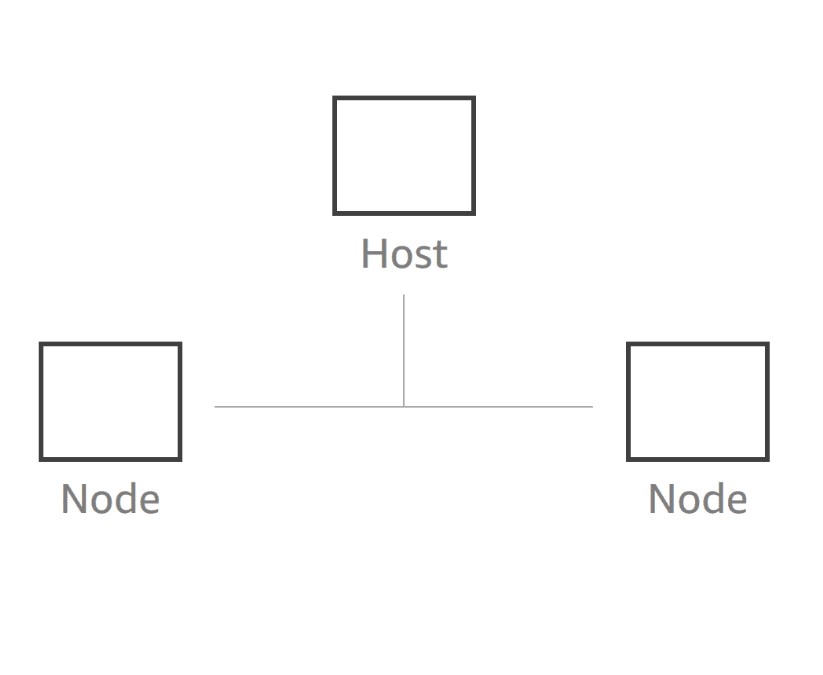
# Introduction to networking

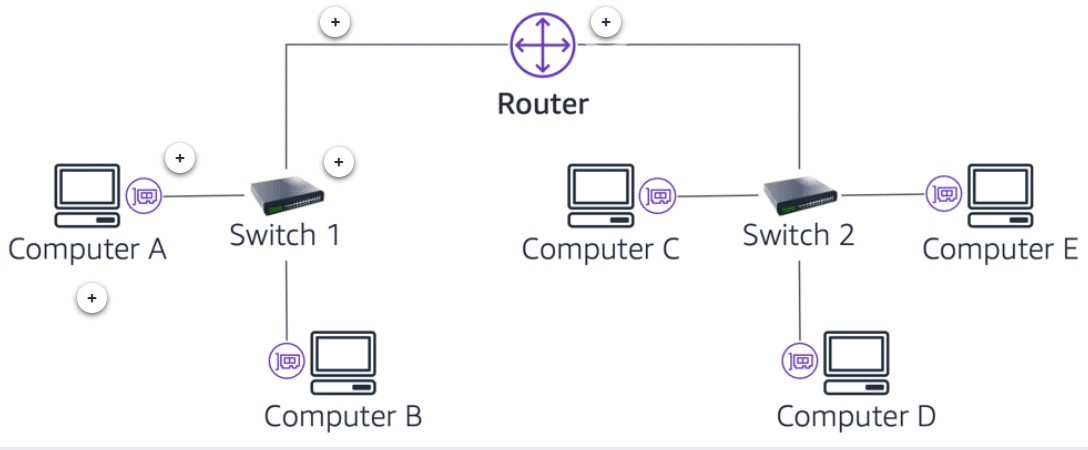
### At the core of the lesson, you will learn how to:

* Define basic networking terms
* Identify the main components of a computer networkWhat is a computer network?
* A *computer network* is a collection of computing devices that are logically connected to communicate and share resources.
* A *node* refers to any device on a network. Examples of nodes can be servers, switches, modems, or even printers.
* A *host* device is a node that has unique function: other devices connect to nodes so they can access data or other services. A server is an example of a host because a server can provide access to data, run an application, or provide a service.



Basic computer network

Elements of a computer network:-



Computer A

A *computer* is a typical node on a network. On the network, it can be either a *client* or a *server*.

Switch

A *switch* connects multiple devices on a network.

Network cable

A *network cable* connects the nodes on a network.

Router

A *router* connects multiple switches on a network.

Client



Computer and mobile device clients

* A *client* is a computer hardware device that accesses the data or a service that is managed by another computer hardware device, which is called a *server*.
* The client connects to the server over a *network*.
* A *client* can also refer to software on the accessing device. For example, a web browser is a client for accessing content from a web server.

Server



Rack-mounted server components

* A *server* provides a response to a request from a client computer over a network.
* Examples:
  + Web server
  + Database server
  + File server
  + Mail server
  + Print server

## Network interface card (NIC)

* A *network interface card (NIC)* connects a computer to a computer network. It is also sometimes referred to as a *network adapter*.
* It uses a *cable* that is connected to a hub or a switch.
* Each NIC has its own *media access control (MAC) address*. The MAC address is a unique physical (hardware) identifier. It is assigned by the manufacturer, and is used to identify the sender and receiver of data.



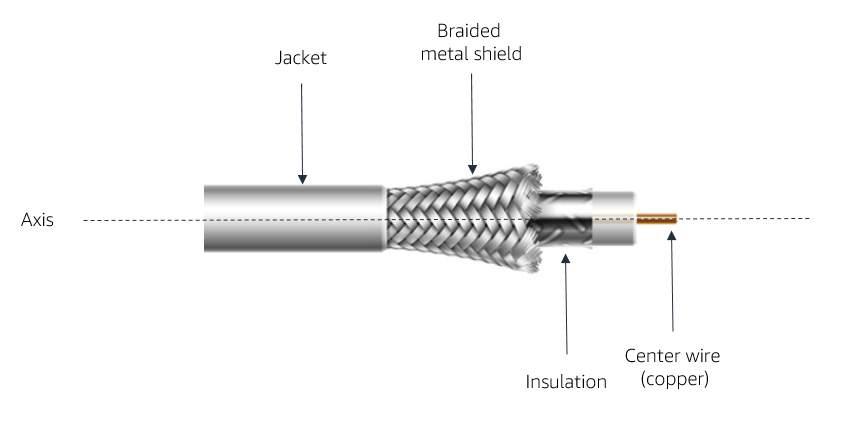
Wired NIC

## Network cable

Most network nodes are linked together by using some type of cabling. Three *types of cables* are commonly used for networks:

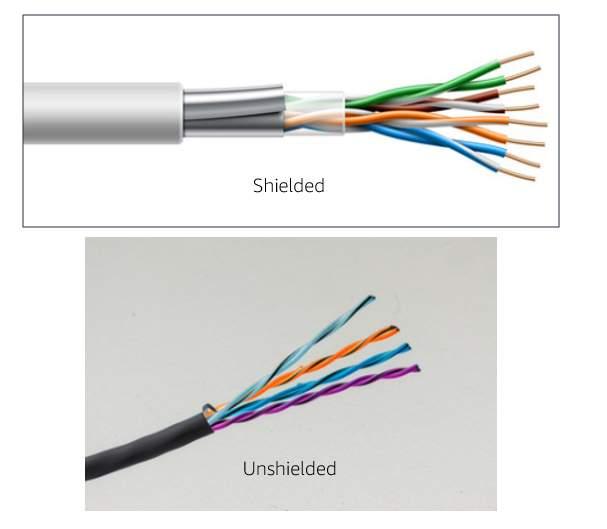
1.COAXIAL

* A *coaxial cable* (or *coax*) contains a central conductor wire (typically copper) that is surrounded by an insulating material. This insulating material, in turn, is surrounded by a braided metal shield. The entire cable is then enclosed in a jacket.
* Today, the most common uses for coaxial cables are to *connect a cable TV modem to an internet service provider (ISP)*, and to connect TVs to cable boxes.



TWISTED PAIR :-

* The *most common type of computer, telephone, and network cable* is known as a *twisted-pair*. It is more commonly known as an *Ethernet* cable. The cable consists of color-coded pairs of insulated copper wires that are bundled together in the same jacket.
* Twisted-pair cabling is available in two types: shielded (STP) and unshielded (UTP). Shielding further prevents electromagnetic interference.

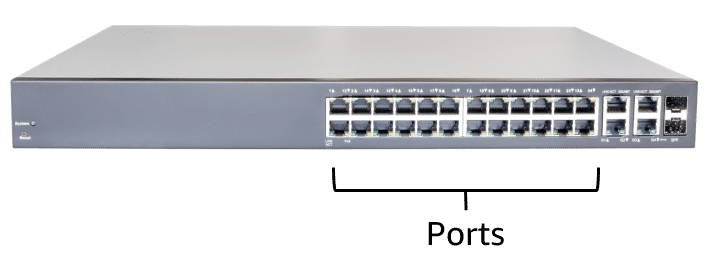


3.FIBER OPTIC

* A *fiber-optic cable* has four components:  
  + Glass fiber (the core)
  + Cladding (the part that makes the light reflect down the fiber)
  + Buffer material (for strength)
  + Insulating jacket
* A fiber-optic cable transmits light instead of electricity. Thus, it is a good choice for use in areas that have high levels of electromagnetic interference (EMI) and for *long-distance transmissions*. A single copper cable cannot carry data more than a few hundred meters, but a single piece of fiber-optic cabling can transmit data over distances of several kilometers.



## Hub and switch

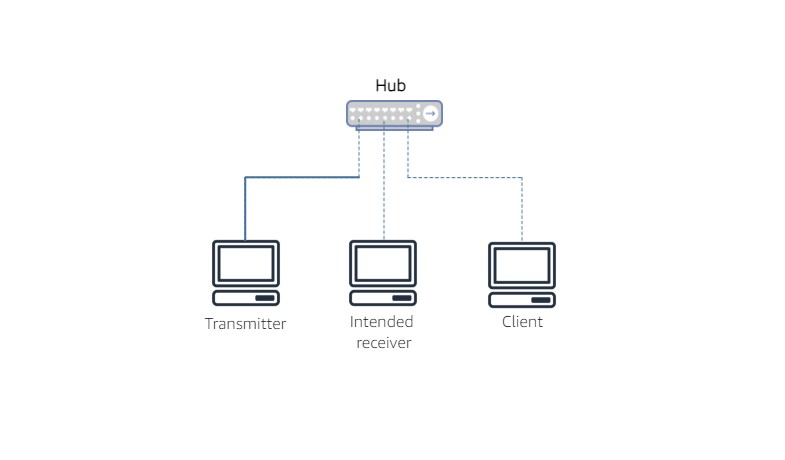


Network hub or switch

A *hub* or *switch* is a device that connects all the nodes of a network together. Every hardwired device in the network uses a network adapter, or NIC, to connect directly to a *port* on the hub or switch through a single cable.

Hub:- A device that transmits data that is received on one port *to all the other ports on the hub*. If one node sends data, all the others receive it, but only the intended recipient listens to it.

Switch:-A device that transmits data *to only the receiving device*. It makes a direct link *by using MAC addresses* between the transmitting device and the receiving device.It improves performance because it does not use bandwidth on unnecessary transmissions.



Similar to switch , a hub is a device that connects all the nodes of a network together . however , there is an important difference , a hub rebroadcast any signals it receives on one port to all the other nodes. So if one nodes sends a signal, all the other receive it , but only the intended recipient listener to it .

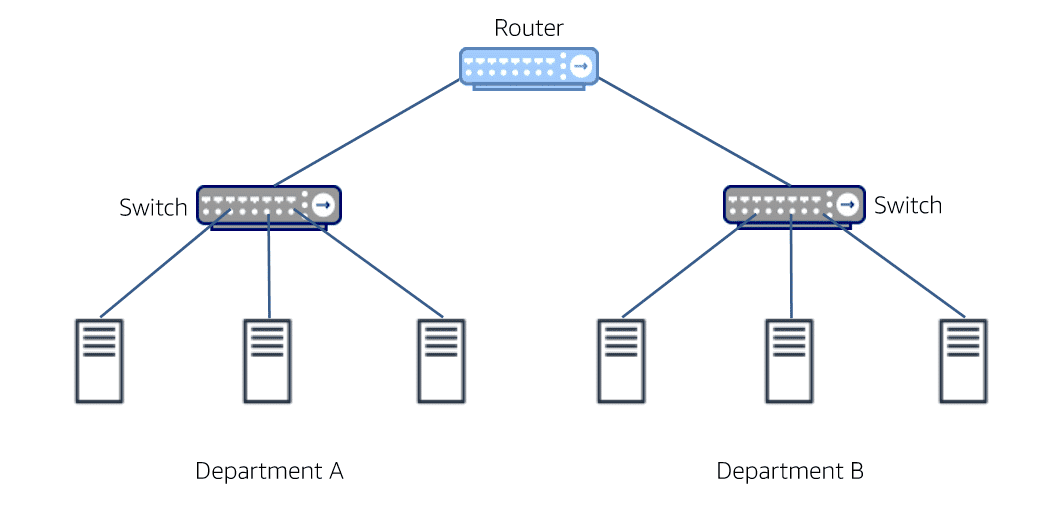
The benefits of a switch over a hub ,is that the switch improves performance ,since it doesn’t waste bandwidth on the unnecessary transmissions .

## Router

A *router* is a network device that connects *multiple network segments into one network*.

* Connects *multiple switches* and their respective networks to form a larger network (that is, it acts as a switch between networks).
* Can also *filter the data* that flows through it, which enables data to be routed differently, depending on need.

The diagram below shows two switched networks (which are also called *subnets*)—Department A and Department B—that are interconnected through a router.



Router diagram

## Key takeaways

* A computer network is a *collection of computing devices that are logically connected to communicate and share resources*.
* The main components of a computer network include:
  + Client devices
  + Servers
  + Network adapters (NICs) and cables
  + Hubs and switches
  + Routers
* Which network device should you use to connect multiple networks?
* Which device makes a direct link (by using MAC addresses) between the transmitting device and receiving device?
* Which type of cable transmits light instead of electricity?

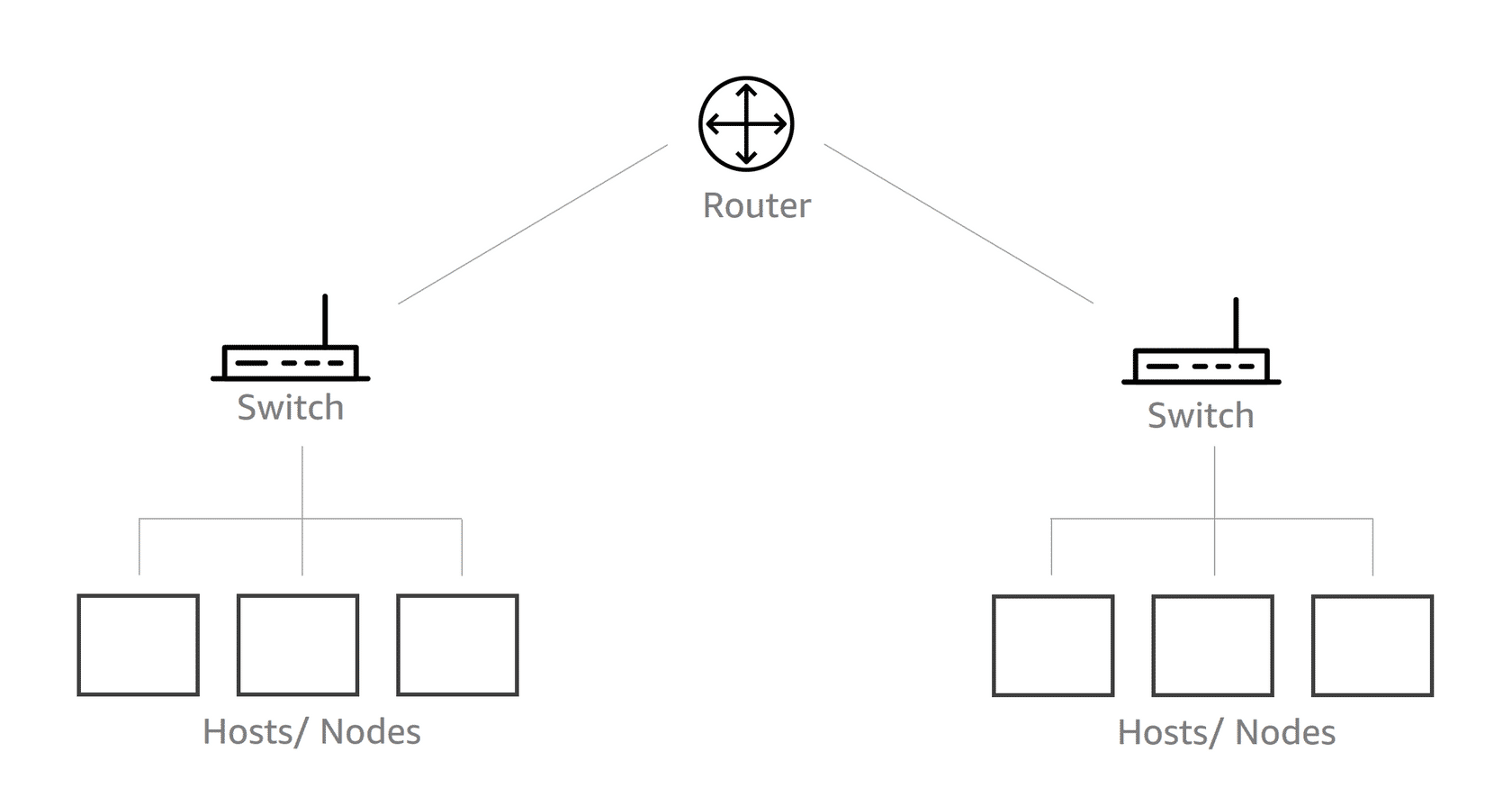
Page 2 . Networking concepts

### At the core of the lesson, you will learn how to:

* Distinguish between different types of networks
* Describe common network management models and network topologies
* Explain the purpose of the Open Systems Interconnection (OSI) communication model
* List different types of network protocols

### Types of computer networks

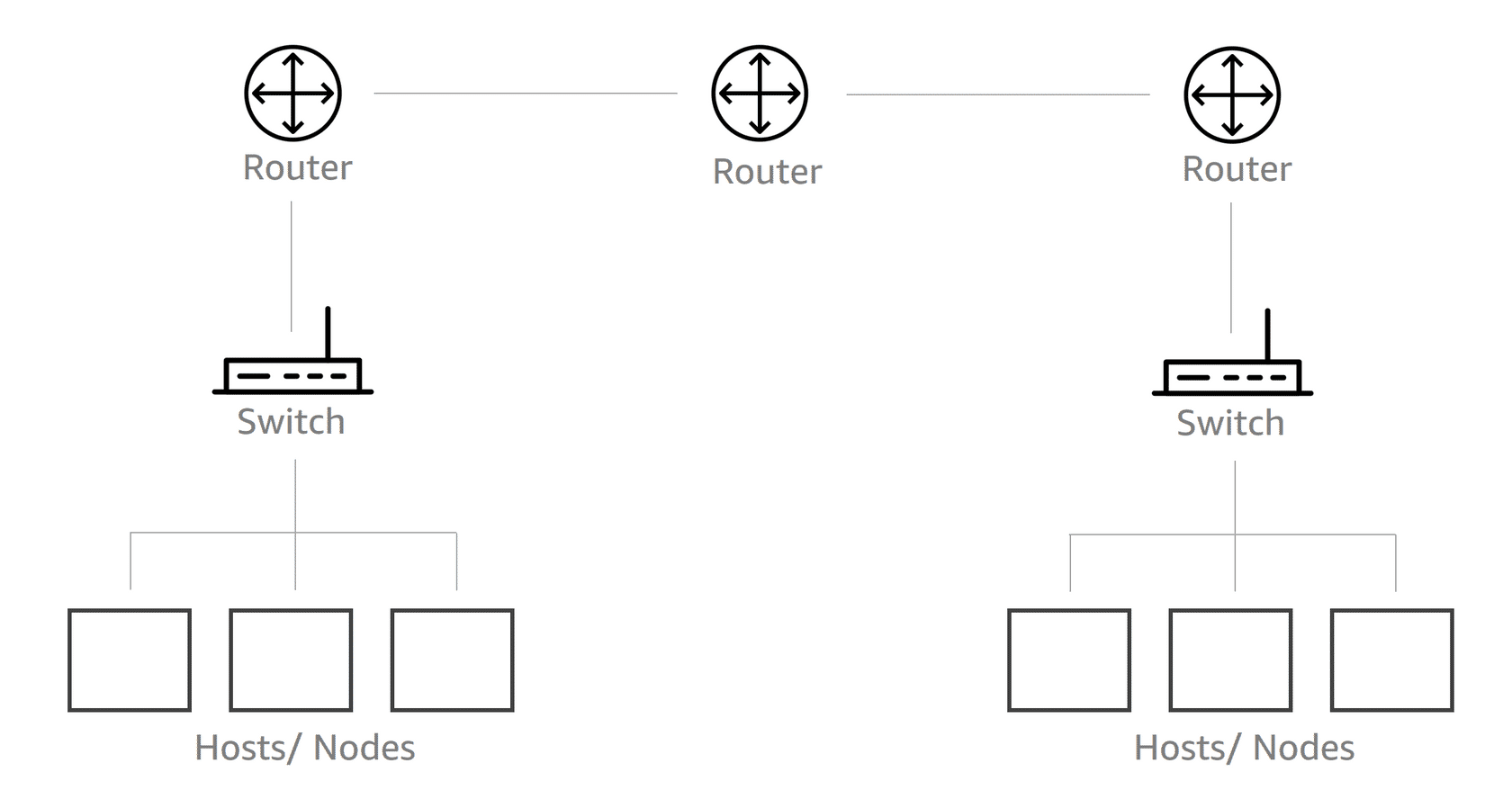
From the standpoint of *geographical span*, two of the most common types of computer networks are *local area networks (LANs)* and *wide area networks (WANs)*.



A LAN with one router, two switches, and three nodes under each switch

Local-area network (LAN)

* A LAN connects devices in a *limited geographical area*, such as a floor, building, or campus.
* LANs commonly use the *Ethernet* standard for connecting devices, and they usually have a high data-transfer rate.
* *Wireless technology* is also commonly used for a LAN.



A WAN with three routers, two switches, and three nodes under each switch

Wide-area network (WAN)

* A WAN connects devices in a *large geographical area*, such as multiple cities or countries.
* WANs are *used to connect LANs*. They use technologies such as fiber-optic cables and satellites to transmit data.
* The *internet* is considered to be the largest WAN.

## Video: LAN versus WAN

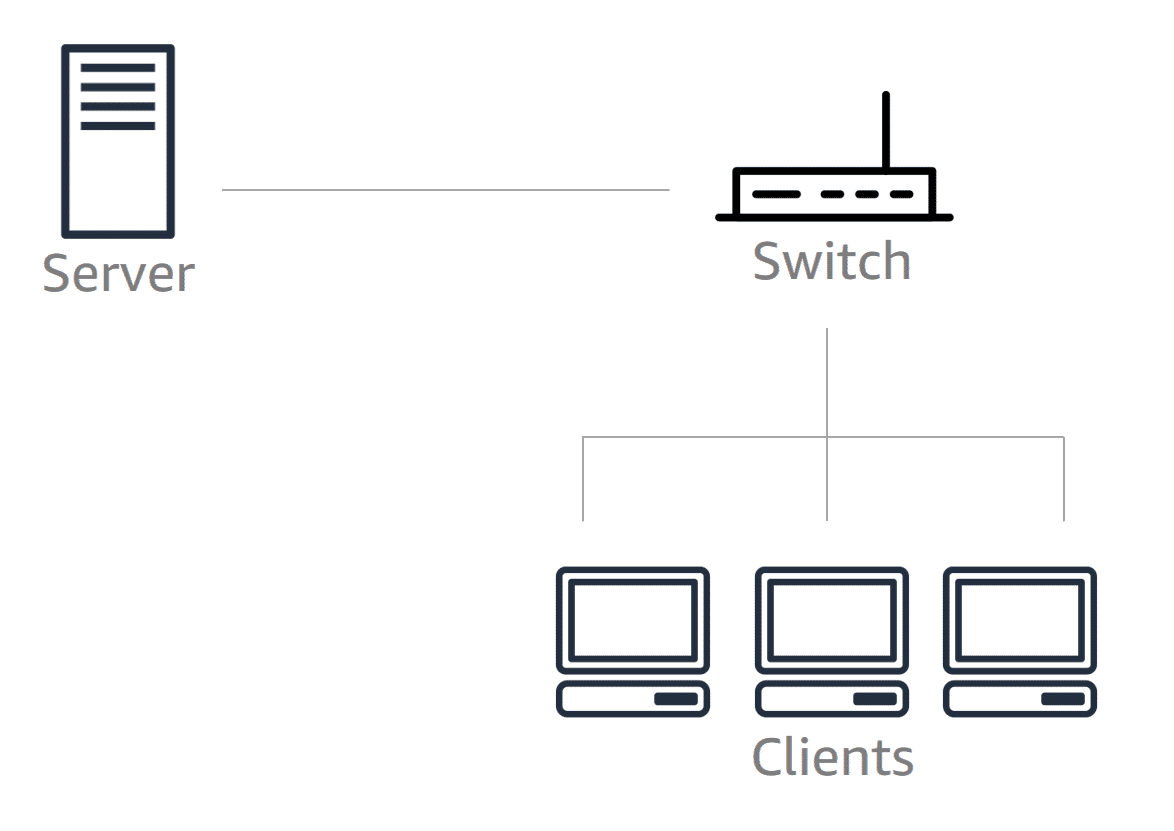
Lets compare local area network with wide ara network .

First see LAN

Suppose your department has three hosts , or servers , that need to communicate with each other over a network , you can configure a switch and connect each host to it . suppose there is another department that has some similarity networked hosts ,and both department wants thair host to communicate with each other

. to connect the two departments , you can configure a router and connect each switch to the router .

## Network management models

A network management model is a representation of *how data is managed, and how applications are hosted in a network*. It is important to understand network management models because they define the roles and relationships of the devices in your network. The two most common network management models for a LAN are *client-server* and *peer-to-peer*. 

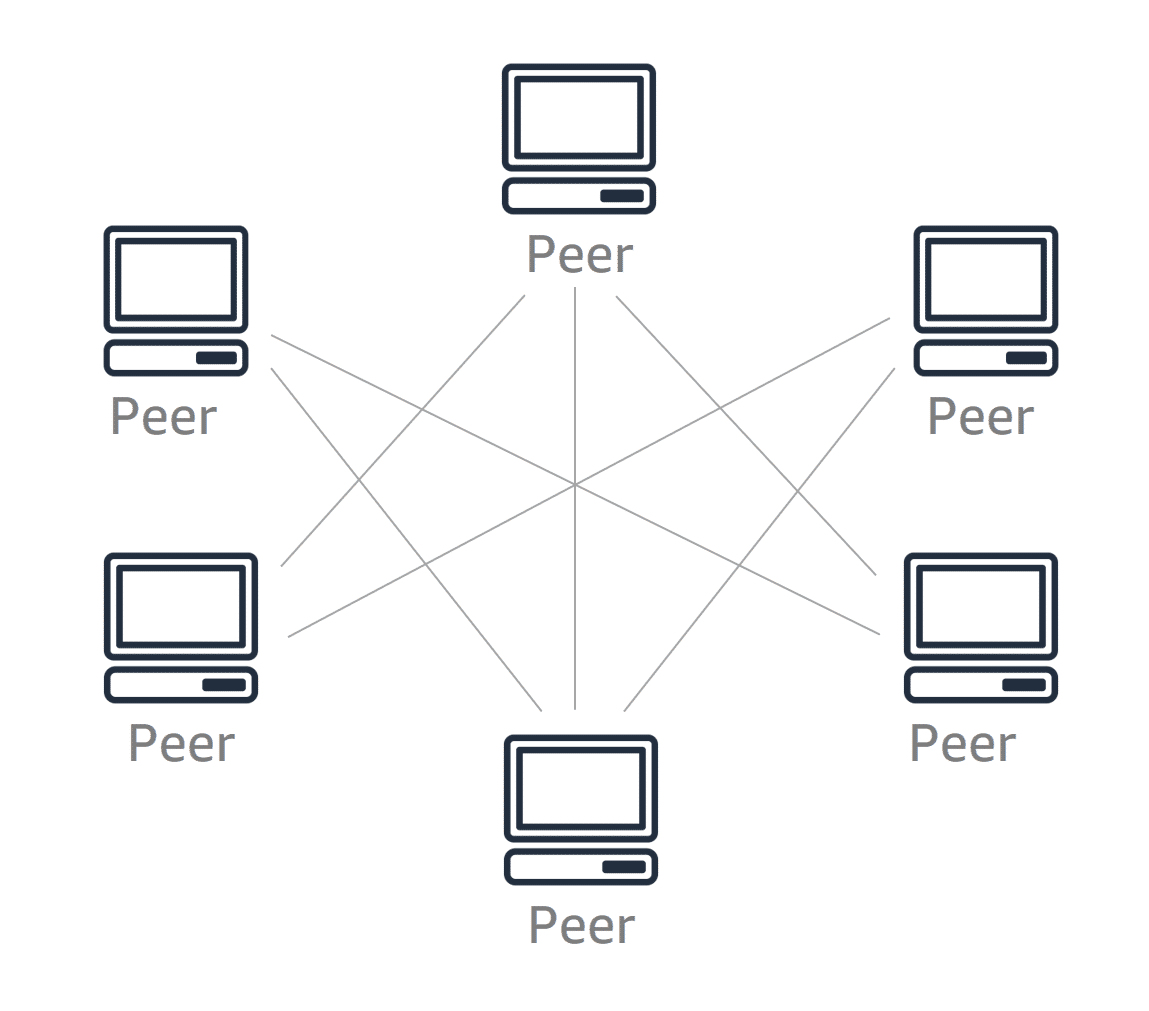
Client-server model

Client-server model

In a client-server network model, the *data* management and *application* hosting *are centralized at the server* and distributed to the clients. All clients on the network must use the designated server to access shared files and information that are stored on the serving computer.

If the server goes down, no client can access the network until the server is restored. Examples of client-server models are:

* File server and desktop clients
* Print server and desktop clients



Peer-to-peer model

Peer-to-peer model

In a peer-to-peer network model, *each node has its own data and applications* and is responsible for its own management and security.

The peer-to-peer model is a distributed architecture that *shares* tasks or workloads among peers. Peers are equally privileged participants in the architecture. For example, files can be shared directly between systems on the network without a central server.

This model might be considered under the following conditions:

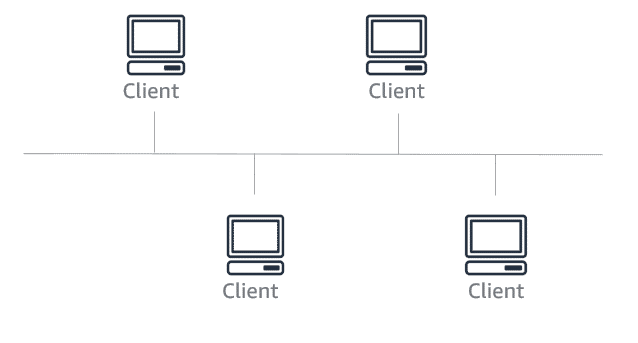
* Users are responsible for backing up each node
* Security requirements are not restrictive
* A limited number of peers are used

### Network topologies

Computer networks use different *topologies* to share information. A topology is a *pattern* (or diagram) *that shows how nodes connect to each other*. Computer networks have both physical topologies and logical typologies.

* Physical topology – Refers to the physical layout of wires in the network
* Logical topology – Refers to how data moves through the network

The following descriptions define historical topologies, such as *bus* and *star*, and more modern *mesh* and *hybrid* topologies.



Bus topology

Bus

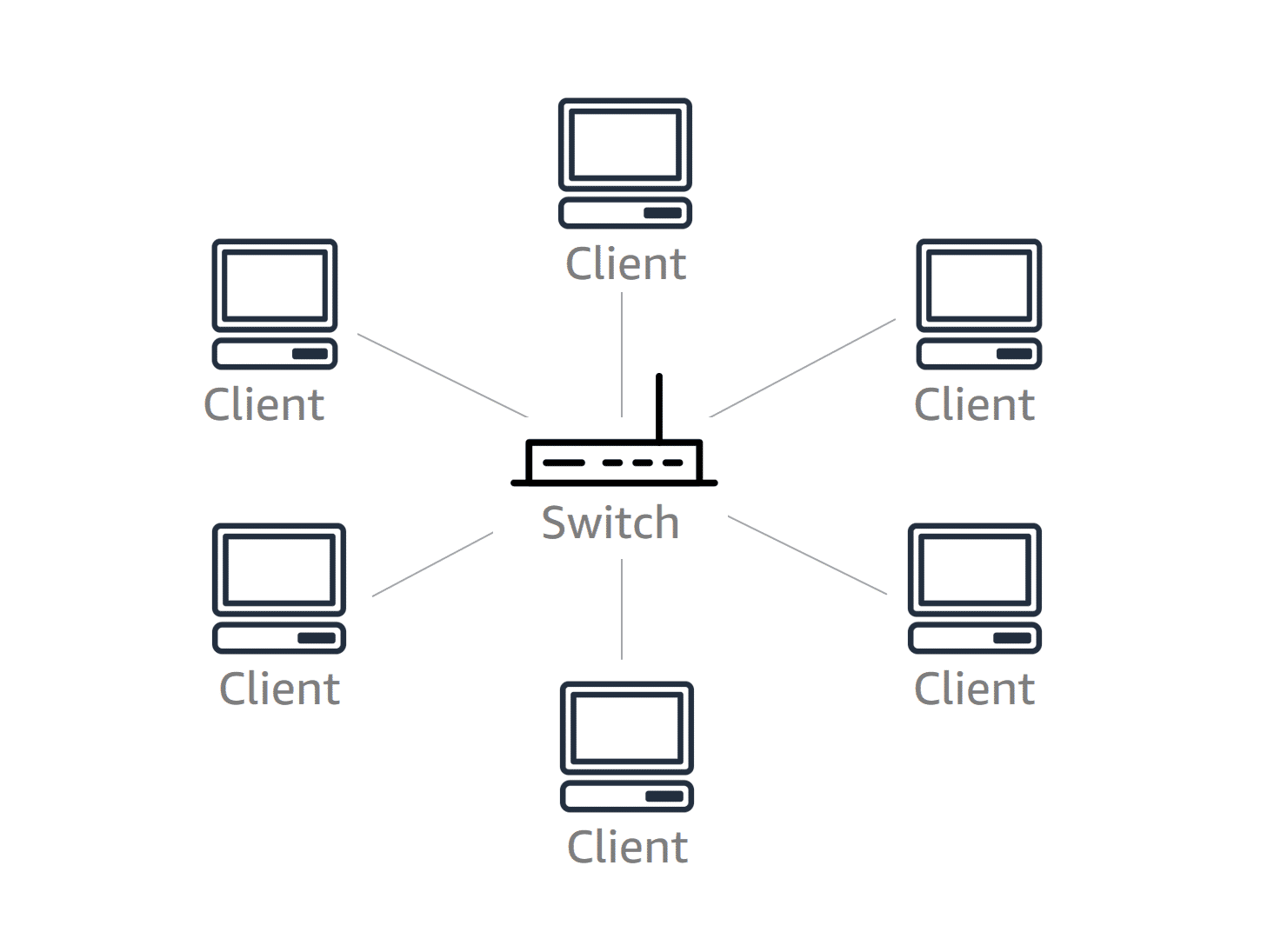
Physical topology

A *bus* topology *positions all the devices on a network along a single cable*. They run in a single direction from one end of the network to the other. A bus topology is also called a *line topology* or *backbone topology*.

Logical topology

The *data flow* on the network also follows the route of the cable, and it *moves in one direction*.

A bus topology is simple to configure. However, it allows only one computer to send a signal at a time, which can cause network collisions that bring down the network.



Star topology

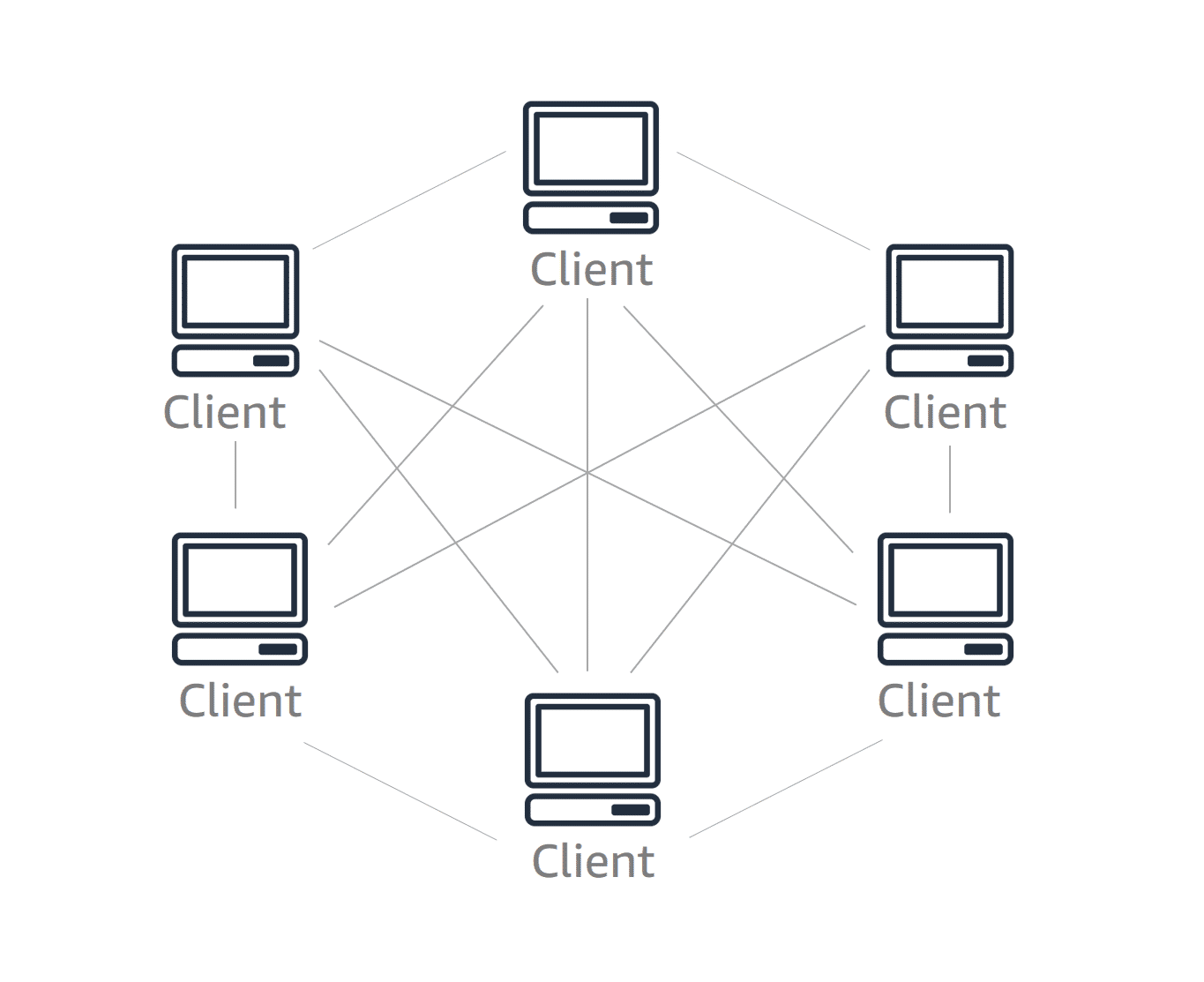
Star

Physical topology

A *star* topology is set up so that *every node* in the network is directly *connected to one central switch* by using coaxial, twisted-pair, or fiber-optic cables.

Logical topology

This *central switch manages data transmission*. Data that is sent from any node on the network must pass through the central switch to reach its destination. The central switch can also function as a repeater to prevent data loss.



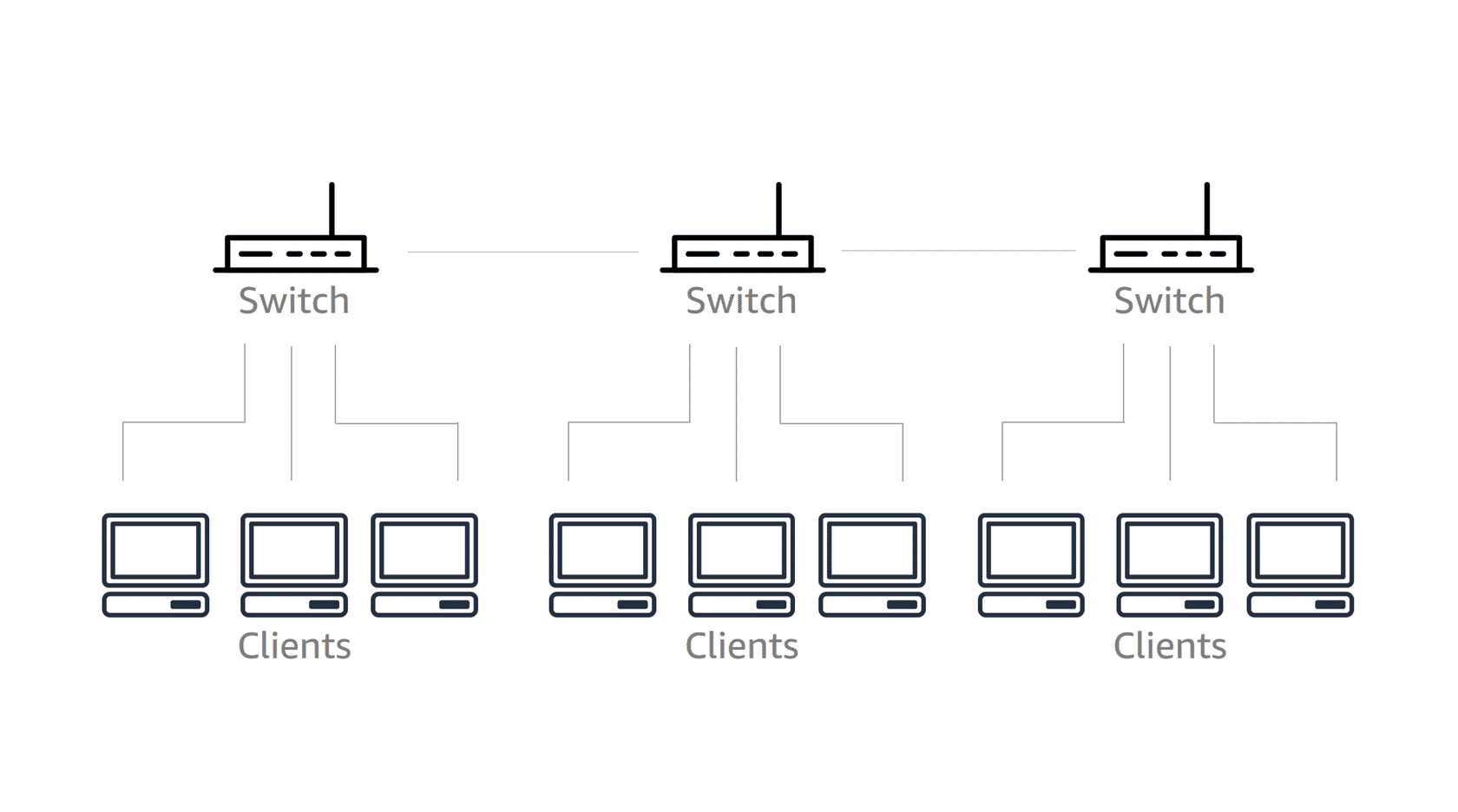
Mesh topology

Mesh

Physical topology

A *mesh* topology is a complex structure of connections that are similar to peer-to-peer, where the nodes are interconnected. Mesh networks can be full mesh or partial mesh. In a *partial-mesh* topology, *all devices are connected to at least two other devices*.

In a *full-mesh* topology, *all nodes are interconnected*. A full-mesh topology provides full redundancy for the network. It is an expensive topology because it requires each node to have multiple network adapters and cables. You will most likely find a full-mesh topology in a WAN.



Star-bus hybrid topology

Hybrid topology

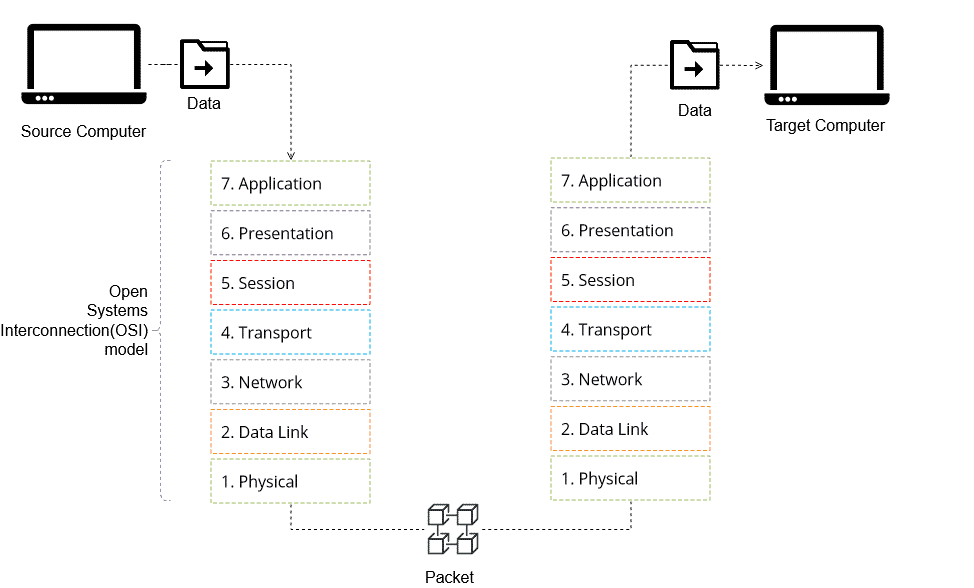
A hybrid topology *combines two or more different topology structures*. It is usually found in large organizations where separate departments have personalized network topologies to accommodate their network usage and other requirements.

Today, the *star-bus* topology is the most common hybrid topology.

### Open Systems Interconnection (OSI) model

The Open Systems Interconnection (OSI) model *defines a standard for how computers can share information over a network*, regardless of the hardware or software that they use. The model divides the processing of data that is sent over a network into seven layers.

The following diagram illustrates how data flows in an *OSI-compliant network* from a source computer to a target computer.



### Network protocols

A network protocol defines the *rules for formatting and transmitting data* between devices on a network. It typically operates at layer 3 (Network) or layer 4 (Transport) of the OSI model.

Network protocols fall into two general categories: *connection-oriented* protocols or *connectionless* protocols.



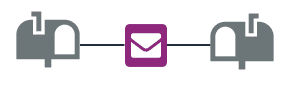
A connection-oriented protocol is similar to a phone call between two people

Connection-oriented protocol

* Establishes a connection
* Waits for a response
* Creates a *session* between the sender and the receiver
* Uses *synchronous* communication

Connectionless protocol

* Sends a message from one endpoint to the other, without ensuring that the destination is available and ready to receive the data
* *Does not require a session* between the sender and the receiver
* Uses *asynchronous* communication



A connectionless protocol is like sending a letter from one mailbox to another mailbox

### Examples of network protocols (To learn more, select a card)

Internet Protocol (IP)

The Internet Protocol establishes the rules for relaying and routing data in the *internet*.

Transmission Control Protocol (TCP)

The Transmission Control Protocol provides a reliable, *connection-oriented*, and ordered delivery of bitstreams over an IP network

TCP/IP

When TCP and IP are combined they form the TCP/IP protocol suite. *TCP/IP implements the set of protocols that the internet runs on*.

User Datagram Protocol (UDP)

The User Datagram Protocol uses a simple, *connectionless* communication model to deliver data over an IP network. Compared to TCP, it provides only a minimum set of functions. It is considered unreliable because it does not guarantee the delivery or ordering of data. Its advantage is that it has a lower overhead and is faster than TCP.

## Key takeaways

* A local-area network (LAN) connects devices in a *limited geographical area*, and a wide-area network (WAN) connects devices in a large geographical area.
* A *network management model* defines how data is managed, and how applications are hosted in a network. *Client-server* and *peer-to-peer* are two common network management models for a LAN.
* A *network topology* shows how nodes connect to each other. *Star* and *hybrid* are example patterns of a network topology.
* The *Open Systems Interconnection (OSI)* model defines a standard for how computers can share information over a network, regardless of the hardware or software that they use.
* A *network protocol* defines the rules for formatting and transmitting data between devices on a network. Which network topology should you use if you want every node in a network to be directly connected to one central hub?
* A business is housed on several floors of the same building. Each employee has their own desktop in their cubicle. The employees must have access to a central file server, and multiple print servers on each floor. Which type of network would best suit the needs of this business?

Page 3. Internet Protocol (IP)

### At the core of the lesson, you will learn how to:

* Describe the Internet Protocol (IP) and its features
* Explain the purpose of an IP address and its notation
* Convert an IP address to binary
* Distinguish between different classes of IP addresses
* Describe port numbering and its use

### What is IP?

IP is a network protocol that establishes the *rules for relaying and routing data in the internet.*

* IP is a critical standard within the larger *TCP/IP* protocol suite when it is combined with the connection-oriented *Transmission Control Protocol (TCP)*. TCP/IP implements the set of protocols that provides a crucial service for the internet because it enables the successful routing of network traffic among devices on a network. Some useful points about TCP/IP include that it:
* Uses *IP addresses* to identify devices
* Uses *port numbers* to identify endpoints
* Supports *subnetting* to subdivide a network

### IP addresses

An IP address *uniquely identifies a device on a network*. Each device on a network has an IP address, and it serves two main functions:

* It identifies a host and a network.
* IP addresses are also used for location addressing.

IP addresses can be assigned to devices in a *dynamic* or *static* way.

* A device with a *dynamic address* has an assigned IP address that can change. This feature is useful in scenarios where devices might leave and come back to a network, like a work-assigned laptop that travels from the work network to the user's home network.
* A device with a *static address* has an IP address that does not change. This feature can be useful in many scenarios, such as servers or printers on a network that other devices often connect to.

IP addresses can also be made *public* or *private*:

* A *public IP address* is an IP address that can be accessed over the internet. A public IP address is similar to a public phone number that is listed in a phone book or on the internet. The phone number is discoverable by anyone who wants to call it. A public IP address is a globally unique IP address that is assigned to a computing device that must access the internet.
* A *private IP address* is assigned to computers within a private network and they cannot be accessed from the internet. Private addresses work well for use cases when an address is assigned to a device that you do not want other devices to know about. Returning to the phone number analogy, a private IP address would be similar to a private listing or a personal phone number that is not made publicly available. For example, the application and database servers in your data center are assigned private IP addresses because you might not want other devices to know about these servers.

When a network is assigned a range of IP addresses, such as 10.0.0.01–10.255.255.255, a few addresses have a special purpose. They are *not* assigned as host addresses. These addresses are the default router address and the broadcast address.

Example: Suppose the range of IP addresses is 10.0.0.0–10.255.255.255.

* The *default router address* is typically the second address in the range: 10.0.0.1
  + The default router address is also known as the *gateway address*, and it is the IP address of the network router.
* The *broadcast address* is the last address in the range: 10.255.255.255
  + The broadcast address is used to transmit messages to all devices that are connected to a network. If a message is sent to a broadcast address, then all nodes on the network can receive it.

### Converting an IP address into binary

An IP address consists of *four numbers from 0 to 255 separated by a period*,which is also known as a *dotted quad* (for example, 10.15.238.21). This format follows the *IPv4 standard.* It is important to note that the numbers in the IP address *identify both the network and the device on the network*.

To understand IP addressing, it is often better to convert the number to binary. A binary number is expressed in the base-2 numeral system, and it is *consists of only zeroes and ones*:

* The value of 0 or 1 is known as a *binary digit*, or *bit*.
* In an IPv4 address, *each of the four numbers* between the dots is *an 8-bit binary number*. Thus, the entire address is a 32-bit binary number.

The following table can be used to convert an 8-bit binary number to a decimal, or a decimal to an 8-bit binary number:

| Bit Position | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Binary Power | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| Decimal  Value | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

Example 1: Converting the decimal value 13 to binary

It is useful to visualize a binary conversion. To do so, consider the decimal value *13*. To convert this value into binary, first consider the following chart based on what you know from the previous table:

| Decimal Value | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Binary Power | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| Bit: 1 or 0 ? |  |  |  |  | 1 | 1 | 0 | 1 |

From the preceding chart, you can determine how many powers of 2 can be used to add up to the value of *13*. If you determine what powers of two come closest to the number 13 *without* going over, you can eliminate 27, 26, 25, and 24. Each of these values has a decimal value that is greater than 13. From the chart above, the correct starting power would be 23 because this value equals *8*. Beginning here, you enter a *1* in the chart beneath 23 to represent the bit.

Subtracting 8 from 13 provides the decimal value remainder of *5*. Repeating the previous step, you can find the next power of 2 that can be used to further reduce the decimal value that is being converted. The decimal value for the remainder of *5* would be 22 because it represents the decimal value of 4. To represent the bit, you can then put a *1* beneath the binary value for 22 in the chart.

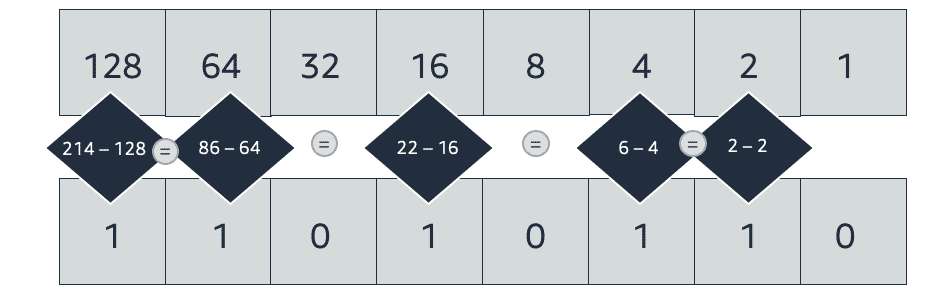
Continuing along the same pattern, you can then subtract 4 from the the remainder decimal value of 5 to get *1*. Working left from right, you can observe that you cannot use the 21 binary power because it would give you a value of 2—which is greater than the remainder. Thus, you can place a bit value of *0* beneath the decimal value 2. The next binary power, 20, has a decimal value of *1*. Because the remainder is currently *1*, 20 can be subtracted from the remainder to provide a remaining decimal value of *0*. A bit value of *1* is placed beneath the binary value 20.

After you convert the number to binary and you have processed the remainders down to 0, you can finally determine the binary value of the number. In this instance, the number *13* has a binary value of 1101.

### Example 2: Converting the decimal value 214 to binary

Evaluate the decimal value *214* to determine its binary value.

1. Find the *highest decimal value that is less than or equal* to 214. This value is *128*.
2. *Set the bit* in the 128 column to *1*.
3. *Subtract* 128 from 214 to *get a remainder* of *86*.
4. Find the *highest decimal value that is less than or equal* to 86. This value is *64*.
5. *Set the bit* in the 64 column to *1*.
6. *Subtract* 64 from 86 to *get a remainder* of *22*.
7. *Repeat the preceding pattern* for each remainder value *until the remainder* is *0*.

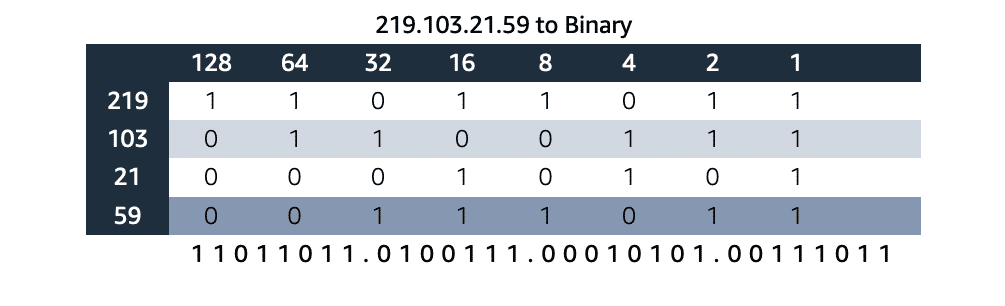


The decimal value *214* coverts to *11010110* in binary

NOTE: If you evaluate the diagram on the left against the instructions in Example 1, you can observe the process of determining a binary value. Bit values are placed beneath the corresponding decimal values for the binary powers that are used to reduce the original value of *214* down to *0*. The bit values for 32, 8, and 1 are all set to *0* because they were not compatible power values for reducing the decimal value remainders down to 0.

### Example 3: Converting the IP address 219.103.21.59 to binary

Use the method described in Example 1 and Example 2 to *convert each of the four numbers in the IP address 219.103.21.59 to binary*.



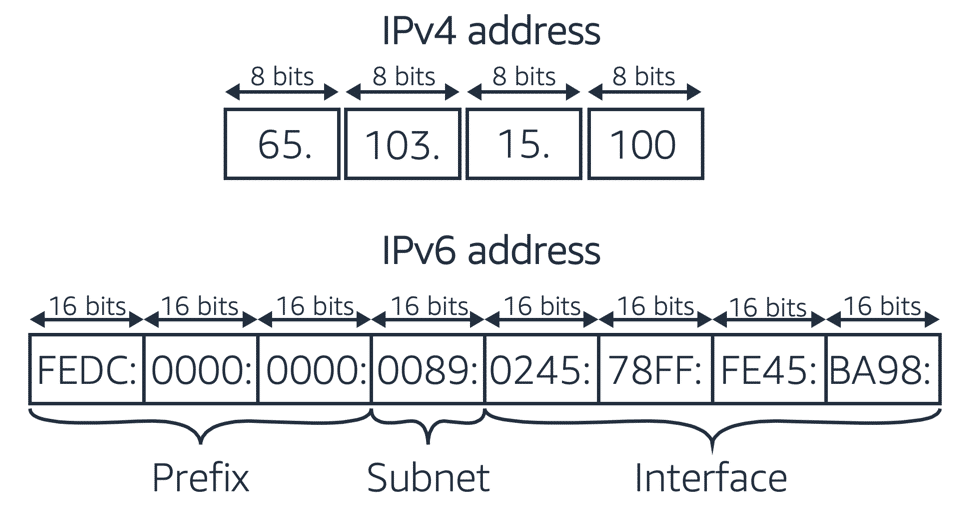
In binary, the IP address *219.103.21.59* converts to *11011011.0100111.00010101.00111011*

### What is IPv6?

A newer *IPv6 standard* extends the range of IPv4 addresses by a factor of 1,028. It uses a *group of hexadecimal numbers that are separated by eight colons (:)*. This emerging standard is designed to replace IPv4 eventually. It is also designed to handle packets more efficiently, increase security, and improve performance. It is important to note that in IPv6 addresses, *the numbers identify both the network and the device on the network*. An IPv6 address looks like the following example:

* 2600:1f18:22ba:8c00:ba86:a05e:a5ba:00FF

When you compare IPv4 to IPv6, you can observe the difference in length between the two numbers. The extra digits in the IPv6 address enable an expanded number of available addresses. Each decimal value now allows for *16 bits* instead of the 8 bits that IPv4 provides. IPv4 provides a maximum of 4.2 billion addresses. However, as the need for connected devices and expansive networks continues to grow, more addresses are needed. IPv6 provides 340 trillion, trillion, trillion addresses—which provides a large amount of space to secure an IP address on a network for the ever-expanding list of smart phones, tablets, computers, servers, internet of things (IoT) devices, and any other connected devices.



Comparison of IPv4 (32-bit) addresses and IPv6 (128-bit) addresses

## 

## 

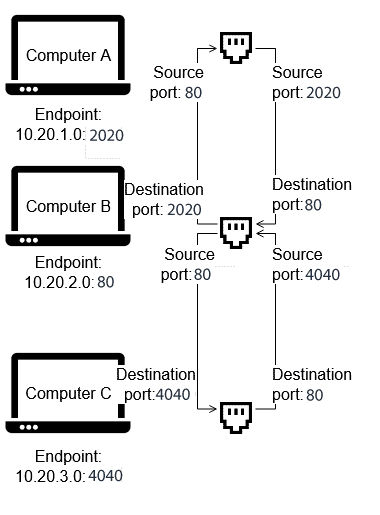
## 

## 

## Port numbers

A *port number* allows a device in a network to *further identify the other devices or applications that communicate with it*. A device might send or receive data from multiple devices at the same time. A port number, in combination with an IP address, enables the device to determine the exact source or destination of the data, which is also known as the *endpoint*.

A good example would be to think of a port number like an extension to a phone number. Multiple people might call a phone number, but each person could be trying to get to a different person based on the extension that they use. This way, everyone can use the same number, but provide unique extensions to ensure that their calls are routed correctly. A computer can receive and send information in a similar way. It can enable multiple types of data to be sent to the same IP address, and still be parsed and acted on in separate and unique ways. This functionality enables a computer download a file over File Transfer Protocol (FTP) from an Amazon Simple Storage Service (Amazon S3) bucket, stream a live video from Twitch, and receive email messages—all at the same time.



Example of using port numbers

Port number example

In this example, an application on Computer B receives data simultaneously from applications on Computer A and Computer C. Both Computer A and Computer C run other applications that communicate to other computers over the network. Because each communicating application on computers A, B, and C is identified by a *unique endpoint* (or *combination of IP address and port number*), messages from the three applications can reach their correct destinations.

## Key takeaways

* An *IP address* uniquely identifies a device on a network and enables communication to it.
* An *IPv4 address* consists of four numbers in the range of 0–255, and each number is separated by a dot (.).
* *Working with IP addresses* often requires converting them to binary.
* A *port number* further identifies an application or a service on a device. When the port number is combined with the device's IP address, it represents an *endpoint*.

Page 4 IP subnetting

### At the core of the lesson, you will learn how to:

* Describe the purpose of IP subnetting
* Use the Classless Inter-Domain Routing (CIDR) notation to specify subnet address ranges

### What is a subnet?

A subnet is *network inside of a network*. Subnetting is process of creating smaller networks by *dividing a large block of addresses* into several contiguous subdivisions. Subnets become useful when you must manage multiple networks. Organizations with multiple office locations can create individual subnets for their multiple locations. A coffee shop might manage a private network for their business operations and a public network for their patrons. Some important aspects of a subnet include these points:

* A subnet *creates multiple logical networks* with different ranges of IP addresses. As an example, an organization could separate their networks. They could have an operational network and another network for machinery and IoT devices.
* *Classless Inter-Domain Routing (CIDR)* notation is used to specify subnet IP address ranges. This system provides a shorthand for describing how the size of a network.
* Devices that are in the *same subnet can communicate with each other without using any routers*.
* Networks with Class A, B, or C IP addresses can be subnetted.

### About IP address classes

An important aspect to understanding a network is understanding the record class that is associated with it. IP addresses are divided into classes *based on their size and the subnetting capabilities* of the network that they support. The potential for an organization to divide their network space is determined by the *class* of their IP addresses. Three of these classes are commonly used, and they provide a logical separation between the networks:

* Class A (0.0.0.0–127.255.255.255): Meant to be used for extremely large networks, like networks that are built by internet service providers (ISPs).
  + A Class A record provides for up to 126 networks, which can each support up to 16,777,214 addresses!
  + It is rare to see a single organization own a Class A record, but it does happen. Examples include Apple, who owns the entire 17.XXX.XXX.XXX space; and the Massachusetts Institute of Technology (MIT), who owns the 18.XXX.XXX.XXX domain.
* Class B (128.0.0.0–191.255.255.255): Suitable for medium to large networks, for example, for enterprises and organizations.
  + Class B records can provide a capacity for up to 16, 384 networks, accounting for approximately 65,000 addresses each.
* Class C (192.0.0.0–233.255.255.255): Suitable for small networks, such as a home network or a small business network.
  + Class C records provide 2,097,150 networks that can each allow for up to 254 addresses.

All networks—no matter what their size is—reserve some IP addresses for specific functions:

* Unknown address (0.0.0.0): This address can have different meanings, depending on context. It might be used to represent an unknown or unroutable request, or it might be used to represent any address on a server. It virtually never refers to an individual device because it is not generally an assignable address.
* Loopback (127.0.0.1): An address that returns a message back to its originating device. This address is used for testing and debugging purposes. It is also used to allow a specific configuration of inter-domain routing between network routers.
* All devices – broadcast (255.255.255.255): An address that represents all devices that are connected to the network. The broadcast address is reserved so that information can be sent to every device on a network. It reduces the routing workload that would be incurred by individually routing information to each address.

It is useful to think about IP address classes when you consider subnetting. The IP address class provides a limit to how many networks can be created in that IP space. A very large organization might have a Class B record that enables them to create up to 16,382 networks, which can each support approximately 65,000 addresses. By using subnets, a network administrator might use this IP address to manage hundreds of individual networks across multiple cities, buildings, floors, and even departments.

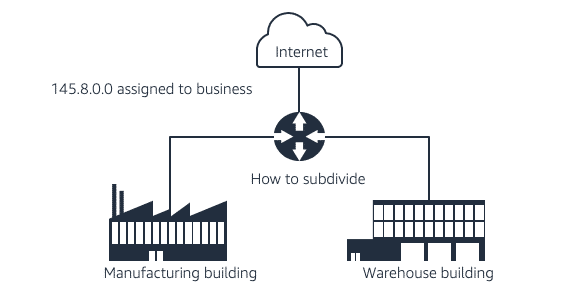
### Purpose of subnetting

Subnetting can be used to organize and optimize a network. For example, you can use subnets to:

* Isolate different parts of the network
* Apply different levels of security to different parts of the network
* Relieve congestion on the network

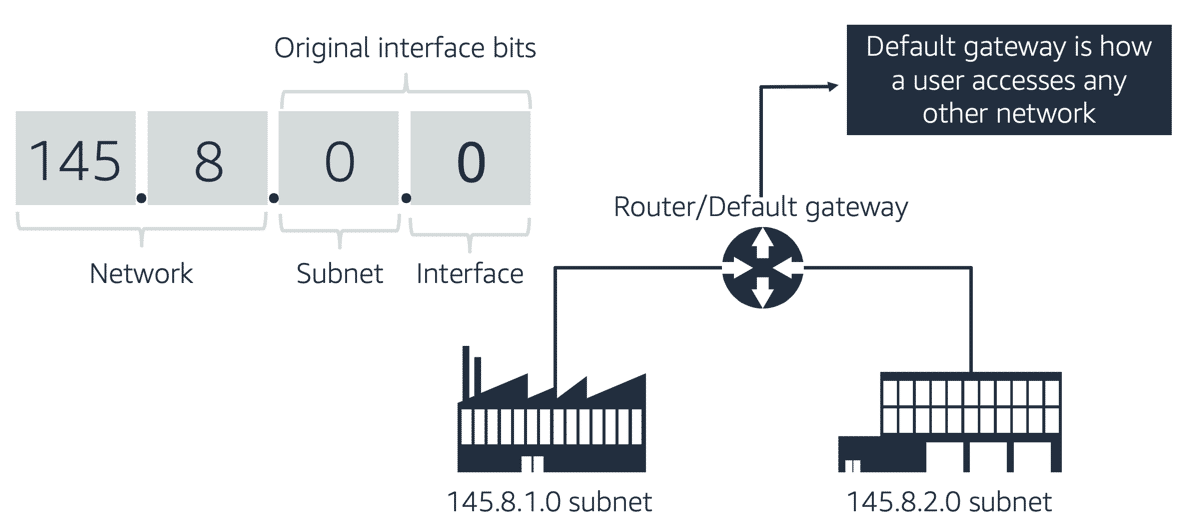
Consider the following example:

* A company has a network that connects two of its buildings, the Manufacturing building and the Warehouse building. The network was assigned the *IP address range of 145.8.x.x.* The company wants to create two separate subnets to provide a level of isolation between the devices in each building.



A manufacturing building and a warehouse that are on different subnets in one network

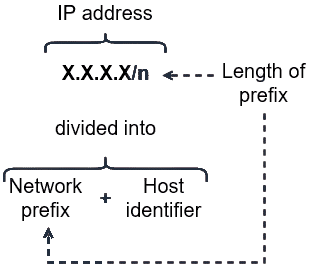
* The *subnet* for the Manufacturing building is *assigned a subset of the network's IP address range*—specifically, *145.8.1.x*.
* Likewise, the Warehouse building is *assigned a different subset of the network's IP address range,* which is *145.8.2.x*.
* A *subnet mask* defines which section of the address *identifies the network*, and which section *identifies the hosts*.
* A *router* or *default gateway* is used by devices on one network to communicate with the devices in another network.



Each subnet's IP address range is a distinct subset of the overall network's IP address range

### Classless Inter-Domain Routing (CIDR) notation

The Classless Inter-Domain Routing (CIDR) format is *used to specify IP address ranges* when you create a network or a subnet.



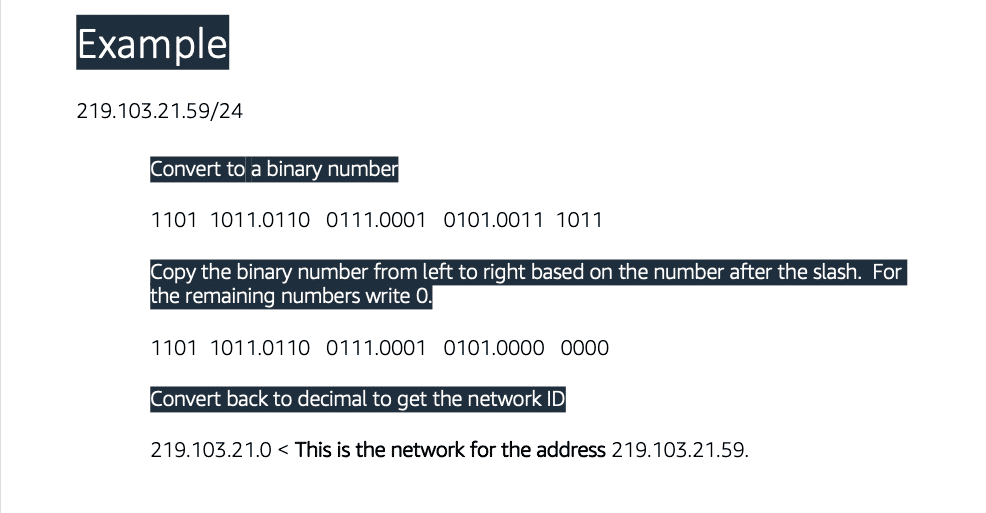
CIDR format

The CIDR format specifies a block (which is known as a *CIDR block*) of IP addresses. This block is represented by using the format *x.x.x.x/n*, where:

* *x.x.x.x* is an *IP address*. Recall that an *IPv4 IP address is a 32-bit number* that is represented as *four numbers separated by dots (.)*. Thus, each *x* is an *8-bit number (a byte) that can have a value between 0–255*. The IP address is logically divided into a *network prefix* and a *host identifier*. The network prefix identifies the network, and the host identifier identifies the host within the network.
* */n* specifies the *length in bits* of the *network prefix portion of the IP address* (starting from the leftmost bit). For an IPv4 IP address, the value of *n* can be from 0–32. The larger the value of *n*, the smaller the range size becomes, which results in a smaller number of usable IP addresses.

Steps for *determining the network ID or prefix* in a CIDR block:1

* *Convert* the IP address to *binary*.
* *Note* the *number of bits that are designated after the slash*, from left to right. *Set all other numbers to 0*.
* *Convert* the resulting binary number *back to decimal*.



Example of how to determine the network ID from CIDR notation

Additional information: To help determine the address range of a CIDR address, you can use the online calculator at the [Online IP CIDR / VLSM Supernet Calculator](http://www.subnet-calculator.com/cidr.php) webpage.

### Examples of CIDR block ranges

The table below provides examples of CIDR block ranges, their corresponding bit representation, and their resulting address range.

| CIDR Block | Bit Representation | Corresponding Address Range |
| --- | --- | --- |
| 10.50.1.0/24 | 00001010.00110010.00000001.xxxxxxxx | 10.50.1.0 - 10.50.1.255 |
| 10.50.1.0/27 | 00001010.00110010.00000000.000xxxxx | 10.50.1.0 - 10.50.1.31 |
| 10.50.1.132/32 | 00001010.00110010.00000001.10000100 | 10.50.1.132  (single address) |
| 0.0.0.0/0 | xxxxxxxx.xxxxxxxx.xxxxxxxxx.xxxxxxxx | 0.0.0.0 - 255.255.255.255  (all addresses) |

## 

## 

## 

## 

## 

## 

## 

## 

## Key takeaways

* IP subnetting enables you to *divide a single network into multiple logical subnetworks*.
* *Classless Inter-Domain Routing (CIDR) notation* is used to specify the IP address range of a subnet.
* A CIDR block uses the format *x.x.x.x/n*, where:
  + *x.x.x.x* is an *IP address*
  + */n* specifies the *length in bits* of the *network prefix portion of the IP address*

PAGE NO 5 :- IP networking in the AWS Cloud: Amazon VPC

### At the core of the lesson, you will learn how to:

* Explain virtual networking in the cloud with Amazon Virtual Private Cloud (Amazon VPC)
* Describe the key components of a VPC
* Relate subnetting and CIDR block addressing to Amazon VPC features

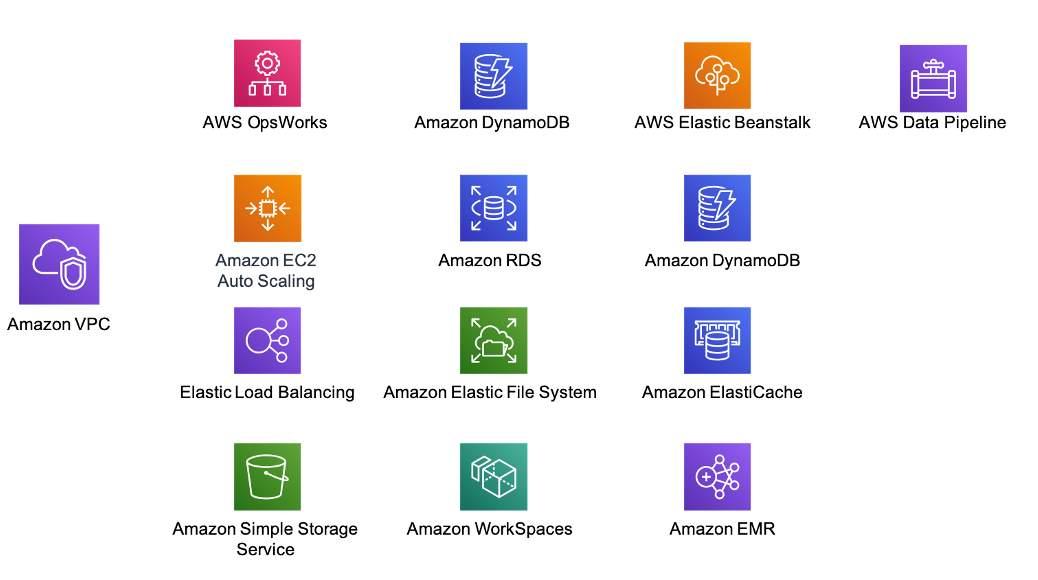
### What is Amazon VPC?

Amazon Virtual Private Cloud (Amazon VPC) is *a service* that you can use to *provision a logically isolated section of the AWS Cloud*, which is called a *virtual private cloud*, or *VPC*. With a VPC, you can launch your AWS resources in *a virtual network that you define*. A VPC:

* Enables you to *create a private network* in the AWS Cloud that uses many of the same concepts and constructs as an on-premises network
* Gives you *control over your virtual networking resources*, including –  
  + Selecting an *IP address range*
  + Creating *subnets*
  + Configuring *route tables* and *network gateways*
* Enables you to *customize its network configuration*
* Enables you to *use multiple layers of security*

### Using other AWS services with Amazon VPC

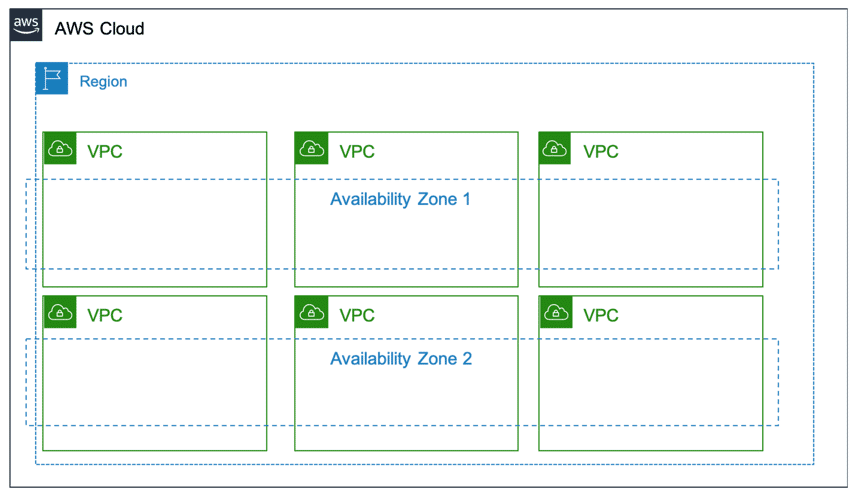
* Amazon VPC is an AWS foundational service and it *works with many AWS services*. For example, Amazon Elastic Compute Cloud (Amazon EC2) instances are deployed into a VPC.
* *Understanding and implementing Amazon VPC will enable you to fully use other AWS services.*

**

Amazon VPC is a foundational AWS service, and it works with many AWS services

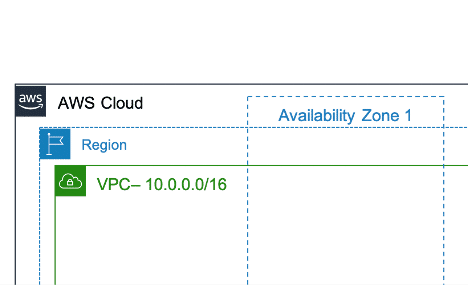
### Amazon VPC features

* A VPC:
  + Is *dedicated* to an AWS account
  + Belongs to a single *AWS Region*
  + Can span multiple *Availability Zones*
  + Is *logically isolated* from other VPCs
* You can create *multiple VPCs in an AWS account* to separate networking environments
* You can create subnets in a VPC, though we recommend fewer subnets to reduce the complexity of the network topology



Multiple VPCs that span different Availability Zones in an AWS Region

### IP addressing in Amazon VPC



The IP address range of a VPC is specified as a CIDR block

* When you *create a VPC*, you must *specify the IPv4 address* range by choosing a *CIDR block*, such as *10.0.0.0/16*.
* A VPC address range could be as large as */16* (65,536 addresses) or as small as */28* (16 addresses).
* *IP addresses should not overlap* with the addresses of other networks that a VPC is connected to.
* The *address range of the VPC cannot be changed* after the VPC is created.

## 

## 

## 

## Amazon VPC components

You can use the following components to configure networking in a VPC:

START

Component 1

## Public subnet or private subnet



A subnet is a segment of a VPC address range. You can launch AWS services in a subnet.

* CIDR blocks define subnets.
* AWS reserves the first four IP addresses and the last IP address of every subnet for internal networking purposes.
* A public subnet is accessible from within the VPC and from the internet.
* A private subnet is only accessible from within the VPC, and it is not accessible from the internet.

1234

Component 2

## Route table



A route table contains a set of rules, called *routes*, that are used to determine where network traffic is directed. Each subnet in a VPC must be associated with a route table. The table controls the routing for the subnet. A subnet can only be associated with one route table, but you can associate multiple subnets with the same route table.

To learn more about route tables, refer to [Route tables](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Route_Tables.html) in the AWS Documentation.

1234

Component 3

## Security group



A security group is a virtual, stateful firewall that controls inbound and outbound network traffic to AWS resources and EC2 instances.

To learn more about security groups, refer to [Security groups for your VPC](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html) in the AWS Documentation

1234

Component 4

## Network access control list (network ACL)



A network access control list (network ACL) is an optional layer of security for a VPC. It acts as a stateless firewall for controlling traffic in and out of one or more subnets. To learn more, refer to [Network ACLs](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html) in the AWS Documentation.

1234

## Key takeaways

* *Amazon Virtual Private Cloud (Amazon VPC)* is a service that you can use to build a custom-defined network in the AWS Cloud
* The IP address range of a VPC is defined by using a *CIDR block*
* You can create the following components in a VPC:  
  + Public subnet or private subnet
  + Route table
  + Security group
  + Network access control list (network ACL)

# Other common protocols

### At the core of the lesson, you will learn how to:

* Identify other types of communication protocols
* Describe common transport, application, and network management protocols
* Use tools to discover information about network communications

### Transport protocols

## Transmission Control Protocol (TCP)

Recall that you first learned about TCP/IP earlier in this module. When Transmission Control Protocol (TCP) is combined with Internet Protocol (IP), they form the TCP/IP protocol suite, a set of protocols that the internet runs on. TCP/IP is a *connection-oriented protocol*. It defines how to establish and maintain network communications where application programs can exchange data. Data that is sent through this protocol is divided into smaller chunks called *packets*.The goal of TCP/IP was to support an interconnection of networks, which was referred to as an *internetwork*, or *internet*. The internet comprises the groups of networks that communicate over this protocol.

Reference: Parziale, L., Britt, D., Davis, C., Forrester, J., Liu, W., Matthews, C., & Rosselot, N. (2006, December). TCP/IP Tutorial and Technical Overview. Retrieved from: [IBM Redbook: TCP/IP Tutorial and Technical Overview](https://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf).

### User Datagram Protocol (UDP)

The User Datagram Protocol (UDP) uses a simple, *connectionless communication* model to deliver data over an IP network. Compared to TCP, UDP only provides a minimum set of functions. It is considered to be *unreliable because it does not guarantee the delivery or ordering of data*. Its advantage is that it has a lower overhead, and it is faster than TCP.UDP is *used by applications that value speed over guaranteed delivery*. Examples include video chat and video streaming. A missed packet might cause a short pause in the video, but the video will still be mostly understandable. However, if the users must wait for all the packets to be confirmed and ordered correctly, the delays can severely impact the quality of their experience.

### Application protocols

### Hypertext Transfer Protocol (HTTP)

HTTP is the protocol that is used to reach webpages. A *full HTTP* address is expressed as a *uniform resource locator (URL)*.



Example of a URL

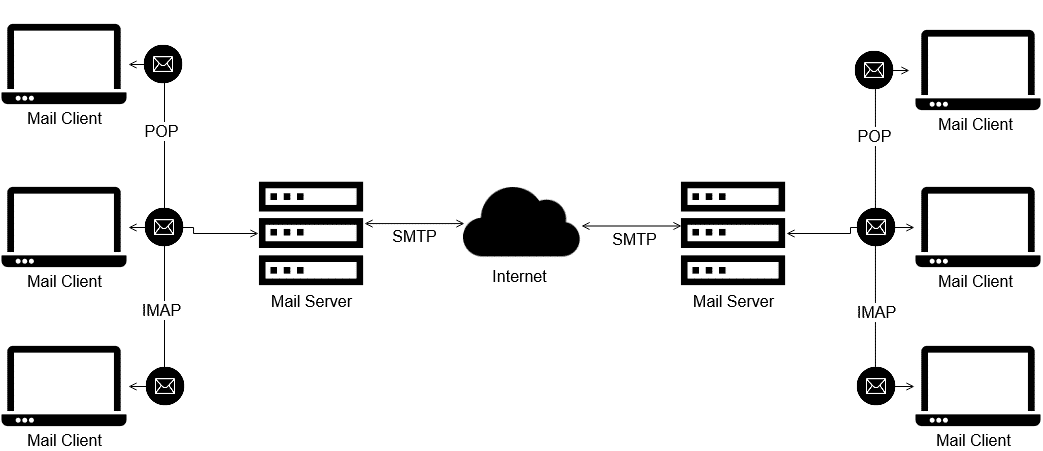
### Secure Sockets Layer (SSL) and Transport Layer Security (TLS)

*Secure Sockets Layer (SSL)* is a *standard for securing and safeguarding communications* between two systems by using encryption.

*Transport Layer Security (TLS)* is an *updated version of SSL* that is *more secure*. Many *security and standards organizations*—such as the *Payment Card Industry Security Standards Council (PCI SSC)*—require organizations to use *TLS version 1.2* to retain certification.

### Mail protocols (SMTP, POP, IMAP)

* The *Simple Message Transfer Protocol (SMTP)* is used to transfer email messages between mail servers.
* *Post Office Protocol (POP)* and *Internet Message Access Protocol (IMAP)* are used *by email clients* such as *Mozilla Thunderbird* or *Microsoft Outlook* to retrieve email messages from the mail server.



Mail protocols: SMTP, POP, and IMAP

### Remote Desktop Protocols (RDP and SSH)

* *Remote Desktop Protocol (RDP)* is a protocol that is used to access the desktop of a remote *Microsoft Windows* computer. *Use port 3389* with clients that are available for Microsoft Windows, Unix, Linux, macOS, and other operating systems.
* *Secure Shell Protocol (SSH)* is a protocol that opens a secure *command line interface (CLI)* on a remote *Linux* or *Unix* computer.

### Application protocol port numbers

Application protocols—such as *Hypertext Transfer Protocol (HTTP)* and *File Transfer Protocol (FTP)*—have assigned port numbers.

These numbers are data endpoints. The ports provide devices with a way to understand what to do with the data that they receive. For example, a computer might download a file over FTP. The computer connects to the server and downloads the data over port 21. The computer will know how to handle that data because of the port that it used. The computer will thus be able to complete the download. The following table shows the network protocol and the port number that are used by common application protocols.

| Application Protocol | Transport Protocol | Port Number |
| --- | --- | --- |
| HTTP | TCP | 80 |
| HTTPS | TCP | 443 |
| FTP | TCP | 21 |
| SSH | TCP | 22 |
| DNS | TCP | 53 |

### 

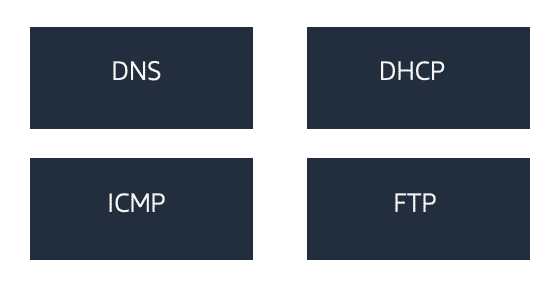
### 

### 

### Management and support protocols

* *Management protocols* are used to configure and maintain network equipment.
* *Support protocols* enable and improve network communications.

The following diagram describes some examples of management and support protocols.



Domain Name System (DNS)

Domain Name System (DNS) is a database for domain names. It is similar to the contacts list on a mobile phone. The contacts list matches people (or organization) names with phone numbers. DNS functions like a contacts list for the internet. It converts human-readable domains names (such as *example.com*) into IP addresses. DNS servers automatically map IP addresses to domain names.

Dynamic Host Configuration Protocol (DHCP)

Dynamic Host Configuration Protocol (DHCP) automatically assigns IP addresses, subnet masks, gateways, and other IP parameters to devices that are connected to a network.

Internet Control Message Protocol (ICMP)

Internet Control Message Protocol (ICMP) is used by network devices to diagnose network communication issues and generate responses to errors in IP networks.

File Transfer Protocol (FTP)

File Transfer Protocol (FTP) is a network protocol that enables the transfer of files from one computer to another. FTP performs two basic functions: PUT and GET. If you have downloaded something, such as an image or a file, then you probably used an FTP server.

### Common network utilities

When you work with networks, it is important to check network performance, bandwidth usage, and network configurations. There are a few common network utilities that you can use to quickly troubleshoot network issues. These tools can help ensure uninterrupted service and prevent long delays.

Example of common network utilities include:

* *ping* tests connectivity. This tool tests if the remote device (server, desktop) is on the network.
* *nslookup* queries the DNS and its servers. It shows the IP addresses that are associated with a given domain name.
* *traceroute* enables users to see the networking path used. It is helpful for troubleshooting connectivity problems.
* *telnet* is used for service response. This tool tests if the service that runs on the remote device is responding to requests.

### Activity: ping and nslookup

Open a Terminal window or a Command Prompt window, and complete the following steps:

1. Enter *ping* [*amazon.com*](http://amazon.com/)
   * This command will return the IP address of the responding server.
   * You can view the additional connectivity information.
2. Enter *nslookup* [*amazon.com*](http://amazon.com/)
   * You can view the path that your computer takes to reach [amazon.com](http://amazon.com/).

### Activity (Linux or macOS only): traceroute

Open a Terminal window or a Command Prompt window, and complete the following steps:

Enter *traceroute* [*amazon.com*](http://amazon.com/)

* You can observe how many hops the request took.
* You can observe the latency of each hop.
* If a hop lists an asterisk (\*), it means that the hop timed out.

## Key takeaways

* TCP and UDP are *transport protocols*. TCP is *connection-oriented* and UDP is *connectionless*.
* Common *application protocols* that are used on the internet include HTTP, TLS/SSL, SMTP, and FTP.
* Common *network management* and *support protocols* include DNS, DHCP, and ICMP.
* Common *utilities* that are used to discover and troubleshoot network communication include *ping*, *nslookup*, and *traceroute*.

Page 7 Introduction to network security

### At the core of the lesson, you will learn how to:

* Recognize the importance of network security
* List types of network security threats
* List techniques to mitigate network security threats

### What is network security?

Network security is a set of *guidelines and configurations* that are designed as preventative measures to protect the underlying networking infrastructure. Network security:

* Is designed to preserve the *usability and integrity of a network, and the data that flows through the network.*
* Is concerned with protecting *a computer network and its resources* from *unauthorized access, misuse, modification, or destruction*.
* Combines *policies and procedures* with *hardware and software solutions* to achieve this goal.

### The importance of network security

According to the International Council of Electronic Commerce Consultants (EC-Council), "Information collected and stored electronically by the organizations is sensitive, crucial, and private information. It most likely includes crucial business data, customers’ personal information, and other information that is valuable for business transactions. Unauthorized access to the network and its data creates a threat to security. Several statutory privacy requirements should be followed like HIPAA, FEPRA, and others to prevent the data from unauthorized dissemination. Failure to do so will leave you responsible for the consequences."

For more information, refer to the [How good is your Network Security?](https://www.eccouncil.org/network-security/) page on the EC-Council website.

### Network security threats

A security threat is any attempt to expose, alter, disable, destroy, steal, or gain unauthorized access to an organization’s network to steal data or perform a malicious activity.

### Example of network security threats (To learn more, select a card)

Malware

Software that purposefully intends to damage a computer, client, server, or network. Ransomware, spyware, and viruses are all examples of malware.

Social engineering

Using deception to trick a person into reveal personal or confidential information. A common example is phishing attacks that disguise themselves as requests for information from trusted service providers.

Network attacks

Actions that seek to gain unauthorized access to a network to steal data or perform a malicious activity. A common example is the distributed denial of service (DDOS) attack, which creates many requests on a service. The service then becomes so overwhelmed that it becomes inaccessible to users or devices that might need it.

### Examples of network security solutions (To learn more, select a card)

Anti-malware

Software that protects both the network and the devices on the network

Encryption

Scrambling data at rest or in transit to keep it private so that it can only be read by the intended recipient

Firewall

A hardware or software solution that filters incoming and outgoing traffic in a network to prevent unauthorized access

Intrusion detection system (IDS)

A hardware or software solution that monitors traffic in the network to detect intrusions, and generates alerts when they occur.

## Key takeaways

* *Network security* is concerned with the protection of a computer network and its resources from unauthorized access, misuse, modification, or destruction.
* Types of network security threats include *malware*, *social engineering*, and *network attacks*.
* Types of network security solutions include *anti-malware*, *encryption*, *firewalls*, and *intrusion detection systems*.

Page 8 Threat mitigation

### At the core of the lesson, you will learn how to:

* Define the goal of threat mitigation
* Describe common threat-mitigation methods

### What is threat mitigation?

If your computing environment has been compromised or infected, you or your organization must take corrective actions to find and eliminate the threat. Corrective actions, such as prevention strategies or solutions, are put in place to combat or reduce IT threats on a computer, server, or network.

Mitigation is a detection and protection policy that is adopted to safeguard networks.

### Threat-mitigation methods

Mitigation methods include:

* Anti-malware – Malware detection
* Encryption
* Authentication
* Firewalls
* IDS
* Training

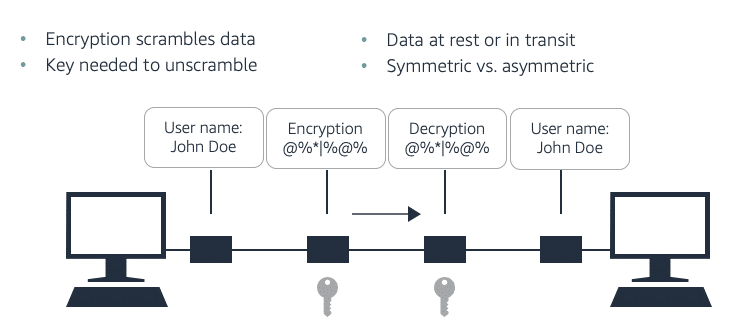
Penetration testing and vulnerability scanning are also essential.

### Anti-malware

* Anti-malware protects the device and the network.
* Malware is malicious computer software that is developed to act out malicious intentions. Malware is designed to damage computer hardware or steal data. Types of malware include viruses, worms, trojan horses, rootkits, ransomware, and keyloggers.
* Malware detection examines networked devices and the data on those systems to detect malware. Malware-detection software is effective at detecting malware because it can involve the use of multiple mechanisms and methods. Types of malware-detection software include anti-virus software, anti-spyware, anti-spam, and firewalls.

### Encryption

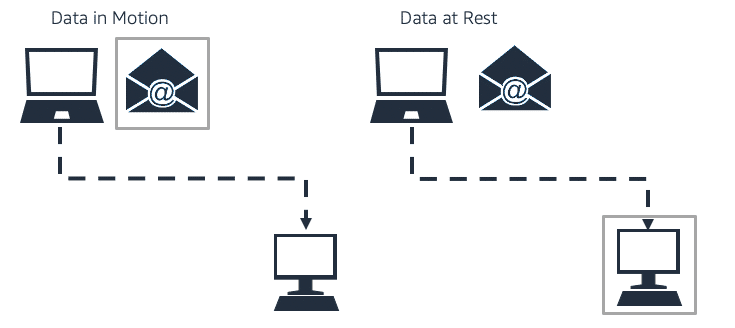
* Encryption uses an algorithm to encrypt (or scramble) data.
* A decryption key is needed to decrypt (or unscramble) the data.
* Sensitive data that is at rest or in motion should be encrypted.



Encryption uses an algorithm to encrypt data. A decryption key is needed to decrypt the data.

### Data in motion versus data at rest

Imagine that an email message is being sent. When the email message is sent from the first computer to the second computer, the data is *in motion*. After the email message reaches the second computer's inbox and is waiting to be opened, the data is *at rest*.



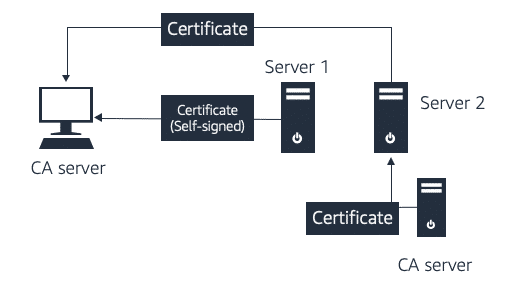
Data in motion versus data at rest

### Encrypting data in motion and data at rest

Encrypting data at rest is vital for regulatory compliance. It helps ensure that sensitive data that is saved on disks is not readable by any user or application that does not have a valid key. Some compliance regulations—such as the Payment Card Industry Data Security Standard (PCI DSS) and the U.S. Health Insurance Portability and Accountability Act (HIPAA)—require the encryption of data at rest throughout the data lifecycle.

### Certificate authentication and encryption

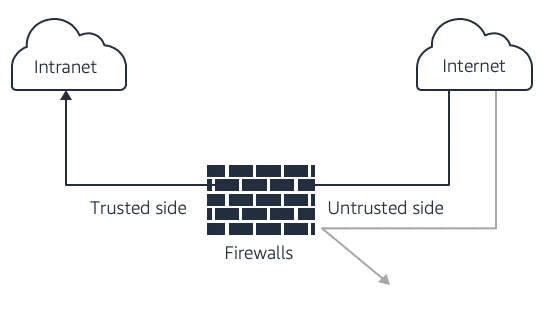
* *Digital certificates* are electronic credentials that are used to *represent the online identities* of individuals, computers, and other entities on a network. Digital certificates are similar to personal ID cards.
* A certificate with a *public key* and a corresponding *private key* can be used for *encryption* and *decryption*. When only the public key is used, the certificate establishes trust and performs encryption.
* There are two types of certificates—*Certificates that are signed by a certificate authority (CA)* and *self-signed certificates*.



How certification works

* Certificates that are signed by a *certificate authority (CA)* are signed by a third-party CA or a trusted CA.
* *Self-signed* certificates are created, issued, and signed by the entities whose identities the certificates are meant to verify.

### Firewalls

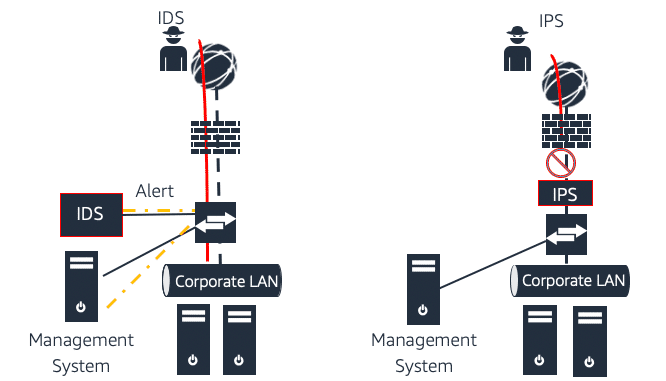


Firewalls monitor incoming and outgoing network traffic, and they filter data packets.

A *firewall* is a network security device that helps to protect software, hardware, or web applications against common web exploits. These exploits might affect availability, compromise security, or consume excessive resources. Firewalls monitor incoming and outgoing network traffic. They allow data packets, or blocks data packets. Firewalls control how traffic reaches your applications by enabling you to create security rules that block common attack patterns, such as SQL injection or cross-site scripting. You can also create rules that filter out specific traffic patterns that you define.

### Intrusion detection systems and intrusion prevention systems

An *intrusion detection system (IDS)* is a combination of hardware and software that can detect network attacks and alert network information security professionals.



How an IDS and an IPS work

An IDS can discover an attack in a couple of ways:

* An *anomaly-based IDS* examines baselines and deviations from baseline behavior.
* A *signature-based IDS* monitors for known patterns of attack.

In addition, an IDS usually is either a *network-based intrusion detection system (NIDS)* or a *host-based intrusion detection system (HIDS)*. A NIDS monitors for *attacks on the network*. Often, all the data that travels through the network also travels through the IDS for inspection and analysis. A NIDS is installed on a server and monitors *logs and critical files on the server*, monitoring for signs of a network attack.

An *intrusion prevention system (IPS)* is a network security tool that *identifies and prevents* threats. An IPS monitors networks for malicious incidents, and acts to prevent these malicious threats from affecting the system. IPSs are often behind the firewall, and they provide an extra layer of protection.

### Security training and education

Security training and education is a continuous effort to acquire knowledge, skills, and competencies that are related to security. Knowledge of security starts with awareness and develops with further training and education.

## Key takeaways

* The goal of threat mitigation is to *combat threats* to an organization's data. It should be a proactive and preventive effort.
* Anti-malware, encryption, certificate authentication, firewalls, IDSs, IPSs, and training are examples of effective *threat-mitigation methods*.

# **Page 9 .A connected world: Exploring emerging technologies**

### **At the core of the lesson, you will learn how to:**

* List some new wireless technologies and their characteristics
* Define bring your own device (BYOD) and mobile device management (MDM)
* Identify the primary goal of the Internet of Things (IoT)
* Identify strategies and techniques for keeping current with emerging technologies

### **Wireless technologies**

Wireless technologies are devices that communicate through the air.

* Wireless technology plays a key role in today’s communications.
* Emerging technologies—including robots, drones, self-driving vehicles, and new medical devices—all have a need for wireless technologies.

### **Examples of emerging wireless technologies**

* **Wireless Gigabit (WiGig)** was first announced in 2009 by the Wireless Gigabit Alliance. WiGig was designed for fast speeds over short distances. You can take full advantage of a WiGig network if you have two or more devices on a WiGig network in the same room, with no obstacles between them. An example use case for WiGig is streaming content wirelessly from a device to a high-resolution TV.
* **HaLow (pronounced *halo*)** was first announced in 2016. It is a low-power, long-range version of Wi-Fi. HaLow was designed and developed for small data payloads and low-power devices. It can extend Wi-Fi to 900 MHz. It works well for sensors and wearable technologies, and it has twice the range of current Wi-Fi.
* **Bluetooth Low Energy (BLE)** optimizes energy consumption. You might also hear Bluetooth Low Energy referred to as *BLE*, *Bluetooth LE*, or *Bluetooth Smart*. BLE technology is primarily used in mobile applications, and it is suitable for IoT. It was initially developed for the periodic transfer of small chunks of data over short ranges. BLE is used in solutions that span a range of domains, including healthcare, fitness, beacons, security, and home entertainment.
* **5G cellular systems** are a new technology that has been deployed throughout 2019 and 2020. The complete rollout of this technology will take a couple of years. It will eventually provide download speeds up to 10 Gbps.

### **What is the Internet of Things (IoT)?**

The **Internet of Things (IoT)** refers to physical devices, or *things*, that are connected to the internet so they can share and collect data.

The primary goal of IoT is to enable devices to self-report in real time, which can improve efficiency. It can also surface important information more quickly than a system that depends on human intervention.

Many devices are in homes, factories, oil wells, hospitals, cars, and other places that need to be connected. Data from these devices can be collected, stored, and analyzed to design better solutions or products. IoT devices include consumer applications (like thermostats) and commercial applications (such as industrial sensors, smart watches, and smart toasters). They also include devices in the Internet of Medical Things (IoMT), such as health monitors.

## [**YOUTUBE**](https://www.youtube.com/)

## **[IoT is Everywhere: From Home to Work](https://cdn.embedly.com/widgets/media.html?src=https%3A%2F%2Fwww.youtube.com%2Fembed%2FbUnOo10z42k%3Ffeature%3Doembed&display_name=YouTube&url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DbUnOo10z42k&image=https%3A%2F%2Fi.ytimg.com%2Fvi%2FbUnOo10z42k%2Fhqdefault.jpg&key=40cb30655a7f4a46adaaf18efb05db21&type=text%2Fhtml&schema=youtube)**

No matter the industry, IoT delivers new opportunities in data, analytics and innovation. Learn how AWS IoT - https://amzn.to/2RQWVoR provides broad and deep...

[**VIEW ON YOUTUBE**](https://cdn.embedly.com/widgets/media.html?src=https%3A%2F%2Fwww.youtube.com%2Fembed%2FbUnOo10z42k%3Ffeature%3Doembed&display_name=YouTube&url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DbUnOo10z42k&image=https%3A%2F%2Fi.ytimg.com%2Fvi%2FbUnOo10z42k%2Fhqdefault.jpg&key=40cb30655a7f4a46adaaf18efb05db21&type=text%2Fhtml&schema=youtube)

### **Enterprise mobility**

Enterprise mobility is a growing trend for businesses. This approach supports remote-working options, which use your personal laptops and mobile devices for business. Remote workers can connect and access data through cloud technology.

Two specific solutions enable enterprise mobility:

* *Bring your own device (BYOD)* is the use of a personal device, like a mobile phone or tablet, on a public or private network. This solution includes the use of a personal device on a corporate network.
* *Mobile device management (MDM)* is a term that describes the management of mobile devices, though it also applies to laptop and desktop computers. Organizations use MDM solutions so that they can provide devices with settings, software, and access to data in a secure way that complies with their needs.

### **Staying current with emerging technologies**

You have many options for keeping up-to-date with the latest technology trends that will impact the future of networking. Many websites, authors, professional networking groups, and other resources are devoted to covering the latest developments in networking. The following list of resources are specific to AWS, which you might find useful.

**The Official AWS Podcast**

The Official AWS Podcast is a podcast for developers and IT professionals who want to learn the latest news and trends in storage, security, infrastructure, serverless, and more.

[**AWS PODCAST**](https://aws.amazon.com/podcasts/aws-podcast/)

**The Jeff Barr AWS Blog**

Jeff Barr is the Chief Evangelist for AWS. He started this blog in 2004, and has been writing posts since then.

[**JEFF BARR@**](https://aws.amazon.com/blogs/aws/author/jbarr/)

**The AWS Tech Chat**

Learn about the latest summary of AWS news, announcements, services, and feature updates. The AWS Tech Chat is brought to you by AWS subject matter experts from the Asia-Pacific region.

[**AWS TECH CHAT**](https://aws.amazon.com/podcasts/aws-techchat/)

## **Key takeaways**

* Emerging technologies in the networking domain include:  
  + Wireless technology
  + BYOD
  + IoT
* AWS provides many resources that enable you to stay current with emerging technologies