CHAPTER 19

Network Layer: Logical Addressing

Solutions to Review Questions and Exercises

Review Questions

1. An ***IPv4*** address is **32** bits long. An ***IPv6*** address is **128** bits long.
2. ***IPv4 addresses*** are usually written in decimal form with a decimal point (dot) separating the bytes. This is called ***dotted-decimal notation***. Each address is **4** bytes. ***IPv6*** addresses are usually written in hexadecimal form with a colon separating the bytes. This is called ***hexadecimal notation***. Each address is **16** bytes or **32** hexadecimal digits.
3. ***Classful addressing*** assigns an organization a Class A, Class B, or Class C block of addresses. ***Classless addressing*** assigns an organization a block of contiguous addresses based on its needs.
4. ***Classes A***, ***B***, and ***C*** are used for **unicast** communication. ***Class D*** is for **multicast** communication and ***Class E*** addresses are **reserved** for special purposes.
5. A ***block in class A*** address is ***too large*** for almost any organization. This means most of the addresses in class A are wasted and not used. ***A block in class C*** is probably ***too small*** for many organizations.
6. A ***mask*** in classful addressing is used to find the first address in the block when one of the addresses is given. The ***default mask*** refers to the mask when there is no subnetting or supernetting.
7. The ***network address*** in a block of addresses is the first address. The ***mask*** can be **ANDed** with any address in the block to find the network address.
8. In ***subnetting***, a large address block could be divide into several contiguous groups and each group be assigned to smaller networks called subnets. In ***supernetting***, several small address blocks can be combined to create a larger range of addresses. The new set of addresses can be assigned to a large network called a supernet. ***A subnet mask*** has ***more*** consecutive 1s than the corresponding default mask. ***A supernet mask*** has ***less*** consecutive 1s than the corresponding default mask.
9. Multicast addresses in ***IPv4*** are those that start with the **1110** pattern. Multicast addresses in ***IPv6*** are those that start with the **11111111** pattern.

10. Home users and small businesses may have created small networks with several hosts and need an IP address for each host. With the shortage of addresses, this is a serious problem. A quick solution to this problem is called ***network address translation (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.

Exercises

11.
    a. $2^8 = \mathbf{256}$
    b. $2^{16} = \mathbf{65536}$
    c. $2^{64} = \mathbf{1.846744737 \times 10^{19}}$
12. $2^x = 1024 \rightarrow x = \log_2 1024 = \mathbf{10}$
13. $3^{10} = \mathbf{59{,}049}$
14.

a.	01110010	00100010	00000010	00001000
b.	10000001	00001110	00000110	00001000
c.	11010000	00100010	00110110	00001100
d.	11101110	00100010	00000010	00000001

15.
    a. **127.240.103.125**
    b. **175.192.240.29**
    c. **223.176.31.93**
    d. **239.247.199.29**
16.
    a. **Class C** (first byte is between 192 and 223)
    b. **Class D** (first byte is between 224 and 239)
    c. **Class A** (first byte is between 0 and 127)
    d. **Class B** (first byte is between 128 and 191)
17.
    a. **Class E** (first four bits are 1s)
    b. **Class B** (first bit is 1 and second bit is 0)
    c. **Class C** (first two bits are 1s and the third bit is 0)
    d. **Class D** (first three bits are 1s and the fourth bit is 0)
18.

a.	netid: 114	**hostid: 34.2.8**
b.	netid: 132.56	**hostid: 8.6**
c.	netid: 208.34.54	**hostid: 12**

19. With the information given, the first address is found by ANDing the host address with the mask 255.255.0.0 (/16).

Host Address:	**25**	.	**34**	.	**12**	.	**56**
Mask (ANDed):	**255**	.	**255**	.	**0**	.	**0**
Network Address (First):	**25**	.	**34**	.	**0**	.	**0**

The last address can be found by ORing the host address with the mask complement 0.0.255.255.

Host Address:	**25**	.	**34**	.	**12**	.	**56**
Mask Complement (ORed):	**0**	.	**0**	.	**255**	.	**255**
Last Address:	**25**	.	**34**	.	**255**	.	**255**

However, we need to mention that this is the largest possible block with 2^{16} addresses. We can have many small blocks as long as the number of addresses divides this number.

20. With the information given, the first address is found by ANDing the host address with the mask 255.255.255.192 (/26).

Host Address:	**182**	.	**44**	.	**82**	.	**16**
Mask (ANDed):	**255**	.	**255**	.	**255**	.	**192**
Network Address (First):	**182**	.	**44**	.	**82**	.	**0**

The last address can be found by ORing the host address with the mask complement 0.0.0.63.

Host Address:	**182**	.	**44**	.	**82**	.	**16**
Mask Complement (ORed):	**0**	.	**0**	.	**0**	.	**63**
Last Address:	**182**	.	**44**	.	**82**	.	**63**

However, we need to mention that this is the largest possible block with 2^6 addresses. We can have several small blocks as long as the number of addresses divides this number.

21.

a. $\log_2 500 = 8.95$ Extra 1s = 9 Possible subnets: **512** Mask: **/17** (8+9)

b. $2^{32-17} = 2^{15} =$ **32,768** Addresses per subnet

c. **Subnet 1:** The first address in the this address is the beginning address of the block or **16.0.0.0**. To find the last address, we need to write 32,767 (one less than the number of addresses in each subnet) in base 256 (0.0.127.255) and add it to the first address (in base 256).

First address in subnet 1:	**16**	.	**0**	.	**0**	.	**0**
Number of addresses:	**0**	.	**0**	.	**127**	.	**255**
Last address in subnet 1:	**16**	.	**0**	.	**127**	.	**255**

d. **Subnet 500:**
Note that the subnet 500 is not the last possible subnet; it is the last subnet used by the organization. To find the first address in subnet 500, we need to add 16,351,232 (499 × 32678) in base 256 (0. 249.128.0) to the first address in subnet 1. We have 16.0.0.0 + 0.249.128.0 = **16.249.128.0**. Now we can calculate the last address in subnet 500.

First address in subnet 500:	**16**	.	**249**	.	**128**	.	**0**
Number of addresses:	**0**	.	**0**	.	**127**	.	**255**
Last address in subnet 500:	**16**	.	**249**	.	**255**	.	**255**

22.

a. $\log_2 1024 = 10$ Extra 1s = **10** Possible subnets: **1024** Mask: **/26**
b. 2^{32-26} = **64** Addresses per subnet
c. **Subnet 1:**
The first address is the beginning address of the block or **130.56.0.0.** To find the last address, we need to write 63 (one less than the number of addresses in each subnet) in base 256 (0.0.0.63) and add it to the first address (in base 256).

First address in subnet 1:	**130**	.	**56**	.	**0**	.	**0**
Number of addresses:	**0**	.	**0**	.	**0**	.	**63**
Last address in subnet 1:	**130**	.	**56**	.	**0**	.	**63**

d. **Subnet 1024:**
To find the first address in subnet 1024, we need to add 65,472 (1023 × 64) in base 256 (0.0.255.92) to the first address in subnet 1. We have 130.56.0.0. + 0.0.255.192 = **130.56.255.192**. Now we can calculate the last address in subnet 500 as we did for the first address.

First address in subnet 1024:	**130**	.	**56**	.	**255**	.	**192**
Number of addresses:	**0**	.	**0**	.	**0**	.	**63**
Last address in subnet 1024:	**130**	.	**56**	.	**255**	.	**255**

23.

a. $\log_2 32 = 5$ Extra 1s = **5** Possible subnets: **32** Mask: **/29** (24 + 5)
b. 2^{32-29} = **8** Addresses per subnet
c. **Subnet 1:**
The first address is the beginning address of the block or **211.17.180.0**. To find the last address, we need to write 7 (one less than the number of addresses in each subnet) in base 256 (0.0.0.7) and add it to the first address (in base 256).

First address in subnet 1:	**211**	.	**17**	.	**180**	.	**0**
Number of addresses:	**0**	.	**0**	.	**0**	.	**7**
Last address in subnet 1:	**211**	.	**17**	.	**180**	.	**7**

d. **Subnet 32:**
To find the first address in subnet 32, we need to add 248 (31 × 8) in base 256 (0.0.0.248) to the first address in subnet 1. We have 211.17.180.0 + 0.0.0.248 or **211.17.180.248.** Now we can calculate the last address in subnet 32 as we did for the first address.

First address in subnet 32:	**211**	.	**17**	.	**180**	.	**248**
Number of addresses:	**0**	.	**0**	.	**0**	.	**7**
Last address in subnet 32:	**211**	.	**17**	.	**180**	.	**255**

24.

a. The mask 255.255.255.0 has **24** consecutive 1s → slash notation: /24
b. The mask 255.0.0.0 has **8** consecutive 1s → slash notation:**/8**
c. The mask 255.255.224.0 has **19** consecutive 1s → slash notation:**/19**
d. The mask 255.255.240.0 has **20** consecutive 1s → slash notation:**/20**

25.

a. The number of address in this block is 2^{32-29} = **8**. We need to add 7 (one less) addresses (0.0.0.7 in base 256) to the first address to find the last address.

From:	**123**	.	**56**	.	**77**	.	**32**
	0	.	**0**	.	**0**	.	**7**
To:	**123**	.	**56**	.	**77**	.	**39**

b. The number of address in this block is 2^{32-27} = **32**. We need to add 31 (one less) addresses (0.0.0.31 in base 256) to the first address to find the last address.

From:	**200**	.	**17**	.	**21**	.	**128**
	0	.	**0**	.	**0**	.	**31**
To:	**200**	.	**17**	.	**21**	.	**159**

c. The number of address in this block is 2^{32-23} = 512. We need to add 511 (one less) addresses (0.0.1.255 in base 256) to the first address to find the last address.

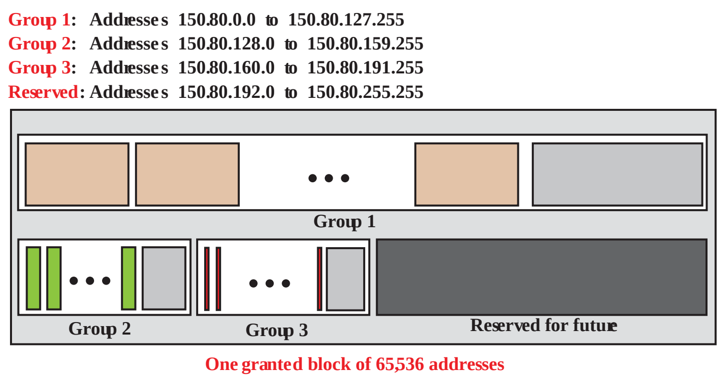
From:	**17**	.	**34**	.	**16**	.	**0**
	0	.	**0**	.	**1**	.	**255**
To:	**17**	.	**34**	.	**17**	.	**255**

d. The number of address in this block is 2^{32-30} = 4. We need to add 3 (one less) addresses (0.0.0.3 in base 256) to the first address to find the last address.

From:	180	.	34	.	64	.	64
	0	.	**0**	.	**0**	.	**3**
To:	180	.	34	.	64	.	67

26. The total number of addresses in this block is 2^{32-16} = **65536**. The ISP can divide this large block in several ways depending on the predicted needs of its customers in the future. We assume that the future needs follow the present pattern. In other words, we assume that the ISP will have customers that belong to one of the present groups. We design four ranges: group 1, group 2, group 3, and one reserved range of addresses as shown in Figure 19.1.

Figure 19.1 *Solution to Exercise 26*



Group 1

In the first group, we have 200 businesses. We augment this number to **256** (the next number after 200 that is a power of 2) to let 56 more customers of this kind in the future. The total number of addresses is 256 × 128 = **32768**. For this group, each customer needs 128 addresses. This means the suffix length is $\log_2 128 = 7$. The prefix length is then 32 – 7 = 25. The addresses are:

1st customer:	**150.80.0.0/25**	**to**	**150.80.0.127/25**
2nd customer:	**150.80.0.128/25**	**to**	**150.80.0.255/25**
...	**...**		**...**
200th customer:	**150.80.99.128/25**	**to**	**150.80.99.255/25**
Unused addresses	**150.80.100.0**	**to**	**150.80.127.255**

Total Addresses in group 1 = 256 × 128 = **32768** Used = 200 × 128 = **25600.**
Reserved: **7168**, which can be assigned to 56 businesses of this size.

Group 2

In the second group, we have 400 business. We augment this number to 512 (the next number after 400 that is a power of 2) to let 112 more customer of this kind in the future. The total number of addresses is = 512 × 16 = **8192**. For this group, each customer needs 16 addresses. This means the suffix length is 4 $\log_2 16 = 4$.

The prefix length is then 32 – 4 = 28. The addresses are:

1st customer:	**150.80.128.0/28**	**to**	**150.80.128.15/28**
2nd customer:	**150.80.128.16/28**	**to**	**150.80.128.31/28**
...	**...**		**...**
400th customer:	**150.80.152.240/28**	**to**	**150.80.152.255/28**
Unused addresses	**150.80.153.0**	**to**	**150.80.159.255**

Total Addresses in group 2 = 512 × 16 = **8192** Used = 400 × 16 = **6400**
Reserved: **1792**, which can be assigned to 112 businesses of this size.

Group 3

In the third group, we have 2000 households. We augment this number to 2048 (the next number after 2000 that is a power of 2) to let 48 more customer of this kind in the future. The total number of addresses is = 2048 × 4 = **8192**. For this group, each customer needs 4 addresses. This means the suffix length is 2 $\log_2 4$ = 2. The prefix length is then 32 – 2 = 30. The addresses are:

1st customer:	**150.80.160.0/30**	**to**	**150.80.160.3/30**
2nd customer:	**150.80.160.4/30**	**to**	**150.80.160.7/30**
...	**...**		**...**
2000th customer:	**150.80.191.60/30**	**to**	**150.80.191.63/30**
Unused addresses	**150.80.191.64**	**to**	**150.80.191.255**

Total Addresses in group 3 = 2048 × 4 = **8192** Used = 2000 × 4 = **8000**
Reserved: **192**, which can be assigned to 48 households.

Reserved Range

In the reserved range, we have 16384 address that are totally unused.

Note that we have unused addresses in each group and a large range of unused addresses in the reserved range.

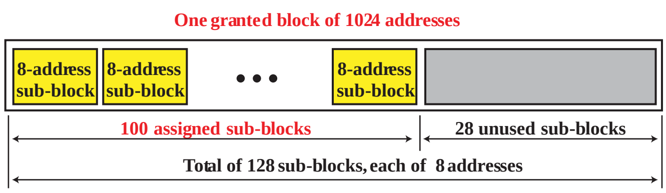
Summary:

The following shows the summary of used and unused addresses:

Group Number	*Total Addresses*	*Used Addresses*	*Unused Addresses*
1	32,768	**25,600**	7168
1	8192	**6400**	1792
1	8192	**8000**	192
Reserved	16,384	**0**	16384
Sum	**65,536**	**40,000**	**25536**

27. The site has $2^{32-22} = 2^{10} =$ **1024** from **120.60.4.0/22** to **120.60.7.255/22** addresses. One solution would be to divide this block into **128** 8-address sub-blocks as shown in Figure 19.2. The ISP can assign the first 100 sub-blocks to the current customers and keep the remaining 28 sub-blocks. Of course, this does not mean the future customer have to use 8-address subblocks. The remaining addresses can later be divided into different-size sub-blocks (as long as the three restrictions mentioned in this chapter are followed). Each sub-block has 8 addresses. The mask for each sub-block is **/29** ($32 - \log_2 8$). Note that the mask has changed from /22 (for the whole block) to /29 for each subblock because we have 128 sub-blocks ($2^7 = 128$).

Figure 19.2 *Solution to Exercise 27*



Sub-blocks:

1st subnet:	**120.60.4.0/29**	to	**120.60.4.7/29**
2nd subnet:	**120.60.4.8/29**	to	**120.60.4.15/29**
...	**...**		**...**
32nd subnet:	**120.60.4.248/29**	to	**120.60.4.255/29**
33rd subnet:	**120.60.5.0/29**	to	**120.60.5.7/29**
...	**...**		**...**
64th subnet:	**120.60.5.248/29**	to	**120.60.5.255/29**
...	**...**		**...**
99th subnet:	**120.60.7.16/29**	to	**120.60.7.23/29**
100th subnet:	**120.60.7.24/29**	to	**120.60.7.31/29**

1024 – 800 = **224** addresses left (from **120.60.7.31** to 120.60.7.155)

28. Each customer has only 1 address and, therefore, only one device. Since we defined a network as 2 or more connected devices, this is not a network

29.

a. **2340:1ABC:119A:A000::0**

b. **0:AA::119A:A231**

c. **2340::119A:A001:0**

d. **0:0:0:2340::0**

30.
   a. 0000:0000:0000:0000:0000:0000:0000:0000
   b. 0000:00AA:0000:0000:0000:0000:0000:0000
   c. 0000:1234:0000:0000:0000:0000:0000:0003
   d. 0123:0000:0000:0000:0000:0000:0001:0002
31.
   a. ***Link local address***
   b. ***Site local address***
   c. ***Multicast address*** (permanent, link local)
   d. ***Loopback address***
32.
   a. ***Unspecified address***
   b. ***Mapped address***
   c. ***Provider based address*** with the address registered through ***INTERNIC*** (North American registry).
   d. ***Provider based address*** with the address registered through ***RIPNIC*** (European registry).
   e. ***Provider based address*** with the address registered through ***APNIC*** (Asian/Pacific registry).
33. **58ABC1**
34.
   a. **0000:0000:0000:0000:0000:0000:8106:0C22** or **0::8106:C22**
   b. **0000:0000:0000:0000:0000:FFFF:8106:0C22** or **0::FFFF:8106:C22**
35.
   a. **FE80:0000:0000:0000:0000:0000:0000:0123** or **FE80::123**
   b. **FEC0:0000:0000:0000:0000:0000:0000:0123** or **FEC0::123**
36. **FF02**: **< Group ID >**
37. The node identifier is **0000:0000:1211**. Assuming a 32-bit subnet identifier, the subnet address is **581E:1456:2314:ABCD:0000** where **ABCD:0000** is the subnet identifier.
38.
   from: **581E:1456:2314:0000:ABCD:0000:0001:XXXX**
   to: **581E:1456:2314:0000:ABCD:0000:00C8:XXXX**
   where **XXXX** is the node identifier.