1. **Describe fundamental characteristics of SDN**

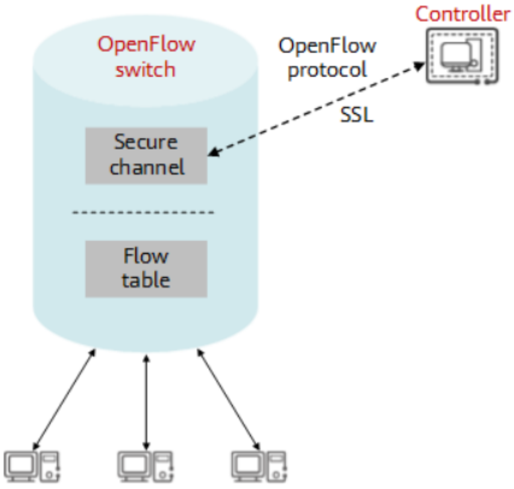
**Ans.**

- **Directly programmable:** Network control is directly programmable because it is decoupled from forwarding functions.
- **Agile:** Abstracting control from forwarding lets administrators dynamically adjust network-wide traffic flow to meet changing needs.
- **Centrally managed:** Network intelligence is (logically) centralized in software-based SDN controllers that maintain a global view of the network, which appears to applications and policy engines as a single, logical switch.
- **Programmatically configured:** SDN lets network managers configure, manage, secure, and optimize network resources very quickly via dynamic, automated SDN programs, which they can write themselves because the programs do not depend on proprietary software.
- **Open standards-based and vendor-neutral:** When implemented through open standards, SDN simplifies network design and operation because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols.

2. **Summarize the working of Openflow**

**Ans.**

- The OpenFlow architecture consists of a controller, OpenFlow switch, and secure channel.
- The controller controls the network in a centralized manner to implement the functions of the control layer.
- The OpenFlow switch is responsible for forwarding at the data layer; it exchanges messages with the controller through a secure channel to receive forwarding entries and report its status.



*OpenFlow architecture*

**OpenFlow Controller**

- An OpenFlow controller is the brain of the SDN architecture and is located at the control layer to instruct data forwarding through the OpenFlow protocol.
- Currently, mainstream OpenFlow controllers are classified into two types: open-source controllers and vendor-developed commercial controllers.
- The widely used open-source controllers include NOX, POX, and OpenDaylight

**OpenFlow Secure Channel**

- A secure channel is established between a controller and an OpenFlow switch. Through this channel, the controller controls and manages the switch, and receives feedback from the switch.
- The messages exchanged over the OpenFlow secure channel must comply with the format specified by the OpenFlow protocol.
- The OpenFlow secure channel is usually encrypted using Transport Layer Security (TLS), Controller-to-Switch message: is sent by the controller to the OpenFlow switch to manage or obtain the OpenFlow switch status.
- Asynchronous message: is sent by the OpenFlow switch to the controller to update network events or status changes to the controller.
- Symmetric message: is sent without solicitation by either the OpenFlow switch or the controller. It is mainly used to set up a connection and detect whether the peer is online.

**OpenFlow Switch**

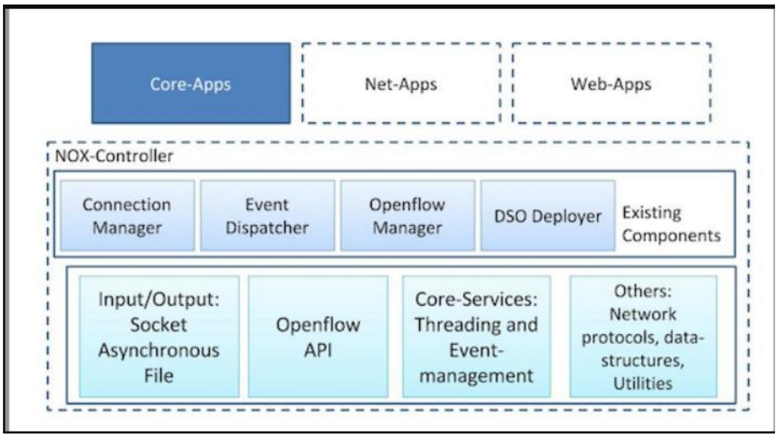
- As a core component of the OpenFlow network, an OpenFlow switch is mainly responsible for forwarding at the data layer. It can be a physical or virtualized switch/router. OpenFlow switches are classified into the following types based on their support for OpenFlow
- Dedicated OpenFlow switch: is a standard OpenFlow device that supports only OpenFlow forwarding. The switch processes all traffic that passes through it in OpenFlow mode, and cannot perform Layer 2 or Layer 3 forwarding on the traffic.
- OpenFlow-compatible switch: supports both OpenFlow forwarding and Layer 2/3 forwarding. It is a commercial switch that supports OpenFlow features such as flow tables and secure channels.
- An OpenFlow switch forwards packets entering the switch based on the flow table, which contains a set of policy entries instructing the switch on how to process traffic. Flow entries are generated, maintained, and delivered by a controller.

**Flow entry**

- Traditional network devices such as switches and routers forward data based on the locally saved Layer 2 MAC address forwarding table, Layer 3 IP address routing table, and transport-layer port numbers.
- OpenFlow switches forward data based on flow tables that contain network configuration information of all layers on the network, instead of 5-tuple information.
- The entries in a flow table are flexible combinations of certain keywords and actions.
- Each flow entry in an OpenFlow flow table consists of match fields and a set of instructions applying to matching packets.
- When receiving a data packet, an OpenFlow switch parses and matches the packet header against match fields in the flow entries, and executes the corresponding instruction if a match is found.

**3. Explain NOX architecture**

**Ans.**


Core-Apps
Net-Apps
Web-Apps
NOX-Controller
Connection Manager
Event Dispatcher
Openflow Manager
DSO Deployer
Existing Components
Input/Output: Socket Asynchronous File
Openflow API
Core-Services: Threading and Event-management
Others: Network protocols, data-structures, Utilities

- The NOX core provides helper methods, such as network packet process, threading and event engine, in addition to OpenFlow APIs for interacting with OpenFlow switches, and I/O operations support.
- At the top, we have applications: Core, Net and Web.
- However, with the current NOX version, there are only two core applications: OpenFlow and switch, and both network and web applications are missing.
- The middle layer shows the in-built components of NOX. The connection manager, event dispatcher and OpenFlow manager are self-explanatory, whereas the dynamic shared object (DSO) deployer basically scans the directory structure for any components being implemented as DSOs.
- All the applications can be viewed as components.
- All applications inherit from the component class. Hence, NOX applications are generally composed of cooperating components that provide the required functionality. In short, a component encapsulates specific functionality that is made available to NOX.

- An event represents a low-level or high-level event in the network.
- Typically, the event only provides the information, and processing of that information is deferred to handlers.
- Many events roughly correlate to something which happens on the network that may be of interest to a NOX component.
- These components, typically, consists a set of event handlers. In this sense, events drive all execution in NOX.

**4. Compare and contrast SDN and traditional networking**

**Ans.**

**similarities between Software Defined Network (SDN) and Traditional Network:**

- Both SDN and traditional networks aim to provide network connectivity between devices.
- Both types of networks use standard networking protocols, such as TCP/IP and Ethernet, for communication between network devices.
- Both SDN and traditional networks have security concerns, such as unauthorized access, data breaches, and network attacks.
- Both types of networks can provide quality of service (QoS) features to ensure that critical applications receive the required bandwidth and priority.

| S.No. | SDN | TRADITIONAL NETWORK |
|---|---|---|
| 01. | Software Defined Network is virtual networking approach. | Traditional network is the old conventional networking approach. |
| 02. | Software Defined Network is centralized control. | Traditional Network is distributed control. |
| 03. | This network is programmable. | This network is non programmable. |
| 04. | Software Defined Network is open interface. | Traditional network is closed interface. |
| 05. | In Software Defined Network data plane and control plane are decoupled by software. | In traditional network data plane and control plane are mounted on same plane. |
| 06. | It supports automatic configuration so it takes less time. | It supports static/manual configuration so it takes more time. |
| 07. | It can prioritize and block specific network packets. | It leads all packets in the same way no prioritization support. |
| 08. | It is easy to program as per need. | It is difficult to program again and to replace existing program as per use. |
| 09. | Cost of Software Defined Network is low. | Cost of Traditional Network is high. |
| 10. | Structural complexity is low in Software Defined Network. | Structural complexity is high in Traditional Network. |
| 11. | Extensibility is high in Software Defined Network. | Extensibility is low in Traditional Network. |
| 12. | In SDN it is easy to troubleshooting and reporting as it is centralized controlled. | In Traditional network it is difficult to troubleshoot and report as it is distributed controlled. |

| | | |
|---|---|---|
| 13. | Its maintenance cost is lower than traditional network. | Traditional network maintenance cost is higher than SDN. |

**5. Build TCP Header format by depicting distinct fields and solve the following: Given a dump of a TCP header in hexadecimal format:05320017 00000001 00000000 500207FF 00000000**

**i. What is the source port number?**

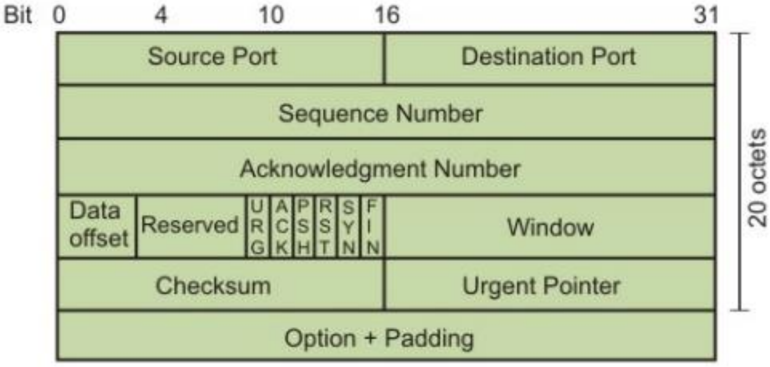
**ii. What is the destination port number?**

**iii. What is the length of the header?**

**iv. What is the type of segment?**

**v. What is the window size?**

**Ans.**



• Source port (16 bits): It defines the port number of the application program in the host of the sender

• Destination port (16 bits): It defines the port number of the application program in the host of the receiver

• Sequence number (32 bits): It conveys the receiving host which octet in this sequence comprises the first byte in the segment

• Acknowledgement number (32 bits): This specifies the sequence number of the next octet that receiver expects to receive

• HLEN (4 bits): This field specifies the number of 32-bit words present in the TCP header

• Control flag bits (6 bits):

- URG: Urgent pointer
- ACK: Indicates whether acknowledge field is valid

- PSH: Push the data without buffering
- RST: Resent the connection
- SYN: Synchronize sequence numbers during connection establishment
- FIN: Terminate the connection

• Window (16 bits): Specifies the size of window

• Checksum (16 bits): Checksum used for error detection.

• User pointer (16 bits): Used only when URG flag is valid

• Options: Optional 40 bytes of information

The dump of a TCP header in hexadecimal format:

```
05320017 00000001 00000000 500207FF  00000000
```

i. Source port number: - (2 byte) -> 0532(in hex) or
$(0532)_{16} = (0 \times 16^3) + (5 \times 16^2) + (3 \times 16^1) + (2 \times 16^0) = 1330$ (in decimal)

ii. Destination port number: - (2 byte) -> 0017(in hex) or
$(0017)_{16} = (0 \times 16^3) + (0 \times 16^2) + (1 \times 16^1) + (7 \times 16^0) = 23$(in decimal)

iii. Length of the header (4 bits) -> 5(in hex and decimal) Means 5*4= 20 bytes header

iv. Type of the segment -> 0X02: - is control field and this indicates a SYN packet.
(5002)->0101 0000 0000 0010->0101 is header, 000000 is reserved,
Urg=0,ack=0,psh=0,rst=0,syn=1,fin=0

v. Window size -> 07FF (in hex) or
$(07FF)_{16} = (0 \times 16^3) + (7 \times 16^2) + (15 \times 16^1) + (15 \times 16^0) = 2047$ (in decimal)

**6. Consider the three-way handshake mechanism followed during TCP connection establishment between hosts P and Q. Let X and Y be two random 32-bit starting sequence numbers chosen by P and Q respectively. Suppose P sends a TCP connection request message to Q with a TCP segment having SYN bit =1, SEQ number =X, and ACK bit**
**=0. Suppose Q accepts the connection request.**
**Construct the following**
i) **Information in TCP segment header that is sent by Q to P with a neat diagram**
ii) **The different phases of congestion control in TCP?**

**Ans.**
**i)**
Host P sends the first SYN packet with SEQ number =X, SYN flag =1 and ACK flag =0 as it’s a connection request.

Host Q will reply back with a SYN packet and acknowledging the arrival of P'S SYN packet.

Host Q will send a packet with **SYN flag =1**, **SEQ number = Y** , to synchronize and establish the connection, and **ACK flag = 1** to acknowledge the P′S SYN packet, with **ACK number = X+1** because ACK number denotes the sequence number of next expecting Byte.

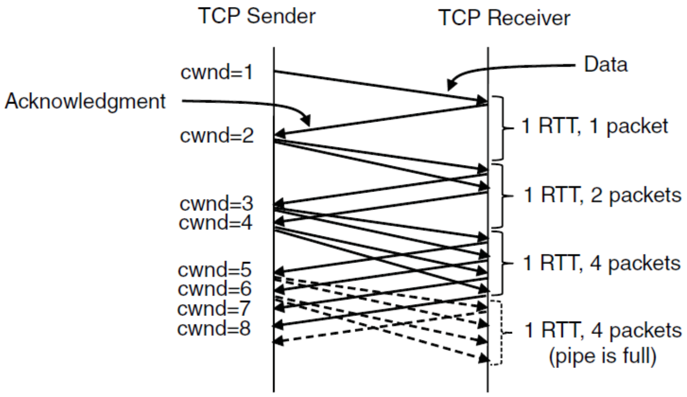
Then P will reply back with an ACK packet to complete the three-way handshake. (not asked here)

FIN flag is used to terminate the connection, and will not be used here, FIN flag = 0.

**ii)**

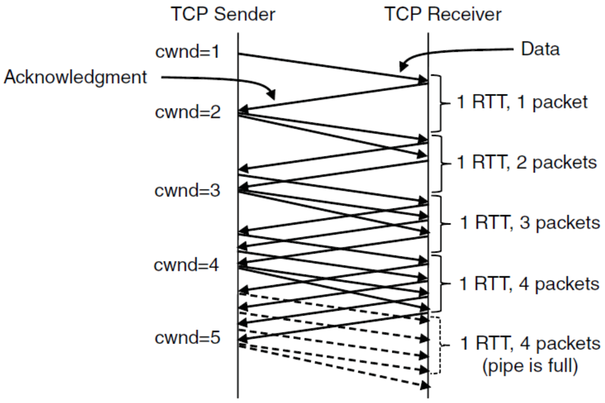
**Slow Start Phase**

- **Exponential increment**: In this phase after every RTT the congestion window size increments exponentially.
- **Example:-** If the initial congestion window size is 1 segment, and the first segment is successfully acknowledged, the congestion window size becomes 2 segments. If the next transmission is also acknowledged, the congestion window size doubles to 4 segments. This exponential growth continues as long as all segments are successfully acknowledged.



**Congestion Avoidance Phase**

- **Additive increment:** This phase starts after the threshold value also denoted as ssthresh. The size of cwnd(congestion window) increases additive. After each RTT cwnd = cwnd + 1.
- **Example:-** if the congestion window size is 20 segments and all 20 segments are successfully acknowledged within an RTT, the congestion window size would be increased to 21 segments in the next RTT. If all 21 segments are again successfully acknowledged, the congestion window size would be increased to 22 segments, and so on.



Additive increase from an initial congestion window of 1 segment

**Congestion Detection Phase**

- **Multiplicative decrement:** If congestion occurs, the congestion window size is decreased. The only way a sender can guess that congestion has happened is the need to retransmit a segment. Retransmission is needed to recover a missing packet that is assumed to have been dropped by a router due to congestion. Retransmission can occur in one of two cases: when the RTO timer times out or when three duplicate ACKs are received.

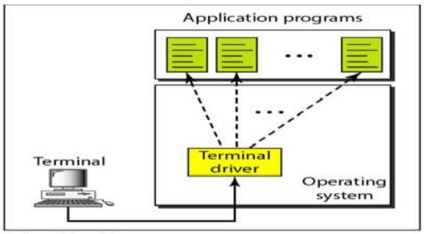
**7. Imagine a user wants a server to display a file (file1) on a remote server. She can type cat file1. However, suppose the name of the file has been mistyped (filea instead of file1).**

i. **Illustrate the user client interactively remote log- in to the server host.**

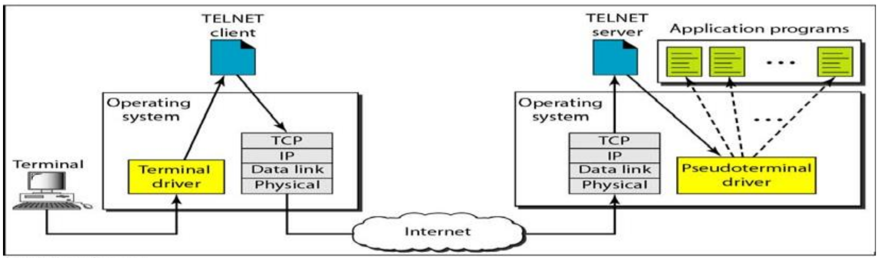
ii. **Show the data typed at remote terminal to correct this situation in default implementation of TELNET.**

**Ans.**

i.



a. Local log-in



b. Remote log-in

ii.

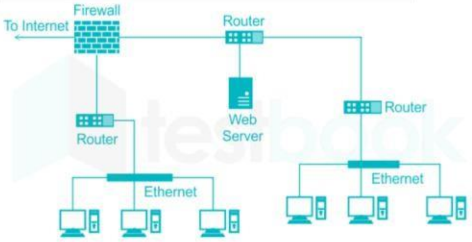
| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| c | a | t | | f | i | l | e | a | IAC | EC | 1 |

Typed at the remote terminal

Embedding TELNET uses only one TCP connection. The server uses the well-known port 23, and the client uses an ephemeral port. The same connection is used for sending both data and control characters. TELNET accomplishes this by embedding the control characters in the data stream. However, to distinguish data from control characters, each sequence of control characters is preceded by a special control character called interpret as control (lAC). For example, imagine a user wants a server to display a file (file1) on a remote server. She can type cat file1. However, suppose the name of the file has been mistyped (filea instead of file1). The user uses the backspace key to correct this situation. However, in the default implementation of TELNET, the user cannot edit locally; the editing is done at the remote server. The backspace character is translated into two remote characters (lAC EC), which are embedded in the data and sent to the remote server. What is sent to the server is shown in Figure.

**8. i) Sketch Cisco Service-Oriented Network Architecture.**

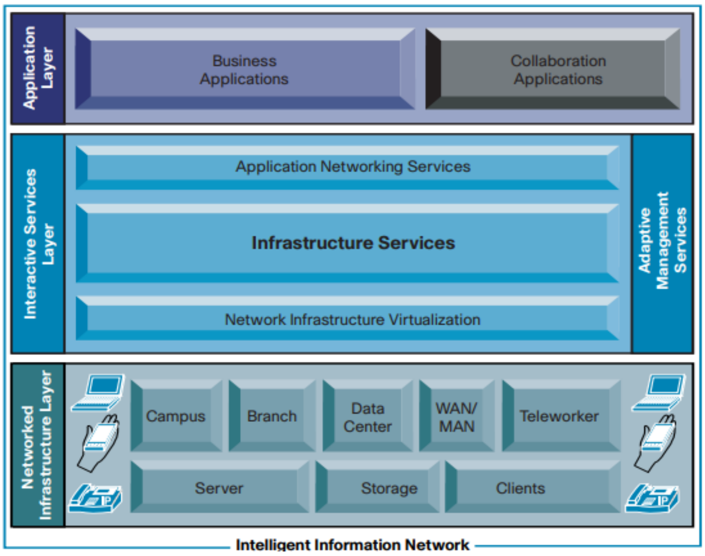
**ii) Consider an enterprise network with two Ethernet segments, a web server and a firewall, connected via three routers as shown below.**



**What is the number of subnets inside the enterprise network? Justify and mark in diagram.**

**Ans.**

**i**


Application Layer
Business Applications
Collaboration Applications
Interactive Services Layer
Application Networking Services
Infrastructure Services
Network Infrastructure Virtualization
Adaptive Management Services
Networked Infrastructure Layer
Campus
Branch
Data Center
WAN/ MAN
Teleworker
Server
Storage
Clients
Intelligent Information Network

- Enabling rapid adoption and deployment of new application services at a reduced cost of development and overhead
- Coordinating application and network events with business process to speed business agility Enforcing business policies in the application and network infrastructure to improve security and reduce risk.
- Aligning network resources to applications to meet business objectives to provide a competitive differentiation.
- **Functionality**: Supports the organizational requirements.
- **Scalability**: Supports growth and expansion of organizational tasks by separating functions and products into layers; this separation makes it easier to grow the network.
- **Availability**: Provides the necessary services, reliably, anywhere, anytime.

- **Performance**: Provides the desired responsiveness, throughput, and utilization on a per application basis through the network infrastructure and services.
- **Manageability**: Provides control, performance monitoring, and fault detection.
- **Efficiency**: Provides the required network services and infrastructure with reasonable operational costs and appropriate capital investment on a migration path to a more intelligent network, through step-by-step network services growth.
- **Security**: Provides for an effective balance between usability and security while protecting information assets and infrastructure from inside and outside threats.

**Three Layers of Cisco SONA**

**The networked infrastructure layer**, where all the IT resources are interconnected across a converged network foundation.

- ➢ At the networked infrastructure validated Cisco enterprise architectures provide complete design guidance for a fully integrated end-to-end system across your entire network.
- ➢ The IT resources include servers, storage, and clients.
- ➢ The Networked Infrastructure layer represents how these resources exist in different places in the network, including the campus, branch, data center, enterprise edge, WAN, metropolitan-area network (MAN), and with the teleworker.
- ➢ The objective of this layer is to provide connectivity, anywhere and anytime.

**The interactive services layer**, which enables efficient allocation of resources to applications and business processes delivered through the networked infrastructure.

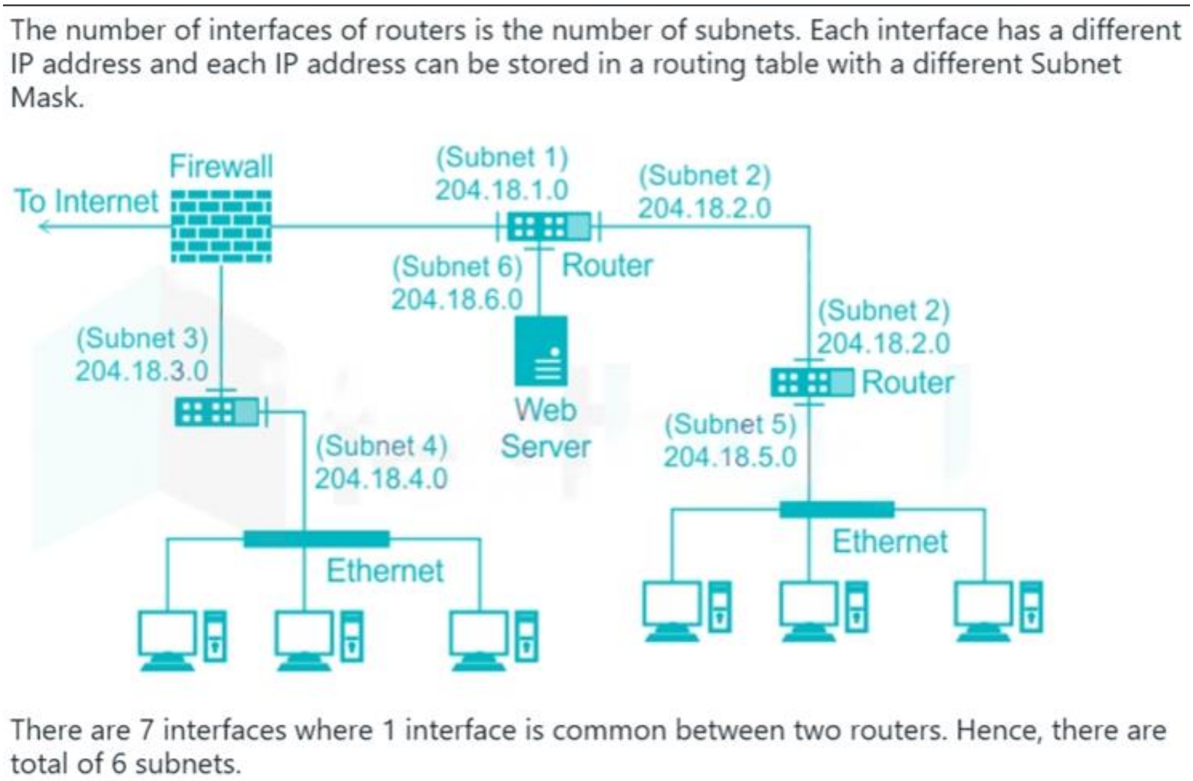
- ➢ At the interactive services layer, Cisco integrates a complete suite of services into intelligent systems that optimize the delivery of business and collaboration applications for more predictable and reliable performance, while lowering operational costs.
- ➢ This layer enables efficient allocation of resources to applications and business processes delivered through the networked infrastructure.

**The applications layer**, which contains the business applications and collaborative applications that take advantage of efficiencies from the interactive services.

- ➢ At the application layer, by deeply integrating with the network fabric, Cisco application networking solutions require no client installation or application changes while maintaining application visibility and security throughout application delivery.

  The objective of this layer is to meet business requirements and achieve efficiencies by leveraging the interactive services layer.

ii.



The number of interfaces of routers is the number of subnets. Each interface has a different IP address and each IP address can be stored in a routing table with a different Subnet Mask.

There are 7 interfaces where 1 interface is common between two routers. Hence, there are total of 6 subnets.

**9. Categorize the network design approach in a Company Management Structure:**
**a) A CEO creates a strategic plan for the company and delegates tasks to department managers, who in turn delegate to their subordinates**
**b) Assembly Line: Workers assemble individual parts to create a final product.**
**ii)Classify both approaches**
**iii)which is better? Justify your answer**

**Ans.**

i) a) Top-Down Approach
b) Bottom-Up Approach

ii)

- **Starting Point**: The top-down approach starts with a high-level understanding of the problem, while the bottom-up approach starts with individual components.
- **Focus:** The top-down approach focuses on high-level planning and decision-making, while the bottom-up approach focuses on the implementation and execution of individual tasks.
- **Prioritization:** The top-down approach prioritizes the end goal and the desired outcome, while the bottom-up approach prioritizes the details and getting each individual component right.

- **Control:** The top-down approach often involves central control and decision-making, while the bottom-up approach empowers individuals and teams to make decisions and drive the process forward.
- **Communication:** The top-down approach relies on communication from the top to the bottom, while the bottom-up approach emphasizes collaboration and communication between different teams working on different components.
- **Flexibility:** The top-down approach can be less flexible, as decisions are made at a high level and the process is more structured, while the bottom-up approach allows for more adaptability and iteration based on feedback and changing requirements.
- **Risk:** The top-down approach can be riskier, as decisions are made at a high level and may not account for all the details and complexities of the problem, while the bottom-up approach addresses risks by focusing on the details and iterating based on feedback.

ii) The top-down approach is best used when:

- The problem is complex and needs to be broken down into smaller, manageable parts.
- There is a need to understand the big picture before diving into details.
- A clear understanding of the end goal is required before starting the project.
- The solution can be divided into smaller subproblems that can be solved independently.
- The problem has multiple potential solutions, and a top-down approach can help prioritize and evaluate them.