75Dineshchandra / Google-IT-Support-Professional-Cert-Notes Public
forked from Clifton893/Google-IT-Support-Professional-Cert-Notes

Code Pull requests Actions Projects Wiki Security Insights Settings

main Google-IT-Support-Professional-Cert-Notes / Course-2 / c2-week-2.md Go to file ...

Clifton893 Update c2-week-2.md Latest commit f220b96 on Nov 28, 2020 History

1 contributor

Week 2: The Network Layer

The Network Layer

Lesson Goals:

- Identify an IP address
- Describe how IP datagrams are encapsulated inside the payload of an ethernet frame
- Correctly identify and describe the many fields of an IP datagram header.

The Network Layer

Address Resolution Protocol (ARP)

IP Addresses

- A single octet is 8 bits of data
- Dotted decimal notation
- IP addresses are distributed in large sections to various organizations and companies instead of being determined by hardware vendors.
  - Thus, IP addresses are more hierarchical, easier to store data about, than physical addresses
- IP addresses belong to networks, not to the devices attached to those networks.
- **Dynamic Host Configuration Protocol (DHCP)** is the technology that automatically assigns IP addresses.

Dynamic IP address vs Static IP address

- A dynamic IP address is distributed by the DHCP
- A static IP address must be configured on a node manually
  - *"In most cases, static IP addresses are reserved for servers and network devices, while dynamic IP addresses are reserved for clients."*

IP Datagrams and Encapsulation

- Much as data packets at the Ethernet layer have a unique name--Ethernet frames--so do data packets at the network layer:
  - Under the IP protocol, a data packet at the network layer is known as an **IP datagram**.
  - An **IP datagram** is a highly-structured series of fields that are strictly defined.

The two primary sections of an IP datagram are the *header* and the *payload*.

IP datagram headers contain much more data than an Ethernet frame header:

- The very first field is the **Version field**, four bits indicating which version of Internet protocol is being used.
  - The most common version of IP is version four (or, IPv4).
- After the version field comes the **Header Length** field.
  - This is also a 4-bit field that declares how long the entire header is.
    - This is almost always 20 bytes in length when dealing with IPv4.
    - In fact, 20 bytes is the minimum length of an IP header.
- Next comes the **Service Type** field.

  - These eight bits can be used to specify details about quality of service (QoS) technologies.
  - QOS services allow routers to make decisions about which IP datagram may be more important than others.

- The next field is a 16 bit field, known as the **Total Length** field.

  - It indicate the total length of the IP datagram it's attached to.

- The **Identification** field, is a 16-bit number that's used to group messages together.

  - The maximum size of a single datagram is the largest number you can represent with 16 bits: 65,535.
  - If the total amount of data that needs to be sent is larger than what can fit in a single datagram, the IP layer needs to split this data up into many individual packets.
    - When this happens, the identification field is used so that the receiving end understands that every packet with the same value in that field is part of the same transmission.

- The **Flag** field is used to indicate if a datagram is allowed to be fragmented, or to indicate that the datagram has already been fragmented.

  - **Fragmentation** is the process of taking a single IP datagram and splitting it up into several smaller datagrams.
  - If a datagram originates from a network allowing for large datagram sizes, and arrives at a network requiring smaller datagram sizes, it will have to be fragmented.

- The **Fragmentation Offset** field contains values used by the receiving end to take all the parts of a fragmented packet and put them back together in the correct order.

- The **Time to Live (TTL)** field is an 8-bit field that indicates how many router hops a datagram can traverse, before it's thrown away.

  - This value is a sort of countdown, decrements the field by one; at zero, routers know they don't have to forward the datagram any further.
  - The main purpose of this field is to make sure that when there's a misconfiguration in routing that causes an endless loop, datagrams don't spend all eternity trying to reach their destination.

- Next is the **Protocol** field, another 8-bit field that contains data about what transport layer protocol is being used.

  - The most common transport Layer protocols are TCP and UDP.

- Next is the **Header Checksum** field, a checksum of the contents of the entire IP datagram header.

  - It functions just like the Ethernet checksum field in an Ethernet frame.
  - Since the TTL field is recomputed at every router hop, the checksum field changes along with it.

- The **Source IP address** and **Destination IP address** fields then follow; each is 32-bits.

- Finally, the **IP Options** field.

  - This is an optional field, and is used to set special characteristics for datagrams primarily used for testing purposes.
  - The IP options field is usually followed by a **Padding** field.
    - Since the IP options field is both optional and variable in length, the padding field is just a series of zeros used to ensure the header is the correct total size.

How does this fit into the big picture?

- The IP datagram is the *data payload section* in an Ethernet header; and this process is known as **encapsulation**.
  - The entire contents of an IP datagram are encapsulated as the payload of an Ethernet frame.
  - The IP datagram also has a payload section: The contents of this payload are the entirety of a TCP or UDP packet.

IP address Classes

IP addresses can be split into two sections:

1. The **network ID**
2. The **host ID**

- The network ID is the first octet in an IP address; while the host ID is the second, third, and fourth octets.
  - Example: `9.100.100.100`
  - The `9.` is the network ID; the `100.100.100` is the host ID

The **Address class system** is a way of defining how the global IP address space is split up.

There are three primary types of address classes:

1. Class A
2. Class B
3. Class C

- **Class A** addresses are those where the first octet is used for the network ID; and the last three are used for the host ID.

- **Class B** addresses are those where the first *two* octets are used for the network ID; and the last two are used for the network ID.
- **Class C** addresses are those where the first *three* octets are used for the network ID; and only the fourth octet is used for the host ID.

Binary trick to Identify IP address classes

- If the very first bit of an IP address is 0, it belongs to a Class A network.
- If the first bits are 10, it belongs to a Class B network.
- If the first bits are 110, it belongs to a Class C network.

The class system has mostly been replaced by a system known as **classless inter-domain routing (CIDR)**.

Address Resolution Protocol

Address resolution protocol (ARP) is a protocol used to discover the hardware address of a node with a certain IP address.

- Once it IP datagram has been fully formed, it needs to be encapsulated inside an Ethernet frame.
  - This means that the transmitting device needs a destination MAC address to complete the Ethernet frame header.
- Almost all network-connected devices will retain a local ARP table.
  - An **ARP table** is a list of IP addresses and the MAC addresses associated with them.

ARP table entries generally expire after a short amount of time, to ensure changes in the network are accounted for.

Further Reading

- Traditional ARP: ARP Process
- ARP explained

Subnetting

Subnetting

Subnetting is the process of taking a large network and splitting it up into many individual and smaller subnetworks, or *subnets*.

"Incorrect subnetting setups are a common problem you might run into as an IT Support Specialist, so it's important to have a strong understanding of how this works."

- Core routers can identify when an address belongs to, for example, a Class A network.
  - They then route the message to the *gateway router* responsible for the network (determined by the network ID).
  - A gateway router specifically serves as the entry and exit path to a certain network.
    - This contrasts with core routers, which (might) only speak to other core routers.
- Once the packet gets to the aforementioned Class A network, the gateway router is now responsible for getting that data to the proper system, by referring to the host ID.
  - But a single Class A network has over 16 million IDs--way too many for one route.
  - This is where subnetting comes into play.
    - With subnets, you can split your large network into many smaller networks.
    - These subnets will each feature their own gateway routers, as entry and exit points for each subnet.

Subnet Masks

- In subnetting, some bits normally used in the host ID are actually used for a *subnet ID*.
  - An example: 10.0.1.10, where `10.0` is the network ID; `1` is the subnet ID; and `10` is the host ID.
- Core Internet routers only care about the network ID, and use it to send the datagram to the appropriate gateway router to that network.
- That gateway router can send the datagram to the destination machine, or the next router in the path to the destination.
- Finally, the host ID is used by the last router, to deliver the datagram to the intended recipient machine.

Subnet Masks

Subnet IDs are calculated by what's known as a *subnet mask*.

- **Subnet masks** are 32-bit numbers that are normally written out as four octets in decimal. (In other words, they look like an IP address.)

A subnet mask is a binary number that has two sections:

- The beginning part -- the mask itself -- is a string of 1s.
- Just 0s come after it.
  - The subnet mask (the part with all the 1s) tells you *what you can ignore* when computing a host ID.

    - The part with all the 0s *tells us what to keep*.
- The purpose of the subnet mask -- again, all the 1s -- *is to tell a router what part of the IP address is the **subnet ID***.

After you've identified the host ID (for example, a Class A), you're left with the last three octets.

- Break these octets into their binary form, and align them with the subnet mask.
    - The binary octet numbers that have a corresponding one in the subnet mask are the *subnet ID*.
    - The binary octet numbers that have a corresponding zero are the *host ID*.

The size of a subnet is entirely defined by its subnet mask.

- Example: `255.255.255.0` tells you that only the last octet is available for host IDs, regardless of what size the network and subnet IDs are.
    - A single 8-bit number can represent 256 different numbers; specifically, the numbers 0-255.

In general, a subnet can only contain *two less* than the total number of host IDs available.

- Going back to that `255.255.255.0` example, 0 is generally not used for an ID; and 255 is normally reserved as a broadcast address for the subnet.
    - Thus, in reality, only the numbers 1-254 are available for assignment to a host.

    - Still, you'll generally refer to the number of hosts available in a subnet as the entire number.
        - EG, you'd still say 8 bits of a host ID's space have 256 addresses available, not 254.
        - That's because those 2 other IPs are still IP addresses, even if they aren't assigned directly to a node on that subnet.

Subnet Masks Beyond Octets

- `255.255.255.254` translates to twenty-seven 1s, and five 0s.
    - Thus, there are 5 bits of host ID space; or 32 addresses.
- Enter a shorthand way of writing subnet masks.
    - `255.255.255.254` could be referenced through the notation `/27`.
    - So an IP address of `9.100.100.100` with the subnet mask of `255.255.255.254` ...
        - ... the entire IP and subnet mask could be written out as:
            - `9.100.100.100/27`
    - Bear in mind, neither notation is necessarily more common than the other.

Basic Binary Math

- Because of the constraints of how logic gates work inside of a processor, it's way easier for computers to think of things only in terms of zero and one.

    - This is also known as binary, or base two.

- You can represent all whole numbers in binary in the same way you can in decimal, it just looks a little different.

    - When you count in decimal you move through all of the numerals upward until you run out then you add a second column with a higher significance.

- For example, in a base ten system, once you count past 9, you start a new column and continue counting.

    - Binary works exactly the same; but instead of counting past 9, a new column initiates when you count past 1.

There's a simple trick to figure out how many decimal numbers can be represented by a certain number of bits.

- 8-bit number = 2^8 = 256, or 0-255.
- 4-bit number = 2^16
- 16-bit number = 2^16 = 65,536

Counting in Binary

- 0 + 0 = 0

- 0 + 1 = 1

- 1 + 0 = 1

- 1 + 1 = 10 (because it can't go past 1, and starts a new column)

- Addition is what's known as an *operator*, and there are many operators that computers use to make calculations.

    - Two of the most important operators are **OR** and **AND**.
        - In computer logic, a **1** represents **true** and a **0** represents **false**.
    - The way the **OR** operator works is you look at each digit, and if either of them is true, the result is true.
    - The operator **AND** does what it sounds like it does: It returns true if both values are true.
        - This is why we say, "1 *and* 1 equals 0."

Bringing It All Together:

A **subnet mask** is a way for a computer to use **and operators** to determine if an IP address exists on the same network.

CIDR

- Address classes were the first attempt at splitting up the global Internet IP space.
- Subnetting was introduced when it became clear that address classes themselves weren't as efficient way of keeping everything organized.
- But as the Internet continued to grow, traditional subnetting just couldn't keep up.
  - See: The limitations of the Internet.

Classless Inter-Domain Routing (CIDR)

- CIDR is an even more flexible approach to describing blocks of IP addresses.
- It expands on the concept of subnetting by using subnet masks to demarcate networks.
  - To *demarcate* something means to set something off.
  - A **demarcation point** refers to where one network or system ends and another one begins.
- CIDR combines the network ID and subnet ID into one.
  - *CIDR notation* is where you get the `9.100.100.100/27` discussed in the last lesson.
- CIDR abandons the concept of address classes entirely, allowing addresses to be defined by only two individual IDs.
  - This simplifies how routers and networks devices need to think about IP address parts.
  - It also avoids enterprise-level challenges, like needing more addresses than a single Class C provided.
  - It also means, that routers now only need to know one entry in their routing table to deliver traffic to these addresses instead of two.

Routing

Basic Routing Concepts

- A **router** is a network device that forwards traffic depending on the destination address of the traffic.
  - A router is a device that has at least 2 network interfaces, since it has to be connected to 2 networks to do its job.

Basic routing has a few steps:

1. A router receives a packet of data on one of its interfaces;
2. The router examines the destination IP of this packet;
3. The router looks up the destination network of this IP in its routing table;
4. The router forwards that out through the interface that's closest to the remote network (determined by info within the routing table).

- These steps are repeated as needed until the traffic reaches its destination.

To protect against breakages, core Internet routers are typically connected in a mesh--meaning there might be many different paths for a packet to take.

Remember, IP addresses belong to networks, *not individual nodes on a network*.

Routers inspect the destination IP; look at the routing table to determine which path is the quickest; and forward the packet along the path.

Routing Tables

- The earliest routers were just regular computers of the era.
  - They had two network interfaces, bridge to networks, and auto-routing table that was manually updated.
  - In fact, all major operating systems today, still have a routing table that they consult before transmitting data.

Routing tables vary; but all share several characteristics.

- The most basic routing table will have 4 columns:

1. **Destination Network:** This column would contain a row for each network the router knows about.

- When the router receives an incoming packet, it examines the destination IP address, and determines which network it belongs to.
- A routing table will generally have a catch-all entry that matches any IP address that it doesn't have an explicit network listing for.

2. **Next Hop:** This is the IP address of the next router that should receive data intended for the destination network in question.

- Or, this could just state the network is directly connected, and that there aren't any additional hops needed.

3. **Total Hops:** The crucial part of understanding routing, and how routing tables work.

- Routers try to pick the shortest possible path at all times, to ensure timely delivery of data.
- But, this shortest possible path can change due to various circumstances.
- So, neighboring routers communicate with each other to keep track of what the shortest/most efficient path to a destination is, at any given time.

4. **Interface:** The router needs to know which of its interfaces it should forward traffic--matching the destination--out of.

Many core Internet routers have *millions* of rows in the routing tables.

Interior Gateway Protocols

In order to learn about the world around them, routers use what are known as *routing protocols*.

- **Routing protocols** are special protocols routers use to speak to each other, in order to share what information they might have.
  - This is how a router on one side of the planet can eventually learn about the best path to a network on the other side of the planet.

Routing protocols fall into two main categories: interior gateway protocols, and exterior gateway protocols.

Interior gateway protocols are used by routers to share information *within* a single autonomous system.

- In networking terms, an **autonomous system** is a collection of networks that all fall under the control of a single network operator.

  - The best example of this would be a large corporation that needs to route data between their many offices an each of which might have their own local area network.

- Interior gateway protocols are further split into two categories:

  i. Link state routing protocols;
  ii. Distance-vector protocols.

- The goals are similar; but the routers employing them share different types of data to accomplish their task.

Distance-vector protocols

- These are an older standard.
- A router using this protocol takes its routing table; then sends this list to every neighboring router (every router directly connected to it).
- In computer science, a *list* is known as a **vector**.
  - This is why a protocol that just sends a list of distances to networks is known as a distance-vector protocol.
- With a distance-vector protocol, routers don't really know that much about the total state of an autonomous system, they just have some information about their immediate neighbors.

Distance vector protocols are pretty simple, but they don't allow for a router to have much information about the state of the world outside of their own direct neighbors.

- Because of this, a router might be slow to react to a change in the network far away from it.
  - This is why link state protocols were eventually invented.

The most common distance-vector protocols are Routing Information Protocol (RIP) and Enhanced Interior Gateway Routing Protocol (EIGRP).

Link state routing protocols

Routers using a link state protocol take a more sophisticated approach to determining the best path to a network.

- Link state protocols get their name because each router advertises the *state* of the *link* of each of its interfaces.
  - These interfaces could be connected to other routers, or they could be direct connections to networks.
  - The information about each router is propagated to every other router on the autonomous system.
- Thus, every router on the system knows every detail about every other router in the system.

Each router then uses this big set of information, and runs complicated algorithms against it, to determine what the best path to any destination network might be.

- Link state protocols require more memory to hold all of this data.
- They also require much more processing power, to compute the quickest path to update the routing tables.

Exterior Gateway Protocols

Exterior gateway protocols are used to communicate data between routers representing the edges of an autonomous system.

Routers use exterior gateway protocols when they need to share information across different organizations [autonomous systems].

The Internet is an enormous mesh of autonomous systems.

- Core Internet routers need to know about autonomous systems in order to properly forward traffic.
- Since autonomous systems are known/defined collections of networks, getting data to the edge router of an autonomous system is the number-one goal of core Internet routers.

The **Internet Assigned Numbers Authority (IANA)** is a non-profit organization, that helps manage things like IP address allocation.

- The Internet couldn't function without a single authority for these sorts of issues.
- The IANA is also responsible for **Autonomous System Number (ASN)** allocation.

  - ASNs are numbers assigned to individual autonomous systems.
  - Just like IP addresses, ASNs are 32-bit numbers.

But unlike IP addresses, they're usually referred to as just a single decimal number (instead of split out into readable bits).

- This is for two reasons:
  i. An ASN never needs to change in order for it to represent more networks or hosts.
  - It's just the core Internet routing tables that need to be updated, to know what the ASN represents.
  ii. ASNs are looked at by humans far less than IP addresses are.
  - ASNs represent entire autonomous systems.
  - ED: AS19604 = IBM

The only exterior gateway protocol standard is Border Gateway Protocol (BGP).

Non-Routable Address Space

RFCs (Request for Comments) started as a way for academics to discuss how their computers might talk to each other.

An RFC would be published, people would leave comments, eventually a consensus would be formed, and a new standard would be developed.

In 1996, **RFC 1989** defined several networks as non-routable address space.

- These are ranges of IPs set aside for use by anyone, that cannot be routed to.
  - After all, not every computer connected to the internet needs to be able to communicate with every other computer connected to the internet. = Non-routable address space allows for nodes on such a network to communicate with each other but no gateway router will attempt to forward traffic to this type of network.
- Network Address Translation (NAT) evolved as a way to allow nodes on non-routable address space networks to communicate with other devices on the Internet.

The primary 3 address ranges of non-routable address space are:

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16

These ranges are free for anyone to use for their internal networks.

- Note: Interior gateway protocols will route these address spaces.
  - Thus, they are appropriate for use within an autonomous system but exterior gateway protocols will not.

Over many decades, RFCs have come to belong to the IETF, orInternet Engineering Task Force, which is an open community charged with developing and maintaining the standards required for the Internet to continue to operate.

Questions for review

Keywords

Concepts

What is the purpose of a TTL field?