29

TCP/IP Suite

The Defense Advance Research Projects Agency (DARPA) originally developed Transmission Control Protocol/Internet Protocol (TCP/IP) to interconnect various defense department computer networks. The Internet, an international Wide Area Network, uses TCP/IP to connect government and educational institutions across the world. TCP/IP is also in widespread use on commercial and private networks. The TCP/IP suite includes the following protocols:

- IP / IPv6: Internet Protocol.
- TCP: Transmission Control Protocol.
- UDP: User Datagram Protocol.

Data Link Layer

- ARP/RARP: Address Resolution Protocol/Reverse Address.

Tunneling protocols

- ATMP: Ascend Tunnel Management Protocol.
- L2F: Layer 2 Forwarding Protocol.
- L2TP: Layer 2 Tunneling Protocol.
- PPTP: Point-to-Point Tunneling Protocol.

Network Layer

- DHCP / DHCPv6: Dynamic Host Configuration Protocol.
- DVMRP: Distance Vector Multicast Routing Protocol.
- ICMP / ICMPv6: Internet Control Message Protocol.
- IGMP: Internet Group Management Protocol.
- MARS: Multicast Address Resolution Server.
- PIM: Protocol Independent Multicast.
- RIP: Routing Information Protocol.
- RIPng for IPv6.
- RSVP: Resource ReSerVation setup Protocol.

Security

- AH: Authentication Header.
- ESP: Encapsulating Security Payload.

Routing

- BGP-4: Border Gateway Protocol.
- EGP: Exterior Gateway Protocol.
- EIGRP: Enhanced Interior Gateway Routing Protocol.
- GRE: Generic Routing Encapsulation.
- HSRP: Cisco Hot Standby Router Protocol.
- IGRP: Interior Gateway Routing.
- NARP: NBMA Address Resolution Protocol.
- NHRP: Next Hop Resolution Protocol.
- OSPF: Open Shortest Path First.

Transport Layer

- Mobile IP.
- Van Jacobson: compressed TCP.
- XOT: X.25 over TCP.

VoIP

- MGCP: Media Gateway Control Protocol.
- SGCP: Simple Gateway Control Protocol.

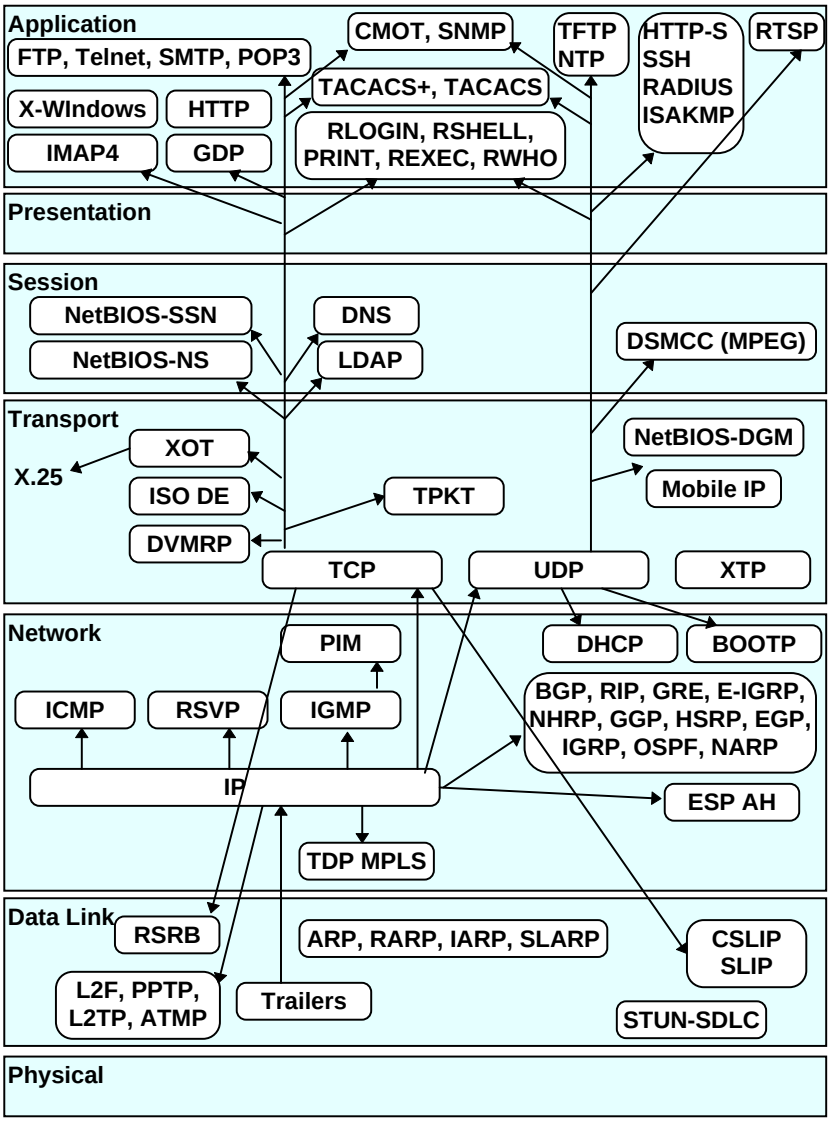
Session Layer

- DNS: Domain Name Service.
- NetBIOS/IP.

Application Layer

- FTP: File Transfer Protocol.
- TFTP: Trivial File Transfer Protocol.
- Finger: User Information Protocol.
- Gopher: Internet Gopher Protocol.
- HTTP: Hypertext Transfer Protocol.
- S-HTTP: Secure Hypertext Transfer Protocol.
- IMAP4: Internet Message Access Protocol rev 4.
- IPDC: IP Device Control.
- ISAPMP: Internet Key Exchange.
- NTP: Network Time Protocol.
- POP3: Post Office Protocol version 3.
- Radius.
- RLOGIN: Remote Login.
- RTSP: Real-time Streaming Protocol.
- SMTP: Simple Mail Transfer Protocol.
- SNMP: Simple Network Management Protocol.
- TACACS+: Terminal Access Controller Access Control System.
- TELNET.
- X-Window.

The following diagram illustrates the TCP/IP suite in relation to the OSI model:



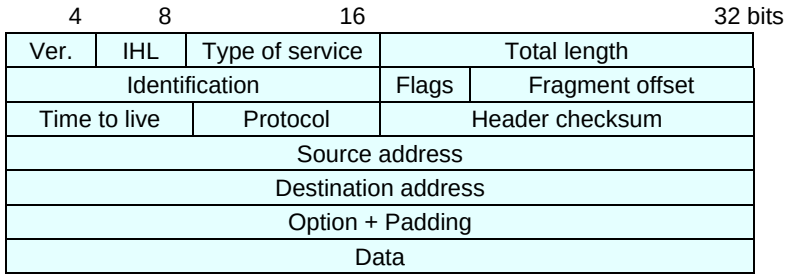
TCP/IP in relation to the OSI model

IP

IETF RFC 791 1981-09 http://www.cis.ohio-state.edu/htbin/rfc/rfc791.html
IETF RFC 1853 1995-1 http://www.cis.ohio-state.edu/htbin/rfc/rfc1853.html

The Internet Protocol (IP) is the routing layer datagram service of the TCP/IP suite. All other protocols within the TCP/IP suite, except ARP and RARP, use IP to route frames from host to host. The IP frame header contains routing information and control information associated with datagram delivery.

The IP header structure is as follows:



IP header structure

Version

Version field indicates the format of the Internet header.

IHL

Internet header length is the length of the Internet header in 32-bit words. Points to the beginning of the data. The minimum value for a correct header is 5.

Type of service

Indicates the quality of service desired. Networks may offer service precedence, meaning that they accept traffic only above a certain precedence at times of high load. There is a three-way trade-off between low delay, high reliability and high throughput.

Bits 0-2: Precedence
111 Network control.
110 Internetwork control.

101 CRITIC/ECP.
100 Flash override.
011 Flash.
010 Immediate.
001 Priority.
000 Routine.

Bit 3: Delay
0 Normal delay.
1 Low delay.

Bit 4: Throughput
0 Normal throughput.
1 High throughput.

Bit 5: Reliability
0 Normal reliability.
1 High reliability.

Bits 6-7: Reserved for future use.

Total length

Length of the datagram measured in bytes, including the Internet header and data. This field allows the length of a datagram to be up to 65,535 bytes, although such long datagrams are impractical for most hosts and networks. All hosts must be prepared to accept datagrams of up to 576 bytes, regardless of whether they arrive whole or in fragments. It is recommended that hosts send datagrams larger than 576 bytes only if the destination is prepared to accept the larger datagrams.

Identification

Identifying value assigned by the sender to aid in assembling the fragments of a datagram.

Flags

3 bits. Control flags:

Bit 0 is reserved and must be zero.

Bit 1: Don't fragment bit:
0 May fragment.
1 Don't fragment.

Bit 2: More fragments bit:
0 Last fragment.
1 More fragments.

Fragment offset
13 bits. Indicates where this fragment belongs in the datagram. The fragment offset is measured in units of 8 bytes (64 bits). The first fragment has offset zero.

Time to live
Indicates the maximum time the datagram is allowed to remain in the Internet system. If this field contains the value zero, the datagram must be destroyed. This field is modified in Internet header processing. The time is measured in units of seconds. However, since every module that processes a datagram must decrease the TTL by at least one (even if it processes the datagram in less than 1 second), the TTL must be thought of only as an upper limit on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded and to bound the maximum datagram lifetime.

Protocol
Indicates the next level protocol used in the data portion of the Internet datagram.

Header checksum
A checksum on the header only. Since some header fields change, e.g., Time To Live, this is recomputed and verified at each point that the Internet header is processed.

Source address / destination address
32 bits each. A distinction is made between names, addresses and routes. A *name* indicates an object to be sought. An *address* indicates the location of the object. A *route* indicates how to arrive at the object. The Internet protocol deals primarily with addresses. It is the task of higher level protocols (such as host-to-host or application) to make the mapping from names to addresses. The Internet module maps Internet addresses to local net addresses. It is the task of lower level procedures (such as local net or gateways) to make the mapping from local net addresses to routes.

Options

Options may or may not appear in datagrams. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation. In some environments, the security option may be required in all datagrams.

The option field is variable in length. There may be zero or more options. There are two possible formats for an option:

- A single octet of option type.
- An option type octet, an option length octet and the actual option data octets.

The length octet includes the option type octet and the actual option data octets.

The option type octet has 3 fields:

1 bit: Copied flag. Indicates that this option is copied into all fragments during fragmentation:
0 Copied.
1 Not copied.

2 bits: Option class.
0 Control.
1 Reserved for future use.
2 Debugging and measurement.
3 Reserved for future use.

5 bits: Option number.

Data

IP data or higher layer protocol header.

IPv6

IETF RFC 1883 http://www.cis.ohio-state.edu/htbin/rfc/rfc1883.html
IETF RFC 1826 http://www.cis.ohio-state.edu/htbin/rfc/rfc1826.html
IETF RFC 1827 1995-12 http://www.cis.ohio-state.edu/htbin/rfc/rfc1827.html

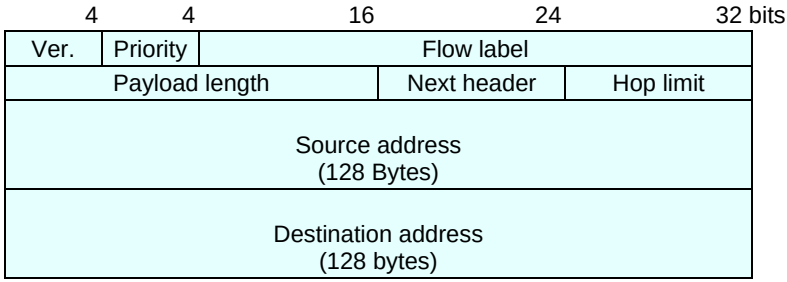
IP version 6 (IPv6) is an updated version of the Internet Protocol based on IPv4. IPv4 and IPv6 are demultiplexed at the media layer. For example, IPv6 packets are carried over Ethernet with the content type 86DD (hexadecimal) instead of IPv4's 0800.

IPv6 increases the IP address size from 32 bits to 128 bits, to support more levels of addressing hierarchy, a much greater number of addressable nodes and simpler auto-configuration of addresses. Scalability of multicast addresses is introduced. A new type of address called an *anycast address* is also defined, to send a packet to any one of a group of nodes.

Improved support for extensions and options - IPv6 options are placed in separate headers that are located between the IPv6 header and the transport layer header. Changes in the way IP header options are encoded allow more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future. The extension headers are: Hop-by-Hop Option, Routing (Type 0), Fragment, Destination Option, Authentication, Encapsulation Payload.

Flow labeling capability - A new capability has been added to enable the labeling of packets belonging to particular traffic flows for which the sender requests special handling, such as non-default Quality of Service or real-time service.

The IPv6 header structure is as follows:



IPv6 header structure

Version
Internet Protocol Version number (IPv6 is 6).

Priority
Enables a source to identify the desired delivery priority of the packets. Priority values are divided into ranges: traffic where the source provides congestion control and non-congestion control traffic.

Flow label
Used by a source to label those products for which it requests special handling by the IPv6 router. The flow is uniquely identified by the combination of a source address and a non-zero flow label.

Payload length
Length of payload (in octets).

Next header
Identifies the type of header immediately following the IPv6 header.

Hop limit
8-bit integer that is decremented by one, by each node that forwards the packet. The packet is discarded if the Hop Limit is decremented to zero.

Source address
128-bit address of the originator of the packet.

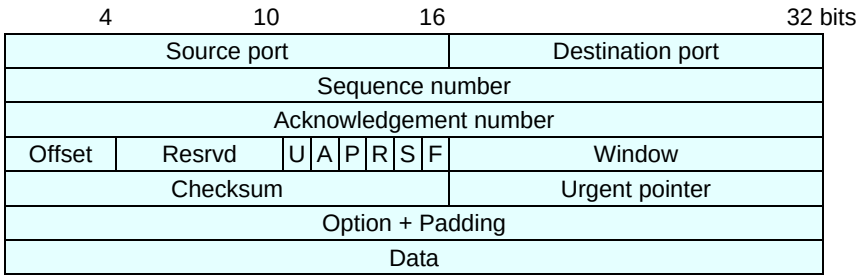
Destination address
128-bit address of the intended recipient of the packet.

TCP

IETF RFC 793 1981-09 http://www.cis.ohio-state.edu/htbin/rfc/rfc793.html
IETF RFC 1072 1988-10 http://www.cis.ohio-state.edu/htbin/rfc/rfc1072.html
IETF RFC 1693 1994-11 http://www.cis.ohio-state.edu/htbin/rfc/rfc1693.html
IETF RFC 1146 1990-03 http://www.cis.ohio-state.edu/htbin/rfc/rfc1146.html
IETF RFC 1323 1992-05 http://www.cis.ohio-state.edu/htbin/rfc/rfc1323.html

The Transmission Control Protocol (TCP) provides a reliable stream delivery and virtual connection service to applications through the use of sequenced acknowledgment with retransmission of packets when necessary.

The TCP header structure is as follows:



TCP header structure

Source port
Source port number.

Destination port
Destination port number.

Sequence number
The sequence number of the first data octet in this segment (except when SYN is present). If SYN is present, the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1.

Acknowledgment number
If the ACK control bit is set, this field contains the value of the next sequence number which the sender of the segment is expecting to receive. Once a connection is established, this value is always sent.

Data offset
4 bits. The number of 32-bit words in the TCP header, which indicates where the data begins. The TCP header (even one including options) has a length which is an integral number of 32 bits.

Reserved
6 bits. Reserved for future use. Must be zero.

Control bits
6 bits. The control bits may be (from right to left):
U (URG) Urgent pointer field significant.
A (ACK) Acknowledgment field significant.
P (PSH) Push function.
R (RST) Reset the connection.
S (SYN) Synchronize sequence numbers.
F (FIN) No more data from sender.

Window
16 bits. The number of data octets which the sender of this segment is willing to accept, beginning with the octet indicated in the acknowledgment field.

Checksum
16 bits. The checksum field is the 16 bit one's complement of the one's complement sum of all 16-bit words in the header and text. If a segment contains an odd number of header and text octets to be checksummed, the last octet is padded on the right with zeros to form a 16-bit word for checksum purposes. The pad is not transmitted as part of the segment. While computing the checksum, the checksum field itself is replaced with zeros.

Urgent Pointer
16 bits. This field communicates the current value of the urgent pointer as a positive offset from the sequence number in this segment. The urgent pointer points to the sequence number of the octet following the urgent data. This field can only be interpreted in segments for which the URG control bit has been set.

Options

Options may be transmitted at the end of the TCP header and always have a length which is a multiple of 8 bits. All options are included in the checksum. An option may begin on any octet boundary.

There are two possible formats for an option:

- A single octet of option type.
- An octet of option type, an octet of option length, and the actual option data octets.

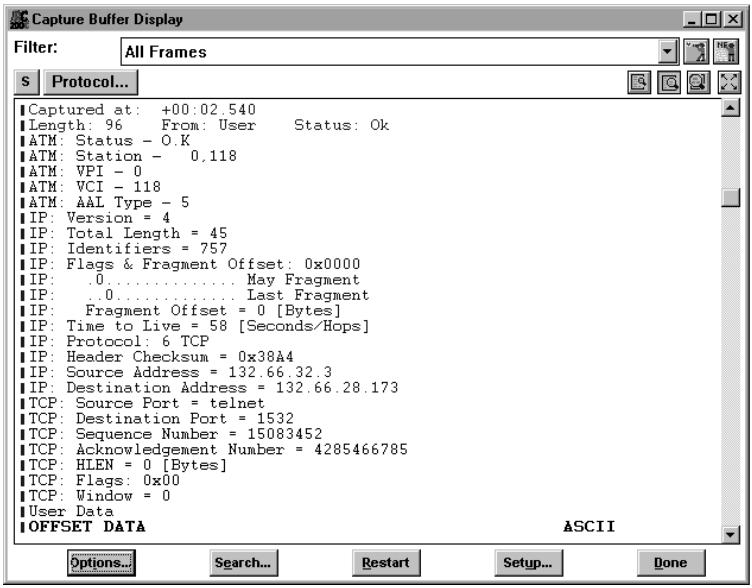
The option length includes the option type and option length, as well as the option data octets.

The list of options may be shorter than that designated by the data offset field because the contents of the header beyond the End-of-Option option must be header padding i.e., zero.

A TCP must implement all options.

Data

TCP data or higher layer protocol.



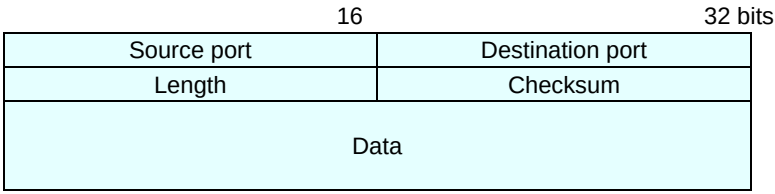
TCP/IP over ATM decode

UDP

RFC 768 1980-08 http://www.cis.ohio-state.edu/htbin/rfc/rfc768.html

The User Datagram Protocol (UDP) provides a simple, but unreliable message service for transaction-oriented services. Each UDP header carries both a source port identifier and destination port identifier, allowing high-level protocols to target specific applications and services among hosts.

The UDP header structure is shown as follows:



UDP header structure

Source port

Source port is an optional field. When used, it indicates the port of the sending process and may be assumed to be the port to which a reply should be addressed in the absence of any other information. If not used, a value of zero is inserted.

Destination port

Destination port has a meaning within the context of a particular Internet destination address.

Length

The length in octets of this user datagram, including this header and the data. The minimum value of the length is eight.

Checksum

The 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.

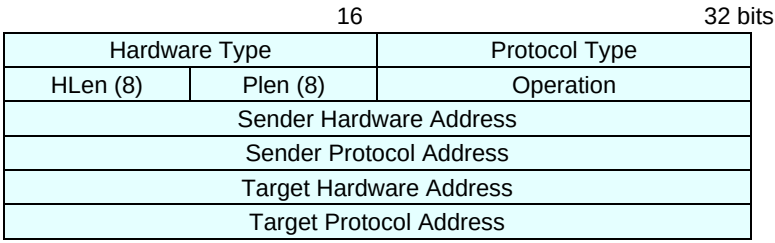
Data
UDP data field.

ARP/RARP

IETF RFC 826 1982-11 http://www.cis.ohio-state.edu/htbin/rfc/rfc826.html
IETF RFC 1390 1993-01 http://www.cis.ohio-state.edu/htbin/rfc/rfc1390.html
IETF RFC 1293 1992-01 http://www.cis.ohio-state.edu/htbin/rfc/rfc1293.html

TCP/IP uses the Address Resolution Protocol (ARP) and the Reverse Address Resolution Protocol (RARP) to initialize the use of Internet addressing on an Ethernet or other network that uses its own media access control (MAC). ARP allows a host to communicate with other hosts when only the Internet address of its neighbors is known. Before using IP, the host sends a broadcast ARP request containing the Internet address of the desired destination system.

The ARP/RARP header structure is shown in the illustration below.



ARP/RARP header structure

Hardware type
Specifies a hardware interface type for which the sender requires a response.

Protocol type
Specifies the type of high-level protocol address the sender has supplied.

HLen
Hardware address length.

PLen
Protocol address length.

Operation

The values are as follows:

1 ARP request.
2 ARP response.
3 RARP request.
4 RARP response.
5 Dynamic RARP request.
6 Dynamic RARP reply.
7 Dynamic RARP error.
8 InARP request.
9 InARP reply.

Sender hardware address

HLen bytes in length.

Sender protocol address

PLen bytes in length.

Target hardware address

HLen bytes in length.

Target protocol address

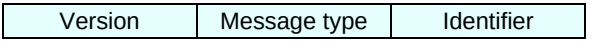
PLen bytes in length.

ATMP

RFC 2107 http://www.cis.ohio-state.edu/htbin/rfc/rfc2107.html

The Ascend Tunnel Management Protocol (ATMP) is a protocol currently being used in Ascend Communication products to allow dial-in client software to obtain virtual presence on a user's home network from remote locations. A user calls into a remote NAS but instead of using an address belonging to a network directly supported by the NAS, the client software uses an address belonging to the user's "Home Network". This address can be either provided by the client software or assigned from a pool of addresses from the Home Network address space. In either case, this address belongs to the Home Network and therefore special routing considerations are required in order to route packets to and from these clients. A tunnel between the NAS and a special "Home Agent" (HA) located on the Home Network is used to carry data to and from the client.

The format of the ATMP header is shown in the following illustration:



ATMP packet structure

Version

The ATMP protocol version must be 1.

Message type

ATMP defines a set of request and reply messages sent with UDP. There are 7 different ATMP message types represented by the following values.

MessageType	***Type Code***
Registration Request	1
Challenge Request	2
Challenge Reply	3
Registration Reply	4
Deregister Request	5
Deregister Reply	6
Error Notification	7

Identifier

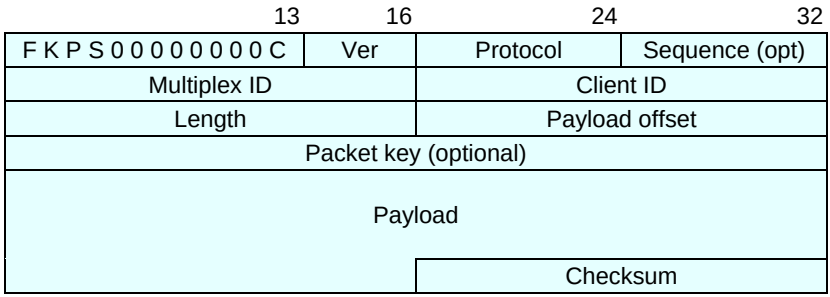
A 16 bit number used to match replies with requests. A new value should be provided in each new request. Retransmissions of the same request should use the same identifier.

L2F

RFC 2341 http://www.cis.ohio-state.edu/htbin/rfc/rfc2341.html

The Layer 2 Forwarding protocol (L2F) permits the tunneling of the link layer of higher layer protocols. Using such tunnels it is possible to divorce the location of the initial dial-up server from the location at which the dial-up protocol connection is terminated and access to the network provided.

The format of the packet is shown in the following illustration:



L2F packet structure

Version
The major version of the L2F software creating the packet.

Protocol
The protocol field specifies the protocol carried within the L2F packet.

Sequence
The sequence number is present if the S bit in the L2F header is set to 1.

Multiplex ID
The packet multiplex ID identifies a particular connection within a tunnel.

Client ID
The client ID (CLID) assists endpoints in demultiplexing tunnels.

Length
The length is the size in octets of the entire packet, including the header, all the fields and the payload.

Payload offset
This field specifies the number of bytes past the L2F header at which the payload data is expected to start. This field is present if the F bit in the L2F header is set to 1.

Packet key
The key field is present if the K bit is set in the L2F header. This is part of the authentication process.

Checksum
The checksum of the packet. The checksum field is present if the C bit in the L2F header is set to 1.

Option Messages

When the link is initiated, the endpoints communicate to verify the presence of L2F on the remote end, and to permit any needed authentication. The protocol for such negotiation is always 1, indicating L2F management. The message itself is structured as a sequence of single octets indicating an option. When the protocol field of an L2F specifies L2F management, the body of the packet is encoded as zero or more options. An option is a single octet message type, followed by zero or more sub-options. Each sub-option is a single byte sub-option value, and followed by additional bytes as appropriate for the sub-option.

Possible option messages are:

Invalid	Invalid message.
L2F_CONF	Request configuration.
L2F_CONF_NAME	Name of peer sending L2F_CONF.
L2F_CONF_CHAL	Random number peer challenges.
L2F_CONF_CLID	Assigned_CLID for peer to use.
L2F_OPEN	Accept configuration.
L2F_OPEN_NAME	Name received from client.
L2F_OPEN_CHAL	Challenge client received.
L2F_OPEN_RESP	Challenge response from client.
L2F_ACK_LCP1	LCP CONFACK accepted from client.
L2F_ACK_LCP2	LCP CONFACK sent to client.
L2F_OPEN_TYPE	Type of authentication used.
L2F_OPEN_ID	ID associated with authentication.
L2F_REQ_LCP0	First LCP CONFREQ from client.
L2F_CLOSE	Request disconnect.

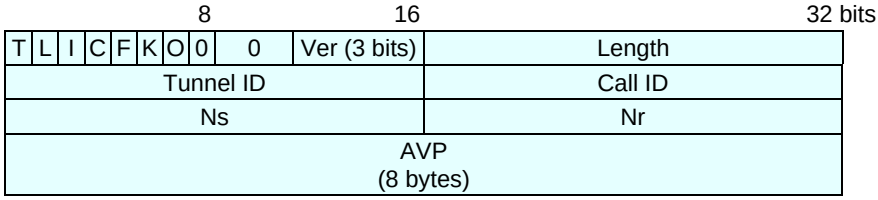
L2F_CLOSE_WHY	Reason code for close.
L2F_CLOSE_STR	ASCII string description.
L2F_ECHO	Verify presence of peer.
L2F_ECHO_RESP	Respond to L2F_ECHO.

L2TP

IETF draft
http://info.internet.isi.edu:80/in-drafts/files/draft-ietf-pppext-l2tp-11.txt

The L2TP Protocol is used for integrating multi-protocol dial-up services into existing Internet Service Providers Point of Presence (hereafter referred to as ISP and POP, respectively). This protocol may also be used to solve the "multilink hunt-group splitting" problem. Multilink PPP, often used to aggregate ISDN B channels, requires that all channels composing a multilink bundle be grouped at a single Network Access Server (NAS). Because L2TP makes a PPP session appear at a location other than the physical point at which the session was physically received, it can be used to make all channels appear at a single NAS, allowing for a multilink operation even when the physical calls are spread across distinct physical NASs.

The format of the L2TP packet is shown in the following illustration:



L2TP packet structure

T

The T bit is 1 for control messages and 0 for payload messages. For control messages, the following seven bits must be set to 1001000, making the header more compatible in encoding with the payload message.

L

When set, this indicates that the Length field is present, indicating the total length of the received packet. Must be set for control messages.

I & C

The I and C bits are reserved and must be set to 0. These bit positions represent options no longer present in L2TP.

F
If the F bit is set, both the Nr and Ns fields are present. F must be set for control messages.

K
The K bit is reserved and must be set to 0.

O
When set, this field indicates that the Offset Size field is present in payload messages.

Ver
The value of the ver bit is always 002. This indicates a version 1 L2TP message.

Length
Overall length of the message, including header, message type AVP, plus any additional AVP's associated with a given control message type.

Tunnel ID
Identifies the tunnel to which a control message applies. If an Assigned Tunnel ID has not yet been received from the peer, Tunnel ID must be set to 0. Once an Assigned Tunnel ID is received, all further packets must be sent with Tunnel ID set to the indicated value.

Call ID
Identifies the user session within a tunnel to which a control message applies. If a control message does not apply to a single user session within the tunnel (for instance, a Stop-Control-Connection-Notification message), Call ID must be set to 0.

Nr
Currently transmitted packet.

Ns
Latest received packet.

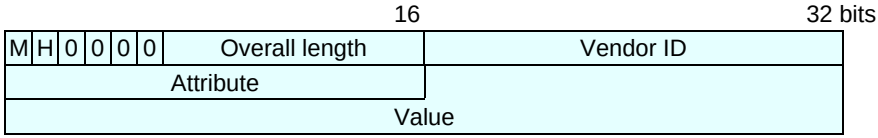
Payload messages have two additional fields before the AVP as follows:

Offset size (16 bits)	Offset pad (16 bits)

Additional fields in L2TP payload message

Offset size
This field specifies the number of bytes past the L2TP header at which the payload data is expected to start. It is recommended that data thus skipped be initialized to 0s. If the offset size is 0, or the O bit is not set, the first byte following the last byte of the L2TP header is the first byte of payload data.

AVP
The AVP (Attribute-Value Pair) is a uniform method used for encoding message types and bodies throughout L2TP. The format of the AVP is given below:



L2TP AVP structure

M
The first six bits are a bit mask, describing the general attributes of the AVP. The M bit, known as the mandatory bit, controls the behavior required of an implementation which receives an AVP which it does not recognize.

H
The hidden bit controls the hiding of the data in the value field of an AVP. This capability can be used to avoid the passing of sensitive data, such as user passwords, as cleartext in an AVP.

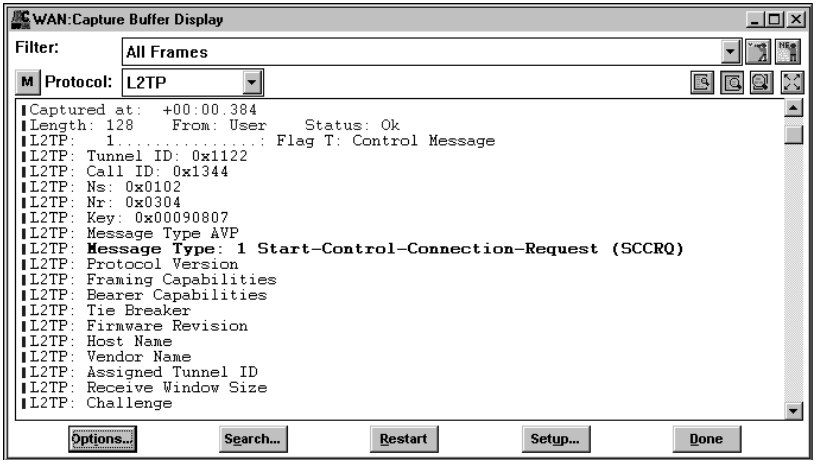
Overall length
Encodes the number of octets (including the overall length field itself) contained in this AVP. It is 10 bits, permitting a maximum of 1024 bytes of data in a single AVP.

Vendor ID
The IANA assigned SMI Network Management Private Enterprise Codes value, encoded in network byte order.

Attribute
The actual attribute, a 16-bit value with a unique interpretation across all AVP's defined under a given Vendor ID.

Value

The value field follows immediately after the Attribute field, and runs for the remaining octets indicated in the overall length (i.e., overall length minus six octets of header).



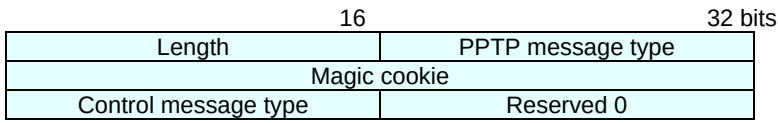
L2TP decode

PPTP

IETF draft
http://info.internet.isi.edu:80/in-drafts/files/draft-ietf-pppext-pptp-04.txt

PPTP (Point to Point Tunneling Protocol) allows PPP to be channeled through an IP network. It uses a client-server architecture to decouple functions which exist in current Network Access Servers and support Virtual Private Networks. It specifies a call-control and management protocol which allows the server to control access for dial-in circuit switched calls originating from a PSTN or ISDN, or to initiate outbound circuit switched connections. PPTP uses a GRE-like (Generic Routing Encapsulation) mechanism to provide a flow- and congestion-controlled encapsulated datagram service for carrying PPP packets.

The format of the header is shown in the following illustration:



PPTP header structure

Length

Total length in octets of this PPTP message including the entire PPTP header.

PPTP message type

The message type. Possible values are:

1 Control message.
2 Management message.

Magic cookie

The magic cookie is always sent as the constant 0x1A2B3C4D. Its basic purpose is to allow the receiver to ensure that it is properly synchronized with the TCP data stream.

Control Message Type

Values may be:

1 Start-Control-Connection-Request.
2 Start-Control-Connection-Reply.
3 Stop-Control-Connection-Request.
4 Stop-Control-Connection-Reply.
5 Echo-Request.
6 Echo-Reply.

Call Management

7 Outgoing-Call-Request.
8 Outgoing-Call-Reply.
9 Incoming-Call-Request.
10 Incoming-Call-Reply.
11 Incoming-Call-Connected.
12 Call-Clear-Request.
13 Call-Disconnect-Notify.

Error Reporting

14 WAN-Error-Notify.

PPP Session Control

15 Set-Link-Info.

Reserved

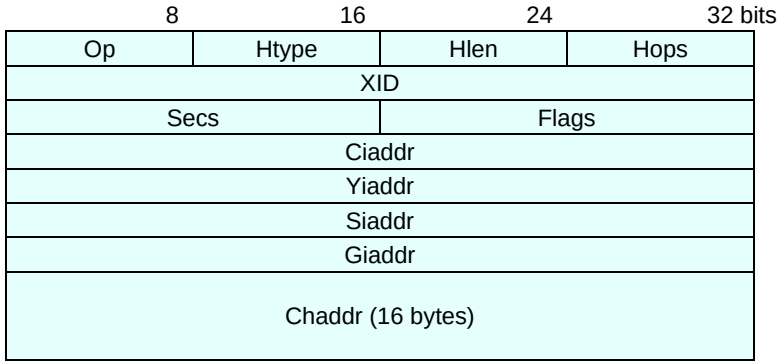
A reserved field, must be set to 0.

DHCP

RFC 1531 http://www.cis.ohio-state.edu/htbin/rfc/rfc1531.html

The Dynamic Host Configuration Protocol (DHCP) provides Internet hosts with configuration parameters. DHCP is an extension of BOOTP. DHCP consists of two components: a protocol for delivering host-specific configuration parameters from a DHCP server to a host and a mechanism for allocation of network addresses to hosts.

The format of the header is shown in the following illustration:



DHCP header structure

Op
The message operation code. Messages can be either BOOTREQUEST or BOOTREPLY.

Htype
The hardware address type.

Hlen
The hardware address length.

XID
The transaction ID.

Secs
The seconds elapsed since the client began the address acquisition or renewal process.

Flags
The flags.

Ciaddr
The client IP address.

Yiaddr
The "Your" (client) IP address.

Siaddr
The IP address of the next server to use in bootstrap.

Giaddr
The relay agent IP address used in booting via a relay agent.

Chaddr
The client hardware address.

DHCPv6

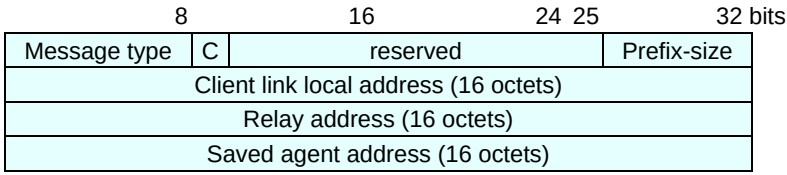
http://www.ietf.org/internet-drafts/draft-ietf-dhc-dhcpv6-14.txt

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) enables DHCP servers to pass configuration information, via extensions, to IPv6 nodes. It offers the capability of automatic allocation of reusable network addresses and additional configuration flexibility. This protocol is a stateful counterpart to the IPv6 Stateless Address Autoconfiguration protocol, and can be used separately or together with the latter to obtain configuration information.

DHCPv6 has 6 different message types: Solicit, Advertise, Request, Reply, Release and Reconfigure.

DHCP Solicit message

A client transmits a DHCP Solicit message over the interface to be configured, to obtain one or more server addresses. Unless otherwise noted, the value of all fields are set by the client.



DHCP Solicit message structure

Message type

Value of 1 specifies a Solicit message.

C

Indicates that the client requests that all servers receiving the message deallocate the resources associated with the client. When set, the client should provide a saved agent address to locate the clients binding by a server.

Prefix size

When non-zero, indicates the number of left-most bits of the agent's IPv6 address which conprise the routing prefix.

Reserved
Set to zero.

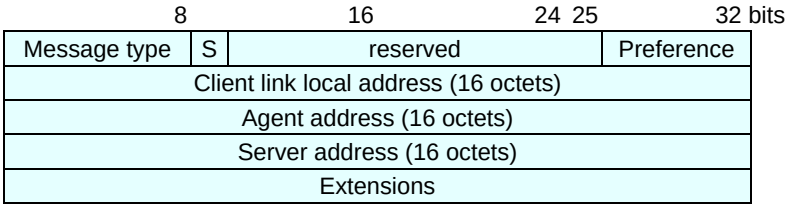
Client link local address
IP link local address of the client interface from which the client issued the DHCP Request message.

Relay address
Set by the client to zero. If received by a DHCP relay this is set by the relay to the IP address of the interface on which the relay received the client's DHCP Solicit message.

Saved agent address
When present, indicates the IP address of an agent's interface retained by the client from a previous DHCP transaction.

DHCP Advertise message

A DHCP agent sends a DHCP Advertise message to inform a prospective client about the IP address of a server to which a DHCP Request message may be sent. When the client and server are on different links, the server sends the advertisement back through the relay whence the solicitation came. The value of all fields in the DHCP Advertise message are filled in by the DHCP server and not changed by any DHCP relay.



DHCP Advertise message structure

Message type
Value of 2 specifies an Advertise message.

S
If set, specifies that the server address is present.

Preference
Indicates a server's willingness to provide service to the client.

Client link local address
IP link local address of the client interface from which the client issued the DHCP Request message.

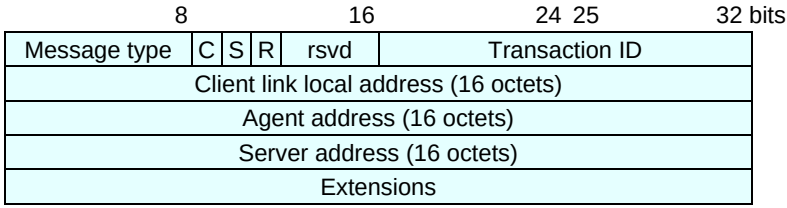
Agent address
IP address of a DHCP agent interface on the same link as the client.

Server address
When present, the IP address of the DHCP server.

Extensions
Described in the standard.

DHCP Request message

In order to request configuration parameters from a server, a client sends a DHCP Request message, and may append extensions. If the client does not know any server address, it must first obtain one by multicasting a DHCP Solicit message. Typically, when a client reboots, it does not have a valid IP address of sufficient scope for the server to communicate with the client. In such cases, the client cannot send the message directly to the server because the server could not return any response to the client. In this case, the client must send the message to the local relay and insert the relay address as the agent address in the message header.



DHCP Request message structure

Message type
Value of 3 specifies a Request message.

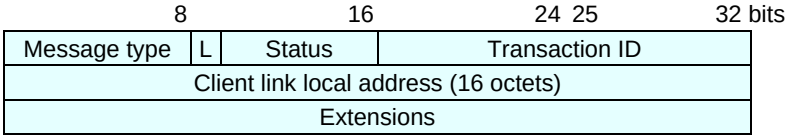
R
If set, specifies that the client has rebooted and requests that all of its previous transaction IDs be expunged and made available for reuse.

Transaction ID
Unsigned integer identifier used to identify this request.

The remaining fields are described in the Solicit and Advertise messages.

DHCP Reply message

The server sends one DHCP Reply message in response to every DHCP Request or DHCP Release received. If the request comes with the S bit set, the client could not directly send the Request to the server and had to use a neighboring relay agent. In that case, the server sends back the DHCP Reply with the L bit set, and the DHCP Reply is addressed to the agent-address found in the DHCP Request message. All the fields in the DHCP Reply message are set by the DHCP server.



DHCP Reply message structure

Message type
Value of 4 specifies a Reply message.

L
If set, the client link local address is present.

Status
May have the following values:

0 Success
16 Failure, reason unspecified
17 Authentication failed or nonexistent
18 Poorly formed Request or Release
19 Resources unavailable
20 Client record unavailable
21 Invalid client IP address in Release

23 Relay cannot find server address
64 Server unreachable (ICMP error)

Transaction ID

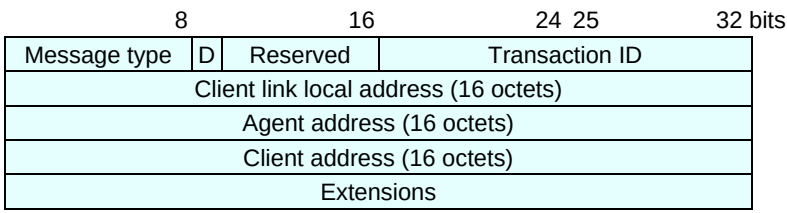
Unsigned integer identifier used to identify this Reply, copied from the client Request.

Client link local address

If present, the IP address of the client interface which issued the corresponding DHCP Request message. If the L bit is set, the client's link-local address is present in the Reply message. Then the Reply is sent by the server to the relay's address which was specified as the agent-address in the DHCP Request message, and the relay uses the link-local address to deliver the Reply message to the client. The transaction-ID in the DHCP Reply is copied by the server from the client Request message.

DHCP Release message

The DHCP Release message is sent without the assistance of any DHCP relay. When a client sends a Release message, it is assumed to have a valid IP address with sufficient scope to allow access to the target server. If parameters are specified in the extensions, only those parameters are released. The values of all fields of the DHCP Release message are entered by the Client. The DHCP server acknowledges the Release message by sending a DHCP Reply.



DHCP Release message structure

Message type

Value of 5 specifies a Release message.

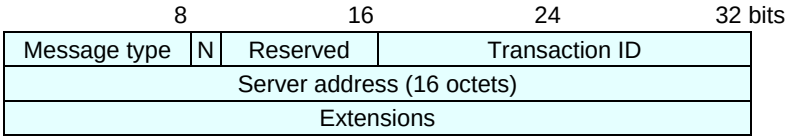
D
When set, the client instructs the server to send the DHCP Reply directly back to the client instead of using the given agent address and link local address to relay the Reply message.

Transaction ID
Unsigned integer identifier used to identify this Release, and copied into the Reply.

The remaining fields are described in the other DHCP messages.

DHCP Reconfigure message

DHCP Reconfigure messages can only be sent to clients which have established an IP address which routes to the link at which they are reachable, hence, the DHCP Reconfigure message is sent without the assistance of any DHCP relay. When a server sends a Reconfigure message, the receivers are assumed to have a valid IP address with sufficient scope to be accessible by the server. Only the parameters which are specified in the extensions to the Reconfigure message need be requested again by the client. A Reconfigure message can either be unicast or multicast by the server. The client extracts the extensions provided by the server and sends a DHCP Request message to the server using those extensions.



DHCP Reconfigure message structure

Message type
Value of 6 specifies a Reconfigure message.

N
Indicates that the client should not expect a DHCP Reply in response to the DHCP Request it sends as a result of the DHCP Reconfigure message.

The remaining fields are described in the other DHCP messages.

DVMRP

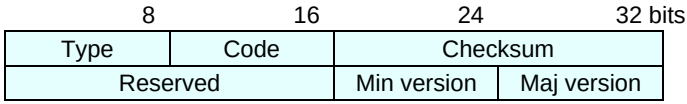
RFC 1075: http://www.cis.ohio-state.edu/htbin/rfc/rfc1075.html
IETF draft:
http://www.ietf.org/internet-drafts/draft-ietf-idmr-dvmrp-v3-08.txt

Distance Vector Multicast Routing Protocol (DVMRP) is an Internet routing protocol that provides an efficient mechanism for connectionless datagram delivery to a group of hosts across an internetwork. It is a distributed protocol that dynamically generates IP multicast delivery trees using a technique called Reverse Path Multicasting

DVMRP combines many of the features of RIP with the Truncated Reverse Path Broadcasting (TRPB) algorithm. DVMRP is developed based upon RIP because an implementation was available and distance vector algorithms are simple, as compared to link-state algorithms. In addition, to allow experiments to traverse networks that do not support multicasting, a mechanism called *tunneling* was developed.

DVMRP differs from RIP in one very important way. RIP routes and forwards datagrams to a particular destination. The purpose of DVMRP is to keep track of the return paths to the source of multicast datagrams. To make the explanation of DVMRP more consistent with RIP, the term *destination* is used instead of the more proper term *source*, however, datagrams are not forwarded to these destinations, but rather, originate from them.

DVMRP packets are encapsulated in IP datagrams, with an IP protocol number of 2 (IGMP). All fields are transmitted in Network Byte Order. DVMRP packets use a common protocol header that specifies the IGMP Packet Type as DVMRP. DVMRP protocol packets should be sent with the Precedence field in the IP header set to Internetwork Control (hexadecimal 0xc0 for the Type of Service Octet). The common protocol header is as shown in the following illustration:



DVMRP structure

Type

Packet type. 0x13 indicates a DVMRP packet.

Code
Determines the type of DVMRP packet. Currently, there are codes for DVMRP protocol message types as well as protocol analysis and troubleshooting packets. The protocol message codes may be as follows:

Probe	Neighbor discovery.
Report	Route exchange.
Prune	Pruning multicast delivery trees.
Graft	Grafting multicast delivery trees.
Graft ack	Acknowledging graft messages.

Checksum
16-bit one's complement of the one's complement sum of the DVMRP message. The checksum must be calculated upon transmission and must be validated on reception of a packet. The checksum of the DVMRP message should be calculated with the checksum field set to zero.

Reserved
Reserved for later use.

Min version
Minor version. Value must be 0xFF for this version of DVMRP.

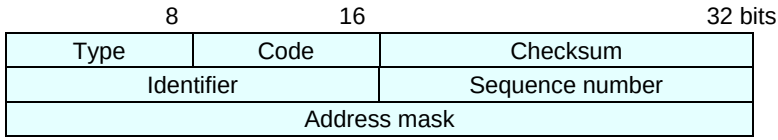
Maj version
Major version. Value must be 3 for this version of DVMRP.

ICMP

RFC 792 http://www.cis.ohio-state.edu/htbin/rfc/rfc792.html

Internet Control Message Protocol (ICMP) messages generally contain information about routing difficulties with IP datagrams or simple exchanges such as time-stamp or echo transactions.

The ICMP header structure is shown as follows:



ICMP header structure

Type	Code	Description
0		Echo reply.
3		Destination unreachable.
3	0	Net unreachable.
3	1	Host unreachable.
3	2	Protocol unreachable.
3	3	Port unreachable.
3	4	Fragmentation needed and DF set.
3	5	Source route failed.
4		Source quench.
5		Redirect.
5	0	Redirect datagrams for the network.
5	1	Redirect datagrams for the host.
5	2	Redirect datagrams for the type of service and network.
5	3	Redirect datagrams for the type of service and host.
8		Echo.
11		Time exceeded.
11	0	Time to live exceeded in transit.
11	1	Fragment reassemble time exceeded.
12		Parameter problem.
13		Timestamp.
14		Timestamp reply.

Type	Code	Description
15		Information request.
16		Information reply.

Checksum
The 16-bit one's complement of the one's complement sum of the ICMP message starting with the ICMP Type. For computing the checksum, the checksum field should be zero.

Identifier
An identifier to aid in matching requests/replies; may be zero.

Sequence number
Sequence number to aid in matching requests/replies; may be zero.

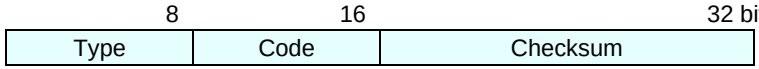
Address mask
A 32-bit mask.

ICMPv6

IETF RFC 1885 1970, 1995-12
http://www.cis.ohio-state.edu/htbin/rfc/rfc1885.html

The Internet Control Message Protocol (ICMP) was revised during the definition of IPv6. In addition, the multicast control functions of the IPv4 Group Membership Protocol (IGMP) are now incorporated with the ICMPv6.

The structure of the ICMPv6 header is shown in the following illustration.



ICMPv6 header structure

Type

The type of the message. Messages can be error or informational messages. Error messages can be Destination unreachable, Packet too big, Time exceed, Parameter problem. The possible informational messages are, Echo Request, Echo Reply, Group Membership Query, Group Membership Report, Group Membership Reduction.

Code

For each type of message several different codes are defined.
An example of this is the Destination Unreachable message, where possible messages are: no route to destination, communication with destination administratively prohibited, not a neighbor, address unreachable, port unreachable. For further details, refer to the standard.

Checksum

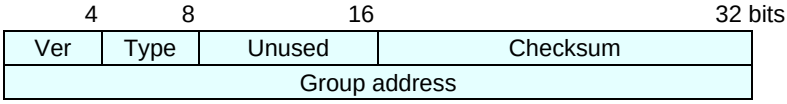
Used to check data corruption in the ICMPv6 message and parts of the IPv6 header.

IGMP

IETF RFC 1112 1989-08 http://www.cis.ohio-state.edu/htbin/rfc/rfc1885.html

The Internet Group Management Protocol (IGMP) is used by IP hosts to report their host group memberships to any immediately neighboring multicast routers. IGMP is a integral part of IP. It must be implemented by all hosts conforming to level 2 of the IP multicasting specification. IGMP messages are encapsulated in IP datagrams, with an IP protocol number of 2.

The format of the IGMP packet is shown in the following illustration:



IGMP packet structure

Version
The protocol version.

Type
The message type:
1 Host Membership Query.
2 Host Membership Report.

Unused
An unused field.

Checksum
The checksum.

Group address
In a Host Membership Report Message this field holds the IP host group address of the group being reported.