...5...
Wireless Network Technologies

Learning Outcomes...

- ❑ Compare the characteristics of 3G, 4G, 5G
- ❑ Illustrate SDN Architecture.
- ❑ Explain Network Functions Visualization.
- ❑ Describe the role of Edge Computing and Edge Networking.
- ❑ Describe role of various Multimedia wireless protocols.

5.0 INTRODUCTION

- A wireless network enables devices to connect and communicate without the need for physical cables, using radio waves, infrared signals or other wireless technologies.
- These wireless networks have transformed how we access the internet, share resources, and communicate, offering flexibility and mobility in various environments.
- Computer networks that are not connected by cables are called wireless networks. They generally use radio waves for communication between the network nodes or devices.
- A wireless network refers to a computer network that makes use of Radio Frequency (RF) connections between nodes in the network.

Types of Wireless Networks:

1. **Wireless LAN (WLAN):** Used for local area networking, like in homes or offices. WLANs connect devices within a small limited geographic area, such as a home, office, school, university, campus, etc., and provide communication and shared access to different devices and resources, like printers, computers, etc. Typically, Ethernet cables and Wi-Fi are used for wireless network connection to provide high data transfer speeds
2. **Wireless Metropolitan Area Network (WMAN):** Larger networks covering a metropolitan area. WMANs typically covering distances of up to 100 kilometers, such as campus or city. The key technology used in WMANs is fiber optic technology providing high speed.
3. **Wireless Personal Area Network (WPAN):** Used for short-range communication between devices, like Bluetooth. WPANs are used for short-range communication, typically up to 100 meters, to connect personal devices like smartphones, tablets, laptops, headphones, and other peripherals.
4. **Wireless Wide Area Network (WWAN):** Used for larger networks, like cellular networks. WWANs Connect multiple LANs over a broad geographic area of a city, country, or continent and enable communication and data transfer across large distances. Typically, they use transmission technologies such as leased lines, satellite links, internet, cellular technology and so on.

Working of Wireless Networks:

- Currently, wireless technologies, such as Wi-Fi, Bluetooth, cellular and others, have become an essential part of our lives due to their ease of use, flexibility and ability to support a large number of connected devices.
- They enable us to stay in touch at all times and in any location, whether at home, in the workplace, or in a public space, etc.
- The term wireless networking comes with the combination of two words i.e. Wireless + Networking. The word wireless is defined as “having no wires” or ‘without cable’ while networking is defined as ‘the interaction of devices/nodes with one other to exchange information’.

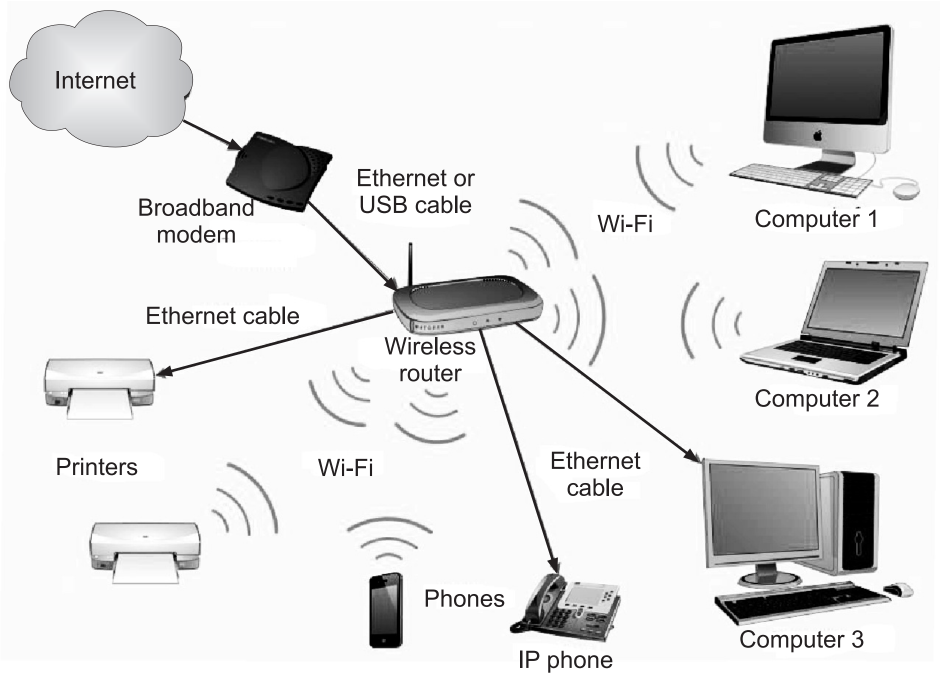


Fig. 5.1: Wireless Home Network

Benefits of Wireless Networks:

1. **Mobility:** Users can access the network while moving within the coverage area.
2. **Flexibility:** No need for physical cables; devices can connect easily.
3. **Cost-Effectiveness:** Reduces infrastructure costs by eliminating extensive wiring.
4. **Scalability:** Easy to expand by adding new devices or access points.
5. **Convenience:** Supports multiple device connections simultaneously.

Challenges of Wireless Networks:

1. **Security Risks:** Vulnerable to unauthorized access and attacks without proper encryption. Security and stability of wireless networks are the most essential issues because of their vulnerability to unauthorized access, interception of data and other attacks.
2. **Limited Range and Speed:** Performance depends on the type of wireless technology used.
3. **Reliability Issues:** Environmental factors like weather can affect signal quality in outdoor networks.

- Wireless networks are a popular solution for homes, businesses and telecommunications networks.

Components of a wireless network:

- A wireless network consists of several essential components, which work together to enable seamless communication without physical connections. The main components include:

Sr. No.	Components	Description
1.	Access Point (AP)	A network device that allows wireless devices to connect to and communicate with wired networks.
2.	Router	A device that connects two or more packet-switched networks or subnetworks, manages traffic between the networks, receives and sends data on computer networks and allows multiple devices to use the same wireless Internet connection.
3.	Wireless Devices	Electronic devices such as smartphones, tablets, laptops, game consoles and other gadgets equipped with wireless adapters that allow them to connect to the network without wires.
4.	Modem	A device that connects the network to the Internet Service Provider (ISP) and converts digital signals for transmission.
5.	Antennas	Devices that enhance signal strength and coverage, either built into electronic devices or attached externally.
6.	Switch	An electrical component used to control multiple devices remotely without physical wiring.
7.	Repeater	An electronic device that receives a wireless signal and amplifies it, extending the network's coverage area.
8.	Extender	A device that boosts the wireless signal range and retransmits it to farther distances, helping to bridge gaps in coverage.
9.	Bridge	A device used to improve and extend Wi-Fi network coverage, connecting the segments of the same network.
10.	Firewall	A network security device that monitors incoming and outgoing traffic and blocks unauthorized access.
11.	Gateway	A device that connects different networks, allowing communication between different protocols, or wired and wireless networks.

5.1 WIRELESS NETWORK COMMUNICATION

- Cellular technology is the foundation of mobile wireless communications and supports users in locations that are not easily served by wired networks.
- Cellular technology is the underlying technology for mobile telephones, personal communications systems, wireless Internet and wireless web applications, and much more.
- The 3G, 4G, and 5G are generations of wireless cellular technology, each offering increased speed, higher capacity, and improved functionality.
- The 3G introduced mobile internet, 4G brought high-speed data and mobile broadband and 5G introduced ultra-fast speeds with low latency and improved device connectivity.

5.1.1 3G

- 3G or third-generation wireless technology, was quite an improvement in mobile data transmission speeds.
- It was introduced in the early 2000s and gave us a taste of things to come, namely accessing the internet from a mobile phone. But that wasn’t all it did for us.
- The objective of the third generation (3G) of wireless communication is to provide fairly high-speed wireless communications to support multimedia, data and video in addition to voice.

- Features included:
  1. Data speeds up to 2 Mbps (Although HSPA eventually reached speeds up to 42 Mbps).
  2. Mobile internet browsing.
  3. Support for video calls and mobile TV.
  4. Advanced security features for mobile devices.
  5. 3G networks operate on frequencies between 850 MHz and 2100 MHz, and they use technologies like WCDMA (UMTS) and CDMA2000.

5.1.2 4G (Mobile Broadband)

- 4G, launched around 2010 and pushed the envelope even further. It allowed more people to adopt mobile data as a primary means of connecting to the internet with its high-speed capabilities.
- The evolution of smartphones and cellular networks has ushered in a new generation of capabilities and standards, which is collectively called 4G.
- 4G systems provide ultra-broadband Internet access for a variety of mobile devices including laptops, smartphones, and tablets.
- 4G networks support Mobile Web access and high-bandwidth applications such as high-definition mobile TV, mobile video conferencing and gaming services.
- These requirements have led to the development of a fourth generation (4G) of mobile wireless technology that is designed to maximize bandwidth and throughput while also maximizing special efficency.
- Some of the new features that came along with 4G included:
  1. Theoretical download speeds up to 1 Gbps, but practical speeds of around 100 Mbps were usually the norm for most users.
  2. Low latency for improved real-time applications like online games and apps.
  3. Support for HD video streaming.
  4. Improved capacity for simultaneous users with better congestion management.
  5. 4G networks primarily use LTE (Long-Term Evolution) technology and operate on various frequency bands, typically between 600 MHz and 2.5 GHz.

4G+ (LTE-Advanced):

- 4G+ was the next big improvement introduced to mobile data standards. Some of the features that it implemented include:
  1. Theoretical download speeds of up to 3 Gbps, but like 4G, the practical speeds that users experienced were much lower, closer to 300 Mbps
  2. Even further reduced latency for better real-time application performance
  3. Carrier aggregation tech, allows users to use multiple frequency bands simultaneously
  4. Enhanced MIMO (Multiple Input, Multiple Output) capabilities

5.1.3 5G (Next Frontier)

- By 2020, the huge amounts of data traffic generated by tablets and smartphones will be augmented by an equally huge and perhaps much larger, amount of traffic from the Internet of Things (IoT), which includes shoes, watches, appliances, cars, thermostats, door locks and much more.
- The 5G, the current generation of wireless technology, is incredibly fast by modern standards.
- It is fast enough to be used as the primary internet connection for homes and small businesses while also adding these new improvements:
  1. Theoretical speeds up to 20 Gbps, with real-world speeds closer to the 1 Gbps
  2. Ultra-low latency, which can be as low as 1 ms
  3. Massive device connectivity, up to 1 million devices per square kilometer

4. Network slicing for customized service delivery, improvising performance
5. 5G operates on a wide range of frequency bands, including sub-6 GHz and mmWave (24-100 GHz) frequencies.

Comparison of 3G vs 4G vs 5G:

Feature	3G	4G	4G+ (LTE Advanced)	5G
Peak Data Rate	Up to 42 Mbps	Up to 1 Gbps	Up to 3 Gbps	Up to 20 Gbps
Latency	100-500 ms	20-30 ms	10-20 ms	1-4 ms
Frequency Bands	850 MHz - 2.1 GHz	600 MHz - 2.5 GHz	600 MHz - 6 GHz	600 MHz - 100 GHz
Network Architecture	Circuit-Switched	Packet-Switched	Packet-Switched	Packet-Switched, Virtualized
Download/Upload Speed	3-7 Mbps / 1 Mbps	10-50 Mbps / 10 Mbps	100-150 Mbps / 50 Mbps	100 Mbps-10 Gbps / Up to 10 Gbps
Use Cases	Voice, SMS, MMS	Streaming, VoIP, Web	HD Streaming, IoT	IoT, VR/AR, Autonomous Vehicles
Backwards Compatibility	2G	3G, some 2G	4G, 3G	4G, 3G, 2G

5.2 SDN (SOFTWARE DEFINED NETWORK)

- Software-defined networking (SDN) is a new networking paradigm that separates the network's control and data planes.
- The traditional networking architecture has a tightly coupled relationship between the data and control planes. This means that network devices, such as routers and switches, are responsible for forwarding packets and determining how the network should operate.
- With SDN, the control plane is decoupled from the data plane and implemented in software, allowing for centralized network control.
- The control plane, also called the network controller, is responsible for making decisions about how traffic should be forwarded, based on the overall network policy.
- The data plane, on the other hand, is responsible for forwarding traffic based on the decisions made by the control plane.
- In SDN, network devices are called switches, and they are typically simple, low-cost devices that forward traffic based on the instructions received from the network controller.
- The controller communicates with the switches using a standard protocol, such as OpenFlow, which allows the controller to program the switches to forward traffic in a particular way.

Definitions of SDN:

Software-Defined Networking (SDN) is an approach to networking in which control is decoupled from hardware and given to a software application called a controller. **OR**

SDN is a technology to networking that allows centralized, programmable control planes so that network operators can control and manage directly their own virtualized networks.

Data Plane:

- In computer networking, the data plane is the part of a network device responsible for forwarding data packets from one interface to another. It is also referred to as the forwarding plane or the user plane.

- The data plane operates at the lowest level of the network stack, typically at Layer 2 (the Data Link layer) and Layer 3 (the Network layer) of the OSI model.
- Its main responsibility is to forward packets from one interface to another based on the destination address contained in the packet header.
- In traditional networking, network devices such as routers and switches have a tightly coupled control plane and data plane.
- This means that the devices are responsible for both forwarding packets and making decisions about how the network should operate.
- However, in software-defined networking (SDN), the control plane is separated from the data plane, allowing for centralized control of the network.
- In SDN, the data plane is implemented in network devices, such as switches, and is responsible for forwarding packets based on the instructions received from the centralized control plane.
- This allows for greater flexibility and scalability in the network, as the data plane can be reprogrammed in real-time to accommodate changing network conditions.

Control Plane:

- In computer networking, the control plane is part of a network device or system that is responsible for managing and controlling the flow of network traffic.
- It is responsible for making decisions about how packets are forwarded across the network based on factors such as network topology, routing protocols, and network policies.
- The control plane operates at a higher network stack level than the data plane, typically at Layer 3 (the Network layer) and above in the OSI model. It is responsible for routing, switching, and traffic engineering tasks.
- In traditional networking, the control plane and data plane are tightly coupled, meaning that network devices such as routers and switches are responsible for forwarding packets and making decisions about how the network should operate.
- However, in software-defined networking (SDN), the control plane is separated from the data plane, allowing for centralized network control.
- In SDN, the controller communicates with the network devices in the data plane using a standard protocol, such as OpenFlow, to program the devices to forward packets in a particular way.
- The benefits of a separate control plane in SDN include greater network flexibility and scalability, as the network policy can be changed in real-time to meet changing network conditions.
- It also allows for easier network management, as the network can be managed from a centralized location.

5.2.1 SDN Architecture

- The architecture of software-defined networking (SDN) consists of three main layers namely, the application layer, the control layer and the infrastructure layer.
- Each layer has a specific role and interacts with the other layers to manage and control the network. Fig. 5.2 shows architecture of SDN.
- Let us see layers in Fig. 5.2 in detail:
  1. **Infrastructure Layer:** The infrastructure layer/tier is the bottom layer of the SDN architecture, also known as the data plane. It consists of physical and virtual network devices such as switches, routers, and firewalls that are responsible for forwarding network traffic based on the instructions received from the control plane.
  2. **Control Layer:** The control layer is the middle layer of the SDN architecture, also known as the control plane. It consists of a centralized controller that communicates with the infrastructure

layer devices and is responsible for managing and configuring the network. The controller interacts with the devices in the infrastructure layer using protocols such as OpenFlow to program the forwarding behavior of the switches and routers. The controller uses network policies and rules to make decisions about how traffic should be forwarded based on factors such as network topology, traffic patterns, and quality of service requirements.

3. **Application Layer:** The application layer is the top layer of the SDN architecture and is responsible for providing network services and applications to end-users. This layer consists of various network applications that interact with the control layer to manage the network. Examples of applications that can be deployed in an SDN environment include network virtualization, traffic engineering, security, and monitoring. The application layer can be used to create customized network services that meet specific business needs.

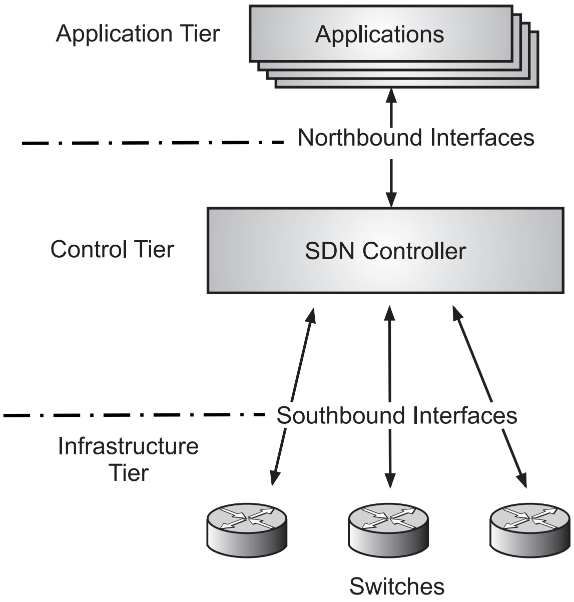


Fig. 5.2: Architecture of Software-Defined Networking (SDN)

- The main benefit of the SDN architecture is its flexibility and ability to centralize control of the network. The separation of the control plane from the data plane enables network administrators to configure and manage the network more easily and in a more granular way, allowing for greater network agility and faster response times to changes in network traffic.

Different Models of SDN:

- While the premise of centralized software controlling the flow of data in switches and routers applies to all software-defined networking, there are following different models of SDN.

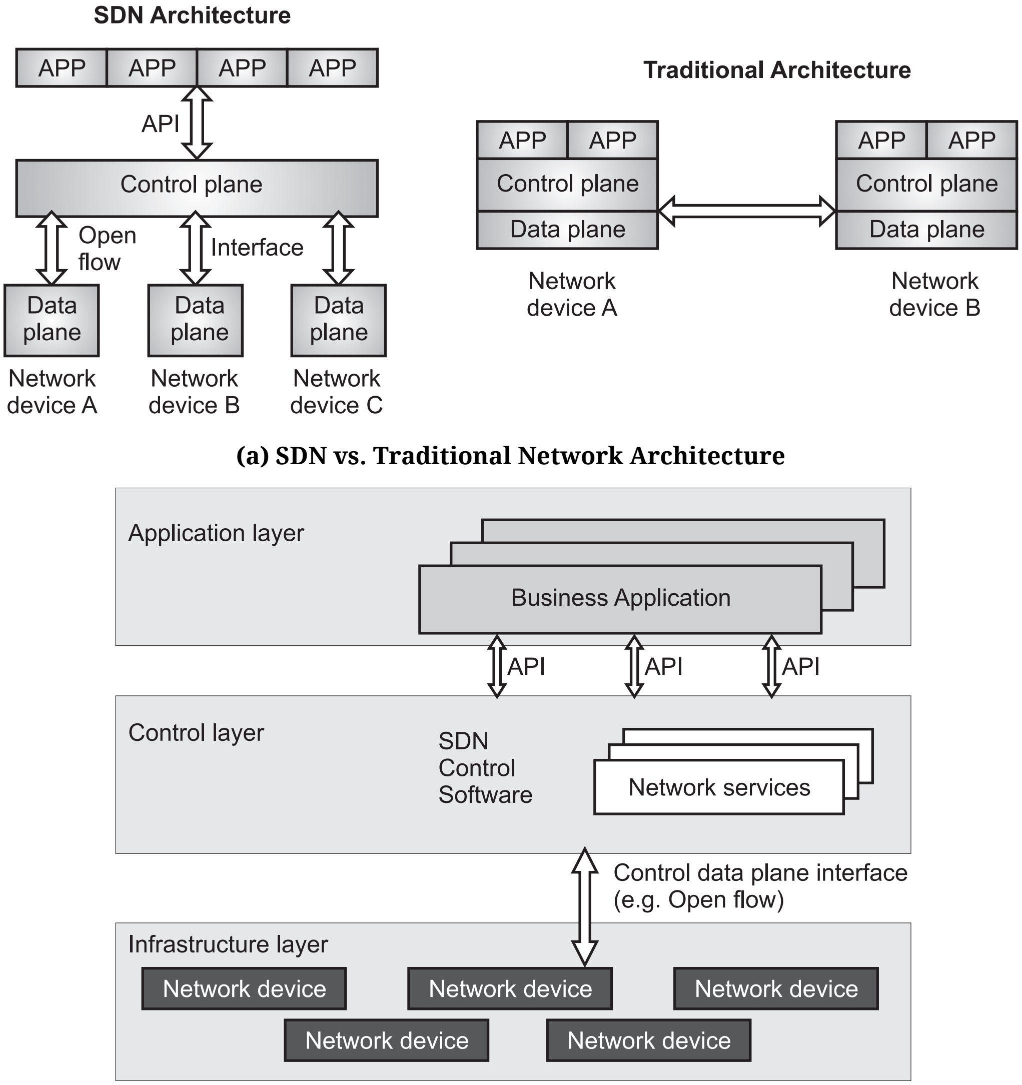
1. **Open SDN:** Network administrators use a protocol like OpenFlow to control the behavior of virtual and physical switches at the data plane level.
2. **SDN by APIs:** Instead of using an open protocol, application programming interfaces control how data moves through the network on each device.
3. **SDN Overlay Model:** Another type of software-defined networking runs a virtual network on top of an existing hardware infrastructure, creating dynamic tunnels to different on-premise and remote data centers. The virtual network allocates bandwidth over a variety of channels and assigns devices to each channel, leaving the physical network untouched.
4. **Hybrid SDN:** This model combines software-defined networking with traditional networking protocols in one environment to support different functions on a network. Standard networking protocols continue to direct some traffic, while SDN takes on responsibility for other traffic, allowing network administrators to introduce SDN in stages to a legacy environment.

5.2.2 Working of SDN

- Software-Defined Networking (SDN) is a modern approach to networking where the control plane is separated from the data plane, allowing centralized, programmable control of network devices.

How SDN Works?

- Devices connect to the network (e.g., a user connects to Wi-Fi).
- The SDN Controller receives a request (e.g., how to route a packet).
- The controller decides the path and installs flow rules in the switches.
- Switches then forward packets according to these rules.
- If there's a network change (like congestion), the controller can re-route traffic instantly.



(a) SDN vs. Traditional Network Architecture

(b) Layers of SDN

Fig. 5.3: Illustrations of SDN

- As illustrated in Fig. 5.3 (b), the essence of the SDN architecture is to well divide the network into three subsystems (usually called layers):
  - **Application Layer:** Provides user applications with programmatic interfaces to the network services regardless of the underlying resources; OpenStack is one implementation of such an interface.

- **Control Layer:** Provides a separate programmable facility for the control and management of the network infrastructure.
- **Infrastructure Layer:** Provides open standards-based interfaces to the basic IT resources (i.e., networks, computing, and storage); OpenFlow is an example.

Advantages of SDN:

- Software-defined networking (SDN) offers several advantages over traditional networking architectures, including:
  1. **Centralized Network Control:** One of the key benefits of SDN is that it centralizes the control of the network in a single controller, making it easier to manage and configure the network. This allows network administrators to define and enforce network policies in a more granular way, resulting in better network security, performance, and reliability.
  2. **Programmable Network:** In an SDN environment, network devices are programmable and can be reconfigured on the fly to meet changing network requirements. This allows network administrators to quickly adapt the network to changing traffic patterns and demands, resulting in better network performance and efficiency.
  3. **Cost Savings:** With SDN, network administrators can use commodity hardware to build a network, reducing the cost of proprietary network hardware. Additionally, the centralization of network control can reduce the need for manual network management, leading to cost savings in labor and maintenance.
  4. **Enhanced Network Security**: The centralized control of the network in SDN makes it easier to detect and respond to security threats. The use of network policies and rules allows administrators to implement fine-grained security controls that can mitigate security risks.
  5. **Scalability:** SDN makes it easier to scale the network to meet changing traffic demands. With the ability to programmatically control the network, administrators can quickly adjust the network to handle more traffic without the need for manual intervention.
  6. **Simplified Network Management:** SDN can simplify network management by abstracting the underlying network hardware and presenting a logical view of the network to administrators. This makes it easier to manage and troubleshoot the network, resulting in better network uptime and reliability.
- Overall, SDN offers a more flexible, programmable, and centralized approach to networking that can result in significant cost savings, enhanced network security, and improved network performance and reliability.

Disadvantages of SDN:

- While software-defined networking (SDN) has several advantages over traditional networking, there are also some potential disadvantages that organizations should be aware of. Here are some of the main disadvantages of SDN:
  1. **Complexity:** SDN can be more complex than traditional networking because it involves a more sophisticated set of technologies and requires specialized skills to manage. For example, the use of a centralized controller to manage the network requires a deep understanding of the SDN architecture and protocols.
  2. **Dependency on the Controller:** The centralized controller is a critical component of SDN, and if it fails, the entire network could go down. This means that organizations need to ensure that the controller is highly available and that they have a robust backup and disaster recovery plan in place.
  3. **Compatibility:** Some legacy network devices may not be compatible with SDN, which means that organizations may need to replace or upgrade these devices to take full advantage of the benefits of SDN.

4. **Security**: While SDN can enhance network security, it can also introduce new security risks. For example, a single point of control could be an attractive target for attackers, and the programmability of the network could make it easier for attackers to manipulate traffic.
5. **Vendor Lock-In:** SDN solutions from different vendors may not be interoperable, which could lead to vendor lock-in. This means that organizations may be limited in their ability to switch to another vendor or integrate new solutions into their existing network.
6. **Performance:** The centralized control of the network in SDN can introduce latency, which could impact network performance in certain situations. Additionally, the overhead of the SDN controller could impact the performance of the network as the network scales.

5.2.3 Applications of SDN

- SDN applications span various domains, including data centers, enterprise networks, service provider networks, and cloud computing, enabling centralized control, improved network visibility and efficient resource management.

1. **Data Centers:**
   - **Fabric Optimization:** SDN facilitates the creation of highly efficient and scalable data center fabrics, enabling rapid application deployment and resource allocation.
   - **Traffic Engineering:** SDN allows for dynamic and centralized traffic engineering, optimizing network performance and ensuring quality of service (QoS).
   - **Security:** SDN enables more targeted and flexible security policies, including distributed firewalls and intrusion detection/prevention.
   - **Virtualization:** SDN simplifies the management of virtualized resources, enabling efficient multi-tenant environments.
2. **Enterprise Networks:**
   - **Campus Networks:** SDN simplifies the management of both wired and wireless networks (Wi-Fi and Ethernet), offering centralized control and automation.
   - **DevOps:** SDN facilitates DevOps practices by automating application deployments and infrastructure changes.
   - **Security:** SDN can improve network security by enabling policy-based segmentation and intrusion detection/prevention.
   - **Application-Aware Networking:** SDN allows for application-level quality of service (QoS) and traffic prioritization, ensuring optimal performance for critical applications.
3. **Service Provider Networks:**
   - **Network Provisioning:** SDN simplifies and automates network provisioning, enabling rapid service deployment and flexible network configurations.
   - **Traffic Management:** SDN enables dynamic traffic management, supporting scalable and flexible service delivery.
   - **Network Slicing:** SDN enables the creation of virtual networks (network slices) to isolate traffic and resources for different services or customers.
4. **Cloud Computing:**
   - **Resource Management:** SDN facilitates the efficient management of virtualized resources in cloud environments, enabling dynamic provisioning and scaling.
   - **Inter-Cloud Networking:** SDN can facilitate the creation of interconnected clouds, enabling seamless data transfer and application deployment across multiple cloud providers.

- **Virtualization:** SDN plays a crucial role in managing virtualized resources in cloud environments, enabling efficient multi-tenant environments and supporting emerging technologies like IoT and 5G networks.
- **Network Function Virtualization (NFV):** SDN enables the virtualization of network functions, such as firewalls and load balancers, allowing for flexible and scalable network services.
- **Cloud Integration:** SDN facilitates the integration of cloud resources with on-premises networks, enabling hybrid cloud deployments and seamless data transfer.
- **IoT and 5G:** SDN plays a crucial role in supporting emerging technologies like IoT and 5G, enabling the management of large-scale and diverse networks.

5.3 NETWORK FUNCTIONS VIRTUALIZATION (NFV)

- In traditional networks, network functions are typically implemented using specialized hardware devices, which can be expensive and difficult to manage.
- For this reason, NFV has become increasingly important in modern networks due to the growing complexity of network infrastructure and the need for more agility, flexibility, and cost-effectiveness.
- In recent years. we have seen the rapid growth of Software-Defined Networking (SDN) in development and production.
- In traditional networking systems, fixed dedicated network devices such as switches. routers are used to control the network traffic.
- However, non-programmable feature, poor network security and performance drawback have created new challenges for future Internet-based information and communication system.
- The complexity in traditional networking makes the system difficult to reconfigure the network to counter faults, load and error.
- To overcome these issues, SDN is taking control over the traditional manually configured network to make proper utilization of physical network infrastructure.
- NFV is a network architecture that virtualizes network functions, like firewalls and load balancers, allowing them to run as software on standard hardware instead of dedicated appliances, leading to greater flexibility, cost savings, and faster service deployment.
- Network Functions Virtualization (NFV) is a modern networking technology that virtualizes network services traditionally performed by dedicated hardware.
- Instead of relying on proprietary devices like routers, firewalls, or load balancers, NFV enables these functions to run as software on industry-standard servers, improving flexibility, scalability, and cost efficiency.
- The purpose of NFV (Network Functions Virtualization) is to decouple the network functions from the network equipment.
- This decoupling enables us to position the software performing the functions of a device on a different machine than the device itself.
- NFV is an approach to network architecture that involves replacing dedicated network hardware devices with software-based Virtualized Network Functions (VNFs) that run on standard servers, storage, and switches.
- In traditional networking, specialized hardware devices such as routers, firewalls, and load balancers are used to perform specific network functions.
- With NFV, these functions are virtualized and can be run as software on commodity hardware, leading to greater flexibility, scalability, and cost-effectiveness.
- NFV enables network operators to quickly and easily deploy and scale network functions as needed, without the need for physical hardware installation or maintenance.

- By virtualizing network functions, operators can also reduce their capital and operational expenditures, and increase their network agility, flexibility, and scalability.
- NFV is often used in conjunction with software-defined networking (SDN), another approach to network architecture that separates the control plane and data plane of the network, allowing for centralized network management and orchestration.
- Together, NFV and SDN enable network operators to build more agile and efficient networks that can adapt to changing business and user requirements.
- The most looked for and promising features of SDN networks are related with:
  - providing centralized control policies, which gives a global view of network configuration and activity as various nodes has different functionalities;
  - capability to dynamically program all features and configuration of network resources conveniently over automated SDN services instead of static manual operation;
  - independent of physical infrastructure as the network administrator can dynamically modify the network traffic flow to meet the changes;
  - implement open standard, which simplifies the network design and operations.

5.3.1 Working of NFV

- Network functions virtualization (NFV) works by virtualizing network functions that were traditionally performed by specialized hardware devices, such as routers, switches, firewalls and load balancers.
- These network functions are abstracted from their underlying hardware and implemented as software-based Virtual Network Functions (VNFs) that run on standard servers, storage, and switches.
- NFV is typically implemented using a virtualization layer that allows multiple VNFs to run on the same physical infrastructure.
- This virtualization layer provides isolation between the VNFs, enabling them to run independently of each other, and also provides resource management and allocation, ensuring that each VNF gets the resources it needs to operate effectively.

5.3.2 Components of NVF Architecture

- The NFV architecture typically includes the following components:
  1. **Virtualization Layer:** This layer provides the virtualization environment that enables multiple VNFs to run on the same physical infrastructure. It includes a hypervisor or container-based virtualization platform that provides isolation, resource allocation, and management for the VNFs.
  2. **Virtual Network Functions (VNFs):** These are the software-based network functions that perform specific network tasks, such as routing, switching, firewalling, load balancing, and encryption. The VNFs are typically deployed as virtual machines or containers and run on the virtualization layer.
  3. **NFV Infrastructure (NFVI):** This is the physical infrastructure that provides the computing, storage, and networking resources needed to support the VNFs. The NFVI can be located in the data centre, at the network edge, or in the cloud.
  4. **Management and Orchestration (MANO):** This component provides, the management and orchestration functions needed to deploy, monitor, and manage the VNFs running on the NFVI. It includes functions such as service orchestration, resource allocation, fault management, and performance management.
- Overall, NFV enables organizations to deploy network functions more quickly and cost-effectively while also providing greater flexibility, scalability, and agility.

- By virtualizing network functions, operators can create, a more dynamic and efficient network architecture that can adapt to changing business and user requirements.

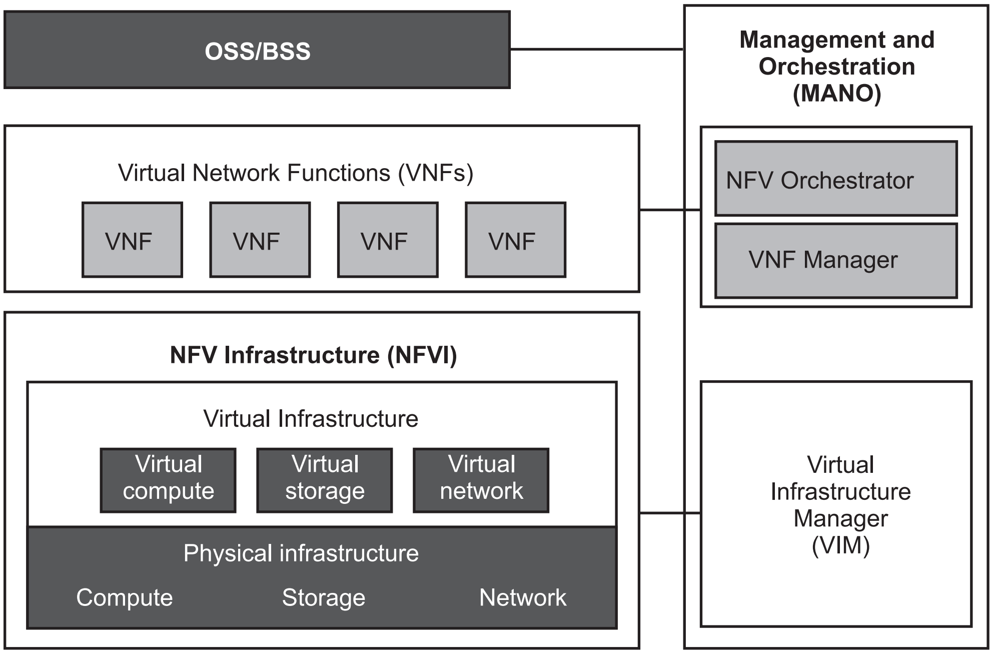


Fig. 5.4: NVF Architecture

Benefits of Network Function Virtualization:

- There are several benefits to using Network Function Virtualization (NFV) in modern network architectures. Here, are a few key reasons why organizations may choose to adopt NFV:
  1. **Cost savings:** NFV can help organizations reduce costs by replacing dedicated hardware devices with software-based virtualized network functions running on commodity hardware. This can help to reduce capital expenditures and operational costs associated with purchasing, deploying, and maintaining specialized hardware.
  2. **Agility and flexibility:** NFV enable network operators to quickly and easily deploy and scale network functions as needed, without the need for physical hardware installation or maintenance. This can help to reduce the time to market for new services and applications, and improve the agility and flexibility of the network.
  3. **Scalability:** NFV can help to improve network scalability by enabling organizations to scale up or down the capacity of virtualized network functions as needed, based on changing traffic patterns and demand.
  4. **Enhanced network security:** NFV can help to improve network security by enabling the deployment of virtualized network functions that can detect and mitigate security threats, such as firewalls, intrusion detection systems, and encryption services.
  5. **Service innovation:** NFV can enable organizations to innovate and introduce new network services and applications more quickly and easily since the virtualized network functions can be easily deployed and scaled as needed.
- Overall, NFV provides a flexible, scalable, and cost-effective approach to network architecture, allowing organizations to build more agile, efficient, and innovative networks that can adapt to changing business and user requirements.

Challenges in Network Function Virtualization:

- While there are many benefits to network functions virtualization (NFV), organizations should consider some risks and challenges before implementing NFV.

- Here are a few potential risks of NFV:
  1. **Complexity:** NFV can add complexity to network architecture, as it involves deploying and managing multiple virtualized network functions on a shared physical infrastructure. This can make troubleshooting issues more challenging and ensure overall network performance.
  2. **Security:** NFV introduces new security risks, as virtualized network functions may be vulnerable to attacks that exploit vulnerabilities in the virtualization software or the underlying hardware infrastructure. Proper security measures must be put in place to mitigate these risks.
  3. **Integration with legacy systems:** Integrating NFV with existing legacy systems and network architectures can be challenging and may require additional investment in new hardware and software to enable compatibility.
  4. **Performance and reliability:** The performance and reliability of NFV may be impacted by the virtualization layer and the underlying hardware infrastructure. Organizations must ensure that they have the right resources in place to support the VNFs and to deliver the desired performance and reliability.
  5. **Vendor lock-in:** Organizations that adopt NFV may become locked into a particular vendor or technology, which can limit their flexibility and ability to switch vendors or technologies in the future.

5.3.3 Applications of NFV

- Network Functions Virtualization (NFV) offers a wide range of applications, primarily focusing on transforming traditional network infrastructure into a more flexible, agile, and cost-effective environment.
- NFV virtualizes various network functions, such as routers, firewalls, and load balancers, onto virtualized infrastructure, enabling on-demand provisioning and scaling of network services.
- The key applications of NFV are:
  1. **Telecom and Network Service Providers:**
     - o **Service Agility and Flexibility:** NFV allows service providers to quickly deploy and scale new network services, enabling them to respond to changing customer demands and market trends.
     - o **Reduced Costs:** By virtualizing network functions, service providers can reduce capital expenditure on hardware and operational costs associated with managing and maintaining physical appliances.
     - o **Improved Time-to-Market:** NFV accelerates the deployment of new services, reducing the time it takes to bring them to market.
     - o **On-Demand Scaling:** NFV enables service providers to scale up or down their network resources as needed, ensuring optimal resource utilization and cost efficiency.
     - o **Network Slicing:** NFV is a key enabler for network slicing, allowing service providers to create multiple isolated virtual networks on a shared physical infrastructure.
     - o **Virtual Customer Premises Equipment (vCPE):** NFV allows for the virtualization of CPE devices, providing more flexibility and control to customers.
  2. **Cloud Data Centers:**
     - o **Dynamic Network Management:** NFV and SDN (Software-Defined Networking) are used in cloud data centers to dynamically manage large-scale, distributed networks, optimizing resource utilization and ensuring high availability.
     - o **Load Balancing and High Availability:** NFV facilitates load balancing and high availability by allowing virtual network functions to be easily scaled and relocated as needed.

3. **Enterprise Networks:**
   - **Virtualized Firewall and Intrusion Detection:** NFV can virtualize firewalls and intrusion detection systems, providing enhanced security and flexibility.
   - **Virtualized Routers and Load Balancers:** NFV allows for the virtualization of routers and load balancers, improving network performance and resource utilization.
   - **WAN Optimization:** NFV can be used to virtualize WAN optimization technologies, improving network performance and reducing bandwidth costs.
   - **Cloud-based Applications:** NFV supports cloud-based applications by providing the necessary network functions in a virtualized environment.
4. **General Applications:**
   - **Network Monitoring:** NFV can be used to virtualize network monitoring tools, providing comprehensive visibility into network performance and traffic patterns.
   - **Security Functions:** NFV enables the deployment of various security functions, such as firewalls, intrusion detection systems, and VPNs, in a virtualized environment.
   - **Network Function as a Service (NFaaS):** NFV enables service providers to offer network functions as a service, providing customers with on-demand access to a wide range of network capabilities.

Difference between SDN and NFV:

Features	SDN	NFV
Scope	SDN is primarily focused on the control and management of network traffic flows.	NFV is focused on the virtualization and management of network functions.
Functionality	SDN separates the control plane (which determines how traffic is routed) from the data plane (which handles the actual transmission of data), allowing for more flexible and programmable network management.	NFV virtualizes network functions such as routing, switching, firewalling, and load balancing, allowing these functions to be deployed and managed as software-based virtual network functions (VNFs).
Deployment	SDN typically requires specialized network hardware, such as switches and routers, that support OpenFlow or other SDN protocols.	NFV can be deployed on standard x86 servers, storage, and switches.
Management and Orchestration	SDN typically relies on centralized controllers that manage and orchestrate network traffic flows.	NFV also requires management and orchestration, but this is typically focused on the deployment and management of VNFs.
Standards	SDN is primarily defined by the Open Networking Foundation (ONF) and the OpenFlow protocol.	NFV is defined by the European Telecommunications Standards Institute (ETSI) and its NFV Industry Specification Group (ISG).
	Note: Both technologies are based on open standards, there are some differences in the specific standards and protocols used by each.	

Contd...

Network Architecture	SDN is typically used to create a centralized, software-defined network architecture that is more programmable and easier to manage.	NFV, on the other hand, is focused on virtualizing network functions to create a more flexible and scalable network architecture.
Network Abstraction	SDN abstracts the network infrastructure from the control plane, allowing network administrators to define network policies and configurations that are separate from the underlying hardware.	NFV abstracts network functions from the underlying hardware, allowing them to be deployed and managed independently of the physical infrastructure.
Service Delivery	SDN can be used to enable new service delivery models, such as network slicing, that allow network resources to be allocated dynamically based on the needs of specific applications or services.	NFV can also enable new service delivery models by allowing network functions to be deployed and scaled up or down based on demand.
Vendor Ecosystem	SDN has a larger and more mature vendor ecosystem than NFV, with a wide range of products and solutions available from established networking vendors as well as startups.	NFV is still a relatively new technology, and the vendor ecosystem is still evolving.

5.4 EDGE COMPUTING AND EDGE NETWORKING

- Edge computing and edge networking are related technologies, but distinct. Edge computing focuses on processing data and performing actions closer to the source, reducing latency and improving efficiency.
- Edge networking refers to the infrastructure and protocols that enable this processing at the edge of the network, often involving technologies like 5G and specialized hardware.
- Edge computing is a distributed computing paradigm that brings computation and data storage closer to data sources.
- This approach minimizes latency, enhances data processing speed, optimizes network bandwidth usage, and ensures data privacy and compliance.
- An edge network is a distributed network architecture designed to provision computing resources at the "edge" of the network.
- It places processing power and data storage closer to devices, reducing reliance on centralized data centers or cloud computing services.
- This architecture aims to improve response times and overall application performance, especially for latency-sensitive and data-intensive applications.

5.4.1 Edge Computing

- Edge computing is a networking philosophy focused on bringing computing as close to the source of data as possible in order to reduce latency and bandwidth use.
- In simpler terms, edge computing means running fewer processes in the cloud and moving those processes to local places, such as on a user's computer, an IoT device, or an edge server.
- Bringing computation to the network's edge minimizes the amount of long-distance communication that has to happen between a client and server.

- Several open source systems for Edge computing have been developed and deployed. Some of them are explained below:
  - **Apache Edgent** is a programming model that can process streams of data locally at the Edge devices or gateways in real-time. Data is determined by Edgent to be stored or analysis at Edge device or back-end systems. Edgent enables the application to transform to sending only essential and useful data to the server from sending the continuous raw data flow.
  - **EdgeX Foundry** is a vendor-neutral open interop platform for the loT and Edge computing. It is an interoperability framework hosted by the Linux Foundation within a full hardware/OS platform. The interested parties of Edge computing can collaborate on IoT solutions freely using current communication standards.
  - **Azure IoT Edge** moves the data analytics from the cloud to the Edge devices. Three components together make up the Azure IoT Edge including the IoT Edge modules, the loT Edge runtime, and the cloud-based interface. The loT Edge runtime enables the cloud logic on Edge devices to manage the communications and operations.
  - **OpenStack** is a cloud operating system which use a data-center to control compute, storage, and networking resources. It also provides management tool through a dashboard, as well as a web interface to the users. The fundamental infrastructure of OpenStack can be deployed at Edge devices and the distribute (software of OpenStack provide support for virtual machines and container technologies, which are vital technologies enable Edge computing.
- Fig. 5.5 shows the two-way computing streams in Edge computing.
- In the Edge computing paradigm, the things not only are data consumers but also play as data producers.
- At the edge, the things cannot only request service and content from the cloud but also perform the computing tasks from the cloud.
- Edge can perform computing offloading, data storage, caching and processing, as well as distribute request and delivery service from cloud to user.
- With those jobs in the network, the edge itself needs to be well designed to meet the requirement efficiently in services such as reliability, security, and privacy protection.

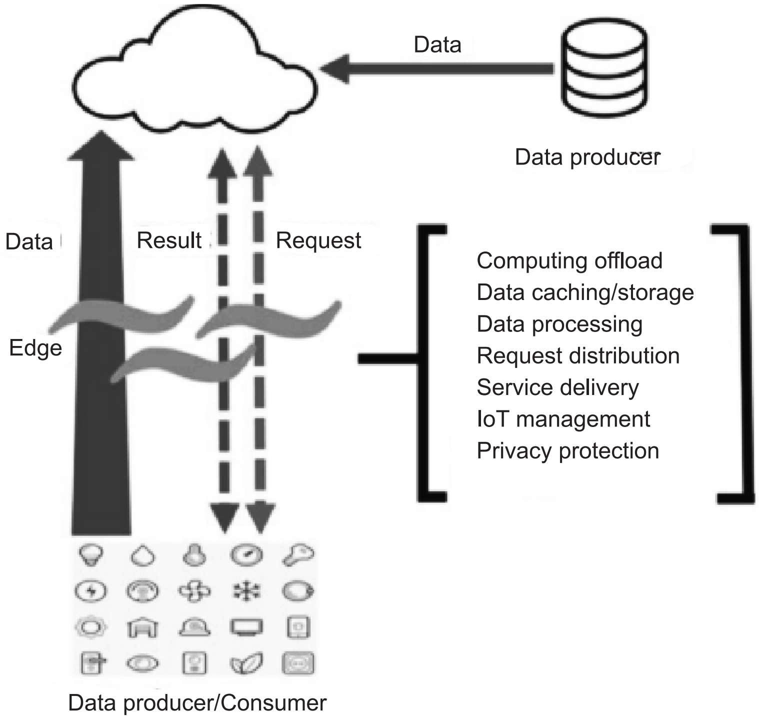


Fig. 5.5 : Edge Computing Paradigm

Definition of Edge Computing:

- Edge computing is a method of optimizing cloud computing systems by performing data processing at the edge of the network near the source of the data.
- Bringing data processing and storage closer to the devices and users that generate and consume the data.
- Edge computing refers to the enabling technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of loT services.
- Here we define "Edge" as any computing and network resources along the path between data sources and cloud data centers.
- For example, a smartphone is an edge between body things and cloud, a gateway in a smart home is the edge between home things and cloud.
- Edge computing involves processing data and running applications at the "edge" of a network, which is closer to the user or the device generating the data, rather than relying solely on a central data center or cloud.

Why it's important:

1. **Reduced Latency:** By processing data locally, edge computing minimizes the time it takes for data to travel across the network, leading to faster response times and real-time decision-making.
2. **Improved Efficiency:** Processing data closer to its source can reduce bandwidth usage and network congestion, making data processing more efficient.
3. **Enhanced Security:** Storing and processing sensitive data locally can improve security by reducing the risk of data breaches during transmission.
4. **Real-Time Applications:** Edge computing is particularly well-suited for applications that require immediate data processing and analysis, such as autonomous vehicles, smart cities, and industrial automation.

Examples:

- **IoT Devices:** Sensors and other IoT devices can process data locally before transmitting it to a cloud server.
- **Local Servers:** Edge servers can be deployed in various locations, such as factories or retail stores, to process data locally.

Components of Edge Computing:

- The various layers of a generic edge computing model are shown in Fig. 5.6.
- The various layers of a generic edge computing model are explained below:
  1. **Perception Layer:** It consists of the edge devices. An edge device is a special-purpose cost-effective hardware designed to perform a specific task effectively. The edge device has limited compute/storage resources. Some examples of such devices include transducers, sensors, actuators, logs, and cameras that perform functionality of gathering and/or transmitting data. Some edge devices have processing power to do additional activities. Analytics applications on image, video, text data gathered by these devices need to be deployed and managed based on the resource availability.
  2. **Networking Layer:** It is responsible for connecting devices, edge systems and cloud systems. It comprises various communication and data transfer protocols.
  3. **Edge Computing Layer:** The major components in this layer include the edge server and gateways. An edge server has higher computational/storage capacity when compared to an IoT device. Workload that cannot be carried out in a resource-constrained edge device can be done

using these general-purpose compute nodes or a cluster of nodes. Edge gateways in this layer perform specialized network functions including protocol mapping, network termination, and tunneling and firewall protection.

4. **Application or Processing Layer:** This layer is responsible to carry out complex data-intensive tasks and to store voluminous data. Private or public cloud servers in this layer act as repositories for sharing data among the nodes. Cloud servers perform complex data-intensive/compute-intensive applications that cannot be hosted in edge servers. Cloud servers can also be used for resource allocation and job management at the edge nodes.

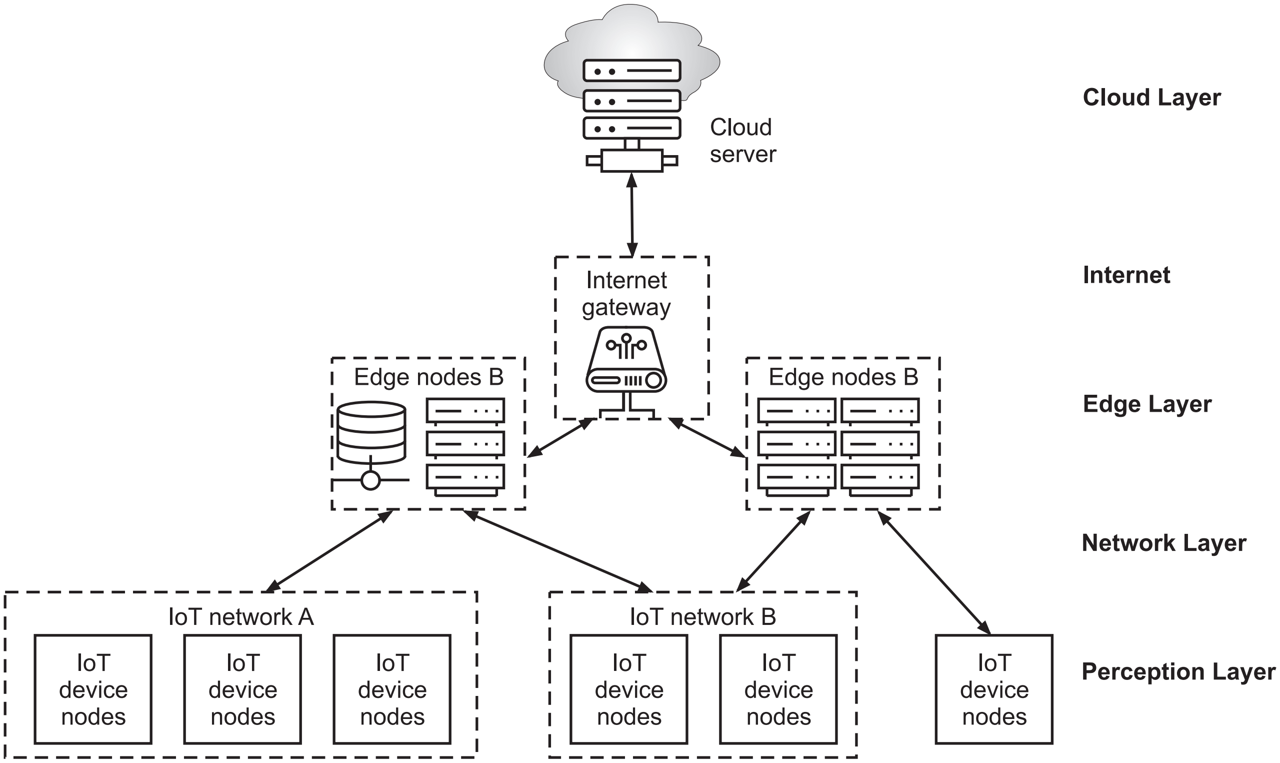


Fig. 5.6: High-level Architecture of an Edge Computing System

- The main components of edge computing are:

1. **Edge Devices**:

○ These are the physical devices or sensors that generate data at the edge of the network. Examples include IoT devices, sensors, cameras, smart devices, industrial machines and autonomous vehicles.

○ They can collect, process, and transmit data to other parts of the system.

2. **Edge Gateways**:

○ Edge gateways act as intermediaries between the edge devices and the cloud or centralized data centers.

○ They typically aggregate data from multiple devices, perform some local processing (such as filtering or aggregating), and then transmit relevant data to the cloud or other centralized systems for further processing.

3. **Edge Servers**:

○ These are more powerful computing resources located closer to the edge of the network that perform intensive processing and analysis of data.

○ Edge servers can execute complex computations and run applications with low latency. They often handle tasks like machine learning, data processing, and analytics, which may not need to be done in the cloud.

4. **Local Data Storage**:

- In some cases, edge devices or gateways need local storage to save data temporarily. This is especially important when there is intermittent connectivity or the data needs to be processed before being sent to the cloud.
- Local storage is often used to store the data that may be used for real-time analysis or for synchronization when connectivity is restored.

5. **Network Infrastructure**:

- A reliable, low-latency network is crucial for edge computing to transmit data between devices, gateways, edge servers, and cloud-based resources.
- This infrastructure can be made up of 4G/5G networks, Wi-Fi, or other wireless technologies that connect edge devices to the cloud and other resources.

6. **Edge Computing Platform/Management Software**:

- This software enables the management, orchestration, and control of edge computing resources and workloads.
- It includes features for provisioning, monitoring, managing devices, security, and enabling edge-to-cloud communication.
- Examples of such platforms include Microsoft Azure IoT Edge, AWS IoT Greengrass, and Google Edge AI.

7. **Cloud or Centralized Data Center**:

- While edge computing is designed to reduce reliance on centralized cloud resources, cloud infrastructure is still part of the overall architecture.
- It provides broader computational power, storage, and services for processing data that cannot be handled locally.
- Data from edge devices is sent to the cloud for long-term storage, deep analytics, and to gain insights that might not be possible on the edge.

8. **Security Components**:

- Security is crucial in edge computing due to the distributed nature of the infrastructure.
- It includes secure communication protocols, encryption, identity and access management, device authentication, and regular security updates.
- Ensuring the integrity of the data and the privacy of users is critical, particularly when edge devices are often deployed in remote or vulnerable locations.

Advantages of Edge Computing:

1. **Speed:** Edge computing reduces latency by facilitating IoT edge computing devices to process data locally or in nearby edge data centers without accessing the data centers of traditional cloud computing.
2. **Security:** Centralized cloud computing systems are vulnerable to distributed denials of services or attacks and power outages. On the contrary, edge computing systems are more secure as they distribute the processing among multiple devices and data centers. However, the distributed nature of edge computing has many entry points of malware, but they can be easily sealed by the respective edge data center as edge computing allows functioning individually.
3. **Scalability:** It is much easier to extend the edge computing systems as per the growing business needs as companies can easily add the IoT devices and edge data centers according to the future needs, unlike cloud computing, wherein large data centers need to be established for accommodating future needs.
4. **Versatility:** Edge computing is based on edge and IoT devices that are always on and interacting with edge data centers making it a versatile system for many applications where network

reachability is an issue. Here unlike cloud computing, the user need not log in and interact with the central cloud system and wait for a response.

5. **Reliability:** Edge computing is based on the principle of distributed computing. Even though some edge data centers have failed, still, edge devices are capable of performing the processing to some extent. Thus, edge computing systems are more reliable compared with their counterparts.
6. **Cost Savings:** Bandwidth and cloud resources are finite and cost money. With every household and office becoming equipped with smart cameras, printers, thermostats, and even toasters, Statista predicts that by 2025 there will be over 75 billion IoT devices installed worldwide. In order to support all those devices, significant amounts of computation will have to be moved to the edge.
7. **New Functionality: E**dge computing can provide new functionality that wasn't previously available. For example, a company can use edge computing to process and analyze their data at the edge, which makes it possible to do so in real time.

Challenges of Edge Computing:

- Edge computing brings numerous benefits, but it also faces several challenges. Here are some of the main challenges associated with implementing and scaling edge computing:

1. **Security and Privacy Risks:**

- o **Decentralized Nature**: Edge computing often involves a large number of distributed devices and edge nodes that can be vulnerable to attacks. Ensuring the security of these devices, especially those deployed in remote locations, can be complex.
- o **Data Privacy**: With sensitive data being processed at the edge, protecting personal information and ensuring compliance with privacy regulations (e.g., GDPR) can be difficult. Edge devices may also lack robust security features found in centralized systems.
- o **Attack Surface Expansion**: As more devices are connected to the edge, the attack surface increases, providing more potential entry points for cyberattacks.

2. **Device and Network Management:**

- o **Heterogeneity of Devices**: Edge computing involves a wide variety of devices, including sensors, smart devices, and industrial machines. Managing the different types of hardware, operating systems, and protocols can be complex.
- o **Remote Management**: Edge devices are often deployed in remote locations, making them harder to access and maintain. Managing, updating, and troubleshooting devices in such locations can be challenging, especially when dealing with large-scale deployments.
- o **Network Reliability**: Edge devices typically rely on wireless networks like 4G/5G or Wi-Fi, which may not always be reliable. In areas with poor connectivity or high latency, maintaining a stable communication link with the cloud or other devices can be problematic.

3. **Scalability:**

- o **Resource Constraints**: Many edge devices have limited computing resources (e.g., CPU, memory, storage). As edge computing networks scale, managing the computational demands and resource allocation can become more difficult.
- o **Interoperability**: As the number of edge devices increases, ensuring that they can communicate seamlessly with each other and with cloud platforms can be challenging. There may be compatibility issues between different vendors' devices, software, and protocols.

4. **Data Synchronization and Consistency:**

- o **Distributed Data**: Edge computing generates large volumes of data from multiple sources. Ensuring that data is synchronized and consistent across various edge nodes and the cloud can be a complex task.

- **Latency**: In some cases, the need for real-time data synchronization can cause delays and affect system performance. This is particularly critical for applications like autonomous vehicles, healthcare monitoring, or industrial automation, where real-time responsiveness is crucial.

5. **Latency and Quality of Service:**

- **Real-time Processing**: While edge computing aims to reduce latency by processing data closer to the source, ensuring that all devices and systems can meet low-latency demands in real-time can be challenging, especially in large-scale deployments.
- **Quality of Service (QoS)**: Providing consistent QoS across a distributed network of edge devices can be difficult due to variability in network performance, processing power, and device capabilities.

6. **Power Consumption:**

- **Edge Device Limitations**: Many edge devices are battery-powered or rely on energy-efficient sources. Managing power consumption, particularly for long-term deployments in remote areas, can be a challenge, especially when devices need to operate continuously or handle complex computations.
- **Balancing Performance and Power**: Finding the right balance between high performance (e.g., intensive processing or high throughput) and low power consumption is crucial for sustainable edge computing systems.

Applications of Edge Computing:

1. **Autonomous Vehicles:** Edge computing enables real-time processing of sensor data for navigation and decision-making. Vehicles can exchange real-time sensory data and corroborate and enhance choices with fewer onboard resources, minimizing the rising cost of autonomous AI systems.
2. **Smart Cities:** Edge computing facilitates real-time monitoring and control of city infrastructure, such as traffic lights and public transportation.
3. **Industrial Automation:** Edge computing enables real-time monitoring and control of industrial processes, such as manufacturing and logistics.
4. **Healthcare:** In a hospital, edge nodes can be used to analyze data from multiple monitoring devices to ensure that analysis is completed in a timely manner.
5. **Telecommunications:** A communications service provider can give real-time statistics for video streaming via edge nodes such as a 5 G router.
6. **Transport and traffic monitoring:** Edge computing enables optimized utilization of public transport through route selection, managing extra lanes, and deciding bus and train frequencies. Real-time traffic data can be transferred to edge devices, and traffic control can be obtained without transferring this data to the centralized cloud.
7. **Smart homes:** Edge computing has enabled the processing and storage of home data acquired through home sensors around the home only, reducing latency and helping to get faster responses for user commands.
8. **IoT devices:** Smart devices that connect to the Internet can benefit from running code on the device itself, rather than in the cloud, for more efficient user interactions.

5.4.2 Edge Networking

- Edge Networking is a networking concept where data processing and computing happen closer to the source of data i.e., at the "edge" of the network - rather than relying entirely on a centralized data center or cloud.

- Instead of sending all the data to a distant server to be processed, edge networking allows devices (like sensors, smartphones, routers, etc.) near the data source to do some or all of the processing themselves.
- **Example:** Imagine a self-driving car. It generates a massive amount of data every second. If all that data had to be sent to a distant cloud server for processing, it would cause delays - not ideal for making split-second decisions. Instead, with edge networking, much of that data is processed locally within the car itself or nearby edge servers.

Need for Edge Networking:

- In traditional cloud-based architectures, data travels to centralized data centers for processing, which can result in delays and inefficiencies, especially for real-time applications.
- Edge networking shifts this approach by placing computing power at the 'edge' of the network, which could be in proximity to IoT devices, mobile users or data-producing sensors.
- This decentralization enables quicker data analysis, faster response times and greater efficiency, making it suitable for applications like autonomous vehicles, industrial automation, and augmented reality.

Definition of Edge Networking:

- Edge networking refers to a distributed computing paradigm that brings computational resources closer to the data source or end-users, reducing latency and improving overall performance. **OR**
- Edge networking is often referred to as a collection of multiple technologies that support connectivity for edge locations, edge devices and edge computing.
- The network infrastructure and protocols that enable edge computing. Edge networking is particularly beneficial for applications like:
  - **Internet of Things (IoT):** Processing sensor data locally enables real-time analysis and control of IoT devices.
  - **Artificial Intelligence (AI):** Edge devices can perform initial AI processing, reducing the need to send all data to the cloud.
  - **5G:** Edge networking is crucial for enabling low-latency and high-speed communication in 5G networks, supporting applications like autonomous vehicles and augmented reality.

Components in Edge Networking:

- The main components used in Edge Networking:

1. Edge Devices:

- These are the devices that generate and sometimes process data at the edge of the network.
- Devices that collect data in the site including sensors, cameras, microphones, recorders, Web monitors, data loggers can function as edge devices. Edge devices can also be actuators and visualization tools.
- Objective of the edge device is to collect information and send upstream or transfer information downstream for actuation.
- Examples of Edge Devices:
  - Sensors
  - IoT devices (like smart thermostats, wearables)
  - Smart cameras
  - Industrial machines
  - Mobile phones
  - Autonomous vehicles

2. Edge Nodes / Edge Servers:

- These are small-scale computing resources placed close to the edge devices. They collect data from the edge devices and process it locally.
- Examples:
  - Mini data centers
  - Local gateways
  - On-premise servers
  - Routers with computing power

3. Network Infrastructure:

- The physical and wireless communication systems that connect edge devices to each other and to the cloud/data center.
- Examples:
  - Wi-Fi
  - 5G/4G cellular networks
  - Ethernet
  - Fiber optics
  - LPWAN (Low-Power Wide-Area Network)

4. Cloud or Central Data Center:

- Even though edge networking brings processing closer to the source, cloud infrastructure still plays a role for:
  - Data storage
  - Heavy processing tasks
  - Centralized control and analytics

5. Security Components:

- To secure the data being processed and transmitted across edge networks.
- Examples:
  - Firewalls
  - Encryption modules
  - Intrusion Detection/Prevention Systems (IDS/IPS)
  - Authentication systems

6. Software and Middleware:

- To manage communication, processing, and analytics between edge and cloud.
- Examples:
  - Edge operating systems (e.g., Azure IoT Edge, AWS IoT Greengrass)
  - Data analytics software
  - Device management platforms

Advantage of Edge Networking:

1. **Higher Performance.** By bringing data closer to the end-user, edge networking speeds up data transmission to devices, minimizing frustrating delays.
2. **Real-time Insights.** Edge networking eliminates lag during critical moments, enabling instant network connections for real-time action.
3. **More Reliable Networks.** Reduce the chance of network disruption with edge networking, creating more dependable digital experiences.

4. **Support for Emerging Technologies.** Enterprises leveraging edge networking can confidently use technologies like IoT and augmented reality. These technologies depend on real-time data transfers and low-latency connectivity to succeed.
5. **Smoother User Experiences.** Ultimately, all of the above benefits stream into user experience. Bringing computing closer to the user means seamless, responsive digital experiences.

Challenges in Edge Networking:

1. **Security and Privacy Risks:**

- Data is processed closer to the user, often on less secure devices.
- Increases the chance of hacking, data leaks, or tampering.
- Each edge device must be protected and there can be thousands of them!

2. **Device Management:**

- Managing and updating a large number of edge devices can be complex.
- Devices may be spread across different locations with different configurations.

3. **Data Consistency and Synchronization:**

- When some data is processed locally and some in the cloud, keeping everything in sync can be difficult.
- Risk of data duplication or loss.

4. **Connectivity Issues:**

- Edge devices often rely on wireless networks, which may be unstable or slow.
- If network goes down, some real-time tasks might fail unless backup systems exist.

5. **Limited Processing Power:**

- Edge devices (like sensors or gateways) usually have less computing power and storage than cloud servers.
- They may not be able to handle heavy processing tasks.

6. **Integration with Existing Systems:**

- Many companies already have cloud-based systems.
- Adding edge computing to existing infrastructure may require significant changes.

7. **Cost and Maintenance:**

- Setting up and maintaining edge nodes and devices can be costly.
- Regular hardware maintenance is needed, especially for outdoor or remote locations.

8. **Skilled Manpower Required:**

- Edge networking involves complex technologies like AI, IoT, security, and real-time systems.
- There’s a shortage of trained professionals who can manage it effectively.

Applications of Edge Networking

1. **Autonomous Vehicles (Self-driving Cars):**

- Self-driving cars need to process data in real time from sensors, cameras, and GPS.
- Edge computing helps the car make instant decisions without waiting for cloud responses (like braking or steering).

2. **Healthcare and Remote Patient Monitoring:**

- Devices like smartwatches or health sensors can monitor heart rate, glucose levels, etc.
- Edge computing allows for real-time alerts if something is wrong, like detecting a heart attack early.

3. **Smart Cities:**
   - Edge networking powers traffic lights, surveillance cameras, waste management, etc.
   - Helps manage traffic flow, detect suspicious activity, and optimize energy usage in real time.
4. **Industrial Automation (Industry 4.0):**
   - In factories, edge computing helps monitor machines, predict failures, and improve efficiency.
   - Example: A machine can shut itself down immediately if a fault is detected - no cloud needed.
5. **Gaming and Augmented/Virtual Reality (AR/VR):**
   - Real-time processing is key for lag-free gaming and immersive AR/VR.
   - Edge servers near users reduce latency, making the experience smooth.
6. **Smart Homes:**
   - Devices like voice assistants, smart lights, thermostats use edge networking.
   - They can process voice commands or automate tasks locally, even without constant internet.
7. **Agriculture:**
   - Sensors in the field measure soil moisture, temperature, etc.
   - Edge computing helps decide when to water or fertilize crops automatically.
8. **Retail:**
   - Smart shelves and checkout systems use edge to track inventory in real time.
   - Edge analytics can even suggest product restocking or detect shoplifting.
9. **Defense and Aerospace:**
   - Used in drones, military vehicles, and aircraft for instant decision-making.
   - Helps in surveillance, navigation, and targeting with high accuracy and low delay.

Difference between Edge computing and Edge networking:

Feature	Edge Computing	Edge Networking
Definition	Processing data at or near the source of data generation.	The communication infrastructure that connects edge devices.
Main Goal	Reduce latency by minimizing the need to send data to the cloud.	Ensure data can travel efficiently between edge devices and networks.
Focus Area	Computation and data processing.	Data transmission and connectivity.
Key Components	Edge servers, gateways, local devices with processing power.	Routers, switches, edge routers, network protocols.
Example Use Case	A factory floor machine analyzing sensor data locally.	Enabling 5G connectivity to smart traffic lights.
Reduces Load On	Cloud servers and data centers. Cloud servers and data centers.	Backbone networks and central routers.
Latency	Ultra-low, as processing happens locally.	Low, optimized by shorter data travel routes.
Dependency	Relies on edge networking for connectivity.	Relies on edge computing to process and act on data locally.
Common Technologies	AI at the edge, local data analytics, edge containers.	SD-WAN, 5G, edge switches, network slicing.

5.5 MULTIMEDIA WIRELESS NETWORKS

- Multimedia wireless networks are systems that transmit multimedia data like audio, video, and images wirelessly using technologies like Wi-Fi, cellular networks, and satellite communication.
- These networks enable a wide range of applications, from video conferencing and streaming to smart home systems and real-time monitoring.
- Multimedia Wireless Networks are wireless communication systems that can transmit multimedia content such as images, audio, video, data like presentations, animations, etc.)
- These networks are specially designed to handle large data and provide quality service for time-sensitive content like video streaming or live calls.
- Examples of Multimedia Wireless Networks: Wi-Fi (IEEE 802.11), 4G/5G Mobile Networks, Wireless Sensor Networks, (WSNs), Bluetooth, WiMAX (IEEE 802.16), Satellite Communication.
- Streaming is a method of transmitting and playing audio or video content over the internet in real time, without downloading the entire file first.
- Instead of waiting for the full video or song to download, streaming lets you start watching or listening almost immediately, while the rest of the data keeps arriving in the background.
- **Example:** When you watch a movie on YouTube, Netflix, or Spotify, we are streaming. The media plays instantly, even though the full file isn’t on your device.

Examples of Applications:

1. **Smart Home Systems:** Using multimedia wireless networks to connect and control various devices in a home.
2. **Real-time Monitoring:** Using WMSNs to monitor environmental conditions, industrial processes, or security systems.
3. **Video Conferencing:** Enabling real-time video communication over wireless networks.
4. **Streaming Services:** Delivering multimedia content like movies and music over the internet.

5.5.1 Streaming Audio and Video

- Streaming audio and video in multimedia wireless networks involves sending compressed audio and video data over a network for immediate playback, rather than downloading the entire file.
- This allows users to consume media content in real-time or near real-time, with features like pausing, rewinding, and fast-forwarding.
- The process involves encoding, compressing, and transmitting the media data in packets over protocols like HTTP.
- Streaming is a method of viewing video or listening to audio content without actually downloading the media files. Streaming is widely used for on-demand video, live broadcasting, and interactive applications like VoIP and teleconferencing.

5.5.2 Voice-over Internet Protocol (VoIP)

- Voice-over Internet protocol (VoIP) is communications technology that allows users to interact by audio through an Internet connection, rather than through an analog connection.
- VoIP converts the voice signal used in traditional phone technology into a digital signal that travels through the Internet instead of through analog telephone lines.
- VoIP technology allows users to make "telephone calls" through Internet connections instead of through analog telephone lines, which renders these calls effectively free wherever the Internet is available.

- VoIP changed the telecommunications industry by making traditional phone lines and services nearly obsolete and reducing demand for them significantly.
- As access to the Internet has become more widely available, VoIP has become ubiquitous both for personal use and for business use.
- VoIP protocols are a set of rules and specifications that enable voice and multimedia communication over the internet.

How VoIP works?

- VoIP also known as IP telephony, is a set of technologies used primarily for voice communication sessions over Internet Protocol (IP) networks, such as the Internet.
- VoIP enables voice calls to be transmitted as data packets, facilitating various methods of voice communication, including traditional applications like Skype, Microsoft Teams, Google Voice, and VoIP phones.
- VoIP converts the voice into a digital format, compresses it, and sends it over the internet. The VoIP service provider (like internet service provider) sets up the call.
- For phone calls, the conversation is exchanged using small data packets. The internet can send these data packets around the world in less than a second. For internet telephony, these packets travel between your phone and a VoIP provider.
- Voice over Internet Protocol bypasses the telephone company entirely. Wherever you have a broadband internet connection like DSL, cable, or fiber, you can use VoIP. It's a major upgrade from an analog phone system.
- Here are some basic steps involved:
  - Your phone connects to your switch or router in your Local Area Network (LAN).
  - When you dial a telephone number, your IP phone tells your VoIP service provider to call the other party.
  - Your VoIP service establishes the call and exchanges data packets from your IP phone.
  - Your VoIP phone converts these digital signals into the sound you can hear.

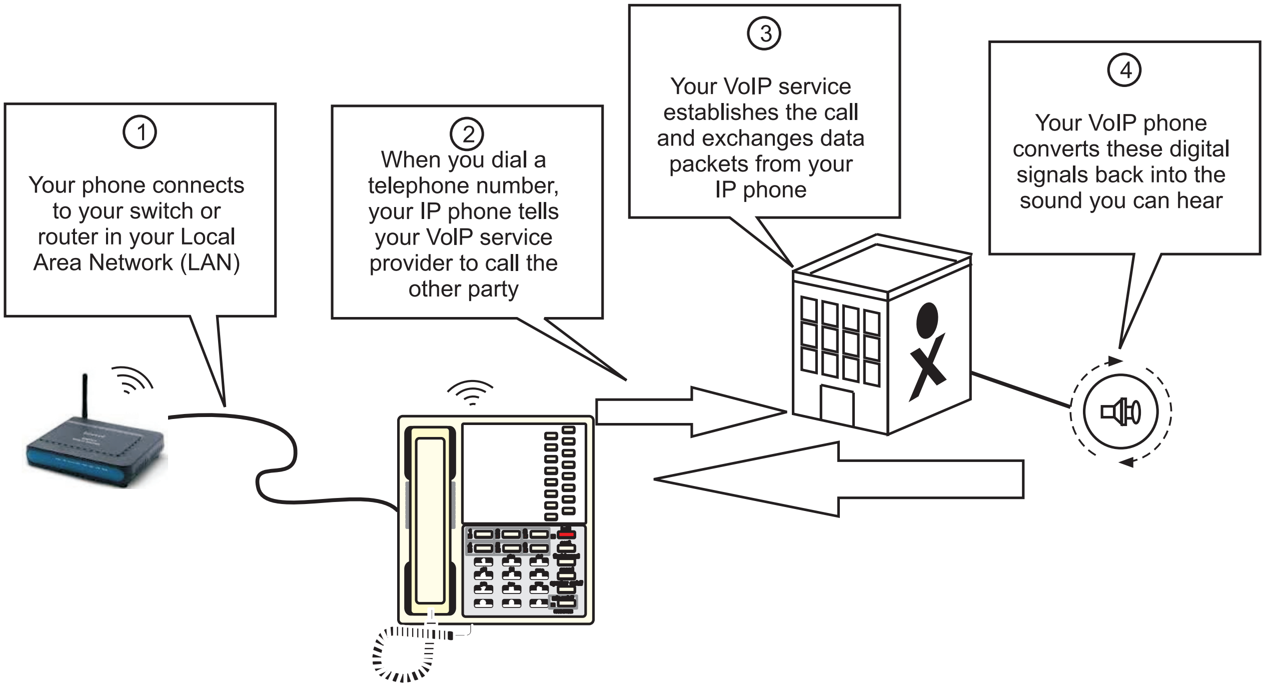

1
Your phone connects to your switch or router in your Local Area Network (LAN)
2
When you dial a telephone number, your IP phone tells your VoIP service provider to call the other party
3
Your VoIP service establishes the call and exchanges data packets from your IP phone
4
Your VoIP phone converts these digital signals back into the sound you can hear

Fig. 5.7: Working of VoIP

5.5.3 VoIP Services

- The first VoIP service was launched in 1995 by a company called VocalTech. The company launched the first Internet-based phone, with the appropriate name InternetPhone.
- This did not come with any video capabilites, and required both users to be logged on to the same software in order to talk.
- Early VoIP services suffered from a poor user experience, with frequent distortions and dropped calls. However, the service steadily improved, until the launch of Skype in 2003 made VoIP attractive and practical for average users.
- This allowed phone calls that were completely free of charge, in addition to video calls and calls to landlines, with much-improved sound quality.
- The COVID-19 pandemic proved another boon for the VoIP industry, as millions of office workers and administrators now began working remotely.
- VoIP and related services such as Zoom became even more critical for office administration, as teleconferencing became the new norm for the average workplace.
- **Examples of VoIP:**
  - Skype
  - WhatsApp
  - Viber
  - Google Hangouts
  - Facebook Messenger

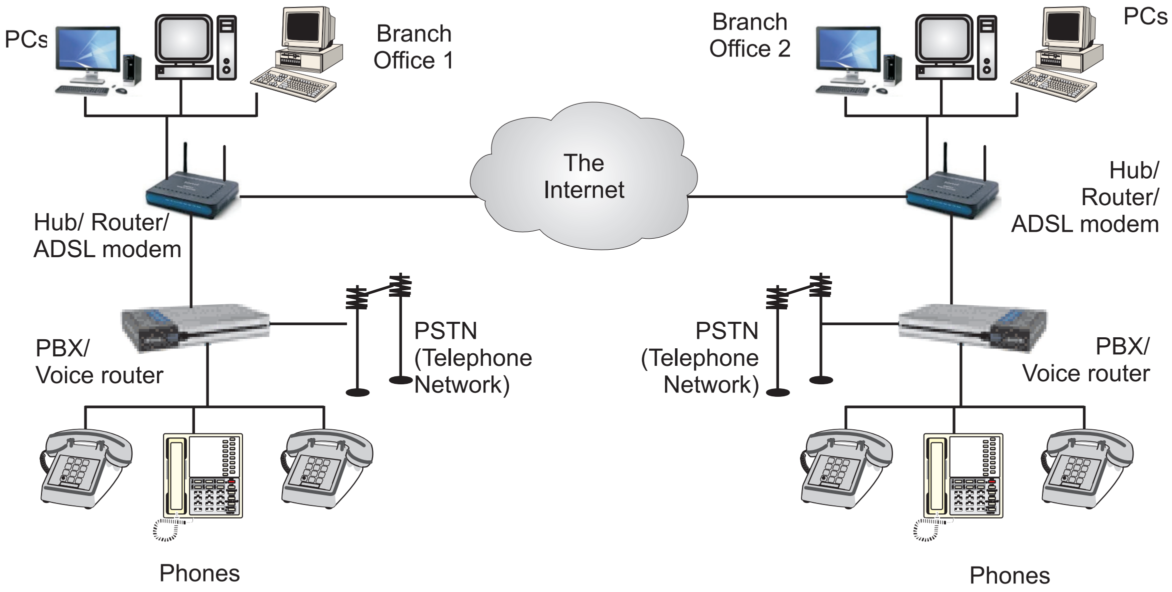


Fig. 5.8

Benefits of VoIP:

1. **Cost-effective:** VoIP calls can be cheaper than traditional phone calls, especially for long-distance or international calls.
2. **Flexibility:** You can make and receive calls from anywhere with an internet connection.
3. **Scalability:** VoIP systems are easy to scale up or down as needed.
4. **Features:** VoIP services often offer additional features such as video calling, instant messaging, and file sharing.

5.5.4 Protocols

- VoIP protocols are sets of rules that enable voice and multimedia communication over the internet. The most common protocols used in VoIP are RTP and RTSP.

5.5.4.1 Real-time Transport Protocol (RTP)

- Real-Time Transport Protocol (RTP) is a network protocol primarily used for delivering audio and video over IP networks in real-time.
- It facilitates the transmission of multimedia data, including live streams, by managing the packaging and delivery of packets.
- RTP is often used with other protocols like UDP and RTCP for reliable and efficient data transfer. The RTP is a network protocol for delivering audio and video over IP networks.
- Real-time Transport Protocol (RTP) is a network protocol for the delivery of audio and video over the internet.
- It is designed to provide end-to-end network transport functions suitable for applications transmitting real-time data, such as audio and video.
- RTP is a network protocol for the delivery of audio and video over the internet. It is designed to provide end-to-end network transport functions suitable for applications transmitting real-time data, such as audio and video.
- RTP is used in conjunction with the Real-time Transport Control Protocol (RTCP), which is used to monitor the quality of the data transmission.
- RTP provides the actual delivery of the media, while RTCP is used to provide feedback on the quality of the transmission and to provide other control information.
- RTP is a packet-based protocol, which means that it breaks the media stream into packets for transmission over the network.
- Each packet is given a sequence number, which allows the receiver to reassemble the packets in the correct order. RTP also includes a timestamp, which allows the receiver to synchronize the audio and video streams.
- RTP is widely used in a variety of applications, including voice over IP (VoIP), video conferencing, and streaming media.
- It is supported by many media players and servers, and it is often used in conjunction with other protocols, such as RTSP and SIP, to deliver audio and video content over the internet.

Features of RTP:

1. **Real-time Delivery:** RTP is designed for applications that require immediate and continuous transmission of data, such as video conferencing, VoIP, and streaming services.
2. **Packetization:** RTP divides multimedia data into packets, which are then transmitted over the network.
3. **Synchronization:** RTP uses sequence numbers and timestamps to help ensure that packets are received in the correct order and can be reassembled at the receiving end.

Benefits of Using RTP:

1. **Real-time Delivery:** RTP ensures that multimedia data is delivered to the receiver in a timely manner, which is crucial for applications like video conferencing and VoIP.
2. **Reliability:** While RTP doesn't guarantee perfect delivery, it provides mechanisms to track packet loss and retransmit if necessary.
3. **Synchronization:** RTP uses sequence numbers and timestamps to maintain the synchronization of the multimedia stream, preventing delays or gaps in the playback.

4. **Scalability:** RTP can be used in both small-scale and large-scale deployments, making it suitable for various applications.

Limitations of RTP:

1. **Security:** RTP itself doesn't provide encryption or authentication, requiring additional protocols to ensure secure transmission.
2. **Complexity:** Implementing RTP can be complex, especially when dealing with multiple streams or complex network topologies.
3. **Packet Loss:** RTP doesn't guarantee 100% packet delivery, which can lead to temporary interruptions in the playback if packet loss is high.

Applications of the Real-time Transport Protocol:

- Real-time Transport Protocol (RTP) is widely used in a variety of applications that require the delivery of real-time audio and video over the internet. Some examples of applications that use RTP include:
  1. **Voice over IP (VoIP):** RTP is commonly used in VoIP systems to transmit audio over the internet.It allows for the real-time delivery of voice calls with low latency.
  2. **Video Conferencing:** RTP is often used in video conferencing systems to transmit audio and video in real time. It allows for the synchronous communication of multiple participants.
  3. **Streaming Media:** RTP is used in many streaming media applications to deliver audio and video over the internet. It is often used in conjunction with other protocols, such as RTSP and HTTP, to stream media to clients.
  4. **Telephony:** RTP is used in many telephony systems to transmit audio and video between devices. It allows for the real-time communication of multiple parties in a call.
  5. **Broadcast Television:** RTP is used in some broadcast television systems to transmit audio and video over the internet. It allows for the delivery of live television streams to viewers.

5.5.4.2 Real-Time Streaming Protocol (RTSP)

- Real Time Streaming Protocol (RTSP) is a network control protocol designed for use in entertainment and communication systems to control streaming media servers.
- The protocol is used to establish and control media sessions between endpoints, and it can support the transmission of video, audio, and other types of data.
- RTSP is similar to HTTP, but it is specifically designed for the control of streaming media. It allows a client to issue commands to a server, such as "play," "pause," and "record," and it can also be used to negotiate the delivery of streaming media.
- RTSP is used in a variety of applications, including internet radio, IPTV, and video-on-demand.
- Real-Time Streaming Protocol (RTSP) is a network control protocol designed to manage the streaming of audio and video content over the Internet or other networks.
- Developed in the late 1990s, RTSP provides applications with the ability to control streaming media sessions with standard commands such as “play,” “pause,” “stop,” and “record.”
- RTSP is an application-layer network control protocol designed to manage and control streaming media sessions between a client and a server.
- It enables real-time transmission of audio and video over IP networks, making it widely used in live streaming, video surveillance, and multimedia applications.

How RTSP Works?

- A client (such as a media player) sends an RTSP request to a server (such as a streaming media server) to establish a media session.
- The server responds with a session description, which includes information about the media being streamed, such as the media format and the transport protocol to be used.

- The client and server exchange RTSP commands and responses to control the flow of the media session. Examples of RTSP commands include "play," "pause," and "record."
- The server streams the media to the client using the agreed-upon transport protocol, such as Real-time Transport Protocol (RTP).
- The client can issue additional RTSP commands to control the media session, such as seeking to a specific point in the media or changing the volume.
- When the media session is finished, the client sends an RTSP "teardown" command to the server to terminate the session.
- RTSP uses TCP (Transmission Control Protocol) as its transport protocol, which provides a reliable connection for the exchange of RTSP commands and responses.
- However, the actual media content is typically delivered using a separate, UDP-based (User Datagram Protocol) protocol, such as RTP. This allows for the efficient delivery of streaming media with low latency.

Components of RTSP:

1. **Clients:** Clients are typically media players or other software that sends RTSP requests to servers in order to establish and control media sessions.
2. **Servers:** Servers are typically streaming media servers that receive RTSP requests from clients and respond with session descriptions and other information. They also stream the media to the client using the agreed-upon transport protocol.
3. **RTSP Requests and Responses:** RTSP uses a set of requests and responses to establish and control media sessions. Examples of RTSP requests include "SETUP," "PLAY," and "TEARDOWN," and examples of RTSP responses include "200 OK" and "404 Not Found."
4. **Transport Protocols:** RTSP uses TCP (Transmission Control Protocol) as its transport protocol for the exchange of RTSP requests and responses. However, the actual media content is typically delivered using a separate, UDP-based (User Datagram Protocol) protocol, such as Real-time Transport Protocol (RTP).
5. **Session Descriptions:** Session descriptions are used to communicate information about the media being streamed, such as the media format, the transport protocol to be used, and the location of the media. Session descriptions are exchanged between the client and server during the setup of a media session.
6. **Media:** The media being streamed is typically audio or video content, although RTSP can also be used to stream other types of data. The media is delivered to the client using the agreed-upon transport protocol.

Advantages of RTSP:

1. Provides real-time interactivity for users.
2. Supports low-latency streaming suitable for live applications.
3. Flexible integration with various codecs and transport protocols.
4. Scalable for both small-scale setups (e.g., home networks) and large-scale deployments (e.g., surveillance systems).

Limitations of RTSP:

1. Not widely supported by modern playback devices compared to newer protocols like HTTP Live Streaming (HLS).
2. Requires additional protocols (RTP/RTCP) for actual data delivery.
3. May face scalability challenges in large-scale environments without proper load balancing.

Applications of Real-Time Streaming Protocol (RTSP):

- RTSP is widely used in various industries and scenarios for real-time audio and video streaming. Below are its key applications:

1. **Video Surveillance Systems:**
   - RTSP is extensively used in security systems to stream video feeds from IP cameras.
   - It enables real-time monitoring, remote playback, and control of surveillance footage, enhancing situational awareness.
   - Users can interact with streams to pause, rewind, or fast forward recorded content.
2. **Live Streaming Services:**
   - RTSP facilitates live broadcasting of events such as sports, concerts, and news.
   - It ensures low latency and real-time delivery, allowing viewers to experience events as they happen.
   - Combined with protocols like RTP, RTSP provides seamless live streaming experiences across platforms.
3. **Video Conferencing and Collaboration:**
   - RTSP is used in video conferencing platforms to enable real-time communication.
   - It supports multimedia sharing, screen sharing, and interactive collaboration among participants.
   - This application ensures smooth audio/video transmission for webinars and remote meetings.
4. **Interactive Education and Training:**
   - RTSP powers virtual classrooms and training sessions by enabling real-time sharing of teaching materials.
   - Features like screen sharing, whiteboard interaction, and live Q&A enhance the learning experience.
   - It allows teachers and students to communicate seamlessly over IP networks.
5. **Multimedia Players and Applications:**
   - RTSP is integrated into multimedia players for streaming audio and video content.
   - Users can control playback with features like pause, rewind, fast forward, and seek.
   - This interactivity improves the user experience in on-demand streaming services.
6. **Content Delivery Networks (CDNs):**
   - RTSP is used in CDNs to deliver high-quality media content efficiently.
   - It supports both unicast (one-to-one) and multicast (one-to-many) streaming for scalability

Comparison between RTP and RTSP:

- Real-Time Transport Protocol (RTP) and Real-Time Streaming Protocol (RTSP) are two key protocols used in multimedia streaming.

Feature	RTP	RTSP
Definition	A transport protocol designed to transmit audio and video data in real time.	A control protocol used to manage and control streaming media sessions between clients and servers.
Primary Function	Handles the actual transmission of multimedia data (e.g., audio, video).	Provides commands for session control (e.g., play, pause, stop, teardown).
Role	Focuses on delivering media packets efficiently with synchronization and jitter control.	Acts as a "remote control" for managing media streams but does not transmit the media itself.

Contd...

Data Transmission	Operates over UDP/IP to ensure low latency in delivering multimedia packets.	Establishes control connections over TCP to manage sessions; uses RTP for actual data transmission.
Session Control	Does not provide session control; only transmits data.	Enables session setup, playback control, and termination using commands like PLAY or PAUSE.
Protocol Interaction	Works alongside RTCP for feedback on stream quality.	Works with RTP to transport media after negotiating session parameters.
Use Cases	Used in live streaming, VoIP (Voice over IP), and video conferencing for transmitting real-time data.	Commonly used for video surveillance (IP cameras), IPTV, and interactive video-on-demand services.
Scenarios	Ideal for transmitting raw multimedia data efficiently across networks.	Suitable for applications requiring user interaction with streams, such as pausing or rewinding content.
Synchronization	Provides mechanisms for synchronizing audio and video streams.	Allows segmented streaming so users can start viewing before full download.
Functionality	RTP focuses solely on transporting multimedia data.	RTSP is responsible for controlling how the multimedia is streamed.
Protocol Dependency	RTP can function independently for raw media transmission.	RTSP relies on RTP (and sometimes RTCP) to handle actual media delivery.
Use Case Focus	RTP is ideal for applications requiring efficient data transfer, such as live broadcasting.	RTSP is better suited for interactive applications like surveillance systems or video-on-demand.

Practice Questions

1. Enlist application layer protocols.
2. What is WWW? Define it?
3. With the help of diagram explain architecture WWW?
4. Define the following terms:
   (i) Web page (ii) URL (iii) Web site (iv) Hypermedia.
5. Enlist types of web pages.
6. What is HTTP? How it Works? Explain its messages diagrammatically.
7. What is meant by file transfer? Which protocols used for file transfer?
8. What is FTP?
9. Explain the working of FTP with diagram.
10. What is e-mail? Give its structure.
11. What is meant by remote login?
12. What is TELNET? Describe in detail.
13. What is edge computing?
14. What is edge networking?
15. Define edge computing and edge networking.
16. List components of edge computing and edge networking.
17. State advantages and applications of edge computing and edge networking.
18. Explain RTSP in deail.

❖❖❖