04a Network Protocol

Notebook:	FGA Cyber		
Created:	7/5/2019 8:16 AM	**Updated:**	7/5/2019 10:43 AM
Author:	clink200032@gmail.com		
URL:	https://static-course-assets.s3.amazonaws.com/CyberOps11/en/course/module4/4.1.1.1...		

Networks come in all sizes. They can range from simple networks consisting of two computers to networks connecting millions of devices. Click the plus signs (+) in the figure to read about networks of different sizes.

Home office networks and small office networks are often set up by individuals that work from a home or a remote office and need to connect to a corporate network or other centralized resources. Additionally, many self-employed entrepreneurs use home office and small office networks to advertise and sell products, order supplies and communicate with customers.

In businesses and large organizations, networks can be used on an even broader scale to provide consolidation, storage, and access to information on network servers. Networks also allow for rapid communication such as email, instant messaging, and collaboration among employees. In addition to internal benefits, many organizations use their networks to provide products and services to customers through their connection to the Internet.

The Internet is the largest network in existence. In fact, the term Internet means a 'network of networks'. The Internet is literally a collection of interconnected private and public networks.

Client-server model

All computers that are connected to a network and that participate directly in network communication are classified as hosts. Hosts are also called end devices, endpoints, or nodes. Much of the interaction between end devices is client-server traffic. For example, when you access a web page on the Internet, your web browser (the client) is accessing a server. When you send an email message, your email client will connect to an email server.

Servers are simply computers with specialized software. This software enables servers to provide information to other end devices on the network. A server can be single-purpose, providing only one service, such as web pages. A server can be multipurpose, providing a variety of services such as web pages, email, and file transfers.

Client computers have software installed, such as web browsers, email, and file transfers. This software enables them to request and display the information obtained from the server. A single computer can also run multiple types of client software. For example, a user can check email and view a web page while listening to Internet radio. Click the plus signs (+) in the figure to read about different clients in a client-server networks.

Protocol

Simply having a wired or wireless physical connection between end devices is not enough to enable communication. For communication to occur, devices must know "how" to communicate. Communication, whether by face-to-face or over a network, is governed by rules called protocols. These protocols are specific to the type of communication method occurring.

For example, consider two people communicating face-to-face. Prior to communicating, they must agree on how to communicate. If the communication is using voice, they must first agree on the language. Next, when they have a message to share, they must be able to format that message in a way that is understandable. For example, if someone uses the English language, but poor sentence structure, the message can easily be misunderstood. Figure 1 shows an example of communication not adhering to protocols for grammar and language.

Network protocol communication is the same way. Network protocols provide the means for computers to communicate on networks. Network protocols dictate the **message encoding, formatting, encapsulation, size, timing, and delivery options**, as shown in Figure 2. As a cybersecurity analyst, you must be very familiar with structure of protocols and how they are used in network communications.

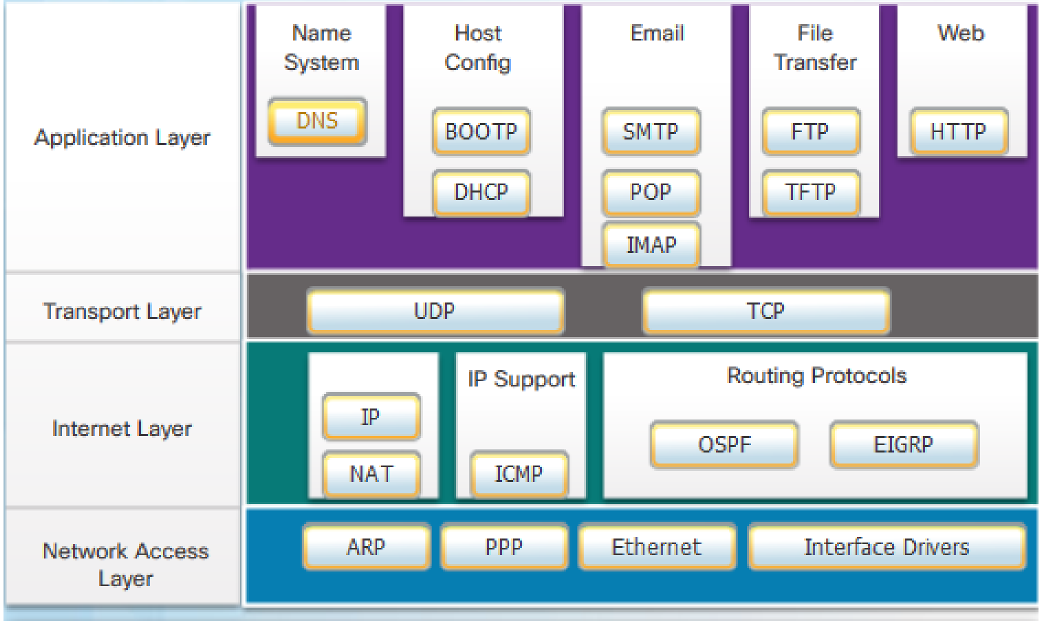
A protocol suite is a set of protocols that work together to provide comprehensive network communication services. A protocol suite may be specified by a standards organization or developed by a vendor.

For devices to successfully communicate, a network protocol suite must describe precise requirements and interactions. Networking protocols define a common format and set of rules for exchanging messages between devices. Some common networking protocols are Hypertext Transfer Protocol (HTTP), Transmission Control Protocol (TCP), and Internet Protocol (IP).

Note: IP in this course refers to both the IPv4 and IPv6 protocols. IPv6 is the most recent version of IP and will eventually replace the more common IPv4.

Figures 1 through 4 illustrate the role of networking protocols:

- How the message is formatted or structured, as shown in Figure 1.
- The process by which networking devices share information about pathways with other networks, as shown in Figure 2.
- How and when error and system messages are passed between devices, as shown in Figure 3.
- The setup and termination of data transfer sessions, as shown in Figure 4.



Protocols define the format, size, and timing of different forms of messages.

Format

When you send an email, protocols of the TCP/IP protocol suite are used by your device to format your message for sending on the network. This is similar to you sending a letter in the mail. You place your letter in an envelope. The envelope has the address of the sender and receiver, each located at the proper place on the envelope, as shown in Figure 1. If the destination address and formatting are not correct, the letter is not delivered. The process of placing one message format (the letter) inside another message format (the envelope) is called encapsulation. De-encapsulation occurs when the process is reversed by the recipient and the letter is removed from the envelope.

Just as a letter is encapsulated in an envelope for delivery, so too are computer messages. Each computer message is encapsulated in a specific format, called a frame, before it is sent over the network. The frame structure is discussed later in the chapter.

Size

Another rule of communication is size. When people communicate in person or over the phone, a conversation is usually made up of many smaller sentences to ensure that each part of the message is received and understood.

Likewise, when a long message is sent from one host to another over a network, it is necessary to break the message into many frames, as shown in Figure 2. Each frame will have its own addressing information. At the receiving host, the individual frames are reconstructed into the original message.

Timing

Timing includes the access method (when can a host send), flow control (how much information can a host send at one time), and response timeout (how long to wait for a response). This chapter will explore how network protocols manage these timing issues.

As you have seen, data is divided into smaller, more manageable pieces to send over the network. This division of data into smaller pieces is called segmentation. Segmenting messages has two primary benefits:

- **Segmentation** – This process increases the efficiency of network communications. If part of the message fails to make it to the destination, due to failure in the network or network congestion, only the missing parts need to be retransmitted.
- **Multiplexing** - By sending smaller individual pieces from source to destination, many different conversations can be interleaved on the network. This is called multiplexing.

In network communications, each segment of the message must be properly labeled to ensure that it gets to the correct destination and can be reassembled into the content of the original message, as shown in Figure 2.

As application data is passed down the protocol stack on its way to be transmitted across the network media, it is encapsulated with various protocol information at each level.

The form that an encapsulated piece of data takes at any layer is called a **protocol data unit (PDU)**. Each succeeding layer encapsulates the PDU that it receives from the layer above in accordance with the protocol being used. At each stage of the process, a PDU has a different name to reflect its new functions. Although there is no universal naming convention for PDUs, in this course, the PDUs are named according to the protocols of the TCP/IP suite, as shown in the Figure 3. Click the plus sign (+) at each PDU in Figure 3 for more information.

When sending messages on a network, the encapsulation process works from top to bottom. At each layer, the upper layer information is considered data within the encapsulated protocol. For example, the TCP segment is considered data within the IP packet. Click Play in Figure 4 to see the encapsulation process as a web server sends a web page to a web client.

Messages sent across the network are first converted into bits by the sending host. Each bit is encoded into a pattern of sounds, light waves, or electrical impulses depending on the network media over which the bits are transmitted. The destination host receives and decodes the signals in order to interpret the message.

This process is reversed at the receiving host, and is known as de-encapsulation. The data is de-encapsulated as it moves up the stack toward the end-user application. Click Play in Figure 5 to see the de-encapsulation process.

Scenario

To summarize network communication processes and protocols, consider the scenario of sending and receiving a web page. Figure 1 lists some of the protocols used between a web server and a web client:

- **HTTP** – This application protocol governs the way a web server and a web client interact.
- **TCP** – This transport protocol manages individual conversations. TCP divides the HTTP messages into smaller pieces, called segments. TCP is also responsible for controlling the size and rate at which messages are exchanged between the server and the client.
- **IP** – This is responsible for taking the formatted segments from TCP, encapsulating them into packets, assigning them the appropriate addresses, and delivering them to the destination host.
- **Ethernet** – This network access protocol is responsible for taking the packets from IP and formatting them to be transmitted over the media.

Figures 2 and 3 demonstrate the complete communication process using an example of a web server transmitting data to a client (Figure 2) and the client receiving the data (Figure 3). Click the Play button to view the animated demonstrations. This process and these protocols will be covered in more detail in later chapters.

1. In Figure 2, the animation begins with the web server preparing the Hypertext Markup Language (HTML) page as data to be sent.

2. The application protocol HTTP header is added to the front of the HTML data. The header contains various information, including the HTTP version that the server is using and a status code indicating it has information for the web client.

3. The HTTP application layer protocol delivers the HTML-formatted web page data to the transport layer. TCP segments the data adding source and destination port numbers.

4. Next, the IP information is added to the front of the TCP information. IP assigns the appropriate source and destination IP addresses. The TCP segment has now been encapsulated in an IP packet.

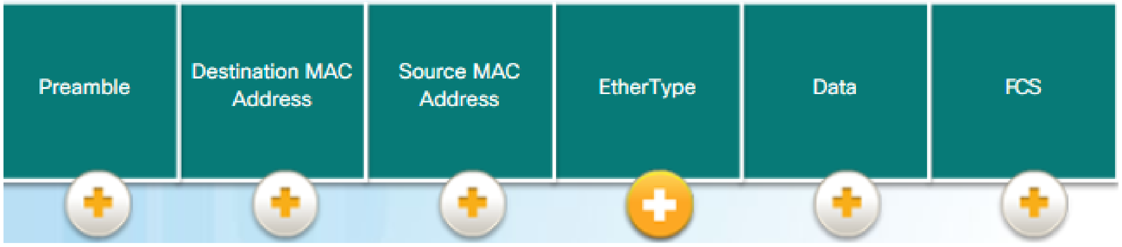
5. The Ethernet protocol adds information to both ends of the IP packet to create a frame. This frame is delivered through the network towards the web client.

6. In Figure 3, the animation begins with the client receiving the data link frames that contain the data. Each protocol header is processed and then removed in the opposite order it was added. The Ethernet information is processed and removed, followed by the IP protocol information, the TCP information, and finally the HTTP information.

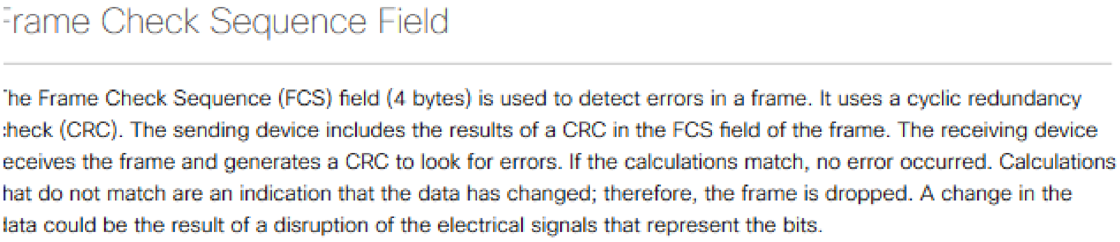
7. The web page information is then passed on to the client's web browser software.

Cybersecurity analysts are adept at using tools to view the behavior of network protocols. For example, Wireshark captures all the details of the protocols encapsulated in packets and data that travels through the network. This course will focus on the use of Wireshark and the interpretation of Wireshark data.

MAC



7-1-6-6-2-x-4



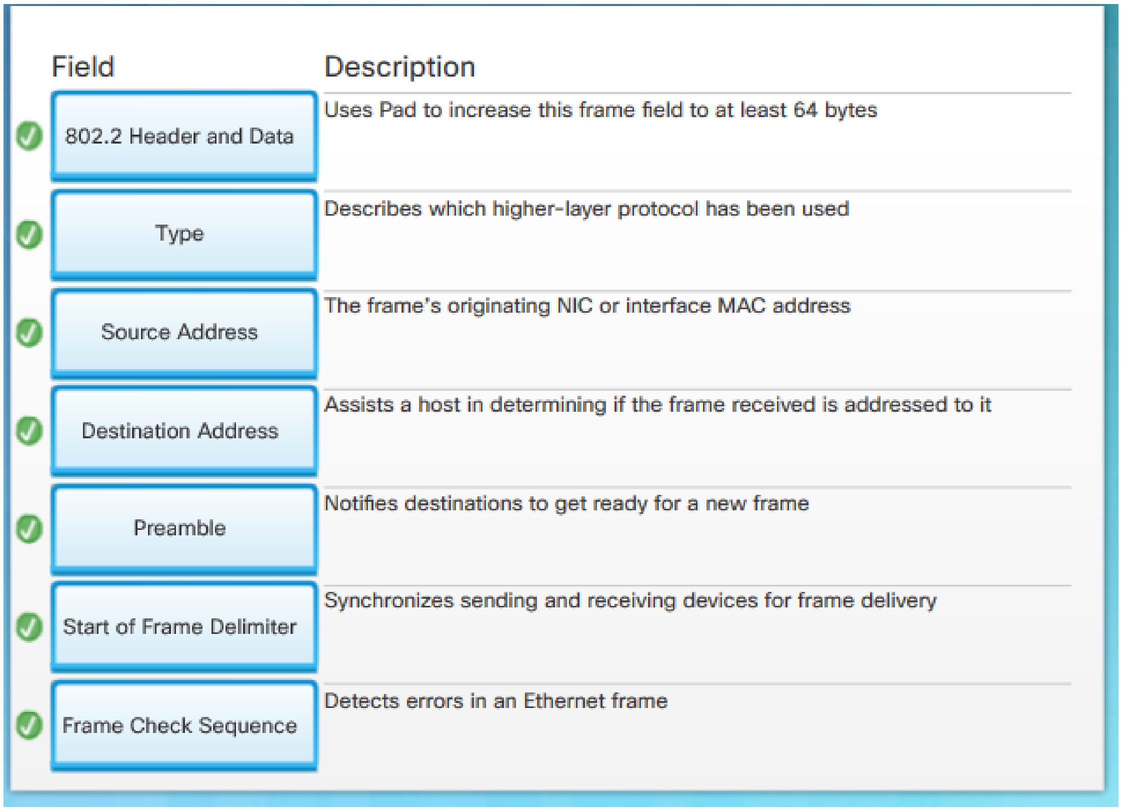
rame Check Sequence Field

he Frame Check Sequence (FCS) field (4 bytes) is used to detect errors in a frame. It uses a cyclic redundancy heck (CRC). The sending device includes the results of a CRC in the FCS field of the frame. The receiving device eceives the frame and generates a CRC to look for errors. If the calculations match, no error occurred. Calculations hat do not match are an indication that the data has changed; therefore, the frame is dropped. A change in the lata could be the result of a disruption of the electrical signals that represent the bits.

The minimum Ethernet frame size is **64 bytes** and the maximum is **1518 bytes**. **This includes all bytes from the Destination MAC Address field through the Frame Check Sequence (FCS) field.** The Preamble field is not included when describing the size of a frame.

Any frame less than 64 bytes in length is considered a “collision fragment” or “runt frame”. Frames with more than 1518 bytes are considered “jumbo” or “baby giant frames”.

If the size of a transmitted frame is less than the minimum or greater than the maximum, **the receiving device drops the frame**. Dropped frames are likely to be the result of collisions or other unwanted signals and are therefore considered **invalid**.



	Field	Description
✓	802.2 Header and Data	Uses Pad to increase this frame field to at least 64 bytes
✓	Type	Describes which higher-layer protocol has been used
✓	Source Address	The frame's originating NIC or interface MAC address
✓	Destination Address	Assists a host in determining if the frame received is addressed to it
✓	Preamble	Notifies destinations to get ready for a new frame
✓	Start of Frame Delimiter	Synchronizes sending and receiving devices for frame delivery
✓	Frame Check Sequence	Detects errors in an Ethernet frame

4.3. Connectivity Verification

Although IP is only a best-effort protocol, the TCP/IP suite does provide for messages to be sent in the event of certain errors. These messages are sent using the services of ICMP. The purpose of these messages is to provide feedback about issues related to the processing of IP packets under certain conditions, not to make IP reliable. ICMP messages are not required and are often not allowed within a network for security reasons.

ICMP is available for both IPv4 and IPv6. ICMPv4 is the messaging protocol for IPv4. ICMPv6 provides these same services for IPv6 but includes additional functionality. In this course, the term ICMP will be used when referring to both ICMPv4 and ICMPv6.

The types of ICMP messages and the reasons why they are sent, are extensive. We will discuss some of the more common messages.

ICMP messages common to both ICMPv4 and ICMPv6 include:

- Host confirmation
- Destination or Service Unreachable
- Time exceeded
- Route redirection

Host Confirmation

An ICMP Echo Message can be used to determine if a host is operational. The local host sends an ICMP Echo Request to a host. If the host is available, the destination host responds with an Echo Reply. Click Play in the figure to see an animation of the ICMP Echo Request/Echo Reply. This use of the ICMP Echo messages is the basis of the ping utility.

Destination or Service Unreachable

When a host or gateway receives a packet that it cannot deliver, it can use an ICMP Destination Unreachable message to notify the source that the destination or service is unreachable. The message will include a code that indicates why the packet could not be delivered.

These are some of the Destination Unreachable codes for ICMPv4:

- **0** - Net unreachable
- **1** - Host unreachable
- **2** - Protocol unreachable
- **3** - Port unreachable

Note: ICMPv6 has similar but slightly different codes for Destination Unreachable messages.

Time Exceeded

An ICMPv4 Time Exceeded message is used by a router to indicate that a packet cannot be forwarded because the Time to Live (TTL) field of the packet was decremented to 0. If a router receives a packet and decrements the TTL field in the IPv4 packet to zero, it discards the packet and sends a Time Exceeded message to the source host.

ICMPv6 also sends a Time Exceeded message if the router cannot forward an IPv6 packet because the packet has expired. IPv6 does not have a TTL field; it uses the **hop limit** field to determine if the packet has expired.

IPv6 RS and RA

The informational and error messages found in ICMPv6 are very similar to the control and error messages implemented by ICMPv4. However, ICMPv6 has new features and improved functionality not found in ICMPv4. ICMPv6 messages are encapsulated in IPv6.

ICMPv6 includes four new protocols as part of the Neighbor Discovery Protocol (ND or NDP).

Messaging between an IPv6 router and an IPv6 device:

- Router Solicitation (RS) message
- Router Advertisement (RA) message

Messaging between IPv6 devices:

- Neighbor Solicitation (NS) message
- Neighbor Advertisement (NA) message

Note: ICMPv6 ND also includes the redirect message, which has a similar function to the redirect message used in ICMPv4.

Figure 1 shows an example of a PC and router exchanging Solicitation and Router Advertisement messages. Click each message for more information.

Neighbor Solicitation and Neighbor Advertisement messages are used for Address resolution and Duplicate Address Detection (DAD).

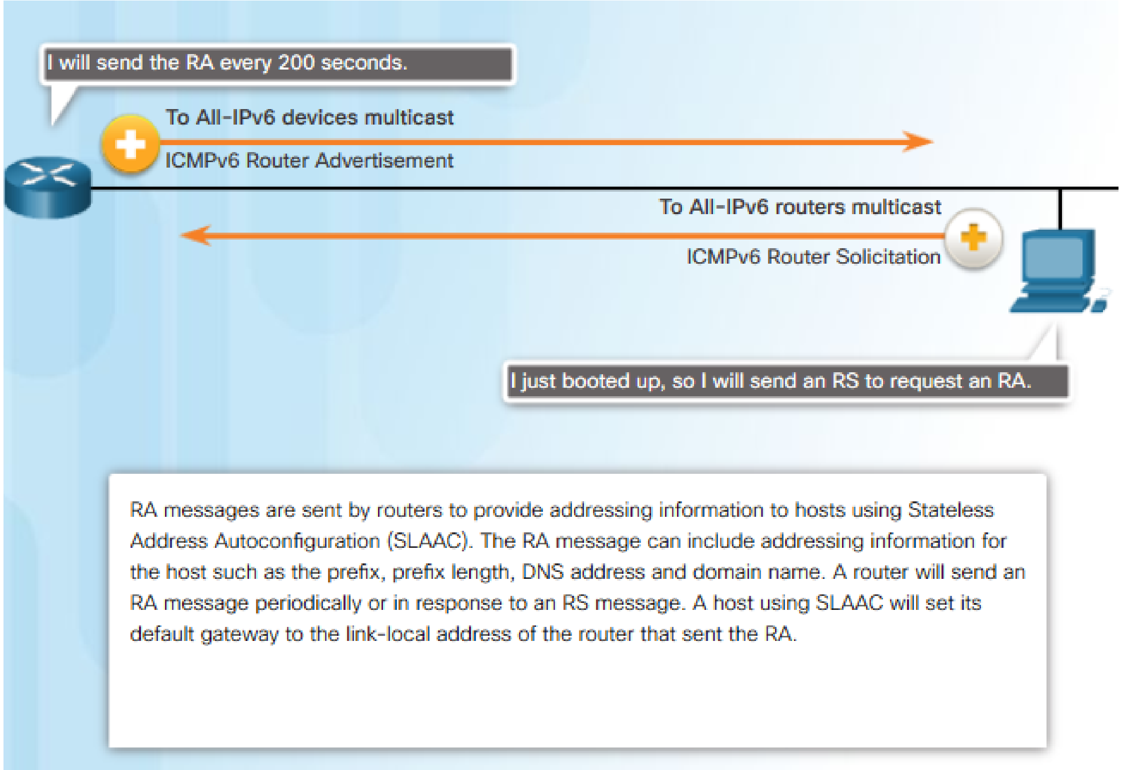
Address Resolution

Address resolution is used when a device on the LAN knows the IPv6 unicast address of a destination but does not know its Ethernet MAC address. To determine the MAC address for the destination, the device will send an NS message to the solicited node address. The message will include the known (targeted) IPv6 address. The device that has the targeted IPv6 address will respond with an NA message containing its Ethernet MAC address. Figure 2 shows two PCs exchanging NS and NA messages. Click each message for more information.

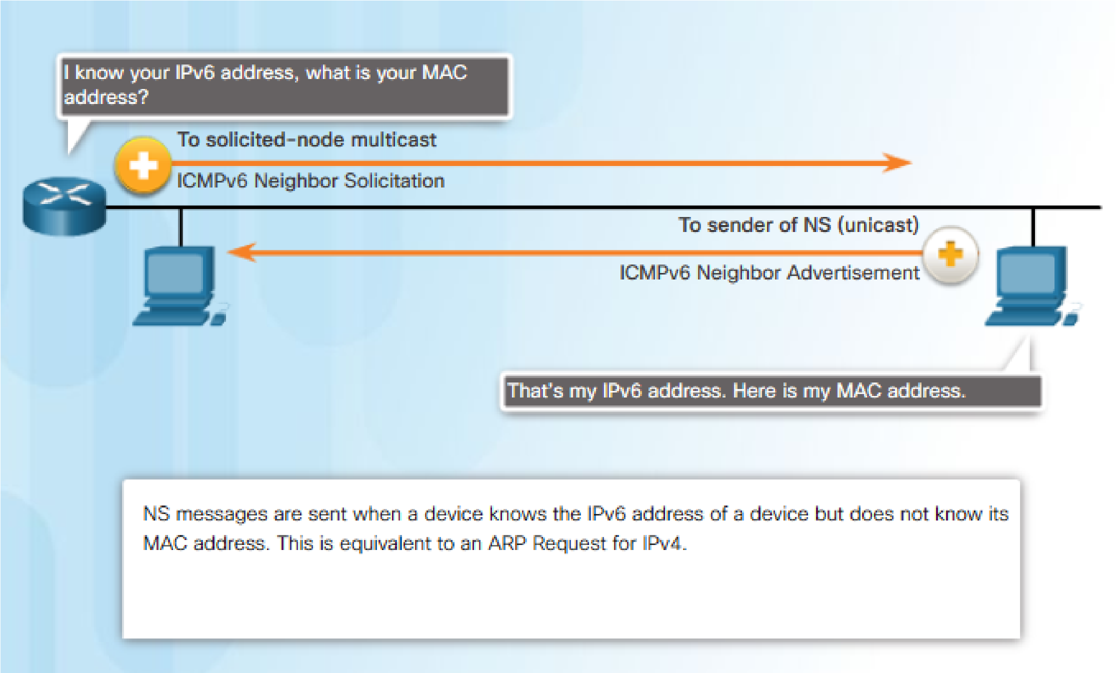
Duplicate Address Detection

When a device is assigned a global unicast or link-local unicast address, it is recommended that DAD is performed on the address to ensure that it is unique. To check the uniqueness of an address, the device will send an NS message with its own IPv6 address as the targeted IPv6 address, shown in Figure 3. If another device on the network has this address, it will respond with an NA message. This NA message will notify the sending device that the address is in use. If a corresponding NA message is not returned within a certain period of time, the unicast address is unique and acceptable for use.

Note: DAD is not required, but RFC 4861 recommends that DAD is performed on unicast addresses.

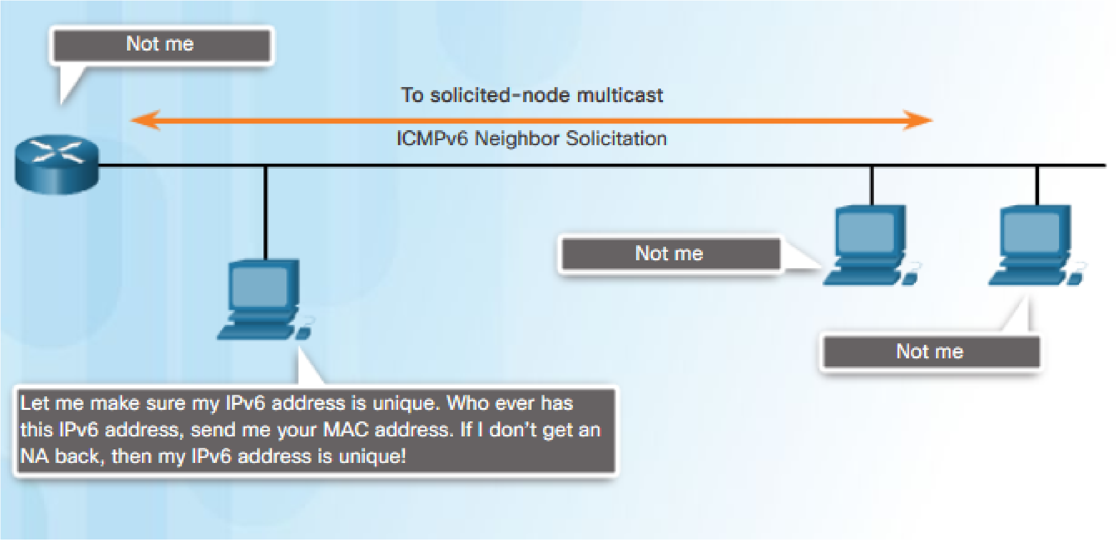

I will send the RA every 200 seconds.
To All-IPv6 devices multicast
ICMPv6 Router Advertisement
To All-IPv6 routers multicast
ICMPv6 Router Solicitation
I just booted up, so I will send an RS to request an RA.

RA messages are sent by routers to provide addressing information to hosts using Stateless Address Autoconfiguration (SLAAC). The RA message can include addressing information for the host such as the prefix, prefix length, DNS address and domain name. A router will send an RA message periodically or in response to an RS message. A host using SLAAC will set its default gateway to the link-local address of the router that sent the RA.

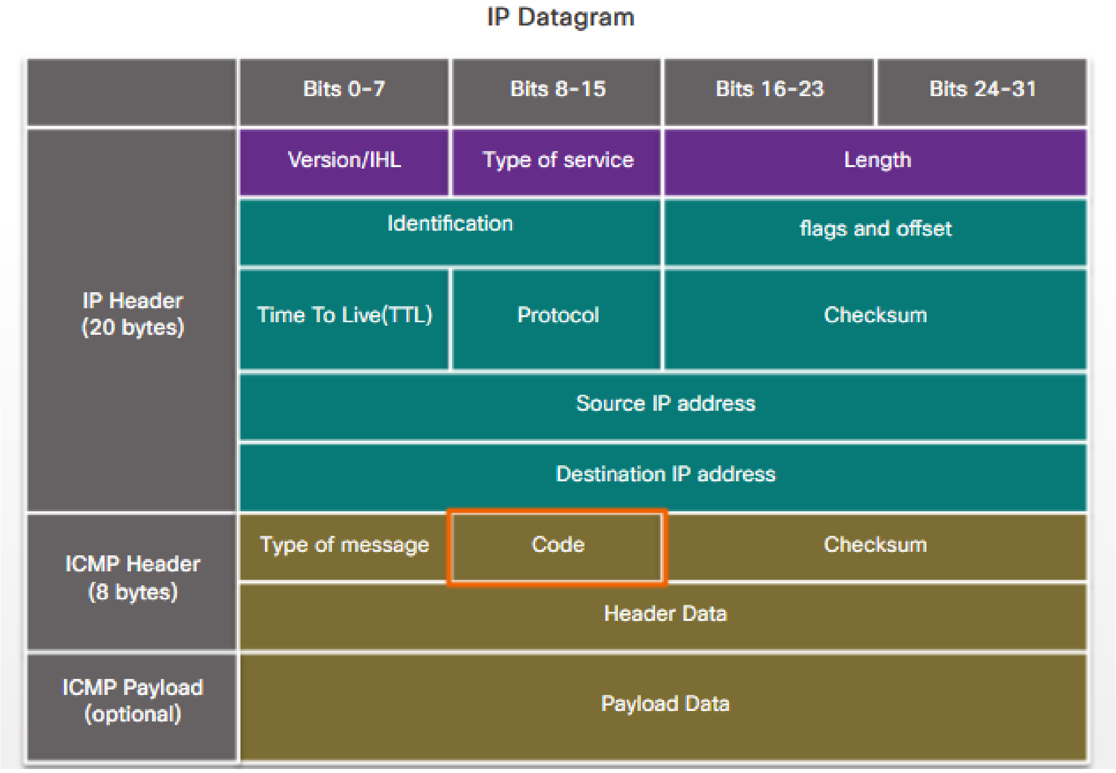

I know your IPv6 address, what is your MAC address?
To solicited-node multicast
ICMPv6 Neighbor Solicitation
To sender of NS (unicast)
ICMPv6 Neighbor Advertisement
That's my IPv6 address. Here is my MAC address.

NS messages are sent when a device knows the IPv6 address of a device but does not know its MAC address. This is equivalent to an ARP Request for IPv4.

DAD =


Not me
To solicited-node multicast
ICMPv6 Neighbor Solicitation
Not me
Not me
Let me make sure my IPv6 address is unique. Who ever has this IPv6 address, send me your MAC address. If I don't get an NA back, then my IPv6 address is unique!

ICMP Packet



IP Datagram

	Bits 0-7	Bits 8-15	Bits 16-23	Bits 24-31
IP Header (20 bytes)	Version/IHL	Type of service	Length	
	Identification		flags and offset	
	Time To Live(TTL)	Protocol	Checksum	
	Source IP address			
	Destination IP address			
ICMP Header (8 bytes)	Type of message	Code	Checksum	
	Header Data			
ICMP Payload (optional)	Payload Data			

ICMP is encapsulated directly into IP packets. In this sense, it is almost like a transport layer protocol, because it is encapsulated into a packet, **however it is considered to be a Layer 3 protocol**. ICMP acts as a **data payload within the IP packet**. It has a special header data field, as shown in the figure.

ICMP uses message codes to differentiate between different types of ICMP messages. These are some common message codes:

- **0** – Echo reply (response to a ping)
- **3** – Destination Unreachable
- **5** – Redirect (use another route to your destination)
- **8** – Echo request (for ping)
- **11** – Time Exceeded (TTL became 0)

As you will see later in the course, a cybersecurity analyst knows that the optional ICMP payload field can be used in an attack vector to exfiltrate data.

For example, some Windows operating systems store ARP cache entries for 2 minutes, as shown in the figure.