A switch is a hardware component in network infrastructure that performs the switching process. The switch connects network devices, such as computers and servers, to one another. A switch enables multiple devices to share a network while preventing each device's traffic from interfering with other devices' traffic. The switch acts as a traffic cop at a busy intersection. When a data packet arrives at one of its ports, the switch determines which direction the packet is headed. It then forwards the packet through the correct port for its destination. Some data packets might come to the switch from devices, like computers or voice-over-IP (VoIP) phones, that are attached directly to it. Other data packets might come to the switch from indirectly connected devices, through a network element such as a hub or router. The switch knows which of the network's devices are connected to it, and it can transfer data packets between those devices directly. In other cases, data packets may be going to more-distant destinations, on other networks. A switch in such a scenario forwards the packets to a router, which then forwards them to their destinations on the network. How is a switch different from a hub? Before there were switches, there were network hubs. Hubs enable many devices to plug into a network through a single shared port on a router. The disadvantage of hubs is that when a hub receives a packet, it sends copies of the packet to every other device connected to it. This can cause problems with traffic congestion and data security. Switches solve this problem by keeping tables of the MAC addresses of all devices sending packets to them and forwarding packets only to their destinations, instead of flooding all connected devices with the packets. How is a switch different from a router? A switch connects devices within a LAN (local-area network) by using MAC addresses to identify where to send data packets. A router connects LANs to other area networks or to the internet. A router uses IP addresses to route data packets. How has switching technology evolved? Switches are still at the core of network infrastructure, but today's advanced switches can do much more than just connect devices in a network or IT environment. Most important, advanced switches can act as both switches and routers. Modern Ethernet switches incorporate features and functions that eliminate the need for some types of additional hardware. For example, switches now include security capabilities that were once handled by dedicated firewalls. Also, multigigabit switches can provide variable speeds to match the throughput needs of wireless access points, which provide Wi-Fi access to devices such as laptops and mobile phones. And advancements in Power over Ethernet (PoE) switches can supply devices over copper Ethernet cable with up to 90 watts of power per switch port. Some switches now incorporate machine learning, so they can act as network sensors, collecting data about the network to help network engineers make informed decisions. Modern switches are also programmable and can include network monitoring applications and network analytics tools. Expect advancements in switching to continue to evolve, with switches taking on even more tasks and further increasing efficiency of data transmission across IT networks. Main types of switches Unmanaged switches Unmanaged switches have basic connectivity and limited capacity. They are essentially "plug and play" devices that can be set up to operate without being configured in some way. Unmanaged switches are typically used in small networks that don't have critical requirements for security or availability. Managed switches Networking experts need to configure managed switches, which are designed for use in large, complex networks that demand reliability and security. These switches offer more capacity than unmanaged switches and provide more operational flexibility and control. Managed switches also can provide network analytics, simplify management, and deploy software updates through automation. Fixed switches Fixed switches have fixed numbers of downlink ports8, 12, 24, or 48. Each downlink port connects a device to a fixed switch and can provide power to the devices. Uplink ports are often modular, with interchangeable network modules that allow for upgrades to the overall throughput of the switch. Ports can be connected with fiber cables for higher throughput or copper cables for PoE. Fixed switches are also stackable. They are often stacked in groups, each of which acts as a single switch. When more than 144 ports are needed, modular switches (described below) can be a good option. Modular switches Modular switches are customizable and thus provide more flexibility than fixed switches. These switches often have 4, 7, or 10 slots that hold line cards with various numbers and types of ports. Networking experts can configure the switches to support an organization's networking needs. For example, line cards for access ports, routing, security, and other features can be removed from the slots and replaced with different versions. Also, fan trays and power-supply placement may offer flexibility. These modular capabilities enable future expansion and lifecycle longevity. Explore switches Some capabilities of advanced switches Broader functionality As noted earlier, advanced modern switches take on roles of other network components, such as routers and wireless LAN (WLAN) controllers. Such functionality reduces the need for additional hardware. Electric power Switches also can provide PoE to power devices such as access points, IP phones, LED lighting, security cameras, and video endpoints. Switches that support Universal Power Over Ethernet (UPOE+) can supply power to devices requiring 15W to 90W. Explore UPOE+ High-speed transmission Highly advanced switches, such as multigigabit switches, can provide speeds of 2.5, 5, or 10 gigabits per second or more. Such speeds support the high throughput needed for data transmission under newer wireless standards, such as Wi-Fi 6/6E. Programmability Todays application-centric organizations, which are embracingsoftware-defined networking (SDN), 5G connectivity, Internet of Things (IoT) applications, and more, rely on programmable networks. Software-defined networks have APIs in their infrastructure that developers can use to program applications and other componentssuch as switchesto interact directly with the network. Programmable switches support high-throughput processing of data packets. They also may include advanced features, such as rate limiting, status monitoring, and security, that network teams can control centrally through a programmatic interface. Also, programmable switches can be used to supply enterprise-grade security, automation, segmentation, and management to operational technology (OT) environments, which typically dont have those features and capabilities. Explore intent-based networking Examples of switching use cases Enterprise network switching Switching in an enterprise network enables data transmission among devices in a LAN. Enterprise switches and other components, such as routers and wireless access points, are part of the critical infrastructure that connects to applications and data that are in the data center or the cloud. Explore enterprise switches Data center switching Data centers are evolving rapidly because of exponential data expansion from hyperconnectivity, the growing use of enterprise apps powered by machine learning and artificial intelligence, and the rise ofintent-based networking (IBN). Intent-based networking captures business intent and translates it into automated network configurations. To support modern data centers shaped by these trends, organizations need data center switches that are high-performing, efficient, reliable, scalable, and programmable. Explore data center switches Industrial Ethernet switching As more organizations look to blend their industrial and operational technology (OT) systems and expand their use of IoT applications, they need to extend critical IT networking features, such as enterprise-grade security, and capabilities such as automation to the far edges of the network. The network edges can include remote areas or outdoor locations with harsh environmental conditions where IoT devices and sensors may be deployed. Such locations include roadways, railways, oil fields, and power substations. Industrial Ethernet switches are hardened devices, designed for resilience in rugged environments. They can provide network managers with visibility into networked endpoints as well as the ability to monitor network health from afar. Explore industrial switching Todays communication networking primarily depends on the capability of transmitting data across different networks with ease. Therefore, no matter whether you are sending an email, watching a video using the Internet, or simply browsing a website, the switching protocols are undeniably significant to the efficiency of information deliverance. These protocols are essential in managing the transmission and flow of data from one device to another to enhance the flow of accurate information. This blog will discuss switching protocols, their importance, classification, and where they are assumed in the networking domain. First, lets understand switching protocols and why they are essential in todays networking systems. Switching protocols are the procedures that determine the transmission of data within two or more devices in a network connection. It describes how a packet of information is switched between nodes in the network to reach the intended destination. In laymans terms, a tricky procedure of selecting the best path or route for a data packet is called switching protocols. These protocols are especially applicable in massive network systems that handle thousands or millions of data types per second. There are many circumstances where switching protocols are needed, whether in Local Area Network (LAN) environments, Wide Area Network (WAN) environments, and, in some instances, even in the World Wide Web environment. There exist various switching protocols that are exercised in a network depending on the networks architecture. Switching protocols in networking facilitates the following key functions: Efficient Data Transmission: They determine the shortest and most efficient route for data, minimizing delays. Collision Prevention: In a network, multiple data packets might try to access the same communication channel simultaneously. Switching protocols help prevent collisions, ensuring smooth data flow. Network Scalability: As networks grow, these protocols allow them to handle larger volumes of data without degrading performance. Reliability: Switching protocols ensure that data can be rerouted through an alternative route, maintaining a stable and reliable connection even if one path fails. Now that we understand the importance of switching protocols in networking, lets explore its different types. Switching Protocols can be categorized based on different layers and techniques: Switching protocols are of two types based on the layers they operate, which are: Layer 2 protocols operate on the Data Link Layer and are responsible for transferring data within a Local Area Network. The most common Layer 2 protocol is STP. Layer 3 protocols operate on the network layer and make decisions based on IP addresses. Some Common Layer 3 Switching Protocols are OSPF and BGP. Switching Protocols can be classified into two types based on the techniques they use: Circuit switching establishes a dedicated communication path between two devices for the duration of a session. Its commonly used in traditional telephony, ensuring a continuous, reliable connection, but it can be less efficient with bandwidth. Packet switching breaks data into small packets that travel independently across the network. Its more efficient for internet traffic as packets can take multiple paths, allowing optimal bandwidth use and faster delivery. Here are some common switching protocols that are used today: The Spanning Tree Protocol (STP) is a network protocol used in Ethernet networks to prevent looping issues that can arise from redundant paths. Ethernet networks often include multiple paths between switches or devices to provide backup connectivity, but this redundancy can also create network loops. A loop allows packets to circulate indefinitely, leading to broadcast storms, high CPU usage on network devices, and potentially a total network breakdown. Rapid Spanning Tree Protocol (RSTP) is an enhanced version of the original Spanning Tree Protocol (STP), designed to offer much faster convergence times and improve overall network stability. Like STP, RSTPs primary goal is to prevent loops in Ethernet networks with redundant links, but it achieves this in a way that allows the network to recover almost instantly from link failures or topology changes. Multiple Spanning Tree Protocol (MSTP) is an advanced network protocol that extends the functionalities of Spanning Tree Protocol (STP) by allowing multiple VLANs to be mapped onto a single spanning tree instance. This approach reduces the number of spanning tree instances required, which helps balance network load, improves efficiency, and conserves resources in larger networks. VLAN Trunking Protocol (VTP) is a Cisco proprietary protocol that helps manage VLAN (Virtual Local Area Network) configurations across multiple network switches. VTP simplifies VLAN management by enabling switches to share VLAN information, so VLANs created or modified on one switch can be automatically updated on other switches. This eliminates the need for manual VLAN configuration on each individual switch, which saves time and reduces the chance of errors. GARP VLAN Registration Protocol (GVRP) is a network protocol used for dynamically managing VLAN configurations across network switches. Built on top of the Generic Attribute Registration Protocol (GARP), GVRP allows switches to automatically share VLAN information with one another, enabling them to register and deregister VLANs as needed. This protocol helps reduce the administrative workload of manually configuring VLANs on every switch and promotes consistent VLAN configurations across a network. Switching protocols manage how data packets are forwarded within a network. They guide routers and switches in making decisions based on predefined rules, ensuring that data reaches its destination efficiently and reliably. Reduced Network Congestion: By managing data flow, they reduce bottlenecks. Optimized Data Flow: They ensure data packets are delivered efficiently. Improved Network Scalability: They are ideal for networks that need to grow over time.Configuration Complexity: They require a detailed setup to avoid errors. Looping Issues: Misconfigured protocols may cause data loops, impacting network performance. The future of switching protocols is likely to be influenced by advancements in AI and network automation. AI and Machine Learning can help streamline protocol configurations, reducing manual intervention and enhancing efficiency. L2 (Layer 2) protocols operate at the Data Link layer of the OSI model and are responsible for local network communication, such as Ethernet and ARP. L3 (Layer 3) protocols function at the Network layer, enabling data transmission across different networks, with IP being a key example. A network switch can be an OSI layer 2 switch or data link layer switch, but it can also be an OSI layer 3 or network layer switch. Layer 2 switches forward data based upon the destination MAC address as defined below, while layer 3 switches forward data depending on the IP address of the destination host. Network protocols are classified based on their protocol type and are mainly categorized into three important types: network management protocols, network communication protocols, and network security protocols. Switching protocols operate at lower levels of the network to manage data packets flow within a local area, while routing protocols focus on directing data across larger, often global networks. The 101 Switching Protocols status code means the server acknowledges the clients request to change protocols. It indicates the new protocol the server will use in response to the Upgrade header. Switching protocols are essential for reliable data delivery across varied networks, from early circuit-switched telephone systems to packet-switched internet protocols. These protocols determine the optimal path for data, enhancing speed and efficiency. With advancements in networking, protocols must meet new demands, including low latency, higher speeds, and adaptability to emerging technologies like SDN, NFV, and 5G. As fundamental components in global communications, evolving switching protocols will drive innovation in network interconnections and future communication systems.Switching is a technique of transferring the information from one computer network to another computer network.Let us discuss about switching in step by step manner as follows Step 1 In a computer network the switching can be achieved by using switches.Step 2 A switch is a small piece of hardware device that is used to join multiple computers together with one local area network (LAN).Step 3 These are devices which are helpful in creating temporary connections between two or more devices that are linked to the switch.Step 4 Switches are helpful in forwarding the packets based on MAC addresses.Step 5 By verifying the destination address to route the packet a Switch is used to transfer the data only to the device that has been addressed.Step 6 It will operate in full duplex mode.Step 7 It works with limited bandwidth, so it does not broadcast the message.The diagram given below depicts the switching technique AdvantagesThe advantages of switching are as follows:The bandwidth of the network increases with the help of a switch.It tries to reduce the workload on individual PCs because it always sends the information to specified addressed devices.It increases the overall performance of the network by reducing the traffic on the network.There will be less frame collision because; switch creates the collision domain for each connection. The disadvantages of switching are as follows A Switch is more expensive than network bridges.It cannot determine the network connectivity issues easily.The proper designing and configuration of the switch are required to handle multicast packets.Types of Switching TechniquesThe different types of switching techniques are depicted below Let us understand all these techniques.Circuit SwitchingIn circuit switching a path will be set-up before the transmission of the data. Now the data follows the path specified.For example, telephone lines.Packet SwitchingThe data packets will contain the source and destination addresses. Every router in between will check the destination address, select the next router to which the packet should be forwarded and send it via an appropriate path. As there is no path specified, different packets may follow different paths.Virtual circuit packet switchingThis is a mix of both packet and circuit switching. A path will be set-up logically i.e. no physical path will be set-up. Packets always follow this logical path. Therefore, these are the advantages of both packet and circuit switching.Message SwitchingA message is transferred as a complete unit and that is routed through intermediate nodes at which it is stored and forwarded. Switching is the process of transferring data packets from one device to another in a network, or from one network to another, using specific devices called switches. A computer user experiences switching all the time for example, accessing the Internet from your computer device, whenever a user requests a webpage to open, the request is processed through switching of data packets only. Switching takes place at the Data Link layer of the OSI Model. This means that after the generation of data packets in the Physical Layer, switching is the immediate next process in data communication. Introduction to SwitchA switch is a hardware device in a network that connects and helps multiple devices share a network without their data interfering with each other.A switch works like a traffic cop at a busy intersection. When a data packet arrives, the switch decides where it needs to go and sends it through the right port.Some data packets come from devices directly connected to the switch, like computers or VoIP phones. Other packets come from devices connected through hubs or routers.The switch knows which devices are connected to it and can send data directly between them. If the data needs to go to another network, the switch sends it to a router, which forwards it to the correct destination.What is Network Switching?A switch is a dedicated piece of computer hardware that facilitates the process of switching i.e., incoming data packets and transferring them to their destination. A switch works at the Data Link layer of the OSI Model. A switch primarily handles the incoming data packets from a source computer or network and decides the appropriate port through which the data packets will reach their target computer or network. A switch decides the port through which a data packet shall pass with the help of its destination MAC(Media Access Control) Address. A switch does this effectively by maintaining a switching table, (also known as forwarding table). A network switch is more efficient than a network Hub or repeater because it maintains a switching table, which simplifies its task and reduces congestion on a network, which effectively improves the performance of the network. The switching process involves the following steps:Frame Reception: The switch receives a data frame or packet from a computer connected to its ports.MAC Address Extraction: The switch reads the header of the data frame and collects the destination MAC Address from it.MAC Address Table Lookup: Once the switch has retrieved the MAC Address, it performs a lookup in its Switching table to find a port that leads to the MAC Address of the data frame.Forwarding Decision and Switching Table Update: If the switch matches the destination MAC Address of the frame to the MAC address in its switching table, it forwards the data frame to the respective port. However, if the destination MAC Address does not exist in its forwarding table, it follows the flooding process, in which it sends the data frame to all its ports except the one it came from and records all the MAC Addresses to which the frame was delivered. This way, the switch finds the new MAC Address and updates its forwarding table.Frame Transition: Once the destination port is found, the switch sends the data frame to that port and forwards it to its target computer/network.Types of SwitchingThere are three types of switching methods:Let us now discuss them individually:Message Switching: This is an older switching technique that has become obsolete. In message switching technique, the entire data block/message is forwarded across the entire network thus, making it highly inefficient. Message SwitchingCircuit Switching: In this type of switching, a connection is established between the source and destination beforehand. This connection receives the complete bandwidth of the network until the data is transferred completely.This approach is better than message switching as it does not involve sending data to the entire network, instead of its destination only.Circuit SwitchingPacket Switching: This technique requires the data to be broken down into smaller components, data frames, or packets. These data frames are then transferred to their destinations according to the available resources in the network at a particular time.This switching type is used in modern computers and even the Internet. Here, each data frame contains additional information about the destination and other information required for proper transfer through network components. Packet SwitchingDatagram Packet Switching: In Datagram Packet switching, each data frame is taken as an individual entity and thus, they are processed separately. Here, no connection is established before data transmission occurs. Although this approach provides flexibility in data transfer, it may cause a loss of data frames or late delivery of the data frames.Virtual-Circuit Packet Switching: In Virtual-Circuit Packet switching, a logical connection between the source and destination is made before transmitting any data. These logical connections are called virtual circuits. Each data frame follows these logical paths and provides a reliable way of transmitting data with less chance of data loss.Read about Difference between Datagram Switching and Virtual Circuit Switching Computer Network Tutorial Basics of Computer Networking Types of Computer Networks Introduction to Internet Types of Network Topology Network Devices (Hub, Repeater, Bridge, Switch, Router, Gateways and Brouter) What is OSI Model? - Layers of OSI Model TCP/IP Model Difference Between OSI Model and TCP/IP Model Physical Layer in OSI Model Types of Network Topology Transmission Modes in Computer Networks (Simplex, Half-Duplex and Full-Duplex) Types of Transmission Media Data Link Layer in OSI Model What is Switching? Virtual LAN (VLAN) Framing in Data Link Layer Error Control in Data Link Layer Flow Control in Data Link Layer Piggybacking in Computer Networks Network Layer in OSI Model Introduction of Classful IP Addressing Classless Addressing in IP Addressing What is an IP Address? IPv4 Datagram Header Difference Between IPv4 and IPv6 Difference between Private and Public IP addresses Introduction To Subnetting What is Routing? Network Layer Protocols Session Layer in OSI model Presentation Layer in OSI model Secure Socket Layer (SSL) PPTP Full Form - Point-to-Point Tunneling Protocol Multipurpose Internet Mail Extension (MIME) Protocol Application Layer in OSI Model Client-Server Model World Wide Web (WWW) Introduction to Electronic Mail What is a Content Distribution Network and how does it work? Protocols in Application Layer What is Network Security? Computer Network | Quality of Service and Multimedia Authentication in Computer Network Encryption, Its Algorithms And Its Future Introduction of Firewall in Computer Network MAC Filtering in Computer Network Wi-Fi Standards Explained What is Bluetooth? Generations of wireless communication Cloud Networking Top 50 Plus Networking Interview Questions and Answers for 2024 Top 50 TCP/IP Interview Questions and Answers 2025 Top 50 IP Addressing Interview Questions and Answers Last Minute Notes for Computer Networks Computer Network - Cheat Sheet ...but your activity and behavior on this site made us think that you are a bot. Note: A number of things could be going on here. If you are attempting to access this site using an anonymous Private/Proxy network, please disable that and try accessing site again. Due to previously detected malicious behavior which originated from the network you're using, please request unblock to site. Computer Networks: A Systems Approach In the simplest terms, a switch is a mechanism that allows us tointerconnect links to form a larger network. A switch is a multi-input,multi-output device that transfers packets from an input to one or moreoutputs. Thus, a switch adds the star topology (seeFigure 56) to the set of possible networkstructures. A star topology has several attractive properties:Even though a switch has a fixed number of inputs and outputs, whichlimits the number of hosts that can be connected to a single switch,large networks can be built by interconnecting a number of switches.We can connect switches to each other and to hosts usingpoint-to-point links, which typically means that we can buildnetworks of large geographic scope.Adding a new host to the network by connecting it to a switch doesnot necessarily reduce the performance of the network for other hostsalready connected.Figure 56. A switch provides a star topology.This last claim cannot be made for the shared-media networks discussedin the last chapter. For example, it is impossible for two hosts on thesame 10-Mbps Ethernet segment to transmit continuously at 10Mbpsbecause they share the same transmission medium. Every host on aswitched network has its own link to the switch, so it may be entirelypossible for many hosts to transmit at the full link speed (bandwidth),provided that the switch is designed with enough aggregate capacity.Providing high aggregate throughput is one of the design goals for aswitch; we return to this topic later. In general, switched networks areconsidered more scalable (i.e., more capable of growing to largenumbers of nodes) than shared-media networks because of this ability tosupport many hosts at full speed.Dense Wavelength Division MultiplexingOur focus on packet-switched networks obscures the fact that,especially in wide-area networks, the underlying physical transportis all-optical: there are no packets. At this layer, commerciallyavailable DWDM (Dense Wavelength Division Multiplexing) equipmentis able to transmit a large numbers of optical wavelengths (colors)down a single fiber. For example, one might send data on 100 or moredifferent wavelengths, and each wavelength might carry as much as 100Gbps of data.Connecting these fibers is an optical device called a ROADM(Reconfigurable Optical Add/Drop Multiplexers). A collection ofROADMs (nodes) and fibers (links) form an optical transport network,where each ROADM is able to forward individual wavelengths along amulti-hop path, creating a logical end-to-end circuit. From theperspective of a packet-switched network that might be constructed ontop of this optical transport, one wavelength, even if it crossesmultiple ROADMs, appears to be a single point-to-point link betweentwo switches, over which one might elect to run SONET or 100-GbpsEthernet as the framing protocol. The reconfigurability feature ofROADMs means that it is possible to change these underlyingend-to-end wavelengths, effectively creating a new topology at thepacket-switching layer.A switch is connected to a set of links and, for each of these links,runs the appropriate data link protocol to communicate with the node atthe other end of the link. A switchs primary job is to receive incomingpackets on one of its links and to transmit them on some other link.This function is sometimes referred to as either switching orforwarding, and in terms of the Open Systems Interconnection (OSI)architecture, it is considered a function of the network layer. (This is acase where OSI layering isnt a perfect reflection of the real world,as well see later.)The question, then, is how does the switch decide which output link toplace each packet on? The general answer is that it looks at the headerof the packet for an identifier that it uses to make the decision. Thedetails of how it uses this identifier vary, but there are two commonapproaches. The first is the datagram or connectionless approach.The second is the virtual circuit or connection-oriented approach. Athird approach, source routing, is less common than these other two,but it does have some useful applications.One thing that is common to all networks is that we need to have a wayto identify the end nodes. Such identifiers are usually calledaddresses. We have already seen examples of addresses, such as the48-bit address used for Ethernet. The only requirement for Ethernetaddresses is that no two nodes on a network have the same address. Thisis accomplished by making sure that all Ethernet cards are assigned aglobally unique identifier. For the following discussion, we assumethat each host has a globally unique address. Later on, we considerother useful properties that an address might have, but globaluniqueness is adequate to get us started.Another assumption that we need to make is that there is some way toidentify the input and output ports of each switch. There are at leasttwo sensible ways to identify ports: One is to number each port, and theother is to identify the port by the name of the node (switch or host)to which it leads. For now, we use numbering of the ports. The idea behind datagrams is incredibly simple: You just include inevery packet enough information to enable any switch to decide how toget it to its destination. That is, every packet contains the completedestination address. Consider the example network illustrated inFigure 57, in which the hosts have addresses A,B, C, and so on. To decide how to forward a packet, a switch consultsa forwarding table (sometimes called a routing table), an exampleof which is depicted in Table 5. Thisparticular table shows the forwarding information that switch2 needsto forward datagrams in the example network. It is pretty easy tofigure out such a table when you have a complete map of a simplenetwork like that depicted here; we could imagine a network operatorconfiguring the tables statically. It is a lot harder to create theforwarding tables in large, complex networks with dynamically changingtopologies and multiple paths between destinations. That harderproblem is known as routing and is the topic of a later section. Wecan think of routing as a process that takes place in the backgroundso that, when a data packet turns up, we will have the rightinformation in the forwarding table to be able to forward, or switch,the packet.Figure 57. Datagram forwarding: an example network.Table 5. Forwarding Table for Switch 2.DestinationPortA3B0C3D3E2F1G0H0Datagram networks have the following characteristics:A host can send a packet anywhere at any time, since any packet thatturns up at a switch can be immediately forwarded (assuming acorrectly populated forwarding table). For this reason, datagramnetworks are often called connectionless; this contrasts with theconnection-oriented networks described below, in which someconnection state needs to be established before the first datapacket is sent.When a host sends a packet, it has no way of knowing if the networkis capable of delivering it or if the destination host is even up andrunning.Each packet is forwarded independently of previous packets that mighthave been sent to the same destination. Thus, two successive packetsfrom hostA to hostB may follow completely different paths (perhapsbecause of a change in the forwarding table at some switch in thenetwork).A switch or link failure might not have any serious effect oncommunication if it is possible to find an alternate route around thefailure and to update the forwarding table accordingly.This last fact is particularly important to the history of datagramnetworks. One of the important design goals of the Internet isrobustness to failures, and history has shown it to be quite effectiveat meeting this goal. Since datagram-based networks are the dominanttechnology discussed in this book, we postpone illustrative examplesfor the following sections, and move on to the two main alternatives. A second technique for packet switching uses the concept of a virtualcircuit (VC). This approach, which is also referred to as aconnection-oriented model, requires setting up a virtual connectionfrom the source host to the destination host before any data issent. To understand how this works, consider Figure 58, where hostA again wants to send packets tohostB. We can think of this as a two-stage process. The first stageis connection setup. The second is data transfer. We consider eachin turn.Figure 58. An example of a virtual circuit network.In the connection setup phase, it is necessary to establish aconnection state in each of the switches between the source anddestination hosts. The connection state for a single connection consistsof an entry in a VC table in each switch through which the connectionpasses. One entry in the VC table on a single switch contains:A virtual circuit identifier (VCI) that uniquely identifies theconnection at this switch and which will be carried inside the headerof the packets that belong to this connectionAn incoming interface on which packets for this VC arrive at theswitchAn outgoing interface in which packets for this VC leave the switchA potentially different VCI that will be used for outgoing packetsThe semantics of one such entry is as follows: If a packet arrives onthe designated incoming interface and that packet contains thedesignated VCI value in its header, then that packet should be sent outthe specified outgoing interface with the specified outgoing VCI valuehaving been first placed in its header.Note that the combination of the VCI of packets as they are received atthe switch and the interface on which they are received uniquelyidentifies the virtual connection. There may of course be many virtualconnections established in the switch at one time. Also, we observe thatthe incoming and outgoing VCI values are generally not the same. Thus,the VCI is not a globally significant identifier for the connection;rather, it has significance only on a given link (i.e., it haslink-local scope).Whenever a new connection is created, we need to assign a new VCI forthat connection on each link that the connection will traverse. We alsoneed to ensure that the chosen VCI on a given link is not currently inuse on that link by some existing connection.There are two broad approaches to establishing connection state. One isto have a network administrator configure the state, in which case thevirtual circuit is permanent. Of course, it can also be deleted by theadministrator, so a permanent virtual circuit (PVC) might best bethought of as a long-lived or administratively configured VC.Alternatively, a host can send messages into the network to cause thestate to be established. This is referred to as signalling, and theresulting virtual circuits are said to be switched. The salientcharacteristic of a switched virtual circuit (SVC) is that a host mayset up and delete such a VC dynamically without the involvement of anetwork administrator. Note that an SVC should more accurately be calleda signalled VC, since it is the use of signalling (not switching) thatdistinguishes an SVC from aPVC.Lets assume that a network administrator wants to manually create anew virtual connection from hostA to hostB. First, the administratorneeds to identify a path through the network from A to B. In theexample network of Figure 58, there is onlyone such path, but in general, this may not be the case. Theadministrator then picks a VCI value that is currently unused on eachlink for the connection. For the purposes of our example, letssuppose that the VCI value 5 is chosen for the link from hostA toswitch1, and that 11 is chosen for the link from switch1 toswitch2. In that case, switch1 needs to have an entry in its VCtable configured as shown in Table 6.Table 6. Example Virtual Circuit Table Entry for Switch 1.Incoming InterfaceIncoming VCIOutgoing InterfaceOutgoing VCI25111Similarly, suppose that the VCI of 7 is chosen to identify thisconnection on the link from switch2 to switch3 and that a VCI of 4 ischosen for the link from switch3 to hostB. In that case, switches 2and 3 need to be configured with VC table entries as shown inTable 7 and Table 8,respectively. Note that the outgoing VCI value at one switch is theincoming VCI value at the next switch.Table 7. Virtual Circuit Table Entry at Switch 2.Incoming InterfaceIncoming VCIOutgoing InterfaceOutgoing VCI31127Table 8. Virtual Circuit Table Entry at Switch 3.Incoming InterfaceIncoming VCIOutgoing InterfaceOutgoing VCI0714Figure 59. A packet is sent into a virtual circuit network.Once the VC tables have been set up, the data transfer phase canproceed, as illustrated in Figure 59. For anypacket that it wants to send to hostB, A puts the VCI value of 5 inthe header of the packet and sends it to switch1. Switch1 receivesany such packet on interface2, and it uses the combination of theinterface and the VCI in the packet header to find the appropriate VCtable entry. As shown in Table 6, the tableentry in this case tells switch1 to forward the packet out ofinterface 1 and to put the VCI value 11 in the header when the packetis sent. Thus, the packet will arrive at switch2 on interface 3bearing VCI 11. Switch2 looks up interface 3 and VCI 11 in its VCtable (as shown in Table 7) and sends thepacket on to switch 3 after updating the VCI value in the packetheader appropriately, as shown in Figure 60. This process continues until it arrives at hostB withthe VCI value of 4 in the packet. To hostB, this identifies thepacket as having come from hostA.In real networks of reasonable size, the burden of configuring VC tablescorrectly in a large number of switches would quickly become excessiveusing the above procedures. Thus, either a network management tool orsome sort of signalling (or both) is almost always used, even whensetting up permanent VCs. In the case of PVCs, signalling is initiatedby the network administrator, while SVCs are usually set up usingsignalling by one of the hosts. We consider now how the same VC justdescribed could be set up by signalling from the host.Figure 60. A packet makes its way through a virtual circuitnetwork.To start the signalling process, hostA sends a setup message into thenetworkthat is, to switch1. The setup message contains, among otherthings, the complete destination address of hostB. The setup messageneeds to get all the way to B to create the necessary connection statein every switch along the way. We can see that getting the setup messageto B is a lot like getting a datagram to B, in that the switches have toknow which output to send the setup message to so that it eventuallyreaches B. For now, lets just assume that the switches know enoughabout the network topology to figure out how to do that, so that thesetup message flows on to switches 2 and 3 before finally reachinghostB.When switch1 receives the connection request, in addition to sendingit on to switch2, it creates a new entry in its virtual circuit tablefor this new connection. This entry is exactly the same as shownpreviously in Table 6. The main difference isthat now the task of assigning an unused VCI value on the interface isperformed by the switch for that port. In this example, the switchpicks the value 5. The virtual circuit table now has the followinginformation: When packets arrive on port2 with identifier5, sendthem out on port1. Another issue is that, somehow, hostA will needto learn that it should put the VCI value of 5 in packets that itwants to send to B; we will see how that happens below.When switch2 receives the setup message, it performs a similar process;in this example, it picks the value 11 as the incoming VCI value.Similarly, switch3 picks 7 as the value for its incoming VCI. Eachswitch can pick any number it likes, as long as that number is notcurrently in use for some other connection on that port of that switch.As noted above, VCIs have link-local scope; that is, they have no globalsignificance.Finally, the setup message arrives at hostB. Assuming that B is healthyand willing to accept a connection from hostA, it too allocates anincoming VCI value, in this case 4. This VCI value can be used by B toidentify all packets coming from hostA.Now, to complete the connection, everyone needs to be told what theirdownstream neighbor is using as the VCI for this connection. HostBsends an acknowledgment of the connection setup to switch3 and includesin that message the VCI that it chose(4). Now switch3 can complete thevirtual circuit table entry for this connection, since it knows theoutgoing value must be4. Switch3 sends the acknowledgment on toswitch2, specifying a VCI of7. Switch2 sends the message on toswitch1, specifying a VCI of11. Finally, switch1 passes theacknowledgment on to hostA, telling it to use the VCI of5 for thisconnection.At this point, everyone knows all that is necessary to allow traffic toflow from hostA to hostB. Each switch has a complete virtual circuittable entry for the connection. Furthermore, hostA has a firmacknowledgment that everything is in place all the way to hostB. Atthis point, the connection table entries are in place in all threeswitches just as in the administratively configured example above, butthe whole process happened automatically in response to the signallingmessage sent from A. The data transfer phase can now begin and isidentical to that used in the PVC case.When hostA no longer wants to send data to hostB, it tears down theconnection by sending a teardown message to switch1. The switch removesthe relevant entry from its table and forwards the message on to theother switches in the path, which similarly delete the appropriate tableentries. At this point, if hostA were to send a packet with a VCI of 5to switch1, it would be dropped as if the connection had never existed.There are several things to note about virtual circuit switching:Since hostA has to wait for the connection request to reach the farside of the network and return before it can send its first datapacket, there is at least one round-trip time (RTT) of delay beforedata is sent.While the connection request contains the full address for hostB(which might be quite large, being a global identifier on thenetwork), each data packet contains only a small identifier, which isonly unique on one link. Thus, the per-packet overhead caused by theheader is reduced relative to the datagram model. More importantly,the lookup is fast because the virtual circuit number can be treatedas an index into a table rather than as a key that has to be lookedup.If a switch or a link in a connection fails, the connection is brokenand a new one will need to be established. Also, the old one needs tobe torn down to free up table storage space in the switches.The issue of how a switch decides which link to forward theconnection request on has been glossed over. In essence, this is thesame problem as building up the forwarding table for datagramforwarding, which requires some sort of routing algorithm. Routingis described in a later section, and the algorithms described thereare generally applicable to routing setup requests as well asdatagrams.One of the nice aspects of virtual circuits is that by the time the hostgets the go-ahead to send data, it knows quite a lot about thenetworkfor example, that there really is a route to the receiver andthat the receiver is willing and able to receive data. It is alsopossible to allocate resources to the virtual circuit at the time it isestablished. For example, X.25 (an early and now largely obsoletevirtual-circuit-based networking technology) employed the followingthree-part strategy:Buffers are allocated to each virtual circuit when the circuit isinitialized.The sliding window protocol is run between each pair of nodes alongthe virtual circuit, and this protocol is augmented with flow controlto keep the sending node from over-running the buffers allocated atthe receiving node.The circuit is rejected by a given node if not enough buffers areavailable at that node when the connection request message isprocessed.In doing these three things, each node is ensured of having the buffersit needs to queue the packets that arrive on that circuit. This basicstrategy is usually called hop-by-hop flow control.By comparison, a datagram network has no connection establishment phase,and each switch processes each packet independently, making it lessobvious how a datagram network would allocate resources in a meaningfulway. Instead, each arriving packet competes with all other packets forbuffer space. If there are no free buffers, the incoming packet must bediscarded. We observe, however, that even in a datagram-based network asource host often sends a sequence of packets to the same destinationhost. It is possible for each switch to distinguish among the set ofpackets it currently has queued, based on the source/destination pair,and thus for the switch to ensure that the packets belonging to eachsource/destination pair are receiving a fair share of the switchsbuffers.In the virtual circuit model, we could imagine providing each circuitwith a different quality of service (QoS). In this setting, the termquality of service is usually taken to mean that the network gives theuser some kind of performance-related guarantee, which in turn impliesthat switches set aside the resources they need to meet this guarantee.For example, the switches along a given virtual circuit might allocate apercentage of each outgoing links bandwidth to that circuit. As anotherexample, a sequence of switches might ensure that packets belonging to aparticular circuit not be delayed (queued) for more than a certainamount of time.There have been a number of successful examples of virtual circuittechnologies over the years, notably X.25, Frame Relay, and AsynchronousTransfer Mode (ATM). With the success of the Internets connectionlessmodel, however, none of them enjoys great popularity today. One of themost common applications of virtual circuits for many years was theconstruction of virtual private networks (VPNs), a subject discussedin a later section. Even that application is now mostly supported usingInternet-based technologies today. Asynchronous Transfer Mode (ATM) is probably the most well-known virtualcircuit-based networking technology, although it is now well past itspeak in terms of deployment. ATM became an important technology in the1980s and early 1990s for a variety of reasons, not the least of whichis that it was embraced by the telephone industry, which at that pointin time was less active in computer networks (other than as a supplierof links from which other people built networks). ATM also happened tobe in the right place at the right time, as a high-speed switchingtechnology that appeared on the scene just when shared media likeEthernet and token rings were starting to look a bit too slow for manyusers of computer networks. In some ways ATM was a competing technologywith Ethernet switching, and it was seen by many as a competitor to IPas well.Figure 61. ATM cell format at the UNI.The approach ATM takes has some interesting properties, which makes itworth examining a bit further. The picture of the ATM packet formatmorecommonly called an ATM cellin Figure 61 will illustratethe main points. Well skip the generic flow control (GFC) bits, whichnever saw much use, and start with the 24 bits that are labelled VPI(virtual path identifier8 bits) and VCI (virtual circuit identifier16bits). If you consider these bits together as a single 24-bit field,they correspond to the virtual circuit identifier introduced above. Thereason for breaking the field into two parts was to allow for a level ofhierarchy: All the circuits with the same VPI could, in some cases, betreated as a group (a virtual path) and could all be switched togetherlooking only at the VPI, simplifying the work of a switch that couldignore all the VCI bits and reducing the size of the VC tableconsiderably.Skipping to the last header byte we find an 8-bit cyclic redundancycheck (CRC), known as the header error check (HEC). It uses CRC-8and provides error detection and single-bit error correction capabilityon the cell header only. Protecting the cell header is particularlyimportant because an error in the VCI will cause the cell to bemisdelivered.Probably the most significant thing to notice about the ATM cell, andthe reason it is called a cell and not a packet, is that it comes inonly one size: 53 bytes. What was the reason for this? One big reasonwas to facilitate the implementation of hardware switches. When ATM wasbeing created in the mid- and late 1980s, 10-Mbps Ethernet was thecutting-edge technology in terms of speed. To go much faster, mostpeople thought in terms of hardware. Also, in the telephone world,people think big when they think of switchestelephone switches oftenserve tens of thousands of customers. Fixed-length packets turn out tobe a very helpful thing if you want to build fast, highly scalableswitches. There are two main reasons for this:It is easier to build hardware to do simple jobs, and the job ofprocessing packets is simpler when you already know how long each onewill be.If all packets are the same length, then you can have lots ofswitching elements all doing much the same thing in parallel, each ofthem taking the same time to do its job.This second reason, the enabling of parallelism, greatly improves thescalability of switch designs. It would be overstating the case to saythat fast parallel hardware switches can only be built usingfixed-length cells. However, it is certainly true that cells ease thetask of building such hardware and that there was a lot of knowledgeavailable about how to build cell switches in hardware at the time theATM standards were being defined. As it turns out, this same principleis still applied in many switches and routers today, even if they dealin variable length packetsthey cut those packets into some sort of cellin order to forward them from input port to output port, but this is allinternal to the switch.There is another good argument in favor of small ATM cells, having to dowith end-to-end latency. ATM was designed to carry both voice phonecalls (the dominant use case at the time) and data. Because voice islow-bandwidth but has strict delay requirements, the last thing you wantis for a small voice packet queued behind a large data packet at aswitch. If you force all packets to be small (i.e., cell-sized), thenlarge data packets can still be supported by reassembling a set of cellsinto a packet, and you get the benefit of being able to interleave theforwarding of voice cells and data cells at every switch along the pathfrom source to destination. This idea of using small cells to improveend-to-end latency is alive and well today in cellular access networks.Having decided to use small, fixed-length packets, the next question waswhat is the right length to fix them at? If you make them too short,then the amount of header information that needs to be carried aroundrelative to the amount of data that fits in one cell gets larger, so thepercentage of link bandwidth that is actually used to carry data goesdown. Even more seriously, if you build a device that processes cells atsome maximum number of cells per second, then as cells get shorter thetotal data rate drops in direct proportion to cell size. An example ofsuch a device might be a network adaptor that reassembles cells intolarger units before handing them up to the host. The performance of sucha device depends directly on cell size. On the other hand, if you makethe cells too big, then there is a problem of wasted bandwidth caused bythe need to pad transmitted data to fill a complete cell. If the cellpayload size is 48bytes and you want to send 1byte, youll need tosend 47bytes of padding. If this happens a lot, then the utilization ofthe link will be very low. The combination of relatively highheader-to-payload ratio plus the frequency of sending partially filledcells did actually lead to some noticeable inefficiency in ATM networksthat some detractors called the cell tax.As it turns out, 48 bytes was picked for the ATM cell payload as acompromise. There were good arguments for both larger and smaller cells,and 48 made almost no one happya power of two would certainly have beenbetter for computers to process. A third approach to switching that uses neither virtual circuits norconventional datagrams is known as source routing. The name derivesfrom the fact that all the information about network topology that isrequired to switch a packet across the network is provided by the sourcehost.There are various ways to implement source routing. One would be toassign a number to each output of each switch and to place that numberin the header of the packet. The switching function is then verysimple: For each packet that arrives on an input, the switch wouldread the port number in the header and transmit the packet on thatoutput. However, since there will in general be more than one switchin the path between the sending and the receiving host, the header forthe packet needs to contain enough information to allow every switchin the path to determine which output the packet needs to be placedon. One way to do this would be to put an ordered list of switch portsin the header and to rotate the list so that the next switch in thepath is always at the front of the list. Figure 62 illustrates this idea.Figure 62. Source routing in a switched network (where the switchreads the rightmost number).In this example, the packet needs to traverse three switches to get fromhost A to host B. At switch 1, it needs to exit on port 1, at the nextswitch it needs to exit at port 0, and at the third switch it needs toexit at port 3. Thus, the original header when the packet leaves host Acontains the list of ports (3, 0, 1), where we assume that each switchreads the rightmost element of the list. To make sure that the nextswitch gets the appropriate information, each switch rotates the listafter it has read its own entry. Thus, the packet header as it leavesswitch 1 en route to switch 2 is now (1, 3, 0); switch 2 performs anotherrotation and sends out a packet with (0, 1, 3) in the header. Althoughnot shown, switch 3 performs yet another rotation, restoring the headerto what it was when host A sent it.There are several things to note about this approach. First, it assumesthat host A knows enough about the topology of the network to form aheader that has all the right directions in it for every switch in thepath. This is somewhat analogous to the problem of building theforwarding tables in a datagram network or figuring out where to send asetup packet in a virtual circuit network. In practice, however, it isthe first switch at the ingress to the network (as opposed to the endhost connected to that switch) that appends the source route.Second, observe that we cannot predict how big the header needs to be,since it must be able to hold one word of information for every switchon the path. This implies that headers are probably of variable lengthwith no upper bound, unless we can predict with absolute certainty themaximum number of switches through which a packet will ever need topass.Third, there are some variations on this approach. For example, ratherthan rotate the header, each switch could just strip the first elementas it uses it. Rotation has an advantage over stripping, however: Host Bgets a copy of the complete header, which may help it figure out how toget back to host A. Yet another alternative is to have the header carrya pointer to the current next port entry, so that each switch justupdates the pointer rather than rotating the header; this may be moreefficient to implement. We show these three approaches inFigure 63. In each case, the entry thatthis switch needs to read is A, and the entry that the next switchneeds to read is B.Figure 63. Three ways to handle headers for source routing:(a) rotation; (b) stripping; (c) pointer. The labels are readright to left.Source routing can be used in both datagram networks and virtual circuitnetworks. For example, the Internet Protocol, which is a datagramprotocol, includes a source route option that allows selected packets tobe source routed, while the majority are switched as conventionaldatagrams. Source routing is also used in some virtual circuit networksas the means to get the initial setup request along the path from sourceto destination.Source routes are sometimes categorized as strict or loose. In astrict source route, every node along the path must be specified,whereas a loose source route only specifies a set of nodes to betraversed, without saying exactly how to get from one node to the next.A loose source route can be thought of as a set of waypoints rather thana completely specified route. The loose option can be helpful to limitthe amount of information that a source must obtain to create a sourceroute. In any reasonably large network, it is likely to be hard for ahost to get the complete path information it needs to construct correctlya strict source route to any destination. But both types of sourceroutes do find application in certain scenarios, as we will see in laterchapters. Built with Sphinx using a theme provided by Read the Docs. What is a Switch in Computer Networks? Data is transmitted through various media as users connect to the internet or other networks. Switching in computer networks is the method of transferring information between different networks.In this article, we have explained about switching and its types in computer networks. We have also covered the working process in switching, modes of switching, and different techniques.Further, if you are interested in learning about switches and switches in detail, you can check out our Networking courses, where you can learn about switching and similar technologies.What is Switching?Switching is the process of transferring data packets between devices using a network switch. A network switch is a hardware component that links multiple computers within a Local Area Network (LAN). Operating at Layer 2 (the Data Link layer) of the OSI model, switches manage the flow of data seamlessly, without requiring user configuration. They forward packets based on MAC addresses, ensuring that data is sent only to the intended device, thus optimizing network performance.Switches operate in full-duplex mode, which allows simultaneous communication between devices, reducing the likelihood of packet collisions. Unlike traditional hubs that broadcast messages to all connected devices, switches utilize bandwidth more efficiently by targeting specific destinations.Why are Switches Used in Networking?As networking demands have grown, the limitations of hub-based systems have become increasingly clear. Switches have emerged as the preferred connecting devices due to their ability to facilitate more efficient data transmission. Unlike hubs, which indiscriminately send data to all devices, switches intelligently direct information based on MAC addresses, minimizing network congestion and improving overall efficiency. CCNA Training Course with Virtual Lab AccessStart your career in networking by training for CCNA certification.Explore course Types of SwitchingThere are three types of network switching:1. Circuit SwitchingCircuit switching establishes a dedicated communication path between the sender and receiver for the duration of the call or session. This technique is analogous to traditional telephone systems, where a continuous circuit is maintained until the conversation ends.Key Phases:Circuit Establishment: A connection request is sent, and an acknowledgment is received before data transmission begins.Data Transfer: Data is transmitted along the established path.Circuit Disconnect: Once communication is complete, the dedicated path is terminated.Technologies Used:Space Division Switching: Utilizes physical separation within a switch to create transmission paths.Crossbar Switch: A type of space division switch with n input and n output lines, having n crosspoints.Multistage Switch: Reduces the number of crosspoints by splitting a crossbar into smaller interconnected units, allowing for path redundancy.Advantages: Dedicated communication channel. Fixed bandwidth ensures consistent performance.Disadvantages: Establishment time can be lengthy (around 10 seconds) with no data transmission during this period. Inefficient use of resources if the channel is idle. More expensive due to the need for dedicated paths.2. Message SwitchingIn message switching, complete messages are routed through intermediate nodes where they are stored and forwarded. There is no need for a dedicated path between sender and receiver.How It Works: Messages are appended with a destination address and dynamically routed based on available information. Each node stores the entire message before forwarding it to the next.Advantages: Efficient use of bandwidth as channels are shared. Reduces traffic congestion through temporary storage at nodes. Supports messages of varying sizes.Disadvantages: Requires substantial storage at each node. Long delays may occur due to the storing and forwarding process.3. Packet SwitchingPacket switching breaks messages into smaller packets, which are transmitted independently across the network. Each packet is identified by its sequence number and contains source and destination addresses.Key Approaches:Datagram Packet Switching: Treats packets as independent entities with no predetermined path. Routing decisions are made at each intermediate node, leading to potential congestion but also flexibility.Virtual Circuit Switching: Establishes a fixed path before data transmission. This connection-oriented approach ensures that all packets follow the same route.Advantages: Cost-effective, as no large storage is needed for packets. Reliable; packets can be rerouted if a node is busy. Efficient, allowing multiple users to share the same channel simultaneously.Disadvantages: Not suitable for applications requiring low latency and high quality. Complex protocols can lead to higher implementation costs. Risk of data loss if packets are not retransmitted during network overload.Why is Network Switching Important?Switching techniques enhance bandwidth utilization across the network. Bandwidth is the maximum data transfer rate of a connection and is a vital and costly resource.Switching technology mitigates this risk by packet collision, ensuring that data packets are routed directly to their destinations.This approach reduces network congestion and improves overall efficiencySwitching ModesIn the realm of computer networking, particularly at the Data Link layer (Layer 2 of the OSI model), switching plays a crucial role in data transmission. Layer 2 switches manage the forwarding of data frames, utilizing MAC addresses to direct traffic efficiently. Within this framework, various switching modes exist, each offering distinct approaches to handling data frames. The primary switching modes include Store-and-Forward, Cut-Through, and Fragment-Free.1. Store-and-Forward SwitchingStore-and-Forward is a robust switching technique where the switch receives the entire frame before any further action is taken. Heres how it works:When a switch receives a frame, it stores the complete frame in its buffer memory. Once the frame is fully received, it undergoes error checking using Cyclic Redundancy Check (CRC) to ensure it is free of errors before transmission. If the frame is error-free, it is forwarded to the next node; if errors are detected, the frame is discarded. Advantages High Reliability: Since corrupted frames are not forwarded, the destination network remains unaffected by errors. Error Checking: Ensures that only valid data frames are transmitted, enhancing network integrity. Disadvantages Higher Latency: Waiting for the entire frame to be received before processing can lead to delays.2. Cut-Through SwitchingCut-through switching offers a different approach, significantly reducing latency. This technique allows the switch to forward packets as soon as the destination address is identified, which occurs after reading the first six bytes of the frame. The switch does not wait for the entire frame to be received, which speeds up the process. Advantages Low Latency: This mode provides rapid forwarding, making it suitable for time-sensitive applications. Reduced Wait Time: By forwarding frames immediately after identifying the destination, overall network efficiency is enhanced. Disadvantages No Error Checking: Frames can be forwarded with potential errors, which might affect network reliability. Collision Handling: Collided frames may also be forwarded, leading to possible data integrity issues.3. Fragment-Free SwitchingFragment-free switching is a hybrid approach that balances speed and error checking. This technique requires the switch to read at least 64 bytes of the incoming frame before forwarding it, allowing for the detection of collisions that typically occur in the initial bytes. By ensuring the switch has enough information to check for errors, Fragment-Free switching merges the speed of Cut-Through with the reliability of Store-and-Forward. Advantages Error Mitigation: By analyzing the first 64 bytes, it reduces the likelihood of forwarding corrupted frames. Efficient Performance: This mode offers a good balance of speed and reliability, making it ideal for many networking scenarios. Disadvantages Moderate Latency: While faster than Store-and-Forward, it still incurs some delay compared to pure Cut-Through switching.Comparison Between Different Switching Modes Feature Store-and-Forward Switching Cut-Through Switching Fragment-Free Switching Frame Reception Waits for the entire frame Checks first 6 bytes, then forwards Reads at least 64 bytes before forwarding Error Checking Yes, discards corrupted frames No, forwards frames regardless of errors Partial check, discards collided frames Latency High Low Moderate Reliability High, forwards only error-free frames Low, can forward error-prone frames Moderate, reduces the chance of forwarding errors Wait Time High, due to full frame requirement Low, forwards immediately upon identifying a destination Moderate, checks partial frame Want to prepare for reputed Cisco Certifications like CCNA, CCNP, or SD-WAN? Check out our Cisco Enterprise Courses or contact our learner advisor.Differences Between Datagram and Virtual Circuit Approaches Feature Datagram Approach Virtual Circuit Approach Routing Decisions Made at each node Fixed path established beforehand Congestion Handling No congestion, packets can take different paths Congestion can occur on the fixed path Flexibility High, packets treated independently Less flexible, fixed routes Advantages of SwitchingBenefits of network switching are:Increased Bandwidth: Switches enhance the overall bandwidth of a network.Reduced Device Workload: By directing information solely to the intended device, switches alleviate the processing load on individual computers.Improved Network Performance: Traffic is minimized, leading to a more efficient network operation.Lower Collision Rates: Each connection has its collision domain, significantly reducing frame collisions.Disadvantages of SwitchingSome disadvantages of network switching are:Cost: Switches tend to be more expensive than simple network bridges.Connectivity Challenges: Diagnosing network connectivity issues can be more complex with switches.Design Complexity: Effective design and configuration are necessary to manage multicast packets efficiently.Future Trends of Switching in Computer NetworksAs network switching technology continues to evolve, it faces several challenges and opportunities:Security Concerns: With increasing network interconnectivity, security threats are also on the rise. Enhancements in encryption, access control, and threat detection are critical for protecting sensitive information.5G Integration: The advent of 5G technology presents both challenges and opportunities for network switching, demanding infrastructure capable of supporting a growing array of connected devices and applications.Edge Computing: The rise of edge computing necessitates switching solutions that effectively handle decentralized architectures.Artificial Intelligence (AI) Integration: Incorporating AI into network switches can improve automation, predictive maintenance, and adaptive optimization of network performance. Machine learning can analyze traffic patterns to foresee and address potential issues.Quantum Networking: The exploration of quantum technologies may revolutionize network switching. Quantum switches, leveraging quantum entanglement, could transform data transmission methods.ConclusionNetwork switching is a cornerstone of efficient data transmission. By understanding the different types of switchingcircuit, message, and packet switchingwe can understand the strength of each type of switching. As the number of devices in a network is increasing, network switching is becoming critical for managing data flow within local networks, enhancing bandwidth utilization, and reducing collisions.The switching modesStore-and-Forward, Cut-Through, and Fragment-Free enable network designers to select the most appropriate switching method based on specific performance requirements and application needs.As we look to the future, the integration of advanced technologies like AI, edge computing, and quantum networking will likely redefine our approach to switching. These innovations promise to enhance the automation, security, and efficiency of network switching.

Switching power supply. Switching power supply efficiency. Switching power supply explained.