echo request” is sent with the ping command
 - The Echo Request means that the host to which it is addressed should reply to the packet
 - The destination host will reply with an “Echo Reply”
 - If the destination host cannot be reached, the packet will time out and a “destination unreachable” message is returned
- Exercise:
 - Try the command “ping www.yahoo.com” at your home computer and record the following data
 - The average time taken for your packet to reach www.yahoo.com and return (average round trip time)
 - The maximum/minimum round trip time

Recall ARP to find MAC addresses from IP addresses



a. ARP request is broadcast



b. ARP reply is unicast

IP addresses: how to get one?

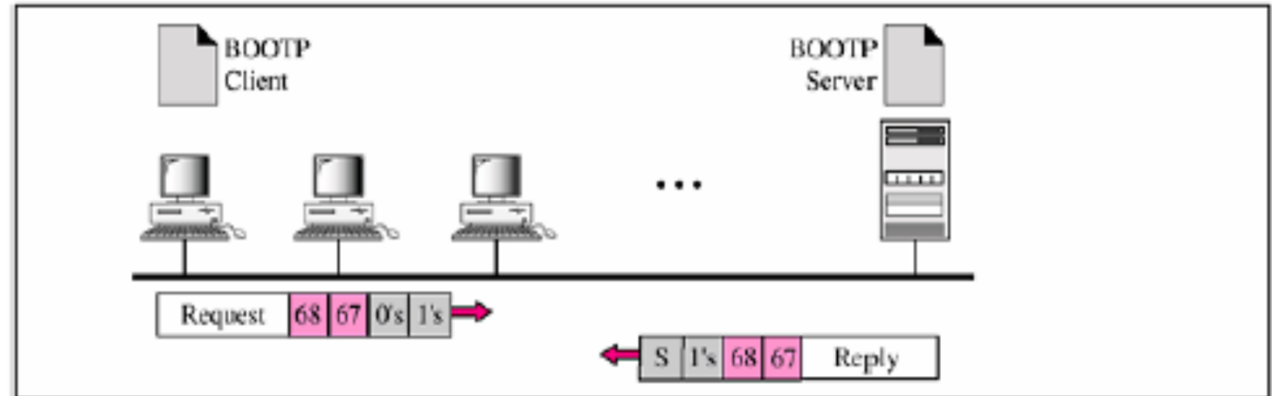
How does *the host get an* IP address?

- hard-coded by sysadmin in config file (e.g., /etc/rc.config in UNIX)
- The station can send (broadcast) its physical address and ask for a short time lease, becoming a link layer problem
 - Bootstrap
 - **DHCP**: **D**ynamic **H**ost **C**onfiguration **P**rotocol:
dynamically get address from the server

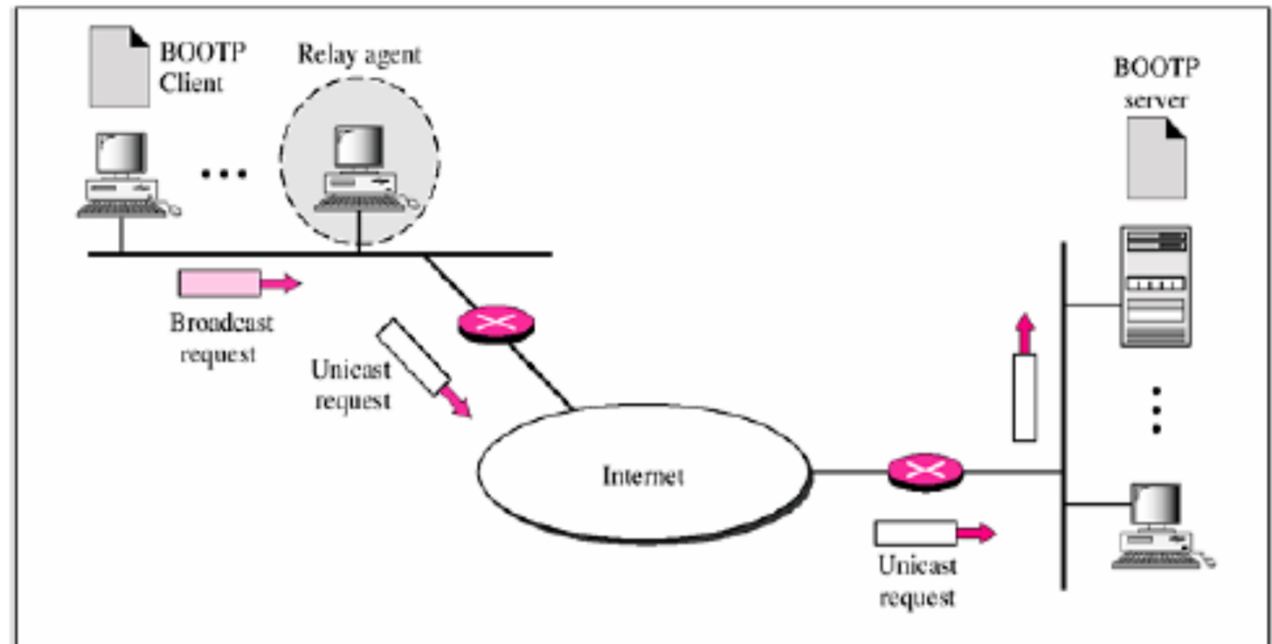
Bootstrap Protocol

A client/server protocol designed to provide physical address to logical address mapping (an application layer protocol)

Server has fixed MAC and IP table mapping



a. Client and server on the same network



b. Client and server on different networks

DHCP: Dynamic Host Configuration Protocol

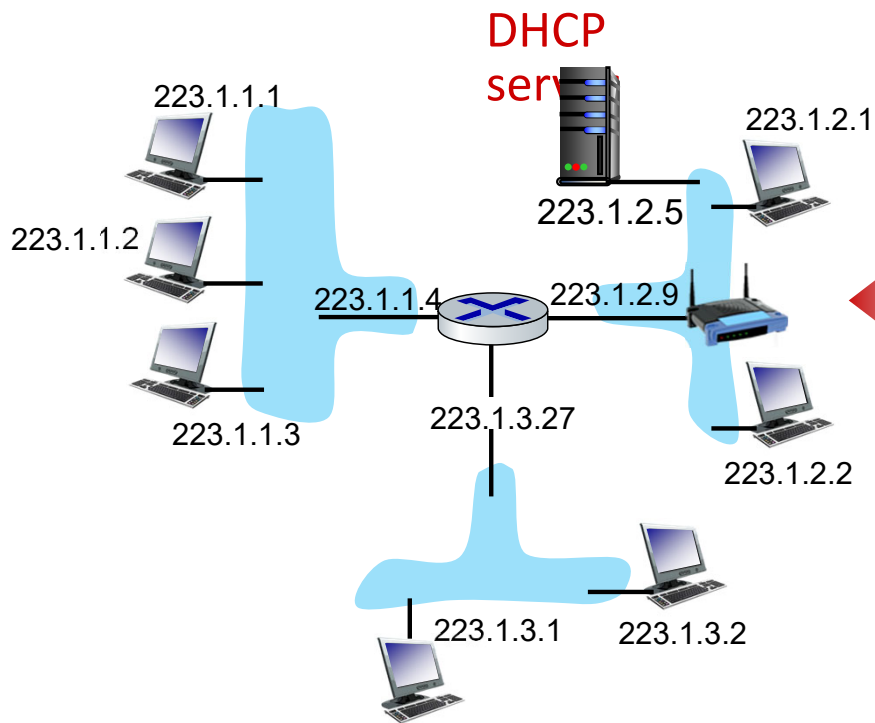
goal: host *dynamically* obtains IP address from IP **pool** at the network server when it “joins” network

- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected/on)
- support for mobile users who join/leave network

DHCP overview:

- host broadcasts **DHCP discover** msg [optional]
- DHCP server responds with **DHCP offer** msg [optional]
- host requests IP address: **DHCP request** msg
- DHCP server sends address: **DHCP ack** msg

DHCP client-server scenario

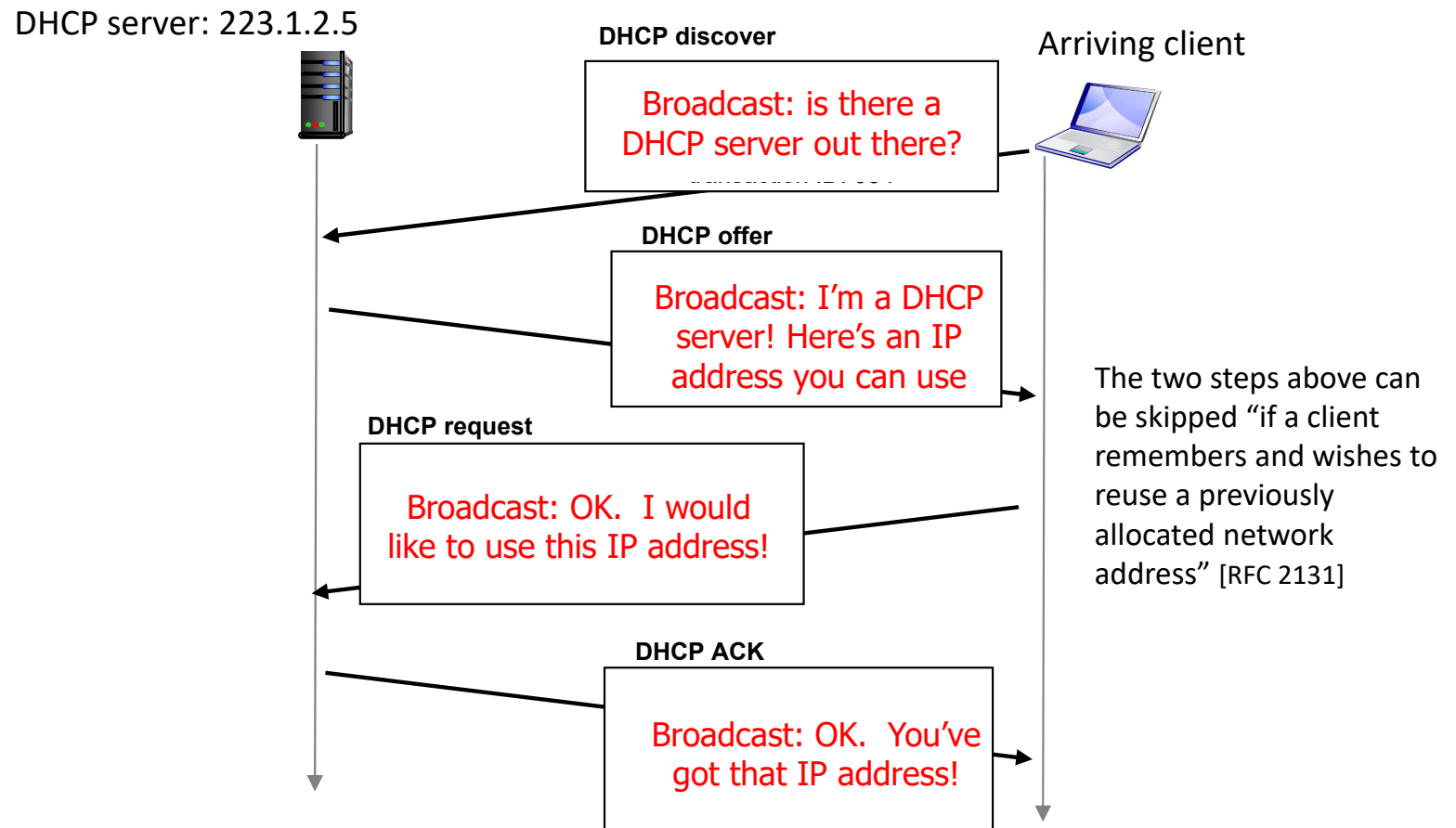


Typically, DHCP server will be co-located in router, serving all subnets to which router is attached



arriving **DHCP client** needs address in this network

DHCP client-server scenario

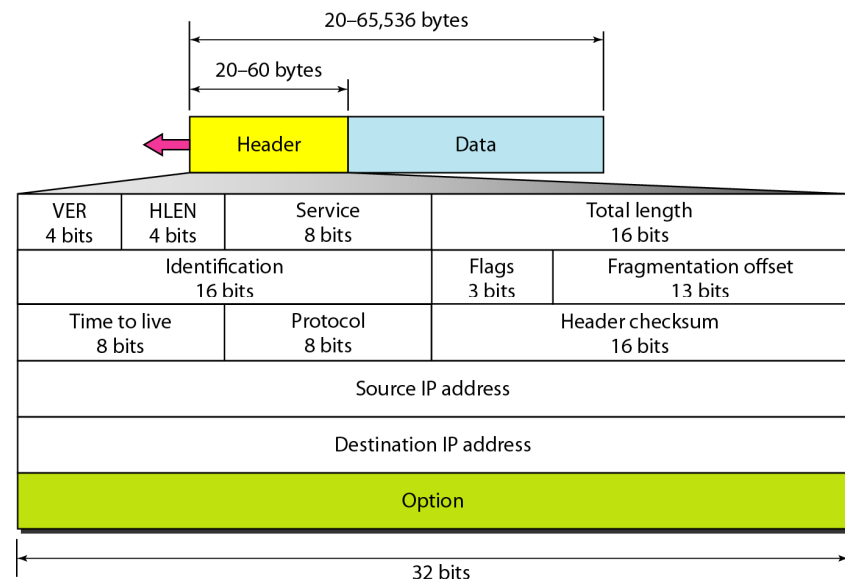


IPv6

- **IPv6** is created to address the fear that address space of IPv4 may run out soon
- **Increasing the size of the address field** dictates a change in IP header format – a new version of Internet Protocol
- In addition to the need to accommodate scalable routing and addressing, other desirable features are:
 - **Support for real-time audio/video service (min delay and resource reservation)**
 - **Security support (encryption and authentication)**
 - **Auto configuration**
 - **Enhanced routing functionality**

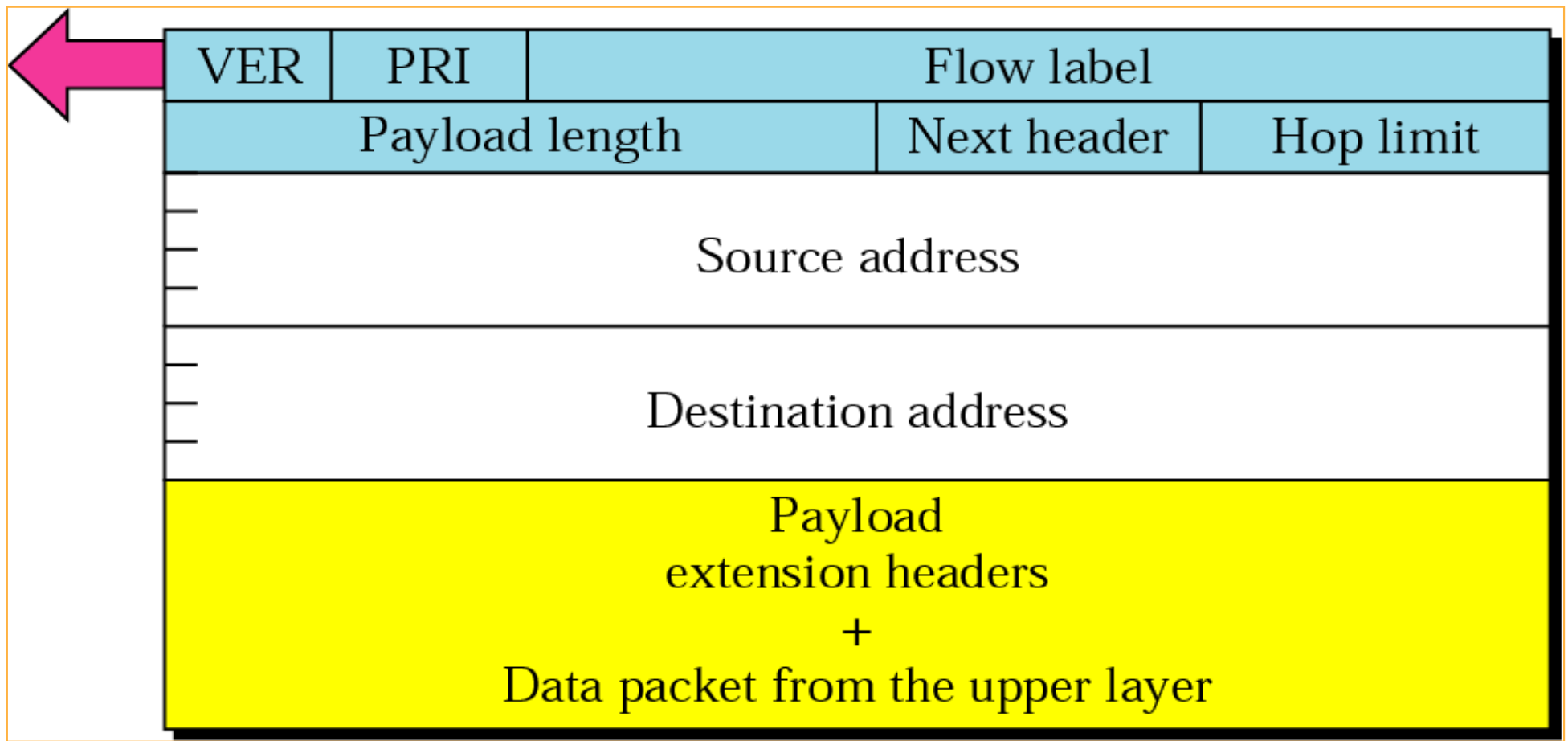
Advantages of IPv6

- **Larger address space** – An IPv6 address is 128 bits long
- **Better header format** – options separated from base header speeds up routing
- **Support for resource allocation** – flow labels used for special handling of packets
- **Support for more security** – encryption and authentication option
- **Allowance for extension**
- **New options**



IPv4 Datagram Format

IPv6 Header Format

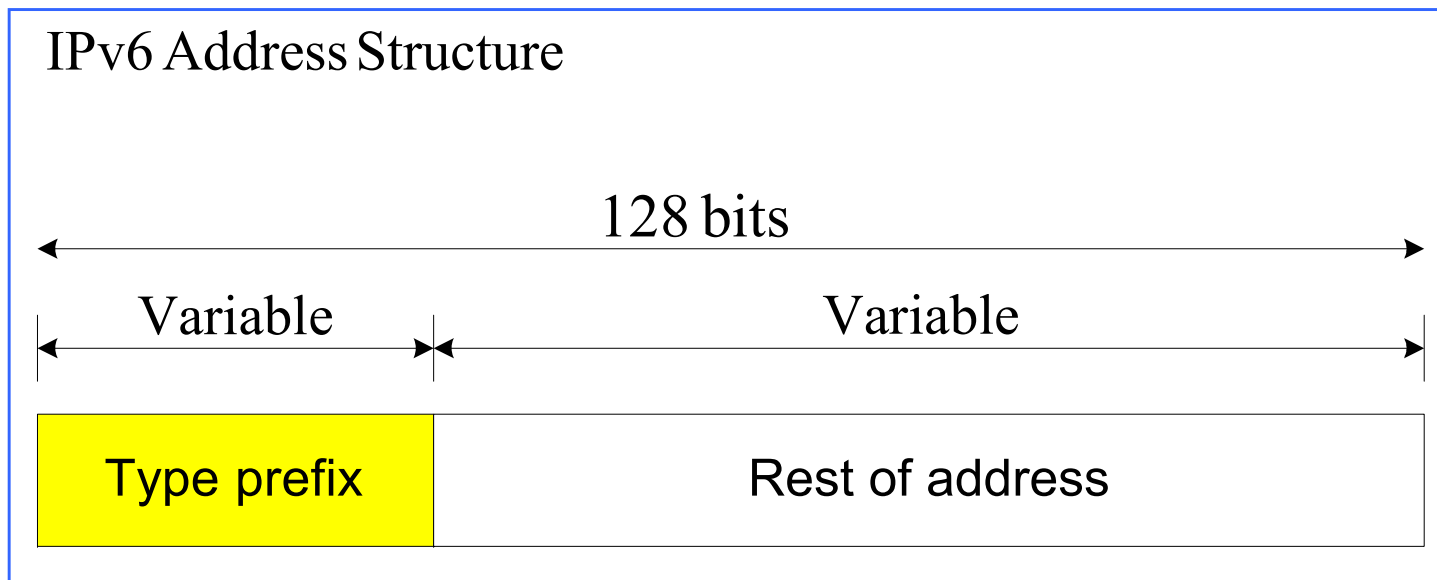


IPv6 Header Format

- Mandatory **Base Header (40 bytes)** has **8 fields**
 - **Version:** indicate the version number of the IP protocol
 - **Priority:** priority of each packets with respect to other packets
 - **Flow label:** can be used to provide special handling for a particular flow of data (e.g., travelling the same path, using same resources, etc.)
 - **PayloadLen:** indicates the length of the packet, excluding the IPv6 header
 - **NextHeader:** replaces optional field and protocol field of IPv4
 - Indicate whether there are more headers (options) follow
 - Indicate higher layer protocols running over IP
 - **HopLimit:** the same as TTL in IPv4

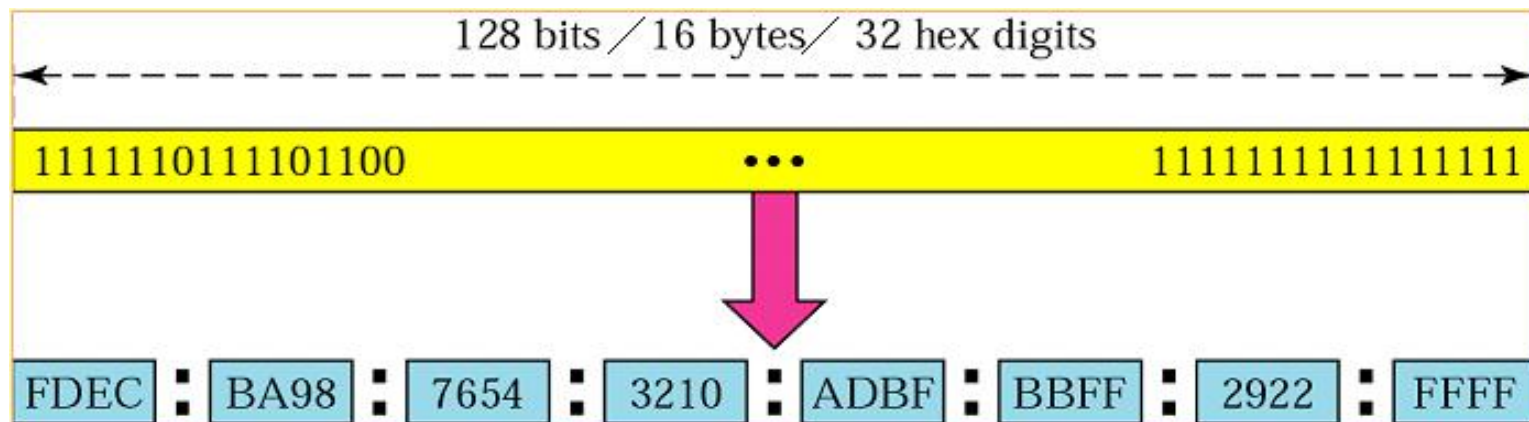
IPv6 Address Types

- The address space is divided into **two parts** with the first part called the *type prefix*
- **Address Types**
 - Unicast, Multicast, Anycast, Reserved, Local



Representation of IPv6 address

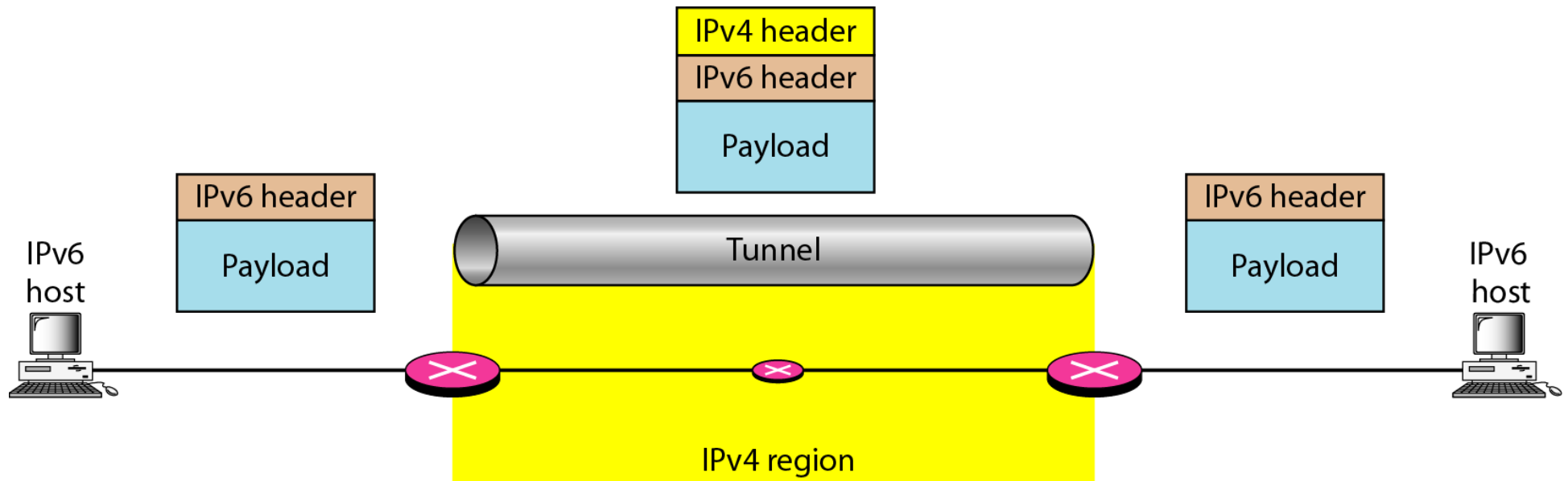
- IPv6 specifies **hexadecimal colon notation** consisting of **32 hex digits**



Transition from IPv4 to IPv6

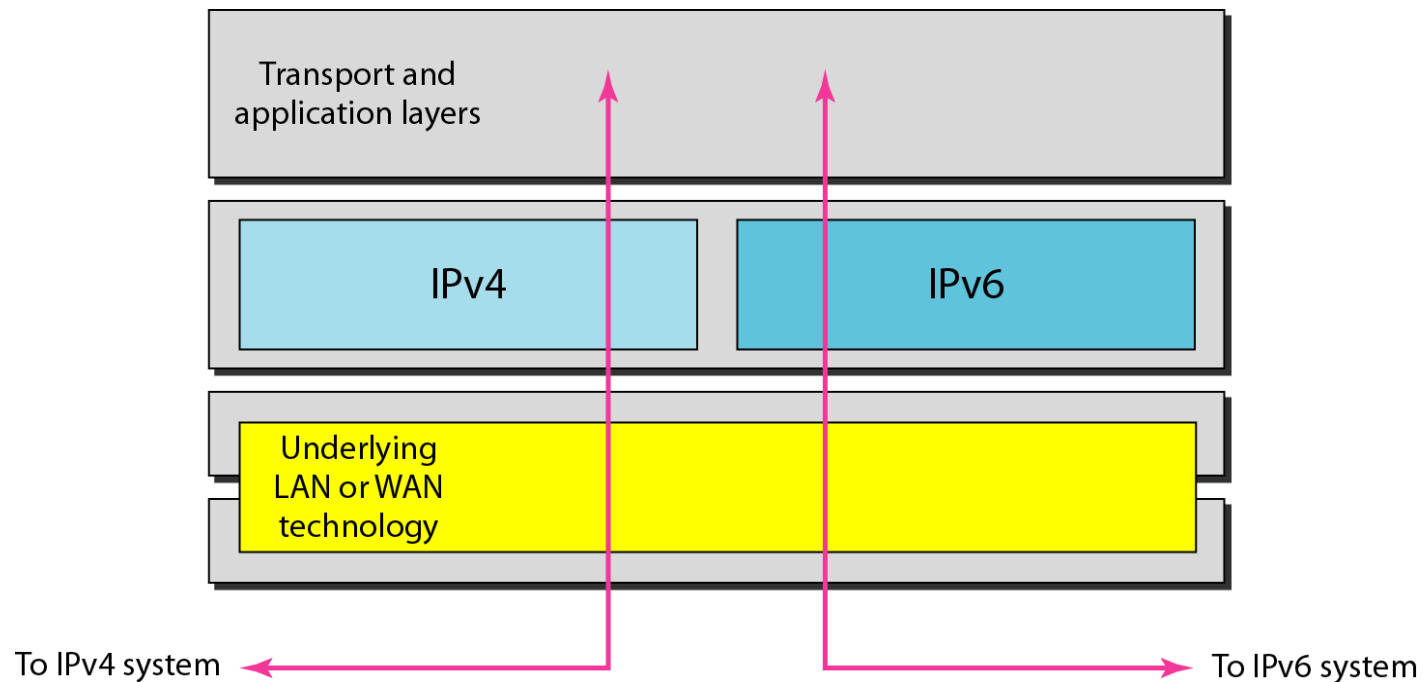
- Not all routers can be upgraded simultaneously
 - no “flag days”
 - How will the network operate with mixed IPv4 and IPv6 routers?
- Two proposed approaches:
 - *Dual Stack*: some routers with dual stack (v6, v4) can “translate” between formats
 - *Tunneling*: IPv6 carried as payload in IPv4 datagram among IPv4 routers

Tunneling



Dual Stack

- IPv6 nodes have a complete IPv4 implementation
 - Use IPv6 when interact with IPv6 nodes
 - Use IPv4 when interact with IPv4 nodes
 - IPv6/IPv4 nodes must have both IPv6 and IPv4 capabilities



Recommended Reading

- Behrouz A. Forouzan, Data Communications and Networking with TCP/IP Protocol Suite, 6th ed., 2022, Chapters 7
- J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach, 8th ed., 2022, Chapter 4