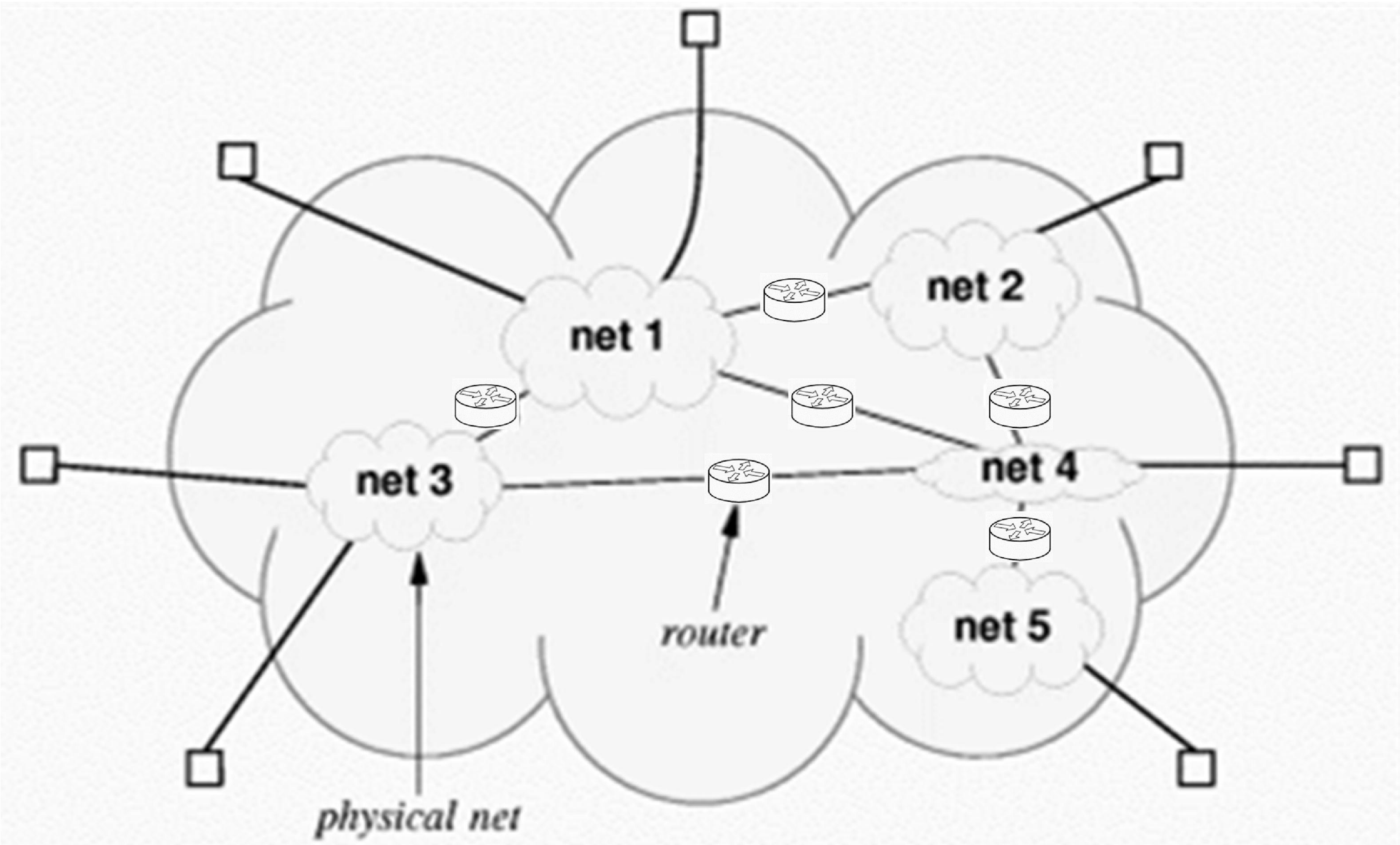
IP Addresses

- An *internet* must operate like any other network:
  - It must allow hosts to send and receive data to each other
  - To facilitate this some type of *addressing scheme* is required
    - i.e. all hosts require a unique address
- So *addressing* is a critical component of the *internet abstraction*
- The *Internet Protocol* (IP) defines such an addressing scheme
  - IP addressing is independent of the underlying physical addressing schemes

A Multi-node Internet



IP Address Hierarchy

- Each host/router *interface* is assigned a unique 32-bit *IP address*
  - This address is used in packets that are sent across an internet
- Each IP address consists of two parts: a *prefix* and a *suffix*
  - Each physical network is assigned a unique *network number* which forms the ***network prefix*** part of a host's IP address.
    - It uniquely identifies the physical network to which the host is attached
  - The ***host suffix*** uniquely identifies the *host* on that network.
- *Network numbers/prefixes* are assigned globally, *host suffixes* are assigned locally

Dotted Decimal Notation

- Representing IP addresses in 32-bit binary form is only suitable for computers.
- A human-friendly version of IP addresses is known as *dotted-decimal notation.*
- Here each <u>octet</u> of the IP address is represented as a decimal value separated by a dot '.'
  - Examples include: 129.52.6.0, 128.10.2.3 and 128.128.255.0
  - Each decimal value represents 8-bits,
  - Consequently, the range of decimal numbers extends from 0 - 255

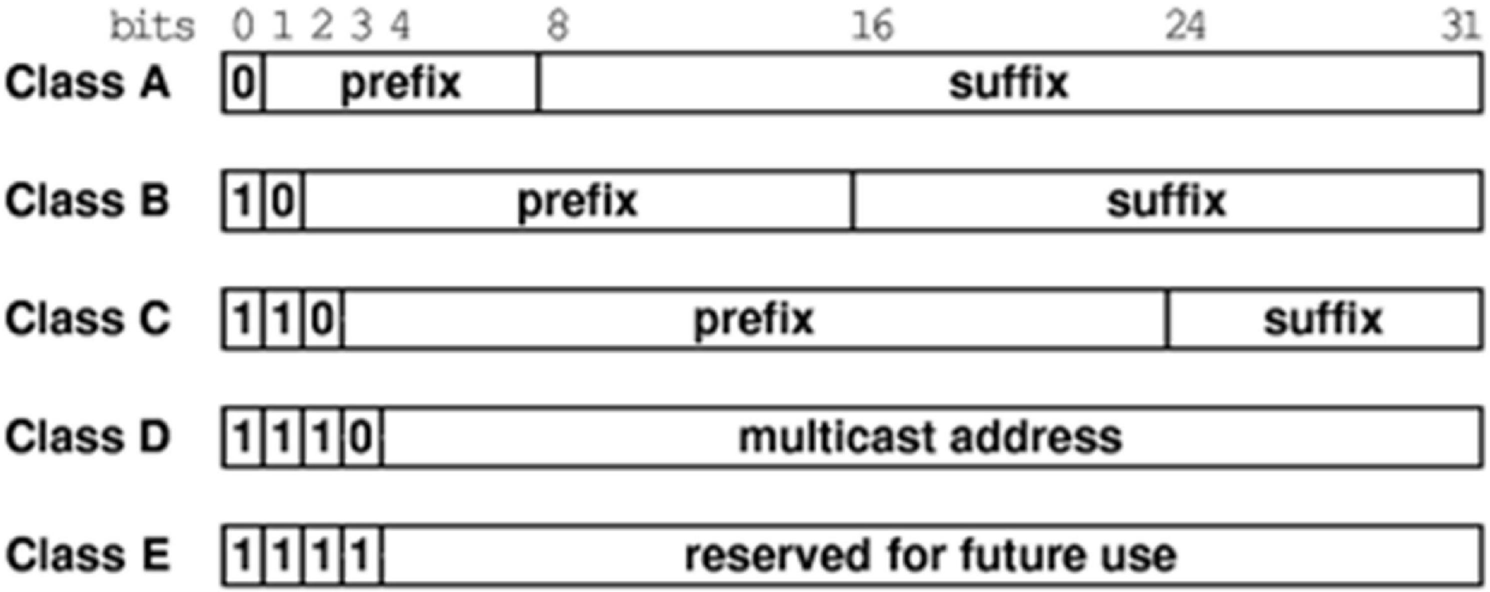
IP Address Hierarchy

- The IP addressing scheme must accommodate large <u>and</u> small internets:
  - This was originally achieved using *classful IP addressing*
- Classful Addressing was the first attempt to organize the IP address space.

Original *Classful* IP Addressing

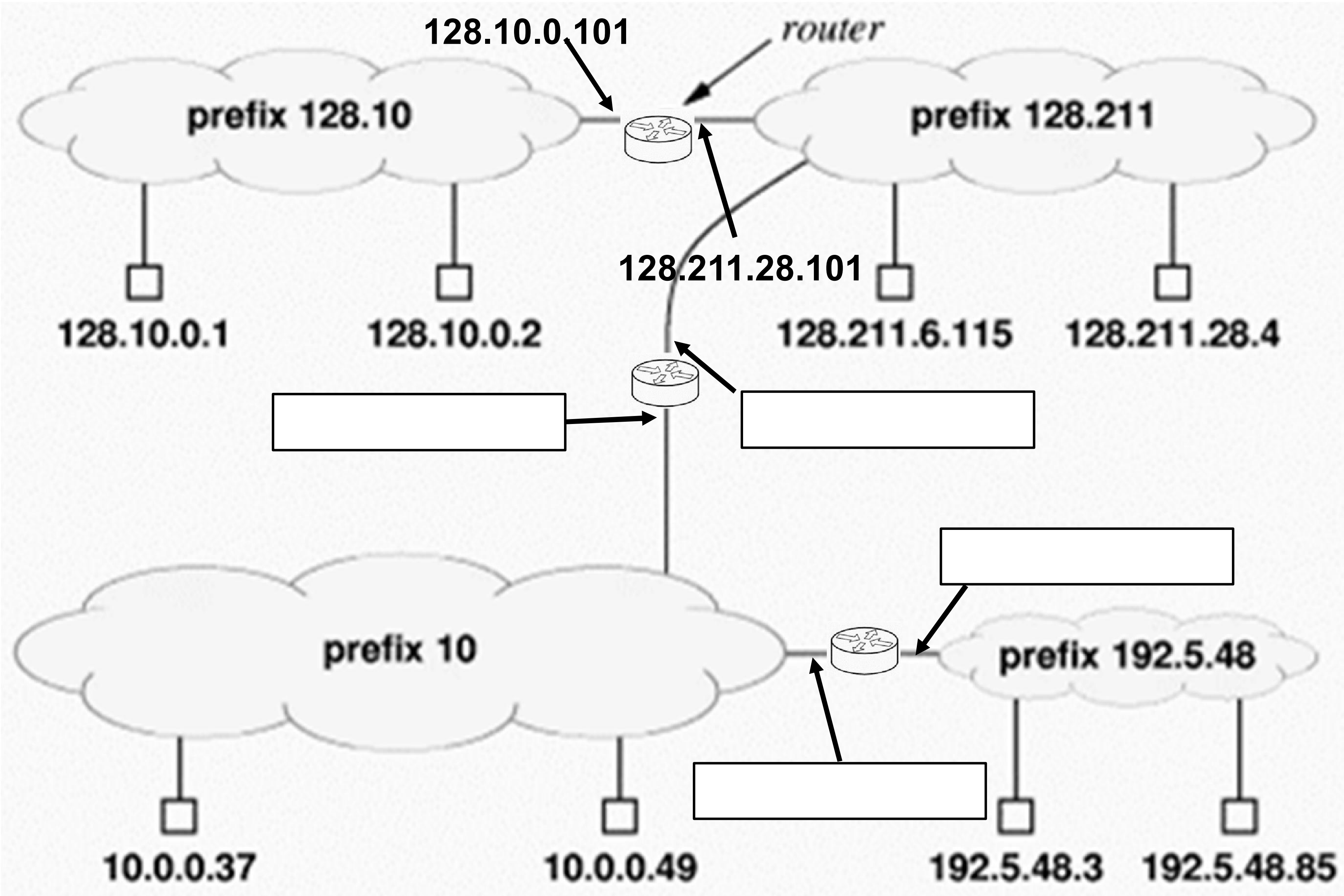
- Here the IP address space was divided into **five** *classes:*
  - Each class had different size *prefix* and *suffix* portions to accommodate large and small networks.
  - The first four bits identify the *class* to which the address belongs (see next slide).
  - **Classes A, B and C** are known as the *primary classes* because they are used for *host addressing.*
  - Class D addresses are used for *multicasting*
  - Class E was reserved for future use.

Classful IP Addressing Scheme



- Note the different sizes of the *prefix* and *suffix* portions for each class:
  - For Class A addresses the first octet is the ***network prefix*** and the last three octets are the **host suffix.**
  - Notice also the split between *network prefix* and *host suffix* for Classes B and C.

A *classful* addressing example



Routing of IP Packets

- Q: Why is the separation of prefix and suffix important?
- A: Routers are responsible for directing datagrams/packets onto their final destination.
  - This is called *routing.*
  - The router needs to look at the address to make a decision about where to route a packet.
  - This is similar to the Post Office delivering letters.

Routing of ‘Letters’

- Letters are routed through *National* and *Local* sorting offices.
- National Sorting Offices tend to be far away from the destination.
- They only need to examine the County/Town to route the letter towards a sorting office closest to the destination.

Routing of ‘Letters’

- The Local Sorting Office closest to the destination examines the Street Name and House Number to make the final delivery.
- This Post Office analogy is similar to the way Routers route incoming IP packets.

Back to routing of ‘Packets’

- Routers far away from the Destination router can be considered *National* Sorting Offices:
  - They only need to examine *network prefixes* (similar to *County/Town*) in an attempt to route the packet towards the final destination router.
- The Router closest to the destination station can be considered a *Local Sorting* Office
  - They examine the entire destination IP address including the *host suffix*.

Routing of IP Packets using Classful Addressing

- Classful IP addresses are *self-identifying* as the *network prefix* portion of an IP address can be computed from the address itself:
  - The first four bits will determine the class and hence the *network prefix*.
  - Refer to the next slide to see how the bits relate to the classes.
- This makes it easy for "far away" routers to route packets towards the destination router.

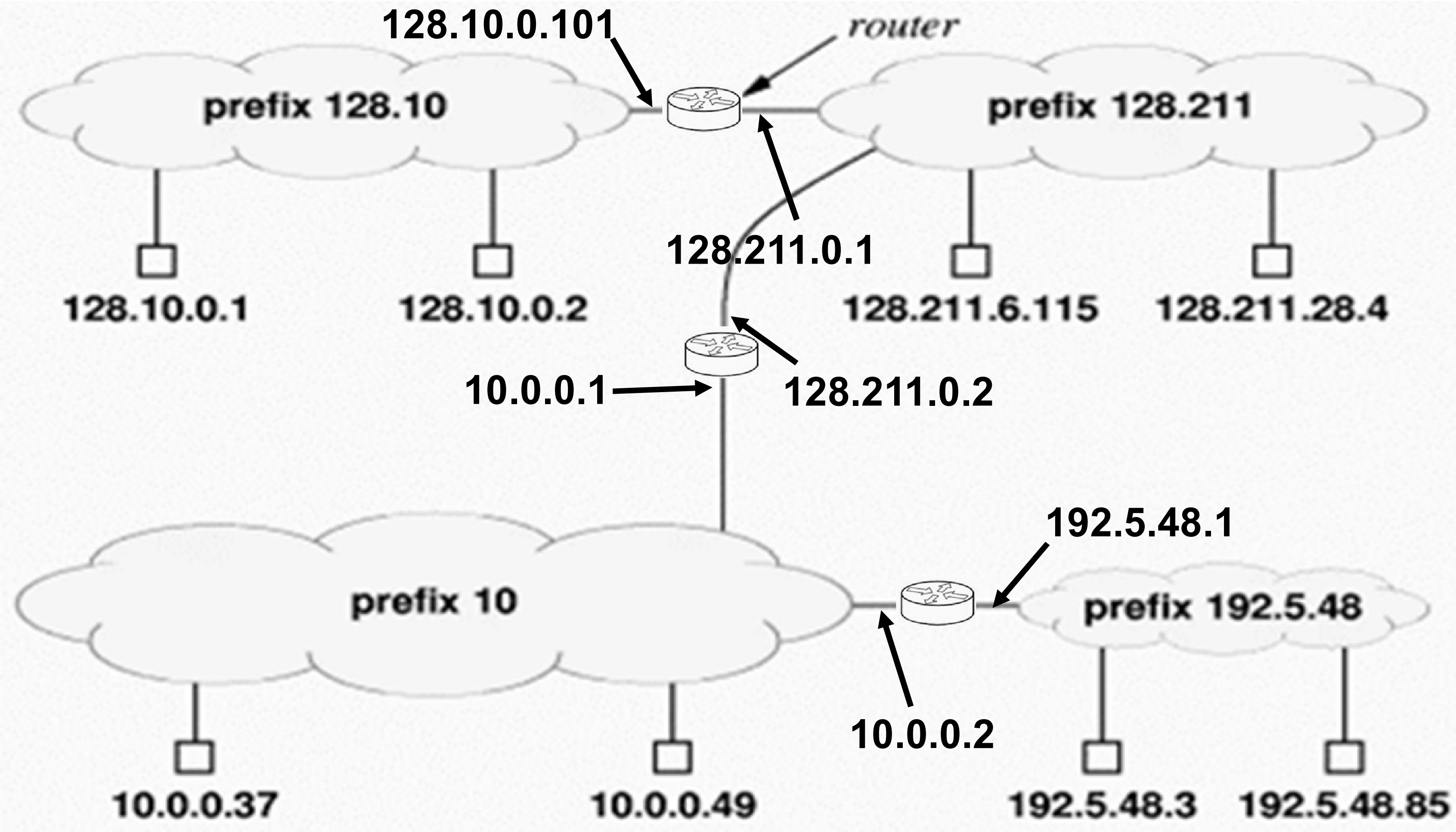
Classful IP address values

First Four Bits Of Address	Table Index (in decimal)	Class of Address
0000	0	A
0001	1	A
0010	2	A
0011	3	A
0100	4	A
0101	5	A
0110	6	A
0111	7	A
1000	8	B
1001	9	B
1010	10	B
1011	11	B
1100	12	C
1101	13	C
1110	14	D
1111	15	E

Routing of IP Packets using Classful Addressing

- Whilst *dotted-decimal notation* presents IP addressing in a simple, human-readable form, it hides the first four bits:
  - Consequently, the Classes are not easily discernible,
  - Instead, humans have to rely on *range values* in the first octet to determine the Class of the address:
    - For Class A addresses the **first** octet is in the range 0-127,
    - For Class B addresses the range is 128-191, and,
    - For Class C addresses the range is 193-223.

A *classful* addressing example



Note: Routers are allocated multiple IP addresses; one per network to which they connect.

Subnet and Classless IP addressing

- A problem with *Classful addressing* is that it results in an unequal division of the *IP address space.*
- As the global Internet has grown, the use of *classful addressing* has become problematic:
  - The IP address space is being exhausted,
  - Many addresses remain unused,
  - Refer to examples in class.
- Two new addressing methods are used to overcome this: *subnet addressing* and *classless addressing*
- Here the division between the prefix and suffix portions can occur on any bit boundary:
  - e.g. a 20-bit *network prefix* + 12-bit *host suffix* is possible.

Subnet and Classless IP addressing

- To facilitate Classless Addressing, an additional piece of information is defined with each address block:
  – Known as an *address mask* or *subnet mask.*
- **Masks** are 32-bit values that facilitate the extraction of the *network prefix* from any given IP address:
  – They are comprised of a contiguous sequence (unbroken sequence) of 1 bits followed by a contiguous sequence of 0 bits.
  – They can be represented in dotted-decimal notation just like IP addresses.
  – Refer to the following slide for some examples.

Example Classless IP Addressing Allocation

Network Number	Mask
30.0.0.0	255.0.0.0
40.0.0.0	255.0.0.0
128.1.0.0	255.255.0.0
192.4.10.0	255.255.255.0

- Notice how each router is assigned an IP address on each of the networks it is connected to.

Subnet and Classless IP addressing

- Note the *address mask* or *subnet mask* for the 192.4.10.0/24 network.
- The ***/XX*** (‘slash’ XX) is another way to represent masks:
  - /24 means that the mask is comprised of 24 ONE bits followed by 8 (32-24) ZERO bits*:*
- This notation is known as Classless Inter-Domain Routing (CIDR):
  - It is a human-friendly way of representing *address masks*
  - In *dotted-decimal notation,* this mask can be represented as:

    255.255.255.0
  - The first three octets are all ONEs, the last octet is all ZEROs.

Using Address Masks to route packets

- Following on from the previous discussion on routing.
- Recall that for any given destination IP address, the router must determine the *network prefix* portion:
  - The router uses the *network prefix* to query its *routing table*.
- With Classless addressing, routers use the ***Mask*** (aka: *network masks*, *subnet masks*) to extract the *network prefix* portion from a destination IP address:
  - Recall that with *classful address*, the router simply examined the first four bits of the address to determine the class.

Using Address Masks to route packets

- For an incoming packet with a destination IP address, D, the router tests the following condition: **A = = (D&&M)**
- Where:
  - **A** is the IP address (network number) of networks that the router knows about,
  - **M** is the mask associated with the network, and,
  - **D** represents a destination IP address that the router needs to make a routing decision.

Address Masks

- For example, consider the following:

  A = 11000000 00000100 00001010 00000000

  D = 11000000 00000100 00001010 00000011

  M = 11111111 11111111 11111111 00000000

- The mask, **M**, is ‘applied’ to the Destination IP address, **D**
  - i.e. **D && M**
  - The **AND** operation effectively zeros out the last eight bits of **D**.
- The result is then compared to the **A** address.

Address Masks

- If they match, then the Destination IP address, **D** is said to **belong** to the network, **A**:
  - The packet containing the Destination IP address, **D**, is then **routed** towards network **A,**
  - The packet is routed to the address indicated by the ***Next Hop*** field in the routing table (refer to next slide).
- Otherwise, the next entry in the routing table will be tried using the above approach.

Example IP Routing Table using Classless Addressing

(a)

Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	deliver direct
128.1.0.0	255.255.0.0	deliver direct
192.4.10.0	255.255.255.0	128.1.0.9

Classless Addressing and the IP Address Space

- *Classless addressing* makes more efficient use of the IP address space.
- Consider an example of a single *class B prefix* (16-bit prefix): 128.211.0.0
  - With *classful addressing* this Class B prefix can only be used to identify a single network with approx. 65K host addresses.
  - With *classless addressing,* this Class B prefix can be considered as having a 16-bit mask:
    - In dotted-decimal notation: 255.255.0.0
    - In CIDR notation: /16
- Using *classless addressing*, there are multiple approaches to dividing this large address block into smaller blocks.

Classless Addressing V's IP Address Space

- Larger value address masks (e.g. /17, /18, /19 etc.) can be used to partition the *Class B address* block into smaller address blocks (known as ***subnets***)
  - For example, two customers could be allocated subnet address blocks with /28 masks (/28 > /16):

    128.211.0.16/28 and, 128.211.0.32/28

  - Although both subnets have the same mask size (28 bits), the network prefixs differ and are *unique.*
  - In addition, most of the original /16 address space is still available for allocation to other customers.

Basic Subnetting

- Subnetting allows for the creation of **multiple** logical networks from a single address block.
  - Subnets are created by taking one, or more, of the *host* bits from the original address block and using them as *network* bits in the smaller subnet blocks:
    - This is achieved by increasing the mask of the original large address block. For example, from **/16** to **/20**
    - The more *host bits* from the original large address block that are used, the more subnets can be defined.

Basic Subnetting

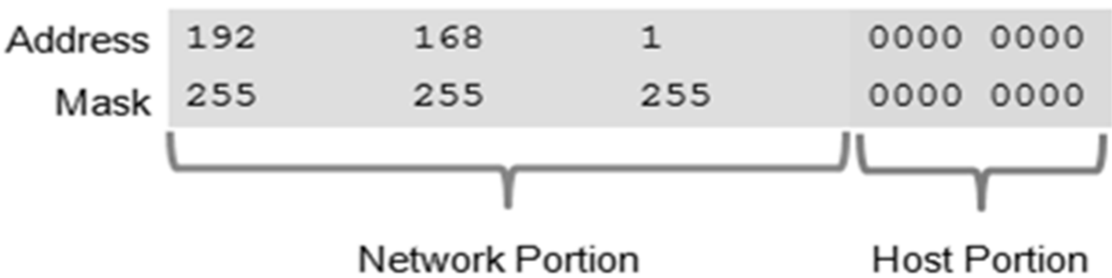
- For each additional host bit **taken**, the number of subnetworks available is doubled:
  - For example, with **one** host bit taken, two subnets can be defined. With two host bits taken, four subnets are created etc.
  - However, with each host bit taken, fewer host addresses are available for each of the subnets.

Basic Subnetting

- For example, if **n**-bits are taken from the original host portion:
  - The number of ***subnets*** created is: $\mathbf{2^n}$
  - The total number of addresses per subnet is $\mathbf{2^m}$ (**m** = the number of host bits remaining)
  - The number of **<u>usable</u>** host addresses: $\mathbf{2^m - 2}$;
    - The first address in the block is used to identify the *network*,
    - The last address in the block is the *Broadcast Address* (explained in class).

Basic Subnetting Example

- Given address block:192.168.1.0/24, divide this network into **2** subnets:
- Looking at the original address block:
  - **/24** means 8 hosts bits i.e. $\mathbf{2^8}$ = 256, Addresses
    - Of these 256 addresses, **254** are usable for host addresses
      - i.e. $\mathbf{2^8 - 2}$ = **254**.



Sizewise, this is similar to a Class C address block with: 8 host bits i.e. 2^8 = 256 addresses, of which 254 are usable.

Basic Subnetting

- To divide this network into 2 subnets, **one** bit is taken from the original <u>host</u> portion:
  - This extends the original mask (255.255.255.0) by one bit as follows:

    11111111.1111111.1111111.**1**0000000

    - In dotted-decimal notation (DDN): **255.255.255.128**
    - In CIDR notation: ***/25***
    - Note: **X** below represents the single host bit taken from the *host* portion of original address block to be used in the network prefix.

Original	192.	168.	1.	X	000 0000
Mask	255.	255.	255.	X	000 0000

Basic Subnetting

Original	192.	168.	1.	x	000 0000	Network 192.168.1.0/24
Mask	255.	255.	255.	x	000 0000	Mask: 255.255.255.0

Taking 1 bit from the original host portion creates 2 subnets, each with the same extended subnet mask: **255.255.255.128** or **/25**

For each new sub-net, there are: 7 Host Bits, i.e. 2^7 = 128 addresses, of which 126 are usable host addresses.

Address allocations for Subnet 0:

Address range: 192.168.1.**0-127**

Network address: 192.168.1.**0**

Mask: 255.255.255.**128** or, /**25**

Address allocations for Subnet 1:

Address range: 192.168.1.**128-255**

Network address: 192.168.1.**128**

Mask: 255.255.255.**128** or, /**25**

Basic Subnetting

- The single host bit taken from the original host portion gives: $\mathbf{2^1 = 2}$ Subnets:
  - Leaving **7** bits for the host portion i.e. so $\mathbf{2^7 = 128}$ addresses per subnet,
  - With the network and broadcast addresses removed this leaves $\mathbf{2^7 - 2 = 126}$ **usable host addresses** on each subnet
  - The *Address Table* needs to be completed:
    - The *Magic Number* approach is one way to determine the addresses in each subnet.

Basic Subnetting using Magic Numbers

- The *Magic Number* approach:
  - Determine the mask for each new subnet.
    - In the above example the mask is determined to be:

      255.255.255.128
  - Looking from left-to-right along the new mask, seek the last octet that is non-zero and less than or equal to 255.
    - In this case, the fourth octet matches this criterion.
    - Subtract this octet from 256 as follows:
      - 256 – 128 = 128 which is *The Magic Number*

Basic Subnetting

- The *Magic Number* (MN) is used to determine the starting network address for <u>each</u> of the Subnets:
  - To determine network address of the **next** subnet, simply add the MN to the previous network address, taking care to add the MN to the octet from which it was derived.
    - In this example, the network address of the <u>first</u> subnet is the first address in the original address block: 192.168.1.0
    - As the MN (128) was found in the fourth octet, the MN is added to the fourth octet: 192.168.1.**<u>0</u>**, to give the network address of the <u>next</u> subnet: 192.168.1.**<u>128</u>**

Basic Subnetting

- With the *Magic Number* approach, the Address Table is easy to complete as follows:

Subnet	Network Address	Host Addresses	Broadcast Address	Mask
0	192.168.1.0	192.168.1.1 – 126	192.168.1.127	/25
1	192.168.1.128	192.168.1.129 - 254	192.168.1.255	/25

Subnets in Use

Subnet 0

Network 192.168.1.**0-127**/**25**

Subnet 1

Network 192.168.1.**128-255**/**25**

Address Range for 192.168.1.0/25 Subnet

Network Address	192. 168. 1. 0 000 0000	= 192.168.1.0
First Host Address	192. 168. 1. 0 000 0001	= 192.168.1.1
Last Host Address	192. 168. 1. 0 111 1110	= 192.168.1.126
Broadcast Address	192. 168. 1. 0 111 1111	= 192.168.1.127

All Host bits = 0

All Host bits = 1

Address Range for 192.168.1.128/25 Subnet

Network Address	192. 168. 1. 1 000 0000	= 192.168.1.128
First Host Address	192. 168. 1. 1 000 0001	= 192.168.1.129
Last Host Address	192. 168. 1. 1 111 1110	= 192.168.1.254
Broadcast Address	192. 168. 1. 1 111 1111	= 192.168.1.255

All Host bits = 0

All Host bits = 1

Basic Subnetting

- Given an address block of 192.168.1.0 /24, we wish to divide this network into 4 subnets:
  - Determine the power of 2 to provide 4 networks:
    - i.e. $\mathbf{2^{?} = 4}$ (Note the number of subnets will be a power of 2)
    - Hence **two** bits are required to be taken from the original host portion.
    - This leaves **6 bits** remaining for host addresses.
  - Having calculated the number of host bits required for each subnet, the mask can be determined as follows:

Basic Subnetting

- The new Subnet mask is: 11111111.1111111.11111111.**11**000000
  - In DDN: 255.255.255.192
  - In CIDR notation: /26
- Using the *Magic Number* approach to determine the addresses in each subnet.

Basic Subnetting

- With a Mask for each subnet of:

255.255.255.192

  – Looking from left-to-right along the new mask, seek the last octet that is non-zero and less than or equal to 255.

  – In this case, the fourth octet matches this criterion.

  – Subtract this octet from 256 as follows:

256 – 192 = 64 which is *The Magic Number*

- The Address Table can be completed as follows:

Basic Subnetting

Subnet	Network Address	Host Addresses	Broadcast Address	Mask
0	192.168.1.0	192.168.1.1 – 62	192.168.1.63	/26
1	192.168.1.64	192.168.1.65 – 126	192.168.1.127	/26
2	192.168.1.128	192.168.1.129 – 190	192.168.1.191	/26
3	192.168.1.192	192.168.1.193 - 254	192.168.1.255	/26

- See how the *MN* (64) is used to determine the *network address* for each subnet.
  - Note: the MN was derived from the fourth octet. Consequently, it is added to this octet in the previous subnet’s network address.

Subnetting a Class B Network

- Given an address block of **172.25.0.0 /16**, we wish to divide this network into 11 subnets with each subnet catering for 3000 hosts:
  - Determine the power of 2 to provide for 3000 hosts:
  - i.e. $\mathbf{2^{?} = 3000}$
  - 2^{12} is sufficient (i.e. $2^{11} = 2048$, $2^{12} = 4096$)
  - This requires four bits to be borrowed from the second octet.
  - Leaving **12 bits** for host addresses.

Subnetting a Class B Network

- The new Subnet mask is:

  11111111.1111111.11110000.00000000
  - In DDN: **255.255.240.0**
  - In CIDR notation: **/20**
- Using the *Magic Number* approach to determine the addresses in each subnet.

Subnetting a Class B Network

- With a Mask for each subnet of:

  255.255.240.0

  – Looking from left-to-right along the new mask, seek the last octet that is non-zero <u>and</u> less than or equal to 255.

  – In this case, the third octet matches this criterion.

  – Subtract this octet from 256 as follows:

  256 – 240 = 16 which is *The Magic Number*

- The Address Table can be completed as follows:

Subnetting a Class B Network

	N/W Add	Host Addresses	Broadcast Address	Mask
0	172.25.0.0	172.25.0.1 – 172.25.15.254	172.25.15.255	/20
1	172.25.16.0	172.25.16.1 – 172.25.31.254	172.25.31.255	/20
2	172.25.32.0	172.25.32.1 – 172.25.47.254	172.25.47.255	/20
3	172.25.48.0	172.25.48.1 – 172.25.63.254	172.25.63.255	/20
4	172.25.64.0	172.25.64.1 – 172.25.79.254	172.25.79.255	/20
5	172.25.80.0	172.25.80.1 – 172.25.95.254	172.25.95.255	/20
6	172.25.96.0	172.25.96.1 – 172.25.111.254	172.25.111.255	/20
7	172.25.112.0	172.25.112.1 – 172.25.127.254	172.25.127.255	/20
8	172.25.128.0	172.25.128.1 – 172.25.143.254	172.25.143.255	/20
9	172.25.144.0	172.25.144.1 – 172.25.159.254	172.25.159.255	/20
10	172.25.160.0	172.25.160.1 – 172.25.175.254	172.25.175.255	/20
11	172.25.176.0	172.25.176.1 – 172.25.191.254	172.25.191.255	/20

Basic Subnetting

	N/W Add
0	172.25.0.0
1	172.25.16.0
2	172.25.32.0
3	172.25.48.0
4	172.25.64.0
5	172.25.80.0
6	172.25.96.0
7	172.25.112.0
8	172.25.128.0
9	172.25.144.0
10	172.25.160.0
11	172.25.176.0

- See how the *MN* (16) is used to determine the *network address* for each subnet.
  - Note: the MN was derived in the third octet. Consequently, it is added to this octet in the previous subnet's network address.

Special IP Addresses

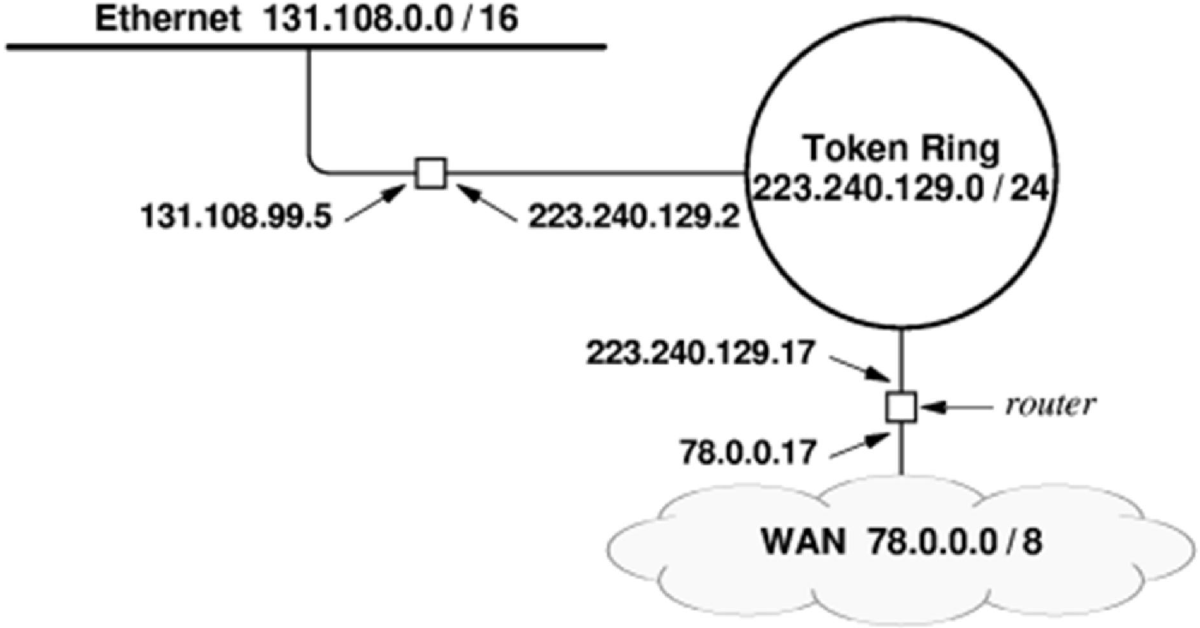
- IP defines a set of special address forms that are reserved and should never be assigned to hosts
- These include:
  - **Directed Broadcast Address**. This is defined for each physical network. A suffix of all 1 bits is added to the network prefix
  - **Limited Broadcast Address**. Here an address consisting of all 1 bits will allow a broadcast on "a single wire"
  - **This Computer Address**. An IP address consisting of all zeros refers to *this computer.* Used by hosts at *boot-up* to obtain its IP address
  - **Loopback Address**. This has a network prefix 127/8; the host suffix is irrelevant but is usually set to 1 i.e. 127.0.0.1

Routers and Multi-Homed Hosts

- Routers <u>and</u> *multi-homed* host computers are assigned two or more IP addresses because:
  - They have connections to multiple physical networks
  - Each IP address prefix specifies <u>only one</u> physical network.
- A fundamental principle of the IP addressing scheme:

"An IP address does not identify a specific computer. Instead, each IP address identifies a connection between a computer and a network. A computer with multiple network connections, e.g. a router, requires one IP address for each connection."

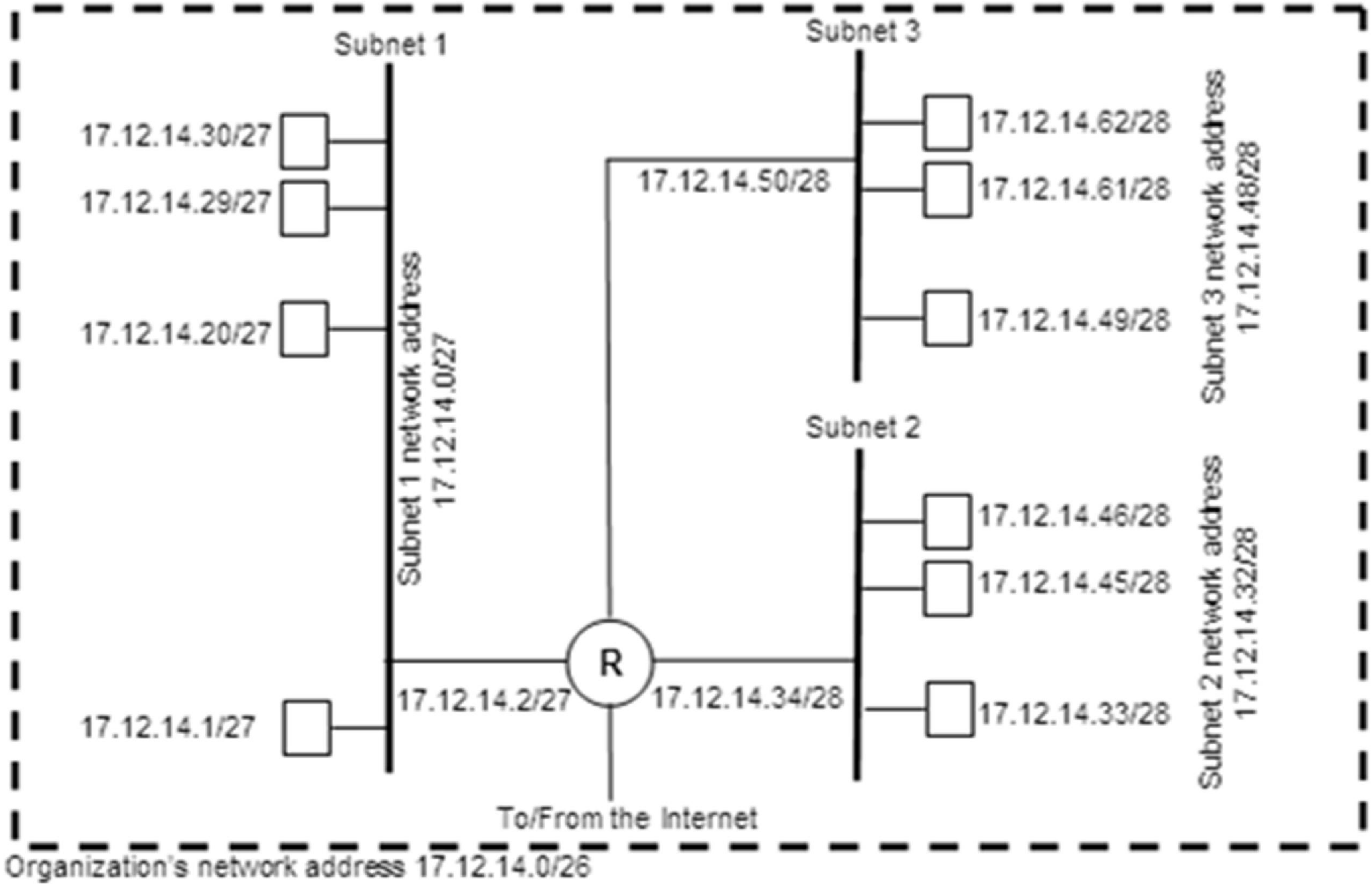
A Router Addressing Example



An example Classless Address Allocation

- An organisation with an address block 17.12.40.0/26 has to divide this block for three separate networks requiring 32, 16 and 16 addresses.
- Firstly the size of the subnet masks for each sub-network needs to be calculated as follows:
  - 32 addresses requires a mask of 27 i.e. $2^{32-27} = 32$
  - 16 addresses requires a mask of 28 i.e. $2^{32-28} = 16$
- An example configuration for this organisation would be:

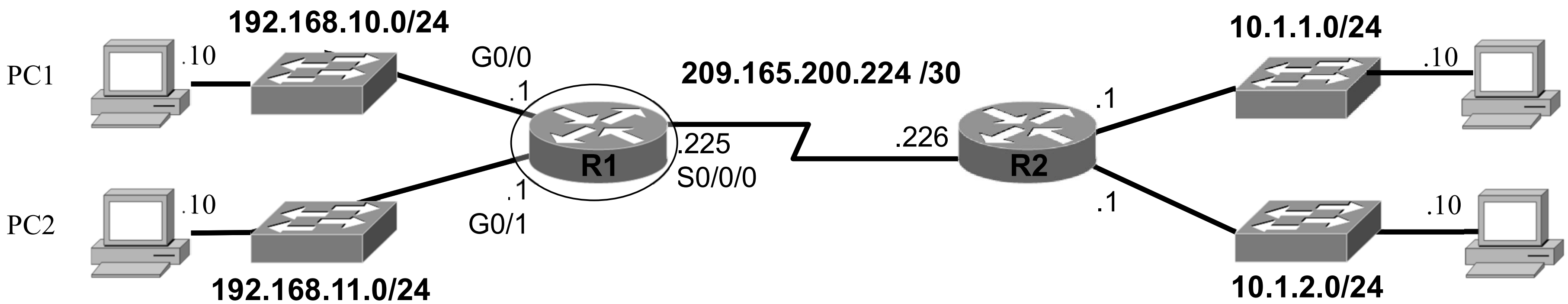
An example Classless Address Allocation



Another example Classless Address Allocation

- An ISP is allocated an address block 190.100.0.0/16.
- There are two groups of customers that need to be allocated addresses as follows:
  – Group 1 - This group consists of 64 customers each requiring 256 addresses.
  – Group 2 - This group consists of 128 customers each requiring 128 addresses.
- To be completed in class.

Example IP Routing Table



- Note the network numbers and the connections to the routers.
- The Routing Table router 1 (R1) is shown on the next slide.

Example IP Routing Table

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP

       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       10.1.1.0/24 [90/2170112] via 209.165.200.226, 00:00:05, Serial0/0/0
D       10.1.2.0/24 [90/2170112] via 209.165.200.226, 00:00:05, Serial0/0/0
     192.168.10.0/24 is variably subnetted, 2 subnets, 3 masks
C       192.168.10.0/24 is directly connected, GigabitEthernet0/0
L       192.168.10.1/32 is directly connected, GigabitEthernet0/0
     192.168.11.0/24 is variably subnetted, 2 subnets, 3 masks
C       192.168.11.0/24 is directly connected, GigabitEthernet0/1
L       192.168.11.1/32 is directly connected, GigabitEthernet0/1
     209.165.200.0/24 is variably subnetted, 2 subnets, 3 masks
C       209.165.200.224/30 is directly connected, Serial0/0/0
L       209.165.200.225/32 is directly connected, Serial0/0/0
R1#
```