Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

**Module**

**7**

**Network Architecture**

* [Module Introduction](javascript://)
* **7-1**[Physical Architecture](javascript://)
  + **7-1a**[Managed Switches](javascript://)
  + **7-1b**[Switch Path Management](javascript://)
  + **7-1c**[Switch Port Security](javascript://)
  + **7-1d**[Hierarchical Design](javascript://)
  + **7-1e**[Software-Defined Networking (SDN)](javascript://)
  + **7-1f**[Storage Area Network (SAN)](javascript://)
* **7-2**[Virtual Architecture](javascript://)
  + **7-2a**[Network Connection Types](javascript://)
  + **7-2b**[Pros and Cons of Virtualization](javascript://)
  + **7-2c**[NFV (Network Functions Virtualization)](javascript://)
* **7-3**[Cloud Architecture](javascript://)
  + **7-3a**[Cloud Service Models](javascript://)
  + **7-3b**[Cloud Deployment Models](javascript://)
  + **7-3c**[Orchestration and Automation](javascript://)
  + **7-3d**[Connectivity and Security](javascript://)
* **7-4**[Network Availability](javascript://)
  + **7-4a**[Fault Tolerance](javascript://)
* **7-5**[Module Review](javascript://)
  + **7-5a**[Module Summary](javascript://)
  + **7-5b**[Key Terms](javascript://)
  + **7-5c**[Review Questions](javascript://)
  + **7-5d**[Hands-On Projects](javascript://)
  + **7-5e**[Capstone Projects](javascript://)

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# Module Introduction

### Objectives

After reading this module and completing the exercises, you should be able to:

* **1**Explain types of abstraction in the design of physical network architecture
* **2**Describe and explain virtualization technologies on a network
* **3**Summarize cloud characteristics, models, and connectivity options
* **4**Identify methods to increase network availability

**On the Job**

A small-business client of our IT consulting firm approached us about upgrading their three servers. They wanted to upgrade to more advanced technology and were also concerned about reducing their energy consumption. The three servers performed three very different functions. One was dedicated to handling the domain, user logins, file and print sharing, and related services. Another hosted a company-wide sales application, which included a large database. The last was dedicated to backing up the database and other important data.

Upon investigating the client’s current systems, we identified a few interesting characteristics. First, average utilization levels for all three servers were well below 50 percent. Moreover, the peak utilization times for the three servers were staggered. The domain server was busiest first thing in the morning, when users first logged into the network, with another small spike after lunchtime. The application server had consistent utilization throughout the day, with the largest peak time at the end of the workday as sales staff input sales for the day. The backup server peaked during off-work hours, as it copied data from the database and other storage while they were not in use.

These findings meant that we could use a hypervisor to run all three servers as virtual machines on a single host server device. This allowed each virtual server to utilize the host’s hardware resources when they were needed the most and, during times of lower utilization, free up those resources for other virtual servers. Not only did this result in a more acceptable level of average server utilization, but it actually decreased power utilization overall, allowing our client to enjoy a lower electric bill.

**Brent M. Ferns, Sr., M.B.A., M.S.A.**

**Associate Dean of Business and Computer Science**

**Palm Beach State College**

Through the past several modules, you’ve learned about the hardware found on a network, the cabling and radio waves that connect this hardware, the way devices on a network are addressed, and the core protocols that make networks work. These concepts cover the basics of how a simple network functions. However, businesses—especially very large businesses—need networking technology that constantly improves to ensure ever faster performance and more reliable service and security.

Consider the progression of telephone technology. In the beginning, the telephone existed to carry a voice conversation between two people. It required manual switching ([Figure 7-1](javascript://)) and the quality of the sound wasn’t all that great. Telephones eventually got better but were still wired to a single location. Then telephones became mobile. Next, they gained the ability to do other things, like texting, taking pictures and video, browsing the web, and playing games. The idea of “phoning someone” became less about having a voice conversation and more about being able to connect to others in a variety of ways, through text, photos, video calls, email, visiting websites, and using social media apps. You might say that today’s smartphones abstracted the idea of “connection” from early telephone conversations and thereby expanded the possibilities of how people connect today.

**Figure 7-1**

A mid-twentieth century telephone switchboard



Enlarge Image

Abstraction is a critical concept in this module and in modern networking. The process of networking isn’t just about bits running along a cable or through the air. Networking also involves decision-making devices that are configured to:

* Prioritize some traffic over other traffic
* Direct traffic to different locations
* Filter traffic for security
* Correct problems with messages that experience errors

This decision making doesn’t always happen directly on the hardware routers and switches—it can be abstracted to another device or even to software running in a VM (virtual machine). You’ve already learned that a network’s physical topology might be configured very differently than its logical topology. Network abstraction—that is, the separation of decision making from hardware—takes this idea further. Examples of network abstraction include the following:

* Networking devices might be controlled remotely by another device.
* Networking devices might exist as virtual devices like a VM.
* Networking devices might only exist as logical constructs that govern traffic in a virtual environment.

You’ll begin this module with a discussion of how to manage physical switches and some of the configuration and organizational considerations for switches in a network design. This overall network design—the devices involved, how they’re configured, the services implemented to support the network, and the way devices are connected to the network—is called a network’s [**architecture**](javascript://). You’ll first read about physical network architecture, such as switch hierarchy and special storage networks. You’ll then explore how virtual devices expand network design parameters while strengthening network availability and security. And then you’ll see how the concept of virtualization in the cloud has further broadened the possibilities for network architecture beyond the walls of the on-site data center. Get ready to think “outside the box” as you learn how network architecture has expanded beyond the confines of physical space.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# 7-1Physical Architecture

### Certification

* 1.2

Explain the characteristics of network topologies and network types.

* 1.7

Explain basic corporate and datacenter network architecture.

* 2.1

Compare and contrast various devices, their features, and their appropriate placement on the network.

* 2.3

Given a scenario, configure and deploy common Ethernet switching features.

* 3.3

Explain high availability and disaster recovery concepts and summarize which is the best solution.

* 4.1

Explain common security concepts.

* 4.3

Given a scenario, apply network hardening techniques.

* 5.5

Given a scenario, troubleshoot general networking issues.

Average reading time: 45 minutes



At this point, you’ve gotten a little experience working with switches in Packet Tracer and perhaps in your school’s lab. You’ve connected devices to switches, and you know how to ensure devices on the same switch can communicate with each other. The responsibility of switch management extends well beyond the tasks associated with connecting devices. As networks rely increasingly on switch-based technologies, managed switches and layer 3 switches (which you’ll learn about shortly) play a much more critical role in an enterprise environment. At the same time, switch security becomes a more important—and more complex—factor in protecting a network’s resources.

Let’s first look at some of the variety of switch types available. Then you’ll learn about some of the configuration changes you can make to these switches to better manage paths between them.

**Note 7-1**

The Cisco switches you’ve worked with in Packet Tracer run a basic operating system called IOS (Internetwork Operating System). Recall from [Capstone Project 3-2](javascript://) in [Module 3](javascript://), you used a command from Cisco’s IOS for switches, show mac address-table. However, there are many other brands of networking devices available, including Huawei, Arista, and Juniper. In this module, as you study commands used to configure switches, you’ll start learning some of the variations of commands used on other vendors’ devices.

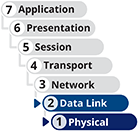
Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 7-1aManaged Switches



An [**unmanaged switch**](javascript://) provides plug-and-play simplicity with minimal configuration options and has no IP address assigned to it. Unmanaged switches are not very expensive, but their capabilities are limited. [**Managed switches**](javascript://), on the other hand, can be configured via a command-line interface or a web-based management GUI, and sometimes can be configured in groups. Sometimes, they are also assigned IP addresses for the purpose of continued management.

Recall that switches are layer 2 devices. However, higher-layer switches also exist:

* [**Layer 3 switch**](javascript://)—Capable of interpreting layer 3 data and can work much like a router, supporting the same routing protocols and making routing decisions. Layer 3 switches are less expensive than routers and are designed to work on large LANs, providing faster layer 3 traffic management within the confines of a known network architecture. The primary difference is the way the hardware is built, but, in fact, it’s often difficult to distinguish between a layer 3 switch and a router. In some cases, the difference comes down to what the manufacturer has decided to call the device to improve sales.
* [**Layer 4 switch**](javascript://)—Capable of interpreting layer 4 data. They operate anywhere from layer 4 to layer 7, and they’re also known as content switches or application switches. Among other things, the ability to interpret higher-layer data enables switches to perform advanced filtering, keep statistics, and provide security functions.

The features of layer 3 and layer 4 switches vary widely depending on the manufacturer and price point, and they can cost significantly more than layer 2 switches. This variability is exacerbated by the fact that key players in the networking trade have not agreed on standards for these switches. They are typically used as part of a network’s backbone and are not appropriate on a single LAN. What they all have in common is that they’re optimized for fast layer 2 data handling.

In this section, you’ll first learn how paths between switches are managed, and you’ll learn about some innovative ways to organize switches for optimized traffic management. Then you’ll examine switch security concerns and some ways switches are used to optimize server performance.

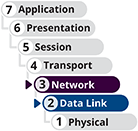
Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

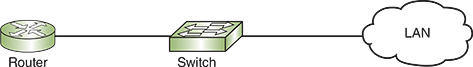
## 7-1bSwitch Path Management



[Figure 7-2](javascript://) shows where a typical layer 2 switch is positioned on a basic network. This switch manages all network traffic on the LAN unless a host on the network wants to communicate with a host on another network, and then that traffic goes through the router.

**Figure 7-2**

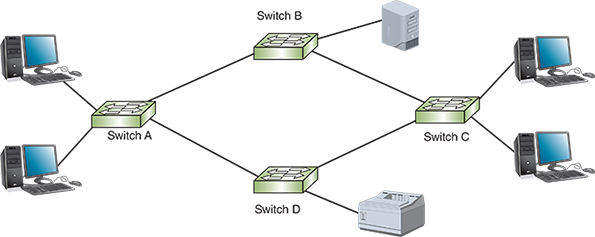
A switch connecting end devices to a router



Suppose you design a larger network with several interconnected switches. To make the network more fault tolerant, you install multiple, or redundant, switches at critical junctures. [**Redundancy**](javascript://) allows data the option of traveling through more than one switch toward its destination and makes your network less vulnerable to hardware malfunctions. For example, if one switch suffers a power supply failure, traffic can reroute through a second switch. Your network might look something like the one pictured in [Figure 7-3](javascript://), where several switches work together to connect hosts in several areas of your building. (In reality, of course, many more hosts would connect to each switch.)

**Figure 7-3**

Network of switches



Enlarge Image

A potential problem with the network shown in [Figure 7-3](javascript://) has to do with traffic loops. What if a server attached to switch B issues a broadcast frame? Switch B then reissues the broadcast to all its ports except the port to which the server is attached. In that case, switch B will issue the broadcast frame to switches A and C, which will both reissue the broadcast frame to switch D, which then reissues the received broadcast frame from each direction back to both switches A and C, and so on around the loop. If not limited in some way, these redundant broadcast transmissions will flood the network (called a [**broadcast storm**](javascript://)), and the high traffic volume will severely impair network performance or possibly disable the network entirely.

Many Huawei, Arista, Juniper, Cisco, and similar devices offer a type of flood guard known as storm control that protects against flooding attacks from broadcast and multicast traffic. Storm control monitors network traffic at one-second intervals to determine if the traffic levels are within acceptable thresholds. Any time traffic exceeds the predefined threshold, all traffic is dropped for the remainder of the time interval. This feature is managed on three of these vendors’ devices using the storm-control command (without the hyphen on Huawei devices: storm control). However, eliminating—or otherwise controlling—switching loops can greatly reduce the potential for a broadcast storm.

To eliminate the possibility of this and other types of traffic loops, [**STP (Spanning Tree Protocol)**](javascript://) was developed by Radia Perlman at Digital Equipment Corporation in 1985 and then adopted by the IEEE in 1990.

The first iteration of STP, defined in IEEE standard 802.1D, functions at the data link layer. It prevents traffic loops, also called switching loops, by calculating paths that avoid potential loops and by artificially blocking the links that would complete a loop. In addition, STP can adapt to changes in the network. For instance, if a switch is removed, STP will recalculate the best loop-free data paths between the remaining switches.

**Note 7-2**

In the following explanation, you can substitute switch wherever the word bridge is used. As you have learned, a switch is really just a glorified bridge. STP terminology refers to a layer 2 device as a bridge because STP was designed and created before switches existed.

So how does STP select and enforce switching paths on a network? Consider the following process:

1. Step 1

STP selects a [**root bridge**](javascript://) that will provide the basis for all subsequent path calculations. Only one root bridge exists on a network. From this root bridge, a series of logical branches, or data paths, emanate like branches on a tree. STP selects the root bridge based on its BID (Bridge ID). The BID is a combination of a 2-byte priority field (which can be set by a network admin) and the bridge’s MAC address. To begin with, all bridges on the network share the same priority number, and so the bridge with the lowest MAC address becomes the root bridge by default.

1. Step 2

STP examines the possible paths between all other bridges and the root bridge. It chooses the most efficient of these paths, called the [**least cost path**](javascript://), for each of the bridges. To enforce this path, STP stipulates that each bridge can have only one [**root port**](javascript://), which is the bridge’s port that is closest to and forwards frames to the root bridge. closest to the root bridge, can forward frames toward the root bridge.

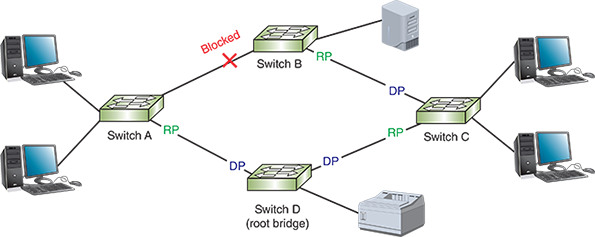
1. Step 3

STP disables links that are not part of a shortest path. To do this, it enables only the lowest-cost port on each link between two bridges to transmit network traffic. This port is called the designated port. All ports can, however, continue to receive STP information in case updates are later needed.

[Figure 7-4](javascript://) illustrates a switched network with certain paths selected and one blocked by STP. In this drawing, root ports pointing upstream toward the root bridge are labeled RP. Designated ports pointing downstream from the root bridge are labeled DP. For example, traffic from the root bridge (switch D) going to switch B would be forwarded through switch C. Even though switch B is also connected to switch A, STP has limited the logical pathway so messages destined for switch B will only go through switch C. Now suppose switch C were to fail. STP would automatically adapt by choosing a different logical pathway for frames destined for switch B. This way, redundancy is maintained while avoiding switching loops.

**Figure 7-4**

DP indicates downstream designated ports, and RP indicates upstream root ports



Enlarge Image

STP information is transmitted between switches via [**BPDUs (Bridge Protocol Data Units)**](javascript://). To protect the integrity of STP paths and the information transmitted by these BPDUs, some security precautions that must be configured on STP-enabled interfaces include the following:

* **BPDU guard**—Blocks BPDUs on any port serving network hosts, such as workstations and servers, and thereby ensures these devices aren’t considered as possible paths. BPDU guards also enhance security by preventing a rogue switch or computer connected to one of these ports from hijacking the network’s STP paths.
* **BPDU filter**—Disables STP on specific ports. For example, you might use a BPDU filter on the demarc, where the ISP’s service connects with a business’s network, to prevent the ISP’s WAN topology from mixing with the corporate network’s topology for the purpose of plotting STP paths.
* **Root guard**—Prevents switches beyond the configured port from becoming the root bridge. For example, an ISP might configure a root guard on an interface facing a customer’s network to ensure that none of the customer’s switches becomes the ISP’s root bridge.

Network developers have repeatedly modified STP to improve and customize its functioning. The original STP is considered too slow for today’s networks. For instance, it could take up to two minutes to detect and account for a link failure. With that kind of lag time, older versions of STP would bog down network transmissions, especially where high-volume, speed-dependent traffic, like telephone or video signals, is involved. Newer technologies to improve on or replace STP include the following:

* RSTP (Rapid Spanning Tree Protocol), defined in IEEE’s 802.1w standard, and MSTP (Multiple Spanning Tree Protocol), originally defined by the 802.1s standard, can detect and correct for link failures in milliseconds.
* TRILL (Transparent Interconnection of Lots of Links) is a multipath, link-state protocol developed by the IETF.
* SPB (Shortest Path Bridging) is a descendent of STP and is defined in IEEE’s 802.1aq standard. SPB differs from earlier iterations of STP in that it keeps all potential paths active while managing the flow of data across those paths to prevent loops. By utilizing all network paths, SPB greatly improves network performance.
* Some switch manufacturers, such as Cisco and Extreme Networks, have designed proprietary versions of STP that are optimized to work most efficiently on their equipment. One popular example is Rapid PVST+ (Per VLAN Spanning Tree Plus) by Cisco. A VLAN (virtual LAN) is similar in concept to subnets, but it functions at OSI layer 2 instead of OSI layer 3. You’ll learn more about VLANs later in this course.

Protocols designed to replace STP, such as SPB, operate at layer 3 instead of or in addition to layer 2, making them more compatible with various types of technologies like the connection protocols used on storage networks, which you’ll learn more about shortly.

When installing switches on your network, you don’t need to enable or configure STP (or the more current version that came with your switch). It will come with the switch’s operating software and should function smoothly by default and without intervention. However, if you want to designate preferred paths between bridges or choose a special root bridge, for example, STP and its relatives allow you to alter default prioritizations.

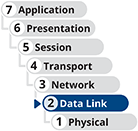
Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 7-1cSwitch Port Security



Switches are designed to offer lots of ports through which devices can access a network by sending and receiving messages. Ideally, you want approved devices connected to a switch’s ports, and no unapproved devices transmitting on those ports. This might seem like a simple goal until you consider how easy it is to unplug an Ethernet cable from an approved computer and plug it into an unapproved laptop. Depending on how the switch’s port, or interface, is configured, this simple vulnerability could give an attacker easy and trusted access to your entire network!

One type of attack (intentional or not) that could be conducted with this kind of unauthorized network access is the broadcast storm you read about as you studied Spanning Tree. For example, if someone connects a hub to two unsecured switch ports, the hub creates a loop that generates a broadcast storm. Another example is connecting both ports on a VoIP phone to the network, or possibly connecting a computer through both a wired and wireless network connection. Controlling who can connect what device to a switch’s port can help prevent this type of attack.

As a first layer of defense, unused physical and virtual ports on switches and other network devices should be disabled until needed. You can do this on Cisco, Huawei, and Arista routers and switches with the shutdown command. To enable them again, use the no shutdown command on Cisco or Arista devices, and use undo shutdown on Huawei devices. On a Juniper device, the corresponding commands are disable and enable, respectively. This will prevent an attacker from plugging into an unused port to conduct their attack. However, what about ports that are being used by legitimate devices?

Another Cisco command (which is also used on Arista devices) to secure switch access ports is switchport port-security (or just port-security on Huawei switches). This is essentially a MAC filtering function that also protects against MAC flooding, which makes it a type of flood guard. Acceptable MAC addresses are stored in a MAC address table. This table can be configured manually or dynamically. By default, the table allows only one MAC address to be active on the port; however, a network administrator can allow more MAC addresses per port. (On a Juniper switch, the mac-limit command restricts the number of MAC addresses allowed in the MAC address table. Approved MAC addresses are configured with the allowed-mac command.) Once the MAC address table is full, a security violation occurs if another device attempts to connect to the port. By default, the switch will shut down the port, or it can be configured to restrict data from the rogue device. Either way, the switch generates a notification to the network administrator. At the end of this module, you’ll have the opportunity to practice securing switch ports on a switch in Packet Tracer, and then you’ll see what happens when a hacker attempts to take over that port.

This type of switch port security is only one layer of defense—more of a deterrent, really. As you already know, MAC spoofing is not difficult to accomplish. The biggest challenge in this attack is learning the port’s approved MAC address so that an attacker can spoof that address. As you’ll learn through the rest of this course and throughout your career in IT, security should always be implemented in layers, which is a strategy called [**defense in depth**](javascript://).

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 7-1dHierarchical Design



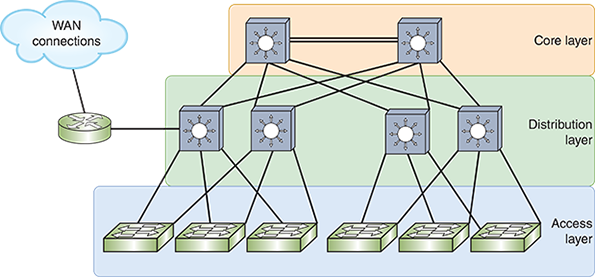
The network shown earlier in [Figure 7-3](javascript://) shows only four switches operating at the data link layer. While this example is larger than the typical home network, it hardly qualifies as a corporate network. Enterprise-grade networks might contain dozens or hundreds of switches (many of which also function at higher OSI layers) in addition to many routers, servers, and firewalls to increase performance and to better protect the network from problems if one or more devices fail. In the interest of increased redundancy, often you’ll find teams or clusters of these devices working together to support higher volumes of traffic, such as when a cluster of web servers all work together to support a single website. A [**load balancer**](javascript://) helps to evenly distribute traffic to each device in a cluster so every device carries a portion of the load.

With all these devices on a network, network engineers must take a more organized approach to designing the network to ensure that every device can function at its best without being swamped with too many kinds of tasks. To this end, Cisco and other manufacturers have developed a hierarchical design for switches on a network. In general, you can think of this hierarchy as having three tiers—that is, a [**three-tiered architecture**](javascript://), as illustrated in [Figure 7-5](javascript://) and described next:

* The [**access layer**](javascript://), or [**edge layer**](javascript://), consists of workgroup switches connected directly to hosts such as servers, printers, and workstations. Ideally, hosts connect only to access switches and never to switches at the other layers, although this is not always the case in the real world. Access switches typically organize traffic according to OSI layer 2 technologies.
* The [**distribution layer**](javascript://), or [**aggregation layer**](javascript://), is a highly redundant mesh of connections between multilayer switches or routers. It provides routing within the corporate network as well as traffic filtering and the network’s connection to one or more WANs, such as a WAN connection to your [**branch offices**](javascript://) at other locations or to the Internet. Other network blocks might also connect to this layer, such as a server network or a storage area network. Ideally, distribution switches manage traffic according to OSI layer 3 technologies; however, this isn’t always possible due to limitations of certain applications needed on the network.
* The [**core layer**](javascript://) consists of highly efficient multilayer switches or routers that support the network’s backbone traffic. Typically, no filtering or routing is performed at this layer so as to minimize the processing required of these switches. Surprisingly, this layer is considered the simplest layer and doesn’t need switches with lots of ports. These switches simply need to pass traffic in and out of a few backbone ports as quickly as possible. These switches nearly always function primarily at OSI layer 3.

**Figure 7-5**

In a three-tiered architecture, switches at each layer are optimized to perform different functions



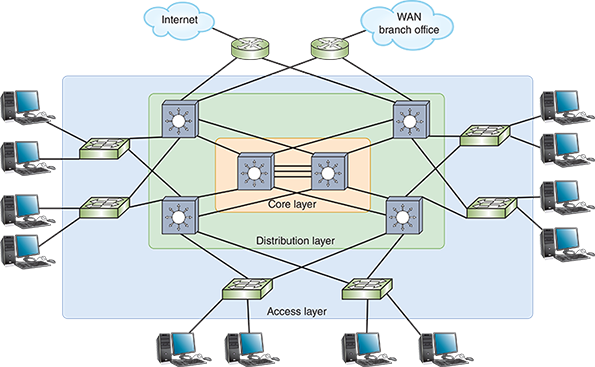
Enlarge Image

In [Figure 7-5](javascript://), notice the multiple switches at each layer, and the multiple connections between switches. This mesh topology provides redundancy to protect the network from device or connection failure. For example, if a distribution switch’s connection with one of the core switches fails, the distribution switch has a redundant connection with the other core switch.

The diagram in [Figure 7-5](javascript://) is simplified so you can see the relationship between the layers of this three-tiered architecture and the necessity for redundant connections. [Figure 7-6](javascript://) lays out the network devices a little more realistically. This figure shows you why the core layer is considered the center, or backbone, of the network. You can also see where hosts connect to the access layer, and other network blocks connect to the distribution layer.

**Figure 7-6**

The distribution layer serves as the interface between the core and access layers and also connects other network blocks to the network

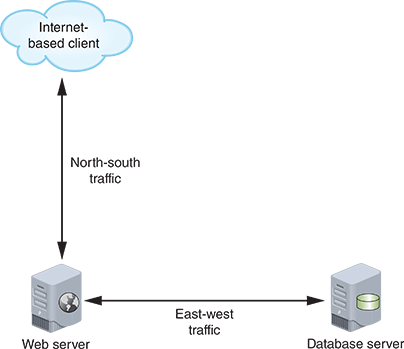


Enlarge Image

One of the advantages of this three-tier design is that traffic destined for nearby nodes can be handled differently and more efficiently than traffic that must traverse longer paths to its destination. The flow of traffic between peers within a network segment is called [**east-west traffic**](javascript://), and messages that must leave the local segment to reach their destinations are called [**north-south traffic**](javascript://). For example, as illustrated in [Figure 7-7](javascript://), traffic from web clients on the Internet requesting information from a web server in the data center is north-south traffic. Traffic from the web server requesting information from a database server within the same data center is east-west traffic. This east-west traffic between servers in the same data center never leaves the access or distribution layers (depending on the network design). However, north-south traffic between the web server and the Internet flows through more switches throughout all layers in the design hierarchy.

**Figure 7-7**

Messages from outside the network create a north-south traffic flow, while messages between peers within the network generate an east-west traffic flow

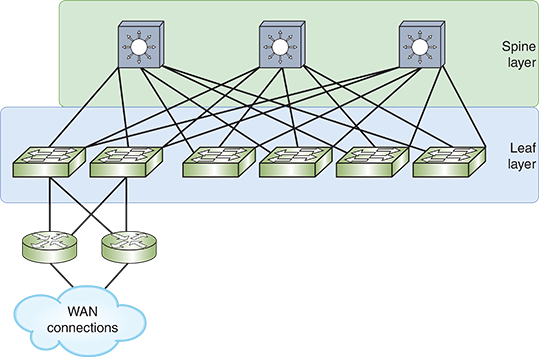


Years ago, the traffic type that needed the most optimization was north-south traffic. However, you’ll soon learn about newer data center technologies, including virtualization, SDN (software-defined networking), cloud computing, and new server technology. As these technologies became more popular, east-west traffic began experiencing significantly worse latency. A new hierarchical design was needed to better optimize this east-west traffic.

Instead of using three layers, newer networks collapse the core and distribution layers into one layer called the spine. Spine switches on the backbone connect in a mesh topology with all leaf switches (but not with each other), and leaf switches connect with servers and other host devices. This design is called a [**spine-and-leaf architecture**](javascript://) (also spine-leaf or leaf-spine), as shown in [Figure 7-8](javascript://). Similar to the three-tiered architecture, spine switches organize traffic and network segments using OSI layer 3 technologies while leaf switches manage traffic by either layer 2 or layer 3 principles.

**Figure 7-8**

Two architecture layers provide more efficient access between any two network resources

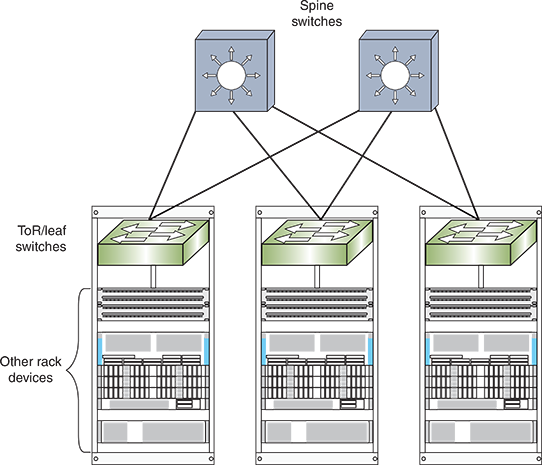


Leaf switches often reside in the same rack as the servers they support. Two rack architectures include the following:

* [**ToR (top of rack) switching**](javascript://)—While ToR switches are not necessarily positioned at the top of the rack, the top location in the rack often provides the best accessibility and cable management. The ToR switch connects all other devices in the rack to the rest of the network, as shown in [Figure 7-9](javascript://).
* [**EoR (end of row) switching**](javascript://)—Where ToR is a vertically oriented arrangement, EoR is more horizontal in orientation. Instead of placing a switch at the top of each rack, EoR switching places several leaf switches in a rack at the end of each row of racks or in the middle of each row—called MoR (middle of row) switching. This approach requires fewer switches (and, therefore, fewer hops for much of the network’s traffic) and less rack space, but more cabling and more work at cable management.

**Figure 7-9**

Leaf switches often server as the ToR switch in a rack



Each of these switch architectures offers benefits such as a reduced need for cabling and lower latency. Keeping the bulk of a network’s cabling within the rack for very short connections means the network requires less cabling overall and provides increased flexibility in the network design. Very short cable runs make very high-speed cables—supporting 10 GbE (Gigabit Ethernet) or even 40 GbE or 100 GbE—an affordable option to connect leaf switches to spine switches.

The spine-and-leaf design also provides many benefits over the older, three-tiered architecture:

* **Improved redundancy**—Every leaf switch is connected to multiple (or perhaps all) spine switches in a full mesh topology, which provides redundant connections in case one link fails.
* **Decreased latency**—Because each leaf switch is connected to every spine switch, messages must traverse fewer hops to reach their destination.
* **Increased performance**—A spine-and-leaf network can replace the older, error-laden STP with newer path management technologies such as TRILL and SPB. These technologies take advantage of the redundant links to increase performance and redundancy without creating problematic switching loops.
* **Improved scalability**—The number of available and usable paths for messages across a network improves scalability. This means a network can support larger numbers of host devices without overwhelming network pathways.
* **Increased security**—Traffic at all layers can be more easily inspected and monitored. With the traditional three-layer hierarchy, typically north-south traffic was filtered through a firewall as it entered the network. However, after that point, traffic was considered trustworthy and was rarely inspected. A spine-and-leaf architecture allows for security inspections of all traffic, including east-west traffic flows. This permutation of security priorities will come up again when you study security in network design later in this course.
* **Reduced expense**—Surprisingly, the hardware needed to build a spine-and-leaf architecture is typically less expensive than the switches needed for a three-tiered architecture.

The improvements offered by spine-and-leaf architecture were partly needed due to modern technologies such as SDN (software-defined networking). Let’s look at what SDN is and why it’s a favored approach to networking today.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 7-1eSoftware-Defined Networking (SDN)

As network infrastructures became more complex, configuring all these networking devices to create and manage the network infrastructure presented a growing challenge. [**SDN (software-defined networking)**](javascript://) is a centralized approach to networking that removes most of the decision-making power from network devices and instead handles that responsibility at a software level. Let’s break that definition down with an analogy.

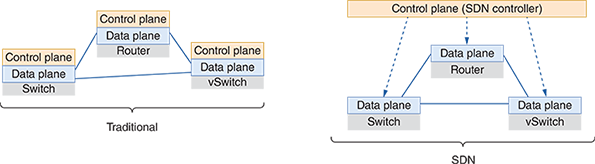
Suppose you’re planning a neighborhood party and need everyone to bring a dish to share. You could take the so-called “pot-luck” approach where everyone brings whatever food they want based on what they like best. In that case, you might end up with a table full of desserts and no salad or veggies (not a terrible thing, but in the end, not well balanced). Instead, suppose you group your neighbors by floor in the building or by street in the neighborhood. You ask one group to bring a main course, another group to bring a side dish or casseroles, and a third group to bring desserts. This coordinated effort results in a more balanced meal that everyone (regardless of diet restrictions) can enjoy.

SDN relies on a similar concept—centralize control of network devices so networking rules (such as switching paths) are applied in a more coordinated manner. You give one device, called the SDN controller, a bird’s eye view of the entire network so it can make more informed decisions on how to direct messages through the network. This [**SDN controller**](javascript://) integrates configuration and management control of all network devices, both physical and virtual, into one cohesive system that is overseen by the network administrator through a single dashboard. Instead of reconfiguring each network device individually, the SDN controller can be used to reconfigure groups of network devices all at one time. It can even make configuration changes automatically in response to changing network conditions.

SDN relies on a form of abstraction called disaggregation, which basically means separating into pieces all the functions of a system so each piece can be handled by separate devices. This is kind of like when each person in an assembly line specializes in one type of task. For example, instead of each person performing all the steps to assemble a car, each person specializes in doing a small part of the work. In the end, this division of labor produces higher quality products at a faster rate. SDN creates the same kind of benefit for networks. Essentially, SDN abstracts the functions of network devices into different layers, or planes, and then relocates those planes in ways that make network management more effective (see [Figure 7-10](javascript://)).

**Figure 7-10**

Distributed control planes in a traditional network versus a centralized control plane in an SDN network



Enlarge Image

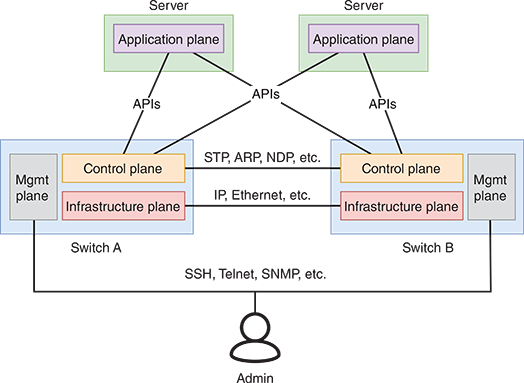
Consider the following explanation of each of these planes.

* [**Infrastructure plane**](javascript://) (also called [**data plane**](javascript://))—This plane is made up of the physical or virtual devices (switches, routers, firewalls, and load balancers) that receive and send network messages on their way to their destinations. This is the plane where bits cross interfaces. This is also the plane where messages are decapsulated to examine layers of headers, MAC addresses are matched to switch ports, addresses are changed for NAT processing, and messages are re-encapsulated for the jump to the next device. The primary function of the infrastructure plane is to forward data on to its destination. Think of the infrastructure plane as the network’s brawn (physical strength).
* [**Control plane**](javascript://)—Think of the control plane as the network’s brain (intelligence)—it handles the decision-making processes. Traditionally, the infrastructure plane and the control plane co-exist on the same device. With SDN, the control plane is abstracted to an SDN controller, which remotely manages networking devices. For example, consider the processing power that goes into building a switch’s MAC address table, or the time and effort STP requires to optimize switching paths. All this work can be done by the SDN controller, which then downloads to network devices the required MAC address tables and other policies. This way, the switch doesn’t have to think about what the network looks like or what devices are connected where. The switch simply compares each incoming message to its list of rules from the controller, and it sends the message on its way. If a message doesn’t match one of the switch’s preconfigured rules, the switch can send the message to the SDN controller for further analysis. This level of insight allows the SDN controller to create more nuanced rules specific to its network’s needs. The SDN controller communicates with the infrastructure plane using APIs defined by an SDN protocol such as the popular and open source OpenFlow. Recall that APIs (application programming interfaces) are access points into available processes to generate a response of some kind. Several vendors offer SDN controller software, including VMware, Cisco, HP, IBM, and Juniper. Open source SDN controllers include ODL (OpenDayLight), ONOS (Open Network Operating System), and OpenKilda.
* [**Application plane**](javascript://)—The SDN controller also communicates with network applications using APIs. Applications the SDN might need to communicate with include web browsers, VoIP software, network services like DNS, and apps specific to SDN. For example, you might install an analytics application that monitors network traffic for signs of a security breach. The application plane corresponds to OSI’s application layer.
* [**Management plane**](javascript://)—While not a typical layer for network communication, this plane could be considered a part of the control plane. It allows network administrators to remotely manage network devices, monitor those devices, and analyze data collected about the devices. Protocols in this plane include SSH, Telnet, SNMP (Simple Network Management Protocol), and even HTTP for web-based user interfaces.

In a traditional network, these planes co-exist within each device and are distributed across the network, as illustrated in [Figure 7-11](javascript://).

**Figure 7-11**

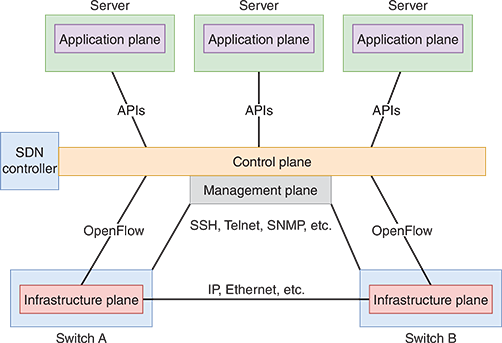
Each plane has its own functions and methods of communication



In an SDN network, the control plane (and, to a degree, the management plane) is moved to the SDN controller so devices can be centrally managed in conjunction with information from network applications, as shown in [Figure 7-12](javascript://).

**Figure 7-12**

SDN centralizes the control plane to an SDN controller



Notice the SDN controller serves as the middleman between network applications and network hardware to ensure the network can best support the needs of those applications. Communication between the SDN controller and network devices is called an SBI (southbound interface). Communication between the SDN controller and applications through APIs is called an NBI (northbound interface).

SDN controller design varies by manufacturer. One of the possible variations is the degree to which the control plane is centralized to the SDN controller. Cisco, for example, achieves an unusual balance so that some of the control plane is centralized while some of it remains distributed to the underlying hardware devices. Other key differences between SDN controllers include the following:

* The level of support for network virtualization tools
* The number of switches the SDN controller can support
* Its ability to function across a WAN connection
* The way the SDN solution scales as your network grows
* The types of security filtering offered
* The ability to provide centralized monitoring of all physical and virtual portions of the network

While SDN does increase complexity, it also increases performance and efficiency. SDN can often be used to manage network devices from multiple manufacturers, obtaining consistent management techniques on the network. Physical and virtual devices can all be managed from a central interface. SDN also creates the potential to implement more sophisticated network functions while using less-expensive devices. Networking hardware in an SDN architecture is significantly less expensive than their more sophisticated counterparts. In the marketplace, these low intelligence, brandless devices are often called white box switches. The SDN controller can also generate more complex rules for managing traffic, such as tables within tables or condition-dependent rules. As you can see, one of the primary advantages to separating the control plane from the data plane is to provide network technicians with more centralized control of network settings and management.

In recent years, the concept of SDN architecture has expanded to include management of fully virtualized network resources and resources that are hosted in places other than an organization’s own network (such as the cloud). All these resources can be centrally managed by a network’s administrator through the framework of modern SDN software. Later in this module, you’ll explore virtualization technologies, which is a significant step of abstraction away from physical devices. Further, virtualization lays the groundwork for a burgeoning IT industry: cloud computing. This module also covers the basics to get you started in the cloud. But first, let’s look at a special kind of network that abstracts physical storage away from network servers.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 7-1fStorage Area Network (SAN)

As you read about SDN, you learned that the control function of switches can be abstracted away from the switches and handled by a controller. This centralized, or consolidated, control allows for more flexible and responsive network path management. A similar principle can be applied to storage space on servers.

Server devices are optimized for compute functions. They might have powerful CPUs and loads of memory resources for fast data processing. Traditionally, each server contains its own storage space, which might include one or more storage drives called DAS (direct-attached storage). Making space on every server device to maximize its storage is not only bulky in your space-limited racks, but it’s also difficult to manage when you have dozens of servers. Controlling access to data stored on each server can also be difficult to manage for a large data center. What if you could consolidate storage from all those servers into one place? What kinds of benefits could that kind of architecture offer your data center?

Large enterprises that require fast access to data and large amounts of storage often have a specialized storage area network connected to their corporate network. A SAN (storage area network) is a distinct network of storage devices that communicate directly with each other and with other portions of the network. Essentially, a SAN abstracts storage services from compute services, and then provides high-speed network services to connect them.

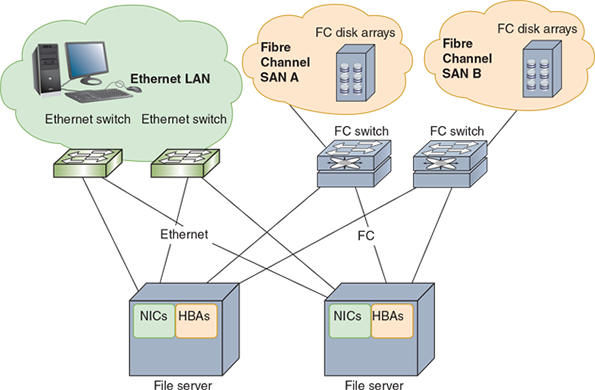
In a typical SAN, specialized SAN devices contain multiple storage drives and are designed to make data available to a network of servers. With multiple connections and clusters of storage devices arranged in RAID (Redundant Array of Independent Disks) arrays, this type of architecture is as fault tolerant as reasonably possible. If one storage device within a SAN suffers a fault, data is automatically retrieved from elsewhere in the SAN. If one connection in a SAN suffers a failure, another connection is already in place to handle network traffic, which is called [**multipathing**](javascript://). Multipathing techniques can also provide load balancing to help ensure that the demand on storage is spread evenly across all storage devices in the SAN.

SANs are not only extremely fault tolerant, but they are also extremely fast. Consider how quickly your computer’s CPU needs to access data stored on your computer’s storage drive. That data travels a very short distance, probably over a SATA (Serial Advanced Technology Attachment) cable, between your computer’s hard drive and its CPU. In a SAN, storage drives are located in one or more separate devices, possibly even in a different rack or a different room. The network connections between servers and SAN devices must support extremely high data throughput. To maximize throughput, SANs rely on one of these networking technologies:

* [**FC (Fibre Channel)**](javascript://) is a storage networking architecture that runs separately from Ethernet networks to maximize speed of data storage and access. Although FC can run over copper cables, fiber-optic cable is much more commonly used. Fibre Channel requires special hardware, which makes it an expensive storage connection technology. Specialized FC switches connect SAN storage devices with each other and with the outside network. Instead of NICs, FC devices connect with the network through HBAs (host bus adapters). At the time of this writing, FC networks can achieve speeds as high as 128 GFC (Gigabit Fibre Channel) over a single lane or approaching 512 GFC on quad lanes using QSFP technology. Specifications are in development to reach speeds up to 1 TFC (Terabit Fibre Channel) as soon as 2024. [Figure 7-13](javascript://) shows a Fibre Channel SAN connected to a traditional Ethernet network. Besides being expensive, Fibre Channel requires extensive training for IT personnel to support it.

**Figure 7-13**

A Fibre Channel SAN connected to an Ethernet LAN

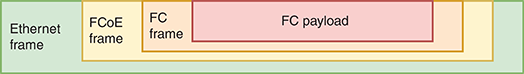


Enlarge Image

* [**FCoE (Fibre Channel over Ethernet)**](javascript://) is a newer technology that allows FC to travel over Ethernet hardware and connections. To do this, the FC frame is encapsulated inside an FCoE frame, which is then encapsulated inside an Ethernet frame, as illustrated in [Figure 7-14](javascript://). This preserves much of the higher speed capabilities of FC, along with the convenience and cost-efficiency of using existing Ethernet network equipment, as shown in [Figure 7-15](javascript://). With the installation of CNAs (converged network adapters), FCoE switches can connect to network servers and to switches for both the LAN and the SAN.

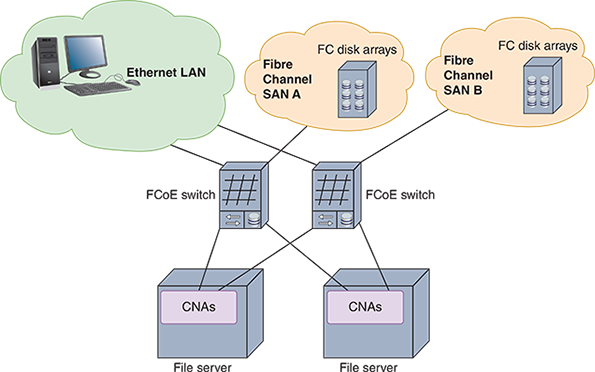
**Figure 7-14**

FCoE encapsulation



**Figure 7-15**

A SAN using FCoE to connect to a LAN



Enlarge Image

* [**iSCSI (Internet SCSI)**](javascript://), pronounced “i-scuzzy,” is a transport layer protocol that runs on top of TCP to allow fast transmissions over LANs, WANs, and the Internet. It can work on a twisted-pair Ethernet network with ordinary Ethernet NICs. iSCSI is an evolution of SCSI (Small Computer System Interface), which is a fast transmission standard used by internal hard drives and operating systems in file servers. The advantages of iSCSI over Fibre Channel are that it is not as expensive, can run on the already established Ethernet LAN by installing iSCSI software (called an iSCSI initiator) on network clients and servers, and does not require as much special training for IT personnel. Some network administrators configure iSCSI to use jumbo frames on the Ethernet LAN. iSCSI architecture is very similar to FC. The primary difference is that Ethernet equipment and interfaces can be used throughout the storage network. In fact, this is the primary advantage of iSCSI over other options, making it relatively straightforward to implement. iSCSI doesn’t offer nearly the same performance benchmarks, however, and currently maxes out around 10 Gbps with 40 Gbps speeds on the horizon.
* [**IB (InfiniBand)**](javascript://), like FC, requires specialized network hardware. Although it’s very fast, InfiniBand tends to serve a few niche markets rather than being widely available. IB falls on the difficult end of the installation and configuration spectrum, and it runs on the expensive side as well.

**Note 7-3**

You can get updated information about Fibre Channel and its related technology, FCoE (Fibre Channel over Ethernet), at [fibrechannel.org](http://fibrechannel.org/" \t "_blank).

A SAN can be installed in a location separate from the LAN it serves. For example, remote SANs can be kept in an ISP’s data center, which can provide greater security and fault tolerance and also allows an organization to outsource the management of its SAN.

SANs are highly scalable and have a very high fault tolerance, massive storage capabilities, and fast data access. SANs are best suited to environments with huge quantities of data that must always be quickly available. For example, consider the storage volumes used by VMs. Hosting large numbers of VMs is a prime use case for a SAN. Let’s look more closely at virtualization technology and how various components of that system take yet another step up in the progression of network abstraction.

**Remember This…**

* Explain how STP works.
* Compare three-tiered architecture with spine-and-leaf architecture.
* Explain the planes of SDN.
* Use common port security best practices.
* Describe SAN technologies, including FC, FCoE, and iSCSI.

**Self-Check**

1. Which STP bridge serves as the basis for all path calculations?

Answer

* 1. Designated bridge
  2. Header bridge
  3. Spanning bridge
  4. Root bridge

1. Which type of switch is best used for connections to web servers?

Answer

* 1. Edge switch
  2. Core switch
  3. Spine switch
  4. Aggregation switch

1. Which SDN plane moves traffic from switch port to switch port?

Answer

* 1. Control plane
  2. Management plane
  3. Application plane
  4. Infrastructure plane

1. Which SAN connection technology can run over ordinary Ethernet NICs without any special equipment needed?

Answer

* 1. FC
  2. iSCSI
  3. SATA
  4. IB

**You’re Ready**

You’re now ready to complete [Capstone Project 7-1: Secure Switch Ports in Packet Tracer](javascript://), or you can wait until you’ve finished reading this module.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# 7-2Virtual Architecture

### Certification

* 1.2

Explain the characteristics of network topologies and network types.

Average reading time: 30 minutes



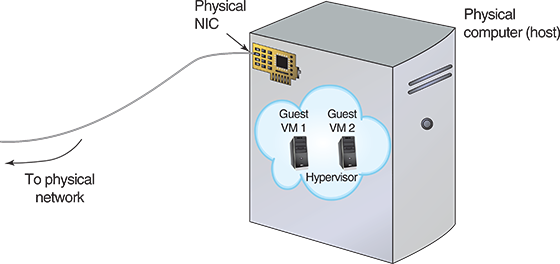
Beginning with the first module in this course, you’ve created and worked with a variety of virtual machines, or VMs, in several of the Capstone Projects. You’ve used Hyper-V or VirtualBox to create workstation and server VMs, and you installed both Windows and Linux operating systems on those VMs. By this point, you’ve had the opportunity to become familiar with the process and some of the terms involved in working with virtualization technology. Now you’re ready for a more thorough examination of what you’ve been accomplishing in these projects.

[**Virtualization**](javascript://) is a virtual, or logical, version of something rather than the actual, or physical, version. For example, when you create an Ubuntu server VM on a Windows PC, the Windows machine is the physical computer, or **host**, and the Ubuntu machine is a logical computer, or [**guest**](javascript://), that is hosted by the physical computer. The Ubuntu operating system acts as if it is installed on a separate, physical machine and is generally not aware of its physical host. In essence, virtualization has abstracted the computer system from the computer hardware so that multiple systems can exist on the same hardware. How is this possible?

The key to this feat is a type of software known as a hypervisor. A [**hypervisor**](javascript://) creates and manages a VM, and it allocates hardware resources for the host and any of its guest VMs. Together, all the virtual devices on a single computer share the same CPU, hard disks, memory, and physical network interfaces. Yet each VM can be configured to use a different operating system and can emulate a different type of CPU, storage drive, or NIC, than the physical computer it resides on. Meanwhile, to users, a VM appears and acts essentially the same as a physical computer running the same software. [Figure 7-16](javascript://) illustrates some of the elements of virtualization.

**Figure 7-16**

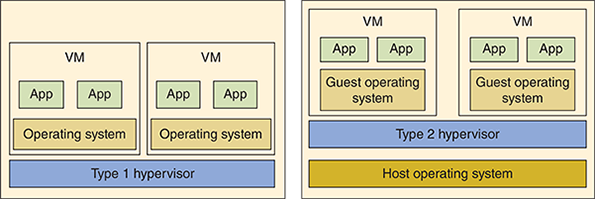
Elements of virtualization



There are two types of hypervisors: Type 1 and Type 2. The differences are diagrammed in [Figure 7-17](javascript://) and explained next.

**Figure 7-17**

Type 1 and Type 2 hypervisors



Enlarge Image

* [**Type 1 hypervisor**](javascript://)—Installs on a computer before any OS and is, therefore, erroneously called a “bare-metal” hypervisor. In reality, a type 1 hypervisor is itself a minimal operating system, many of which are built on a Linux kernel. Like any OS, the hypervisor relies on firmware to enable communication with the underlying hardware. It partitions the hardware computing power to multiple VMs, each with their own OS. Popular examples include Citrix Hypervisor, ESXi by VMware, and Hyper-V by Microsoft.

* **[Type 2 hypervisor](javascript://)**—Installs in a host OS as an application and is called a hosted hypervisor. VirtualBox, which you’ve seen in the Capstone Projects, is an example of a type 2 hypervisor, as is the popular VMware Player. A type 2 hypervisor is not as powerful as a type 1 hypervisor because it is dependent on the host OS to allot its computing power. VMs hosted by a type 2 hypervisor also are not as secure or as fast as a type 1 hypervisor’s VMs.

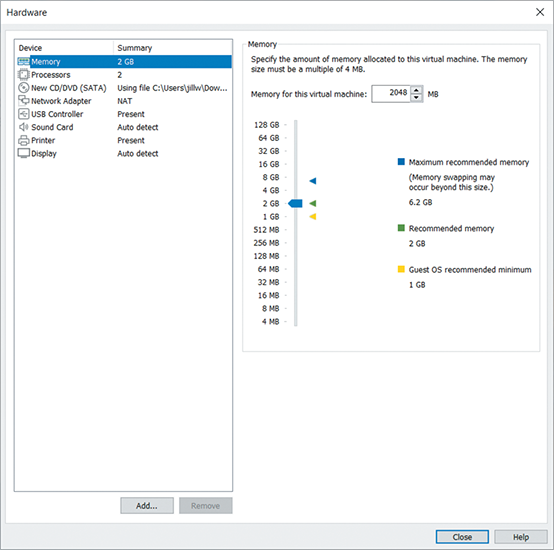
Hyper-V has elements of both categories. Hyper-V is embedded in Windows Server 2008 and beyond, and in Windows 10 (sometimes called Client Hyper-V), which you might have worked with in the Capstone Projects. Hyper-V Manager appears to run as an application and, therefore, looks more like a type 2 hypervisor. In reality, when Hyper-V is enabled in Windows, it creates a virtualization layer underneath the existing OS installation, thereby establishing its role as a type 1 hypervisor. This arrangement is unusual, however, in that the existing Windows OS continues to be given privileged access through the virtualization layer to the underlying hardware, while guest VMs are not given this level of access.

Another hypervisor, KVM (Kernel-based Virtual Machine), is native to Linux OSs and also has elements of both a type 1 and type 2 hypervisor. When installed, KVM effectively converts the existing Linux OS to a type 1 hypervisor. However, the original OS is still accessible and can still host other applications.

A VM’s software and hardware characteristics are assigned when it is created in the hypervisor. As you have learned, these characteristics can differ completely from those of the host machine. Keep in mind that a VM is entirely a logical entity—it’s not confined to the features of the local hardware in the same way that a physical machine is. You can customize the VM with a guest operating system, amount of memory, hard disk size, and processor type, to name just a few options. [Figure 7-18](javascript://) shows a screen from VMware Player’s VM creation wizard that allows you to specify the amount of memory allocated to a VM. Notice in the figure you could click on other devices in the hardware list, such as processors, optical disc drives, and the network adapter, to make changes to those specifications as well.

**Figure 7-18**

Specifying a VM’s memory in VMware



Enlarge Image

Source: VMware, Inc.

While there are limits imposed by the physical hardware, such as total available RAM or storage space, the hypervisor makes it possible for a VM guest to function differently than the host machine or other guest machines. As you learn more about virtualization, train yourself to notice how logical functions operate on a different layer, or plane, than what might be implied by the physical hardware. To help you understand the difference between physical systems and virtual systems, consider the following analogy.

Think about the foundation of a house. In many cases, a foundation is made of cement blocks or poured concrete, it usually reaches deep into the ground, and it defines the outline of the house to be built above it. Looking only at the foundation, however, would not necessarily indicate how many floors the house will have, what colors the walls will be, or what materials will be used on the inside or the outside. On a single foundation, you might even build two townhouses or four apartments. Similarly, the physical hardware of a computer defines some outer limits of capabilities, such as how much RAM is available to all running VMs or how much storage space is available for all VM-associated data. However, within those limits, you can create many virtual machines with a variety of characteristics. These VMs are managed by the hypervisor without being directly defined by the hardware supporting it all.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 7-2aNetwork Connection Types

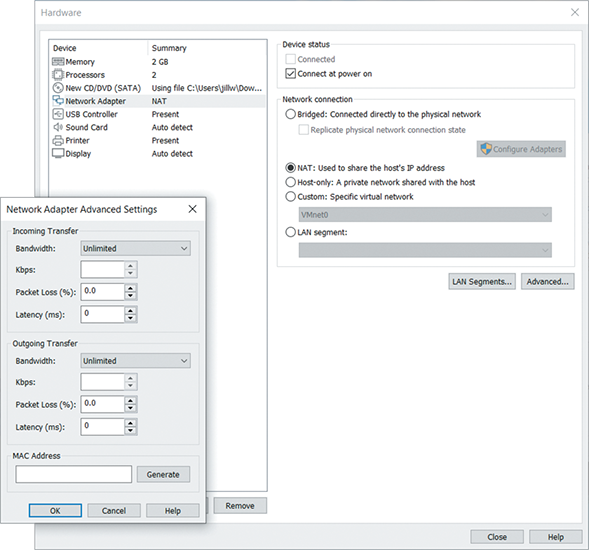
In [Module 4’s Project 4-2](javascript://), you used Remote Desktop to connect two computers. You had the opportunity to use a VM for one or both of the computers. And if you were using the VirtualBox hypervisor, you had to ensure the VM’s network adapter was set to Bridged mode. Now you’re ready to learn what that setting means.

Every VM has its own virtual network adapter, or **[vNIC (virtual NIC)](javascript://)**, that can connect the VM to a network, either virtual or physical. Just like a physical NIC, a vNIC operates at the data link layer and provides the computer with network access. Each VM can have several vNICs, no matter how many NICs the host machine has. The maximum number of vNICs on a VM depends on the limits imposed by the hypervisor. For example, VirtualBox allows up to eight vNICs per VM. Upon creation, each vNIC is automatically assigned a MAC address.

[Figure 7-19](javascript://) shows a dialog box from the VMware Player wizard that allows you to customize properties of a virtual workstation’s vNIC. One of many options you can configure for each vNIC is its inbound and outbound transmission speeds. For example, you could select transmission speeds that simulate a DSL or cable broadband connection, which you’ll learn more about later.

**Figure 7-19**

Customizing vNIC properties in VMware



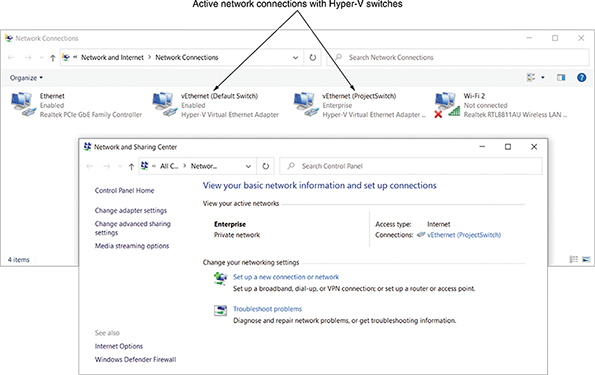
Enlarge Image

Source: VMware, Inc.

As soon as the virtual machine’s vNIC is selected, the hypervisor creates a connection between that VM and the host. Depending on the hypervisor, this connection might be called a bridge or a switch. This **[vSwitch (virtual switch)](javascript://)** or bridge is a logically defined device that operates at the data link layer to pass frames between nodes. Thus, it can allow VMs to communicate with each other and with nodes on a physical LAN. In a type 1 hypervisor, such as Hyper-V, the host and its guests all use the vSwitch (see [Figure 7-20](javascript://)). In a type 2 hypervisor, such as VirtualBox, the guests rely on the vSwitch while the host remains connected directly to the physical network.

**Figure 7-20**

The OS of a host computer running Hyper-V is connected to the physical network through Hyper-V’s vSwitch

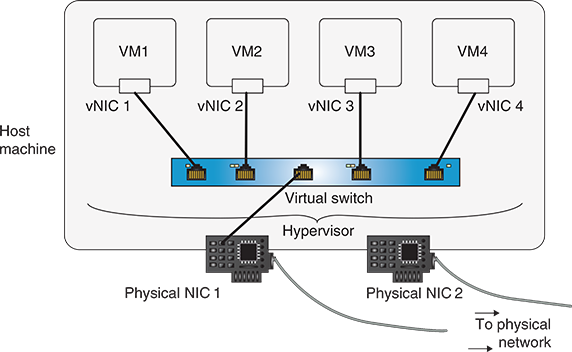


Enlarge Image

One host can support multiple virtual switches, which are controlled by the hypervisor. [Figure 7-21](javascript://) illustrates a host machine with two physical NICs that support several virtual machines and their vNICs. A virtual switch connects the vNICs to the network.

**Figure 7-21**

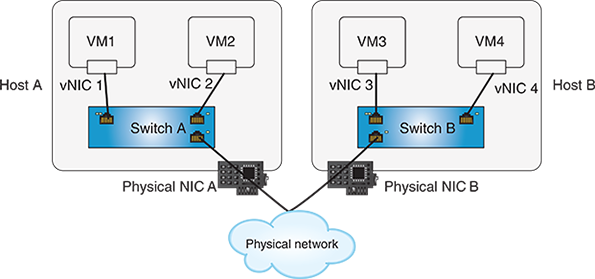
Virtual servers on a single host connect with a virtual switch



VMs can go through a virtual switch on the host computer to reach the physical network and can communicate with physical or virtual routers, other network devices, and other hosts on the local or another network. For example, in [Figure 7-22](javascript://), a VM on Host A can communicate with a VM on Host B.

**Figure 7-22**

Virtual switches exchange traffic across the physical network



Enlarge Image

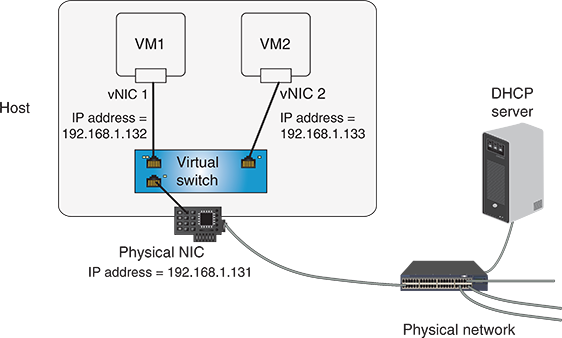
The way a vNIC is configured determines whether the VM is joined to a virtual network or attempts to join the physical LAN that the host machine is connected to. These various configurations are called networking modes, the most common of which are bridged, NAT, and host-only, as described next. These descriptions are specific to the type 2 hypervisors you’ve been using in your projects. However, type 1 hypervisors offer these and other network configurations as well.

### Bridged Mode

In [**bridged mode**](javascript://), a vNIC accesses a physical network using the host machine’s NIC, as shown in [Figure 7-23](javascript://). In other words, the virtual interface and the physical interface are bridged. If your host machine contains multiple physical adapters—for example, a wireless NIC and a wired NIC—you can choose which physical adapter to use as the bridge when you configure the virtual adapter.

**Figure 7-23**

This vNIC accesses the physical network directly in bridged mode



Although a bridged vNIC communicates through the host’s adapter, it obtains its own IP address, default gateway, and subnet mask from a DHCP server on the physical LAN. For example, suppose your DHCP server is configured to assign addresses in the range of 192.168.1.100 through 192.168.1.254 to nodes on your LAN. The router might assign your host machine’s physical NIC an IP address of 192.168.1.131. A guest on your host might obtain an IP address of 192.168.1.132. A second guest on that host might obtain an IP address of 192.168.1.133, and so on.

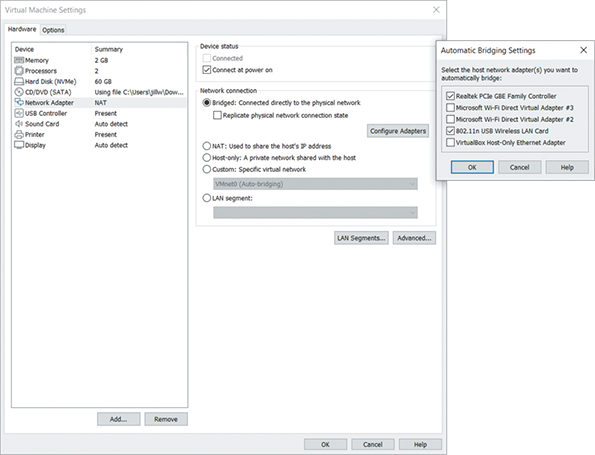
When connected using bridged mode, a VM appears to other nodes as just another client or server on the network. Other nodes communicate directly with the VM without realizing it is virtual.

**Note 7-4**

In VMware and VirtualBox, you can choose the bridged connection type when you create or configure the virtual adapter. In KVM, you create a bridge between the VM and your physical NIC when you modify the vNIC’s settings. In Hyper-V, you create a bridged connection type by assigning VMs to an external network switch. Additionally, bridged mode is the most common networking mode for VMs hosted by type 1 hypervisors such as ESXi. [Figure 7-24](javascript://) shows the bridging options for a virtual machine in VMware Player with the Bridged network connection type selected.

**Figure 7-24**

Selecting the Bridged option for a vNIC in VMware Player



Enlarge Image

Source: VMware, Inc.

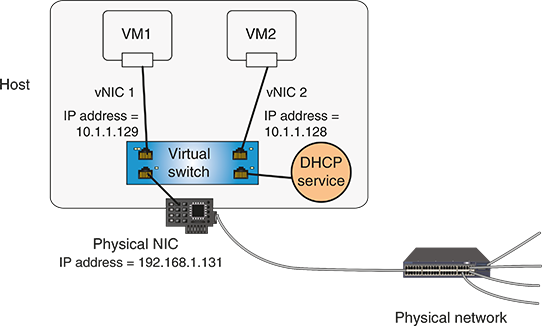
VMs that must be available at a static IP address, such as mail servers or web servers, should be assigned bridged network connections. However, VMs that other nodes do not need to access directly can be configured to use the NAT networking mode.

### NAT Mode

In [**NAT Mode**](javascript://), a vNIC relies on the host machine to act as a NAT (network address translation) device. In other words, the VM obtains IP addressing information from its host rather than from a server or router on the physical network. To accomplish this, the hypervisor acts as a DHCP server. A vNIC operating in NAT mode can still communicate with other nodes on the network and vice versa. However, other nodes communicate with the host machine’s IP address to reach the VM; the VM itself is invisible to nodes on the physical network. [Figure 7-25](javascript://) illustrates a VM operating in NAT mode.

**Figure 7-25**

The vNIC accesses the physical network via NAT in NAT mode

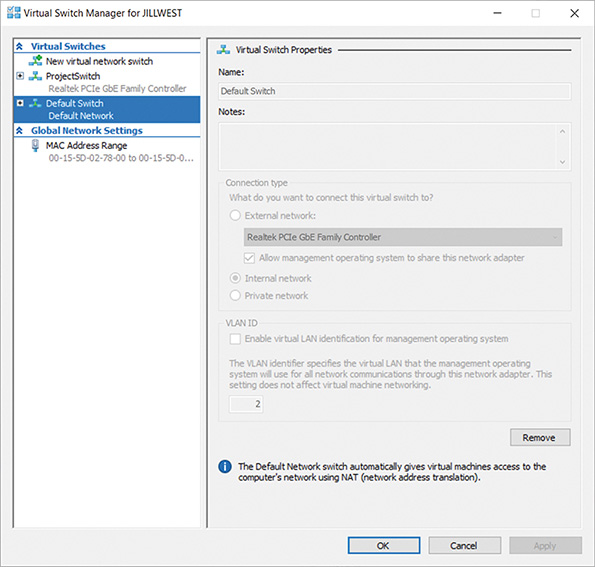


**Note 7-5**

NAT is the default network connection type selected when you create a VM in VMware Player, VirtualBox, or KVM. In Hyper-V, the NAT connection type is created by assigning VMs to an internal network run on a virtual switch. [Figure 7-26](javascript://) shows the default switch in Hyper-V, which hosts an internal network type using NAT. Notice all other options are grayed out. The only way to change these other options is to create a new virtual network switch in the Hyper-V Virtual Switch Manager.

**Figure 7-26**

The default switch in Hyper-V gives VMs access to the host’s network using NAT



Enlarge Image

Once you have selected the NAT configuration type, you can configure the pool of IP addresses available to the VMs on a host. For example, suppose, as shown in [Figure 7-25](javascript://), your host machine has an IP address of 192.168.1.131. You might configure your host’s DHCP service to assign IP addresses in the range of 10.1.1.120 through 10.1.1.254 to the VMs you create on that host. Because these addresses will never be evident beyond the host, you have flexibility in choosing their IP address range.

The NAT network connection type is appropriate for VMs that do not need to be accessed at a known address by other network nodes. For example, virtual workstations that are mainly used to run stand-alone applications, or that serve as test beds to test applications or operating system installations, are good candidates for NAT network connections.

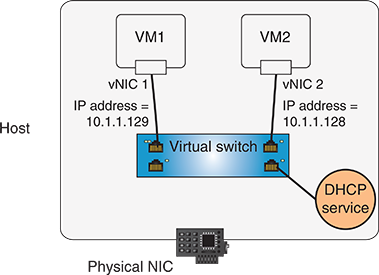
### Host-Only Mode

In [**host-only mode**](javascript://), VMs on one host can exchange data with each other and with their host, but they cannot communicate with any nodes beyond the host. In other words, the vNICs never receive or transmit data via the host machine’s physical NIC. In host-only mode, as in NAT mode, VMs use the DHCP service in the host’s virtualization software to obtain IP address assignments.

[Figure 7-27](javascript://) illustrates how the host-only option creates an isolated virtual network. Host-only mode is appropriate for test networks or if you simply need to install a different operating system on your workstation to use an application that is incompatible with your host’s operating system. For example, suppose a project requires you to create diagrams in Microsoft Visio and your workstation runs Red Hat Linux. You could install a Windows 10 VM solely for the purpose of installing and running Visio.

**Figure 7-27**

vNICs in a host-only network can only talk to other VMs running on that host

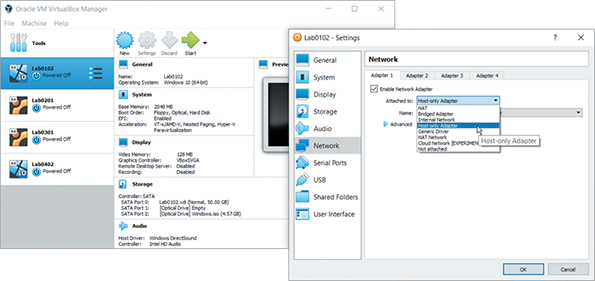


Obviously, because host-only mode prevents VMs from exchanging data with a physical network, this configuration cannot work for virtual servers that need to be accessed by clients across a LAN. Nor can it be used for virtual workstations that need to access LAN or WAN services, such as email or web pages. Host-only networking is less commonly used than NAT or bridged mode networking.

[Figure 7-28](javascript://) shows the “Host-only Adapter” mode selected in VirtualBox. Notice a similar mode called Internal Network. In VirtualBox, the difference is that with Host-only networking, VMs can communicate with the physical host. With Internal networking, VMs are isolated even from the host and can only communicate with each other.

**Figure 7-28**

VirtualBox offers both Host-only mode and Internal mode



Enlarge Image

Source: Oracle Corporation

**Note 7-6**

You can choose host-only networking when you create or configure a VM in VMware or VirtualBox. In Hyper-V, the host-only connection type is created by assigning VMs to a private virtual network. In KVM, host-only is not a predefined option, but can be assigned to a vNIC via the command-line interface.

Virtualization software gives you the flexibility of creating several networking types on one host machine. For example, on one host you could create a host-only network to test multiple versions of Linux. On the same host, you could create a group of Windows Server machines that are connected to your physical LAN using the bridged connection type. Or, rather than specifying one of the networking connection types described previously, you could also create a VM that is not connected to any nodes, whether virtual or physical. Preventing the VM from communicating with other nodes keeps it completely isolated. This might be desirable when testing unpredictable software or an image of untrusted origin.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 7-2bPros and Cons of Virtualization

Virtualization has become the de facto standard for many resources on enterprise networks worldwide. It’s unlikely you’ll work with a business-grade network that does not incorporate virtualization in some way. Virtualization offers several advantages, including the following:

* **Efficient use of resources**—Physical clients or servers devoted to one function typically use only a fraction of their capacity. Without virtualization, a company might purchase five computers to run five different services—for example, an email server, a file server, two web servers, and a database server. The [On the Job](javascript://) story at the beginning of this module gave an example of a company running three different servers for various tasks. Each server needs its own, dedicated system, resulting in a lot of waste because each service might demand only 10–20 percent of its computer’s processing power and memory. With virtualization, a single, powerful computer can support all five services. This creates a significant single point of failure, however, if this one server goes down for any reason. Therefore, in actual practice, most of these network services are also duplicated across multiple physical servers.
* **Cost and energy savings**—With virtualization, organizations save money by purchasing fewer and less-expensive physical machines. They also save electricity because there are fewer and more efficient computers drawing power and less demand for air conditioning in the computer room, as you saw in this module’s [On the Job](javascript://) story. Some institutions with thousands of users, such as Stanford University, are using virtualization to conserve energy and are promoting it as part of campus-wide sustainability efforts. Thin clients, for example, are very small, energy-efficient computers that can be used to populate large computer labs on a college campus. Thin clients connect to a central server to perform most of their processing functions. When a user signs into a domain account on the thin client, the thin client then contacts the server for all other functions. The server hosts the thin client’s software, including the operating system and most or all applications. In other words, the thin client’s entire desktop is virtualized and hosted by the server.
* **Fault and threat isolation**—In a virtual environment, the isolation of each guest system means that a problem with one guest does not affect the others. For example, an instructor might create multiple instances of an operating system and applications on a single computer that’s shared by several classes. This allows each student to work on their own instance of the OS environment. Any configuration errors or changes they make on their guest machine will not affect other students. In another example, a network administrator who wants to try a beta version of an application might install that application on a guest machine rather than on the host, in case the untested software causes problems. Furthermore, because a VM is granted limited access to hardware resources, security attacks on a guest pose less risk to a host or the physical network to which it’s connected.
* **Simple backups, recovery, and replication**—Virtualization software enables network administrators to save backup images of a guest machine. The images can later be used to recreate that machine on another host or on the same host. This feature allows for simple backups and quick recovery. It also makes it easy to create multiple, identical copies of one VM, called clones. Some virtualization software allows you to save image files of VMs that can be imported into a competitor’s virtualization software.

Not every type of client or server is a good candidate for virtualization, however. Potential disadvantages to creating multiple guests on a single host machine include the following:

* **Compromised performance**—When multiple VMs contend for finite physical resources, one VM could monopolize those resources and impair the performance of others on the same computer. In theory, careful management and resource allocation should prevent this. In practice, however, it is unwise to force a critical application—for example, a hospital’s emergency medical systems or a factory’s real-time control systems—to share resources and take that risk. Imagine a brewery that uses computers to measure and control tank levels, pressure, flow, and temperature of liquid ingredients during processing. These functions are vital for product quality and safety. In this example, where specialty software demands real-time, error-free performance, it makes sense to devote all a computer’s resources to this set of functions, rather than share that computer with, for example, the brewery’s human resources database server. In addition to multiple guest systems vying for limited physical resources, a hypervisor also requires some overhead.
* **Increased complexity**—Although virtualization reduces the number of physical machines to manage, it increases complexity and administrative burden in other ways. For instance, a network administrator who uses virtual servers and switches must thoroughly understand virtualization software. In addition, managing addressing and switching for multiple VMs is more complex than doing so for physical machines. Finally, because VMs are so easy to set up, they might be created capriciously or as part of experimentation, and then forgotten. As a result, extra VMs might litter a server’s hard disk, consume resources, and unnecessarily complicate network management. By contrast, abandoned physical servers might only take up rack space.
* **Increased licensing costs**—Because every instance of commercial software requires its own license, every VM that uses such software comes with added cost. In some cases, the added cost brings little return. For example, a software developer might want to create four instances of Windows Server on a single computer to test new software using four testing procedures on four different OS installation configurations. To comply with Microsoft’s licensing restrictions, the developer will have to purchase four licenses for Windows Server. Depending on the developer’s intentions, it might make more sense, instead, to share one installation of Windows Server and separate the four testing procedures by using four different logon IDs. Alternatively, the developer could save the initial VM image and start over fresh for each test.
* **Single point of failure**—If a host machine fails, all its guest machines will fail, too. As mentioned earlier, an organization that creates VMs for its email server, file server, web servers, and database server on a single physical computer would lose all those services if the one, physical computer went down. Wise network administrators implement measures such as clustering and automatic failover to prevent that from happening.

Most of the potential disadvantages in this list can be mitigated through thoughtful design and virtualization control. Similarly, the same advantages and disadvantages of client virtualization apply to virtualizing other network devices. Next, let’s look at what can be accomplished when virtualization technology is used elsewhere on the network.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 7-2cNFV (Network Functions Virtualization)

You’ve seen how a single workstation can host many VM workstations or servers, each with its own network connection, operating system, and applications. Networking devices can also be virtualized. For example, instead of purchasing an expensive hardware firewall to protect a LAN, suppose you were to install a firewall’s operating system in a VM on an inexpensive server. Suppose you also install a router VM on that server instead of purchasing an expensive hardware router. You’ve now provided your network with two sophisticated, virtualized devices—a virtual firewall and a virtual router—on one, inexpensive server instead of paying for two, expensive, dedicated devices.

**Note 7-7**

To clarify, a software firewall is merely an application, like Windows Firewall. It’s very limited in scope and features, and only services a single client. A dedicated firewall device, such as those made by Fortinet, Cisco, or Palo Alto Networks, serves an entire network (or portion of a network). It has many more features than a firewall app and runs its own OS.

A virtual firewall emulates a hardware firewall, and it’s hosted in a virtualized environment. Examples include the pfSense virtual firewall by Netgate and Barracuda’s CloudGen firewall, which can also provide protection for cloud-based portions of the network. There must be a hypervisor present (usually type 1) to host a virtual firewall.

These distinctions apply to other devices as well, such as routers, switches, and load balancers.

Now replicate these savings over dozens of devices for a large network, and you can begin to see some of the advantage of virtualizing network functions. Other advantages include the following:

* Virtual devices can be quickly and sometimes automatically migrated, or moved, from one server to another in the event of a hardware failure or maintenance.
* Resources, such as hardware, energy usage, and physical space, are utilized more efficiently.
* Services can be easily scaled to meet the changing needs of a network.

Merging physical and virtual network architecture is called [**NFV (Network Functions Virtualization)**](javascript://). NFV provides flexible, cost-saving options for many types of network devices, including virtual servers, data storage, load balancers, and firewalls. However, there are a few caveats and considerations to keep in mind:

* You’ll need licenses for each of the virtualized devices as well as for the type 1 hypervisor that will host them. Fortunately, the cost of these licenses amounts to a fraction of the cost of similarly featured hardware devices.
* The interaction between physical and virtual devices introduces a small degree of latency as data passes through the hypervisor and its connections. Usually, this delay is negligible. However, it might be a relevant consideration in some cases.
* Even some of the most die-hard virtualization fans are uncomfortable using a virtual firewall to protect the entire network. The server hosting a virtual firewall occasionally needs to be restarted in the course of regular maintenance or some kind of failure, and in that event, the hosted firewall goes down with the server. Instead, many network admins believe that virtual firewalls are only appropriate for securing virtual-only portions of the network, or for serving as a backup to physical firewall devices.

**Remember This…**

* Compare type 1 and type 2 hypervisors.
* Describe virtual network devices, such as a vSwitch and a vNIC.
* Explain the role of NFV in optimizing network architecture.

**Self-Check**

1. Which virtual network connection type assigns a VM its IP address from the physical network?

Answer

* 1. NAT
  2. Bridged
  3. Private
  4. Host-only

1. Which network architecture technique are you using when you run a virtual router on a network?

Answer

* 1. STP
  2. SAN
  3. NFV
  4. SDN

**You’re Ready**

You’re now ready to complete [Capstone Project 7-2: Explore Virtual Network Configuration Options in Hyper-V](javascript://) or [Capstone Project 7-3: Explore Virtual Network Configuration Options in VirtualBox](javascript://), depending on which hypervisor you’ve been using throughout this course. Or you can wait until you’ve finished reading this module.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# 7-3Cloud Architecture

### Certification

* 1.7

Explain basic corporate and datacenter network architecture.

* 1.8

Summarize cloud concepts and connectivity options.

* 3.1

Given a scenario, use the appropriate statistics and sensors to ensure network availability.

Average reading time: 26 minutes



On network diagrams, the Internet is frequently drawn as a cloud allowing access to information stored on web servers around the world. When it was first introduced, the image captured something essential about modern networking, so now the term [**cloud computing**](javascript://), or simply cloud, refers to the flexible provision of data storage, applications, or services to clients over the Internet. You might already be familiar with cloud storage services such as Dropbox, OneDrive, and Google Drive, which let you store your own data on web-based servers. Web-based email is another example of cloud computing. Most cloud service providers use virtualization software to supply multiple platforms to users. For example, industry leaders Rackspace (in its private, public, or hybrid cloud products) and Amazon (in its Elastic Compute Cloud, or EC2, service) use Xen virtualization software by Citrix to create virtual environments for their customers.

Cloud computing covers a broad range of services, from hosting websites and database servers to providing virtual servers for collaboration or software development. You can think of the cloud as an abstraction of IT from the data center. That’s not entirely accurate, as you can run your cloud in your own data center. However, a cloud architecture gives you a lot more flexibility in choosing and configuring your resources. Let’s look at this distinction a little more closely. All cloud services have the following features in common, according to NIST (National Institute of Standards and Technology):

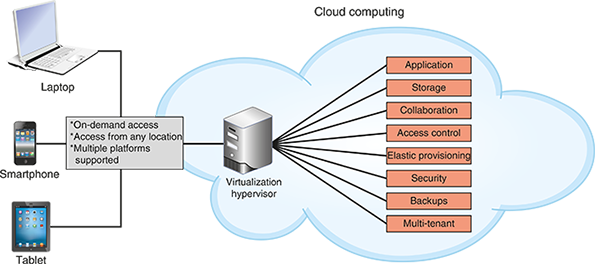
* **On-demand self-service**—Services, applications, and storage in a cloud are available to users at any time, upon the user’s request.
* **Broad network access**—Client devices of all types, including smartphones, laptops, desktops, thin clients, and tablet computers, can access services, applications, and storage in a cloud, no matter what operating system they run or where they are located, if they have an Internet connection.
* **Resource pooling**—Host computers in the cloud provide multiple services or resources such as disk space, applications, and services that are pooled, or consolidated. For example, a single cloud provider can host hundreds of websites for hundreds of different customers on just a few servers. This is called [**multitenancy**](javascript://) where several customers (tenants) pay for resources running on the same hardware.
* **Measured service**—Everything offered by a cloud provider, including applications, desktops, storage, and other services, is measured. A provider might limit or charge by the amount of bandwidth, processing power, storage space, or client connections available to customers.
* **Rapid elasticity**—Services and storage capacity can be scaled up or down without negatively affecting the efficiency and effectiveness of the workload, which is called scalability. The fact these resources can be changed quickly and dynamically—even automatically—in response to changing demands, refers to [**elasticity**](javascript://). For example, if your website suddenly receives a burst of traffic following a major marketing campaign, scalability ensures you can increase the cloud resources the website needs without disrupting your web services, while elasticity refers to your web servers’ ability to scale up quickly and automatically as soon as the increased traffic is detected.

Let’s consider a scenario where cloud enables a company to partner with people who are scattered across the globe. Suppose an organization that develops graphic design software employs dozens of creative and highly skilled developers on a project. These developers, half of them working from home, are located in six different countries. How can these employees, located so far away from the central office and from each other, collaborate successfully?

The company contracts with a [**CSP (cloud services provider)**](javascript://) to host its servers, making the company’s test platforms easily accessible to any of its employees via the Internet. The company’s developers can load any kind of software on the servers, test it from afar, and share this content with distant members of the team. If additional storage space is needed, that resource can be dynamically allocated. This means the storage space reserved for the software can be increased automatically as the need arises. Later, when it’s no longer needed, that space can be freed up again for other developers. In addition, the CSP ensures the underlying hardware servers hosting the cloud resources are secure and regularly backed up. Cloud removes from the company’s IT personnel the burden of managing the underlying hardware so they can focus on other priorities. [Figure 7-29](javascript://) illustrates some of the benefits of cloud for this organization.

**Figure 7-29**

Characteristics of cloud services



Enlarge Image

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

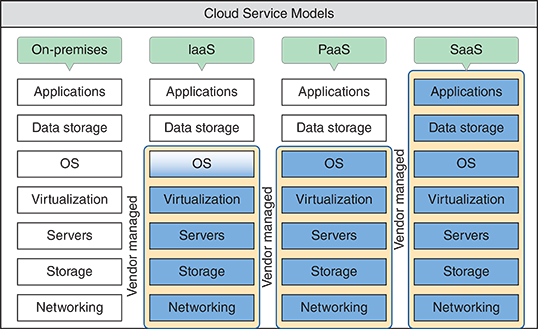
## 7-3aCloud Service Models

As you can see, managing cloud resources involves shared responsibilities—the ISP has some responsibility, and the cloud customer has other responsibilities. The dividing line between those groups of responsibilities varies according to the service being used. It’s important for a cloud customer to understand exactly what they’re responsible for and exactly what they can count on the CSP to provide for each service the customer uses. Most of the time, this dividing line is consistent for services within certain service models.

[**Cloud service models**](javascript://) are categorized by the types of services they provide. NIST has developed a standard definition for each category, which varies by the division of labor implemented. For example, as shown in [Figure 7-30](javascript://), an organization is entirely responsible for their own network, top to bottom. In this traditional arrangement for an [**on-premises data center**](javascript://), which is the physical location of the customer and the hardware they own, the organization maintains its own network infrastructure devices, manages its own network services and data storage, and purchases licenses for its own applications. Three of the many cloud service models are also illustrated in [Figure 7-30](javascript://). These common service models incrementally increase the amount of management responsibilities outsourced to cloud vendors.

**Figure 7-30**

At each progressive level of these cloud service models, the vendor takes over more computing responsibility for the organization



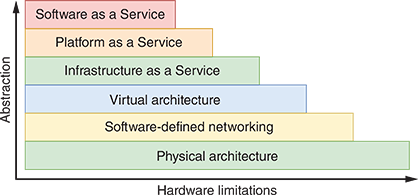
To understand the various service models, it’s helpful to compare them to the many ways to acquire a pizza for dinner. On the traditional end of the scale, you can make the pizza yourself. On the other end of the scale, you can have someone else make it and serve it to you. Check out each service model in [Figure 7-30](javascript://), and explore their differences using a pizza analogy:

* **On-premises**—All the hardware, software, and everything else is located and managed at your location. This would be like making your own pizza from scratch at home. You provide all the ingredients, bake it in your own oven, and eat it at your own table. For example, you install Microsoft Office on your laptop and keep all your documents on your hard drive. You can work with Office and your documents without being connected to the Internet.
* [**IaaS (Infrastructure as a Service)**](javascript://)—Hardware services are provided virtually, including network infrastructure devices such as virtual servers and DNS services. These services rely on the network infrastructure at the vendor’s site, but customers are responsible for their own application installations, data management and backup, and possibly operating systems. For example, customers might use the vendor’s servers to store data, host websites, and provide email, DNS, or DHCP services, but must provide their own NOS licenses and productivity software, such as customer tracking, sales management, and an office suite.

In the pizza analogy, this would be like a take-and-bake restaurant. You decide the type of crust you want and the toppings; the restaurant puts it all together for you. Then you take the unbaked pizza home, bake it yourself, and eat it at your own table. AWS (Amazon Web Services) provides many IaaS services, such as EC2 (Elastic Compute Cloud), which allows you to create and run your own VMs in the cloud. AWS provides the processing power, storage space, and deployment services. You create VMs and choose OSs to install on them. You load applications, databases, etc., and run Internet and other network services on them. You can think of IaaS as the least abstracted of the cloud service models, as shown in [Figure 7-31](javascript://).

**Figure 7-31**

Abstraction levels of cloud service models



* [**PaaS (Platform as a Service)**](javascript://)—Developers often require access to multiple platforms during the development process. A platform in this context includes the operating system, the runtime libraries or modules the OS provides to applications, and the hardware on which the OS runs. Rather than purchasing and maintaining a separate device for each platform, another option is to subscribe to PaaS services. Developers can build and test their applications within these virtual, online environments, which are tailored to the specific needs of the project. Many new cloud technologies have emerged over recent years in relation to the PaaS category, including [**containers**](javascript://), which are essentially micro-versions of servers that provide only the resources needed to run an application, and [**serverless compute**](javascript://), which are services that allow customers to run their code directly in the cloud without having to manage a server environment at all. Alternatively, an organization’s entire network might be built on platform services provided by a vendor. Any platform managed by a vendor resides on the vendor’s hardware and relies on their uptime and accessibility to meet performance parameters. However, the customers are responsible for their own applications and/or data storage, including maintaining backups of the data.

In the pizza analogy, this is the delivery option. You decide on the crust and toppings, the restaurant bakes it for you, and then they bring it to your front door within 30 minutes. You provide your own table and do the cleanup after dinner. GCP (Google Cloud Platform at [cloud.google.com](http://cloud.google.com/" \t "_blank)) specializes in PaaS where you can run code directly in the cloud without needing to create a server environment. Similarly, Alexa (Amazon’s personal assistant app) runs code in a PaaS or FaaS (Function as a Service) called Lambda—when you talk to Alexa, you’re talking to AWS’s cloud.

* [**SaaS (Software as a Service)**](javascript://)—Applications are provided through an online user interface and are compatible with a variety of devices and operating systems. Online email services such as Gmail and Yahoo! are good examples of SaaS, as are CRM (customer relationship management) apps, such as Salesforce and Zoho, and online office productivity tools such as Microsoft’s Office 365 and Google Docs. Except for the interface itself (the device and whatever browser software is required to access the website), the vendor provides every level of support from network infrastructure through data storage and application implementation.

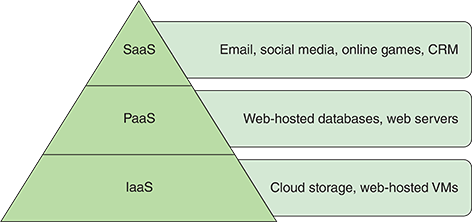
Here you see the full capability of pizza provider services. The restaurant provides the crust and all the ingredients, bakes it for you, and serves it directly to the table in the restaurant they have provided. You had to get yourself to the restaurant, but you didn’t need to bring anything to make it all work (except your payment, of course), and they do the cleanup after you leave. This is similar to applications you run online, like email, office productivity apps, or CRM software.

* [**XaaS (Anything as a Service)**](javascript://)—In this broader model, the “X” represents an unknown, just as it does in algebra. (And you thought you would never again use algebra.) Here, the cloud can provide any combination of functions depending on a client’s exact needs. This includes monitoring, storage, applications, and virtual desktops. For example, [**DaaS (Desktop as a Service)**](javascript://) is a type of SaaS where you can access a virtual desktop through your browser. The virtual desktop will include an OS and a range of installed applications, depending on how you have the service configured, what licenses you bring to the table, and how much you’re willing to pay. Recall that VDI is based on the same concept—DaaS uses the same technology except the CSP hosts the back end of the VDI deployment.

Consider the service models as they’re shown in [Figure 7-32](javascript://). The smaller, upper end of the pyramid indicates how little a SaaS customer needs to understand and interact with a cloud provider’s infrastructure for the customer to perform their work. In contrast, an IaaS customer interacts more heavily with their service provider’s infrastructure for every aspect of their cloud needs. IaaS is more pervasively integrated with a client’s computer network than is SaaS.

**Figure 7-32**

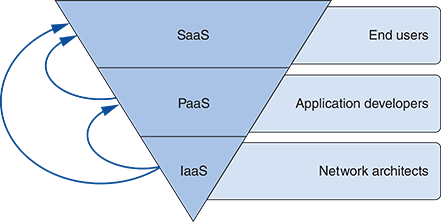
IaaS customers must understand more about a cloud provider’s platform and services than SaaS customers



At the same time, consider how accessible each type of cloud service is to end users. In [Figure 7-33](javascript://), the triangle is upside-down. End users, the largest group of cloud computing consumers, can easily access and use SaaS products without much setup, whereas IaaS products require extensive preparation by a much smaller group of more skilled network architects and administrators, who provide systems for their users. In the middle of this pyramid is PaaS, which is often used by application developers, both professionals and laypersons, for testing their products. Customers at the lower layers of this pyramid build products that support customers at the higher layers, such as when a company subscribes to an IaaS product, on which it offers its own PaaS products to its own unique market of customers.

**Figure 7-33**

SaaS are more immediately accessible to a wide market of end users than other categories of cloud services



Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 7-3bCloud Deployment Models

Cloud services are delivered in a variety of deployment models, depending on who manages the underlying hardware and who has access to it. The main deployment models you are likely to encounter are the following:

* [**Public cloud**](javascript://)—Service provided over public transmission lines, such as the Internet. Most of the examples discussed in this part of the module operate in public clouds.
* [**Private cloud**](javascript://)—Service established on an organization’s own servers in its own data center, or established virtually for a single organization’s private use and made available to users over a WAN connection through some type of remote access. If hosted internally, this arrangement allows an organization to use existing hardware and connectivity, potentially saving money. If hosted virtually, the organization benefits from the usual advantages of virtual services, such as scalability and accessibility.
* [**Community cloud**](javascript://)—Service shared between multiple organizations, but not available publicly. Organizations with common interests, such as regulatory requirements, performance requirements, or data access, might share resources in this way. For example, a medical database might be made accessible to all hospitals in a geographic area. In that case, the community cloud could be hosted internally by one or more of the organizations involved or hosted by a third-party provider. But it would not be made available to the public.
* [**Hybrid cloud**](javascript://)—A combination of public and private cloud resources. In the real world, the hybrid cloud infrastructure is a common result of transitory solutions. (In IT, “solution” refers to a product, service, or combination of products and services, and often includes extra features such as ongoing customer service.) An example of a hybrid cloud by design might arise when a company stores data in a private cloud but uses a public cloud email service.
* [**Multicloud**](javascript://)—A combination of the other service models in a single deployment, and probably the most common service model in use. In the real world, multicloud infrastructure is a common result of “best in class” selections of available cloud services. A company might choose the best cloud service for their databases from one CSP, the best cloud service for their web servers from another CSP, and the best cloud service for their CRM software from another CSP. The challenge here is finding ways to link the various cloud services in productive ways. This is becoming easier, especially as third-party providers such as Aviatrix develop products designed to support a multicloud environment.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 7-3cOrchestration and Automation

As you read about in the virtualization section, one potential drawback of working in a virtualized or cloud environment is that it’s easy to spin up, or create, virtual resources (such as a cloud-based VM) and then forget about them as they continue to accrue charges. Another, related challenge in the cloud is creating identical resources over time and tracking what changes are made to the cloud infrastructure and when. As companies rely more heavily on cloud services and give more privileged access to more employees, it can be difficult to track who makes what changes and how those changes affect other resources.

In some projects at the end of this module, you’ll use AWS’s management console, a web-based GUI, to spin up cloud resources. However, cloud resources can also be managed through a CLI. In fact, CLI-based management provides a more traceable and consistent method of making changes in the cloud. For example, suppose you want to create an EC2 instance using the t2.micro instance type, as you’ll be doing in [Project 7-2](javascript://). Instead of clicking through the GUI, you could enter the following command at the AWS CLI:

The command entered in the Amazon Web Services command line utility for creating an Elastic Compute 2 virtual machine instance using the t 2 dot micro instance type. a w s e c 2 run hypen instances hyphen hyphen image hyphen i d a m i hyphen 0 f f 8 a 9 1 5 0 7 f 7 7 f 8 6 7 hyphen hyphen count 1 hyphen hyphen instance hyphen type t 2 dot micro hyphen hyphen key hyphen name My Key Pair hyphen hyphen security hyphen groups My S G.

Enlarge Image

While this command might seem overwhelming, once you have it built with the proper parameters, it’s a simple matter of copy-and-paste to run it again and again. Suppose, instead of entering this command at a CLI, you built a series of commands in a script that you could feed the cloud platform. This process of using text-based commands in a computer-readable configuration file to create and manage cloud resources is called **[IaC (infrastructure as code)](javascript://)**. IaC allows you to log changes made to your cloud resources—you can track who made the changes, when changes were made, and what was the state of your cloud resources before and after each change. You can even use this information to revert to an earlier state if needed.

As you’ve already read, many cloud resources can be scaled up or down automatically. Sensibly, a programmed, computer-generated response to a specific event is referred to as [**automation**](javascript://). However, automation is limited to specific, single events. As you convert more of your cloud maintenance and security tasks into code that can be run from scripts, you can automate many tasks to work together in a complex and lengthy workflow, which is called [**orchestration**](javascript://). A robust and well-designed cloud deployment is almost fully orchestrated to minimize hands-on time required by cloud admins and to reduce the chances of human error when changes need to be made.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 7-3dConnectivity and Security

While cloud offers many significant advantages and opportunities for expansion, there are some drawbacks worth considering. Dependence on the Internet means dependence on your network’s connection to the ISP and reliance on other third parties as well. Potential risks to your cloud’s availability and security include the following:

* ISP outages
* ISP-imposed bandwidth limitations in response to increased demand on its network
* Cloud provider outages
* Failure of the cloud provider’s backup and security systems
* Misconfiguration that exposes one client’s data to another client
* Unauthorized access to data by cloud provider employees or by illegitimate users
* Breaches of confidentiality agreements when data is stored online
* Failure to properly comply with data security regulations (such as for healthcare, financial, or government entities)
* Questions over ownership of intellectual property stored in the cloud (for example, photos or comments made on social media websites, or files saved in online storage accounts)
* Questions over data maintenance if a payment is not made on time
* Risks to the network, proprietary data, or customer information caused by BYOC (bring your own cloud) services on users’ personal devices
* Reduced consumer confidence, fines, lawsuits, and possibly criminal charges when cloud breaches occur

One way to reduce the inherent risks of cloud computing is to use encryption. Another way is to carefully choose the method by which your network connects to your cloud resources. Business requirements, risk management, and cost all factor into this decision. Cloud providers will often offer attractive SLAs (service-level agreements) based upon their own technology’s availability. However, the WAN connection that links their resources with your network is just as important. To this end, organizations generally have an array of options:

* **Internet**—Provides the simplest and cheapest option, but with high and unpredictable latency as well as significant security concerns.
* **VPN (virtual private network)**—Relies on the same VPN technologies used to connect on-premises networks with branch offices and remote workers.
* **Remote access connections**—Uses tunneling or terminal emulation technologies to increase security, including SSH and RDP.
* **Leased line**—Relies on private WAN options to reserve a dedicated amount of bandwidth between the cloud provider and the customer’s premises. Depending on the respective locations of provider and customer, this might require the cooperation of multiple ISPs to reach the cloud provider’s servers. Hybrid pay-per-use models are available where the customer reserves a portion of anticipated bandwidth needs, and then is invoiced for additional bandwidth used during the pay period. Works in conjunction with a private or dedicated direct connection.
* **Private or dedicated direct connection**—Maximizes predictability and minimizes latency, and of course comes with a high price tag. Some of the larger cloud service providers maintain multiple **[PoPs (Points of Presence)](javascript://)** around the world. This means the provider rents space at a data center facility, called a [**colocation facility**](javascript://), that is shared by a variety of providers. In many cases, ISPs can provide dedicated access from a customer’s premises to a cloud provider’s PoP. This is more cost effective when an organization subscribes to multiple cloud providers who all use the same colocation. Amazon’s Direct Connect and Microsoft’s Azure ExpressRoute both offer dedicated connection services.

As you can see, the cloud offers some significant advantages over conventional, on-prem networks while also offering the ability to integrate with on-prem resources in a hybrid deployment. However, the cloud presents many concerns that will be familiar to those who work with physical networks, such as performance and availability. CSPs have integrated many cloud services that address performance concerns. For example, in a project at the end of this module, you’ll practice working with AWS’s CloudWatch service to set up an alarm to track expenses in your account. CloudWatch can be used to monitor performance metrics for cloud resources, such as CPU (central processing unit) and memory usage in a cloud VM, and alert admins if a resource encounters problems. Other techniques can also be used to monitor and manage cloud resource availability. Many of the same availability principles for the on-prem network can be applied to the cloud environment. Let’s look at what these principles are and examine strategies for maximizing network availability.

**Remember This…**

* Identify the defining characteristics of cloud services.
* Compare the primary cloud service models: IaaS, PaaS, and SaaS.
* Describe popular cloud deployment models, including public, private, community, multicloud, and hybrid.
* Explain the benefits of IaC.
* Compare automation and orchestration.

**Self-Check**

1. Which cloud characteristic ensures you can manage cloud resources from an iPad?

Answer

* 1. Rapid elasticity
  2. Resource pooling
  3. Multitenancy
  4. Broad network access

1. When you set private IP address ranges for servers in your cloud, what service model are you using?

Answer

* 1. IaaS
  2. SaaS
  3. DaaS
  4. PaaS

**You’re Ready**

You’re now ready to complete [Project 7-1: Create a CloudWatch Alarm in AWS](javascript://), or you can wait until you’ve finished reading this module.

**You’re Ready**

You’re now ready to complete [Project 7-2: Create an EC2 Instance in AWS](javascript://), or you can wait until you’ve finished reading this module.

**You’re Ready**

You’re now ready to complete [Project 7-3: Remote into a Cloud VM Instance in AWS](javascript://), or you can wait until you’ve finished reading this module.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# 7-4Network Availability



### Certification

* 1.4

Given a scenario, configure a subnet and use appropriate IP addressing schemes.

* 2.1

Compare and contrast various devices, their features, and their appropriate placement on the network.

* 2.3

Given a scenario, configure and deploy common Ethernet switching features.

* 3.1

Given a scenario, use the appropriate statistics and sensors to ensure network availability.

* 3.3

Explain high availability and disaster recovery concepts and summarize which is the best solution.

Average reading time: 24 minutes

In the world of networking, the term [**availability**](javascript://) refers to how consistently and reliably a connection, system, or other network resource can be accessed by authorized personnel. It’s often expressed as a percentage, such as 98% or 99.5%. The term [**HA (high availability)**](javascript://) refers to a system that functions reliably nearly all the time. For example, a server that allows staff to log on and use its programs and data 99.999 percent of the time is considered highly available, whereas one that is functional only 99.9 percent of the time is significantly less available. In fact, the number of 9s in a system’s availability rating is sometimes referred to colloquially as “four 9s” (99.99 percent) or “three 9s” (99.9 percent) availability. You might hear a network manager use the term in a statement such as, “We’re a four 9s shop.” This could be an impressive track record for a small ISP or a school’s LMS (learning management system). For a hospital network, however, where lives are at stake, four nines likely wouldn’t be enough.

**Note 7-8**

Various cloud services and ISPs offer three nines, four nines, five nines, or better availability, depending on what’s defined in their SLAs (service-level agreements). When shopping for cloud services, examine the SLA carefully so you’ll know what aspects of a service are guaranteed available.

However, be aware there’s a difference between availability (the ability to access a resource) and [**durability**](javascript://) (the resource’s ongoing existence). For example, AWS lists its storage service, S3, at 99.999999999% durability (that’s 11 nines!), but S3’s availability is 99.99% for its Standard storage class. Why the discrepancy?

That 11 nines durability means you could store 10,000,000 objects in S3 and expect to lose one of those objects every 10,000 years on average (okay, not bad). This is because S3 stores each object on multiple devices in multiple, physical data centers. The four nines availability means that, each year on average, there should only be 52.6 minutes when you can’t get to your objects in S3—this is also pretty good, considering you can relax in knowing that your stored data isn’t lost during that 52 minutes, even if you can’t get to it for a bit.

Similar terms include [**reliability**](javascript://), which refers to how well a resource functions without errors, and [**resiliency**](javascript://), which refers to a resource’s ability to recover from errors even if it becomes unavailable during the outage.

One way to consider availability is by measuring a system or network’s [**uptime**](javascript://), which is the duration or percentage of time it functions normally between failures. As shown in [Table 7-1](javascript://), a system that experiences 99.999 percent uptime is unavailable, on average, only 5 minutes and 15 seconds per year.

**Table 7-1**

### Availability and Downtime Equivalents

| **Availability** | **Downtime per day** | **Downtime per month** | **Downtime per year** |
| --- | --- | --- | --- |
| 99% | 14 minutes, 23 seconds | 7 hours, 18 minutes, 17 seconds | 87 hours, 39 minutes, 29 seconds |
| 99.9% | 1 minute, 26 seconds | 43 minutes, 49 seconds | 8 hours, 45 minutes, 56 seconds |
| 99.99% | 8 seconds | 4 minutes, 22 seconds | 52 minutes, 35 seconds |
| 99.999% | 0.4 seconds | 26 seconds | 5 minutes, 15 seconds |

On a computer running Linux or UNIX, you can view the length of time your system has been running with the command uptime. On a Windows 10 system, uptime information is found in Task Manager.

**Applying Concepts 7-1**

### Windows Task Manager

Windows 10 provides uptime data, along with a great deal of additional performance information, in Task Manager. Complete the following steps to view this information on a Windows 10 computer:

1. 1

Right-click **Start** and click **Task Manager**.

1. 2

On the **Performance** tab, examine the CPU and Memory utilization statistics. What is the current uptime?

1. 3

Click **Open Resource Monitor** to view additional performance data and graphs.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 7-4aFault Tolerance

A key factor in maintaining the availability of network resources is [**fault tolerance**](javascript://), or the capacity of a system to continue performing despite an unexpected hardware or software malfunction. The key to fault tolerance in network design is supplying multiple paths that data can use to travel from any one point to another. Therefore, if one connection or component fails, data can be rerouted over an alternate path.

To better understand the issues related to fault tolerance, it helps to know the difference between failures and faults as they apply to networks, as described next:

* [**Failure**](javascript://)—A deviation from a specified level of system performance for a given period of time. In other words, a failure occurs when something doesn’t work as promised or as planned. For example, if your car breaks down on the highway, you can consider the breakdown to be a failure.
* [**Fault**](javascript://)—A malfunction of one component of a system. A fault can result in a failure. For example, the fault that caused your car to break down might be a leaking water pump. The goal of fault-tolerant systems is to prevent faults from progressing to failures.

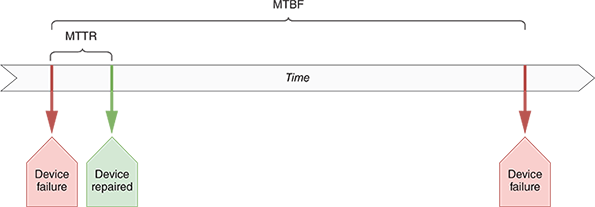
Fault tolerance can be realized in varying degrees; the optimal level of fault tolerance for a system depends on how critical its services and files are to productivity. At the highest level of fault tolerance, a system remains unaffected by even the most drastic problem, such as a regional power outage. In this case, a backup power source, such as an electrical generator, is necessary to ensure fault tolerance. However, less dramatic faults, such as a malfunctioning NIC on a router, can still cause network outages, and you should guard against them.

### Redundancy

Devices on a network typically have a calculated [**MTBF (mean time between failures)**](javascript://). This is the average amount of time that will pass for devices exactly like this one before the next failure is expected to occur. While any single device might experience a failure much sooner or later, vendors and technicians budget for repairs or replacement of devices based on the advertised MTBF. Once a device fails, there is an average amount of time required to repair the device. This is called [**MTTR (mean time to repair)**](javascript://), and this cost must also be considered. [Figure 7-34](javascript://) shows how these concepts are related.

**Figure 7-34**

Every device eventually fails, it’s just a question of when



Enlarge Image

MTBF, MTTR, and related concepts can all apply to services or systems as well. An ISP service might have an advertised MTBF and MTTR that are defined in the SLA. For example, any time your WAN connection goes down, the ISP might guarantee that it will be back up within two to four hours. Of course, these numbers vary according to provider, connection type, and subscription level, and should be taken into account when selecting WAN service options. You’ll learn more about WAN technologies later.

To help protect against faults and failures, networks are often designed with two or more of the same item, service, or connection filling the same role on the network. If one part, service, or connection fails, the other takes over. Recall this is called redundancy and refers to an implementation in which more than one component is installed and ready to use for storing, processing, or transporting data. Redundancy is intended to eliminate single points of failure. To maintain high availability, you should ensure that critical network elements, such as your connection to the Internet or your file server’s hard disk, are redundant. Some types of redundancy—for example, redundant sources of electrical power for a building—require large investments, so your organization should weigh the risks of losing connectivity or data against the cost of adding duplicate (or triplicate) components.

As you can see, the main disadvantage of redundancy is its cost. Redundancy is like a homeowner’s insurance policy: You might never need to use it, but if you don’t get it, the cost when you do need it can be much higher than your premiums. Redundant ISP services, for example, can be costly. Compared to the cost to a business of not having Internet access if a trunk line is severed, however, the additional WAN interface might make sense. As a rule, you should invest in connection redundancies for any connection that is absolutely necessary.

Even when dedicated links and VPN connections remain sound, a faulty device or interface in the data path can affect service for a user, a whole segment, or the whole network. To understand how to increase the fault tolerance of a connection from end to end, consider a typical link to the Internet. [Figure 7-35](javascript://) provides a representation of this arrangement.

**Figure 7-35**

Single Internet link connectivity

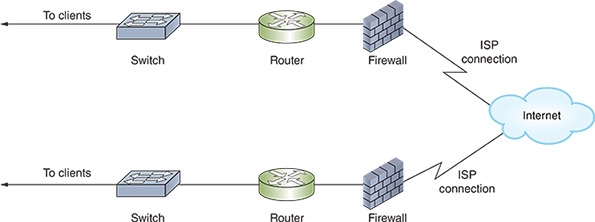


Enlarge Image

Notice the many single points of failure in the arrangement depicted in [Figure 7-35](javascript://). In addition to the ISP link failing—for example, if a backhoe accidentally cuts a cable during road construction—any of the critical nodes in the diagram (firewall, router, or switch) could suffer a fault or failure and impair connectivity or performance. [Figure 7-36](javascript://) illustrates a network design that ensures full redundancy for all the components linking two locations to an ISP.

**Figure 7-36**

Fully redundant ISP connectivity



Enlarge Image

To achieve the utmost fault tolerance, each critical device requires redundant NICs, power supplies, cooling fans, and processors, all of which should, ideally, be able to immediately assume the duties of an identical component, a capability known as [**automatic failover**](javascript://). If one NIC in a router fails, for example, automatic failover ensures that the router’s other NIC can automatically handle the first NIC’s responsibilities.

In cases where failover-capable components are impractical, you can provide some level of fault tolerance by using hot-swappable parts. Recall that hot-swappable refers to identical components that can be changed (or swapped) while a machine is still running (hot). There are two approaches to this:

* **Hot spare**—A duplicate component that is already installed in a device and can immediately assume the original component’s functions in case that component fails.
* **Cold spare**—A duplicate component that is not installed but that can be installed in case of a failure. Relying on a cold spare results in an interruption of service.

When you purchase switches or routers to support critical links, look for those that contain failover capable or hot-swappable components. As with other redundancy provisions, these features add to the cost of your device.

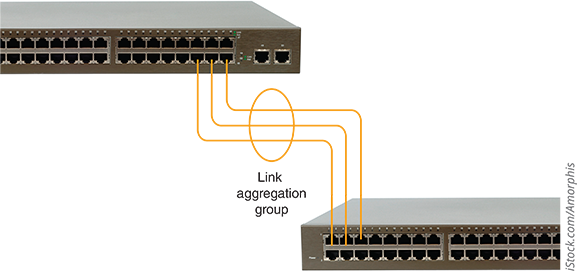
### Redundant Links

Besides using redundant devices, you can also use redundant connections, or links, between devices. [**Link aggregation**](javascript://) is the seamless combination of multiple network interfaces or ports to act as one logical interface, and it can help solve problems like network bottlenecks. This implementation is also known by a variety of other terms, such as [**port aggregation**](javascript://) on Cisco devices, [**NIC teaming**](javascript://) on Windows devices, and a variety of others such as bonding, bundling, or Cisco’s EtherChannel. Regardless of the terms used, link aggregation causes two or more NICs to work in tandem handling traffic between two or more devices (usually switches and servers). All the physical links involved in creating the one logical link are called a LAG (link aggregation group), bundle, or team, as shown in [Figure 7-37](javascript://). This configuration allows for three major advantages:

* Increased total throughput
* Automatic failover between the aggregated NICs
* [**Load balancing**](javascript://), which is a distribution of traffic over multiple components or links to optimize performance and fault tolerance

**Figure 7-37**

Two switches treat these three physical links as one logical link

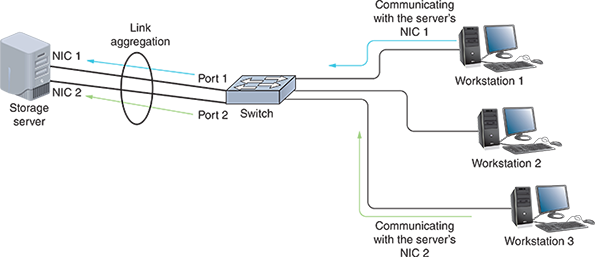


[iStock.com/Amorphis](http://istock.com/Amorphis" \t "_blank)

Link aggregation isn’t about speed of network traffic so much as bandwidth, or total potential to handle more network traffic at one time. Because packets and sessions generally aren’t separated between the duplicate connections, the benefits of link aggregation are primarily noticed on busy networks. For example, if a single session is all handled on only one of the aggregated connections, that session doesn’t reach its destination any faster. However, if two sessions are being transmitted at the same time, one session can traverse one of the aggregated links, and the other session can traverse the other link at the same time. Neither session must wait on the other (see [Figure 7-38](javascript://)).

**Figure 7-38**

Link aggregation allows two workstations to communicate with a server at the same time



Enlarge Image

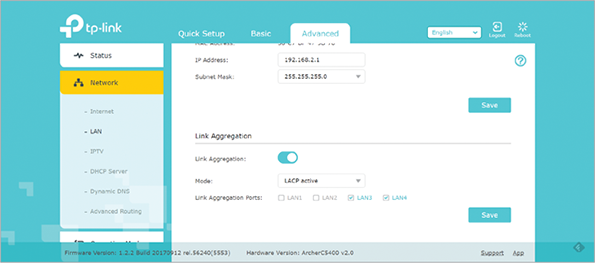
For multiple NICs or ports to use link aggregation, they must be properly configured in each device’s operating system. For example, all involved interfaces must be configured for full duplex, and have the same speed, VLAN, and MTU settings. Many manufacturers now use [**LACP (Link Aggregation Control Protocol)**](javascript://), which was initially defined by IEEE’s 802.3ad standard and currently defined by the 802.1AX standard (notice the change in working group from 802.3 to 802.1). LACP dynamically coordinates communications between hosts on aggregated connections, kind of like what DHCP does for IP addressing. Most of these devices offer similar configuration options, such as the following:

* **Static configuration**—Both hosts are manually configured to handle the division of labor between the redundant links according to particular rules without the ability to compensate for errors.
* **Passive mode**—The port passively listens for LACP-defined link aggregation requests, but it will not initiate the request.
* **Active mode**—The port is set to automatically and actively negotiate for link aggregation using LACP. This allows for fault tolerance should one or more links fail, as LACP will automatically reconfigure active links to compensate. In reality, this is the most common configuration for all ports involved in link aggregation, and it provides the most protection against link misconfigurations or failures.

[Figure 7-39](javascript://) shows the link aggregation options on a SOHO router. Here, you can aggregate two or more of the router’s four LAN ports, depending on which ones are currently connected to another device.

**Figure 7-39**

Aggregate LAN3 and LAN4 to a network server



Enlarge Image

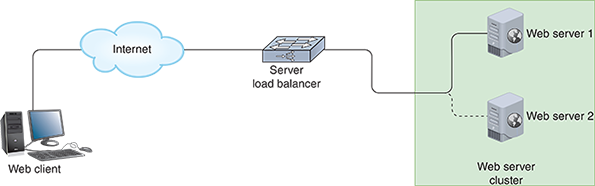
Source: TP-Link Technologies Co., Ltd.

More sophisticated load balancing for all types of servers can be achieved using a load balancer which, as you’ve already read, is a device dedicated to the task of distributing traffic intelligently among multiple computers. It can determine which among a pool of servers is experiencing the most traffic before forwarding the request to a server with lower utilization. This server pool might be configured as a cluster. [**Clustering**](javascript://) refers to the technique of grouping redundant resources such as servers so they appear as a single device to the rest of the network. Clustering can be configured with groups of servers, routers, or applications. Although it usually accompanies load balancing, it doesn’t have to.

Let’s look at an example of how clustering and load balancing might work. Suppose you have two web servers that, together, host a single website (see [Figure 7-40](javascript://)). To access the website, web clients direct requests to a single [**VIP (virtual IP address)**](javascript://) that represents the entire cluster. To the client, the cluster looks like a single web server. On the back end, though, a load balancer directs traffic evenly between the web servers, and both servers have access to all the data needed to respond to any web page requests from clients. The clients, however, are not aware that two physical machines are at work. As far as a client is concerned, it’s talking with a single server.

**Figure 7-40**

Two web servers work together in a cluster to host a single website



Enlarge Image

In this way, a popular website can respond more quickly to the high number of visitors interacting with the site at any one time. Each web server also serves as a backup to the other one. Should one server fail, the other can take over the full load until the malfunctioning server can be fixed or replaced. In the cloud or other virtualized environments, web server VMs or containers can be configured to automatically scale up to handle higher volumes of traffic. These autoscaling groups typically run a minimum number of servers at all times and then add more servers to the cluster only as needed. When the traffic subsides, the extra servers are removed so the website owner is not charged for unneeded server resources.

**Applying Concepts 7-2**

### Add a Virtual IP Address to Windows 10

You can add multiple, virtual IP addresses to a Windows computer. This is not the same use case as previously described with load balancing. Instead, it might be used to assign a different IP address to multiple instances of the same service running on a single machine. For example, you might have three different websites running on one machine, and each website would need its own IP address. To see how to add multiple IP addresses to one network adapter on a Windows computer, complete the following steps:

1. 1

Use PowerShell or Command Prompt to determine your computer’s current IP address, subnet mask, default gateway, and DNS servers. What command did you use? What information did you find?

1. 2

Open the **Network and Sharing Center**. Click **Change adapter settings**.

1. 3

Open the properties box for the active network connection. Open the properties box for TCP/IPv4.

1. 4

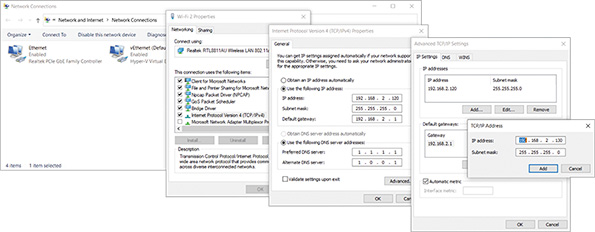
Configure a static IP address using the information you gathered in [Step 1](javascript://). Do not click OK. Instead, click **Advanced**.

1. 5

On the IP Settings tab, under IP addresses, click **Add**, as shown in [Figure 7-41](javascript://). Enter a second IP address in the same subnet as the original IP address and using the same default gateway. For example, if your first IP address is 192.168.2.123, you might add a second IP address at 192.168.2.124. Click **Add**.

**Figure 7-41**

Add multiple, virtual IP addresses to a single, physical network connection



Enlarge Image

1. 6

Click **OK** two more times, and close all open windows except PowerShell or Command Prompt. Run **ipconfig** again to determine your current IP addresses, subnet mask, and default gateway. What information is reported this time?

1. 7

Ping your VIP (the second IP address you added). Was it successful?

1. 8

If you have another computer on this subnet, ping each of the first computer’s two IP addresses from the other computer. Are the pings successful? Why do you think this is?

1. 9

On the second computer, run the command **arp -a**. What is significant about the MAC addresses listed for the first computer’s two IP addresses?

1. 10

What steps do you need to take to return your computer to the IP configuration it had when you started? If desired, do this now.

In some cases, you might have a set of IP addresses to share among multiple hosts. For example, if you have multiple routers that support multiple interfaces, and you want to interlace those routers as a fault-tolerant cluster, you would have a list of several IP addresses pointing to the cluster as a group. This is accomplished with [**CARP (Common Address Redundancy Protocol)**](javascript://), which allows a pool of computers or interfaces to share one or more IP addresses. This pool is known as a group of redundancy or redundancy group. When using CARP, one device, acting as the group master, receives requests for an IP address, then parcels out the requests to one of several devices in the group.

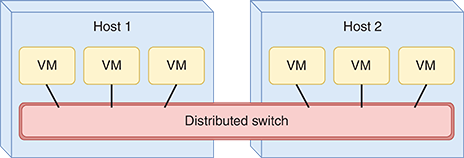
### Exam Tip

CARP is a free alternative to [**VRRP (Virtual Router Redundancy Protocol)**](javascript://), or Cisco’s propriety version called [**HSRP (Hot Standby Routing Protocol)**](javascript://). Although VRRP and HSRP function somewhat differently than CARP and are used solely for routers, the general idea is the same.

Clustering servers is used in many different ways to pool resources on a network and provide redundancy for fault tolerance. Another scenario is when pooling servers that host VMs. In a server cluster, the VMs are configured with varying amounts of redundancy to provide fault tolerance if one server fails. This can also allow for—and, in fact, necessitates—more efficient networking solutions between the VMs. Recall that VMs connect to a network via a vSwitch that exists in the host’s hypervisor. In a server cluster, a single, distributed vSwitch can service VMs across multiple hosts, as illustrated in [Figure 7-42](javascript://). This is called [**distributed switching**](javascript://). It centralizes control of the VMs, simplifies network operations, and minimizes the chances for configuration errors. To do this, an agent is installed on each physical host and is then controlled by a supervisor module in the distributed switch. Examples of distributed switch products are VMware’s VDS (vSphere Distributed Switch) that is native to its vSphere platform and a variety of third-party products, including Cisco’s Nexus 1000v series.

**Figure 7-42**

Distributed switching centralizes management of VM network connections



**Remember This…**

* Compare MTTR and MTBF.
* Explain key redundancy concepts, including load balancing, NIC teaming, link aggregation, and port aggregation.
* Configure a VIP.

**Self-Check**

1. Which is longer for a reliable device?

Answer

* 1. MTBF
  2. MTTR

1. Which protocol balances traffic across multiple links?

Answer

* 1. CARP
  2. LACP
  3. VIP
  4. VRRP

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# Module Review

## 7-5a**Module Summary**

### Physical Architecture

* The overall network design—the devices involved, how they’re configured, the services implemented to support the network, and the way devices are connected to the network—is called a network’s architecture.
* An unmanaged switch provides plug-and-play simplicity with minimal configuration options and has no IP address assigned to it. Unmanaged switches are not very expensive, but their capabilities are limited. Managed switches, on the other hand, can be configured via a command-line interface or a web-based management GUI, and sometimes can be configured in groups.
* Redundancy allows data the option of traveling through more than one switch toward its destination, and it makes your network less vulnerable to hardware malfunctions. A potential problem with redundancy has to do with traffic loops. STP (Spanning Tree Protocol) and similar technologies eliminate or otherwise control switching loops to greatly reduce the potential for a broadcast storm.
* The Cisco command (which is also used on Arista devices) to secure switch access ports is switchport port-security (or just port-security on Huawei switches). This is essentially a MAC filtering function that also protects against MAC flooding, which makes it a type of flood guard. Acceptable MAC addresses are stored in a MAC address table. Once the MAC address table is full, a security violation occurs if another device attempts to connect to the port.
* Cisco and other manufacturers have developed a hierarchical design for switches on a network. The three tiers are the access layer (edge layer), the distribution layer (aggregation layer), and the core layer. Hosts connect to the access layer, and other network blocks connect to the distribution layer.
* A spine-and-leaf architecture (also spine-leaf or leaf-spine) collapses the core and distribution layers into one layer called the spine. Spine switches on the backbone connect in a mesh topology with all leaf switches (but not with each other), and leaf switches connect with servers and other host devices.
* SDN (software-defined networking) is a centralized approach to networking that removes most of the decision-making power from network devices and instead handles that responsibility at a software level. This SDN controller integrates configuration and management control of all network devices, both physical and virtual, into one cohesive system that is overseen by the network administrator through a single dashboard. To do this, SDN separates into pieces all the functions of a system into three layers, or planes: infrastructure plane (data plane), control plane, and application plane. While not a typical layer for network communication, the management plane could be considered a part of the control plane. It allows network administrators to remotely manage network devices, monitor those devices, and analyze data collected about the devices.
* A SAN (storage area network) is a distinct network of storage devices that communicate directly with each other and with other portions of the network. Essentially, a SAN abstracts storage services from compute services and then provides high-speed network services to connect them. The network connections between servers and SAN devices must support extremely high data throughput. To maximize throughput, SANs often rely on one of these networking technologies: FC (Fibre Channel), FCoE (Fibre Channel over Ethernet), iSCSI (Internet SCSI), and IB (InfiniBand).

### Virtual Architecture

* Virtualization is a virtual, or logical, version of something rather than the actual, or physical, version. A hypervisor creates and manages a VM, and it allocates hardware resources for the host and any of its guest VMs. Together, all the virtual devices on a single computer share the same CPU, hard disks, memory, and physical network interfaces. A type 1 hypervisor installs on a computer before any OS and is, therefore, erroneously called a “bare-metal” hypervisor. A type 2 hypervisor installs in a host OS as an application and is called a hosted hypervisor.
* The way a vNIC is configured determines whether the VM is joined to a virtual network or attempts to join the physical LAN that the host machine is connected to. These various configurations are called networking modes, the most common of which are bridged, NAT, and host-only.
* Virtualization offers several advantages, including efficient use of resources; cost and energy savings; fault and threat isolation; and simple backups, recovery, and replication. Potential disadvantages to creating multiple guests on a single host machine include compromised performance, increased complexity, increased licensing costs, and a single point of failure.
* Merging physical and virtual network architecture is called NFV (Network Functions Virtualization). NFV provides flexible, cost-saving options for many types of network devices, including virtual servers, data storage, load balancers, and firewalls.

### Cloud Architecture

* Cloud computing refers to the flexible provision of data storage, applications, or services to clients over the Internet. All cloud services have the following features in common, according to NIST (National Institute of Standards and Technology): on-demand self-service, broad network access, resource pooling, measured service, and rapid elasticity.
* Cloud service models incrementally increase the amount of management responsibilities outsourced to cloud vendors. Common cloud service models include IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service), and XaaS (Anything as a Service).
* Cloud services are delivered in a variety of deployment models, depending on who manages the underlying hardware and who has access to it. The main deployment models you are likely to encounter are public cloud, private cloud, community cloud, hybrid cloud, and multicloud.
* The process of using text-based commands in a computer-readable configuration file to create and manage cloud resources is called IaC (infrastructure as code). A programmed, computer-generated response to a specific event is referred to as automation. However, automation is limited to specific, single events. Many cloud management tasks can be automated to work together in a complex and lengthy workflow, which is called orchestration.
* The WAN connection that links cloud resources with the local network is critical. Connectivity options include Internet, VPN, remote access connections, leased line, and private or dedicated direct connection.

### Network Availability

* In the world of networking, the term availability refers to how consistently and reliably a connection, system, or other network resource can be accessed by authorized personnel. It’s often expressed as a percentage, such as 98% or 99.5%. There’s a difference between availability (the ability to access a resource) and durability (the resource’s ongoing existence). Similar terms include reliability, which refers to how well a resource functions without errors, and resiliency, which refers to a resource’s ability to recover from errors even if it becomes unavailable during the outage.
* A key factor in maintaining the availability of network resources is fault tolerance, or the capacity of a system to continue performing despite an unexpected hardware or software malfunction. The key to fault tolerance in network design is supplying multiple paths that data can use to travel from any one point to another.
* Devices on a network typically have a calculated MTBF (mean time between failures). This is the average amount of time that will pass for devices exactly like this one before the next failure is expected to occur. Once a device fails, there is an average amount of time required to repair the device. This is called MTTR (mean time to repair), and this cost must also be considered.
* Link aggregation is the seamless combination of multiple network interfaces or ports to act as one logical interface, and it can help solve problems like network bottlenecks. This implementation is also known by a variety of other terms, such as port aggregation on Cisco devices, NIC teaming on Windows devices, and a variety of others such as bonding, bundling, or Cisco’s EtherChannel. Regardless of the terms used, link aggregation causes two or more NICs to work in tandem handling traffic between two or more devices (usually switches and servers).

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# Module Review

## 7-5b**Key Terms**

* [**access layer**](javascript://)
* [**aggregation layer**](javascript://)
* [**application plane**](javascript://)
* [**architecture**](javascript://)
* [**automatic failover**](javascript://)
* [**automation**](javascript://)
* [**availability**](javascript://)
* [**BPDU (Bridge Protocol Data Unit)**](javascript://)
* [**branch office**](javascript://)
* [**bridged mode**](javascript://)
* [**broadcast storm**](javascript://)
* [**CARP (Common Address Redundancy Protocol)**](javascript://)
* [**cloud computing**](javascript://)
* [**cloud service model**](javascript://)
* [**clustering**](javascript://)
* [**colocation facility**](javascript://)
* [**community cloud**](javascript://)
* [**container**](javascript://)
* [**control plane**](javascript://)
* [**core layer**](javascript://)
* [**CSP (cloud services provider)**](javascript://)
* [**DaaS (Desktop as a Service)**](javascript://)
* [**data plane**](javascript://)
* [**defense in depth**](javascript://)
* [**distributed switching**](javascript://)
* [**distribution layer**](javascript://)
* [**durability**](javascript://)
* [**east-west traffic**](javascript://)
* [**edge layer**](javascript://)
* [**elasticity**](javascript://)
* [**EoR (end of row) switching**](javascript://)
* [**failure**](javascript://)
* [**fault**](javascript://)
* [**fault tolerance**](javascript://)
* [**FC (Fibre Channel)**](javascript://)
* [**FCoE (Fibre Channel over Ethernet)**](javascript://)
* [**guest**](javascript://)
* [**HA (high availability)**](javascript://)
* **host**
* [**host-only mode**](javascript://)
* [**HSRP (Hot Standby Routing Protocol)**](javascript://)
* [**hybrid cloud**](javascript://)
* [**hypervisor**](javascript://)
* [**IaaS (Infrastructure as a Service)**](javascript://)
* [**IaC (infrastructure as code)**](javascript://)
* [**IB (InfiniBand)**](javascript://)
* [**infrastructure plane**](javascript://)
* [**iSCSI (Internet SCSI)**](javascript://)
* [**LACP (Link Aggregation Control Protocol)**](javascript://)
* [**layer 3 switch**](javascript://)
* [**layer 4 switch**](javascript://)
* [**least cost path**](javascript://)
* [**link aggregation**](javascript://)
* [**load balancer**](javascript://)
* [**load balancing**](javascript://)
* [**managed switch**](javascript://)
* [**management plane**](javascript://)
* [**MTBF (mean time between failures)**](javascript://)
* [**MTTR (mean time to repair)**](javascript://)
* [**multicloud**](javascript://)
* [**multipathing**](javascript://)
* [**multitenancy**](javascript://)
* [**NAT mode**](javascript://)
* [**NFV (Network Functions Virtualization)**](javascript://)
* [**NIC teaming**](javascript://)
* [**north-south traffic**](javascript://)
* [**on-premises data center**](javascript://)
* [**orchestration**](javascript://)
* [**PaaS (Platform as a Service)**](javascript://)
* [**PoP (Point of Presence)**](javascript://)
* [**port aggregation**](javascript://)
* [**private cloud**](javascript://)
* [**public cloud**](javascript://)
* [**redundancy**](javascript://)
* [**reliability**](javascript://)
* [**resiliency**](javascript://)
* [**root bridge**](javascript://)
* [**root port**](javascript://)
* [**SaaS (Software as a Service)**](javascript://)
* [**SDN (software-defined networking)**](javascript://)
* [**SDN controller**](javascript://)
* [**serverless compute**](javascript://)
* [**spine-and-leaf architecture**](javascript://)
* [**STP (Spanning Tree Protocol)**](javascript://)
* [**three-tiered architecture**](javascript://)
* [**ToR (top of rack) switching**](javascript://)
* [**type 1 hypervisor**](javascript://)
* [**type 2 hypervisor**](javascript://)
* [**unmanaged switch**](javascript://)
* [**uptime**](javascript://)
* [**VIP (virtual IP address)**](javascript://)
* [**virtualization**](javascript://)
* [**vNIC (virtual NIC)**](javascript://)
* [**VRRP (Virtual Router Redundancy Protocol)**](javascript://)
* [**vSwitch (virtual switch)**](javascript://)
* [**XaaS (Anything as a Service)**](javascript://)

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# Module Review

## 7-5c**Review Questions**

1. What software allows you to define VMs and manage resource allocation and sharing among VMs on a host computer?
   1. Hypervisor
   2. NFV (Network Functions Virtualization)
   3. SDN (software-defined networking)
   4. Terminal emulation
2. What virtual, logically defined device operates primarily at the data link layer to pass frames between nodes?
   1. Virtual firewall
   2. Virtual switch
   3. Virtual router
   4. Virtual load balancer
3. Which device can manage traffic to multiple servers in a cluster so all servers equally share the traffic?
   1. Router
   2. Firewall
   3. Switch
   4. Load balancer
4. With which network connection type does the VM obtain IP addressing information from its host?
   1. Bridged mode
   2. Managed mode
   3. NAT mode
   4. Isolation mode
5. Which type of switch connects all devices in a rack to the rest of the network?
   1. ToR switch
   2. Spine switch
   3. EoR switch
   4. Core switch
6. Which cloud service model gives software developers access to multiple platforms for testing code?
   1. IaaS
   2. PaaS
   3. SaaS
   4. XaaS
7. When shopping for a new router, what does the MTBF tell you?
   1. How long until that device fails
   2. How much it will cost to repair that device
   3. How long devices like this one will last on average until the next failure
   4. How long it will usually take to repair that device
8. What information does the switchport port-security command use to restrict access to a switch’s interface?
   1. MAC address
   2. Port number
   3. IP address
   4. Broadcast address
9. Which of the following features of a network connection between a switch and server is not improved by link aggregation?
   1. Bandwidth
   2. Fault tolerance
   3. Speed
   4. Availability
10. Which cloud management technique executes a series of tasks in a workflow?
    1. Automation
    2. IaC
    3. SLA
    4. Orchestration
11. List two advantages to using virtualization on a network.
12. List available options for connecting to cloud resources.
13. How does a vNIC get a MAC address without manual intervention?
14. What type of adapters are required on servers in an FCoE storage network?
15. What are two use cases for a VIP (virtual IP address)?
16. Which type of hypervisor is installed directly on top of the server’s firmware?
17. Only one  exists on a network using STP.
18. What protocol is most often used to bond ports between a switch and a busy server?
19. How is licensing an important concern when using virtualization?
20. What kind of device can be used to configure and manage physical and virtual networking devices across the network?

Go to pg.

[**help**](javascript://)