UBIQUITI ACADEMY

UBRSS

UBIQUITI Broadband Routing & Switching Specialist

Ubiquiti Broadband Routing & Switching Specialist

Table of Contents

Foreword

The *Ubiquiti Broadband Routing and Switching Specialist* (UBRSS) training book is made freely available to you as a learning resource to prepare you for taking Ubiquiti certification exams. During classroom training events, students engage in real-world lab activities using the latest Ubiquiti hardware, led by a Ubiquiti-Certified Trainer proficient in the course topics to guide class discussions.

To empower our global user base, the Ubiquiti Academy provides this training book as a reference, to be used to begin and accelerate your learning - it is not a substitute for training courses led by a qualified instructor. When you're ready, sign up for an official Ubiquiti training course and gain recognition as a Ubiquiti-certified professional in your industry expertise.

Ubiquiti acknowledges that professional success in the rapidly-evolving technological world of today requires a strong commitment to continued learning through diverse methods of study. As you read this training book, be sure to participate in our active User Community, where thousands of users come together daily to discuss best practices for configuring, deploying, and troubleshooting real-world projects designed and built on Ubiquiti's cutting-edge platforms.

Jamie Higley
Global Director of Training
Ubiquiti Networks, Inc.
April 2017

I. UBRSS Course Objectives

Welcome to the Ubiquiti Broadband Routing & Switching Specialist course! This is an entry-level course specially designed for networking professionals in the service provider background. The topics include:

- First-time use, setup & management of network equipment
- Basic network design, protocol stacks and data models
- Addressing and subnetting for IPv4 networks
- Anatomy of a router and essential routing protocols
- Standard network services and security across different OSI layers

Routing & Switching Certification Track

The UBRSS course is designed for students with virtually no prior knowledge of network theory. While not a prerequisite to the UBRSA course, UBRSS lays the foundations for the fundamental routing and switching concepts that surround service provider networks, including VLANs, Policy-Based Routing, multi-area OSPF, as well as intro-to-BGP. Mastery of the UBRSS course concepts is crucial to your success in the networking world and advancement through the Ubiquiti Academy.

Lab Overview

The UBRSS course is designed with plenty of hands-on lab activities to accelerate the learning process. The trainer will supply each student with an EdgeRouter-X (ER-X), EdgeSwitch-8-150W (ES-8-150W), and airMAX-ac device to simulate their own local area network. Students will also connect their ER-X to the airMAX-ac radio, which connects to the trainer's airMAX-ac, which connects to the trainer's EdgeRouter (and upstream Internet). Read the description at the beginning of each lab activity to understand the purpose of each lab, then follow the instructions step-by-step to complete the activity.

Your trainer will assign you a unique number (X) to differentiate your IP settings from that of others. Later, you will work in groups (Student A and B) to complete lab activities, where your unique number (Student X) is still used for reference.

As an example, Student A and B work in a group and use their unique numbers (1 and 2, respectively). If the lab activity requires Student B to set an interface address to "10.1.(100+A).B", then Student B would set the interface address to "10.1.101.2", since (100+A) = (100+1) = 101 and B = 2.

II. Device Management

TOUGHSwitch

Ubiquiti's TOUGHSwitch products are simple, yet powerful managed switches. Compared to an unmanaged switch, a managed switch provides advanced options for configuration. In the case of TS-5-POE, users can begin configuring the switch on the management web GUI at **https://192.168.1.20/**. Restricting management access to only authorized users is an important first step in securing the network. As with most Ubiquiti devices, TOUGHSwitch has a default username and password of **ubnt / ubnt** for fast and convenient setup. Administrators who do not modify the default account credentials risk snooping by a casual customer or a menacing attacker. These settings, as well as settings for the web server management (such as IP address, server port and protocols), can be changed, enabled, or disabled as needed on the *Device* tab.

One common method for securing the TOUGHSwitch is through the *Management* port. When enabled, this port can serve as an **out-of-band** management interface. This means that **in-band** hosts connected to the normal switch ports (also called *Data* ports) cannot access (let alone ping) the TOUGHSwitch unless physically connected on the *Management* port. The *Management* port is not recommended as a *Data* port (whether upstream or downstream), since it is limited to 10/100Mbps speeds and could lead to high packet loss. Instead, use *Data* ports for this purpose since they support the high 1000Mbps (Gigabit) speeds needed when passing normal, non-Management traffic.

The **Power-Over-Ethernet** (POE) capabilities of the TOUGHSwitch make it ideal as a managed switch for powering radios, IP cameras and other devices in a service provider network. Although the TS-5-POE allows for up to 24V of **passive POE** per port, the TS-8-PRO and TS-16-CARRIER allow for up to 48V of passive POE across all switch ports. The TS-16-CARRIER is simply two TS-5-POE devices in a rack-mount, so if interconnecting the two switches is desired, run a cable from one port on each switch.

An essential reason for deploying the TOUGHSwitch in service provider settings is the ability to monitor port status and track data distribution on the *Status* tab. Ping Watchdog, System Log and Alerts are all useful tools for maintaining a functional carrier network on a port-by-port basis.

EdgeRouter

EdgeMAX is the family name given to the EdgeRouter hardware and EdgeOS software used to route and filter packets from the edge to the very core of today's carrier-grade networks. Similar to the TOUGHSwitch, the EdgeRouter features a built-in web GUI for simple management access on **https://192.168.1.1/** (with default configuration). Before placing the EdgeRouter in a live network, be sure to change the username and password to something different from the default **ubnt / ubnt**.

Once logged into EdgeOS, the *System* tab contains most of the important management settings for the router. Setting a **hostname** is useful in a local network, where via a name server, IP addresses (e.g., 192.168.87.1) are mapped to simple router names (e.g., Tower3-Shanghai). This area is where backup configurations can be downloaded or uploaded to the router, as well as upgrades to system firmware.

Similar to the TOUGHSwitch *Management* port, all EdgeRouter models feature a *Console* port for **out-of-band** management. This is especially useful should you desire a dedicated line for remote management of the router. In all cases, a serial console cable (RJ45-to-DB9) should be used to connect to the *Console* port, since it is just a serial port with an RJ45 connector. If the computer does not have a DB9 port, then a DB9-to-USB adapter is also required. Use PuTTY (Windows) or Terminal (Mac/Linux) to connect using the following settings:

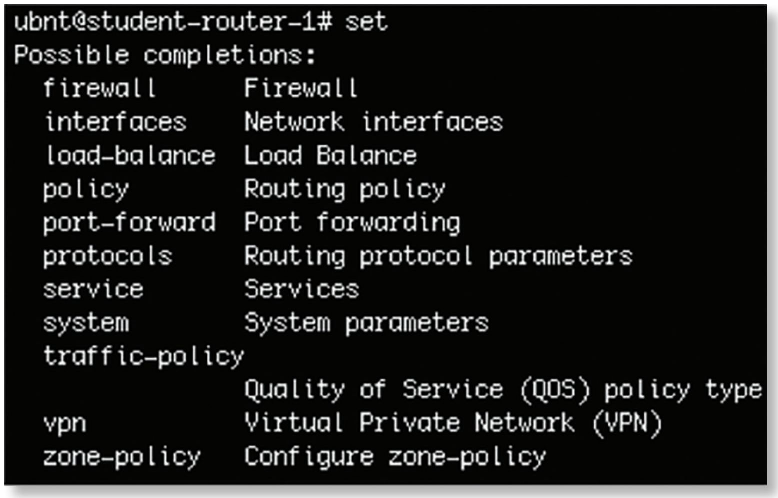
• Baud rate:	115200
• Data bits:	8
• Parity:	NONE
• Stop bits:	1
• Flow control:	NONE

The *Console* port allows users to access the **Command Line Interface** of the EdgeRouter to issue commands, albeit without the intuitive GUI of EdgeOS. Users can also access the CLI using the SSH protocol across any of the Ethernet ports of the EdgeRouter. Given its secure protocol, SSH is preferred over telnet for remote management purposes. The CLI gives users absolute freedom to configure the router, often enabling functionality that is not currently supported via the GUI. You should also be aware of a few handy **shortcuts** that make CLI-based configuration more easy and convenient.

- The "tab" key will attempt to complete the current string
- The "?" symbol allows users to view possible string completions
- The **`configure`** command allows users to leave operational mode and enter configuration mode

- The **`run`** command allows operational mode commands while in configuration mode
- The **`exit`** command allows users to leave modes or terminate a CLI session.

EdgeOS features two modes of management when accessing the CLI: **Operational** and **Configuration** mode. Operational mode is useful for performing commands and reviewing the state of the router. Entering Configuration mode is necessary to make changes to any of the configuration files stored on the router.



```
ubnt@student-router-1# set
Possible completions:
  firewall      Firewall
  interfaces    Network interfaces
  load-balance  Load Balance
  policy        Routing policy
  port-forward  Port forwarding
  protocols     Routing protocol parameters
  service       Services
  system        System parameters
  traffic-policy
                Quality of Service (QOS) policy type
  vpn           Virtual Private Network (VPN)
  zone-policy   Configure zone-policy
```

When using the EdgeOS CLI, there are three **configuration states** used:

- **Working** When making changes to the working configuration, they are not applied until changes are *commited* (**`commit`**) to the active configuration.
- **Active** After changes are *committed* to the active configuration, they are applied; however, the changes do not become part of the *boot* configuration until changes are *saved* to the boot configuration (the equivalent of a *running* configuration).
- **Boot** After changes are *saved* (**`save`**), they are applied to the *boot* configuration. When the EdgeRouter *reboots* (**`reboot`**), it loads the *boot* configuration for use.

Whether making changes to the IP settings or firewall rules on the EdgeRouter, it is often best to use the **`commit-confirm`** command. Following the command, the admin must return to the console to issue the **`confirm`** command. However, if the admin lost access to the router and cannot issue the **`confirm`** command after **10 minutes**, the router will return to previous configuration state.

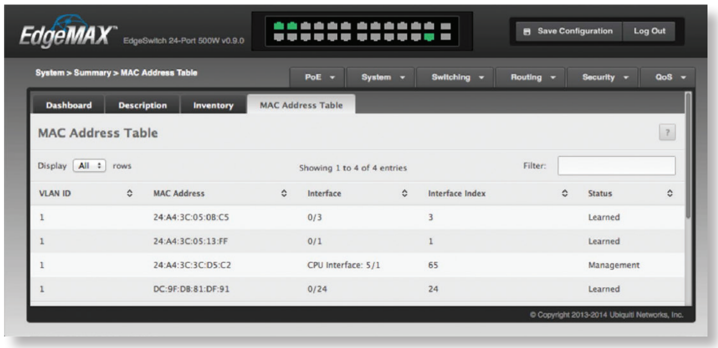
You can always download a backup configuration file over HTTPS via the *System* tab of the EdgeOS GUI. Be sure to keep backup configurations of routers in a production network, should anything happen where a quick restoration is needed. EdgeOS supports automatic, remote backups after every commit, using the **`commit-archive`** command and SCP, FTP or TFTP protocols. You can also keep a specified number of revisions of the configuration file on the local EdgeRouter using the **`commit-archive`** command. See the **EdgeOS Commands** Appendix (located at the back of this manual) for more details.

EdgeSwitch

EdgeSwitch is the newest addition to the EdgeMAX family, combining advanced layer-2 switch functions with needed layer-3 routing into a single device (more on **layers** in the next chapter). Featuring GigE across as many as 48 ports, 2 SFP+ and 2 SFP ports, the **multi-layer switch** pushes as much as 70 Gbps non-blocking throughput. Support for so many different layer-2 and layer-3 protocols and functions places the EdgeSwitch in a unique position to work virtually anywhere in the service provider topology.

Compared to the TOUGHSwitch models, EdgeSwitch supports **POE+ IEEE 802.3at/af** as well as **Passive POE**. A variety of models exist to cater to each unique carrier deployment, including:

- ES-24-250W
- ES-24-500W
- ES-48-500W
- ES-48-750W



System > Summary > MAC Address Table

MAC Address Table

Display All rows   Showing 1 to 4 of 4 entries   Filter:

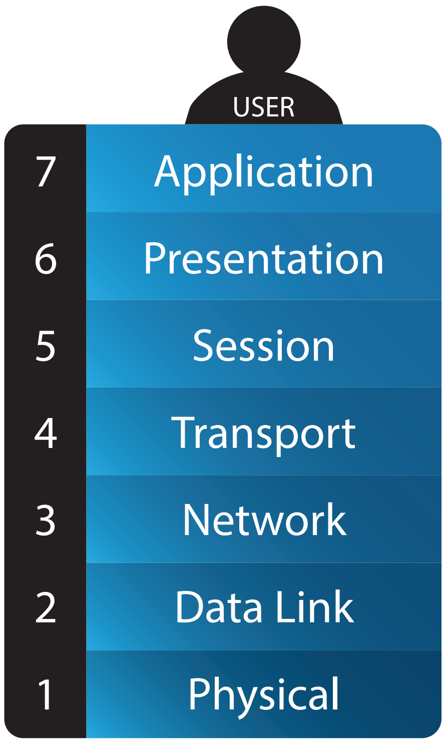
VLAN ID	MAC Address	Interface	Interface Index	Status
1	24:A4:3C:05:0B:C5	0/3	3	Learned
1	24:A4:3C:05:13:FF	0/1	1	Learned
1	24:A4:3C:3C:D5:C2	CPU Interface: 5/1	65	Management
1	DC:9F:DB:81:DF:91	0/24	24	Learned

Note for Student: Ask your trainer to share a read-only account so that you can explore the EdgeOS featureset for EdgeSwitch. For tutorials and instructional guides, visit the EdgeSwitch Knowledge Base located on Ubiquiti's Community.

III. Network Design

OSI Model & Encapsulation

The **Open Systems Interconnection** (OSI) **Model** is a 7-layer data model designed to organize the various software protocols and hardware involved in network communication. At the top of the stack, the **Application Layer** represents protocols such as FTP, DHCP or HTTP. Most end-users work comfortably at this layer and are not concerned with the work that occurs in lower layers. Thus the **payload** (original data) is passed down layer by layer. In this way, each layer exists to serve the layer above it.

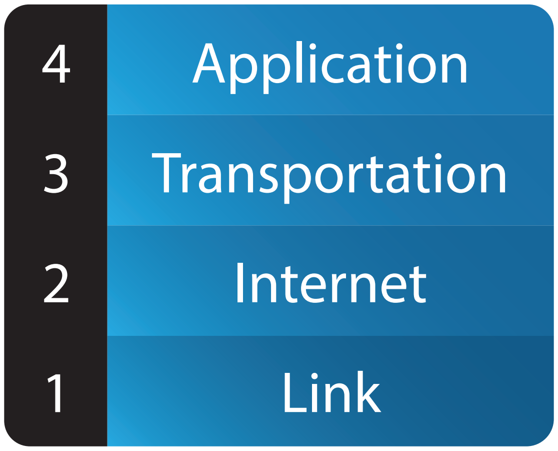


Eventually, application data arrives at the **Transport Layer**, where network port numbers are assigned and the "payload" becomes a "segment." This process by which layer-relevant information is added and enveloped around the original data payload is called **encapsulation**. At the **Network Layer**, the "segment" is encapsulated with IP information and becomes a "packet." After arriving at the **Data Link Layer**, the packet is encapsulated with relevant MAC information and becomes a "frame." The frame finally passes to the **Physical Layer** where the frame becomes a transmitted "signal" on the physical medium. At the other end of the link the transmitted Physical Layer signal arrives at the receiver.

Based on the received Physical Layer signal, the receiver begins to reconstruct the Layer-2 frame. If the destination MAC address contained in the frame belongs to the receiver, the frame will be stripped of MAC information and passed up to the Network Layer. This process by which layer-relevant information is stripped off layer-by-layer is called **decapsulation.** In a properly designed and configured network, the original data payload will eventually arrive at its intended destination.

TCP/IP Model & End-to-End Communication

As a simplified version of the OSI model, the **TCP/IP Model** is useful in diagramming end-to-end communication in today's networks, including the Internet. The **TCP/IP stack** groups protocols into four layers: **Application**, **Transport**, **Internet**, and **Link**. As with the OSI model, the Application Layer initiates the payload and the Transport Layer assigns ports. The Internet Layer deals with internetwork (IP) packet routing while the Link Layer works with local area (MAC) communication and physical transmission methods.



The topic of intranetwork communication on local networks will be explored later (see ARP). For now, understand that the Link Layer encapsulates and decapsulates the original data payload with information such as the MAC address so that it can reach a local node. Regarding internetwork communication, including Internet traffic, the TCP/IP model is synonymous to a **postal system**. The user writes the **letter** (data) and puts the letter in an **addressed envelope** (packet). After arriving at the **post office** (gateway), the **recipient address** (destination IP) is checked against **records** (routing tables), then **forwarded** to another **postal office** (neighbor router), repeating the process (hop-by-hop) until the packet arrives.

Transport Layer & Connections

At the transport layer, network devices assign port numbers and decide which transport protocol to use for end-to-end communication. The transport protocol used governs behavior for connectivity; there are two major protocols used in today's networks:

- **Transmission Control Protocol (TCP)** connection-oriented, reliable, orderly, and supports error-checking.
- **User Datagram Protocol (UDP)** connectionless, unreliable, disorderly, and stateless (but faster than TCP).

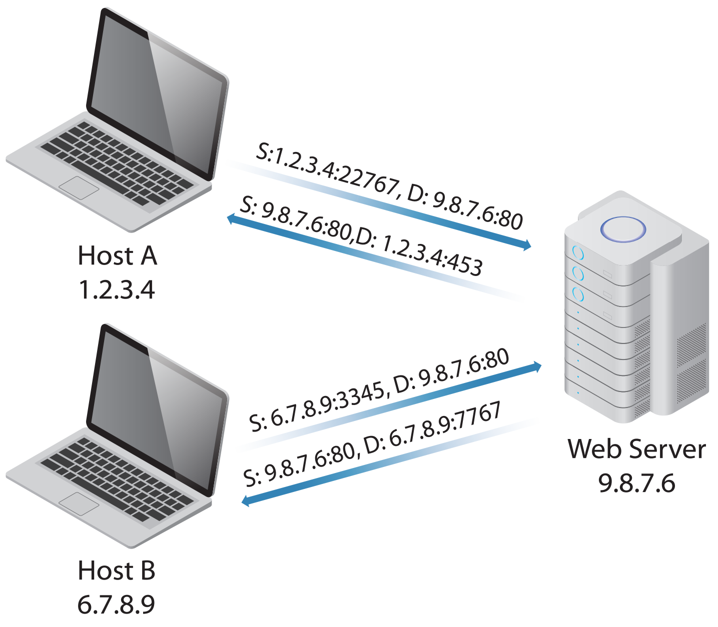
Compared to UDP, TCP is used in a wide range of applications, e.g., HTTP (80), HTTPS (443), FTP (20, 21), SSH (22), and more (ports in parentheses). TCP demonstrates the **handshake** model for establishing and acknowledging data connections between hosts. The handshake is as follows:

1. Sender requests **synchronization**.
2. Receiver **acknowledges** and **syncs** as well.
3. Sender **acknowledges**.

Due to the connectionless nature of UDP, it is useful with real-time network applications such as Skype, VOIP, and gaming. Other common UDP applications include DNS (53), DHCP (67, 68), TFTP (69), and SNMP (161, 162).

Compared to TCP and UDP, the **Internet Control Messaging Protocol** (ICMP) is considered a **Network Layer** protocol, not used to carry data, but instead send control messages about errors, routing, etc. The network utility **ping** sends ICMP **echo** requests.

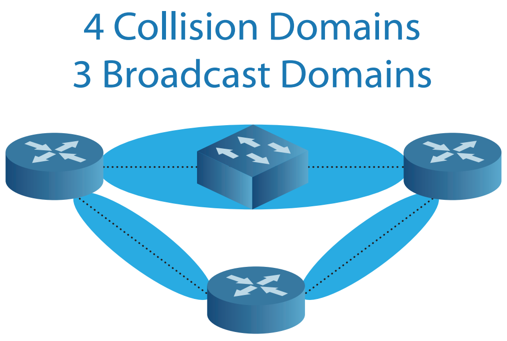
As previously mentioned, **port assignment** also occurs at the Transport Layer. Because hosts may establish connections with multiple hosts using the same transport protocol, hosts assign different port numbers in order to track connections (and serve the application layer above it). The following example demonstrates a web server (9.8.7.6) that receives connections from two hosts on **destination port** 80. The hosts randomly choose **source ports** to establish TCP connections and request to SYN. The server sends an ACK reply and requests to SYN with the host. The host then sends an ACK and begins passing HTTP data.

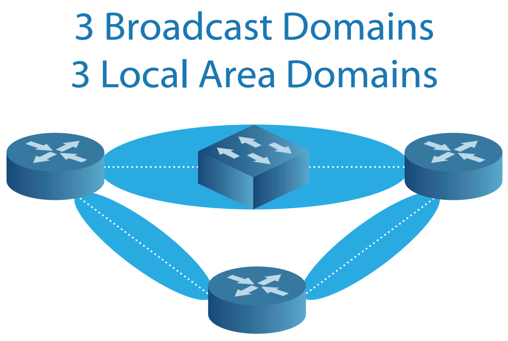

S:1.2.3.4:22767, D: 9.8.7.6:80
S: 9.8.7.6:80,D: 1.2.3.4:453
Host A
1.2.3.4
S: 6.7.8.9:3345, D: 9.8.7.6:80
S: 9.8.7.6:80, D: 6.7.8.9:7767
Web Server
9.8.7.6
Host B
6.7.8.9

Network Topology Foundations

Compared to today's networks, early networks lacked some efficient foundational protocols and technologies. "**Thicknet**," was one of the first implementations of Ethernet that joined devices on the same wired segment. In this way, all connected devices competed for use of the same transmission medium. Whenever two hosts transmitted simultaneously, a collision occurred. Fortunately, collisions were detectable via the **CSMA/CD** (Carrier Sense Multiple Access / Collision Detection) protocol, reducing the chance for errors and latency.

The area of a network where a collision between two or more network devices is possible is known as a **collision domain**. After the introduction of network **bridges**, collision domains became much easier to manage. By design, bridges join segments of the network while dividing the collision domain between them. The effectiveness of bridges was compounded by successor technologies such as network **switches** and twisted-pair Ethernet. Featuring full-duplex, bi-directional communication and collision domains across each switch port, collisions are all but eliminated in wired networks.





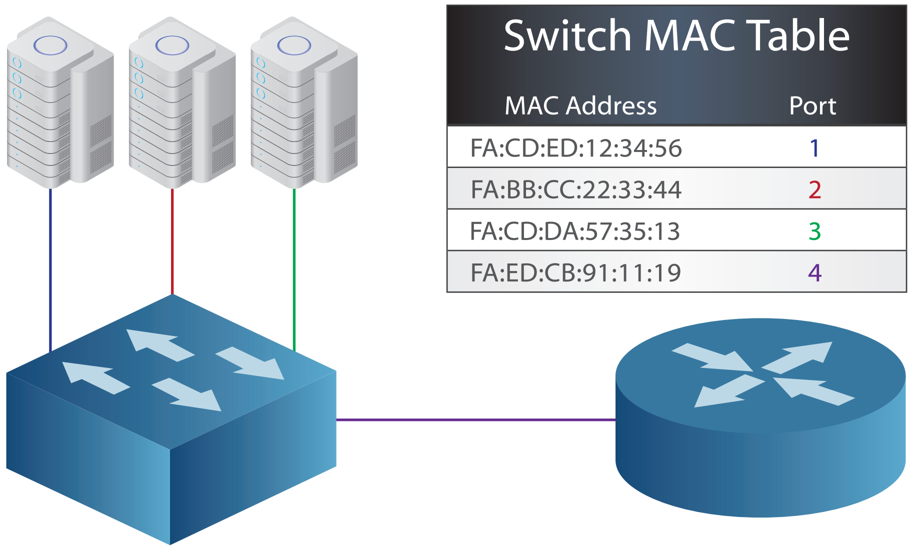
Despite segmenting collision domains, bridges and switches actually expand the size of the network **broadcast domain**. Broadcast domains represent the total area of the network where a host can broadcast (announce) a message and reach. Whenever a broadcast reaches a switch, it forwards the message out all ports contained in the broadcast domain (except the port on which the message was received). After the broadcast domain becomes too large, broadcast messages create crippling **broadcast storms**. As switch after switch flood the network (often unnecessarily), overall performance begins to degrade. Only **routers**, which are designed to stop broadcast messages, can segment the broadcast domain. Today's carrier networks are built on both switches and routers, as well as wireless bridges.

Network Interfaces

A **network interface** represents the software and hardware components that work together to allow a device to communicate on a network. Although network interfaces vary in hardware, software, and protocols, some of the interfaces you should recognize in the service provider setting include **Ethernet**, **wireless**, **PPP** and **loopback**. In computing, system kernels permit an interface's software drivers and hardware to interact. Depending on the type of interface, a unique logical (IP) and/or hardware (MAC) address is assigned to begin communicating on the network.

The primary purpose of a switch is to join hosts on the same network segment, often by way of a single bridge interface (e.g., *br0*) connecting all the ports together. On the other hand, the principal job of the router is to separate networks, which it does through multiple interfaces (e.g., *eth0*, *eth1*, and *eth2*) across multiple ports. For this reason, routers are appropriately placed at the network **core** and **edge** to contain broadcast traffic, while moving packets between different networks, such as the **LAN** and **Internet**.

While on the same LAN segment, devices communicate based on MAC (Media Access Control) addresses, being unique to each network interface. Compared to a router, which has multiple MACs for each of its Ethernet interfaces, a desktop PC might have just one MAC for its single network interface controller (NIC). Despite having multiple ports, an Ethernet switch typically has a single network interface that bridges all of the ports and consequently one MAC address advertised on all ports.



Switch MAC Table

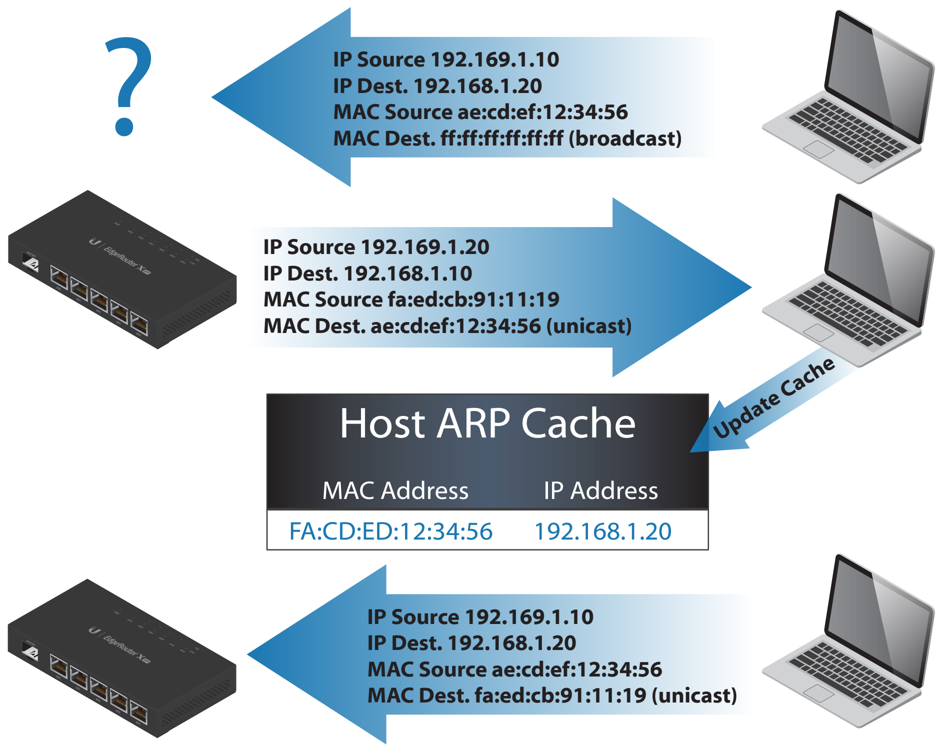
MAC Address	Port
FA:CD:ED:12:34:56	1
FA:BB:CC:22:33:44	2
FA:CD:DA:57:35:13	3
FA:ED:CB:91:11:19	4

The Local Area Network

The concept of a **Local Area Network** (LAN) enables you to create your own private network and join **hosts**. In the absence of a router, a simple, Layer-2 switch can interconnect hosts, such as printers, servers, and end-users on a private network. But to join two different networks (like two LANs), a Layer-3 router should be used.

To understand LAN communication, consider the following example, which prefaces the next lab activities. Two hosts (Student A and B) are connected to the same LAN segment (a single switch). To test connectivity, student A sends pings to the known IP address for student B. Before sending any packet, student A's IP interfaces must determine whether the destination address (student B) exists on the local network. If not, then the packet is encapsulated with the destination MAC of the local gateway and forwarded. If, however, the destination address is local, then the packet is encapsulated and forwarded directly.

After determining that student B is on the same network, student A uses the **Address Resolution Protocol** (ARP) to resolve student B's IP address to MAC address. It first checks its **ARP table** for an existing MAC entry belonging to Host B's IP address, which it does not find. Student A then broadcasts the message "who is 192.168.1.11?" using the layer-2 broadcast address ff:ff:ff:ff:ff:ff. When student B receives the broadcast, it updates its ARP table with student A's MAC address, retrieves the payload, and creates a new packet destined to student A (encapsulated in a frame destined to student A's MAC).

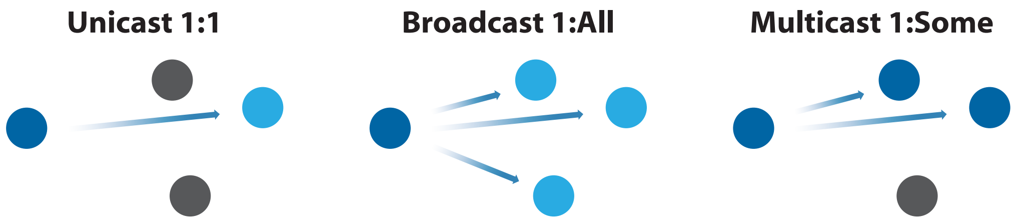


By design, layer-2 switches are not concerned with IP addresses, only MAC address. Upon receiving a frame, a switch inspects the destination MAC to consult its **MAC table** before either forwarding, flooding, or filtering the frame. If the switch's MAC table contains an entry for the destination MAC address, it will forward the frame out that port. If the frame is a broadcast or no entry exists for the destination MAC, the switch flood the frame out to all ports except the port on which the frame was received. Switches also filter frames in case an **Access Control List** (ACL) exists that blacklists the destination MAC address.

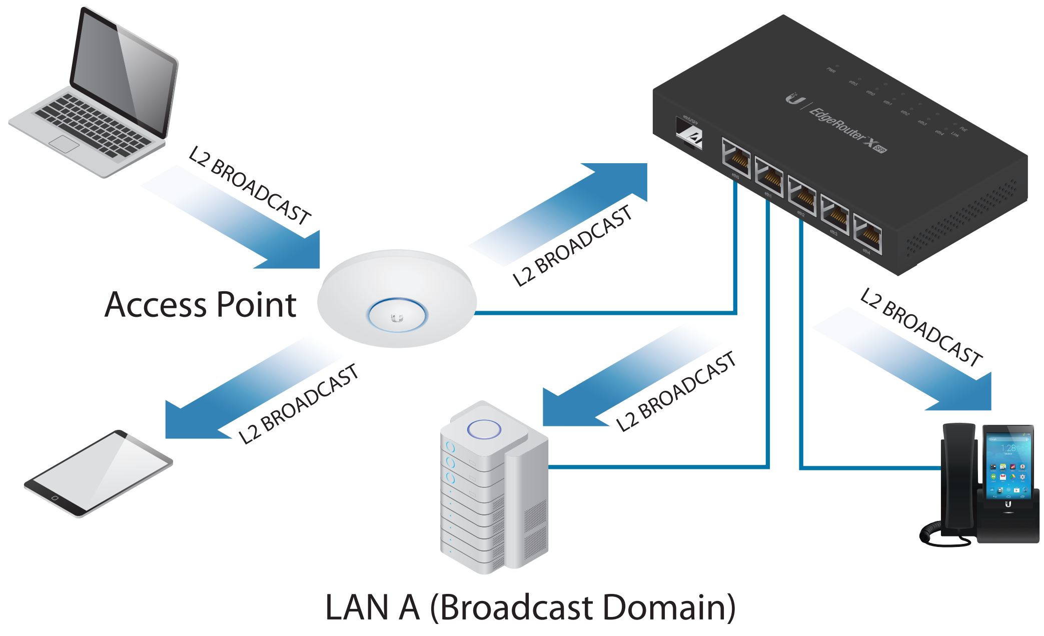
Network Communication

In IPv4 networks, network hosts communicate using one of three methods:

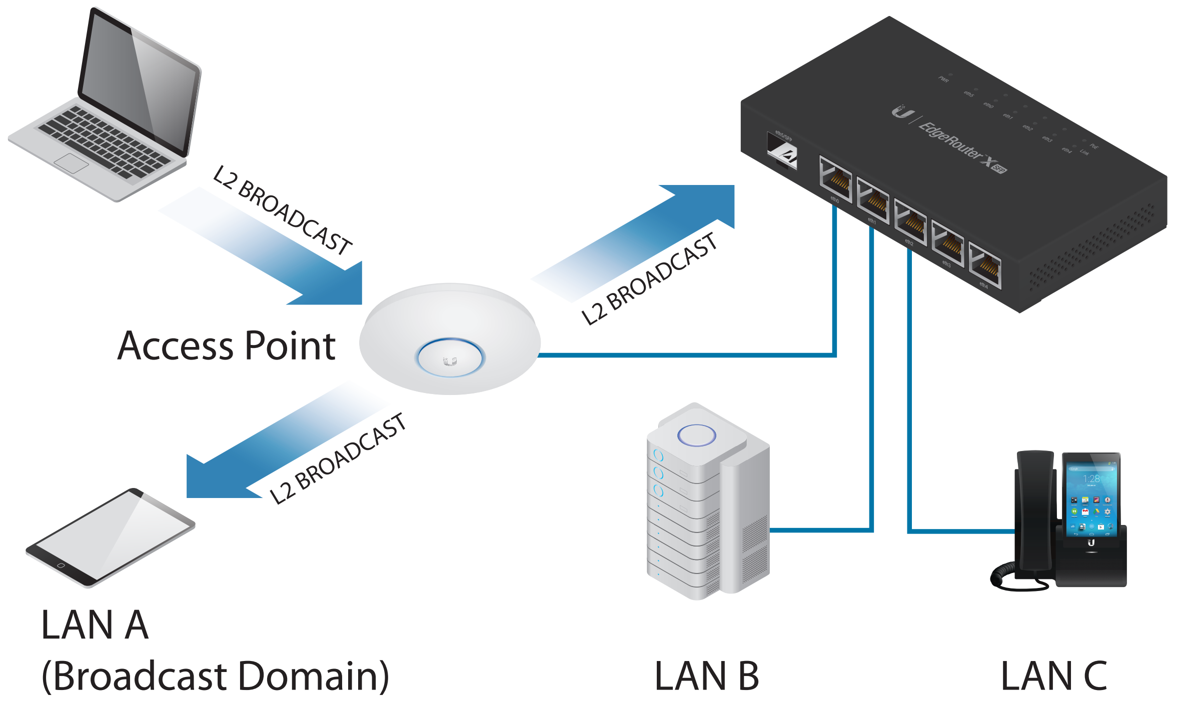
- **Unicast** one-to-one communication between two hosts
- **Broadcast** one-to-all communication between one host and all other hosts on the local network.
- **Multicast** one-to-some communication between one host and multiple hosts.



As networks grow larger, broadcasts become more common, increasing the amount of potentially unnecessary traffic reaching hosts. Hosts check broadcast traffic, even when it is not intended for them, so increased **broadcast storms** can affect network performance. The following image demonstrates a user network (LAN A) whose broadcast domain reaches unintended hosts (servers, phones) via a layer-2 switch.



The following image demonstrates how **LAN segmentation** can limit the severity of broadcast storms in the network environment by replacing a switch with a layer-3 router.



Multicast traffic is a special kind of traffic. Unlike with broadcast traffic, typical network hosts are not configured to listen to multicast traffic. Multicast is seen in networks with **dynamic routing protocols** where routers multicast to share common routing updates. Although multicasting is more efficient than broadcasting, it is common to see multicast traffic not pass across layer-2 network segments (e.g., a wireless bridge) if **multicast support** is disabled. Digital service providers also use multicasting when pushing content such as IPTV services to end subscribers to better conserve network resources.

The Wide Area Network

Routers are responsible for delivery of packets between networks. However, hosts are still responsible for determining who should receive the packet. The IP address and network mask assigned to the network interface of the host is important in the ARP process. By checking the **source** and **destination IP** against the **network mask** (subnet mask), a host knows whether to forward the packet directly to the local recipient or whether to send the packet to the **gateway**. In either case, layer-3 packets are always encapsulated inside layer-2 frames containing source and destination MAC addresses for local communication.

From the perspective of the LAN, the **Wide Area Network** (WAN) refers to the larger, **segmented network** that connects the LAN to the Internet. In the **ISP world**, this is called the **upstream provider**. The TCP/IP model outlines the model for end-to-end communication across the Internet, like when a host requests data from a remote web server. Between the host and web server, there may be a number of **hops**, or, routers that select the best path based on different metrics such as distance and link speed.

Since WAN routers serve as **gateways** to LAN hosts, they may employ a **default route** for all traffic (unless a more specific route exists to the destination IP address). The packet routing process will be examined later but for now it is important to understand that while **all network** communication is concerned with **local** (**MAC**) **addresses**, **internetwork** communication is mostly concerned with **IP addresses**.

The Big Picture

The **Internet** is the largest Wide Area Network in existence, connecting billions of people and servers across the world. **Internet Service Providers** (ISPs) including **Wireless ISPs** (WISPs) provide connectivity to consumers at the edge of the carrier network. Away from the customers and at the ISP border, service providers **peer** at **Internet Exchange Points** (IXPs) to improve their WAN presence and performance through redundant paths, increased capacity, or enhanced routing controls. The **traceroute** tool illustrates the path that packets take as they cross the Internet on a "hop-by-hop" basis.

EdgeMAX and TOUGHSwitch products are specifically designed as routing and switching solutions for ISPs. They are used necessarily at customer and tower sites, across the carrier infrastructure, and finally, at the ISP **Point-of-Presence** (POP).

IV. IPv4 & Subnetting

Before beginning this chapter, know that network addressing is crucial to all manner of service provider tasks, including network architecture, address assignment, and subnetting. Learning the **Internet Protocol** from the perspective of devices such as laptops, switches, and routers is fundamental to building functional networks.

Subnetting is how a large network is broken up into smaller networks. Although there are *many* different methods for deriving subnets, this chapter is designed with examples to help students understand 1) how IP addressing affects network communication, 2) the purpose for subnetting, and, of course, 3) how to subnet!

Addressing Basics

Network Address Anatomy

Internet Protocol version 4 is used in most of today's networks as the standard for addressing. In this way, hosts can communicate with other hosts whether locally on the same network or across network boundaries. A **host** is any device configured with an address to communicate on the network. Layer-2 communication involves **hardware** or **MAC addresses** while Layer-3 communication deals with **logical** or **IP addresses**.

MAC addresses are expressed in **hexadecimal**, containing numbers 0-9 and letters A-F in this format: A1:B2:C3:D4:E5:F6. An IPv4 address is composed of four decimal **octets**, each of which contains a value from 0 to 255, for example: 99.1.101.199. Although an IP address is read from left-to-right, addresses **sequentially grow** from **right-to-left**, as rightmost octets fill up and carry over to left octets. For example, the next IP address in the sequence following 192.168.0.255 is 192.168.1.0, since 255 is the maximum value of an IPv4 octet.

Binary

In computing, **binary numbers** are used to represent all data. Network devices **encode** and **decode** data using binary numbers 0 and 1, which individually are called **bits**. From the context of the OSI model, all information from layers 2-7 is eventually broken down into 0's and 1's to be transmitted at layer 1.

Interestingly, you may have noticed that an IPv4 address never exceeds the value 255 in any of its four decimal places. The name **octet** is derived from the **eight binary bits** contained in each decimal place of the IPv4 address. Each bit has a **binary value** relative to its position in the octet. The following binary examples show how IP addresses grow as **bits** are **summed** across **octets** from **left to right**:

Decimal to Binary Conversion

Decimal	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Exponent
0 - 255	128	64	32	16	8	4	2	1	Binary
192	1	1	0	0	0	0	0	0	11000000
168	1	0	1	0	1	0	0	0	10101000
37	0	0	1	0	0	1	0	1	00100101
36	0	0	1	0	0	1	0	0	00100100
192. 168. 37. 36 =									11000000. 10101000. 00100101. 00100100

The binary equivalent of the decimal address 192.168.37.36 is 11000000.10101000.00100101.00100100. Each bit carries a value relative to its position in the octet, ranging from 2^7 to 2^0. How would the IPv4 decimal address 0.0.0.0 be expressed in binary?

ANDing, ARP, and Forwarding Packets

Whenever a host needs to send a packet to another host, it must first perform ANDing on the target host's address. Once the networks of both sender and recipient are determined, the packet can be forwarded. If the two hosts are on the **same network**, then the sender retrieves the recipient's MAC address using the **Address Resolution Protocol**, then **forwards** the packet **directly**. If the two hosts are on **different networks**, then the packet must be **forwarded** to the **gateway**.

ANDing for Network ID of Host 1

Address Type	Decimal Address	Octet 1	Octet 2	Octet 3	Octet 4
Host Address	192.168.50.50	11000000	10101000	001100010	00110010
Subnet Mask	255.255.255.224	11111111	11111111	11111111	11100000
Network ID	192.168.50.32	11000000	10101000	00110010	00100000

ANDing for Network ID of Host 2

Address Type	Decimal Address	Octet 1	Octet 2	Octet 3	Octet 4
Host Address	192.168.50.65	11000000	10101000	00110010	01000001
Subnet Mask	255.255.255.224	11111111	11111111	11111111	11100000
Network ID	192.168.50.64	11000000	10101000	00110010	01000000

Public vs. Private Addresses

It is possible that you are already familiar with addresses such as 192.168.0.0, 10.0.0.0, or 172.16.0.0. That is because these addresses belong to the private network IP range. Private network addresses are used when creating **local area networks** (LANs). The main difference between private and public network IP addresses is that public IP addresses are assigned to hosts on the global Internet, such as web servers and ISP routers. Later, we will explore how the **Network Address Translation** (NAT) protocol allows hosts with private addresses to communicate on the Internet.

Class	Starting Bits	Start Address	End Address	Use
A	0	10.0.0.0	10.255.255.255	Private
B	10	172.16.0.0	172.31.255.255	Private
C	110	192.168.0.0	192.168.255.255	Private

In the previous table, the first column indicates the class of addresses to which each private IP range belongs. Though address class is not significant in networking today, it is important to recognize classes as they relate to the use of addresses and how subnetting was performed in the past.

Classful Addressing in Legacy Networks

After the introduction of the **Internet Protocol version 4**, **classful addressing** came to define a global system for Internet addressing. Although classful networking worked, it did not come without design flows. First, it did not prevent the **foreseeable exhaustion** of **public IPv4 addresses**. Second, it caused the **Internet routing tables** of ISP routers to grow to **unsustainable sizes**. Service providers faced a serious limitation in the growth of their businesses if no viable solution could be found.

Classful networking's successor, **classless networking**, ultimately solved the issue faced by Internet routing tables. It even alleviated, in part, the rapid depletion of IPv4 addresses. Ultimately however, the solution to the exhaustion of Internet addresses is IPv4's successor, **Internet Protocol version 6** (more on this in the IPv6 Appendix).

Class	Starting Bits	Start Address	End Address	Use
A	0	0.0.0.0	127.255.255.255	All
A	0	0.0.0.0	0.255.255.255	Test
A	0	10.0.0.0	10.255.255.255	Private
A	0	127.0.0.0	127.255.255.255	Loopback
B	10	128.0.0.0	191.255.255.255	All
B	10	172.16.0.0	172.31.255.255	Private
C	110	192.0.0.0	223.255.255.255	All
C	110	192.168.0.0	192.168.255.255	Private
D	1110	224.0.0.0	239.255.255.255	Multicast
E	11110	240.0.0.0	255.255.255.255	Reserved

An address' **network class** was determined by the **starting bits** contained in the address. While classes are not relevant in networking today, IPv4 address ranges belonging to **loopback**, **test**, **multicast** and **private networks** still apply. For more information, consult the Addressing & Routing Tables in the Appendix (at the back of this manual).

Network Mask (Network Range)

With classful addressing came default network mask assignments. The purpose of the **network mask** is to define the **range of addresses** that belong to a **network**. The following table shows the **default network masks** for each class of IPs, as well as their shorthand **prefixes**:

Class	Default Mask	Prefix
A	255.0.0.0	/8
B	255.255.0.0	/16
C	255.255.255.0	/24

When converted to binary, network masks produce an interesting **pattern** of **1 bits** (on the left) and **0 bits** (on the right). Take the default masks for example:

Network Bits Host Bits

255.0.0.0 = 11111111.00000000.00000000.00000000

255.255.0.0 = 11111111.11111111.00000000.00000000

255.255.255.0 = 11111111.11111111.11111111.00000000

The **1 bits represent** *network bits*, while the **0 bits represent** *host bits*. The shorthand **prefix** contains the total number of **network bits** contained in the **mask**.

How then is the network range determined by the mask? Really, it is determined by how many network/host bits exist. The *more network bits*, the *larger* the size of the network. However, the *more host bits*, the *greater* the number of hosts per network. Let us take a look at some private network examples which use the default network masks.

Network Start Address (Network ID)

192.168.0.0/24 = 11000000.10101000.01100100.00000000

255.255.255.0 = 11111111.11111111.11111111.00000000 /24

Network Bits Host Bits

The /24 prefix means that there are exactly 24 network bits in the mask.

Any octet containing a value of less than 255 (e.g., 254, 0) means the range of the network extends through the octet. By analyzing the rightmost network bit, you can determine the **increment** by which each **network** (or **subnetwork**) grows. The mask of the network 192.168.0.0/24 is expressed in binary below:

```
                      Network Bits            Host Bits

255.255.255.0 = 11111111.11111111.11111111.00000000

      3rd Octet = 128   64   32   16   8   4   2   1
                    1    1    1    1   1   1   1   1

      4th Octet = 128   64   32   16   8   4   2   1
                    0    0    0    0   0   0   0   0
```

Since the **rightmost** network bit is contained in the third octet, sequential networks will grow in **increments** of 1 (in the third octet). This means that the network start addresses follow a sequence such as this example:

Network 1: 192.168.0.0/24

Network 2: 192.168.1.0/24

Network 3: 192.168.2.0/24

. . .

Network 255: 192.168.254.0/24

Network 256: 192.168.255.0/24

The reason that networks grow in **increments** of 1 in the 3rd octet is due to the /24 mask. If the network mask changed to say, /25, resulting in a 255.255.255.128 address (instead of /24, 255.255.255.0), the network would grow in **increments** of 128 in the 4th octet, as so:

```
                      Network Bits            Host Bits

255.255.255.128 = 11111111.11111111.11111111.10000000

      3rd Octet = 128   64   32   16   8   4   2   1  ]  Old Mask
                    1    1    1    1   1   1   1   1  ]  Increment

      4th Octet = 128   64   32   16   8   4   2   1
                    1    0    0    0   0   0   0   0

                  New Mask
                  Increment
```

The new subnet mask causes sequential subnetworks to grow in **increments** of 128 in the 4th octet:

Network 1: 192.168.0.0/25

Network 2: 192.168.0.128/25

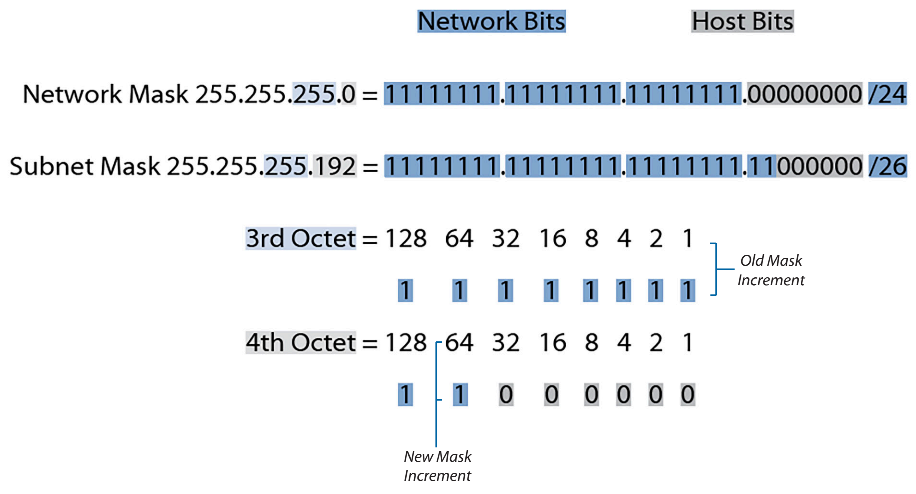
Network 3: 192.168.1.0/25

. . .

Network 511: 192.168.255.128/25

Network 512: 192.168.255.128/25

This is, in reality, how subnetting occurs. As *more network bits* are added to the mask, the *number of networks increases* while *the range of addresses decreases*. Similarly, if the network mask were extended to /**26** (255.255.255.**192**), then sequential subnetworks would grow in increments of **64**.


Network Bits
Host Bits
Network Mask 255.255.255.0 = 11111111.11111111.11111111.00000000 /24
Subnet Mask 255.255.255.192 = 11111111.11111111.11111111.11000000 /26
3rd Octet = 128 64 32 16 8 4 2 1
1 1 1 1 1 1 1 1
Old Mask Increment
4th Octet = 128 64 32 16 8 4 2 1
1 1 0 0 0 0 0 0
New Mask Increment

Network 1: 192.168.0.0/26

Network 2: 192.168.0.64/26

Network 3: 192.168.0.128/26

Network 4: 192.168.0.192/26

Network 5: 192.168.1.0/26

. . .

Network 1023: 192.168.255.128/26

Network 1024: 192.168.255.192/26

To reiterate: The **mask** determines the **range** of addresses contained in a each network. Now that the starting address for each sequential network is known, you can quickly determine the last address in each network.

Network End Address (Broadcast ID)

The last address in a network is always one less than the next network. However, this address is reserved as a broadcast ID, and therefore, cannot be assigned to hosts. Once the broadcast ID and network ID are known, the complete address range is known. **Valid host addresses lie between the network and broadcast IDs.**

Network Bits Host Bits

Network 1: 192.168.0.0/24

Network ID: 192.168.0.0

Broadcast ID: 192.168.0.255

Valid Host Range: 192.168.0.1 - 192.168.0.254

Network 2: 192.168.1.0/24

Network ID: 192.168.1.0

Broadcast ID: 192.168.1.255

Valid Host Range: 192.168.1.1 - 192.168.1.254

Network 3: 192.168.2.0/24

Network ID: 192.168.2.0

Broadcast ID: 192.168.2.255

Valid Host Range: 192.168.2.1 - 192.168.2.254

It is therefore the **network mask** that determines the range of the network. Although there is functionally no difference between a network and a subnet, the **network mask** is more often called the **subnet mask** when networks receive custom masks. The concept of **subnetting** is the process by which the network mask is altered to create more (sub) networks. Each subnetwork is smaller than the network that is subdivided. This results in a smaller network range and consequently, fewer hosts per subnetwork.

Demise of Classful Addressing

The introduction of a default mask distinguished how the **Internet Assigned Numbers Authority** (IANA) allocated public address blocks to companies. Large companies received class A /8 blocks, while mid-to-small size companies received class B /16 and class C /24 blocks. This meant that a large organization receiving a class A address block of 20.0.0.0/8 (20.0.0.0 - 20.255.255.255) had over 16,000,000 public addresses for hosts! A medium-sized company might receive a class B address block of 130.0.0.0/16 (130.0.0.0 - 130.0.255.255), for over 65,000 public host addresses. Finally, a small company would receive a class C address block of 199.0.0.0/24 (199.0.0.0 - 199.0.0.255).

Class	Default Mask	Total Addresses
A	255.0.0.0	16, 777, 216
B	255.255.0.0	65, 536
C	255.255.255.0	256

For most companies, class A address blocks proved far too large for the needs of most organizations. As a result, an overwhelming number of class B and C address blocks were used. Eventually, the demise of classful networking became apparent as class B address blocks all but disappeared.

Classless Addressing in Today's Networks

Classless addressing was introduced as a solution to the inefficiencies of the classful address system—primarily, helping in address allocation. Instead of requiring that all organizations receive public address **blocks** with class-based network masks (/8, /16 or /24), classless addressing proposed the concept of **classless inter-domain routing** (CIDR) and **variable length subnet masks** (VLSM).

CIDR proposes address blocks, regardless of class, can be grouped with relative routing prefixes. For example, each of the /24 networks from 192.168.0.0/24 to 192.168.255.0/24 are contained in the 192.168.0.0/16 network. In routing, this concept is known as **summarization**, or **route aggregation**, where groups of networks are combined whenever possible, to minimize the size of routing tables.

Furthermore, **VLSM** allows owners to apply different subnet masks with groups of addresses contained in a network range. For example, the network 192.168.0.0/24 can be subnetted **variably** to contain various subnetworks. One possible subnet combination is 192.168.0.0/25 and 192.168.0.128/25. Another possible subnet combination is 192.168.0.0/26, 192.168.0.64/26, and 192.168.128.0/25.

For private and public network address blocks alike, classless addressing is very useful and has become the standard for how networks are assigned today. Modern routing protocols rely on classless addressing since networks of varied sizes are common in the service provider setting.

Subnetting

What is a Subnet?

For all intents and purposes, a subnet is the same thing as a network. With subnetting, it is similar to taking a cake and cutting it into smaller pieces. While not quite as delicious as a cake, subnetting is useful in scenarios where you want to:

- **Reduce** the amount of **broadcast traffic** on the local area network
- **Divide** up a larger network of users in **smaller networks** relating to location or department
- **Create dedicated networks** / **links** between routers for just about any IP-related concept (DHCP pools, site-to-site VPNs, VRRP)
- **Master** the most important **routing protocols** used in today's enterprise / ISP networks
- **Rely** on **route summarization** for faster, more efficient routing

There are really two questions to ask when subnetting a network. First, how many **subnets** are desired? Second, how many **hosts** are needed per subnet? Being able to calculate the number of subnets and hosts within a network hinges on two **formulas** that relate back to the **subnet mask**.

Revisiting the Network Mask

In the lab activity "Determine Host Network," we used the network mask to find the network of various hosts. Recall that the network mask is broken up into two parts: **network bits** and **host bits**. The network bits are represented by 1's (on the left side of the mask), while the host bits are represented by 0's (on the right side of the mask). Classful networking introduced the default network masks for class A, B, and C networks.

Network Bits Host Bits

255.0.0.0 = 11111111.00000000.00000000.00000000 /8 Mask

255.255.0.0 = 11111111.11111111.00000000.00000000 /16 Mask

255.255.255.0 = 11111111.11111111.11111111.00000000 /24 Mask

There is a special pattern unique to default masks: The "255" octets contain **all network bits**, while the "0" octets contain **all host bits**. The boundary between these network and host bits is called the **octet boundary**. As the boundary between network and host bits moves right, the **length** of the network mask **increases**, creating **subnets** in the process. Technically speaking, subnetting occurs as **host bits** are **converted** to **network bits**.

Let us practice subnetting and learn from the following example:

Network Bits   Host Bits

Host IP 192.168.0.129 = 11000000.10101000.00000000.10000001

Net Mask 255.255.255.0 = 11111111.11111111.11111111.00000000

Net ID 192.168.0.0 = 11000000.10101000.00000000.00000000

Mask 3rd Octet =	128	64	32	16	8	4	2	1
	1	1	1	1	1	1	1	1
Mask 4th Octet =	128	64	32	16	8	4	2	1
	0	0	0	0	0	0	0	0

Notice we call it a "network mask". After we convert host bits to network bits, we'll call it a "subnet mask". We have also performed ANDing to determine the Net ID (which is another way to represent the **network** to which a **host** belongs).

A quick look at previous digram indicates that the **rightmost network bit** of the mask lies in the **3rd octet** and has a value of 1. This is ultimately due to the 255.255.255.0 (/24) mask. Let's now see what happens to the host 192.168.0.129 as we apply a different network mask:

Network Bits   Host Bits

Host IP 192.168.0.129 = 11000000.10101000.00000000.10000001

Subnet Mask 255.255.255.128 = 11111111.11111111.11111111.10000000

Net ID 192.168.0.128 = 11000000.10101000.00000000.10000000

Mask 3rd Octet =	128	64	32	16	8	4	2	1
	1	1	1	1	1	1	1	1
Mask 4th Octet =	128	64	32	16	8	4	2	1
	1	0	0	0	0	0	0	0

After converting a single host bit to a network bit, the network mask changes to 255.255.255.128. Likewise, the network to which host 192.168.0.129 belongs also changes from 192.168.0.0/24 to 192.168.0.128/25. Now that we have seen how the network mask affects the (sub)network ID, let's see how we can practically create a pre-determined number of subnets given any network mask.

Practice

Calculate Subnets

As previously mentioned, subnetting any network, such as 192.168.0.0/24, means we must **convert host bits** contained in the mask to **network bits**. This is accomplished by moving the boundary from its place in the default mask, any number of places to the **right**. Let us examine the previous example again:

	Network Bits	Host Bits	
Network ID 192.168.0.0 =	11000000.10101000.00000000.	00000000	
Default Mask 255.255.255.0 =	11111111.11111111.11111111.	00000000	/24
Subnet Mask 255.255.255.128 =	11111111.11111111.11111111.1	0000000	/25

Just how many subnets were created by borrowing one host bit and converting it to network bits? The **formula** to calculate how many subnets are created follows, where variable “n” = the number of host bits converted to subnet bits:

2^n = number of newly created subnets.

In the previous Class C example, a single host bit was converted to a network bit. We’ll use the formula with this example to find the number of newly created subnets.

Network 192.168.0.0/24 is subnetting using /25 masks.

2^n = number of created subnets (where n = number of borrowed host bits)

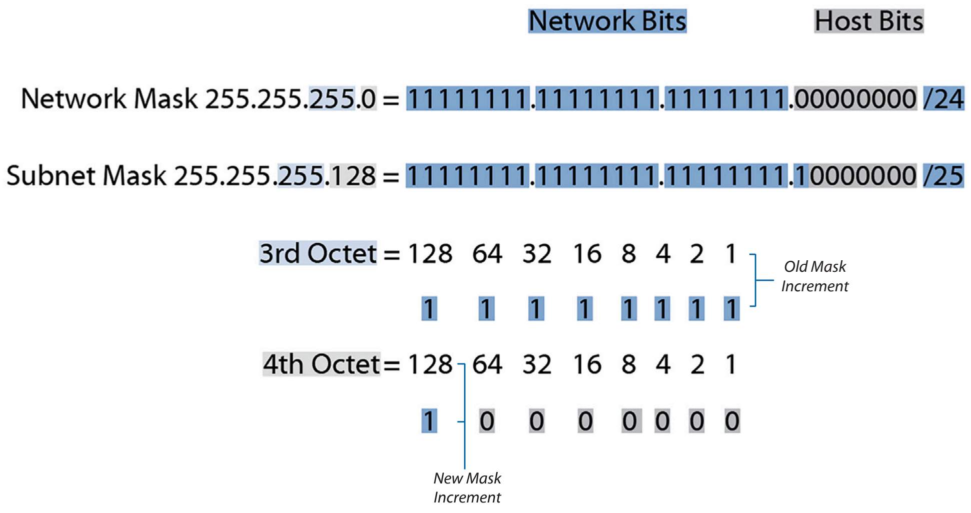
2^(subnetwork bits - original network bits) = number of created subnets

2^(25-24) = number of created subnets

2^(1) = number of created subnets

2 = number of created subnets

What then is the range for each of the newly created subnetworks? You can find the new subnetwork ranges by applying the same steps for determining any network range.—just look for the rightmost network bit. Because the rightmost subnetwork bit is located in the fourth octet, that is where the first subnetwork begins:


Network Bits
Host Bits
Network Mask 255.255.255.0 = 11111111.11111111.11111111.00000000 /24
Subnet Mask 255.255.255.128 = 11111111.11111111.11111111.10000000 /25
3rd Octet = 128 64 32 16 8 4 2 1
1 1 1 1 1 1 1 1
Old Mask Increment
4th Octet = 128 64 32 16 8 4 2 1
1 0 0 0 0 0 0 0
New Mask Increment

The **value** assigned to the **rightmost subnet bit** is 128. The subnetworks grow in increments of 128 (in the fourth octet). The first subnetwork begins at 192.168.0.**0**, while the second, starts at 192.168.0.**128**. Compared to a /24 network, which grows in intervals of 1 in the third octet, /25 networks grow by intervals of 128 in the fourth octet.

Also recall that each network ends just before the address at which the next network begins. So the range of the subnetwork 192.168.0.0/25 is 192.168.0.0 - 192.168.0.127. The range of the next subnetwork 192.168.0.128/25 is 192.168.0.128 - 192.168.0.255, which is just before the next network, 192.168.1.0/24 (assuming a default class C mask).

Calculate Hosts

Often, you will you need to subdivide a network based on a minimum number of hosts per subnetwork. The fastest way to calculate the total number of valid hosts can be assigned to a subnetwork is to use the following formula, where variable "x" = the number of host bits in the subnet mask:

(2^x) - 2 = number of valid hosts per subnet

Let's revisit the same Class C example from before, 192.168.0.0/24. How many valid hosts can exist on each subnet when a /25 mask is applied to 192.168.0.0/24?

Network 192.168.0.0/24 is subnetting using /25 masks.

2^x = number of valid hosts per subnet

2^ (32 - subnet mask bits) - 2 = number of valid hosts per subnet

2^ (32 - 25) -2 = number of valid hosts per subnet

2^ (7) - 2 = number of valid hosts per subnet

128 - 2 = number of valid hosts per subnet

126 = number of valid hosts per subnet

4 network allows for a maximum of 126 **valid host addresses** per subnetwork. This is because the first and last address contained in every network are reserved for the network and broadcast ID.

Subnetting Practice

Below are a few more lab activities to practice your subnetting skills. Feel free to work with your instructor to set up more examples. Although subnet calculators take all of the work out of subnetting, it is important to conceptualize subnetting through practice. You can even use subnet calculators to create more subnetting examples and double-check your answers. Remember, practice makes perfect!

VLSM for Service Providers

Suppose you have been allocated a **block** of public network addresses: **100.0.0.0/20**. While it is tempting to begin handing out addresses to users, the first thing to do is determine the **entire range of network addresses** contained in your **block**. After identifying the network range, you can begin to subnet the block of public addresses as you re-allocate addresses to your customers.

Network Block = 100.0.0.0/20

Subnet Mask = 255.255.240.0

CIDR Mask = 11111111.11111111.11110000.00000000 /20

3rd Octet =	128	64	32	16	8	4	2	1
	1	1	1	1	0	0	0	0
	Network Bits				Host Bits			

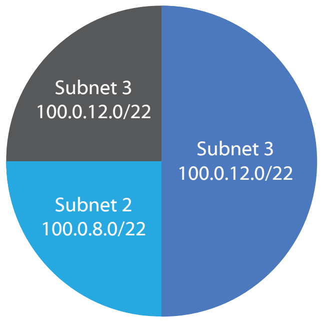
Once again, the **rightmost** network bit in the /20 mask is used as reference. Since the rightmost network bit has a value of 16 (in the third octet), add 16 to the third octet of the block 100.0.**0**.0/20. Because the next block of addresses starts at 100.0.**16**.0, the range of addresses contained in the block 100.0.**0**.0/20 is 100.0.**0**.0 - 100.0.**15**.255.

According to **VLSM**, the address block can be subnetted in any combination, so long as the new networks do not overlap addresses outside the block address range (beyond 100.0.15.255). As a general rule with VLSM, new subnet masks applied to the network block cannot be greater (e.g., /18, /19) than the entire block itself (e.g., /20).

The example below illustrates how the block 100.0.0.0/20 is variably subnetted into 3 smaller networks. The octet containing the rightmost subnet bit is in **bold**.

VLSM Example 1
100.0.0.0/20 is
Variably Subnetted

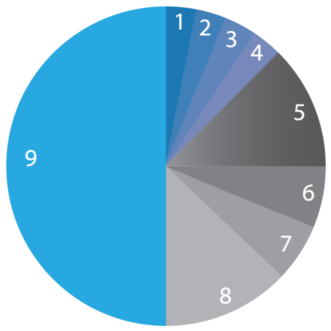
Subnet 1: 100.**0**.0/21
Subnet 2: 100.**8**.0/22
Subnet 3: 100.**12**.0/22



The next example takes the same address block 100.0.0.0/20 and variably subnets it into 9 smaller networks. Once again, the octet containing the rightmost subnet bit is in **bold**.

VLSM Example 1
100.0.0.0/20 is
Variably Subnetted

Subnet 1: 100.0.0.**0**/25
Subnet 2: 100.0.0.**128**/25
Subnet 3: 100.0.1.**0**/25
Subnet 4: 100.0.1.**128**/25
Subnet 5: 100.0.**2**.0/23
Subnet 6: 100.0.**4**.0/24
Subnet 7: 100.0.**5**.0/24
Subnet 8: 100.0.**6**.0/23
Subnet 9: 100.0.**8**.0/21



As the examples have shown, **variable length subnet masks** allows for custom address allocation based on need. It is a flexible system used by all service providers. Just remember that when dividing up address blocks, make sure that the subnet mask does not cause the new subnet ranges to overlap beyond the starting address pool.

Summarization

Route summarization, also known as **route aggregation**, is the process by which a **CIDR prefix** is appended to a larger network group containing multiple networks for which a route can be selected. Instead of advertising hundreds or thousands of addresses, a router can summarize a group of known networks based on a network range, which greatly improves the efficiency and performance of the router. The following example shows a group of /24 networks contained in the private network range 10.0.0.0/8.

Local networks to be advertised in the routing protocol:

- 10.20.0.0/24
- 10.20.30.0/24
- 10.20.31.0/24
- 10.20.46.0/24
- 10.20.100.0/24
- 10.20.110.0/24
- 10.21.0.0/24

While the 10.0.0.0/8 network could be used to summarize all the networks, it is somewhat “messy” in that it includes a very broad range of addresses which do not actually exist on the network (e.g., 10.1.1.0/24, 10.2.2.0/24). Instead, a more specific summary should be used. The method for determining the **supernetwork** to be advertised is simple:

1. Line up the networks
2. **Convert** to **binary**
3. Perform **AND** across the octets
4. As soon as a column of bits differs, stop matching and carry zeros across.
5. Convert back from binary to decimal to indicate the advertised network range
6. Count the number of matching CIDR value is retrieved by counting from left to right the number of bits that matched before stopping.

(Decimal)	(Binary)
10.20.0.0	00001010.00010100.00000000.00000000
10.20.30.0	00001010.00010100.00011110.00000000
10.20.31.0	00001010.00010100.00011111.00000000
10.20.46.0	00001010.00010100.00101110.00000000
10.20.100.0	00001010.00010100.01100100.00000000
10.20.110.0	00001010.00010100.01101110.00000000
10.21.0.0	00001010.00010101.00000000.00000000
10.20.0.0	00001010.00010100.00000000.00000000

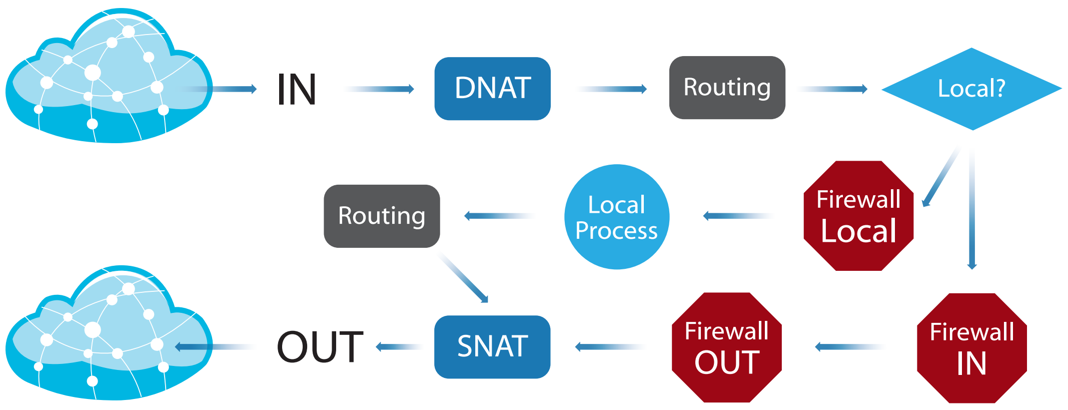
00001010.0001010 = 15 matching bits (network bits)

Single route to be advertised is 10.20.0.0/15.

V. Routing

Routing builds on the **subnetting** fundamentals taught in the previous chapter.

Although the functions of a network router vary greatly, **packet routing** is its primary purpose. Routers perform actions on received packets, based on network decisions and firewall rules. The following diagram describes the flow of packets as they arrive on a router interface, pass through a router, and leave a router interface.



At each of the black "routing" boxes in the previous figure, the router consults its **routing table**. The routing table is the database containing suggested routes to different networks; each suggested route contains the following information:

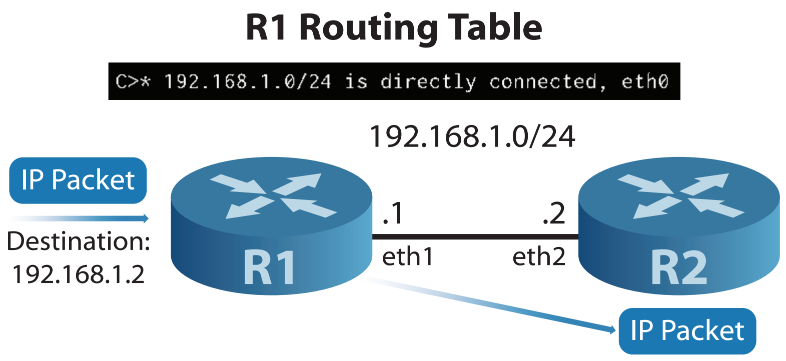
- **Network ID**, which includes the **subnet mask** (**prefix**)
- **Cost/Metric** associated with the path/route type
- **Next Hop**, or the address of the next **gateway** to reach the destination address

The routing table may contain other information relevant to the target network path, such as **interface state** (e.g., up/down) or **quality of service**. In service provider networks, it is common to have redundant links to the same network for failover. In such cases, the routing table may contain **multiple routes** to the **same network**. The chosen route is selected according to **administrative distance** (and metric if the administrative distances are equal). The following table lists the associated with different routing protocols. Routes with lower administrative distances are considered more "reliable" as compared to routes with higher administrative distances.

Route Type	Administrative Distance
Connected	0
Static	1
eBGP	20
OSPF	110
RIP	120
iBGP	200

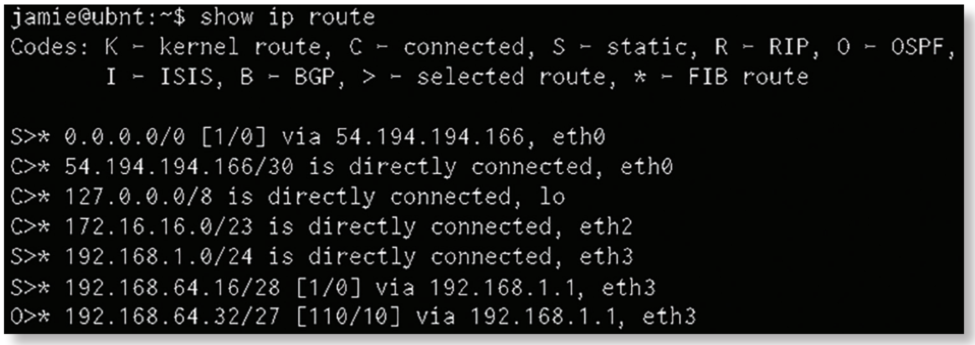
Connected & Static Routes

Routes are simply paths to networks that a router can select when forwarding packets on the way to their final destination. A router whose interfaces are configured with IP addresses is said to be **directly connected** to a particular network. In routing, any connected network interface is called a **link**. Once configured, the networks appear as **connected routes**, and are automatically added to the routing table.



A **static route** is a route that is manually added to the routing table. A common instance of static routing is when a **default route** (sometimes called the **gateway of last resort**) is required. The default route **0.0.0.0/0** is a kind of last resort gateway route since it matches all the destination network of all incoming packets. A **next hop** address is assigned to the default route where packets can then be forwarded. This default route serves a similar purpose to that of the **gateway address** assigned to a network **host**.

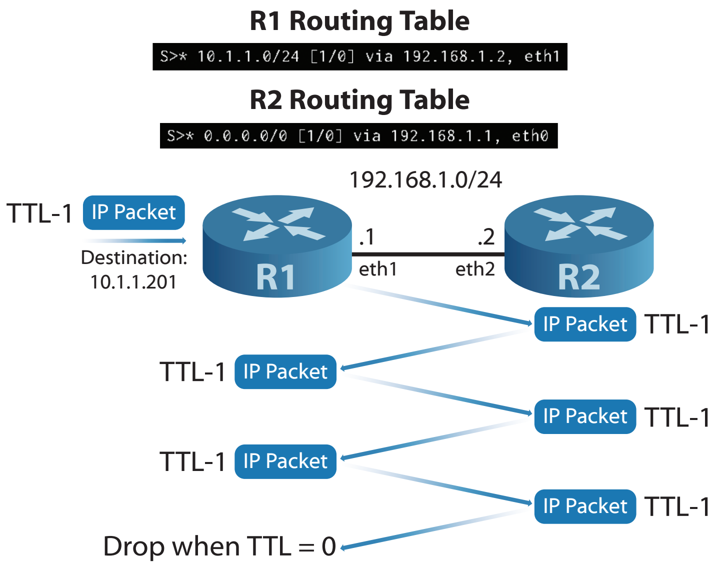
Although the static route 0.0.0.0/0 matches all packets, it is not always the chosen route. The foremost reason for this is due to its **prefix**. The more specific the prefix, the more "desirable" the route, provided the destination address of the packet is part of the target network. In the following example, a router receives a packet destined for **192.168.64.44**. Although the routing table contains an entry for **0.0.0.0/0**, it prefers to forward the packet to an OSPF neighbor since the prefix is more specific (longer) and the target network routing entry (**192.168.64.32/27**) matches the packet's destination.



```
jamie@ubnt:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

S>* 0.0.0.0/0 [1/0] via 54.194.194.166, eth0
C>* 54.194.194.166/30 is directly connected, eth0
C>* 127.0.0.0/8 is directly connected, lo
C>* 172.16.16.0/23 is directly connected, eth2
S>* 192.168.1.0/24 is directly connected, eth3
S>* 192.168.64.16/28 [1/0] via 192.168.1.1, eth3
O>* 192.168.64.32/27 [110/10] via 192.168.1.1, eth3
```

Be careful when creating a static route to a non-local network that is unreachable, since this can cause routing loops in the network, as seen in the figure below. Although a Time-to-Live (TTL) marker contained in the IP packet counts down the number of hops it can take before being discarded, routing loops can cause confusion and unnecessary traffic. Using the `traceroute` tool, you can identify the problematic "hops" and examine their routing tables to identify the cause of the loop.



R1 Routing Table

```
S>* 10.1.1.0/24 [1/0] via 192.168.1.2, eth1
```

R2 Routing Table

```
S>* 0.0.0.0/0 [1/0] via 192.168.1.1, eth0
```

Dynamic Routing

Compared to static routing, **dynamic routing protocols** are used to automatically create, maintain and update the routing table. In a fully static network, routes must be individually built on each router in order for the network to reach full connectivity. Needless to say, this can be quite time consuming. If even a single router is added or removed from the network, you have to update every router individually! Dynamic routing protocols work differently, with useful characteristics for the service provider setting, including:

- Quick, **automatic** updates whenever network topology changes.
- Routers systematically choose the **best path** based on **metrics**.
- Capable of **load-balancing** between links.

Dynamic routing protocols can be grouped into two different types. The first group is **distance-vector** protocol, which calculates the **cost/metric** (best possible path) to a remote network based on its **distance/hops** to the router. The **Routing Information Protocol** (RIP) is an example of a distance-vector protocol that chooses the route with the fewest number of **hops** to the target network. With RIP, routers send out **periodic updates** advertising the networks to which they are directly connected. Routers that receive the announcement will re-advertise the route to the remote network, adding a single hop (up to 15 times). This has led to the description for RIP, "**routing by rumor**," a behavior that illustrates the unreliability and instability of the protocol. It is not often seen in modern networks, but nonetheless, is supported in EdgeOS for backward compatibility in **legacy** networks.

The second group of dynamic protocols is classified as **link-state** protocol. Link-state routing considers different metrics in finding the best possible path, primarily **link speed** (e.g., 1Gbps preferred to 100/10Mbps). In the link-state topology, the behavior of participating routers is "**tell the world about my neighbors**." Each router has absolute certainty about the existence of the networks it announces, since each router is directly connected to network it is advertising. To conceptualize link-state routing, each router constructs a map of the entire, link-state network, **calculating routes** to each area. Though multiple routes may exist to the same area, only the best route is selected and added to the routing table. Link-state protocols such as OSPF are common in the service provider and enterprise setting. They allow groups of routers to quickly reach **convergence**, where the complete, up-to-date topology is shared between all participating routers. As the topology of an OSPF network changes (for example when a router is added or removed) routers pass link-state updates (also known as **link-state advertisements**) advising about the new link-state. Once the OSPF routers have decided on a finalized topology table (also known as the **link-state database**), the network has reached convergence. After calculating the shortest path to each of the OSPF networks, the router adds the shortest possible path to its routing table.

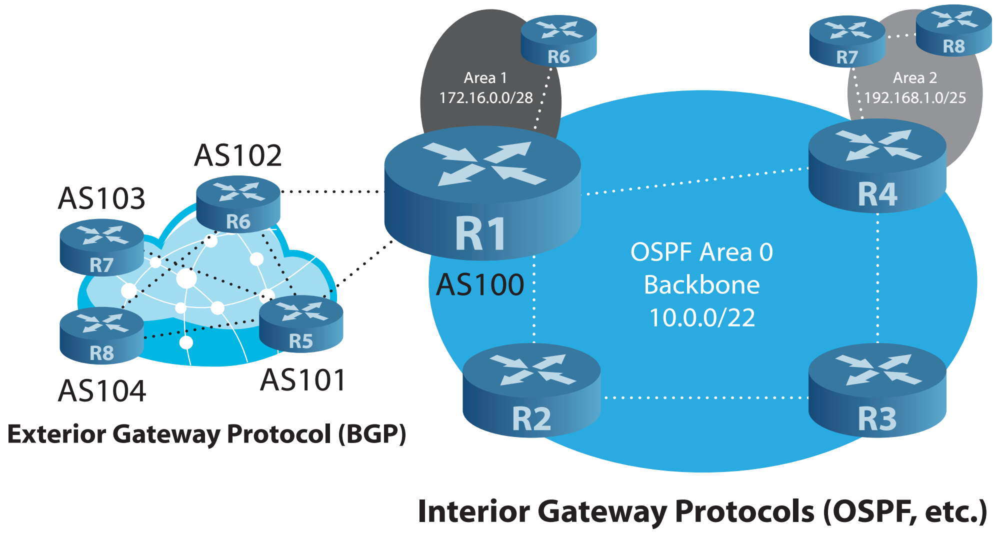
Although dynamic routing protocols differ in implementation, each can be categorized as either an **Interior Gateway Protocol** (IGP) or **Exterior Gateway Protocol** (EGP). The main difference between the two types is that an IGP seeks to exchange routing information between routers contained in the same **Autonomous System** (AS), while an EGP is used to route between autonomous systems. **Autonomous systems** (ASes) are groups of routers under solitary control, like a service provider or large enterprise network.

Open Shortest Path First

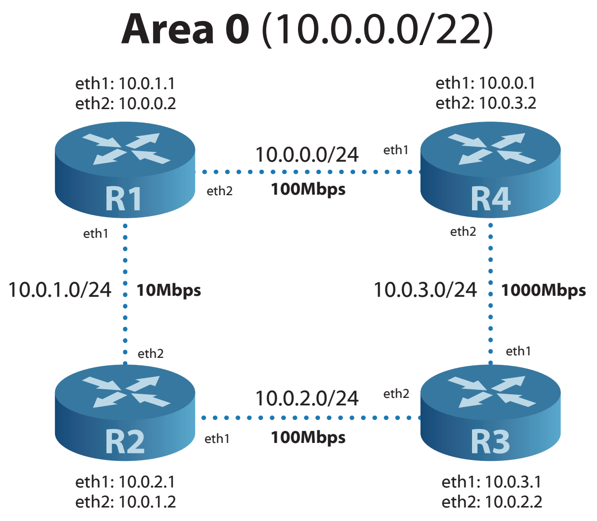
OSPF Overview

Open Shortest Path First (OSPF) is perhaps the most common **interior gateway routing protocol**, having support for VLSM, IPv4/IPv6 and interoperability with today's routers. OSPF is able to achieve fast convergence through creation of **areas** while typical **link-state updates** are contained to a **single area**. The default area of the OSPF network is considered the **backbone**. It is expected that all other areas directly connect to the backbone (**Area 0**) via **area border routers**.

Areas allow for network **summarization**, which vastly improves network efficiency. Whenever a link **flaps** (meaning a network goes offline then back online) the OSPF network begins to **flood** with **updates**. Links that flap often can lead to instability in the OSPF network. By confining groups of networks to individual areas (e.g., Area 0 and 10.0.0.0/22, remote areas are largely unaffected by flapping. **OSPF border routers** are responsible for performing **route summarization** between **OSPF areas**.



Area	Networks	Area Routers	Route Summary	Border Routers
0	10.0.0.0/24 10.0.1.0/24 10.0.2.0/24 10.0.3.0/24	R1 R2 R3 R4	10.0.0.0/22	R1 R4
1	172.16.0.0/30 172.16.0.4/30 172.16.0.8/30 172.16.0.12/30	R1 R6	172.16.0.0/28	R1
2	192.168.1.0/27 192.168.1.32/27 192.168.1.64/27 192.168.1.96/27	R4 R7 R8	192.168.1.0/25	R4



OSPF Area Details

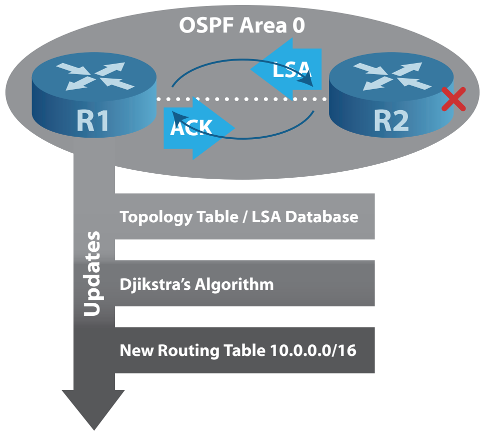
Routers in the same area receive a common area ID, such as 0.0.0.0 (in the case of Area 0). Two or more **OSPF routers** connected on the **same LAN segment** are said to share an **OSPF link** in a **broadcast network**. If certain parameters match, the "**OSPF neighbors**" can form **adjacencies** to begin sharing link-state updates about the network topology. Adjacencies occur as a result of matching **hello packets** being sent out and received on neighbor OSPF routers. Several parameters contained in the hello packet must match in order for OSPF routers to form an adjacency:

- **Area** OSPF neighbors share a common OSPF **area ID** (e.g., 0.0.0.0) as well as a common network **link** on which the interfaces may communicate (e.g., 10.0.0.0/24).
- **Timers** The **Hello Interval** relates the time at which hello packets are sent across the OSPF link; the **Dead Interval** relates the time after which a link is considered down if no hello packet is received.
- **Authentication** Authentication helps secure the OSPF process between authorized routers and is defined when initially creating the **OSPF area**. If authentication is chosen, assign a matching key to **OSPF interfaces** on a common link (that will form an adjacency).

After forming a neighborship, the routers elect an interface in the adjacency to serve as the **designated router** (DR), as well as another interface as the **backup designated router** (BDR). After which, routers can begin to share **link-state advertisements** (LSAs), important information contained in each router's **OSPF database**, including:

- **OSPF networks** and prefixes (e.g., R1 shares 10.0.0.0/24 and 10.0.1.0/24)
- **Metrics/costs** (e.g., **default** costs based on **link speed**)
- **Link state** (e.g., up/down)
- **Neighbors list** (e.g., R1 has R2 and R4 as neighbors)

A **link-state database** (LSDB) containing all of the LSAs is stored on all the routers in the OSPF area. After reaching **convergence**, the LSDB (or **topology map**) is finally finished. The topology map is identical across all routers in the OSPF area after convergence is reached. Each OSPF router then calculates the best possible path to remote networks using Djikstra's Algorithm. OSPF sums the cost of each link along the path to the remote network (default cost is based on link speed). Finally, the OSPF protocol selects the best route to the remote network and forwards it to the **routing table**. When properly configured, the OSPF protocol is rapid, reliable, and robust, making it the ideal choice for IGP routing today.

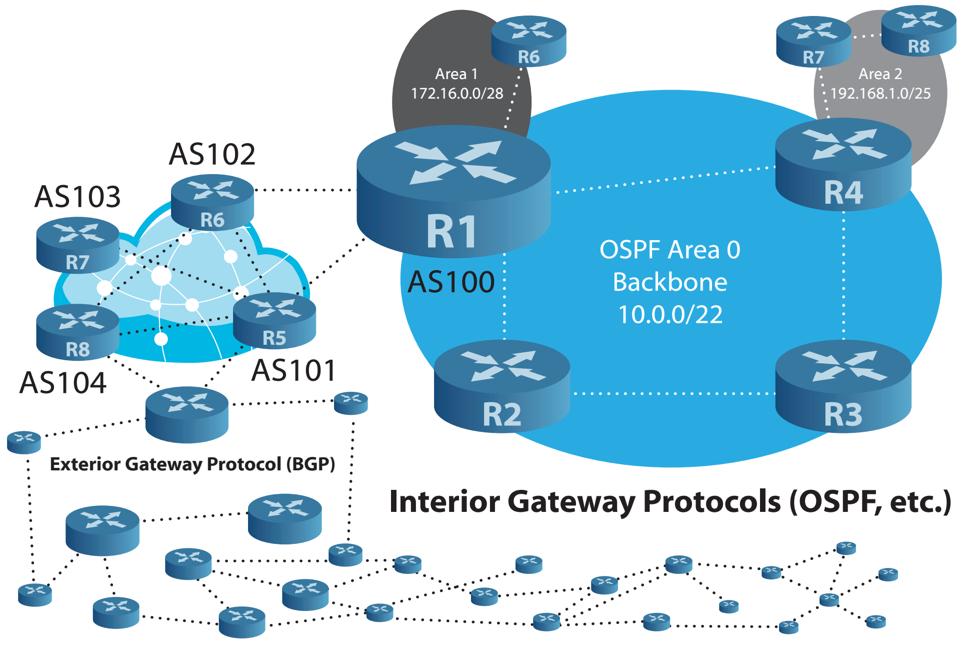


Border Gateway Protocol

As an EGP, the Border Gateway Protocol (BGP) is responsible for interconnecting ASes so that packets can be routed across the Internet. There are three types of AS including:

- **Multihomed** an AS connected to multiple other AS
- **Stub** an AS connected to a single AS
- **Transit** an AS used by another AS to reach a different AS

Routers located on the border of the AS are called edge routers, typically running **external BGP** (eBGP) to advertise routes. Routers that exchange information about routes to networks through other ASes are called **BGP peers**. Recall from *Lab: Display Routing Tables* that the public Internet router ran BGP and as a result, contained a very large number of routes. The greater the number of peers, the higher the RAM requirements. For this reason, the EdgeRouter-8 and EdgeRouter-8-PRO are recommended over the EdgeRouter-Lite in settings where BGP is required.



VI. Services & Security

Whether working in a service provider or enterprise setting, you should familiarize yourself with a number of commonly used network **protocols** and **services**. Later in the chapter, you will complete lab activities designed to emulate real-world situations for which the router would require a specific configuration.

The *Wizards* tab in EdgeOS allows users to quickly configure a variety of **routing**, **service** and **security** settings through the **web GUI**. Here are just a few of the feature wizards included on the latest version of EdgeOS:

- Static & OSPF Routing
- NAT & Port Forwarding
- DHCP Server & Static Mapping
- DNS Forwarding & DynDNS
- PPTP Remote & IPSec Site-to-Site Tunnels
- TCP MSS Clamping & UPnP
- Load Balancing
- WAN+2LAN (SOHO Config)

Dynamic Host Configuration Protocol

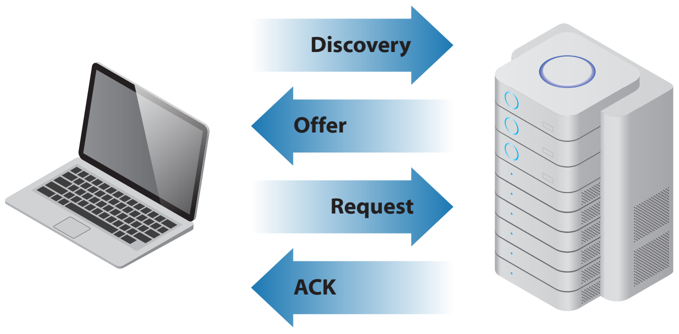
The **Dynamic Host Configuration Protocol** (DHCP) is a method by which network **nodes** can automatically receive IP information to begin passing traffic on the network. At **layer-2**, the process occurs in the following manner:

1. DHCP Client **broadcasts** to **DISCOVER** a server
2. DHCP Server **unicast** replies with DHCP **OFFER**
3. DHCP Client **broadcasts** the **REQUEST**
4. DHCP Server **unicast** replies with **ACK**

The DHCP OFFER (also called the **DHCP lease**) may contain a number of **options**, which correspond to different network information. Typical options include:

- IP Address
- Network Mask
- Router
- DNS Server

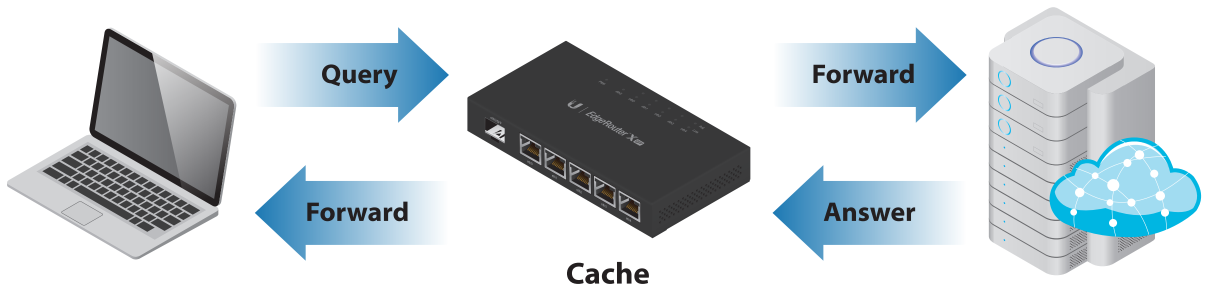
DHCP Option 43 is a vendor-specific option used with access points, for example, UniFi for adoption en-masse. EdgeOS DHCP Server can be configured with Option 43.



Domain Name System

A **Domain Name System** (DNS) server is responsible for translating network hosts from their **hostname** (e.g., new-york-office-router) or **fully-qualified domain name** (e.g., store. ubnt.com) to an IP address. DNS is commonly compared to a phone book, where people's names are associated with a phone number, which can then be used to call the person. DNS is used in private and public networks alike. The foremost example is the Internet, where global, authoritative DNS servers map millions of IP addresses to domain names (which just like names, are easier for people to remember).

A **DNS forwarder** is common on private networks where a gateway located on a LAN and/ or WAN boundary may receive multiple, distinct DNS queries for the same domain name. Once the DNS address is retrieved, the forwarder will cache it for future requests. This can help minimize network **latency** experienced by hosts that require DNS.



Firewalls

Given their design, function, and location on networks, routers are well-suited to run firewalls. Firewalls that keep track of network connections are said to be **stateful**, while those that do not and act on each individual packet are called **stateless**.

EdgeOS supports both firewall types through **rulesets** that precede and/or follow routing decisions. The ruleset applies a default action to every packet, including:

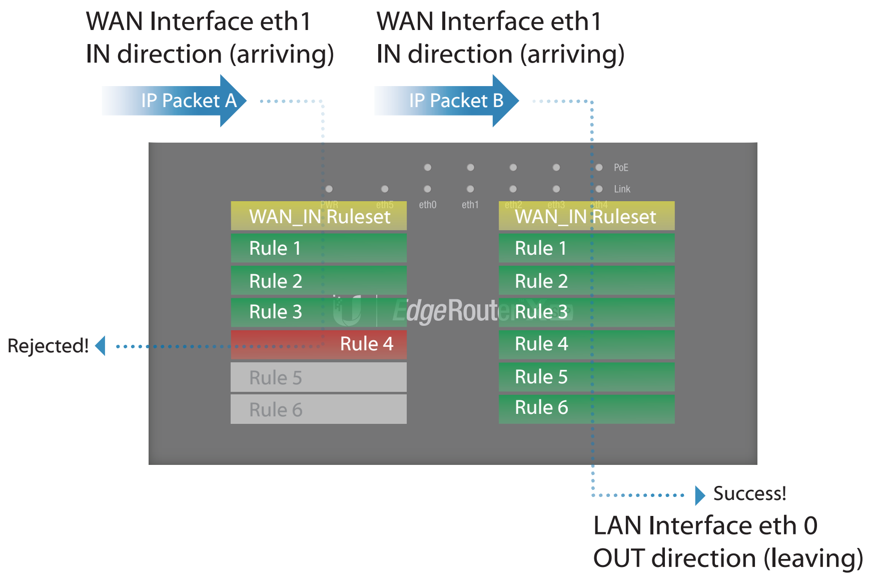
- **Drop**, where the received packet is dropped and not replied
- **Reject**, where the received packet is refused but replied
- **Accept**, where the received packet is received

Once the ruleset is created, it can be enhanced or overruled with special **rules** that are triggered by the following parameters:

- **Protocol**, only acting on a select packet types (e.g., TCP, UDP)
- **State**, stateful firewall means the type of connection associated with the packet (e.g., new, established, related, or invalid)
- **Source**
- **Destination**
- **Time**, for schedule-based firewall rules

The ruleset can be applied to one or more interfaces on the EdgeRouter. It is important to specify the **direction** for which packets should receive the rules. There are three directions a packet can take with respect to an interface:

- **In** packets *arriving* on the interface (also ingress, inbound)
- **Out** packets *leaving* the interface (also egress, outbound)
- **Local** packets *destined* for the interface of the router (the router itself)



Tunnels & VPN

Tunneling is simply a means by which data is encapsulated inside of another protocol, serving as the basis for **Virtual Private Networks** (VPNs). VPNs extend a LAN beyond its WAN border, as in the case of remote network access. The **Point-to-Point Tunneling Protocol** (PPTP) is among the simplest types of **remote access VPNs** to set up and troubleshoot. **Hosts** with a **PPTP client** can establish connection to a **PPTP server** running on the EdgeRouter, provided they have the right credentials.

Another useful tunneling technology is the **Point-to-Point Protocol over Ethernet** (PPPoE). **PPPoE** is often used by service providers to let **subscribers** establish a connection and **authenticate** prior to beginning an **Internet session**. Whether at the customer premise or on location at a carrier site, EdgeRouter supports both PPPoE client and server mode. In this way, PPPoE clients connect to a PPPoE server, which contacts a **RADIUS** server to authenticate the PPPoE client (subscribers) against a database, using one of many different methods. The EdgeRouter also supports **local authentication**, where incoming PPPoE client connections are matched against the local users. An overview of the PPPoE process is described below:

1. End-to-end, **physical link** is created between server and client.
2. Logical **session window** is tracked for connecting client.
3. User identity is **authenticated** and authorized by servers.
4. PPPoE Server Pool **leases** IP address to client.

A. Glossary

- **Adjacency:** Formed whenever two or more neighbor routers share a common OSPF link; necessary to share LSAs.
- **Address Resolution Protocol:** A network protocol used to discover and cache the MAC address of a local host (IP to MAC mapping).
- **Administrative Distance (AD):** The reliability or trustworthiness of a route (lower = more reliable).
- **Advertisement:** See LSA.
- **Aggregation:** See Summarization.
- **Application Layer:** In the OSI Model, the layer at which users typically interact (e.g., HTTP application protocol corresponds to web browser program).
- **Area:** A group of routers that share link-state advertisements, typically participating in a summarized network.
- **Area Border Router (ABR):** Router with interfaces connecting two or more areas.
- **ARP Cache:** A host's cache containing MAC addresses mapped to IP addresses.
- **Autonomous System (AS):** A group of routers under solitary control by a single domain or company.
- **Autonomous System Border Router (ASBR):** The router connecting an AS to other ASes.
- **Backbone:** The OSPF area connecting to all other areas; always area 0 (0.0.0.0).
- **Border Gateway Protocol:** A path vector protocol widely used to move packets between ASes and route packets across the Internet.
- **Broadcast:** One-to-all network communication, intended to reach all link-local hosts.
- **Broadcast Domain:** The area of a local network where a broadcast message reaches.
- **Collision Domain:** A transmission line or medium where collisions between nodes are possible (e.g., wireless radio, CAT5 Ethernet).
- **Data Link:** Refers to local network communication, where MAC addresses are referenced to pass data on/across the same network.

- **Dead Timers:** The time that passes without Hello packets arriving on an OSPF interface, at which point the router determines that the link to the neighbor is down.
- **Discovery:** A broadcast protocol specific to Ubiquiti that finds devices on the link-local network.
- **Exterior Gateway Protocol (EGP):** A routing protocol used to move traffic between different autonomous systems.
- **Gateway:** A layer-3 device (typically a router) that forwards packets destined to a different network from the source host.
- **Host:** A network node that has a logical address.
- **Interface:** The software and hardware components that together allow a device to communicate on a network.
- **L3 Switch:** A special kind of switch that has routing capabilities.
- **Link-local:** Refers to the Layer 2 network.
- **Local Area Network:** A group of devices that share a common network ID, similar to the broadcast domain.
- **MAC Table:** The table consulted by a switch when deciding whether a frame will be forwarded, filtered or flooded.
- **Multicast:** One-to-some network communication, seen in dynamic routing protocols (OSPF), video streaming protocols (IPTV), etc.
- **OSI Model:** A 7-layer model that outlines the network protocols involved in communication.
- **Point-to-Point Protocol (PPP):** A popular, layer-2 transport mechanism for establishing and authenticating subscribers in service provider networks.
- **Router:** A layer-3 device capable of moving packets between networks.
- **Routing Table:** The table consulted by a router when deciding where to send a packet.
- **Switch:** A layer-2 device used to moving traffic (frames) between link-local hosts.
- **TCP/IP Model:** A 4-layer model that outlines the network protocols involved in end-to-end communication, as with the Internet.
- **Unicast:** One-to-one network communication.
- **Wide Area Network (WAN):** Represents the network area, outside of the LAN.

B. Appendices

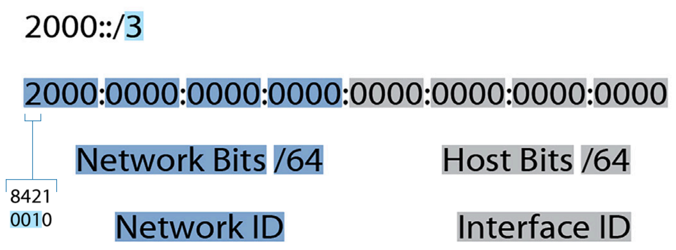
Internet Protocol version 6

IPv4 Shortcomings

Although IPv4 is the standard for how today's Internet works, it is currently being superseded by IPv6, a better, scalable protocol. The rapid transformation of the Internet revealed a number of IPv4's shortcomings:

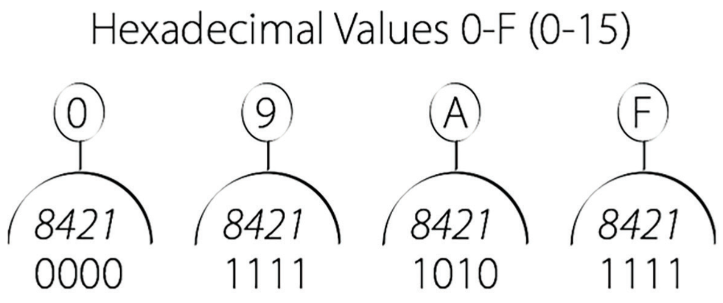
- Rapid **exhaustion** of public, routable **Internet addresses**
- Extremely large, **inefficient routing tables** on Internet routers
- Wasteful traffic due to unnecessary **broadcast** messages
- **Complex** network auto-configuration
- **Lack of IPSec** (end-to-end encryption) and **QoS** at Layer-3

IPv6 Introduction



Although similarities exist between the two Internet Protocol versions, IPv6 features a number of enhancements in addressing format:

- **128-bit** addresses instead of 32-bit, half being network bits (network ID), while the other half, host bits (interface ID).
- **Hexadecimal** addressing, where each character (0-F) represents a value (0-15); in bits (0000-1111). Each character in the IPv6 address now represents 4 bits (A = 1010, 9 = 1111).
- A single interface often receives **multiple addresses**, for local and non-local communication.



Types of Communication & Addresses

There are three **types** of communication (no broadcasts) and a number of different *addresses* to recognize in the IPv6 world.

- **Unicast**, one-to-one messages. *Global unicast addresses* are most similar to a public, routable IPv4 address and start with *2000::/3*.
- **Multicast**, one-to-many messages. Like in IPv4 world, a packet with multicast destination address is sent to all interfaces listening to the multicast address. *Multicast addresses* start with *FF00::/8*.
- **Anycast**, similar to multicast since packet is destined to multiple interfaces/devices, but only used with routers, anycast packet reaches only a device (closest, routable interface).
- *Link-local addresses* are based on concept of self-assigned IP addresses (like when no DHCP server for exists in IPv4 world, 169.254...), starts with FE80::/10. **EUI-64** is a method for configuring link-local addresses by placing **FFFE** between 3rd/4th hexadecimal places of the original MAC address and inserting it into the interface ID

  ex. 00:11:22:33:44:55 >>>> 0011:22FF:FE33:4455

  MAC IPv6

Here are three other examples of IPv6 addresses to recognize:

- **0:0:0:0:0:0:0:0**, or, ::, belonging to a host before an address is assigned.
- **0:0:0:0:0:0:0:1**, or, ::1, similar to 127.0.0.1 in IPv4, a loopback address.
- **0:0:0:0:192:168:1:1**, how an IPv4 address is expressed in a network with both IPv4 and IPv6 running.

Subnetting & IPv6

Suppose you have a block 2000::/56

```
Start Address: 2000:0000:0000:0000:0000:0000:0000:0000
End Address: 2000:0000:0000:00FF:FFFF:FFFF:FFFF:FFFF
```

Total Number of Valid Host Addresses: (2^72) - 2 (128 total bits - 56 network bits = 72 remaining bits for hosts)

Subnetting Example 1 (2 Subnets = 2^n, since 57 network bits - 56 network bits = 1, in 2^n):

```
Subnet 1: 2000::/57
Start Address: 2000::
End Address: 2000::7F:FFFF:FFFF:FFFF:FFFF
Subnet 2: 2000:0:0:80::/57
Start Address: 2000:0:0:80::0
End Address: 2000::FF:FFFF:FFFF:FFFF:FFFF
```

Subnetting Example 2 (4 Subnets = 2^n, since 58 network bits - 56 network bits = 2, in 2^n equation)

```
Subnet 1: 2000::/58
Start Address: 2000:0000:0000:0000:0000:0000:0000:0000
End Address: 2000:0000:0000:003F:FFFF:FFFF:FFFF:FFFF
Subnet 2: 2000:0:0:4::/58
Start Address: 2000:0000:0000:0040:0000:0000:0000:0000
End Address: 2000:0000:0000:007F:FFFF:FFFF:FFFF:FFFF
Subnet 3: 2000:0:0:8::/58
Start Address: 2000:0000:0000:0080:0000:0000:0000:0000
End Address: 2000:0000:0000:00BF:FFFF:FFFF:FFFF:FFFF
Subnet 4: 2000:0:0:C::/58
Start Address: 2000:0000:0000:00C0:0000:0000:0000:0000
End Address: 2000:0000:0000:00FF:FFFF:FFFF:FFFF:FFFF
```

Important EdgeOS Commands

Operational

`configure` (change to Configuration mode, available to admins)

`reboot` (reboots the EdgeRouter upon confirmation)

`show`

    `arp` (display arp neighbors on ethX interface)

    `configuration` (view running config)

    `conntrack` (display connections table; e.g. netstat)

    `dhcp`

        leases pool “poolname”

    `ip ospf`

        neighbors

Configuration

`commit` (apply changes to active configuration)

`commit-confirm X` (apply changes but require the user to confirm before X minutes)

`delete` (delete part of the configuration)

`edit` (move to a line in the configuration file to make fast edits)

    `firewall`

    `services`

`exit` (leave configuration mode)

`save` (save to changes to boot configuration)

`set` (add changes to the configuration)

- `firewall` (create, modify, delete firewall rules and groups)
  - group
  - name
- `interfaces` (create, modify, delete interfaces)
  - bridge
  - ethernet
  - loopback
  - tunnel
- `system`
  - config-management (create backups)
    - commit-archive location (scp, ftp, tftp)
    - commit revisions #
  - gateway-address (default gateway of last resort)
  - host-name
  - name-server (name lookup server)

Addressing & Routing Tables

Table: IPv4 Address Classes & Types

Class	Starting Bits	Start Address	End Address	Use
A	0	0.0.0.0	127.255.255.255	All
A	0	0.0.0.0	0.255.255.255	Test
A	0	10.0.0.0	10.255.255.255	Private
A	0	127.0.0.0	127.255.255.255	Loopback
B	10	128.0.0.0	191.255.255.255	All
B	10	172.16.0.0	172.31.255.255	Private
C	110	192.0.0.0	223.255.255.255	All
C	110	192.168.0.0	192.168.255.255	Private
D	1110	224.0.0.0	239.255.255.255	Multicast
E	11110	240.0.0.0	255.255.255.255	Reserved

Table: Default Masks for Classful

Class	Default Mask	Prefix
A	255.0.0.0	/8
B	255.255.0.0	/16
C	255.255.255.0	/24

Table: CIDR to Subnet Mask

CIDR	Subnet Mask	CIDR	Subnet Mask	CIDR	Subnet Mask
/0	0.0.0.0	/11	255.224.0.0	/22	255.255.252.0
/1	128.0.0.0	/12	255.240.0.0	/23	255.255.254.0
/2	192.0.0.0	/13	255.248.0.0	/24	255.255.255.0
/3	224.0.0.0	/14	255.252.0.0	/25	255.255.255.128
/4	240.0.0.0	/15	255.254.0.0	/26	255.255.255.192
/5	248.0.0.0	/16	255.255.0.0	/27	255.255.255.224
/6	252.0.0.0	/17	255.255.128.0	/28	255.255.255.240
/7	254.0.0.0	/18	255.255.192.0	/29	255.255.255.248
/8	255.0.0.0	/19	255.255.224.0	/30	255.255.255.252
/9	255.128.0.0	/20	255.255.240.0	/31	255.255.255.254
/10	255.192.0.0	/21	255.255.248.0	/32	255.255.255.255

Table: Route Type & Administrative Distance

Route Type	Administrative Distance
Connected	0
Static	1
eBGP	20
OSPF	110
RIP	120
iBGP	200

Table: OSPF Areas, Networks & Summarization

Area	Networks	Area Routers	Route Summary	Border Routers
0	10.0.0.0/24 10.0.1.0/24 10.0.2.0/24 10.0.3.0/24	R1 R2 R3 R4	10.0.0.0/22	R1 R4
1	172.16.0.0/30 172.16.0.4/30 172.16.0.8/30 172.16.0.12/30	R1 R6	172.16.0.0/28	R1
2	192.168.1.0/27 192.168.1.32/27 192.168.1.64/27 192.168.1.96/27	R4 R7 R8	192.168.1.0/25	R4

UBIQUITI®
NETWORKS

For information about future training dates, locations, and courses, visit the official training portal of Ubiquiti Networks: **www.ubnt.com/training**

We'd like to hear your feedback!
Contact us at **training@ubnt.com**

NBPR041217V1.2