IP Protocols and IP Addressing

The ***TCP/IP***, known as ***Transmission Control Protocol/Internetworking Protocol***, is a four-layer representation of the network that defines how information or data should travel between networks and devices across the Internet. The two (2) most important protocols in the TCP/IP suite are as follows:

- ***Transmission Control Protocol (TCP):*** This is responsible for breaking messages into packets, handling them off to the IP software for delivery, and then orders and reassembles the packets at their destination.
- ***Internet Protocol (IP):*** This is responsible for identifying the hosts through an IP addressing scheme and for the routing of packets through the maze of interconnected networks to their final destination.

OSI and TCP/IP Comparison

Similarities	Differences
Both have layers	The OSI has seven (7) layers, while the TCP/IP appears simpler because it has fewer layers (four [4] layers)
Both have application layers, though they include very different services	The TCP/IP combines the presentation and session layers into its application layer; the characteristics of presentation layer are provided by the application layer, whereas, the characteristics of the session layer are provided by the transport layer
Both have comparable transport and network layers	The TCP/IP combines the OSI data link and physical layers into one layer
Packet-switched (not circuit-switched) technology is assumed	The TCP/IP transport layer using UDP does not always guarantee reliable delivery of packets as the transport layer in the OSI model does
Networking professionals need to know both	The OSI is truly a general model, while TCP/IP cannot be used for any other application

Internet Protocol

The Internet Protocol (IP) had undergone several version changes. Currently, there are two (2) versions of an IP, which are the IP version 4 (IPv4) and IP version 6 (IPv6).

	IP version 4 (IPv4)	IP version 6 (IPv6)
Deployed	1981	1992
Address Size	32 – bit (4 bytes) addressing scheme	128 – bit (16 bytes) addressing scheme
Address Format	Dotted Decimal Notation: 192.149.252.76	Hexadecimal Notation: 3FFE:F200:0234:AB00:0123:4567:8901:ABCD
Prefix Notation	192.149.0.0/24	3FFE:F200:0234::/48
Number of Addresses	2^{32} =~4,294,967,296	2^{128} = ~340,282,366,920,938,463,374,607,431,768,211,456

Information required to identify computers on a TCP/IP network

- ***MAC (Media Access Control) address (aka physical address)*** refers to the unique physical address of all computers, which are assigned by the manufacturer of the network interface card.
  - ***Organizational Unique Identifier (OUI)*** – This is the first 24 bits of the MAC address that is vendor specific; it identifies the company that manufactured or sold the device (e.g. network interface card, or router ports, etc.).

- *Vendor Assigned* – This is the remaining 24 bits of the MAC address that is incrementally and uniquely assigned by the specific vendor of the hardware – denotes the serial number of the individual device.

- *An **IP address***, on the other hand, refers to an address that is usually assigned by the network administrator or internet service provider in order to uniquely and universally identify each device on an IP network.
    - *A **public IP address*** is what computers use to find each other online and exchange information. It is assigned to the computer by the Internet Service Provider as soon as the computer is connected to the Internet gateway.
        - ***Static public IP address*** – It is a fixed IP address and is used primarily for hosting web pages or services on the Internet.
        - ***Dynamic public IP address*** – It is chosen from a pool of available addresses and changes each time one connects to the Internet.
    - *A **private IP address*** is what computers on a network used to talk to the router. It can change each time they are connected. An IP address is considered private if the IP number falls within the class A, B, and C address ranges.

Address Class	**Address Range**
Class A	10.0.0.0 – 10.255.255.255 (Total Addresses: 16,777,216)
Class B	172.16.0.0 – 172.31.255.255 (Total Addresses: 1,048,576)
Class C	192.168.0.0 – 192.168.255.255 (Total Addresses: 65,536)

Reserved for special purposes		
Diagnostic	127.0.0.0 – 127.255.255.255	It is used for testing and debugging of programs or hardware
Default Network	0.0.0.0	Used for routing
Network Broadcast	255.255.255.255	It is used for broadcasting messages to the entire network

- **Parts of an IP Address:**
    - The ***network ID/field*** identifies the host that is located on the same physical network.
    - The ***host ID*** (also known as a host address) identifies the individual host (e. g. workstation, server, router, or other TCP/IP host) within a network.

- *The **subnet mask*** or a default mask (shown in Table 6.5) determines which portion of an IP address identifies the network and which portion identifies the host. Like the IP address, it is represented by four octets. The network bits are represented by the 1s in the network mask, and the node bits are represented by the 0s.

Class	**Default Mask**		
	Decimal	Binary	Shorthand
A	255.0.0.0	11111111.00000000.00000000.00000000	/8
B	255.255.0.0	11111111.11111111.00000000.00000000	/16
C	225.255.255.0	11111111.11111111.11111111.00000000	/24

Notes: Class D & E are used for Multicast and Research purposes.

- ***Default Gateway*** (shown in Figure 6.5) is used to specify the address of the nearest routing device that is used by the host device to forward addressed packets on to the network.

Dotted-Binary to Dotted-Decimal Conversion

01110101 . 10010101 . 00011101 . 00000010 → N_{10}

Solution: *Using Positional Notation*

For: 01110101

Bits	0	1	1	1	0	1	0	1
Place values	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Position	128	64	32	16	8	4	2	1
Add each position of an octet with "1"	64 + 32 + 16 + 4 + 1							
Decimal	**117**							

For: 10010101

Bits	1	0	0	1	0	1	0	1
Place values	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Position	128	64	32	16	8	4	2	1
Add each position of an octet with "1"	128 + 16 + 4 + 1							
Decimal	**149**							

For: 00011101

Bits	0	0	0	1	1	1	0	1
Place values	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Position	128	64	32	16	8	4	2	1
Add each position of an octet with "1"	16 + 8 + 4 + 1							
Decimal	**29**							

For: 00000010

Bits	0	0	0	0	0	0	1	0
Place values	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Position	128	64	32	16	8	4	2	1
Add each position of an octet with "1"	2							
Decimal	2							

The IP address in the form of binary notation above is equivalent to **117.149.29.2** when written in dotted-decimal notation.

Classes of IPv4 Address

Class A address – This uses only the first octet (8 bits) of the 32-bit number to indicate the network address. The entire second to the fourth octet is used for host addresses, which is now equivalent to a total of 24 bits.	Network.Host.Host.Host *Number of Bits used in Network / Host – 24* *Maximum Host of Network – 16,777,214*
Class B address – This uses two (2) of the four (4) octets (16 bits) to indicate the network address. The two other octets, which specify the host addresses, have now a total of 16 bits.	Network.Network.Host.Host *Number of Bits used in Network / Host – 16* *Maximum Host of Network – 65,534*
Class C address – This uses the first three octets (24 bits) of the IP address to identify the network portion, with the remaining octet reserved for the host portion, which is equivalent to eight (8) bits.	Network.Network.Network.Host *Number of Bits used in Network / Host –* 8 *Maximum Host of Network –* 254
Class D address – This is created to enable multicasting using an IP address. A multicast address is a unique address that directs packets with that destination address to predefined groups of hosts.	Host.Host.Host.Host *The first four bits of Class D must be 1110.* *The first octet range for this class is **1110**0000 to **1110**1111, or 224 to 239.*

Class E address – This is reserved by the Internet Engineering Task Force (IETF) for its own research.	*The first four bits of Class E is set to "1111". Therefore, the first octet range for Class E addresses is **11110000 to 11111111**, 240 to 255.*

Subnet Masking

For our computer to tell where to break the IP address apart to get the abovementioned parts of an IP address, the IP address must be "ANDed" with its default mask in binary.

Input		Output
x	**y**	**F = xy**
0	0	0
0	1	0
1	0	0
1	1	1

Example: Determine the network and the host portion of an IP address, 200.133.175.33.
Step 1: By looking at the first octet of the given IP address, we can conclude that it is Class C since it belongs to the range of 192 to 223.

Network	.	Network	.	Network	.	Host	
200	.	**133**	.	**175**	.	33	**Network Portion**
200	.	133	.	175	.	**33**	**Host Portion**

Thus, the binary equivalent of an IP address, 200.133.175.33 is **11001000.10000101.10101111. 00100001**.
Step 2: "AND" the binary equivalent of the IP address with its equivalent default mask in binary.
Note: Default mask to be used is 225.255.255.0 and/or 11111111.11111111.11111111.00000000 in Class C.

IP Address	11001000	.	10000101	.	10101111	.	00100001	200.133.175.33
Default Mask	11111111	.	11111111	.	11111111	.	00000000	225.255.255.0
AND	11001000	.	10000101	.	10101111	.	00000000	**200.133.175.0**

Subnetting

It refers to a process of borrowing bits from the host ID field to form a new subnet ID field.
Example: A company is granted a site address 138.45.0.0 and it needs to be subnet into 45 individual networks.

Step 1: Determine the class and the default mask of the IP address you have been given.

First Octet	Second Octet	Third Octet	Fourth Octet
138	45	0	0

Class B Range: 128 to 191.
Default Mask is **255.255.0.0** or **11111111.11111111.00000000.00000000**

Step 2: Identify the number of subnetworks (subnets) that are required.
Subnets Needed: 45

Step 3: Determine how many bits are required to support the total number of subnets.

	Quotient	Remainder
45/2	22	1
22/2	11	0
11/2	5	1
5/2	2	1
2/2	1	0
½; cannot be		1

6 bits, 101101

This tells us that it takes a total of <u>6 bits</u> to support the value of 45, 101101.

Step 4: Determine the custom subnet mask for our network by masking.

Default Mask	11111111	.	11111111	.	00000000	.	00000000
Custom Subnet Mask	11111111	.	11111111	.	11111100	.	00000000

6 bits

Bits	1	1	1	1	1	1	0	0
Place values	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Position	128	64	32	16	8	4	2	1
Add each position of an octet with "1"	128 + 64 + 32 + 16 + 8 + 4							
Decimal	**252**							

Thus, the custom subnet mask in dotted-decimal notation is **255.255.252.0**; to support 45 individual networks with a given network ID of 138.45.0.0, we would have to use a subnet mask of 255.255.252.0

Step 5: Given our previous IP address of 138.45.0.0 with a subnet mask of 255.255.252.0, we can also determine all of the valid network IDs by using the smallest bit of the custom subnet mask.

Custom Subnet Mask	11111111	.	11111111	.	11111100	.	00000000

Bits	1	1	1	1	1	1	0	0
Place values	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Position	128	64	32	16	8	4	2	1

Increments of 4 - **138.45.4.0** - **138.45.8.0**

Step 6: List each subnet address starting with the incremental value.

Total number of subnets = $2^s – 2$;
S = 6 bits
Total number of subnets = $2^6 – 1 = 62$.

Original Network ID (Not a valid subnetwork address)	138.45.**0**.0
Network ID for Subnet 1	138.45.**4**.0
Network ID for Subnet 2	138.45.**8**.0
Network ID for Subnet 3	138.45.**12**.0
Network ID for Subnet 4	138.45.**16**.0
Network ID for Subnet 5	138.45.**20**.0
Network ID for Subnet 6	138.45.**24**.0
Network ID for Subnet 7	138.45.**28**.0
Network ID for Subnet 8	138.45.**32**.0
Network ID for Subnet 9	138.45.**36**.0
Network ID for Subnet 10	138.45.**40**.0
Network ID for Subnet 11	138.45.**44**.0
. .	
Network ID for Subnet 62	138.45.**248**.0
Custom Subnet Mask value (Not a valid subnetwork address)	146.168.**252**.0

Step 7: Determine the number of host addresses that could support each subnetwork.
Count the remaining bits **(zero's)** to the right-hand side of the custom subnet mask

Default Mask	11111111	.	11111111	.	00000000	.	00000000
Custom Subnet Mask	11111111	.	11111111	.	11111100	.	00000000

Total Hosts = $2^h - 2$; **h = 10**
Total Hosts = $2^{10} - 2 = 1{,}024 - 2$ (for the invalid addresses) = **1,022** total hosts per subnetwork

References:
Mueller, S. (2013). *Upgrading and Repairing PC's 21st Edition*. Indianapolis, Ind.: Que
Oliviero, A. (2014)., *Cabling: the complete guide to copper and fiber-optic networking, 5th ed.* Indianapolis, IN: John Wiley and Sons
Sosinsky, B. (2009). *Networking bible*. Indianapolis, IN: Wiley Pub., inc.
Tanenbaum, A. (2011). *Computer Networks (5th Edition)*. Boston: Pearson Prentice Hall