Network Devices

5.4.1

Ethernet Switches

Earlier in this module, you explored both switching and routing functions in the network layer. In this topic, you will explore in more detail the networking devices that perform the switching and routing functions.

A key concept in Ethernet switching is the broadcast domain. A broadcast domain is a logical division in which all devices in a network can reach each other by broadcast at the data link layer. Broadcast frames must be forwarded by the switch on all its ports except the port that received the broadcast frame. By default, every port on a switch belongs to the same broadcast domain. A Layer 3 device, such as a router is needed to terminate the Layer 2 broadcast domain. As discussed previously, VLANs correspond to a unique broadcast domain.

In legacy shared Ethernet, a device connected to a port on a hub can either transmit or receive data at a given time. It cannot transmit and receive data at the same time. This is referred to as half-duplex transmission. Half-duplex communication is similar to communication with walkie-talkies in which only one person can talk at a time. In half-duplex environments, if two devices do transmit at the same time, there is a collision. The area of the network in which collisions can occur is called a collision domain.

One of the main features of Ethernet switches over legacy Ethernet hubs is that they provide full-duplex communications, which eliminates collision domains. Ethernet switches can simultaneously transmit and receive data. This mode is called full-duplex. Full-duplex communication is similar to the telephone communication, in which each person can talk and hear what the other person says simultaneously.

Switches have the following functions:

- Operate at the network access layer of the TCP/IP model and the Layer 2 data link layer of the OSI model
- Filter or flood frames based on entries in the MAC address table
- Have a large number of high speed and full-duplex ports

The figure shows an example of switches with multiple high speed and full-duplex ports.

The switch dynamically learns which devices and their MAC addresses are connected to which switch ports. It builds the MAC address table and filters or floods frames based on that table. A MAC address table on a switch looks similar to the one below:

VLAN	MAC Address	Type	Ports
1	001b.10a0.2500	Dynamic	Gi0/1
1	001b.10ae.7d00	Dynamic	Gi0/2
10	0050.7966.6803	Dynamic	Gi0/3

The switching mode determines whether the switch begins forwarding the frame as soon as the switch has read the destination details in the packet header, or waits until the entire frame has been received and checked for errors, by calculating the cyclic redundancy check (CRC) value, before forwarding on the network. The switching mode is applicable to all packets being switched or routed through the hardware and can be saved persistently through reboots and restarts.

The switch operates in either of the following switching modes:

- **Cut-Through Switching Mode** - Switches operating in cut-through switching mode start forwarding the frame as soon as the switch has read the destination details in the frame header. A switch in cut-through mode forwards the data before it has completed receiving the entire frame. The switching speed in cut-through mode is faster than the switching speed in store-and-forward switching mode. Fragment free switching is a modified form of cut-through switching in which the switch only starts forwarding the frame after it has read the Type field. Fragment free switching provides better error checking than cut-through, with practically no increase in latency.
- **Store-and-Forward Switching Mode** - When store-and-forward switching is enabled, the switch checks each frame for errors before forwarding it. Each frame is stored until the entire frame has been received and checked. Because it waits to forward the frame until the entire frame has been received

and checked, the switching speed in store-and-forward switching mode is slower than the switching speed in cut-through switching mode.

These are some characteristics of LAN switches:

- **High port density** - Switches have a large number of ports, from 24 to 48 ports per switch in smaller devices, to hundreds of ports per switch chassis in larger modular switches. Switch ports usually operate at 100 Mbps, 1 Gbps, and 10 Gbps.
- **Large frame buffers** - Switches have the ability to store received frames when there may be congested ports on servers or other devices in the network.
- **Fast internal switching** - Switches have very fast internal switching. They are able to switch user traffic from the ingress port to the egress port extremely fast. Different methods are used to connect the ports which affects the overall performance of the switch including a fast internal bus, shared memory, or an integrated crossbar switch fabric.

5.4.2

Routers

While switches are used to connect devices on a LAN and exchange data frames, routers are needed to reach devices that are not on the same LAN. Routers use routing tables to route traffic between different networks. Routers are attached to different networks (or subnets) through their interfaces and have the ability to route the data traffic between them.

Routers have the following functions:

- They operate at the internet layer of TCP/IP model and Layer 3 network layer of the OSI model.
- They route packets between networks based on entries in the routing table.
- They have support for a large variety of network ports, including various LAN and WAN media ports which may be copper or fiber. The number of interfaces on routers is usually much smaller than switches but the variety of interfaces supported is greater. IP addresses are configured on the interfaces.

The figure shows a modular router with integrated switch ports.

Recall that the functions of a router are path determination and packet forwarding. There are three packet-forwarding mechanisms supported by routers:

- **Process switching** - When a packet arrives on an interface, it is forwarded to the control plane where the CPU matches the destination address with an entry in its routing table, and then determines the exit interface and forwards the packet. The router does this for every packet, even if the destination is the same for a stream of packets. This process-switching mechanism is very slow and is rarely implemented in modern networks. Contrast this with fast switching.
- **Fast switching** - Fast switching uses a fast-switching cache to store next-hop information. When a packet arrives on an interface, it is forwarded to the control plane where the CPU searches for a match in the fast-switching cache. If it is not there, it is process-switched and forwarded to the exit interface. The flow information for the packet is also stored in the fast-switching cache. If another packet going to the same destination arrives on an interface, the next-hop information in the cache is re-used without CPU intervention.
- **Cisco Express Forwarding (CEF)** - CEF is the most recent and default Cisco IOS packet-forwarding mechanism. Like fast switching, CEF builds a Forwarding Information Base (FIB), and an adjacency table. However, the table entries are not packet-triggered like fast switching but change-triggered, such as when something changes in the network topology. Therefore, when a network has converged, the FIB and adjacency tables contain all the information that a router would have to consider when forwarding a packet. Cisco Express Forwarding is the fastest forwarding mechanism and the default on Cisco routers and multilayer switches.

A common analogy used to describe these three different packet-forwarding mechanisms is as follows:

- Process switching solves a problem by doing math long hand, even if it is the identical problem that was just solved.
- Fast switching solves a problem by doing math long hand one time and remembering the answer for subsequent identical problems.
- CEF solves every possible problem ahead of time in a spreadsheet.

5.4.3

Firewalls

A firewall is a hardware or software system that prevents unauthorized access into or out of a network. Typically, firewalls are used to prevent unauthorized internet users from accessing internal networks. Therefore, all data leaving or entering the protected internal network must pass through the firewall to reach its destination, and any unauthorized data is blocked. The role of the firewall in any network is critical. Additional details on how firewalls interact with applications are presented in the Application Development and Security module of the course.

The figure shows an example of a hardware firewall.

Stateless Packet Filtering

The most basic (and the original) type of firewall is a stateless packet filtering firewall. You create static rules that permit or deny packets, based on packet header information. The firewall examines packets as they traverse the firewall, compares them to static rules, and permits or denies traffic accordingly. This stateless packet filtering can be based on several packet header fields, including the following:

- Source and/or destination IP address
- IP protocol ID
- Source and/or destination TCP or UDP Port number
- ICMP message type
- Fragmentation flags
- IP option settings

The static rules are fairly simple, but they do not work well for applications that dynamically use different sets of TCP and/or UDP port numbers. This is because they cannot track the state of TCP or UDP sessions as they transition from initial request, to fulfilling that request, and then the closing of the session. Also, these static rules are built using a restrictive approach. In other words, you write explicit rules to permit the specific traffic deemed acceptable, and deny everything else.

Static rules are transparent to end systems, which are not aware that they are communicating with a destination through a high-performance firewall. However, implementing static rules requires deep understanding of packet headers and application processes.

Stateful Packet Filtering

The stateful packet filtering firewall performs the same header inspection as the stateless packet filtering firewall but also keeps track of the connection state. This is a critical difference. To keep track of the state, these firewalls maintain a state table.

A typical simple configuration works as follows. Any sessions or traffic initiated by devices on trusted, inside networks are permitted through the firewall. This includes the TCP connection request for destination port 80. The firewall keeps track of this outbound request in its state table. The firewall understands that this is an initial request, and so an appropriate response from the server is allowed back in through the firewall. The firewall tracks the specific source port used and other key information about this request. This includes various IP and TCP flags and other header fields. This adds a certain amount of intelligence to the firewall.

It will allow only valid response packets that come from the specific server. The response packets must have all the appropriate source and destination IP addresses, ports, and flags set. The stateful packet filtering firewall understands standard TCP/IP packet flow including the coordinated change of information between inside and outside hosts that occurs during the life of the connection. The firewall allows untrusted outside servers to respond to inside host requests, but will not allow untrusted servers to initiate requests.

Of course, you can create exceptions to this basic policy. Your company might consider certain applications to be inappropriate during work hours. You might want to block inside users from initiating connections to those applications. However, with traditional stateful packet filtering, this capability is limited. These traditional firewalls are not fully application-aware.

Also, you might have a web server hosted on the premises of a corporation. Of course, you would like everyone in the world to access your web server and purchase your products or services. You can write rules that allow anyone on the untrusted internet to form appropriate inbound connections to the web server.

These stateful firewalls are more adept at handling Layer 3 and Layer 4 security than a stateless device. However, like stateless packet filters, they have little to no insight into what happens at OSI model Layers 5–7; they are “blind” to these layers.

Application Layer Packet Filtering

The most advanced type of firewall is the application layer firewall which can perform deep inspection of the packet all the way up to the OSI model's Layer 7. This gives you more reliable and capable access control for OSI Layers 3–7, with simpler configuration.

This additional inspection capability can impact performance. Limited buffering space can hinder deep content analysis.

The application layer firewall can determine an File Transfer Protocol (FTP) session, just like a stateless or stateful firewall can. However, this firewall can look deeper, into the application layer to see that this is specifically an FTP "put" operation, to upload a file. You could have rules that deny all FTP uploads. Or you can configure a more granular rule such as one that denies all FTP uploads except those from a specific source IP and only if the filename is "os.bin".

The deeper packet inspection capability of the application layer firewall enables it to verify adherence to standard HTTP protocol functionality. It can deny requests that do not conform to these standards, or otherwise meet criteria established by the security team.

5.4.4

Load Balancers

Load balancing improves the distribution of workloads across multiple computing resources, such as servers, cluster of servers, network links, and more. Server load balancing helps ensure the availability, scalability, and security of applications and services by distributing the work of a single server across multiple servers.

The load balancer decides which server should receive a client request such as a web page or a file. The load balancer selects a server that can successfully fulfill the client request most effectively, without overloading the selected server or the overall network.

At the device level, the load balancer provides the following features to support high network availability:

- **Device redundancy** – Redundancy allows you to set up a peer load balancer device in the configuration so that if one load balancer becomes inoperative, the other load balancer can take its place immediately.
- **Scalability** – Virtualization allows running the load balancers as independent virtual devices, each with its own resource allocation.
- **Security** – Access control lists restrict access from certain clients or to certain network resources.

At the network service level, a load balancer provides the following advanced services:

- **High services availability** – High-performance server load balancing allows distribution of client requests among physical servers and server farms. In addition, health monitoring occurs at the server and server farm levels through implicit and explicit health probes.

- **Scalability** – Virtualization allows the use of advanced load-balancing algorithms (predictors) to distribute client requests among the virtual devices configured in the load balancer. Each virtual device includes multiple virtual servers. Each server forwards client requests to one of the server farms. Each server farm can contain multiple physical servers.
- **Services-level security** – This allows establishment and maintenance of a Secure Sockets Layer (SSL) session between the load balancer and its peer, which provides secure data transactions between clients and servers.

Although the load balancer can distribute client requests among hundreds or even thousands of physical servers, it can also maintain server persistence. With some e-commerce applications, all client requests within a session are directed to the same physical server so that all the items in one shopping cart are contained on one server.

You can configure a virtual server to intercept web traffic to a website and allow multiple real servers (physical servers) to appear as a single server for load-balancing purposes.

A virtual server is bound to physical hardware and software resources that run on a real, physical server in a server farm. They can be configured to provide client services or as backup servers.

Physical servers that are all perform the same or similar functions are grouped into server farms. Servers in the same server farm often contain identical content (referred to as mirrored content) so that if one server becomes inoperative, another server can take over its functions immediately. Mirrored content also allows several servers to share the load during times of increased demand.

You can distribute incoming client requests among the servers in a server farm by defining load-balancing rules called predictors using IP address and port information.

When a client requests an application service, the load balancer performs server load balancing by deciding which server can successfully fulfill the client request in the shortest amount of time without overloading the server or server farm. Some sophisticated predictors take into account factors such as the server load, response time, or availability, allowing you to adjust load balancing to each match the behavior of a particular application.

You can configure the load balancer to allow the same client to maintain multiple simultaneous or subsequent TCP or IP connections with the same real server for the duration of a session. A session is defined as a series of interactions between a client and a server over some finite period of time (from several minutes to several hours). This server persistence feature is called stickiness.

Many network applications require that customer-specific information be stored persistently across multiple server requests. A common example is a shopping cart used on an e-commerce site. With server load balancing in use, it could potentially be a problem if a back-end server needs information generated at a different server during a previous request.

Depending on how you have configured server load balancing, the load balancer connects a client to an appropriate server after it has determined which load-balancing method to use. If the load balancer determines that a client is already stuck to a particular server, then the load balancer sends subsequent client requests to that server, regardless of the load-balancing criteria. If the load balancer determines that the client is not stuck to a particular server, it applies the normal load-balancing rules to the request.

The combination of the predictor and stickiness enables the application to have scalability, availability, and performance as well as persistence for transaction processing.

SSL configuration in a load balancer establishes and maintains an SSL session between the load balancer and its peer, enabling the load balancer to perform its load-balancing tasks on the SSL traffic. These SSL functions include server authentication, private-key and public-key generation, certificate management, and data packet encryption and decryption. Depending on how the load balancer is configured, it can also perform SSL offloading, by terminating the SSL session from the client on the load balancer itself. This way, the resource intensive SSL processes are offloaded on the load balancer itself, instead of terminating on the backend servers.

Application services require monitoring to ensure availability and performance. Load balancers can be configured to track the health and performance of servers and server farms by creating health probes. Each health probe created can be associated with multiple real servers or server farms.

When the load balancer health monitoring is enabled, the load balancer periodically sends messages to the server to determine the server status. The load balancer verifies the server response to ensure that a client can access that server. The load balancer can use the server response to place the server in or out of service. In addition, the load balancer can use the health of servers in a server farm to make reliable load-balancing decisions.

Additional details on how load balancers interact with applications and load balancing algorithms is covered in the Application Development and Security module of the course.

5.4.5

Network Diagrams

It is very important to document your code, not only to make it easier to understand and follow by other people who will be reading and reviewing it, but also for yourself. Six months down the road, when you come back and look at your code, you might find it very difficult and time consuming to remember what exactly went through your mind when you wrote that amazing and aptly named f() function.

Network diagrams are part of the documentation that goes with a network deployment and play just as an important role as the documentation steps in programming code. Network diagrams typically display a visual and intuitive representation of the network, depicting how are all the devices are connected, and in which buildings, floors, closets are they located, as well as what interface connects to each device.

Imagine being dropped into a place you have never been to, without GPS, without a map, with the instruction to find the closest grocery store. This is what it feels like to manage a network of devices without a network diagram and network documentation. Instead of finding the grocery store, you have to figure out why a large number of devices are no longer connected to the network. You might be able to find the grocery store eventually, if you set off in the right direction. Similarly, you also might be able to figure

out the network problem. But it would take you a lot less time if you had access to a map, a network diagram.

As networks get built and configured and go through their lifecycle of ordering the devices, receiving them on site, bringing them online and configuring them, maintaining and monitoring them, upgrading them, all the way to decommissioning them, and starting the process over again, network diagrams need to be updated and maintained to document all these changes.

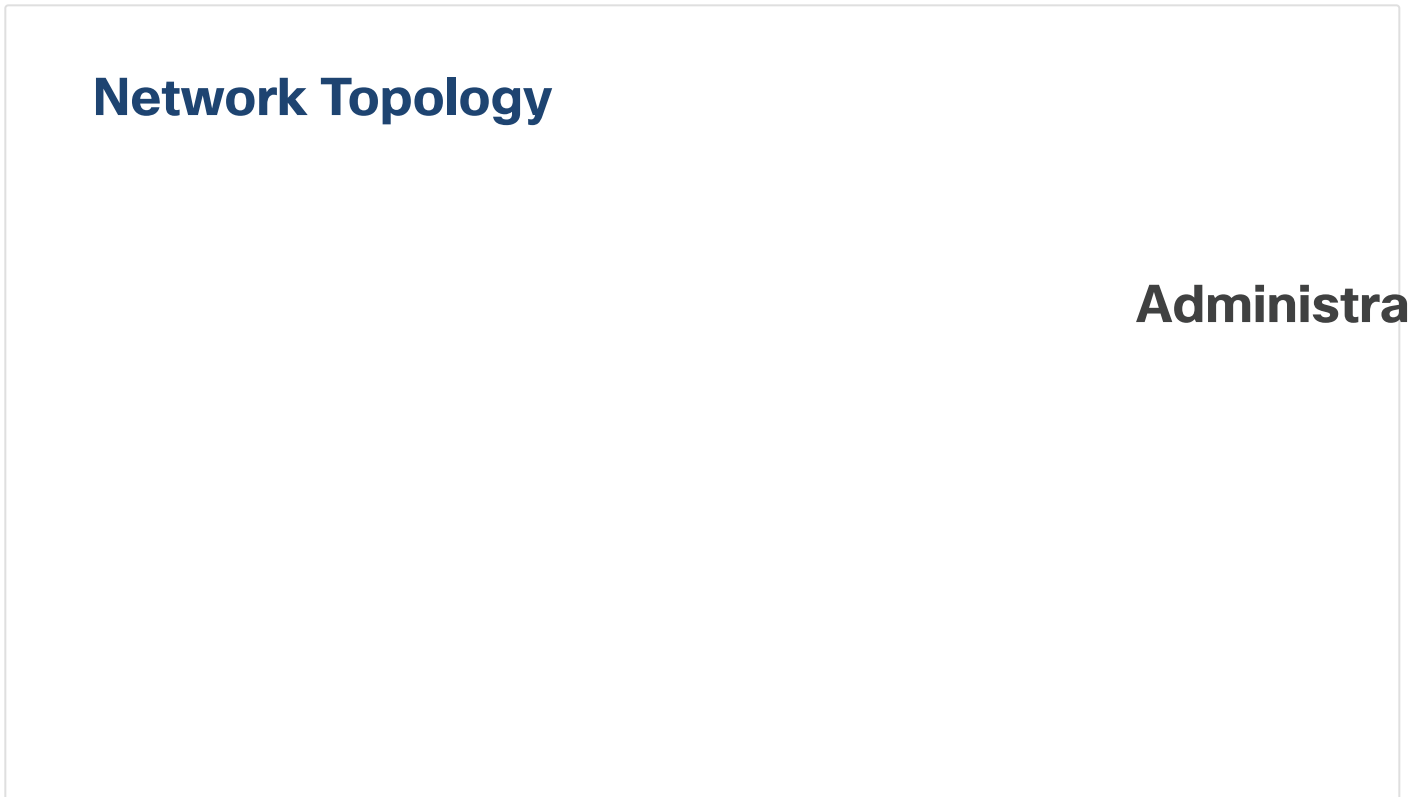
There are generally two types of network diagrams:

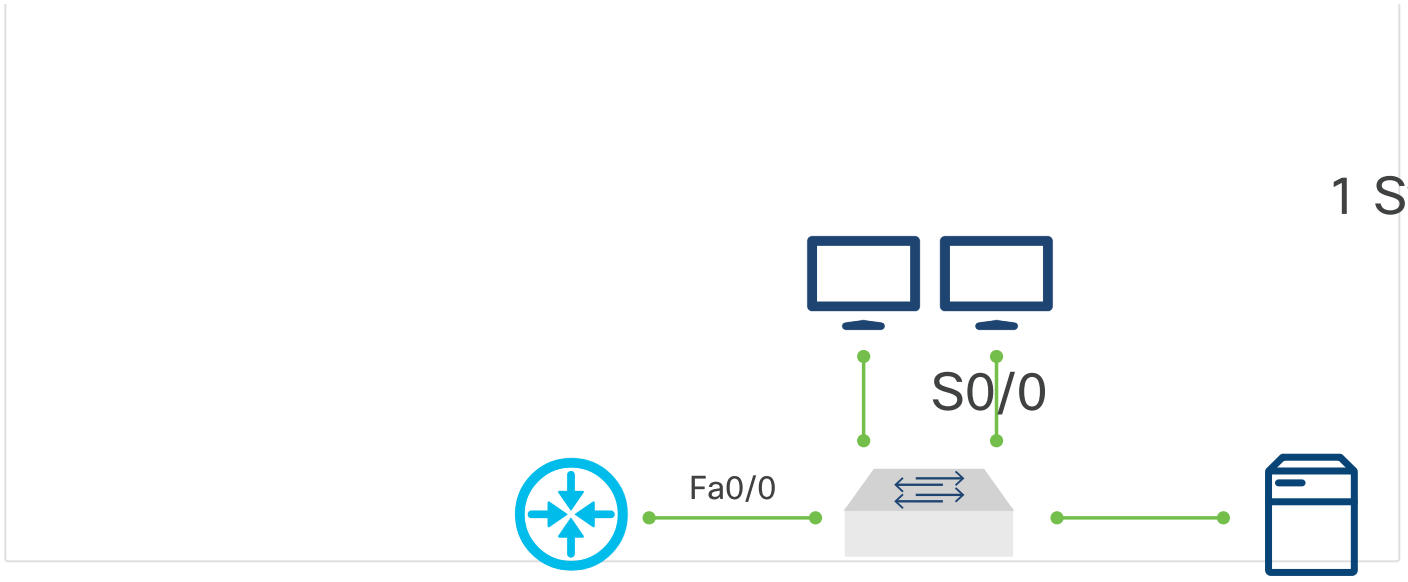
- Layer 2 physical connectivity diagrams
- Layer 3 logical connectivity diagrams

Layer 2, or physical connectivity diagrams are network diagrams representing how devices are physically connected in the network. It is basically a visual representation of which network port on a network device connects to which network port on another network device. Protocols like Cisco Discovery Protocol (CDP) or Link Layer Discovery Protocol (LLDP) can be used to display the physical network port connectivity between two or more devices. This network diagram is useful especially when troubleshooting direct network connectivity issues.

Layer 3, or logical connectivity diagrams are network diagrams that display the IP connectivity between devices on the network. Switches and Layer 2 devices are usually not even displayed in these diagrams as they do not perform any Layer 3 functions and from a routing perspective, they are the equivalent of a physical wire. This type of network diagram is useful when troubleshooting routing problems. Redundant connections and routing protocols are usually present in networks that require high availability.

An example of a simplified Layer 2 network diagram is displayed in the figure. Notice that there is no Layer 3 information documented, such as IP addresses or routing protocols.





Looking at this diagram you can get a general idea of how the clients connect to the network and how the network devices connect to each other so that end to end connectivity between all clients is accomplished. Router RTR1 has two active interfaces in this topology: FastEthernet 0/0 and Serial 0/0. Router RTR2 has three active interfaces: FastEthernet0/0, FastEthernet 1/0, and Serial 0/0.

Most Cisco routers have network slots that support modular network cards. This means that the routers are a future proof investment in the sense that when upgrading the capacity of the network, for example from 100 Mbps FastEthernet to 1 Gbps GigabitEthernet and further to 10 Gbps TenGigabitEthernet, you can simply swap between modular interfaces and still use the same router. Modular Ethernet cards for Cisco routers usually have multiple Ethernet ports on each card.

In order to uniquely identify the modular cards and the ports on each one of these cards, a naming convention is used. In the figure above, FastEthernet 0/0 specifies that this FastEthernet modular card is inserted in the first network module on the router (module 0, represented by the first 0 in 0/0) and is the first port on that card (port 0, represented by the second 0 in 0/0). Following this logic, FastEthernet 0/1, references the second FastEthernet port on the first FastEthernet module and FastEthernet 1/2, references the third FastEthernet port on the second FastEthernet module. Cisco routers support a large number of network modules implementing different technologies including the following: FastEthernet (rarely used these days), GigabitEthernet, 10GigabitEthernet, 100GigabitEthernet, point to point Serial, and more.

Going back to the network diagram above, we see there are 2 routers RTR1 and RTR2 in the diagram connected through a serial network connection. Interface FastEthernet 0/0 on RTR1 connects to a switch that provides network connectivity to a server and 20 hosts in the Administration organization. Interface FastEthernet 0/0 on router RTR2 connects to 4 switches that provide network connectivity to 64 hosts in the Instructor group. Interface FastEthernet 1/0 on Router RTR2 connects to 20 switches that provide network connectivity to 460 hosts in the Student group.

5.4.6

Packet Tracer – Explore a Simple Network

Packet Tracer is a great tool for building and testing networks and network equipment. As a developer, it is important that you are familiar with network devices and how they communicate with each other. The simple network in this Packet Tracer activity is pre-configured to give you an opportunity to explore the devices.

You will complete these objectives:

- Part 1: Add PCs to the Topology
- Part 2: Test Connectivity Across the Network
- Part 3: Create a Web Page and View it
- Part 4: Examine the FIREWALL Access Lists

Explore a Simple Network

Explore a Simple Network

5.3
Internetwork Layer

5.5
Networking Protocols