IP Addressing & Subnetting Made Easy

Working with IP Addresses

Introduction

- You can probably work with decimal numbers much easier than with the binary numbers needed by the computer.
- Working with binary numbers is time-consuming & error-prone.

Octets

- The 32-bit IP address is broken up into 4 octets, which are arranged into a dotted-decimal notation scheme.
- An octet is a set of 8 bits & not a musical instrument.
- Example of an IP version 4:

172.64.126.52

Thinking in Binary

- The binary system uses only 2 values "0 & 1" to represent numbers in positions representing increasing powers of 2.
- We all are accustomed to thinking & working in the decimal system, which is based on the number 10.

Thinking in Binary *(Cont.)*

- To most humans, the number 124 represents 100 + 20 + 4.
- To the computer, this number is 1111100, which is 64 (2^6) + 32 (2^5) + 16 (2^4) + 8 (2^3) + 4 (2^2) + 0 + 0

- Each position in a binary number represents, right to left, a power of two beginning with 2^0 & increasing by one power as it moves left: 2^0, 2^1, 2^2, 2^4, etc.

Converting to Decimal

- You'll need to convert binary to decimal & vice versa to compute subnets & hosts.
- So, it's time for a quick review lesson in binary-to-decimal conversion.
- There are 8 bits in an octet & each bit can only be a 1 or a 0.

Converting to Decimal *(Cont.)*

- What then do you suppose is the largest decimal number that can be expressed in an octet?

Eight 1’s (1111 1111)

Converting to Decimal *(Cont.)*

- Now, for double the money, what is its equivalent decimal value?

2^7	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**
128	**64**	**32**	**16**	**8**	**4**	**2**	**1**

The binary number 1111 1111 converts into the decimal number:
128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255

Converting to Decimal *(Cont.)*

- Therefore, the largest decimal number that can be stored in an IP address octet is 255.
- The significance of this should become evident later in this presentation.

IP Address Classes

- IP addresses are divided into 5 classes, each of which is designated with the alphabetic letters A to E.
- Class D addresses are used for multicasting.
- Class E addresses are reserved for testing & some mysterious future use.

IP Address Classes *(Cont.)*

- The 5 IP classes are split up based on the value in the 1st octet:

IP Address Class Assignments	
Class	***First Octet Value***
Class A	0 ~ 127
Class B	128 ~ 191
Class C	192 ~ 223
Class D	224 ~ 239
Class E	240 ~ 255

IP Address Classes *(Cont.)*

- Using the ranges, you can determine the class of an address from its 1st octet value.
- An address beginning with 120 is a Class A address, 155 is a Class B address & 220 is a Class C address.

Are You the Host or the Network?

- The 32 bits of the IP address are divided into Network & Host portions, with the octets assigned as a part of one or the other.

Network & Host Representation By IP Address Class				
Class	*Octet1*	*Octet2*	*Octet3*	*Octet4*
Class A	Network	Host	Host	Host
Class B	Network	Network	Host	Host
Class C	Network	Network	Network	Host

Are You the Host or the Network? *(Cont.)*

- Each Network is assigned a network address & every device or interface (such as a router port) on the network is assigned a host address.
- There are only 2 specific rules that govern the value of the address.

Are You the Host or the Network? *(Cont.)*

- A host address cannot be designated by all zeros or all ones.
- These are special addresses that are reserved for special purposes.

Class A Addresses

- Class A IP addresses use the 1st 8 bits (1st Octet) to designate the Network address.
- The 1st bit which is always a 0, is used to indicate the address as a Class A address & the remaining 7 bits are used to designate the Network.
- The other 3 octets contain the Host address.

Class A Addresses *(Cont.)*

- There are 128 Class A Network Addresses, but because addresses with all zeros aren’t used & address 127 is a special purpose address, 126 Class A Networks are available.

Class A Addresses *(Cont.)*

- There are 16,777,214 Host addresses available in a Class A address.
- Rather than remembering this number exactly, you can use the following formula to compute the number of hosts available in any of the class addresses, where **"n"** represents the number of bits in the host portion:

$(2^n - 2)$ = Number of available hosts

Class A Addresses *(Cont.)*

- For a Class A network, there are:

  $2^{24} - 2$ or 16,777,214 hosts.
- Half of all IP addresses are Class A addresses.
- You can use the same formula to determine the number of Networks in an address class.
- Eg., a Class A address uses 7 bits to designate the network, so $(2^7 - 2) = 126$ or there can be 126 Class A Networks.

Class B IP Addresses

- Class B addresses use the 1st 16 bits (two octets) for the Network address.
- The last 2 octets are used for the Host address.
- The 1st 2 bit, which are always 10, designate the address as a Class B address & 14 bits are used to designate the Network. This leaves 16 bits (two octets) to designate the Hosts.

Class B IP Addresses *(Cont.)*

- So how many Class B Networks can there be?
- Using our formula, $(2^{14} – 2)$ there can be 16,382 Class B Networks & each Network can have $(2^{16} – 2)$ Hosts, or 65,534 Hosts.

Class C IP Addresses

- Class C addresses use the 1st 24 bits (three octets) for the Network address & only the last octet for Host addresses.the 1st 3 bits of all class C addresses are set to 110, leaving 21 bits for the Network address, which means there can be 2,097,150 ($2^{21} – 2$) Class C Networks, but only 254 ($2^8 – 2$) Hosts per Network.

Class C IP Addresses *(Cont.)*

Characteristics of the IP Address Classes						
Class	Address Range	Identify Bits (binary value)	Bits in Network ID	Number of Networks	Bits in Host ID	Number of Hosts/ of Network
A	0 ~ 127	1 (0)	7	126	24	16,777,214
B	128~191	2 (10)	14	16,382	16	5,534
C	192~223	3 (110)	21	2,097,150	8	254

Special Addresses

- A few addresses are set aside for specific purposes.
- Network addresses that are all binary zeros, all binary ones & Network addresses beginning with 127 are special Network addresses.

Special Addresses *(Cont.)*

Special IP Addresses			
Network Address	***Host Address***	***Description***	***Example***
0's	0's	Default Cisco Route	0.0.0.0
0's	Host Address	Local Network Hosts	0.0.0.115
1's	1's	Broadcast to Local Network	255.255.255.255
Network Address	1's	Broadcast to Network Address	192.21.12.255
127	Anything	Loopback Testing	127.0.0.1

Special Addresses *(Cont.)*

- Within each address class is a set of addresses that are set aside for use in local networks sitting behind a firewall or NAT (Network Address Translation) device or Networks not connected to the Internet.

Special Addresses *(Cont.)*

- A list of these addresses for each IP address class:

Special Local Network Addresses	
IP Class	***Address Range***
Class A	10.0.0.0 ~ 10.255.255.255
Class B	172.16.0.0 ~ 172.31.255.255
Class C	192.168.0.0 ~ 192.168.255.255

Subnet Mask

- An IP address has 2 parts:
  - The Network identification.
  - The Host identification.
- Frequently, the Network & Host portions of the address need to be separately extracted.
- In most cases, if you know the address class, it's easy to separate the 2 portions.

Subnet Mask *(Cont.)*

- With the rapid growth of the internet & the ever-increasing demand for new addresses, the standard address class structure has been expanded by borrowing bits from the Host portion to allow for more Networks.
- Under this addressing scheme, called Subnetting, separating the Network & Host requires a special process called Subnet Masking.

Subnet Mask *(Cont.)*

- The subnet masking process was developed to identify & extract the Network part of the address.
- A subnet mask, which contains a binary bit pattern of ones & zeros, is applied to an address to determine whether the address is on the local Network.
- If it is not, the process of routing it to an outside network begins.

Subnet Mask *(Cont.)*

- The function of a subnet mask is to determine whether an IP address exists on the local network or whether it must be routed outside the local network.
- It is applied to a message's destination address to extract the network address.
- If the extracted network address matches the local network ID, the destination is located on the local network.

Subnet Mask *(Cont.)*

- However, if they don't match, the message must be routed outside the local network.
- The process used to apply the subnet mask involves Boolean Algebra to filter out non-matching bits to identify the network address.

Boolean Algebra

- Boolean Algebra is a process that applies binary logic to yield binary results.
- Working with subnet masks, you need only 4 basic principles of Boolean Algebra:
  - 1 and 1 = 1
  - 1 and 0 = 0
  - 0 and 1 = 0
  - 0 and 0 = 0

Boolean Algebra *(Cont.)*

- In another words, the only way you can get a result of a 1 is to combine 1 & 1. Everything else will end up as a 0.
- The process of combining binary values with Boolean Algebra is called Anding.

Default Standard Subnet Masks

- There are default standard subnet masks for Class A, B and C addresses:

Default Subnet Masks	
Address Class	***Subnet Mask***
Class A	255.0.0.0
Class B	255.255.0.0
Class C	255.255.255.0

A Trial Separation

- Subnet masks apply only to Class A, B or C IP addresses.
- The subnet mask is like a filter that is applied to a message's destination IP address.
- Its objective is to determine if the local network is the destination network.

A Trial Separation *(Cont.)*

- The subnet mask goes like this:

1. If a destination IP address is 206.175.162.21, we know that it is a Class C address & that its binary equivalent is:
   11001110.10101111.10100010.00010101

A Trial Separation *(Cont.)*

2. We also know that the default standard Class C subnet mask is: 255.255.255.0 and that its binary equivalent is:

11111111.11111111.11111111.00000000

A Trial Separation *(Cont.)*

3. When these two binary numbers (the IP address & the subnet mask) are combined using Boolean Algebra, the Network ID of the destination network is the result:

206.175.162.21	11001110.10101111.10100010.00010101
and	
255.255.255.0	11111111.11111111.11111111.00000000
yields	
	11001110.10101111.10100010.00000000

A Trial Separation *(Cont.)*

4. The result is the IP address of the network which in this case is the same as the local network & means that the message is for a node on the local network.

Routing IP Addresses

- When you build a network, you need to figure out how many network Ids your network requires.
- To do so, you must account for every WAN connection & subnet on the Network.
- Every node & router interface requires a Host address, or ID.

Routing IP Addresses *(Cont.)*

- There's no hard & fast rule on how you should dole out your allotted IP addresses.
- Commonly, though, the lowest numbers (1 through 10) are assigned to routers & servers but how you assign addresses is strictly up to you & your network policies & guidelines.

Configuring an IP Address

- The proper way to configure an IP address on the router is through the IP Address command, which assigns each router interface its unique IP address.
- A router with 4 interfaces needs 4 separate IP addresses because, technically each interface (& address) is on a different network.

Configuring an IP Address *(Cont.)*

- The IP Address command is entered from the config-if mode because the action affects only that interface.
- Both the IP address & the subnet mask are defined in the command.

Verifying an IP Address

- IP addresses are verified using PING, Trace & Telnet.
- It is important that you know that PING is used to verify IP address connections to the Network Layer & that Telnet is used to verify network IP address connections to the Application Layer.

Verifying with Telnet

- The reason you need to verify IP addresses is to ensure that the various parts of a network can properly communicate with the other parts.
- Eg., if you can Telnet (Terminal Emulation Protocol) into a router from a remote location on the same network, you can verify that the interface & route are up and available.

Verifying with Telnet *(Cont.)*

- Because Telnet operates on the OSI Model's Application Layer, when it's functioning, it's safe to assume that all lower layers are also functioning.

Verifying with PING

- The PING (Packet Internet Groper) command verifies OSI Layer 3 (Network Layer) connectivity.
- It sends out ICMP (Internet Control Message Protocol) messages to verify both the logical addresses & the Physical connection.

Verifying with PING *(Cont.)*

- **The PING command issued from a Cisco router responds with a number of single character responses.**

Cisco PING Response Codes	
Response	***Meaning***
! (exclamation Mark)	Success
. (period)	Timed out waiting for reply
U	Destination unreachable
| (vertical bar)	Ping process interrupted
? (question mark)	Unknown packet type
C	Congestion-experienced
& (ampersand)	Time to live exceeded

Verifying with Traceroute

- The Traceroute or Trace command is used to show the complete route from a source to a destination.
- Trace sends out probe packets one at a time to each router or switch in the path between the source & the destination IP address entered.

Verifying with Traceroute *(Cont.)*

- Traceroute displays the round-trip time for each packet sent to each upstream router.
- Traceroute has really only 2 results:
  - Time exceeded or
  - Destination unreachable.
- Trace is used to determine where a breakdown in a route may be occurring.

Verifying with Traceroute *(Cont.)*

- Example on how Trace is used:
  - A network has 4 routers (**A**, **B**, **C** & **D**). A Trace command is issued on router **A** to trace the route from itself to router **D**.
  - A timing response comes back from router **B**, but the next message indicates that router **C** is unreachable. You can be fairly certain that the problem lies somewhere on the route between router **B** & router **C**.

Verifying with Traceroute *(Cont.)*

- Like PING, Trace has its own set of response codes:

Trace Command Response Codes	
Response	***Meaning***
*	Timed out
!H	Router received packet but did not forward it (usually due to an access list)
N	Network unreachable
P	Protocol unreachable
U	Port unreachable

Subnetting

Introduction

- Subnetting is the foundation underlying the expansion of both Local Networks & the Internet in today's world.
- Subnetting has become essential knowledge for the Administrator of any network.
- There are 2 fundamental reasons why subnetting has so much importance in today's networking environment:

Introduction *(Cont.)*

1) The world is running out of available IP addresses. There just isn't an unlimited number of IP addresses available & subnetting helps extend the existing addresses until either the next version of IP is rolled out or some other technology charges on the scene.

Introduction *(Cont.)*

2) Subnetting reduces the size of the routing tables stored in routers. Subnetting extends the existing IP address base & restructures the IP address. As a result, routers must have a way to extract from a IP address both the Network address & the Host address.

Introduction *(Cont.)*

- There are only 3 usable IP address classes:
  - Class A
  - Class B
  - Class C
- Class A networks have the highest number of available hosts.
- Class C networks have the fewest number of hosts.

Subnetting Networks ID

- A 3-step example of how the default Class A subnet mask is applied to a Class A address:

Decimal		Binary
IP Address	123.123.123.001	01111011.01111011.01111011.00000001
Subnet Mask	255.0.0.0	11111111.00000000.00000000.00000000
Network ID	123.0.0.0	01111011.00000000.00000000.00000000

Subnetting Networks ID *(Cont.)*

- In the previous slide, the default Class A subnet mask (255.0.0.0) is AND'd with the Class A address (123.123.123.001) using Boolean Algebra, which results in the Network ID (123.0.0.0) being revealed.
- The default Class B subnet mask (255.255.0.0) strips out the 16-bit network ID & the default Class C subnet mask (255.255.255.0) strips out the 24-bit network ID.

Subnetting, Subnet & Subnet Mask

- Subnetting, a subnet & a subnet mask are all different.
- In fact, the 1st creates the 2nd & is identified by the 3rd.
- Subnetting is the process of dividing a network & its IP addresses into segments, each of which is called a subnetwork or subnet.

Subnetting, Subnet & Subnet Mask *(Cont.)*

- The subnet mask is the 32-bit number that the router uses to cover up the network address to show which bits are being used to identify the subnet.

Subnetting

- A network has its own unique address, such as a Class B network with the address 172.20.0.0 which has all zeroes in the host portion of the address.
- From the basic definitions of a Class B network & the default Class B subnet mask, you know that this network can be created as a single network that contains 65,534 individual hosts.

Subnetting *(Cont.)*

- Through the use of subnetting, the network from the previous slide can be logically divided into subnets with fewer hosts on each subnetwork.
- It does not improve the available shared bandwidth only, but it cuts down on the amount of broadcast traffic generated over the entire network as well.

Subnetting *(Cont.)*

- The 2 primary benefits of subnetting are:
  1. Fewer IP addresses, often as few as one, are needed to provide addressing to a network & subnetting.
  2. Subnetting usually results in smaller routing tables in routers beyond the local internetwork.

Subnetting *(Cont.)*

- Example of subnetting: when the network administrator divides the 172.20.0.0 network into 5 smaller networks – 172.20.1.0, 172.20.2.0, 172.20.3.0, 172.20.4.0 & 172.20.5.0 – the outside world stills sees the network as 172.20.0.0, but the internal routers now break the network addressing into the 5 smaller subnetworks.

Subnetting *(Cont.)*

- In the example, only a single IP address is used to reference the network & instead of 5 network addresses, only one network reference is included in the routing tables of routers on other networks.

Borrowing Bits to Grow a Subnet

- The key concept in subnetting is borrowing bits from the host portion of the network to create a subnetwork.
- Rules govern this borrowing, ensuring that some bits are left for a Host ID.
- The rules require that two bits remain available to use for the Host ID& that all of the subnet bits cannot be all 1s or 0s at the same time.

Borrowing Bits to Grow a Subnet *(Cont.)*

- For each IP address class, only a certain number of bits can be borrowed from the host portion for use in the subnet mask.

Borrowing Bits to Grow a Subnet *(Cont.)*

Bits Available for Creating Subnets		
Address Class	***Host Bits***	***Bits Available for Subnet***
A	24	22
B	16	14
C	8	6