Chapter 2

TCP/IP Architecture

The four-level model illustrated in Figure 1.1. This model provides a reasonable pictorial representation of the layers in the TCP/IP protocol hierarchy.

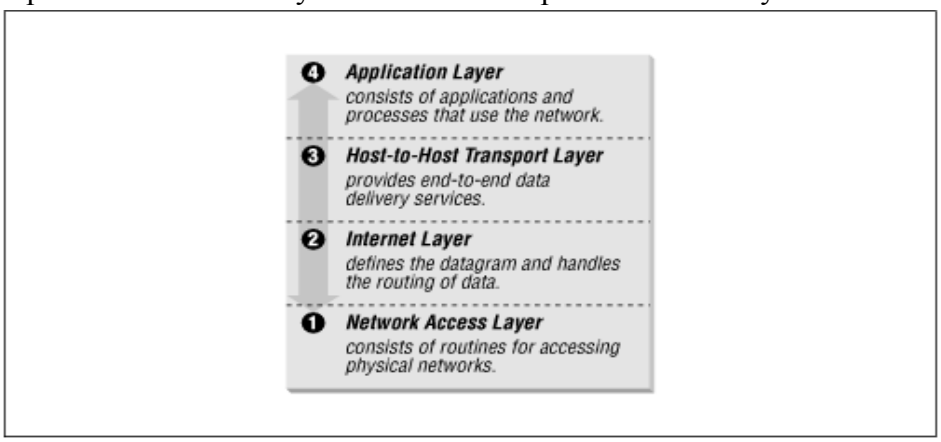


Figure 1.1: Layers in the TCP/IP protocol architecture

As in the OSI model, data is passed down the stack when it is being sent to the network, and up the stack when it is being received from the network. The four-layered structure of TCP/IP is seen in the way data is handled as it passes down the protocol stack from the Application Layer to the underlying physical network. Each layer in the stack adds control information to ensure proper delivery. This control information is called a header because it is placed in front of the data to be transmitted. Each layer treats all of the information it receives from the layer above as data and places its own header in front of that information. The addition of delivery information at every layer is called encapsulation . (See Figure 1.2 for an illustration of this.) When data is received, the opposite happens. Each layer strips off its header before passing the data on to the layer above. As information flows back up the stack, information received from a lower layer is interpreted as both a header and data.

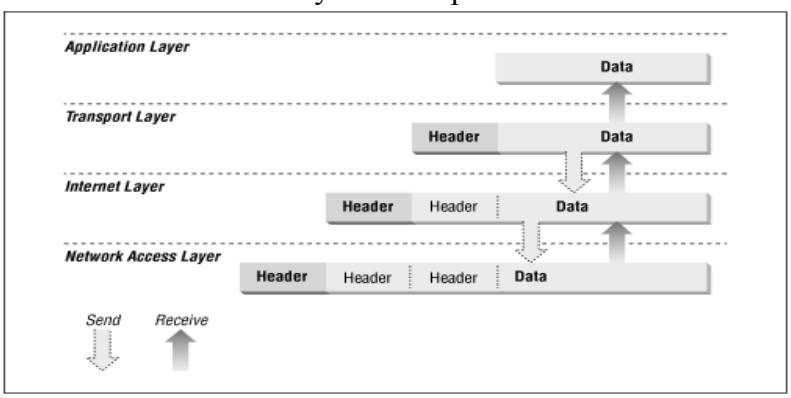


Figure 1.2: Data encapsulation

Each layer has its own independent data structures. Conceptually, a layer is unaware of the data structures used by the layers above and below it. In reality, the data structures of a layer are designed to be compatible with the structures used by the surrounding layers for the sake of more efficient data transmission. Still, each layer has its own data structure and its own terminology to describe that structure. Figure 1.4 shows the terms used by different layers of TCP/IP to refer to the data being transmitted. Applications using TCP refer to data as a

stream , while applications using the User Datagram Protocol (UDP) refer to data as a message . TCP calls data a segment , and UDP calls its data structure a packet . The Internet layer views all data as blocks called datagrams . TCP/IP uses many different types of underlying networks, each of which may have a different terminology for the data it transmits. Most networks refer to transmitted data as packets or frames . In Figure 1.3 we show a network that transmits pieces of data it calls frames .

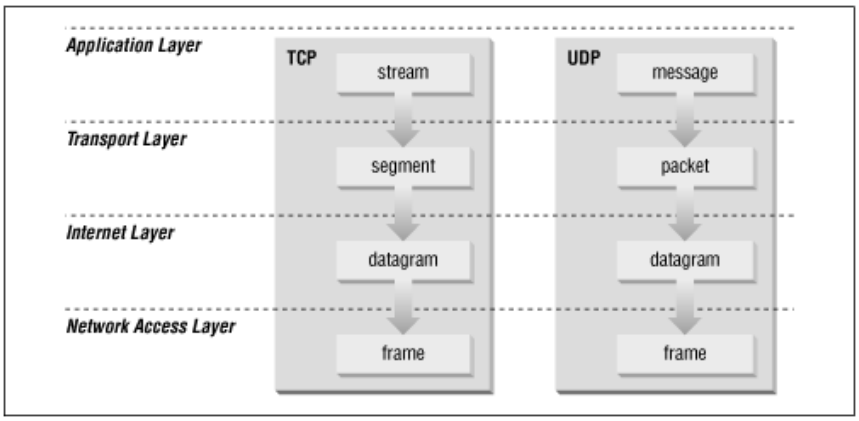


Figure 1.3: Data structures

TCP/IP and the DoD Model

DoD stands for Department of Defense. It is a smaller version of the OSI reference model. TCP/IP DoD model has four layers that are:

- Process/Application layer
- Host-to-Host layer
- Internet layer
- Network Access Layer

Process/Application layer

- The Application layer of the DoD model is equivalent to the upper three layers of the OSI model, i.e., Session layer, Presentation layer, and Application layer.
- The Process/Application layer of the DoD model provides the following capabilities –
- Enable applications to communicate with each other.
- Provides access to the services that operate at the lower layers of the DoD model.
- It contains a protocol that implements user-level functions such as mail delivery, file transfer, and remote login.

Host-to-Host layer

- A host-to-host layer of the DoD model performs the same functions as the Transport layer of the OSI reference model.
- It handles issues such as flow control, reliable end-to-end communication, and ensuring error-free delivery of the data.
- Protocols that operate on the Host-to-Host layer are: TCP and UDP.

Internet layer

- Internet layer of the DoD model performs the same functions as the Network layer of the OSI reference model.
- It handles the packaging, addressing, and routing of packets among multiple networks.
- This layer also establishes a connection between two computers to exchange the data.

Network Access Layer

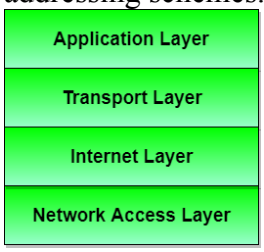
- The Network Access layer of the DoD model is equivalent to the lower two layers of the OSI model, i.e., Data link layer, and Physical layer.

- The Hardware connected to Network access layer are:
- Network medium: Cables like coaxial, twisted pair. Today, mostly, we use a wireless medium such as Bluetooth, WI-FI.
- Network Interface Card (NIC) has two types of addresses.

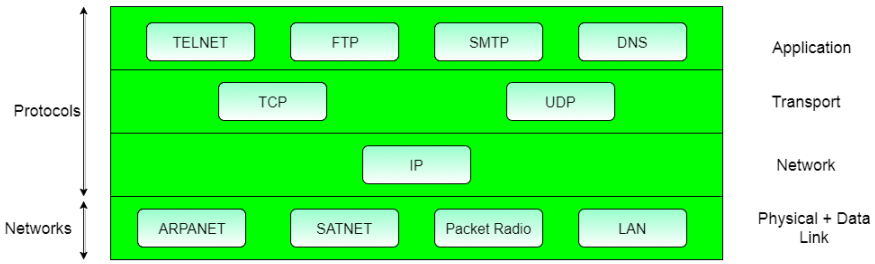
1. MAC Address- It is a 48 bits physical address.
2. IP Address – It is a 32 bits logical address.

Block diagram of TCP/IP Protocol Suite

TCP/IP means Transmission Control Protocol and Internet Protocol. It is the network model used in the current Internet architecture as well. Protocols are set of rules which govern every possible communication over a network. These protocols describe the movement of data between the source and destination or the internet. They also offer simple naming and addressing schemes.



Protocols and networks in the TCP/IP model:



Application layer

This is the top layer of TCP/IP protocol suite. This layer includes applications or processes that use transport layer protocols to deliver the data to destination computers.

At each layer there are certain protocol options to carry out the task designated to that particular layer. So, application layer also has various protocols that applications use to communicate with the second layer, the transport layer. Some of the popular application layer protocols are :

- HTTP (Hypertext transfer protocol)
- FTP (File transfer protocol)
- SMTP (Simple mail transfer protocol)
- SNMP (Simple network management protocol) etc

2. Transport Layer

This layer provides backbone to data flow between two hosts. This layer receives data from the application layer above it. There are many protocols that work at this layer but the two most commonly used protocols at transport layer are TCP and UDP.

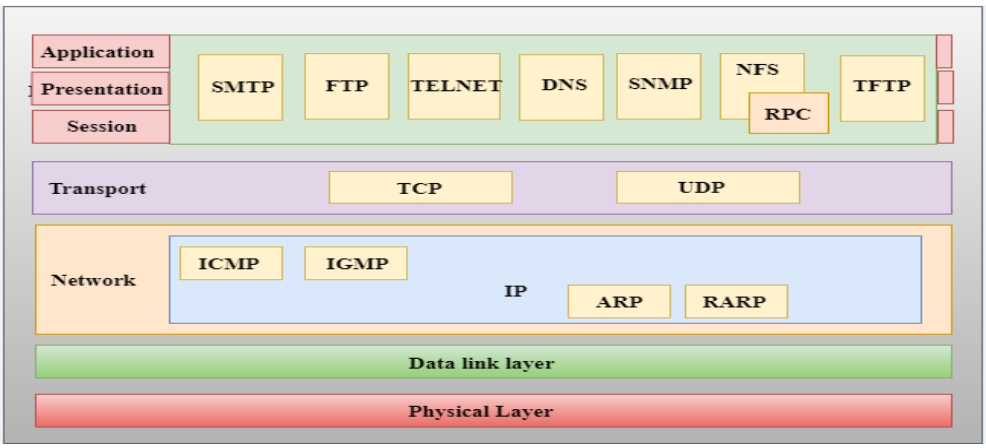
3. Network Layer

This layer is also known as Internet layer. The main purpose of this layer is to organize or handle the movement of data on network. By movement of data, we generally mean routing of data over the network. The main protocol used at this layer is IP. While ICMP(used by popular ‘ping’ command) and IGMP are also used at this layer.

4. Data Link Layer

This layer is also known as network interface layer. This layer normally consists of device drivers in the OS and the network interface card attached to the system. Both the device drivers and the network interface card take care of the communication details with the media being used to transfer the data over the network. In most of the cases, this media is in the form of cables. Some of the famous protocols that are used at this layer include ARP(Address resolution protocol), PPP(Point to point protocol) etc.

Different Protocols in TCP/IP



TCP:

TCP (Transmission Control Protocol) is a standard that defines how to establish and maintain a network conversation through which application programs can exchange data. TCP works with the Internet Protocol (IP), which defines how computers send packets of data to each other. Together, TCP and IP are the basic rules defining the Internet.

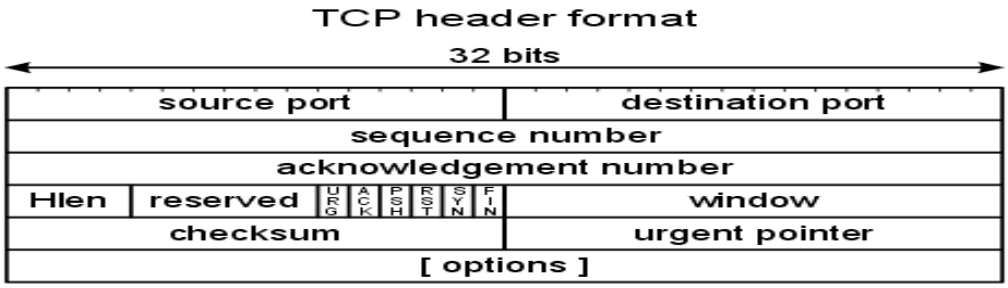
1. Connection oriented service

   Transmission Control Protocol (TCP) is a connection-oriented protocol. For connection-oriented communications, each end point must be able to transmit so that it can communicate. ... Because they can keep track of a conversation, connection-oriented protocols are sometimes described as stateful.
2. Flow Control

   Flow Control basically means that TCP will ensure that a sender is not overwhelming a receiver by sending packets faster than it can consume. ... Congestion control is about preventing a node from overwhelming the network (i.e. the links between two nodes), while Flow Control is about the end-node.
3. Multiplexing

   Gathering data from multiple application processes of sender, enveloping that data with header and sending them as a whole to the intended receiver is called as multiplexing. Delivering received segments at receiver side to the correct app layer processes is called as demultiplexing.

   TCP PROTOCOL HEADER FORMAT:



Source and Destination Port Number
Identification of the sending and receiving application. Along with the source and destination IP addresses in the IP - header identify the connection as a socket.
Sequence Number
The sequence number of the first data byte in this segment. If the SYN bit is set, the sequence number is the initial sequence number and the first data byte is initial sequence number + 1.
Acknowledgement Number
If the ACK bit is set, this field contains the value of the next sequence number the sender of the segment is expecting to receive. Once a connection is established this is always sent.
Hlen
The number of 32-bit words in the TCP header. This indicates where the data begins. The length of the TCP header is always a multiple of 32 bits.
Flags
There are six flags in the TCP header. One or more can be turned on at the same time.

URG	The URGENT POINTER field contains valid data
ACK	The ackowledgement number is valid
PSH	The receiver should pass this data to the application as soon as possible
RST	Reset the connection
SYN	Synchronize sequence numbers to initiate a connection.
FIN	The sender is finished sending data.

Window
This is the number of bytes, starting with the one specified by the acknowledgment number field, that the receiver is willing to accept. This is a 16-bit field, limiting the window to 65535 bytes.
Checksum
This covers both the header and the data. It is calculated by prepending a pseudo-header to the TCP segment, this consists of three 32 bit words which contain the source and destination IP addresses, a byte set to 0, a byte set to 6 (the protocol number for TCP in an IP datagram header) and the segment length (in words). The 16-bit one's complement sum of the header is calculated (i.e., the entire pseudo-header is considered a sequence of 16-bit words). The 16-bit one's complement of this sum is stored in the checksum field. This is a mandatory field that must be calculated and stored by the sender, and then verified by the receiver.
Urgent Pointer

The urgent pointer is valid only if the URG flag is set. This pointer is a positive offset that must be added to the sequence number field of the segment to yield the sequence number of the last byte of urgent data. TCP's urgent mode is a way for the sender to transmit emergency data to the other end. This feature is rarely used.

UDP-Connectionless service

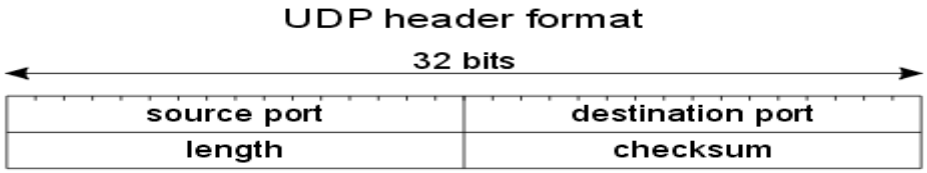
The message or datagram is sent without prior arrangement, which is less reliable but faster transaction than a connection-oriented service. User Datagram Protocol (UDP) is a connectionless protocol, while Transmission Control Protocol (TCP) is a connection-oriented network protocol.

User datagram protocol (UDP) operates on top of the Internet Protocol (IP) to transmit datagrams over a network. UDP does not require the source and destination to establish a three-way handshake before transmission takes place. Additionally, there is no need for an end-to-end connection.

Since UDP avoids the overhead associated with connections, error checks and the retransmission of missing data, it's suitable for real-time or high performance applications that don't require data verification or correction. If verification is needed, it can be performed at the application layer.

UDP Header Format

Each UDP message is called a user datagram. Conceptually, a user datagram consists of two parts: a UDP Header and a UDP data area. The header is divided in four 16-bit fields as shown:



Source and Destination Port

The port numbers identify the sending process and the receiving process. TCP and UDP use the destination port number to demultiplex incoming data from IP. Since IP has already demultiplexed the incoming IP datagram to either TCP or UDP (based on the protocol value in the IP header), this means the TCP port numbers are looked at by TCP, and the UDP port numbers by UDP. The TCP port numbers are independent of the UDP port numbers.

Length

The length in bytes of the UDP header and the encapsulated data. The minimum value for this field is 8.

Checksum

This is computed as 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 as needed with zero bytes at the end to make a multiple of two bytes. If the checksum is set to zero, then checksuming is disabled. The designers chose to make the checksum optional to allow implementations to operate with little computational overhead

ICMP

IP identifies ICMP messages contained within an IP datagram with protocol type 1. The first four bytes (1-byte type field, 1-byte code field, and 2-byte checksum) have the same format for all message types. All other fields and information contained within the ICMP header vary depending on the message type being sent.

Hosts and gateways use ICMP as a messaging, control, and diagnostic protocol to alert a host of problems or test connectivity. Note the protocol value contained within the IP header is 1, indicating this is an ICMP message. To determine the type of ICMP message, look in the ICMP header at the type field. Once you determine the type of ICMP message, you can use the code field to further identify the purpose of the message. The type field identifies the particular ICMP messages. Some ICMP messages use different values in the code field to further specify the error condition. The checksum field covers the entire ICMP message.

Codes

Many of the type fields contain more specific information about the error condition identified by a code value. ICMP messages have two types of codes:

- Query
- Error

Queries contain no additional information because they merely ask for information and will show a value of 0 in the code field. ICMP uses the following queries:

- Type 0 = Echo Reply
- Type 8 = Echo Request
- Type 9 = Router Advertisement
- Type 10 = Router Solicitation
- Type 13 = Timestamp Request
- Type 14 = Timestamp Reply
- Type 15 = Information Request (obsolete)
- Type 16 = Information Reply (obsolete)
- Type 17 = Address Mask Request
- Type 18 = Address Mask Reply

Error messages give specific information and will have varying values that further describe conditions. Error messages always include a copy of the offending IP header and up to 8 bytes of the data that caused the host or gateway to send the error message. The source host uses this information to identify and fix the problem reported via the ICMP error message. ICMP uses the following error messages:

- Type 3 = Destination Unreachable
- Type 4 = Source Quench
- Type 5 = Redirect
- Type 11 = Time Exceeded
- Type 12 = Parameter Problems

Checksum

The checksum verifies the validity of the ICMP header. The sending host performs the initial checksum calculation and places the results in this field. The receiving host performs the same calculations to assure that it does not receive data damaged in transit. If the checksum values do not match, it trashes the datagram.

Identifier

The user on the source host can set this optional value to match sent echo requests with received replies.

Sequence Number

The user on the source host can set this optional value to match sent echo requests with received replies.

BOOTP

The Bootstrap Protocol (BOOTP) is a computer networking protocol used in Internet Protocol networks to automatically assign an IP address to network devices from a configuration server. Bootstrap protocol was intended to allow computers to find what they need to function properly after booting up. BOOTP uses a relay agent, which allows packet forwarding from the local network using standard IP routing, allowing one BOOTP server to serve hosts on multiple subnets.

BOOTP requests and replies are encapsulated in UDP datagrams, as shown in Figure

Figure Encapsulation of BOOTP requests and replies within a UDP datagram.

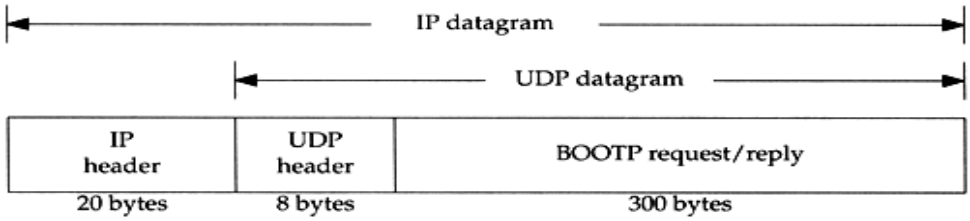
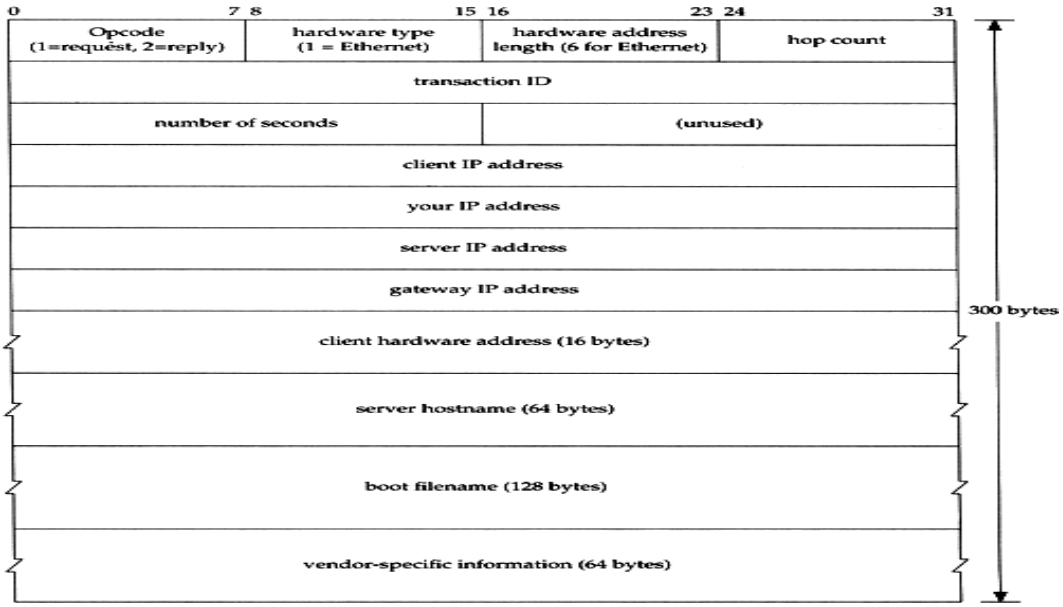


Figure shows the format of the 300-byte BOOTP request and reply.

Figure 16.2. Format of BOOTP request and reply.



Opcode is 1 for a request and 2 for a reply. The hardware type field is 1 for a 10 Mbits/sec Ethernet, the same value that is in the field of the same name in an ARP request or reply. Similarly, the hardware address length is 6 bytes for an Ethernet.

The hop count is set to 0 by the client, but can be used by a proxy server.

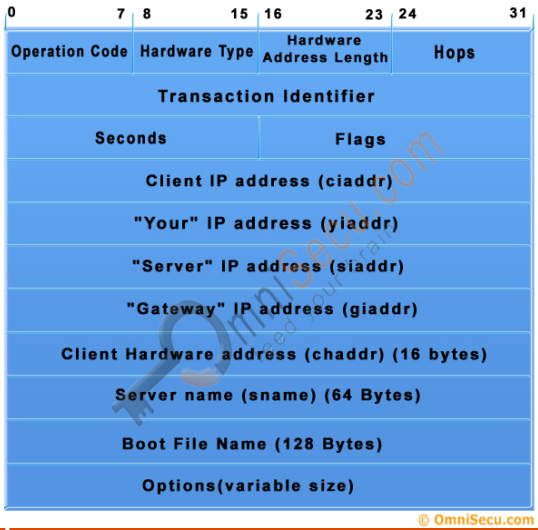
The transaction ID is a 32-bit integer set by the client and returned by the server. This lets the client match a response with a request. The client should set this to a random number for each request.

Number of seconds can be set by the client to the time since it started trying to bootstrap. The servers can look at this value, and perhaps a secondary server for a client won't respond until the number of seconds has exceeded some value, implying that the client's primary server is down.

If the client already knows its IP address, it fills in the client IP address. Otherwise, the client sets this to 0. In the latter case the server fills in your IP address with the client's IP address. The server IP address is filled in by the server. If a proxy server is used, that proxy server fills in its gateway IP address.
The client must set its client hardware address. Although this is the same value as in the Ethernet header, by placing the field in the UDP datagram also, it is easily available to any user process (e.g., a BOOTP server) that receives the datagram. It is normally much harder (or impossible) for a process reading UDP datagrams to determine the fields in the Ethernet header that carried the UDP datagram.
The server hostname is a null terminated string that is optionally filled in by the server. The server can also fill in the boot filename with the fully qualified, null terminated pathname of a file to bootstrap from.
The vendor-specific area is used for various extensions to BOOTP. Section 16.6 describes some of these extensions.
When a client is bootstrapping using BOOTP (an opcode of 1) the request is normally a link-layer broadcast and the destination IP address in the IP header is normally 255.255.255.255. The source IP address is often 0.0.0.0 since the client does not know its own IP address yet. Recall from Figure 3.9 that 0.0.0.0 is a valid source IP address when a system is bootstrapping itself.
Port Numbers
There are two well-known ports for BOOTP: 67 for the server and 68 for the client. This means the client does not choose an unused ephemeral port, but uses 68 instead. The reason two port numbers were chosen , instead of just one for the server, is that a server's reply can be (but normally isn't) broadcast.
If the server's reply were broadcast, and if the client were to choose an ephemeral port number, these broadcasts would also be received by other applications on other hosts that happen to be using the same ephemeral port number. Hence, it is considered bad form to broadcast to a random (i.e., ephemeral) port number.
If the client also used the server's well-known port (67) as its port, then all servers on the network are awakened to look at each broadcast reply. (If all the servers were awakened, they would examine the opcode, see that it's a reply and not a request, and go back to sleep.) Therefore the choice was made to have all clients use a single well-known port that differs from the server's well-known port.
If multiple clients are bootstrapping at the same time, and if the server broadcasts the replies, each client sees the replies intended for the other clients. The clients can use the transaction ID field in the BOOTP header to match replies with requests, or the client can examine the returned client hardware address.

DHCP

DHCP (Dynamic Host Configuration Protocol) is a protocol that provides quick, automatic, and central management for the distribution of IP addresses within a network. DHCP is also used to configure the subnet mask, default gateway, and DNS server information on the device.
Dynamic Host Configuration Protocol (DHCP) Message Format
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.



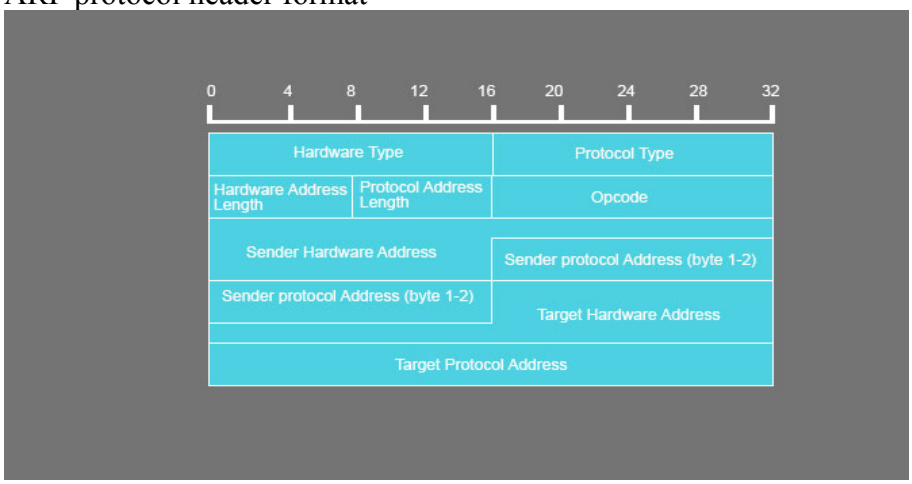
DHCP Message Field	Description
Operation Code	Specifies the type of the Dynamic Host Configuration Protocol (DHCP) message. Set to 1 in messages sent by a client (requests) and 2 in messages sent by a server (response).
Hardware Type	Specifies the network LAN architecture. For example, the ethernet type 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).

ARP

Address Resolution Protocol (ARP) is a procedure for mapping a dynamic Internet Protocol address (IP address) to a permanent physical machine address in a local area network (LAN). The physical machine address is also known as a Media Access Control or MAC address. The job of the ARP is essentially to translate 32-bit addresses to 48-bit addresses and vice-versa. This is necessary because in IP Version 4 (IPv4), the most common level of Internet Protocol (IP) in use today, an IP address is 32-bits long, but MAC addresses are 48-bits long.

ARP protocol header format



Hardware type (HTYPE)
: This field specifies the network link protocol type. Example: Ethernet is 1.

Protocol type (PTYPE)
: This field specifies the internetwork protocol for which the ARP request is intended. For IPv4, this has the value 0x0800. The permitted PTYPE values share a numbering space with those for EtherType.[3][4][5]

Hardware length (HLEN)
: Length (in octets) of a hardware address. Ethernet address length is 6.

Protocol length (PLEN)
: Length (in octets) of internetwork addresses. The internetwork protocol is specified in PTYPE. Example: IPv4 address length is 4.

Operation
: Specifies the operation that the sender is performing: 1 for request, 2 for reply.

Sender hardware address (SHA)

Media address of the sender. In an ARP request this field is used to indicate the address of the host sending the request. In an ARP reply this field is used to indicate the address of the host that the request was looking for.

Sender protocol address (SPA)

Internetwork address of the sender.

Target hardware address (THA)

Media address of the intended receiver. In an ARP request this field is ignored. In an ARP reply this field is used to indicate the address of the host that originated the ARP request.

Target protocol address (TPA)

Internetwork address of the intended receiver.

RARP

When a system with a local disk is bootstrapped it normally obtains its IP address from a configuration file that's read from a disk file. But a system without a disk, such as an X terminal or a diskless workstation, needs some other way to obtain its IP address.

Each system on a network has a unique hardware address, assigned by the manufacturer of the network interface. The principle of RARP is for the diskless system to read its unique hardware address from the interface card and send an RARP request (a broadcast frame on the network) asking for someone to reply with the diskless system's IP address (in an RARP reply).

The format of an RARP packet is almost identical to an ARP packet. The only differences are that the frame type is for an RARP request or reply, and the op field has a value of 3 for an RARP request and 4 for an RARP reply.

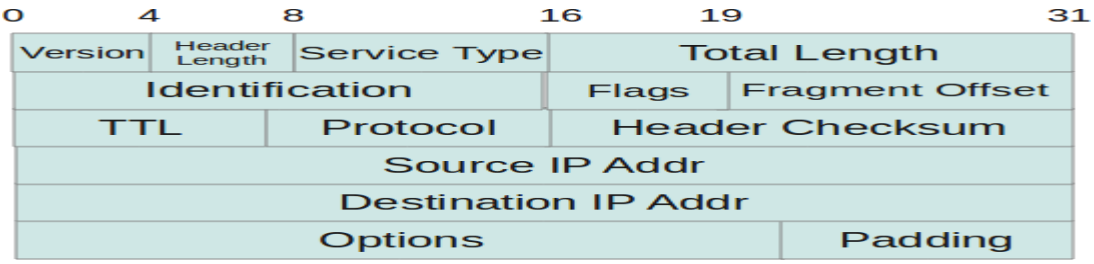
As with ARP, the RARP request is broadcast and the RARP reply is normally unicast.

IP

IP protocol is one of the main protocols in the TCP/IP stack.It is in the form of IP datagrams that all the TCP, UDP, ICMP and IGMP data travels over the network.

IP is connection less and unreliable protocol. It is connection less in the sense that no state related to IP datagrams is maintained either on source or destination side and it is unreliable in the sense that it not guaranteed that an IP data gram will get delivered to the destination or not.

If an IP datagram encounters some error at the destination or at some intermediate host (while traveling from source to destination) then the IP datagram is generally discarded and an ICMP error message is sent back to the source.



- Protocol Version(4 bits) : This is the first field in the protocol header. This field occupies 4 bits. This signifies the current IP protocol version being used. Most common version of IP protocol being used is version 4 while version 6 is out in market and fast gaining popularity.
- Header Length(4 bits) : This field provides the length of the IP header. The length of the header is represented in 32 bit words. This length also includes IP options (if any). Since this field is of 4 bits so the maximum header length allowed is 60 bytes. Usually when no

options are present then the value of this field is 5. Here 5 means five 32 bit words ie 5 *4 = 20 bytes.

- Type of service(8 bits) : The first three bits of this field are known as precedence bits and are ignored as of today. The next 4 bits represent type of service and the last bit is left unused. The 4 bits that represent TOS are : minimize delay, maximize throughput, maximize reliability and minimize monetary cost.
- Total length(16 bits): This represents the total IP datagram length in bytes. Since the header length (described above) gives the length of header and this field gives total length so the length of data and its starting point can easily be calculated using these two fields. Since this is a 16 bit field and it represents length of IP datagram so the maximum size of IP datagram can be 65535 bytes. When IP fragmentation takes place over the network then value of this field also changes. There are cases when IP datagrams are very small in length but some data links like ethernet pad these small frames to be of a minimum length ie 46 bytes. So to know the exact length of IP header in case of ethernet padding this field comes in handy.
- Identification(16 bits): This field is used for uniquely identifying the IP datagrams. This value is incremented every-time an IP datagram is sent from source to the destination. This field comes in handy while reassembly of fragmented IP data grams.
- Flags(3 bits): This field comprises of three bits. While the first bit is kept reserved as of now, the next two bits have their own importance. The second bit represents the 'Don't Fragment' bit. When this bit is set then IP datagram is never fragmented, rather its thrown away if a requirement for fragment arises. The third bit represents the 'More Fragment' bit. If this bit is set then it represents a fragmented IP datagram that has more fragments after it. In case of last fragment of an IP datagram this bit is not set signifying that this is the last fragment of a particular IP datagram.
- Fragment offset(13 bits): In case of fragmented IP data grams, this field contains the offset(in terms of 8 bytes units) from the start of IP datagram. So again, this field is used in reassembly of fragmented IP datagrams.
- Time to live(8 bits) : This value represents number of hops that the IP datagram will go through before being discarded. The value of this field in the beginning is set to be around 32 or 64 (lets say) but at every hop over the network this field is decremented by one. When this field becomes zero, the data gram is discarded. So, we see that this field literally means the effective lifetime for a datagram on network.
- Protocol(8 bits) : This field represents the transport layer protocol that handed over data to IP layer. This field comes in handy when the data is demultiplex-ed at the destination as in that case IP would need to know which protocol to hand over the data to.
- Header Checksum(16 bits) : This fields represents a value that is calculated using an algorithm covering all the fields in header (assuming this very field to be zero). This value is calculated and stored in header when IP data gram is sent from source to destination and at the destination side this checksum is again calculated and verified against the checksum present in header. If the value is same then the datagram was not corrupted else its assumed that data gram was received corrupted. So this field is used to check the integrity of an IP datagram.
- Source and destination IP(32 bits each) : These fields store the source and destination address respectively. Since size of these fields is 32 bits each so an IP address os maximum length of 32 bits can be used. Options(Variable length) : This field represents a list of options that are active for a particular IP datagram. This is an optional field that could be or could not be present. If any option is present in the header then the first byte is represented as follows.