SecureSet Prep
Introduction to Network Security 1

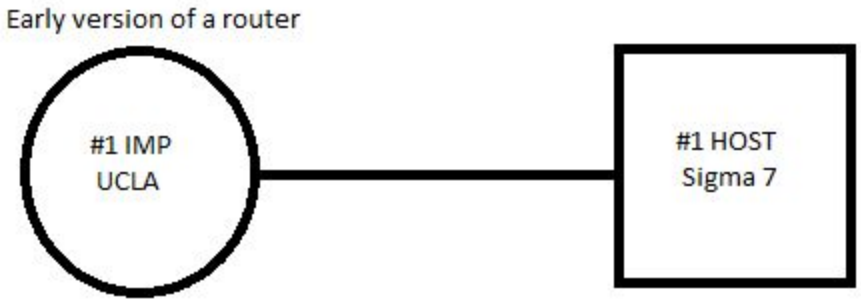
Glenn Webb, CISSP, GSEC

Table of Contents

History of TCP/IP

The early beginnings of networking

- In the mid-1960's ARPA (Advanced Research Projects Agency) was planning a computer network to connect researchers across the country. This project was funded by the U.S. Department of Defense.
- In 1969 the first network link for this project was created between a host and an Interface Message Processor (IMP). The IMP was a host running Network Control Program (NCP) software and it was an early version of what we now call a router.

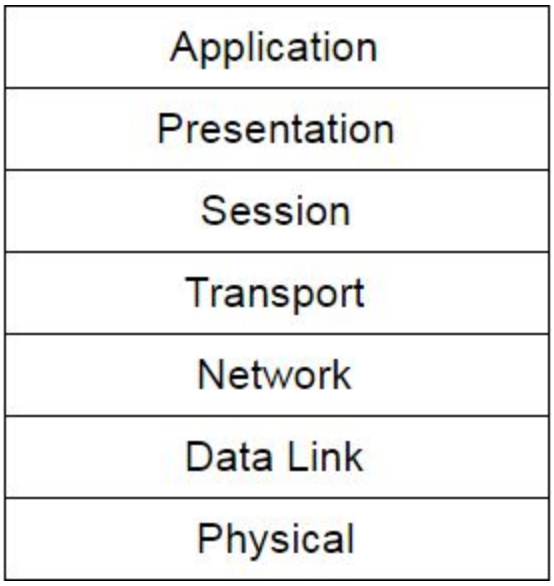


- In 1973 a specification for TCP was written
- In 1977 the details of IP were added to version 3 of the specification
- In 1980 version 4 of the specification was published as an RFC (Request For Comments) and is now what we know as IPv4
- In 1983 IPv4 became the official networking protocol for use on the ARPANET

Networking Models

The OSI Model

> "The OSI model is a generic, protocol-agnostic architectural model that ... is more about describing the ways in which protocols communicate and function together than it is about specifying or describing any particular protocol." *Ric Messier*



1. Physical Layer - is everything about the physical connection
    a. network interfaces
    b. wires
    c. cables
    d. jacks
    e. switches
    f. anything else hardware related
2. Data Link Layer - describes how data flowing down from the top of the stack gets translated onto the physical transmission medium and transferred to other nodes on the same LAN segment. Also covers things like
    a. error detection and correction mechanisms
    b. a set of addresses to ensure that systems can communicate with each other
    c. Ethernet which is the most common protocol for this layer
        i. Common Ethernet speeds are 100 Mb, 1 Gb, 10 Gb and can even go as high as 400 Gb
        ii. Ethernet specifies Media Access Control addresses which are assigned to network interfaces (WiFi and LAN), switch ports, and router ports
        iii. Communication between devices on a subnet is based upon the MAC address. Another way to say this is getting traffic from one system to another on the same network
3. Network Layer - this layer manages the network communication.
    a. Primarily has to do with addressing traffic and ensuring it gets to where it is intended to go.
    b. Routing takes place at this layer. The network layer get traffic from one network to another.
    c. The addressing which occurs at this layer is logical addressing. This is distinct from the physical addresses which exist in the data link layer.

    d. IP (internet protocol) is the most common protocol for this layer
    e. IP addresses provide logical addressing and the ability to aggregate addresses that MAC addresses don't offer. This aggregation enables routing which happens at this layer.
4. Transport Layer - this layer provides connection multiplexing and optionally connection services, reliable delivery, and guaranteed ordering of messages.
    a. Connection multiplexing is done through the use of ports. Each system has 65536 ports for both UDP (user datagram protocol) and TCP (transmission control protocol) packets.
        i. Connections to a system are always done with an address:port pair as well soon see. Using ports allows to system to have up to 65536 easily distinguishable connections from other systems.
    b. Connection services, reliable delivery, and ordering of messages are done by using the TCP protocol in this layer. UDP is a connectionless "send and forget" protocol which does not offer any of these options.
5. Session Layer - A session is like a conversation, it is not just about sending a few messages back and forth, but instead a steady and coherent stream designed to accomplish a particular task.
    a. This layer establishes identity and permissions and also makes sure the session stays open until the task is completed.
        i. If for example a network cable falls out, this layer will re-establish the session (assuming the cable was plugged back-in in a reasonably short time).
    b. This layer offers authentication and authorization as well. It makes sure you are who you say you are but also ensures that you have the right permissions to access a particular resource.
6. Presentation Layer - is responsible for the representation of data.
    a. For example, XML and JPEG formats are at this layer.
    b. Encrypting and decrypting data are examples of functions that would take place at this layer.
7. Application Layer - this layer is where the functions closest to the user exist.
    a. This layer is responsible for generating and handling the data that goes to or comes from the lower layers.
    b. For example, IMAP, SMTP, and POP are all protocols relating to email which reside at this layer.

For the purpose of this course we will only concern ourselves with the first 4 layers of the OSI model as they pertain to networking.

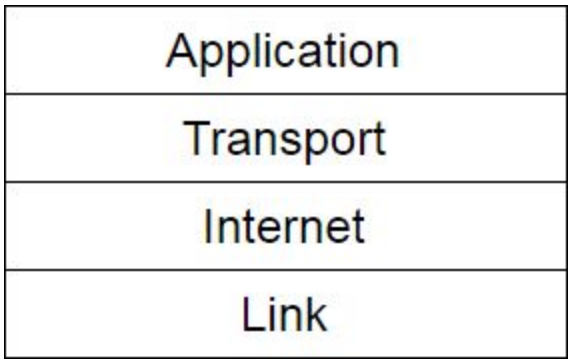
Layer-to-Layer Communication

Application data originates at the top layer of the OSI model (the application layer). As the data descends from top to bottom each subsequent layer tacks information on to the existing data until it gets put onto the transmission media at the Data Link Layer.

After a packet reaches its final destination the data travels up the OSI layers and each layer removes the additional data that was tacked on by the corresponding layer at the originating host.

The TCP/IP Model

The TCP/IP model, also called the Department of Defense model is defined in RFC-1122 dated from 1989, well after TCP/IP had been in widespread use. This RFC specifies four layers which any host on the Internet must implement. This means there must be at least one protocol in each layer.



1. Link Layer - this layer corresponds to the bottom two layers of the OSI model (the Physical and Data Link layers).
    a. When the Link Layer receives a chunk of data it formats it into a **frame** and puts it onto the transmission medium.
    b. This layer also handles the Address Resolution Protocol (ARP), which translates higher layer addresses to Link layer addresses.
    c. Ethernet clearly falls into this layer.
2. Internet Layer - this layer corresponds to the Network layer in the OSI model
    a. The Internet Protocol (IP) resides in this layer
    b. RFC-1122 states that this layer has 2 primary function
        i. the first is choosing the next hop gateway which means it is responsible for routing functions
        ii. the second one is reassembling fragmented IP **datagrams**
    c. Internet Control Messaging Protocol (ICMP) is at this layer to provide the error and diagnostic functionality

3. Transport Layer - this layer corresponds to the Transport layer in the OSI model and handles end-to-end communication for systems
    a. UDP - this protocol falls within this layer
        i. UDP is a "send-and-forget" protocol in which the packets are sent to the destination and no effort is made to verify that the packets have successfully reached their destination. There is no reliability and no connection establishment
    b. TCP - this protocol falls within this layer
        i. TCP is a connection-based protocol which guarantees reliability, error-correction, and in-order delivery of **packets**
4. Application Layer - this layer corresponds to the Session, Presentation, and Application layers of the OSI model.
    a. This layer is not covered in RFC-1122 as the RFC is only concerned with the networking aspects
    b. Examples of protocol at this layer include telnet, Simple Network Management Protocol (SNMP), and Domain Name Service (DNS).

Ref: *GIAC Security Essentials Certification, ch. 2, by Ric Messier*

OSI to TCP/IP Mapping

OSI Layers	TCP/IP Layers
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data Link	Link
Physical	

A more detailed look into TCP/IP

The Link Layer

This layer corresponds to the two bottom layers of the OSI model (physical and data link). The physical components of a network are in this layer such as cables, fiber optics, wifi transmitters, network cards, etc. The data link portion of the Link Layer corresponds the to the Data Link layer in the OSI model and performs two separate functions:

- Media Access Control - this sublayer provides an interface to the network adapter.
- The Logical Link Control - this sublayer provides error checking for arriving frames and manages links between communicating devices.

When transmitting data, packets which are passed to the Link Layer from the Internet Layer are formatted into an Ethernet frame and put onto the transmission media. This frame will also include a CRC. When receiving data, the Link Layer will receive packets from the transmission media, check the CRC (and drop the packet if corrupt), strip off the Link Layer header, and optionally put on a new Link Layer header and forward the packet onto the next broadcast domain.

- ARP is also part of the Link Layer. An ARP request begins when a computer wants to send a packet to a specific IP address. The sending computer broadcasts (FF:FF:FF:FF:FF:FF) an ARP request known as “WHO IS” that requests the MAC address of the computer with the specific IP address. The computer with the specific address then sends an ARP reply “IS AT” to the original sender (unicast) with its MAC address. Optionally, the sender of the ARP reply can send a “Gratuitous ARP” which is a broadcast ARP reply with the purpose of updating the ARP caches of all the machines in the broadcast domain. Address Resolution Protocol is only used in the Link Layer and ARP packets are not routed, they stay within the broadcast domain.
    - Broadcast Domain - is a logical division of a computer network, in which all nodes can reach each other by broadcast at the data link layer (FF:FF:FF:FF:FF:FF). A broadcast domain can be within the same LAN segment or it can be bridged to other LAN segments. Routers and other higher-layer devices form boundaries between broadcast domains.
- RARP (Reverse ARP) is an obsolete protocol you may see mentioned. A RARP query is when a Link Layer broadcast is sent asking for the IP address of the system who owns the MAC address just queried about. RARP has been replaced by DHCP (Dynamic Host Configuration Protocol) which supports many features included an expanded feature set of RARP.

The Link Layer can support several types of transmission mediums such as 802.11 Wireless, Ethernet, and Modems (Point-to-Point Protocol).

Protocols and Hardware

The Network Access Layer manages all the services and functions necessary to prepare data for the physical network such as

- Interfacing with the computer's network adapter
- Converting the data into a format that will be transmitted into the stream of electric or analog pulses across the transmission medium
- Checking for errors in incoming data (a frame check sequence FCS in the form of a cyclic redundancy check CRC calculation)
- Adding error-checking information to outgoing data so the other end of the transmission can check for errors (adding an FCS)

Ethernet

Wired Ethernet has been the dominant LAN technology for years but recently Wireless LAN technology has become prevalent in many spaces (for example, home networks) and Mobile Internet Technologies is the LAN technologies for tablets and cell phones. Ethernet typically offers better speed, security, and reliability than over-the-air technologies such as WiFi, Bluetooth, and 3G/4G cell phone technology.

Anatomy of an IEEE 802.3 Ethernet Frame

When the lower level of the Link Layer receives a datagram from the Internet Layer it

1. Breaks the data into smaller chunks (if necessary). Typically the maximum size of a data chunk is 1500 bytes called the Maximum Transmission Unit (MTU). Some systems support an MTU of up to 9000 bytes. The total size of a regular Ethernet frame must be between 64 bytes and 1518 bytes.
2. The chunks of data are packed into the data field of the Ethernet frames. Each frame contains the data along with the other fields described in the table below
3. Passes the data frame to the low-level physical layer for transmission over the medium.

Field	Length	Purpose
Preamble	64 bits (8 bytes)	A sequence of bits used to mark the beginning of the frame. The last byte is the 1-byte Start Frame Delimiter
Destination Address	48 bits (6 bytes)	The MAC address of the recipient
Source Address	48 bits (6 bytes)	The MAC address of the sender
VLAN Tag	16 bits (2 bytes)	This optional field is designed

		to allow multiple Virtual LANs to operate on the same network switch
Length	16 bits (2 bytes)	Indicates the size of the data field
Data	Variable	The data transmitted in the frame
Frame Check Sequence	32 bits (4 bytes)	A Cyclic Redundancy Check used to verify the integrity of the transmitted data

The Internet Layer

The Internet Layer is responsible for applying routing headers to packets being sent and interpreting routing headers from packets being received.

IP Headers

The IPv4 header is defined to be up to 24 bytes in length but it is typical to see 20 byte headers.

Anatomy of an IPv4 Header

Field	Length	Purpose
Version	4 bits	Currently v.4, 0100
Internet Header Length	4 bits	The length of the IP header in 32-bit words. The minimum value is 5 which equals 20 bytes
Differentiated Services Code Point	6 bits	Used to differentiate priorities of datagrams. Voice over IP uses this field
Explicit Congestion Notification	2 bits	Helps to provide end-to-end notification of network congestion
Total Length	16 bits (2 bytes)	Total length of the datagram in bytes
Identification	16 bits (2 bytes)	Used to identify an IP datagram for reassembling fragmented datagrams
Flags	3 bits	Fragmentation notification bits. Bit 0 is reserved, Bit 1: Don't Fragment, Bit 2: More Fragments

Fragment Offset	13 bits	Specifies the offset of a particular fragment relative to the beginning of the original unfragmented IP datagram
Time To Live (TTL)	8 bits (1 byte)	This field represents a packet's maximum lifetime in hops
Protocol	8 bits (1 byte)	Indicates what protocol is in the next set of headers (ICMP=1, TCP=6, UDP=11)
Header Checksum	16 bits (2 bytes)	Used to verify integrity. Recalculated on every hop
Source Address	32 bits (4 bytes)	Sender's IP address
Destination Address	32 bits (4 bytes)	Recipient's IP address
Options	Variable length	Optional header values can be inserted here. Not often used.

IPv4 Header Format

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

https://en.wikipedia.org/wiki/IPv4

Sample IP Header from Wireshark

```
45 00 00 25 00 00 40 00 40 11 B7 FD C0 A8 00 7B C0 A8 00 FF
```

Step through the fields of this sample output. Each pair is 1 byte (8 bits). Each character is 4 bits.

ICMP (Internet Control Message Protocol) is also part of the Internet Layer. ICMP is a support protocol for other protocols in TCP/IP. It is designed to provide error and diagnostic messages to assist in the delivery of messages. When packets run into problems on the network, ICMP messages are sent back to the original sender. For example if a packet from network A is destined for a host on remote network D but the host on network D is down, the router attached to network D will be the one which detects that the node is down. The router on network D would send back an ICMP message that the host is down. It will do the same if the node is up

but the port requested is unavailable. Two common Linux utilities which use ICMP packets are `ping` and `traceroute`. The ping command sends an ICMP echo request to the desired host. If the host can reply it will send back an ICMP reply to the originating sender.
is a very common way to determine if a host is up and available. However it should be noted that failing to receive an ICMP reply does not always mean the host in question is down. A firewall or host settings can prevent the host from replying. The traceroute utility uses a decrementing time-to-live (TTL) setting in the ICMP packet to determine the path a packet takes through a network to reach its destination.

ARP is sometimes considered to be a part of the Internet Layer but other sources place it into the Link Layer. We will cover ARP in the Link Layer Session.

Facts about IPv4 and IPv6

The IP protocol has been around for nearly 4 decades but is still predominantly used in networking. IPv4 addresses take the form of 4 dot-separated values from 0-255 (called octets). A total of 32 bits are required to represent an IPv4 address, for example `209.121.131.14`. IPv4 allows for 4,294,967,296 addresses of which 3,706,452,992 are available for public use. In February 2011 the Internet Corporation for Assigned Names and number announced that the Internet had run out of IPv4 addresses. Network Address Translation (NAT) is a solution which allows many hosts to share one public IPv4 address.

IPv4 binary representation

It is important to understand the binary representation of IPv4 addresses as they will be used with netmasks and CIDR (Classless Interdomain Routing) notation, to be covered later. As an example we will break down the address `209.121.131.14` to its binary representation.
`11010001 01111001 10000011 00001110`
The numeric value of the columns in each grouping is as follows $2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$
Converting the first group 11010001 to decimal is as follows:
$1x2^7 + 1x2^6 + 0x2^5 + 1x2^4 + 0x2^3 + 0x2^2 + 0x2^1 + 1x2^0$ = 128+64+0+16+0+0+0+1 = 209
Converting the second group 01111001 to decimal is as follows:
$0x2^7 + 1x2^6 + 1x2^5 + 1x2^4 + 1x2^3 + 0x2^2 + 0x2^1 + 1x2^0$ = 0+64+32+16+8+0+0+1 = 121
Converting the third group 10000011 to decimal is as follows:
$1x2^7 + 0x2^6 + 0x2^5 + 0x2^4 + 0x2^3 + 0x2^2 + 1x2^1 + 1x2^0$ = 128+0+0+0+0+0+2+1 = 131
Converting the last group 00001110 to decimal is as follows:
$0x2^7 + 0x2^6 + 0x2^5 + 0x2^4 + 1x2^3 + 1x2^2 + 1x2^1 + 0x2^0$ = 0+0+0+0+8+4+2+0 = 14

Usage of IP version 6 is increasing with help from the limitations of IPv4. Internet Protocol version 6 (IPv6) addresses take the form of eight colon-separated hexadecimal values ranging from 65535. A total of 128 bits are required to represent an IPv6 address, for example `3FFE:1901:3875:0009:034A:F8FF:FB11:C88F`. IPv6 allows for 340,282,366,920,938,463,463,374,607,431,768,211,456 IP addresses. Some sources say that

is enough for over 100 IP addresses for every atom on the face of the earth or 39,614,081,257,132,168,796,771,975,168 IP addresses for each living human being. There are plenty.
The smallest IPv6 address is `0000:0000:0000:0000:0000:0000:0000:0000`
The largest IPv6 address is `FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF`

Network Classes and Subnets

An IP address contains the network ID and the host ID for a given node on a network. For example, the IP address we were using earlier `209.121.131.14` is a Class C address. The network ID is `209.131.131.0` and the host ID is `14`. We will look look at how and why we have network classes.
Since an IP address contains a network ID and a host ID there needs to be a way to distinguish the two parts. The original way to distinguish the network ID and host ID was by a system of Network Classes.

- If the 32-bit binary address starts with a 0 bit, the address is a Class A address. Class A addresses use the first 8 bits for the network ID and the remaining 24 bits are for the host ID
- If the 32-bit binary address starts with 10, the address is a Class B address. Class B addresses use the first 16 bits for the network ID and the remaining 16 bits are for the host ID
- If the 32-bit binary address starts with 110, the address is a Class C address. Class C addresses use the first 24 bits for the network ID and the remaining 8 bits are for the host ID

Our example IP address is a Class C address because the first octet `11010001` starts with 110. A Class C address implies that you have a manageable size subnet. The subnet can contain 254 hosts, which is not too large. What if you have a Class A address? Your subnet is too large to be manageable because it can contain up to 16 million hosts. We will explore how to make smaller subnets in a later session.

Address Ranges for Class A, B, and C networks

Address Class	Begins With	First Term	Excluded Addresses (private addresses)
A	0	0 to 127	10.0.0.0 to 10.255.255.255 127.0.0.0 to 127.255.255.255
B	10	128 to 191	172.16.0.0 to 172.31.255.255
C	110	192 to 223	192.168.0.0 to 192.168.255.255

Special IP Addresses

Network ID Notation - an IP address in which the host ID portion is zeroed-out is the Network ID. For example 129.152.0.0 refers to the Class B network with Network ID 129.152.

If the binary representation of an IP address has all 1's in the host ID portion, this indicates a broadcast identifier for the given network. For example the Class C Network ID from a previous example was `209.121.131.0`. The binary representation of the Network ID for this is

```
11010001 01111001 10000011 00000000
```

To represent a broadcast address for this Network ID the last octet would be all ones like so

```
11010001 01111001 10000011 11111111
```

The decimal representation of this address is `209.121.131.255`. Another way to think of this is the broadcast address is the highest numbered address you can have on a network and the network ID is the lowest numbered address you can have on a network. Another broadcast address 255.255.255.255 accomplishes the same thing as using the highest numbered address on the network. Not all broadcast packets will cross a router border. It depends on the type of packet being broadcast. By default DHCP packets will not cross a router border but DNS packets will.

IP addresses which begin with 127 are loopback addresses. The address 127.0.0.1 is most commonly to represent the loopback device.

Subnetting

We have discussed how class A networks can be unwieldy because of the host ID space (16 million host IDs). This session will discuss methods to break up a physical network into smaller entities called subnets.

Address Ranges for Class A, B, and C networks

Address Class	Begins With	First Term	Excluded Addresses (private addresses)
A	0	0 to 127	10.0.0.0 to 10.255.255.255 127.0.0.0 to 127.255.255.255.255
B	10	128 to 191	172.16.0.0 to 172.31.255.255
C	110	192 to 223	192.168.0.0 to 192.168.255.255

Dividing a network

The IP address classes (A, B, and C) can give you an idea of what the Network ID and Host ID are when looking at an IP address, however the address class system is too rigid to do the job alone. In real-world environments networks are divided into smaller subnets by using the subnetting technique we will discuss in this session.

Subnetting lets you break a network into smaller subnets. The concept of subnets came about concurrently with the address class system. As time went on the internet community and hardware vendors settled on a new system called Classless Inter-Domain Routing (CIDR) which does not emphasize the address class system.

In order to subnet a larger network a tool called a subnet mask is used. A subnet mask allows the network administrator to borrow bits which were originally in the host ID portion of an IP address to supplement the Network ID portion. Like an IP address, a subnet mask is a 32-bit binary number. The bits of the subnet mask are arranged in a pattern that reveals the subnet ID of the IP address to which the mask is associated. The netmask uses a 1 for every bit in the network ID and a 0 to designate any bits which are part of the host ID. The bits from the binary IP addresses are Bitwise And'd with the bits in the subnet mask.

Bitwise-And Operations
0 and 0 = 0
0 and 1 = 0
1 and 0 = 0
1 and 1 = 1

For example
IP address: `11010000001000110110100100110011`
Subnet mask: `11111111111111111111111100000000`
Resulting Network ID *`11010000001000110110100`*`100000000`

Simply put, the 1's portion of the subnet mask refers to the Network ID and the 0's portion refers to the host ID section. If we have an IP address of
`219.55.179.245` or `11011011:00110111:10110011:11110101`
and the netmask is
`255.255.255.240` or `11111111:11111111:11111111:11110000`

then the Network ID would be

```
11011011:00110111:10110011:11110101   (219.55.179.245)
11111111:11111111:11111111:11110000   (255.255.255.240)
-----------------------------------
11011011:00110111:10110011:11110000  = 219.55.179.240 (network ID)
```

Recall that the number of 0's in the right hand side of the netmask indicates how many host ID's there will be in the network. The netmask we just used had four 0's so the number of host IDs in this network is $2^4 = 16$

If we have another host from this network at `219.55.179.252` we already know the network number is `219.255.179.240` but what is the Host ID for this IP address? Just looking at the last octet `252` in binary it is `11111100`. We know the Network ID extends into the first 4 bits of this octet because of the netmask we used so the Host ID is comprised of the last 4 bits which are `1100`. This is 12 in decimal, and this is the Host ID.

All network addresses have a minimum value and a maximum value. As we have seen, the minimum value is the network number. The maximum value is by definition the broadcast address. This broadcast address is different than the Link Layer broadcast address we saw earlier.

CIDR Notation

Classless Inter-Domain Routing (CIDR) which reduces the need for the classical approach to subnetting builds upon the concepts of subnetting will be introduced in this discussion.

Classless Inter-Domain Routing is a more flexible technique for defining blocks of addresses in routing tables. The CIDR system does not depend of Class A, B, and C networks where the network ID is defined by 8, 16, or 24 bits, respectively. Instead, a single number called the CIDR prefix specifies the number of bits in the address which are the Network ID. This prefix is also called the Variable Length Subnet Mask (VLSM). CIDR notation uses a slash (/) separator followed by a decimal number to specify the number of bits in the network portion of the address.
For example, the CIDR address `219.55.179.252/25` specifies that 25 bits of the IP address refer to the network portion. The equivalent subnet mask is `255.255.255.128`.
Another aspect of CIDR is that the VLSM can be used to aggregate multiple consecutive Class C networks into a single entity. The VLSM in this case would be called a supernet mask. For example, an ISP may lease a series of consecutive Class C networks like 204.21.128.0 to 204.21.255.255.

```
11001100000101011000000000000000  (204.21.128.0)
11001100000101011111111111111111  (204.21.255.255)
```

The number of bits in the Network ID of a Class C network would normally be 24 bits but if we use CIDR and make the number of bits to be 17 in the Network ID we have effectively combined the range of Class C networks listed above into one large subnet. The address range of this

new network is specified by using the lowest address in the range followed by the VLSM. Like so `204.21.128.0/17`

The important thing to remember about subnetting and CIDR notation is that they both do the same thing. They tell routers how many bits of the address to use when making routing decisions.

The Transport Layer

UDP Headers

The User Datagram Protocol is considered an unreliable protocol. This means that is has no built-in mechanism to insure that messages get delivered and in the correct order. UDP does not fail as often as one may assume for an "unreliable" protocol.

With nothing to prevent messages from getting dropped or arriving out of order any application using UDP should not assume that every packet will arrive and in the correct order. Some network services do better using UDP like voice or media streaming, IP telephony, and other speed dependent or real-time applications. Speed is key and so UDP headers are minimal and easy to process. UDP also offers multiplexing via port numbers.

Anatomy of a UDP header

Field	Length	Purpose
Source Port	16 bits (2 bytes)	The port assigned to the application which is originating the packets. This is the same port which the response would get sent to
Destination Port	16 bits (2 bytes)	The port that messages are being sent to on the destination system
Length	16 bits (2 bytes)	The length of the UDP message. 16 bits can hold values of up to 65535, indicating a UDP packets can be up to 64k in length
Checksum	16 bits (2 bytes)	Validation that a message has arrived unaltered in transit

The simplicity of UDP headers demonstrates what little header information is required for a forward-and-forget packet.

TCP Headers

TCP provides reliable communication along with multiplexing (ports). TCP uses 16 bits (2 bytes) to designate port numbers meaning there can be up to 65536 ports for multiplexing per IP address. An outgoing TCP/IP packet will have a source IP and Port pair and a destination IP and Port pair. When the recipient of a TCP/IP packet needs to respond back to the sender it reverses these pairs so the sender IP:Port becomes the destination IP:Port and vice versa.

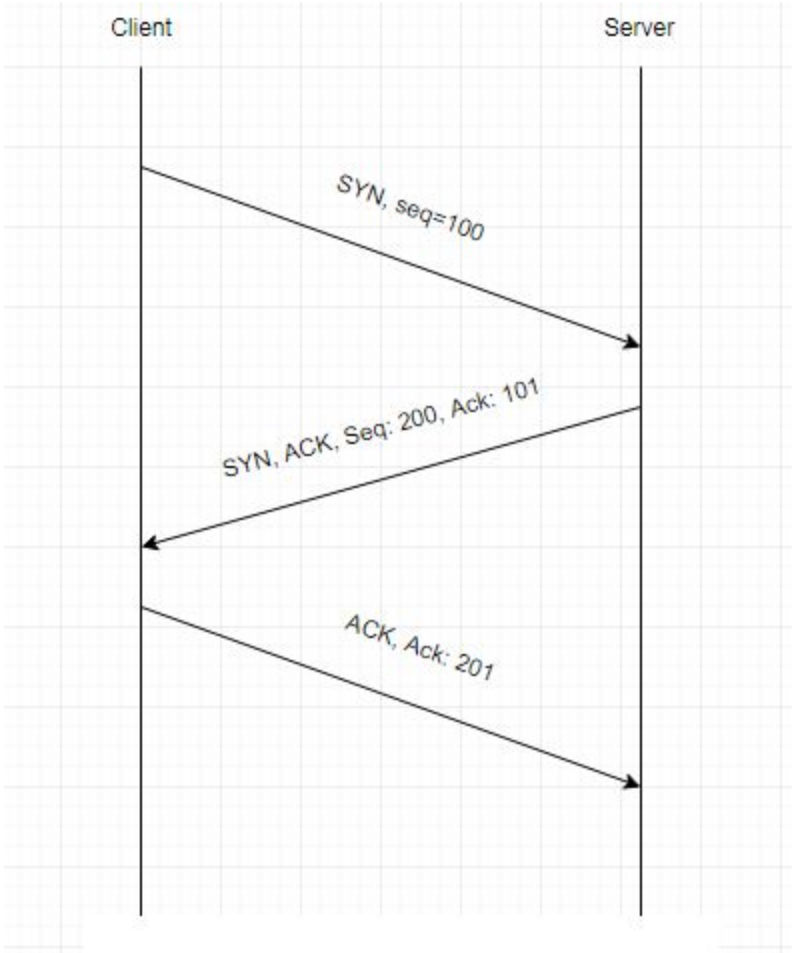
Anatomy of a TCP header

Field	Length	Purpose
Source Port	16 bits (2 bytes)	The port assigned to the application which is originating the packets. This is the same port which the response would get sent to
Destination Port	16 bits (2 bytes)	The port that messages are being sent to on the destination system
Sequence Number	32 bits (4 bytes)	It is shared between sender and receiver to insure that data is received and in the correct order
Acknowledgment Number	32 bits (4 bytes)	Used in conjunction with the sequence number to insure that the data is received and in the correct order
Data offset	4 bits	Number of 32-bit words that the data payload is offset from the beginning of the TCP packet
Reserved	3 bits	Unused, should be 0
Flags	9 bits	Nine single flags for values like ACK, SYN, FIN, and RST
Window size	16 bits (2 bytes)	The number of bytes which can be sent without receiving an ACK from the destination
Checksum	16 bits (2 bytes)	Used to verify the integrity of the segment
Urgent Pointer	16 bits (2 bytes)	Points to urgent data within the segment
Options	0 - 10 32 bit words	There can be up to 320 bits of optional header information
Padding	Variable size	The TCP header must end on a 32 bit word boundary so padding will be added as necessary

3-way handshake

TCP achieves reliable delivery through a mechanism called the 3-way handshake which is used to establish a connection. For the purpose of this discussion, the client will be the host initiating a TCP/IP connection and the server will be host being connected to. The 3-way handshake

which takes place between a client and a server is a fairly simple process. The diagram and steps are outlined below.



1. The client sends a synchronize (SYN) message to the server. A SYN message is one in which the SYN flag is set in the Flags section of a TCP header. This is how the client notifies the server that it wants to reliably communicate with it. The client will also include an initial sequence number in the first packet. The sequence number establishes a baseline so that messages can be ordered correctly. The sequence number helps the server put the messages it has received from the client into their proper order. The sequence number also enables guaranteed delivery because the server can send a message back to the client verifying that specific messages from the client were received.
2. The second message is sent from the server to the client. The second message has two flags set, ACK and SYN. The ACK flag acknowledges to the client that the server received the first SYN message and it acknowledges that it received the sequence number by incrementing it by one and sending it back to the client. The SYN flag in the second message is the server initiating reliable communication with the client and the server sends its own sequence number.

3. The final message is the client sending back an ACK message to the server acknowledging that it received the SYN message from the server and acknowledging the sequence number sent by the server by incrementing it by one and sending it back.

This finishes the initiation of a reliable connection where both systems have sent their own sequence number to the other host and they have had their sequence numbers acknowledged. Note, that for simplicity the sequence numbers were made to be small values, 100 and 200. In reality, the TCP sequence numbers are large and randomly generated. Having their start value be a random number makes them less predictable and thus safer. In the 1980's and 1990's there was a hacker named Kevin Mitnick who was able to exploit TCP sequence number predictability (this was before the initial sequence number was randomized) and break into systems. He broke into the system of a graduate student named Tsuturo Shimomura. Shimomura was able to capture the packets and figure out how the attack took place. Mitnick was eventually caught and spent 5 years in prison. He now owns a computer security consulting business.

The termination of a TCP session is similar to the 3-way handshake with the communicating hosts exchanging FIN and ACK messages.

Routing refresher

```
netstat -rn
```

```
Kernel IP routing table
Destination     Mask            Gateway           Fl   MSS    If
132.236.227.0   255.255.255.0   132.236.227.93    U    1500   eth0
default         0.0.0.0         132.236.227.1     UG   1500   eth0
132.236.212.0   255.255.255.192 132.236.212.1     U    1500   eth1
132.236.220.64  255.255.255.192 132.236.212.6     UG   1500   eth1
127.0.0.1       255.255.255.255 127.0.0.1         U    3584   lo0
```

- The destination field is usually a network address
- The gateway must be a host address
- The second entry is a default route; packets not explicitly addressed to any of the three networks listed (or to the machine itself) will be sent to the default gateway host, 132.236.227.1
- Hosts can route packets only to gateway machines that are directly attached to their same network
- The final route is added at boot time. It configures a pseudo-device called the loopback interface. The loopback prevents packets sent from the host to itself from going out on the network; instead they are transferred directly from the network output queue to network input queue inside the kernel
- cat /proc/net/route
- route -n

References:

https://en.wikipedia.org/wiki/ARPANET
https://technet.microsoft.com/en-us/library/bb726991.aspx
Sams Teach Yourself TCP/IP in 24 Hours, Joe Casad
GSEC GIAC Security Essentials Certification Exam Guide, Ric Messier
All In One CISSP Exam Guide, Shon Harris
https://en.wikipedia.org