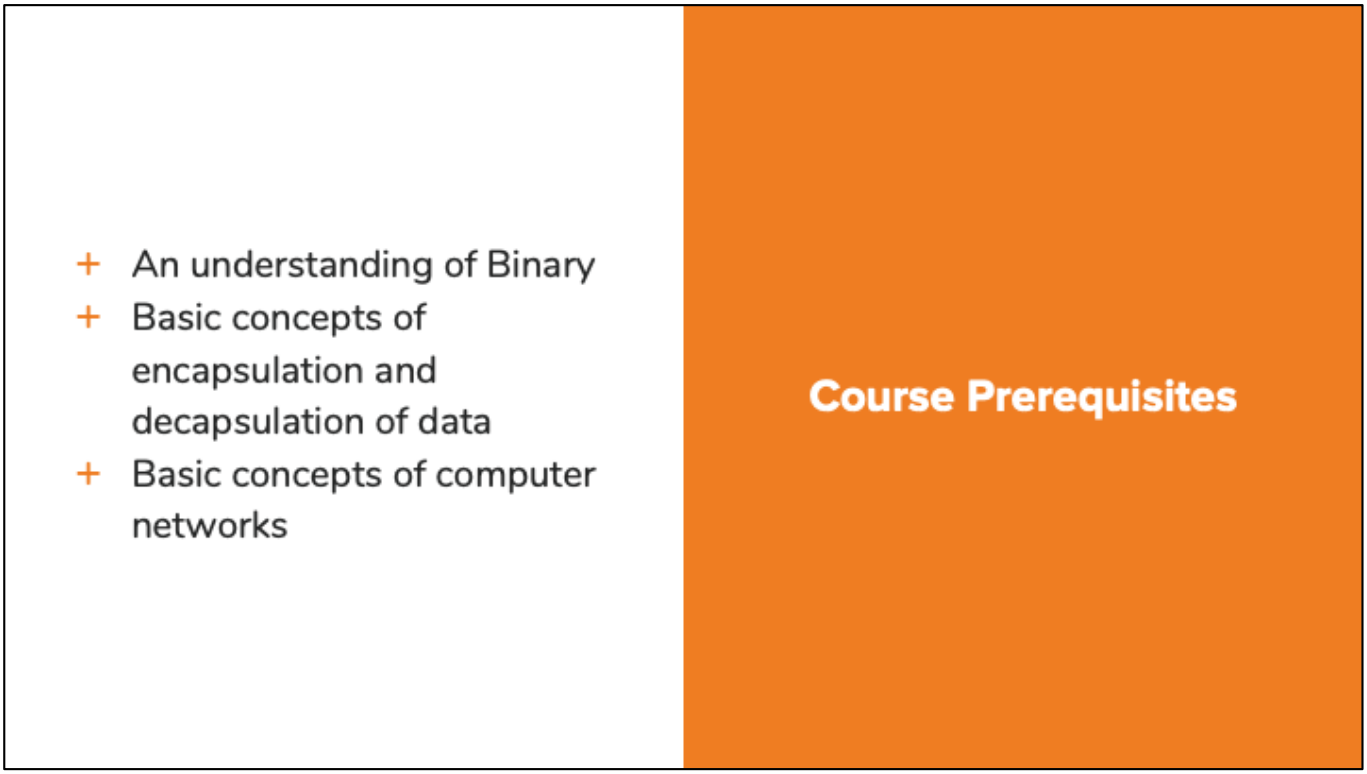

Introducing IPv4, UDP & TCP
ine.com


Keith Bogart
CCIE #4923
kbogart@ine.com
@keithbogart1
linkedin.com/in/keith-bogart-2a75042
CCIE Routing & Switching



- In order to understand the concepts in this course you need to know what a network is, (why we use them and their benefits)

Course Objectives

- To help you understand the purpose and use of the Internet Protocol
- To explain the fields in the IPv4 header
- To understand how IPv4 addresses are structured
- To summarize the differences between UDP and TCP
- To be able to explain how TCP/UDP port numbers are created and used
- To gain basic familiarity with how TCP implements a connection-oriented session


History & Purpose Of The Internet Protocol
ine.com

Topic Overview

+ Early Development Of Packet Switching
+ Concepts Of Packet Switching
+ Overview Of The Internet Protocol

Early Development Of Packet Switching

+ Early networks (late 1950s to mid 1960s) relied on dedicated telephone lines between disparate networks to transport data

+ Early 1960s, concepts of Packet Switching were developed by American computer scientist Paul Baran (concept was called *Distributed Adaptive Message Block Switching*)

It was of course, the military who originally provided the motivation for this development. In the late 1950s, the US Air Force established a wide area network for the Semi-Automatic Ground Environment (SAGE) radar defense system. They sought a system that might survive a nuclear attack to enable a response, thus diminishing the attractiveness of the first strike advantage by enemies. Baran developed his concept as a response to their needs.

Although another scientist (Donald Davies from the U.K.) was credited with actually coining the term “Packet Switching”, Paul Baran is the one who originally had the idea that data could be divided into small chunks, each chunk having its own distinct header (with an address) and these “packets” could be independently sent through a routed network.

At the time, once a channel (i.e. Phone circuit connection) was established between two offices, the bandwidth of that channel was 100% guaranteed, 24-hours a day for those two offices. So even when no data was being exchanged that bandwidth was available (although wasted).

Baran’s idea was that by dividing data into discreet, addressable packets, the bandwidth of a communications circuit would only be needed during the transmission of that packet. Once the packet was transmitted, the bandwidth would be freed to service someone else’s packet.

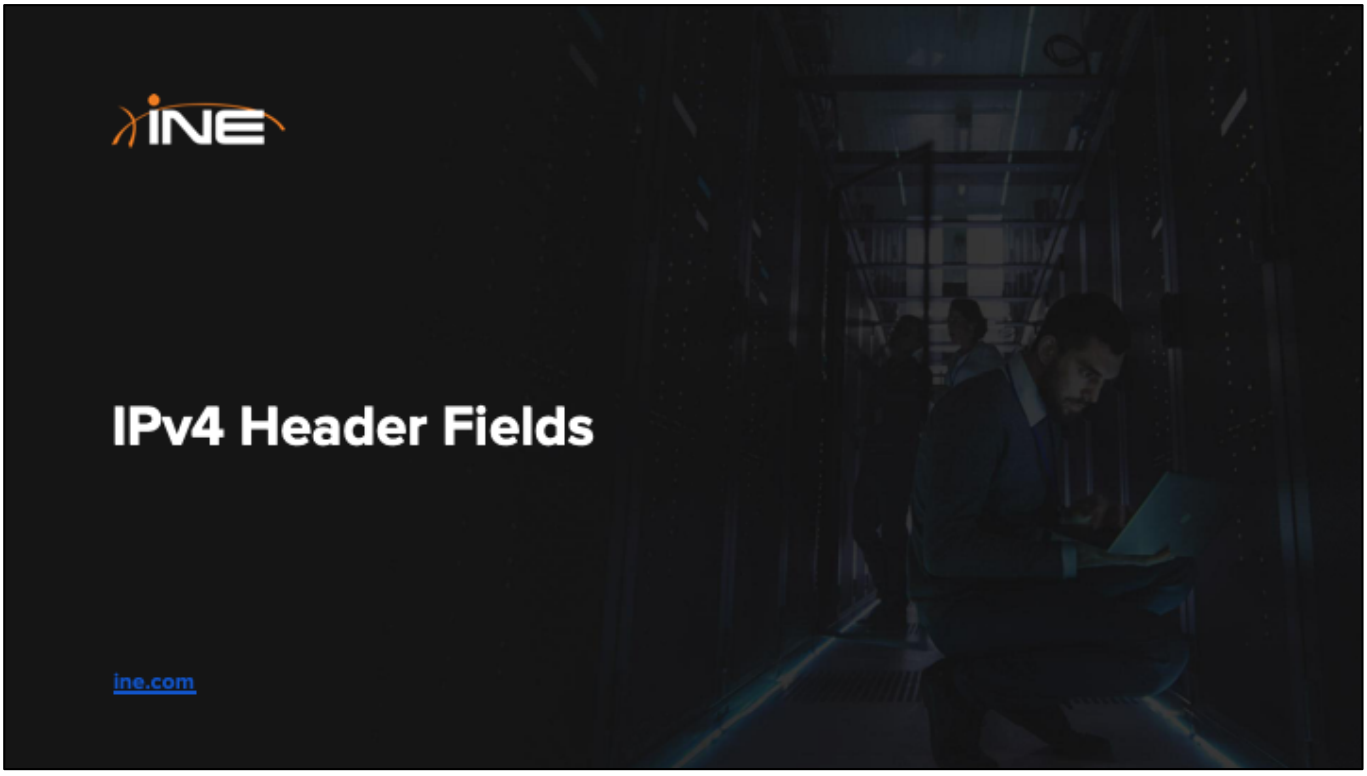
Concepts Of Packet Switching

- Packet switching implementation relies on three components:
  - Use of a decentralized network with multiple paths between any two points
  - Dividing user messages into *message blocks*
  - Delivery of these messages by store and forward switching
- 1973: U.S. Defense Advanced Research Projects Agency (DARPA) initiates a research program to investigate techniques and technologies for interlinking packet networks of various kinds
  - TCP/IP invented

Overview Of The Internet Protocol

- Internet Protocol version 4
- Resides at OSI Layer-3 (Network Layer)
- Connectionless
- Provides an encapsulation method (i.e. header) and addressing structure for data to be transmitted across routed networks

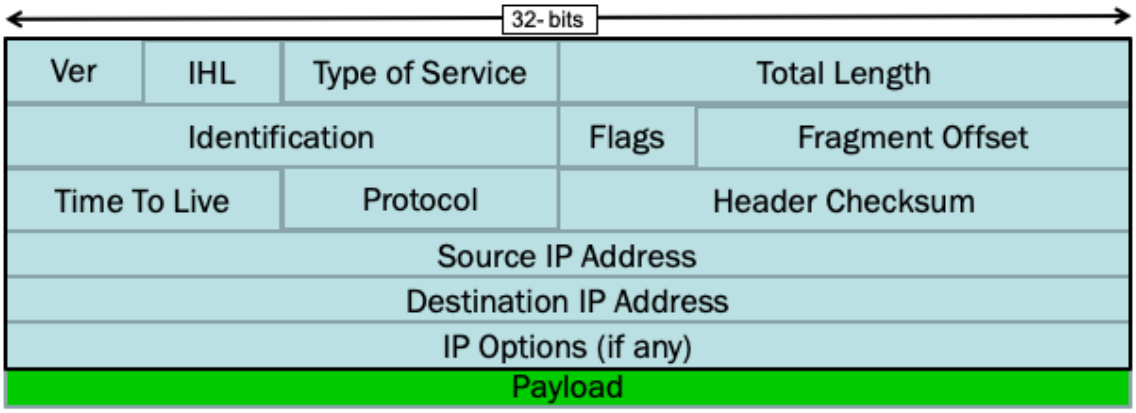
Thanks for Watching!


IPv4 Header Fields
ine.com

Topic Overview

+ IPv4 Header Fields

IPv4 Header Fields



IHL = number of 32-bit “words” in the IP header.
Minimum size of IP packet = 20bytes
Maximum size of IP packet = 65,535 bytes

Common IP Protocol Numbers:
ICMP = 1
IGMP = 2
TCP = 6
UDP = 17
EIGRP = 88
OSPF = 89

Mention that IP is not the ONLY protocol to work at Layer-3, there are other options:
IPv6, IPX, Appletalk, etc

Thanks for Watching!


Why Do We Need IPv4 Addresses?
ine.com

Topic Overview

+ Introduction To IPv4 Addressing
+ Network & Host Bits
+ Packet Routing Decisions

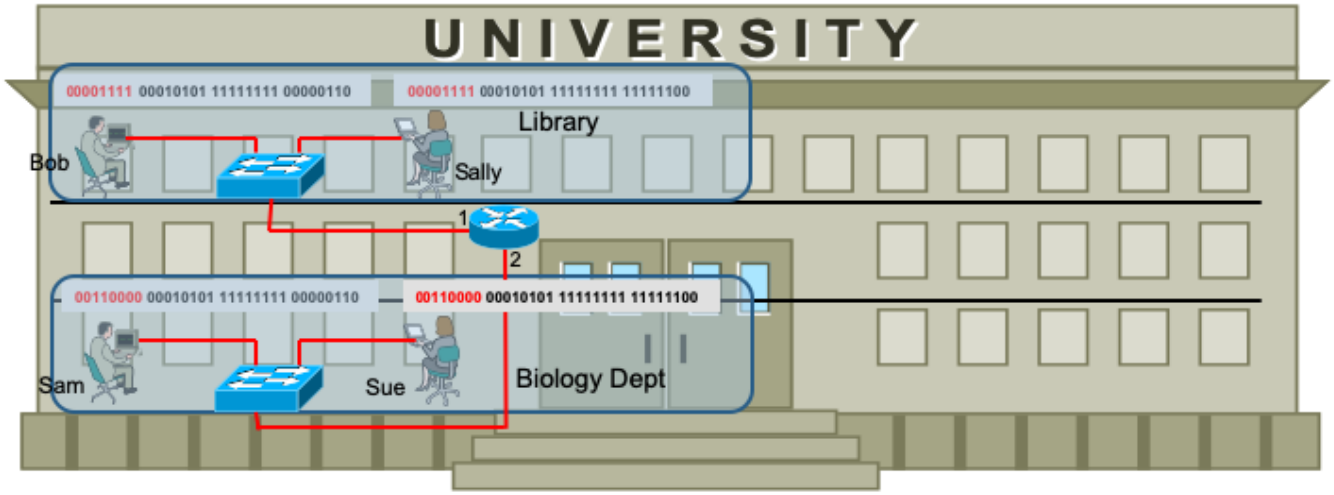
Introduction to IPv4 Addressing

- 32-bit addressing system
  11000000000000010000000100000011
- Logical address for a network defined by IANA
  - Network portion of address
  - Host portion of address
- IPv4 addresses are comprised of 4 octets
  11000000 00000001 00000001 00000011
- Dotted decimal notation is used to segment the octet
  192. 1. 1. 3

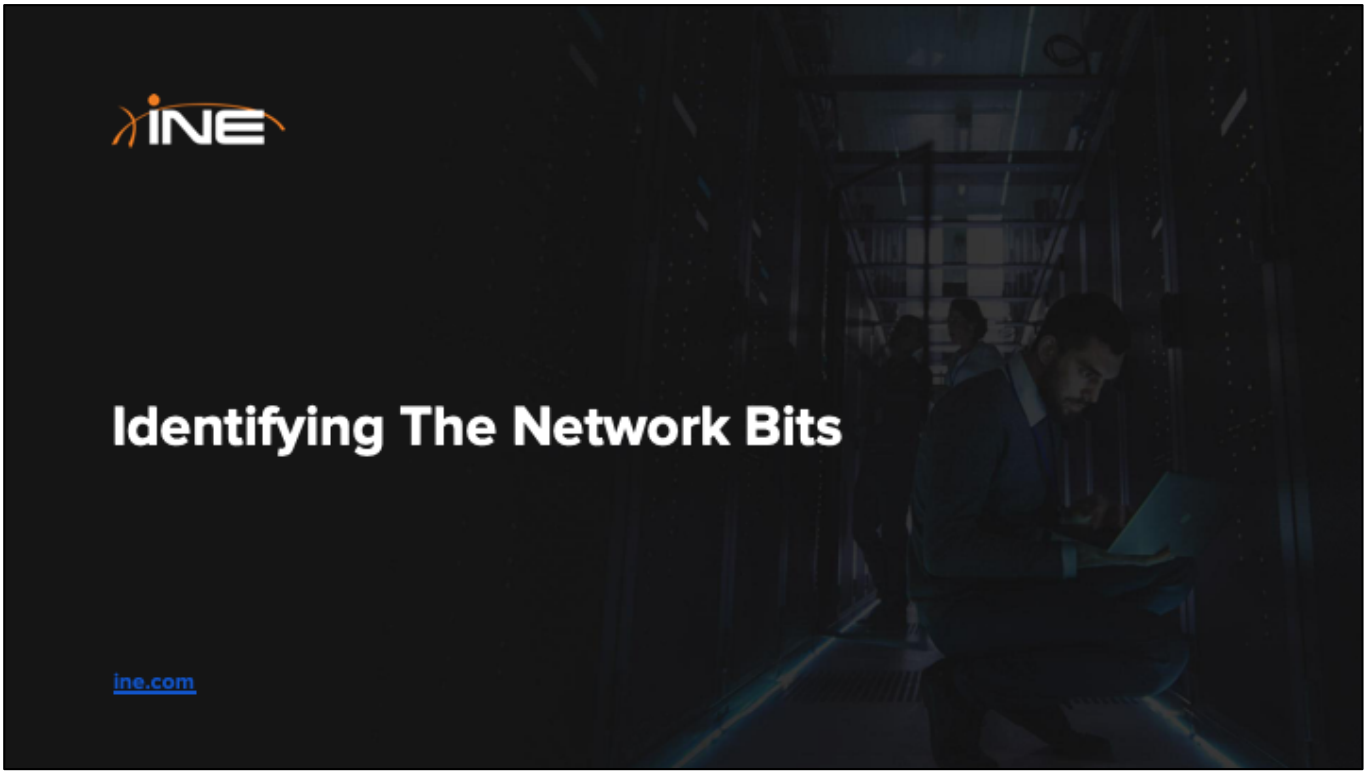
Network & Host Bits

- Devices with IPv4 addresses need to be able to identify the “network” portion of their address
  - Determines the necessity of directing packets to a default gateway for off-network destinations
  - Determines if ARP is needed to reach local hosts

Packet Routing Decisions



Thanks for Watching!


Identifying The Network Bits
ine.com

Topic Overview

+ IPv4 Address Classes
+ Determining IPv4 Address Classes
+ IPv4 Subnet Masks
+ IPv4 Address Types

IPv4 Address Classes

- Originally, IPv4 addresses were divided into Classes
  - Class was determined by specific bit patterns of first few bits in the address
  - Class of an address determined demarcation of "network" bits from "host" bits

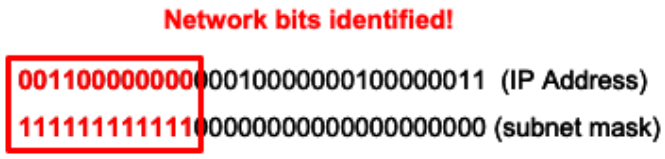
Determining IPv4 Address Classes

- Class-A:
  0xxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
- Class-B:
  10xxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
- Class-C:
  110xxxxx xxxxxxxx xxxxxxxx xxxxxxxx
- Class-D:
  1110xxxx xxxxxxxx xxxxxxxx xxxxxxxx
- Class-E:
  11110xxx xxxxxxxx xxxxxxxx xxxxxxxx

Class-D addresses were utilized for multicast traffic, more on that in the next slide. Class-E was experimental and was never used. Even today, computer operating systems will not allow you to configure a network interface with a Class-E address.

IP Subnet Masks

- Classful addressing was not scalable
- Another method was needed to arbitrarily set the boundary between network and host bits
- Subnet mask introduced in 1985 with RFC 950
- CIDR (introduced in 1993) formally deprecated classful addressing

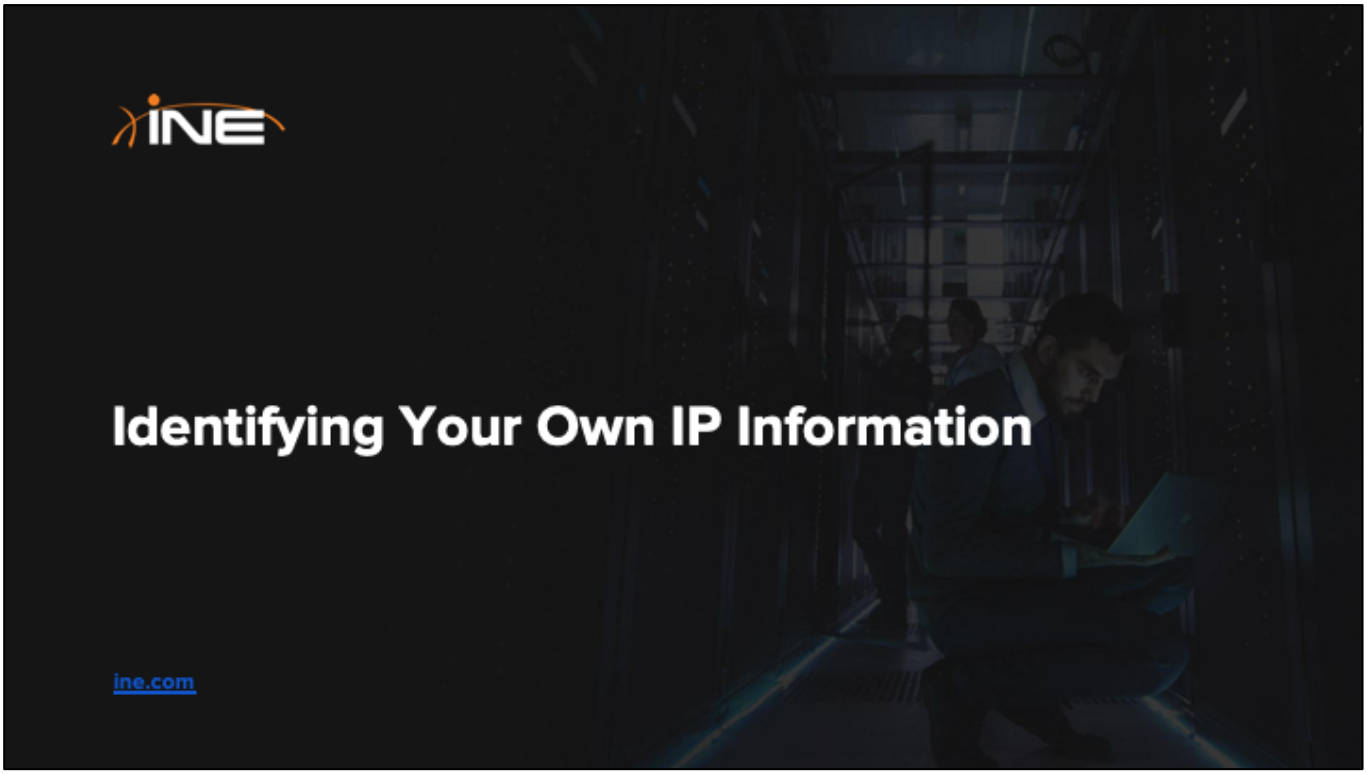


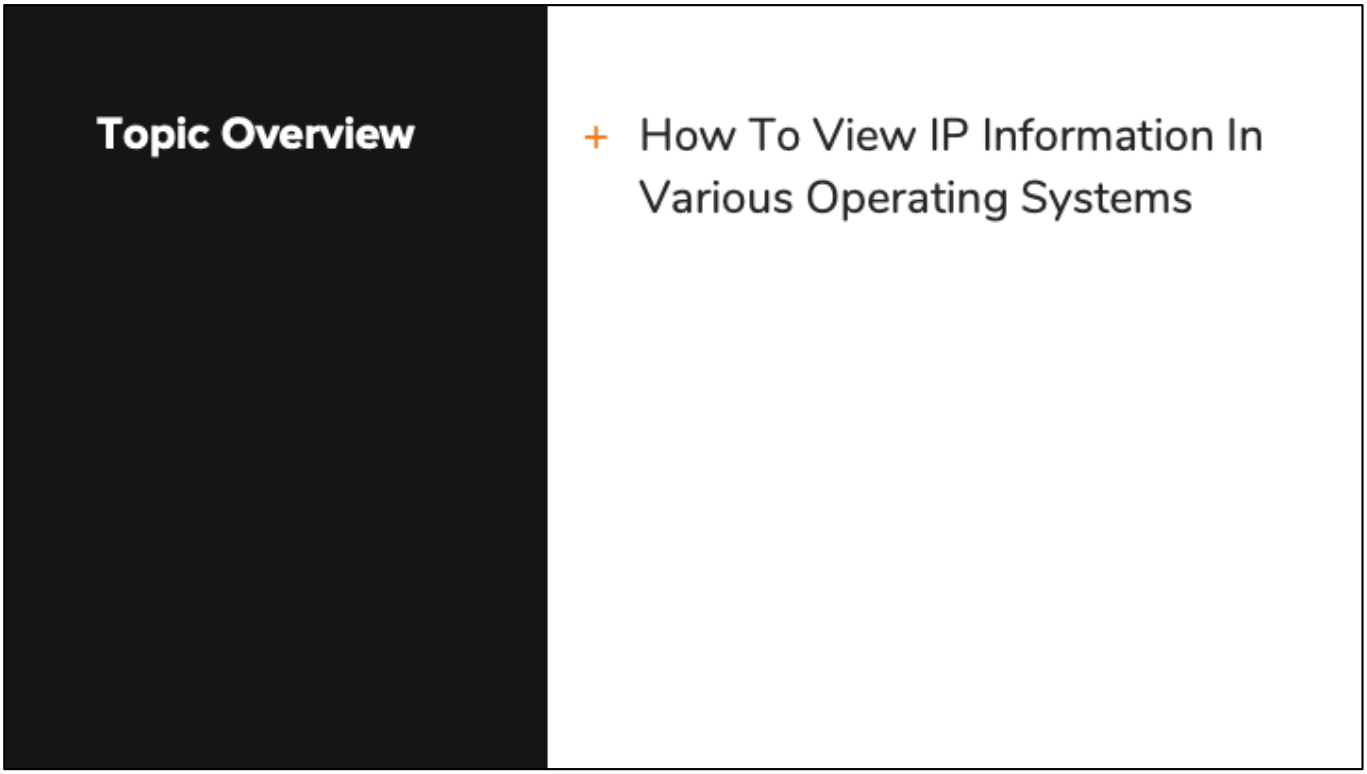
CIDR (Classless Interdomain Routing) was introduced IN 1993 to replace classful addressing which formally deprecated classful addressing and incorporated subnetting which had been introduced 8-years earlier with RFC 950

IPv4 Address Types

- Unicast
  - One-to-one communication
  - IPv4: Utilizes Class-A, B & C address space
- Multicast
  - One-to-many communication
  - IPv4: Utilizes Class-D address space
- Broadcast
  - One-to-all communication
  - General broadcast
  - Directed broadcast

Thanks for Watching!


Identifying Your Own IP Information
ine.com



I'll be showing you how to view your own IP address, subnet mask and default gateway information whether you're using a Windows OS, Linux OS or Mac OS

Viewing IP Information In Windows

- Open the DOS Command window
- Enter the command “ipconfig”
- Alternatively:
  - Open “Settings”
  - Select “Network and Internet”
  - Select “Ethernet” (or whatever network adapter you are using)
  - Select “Network & Sharing Center”
  - Click on “Ethernet” followed by “Details”

Viewing IP Information In Mac OS

- Open System Preferences
- Select the “Network” icon
  - IP and Subnet Mask are now displayed
- Click on “Advanced”
  - Default gateway (Router) is now displayed
  - Other items are now viewable

Viewing IP Information In Linux

- Open the Terminal
- From the CLI type the command, “ifconfig”
  - IP address is now displayed
  - Subnet mask is now displayed (in hexadecimal)
- From the CLI type “netstat –rn”
  - IPv4/IPv6 routing table is now displayed
  - You can view your default gateway
- Alternative command, “ipconfig getpacket <interface>
  - Replace “interface” with the interface that contained your IP address in the output of “ifconfig”

Thanks for Watching!

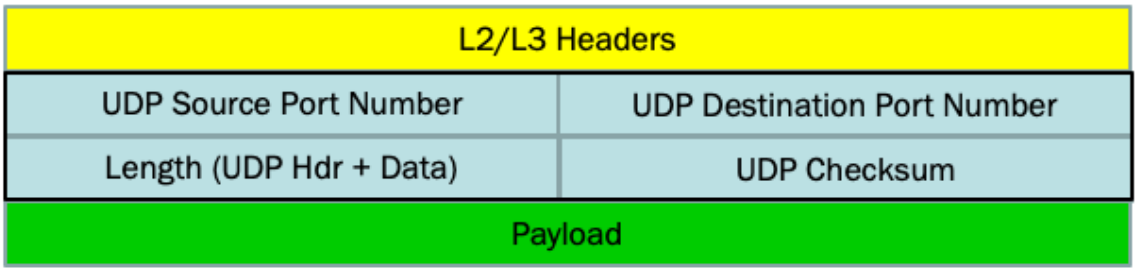

Overview Of UDP
ine.com

Topic Overview

+ Introduction To UDP
+ Transport Layer Port Numbers
+ How Port Numbers Work

Introduction To UDP

- User Datagram Protocol
  - PDUs created with UDP headers called, "Datagrams"
  - Documented in RFC 768
- Connectionless
- UDP header placed after the IP header
- Often used by time-sensitive applications intolerant of delay

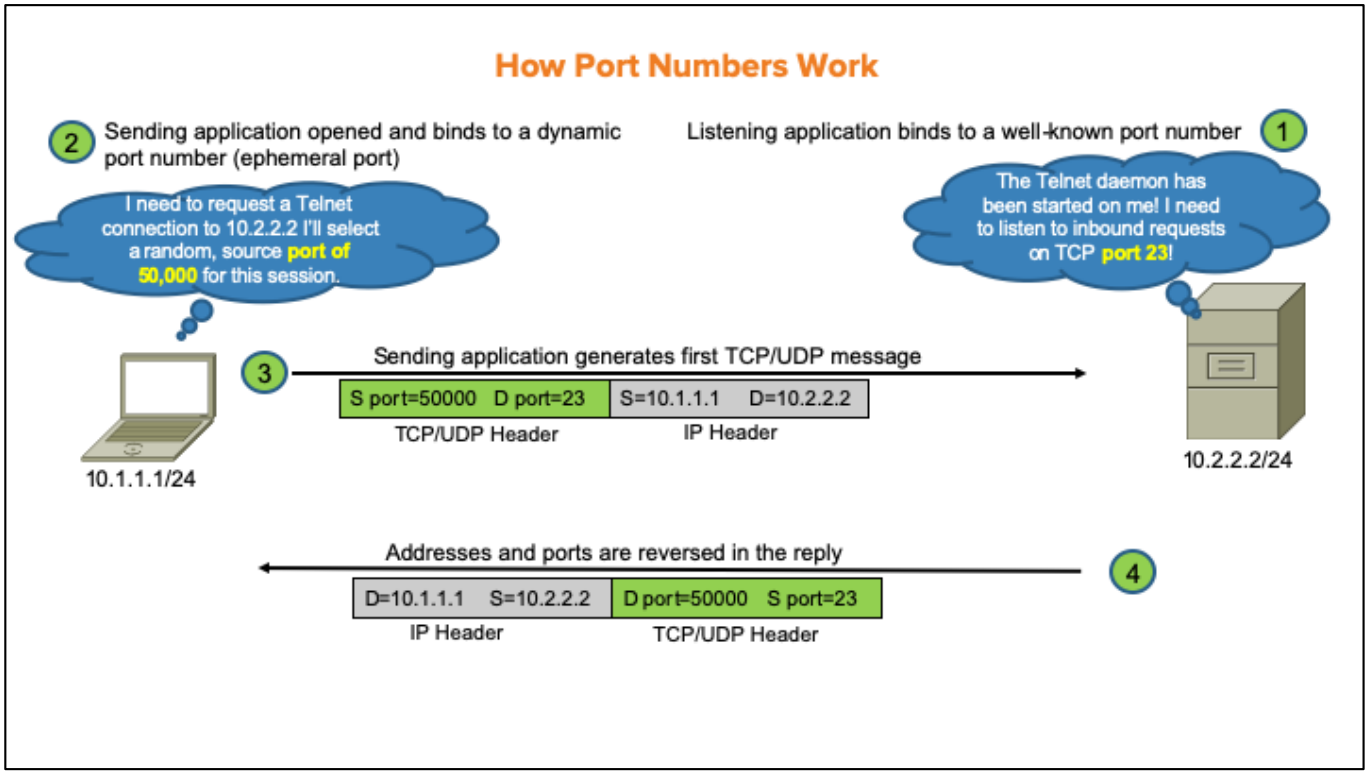


Examples of common protocols that use UDP:
TFTP – 69
DNS – 53 (can also operate with TCP)
RIP (520)

Transport-Layer Port Numbers

- TCP/UDP Port numbers:
  - Logical entities
  - Used by end system applications to bind their transport sessions to
- Every TCP/UDP segment/datagram has a source and destination port number
  - Destination port typically a well-known, registered application port
  - Source port typically randomly derived
- TCP/UDP ports separated into three (3) ranges by the IANA
  - Port numbers 0 through 1023 are used for common, well-known services
  - Port numbers 1024 through 49151 are the registered ports used for IANA-registered services
  - Ports 49152 through 65535 are dynamic ports (ephemeral ports) that are not officially designated for any specific service, and may be used for any purpose

- The port numbers in the range from 0 to 1023 are the well-known ports or system ports. They are used by system processes that provide widely used types of network services. On Unix-like operating systems, a process must execute with superuser privileges to be able to bind a network socket to an IP address using one of the well-known ports
- The range of port numbers from 1024 to 49151 are the registered ports. They are assigned by IANA for specific service upon application by a requesting entity.[1] On most systems, registered ports can be used by ordinary users.


How Port Numbers Work
2
Sending application opened and binds to a dynamic port number (ephemeral port)
I need to request a Telnet connection to 10.2.2.2 I'll select a random, source port of 50,000 for this session.
Listening application binds to a well-known port number
1
The Telnet daemon has been started on me! I need to listen to inbound requests on TCP port 23!
3
Sending application generates first TCP/UDP message
S port=50000 D port=23
S=10.1.1.1 D=10.2.2.2
TCP/UDP Header
IP Header
10.1.1.1/24
10.2.2.2/24
Addresses and ports are reversed in the reply
4
D=10.1.1.1 S=10.2.2.2
D port=50000 S port=23
IP Header
TCP/UDP Header

Thanks for Watching!


Introduction To TCP
ine.com

Topic Overview

+ TCP History
+ TCP Compared To UDP
+ Connection Oriented Defined
+ TCP Header Fields

TCP History

- Developed by Stanford University in 1970's
- Originally TCP and IP were part of the same standard called TCP (Internet Transmission Control PROGRAM)
- IP is the protocol suite, ARPANET was the network infrastructure
- TCP was de-coupled from IP to stay consistent with a layered approach to networking

TCP originally defined in RFC 793

Originally designed to run on the United States Defense Advanced Research Projects Agency network (DARPA or ARPA).

Originally the ARPAnet used a protocol called NCP (Network Control Protocol) but that was found to be limited.

Development began on a new protocol that would be better suited to a growing internetwork. This new protocol, first formalized in RFC 675, was called the Internet Transmission Control Program (TCP).

TCP Compared To UDP

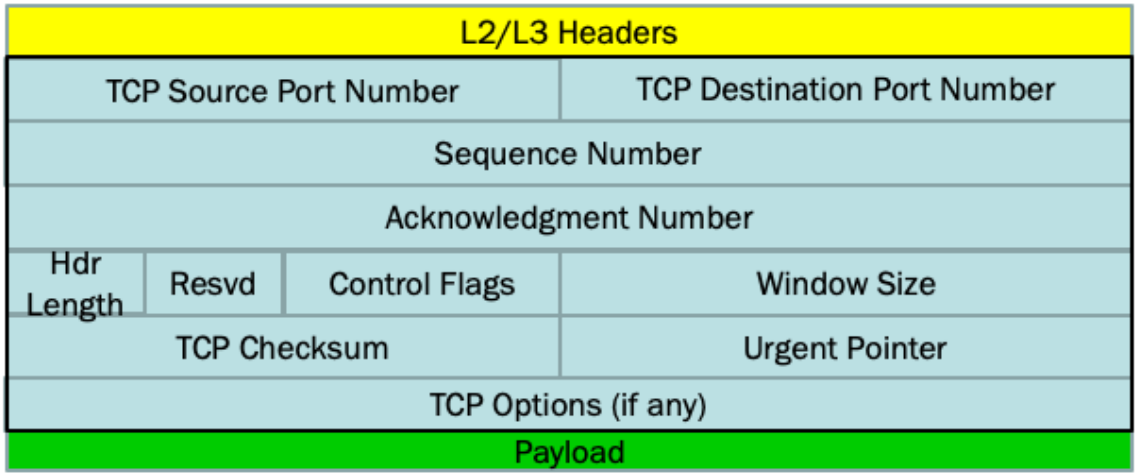
- What makes TCP different from UDP?
- UDP
  - Connectionless
  - Expects to receive small, discrete messages from upper-layer protocols
  - No reliability or retransmission
- TCP
  - Connection-Oriented
  - Can work with any format of data (including a constant stream of bytes) from upper-layer protocols
  - Reliability with Acknowledgements and Retransmissions

UDP was not designed to receive large, long blocks of data. If a long, continuous stream of bytes were sent down to UDP, it has no mechanism to answer the question, “at what point do I package this into a datagram?”

Connection Oriented

- What is meant by “connection oriented”?
  - TCP verifies existence of peer prior to data exchange
  - TCP peers negotiate parameters used to control data exchange
  - TCP data is exchanged reliably using sequence numbers, acknowledgements, flow-control, and retransmissions
  - TCP can gracefully inform peer of the need to close a connection

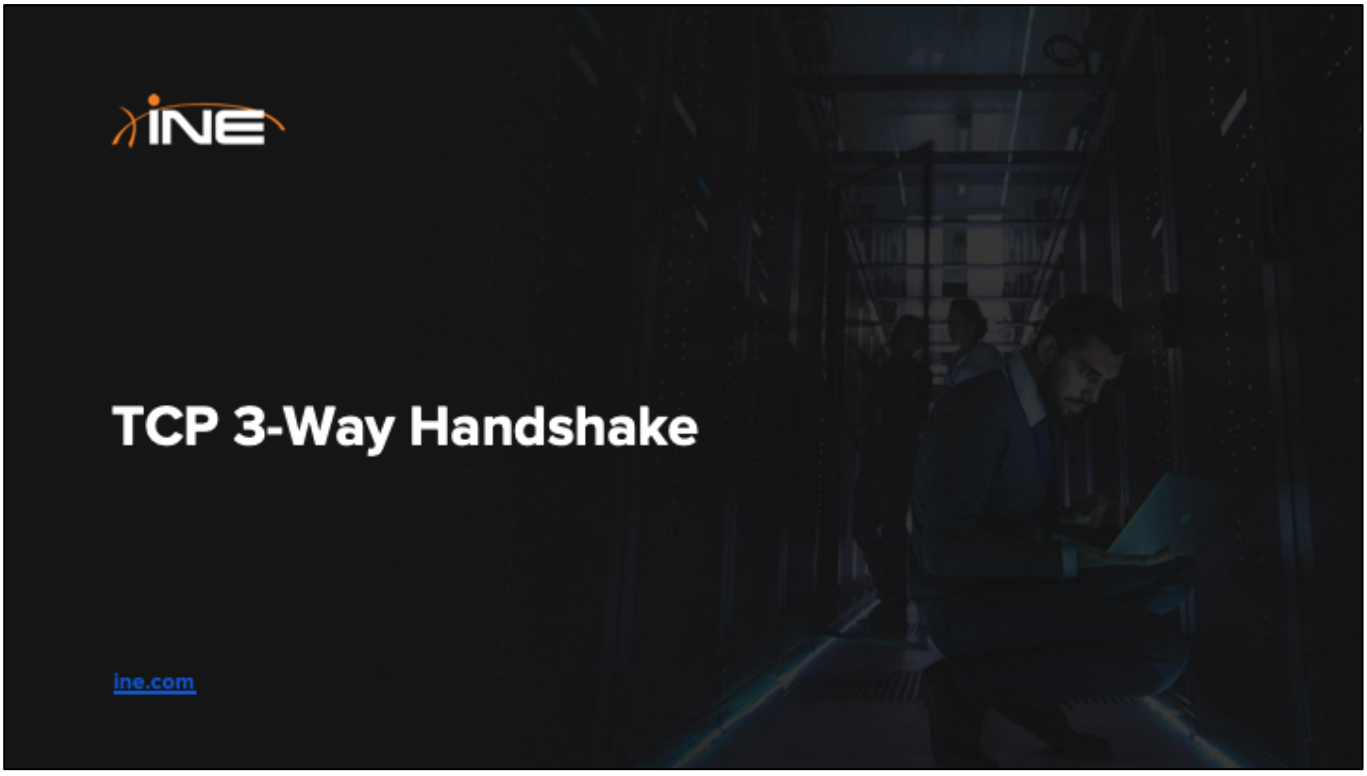
TCP Header Fields



In this section let's focus primarily on everything BUT the Control Flags. We'll look more closely at those in the next video.

Window Size = Maximum quantity of segments (in bytes) the sender is willing to accept.
Talk about: MSS (Maximum Segment Size) that is sent during TCP 3-way handshake.
Talk about: Common protocols that use TCP:
Telnet (23)
FTP (20 and 21)
POP3 (110)
SMTP (25)
HTTP (80)
HTTPS (443)

Thanks for Watching!


TCP 3-Way Handshake
ine.com

Topic Overview

+ Purpose Of The 3-Way Handshake
+ TCBs & Sockets
+ TCP Active Opens
+ TCP Passive Opens
+ TCP Header Fields
+ Creating TCP Connections

Purpose Of The Three-Way Handshake

- The TCP 3-Way Handshake
  - Starts the process of “Opening” a TCP connection
- Main objectives of the handshake:
  - Initial contact and proof-of-existence
  - Sequence Number Synchronization
  - Exchange of Parameters

TCP Terminology: TCBs and Sockets

- TCB = Block of memory space allocated by CPU to maintain state-information for a single TCP session
- Contain TCP "Socket" information
- Creation of TCBs can happen in one-of-two ways:
  - Active Opens
  - Passive Opens

TCP Socket is a combination of four parts of data:
---Src IP: Src TCP port number
---Destination IP: Dest TCP Port Number

So clients and servers typically maintain multiple TCBs at one time…each TCB containing socket information for a single flow of bi-directional traffic.

TCB contains:
---socket numbers that identify
---pointers to buffers where incoming and outgoing data are held.
---implement the sliding window mechanism.
---holds variables that keep track of the number of bytes received and acknowledged, bytes received and not yet acknowledged, current window size and so forth.

TCP Active Opens

- TCP Clients:
  - Clients have many potential applications that could use TCP: FTP, Email, HTTP, etc
  - TCB on Client is not created until an Application requests the services of TCP
- Client pre-determines elements required for TCP socket
- TCB created utilizing socket information, ISN, etc. TCP "SYN" transmitted

ISN = Initial Sequence Number (more on this later)

Socket information determined by:
SRC IP: Client already knows this via DHCP or static assignment
SRC Port: Most likely an ephemeral port

Dest IP: Probably determined via DNS lookup or static assignment.
Dest Port: Well-known TCP port for that protocol.

TCP Passive Opens

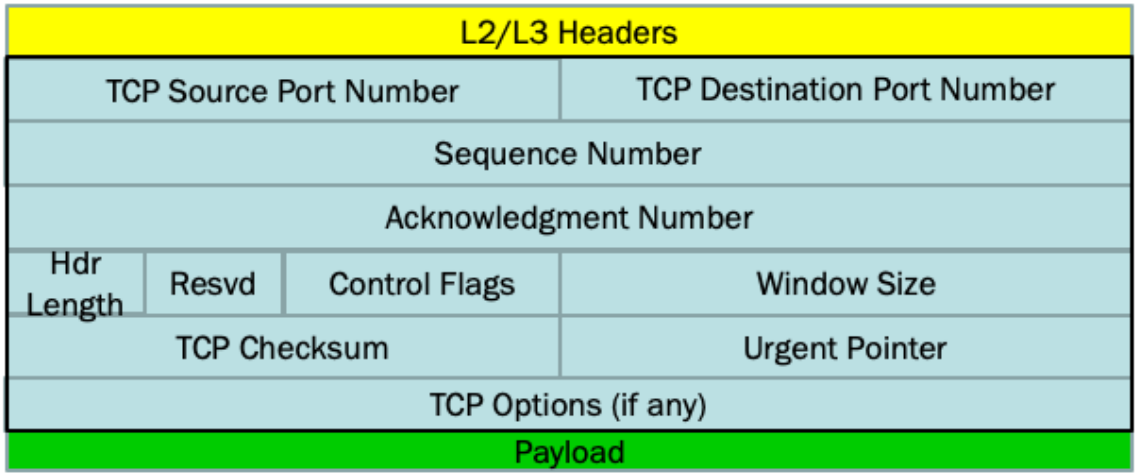
- TCP Servers:
  - Typically designed to only recognize certain TCP applications
  - TCB created “in advance” to allow capability of listening for any incoming requests
- Server pre-determines elements required for TCP socket
- TCB created implementing partial socket information, ISN, etc
- TCB “passively” waits for any incoming requests

Socket information determined by:
SRC IP: Client already knows this via DHCP or static assignment
SRC Port: Well-known TCP port for that protocol

Dest IP: This field is left blank until incoming client is detected.
Dest Port: This field is left blank until incoming client is detected.

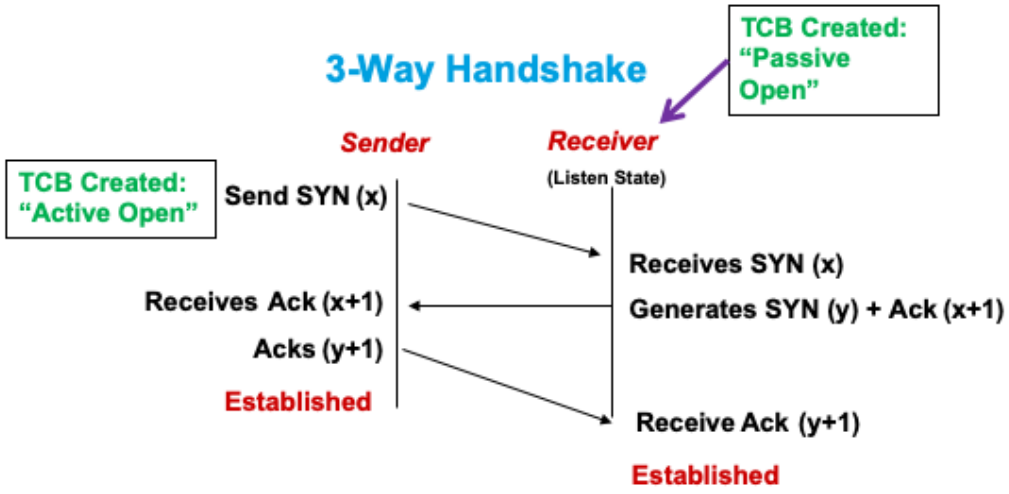
This is called an “unspecified passive open”. Client socket (after receiving a “sync”) is bound to this unspecified passive open.

TCP Header Fields



In order to talk about the TCP 3-way handshake, the main field we need to understand is the Control Flags field.

Creating TCP Connections



Thanks for Watching!