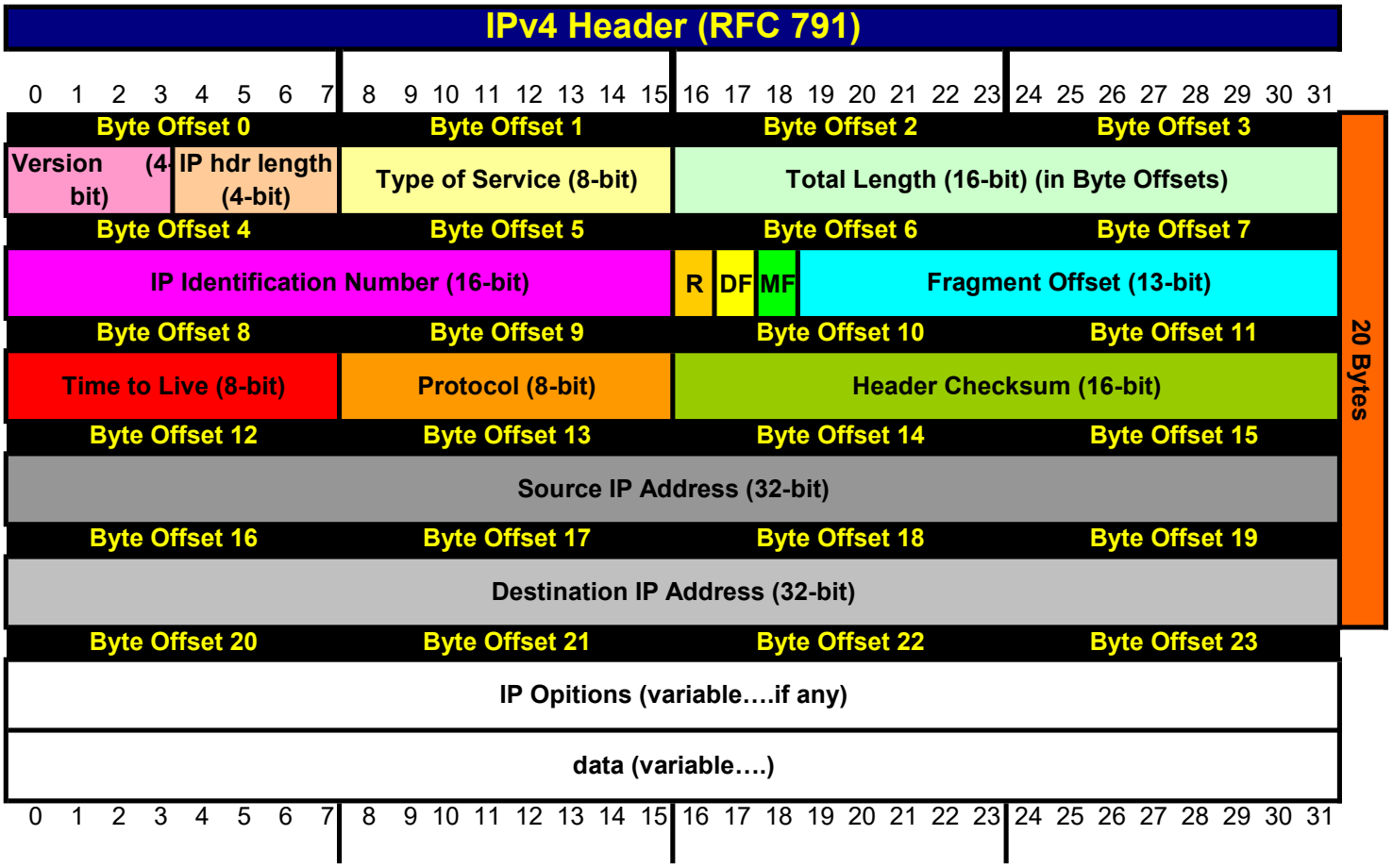
Header Offset Shortcuts

Field	Length (bits)	TCPDUMP Filter	Notes
IP Header Length	4	ip[0] &0x0F	Remember to use a 4 byte multiplier to find header length in bytes
IP Packet Length	16	ip[2:2]	The is no multiple for this length field
IP TTL	8	ip[8]	
IP Protocol	8	ip[9]	

	D	Hex	Proto	D	Hex	Proto	D	Hex	Proto
	1	0x01	ICMP	9	0x09	IGRP	47	0x2F	GRE
	2	0x02	IGMP	17	0x11	UDP	50	0x32	ESP
	6	0x06	TCP	47	0x2F	GRE	51	0x33	AH

IP Address - Src	32	ip[12:4]	
IP Address - Dst	32	ip[16:4]	
IP Fragmentation	flag=3	ip[6] &0x20 = 0x20 More Fragment bit is set.	
	offset=13	ip[6:2] &0x1fff != 0x000 fragment offset in not 0	
ICMP Type	8	icmp[0]	
ICMP Code	8	icmp[1]	
TCP Src Port	16	tcp[0:2]	
TCP Dst Port	16	tcp[2:2]	
TCP Header Length	4	tcp[12] &0x0F	Remember to use a 4 byte multiplier to find header length in bytes
TCP Flags	8	tcp[13]	
TCP Windows Size	16	tcp[14:2]	
UDP Src Port	16	udp[0:2]	
UDP Dst Port	16	udp[2:2]	
UDP Header Length	16	upd[4:2]	The is no multiple for this length field



IP Version Number

Valid values are: 4 for IP version 4 6 for IP version 6

IP Header Length (4 byte multiplier)

Number of 32-bit words in IP header minimum value 5 (5 x 4 = 20 bytes) maximum value 15 (15 x 4 = 60 bytes)

Type of Service (Used by gateways as a QoS type field) (Most OS's default to 0) (Over)

Total Length (No multiplier)

Number of bytes in packet maximum length = 65,535

IP Identification Number

Uniquely identifies every datagram sent by host, value typically incremented by 1 (AKA Fragment ID)

Flags

R is reserved and must be set to 0

D is Don't Fragment Flag 1=Don't Fragment 0=Can Fragment

MF is More Fragments 1=More Fragments 0=No Fragment or no more Fragments

(frag x:y@z where x is the fragment ID, y is # of bytes (must be divisible by 8) and z is the fragment offset)

(In Ethernet the MTU 1500 should see middle fragments of size 1480 (1480 data + 20 ip header = 1500)

Fragment Offset (8 byte multiplier) (Max fragment offest 65528)

Position of this fragment in the original datagram value is multipled by 8 to get bytes

Time To Live

IP Protocol

D	Hex		D	Hex		D	Hex		D	Hex	
1	0x01	ICMP	9	0x09	IGRP	47	0x2F	GRE	88	0x58	EIGRP
2	0x02	IGMP	17	0x11	UDP	50	0x32	ESP	89	0x59	OSPF
6	0x06	TCP	47	0x2F	GRE	51	0x33	AH			

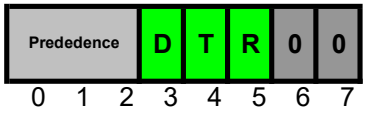
Header Checksum

Covers IP header only Validated along the path from source to destination

Options (0-40 bytes; 1st @ 20th byte offset; padded 4-byte boundary) (Processed by each router as packet passes)

D	Hex		D	Hex	
0	0x00	End of Option list	68	0x44	Timestamp
1	0x01	No operation (pad)	131	0x83	Loose source route (security risk)
7	0x07	Record Route (security risk)	137	0x89	Strict source route (security risk)

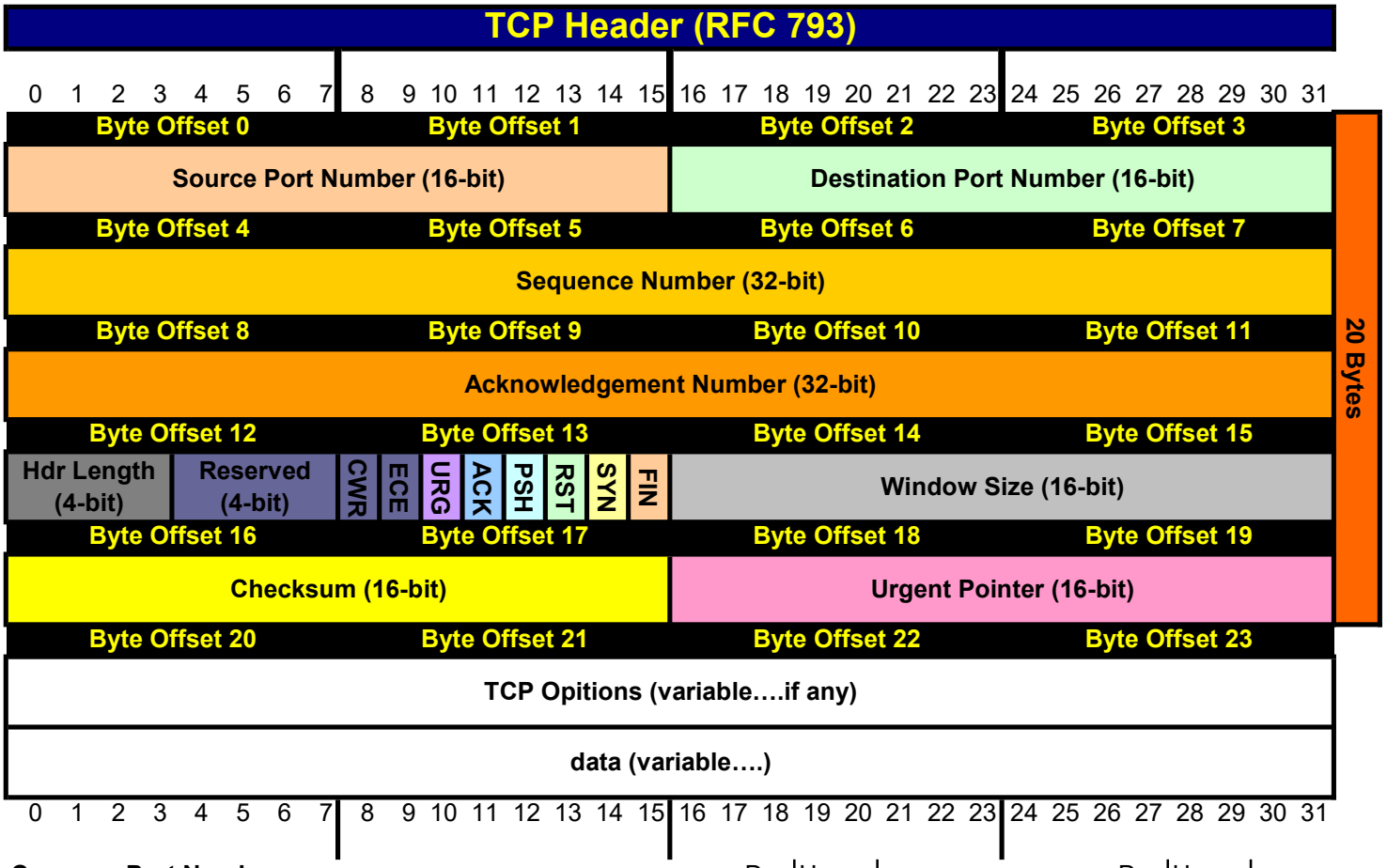
Type of Service (Used by gateways as a QoS type field) (Most OS's default to 0)



Bit	0	-	2	Precedence		
Bit	3			0 = Normal Delay	1 =	Low Delay
Bit	4			0 = Normal Throughput	1 =	High Throughput
Bit	5			0 = Normal Relibility	1 =	High Relibility
Bit	6	&	7	Reserved for future use (Always set to 0)		

Precedence

1	1	1	Network Control
1	1	0	Internetwork Control
1	0	1	CRITIC / ECP
1	0	0	Flash Override
0	1	1	Flash Override
0	1	0	Immediate
0	0	1	Priority
0	0	0	Routine



Common Port Numbers

D	Hex		D	Hex		D	Hex		D	Hex	
7	0x07	echo	22	0x16	ssh	80	0x50	http	143	0x8F	imap
19	0x13	chargen	25	0x19	smtp	110	0x6E	pop3	179	0xB3	bgp
20	0x14	ftp-data	53	0x35	domain	119	0x77	nntp	389	0x185	ldap
21	0x15	ftp-control	79	0x4F	finger	137	0x89	netbios-ns	443	0x1BB	https (ssl)
						139	0x8B	netbios-ssn	445	0x1BD	ms-ds

Sequence Number
32-bit number uniquely identifies initial byte of segment data.

Acknowledgement Number
Represents next byte of data receiving host expects: (last received sequence number + 1)

Header Length (4 byte multiplier)
Number of 32-bit words in TCP header minimum value 5 (5x4=20bytes) maximum value 15 (5x15=60bytes)

Reserved 4 bits set to 0

Congestion Window Reduced (CWR)
Set to 0 unless ECN is used. (1 = sender has cut congestion window in half)

Explicit Congestion Notification Echo (ECE)
Set to 0 unless ECN is used. (1 = receiver cuts congestion window in half)

Flags
URG = Urgent ACK = Acknowledgment PSH = Push RST = Reset SYN = Syncronize
FIN = Finish

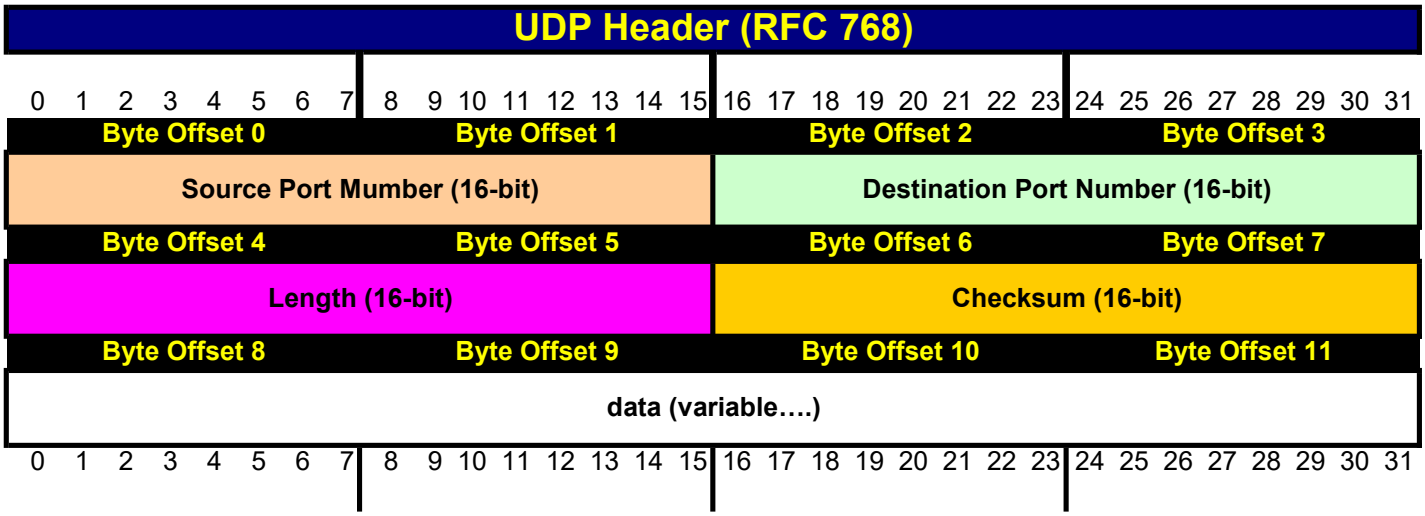
Window Size
Acts as flow control. Window size dynamically changes as data is received. A 0 window size tells src host to wait.

Checksum
Covers psedoheader and entire TCP segment

Urgent Pointer
Points to the sequence number of the byte following urgent data

Options

0 End of Options List	2 Maximum segment size	4 Selective ACK ok
1 No Operation (pad)	3 Window scale	8 Timestamp



Common Port Numbers

D	Hex		D	Hex		D	Hex	
7	0x07	echo	69	0x45	tftp	514	0x202	syslog
19	0x13	chargen	137	0x89	netbios-ns	520	0x208	rip
37	0x25	time	138	0x8A	netbios-dgm	33434	829A	traceroute
53	0x35	domain	161	0xA1	snmp			
67	0x43	bootps	162	0xA2	snmp-trap			
68	0x44	bootpc	500	0x1F4	isakmp			

Length

Number of bytes in the entire datagram including header

minimum value 8

(Which is the length of just the header with no data)

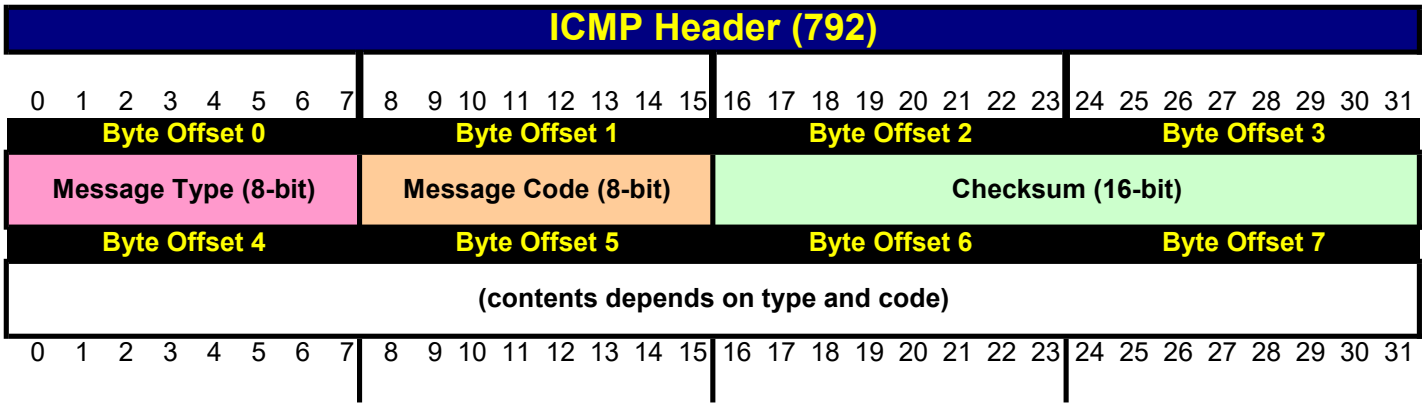
maximum value 65515 (or 65507 bytes of UDP data)

(Max IP is 65535 bytes - 20 byte header = 65515 bytes for UDP packet - 8 bytes UDP header = 65507)

Checksum

Covers psedoheader and entire UDP datagram

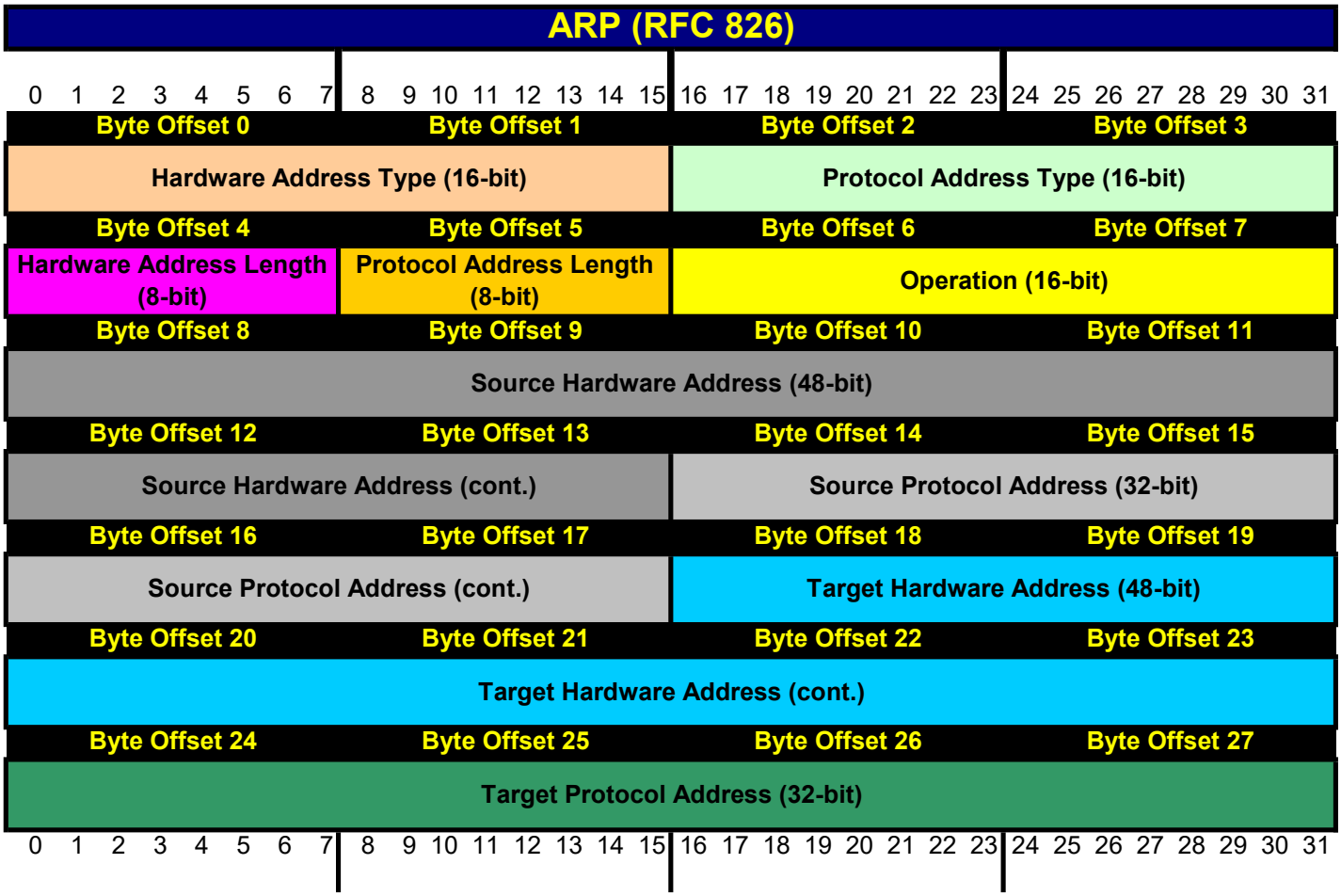
(Note: By RFC, the crc is not required)



Common Types & Codes

- 0 Echo reply
- 3 Destination Unreachable
  - 0 Net Unreachable
  - 1 Host Unreacheable
  - 2 Protocol Unreachable
  - 3 Port Unreachable
  - 4 Fragmentation Needed & Don't Fragment Flag Set
  - 5 Source Route Failed
  - 6 Destionation Network Unknown
  - 7 Destination Host Unknown
  - 8 Source Route Isolated
  - 9 Network Administratively Prohibited
  - 10 Host Administratively Prohibited
  - 11 Network Unreachable for TOS
  - 12 Host Unreachable for TOS
  - 13 Communication Administratively Prohibited
- 4 Source Quench
- 5 Redirect
  - 0
  - 1
  - 2
  - 3
- 8 Echo
- 9 Router Advertisement
- 10 Router Selection
- 11 Time Exceeded
  - 0 Time to Live exceeded in transit
  - 1 Fragment Reassembly Time Exceeded
- 12 Parameter Problem
  - 0 Pointer indicates the error
  - 1 Missing a Required Option
  - 2 Bad Length
- 13 Timestamp Request
- 14 Timestamp Reply
- 15 Information Request
- 16 Information Reply
- 17 Address Mask Request
- 18 Address Mask Reply
- 30 Traceroute

(Note: Byte offset 4-5: identification #)
(Note: Byte offset 6-7: sequence #)



ARP maps the logical address (IP) to the physical address (MAC)

Hardware Address Type

1	Ethernet
6	IEEE 802 Lan

Protocol Address Type

2048	IPv4 (0x0800)

Hardware Address Length

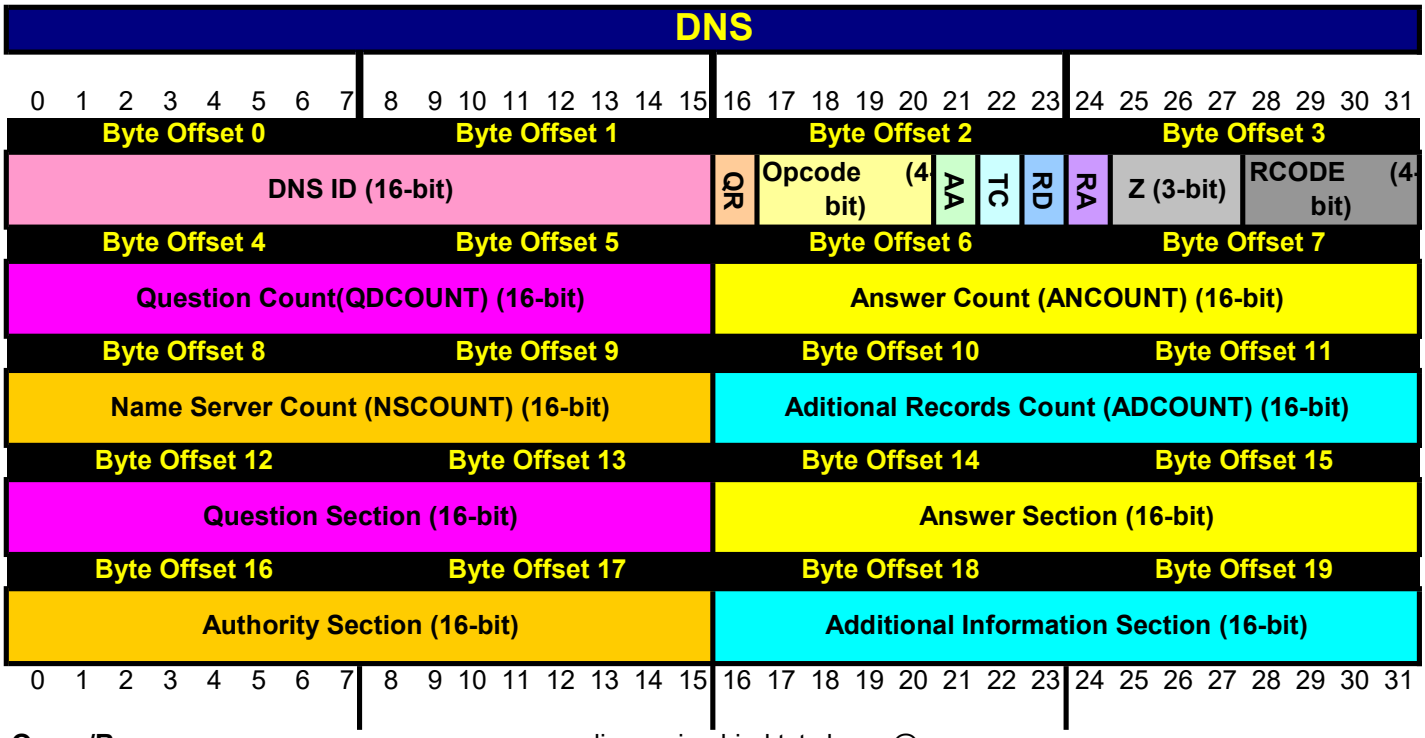
6	for Ethernet/IEEE 802

Protocol Address Length

4	for IPv4

Operation

1	Request
2	Reply



Query/Response

0 Query
1 Response

dig version.bind txt chaos @ *server name*
dig @ *server name* txt chaos version.bind

Opcode

0 Standard query (QUERY)
1 Inverse query (IQUERY)
2 Server status request (STATUS)

AA

1 Athoritative Answer

TC

1 Truncation

RD

1 Recursion Desired

RA

1 Recursion Available

Z

Reserved; set to 0

Response Code

0 No Error
1 Format Error
2 Server Failure
3 Non-existant Domain (NXDOMAIN)
4 Query Type Not Implemented
5 Query Refused

QDCOUNT

(Number of entries in Question section)

ANCOUNT

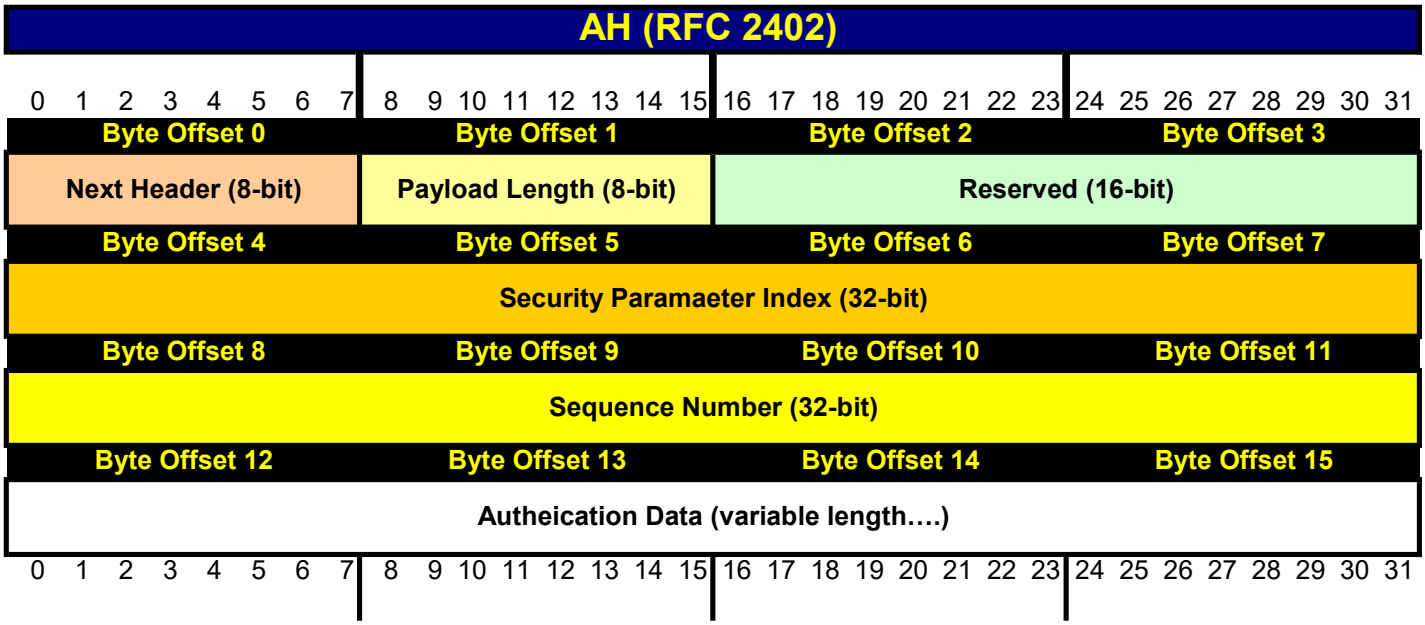
(Number of resource records in Answer section)

NSCOUNT

(Number of name server resource records in Authority section)

ARCOUNT

(Number of resource records in Additional Information section)



Next Header

Equivalent to the IP Protocol Identifier field in IPv4

D	Hex		D	Hex		D	Hex		D	Hex	
1	0x01	ICMP	9	0x09	IGRP	47	0x2F	GRE	88	0x58	EIGRP
2	0x02	IGMP	17	0x11	UDP	50	0x32	ESP	89	0x59	OSPF
6	0x06	TCP	47	0x2F	GRE	51	0x33	AH			

Payload Length

Specifies the length of the Authentication Header (number of 32-bit words - 2 for IPv6 compatibility)

Reserved

Zero filled field

Security Parameter Index (SPI)

Random 32-bit value used with dst IP address and IP Sec protocol to uniquely identify the SA.

The SPI is generally selected by the dst IP Sec node.

Sequence Number

A 32-bit sequence number starting at zero and incremented by one for each packet.

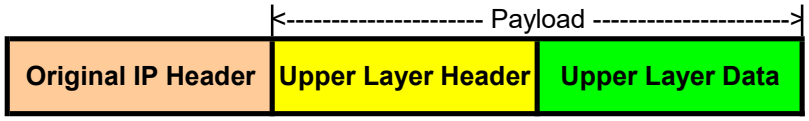
This monotonically increasesing sequence number is the AH anti-replay mechanism.

Authentication Data

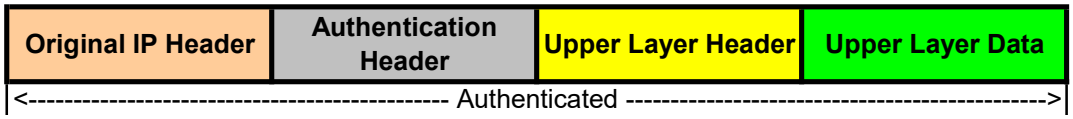
A variable-length field that contains the Integrity Check Value (ICV) for the packet.

The length of the IVC must be an integral multiple of 32 bits; will ne padded or truncated to meet the requirement.

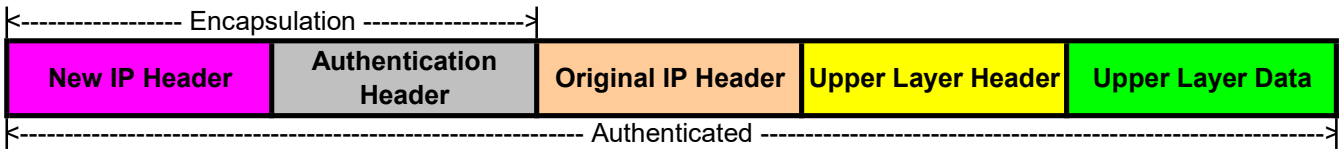
Original Packet

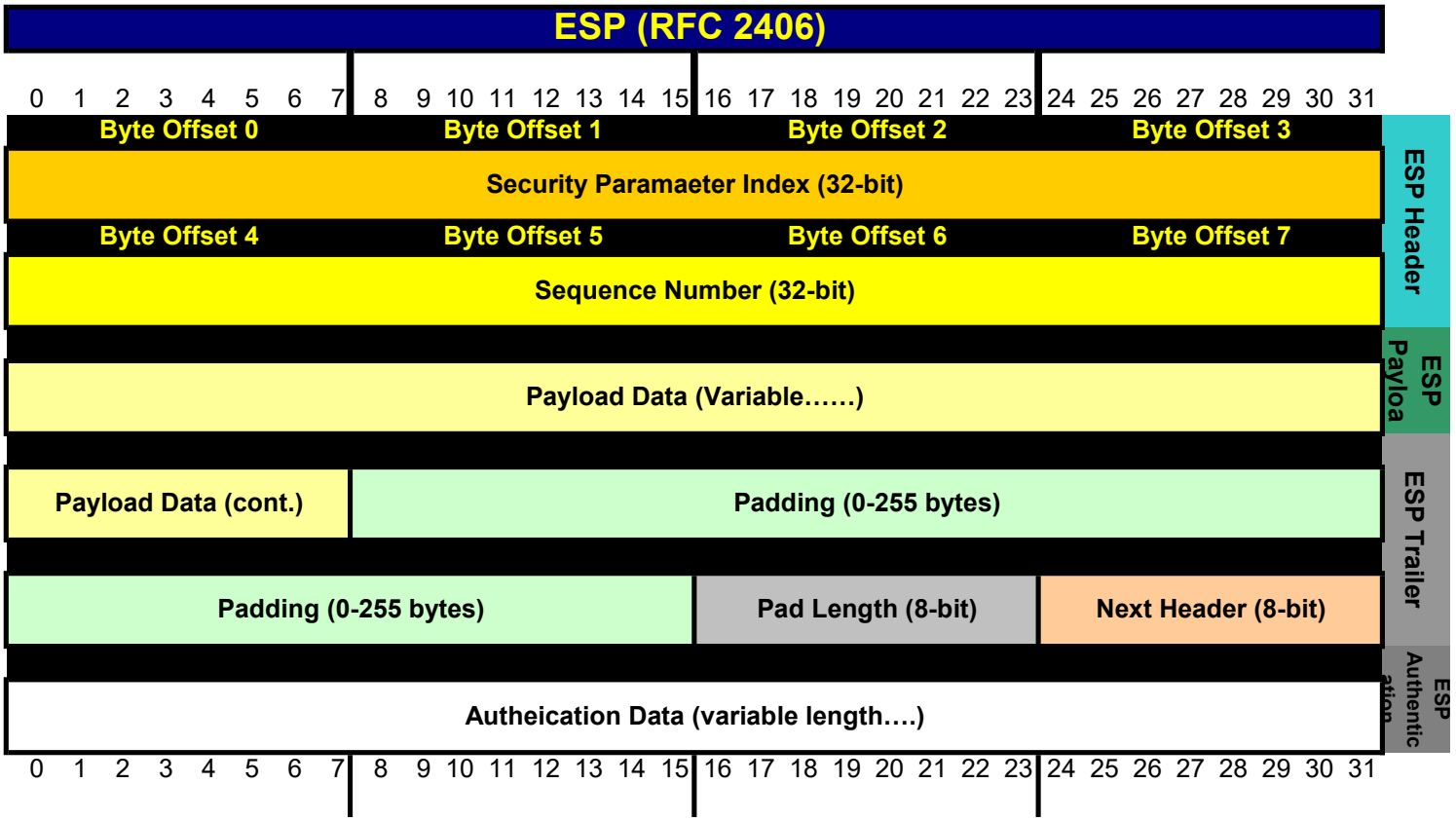


AH Tranport Mode Packet



AH Tunnel Mode Packet





ESP Header

Security Parameter Index (SPI)

Random 32-bit value used with dst IP address and IP Sec protocol to uniquely identify the SA.
The SPI is generally selected by the dst IP Sec node.

Sequence Number

A 32-bit sequence number starting at zero and incremented by one for each packet.
This monotonically increasesing sequence number is the AH anti-replay mechanism.

ESP Payload

Payload Data

A variable-length field containing the data to be protected by the ESP protocol; i.e., the original IP packet

ESP Trailer

Padding

A 0-255 byte field used for varity of purposes. It is primarily used to ensure that the Payload, Pad Length, & Next Header align on a 32-bit boundary. It can also be used if the ESP encryption algorithm requires a certain minimum number of bytes. Finally, it may be used to hide the real size of the payload (protect againts traffic flow analysis)

Pad Length

8-bit value indicating the number of Pad bytes that were inserted.

Next Header

Equivalent to the IP Protocol Identifier field in IPv4

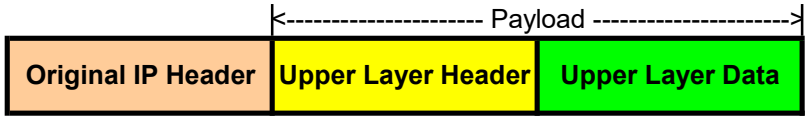
D	Hex		D	Hex		D	Hex		D	Hex	
1	0x01	ICMP	9	0x09	IGRP	47	0x2F	GRE	88	0x58	EIGRP
2	0x02	IGMP	17	0x11	UDP	50	0x32	ESP	89	0x59	OSPF
6	0x06	TCP	47	0x2F	GRE	51	0x33	AH			

ESP Authentication

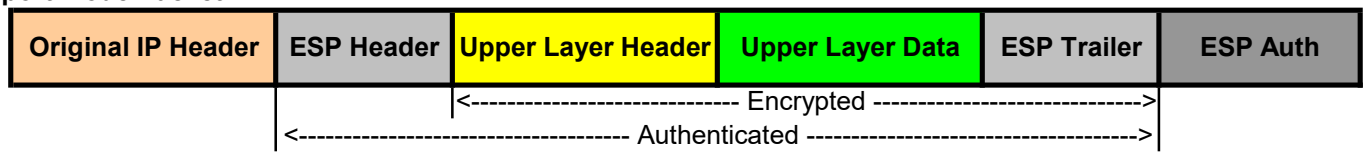
Authentication Data

A variable-length field that contains the Integrity Check Value (ICV) for ESP the packet. The length of the this field is dependent upon the authentication function used. This field is peresent only if an authenication service is being employed in the SA.

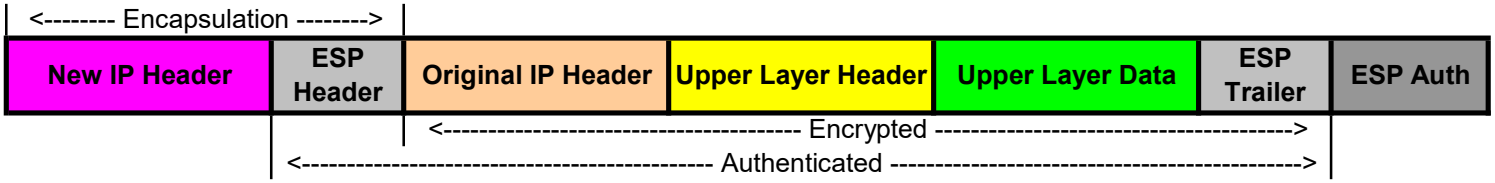
Original Packet

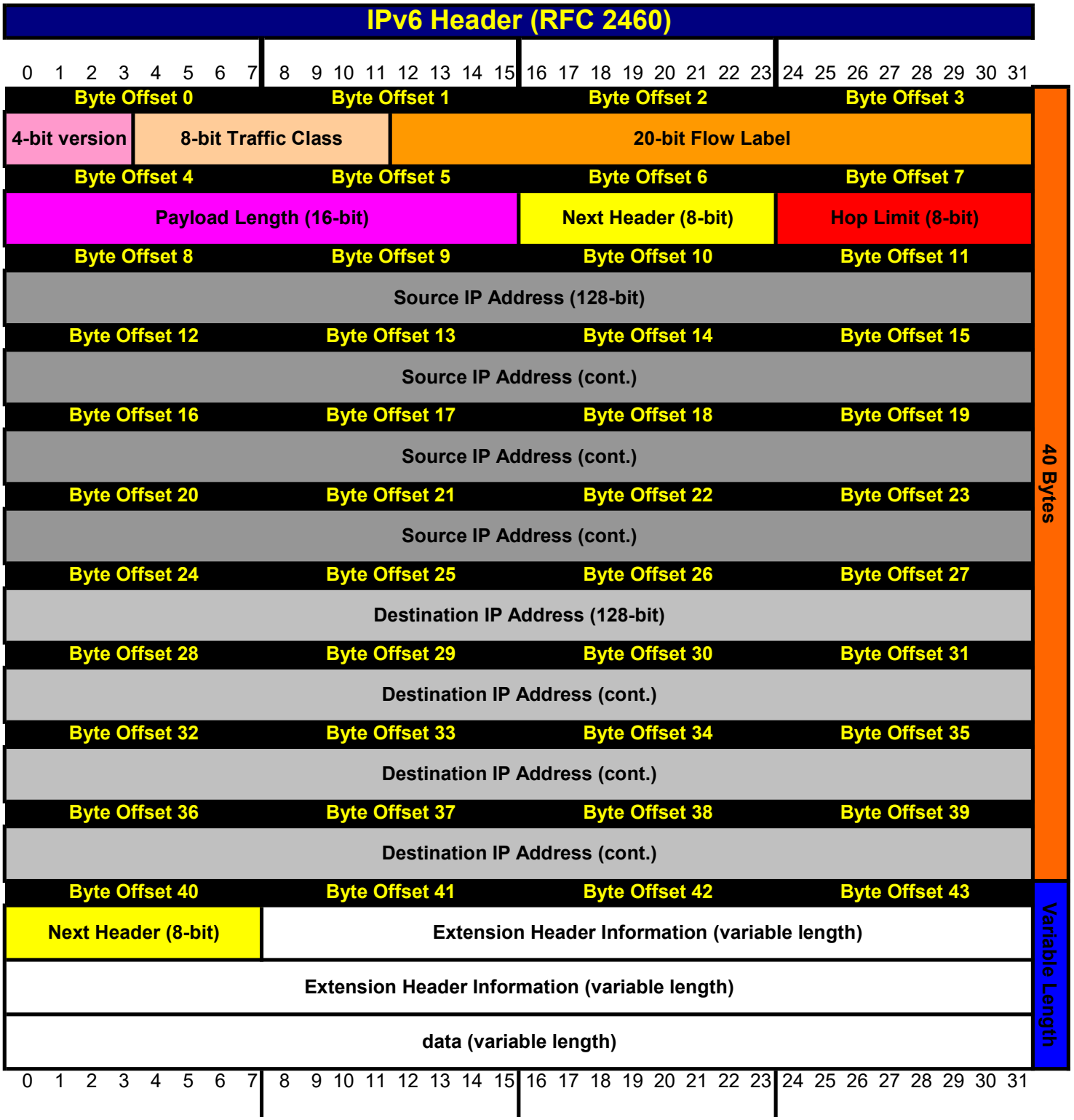


ESP Tranport Mode Packet



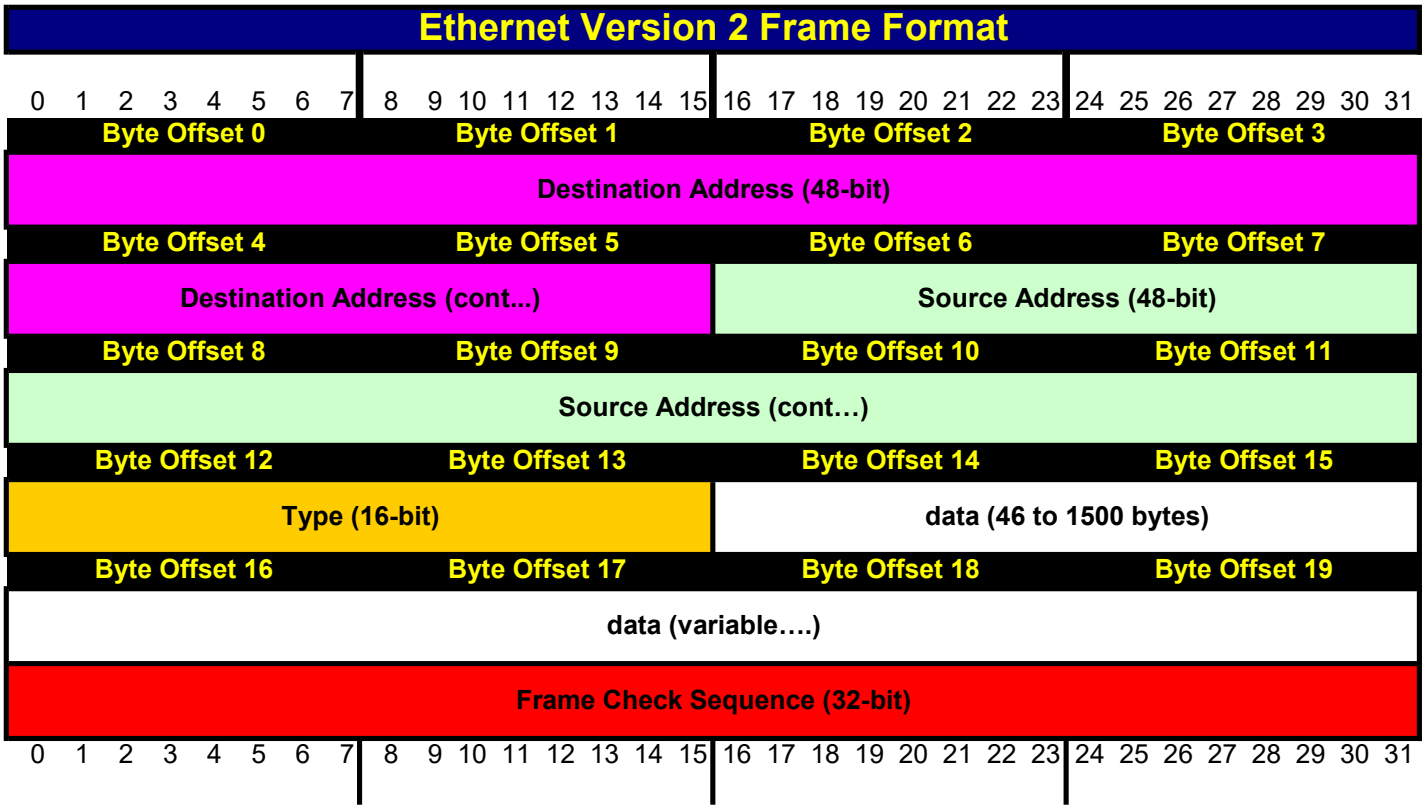
ESP Tunnel Mode Packet





IP Version Number	6 for IP version 6 4 for IP version 4
Traffic Class	8-bit field similar to IPv4 type of service field
Flow Label	To tag packets of a specific flow to differentiate the packets at the network layer. **QoS**
Payload Length	The total length of the data portion of the packet
Next Header	Similar to the protocol field of IPv4 packet header
Hop Limit:	Similar to Time to Live field in IPv4 packet header
Source Address	128-bit source address field
Destination Address	128-bit destination address field

IPv6 Hdr



Preamble: 8 bytes (64 bite) At the head of each frame is a preamble used for sychronization 1010…10101011

Destinnation Address: 6 byte (48 bit) desination media access control (MAC) address

Soutce Address: 6 byte (48 bit) source media access control (MAC) address

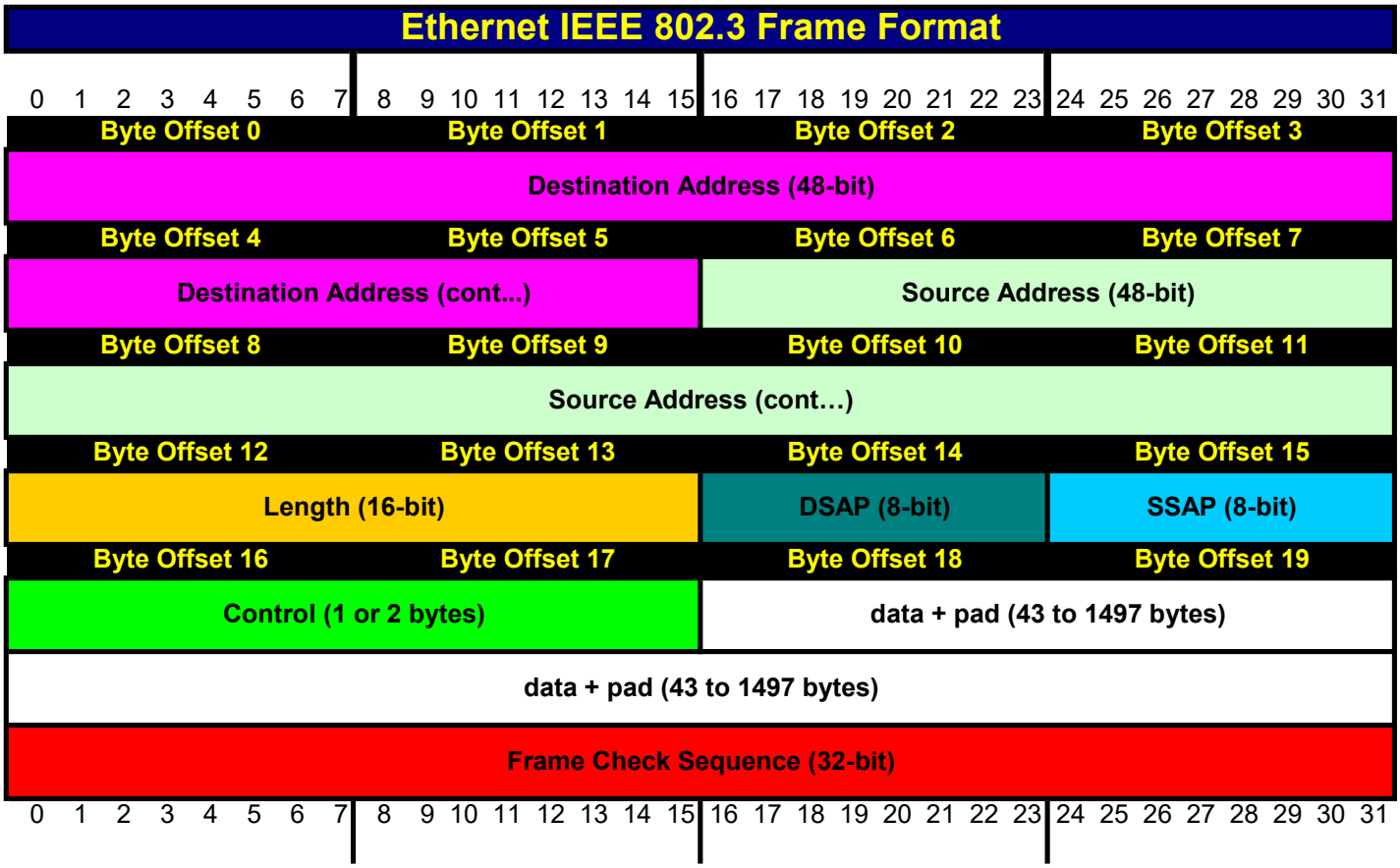
Type: 2 byte (16 bit) field that specifies the upper-layer protocol

Type	Value
NetWare	8137
XNS	0600, 0807
IP	800
IP (VINES)	0BAD, 80C4
ARP	806
RARP	8035
DRP	6003
LAT	6004
LAVC	6007
ARP (Atalk)	80F3

Data: 46 to 1500 bytes of upper-layer protocol information

Frame Check Sequence: The cyclic redundancy check (CRC) or checksum for the Ethernet Frame

Ether v2 Hdr



Preamble: 8 bytes (64 bite) At the head of each frame is a preamble used for sychronization 1010…10101011

Destinnation Address: 6 byte (48 bit) desination media access control (MAC) address

Soutce Address: 6 byte (48 bit) source media access control (MAC) address

Length: 2 byte (16 bit) field that specifies the number of bytes (3-1500) in the LLC and data fields

Logical Link control The logical link control (LLC) is made up of the DSAP, SSAP and Control fields. This is a mothed for telling the 802.3 IEEE and Netware (RAW) formats. The IEEE 802.3 format has the LLS and the NetWare 802.3 "Raw" for mate does not.

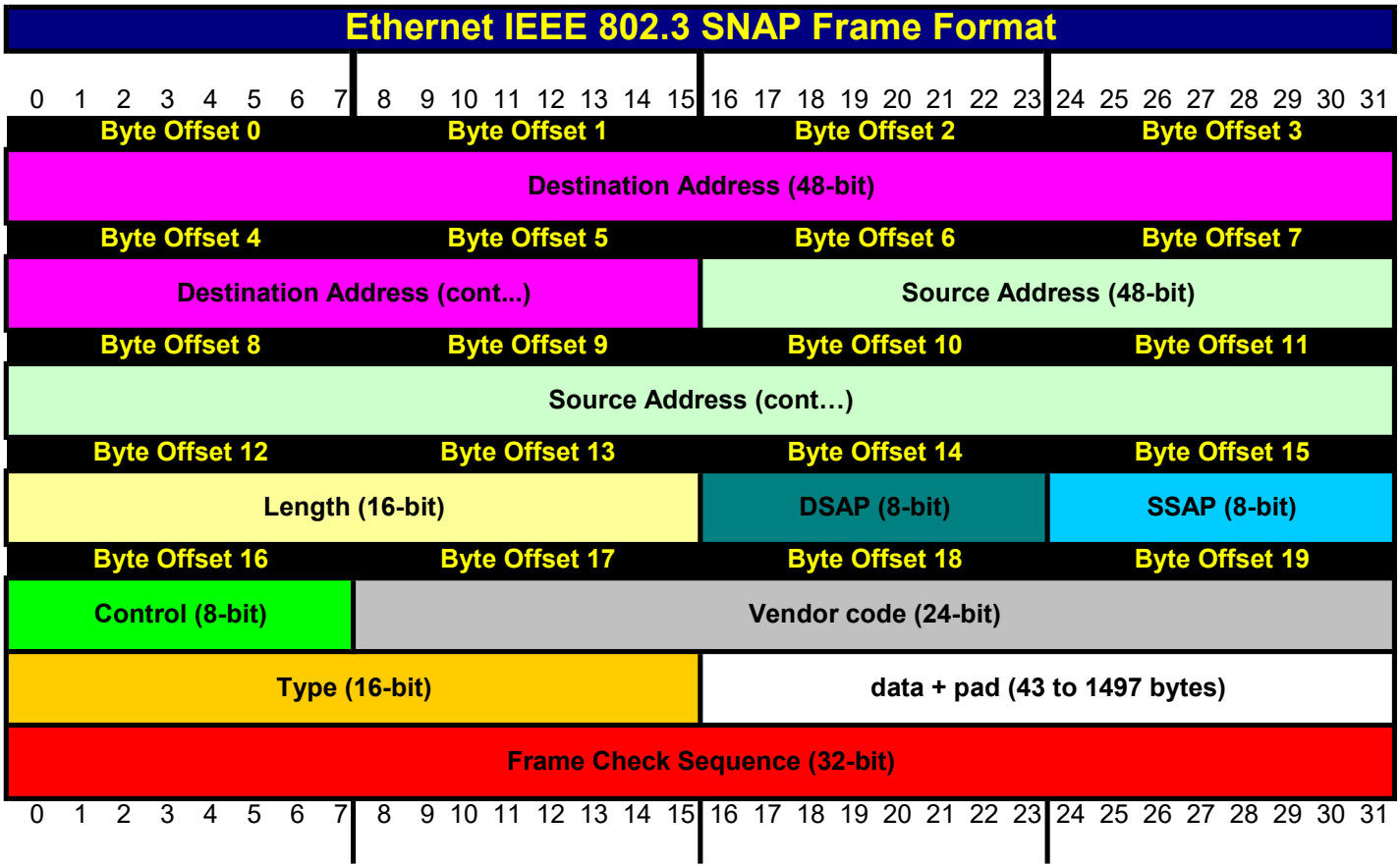
- **DSAP:** 1 byte destination service access point; receiving process at destination
- **SSAP:** 1 byte source service access point; sending process at source
- **Control:** 1 byte is various control information (Connection less)
  2 bytes are for connection-oriented LLC

Pad: Pads the frame to minimum of 46 bytes of data and LLC (so collisions can be detected)

Data: 46 to 1500 bytes of upper-layer protocol information

Frame Check Sequence: The cyclic redundancy check (CRC) or checksum for the Ethernet Frame

Ether 802.3 Hdr



Preamble: 8 bytes (64 bite) At the head of each frame is a preamble used for sychronization 1010…10101011

Destinnation Address: 6 byte (48 bit) desination media access control (MAC) address

Soutce Address: 6 byte (48 bit) source media access control (MAC) address

Length: 2 byte (16 bit) field that specifies the number of bytes (3-1500) in the LLC and data fields

Logical Link control The logical link control (LLC) is made up of the DSAP, SSAP and Control fields. This is a mothed for telling the 802.3 IEEE and Netware (RAW) formats. The IEEE 802.3 format has the LLS and the NetWare 802.3 "Raw" for mate does not.

- **DSAP:** 1 byte destination service access point; receiving process at destination (**Always AA**)
- **SSAP:** 1 byte source service access point; sending process at source (**Always AA**)
- **Control:** 1 byte is various control information (Connection less)
  2 bytes are for connection-oriented LLC

SNAP Header The Subnet Access Protocal Header consists of the Vendor Code and Type fields

- **Vendor Code:** 3 byte (24 bit) field to identify the vendor
- **Type:** 2 byte (16 bit) field that specifies the upper-layer protocol

Type	Value
NetWare	8137
XNS	0600, 0807
IP	800
IP (VINES)	0BAD, 80C4
ARP	806

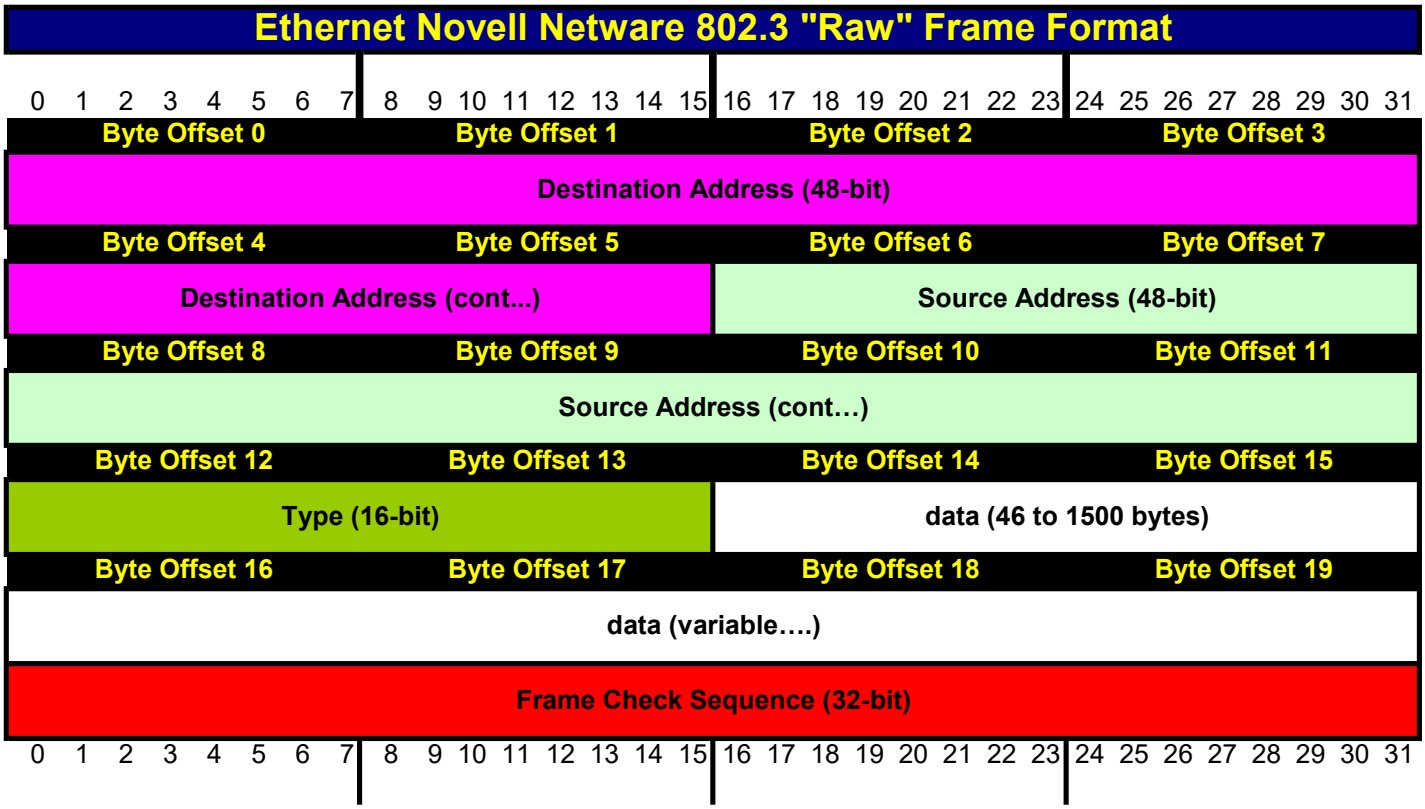
Type	Value
RARP	8035
DRP	6003
LAT	6004
LAVC	6007
ARP (Atalk)	80F3

Pad: Pads the frame to minimum of 46 bytes of data and LLC (so collisions can be detected)

Data: 46 to 1500 bytes of upper-layer protocol information

Frame Check Sequence: The cyclic redundancy check (CRC) or checksum for the Ethernet Frame

Ether 802.3 SNAP Hdr



IP Version Number

Preamble: 8 bytes (64 bite) At the head of each frame is a preamble used for sychronization 1010…10101011

Destinnation Address: 6 byte (48 bit) desination media access control (MAC) address

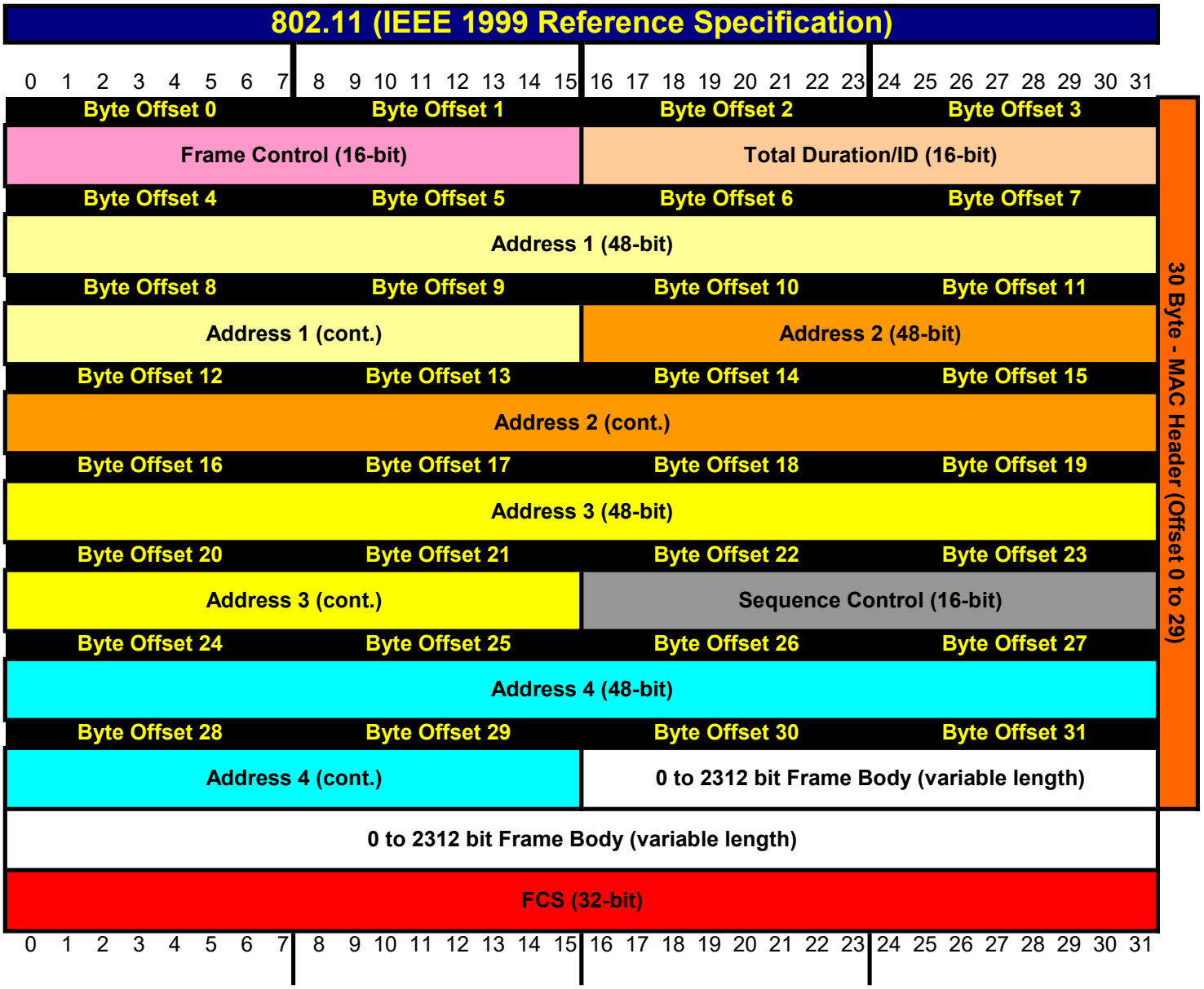
Soutce Address: 6 byte (48 bit) source media access control (MAC) address

Length: 2 byte (16 bit) field that specifies the number of bytes (46-1500) in the LLC and data fields Note the lack of the LLC fields, this is who you tell Netware 802.3 from IEEE 802.3

Data: 46 to 1500 bytes of upper-layer protocol information. IPX header starting with 2 byte checksum (usually FFF) followed by NetWare higher layers ('data')

Frame Check Sequence: The cyclic redundancy check (CRC) or checksum for the Ethernet Frame

Ether 802.3 RAW Hdr



Frame Control Consists of the following subfields: Protocol Version (bits 0-1), Type (bits 2-3), Subtyoe (bits 4-7), To DS (bit 8), From DS (bit 9), More Fragment (bit 10), Retry (bit 11), Power management (bit 12), More Data (bit 13), WEP (bit 14) and Order (bit 15)

Duration / ID

Duration/ID field encoding

15	14	bit 13 - 0	Usage
0	0 - 32767		Duration
1	0	0	Fixed value within frames transmitted during the CFP
1	0	1-16383	Reserved
1	1	0	Reserved
1	1	1-2007	AID in PS-Poll frames
1	1	2008 - 16383	Reserved

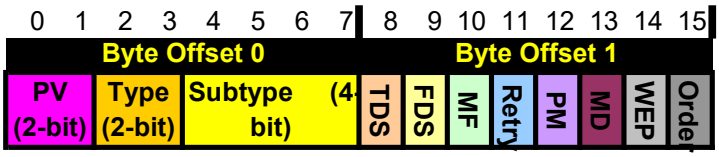
Address Fields There are 4 address fields in the MAC frame format. These fields are used to indicate the BSSID, source address (SA), destination address (DA), transmitting station address (TA), and the receiving station address (RA).

Sequence Control Consists of the following subfields: Fragment Number (bits 0-3) and Sequence Number (bits 4-15).

Frame Body Variable length field that contains information specific to individual frame types and subtypes.

FCS 32-bit check sum field calculated over all the fields of the MAC header and Frame body

Frame Control



Protocol Version Currently the value should always be 0

Type / Subtype The type and subtype field together identify the function of the frame

Type		Type	Subtype				
b3	b2	Description	b7	b6	b5	b4	Subtype Description
0	0	Management	0	0	0	0	Association Request
0	0	Management	0	0	0	1	Association Response
0	0	Management	0	0	1	0	Reassociation Request
0	0	Management	0	0	1	1	Reassociation Response
0	0	Management	0	1	0	0	Probe Request
0	0	Management	0	1	0	1	Probe Response
0	0	Management	0110-0111				Reserved
0	0	Management	1	0	0	0	Beacon
0	0	Management	1	0	0	1	Announcement traffic indication message (ATIM)
0	0	Management	1	0	1	0	Disassociation
0	0	Management	1	0	1	1	Authentication
0	0	Management	1	1	0	0	Deauthentication
0	0	Management	1101-1111				Reserved
0	1	Control	0000-1001				Reserved
0	1	Control	1	0	1	0	Power Save (PS)-Poll
0	1	Control	1	0	1	1	Request To Send (RTS)
0	1	Control	1	1	0	0	Clear To Send (CTS)
0	1	Control	1	1	0	1	Acknowledgment (ACK)
0	1	Control	1	1	1	0	Contention-Free (CF)-End
0	1	Control	1	1	1	1	CF-End + CF-Ack
1	0	Data	0	0	0	0	Data
1	0	Data	0	0	0	1	Data + CF-Ack
1	0	Data	0	0	1	0	Data + CF-Poll
1	0	Data	0	0	1	1	Data + CF-Ack + CF-Poll
1	0	Data	0	1	0	0	Null function (no data)
1	0	Data	0	1	0	1	CF-Ack (no data)
1	0	Data	0	1	1	0	CF-Poll (no data)
1	0	Data	0	1	1	1	CF-Ack + CF-Poll (no data)
1	0	Data	1000-1111				Reserved
1	1	Reserved	0000-1111				Reserved

To DS (a) Set to 1 in data type frames destined for the DS. This includes all data type frames sent by STAs associated with an AP. The To DS field is set to 0 in all other frames.

From DS (b) Set to 1 in data type frames exiting the DS. It is set to 0 in all other frames.

TO/From DS Values

a	b	Meaning
0	0	A data frame direct from one STA to another STA within the same IBSS, as well as all management and control type frames.
1	0	Data frame destined for the DS
0	1	Data frame exiting the DS
1	1	Wireless distribution system (WDS) frame being distributed from one AP to another AP

More Fragments Set to 1 in all data management type frames that have another fragment of the current MSDU or current MMPDU to follow. It is set to 0 in all other frames.

Retry Set to 1 in any data or management type frame that is a retransmission of an earlier frame. It is set to 0 in all other frames. Areceiving station uses this indication to aid in the process of eliminating duplicate frames.

Power Management

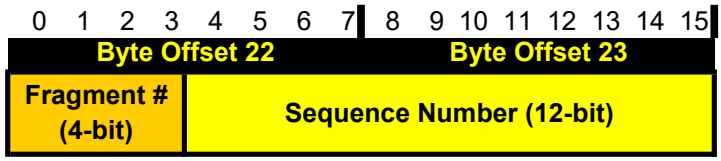
Set to 1 indicates that the STA will be in power-save mode. A value of 0 indicates that the STA will bein active mode. This field is always set to 0 in frames transmitted by an AP.

More Data Set to 1 in directed data type frames transmitted by a contention-free (CF)-Pollable STA to the point coordinator (PC) in response to a CF-Poll to indicate that the STA has at least one additional buffered MSDU available for transmission in response to a subsequent CF-Poll. Set to 0 in all other directed frames.

WEP Set to 1 if the Frame Body field contains information that has been processed by the WEP algorithm. The WEP field is set to 0 in all other frames. When the WEP bit is set to 1, the Frame Body field is expanded.

Order Set to 1 if any data type frame that contains an MSDU, or fragment thereof, which is being trasferred using the StrictlyOrdered service class. Set to 0 in all other frames.

Sequence Control



TCPDUMP / WINDUMP

windump -i <interface> -nx	capture from interface (-i <interface>) do not convert names(-n) and print on hex and ascii (-x)
windump -i <interface> -nx -s0	capture from interface (-i <interface>) do not convert names(-n), print on hex and ascii (-x) and capture all the packet
windump -r <file> -nxp	capture from file (-r <file>), do not convert names (-n), print out hex and ascii (-x), not in permiscous mode (-p)

Keywords

host (host)	**ip**	**vrrp**	**ether multicast**
src host (host)	**ip6**	**ip broadcast**	**vlan** (vlan_id)
dst host (host)	**arp**	**ip proto** (protocol)	**atalk**
gateway (host)	**icmp**	**ip protochan** (protocol)	**decnet**
net (net/len)	**icmp6**	**ip6 proto** (protocol)	**decnet src**
src net (net)	**tcp**	**ip6 protochain** (protocol)	**decnet host**
dst net (net)	**udp**	**ip multicast**	**iso**
port (port)	**ah**	**ip6 multicast**	**stp**
src port (port)	**esp**	**ether host** (MAC)	**ipx**
dst port (port)	**igmp**	**ether src** (MAC)	**netbeui**
less (length)	**igrp**	**ether dst** (MAC)	
greater (length)	**rarp**	**ether proto** (protocol)	

Bit Masking	tcpflags	icmptype			
And unwanted bits with 0	tcp-fin		icmp-echoreply	icmp-echo	icmp-paramprob
And wanted bits with 1	tcp-syn		icmp-unreachable	icmp-ireq	icmp-tstamp
0 AND 0 = 0	tcp-rst		icmp-sourcequench		icmp-tstampreply
0 AND 1 = 0	tcp-push		icmp-redirect		icmp-ireq
1 AND 0 = 0	tcp-ack		icmp-routeradvert		icmp-ireqreply
1 AND 1 = 1	tcp-urg		icmp-routersolicit		icmp-maskreq
			icmp-timxceed		icmp-maskreply

Expressions: >, <, >=, <=, =, !=, +, -, *, /, &, | **!** or **not** **&&** or **and** **||** or **or**

filter format <protocol header>[offset:length]<relation><value>

tcpdump [command line options] ['filter']

windump [command line options] ["filter"]

Examples

host A and B	Connections between host and and host B				
ip[9] = 1	icmp	ip[9] = 6	tcp	ip[9] = 17	udp
tcp[2:2] < 20	The TCP dst port is greater than 20			udp[6:2] != 0	Non-zero UDP checksum
tcp[tcpflags]=tcp-syn	Only Syn	tcp[13] &0x02 != 0	At minimum the SYN bit set		
tcp[tcpflags]=tcp-ack	Only Ack	tcp[13] &0x10 != 0	At minimum the ACK bit set		
tcp[tcpflags]=tcp-fin	or	tcp[13] &0xff = 0x1	Only the FIN bit is set		tcp[13] &0xff = 1
tcp[13] &0xff =16	or	tcp[13] &0xff = 0x10	Only the ACK bit is set		

icmp[0]=3 and icmp[1]=2	icmp type 3 is destination unreachable category and a code of 2 specifies that this is an ICMP protocol unreachable (**Good filter for detecting protocol scans**)
(tcp and (tcp[13] &0x0f != 0) and not port 25 and not port 20)	A tcp packet where any combination of PSH, RST, SYN, FIN are set and the packet is not port 25 or 20
udp[21:4]=0x56455253	Looks for “VERS” in udp payload for VERSION.BIND
tcp[20:4] = 0x5353482d	Looks for “SSH-” in TCP payload
ip[6:2] & 0x3fff != 0	Look for ALL fragmented ip packets
ip[6] &0x20 = 0x20 or ip[6:2] &0x1fff != 0	Look for more fragment bit set **or** fragment offset greater than 0 (**Look for ALL fragmented ip packets**)
ip[6] &0x20 = 0 and ip[6:2] &0x1fff != 0	Look for more fragment bit **not** set **and** fragment offset greater than 0 (**Last fragment packets**)

Command Line Options	
Options	**Description**
-a	Attempt to convert network and broadcast addresses to names
-A	
-B <size>	Set driver's buffer size to size in KiloBytes. The default buffer size is 1 megabyte (i.e 1000).
-c <count>	Exit after receiving <count> of packets
-C <file size>	Before writing a raw packet to a savefile, check whether the file is currently larger than file_size and, if so, close the current savefile and open a new one.
-d	Dump the compiled packet-matching code in a human readable form to standard output and stop
-dd	Dump packet-matching code as a C program fragment
-ddd	ddd Dump packet-matching code as decimal numbers (preceded with a count)
-D	Print the list of the interface cards available on the system. WINDUMP ONLY
-e	Print the link-level header on each dump line
-E <algo:secret>	Use algo:secret for decrypting IPsec ESP packets where algorithms may be des-cbc, 3des-cbc, blowfish-cbc, rc3-cbc, cast128-cbc, or none.
-f	Print 'foreign' internet addresses numerically rather than symbolically
-F <file>	Use file as input for the filter expression
-i <interface>	Listen on interface (defaults to lowest numbered interface)
-l	Make stdout line buffered. ``tcpdump -l | tee dat'' or ``tcpdump -l > dat & tail -f dat''
-L	
-m <module>	Load SMI MIB module definitions from file module
-n	Don't convert addresses to names
-N	Don't print domain name qualification of host names
-O	Do not run the packet-matching code optimizer
-p	Don't put the interface into promiscuous mode
-q	Quick output – print less protocol information
-r <file>	Read packets from file (created with the –w option)
-R	Assume ESP/AH packets to be based on old specs
-s <snaplen>	Snarf snaplen bytes of data from each packet (default is 68)
1518 Max Ethernet Frame (14 byte Ethernet header + 1500 byte IP + 4 byte Ethernet trailer) 64 Min Ethernet Frame (14 byte Ethernet header + 64 byte IP + 4 byte Ethernet trailer) **Note: -s0 mean full ethernet packet**	
-S	Print absolute, rather than relative TCP sequence numbers
-t	Don't print a timestamp on each dump line
-T <type>	Force packets selected by "expressions" to be interpreted the specified type (cnfp, rpc, rtp, snmp, wb)
-tt	Print an unformatted timestamp on each dump line
-ttt	Print a delta (in micro-seconds) between current and previous line on each dump line
-tttt	Print a timestamp in default format proceeded by date on each dump line
-u	Print undecoded NFS handles
-U	
-v	Verbose output (TOS, TTL, IP ID, Fragment Offset, IP Flags, length)
V	
-w <file>	Write the raw packet to file rather than parsing and printing to stdout
-x	Print each packet (minus link level header) in hex
-X	Print each packet in hex and ascii
-y <datalinktype>	

http://www.tcpdump.org/tcpdump_man.html
http://windump.polito.it/docs/manual.htm#Wdump

NGREP

ngrep
<-hXViwqpevxlDtT> <-IO pcap_dump> <-n num> <-d dev> <-A num> <-s snaplen> <-S limitlen>
<match expression>
<bpf filter>

Command Line Options	
-A (num)	is dump num packets after a match
-D	is replay pcap_dumps with their recorded time intervals
-d (device)	is use a device different from the default (pcap)
-e	is show empty packets
-h	is help/usage
-i	is ignore case
-I (file)	is read packet stream from pcap format file pcap_dump (**Capitol i**)
-l	is make stdout line buffered
-n (num)	is look at only num packets
-O (file)	is dump matched packets in pcap format to pcap_dump
-p	is don't go into promiscuous mode
-q	is be quiet
-S (limitlen)	is set the limitlen on matched packets
-s (snaplen)	is set the bpf caplen
-t	is print timestamp every time a packet is matched
-T	is print delta timestamp every time a packet is matched
-V	is version information
-v	is invert match
-w	is word-regex (expression must match as a word)
-X	is interpret match expression as hexadecimal
-x	is print in alternate hexdump format

<match expression>	is either an extended regular expression or a hexadecimal string. see the man page for more information.
<bpf filter>	is any bpf filter statement.

Examples:

ngrep '' icmp	print all UDP packets
ngrep '' tcp	print all TCP packets
ngrep '' udp	print all UDP packets
ngrep '' port 53	print all packets to or from TCP or TDP port 53
ngrep '' tcp port 53	print all packets to or from only TCP port 53
ngrep - v '' tcp port 53	print all packets but those to or from TCP port 53
ngrep 'USER|PASS' tcp port 21	print all packets to or from TCP port 21 where USER or PASS
ngrep 'SSH-' port tcp 22	print all packets to or from TCP port 22 where SSH-
ngrep 'LILWORD' port 138	print Microsoft browsing traffic for NT domain LILWORLD
ngrep -iq 'rcpt to|mail from' tcp port 25	monitor current delivery and print sender and recipients
ngrep 'user' port 110	monitor POP3
ngrep -q 'abcd' icmp	"pinging" host running a Microsoft operating system?
ngep -i -I <input file> "Yahoo"	read from input file and search for case insensitive "Yahoo"

OS	Version	Platform	TTL	Window	DF	TOS	TCP Options
DC-Osx	1.1-95	Pyramid/NILE	30	8192	n	0	
Windows	9x/NT	Intel	32	5000-9000	y	0	
NetApp	OnTap	5.1.2-5.2.2	54	8760	y	0	
HPJetDirect	?	HP_Printer	59	2100-2150	n	0	
AIX	4.3.X	IBM/RS6000	60	16000-16100	y	0	MSS
AIX	4.2.X	IBM/RS6000	60	16000-16100	n	0	
Cisco	11.2	7507	60		y	0	
DigitalUnix	4	Alpha	60		y	16	
IRIX	6.x	SGI	60		y	16	
OS390	2.6	IBM/S390	60		n	0	
Reliant	5.43	Pyramid/RM1000	60		n	0	
FreeBSD	3.x	Intel	64		y	16	
JetDirect	G.07.x	J311A	64		n	0	
Linux	2.2.x	Intel	64	32120	y	0	MSS, SackOK, wscale, Timestamp, one NOP
Linux	2.4	Intel	64	5840			MSS, SackOK, wscale, Timestamp, one NOP
OpenBSD	2.x	Intel	64		n	16	MSS, Timestamp, wscale, sacks OK, 5 nops
0s/400	r4.4	AS/400	64		y	0	
SCO	R5	Compaq	64		n	0	
Solaris	8	Intel/Sparc	64		y	0	
FTX(Unix)	3.3	STRATUS	64	32678	n	0	
Unisys	x	Mainframe	64	32768	n	0	
Netware	4.11	Intel	126	32000-32768	y	0	
Windows	9x/NT	Intel	128	5000-9000	y	0	
Windows	2000	Intel	128	17000-18000	y	0	MSS, SackOK, 2 NOPs
Cisco	12	2514	255	3800-5000	n	192	
Solaris	2.x	Intel/Sparc	255	8760	y	0	

ADDITIONAL NOTES
#
Cisco IOS 12.0 normally starts all IP sessions with IP ID of 0
Solaris 8 uses a smaller TTL (64) then Solaris 7 and below (255).
Windows 2000 uses a much larger Window Size then NT.

Decimal to hexadecimal to ASCII Chart

Dec	Hex	ASCII
0	0	NUL
1	1	SOH
2	2	STX
3	3	ETX
4	4	EOT
5	5	ENQ
6	6	ACK
7	7	BEL
8	8	BS
9	9	HT
10	A	LF
11	B	VT
12	C	FF
13	D	CR
14	E	SO
15	F	SI
16	10	DLE
17	11	DC1
18	12	DC2
19	13	DC3
20	14	DC4
21	15	NAK
22	16	SYN
23	17	ETB
24	18	CAN
25	19	EM
26	1A	SUB
27	1B	ESC
28	1C	FS
29	1D	GS
30	1E	RS
31	1F	US

Dec	Hex	ASCII
32	20	SP
33	21	!
34	22	"
35	23	#
36	24	$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?

Dec	Hex	ASCII
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_

Dec	Hex	ASCII
96	60	'
97	61	a
98	62	b
99	63	c
100	64	DEL
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	DEL

Dec	Hex	ASCII
128	80	Ç
129	81	ü
130	82	é
131	83	â
132	84	ä
133	85	à
134	86	å
135	87	ç
136	88	ê
137	89	ë
138	8A	è
139	8B	ï
140	8C	î
141	8D	ì
142	8E	Ä
143	8F	Å
144	90	É
145	91	æ
146	92	Æ
147	93	ô
148	94	ö
149	95	ò
150	96	û
151	97	ù
152	98	ÿ
153	99	Ö
154	9A	Ü
155	9B	¢
156	9C	£
157	9D	¥
158	9E	Pts
159	9F	ƒ

Dec	Hex	ASCII
160	A0	á
161	A1	í
162	A2	ó
163	A3	ú
164	A4	ñ
165	A5	Ñ
166	A6	ª
167	A7	º
168	A8	¿
169	A9	⌐
170	AA	¬
171	AB	½
172	AC	¼
173	AD	¡
174	AE	«
175	AF	»
176	B0	▒
177	B1	▒
178	B2	▓
179	B3	│
180	B4	┤
181	B5	╡
182	B6	╢
183	B7	╖
184	B8	╕
185	B9	╣
186	BA	║
187	BB	╗
188	BC	╝
189	BD	╜
190	BE	╛
191	BF	┐

Dec	Hex	ASCII
192	C0	└
193	C1	┴
194	C2	┬
195	C3	├
196	C4	─
197	C5	┼
198	C6	╞
199	C7	╟
200	C8	╚
201	C9	╔
202	CA	╩
203	CB	╦
204	CC	╠
205	CD	═
206	CE	╬
207	CF	╧
208	D0	╨
209	D1	╤
210	D2	╥
211	D3	╙
212	D4	╘
213	D5	╒
214	D6	╓
215	D7	╫
216	D8	╪
217	D9	┘
218	DA	┌
219	DB	▉
220	DC	▄
221	DD	▌
222	DE	▐
223	DF	▀

Dec	Hex	ASCII
224	E0	α
225	E1	ß
226	E2	Γ
227	E3	π
228	E4	Σ
229	E5	σ
230	E6	µ
231	E7	τ
232	E8	Φ
233	E9	Θ
234	EA	Ω
235	EB	δ
236	EC	∞
237	ED	φ
238	EE	ε
239	EF	∩
240	F0	≡
241	F1	±
242	F2	≥
243	F3	≤
244	F4	⌠
245	F5	⌡
246	F6	÷
247	F7	≈
248	F8	°
249	F9	∙
250	FA	·
251	FB	√
252	FC	ⁿ
253	FD	²
254	FE	■
255	FF	Hardspace

										128	(1)					
						192	(2)			192	(2)			192	(2)	
				00000000	0		01000000	64		10000000	128		11000000	192
				00000001	1		01000001	65		10000001	129		11000001	193
				00000010	2		01000010	66		10000010	130		11000010	194
			252	00000011	3		01000011	67		10000011	131		11000011	195
			(6)	00000100	4		01000100	68		10000100	132		11000100	196
				00000101	5		01000101	69		10000101	133		11000101	197
				00000110	6		01000110	70		10000110	134		11000110	198
		248	252	00000111	7		01000111	71		10000111	135		11000111	199
		(5)	(6)	00001000	8		01001000	72		10001000	136		11001000	200
				00001001	9		01001001	73		10001001	137		11001001	201
				00001010	10		01001010	74		10001010	138		11001010	202
			252	00001011	11		01001011	75		10001011	139		11001011	203
			(6)	00001100	12		01001100	76		10001100	140		11001100	204
				00001101	13		01001101	77		10001101	141		11001101	205
				00001110	14		01001110	78		10001110	142		11001110	206
	240	248	252	00001111	15		01001111	79		10001111	143		11001111	207
	(4)	(5)	(6)	00010000	16		01010000	80		10010000	144		11010000	208
				00010001	17		01010001	81		10010001	145		11010001	209
				00010010	18		01010010	82		10010010	146		11010010	210
			252	00010011	19		01010011	83		10010011	147		11010011	211
			(6)	00010100	20		01010100	84		10010100	148		11010100	212
				00010101	21		01010101	85		10010101	149		11010101	213
				00010110	22		01010110	86		10010110	150		11010110	214
		248	252	00010111	23		01010111	87		10010111	151		11010111	215
		(5)	(6)	00011000	24		01011000	88		10011000	152		11011000	216
				00011001	25		01011001	89		10011001	153		11011001	217
				00011010	26		01011010	90		10011010	154		11011010	218
			252	00011011	27		01011011	91		10011011	155		11011011	219
			(6)	00011100	28		01011100	92		10011100	156		11011100	220
				00011101	29		01011101	93		10011101	157		11011101	221
				00011110	30		01011110	94		10011110	158		11011110	222
	224	240	248	252	00011111	31		01011111	95		10011111	159		11011111	223
	(3)	(4)	(5)	(6)	00100000	32		01100000	96		10100000	160		11100000	224
					00100001	33		01100001	97		10100001	161		11100001	225
					00100010	34		01100010	98		10100010	162		11100010	226
				252	00100011	35		01100011	99		10100011	163		11100011	227
				(6)	00100100	36		01100100	100		10100100	164		11100100	228
					00100101	37		01100101	101		10100101	165		11100101	229
					00100110	38		01100110	102		10100110	166		11100110	230
			248	252	00100111	39		01100111	103		10100111	167		11100111	231
			(5)	(6)	00101000	40		01101000	104		10101000	168		11101000	232
					00101001	41		01101001	105		10101001	169		11101001	233
					00101010	42		01101010	106		10101010	170		11101010	234
				252	00101011	43		01101011	107		10101011	171		11101011	235
				(6)	00101100	44		01101100	108		10101100	172		11101100	236
					00101101	45		01101101	109		10101101	173		11101101	237
					00101110	46		01101110	110		10101110	174		11101110	238
		240	248	252	00101111	47		01101111	111		10101111	175		11101111	239
		(4)	(5)	(6)	00110000	48		01110000	112		10110000	176		11110000	240
					00110001	49		01110001	113		10110001	177		11110001	241
					00110010	50		01110010	114		10110010	178		11110010	242
				252	00110011	51		01110011	115		10110011	179		11110011	243
				(6)	00110100	52		01110100	116		10110100	180		11110100	244
					00110101	53		01110101	117		10110101	181		11110101	245
					00110110	54		01110110	118		10110110	182		11110110	246
			248	252	00110111	55		01110111	119		10110111	183		11110111	247
			(5)	(6)	00111000	56		01111000	120		10111000	184		11111000	248
					00111001	57		01111001	121		10111001	185		11111001	249
					00111010	58		01111010	122		10111010	186		11111010	250
				252	00111011	59		01111011	123		10111011	187		11111011	251
				(6)	00111100	60		01111100	124		10111100	188		11111100	252
					00111101	61		01111101	125		10111101	189		11111101	253
					00111110	62		01111110	126		10111110	190		11111110	254
					00111111	63		01111111	127		10111111	191		11111111	255