Solution of Chapter 18

Q18-1. Why does the network-layer protocol need to provide packetizing service to the transport layer? Why can't the transport layer send out the segments without encapsulating them in datagrams?

- **Packetizing:** The network layer encapsulates transport-layer segments into **datagrams** (packets).
- **Reason:**
  - The **transport layer** only ensures process-to-process delivery, but it is not aware of the physical or logical paths that the packets will take.
  - The **network layer** adds source and destination IP addresses to ensure **end-to-end communication** between devices.
  - Without encapsulating in datagrams, there's no way for the routers and intermediate devices to route the data correctly.

Q18-2. Why is routing the responsibility of the network layer? Why can't the routing be done at the transport layer or the data-link layer?

- **Routing at Network Layer:**
  - The **network layer** operates on IP addresses, which are hierarchical and global.
  - It determines the best path for data to travel from the source to the destination using routing protocols.
- **Why Not Transport Layer?**
  - The transport layer deals with **process-to-process** communication, not **host-to-host** or network routing.
  - It cannot decide on physical paths between routers.
- **Why Not Data-Link Layer?**
  - The data-link layer only handles communication between **adjacent nodes** on the same local network (MAC addresses).
  - It has no knowledge of multiple hops across different networks.

Q18-3. Distinguish between the process of routing a packet from the source to the destination and the process of forwarding a packet at each router.

- **Routing:**
  - A **global process** that determines the **path** a packet should take to reach its destination.
  - Performed using routing algorithms and protocols like OSPF, RIP, or BGP.
  - Happens **once per session**.

- **Forwarding:**
  - A **local process** performed at each router to send the packet to the next-hop router or destination.
  - Based on the routing table and IP address lookup.
  - Happens **per packet**.

Q18-4. What is the piece of information in a packet upon which the forwarding decision is made in each of the following approaches to switching?

- **a. Datagram Approach:**
  - Forwarding is based on the **destination IP address** in each packet.
  - Each packet is treated independently.
- **b. Virtual-Circuit Approach:**
  - Forwarding is based on the **VCI (Virtual Circuit Identifier)** or **Label**.
  - A pre-established path is used for all packets in the same connection.

Q18-5. If a label in a connection-oriented service is 8 bits, how many virtual circuits can be established at the same time?

- **Number of Virtual Circuits:**
  - A label with **8 bits** can represent $\mathbf{2^8 = 256}$ different virtual circuits.

Q18-6. List the three phases in the virtual-circuit approach to switching.

1. **Connection Setup:**
   - Establish a path and allocate resources.
2. **Data Transfer:**
   - Send packets along the established path.
3. **Connection Teardown:**
   - Release resources after communication is complete.

Q18-7. Do we have any of the following services at the network layer of TCP/IP? If not, why?

- **a. Flow Control:** No
  - Flow control is handled at the **transport layer** (TCP), not at the network layer.
- **b. Error Control:** No
  - The network layer does not perform error correction but only detects errors through IP header checksums.

    - Error control is managed by the **transport layer**.
- **c. Congestion Control:** No
    - Congestion control is performed at the **transport layer** (TCP), not at the network layer.

Q18-8. List four types of delays in a packet-switched network.

1. **Propagation Delay:** Time for a signal to travel through the medium.
2. **Transmission Delay:** Time to push all bits into the link.
3. **Queuing Delay:** Time spent waiting in queues at routers.
4. **Processing Delay:** Time taken by routers to process and forward the packet.

Q18-9. In Figure 18.10, assume that the link between R1 and R2 is upgraded to 170 kbps and the link between the source host and R1 is now downgraded to 140 kbps. What is the throughput between the source and destination after these changes? Which link is the bottleneck now?

- **Throughput:**
    - Throughput is determined by the **slowest link**.
    - Since the link between **source and R1** is downgraded to **140 kbps**, that becomes the **bottleneck**.
- **New Throughput:** 140 kbps.

Q18-10. In classless addressing, we know the first and the last address in the block. Can we find the prefix length? If the answer is yes, show the process.

- **Yes, the prefix length can be determined.**
        - **Formula:** Number of addresses=last address−first address+1
    - Convert the number of addresses to binary and calculate **$\log_2$(number of addresses)**.
    - Subtract the result from **32** to get the **prefix length**.

Q18-11. In classless addressing, we know the first address and the number of addresses in the block. Can we find the prefix length? If the answer is yes, show the process.

- **Yes, the prefix length can be calculated.**
- **Process:**
    - Convert the **number of addresses** to binary.
    - Calculate **$\log_2$(number of addresses)** to find the number of host bits.
    - Subtract the host bits from **32** to get the prefix length.

$$\text{Prefix length} = 32 - \log_2(\text{number of addresses})$$

Q18-12. In classless addressing, can two different blocks have the same prefix length? Explain.

Yes, two different blocks can have the same prefix length.

Explanation:

- **Classless Inter-Domain Routing (CIDR)** allows IP blocks to be defined with flexible prefix lengths.
- The **prefix length** defines the number of bits used for the **network portion** of the address.
- Two different IP address blocks (subnets) can have the **same prefix length** but belong to **different network ranges**.

Example:

- **Block 1:** 192.168.1.0/24
    - First address: 192.168.1.0
    - Last address: 192.168.1.255
    - Prefix length: 24 (indicating 256 addresses)
- **Block 2:** 10.0.0.0/24
    - First address: 10.0.0.0
    - Last address: 10.0.0.255
    - Prefix length: 24 (indicating 256 addresses)

Problems:

P18-1. What is the size of the address space in each of the following systems?

---

a. A system in which each address is only 16 bits.

- **Address space size:** $2^{16} = 65{,}536$ addresses.

---

b. A system in which each address is made of six hexadecimal digits.

- **Each hexadecimal digit represents 4 bits.**
- Number of bits: $6 \times 4 = 24$ bits.

Address space size: $2^{24} = 16{,}777{,}216$ addresses.

---

c. A system in which each address is made of four octal digits.

- **Each octal digit represents 3 bits.**
- Number of bits: $4 \times 3 = 12$ bits.

- **Address space size:**2^12=4,096 addresses.

P18-2. Rewrite the following IP addresses using binary notation:

---

a. 110.11.5.88

- **Convert to binary:**
- 110 → 01101110
- 11 → 00001011
- 5 → 00000101
- 88 → 01011000

Binary notation:

01101110.00001011.00000101.01011000

P18-3. Rewrite the following IP addresses using dotted-decimal notation:

---

a. 01011110 10110000 01110101 00010101

- Split into 8-bit sections:

01011110=94,10110000=176,01110101=117,00010101=21

- **Dotted-decimal:** 94.176.117. 21

---

b. 10001001 10001110 11010000 00110001

10001001=137,10001110=142,11010000=208,00110001=49

- **Dotted-decimal:**137.142.208.49

---

c. 01010111 10000100 00110111 00001111

01010111=87,10000100=132,00110111=55,00001111=15

- **Dotted-decimal:** 87.132.55.15

P18-4. Find the class of the following classful IP addresses:

---

a. 130.34.54.12

- **First octet:** 130
- **Range:** 128 - 191 (Class B)

- **Class:** B

b. 200.34.2.1

- **First octet:** 200
- **Range:** 192 - 223 (Class C)
- **Class:** C

P18-5. Find the class of the following classful IP addresses:

---

a. 01110111 11110011 10000111 11011101

- **First octet:** 01110111 (Binary → Decimal) = 119
- **Range:** 0 - 127 (Class A)
- **Class:** A

---

b. 11101111 11000000 11110000 00011101

- **First octet:** 11101111 (Binary → Decimal) = 239
- **Range:** 224 - 239 (Class D)
- **Class:** D (Multicast)

---

c. 11011111 10110000 00011111 01011101

- **First octet:** 11011111 (Binary → Decimal) = 223
- **Range:** 192 - 223 (Class C)
- **Class:** C

P18-6. In classless addressing, show the whole address space as a single block using CIDR notation.

- **Address space:**0.0.0.0/0
- CIDR block **/0** means all 32 bits are available for network addressing.

P18-7. In classless addressing, what is the size of the block (N) if the value of the prefix length (n) is one of the following?

---

a. n = 0; Size of block: N=232−0=232=4,294,967,296 addresses.

b. n = 14; Size of block: N=232−14=218=262,144 addresses.

P18-8. In classless addressing, what is the value of the prefix length (n) if the size of the block (N) is one of the following?

---

a. N = 1

- **Prefix length:** $n = 32 - \log 2(1) = 32$

---

b. N = 1024

- **Prefix length:** $n = 32 - \log 2(1024) = 32 - 10 = 22$

P18-9. Change each of the following prefix lengths to a mask in dotted-decimal notation:

---

a. n = 0

- **Mask:** 0.0.0.0

---

b. n = 14

- **Mask:** 11111111.11111100.00000000.00000000 ⟹ 255.252.0.0

P18-10. Change each of the following masks to a prefix length:

---

a. 255.224.0.0

- **Binary:** 11111111.11100000.00000000.00000000
- **Prefix length:** $n=11$

---

b. 255.240.0.0

- **Binary:** 11111111.11110000.00000000.00000000
- **Prefix length:** $n=12$

P18-11. Which of the following cannot be a mask in CIDR?

---

a. 255.225.0.0

- **Invalid:** Not a contiguous mask.

---

b. 255.192.0.0

- **Valid:** Prefix length = 10

P18-12. Each of the following addresses belongs to a block. Find the first and the last address in each block.

---

a. 14.12.72.8/24

- **First address:** 14.12.72.0
- **Last address:** 14.12.72.255

---

b. 200.107.16.17/18

- **First address:** 200.107.0.0
- **Last address:** 200.107.63.255

P18-13. Show the n leftmost bits of the following network addresses/masks that can be used in a forwarding table.

---

a. 170.40.11.0/24

- **Leftmost 24 bits:**

10101010.00101000.00001011

---

b. 110.40.240.0/22

- **Leftmost 22 bits:**

01101110.00101000.11110001

P18-14. Explain how DHCP can be used when the size of the block assigned to an organization is less than the number of hosts in the organization.

Dynamic Host Configuration Protocol (DHCP) Usage:

- **Problem:**
  If an organization has more hosts than available IP addresses, DHCP can dynamically allocate IP addresses for a limited period of time.
- **DHCP Solution:**

1. **Lease Time:** DHCP assigns IP addresses to devices temporarily through a lease period.

2. **Address Recycling:** When a device disconnects, its IP address is returned to the DHCP pool for reuse.
3. **Temporary Allocation:** Devices receive addresses only when they are online. Devices that are not connected do not hold onto an IP address.

- **Scenario:**

100 hosts but only 50 IP addresses. DHCP assigns IPs dynamically, ensuring that only active devices use the IPs. If 50 devices disconnect, their IPs are available for new devices.

P18-15. Compare NAT and DHCP. Both can solve the problem of a shortage of addresses in an organization, but by using different strategies.

Network Address Translation (NAT):

- **How It Works:**
- NAT maps multiple private IP addresses inside the network to a single public IP address used for internet communication.
- **Type:** Static NAT, Dynamic NAT, and PAT (Port Address Translation).
- **Pros:** Saves public IP addresses.
  Improves security by hiding private IPs.
- **Cons:** Slows down performance due to address translation.
  Can cause issues with some applications (e.g., VoIP, VPN).

Dynamic Host Configuration Protocol (DHCP):

- **How It Works:**
- DHCP dynamically assigns IP addresses to hosts from a pool of available addresses.
- **Type:** Automatic, Manual, and Dynamic Allocation.
- **Pros:** Reduces administrative effort.
  Recycles IP addresses effectively.
- **Cons:** Short-term assignments can lead to conflicts if poorly managed.
  Devices might lose connection when their lease expires.

P18-16. Assume we have an internet with an 8-bit address space. The addresses are equally divided between four networks (N0 to N3). The internetwork communication is done through a router with four interfaces (m0 to m3). Show the internet outline and the forwarding table (with two columns: prefix in binary and the interface number) for the only router that connects the networks. Assign a network address to each network.

Address Space: 2^8=256 addresses.

Network Assignment:

- N0: 00000000 to 00111111 (/6) → 0.0 to 63.255
- N1: 01000000 to 01111111 (/6) → 64.0 to 127.255
- N2: 10000000 to 10111111 (/6) → 128.0 to 191.255
- N3: 11000000 to 11111111 (/6) → 192.0 to 255.255

Router Forwarding Table:

Prefix (Binary)	Interface
00xxxxxx	m0
01xxxxxx	m1
10xxxxxx	m2
11xxxxxx	m3

P18-17. Assume we have an internet with a 12-bit address space. The addresses are equally divided between eight networks (N0 to N7). The internetwork communication is done through a router with eight interfaces (m0 to m7). Show the internet outline and the forwarding table (with two columns: prefix in binary and the interface number) for the only router that connects the networks. Assign a network address to each network.

Address Space: 2^12=4096 addresses.

Network Assignment:

- N0: 000 → 0 to 511
- N1: 001 → 512 to 1023
- N2: 010 → 1024 to 1535
- N3: 011 → 1536 to 2047
- N4: 100 → 2048 to 2559
- N5: 101 → 2560 to 3071
- N6: 110 → 3072 to 3583
- N7: 111 → 3584 to 4095

Router Forwarding Table:

Prefix (Binary)	Interface
000	m0
001	m1
010	m2
011	m3
100	m4
101	m5
110	m6
111	m7

P18-18. Assume we have an internet with a 9-bit address space. The addresses are divided between three networks (N0 to N2), with 64, 192, and 256 addresses respectively. The internetwork communication is done through a router with three interfaces (m0 to m2). Show the internet outline and the forwarding table (with two columns: prefix in binary and the interface number) for the only router that connects the networks. Assign a network address to each network.

Address Space: 2^9=512 addresses.

Network Assignment:

- N0: 0000000xx → 0 to 63 (/6)
- N1: 00001xxxx → 64 to 255 (/5)
- N2: 001xxxxx → 256 to 511 (/3)

Router Forwarding Table:

Prefix (Binary)	Interface
0000000	m0
00001	m1
001	m2

P18-19. Combine the following three blocks of addresses into a single block:

---

a. 16.27.24.0/26 → 64 addresses

b. 16.27.24.64/26 → 64 addresses

c. 16.27.24.128/25 → 128 addresses

Combined Block:

- Start: 16.27.24.0
- End: 16.27.24.255
- **Single Block:** 16.27.24.0/24
- **New CIDR Notation:** /24 (256 addresses)

P18-20. A large organization with a large block address (12.44.184.0/21) is split into one medium-size company using the block address (12.44.184.0/22) and two small organizations. If the first small company uses the block (12.44.188.0/23), what is the remaining block that can be used by the second small company?

Available Block:

- **12.44.184.0/21** → 2048 addresses
- **Used:**
  - 12.44.184.0/22 → 1024 addresses
  - 12.44.188.0/23 → 512 addresses
- **Remaining Block:** 12.44.190.0/23

P18-21. An ISP is granted the block 16.12.64.0/20. The ISP needs to allocate addresses for 8 organizations, each with 256 addresses.

---

a. Number and Range of Addresses:

- **Block Size:** 2^32−20=4096 addresses.
- **Range:** 16.12.64.0 to 16.12.79.255

---

b. Range of Addresses for Each Organization:

- Each organization needs 256 addresses.
- CIDR for 256 addresses → **/24.**
- **Ranges:** Org 1: 16.12.64.0/24
- Org 2: 16.12.65.0/24

- Org 3: 16.12.66.0/24
- …
- Org 8: 16.12.71.0/24

P18-22. An ISP is granted the block 80.70.56.0/21. The ISP needs to allocate addresses for two organizations each with 500 addresses, two organizations each with 250 addresses, and three organizations each with 50 addresses.

---

a. Number and Range of Addresses:

- **Block Size:** 2^32−21=2048 addresses
- **Range:** 80.70.56.0 to 80.70.63.255

---

b. Range of Addresses for Each Organization:

- 500 addresses → **/23**
- 250 addresses → **/24**
- 50 addresses → **/26**

Allocation:

- Org 1: 80.70.56.0/23
- Org 2: 80.70.58.0/23
- Org 3: 80.70.60.0/24
- Org 4: 80.70.61.0/24
- Org 5: 80.70.62.0/26
- Org 6: 80.70.62.64/26
- Org 7: 80.70.62.128/26

P18-23. An organization is granted the block 130.56.0.0/16. The administrator wants to create 1024 subnets.

---

a. Find the number of addresses in each subnet.

- **Given Block:** 130.56.0.0/16
- **Number of Subnets:** 1024=2^10 ⟹ 10 bits for subnetting.
- **New Prefix:** /16+10=/26
- **Number of Addresses per Subnet:** 2^(32−26)=64 addresses.

Each subnet contains 64 addresses.

---

b. Find the subnet prefix.

- **Subnet Prefix:** 130.56.0.0/26

---

c. Find the first and last address in the first subnet.

- **First Subnet Address:**130.56.0.0/26
- **First Address:** 130.56.0.0
- **Last Address:** 130.56.0.63

First Subnet Range: 130.56.0.0 to 130.56.0.63

---

d. Find the first and last address in the last subnet.

- **Last Subnet Address:** 130.56.255.192/26
- **First Address in Last Subnet:** 130.56.255.192
- **Last Address in Last Subnet:** 130.56.255.255

Last Subnet Range: 130.56.255.192 to 130.56.255.255

P18-24. Can router R1 in Figure 18.35 receive a packet with destination address 140.24.7.194? What will happen to the packet if this occurs?

Check if Router R1 Can Receive the Packet:

- **IP Address:** 140.24.7.194
- Check if this address belongs to any subnet directly connected to R1.

Router R1's Interface Subnets:

- Let's assume R1's connected subnets are:
    - 140.24.7.0/24 → Covers addresses from 140.24.7.0 to 140.24.7.255.

Verification:

- 140.24.7.194 falls within the subnet **140.24.7.0/24.**

Yes, R1 can receive the packet.

What Will Happen?

- If the destination address is within one of R1's directly connected networks, R1 will forward the packet directly to the host.
- Otherwise, R1 will consult its forwarding table. If no match is found, the packet is discarded.

P18-25. Assume router R2 in Figure 18.35 receives a packet with destination address 140.24.7.42. How is the packet routed to its final destination?

Check if Router R2 Can Route the Packet:

- **Destination IP:** 140.24.7.42
- **Subnet Check:**
  - Assuming R2 has a route for 140.24.7.0/24.

Routing Decision:

- **Match Found:**
  - 140.24.7.42 falls in **140.24.7.0/24.**
  - R2 forwards the packet to the appropriate next hop (either directly to the host or to another router if necessary).

Packet will be routed to the corresponding host or the next router.

****