CS-542

Computer Networks-1

Homework-1

Fall 2020

Rupa Rajendran-A20460455
Deepak Kumar Yadav-A20447332

1. **What is the range of address of the 100th block of Class A? (2 points)**

<u>Solution:</u> For Class-A addresses the first byte defines the **netid** and the last 3 bytes define the **host-id**.
For Class-A addresses the first byte(netid) has the range 0 - 127.
So,
- Number of blocks in Class A is : (127-0)+1 = **128**
- Number of addresses in Class A is : 2 ^ (32-8) = 2 ^ (24) = **16,777,216**
- First address in the first block of class A is : **0.0.0.0**
- Last address in the last block of class A is : **0.255.255.255**

To get the range of addresses in the 100th block of Class A. We have to find the first address and the last address of the 100th block

<u>- First Address:</u>
To find the first address of the 100th block we just add **100-1 = 99** to the net id part of the first block in class A
Net id of class A is : 0
To get 100 block starting address : 99

99
So, the first address in the 100th block is **99.0.0.0**

<u>- Last Address:</u>
To find the last address of the 100th block we add 255.255.255 to the host id part of the first address in the 100th block
First address in 100th block : 99 . 0. 0. 0
To get last address : 0.255.255.255
----
99. 255. 255. 255

So, the last address in the 100th block is **99.255.255.255**

Range of address of the 100th block in class A is: **99.0.0.0 to 99.255.255.255**

2. **What are the network address, net id, and host id for the IP address 172.168.5.4? (Assume classful addressing) (1.5 points)**

<u>Solution:</u> As the given I.P address is classful addressing we look at the first byte value to determine the class. In our case the first byte value is **172**

For class-B the range of the first byte is **127-191** hence the given address belongs to **Class-B**

For Class-B addresses

- The first two bytes define the **netid**.
- The last two bytes define the **hostid**.
- Default mask is **/16**

To get the **network address** from the IP Address we keep the first 16 bits as it is and make the remaining 16 bits zero.

Network Address: 172.168.0.0

So, the netid and hostid are

Net Id: 172.168
Host Id: 5.4

3. **Convert the number ADBC91E0 in the hexadecimal base to the 256-base system directly.DO NOT convert these numbers into the binary or decimal system (2 points)**

<u>**Solution:**</u>

$(ADBC91E0)_{16}$ = $((A*16^7+D*16^6)) + ((B*16^5+C*16^4)) + ((9*16^3+1*16^2)) + ((E*16^1+0*16^0))$

To convert to base-256 equivalent we can write the above as

$= ((A*16 \quad +D) * 256^3) + ((B*16 \quad +C) * 256^2) + ((9*16 \quad +1) * 256^1) \quad + ((E*16^1+0*16^0) * 256^0)$

$= ((160+13) * 256^3) + ((176+12) * 256^2) + ((144+1) * 256^1) \quad + ((224+0) * 256^0)$

$= (173 * 256^3) + (188 * 256^2) + (145 * 256^1) + (224 * 256^0)$

$= (173.188.145.224)_{256}$

4. **Subtract the number 254.193.47.26 from 192.34.29.255. Give the result in the 256-base system. DO NOT convert these numbers into the binary or decimal, or hexadecimal system. (3 points)**

192.34.29.255
\- **254.193.47.26**

-62.159.17.27

5. **What is the 199th address in the 2^{20} (2 ^ 20) block of Class C? (3 points)**

Solution: In class-C the range of the first byte is 192-223. The net id is defined by the first three bytes and the host id is defined by the last byte.

Total number of blocks available in class C is : ((223-192) + 1) * 256 * 256 = 2,097,152

So, the range of addresses in class-C will be: **192.0.0.X - 223.255.255.X.**

To get the 199th address in the 2^{20} (2 ^ 20) block. We need to first find the first address of the block

Here, (2^ 20) is **1,048,576** which in base-256 notation is $(0.16.0.0)_{256}$

We add this to the net id of the first block to get the starting address on the block

First Address is : Net id of first block + 0.16.0.0

= 192.0.0 + 0.16.0

= 192.16.0

So, the first address in the block is 192.16.0.0

To get the 199th address in the block we add 199-1 = 198 to the first address of the block. 198 in base-256 is $(0.0.0.198)_{256}$

199th Address in the block is = First address of block + 0.0.0198

= 192.16.0.0 + 0.0.0.198

= $(192.16.0.198)_{256}$

6. **Consider the following routing table (the next-hop address is omitted): (3.5 points)**

Mask	**Network Address**	**Interface**
/27	**155.232.55.0**	**M0**

/26	**132.176.97.0**	**M1**
/25	**132.176.97.128**	**M2**
/24	**198.104.162.0**	**M3**
Default	**Default**	**M4**

Give the interface number for a packet whose destination IP address is:

a. 132.176.97.173

We apply the mask available in the routing table to the given destination IP address to find out if this will be routed using that router.

- **/27**

  The mask 27 can be represented as 255.255.255.224 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.

  AND operation on the binary representation of IP address and the mask:

  Address **:** 10000100 10110000 01100001 10101101
  Mask **:** 11111111 11111111 11111111 11100000
  --
  10000100 10110000 01100001 10100000
  = 132.176.97.160

  This doesn't match the network address given in the routing table. So, we check the next available mask.

- **/26**

  The mask 26 can be represented as 255.255.255.192 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.

  AND operation on the binary representation of IP address and the mask:

  Address **:** 10000100 10110000 01100001 10101101
  Mask **:** 11111111 11111111 11111111 11000000

10000100 10110000 01100001 1000000
= 132.176.97.128

This doesn't match the network address given in the routing table. So, we check the next available mask.

- **/25**

  The mask 25 can be represented as 255.255.255.128 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.

  AND operation on the binary representation of IP address and the mask:

  Address **:** 10000100 10110000 01100001 10101101
  Mask **:** 11111111 11111111 11111111 10000000

  10000100 10110000 01100001 1000000
  = 132.176.97.128

  This matches the network address given in the routing table. So, **132.176.97.173** will be routed using the interface **M2**

b. 132.176.97.32

We apply the mask available in the routing table to the given destination IP address to find out if this will be routed using that router.

- **/27**

  The mask 27 can be represented as 255.255.255.224 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.

  AND operation on the binary representation of IP address and the mask:

  Address **:** 10000100 10110000 01100001 00100000
  Mask **:** 11111111 11111111 11111111 11100000

  10000100 10110000 01100001 00100000
  = 132.176.97.32

  This doesn't match the network address given in the routing table. So, we check the next available mask.

- **/26**

  The mask 26 can be represented as 255.255.255.192 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.

AND operation on the binary representation of IP address and the mask:

Address	**:**	10000100 10110000 01100001 00100000
Mask	**:**	11111111 11111111 11111111 11000000

10000100 10110000 01100001 0000000
= 132.176.97.0

This matches the network address given in the routing table. So, **132.176.97.32** will be routed using the interface **M1**

c. 155.232.55.192

We apply the mask available in the routing table to the given destination IP address to find out if this will be routed using that router.

- ***/27***
  The mask 27 can be represented as 255.255.255.224 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
  AND operation on the binary representation of IP address and the mask:

Address	**:**	10011011 11101000 00110111 11000000
Mask	**:**	11111111 11111111 11111111 11100000

10011011 11101000 00110111 11000000
= 155.232.55.192

This doesn't match the network address given in the routing table. So, we check the next available mask.

- ***/26***
  The mask 26 can be represented as 255.255.255.192 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
  AND operation on the binary representation of IP address and the mask:

Address	**:**	10011011 11101000 00110111 11000000
Mask	**:**	11111111 11111111 11111111 11000000

10011011 11101000 00110111 1100000
= 155.232.55.192

This doesn't match the network address given in the routing table. So, we check the next available mask.

- **/25**
  The mask 25 can be represented as 255.255.255.128 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
  AND operation on the binary representation of IP address and the mask:

  Address **:** 10011011 11101000 00110111 11000000
  Mask **:** 11111111 11111111 11111111 10000000

  10011011 11101000 00110111 1000000
  = 155.232.55.128
  This doesn't match the network address given in the routing table. So, we check the next available mask.

- **/24**
  The mask 24 can be represented as 255.255.255.0 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
  AND operation on the binary representation of IP address and the mask:

  Address **:** 10011011 11101000 00110111 11000000
  Mask **:** 11111111 11111111 11111111 00000000

  10011011 11101000 00110111 0000000
  = 155.232.55.0
  This doesn't match the network address given in the routing table. So, we check the next available mask.

  Now there are no more masks available for checking. So, the given packet will be routed using the default entry. Therefore, the packet with destination IP of **155.232.55.192 will be routed using M4**

d. 155.232.55.27

We apply the mask available in the routing table to the given destination IP address to find out if this will be routed using that router.

- **/27**
  The mask 27 can be represented as 255.255.255.224 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
  AND operation on the binary representation of IP address and the mask:
  Address **:** 10011011 11101000 00110111 00011011
  Mask **:** 11111111 11111111 11111111 11100000

10011011 11101000 00110111 00000000
= 155.232.55.0

This matches the network address given in the routing table. So,
155.232.55.27 will be routed using the interface **M0**

e. 198.104.162.128

We apply the mask available in the routing table to the given destination IP address to find out if this will be routed using that router.

- **/27**
  The mask 27 can be represented as 255.255.255.224 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
  AND operation on the binary representation of IP address and the mask:
  Address **:** 11000110 01101000 10100010 10000000
  Mask **:** 11111111 11111111 11111111 11100000

  11000110 01101000 10100010 10000000
  = 198.104.162.128

  This doesn't match the network address given in the routing table. So, we check the next available mask.

- **/26**
  The mask 26 can be represented as 255.255.255.192 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
  AND operation on the binary representation of IP address and the mask:

  Address **:** 11000110 01101000 10100010 10000000
  Mask **:** 11111111 11111111 11111111 11000000

  11000110 01101000 10100010 1000000
  = 198.104.162.128
  This doesn't match the network address given in the routing table. So, we check the next available mask.

- **/25**

The mask 25 can be represented as 255.255.255.128 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
AND operation on the binary representation of IP address and the mask:

Address	**:**	11000110 01101000 10100010 10000000
Mask	**:**	11111111 11111111 11111111 10000000

--

11000110 01101000 10100010 10000000
= 198.104.162.128

This doesn't match the network address given in the routing table. So, we check the next available mask.

- **/24**
  The mask 25 can be represented as 255.255.255.0 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
  AND operation on the binary representation of IP address and the mask:

Address	**:**	11000110 01101000 10100010 10000000
Mask	**:**	11111111 11111111 11111111 00000000

--

11000110 01101000 10100010 00000000
= 198.104.162.0

This matches the network address given in the routing table. So, **198.104.162.0** will be routed using the interface **M3**

f. 198.104.161.192

We apply the mask available in the routing table to the given destination IP address to find out if this will be routed using that router.

- **/27**
  The mask 27 can be represented as 255.255.255.224 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
  AND operation on the binary representation of IP address and the mask:

Address	**:**	11000110 01101000 10100001 11000000
Mask	**:**	11111111 11111111 11111111 11100000

11000110 01101000 10100001 11000000
= 198.104.161.192

This doesn't match the network address given in the routing table. So, we check the next available mask.

- **/26**
The mask 27 can be represented as 255.255.255.192 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
AND operation on the binary representation of IP address and the mask:

Address	**:**	11000110 01101000 10100001 11000000
Mask	**:**	11111111 11111111 11111111 11000000

11000110 01101000 10100001 11000000
= 198.104.161.192

This doesn't match the network address given in the routing table. So, we check the next available mask.

- **/25**
The mask 27 can be represented as 255.255.255.128 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
AND operation on the binary representation of IP address and the mask:

Address	**:**	11000110 01101000 10100001 11000000
Mask	**:**	11111111 11111111 11111111 10000000

11000110 01101000 10100001 10000000
= 198.104.161.128

This doesn't match the network address given in the routing table. So, we check the next available mask.

- **/24**
The mask 27 can be represented as 255.255.255.0 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
AND operation on the binary representation of IP address and the mask:

Address	**:**	11000110 01101000 10100001 11000000
Mask	**:**	11111111 11111111 11111111 00000000

11000110 01101000 10100001 00000000
= 198.104.161.0

This doesn't match the network address given in the routing table. So, we check the next available mask.

Now there are no more masks available for checking. So, the given packet will be routed using the default entry.
Therefore, the packet with destination IP of **198.104.161.192 will be routed using M4**

g. 132.176.97.120

We apply the mask available in the routing table to the given destination IP address to find out if this will be routed using that router.

- **/27**
  The mask 27 can be represented as 255.255.255.224 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
  AND operation on the binary representation of IP address and the mask:
  Address **:** 10000100 10110000 01100001 01111000
  Mask **:** 11111111 11111111 11111111 1110000

  10000100 10110000 01100001 01110000
  = 132.176.97.112

  This doesn't match the network address given in the routing table. So, we check the next available mask.

- **/26**
  The mask 26 can be represented as 255.255.255.192 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
  AND operation on the binary representation of IP address and the mask:
  Address **:** 10000100 10110000 01100001 01111000
  Mask **:** 11111111 11111111 11111111 11000000

  10000100 10110000 01100001 01000000
  = 132.176.97.64

  This doesn't match the network address given in the routing table. So, we check the next available mask.

- **/25**

The mask 25 can be represented as 255.255.255.128 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
AND operation on the binary representation of IP address and the mask:

Address	**:**	10000100 10110000 01100001 01111000
Mask	**:**	11111111 11111111 11111111 10000000

10000100 10110000 01100001 00000000
= 132.176.97.0

This doesn't match the network address given in the routing table. So, we check the next available mask.

- **/24**
  The mask 24 can be represented as 255.255.255.0 .We perform AND operation on the destination address with the mask to check if this router will be used for routing this packet.
  AND operation on the binary representation of IP address and the mask:

Address	**:**	10000100 10110000 01100001 01111000
Mask	**:**	11111111 11111111 11111111 00000000

10000100 10110000 01100001 00000000
= 132.176.97.0

This doesn't match the network address given in the routing table. So, we check the next available mask.

Now there are no more masks available for checking. So, the given packet will be routed using the default entry.
Therefore, the packet with destination IP of **132.176.97.120 will be routed using M4**

7. **Give the mask in the dotted-decimal notation: (4 points)**
   a. **For a block of Class A which results in 32 subnets**
   b. **Which combines 2048 Class C blocks into a supernet**
   c. **For a block of Class B which results in 64 subnets**
   d. **Which combines 16 Class B blocks into a supernet**

a.) For a block of Class A which results in 32 subnets

For class-A the default mask is /8. In order to have 32 subnets the extra 1 bits we need is calculated using the below formulae.

=> $log_2 32 = log_2 2^5$ = 5

So, the mask to have 32 subnets in class-A is 8+5 = 13 or 255.248.0.0

b.) Which combines 2048 Class C blocks into a supernet

For class-C the default is mask is /24. In order to combine 2048 blocks into a supernet 11 bits which is the complete 3rd byte and 3 bits in the 2nd byte of the default mask are changed from 1 to 0. As 2^11 = 2048, hence 11 bits are reduced from the default mask.

Default Mask **:** 255.255.255.0
Mask(Binary rep.) : 11111111. 11111111. 11111111. 00000000
After modifying the mask we get
ModifiedMask(Binary rep.):
11111111. 11111000. 00000000. 00000000

So, the mask to have 2048 blocks in a supernet is 13 or 255.248.0.0

c.) For a block of Class B which results in 64 subnets

For class-B the default mask is /16. In order to have 64 subnets the extra 1 bits we need is calculated using the below formulae.

=> $log_2 64 = log_2 2^6$ = 6

So, the mask to have 32 subnets in class-A is 16+6 = 22 or 255.255.252.0

d.) Which combines 16 Class B blocks into a supernet

For class-B the default is mask is /16. In order to combine 16 blocks into a supernet 4 bits which are the last 4 bits in the 2nd byte of the default mask are converted from 1 to 0. As 2^4 = 16, hence 4 bits are reduced from the default mask.

Default Mask **:** 255.255.0.0
Mask(Binary rep.) : 11111111. 11111111. 00000000. 00000000
After modifying the mask we get
ModifiedMask(Binary rep.):
11111111. 11110000. 00000000. 00000000

So, the mask to have 32 blocks in a supernet is 12 or 255.240.0.0

8. **Find the network address, the direct broadcast address, and the number of addresses in a block; if one of the addresses in this block is 175.120.240.17/19 (3 points)**

<u>Solution:</u> To find the network address from an address in the block we need to perform AND operation on the given address and the specified mask.

<u>Binary representations are:</u>
Address **:** 10101111 01111000 11110000 00010001
Mask **:** 11111111 11111111 11100000 00000000

: 10101111 01111000 11100000 00000000

Network Address is : 175.120.240.0 / 19

Direct broadcast address, is the last address in the block. In order to find this we need to find the total number of addresses available in this block and use this with the network address to find the last address in the block.(Ans is below)

Total number of addresses available in a block can be calculated using the below formulae.

$2^{32-Mask} = 2^{32-19} = 2^{13} =$ **8,192 addresses**.

The last address in the block can be found by adding 8192-1 = 8191 to the network address of the block

=> 8191 in base 256 = (31* 256^1) + (255 * 256^0)

So, Last address is = 175.120.240.0/19 + 0.0.31.255. Below is the sum operation in binary

175.120.240.0 = 10101111 01111000 11110000 00000000
0.0.31.255 = 00000000 00000000 00011111 11111111

= 10101111 01111001 00001111 11111111

Last address is : 175.121.31.255/19

9. **Convert the decimal number 1819111023.439246416091794 to the base 256 system. Show all the intermediate steps. Don't miss a decimal point in this number. (5 points)**

<u>Solution:</u> The decimal number **1819111023.439246416091794** can be converted to base 256 in the following manner

We first convert **1819111023** by dividing with 256 until it terminates

256	1819111023	
256	7105902	111
256	27757	110
256	108	109
256	0	108

So, we can write $(1819111023)_{10}$= $(108.109.110.111,)_{256}$

Now, we work on the other decimal part i.e., 0.439246416091794

0.439246416091794 * 256	112.44708251949926137	112
0.44708251949926137 * 256	114.4531249918109097	114
0.4531249918109097 * 256	115.9999979035928845	115
0.9999979035928845 * 256	255.9994633197784424	255
0.9994633197784424 * 256	255.86260986328125	255
0.86260986328125 * 256	220.82812500128	220
0.82812500128 * 256	212.0000000032768	212
0.0000000032768 * 256	0.00000008388608	0

So, we can write $(0.439246416091794)_{10}$= (,112.114.115.255.255.220.212.0$)_{256}$

1819111023.439246416091794 in base-256 is

(108.109.110.111,112.114.115.255.255.220.212.0$)_{256}$

10. **The 49th address of a block assigned to a specific organization is 185.175.79.48. The organization needs a total of 128 addresses. Find the mask and define this block of addresses. Is there any wastage of the IP addresses? If yes, how many? (4 points)**

Solution: The organization needs 128 addresses. Address are allocated in power of 2 so the number nearest to 128 which is a power of 2 is 128 itself so the suffix length will be 7 as $log_2 128$ = 7 and prefix will be 25 as 32-7 = 25.

The **subnet mask for the organization will be /25 which is 255.255.255.128.**

We are provided with the 49th address in the block: 185.175.79.48

To get the first address of the block we will subtract 49-1 = 48 from the given 49th address. **48** in base-256 notation is 0.0.0.48

First address is **=** 49th address - 0.0.0.48
= 185.175.79.48 - 0.0.0.48
= $(185.175.79.0/25)_{256}$

To get the last address we add 128-1 = 127 to the first address in the block. **127** in base 256 notation is 0.0.0.127

Last address is **=** 1st address + 0.0.0.127
= 185.175.79.0 + 0.0.0.127
= $(185.175.79.127/25)_{256}$

Here, the total number of addresses allocated is 128 and the number of addresses requested by the organization is 128. **So, the number of addresses wasted in 0.**

11. An organization is granted the block 178.49.240.0/20. The administrator wants to create 128 subnets. Find the following. (4 points)

a. The subnet mask (give value in /n notation)
b. The number of addresses in each subnet
c. Subnet address of the 100th subnet.

A. The subnet mask (give value in /n notation)

The block given to us is 178.49.240.0/20. The administrator wants to create 128 subnets. So, the number of extra bits that needs to be added to the default mask is

=> $log_2 128$ = $log_2 2^7$ = 7

Subnet mask will be 20 + 7 = **/27 or 255.255.255.224**

B. The number of addresses in each subnet

The number of addresses in each subnet is given by

=> $2^{32-mask}$ = 2^{32-27} = 2^5= **32 address per subnet**

C. Subnet address of the 100th subnet.

Each subnet has 32 addresses, so to get the subnet address of the 100th subnet we need to add 32 * 99 (this represent the address in the 99 subnet which are present before this subnet) to the first block

=> 32 * 99 = 3168_{10} = $(12.96)_{256}$ = $(0.0.12.96)_{256}$
=> **Subnet address of the 100th subnet is** = first block address + 0.0.12.96
= 178.49.240.0 + 0.0.12.96
= **178.49.252.96/20**

12. An Internet Service Provider (ISP) has the following block of IP address192.37.128.0/17. The ISP gave the first 1024 addresses to Organization A and the next available subblock of 16384 addresses to Organization B and retained the remaining IP addresses. Give the subblocks and the valid range of addresses allocated to organizations A & B and the range of remaining addresses. (4 points)

Solution: Given block of IP addresses is 192.37.128.0/17
The mask given here is **17** that means total number of addresses available for allocation is:
2^{32-17}= 2^{15}= **32,768 addresses**

- **Organization-A address allocation:**

Organization-A has been assigned with the first 1024 addresses. So, the subnet mask for organization-A is **22** as
=> $2^{32-x} = 2^{10}$
=> 32-x = 10
=> x = 22
First Address in Organization-A is : **192.37.128.0/22**
To get the last address in organization-A we need to add 1024-1= 1023 to the first address.
1023 in base-256 representation is $(3.255)_{256}$= $(0.0.3.255)_{256}$
Last Address in Organization-A is: First address in block + 0.0.3.255
= 192.37.128.0 + 0.0.3.255
= **192.37.131.255/22**

- **Organization-B address allocation:**
Organization-B has been assigned with the next 16384 addresses. So, the subnet mask for organization-A is **18** as
=> $2^{32-x} = 2^{14}$
=> 32-x = 14
=> x = 18
To get the first address in organization-B we need to add 1024 to the first address of organization-A or we can add 1 to the last address in the block of organization-A. Here we add 1024 to the first address in org-A first address
1024 in base 256 is $(4.0)_{256}$ = $(0.0.4.0)_{256}$

First Address in Organization-B is : first address in org-A + 0.0.4.0
= 192.37.128.0 + 0.0.4.0
= **192.37.132.0/18**

To get the last address in organization-B we need to add 16384-1= 16383 to the first address.
16383 in base-256 representation is $(63.255)_{256}$=$(0.0.63.255)_{256}$
Last Address in Organization-B is: First address in block + 0.0.63.255
= 192.37.132.0 + 0.0.63.255
= **192.37.195.255/18**

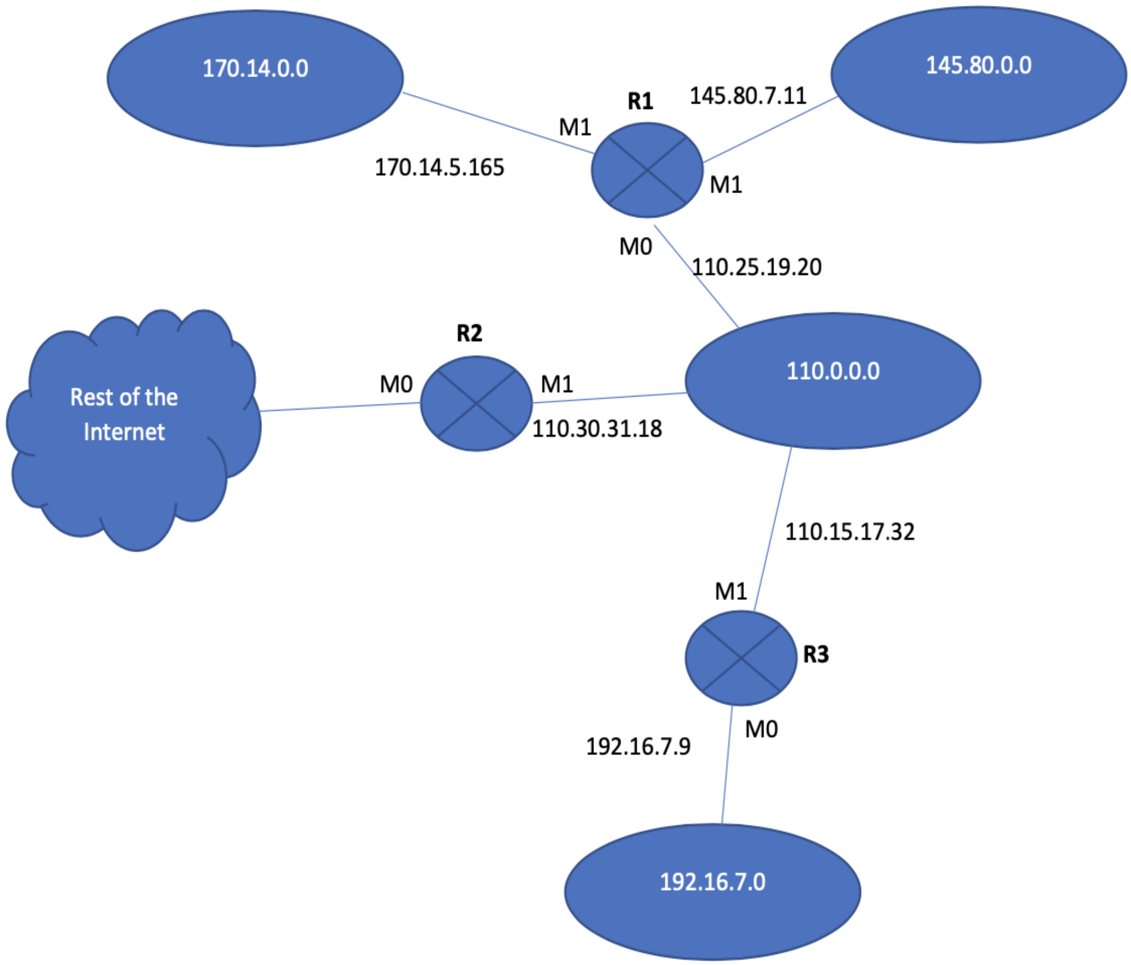
- **ISP address available:**
  After, the address allocation to organization-A and organization-B the range of addresses remaining with ISP is calculated as follows
  Total address available with ISP after address allocation is
  = (Total address before allocation) - (OrgA address count) - (OrgB address count)
  = 32768 - 1024 - 16384
  = 15,360

  As, the address allocation can only be a power of 2. So, the subnet mask for the remaining addresses is **/14.**

  To get the first address for the remaining addresses add 1 to the last block of organization-B address range.
  1 in base-256 representation is $(0.1)_{256}$= $(0.0.0.1)_{256}$
  First Address in remaining address is : Last address in org-B + 0.0.0.1
  = 192.37.195.255 + 0.0.0.1
  = **192.37.196.0/14**

  To get the last address for the remaining addresses we need to add 15360-1=15359 to the first address
  15359 in base-256 representation is $(59.255)_{256}$= $(0.0.59.255)_{256}$
  Last Address in remaining address is : First address in block + 0.0.59.255
  = 192.37.196.0 + 0.0.59.255
  = **192.37.255.255/14**

13. Consider the network configuration below. A packet arrived at the router R3 with the destination address 170.14.7.47. Show how it is forwarded. (Assume classful addressing). Create a routing table for router R3. (6 points)



<u>Solution:</u> The following is the routing table for Router R3

	CLASS A	
Network Address	**Next Hop Address**	**Interface**
110.0.0.0	---------------------	m1

	CLASS B	
Network Address	**Next Hop Address**	**Interface**
145.80.0.0	110.25.19.20	m1
170.14.0.0	110.25.19.20	m1

	CLASS C	
Network Address	**Next Hop Address**	**Interface**
192.16.7.0	-----------------------------	m0

Network Address	**Next Hop Address**	**Interface**
Default	110.30.31.18	m1

The destination address of the packet that arrives in router 3 is 170.14.7.47. So, first we have to convert the destination address into binary and that will be 10101010 00001110 00000111 00111111.
A copy of the address is shifted 28 bits to the right. The result is 00000000 00000000 00000000 00001010 or 10. **The class is B.**
The network address can be found by masking off 16 bits of the destination address, the result is 170.14.0.0. The table for class B is searched. The network address is found in the second row. The next hop address is **110.25.19.20** and the interface **m1** is passed to the ARP.

(Note: In the diagram above, the interface for 170.14.0.0 is given as M1, so we have taken that only into consideration.)