CHAPTER SIX

Internet Protocol (IP) and IP Addressing

Internet Protocol (IP)

- The **Internet Protocol (IP)** is a protocol used for **communicating data** across a **packet-switched** internetwork using the **Internet Protocol Suite**, also referred to as TCP/IP.
- IP is the **primary protocol** in the **Internet Layer** of the **Internet Protocol Suite** and has the task of **delivering** distinguished protocol datagrams (packets) from the **source host** to the **destination host** solely **based on their addresses.**

Internet Protocol (IP) continued

- For this purpose the **Internet Protocol** defines **addressing methods** and **structures** for **datagram encapsulation**.
- The first **major version** of **addressing structure**, now referred to as **Internet Protocol Version 4** (IPv4) is still the **dominant protocol** of the **Internet**, although the successor, **Internet Protocol Version 6** (IPv6), is being deployed actively **worldwide (128 bits).**
- **Communication** at the **network layer** is **host-to-host (computer-to-computer)**; a computer somewhere in the world needs to **communicate** with another **computer somewhere** else in the **world**.

Internet Protocol (IP) continued

- Usually, **computers communicate** through the **Internet**.
- The **packet transmitted** by the **sending computer** may pass through several **LANs** or **WANs** before reaching the **destination computer**.
- For this **level** of **communication**, we need a **global addressing scheme**; we use the term **IP address** to mean a **logical address** in the **network layer** of the **TCP/IP protocol suite**.

IPv4 Addresses

- An **IPv4** address is a **32-bit** address that **uniquely** and **universally defines** the **connection** of a **device** (**for example**, a **computer** or a **router**) to the **Internet**.
- **IPv4 addresses** are **unique** and **universal**.

✓ They are **unique** in the **sense** that **each address** defines **one,** and **only one**, **connection** to the **Internet**.

- **Two devices** on the **Internet** **can never** have the **same address at the same time**.

Address Space

- A **protocol** such as IPv4 that **defines addresses** has an **address space**.

✓ An **address space** is the **total number** of **addresses** used by the **protocol**.

- If a **protocol** uses **N bits** to **define** an **address**, the **address space** is 2^N because **each bit** can have **two different values** (**0 or 1**) and **N bits** can have $\mathbf{2^N}$ **values**.

Address Space continued

- IPv4 uses **32-bit addresses**, which means that the **address space** is 2^{32} or **4,294,967,296** (more than **4 billion**).

✓ This means that, **theoretically**, if there were **no restrictions**, more **than 4 billion devices** could be **connected** to the **Internet**.

- We will see shortly that the **actual number** is **much less** because of the **restrictions** imposed on the **addresses**.

IP Address Notations

- There are **two prevalent notations** to show an **1Pv4 address**: **binary notation** and **dotted-decimal notation**.

1. Binary Notation

- In **binary notation**, the IPv4 address is displayed as **32 bits**.
- Each **octet** is often referred to as a **byte**.
- So it is common to hear an **IPv4 address** referred to as a **32-bit address** or a **4-byte address**.

IP Address Notations continued

✓ The following is an **example** of an **IPv4 address** in **binary notation:**

01110101 10010101 00011101 00000010

2. Dotted-Decimal Notation

- To make the **IPv4 address more compact** and **easier to read**,

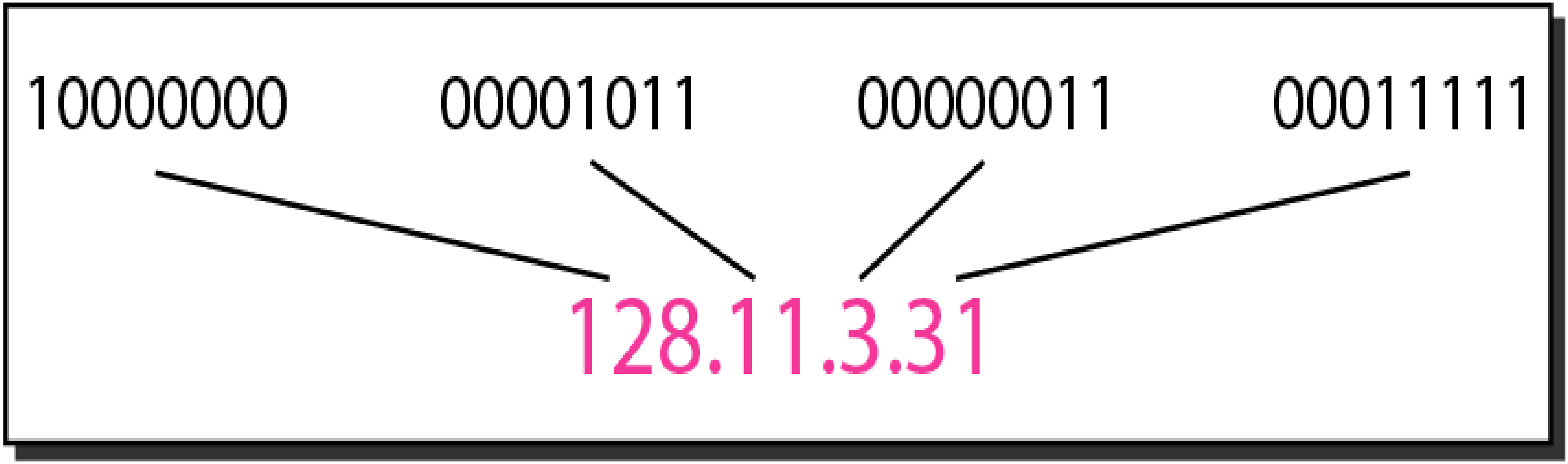
Internet addresses are usually written in **decimal** form with a **decimal point (dot)** separating the **bytes**.

✓ The following is the **dotted-decimal notation** of the above address: **117.149.29.2**

IP Address Notations continued

- **Example**:

- **Dotted-decimal notation** and **binary notation** for an **IPv4 address**



Example 1

- Change the following IPv4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

- **Solution**

- We replace each group of 8 bits with its equivalent decimal number and add dots for separation.

a. 129.11.11.239

b. 193.131.27.255

Example 2

- Change the following IPv4 addresses from dotted-decimal notation to binary notation.

a. 111.56.45.78

b. 221.34.7.82

- Solution

- We replace each decimal number with its binary equivalent.

a. 01101111 00111000 00101101 01001110

b. 11011101 00100010 00000111 01010010

Example 3

- Find the error, if any, in the following IPv4 addresses.

  a. 111.56.045.78

  b. 221.34.7.8.20

  c. 75.45.301.14

  d. 11100010.23.14.67

- Solution

a. There must be no leading zero (045).

b. There can be no more than four numbers.

c. Each number needs to be less than or equal to 255.

d. A mixture of binary notation and dotted-decimal notation is not allowed.

Classful Addressing

- IPv4 addressing, at its inception, used the concept of **classes**.

✓ This **architecture** is called **classful addressing**.

- In **classful addressing,** the **address space** is divided into **five classes: A, B, C, D,** and **E**.

✓ Each class **occupies** some part of the **address space**.

	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

Example 4

- Find the class of each address.

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

c. 14.23.120.8

d. 252.5.15.111

- **Solution**

a. The first bit is 0. This is a class A address.

b. The first 2 bits are 1; the third bit is 0. This is a class C address.

c. The first byte is 14; the class is A.

d. The first byte is 252; the class is E.

Classes and Blocks

- One **problem** with **classful addressing** is that each class is **divided into a fixed number** of **blocks** with each **block** having a **fixed size**

Class	*Number of Blocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

Number of Blocks for class A = 2^7

Number of Blocks for class B = 2^{14}

Number of Blocks for class C = 2^{21}

Block size for class A = 2^{24}

Block size for class B = 2^{16}

Block size for class C = 2^8

Classes and Blocks continued

- Previously, when an organization requested a **block** of **addresses**, it was granted one in **class A, B,** or **C**.
- **Class A** addresses were designed for **large organizations** with a **large number** of **attached hosts** or **routers**.
- **Class B** addresses were designed for **midsize organizations** with **tens** of **thousands** of **attached hosts** or **routers**.
- **Class C** addresses were designed for **small organizations** with a **small number** of **attached hosts** or **routers**.

Classes and Blocks continued

- A **block** in **class A address** is **too large** for almost **any organization.**

✓ This means most of the **addresses** in **class A** were **wasted** and were **not used**.

- A **block** in **class B** is also **very large**, probably **too large** for many of the **organizations** that received a **class B block.**

- A **block** in **class C** is probably **too small** for many **organizations.**

Network ID and Host ID

- In classful addressing, an **IP address** in class A, B, or C is divided into **network ID** and **host ID**.

✓ These parts are of varying **lengths**, depending on the **class** of the **address**.

- In **class A, one byte defines** the **network ID** and **three bytes define** the **host ID.**
- In **class B**, **two bytes** define the **network ID** and **two bytes** define the **host ID**.
- In **class C, three bytes** define the **network ID** and **one byte** defines the **host ID**.

Subnet Mask

- Every device has an **IP address** with **two pieces**: the **server** or **network address** and the **client** or **host address**.

- The **subnet mask splits** the **IP address** into the **network addresses** and **host**,

  thereby **defining** which part of the **IP address** belongs to the **network** and which part belongs to the

device (host).

- The **device called** a **gateway** or **default gateway**

connects local devices to other **networks**

Subnet Mask-------

- This means that when a **local device** wants to **send information** to a **device** at an **IP address** on **another network**,

  it first **sends** its packets to the **gateway**, which then **forwards** the **data** on to its **destination** **outside** of the **local network**.

- A **subnet mask** is like an **IP address**, but for only **internal usage** within a **network**.
- **Routers** use **subnet masks** to **route data packets** to the **right place**.

Subnet Mask-------

- A **subnet mask** is a **32-bit number** created by **setting host bits** to all **0s** and **setting network bits** to all **1s.**
- In this way, the **subnet mask separates** the **IP address** into the **network** and **host addresses**.
- The **“255” address** is always **assigned** to a **broadcast address**, and the **“0” address** is always **assigned** to a **network address**.

✓ The **masks** for **classes A, B,** and **C** are shown on the next slide

Mask continued

- For **example**, the **mask** for a **class A address** has **eight 1s**, which means the **first 8 bits** of **any address** in **class A** define the **network ID**; the **next 24 bits** define the **host ID.**

Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

Subnetting

- **Subnetting** is a **method** of **dividing** a **single physical network** into **logical sub-networks (subnets)**.
- **Subnetting** allows a **business** to **expand** its **network** without requiring a **new network number** from its **Internet service provider.**
- The **goal** of **subnetting** is to **create** a **fast**, **efficient**, and **resilient computer network**.

➢ As **networks** become **larger** and more **complex**, the **traffic traveling** through them **needs more efficient routes**

Subnetting-----

- If all **network traffic** was **traveling** across the **system** at the **same time** using the **same route**, **bottlenecks** and **congestion** would **occur** resulting in **sluggish** and **inefficient backlogs**.
- **Creating** a **subnet** allows you to **limit** the **number** of **routers** that **network traffic** must **pass through**.
- **Subnetting** helps to **reduce** the **network traffic** and also **conceals network complexity**.
- A network is divided into **several smaller networks**.

✓ Each **smaller network** is called a **subnetwork** *or a* **subnet**

What is the use of Subnetting?

1. Reallocating IP Addresses

- A **limited number** of **host allocations** are **available** for each **class**; **for example**, **networks** with more than **254 devices require** a **Class B allocation**.
- Suppose a **network administrator** works with a **Class B** or **C network** and needs to **allocate 150 hosts** across **three physical networks** in **three different cities**.
- In that case, they must either **request** more **address blocks** for **each network** or **divide** the **network** into **subnets;**

that allow **administrators** to use **one block** of **addresses across multiple physical networks**.

What is the use of Subnetting?------

2. Improves Network Speed

- Divides **large network** into **small subnets**, and the purpose of these **subnets** is to **divide** a **huge network** into a **collection** of **smaller**, **interconnected networks** to **reduce network traffic**.
- **Subnets** **eliminate** the need for **traffic** to pass through **extraneous routs**, resulting in **faster network speeds**.

3. Improving Network Security

- **H**elps **network administrators** to **reduce network-wide threats** by **quarantining compromised areas** of the **network** and making it **more complex** for **trespassers (intruders)** to travel throughout an **organization's network**.

What is the use of Subnetting?------

4. Reliving Network Congestion

- If a **large portion** of an **organization's traffic** is intended to be **shared** regularly **across** a **group** of **computers**, putting them all on the **same subnet** **can help** **reduce network traffic**.
- Without a **subnet**, **data packets** from e**very** other **compute**r on the **network** would be **visible** to all **computers** and **servers**.

5. Efficiency

- To **simplify network traffic** by **eliminating** the need for **additional routers**.
- This **ensures**, **data** being **sent** can **quickly** as possible to its **destinatio**n, **avoiding** any **potential detours** that can **slow** it **down**.

Address Depletion

- The **flaws** in **classful addressing scheme** combined with the **fast growth** of the **Internet** led to the **near depletion** of the **available addresses**.
- Yet the **number** of **devices** on the **Internet** is much **less than** the 2^{32} **address space**.
- We have **run out** of **class A** and **B addresses**, and a **class C block** is **too small** for most **midsize organizations.**
- One **solution** that has **alleviated** the **problem** is the idea of **classless addressing**.

Classless Addressing

- To overcome **address depletion** and give more organizations access to the **Internet**, **classless addressing** was **designed** and **implemented**.

✓ In this **scheme**, there are **no classes**, but the addresses are **still granted** in **blocks**.

- **Address Blocks**
  - ✓ In **classless addressing**, when an **entity**, **small** or **large**, needs to be **connected** to the **Internet**, it is granted a **block (range)** of **addresses**.

Classless Addressing continued

- The **size** of the **block** (the **number** of **addresses**) varies based on the **nature** and **size** of the **entity**.
- For example, a **household** may be given only **two addresses**; a **large organization** may be given **thousands** of **addresses**.

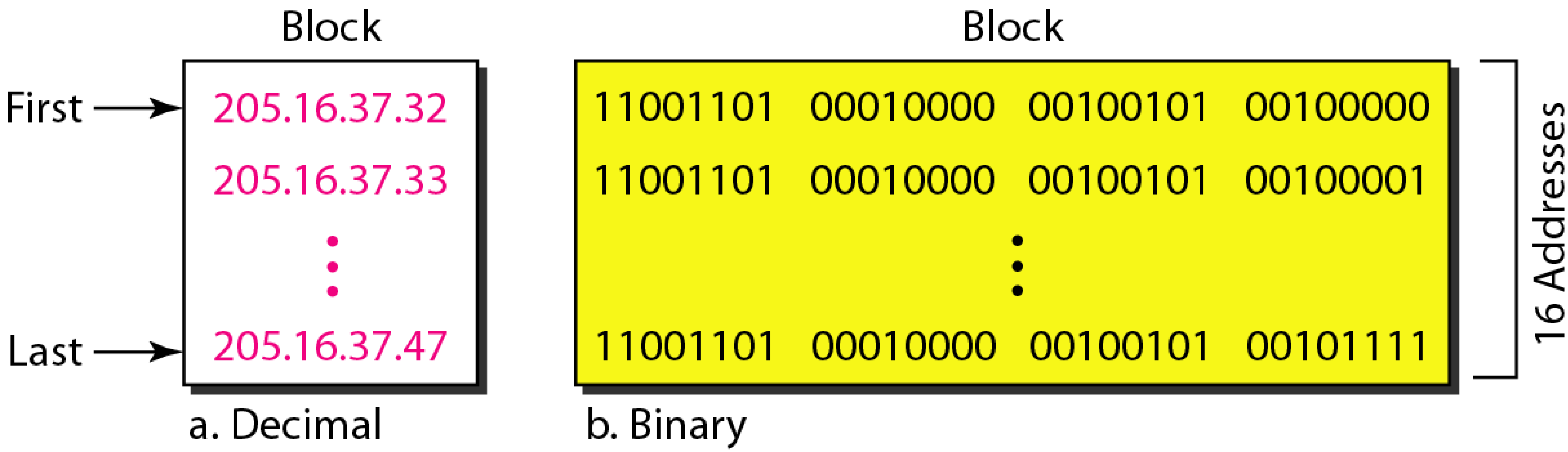
✓ An ISP, as the Internet service provider, may be given **thousands** or **hundreds** of **thousands** based on the **number** of **customers** it may **serve**.

Restriction

- To simplify the handling of **addresses**, the **Internet authorities** impose **three restrictions** on **classless address blocks:**

  1. The **addresses** in a **block** must be **contiguous, one after another.**
  2. The **number** of **addresses** in a **block must** be a **power** of **2 (1, 2, 4, 8 ....).**
  3. The **first address** must be **evenly divisible** by the **number** of **addresses.**

Restriction continued



- This figure shows a **block** of **addresses**, in both **binary** and **dotted-decimal notation**, granted to a **small business** that needs **16 addresses**.
- We can see that the restrictions are applied to this block. The **addresses** are **contiguous**.

Restriction continued

- The **number** of **addresses** is a power of 2 (16= 2^4), and the first **address** is divisible by **16**.
- The first address, when converted to a decimal number, is **3,440,387,360**, which when divided by **16** results in **215,024,210**.

Classless Addressing Mask

- A better way to define a **block** of **addresses** is to select any address in the **block** and the **mask**.
- As we discussed before, a mask is a **32-bit number** in which the **n** **leftmost bits** are **1s** and the **32-n** **rightmost** **bits** are **0s**.
- However, in **classless addressing** the **mask** for a **block** can take any value from **0 to 32**.

Classless Addressing Mask continued

- It is very convenient to give just the **value** of **n** preceded by a **slash (CIDR – Classless Inter Domain Routing notation).**
- The **address** and the **/n notation** completely **define** the whole **block** (the **first address,** the **last address,** and the **number of addresses**).

1. First Address

- The **first address** in the **block** can be found by setting the **32-n rightmost bits** in the **binary notation** of the **address** to **0s**.

➢ Example 5: A block of addresses is granted to a small organization.

- We know that one of the addresses is **205.16.37.39/28**. What is the **first address** in the **block**?

➢ **Solution**

- The **binary representation** of the given **address** is

11001101 00010000 00100101 00100111

- If we set **32−28 rightmost bits** to **0**, we get

11001101 00010000 00100101 00100000 or

205.16.37.32.

2. Last Address

- The **last address** in the **block** can be found by **setting** the **32- n rightmost bits** in the **binary notation** of the **address** to **1s**.
- Example 6: Find the **last address** for the **block** in **Example 5**.

➢ **Solution**

- The **binary representation** of the given **address** is

  11001101 00010000 00100101 00100111

- If we **set 32−28 rightmost bits** to **1**, we get

  11001101 00010000 00100101 00101111

  or

  205.16.37.47

3. Number of Addresses

- The **number** of **addresses** in the **block** is the difference between the **last** and **first address**.

✓ It can easily be found using the formula $\mathbf{2^{32-n}}$.

➢ **Example 7**

- Find the **number** of **addresses** in Example 5.

✓ **Solution**

- The **value** of **n** is **28**, which means that **number** of **addresses** is $\mathbf{2^{32-28}}$ or **16**.

Number of Addresses continued

- Another way to find the **first address**, the **last address**, and the **number** of **addresses** is to **represent** the **mask** as a **32-bit binary** (or **8-digit hexadecimal**) **number**.
- This is particularly **useful when** we are **writing** a **program** to **find** these **pieces** of **information**. In the above example the **/28** can be represented as

  11111111 11111111 11111111 11110000

  (**twenty-eight 1s** and **four 0s**). Find

a. The **first address**

b. The **last address**

c. The **number** of **addresses**.

- **Solution**

a. The **first address** can be found by **ANDing** the given **addresses** with the **mask**.

- **ANDing** here is done **bit** by **bit**.
- The result of **ANDing 2 bits** is **1** if both **bits** are **1s**; the **result** is **0** otherwise.

Address:	11001101	00010000	00100101	00100111
Mask:	**11111111**	**11111111**	**11111111**	**11110000**
First address:	11001101	00010000	00100101	00100000

b. The **last address** can be found by **ORing** the given **addresses** with the **complement** of the **mask**.

- **Oring** here is done **bit** by **bit**.
- The result of **ORing** **2 bits** is **0** if both **bits** are **0s**; the **result** is **1** otherwise.
- The **complement** of a **number** is found by changing **each 1** to **0** and **each 0** to **1**.

Address:	11001101	00010000	00100101	00100111
Mask complement:	**00000000**	**00000000**	**00000000**	**00001111**
Last address:	11001101	00010000	00100101	00101111

c. The **number** of **addresses** can be found by **complementing** the **mask**, interpreting it as a **decimal number**, and **adding 1** to it.

Mask complement:	**000000000 00000000 00000000 00001111**
Number of addresses:	15 + 1 = 16

Network Addresses

- A very important concept in **IP addressing** is the **network address**.
- When an organization is given a **block** of **addresses**, the organization is **free** to **allocate** the **addresses** to the **devices** that need to be **connected** to the **Internet**.
- The **first address** in the **class**, however, is normally (not always) treated as a **special address**.
- The **first address** is called the **network address** and **defines** the **organization network.**
- It **defines** the **organization itself** to the **rest** of the **world**.
- The **first address** is the one that is used by **routers** to **direct** the **message sent** to the **organization** from the **outside**.

Hierarchy

- IP addresses, like other addresses or identifiers we encounter these days, have levels of hierarchy.
- For example, a telephone network in Ethiopia has three levels of hierarchy.
- The leftmost three digits (251) define the country code, the next three digits (011, for example) define the area, the last seven digits (1112343, for example) define the subscriber number.

Two-Level Hierarchy: No Subnetting

- An IP address can define only two levels of hierarchy when not subnetted.
- The n leftmost bits of the address **x.y.z.t/n** define the network (organization network); the 32 – n rightmost bits define the particular host (computer or router) to the network.
- The two common terms are prefix and suffix.
- The part of the address that defines the network is called the prefix; the part that defines the host is called the suffix.
- The prefix is common to all addresses in the network; the suffix changes from one device to another.

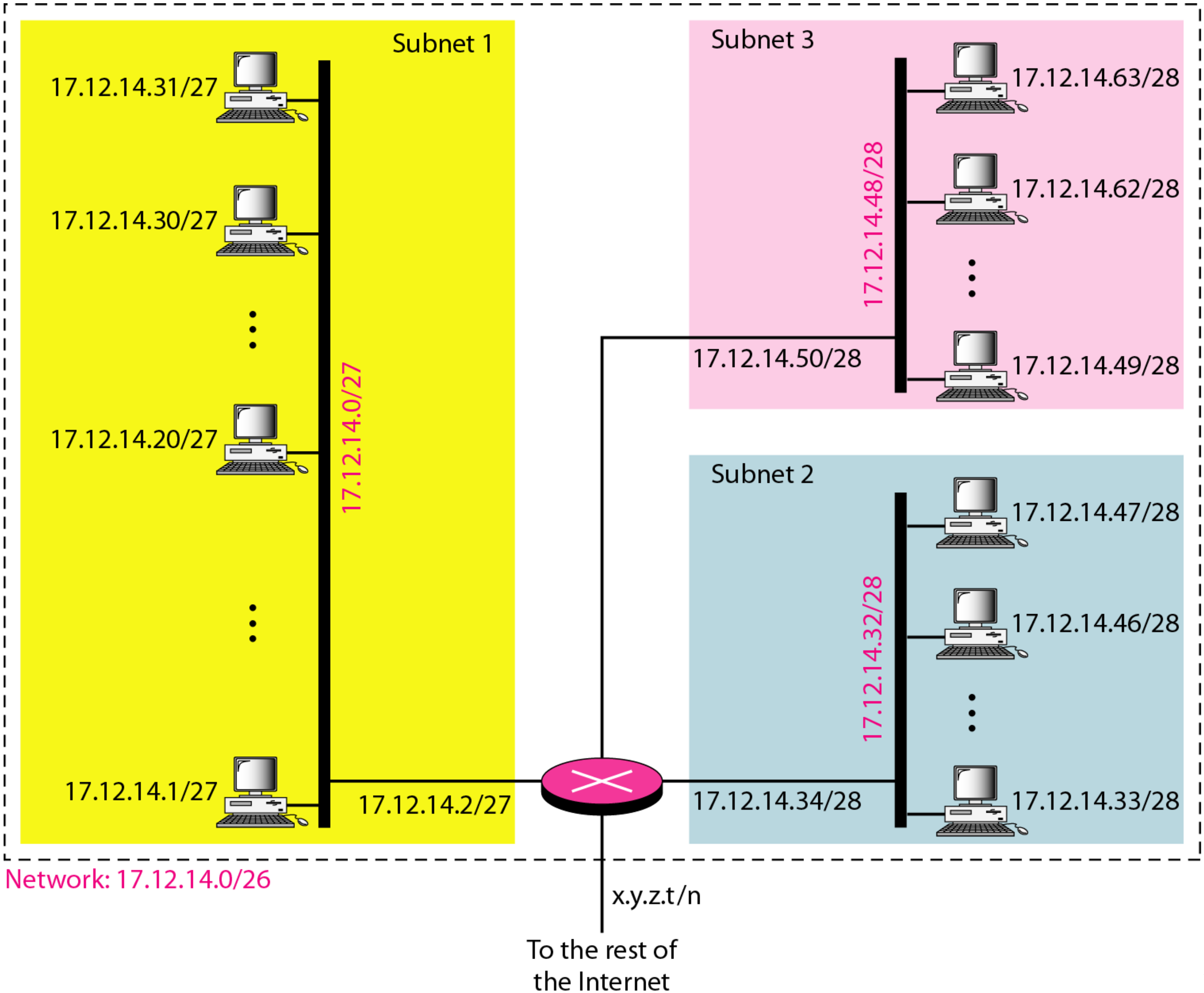
Three-Levels of Hierarchy: Subnetting

- An organization that is granted a large block of addresses may want to create clusters of networks (called subnets) and divide the addresses between the different subnets.
- The rest of the world still sees the organization as one entity; however, internally there are several subnets.
- All messages are sent to the router address that connects the organization to the rest of the Internet; the router routes the message to the appropriate subnets.
- The organization, however, needs to create small sub blocks of addresses, each assigned to specific subnets.

✓ The organization has its own mask; each subnet must also have its own

Example 7

- Suppose an organization is given the block 17.12.14.0/26, which contains 64 addresses.
- The organization has three offices and needs to divide the addresses into three sub blocks of **32**, **16**, and **16** addresses.
- We can find the new masks by using the following arguments:
  1. Suppose the mask for the first subnet is n1, then $\mathbf{2^{32-n1}}$ must be 32, which means that **n1 = 27**.
  2. Suppose the mask for the second subnet is n2, then 2^{32-n2} must be 16, which means that **n2 = 28**.
  3. Suppose the mask for the third subnet is n3, then 2^{32-n3} must be 16, which means that **n3 = 28**.
- This means that we have the masks 27, 28, 28 with the organization mask being 26.



Subnet 1
17.12.14.31/27
17.12.14.30/27
17.12.14.20/27
17.12.14.1/27
17.12.14.0/27
17.12.14.2/27
Subnet 3
17.12.14.63/28
17.12.14.62/28
17.12.14.48/28
17.12.14.50/28
17.12.14.49/28
Subnet 2
17.12.14.47/28
17.12.14.46/28
17.12.14.32/28
17.12.14.34/28
17.12.14.33/28
Network: 17.12.14.0/26
x.y.z.t/n
To the rest of the Internet

More Levels of Hierarchy

- The structure of classless addressing does not restrict the number of hierarchical levels.
- An organization can divide the granted block of addresses into sub blocks.
- Each sub block can in turn be divided into smaller sub blocks. And so on.
- One example of this is seen in the ISPs. A national ISP can divide a granted large block into smaller blocks and assign each of them to a regional ISP. A regional ISP can divide the block received from the national ISP into smaller blocks and assign each one to a local ISP.
- A local ISP can divide the block received from the regional ISP into smaller blocks and assign each one to a different organization.
- Finally, an organization can divide the received block and make several subnets out of it.

Address Allocation

- The next issue in classless addressing is address allocation. How are the blocks allocated?
- The ultimate responsibility of address allocation is given to a global authority called the **Internet Corporation for Assigned Names and Addresses (ICANN).**
- However, ICANN does not normally allocate addresses to individual organizations. It assigns a large block of addresses to an ISP.
- Each ISP, in turn, divides its assigned block into smaller sub blocks and grants the sub blocks to its customers.
- In other words, an ISP receives one large block to be distributed to its Internet users. This is called **address aggregation**: many blocks of addresses are aggregated in one block and granted to one ISP.

Example 8

- An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:

a. The first group has 64 customers; each needs 256 addresses.

b. The second group has 128 customers; each needs 128 addresses.

c. The third group has 128 customers; each needs 64 addresses.

- **Design the sub blocks and find out how many addresses are still available after these allocations.**

- Group 1: For this group, each customer needs 256 addresses. This means that 8 bits are needed to define each host. The prefix length is then 32 − 8 = 24. The addresses are

1st Customer:	*190.100.0.0/24*	*190.100.0.255/24*
2nd Customer:	*190.100.1.0/24*	*190.100.1.255/24*
. . .		
64th Customer:	*190.100.63.0/24*	*190.100.63.255/24*
Total $= 64 \times 256 = 16{,}384$		

- Group 2: For this group, each customer needs 128 addresses. This means that 7 bits are needed to define each host. The prefix length is then 32 − 7 = 25. The addresses are

1st Customer:	*190.100.64.0/25*	*190.100.64.127/25*
2nd Customer:	*190.100.64.128/25*	*190.100.64.255/25*
. . .		
128th Customer:	*190.100.127.128/25*	*190.100.127.255/25*
Total $= 128 \times 128 = 16{,}384$		

Contd.

- Group 3: For this group, each customer needs 64 addresses. This means that 6 bits are needed to each host. The prefix length is then 32 − 6 = 26. The addresses are

1st Customer:	*190.100.128.0/26*	*190.100.128.63/26*
2nd Customer:	*190.100.128.64/26*	*190.100.128.127/26*
. . .		
128th Customer:	*190.100.159.192/26*	*190.100.159.255/26*

Total $= 128 \times 64 = 8192$

- Number of granted addresses to the ISP: 65,536
- Number of allocated addresses by the ISP: 40,960
- Number of available addresses: 24,576

Network Address Translation (NAT)

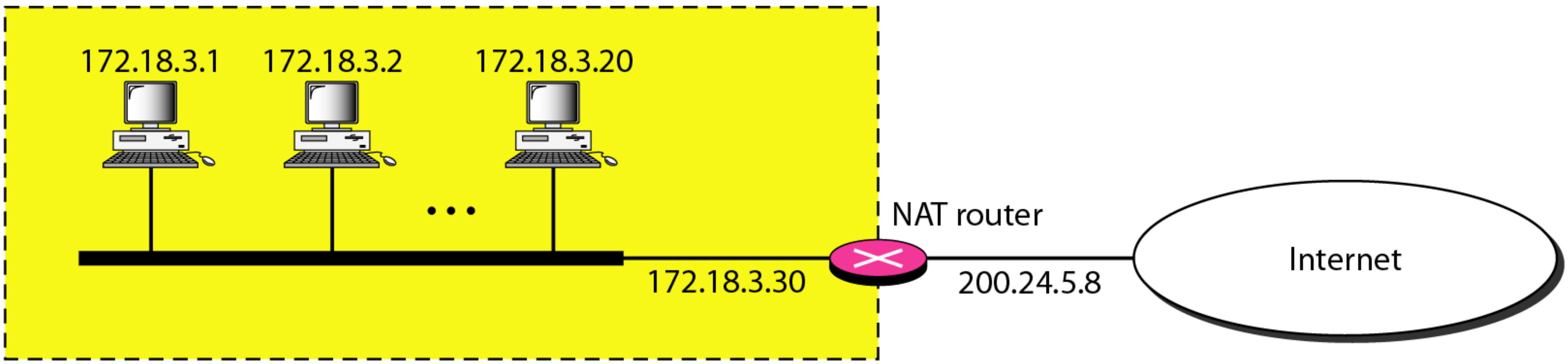
- Many users start to have more hosts to be connected to the internet
- IP addresses are in depletion

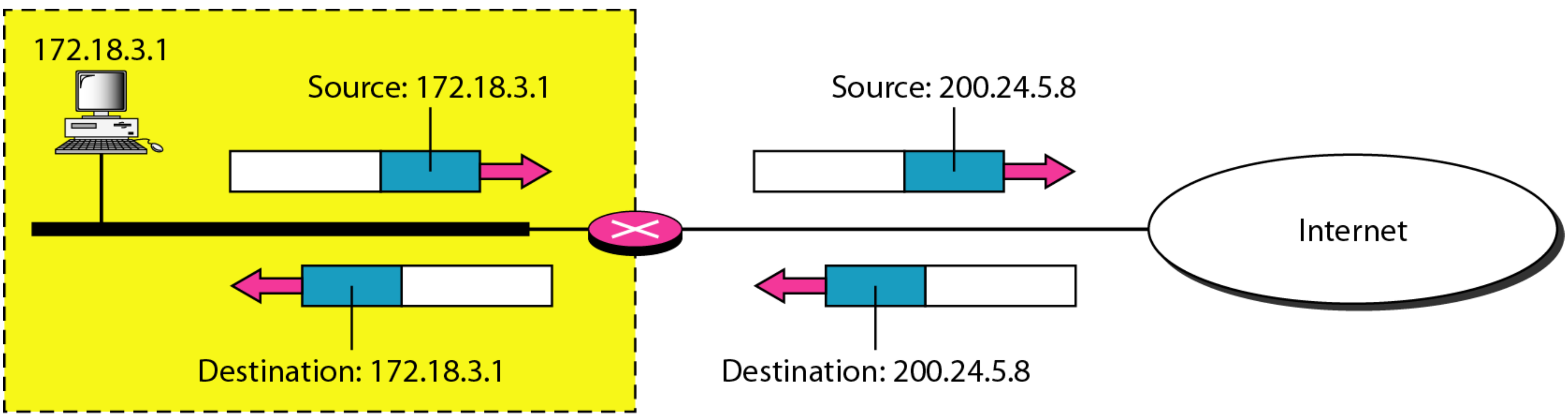
✓ **Solution:** NAT

- NAT enables a user to have a large set of addresses internally and one address, or a small set of addresses, externally.
- The traffic inside can use the large set; the traffic outside, the small set.

Range			*Total*
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}

Site using private addresses





Reading Assignment:

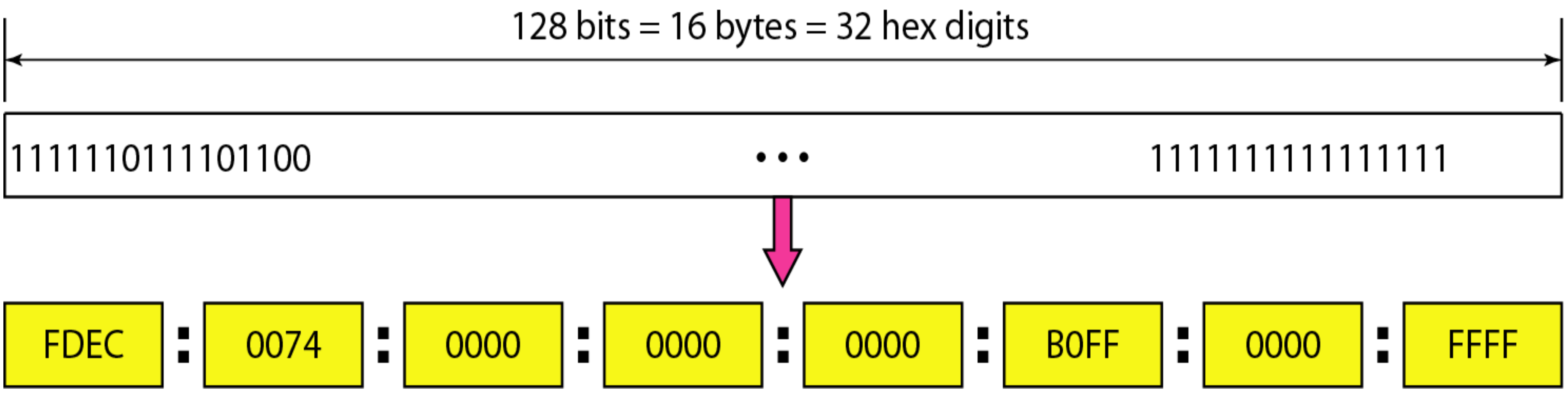
1. Read how NAT works
2. Investigate the global IP address scheme of Adama University and how they used it in NAT

IPv6 ADDRESSES

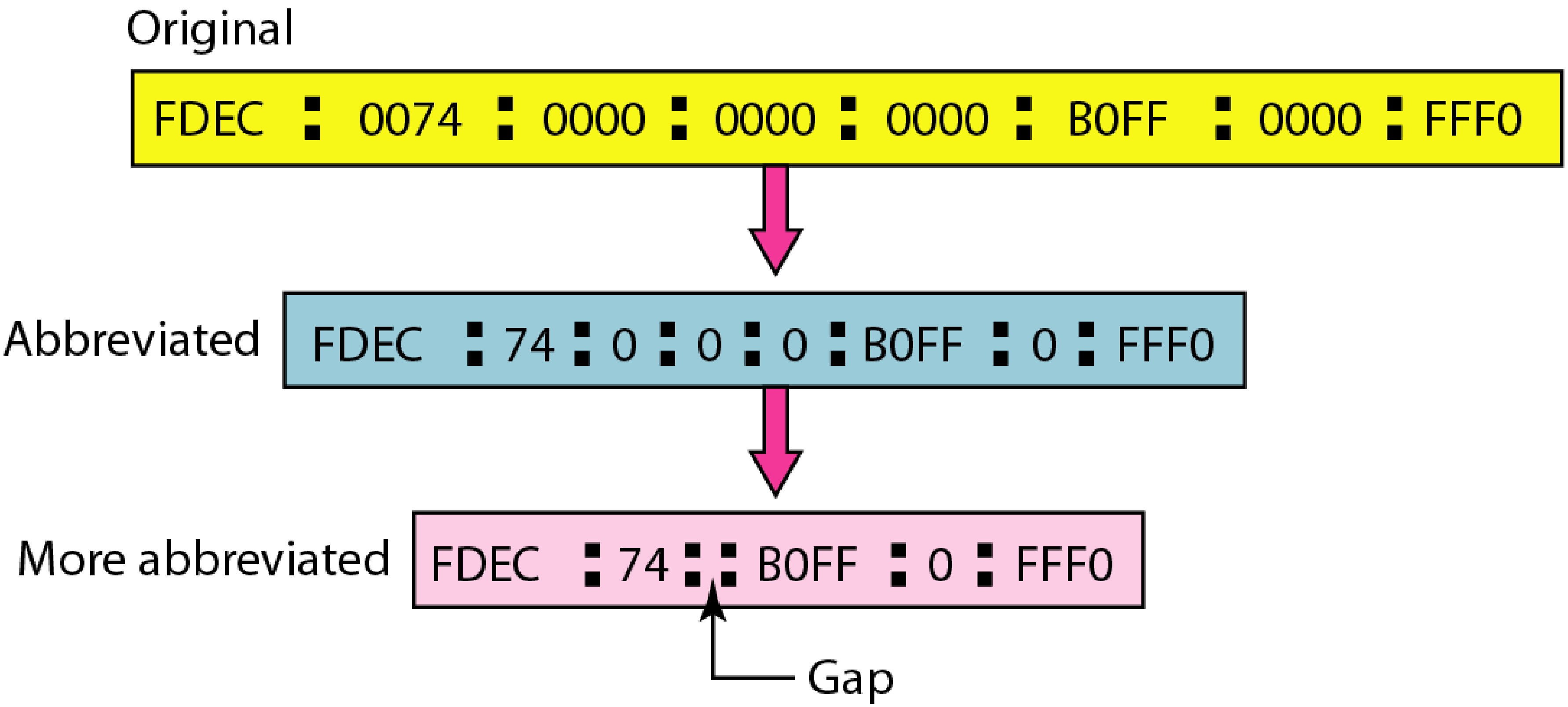
- Despite all short-term solutions, address depletion is still a long-term problem for the Internet.

✓ This and other problems in the IP protocol itself have been the motivation for IPv6.

- An IPv6 address is 128 bits or 32 hexadecimal digits long.



Abbreviated IPv6 addresses



Example 9

Expand the address 0:15::1:12:1213 to its original.

- Solution

✓ We first need to align the left side of the double colon to the left of the original pattern and the right side of the double colon to the right of the original pattern to find how many 0s we need to replace the double colon.

```
xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
   0:  15:                 :   1:  12:1213
```

This means that the original address is.

```
0000:0015:0000:0000:0000:0001:0012:1213
```

ADDRESS MAPPING

- The delivery of a packet to a host or a router requires two levels of addressing:

✓ logical and physical.

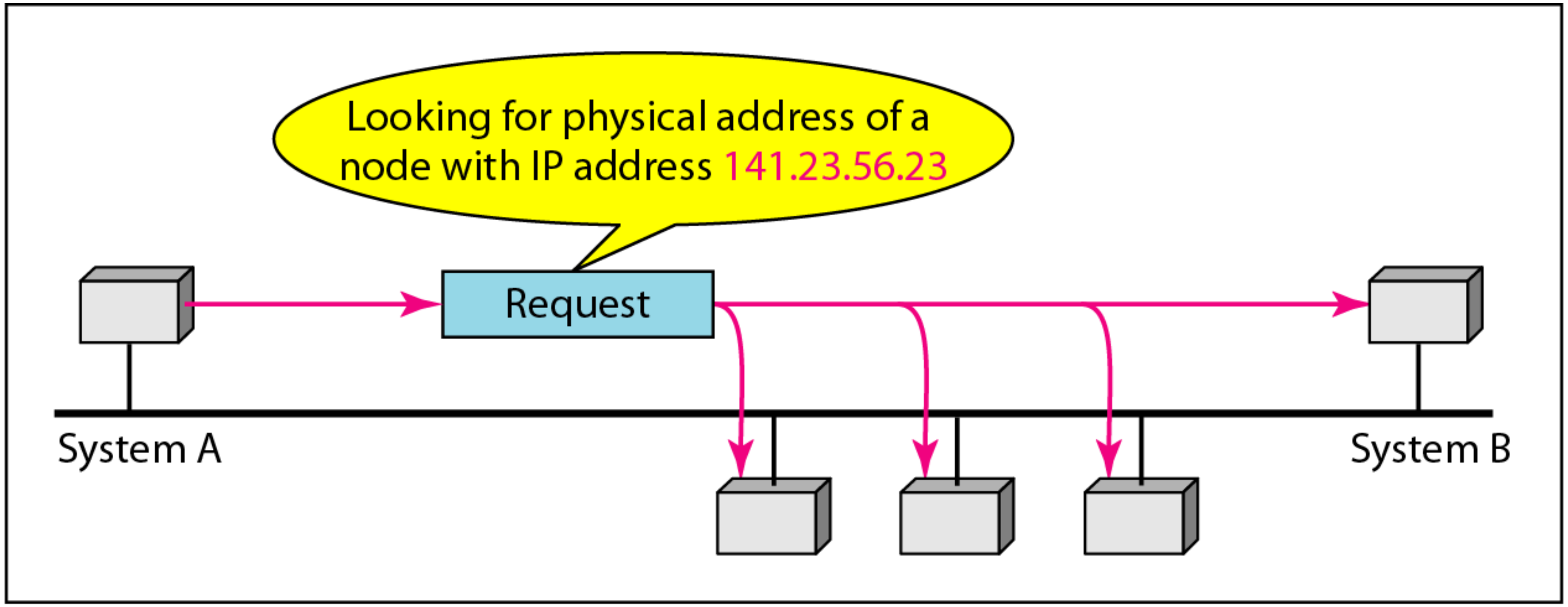
- We need to be able to map a logical address to its corresponding physical address and vice versa.
- This can be done by using either static or dynamic mapping.
- IP is used for logical addressing
- MAC is used for physical addressing in a local network such as Ethernet

Mapping Logical to Physical Address: ARP

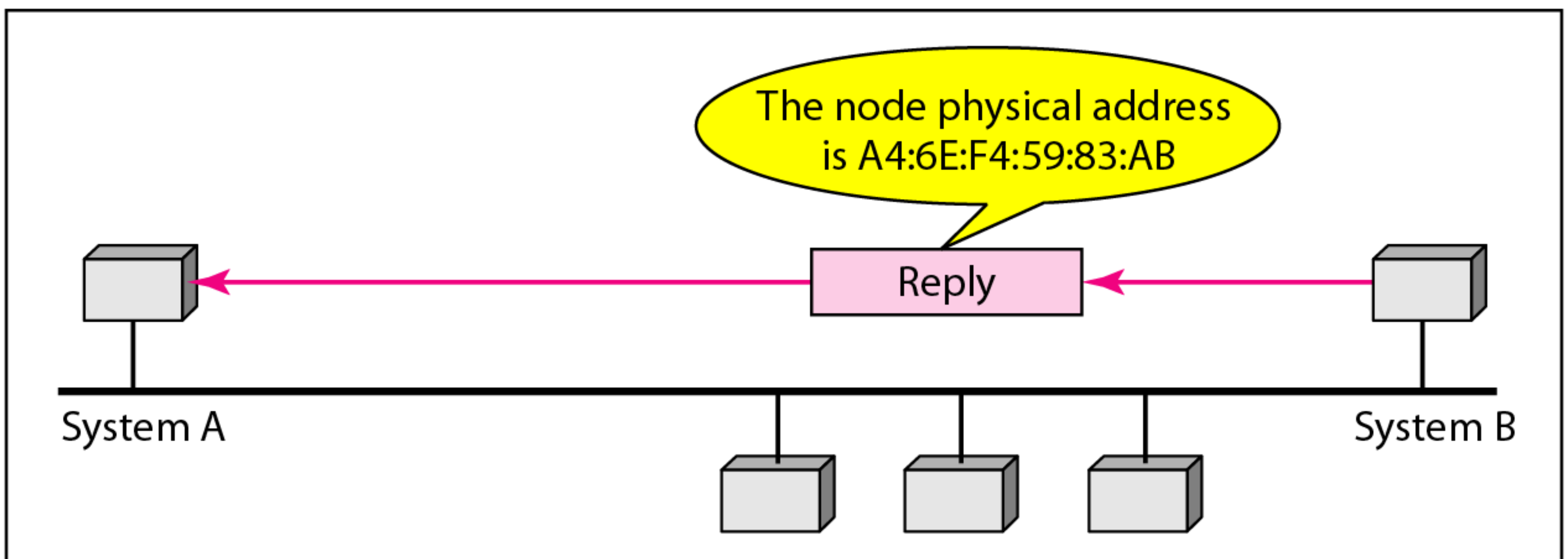
- Anytime a host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver.
- The logical (IP) address is obtained from the DNS if the sender is the host or it is found in a routing table if the sender is a router.
- But the IP datagram must be encapsulated in a frame to be able to pass through the physical network.

Mapping Logical to Physical Address: ARP-------

- This means that the sender needs the physical address of the receiver. The host or the router sends an ARP query packet.
- The packet includes the physical and IP addresses of the sender and the IP address of the receiver.
- Because the sender does not know the physical address of the receiver, the query is broadcast over the network



a. ARP request is broadcast



b. ARP reply is unicast

Mapping Physical to Logical Address: RARP

- There are occasions in which a host knows its physical address, but needs to know its logical address.
- This may happen in two cases:

1. A diskless station is just booted.

- The station can find its physical address by checking its interface, but it does not know its IP address.

2. An organization does not have enough IP addresses to assign to each station; it needs to assign IP addresses on demand.

- The station can send its physical address and ask for a short time lease.

RARP-------------------

- Reverse Address Resolution Protocol (RARP) finds the logical address for a machine that knows only its physical address.
- Each host or router is assigned one or more logical (IP) addresses, which are unique and independent of the physical (hardware) address of the machine.
- To create an IP datagram, a host or a router needs to know its own IP address or addresses.
- The IP address of a machine is usually read from its configuration file stored on a disk file.

RARP-------------------

- The machine can get its physical address (by reading its NIC, for example), which is unique locally.
- It can then use the physical address to get the logical address by using the RARP protocol.
- A RARP request is created and broadcast on the local network.
- Another machine on the local network that knows all the IP addresses will respond with a RARP reply.

ICMP

- The IP protocol has no error-reporting or error-correcting mechanism.
- The IP protocol also lacks a mechanism for host and management queries.
- The Internet Control Message Protocol (ICMP) has been designed to compensate for the above two deficiencies.

✓ It is a companion to the IP protocol.

- PING and TRACEROUTE are two tools for ICMP