Networking And Scripting

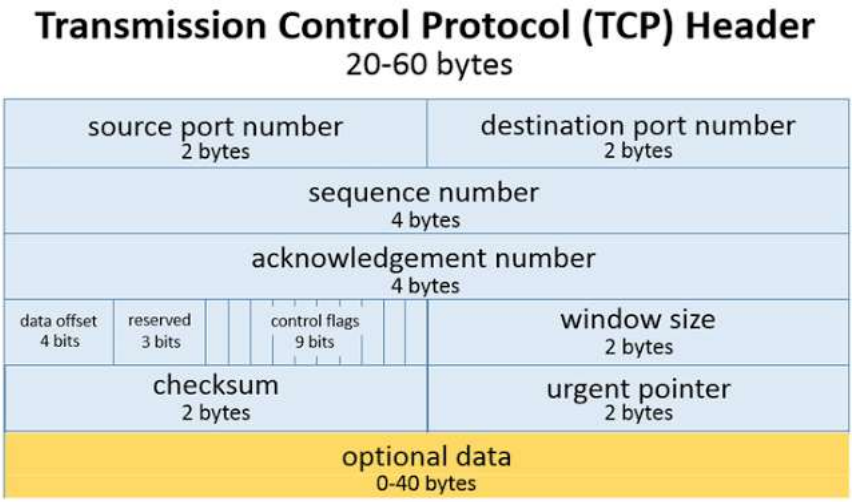
MONDAY, 2 JANUARY 2017

Packet Formats to Remember

0x0800	Internet Protocol version 4 (IPv4)
0x0806	Address Resolution Protocol (ARP)
0x888E	EAP over LAN (IEEE 802.1X)

0x8100	VLAN-tagged frame (IEEE 802.1Q)

0x8035	Reverse Address Resolution Protocol



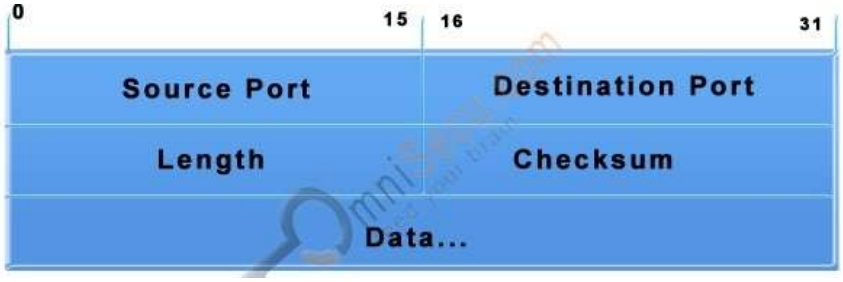
TCP Header Format

Source port (16 bits) : identifies the sending port.
Destination port (16 bits): identifies the receiving port.
Sequence number (32 bits):Has a dual role:
If the SYN flag is set (1), then this is the initial sequence number. The sequence number of the actual first data byte and the acknowledged number in the corresponding ACK are then this sequence number plus 1.
If the SYN flag is clear (0), then this is the accumulated sequence number of the first data byte of this segment for the current session.
Acknowledgement number (32 bits)
if the ACK flag is set then the value of this field is the next sequence number that the receiver is expecting. This acknowledges receipt of all prior bytes (if any). The first ACK sent by each end acknowledges the other end's initial sequence number itself, but no data.
Data offset (4 bits)
specifies the size of the TCP header in 32-bit words. The minimum size header is 5 words and the maximum is 15 words thus giving the minimum size of 20 bytes and maximum of 60 bytes, allowing for up to 40 bytes of options in the header. This field gets its name from the fact that it is also the offset from the start of the TCP segment to the actual data.
Reserved (3 bits):for future use and should be set to zero
Flags (9 bits) (aka Control bits)
contains 9 1-bit flags
NS (1 bit) – ECN-nonce concealment protection (added to header by RFC 3540).
CWR (1 bit) – Congestion Window Reduced (CWR) flag is set by the sending host to indicate that it received a TCP segment with the ECE flag set and had responded in congestion control mechanism (added to header by RFC 3168).
ECE (1 bit) – ECN-Echo has a dual role, depending on the value of the SYN flag. It indicates:
If the SYN flag is set (1), that the TCP peer is ECN capable.
If the SYN flag is clear (0), that a packet with Congestion Experienced flag in IP header set is received during normal transmission (added to header by RFC 3168).
URG (1 bit) – indicates that the Urgent pointer field is significant

ACK (1 bit) – indicates that the Acknowledgment field is significant. All packets after the initial SYN packet sent by the client should have this flag set.
PSH (1 bit) – Push function. Asks to push the buffered data to the receiving application.
RST (1 bit) – Reset the connection
SYN (1 bit) – Synchronize sequence numbers. Only the first packet sent from each end should have this flag set. Some other flags and fields change meaning based on this flag, and some are only valid for when it is set, and others when it is clear.
FIN (1 bit) – No more data from sender
Window size (16 bits)
the size of the receive window, which specifies the number of window size units (by default, bytes) (beyond the sequence number in the acknowledgment field) that the sender of this segment is currently willing to receive (see Flow control and Window Scaling)
Checksum (16 bits)
The 16-bit checksum field is used for error-checking of the header and data
Urgent pointer (16 bits)
if the URG flag is set, then this 16-bit field is an offset from the sequence number indicating the last urgent data byte

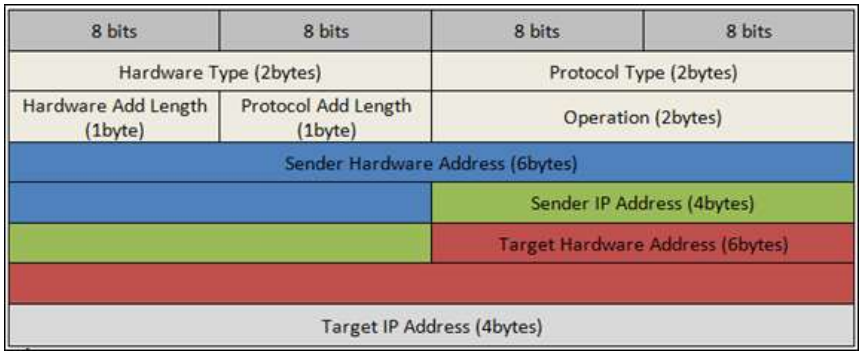
Options (Variable 0–320 bits, divisible by 32)

UDP



Source Port Number: The first 16 bits of the UDP header contain the port number of the application sending the data.
Destination Port Number: The next 16 bits contain the port number of the application that receives this data.
Length: The next 16 bits identify how long the datagram is in bits**.**
Checksum: The last 16 bits of the UDP header are reserved for the checksum value. Checksum is used as an error-detection mechanism. The source machine runs a mathematical algorithm on the datagram. The destination, or recipient, machine runs the same mathematical algorithm on the datagram. If the both values match we can assume that the datagram wasn't damaged while its journey.
The checksum field includes a 12-byte 'pseudo header' that includes the source and destination IP addresses, the 8-bit reserved field containing 0, the 8-bit protocol ID and the 16-bit UDP length field. The pseudo header is useful to check that the IP datagram arrived at the correct station.
Important protocols which use UDP: TFTP, DNS, SNMP, LDAP.

ARP:



Hardware Type [2 bytes]: It specifies the type of hardware used for the local network transmitting the ARP message. *Ethernet is the common Hardware Type and he value is 1. The size of this field is 2 bytes.*
Protocol Type [2 bytes]: Each protocol is assigned a number used in this field*, IPv4 is 2048 (0x0800 in Hexa).*

Hardware Address Length: Hardware Address Length in the ARP Message is length in bytes of a hardware (MAC) address. *Ethernet MAC addresses are 6 bytes long.*

Protocol Address Length: Length in bytes of a logical address (IPv4 Address). *IPv4 addresses are 4 bytes long.*

Opcode [Operation] [2 bytes]: Opcode field in the Address Resolution Protocol (ARP) Message specifies the nature of the ARP message. *1 for ARP request and 2 for ARP reply.*

Sender Hardware Address [4 bytes]: Layer 2 [MAC] address of the device sending the message.

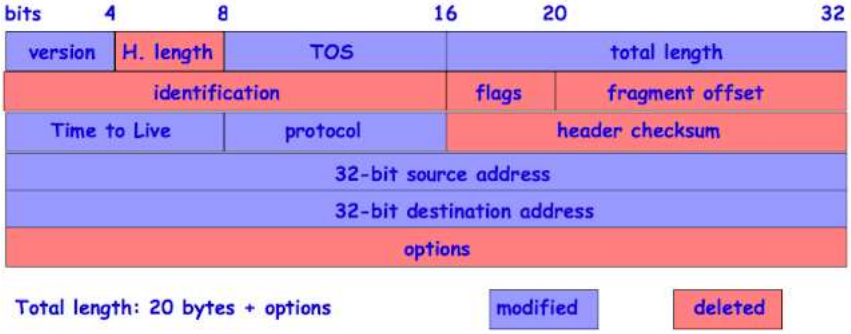
Sender IP Address [4bytes]: The protocol address (IPv4 address) of the device sending the message

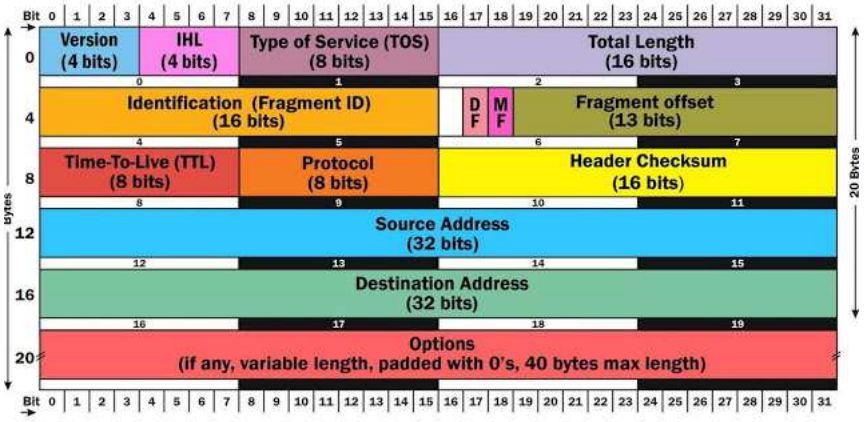
Target Hardware Address [6 bytes]: Layer 2 [MAC] address of the intended receiver. This field is ignored in requests.

Target IP Address [4 bytes]: The protocol address (IPv4 Address) of the intended receiver.

Networking Interview Q&
PROXY ARP
Packet Flow through Cisc
Packet Formats to Reme
Packet Transmission: Ro
Protocol Stack Layers
Ping And Traceroute
Puzzles and Riddles
Python 2.7 Vs 3.4
Python Basic Programs-1
Python Basic Programs-2
Python Basic Programs-3
Python Basics -1
Python Basics -2
Python Basics:Cheat She
Python Dictionaries & Dic
Python Functions
Python IN Nutshell
Python Important Notes
Python Interview Questio
Python Interview Questio
Python Interview Questio
Python Lists & Lists Prog
Python OOPS Basic Exa
Python OOPS Basic Tips
Python Quick Reference
Python Script:Generate F
Python Sets & Sets Progr
Python String & String Pr
Python Tuple & Tuple Pro
Python: (_ & __) in variab
Python: Difference betwe
Python: Interview Questic
Python: Removing Duplic
Python: Script to open a
Python: Subinterfaces on
Python: if __name__ ==
Python:Assertion
Python:Basic Notes
Python:Basic Script Expla
Python:Decorators
Python:Fibonacci Sequer
Python:IMP Programs to
Python:Lambda Expressi
Python:Module & Packag
Python:Reverse a String
RADIUS [Remote Authen
RIB Vs FIB
Rapid Spanning Tree Pro
Routing Basics
STP NOTES
STP Vs RSTP
STP-BPDU GAURD AND
STP-BPDU TYPES
STP-PortFast
STP-ROOT GUARD
Scenarios Based Q&A
Spanning Tree Interview
Spanning Tree Protocol
Spanning Tree Protocol:F
Spirent [STC] points to re
Static Route Explanation

IPv4 Header Format





- **Version**: - The first header field in an IP packet is the four-bit version field. Version identifies the IP version to which the packet belongs. This four-bit field is set to binary 0100 to indicate version 4 (IPv4) or binary 0110 to indicate version 6 (IPv6).

- **Header length or Internet Header Length (IHL) :-** The second field (4 bits) is the Internet Header Length (IHL) telling the number of 32-bit words in the header. Since an IPv4 header may contain a variable number of options, this field specifies the size of the header (this also coincides with the offset to the data). The minimum value for this field is 5 , which is a length of 5×32 = 160 bits = 20 bytes. Being a 4-bit value, the maximum length is 15 words (15×32 bits) or 480 bits = 60 bytes.

- **Type of Service (ToS):–** now known as **Differentiated Services Code Point (DSCP).** The TOS field is used to carry information to provide quality of service features. New technologies are emerging that require real-time data streaming and therefore make use of the DSCP field. An example is Voice over IP (VoIP) that is used for interactive data voice exchange.

- **Total Length:-** This 16-bit field defines the entire datagram size, including header and data, in bytes. The minimum-length datagram is 20 bytes (20-byte header + 0 bytes data) and the maximum is 65,535 bytes — the maximum value of a 16-bit word. The minimum size datagram that any host is required to be able to handle is 576 bytes, but most modern hosts handle much larger packets. Sometimes subnetworks impose further restrictions on the size, in which case datagrams must be fragmented. Fragmentation is handled in either the host or packet switch in IPv4.
- **Identification: –** This field is an identification field and is primarily used for uniquely identifying fragments of an original IP datagram. Some experimental work has suggested using the ID field for other purposes, such as for adding packet-tracing information to datagrams in order to help trace back datagrams with spoofed source addresses.
- **Flags:–**A three-bit field follows and is used to control or identify fragments. They are (in order, from high order to low order):
  - Bit 0: Reserved; must be zero.
  - bit 1: Don't Fragment (DF)
  - bit 2: More Fragments (MF)
- **Don't Fragment: -** Sets the Don't Fragment bit in sent packets. When an IP datagram has its DF flag set, intermediate devices are not allowed to fragment it so if it needs to travel across a network with a MTU (Maximum Transmission Unit) smaller that datagram length the datagram will have to be dropped. Normally an ICMP Destination Unreachable message is generated and sent back to the sender.
- **More Fragments: -** Sets the More Fragments bit in sent packets. The MF flag is set to indicate the receiver that the current datagram is a fragment of some larger datagram. When set to zero it indicates that the current datagram is either the last fragment in the set or that it is the only fragment.
- **Fragment Offset:-**The fragment offset field, measured in units of eight-byte blocks, is 13 bits long and specifies the offset of a particular fragment relative to the beginning of the original unfragmented IP datagram. The first fragment has an offset of zero. This allows a maximum offset of ($2^{13} – 1$) × 8 = 65,528 bytes which would exceed the maximum IP packet length of 65,535 bytes with the header length included (65,528 + 20 = 65,548 bytes).
- **Time To Live (TTL):-**It is of 8 bit field. This field indicates the maximum time the datagram is allowed to remain in the internet system. If this field contains the value zero, then the datagram must be destroyed. This field is modified in internet header processing. The time is measured in units of seconds, but since every module that processes a datagram must decrease the TTL by at least one even if it process the datagram in less than a second, the TTL must be thought of only as an upper bound on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded, and to bound the maximum datagram lifetime. <hops> must be a number in the range [0–255].

TCL EXPECT TUTORIAL
TCL File Handling with E
TCL IMP THINGS TO RE
TCL INTERVIEW QUEST
TCL Interview Question -
TCL List & Basic List Pro
TCL NAMESPACE
TCL Overview
TCL PACKAGES
TCL PROC
TCL String
TCL-Arrays with Example
TCL-Regular Expression
TCL-Regular Expression
TCP Interview Questions
TCl Basic Programs
TestCases for Elevator
Traceroute "Working & E
VLAN & VLAN Types
VPN:Detailed Notes
VRRP virtual MAC addre
VRRP-Explanation with E
Virtual Private Network:V
What happens when you
Writing Test Cases:Basic

BLOG ARCHIVE
► 2016 (35)
▼ 2017 (71)
▼ January (14)
TCL NAMESPACE
Packet Formats to Remember
BGP Overview
BGP MCQ
BGP Q&A
EAP: Extensible Authentication Protocols
BGP Basics
IEEE 802.1X (dot1x) Port Based Authentication
DHCP OPTION 82
BGP "IMP" NOTES
BGP In Nutshell "With"Interview Question on BGP
BGP Scenarios Based Q&A
Routing Basics
RIB Vs FIB
► February (13)
► March (3)
► August (14)
► September (6)
► October (12)
► November (9)
► 2018 (24)
► 2019 (1)

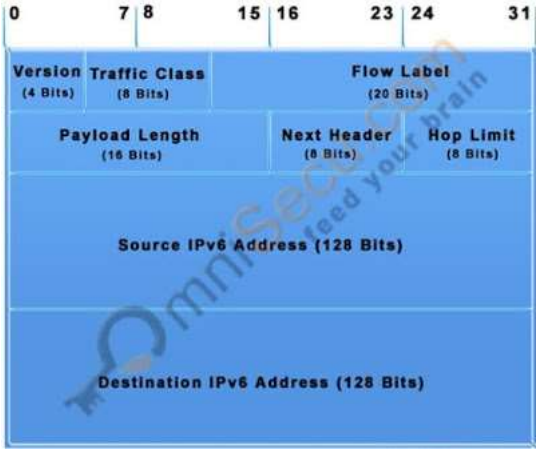
- **Protocol:-**This field defines the protocol used in the data portion of the IP datagram. The Internet Assigned Numbers Authority maintains a list of IP protocol numbers.
- **Header Checksum:-** The 16-bit checksum field is used for error-checking of the header. At each hop, the checksum of the header must be compared to the value of this field. If a header checksum is found to be mismatched, then the packet is discarded. Errors in the data field must be handled by the encapsulated protocol and both UDP and TCP have checksum fields.

As the TTL field is decremented on each hop, a new checksum must be computed each time. *The checksum field is the 16-bit one's complement of the one's complement sum of all 16-bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero.*

- **Source address :-** Sets the source IP address. This option lets you specify a custom IP address to be used as source IP address in sent packets. This allows spoofing the sender of the packets. <*addr*> can be an IPv4 address or a hostname.
- **Destination address :-** An IPv4 address indicating the receiver of the packet. As with the Source address, this may be changed in transit by a network address translation device.

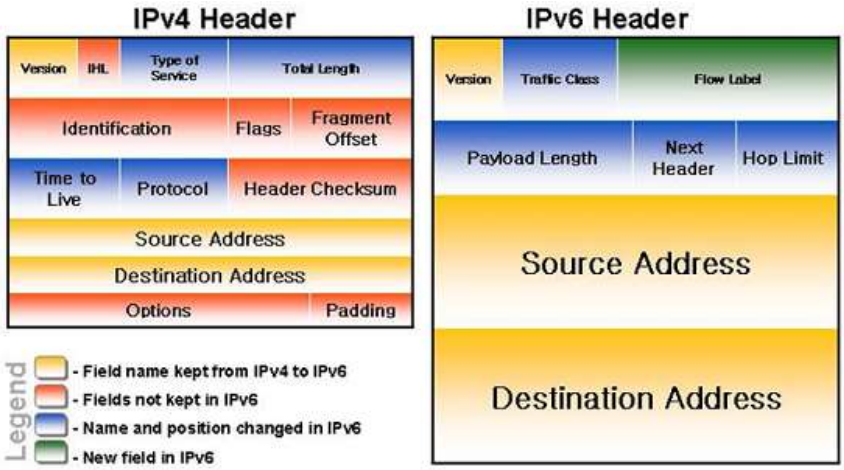
Options:-Additional header fields may follow the destination address field, but these are not often used. The value in the IHL field must include enough extra 32-bit words to hold all the options (plus any padding needed to ensure that the header contains an integral number of 32-bit words). The list of options may be terminated with an EOL (End of Options List) option; this is only necessary if the end of the options would not otherwise coincide with the end of the header.

IPv6 Datagram Packet Header and Fields:



• **Version:** The size of the Version field is 4 bits. The Version field shows the version of IP and is set to 6.
• **Traffic Class:** The size of Traffic Class field is 8 bits. Traffic Class field is similar to the IPv4 Type of Service (ToS) field. The Traffic Class field indicates the IPv6 packet's class or priority.
• **Flow Label:** The size of Flow Label field is 20 bits. The Flow Label field provide additional support for real-time datagram delivery and quality of service features. The purpose of Flow Label field is to indicate that this packet belongs to a specific sequence of packets between a source and destination and can be used to prioritized delivery of packets for services like voice.
• **Payload Length:** The size of the Payload Length field is 16 bits. The Payload Length field shows the length of the IPv6 payload, including the extension headers and the upper layer protocol data
• **Next Header:** The size of the Next Header field is 8 bits. The Next Header field shows either the type of the first extension (if any extension header is available) or the protocol in the upper layer such as TCP, UDP, or ICMPv6.
• **Hop Limit:** The size of the Hop Limit field is 8 bits The Hop Limit field shows the maximum number of routers the IPv6 packet can travel. This Hop Limit field is similar to IPv4 Time to Live (TTL) field.
This field is typically used by distance vector routing protocols, like Routing Information Protocol (RIP) to prevent layer 3 loops (routing loops).
• **Source Address:** The size of the Source Address field is 128 bits. The Source Address field shows the IPv6 address of the source of the packet.
• **Destination Address:** The size of the Destination Address field is 128 bits. The Destination Address field shows the IPv6 address of the destination of the packet.

V4/V6 Packet Format Differences.



IPv4 Packet Header

IP Version Number (4)	IHL (4 Bits)	Type of Service (8 Bits)	Total Length (16 Bits)
Identification (16 Bits)	Flags (4 Bits)	Fragment Offset (12 Bits)	
Time to Live (8 Bits)	Protocol (8 Bits)	Header Checksum (16 Bits)	
Source Address (32 Bits)			
Destination Address (32 Bits)			
Options (variable)	Padding (variable)		

IPv6 Packet Header

IP Version Number (6)	Traffic Class (8 Bits)	Flow Label (20 Bits)
Payload Length (16 bits)	Next Header (8 Bits)	Hop Limit (8 Bits)
Source Address (128 Bits)		
Destination Address (128 Bits)		

IPv6 Packet Structure

<----------Encrypted---------->

IPv6 Header	Hop-by-Hop Extension Header	AH Header	ESP Extension Header	Transport Header (TCP, etc.)	Payload

VRRP Packet Format:

VRRP Packet Format

- The purpose of the VRRP packet is to communicate to all VRRP routers the priority and the state of the Master router associated with the Virtual Router ID.
- VRRP packets are sent encapsulated in IP packets. They are sent to the IPv4 multicast address assigned to VRRP.

Relevant fields in the VRRP header

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Type  | Virtual Rtr ID|   Priority    | Count IP Addrs|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Auth Type   |   Adver Int   |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         IP Address (1)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            .                                  |
|                            .                                  |
|                            .                                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         IP Address (n)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication Data (1)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication Data (2)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This section defines the format of the VRRP packet and the relevant fields in the IP header.

IP Field Descriptions

- Source Address :

  The primary IP address of the interface the packet is being sent from.

- Destination Address :

  The IP multicast address as assigned by the IANA for VRRP is:224.0.0.18

  This is a link local scope multicast address. Routers MUST NOT forward a datagram with this destination address regardless of its TTL.

- TTL :

  The TTL MUST be set to 255. A VRRP router receiving a packet with the TTL not equal to 255 MUST discard the packet.

- Protocol :

  The IP protocol number assigned by the IANA for VRRP is 112 (decimal).

VRRP Field Descriptions

- **Version :** The version field specifies the VRRP protocol version of this packet.
- **Type :** The type field specifies the type of this VRRP packet. The only packet type defined in this version of the protocol is advertisement.
- **VRID:** The Virtual Router Identifier field identifies the virtual router this packet is reporting status for. Configurable item in the range 1-255 (decimal). There is no default
- **Priority:** A value between 0-255.
  - **0**: indicate the current Master has stopped participating in VRRP
  - **255**: for the VRRP router that owns the IP address(es) associated with the virtual router Note that if the IP address owner is available, then it will always become the Master.
  - **1-254**: for the VRRP routers backing up a virtual router
- **Count IP Address:** The number of IP addresses contained in this VRRP advertisement.
- **Auth Type:** The authentication type field identifies the authentication method being utilized. Authentication type is unique on a Virtual Router basis The authentication type field is an 8 bit unsigned integer. A packet with unknown authentication type or that does not match the locally configured authentication method MUST be discarded. Note. The authentication methods currently defined are:

  0 - No Authentication -The use of this authentication type means that VRRP protocol exchanges are not authenticated.

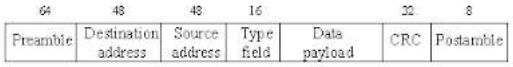
  The contents of the Authentication Data field should be set to zero on transmission and ignored on reception.

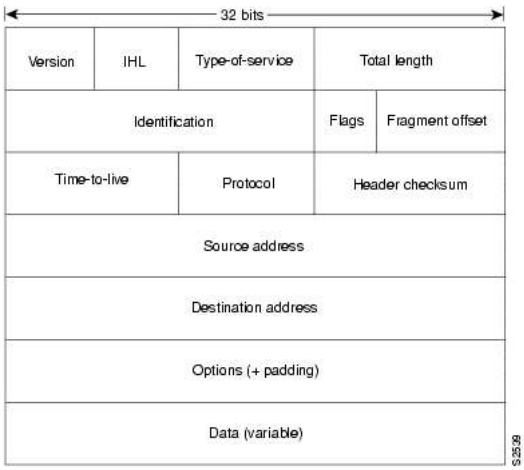
  1 & 2 – Reserved : Both is are maintained to maintain a backward compatibility with older versions.
- **Adver Int:** The Advertisement interval indicates the time interval (in seconds) between ADVERTISEMENTS. The default is 1 second.
- **Checksum:** The checksum field is used to detect data corruption in the VRRP message.
- **IP Address:** One or more IP addresses that are associated with the virtual router. The number of addresses included is specified in the "Count IP Addrs" field.
- **Authentication Data**: The authentication string is currently only used to maintain backwards compatibility with RFC 2338. It SHOULD be set to zero on transmission and ignored on reception

STP:

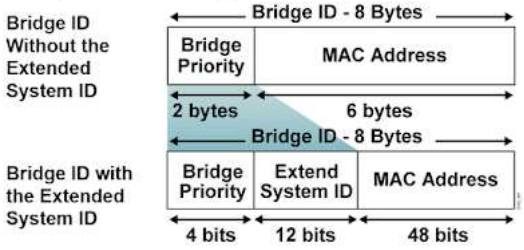
- Broadcasts and Layer 2 loops can be a dangerous combination.
- Ethernet frames have no TTL field
- After an Ethernet frame starts to loop, it will probably continue until someone shuts off one of the switches or breaks a link.

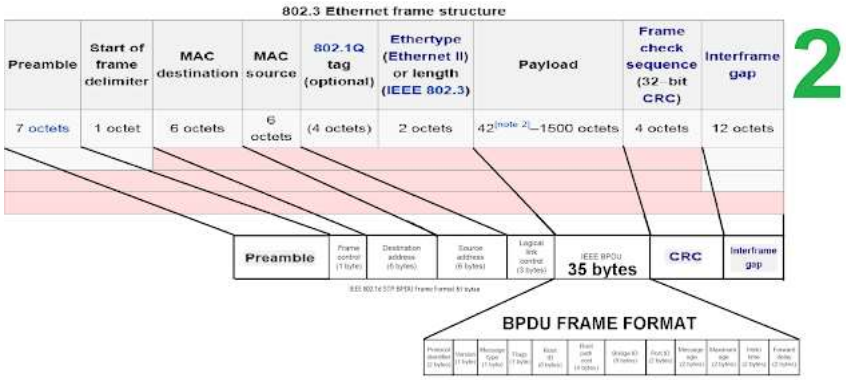
Ethernet Frame Format





- Bridge ID (BID) is used to identify each bridge/switch.
- The BID is used in determining the center of the network, in respect to STP, known as the root bridge.





Bytes	Field
2	Protocol ID
1	Version
1	Message type
1	Flags
8	Root ID
4	Cost of path
8	Bridge ID
2	Port ID
2	Message age
2	Max age
2	Hello time
2	Forward delay

Ethernet:

The data packets from Internet Layer is moved to Network Access Layer as it moves down the TCP/IP protocol stack. There is a size limitation for Ethernet Frame. The total size of the Ethernet frame must be between 64 bytes and 1,518 bytes (not including the preamble). Network Access Layer Breaks Internet Layer data (IP Datagram) into smaller chunks, if necessary, which will become the payload of ethernet frames. A Frame includes data to be transmitted and also a header and a trailer which contain information that the network adapters on the ethernet need to process the frame.
The total size of the ethernet frame must be between 64 bytes and 1,518 bytes (not including the preamble). A frame shorter than the minimum 64 bytes but with a valid CRC is called as a runt. In most cases, such frames arise from a collision. Any frame which is received and which is greater than the maximum frame size, is called a "giant". A "giant" is longer than 1518 bytes yet have a valid CRC. Both runts and giants are considered as invalid.

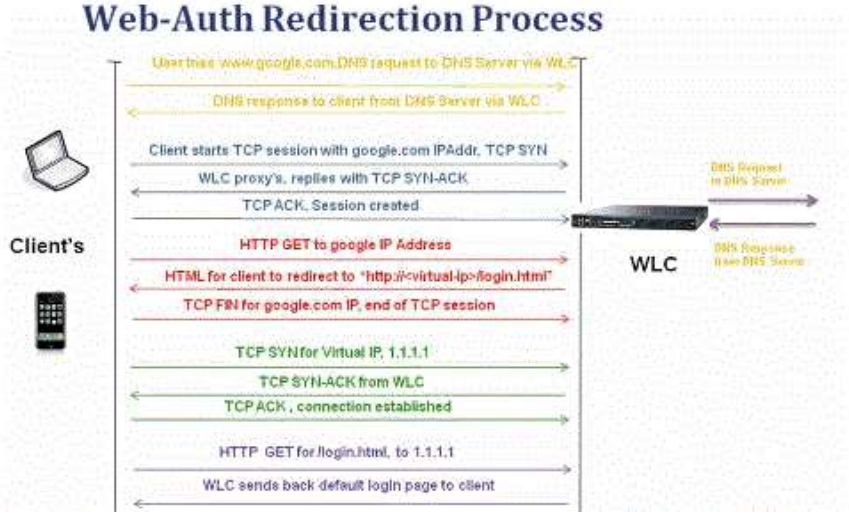
Structure of an Ethernet Frame

The Ethernet Frame fields are explained below.
Preamble: A sequence of 56 bits having alternating 1 and 0 values that are used for synchronization. They serve to give components in the network time to detect the presence of a signal, and being reading the signal before the frame data arrives.
SFD (Start Frame Delimiter): A sequence of 8 bits having the bit configuration 10101011 that indicates the start of the frame.
Source and Destination MAC Addresses: The Source MAC Address is the MAC Address of the device this frame is coming from. The Destination MAC Address is the MAC Address of the device which is going to receive this frame. Both of these fields are 6 bytes long.
Length/Type: A 2-byte (16-bit) field contains the number of bytes in the Data field or the nature of the MAC client protocol.
Data: This field contains the actual data transferred from the source device to the destination device. The maximum size of this field is 1500 bytes. If the size of this field is less than 46 bytes, then use of the subsequent "Pad" field is necessary to bring the frame size up to the minimum length.
Pad: If necessary, extra data bytes are appended in this field to bring the frame length up to its minimum size. A minimum Ethernet frame size is 64 bytes from the Destination MAC Address field through the Frame Check Sequence.

Frame Check Sequence: This field contains a 4-byte Cyclic Redundancy Check (CRC) value used for error checking. When a source device assembles a frame, it performs a Cyclic Redundancy Check (CRC) calculation on all fields in the frame except the Preamble, SFD (Start Frame Delimiter), and frame check sequence using a predetermined algorithm. The source device stores the value in this field and transmits it as part of the frame. When the frame is received by the destination device, it performs an CRC test again using the same algorithm. If the CRC value calculated at the destination device does not match the value in the FCS (Frame Check Sequence) field, the destination device will discards the frame, considering this as a transmission error.

WebAuth:

http://www.cisco.com/c/en/us/support/docs/wireless/5500-series-wireless-controllers/108501-webauth-tshoot.html



RADIUS - Operations

Before the Client starts communicating with the Radius Server, it is required that the secret key is shared between the Client and the Server and the Client must be configured to use Radius server to get service.

Once Client is configured properly then:

- The Client starts with Access-Request.
- The Server sends either Access-Accept, Access-Reject, or Access-Challenge.
- Access-Accept keeps all the required attributes to provide service to the user.

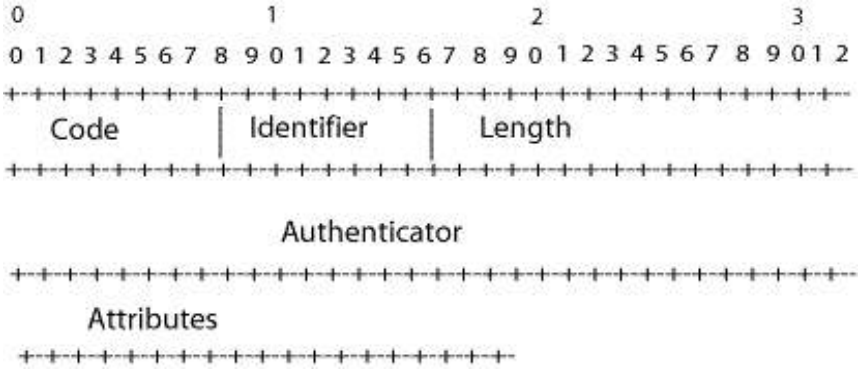
Radius Codes (decimal) are assigned as follows:

- 1 Access-Request
- 2 Access-Accept
- 3 Access-Reject
- 4 Accounting-Request
- 5 Accounting-Response
- 11 Access-Challenge
- 12 Status-Server (experimental)
- 13 Status-Client (experimental)
- 255 Reserved
- No Keep Alive concept - Good or Bad??

Codes 4 and 5 are related to Radius Accounting Functionality. Codes 12 and 13 are reserved for possible use.

RADIUS - Packet Format

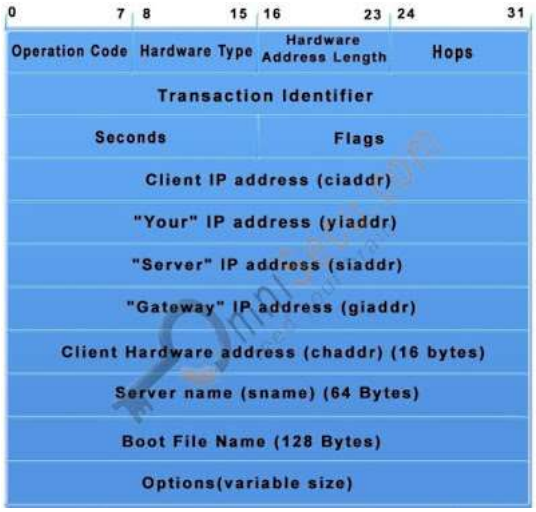
The packet format of Radius is as shown below:



Code: This is 1 Octet (1 byte) long and identifies various types of packets. Normally 1 Octet means 1 Byte.
Identifier: This is again 1 Octet long and aids in matching responses with requests.
Length: This is 2 Octets long and specifies the length of the packet including code, identifier, length, and authenticator. (Min packet is 20 Octets and max is 4096 Octets).
Authenticator: This is 16 Octets long and filled up in case of some requests and responses.

DHCP Packet:

All Dynamic Host Configuration Protocol (DHCP) messages include a FIXED format section and a VARIABLE format section. The fixed format section consists of several fields that are the same in every Dynamic Host Configuration Protocol (DHCP) message. The variable format section in the Dynamic Host Configuration Protocol (DHCP) contains "OPTIONS", which carry additional configuration parameters.



DHCP Message Field	Description
Operation Code	Specifies the type of the Dynamic Host Configuration Protocol (DHCP) message. Set to 1 in messages sent by a client (requests) and 2 in messages sent by a server (response).
Hardware Type	Specifies the network LAN architecture. For example, the ethernet type is specified when htype is set to 1.
Hardware Address Length	Layer 2 (Data-link layer) address length (MAC address) (in bytes); defines the length of hardware address in the chaddr field. For Ethernet (Most widely used LAN Standard), this value is 6.
Hops	Number of relay agents that have forwarded this message.
Transaction Identifier	Used by clients to match responses from servers with previously transmitted requests.
seconds	Elapsed time (in seconds) since the client began theDynamic Host Configuration Protocol (DHCP) process.
Flags	Flags field is called the broadcast bit, can be set to 1 to indicate that messages to the client must be broadcast
ciaddr	Client's IP address; set by the client when the client has confirmed that its IP address is valid.
yiaddr	Client's IP address; set by the server to inform the client of the client's IP address.
siaddr	IP address of the next server for the client to use in the configuration process (for example, the server to contact for TFTP download of an operating system kernel).
giaddr	Relay agent (gateway) IP address; filled in by the relay agent with the address of the interface through which Dynamic Host Configuration Protocol (DHCP) message was received.
chaddr	Client's hardware address (Layer 2 address).
sname	Name of the next server for client to use in the configuration process.
file	Name of the file for the client to request from the next server (for example the name of the file that contains the operating system for this client).

BGP Packet Formats:

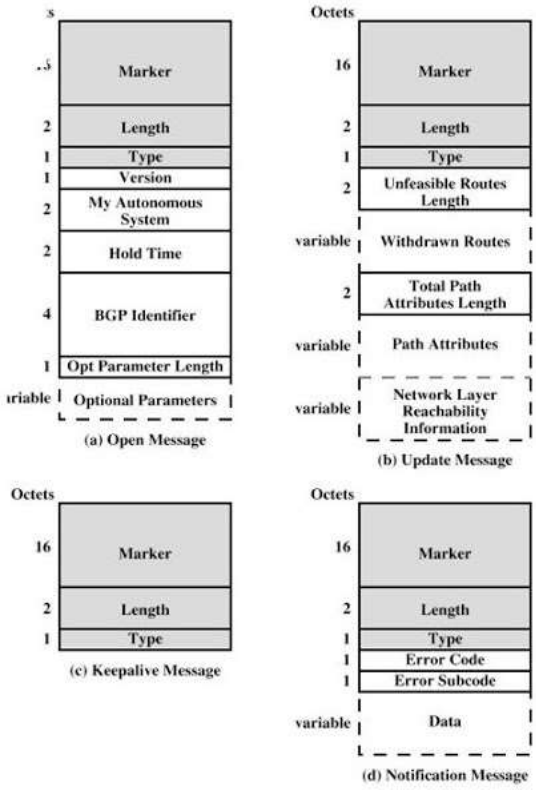
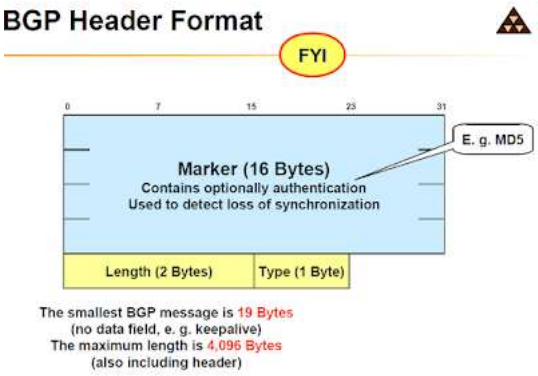
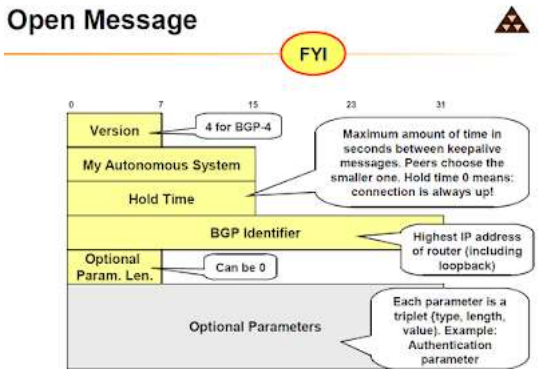
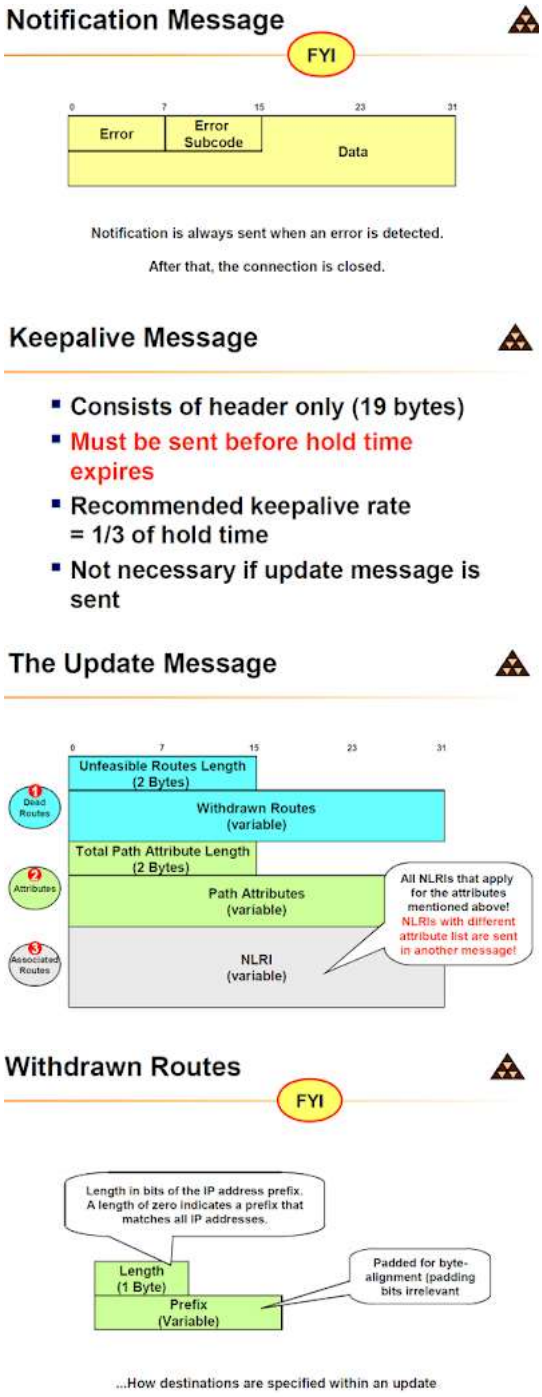


Figure 14.12 BGP Message Formats







Posted by Manish Bidsar at 23:19

Labels: Packet Formats to Remember

3 comments:

Vijay Kumar 27 April 2018 at 23:22

Nice Concept !!

Reply

Unknown 30 July 2018 at 20:00

All concept are very clearly define. All types of header are available on single place.
Thanks a lot...

Reply

Valentin Dugan 30 October 2018 at 12:05

Amazing, very informative.

Reply

Enter your comment...

Comment as: helloroxor@gm ▾ Sign out

Publish Preview Notify me

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)