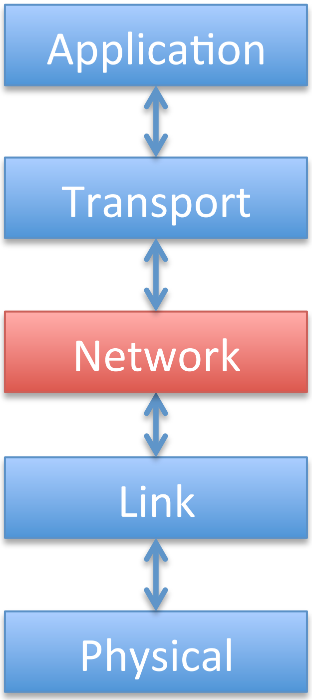
Computer Networks and Communication

Lecture 8

Network Layer

Network Layer

- **Task**: Move packets from a sending host to a receiving host
- Getting to the destination may involve many hops at intermediate routers
- The network layer must determine the **network wide** end-to-end path to send packets from sender to receiver
  - This process is called **routing**
  - Algorithms to determine the paths of packets are called **routing algorithms**



Internet Network-Layer Components

- The Internet's network layer consists of three components
  - **IP Protocol**: Responsible for host addressing and datagram format and forwarding
  - **Routing Protocols**: Determines the path a datagram follows from source to destination
    - Example: RIP, OSPF, BGP
  - **ICMP Protocol**: Error reporting and provides certain network-layer information

Internet Protocol

- The Internet is a network of networks
- Different **subnetworks** or **Autonomous Systems (ASes)** in the Internet might be implemented on different sets of hardware, software and protocols
- **Problem**: How can two nodes in different ASes communicate with each other?
  - IP is designed to solve this very problem

Internet Protocol (2)

- Main protocol of the Internet
- Defined in RFC 791
- We are moving from IPv4 to IPv6
- IP features
  - Best-effort service to transport datagrams from sources to destinations
  - The datagram transportation works within the same network and across networks
  - Host addressing conventions

IPv4 Protocol Header

<table>
<tr><th colspan="3">← 16 bit →</th><th colspan="2">← 16 bit →</th></tr>
<tr><td>Version</td><td>Header length</td><td>Type of Service</td><td colspan="2">Total datagram length (bytes)</td></tr>
<tr><td colspan="3">16-bit Identifier</td><td>Flags</td><td>13-bit Fragmentation Offset</td></tr>
<tr><td colspan="2">Time-to-Live</td><td>Upper-layer Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="5">Source IP Address (32 bit)</td></tr>
<tr><td colspan="5">Destination IP Address (32 bit)</td></tr>
<tr><td colspan="5">Options (if any)</td></tr>
<tr><td colspan="5">Payload (variable length)</td></tr>
</table>

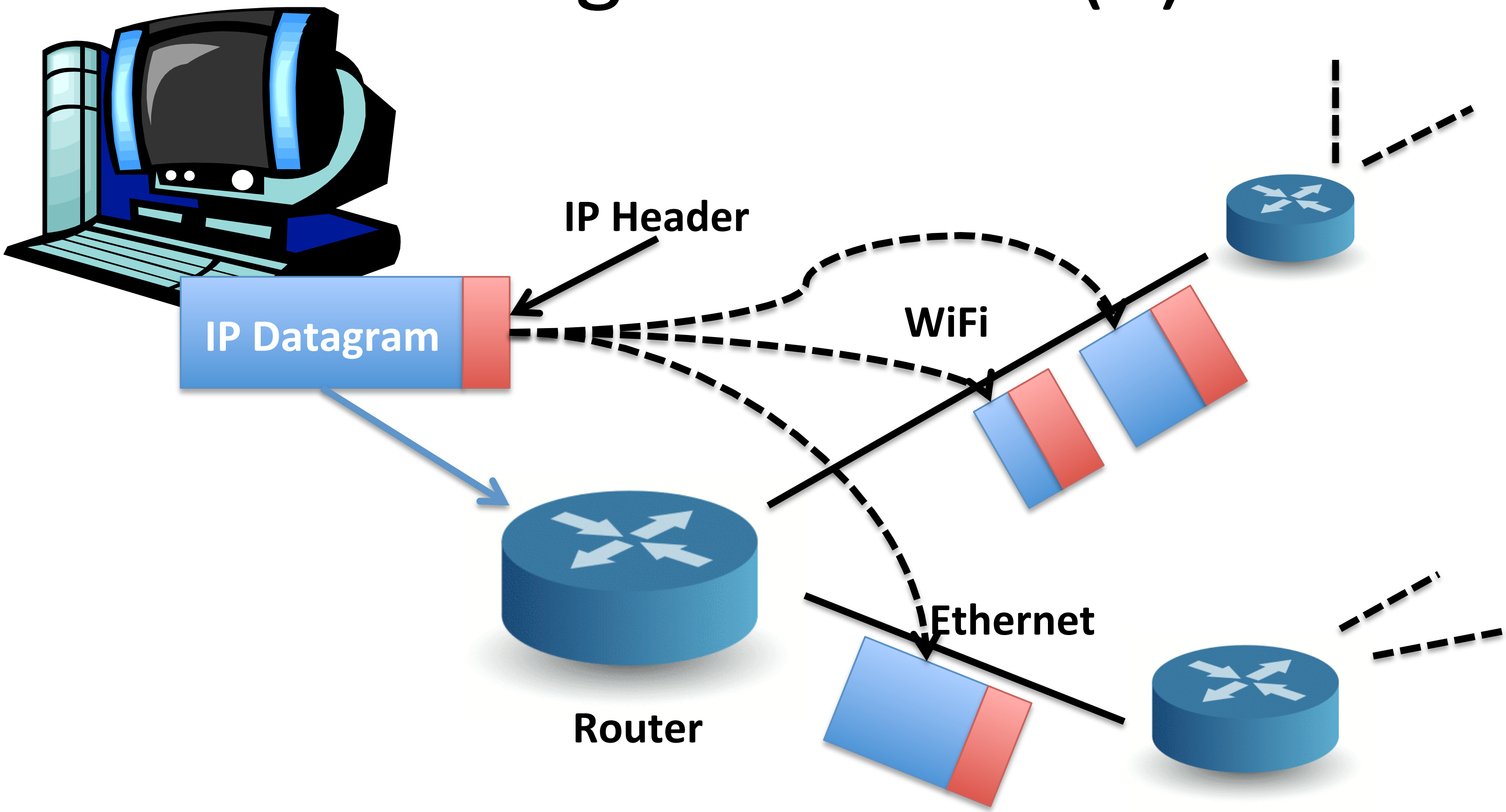
IP Fragmentation

- Network layer has to communicate with the link layer which controls the physical medium
  - Maximum size of a packet (or **frame**) in link layer is specified as **Maximum Transfer Unit (MTU)**
  - Different link types => Different MTU
- **Problem**: You have datagrams with the same size, how can you be sure that they will fit in different frames of different link types with different MTU?

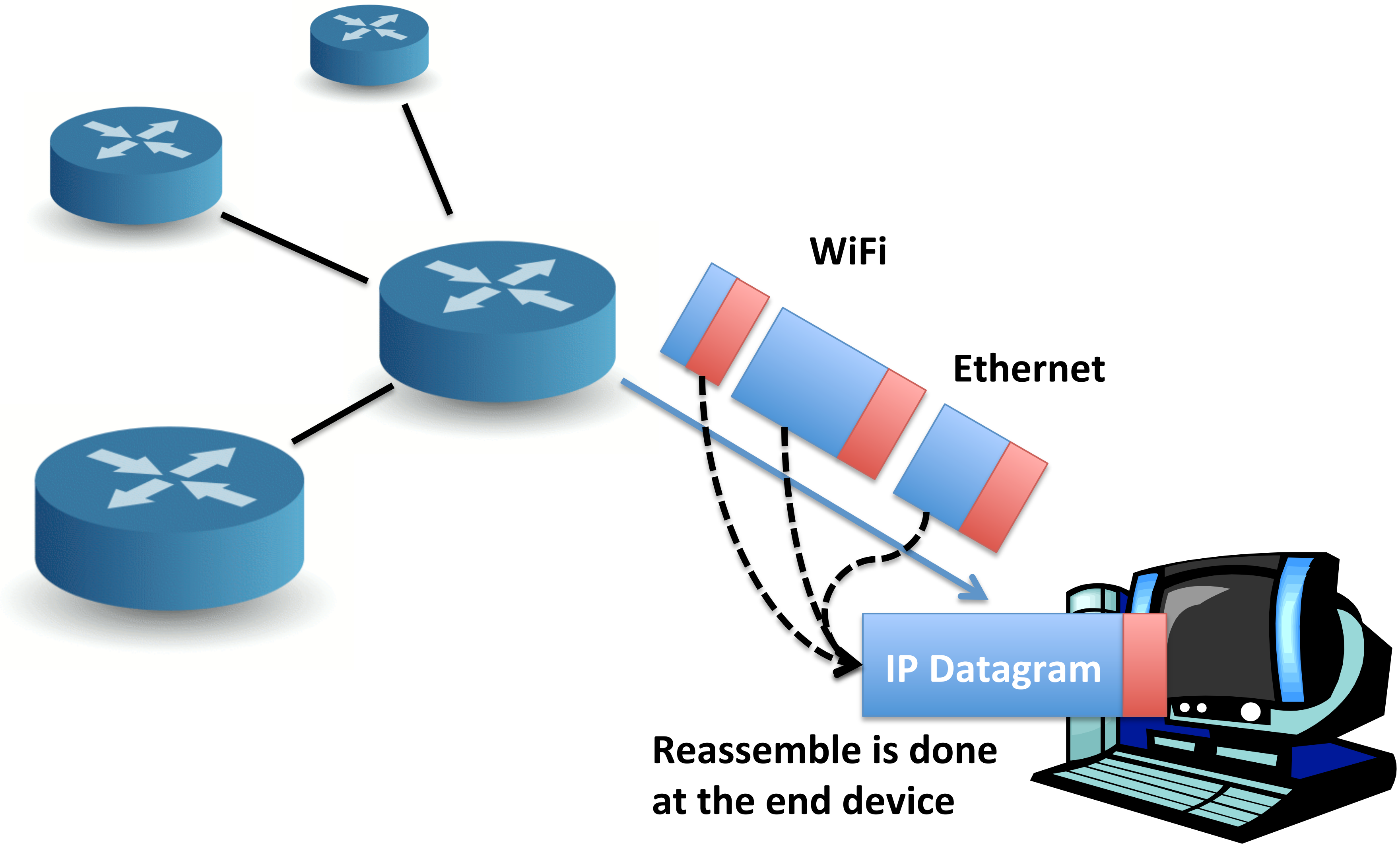
IP Fragmentation (2)

- **Solution**: You divide each datagram smaller datagrams such that the divided datagrams fit the link-layer frames. This process is called "**fragmentation**"
- Fragmentation
  - Large datagram is divided into several datagram
  - A sub-datagram is called a **fragment**
  - Fragments are resembled back into original datagram at the final destination
  - A number of IP header fields are used for the fragmentation

IP Fragmentation (3)



IP Fragmentation (4)

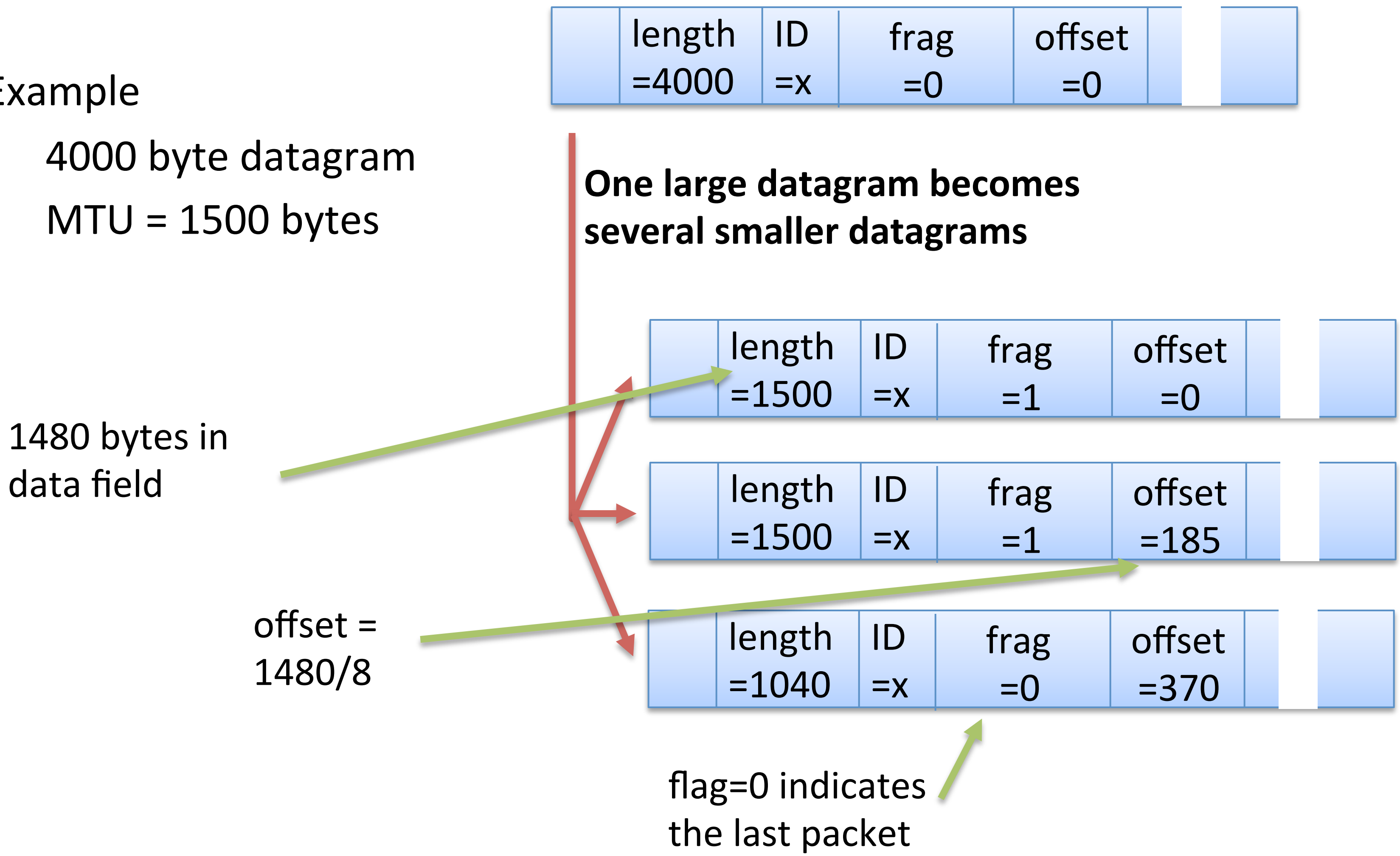


Reassemble is done at the end device

IP Fragmentation Example

Example

- 4000 byte datagram
- MTU = 1500 bytes



Network Interface

- A **network interface** is a component which a network device used to connect to a network
  - WiFi
  - Ethernet
  - Bluetooth
- Of course, a device can have more than one interface of different or the same type
- Each router has at least two

IP Address and Network Interface

- IP requires that each interface in a network must have a unique IP address
- This means, a router have multiple addresses associated to different interfaces

IP Address

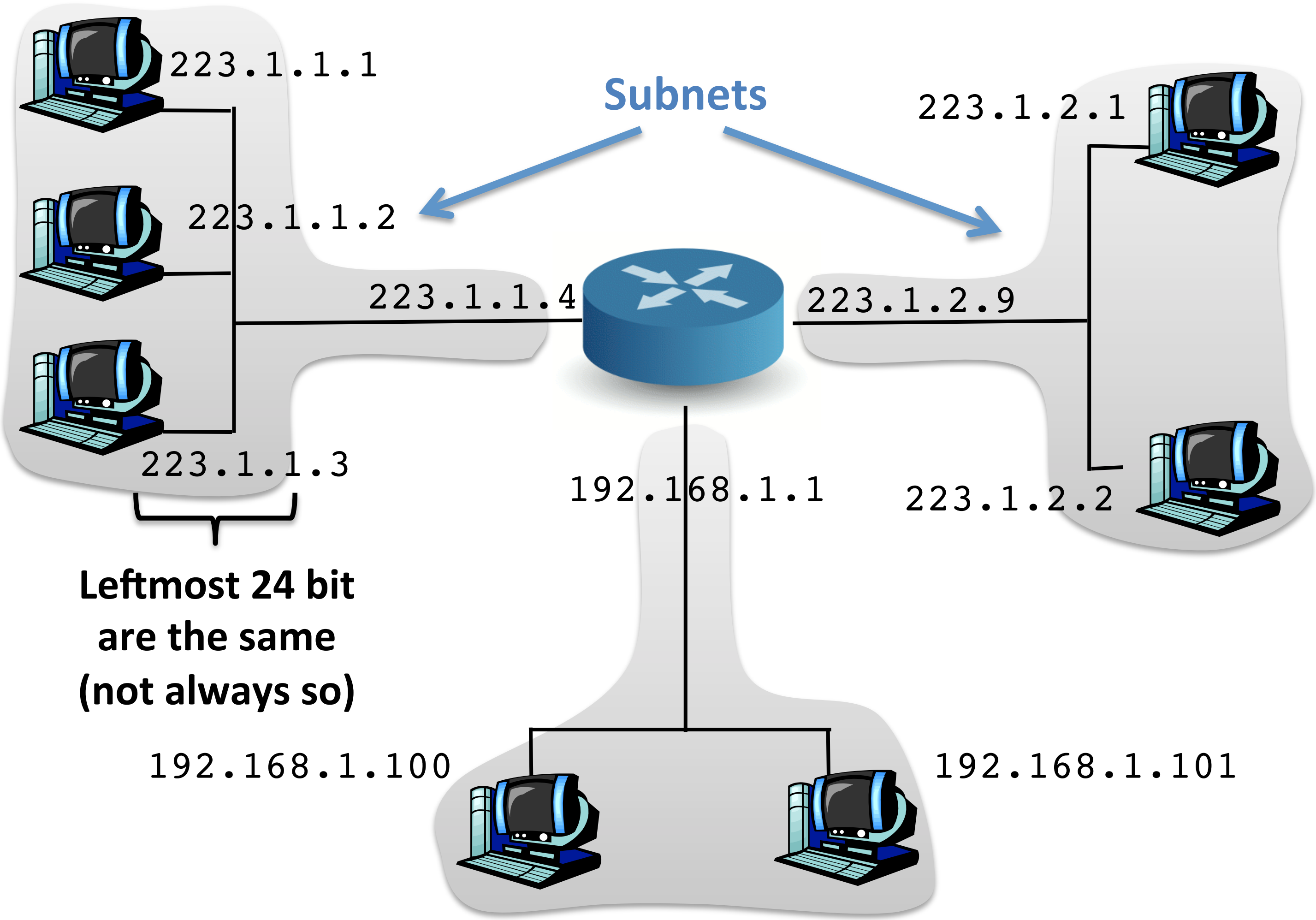
- An **IPv4** address is a 32 binary number

  `11000001 00100000 11011000 00001001`

- Can refer up to 2^{32} = ~4 billion addresses
- An IP address is usually written using **dotted-decimal notation**
  - Each byte (8 bit) of the address is written in decimal form
  - Separated by a period (dot)

`193.32.216.9`

IP Address (2)


223.1.1.1
Subnets
223.1.2.1
223.1.1.2
223.1.1.4
223.1.2.9
223.1.1.3
192.168.1.1
223.1.2.2
Leftmost 24 bit are the same (not always so)
192.168.1.100
192.168.1.101

IP Subnet

- A **subnet** is a sub-division of an IP network
  - All hosts in the same subnet share the same *X* digits of IP addresses
  - For example first 24 bits of the following addresses are the same

    `223.1.1.1, 223.1.1.2, 223.1.1.3`

  - We can refer to this subnet as

    `223.1.1.0`

CIDR

- Typical address assignment strategy is called **Classless Inter-Domain Routing (CIDR)**
- It divides an address into two parts
  - **Network number**: The first *X* bits
  - **Host number**: The rest
  - An IP address can be written as

    `223.1.1.1/24`

    where 24 denotes that the first 24 bits is the network-number portion
  - A subnet, in turn, is denoted by

    `223.1.1.0/24`

CIDR (2)

Example:

223.1.1.0/24

11011111 00000001 00000001 00000000

← network number (subnet) part → ← host part →

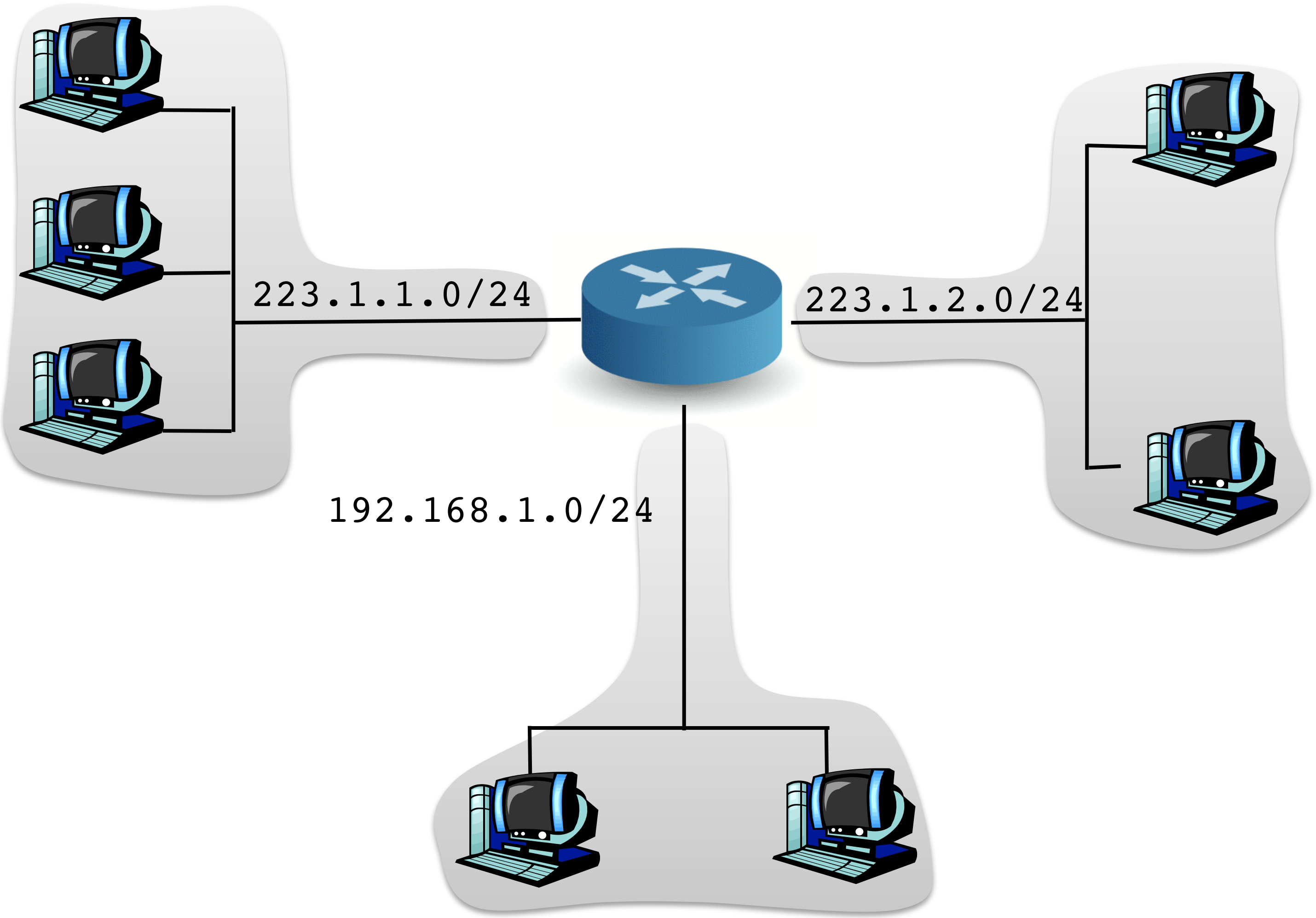
Micro exercise:

200.23.16.0/23

11001000 00010111 00010000 00000000

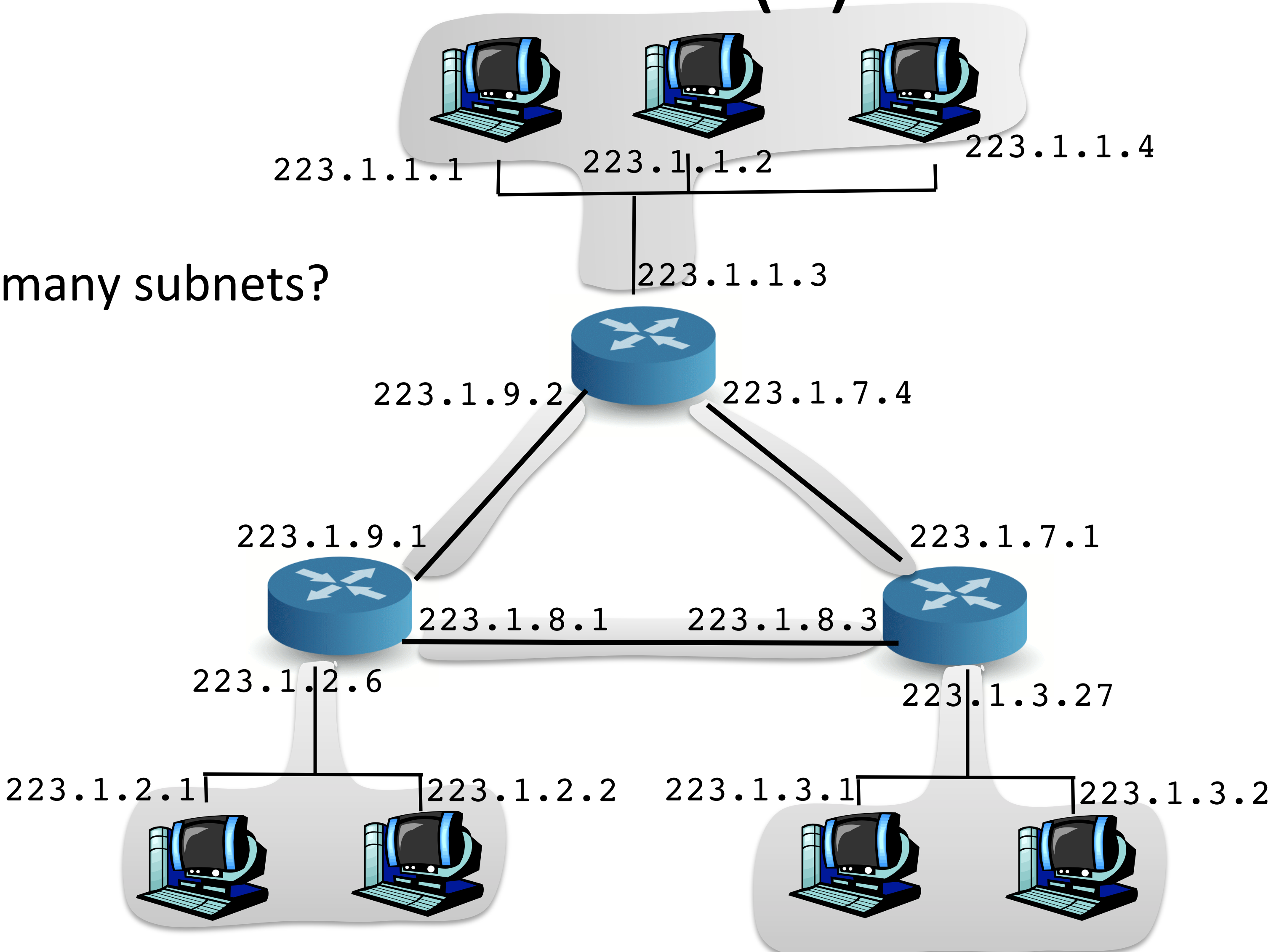
← network number (subnet) part → ← host part →

IP Subnet (2)



IP Subnet (3)

How many subnets?

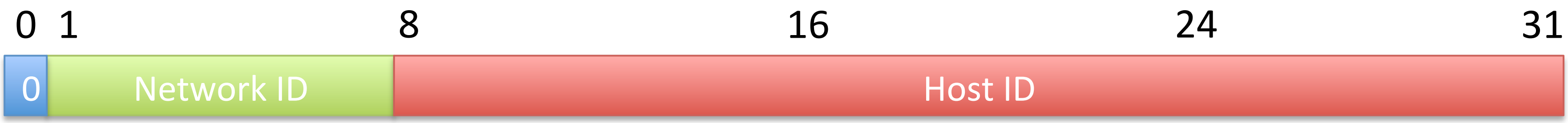


Classful Network (Obsolete)

- It is an addressing strategy used in 1980s before CIDR was introduced
- IP Addresses are divided into 5 **classes**, A – E
- Each class has different (and fixed) size of network number portion and host number portion
- **It is obsolete**: It does not scale to the size of the Internet

Classful Network (2)

Class A: For networks which have up to 16 million hosts (0.0.0.0 – 127.255.255.255)



Class B: For networks which have up to 65,536 hosts (128.0.0.0 – 191.255.255.255)

Class C: For networks which have up to 256 hosts (192.0.0.0 – 223.255.255.255)

Class D: For multicast networks (224.0.0.0 – 239.255.255.255)

Class E: Reserved for future uses (240.0.0.0 – 255.255.255.255)

Netmask

- **Mask** or **netmask** is a form of denoting the separation between network and host parts in the IP address
- To specify that the network part is first 24 bit, it is written as:

```
11111111 11111111 11111111 00000000
```

- Or

```
255.255.255.0
```

- If the network part is the first 18 bit, , it is written as:

```
11111111 11111111 11000000 00000000
```

- Or

```
255.255.192.0
```

- This notation is valid only in IPv4, but not IPv6

Private Addresses

- **Problem**: If all hosts in the Internet have IP addresses, some might use the same addresses as our private machines
  - Example: `www.google.com` IP is `74.125.135.104`
    If an host in my subnet also has the address `74.125.135.104`, how could my router know which host I want to connect to?

- **Solution**: Specific address spaces that are assigned only to private addresses or "**private Internets**"

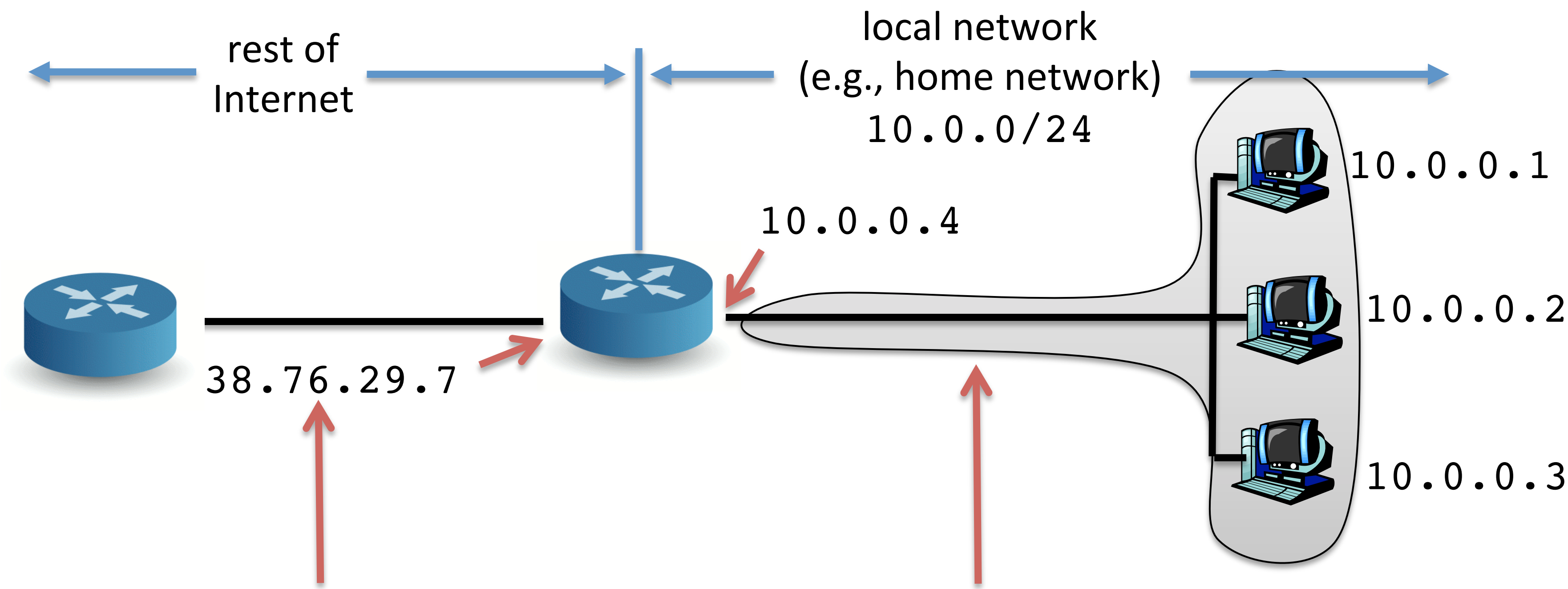
Private Addresses (2)

- IANA has reserved the following ranges for private Internets:
  - `10.0.0.0 – 10.255.255.255`
    10/8 prefix or 24-bit block
  - `172.16.0.0 – 172.31.255.255`
    172.16/12 prefix or 20-bit block
  - `192.168.0.0 – 192.168.255.255`
    192.168/16 prefix or 16-bit block
- The address ranges are different in sizes.
- Their uses can be chosen based on the size of the networks or enterprises.
- This is the reason why your routers usually begin with `192.168.1` or `192.168.0`

Network Address Translation (NAT)

- What we learned: Each host has to have an IP address
- Problems:
  - Does your ISP has to assign IP to each of your network devices (e.g. Desktop, MacBook, and BB)?
  - Is there enough IP addresses for every device on earth?
- Solution: **Network Address Translation (NAT)**
  - A mapping between IP addresses and ports across two interfaces of a router

Network Address Translation (2)



All datagrams ***leaving*** local network have same single source NAT IP address: `138.76.29.7`, different source port numbers

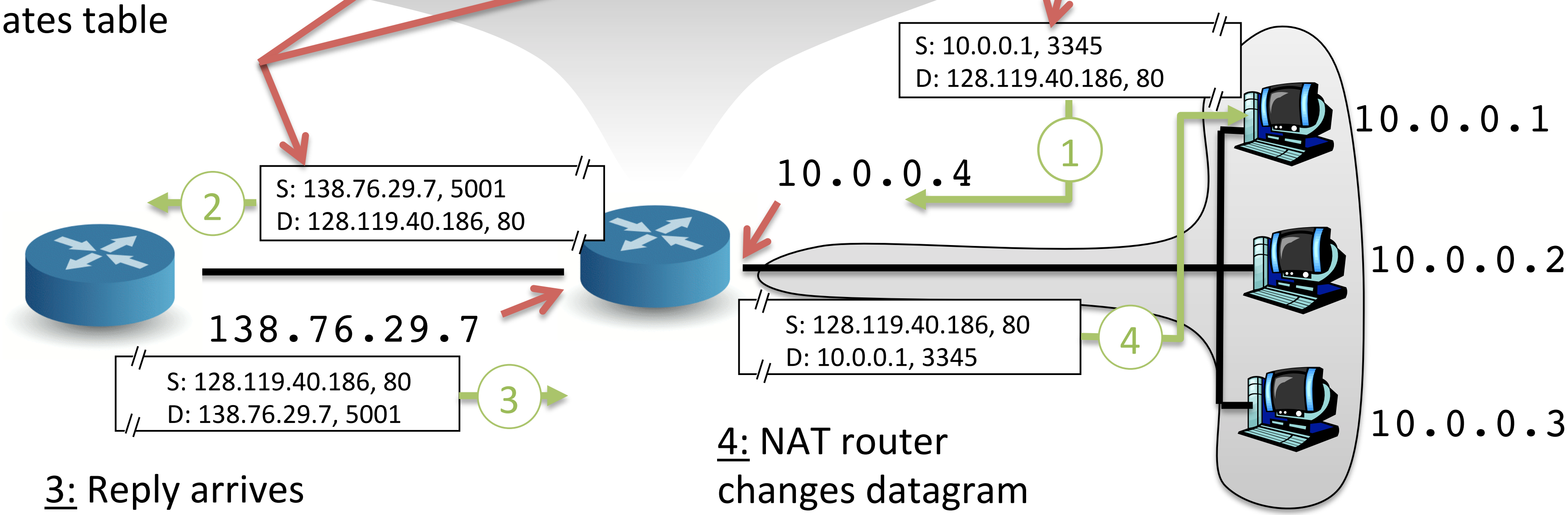
Datagrams with source or destination in this network have `10.0.0/24` address for source, destination (as usual)

Network Address Translation (3)

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

NAT translation table	
WAN side addr	LAN side addr
138.76.29.7,5001	10.0.0.1,3345
……	……

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80



3: Reply arrives dstIP: 138.76.29.7, 5001

4: NAT router changes datagram dstIP from 138.76.29.7, 5001 to 10.0.0.1, 3345

Problems with NAT

- A host behind NAT can establish connection to any outside host
- But it cannot act as a server waiting for any connection
  - Because the outside client would not know which WAN-side port to connect to
  - If it wants to connect to specific port, the outside client cannot know if the server is waiting inside the NAT

Connection Reversal

- NAT Problem can be circumvented using **connection reversal** technique
- Connection reversal works as follows:
  - A wants to connect to B and B is behind NAT
  - A is **not** behind NAT
  - B has to connect to intermediate server, C, first
  - Then, A connects to C, asking C to ask B to connect back to A
- However, if A is also behind NAT, we would end up with the same problem
  - Solution: **Relaying technique (Homework)**
  - Solution: **UPnP (Homework)**

DHCP

- **Dynamic Host Configuration Protocol** (**DHCP**)
- Automatically configures the hosts which are connected to a network
- Provides network-configuration information to hosts, e.g.
  - IP address
  - DNS server address
  - Subnet mask
  - Router address
- DHCP is a client-server protocol
  - The hosts who wants to connect to a network must send DHCP configuration request to a DHCP server

DHCP Configuration Process

- A host (client) who wants to connect to a network finds a DHCP server
  - Sending **DHCP discover message** to the **broadcast address** (**255.255.255.255**) using **UDP at port 67**
  - Using broadcast address, **all** hosts in the network receive the message

- DHCP server(s) in the network responses to the message by sending **DHCP offer message** back to the broadcast address, now to **port 68**.
  (**Why broadcast address?**)
  - The offer message contains the network configuration information

- DHCP client sends **DHCP request message** to the responded servers confirming the configuration info

- DHCP server sends **DHCP ACK message** confirming the request

IPv6

- Upcoming version of IP protocol
- Motivation
  - IPv4 addresses are running out
    - IPv4 address space: **2^{32} = ~4 billion** addresses
    - IPv6 address space: **2^{128} =3.4 x 10^{38}** (6 x 10^{23} addresses per m^2 of Earth surface)
  - Improve packet routing speed and efficiency
    - The header has only 8 instead of 13 fields
    - The header has fixed length (40 bytes)
    - Eliminate the option field
    - Introduce **Flow Label** and **Traffic Class** fields to accommodate **Quality-of-Service (QoS)**
    - No fragmentation. No checksum

IPv6 Protocol Header