14. TCP/IP Protocol Suite

Contents

a. TCP/IP – Internet – OSI

b. Network level – IP protocol

c. Addressing and sub-networks

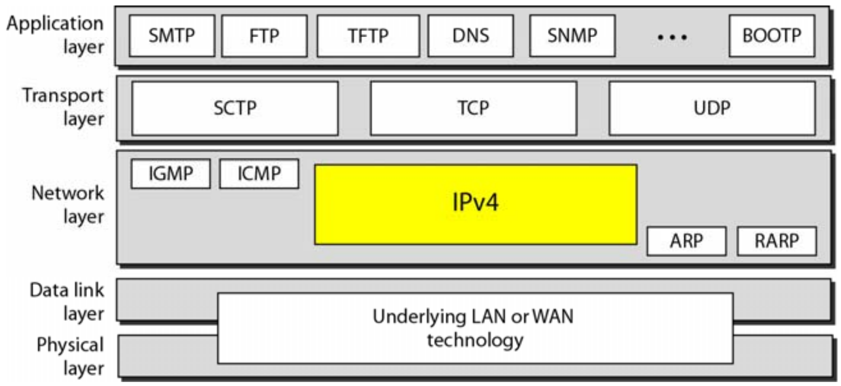
d. Other network-level protocols

e. Transport level

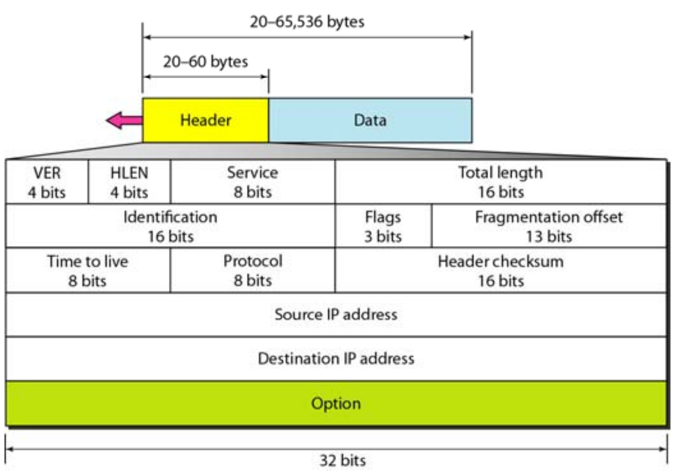
a. TCP/IP – Internet – OSI

b. Network level – IP protocol

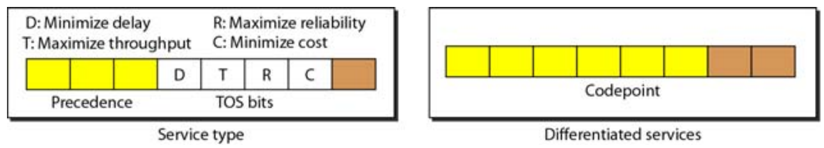
Position of IPv4 in TCP/IP protocol suite



IPv4 datagram format



Service type or differentiated services



The precedence subfield was part of version 4, but never used.

Types of service

TOS Bits	*Description*
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

Default types of service

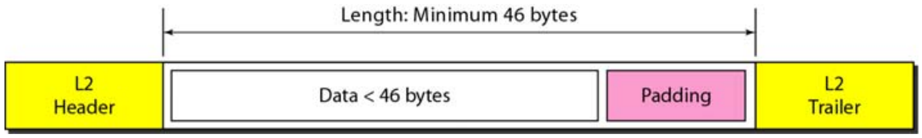
Protocol	*TOS Bits*	*Description*
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

Values for codepoints

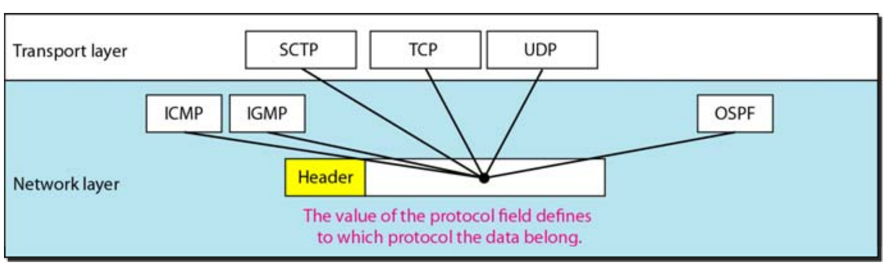
Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

The total length field defines the total length of the datagram including the header.

Encapsulation of a small datagram in an Ethernet frame



Protocol field and encapsulated data



Protocol values

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

Example 1

An IPv4 packet has arrived with the first 8 bits as shown:

01000010

The receiver discards the packet. Why?

Solution

There is an error in this packet.

The 4 leftmost bits (0100) show the version, which is correct.

The next 4 bits (0010) show an invalid header length (2 × 4 = 8). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

Example 2

In an IPv4 packet, the value of HLEN is 1000 in binary. How many bytes of options are being carried by this packet?

Solution

The HLEN value is 8, which means the total number of bytes in the header is 8 × 4, or 32 bytes.

The first 20 bytes are the base header, the next 12 bytes are the options.

Example 3

In an IPv4 packet, the value of HLEN is 5, and the value of the total length field is 0x0028. How many bytes of data are being carried by this packet?

Solution

The HLEN value is 5, which means the total number of bytes in the header is 5 × 4, or 20 bytes (no options)

The total length is 40 bytes, which means the packet is carrying 20 bytes of data (40 − 20).

Example 4

An IPv4 packet has arrived with the first few hexadecimal digits as shown.

0x45000028000100000102 . . .

How many hops can this packet travel before being dropped? The data belong to what upper-layer protocol?

Solution

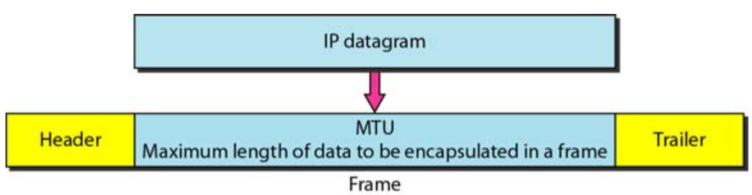
To find the time-to-live field, we skip 8 bytes.

The time-to-live field is the ninth byte, which is 01.

This means the packet can travel only one hop.

The protocol field is the next byte (02), which means that the upper-layer protocol is IGMP.

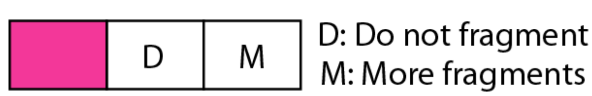
Maximum transfer unit (MTU)



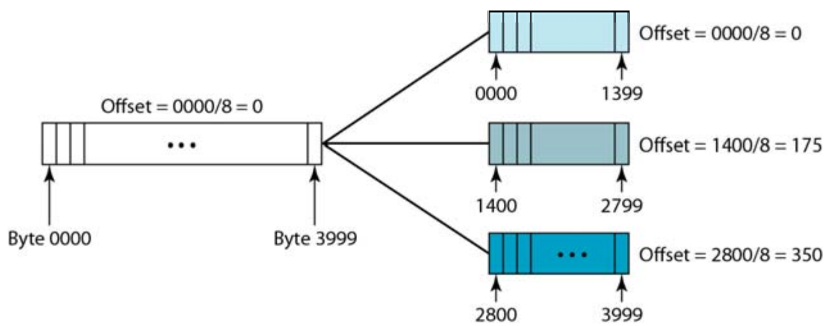
MTUs for some networks

Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

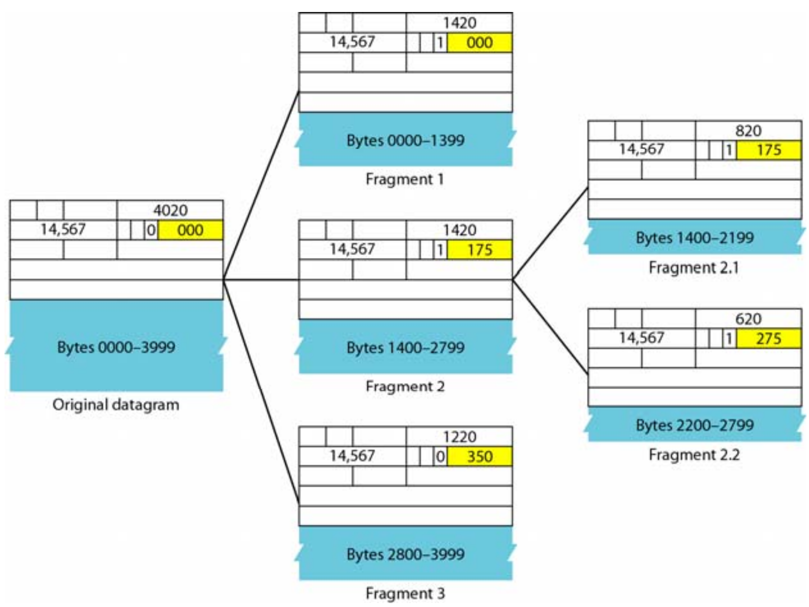
Flags used in fragmentation



Fragmentation example



Detailed fragmentation example



Example 5

A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 0, it means that there are no more fragments; the fragment is the last one.

However, we cannot say if the original packet was fragmented or not.

A non-fragmented packet is considered the last fragment.

Example 6

A packet has arrived with an M bit value of 1. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 1, it means that there is at least one more fragment.

This fragment can be the first one or a middle one, but not the last one.

We don't know if it is the first one or a middle one; we need more information (the value of the fragmentation offset).

Example 7

A packet has arrived with an M bit value of 1 and a fragmentation offset value of 0. Is this the first fragment, the last fragment, or a middle fragment?

Solution

Because the M bit is 1, it is either the first fragment or a middle one.

Because the offset value is 0, it is the first fragment.

Example 8

A packet has arrived in which the offset value is 100. What is the number of the first byte? Do we know the number of the last byte?

Solution

To find the number of the first byte, we multiply the offset value by 8.

This means that the first byte number is 800.

We cannot determine the number of the last byte unless we know the length.

Example 9

A packet has arrived in which the offset value is 100, the value of HLEN is 5, and the value of the total length field is 100. What are the numbers of the first byte and the last byte?

Solution

The first byte number is 100 × 8 = 800.

The total length is 100 bytes, and the header length is 20 bytes (5 × 4), which means that there are 80 bytes in this datagram.

If the first byte number is 800, the last byte number must be 879.

c. Addressing and sub-networks

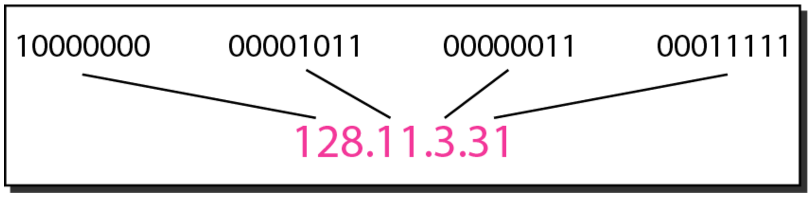
Note

An IPv4 address is 32 bits long.

Note

The IPv4 addresses are unique and universal.

Dotted-decimal notation and binary notation for an IPv4 address



Numbering systems are reviewed in Appendix B.

Example 1

Change the following IPv4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

Solution

We replace each group of 8 bits with its equivalent decimal number (see Appendix B) and add dots for separation.

a. 129.11.11.239

b. 193.131.27.255

Example 2

Change the following IPv4 addresses from dotted-decimal notation to binary notation.

a. 111.56.45.78

b. 221.34.7.82

Solution

We replace each decimal number with its binary equivalent

a. 01101111 00111000 00101101 01001110

b. 11011101 00100010 00000111 01010010

Example 3

Find the error, if any, in the following IPv4 addresses.

a. 111.56.045.78
b. 221.34.7.8.20
c. 75.45.301.14
d. 11100010.23.14.67

Solution

a. There must be no leading zero (045).

b. There can be no more than four numbers.

c. Each number needs to be less than or equal to 255.

d. A mixture of binary notation and dotted-decimal notation is not allowed.

In classful addressing, the address space is divided into five classes: A, B, C, D, and E.

Finding the classes in binary and dotted-decimal notation

	First byte	Second byte	Third byte	Fourth byte
Class A	0			
Class B	10			
Class C	110			
Class D	1110			
Class E	1111			

a. Binary notation

	First byte	Second byte	Third byte	Fourth byte
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–239			
Class E	240–255			

b. Dotted-decimal notation

Example 4

Find the class of each address.

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

c. 14.23.120.8

d. 252.5.15.111

Solution

a. The first bit is 0. This is a class A address.

b. The first 2 bits are 1; the third bit is 0. This is a class C address.

c. The first byte is 14; the class is A.

d. The first byte is 252; the class is E.

Number of blocks and block size in classful IPv4 addressing

Class	*Number of Blocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

In classful addressing, a large part of the available addresses were wasted.

Default masks for classful addressing

Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	**11111111** 00000000 00000000 00000000	**255**.0.0.0	/8
B	**11111111 11111111** 00000000 00000000	**255.255**.0.0	/16
C	**11111111 11111111 11111111** 00000000	**255.255.255**.0	/24

Classful addressing, which is almost obsolete, is replaced with classless addressing.

Example 5

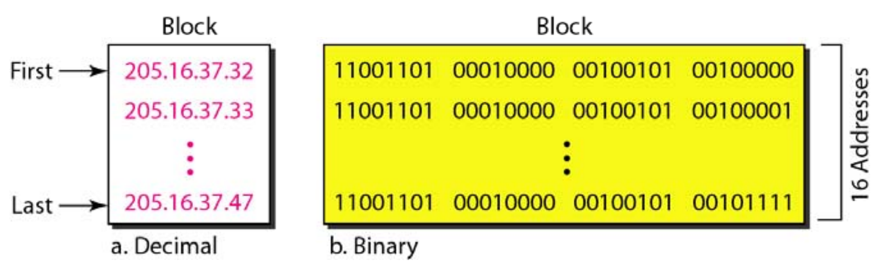
The following figure shows a block of addresses, in both binary and dotted-decimal notation, granted to a small business that needs 16 addresses.

We can see that the restrictions are applied to this block. The addresses are contiguous.

The number of addresses is a power of 2 ($16 = 2^4$), and the first address is divisible by 16.

The first address, when converted to a decimal number, is 3,440,387,360, which when divided by 16 results in 215,024,210.

A block of 16 addresses granted to a small organization



In IPv4 addressing, a block of addresses can be defined as
x.y.z.t /*n*
in which x.y.z.t defines one of the addresses and the /*n* defines the mask.

The first address in the block can be found by setting the rightmost $32 - n$ bits to 0s.

Example 6

A block of addresses is granted to a small organization. We know that one of the addresses is 205.16.37.39/28. What is the first address in the block?

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

If we set 32−28 rightmost bits to 0, we get

11001101 00010000 00100101 0010000

or

205.16.37.32.

This is actually the block shown in figure.

The last address in the block can be found by setting the rightmost 32 − n bits to 1s.

Example 7

Find the last address for the block in Example 19.6.

Solution

The binary representation of the given address is

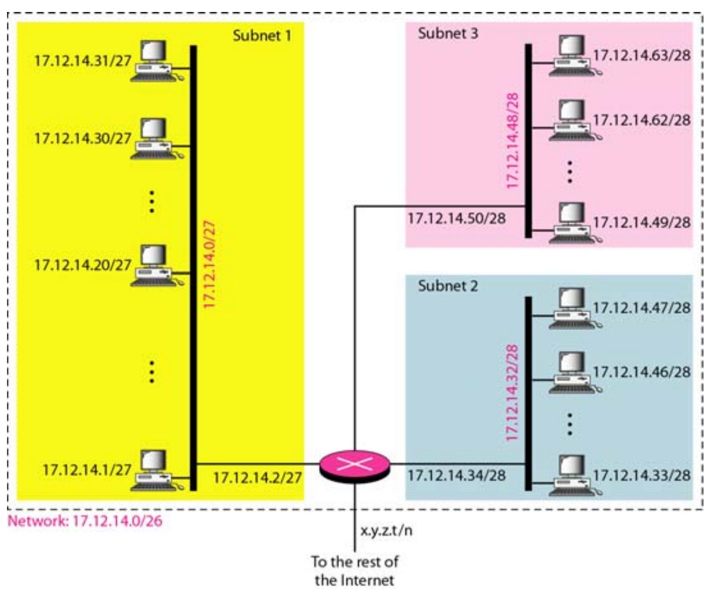
11001101 00010000 00100101 00100111

If we set 32 − 28 rightmost bits to 1, we get

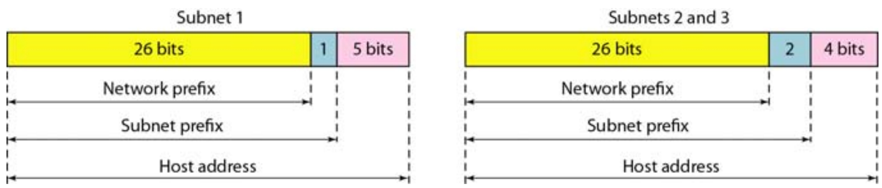
11001101 00010000 00100101 00101111
or
205.16.37.47

This is actually the block shown in figure.

Configuration and addresses in a subnetted network



Three-level hierarchy in an IPv4 address



Example

An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to 3 groups of customers as follows:

a. The first group has 64 customers; each needs 256 addresses.
b. The second group has 128 customers; each needs 128 addresses.
c. The third group has 128 customers; each needs 64 addresses.

Design the subblocks and find out how many addresses are still available after these allocations.

Example

Solution

Figure shows the situation.

Group 1

For this group, each customer needs 256 addresses. This means that 8 (log2 256) bits are needed to define each host. The prefix length is then 32 − 8 = 24. The addresses are

1st Customer:	*190.100.0.0/24*	*190.100.0.255/24*
2nd Customer:	*190.100.1.0/24*	*190.100.1.255/24*
. . .		
64th Customer:	*190.100.63.0/24*	*190.100.63.255/24*
Total = 64 × 256 = 16,384		

Example

Group 2

For this group, each customer needs 128 addresses. This means that 7 (log2 128) bits are needed to define each host. The prefix length is then 32 − 7 = 25. The addresses are

1st Customer:	*190.100.64.0/25*	*190.100.64.127/25*
2nd Customer:	*190.100.64.128/25*	*190.100.64.255/25*
. . .		
128th Customer:	*190.100.127.128/25*	*190.100.127.255/25*
Total = 128 × 128 = 16,384		

Example

Group 3

For this group, each customer needs 64 addresses. This means that 6 (log264) bits are needed to each host. The prefix length is then 32 − 6 = 26. The addresses are:

1st Customer:	*190.100.128.0/26*	*190.100.128.63/26*
2nd Customer:	*190.100.128.64/26*	*190.100.128.127/26*
. . .		
128th Customer:	*190.100.159.192/26*	*190.100.159.255/26*
Total = 128 × 64 = 8192		

Number of granted addresses to the ISP: 65,536
Number of allocated addresses by the ISP: 40,960
Number of available addresses: 24,576

An example of address allocation and distribution by an ISP

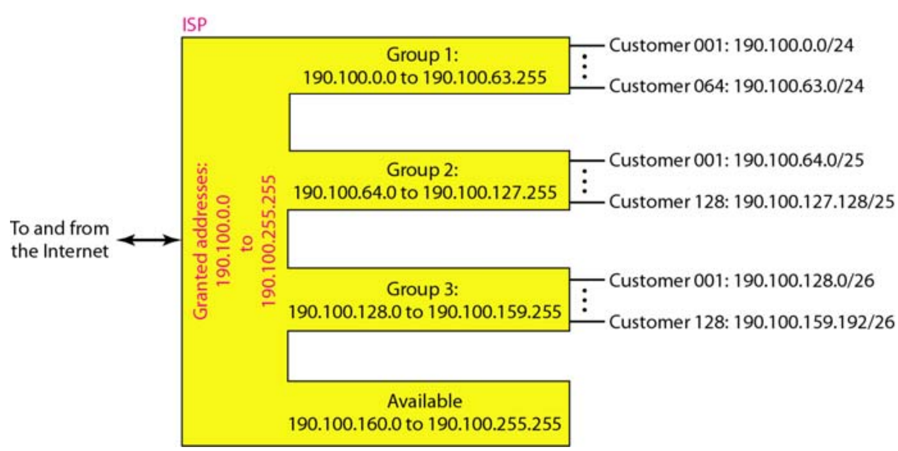
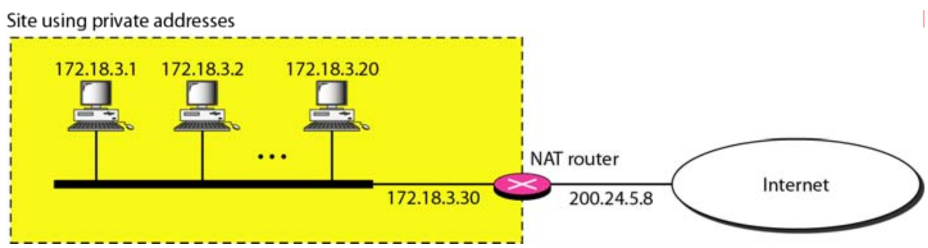


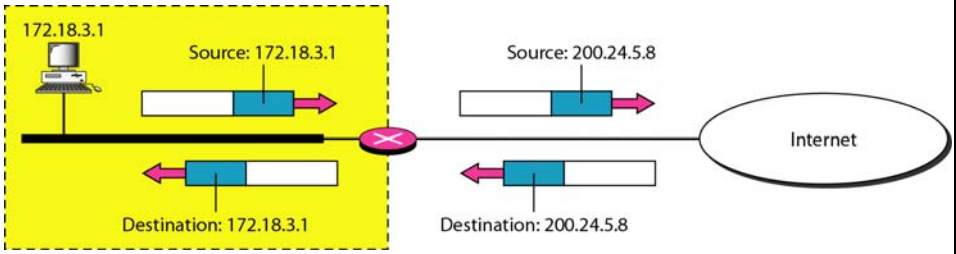
Table. Addresses for private networks

Range			*Total*
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

A NAT implementation



Addresses in a NAT



NAT address translation

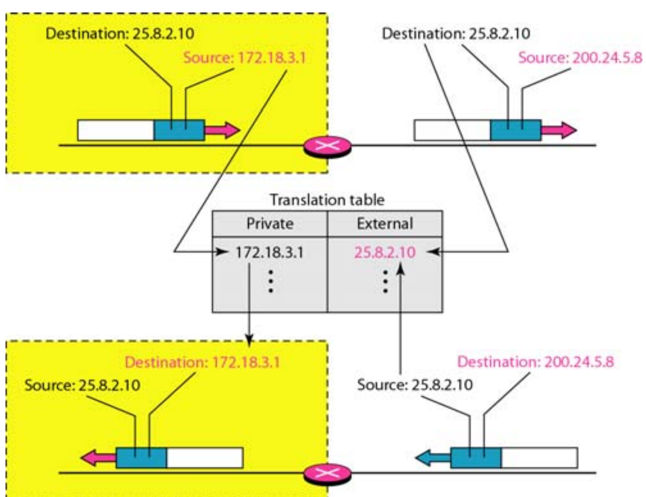
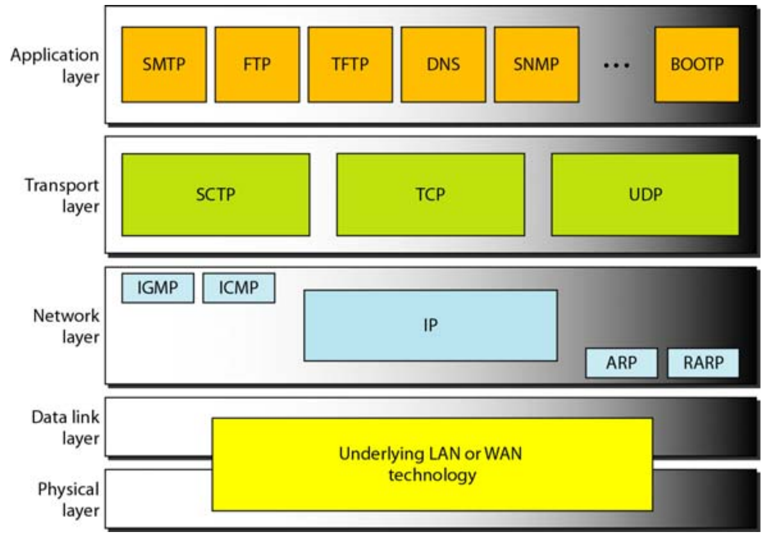


Table. Five-column translation table

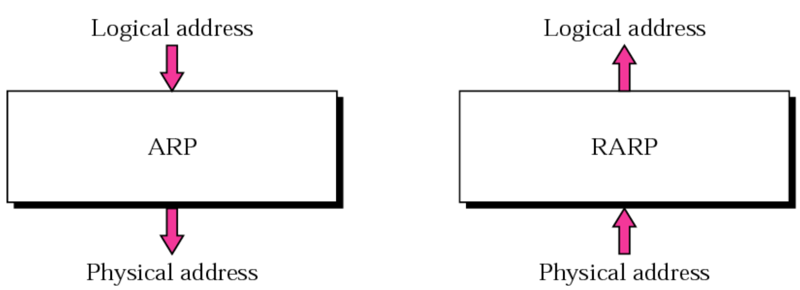
Private Address	*Private Port*	*External Address*	*External Port*	*Transport Protocol*
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
. . .	. . .	. . .	. . .	. . .

d. Other network-level protocols

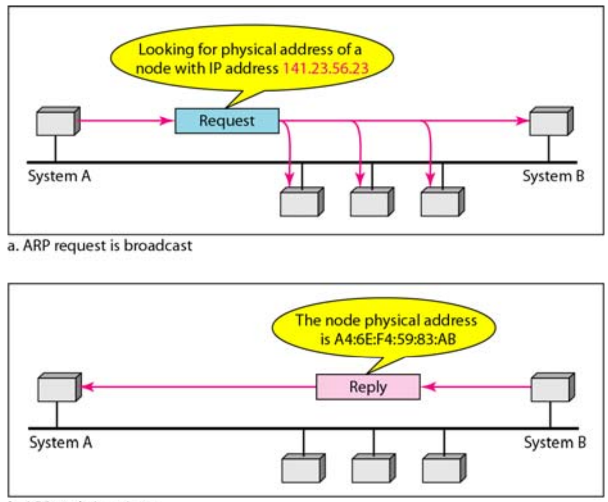
Position of UDP, TCP, and SCTP in TCP/IP suite



ARP and RARP



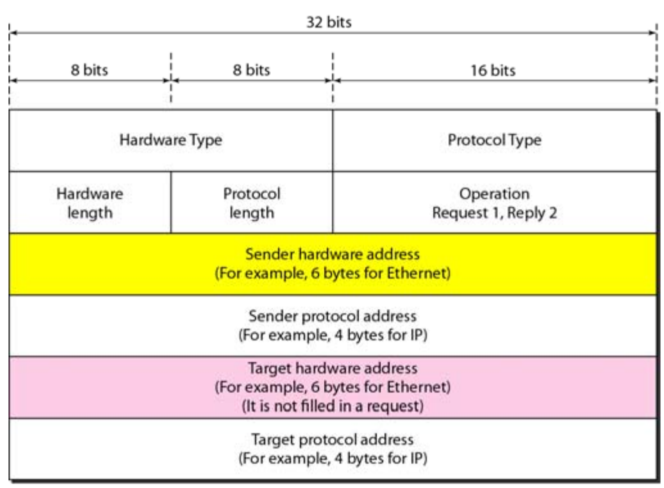
ARP operation



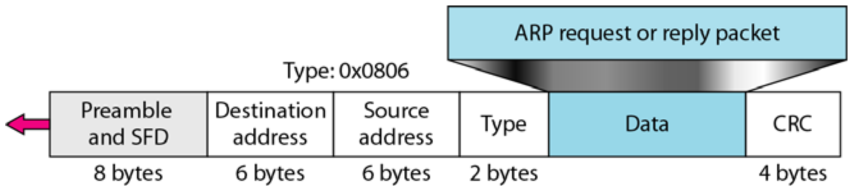
a. ARP request is broadcast

b. ARP reply is unicast

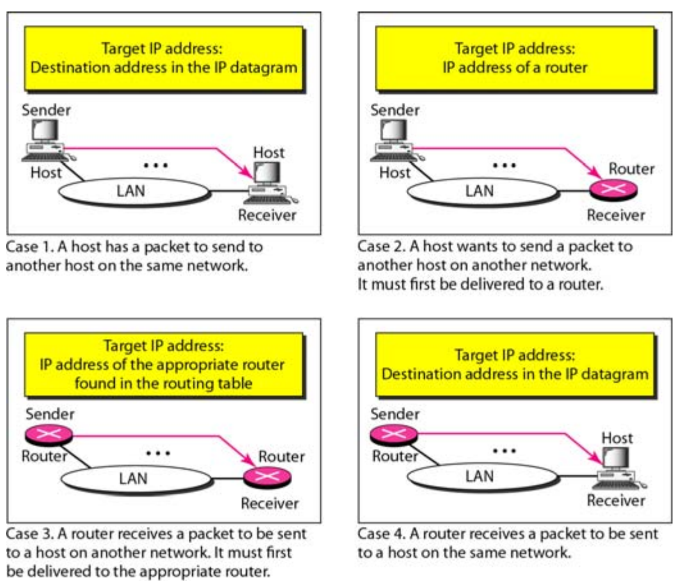
ARP packet



Encapsulation of ARP packet



Four cases using ARP



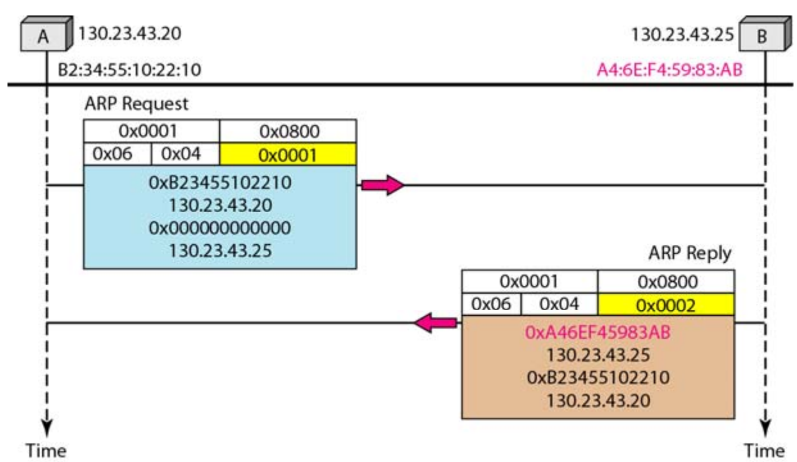
Example

A host with IP address 130.23.43.20 and physical address B2:34:55:10:22:10 has a packet to send to another host with IP address 130.23.43.25 and physical address A4:6E:F4:59:83:AB. The two hosts are on the same Ethernet network. Show the ARP request and reply packets encapsulated in Ethernet frames.

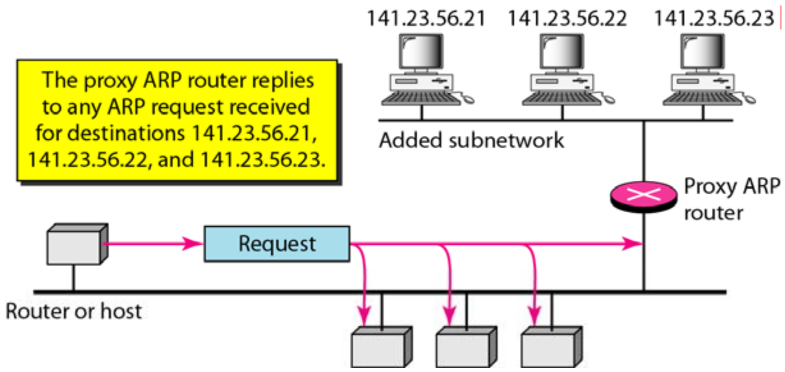
Solution

The following figure shows the ARP request and reply packets. Note that the ARP data field in this case is 28 bytes, and that the individual addresses do not fit in the 4-byte boundary. That is why we do not show the regular 4-byte boundaries for these addresses.

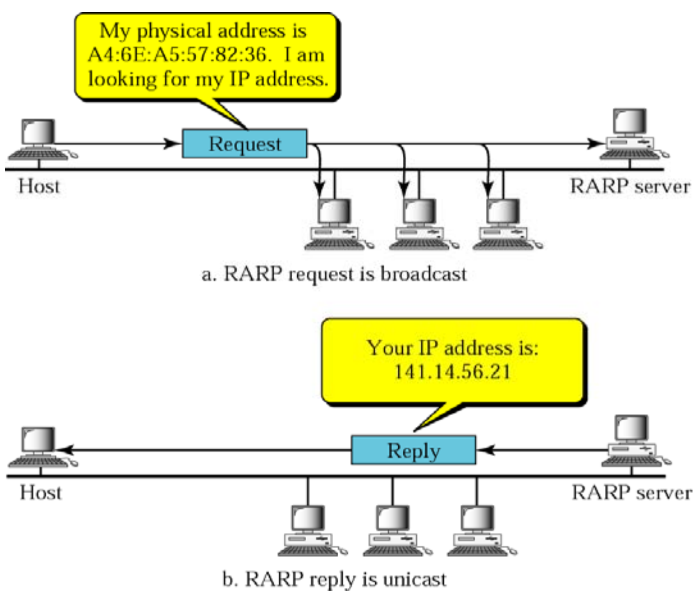
Example. An ARP request and reply



Proxy ARP



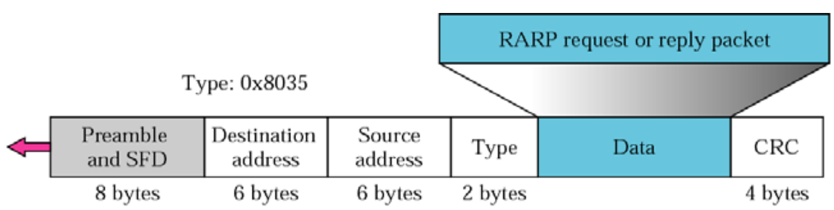
RARP operation



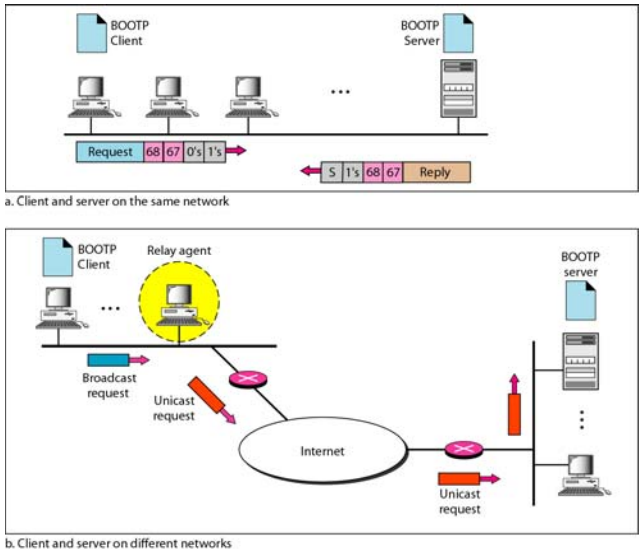
RARP packet

Hardware type		Protocol type
Hardware length	Protocol length	Operation Request 3, Reply 4
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP) (It is not filled for request)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled for request)		
Target protocol address (For example, 4 bytes for IP) (It is not filled for request)		

Encapsulation of RARP packet



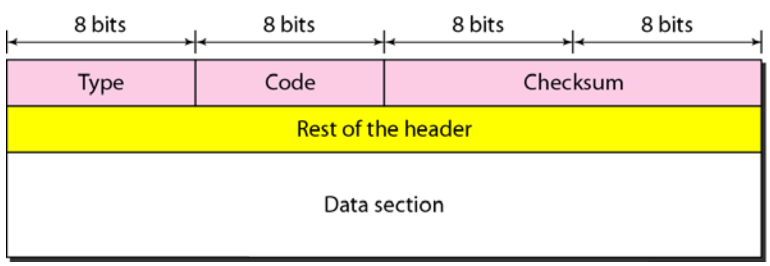
BOOTP client and server on the same and different networks



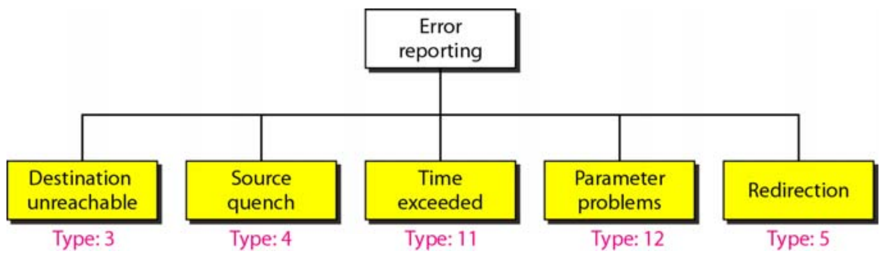
a. Client and server on the same network

b. Client and server on different networks

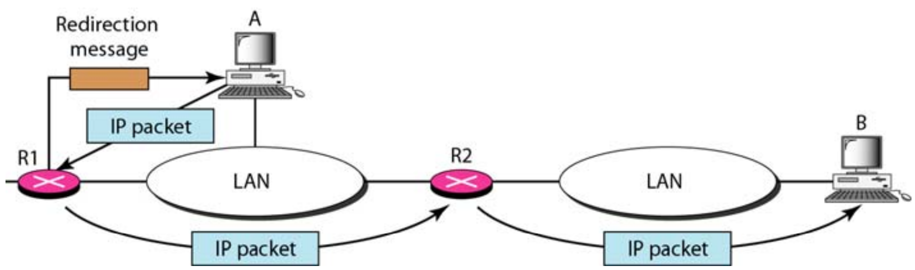
General format of ICMP messages



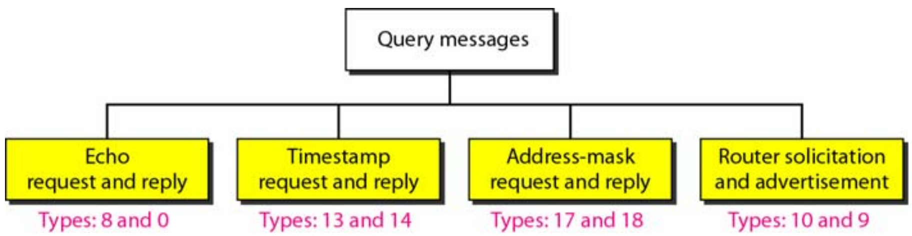
Error-reporting messages



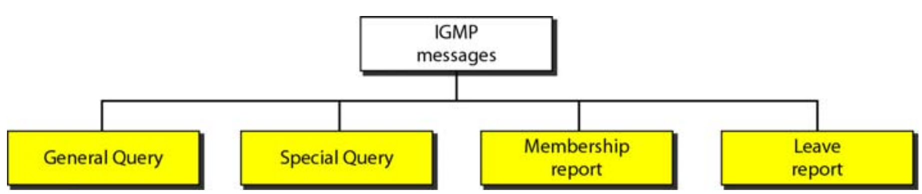
Redirection concept



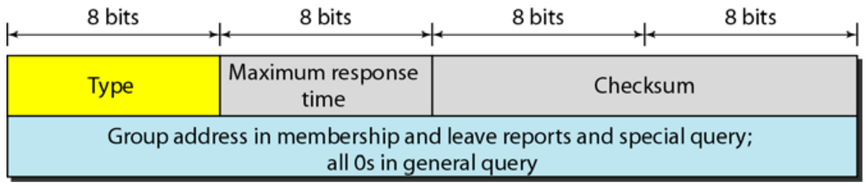
Query messages



IGMP message types



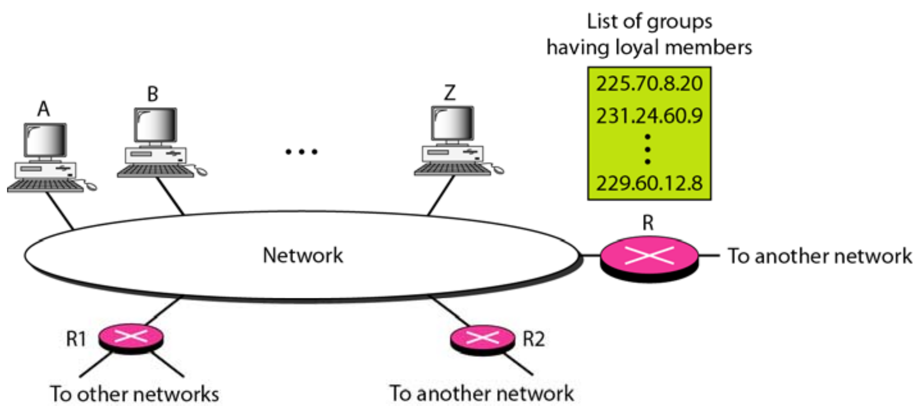
IGMP message format



IGMP type field

Type	*Value*
General or special query	0x11 or 00010001
Membership report	0x16 or 00010110
Leave report	0x17 or 00010111

IGMP operation



Example

Imagine there are three hosts in a network, as shown in the following figure.

A query message was received at time 0; the random delay time (in tenths of seconds) for each group is shown next to the group address.

Show the sequence of report messages.

Solution

The events occur in this sequence:

a. Time 12: The timer for 228.42.0.0 in host A expires, and a membership report is sent, which is received by the router and every host including host B which cancels its timer for 228.42.0.0.

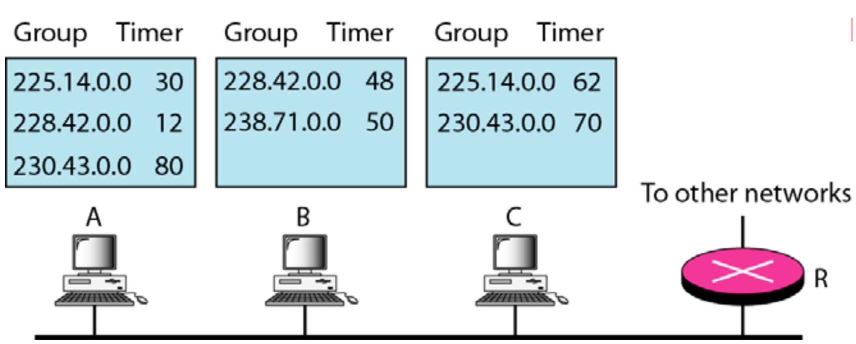
Example

b. Time 30: The timer for 225.14.0.0 in host A expires, and a membership report is sent which is received by the router and every host including host C which cancels its timer for 225.14.0.0.

c. Time 50: The timer for 238.71.0.0 in host B expires, and a membership report is sent, which is received by the router and every host.

d. Time 70: The timer for 230.43.0.0 in host C expires, and a membership report is sent, which is received by the router and every host including host A which cancels its timer for 230.43.0.0.

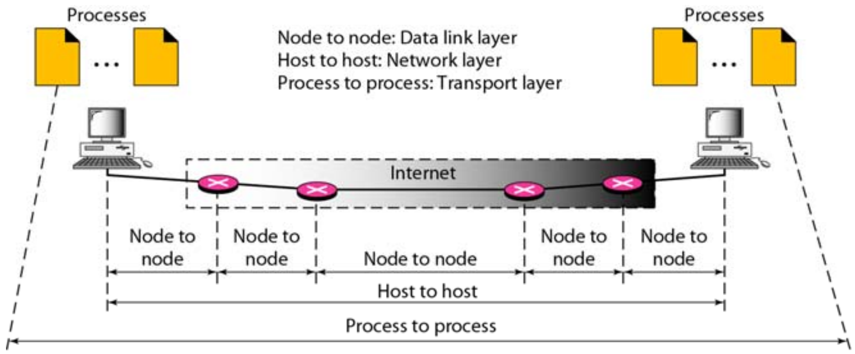
Example



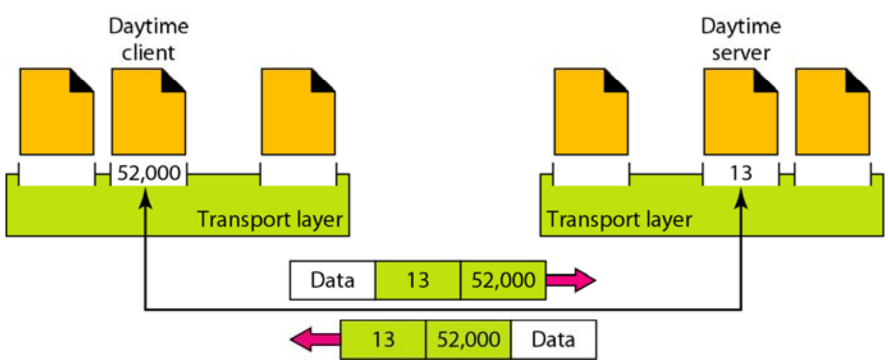
e. Transport level

The transport layer is responsible for process-to-process delivery.

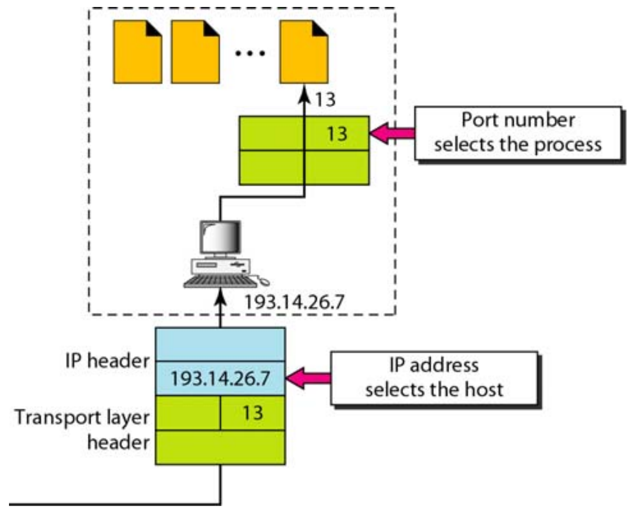
Types of data deliveries



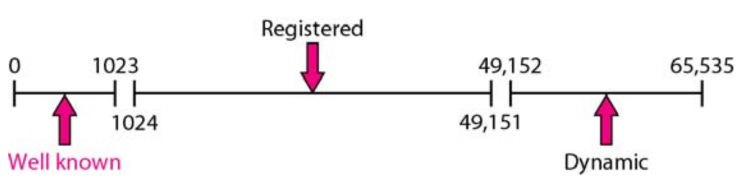
Port numbers



IP addresses versus port numbers

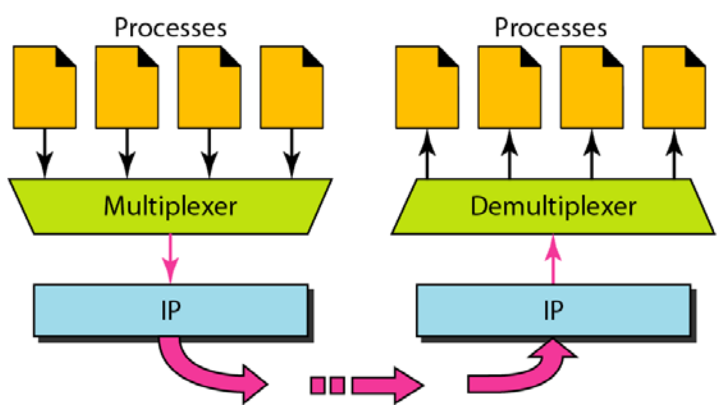


IANA ranges

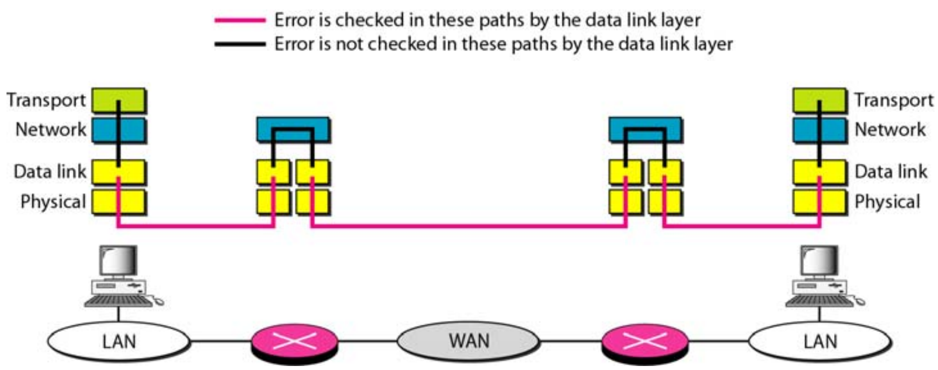


Socket address

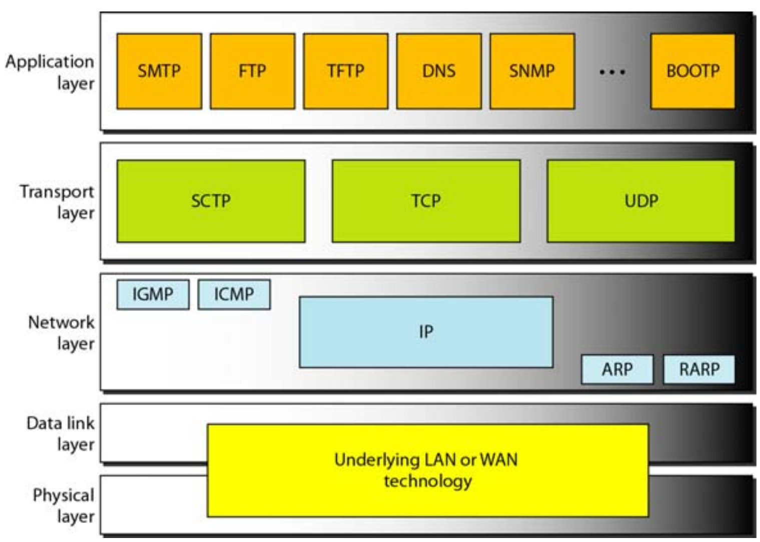
Multiplexing and demultiplexing



Error control



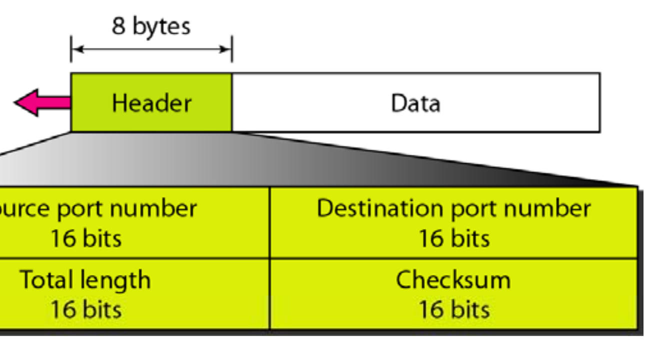
Position of UDP, TCP, and SCTP in TCP/IP suite



Well-known ports used with UDP

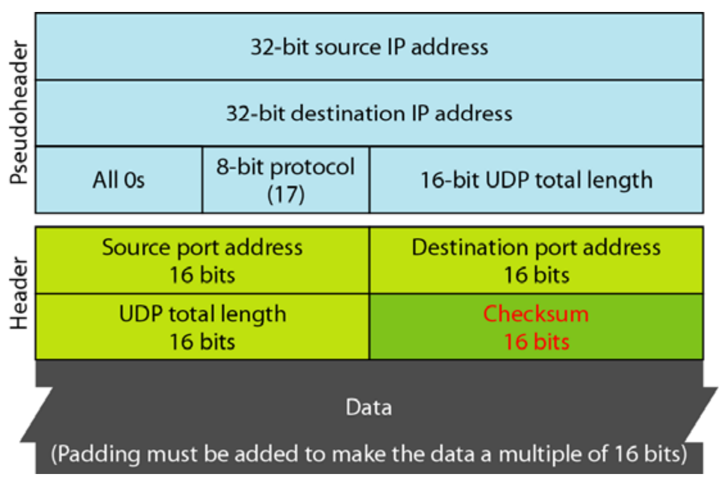
Port	Protocol	Description
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Nameserver	Domain Name Service
67	BOOTPs	Server port to download bootstrap information
68	BOOTPc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

User datagram format



UDP length
= IP length – IP header's length

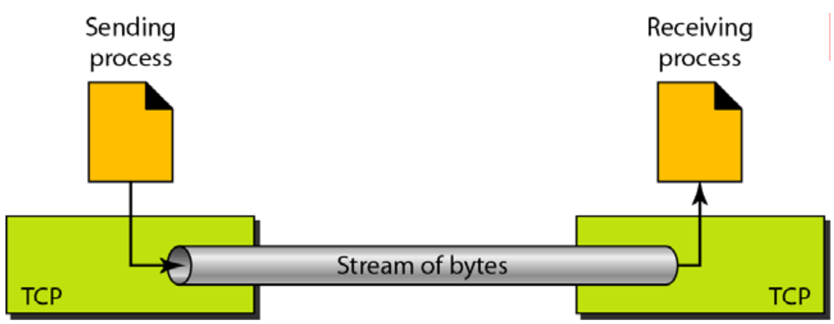
Pseudoheader for checksum calculation



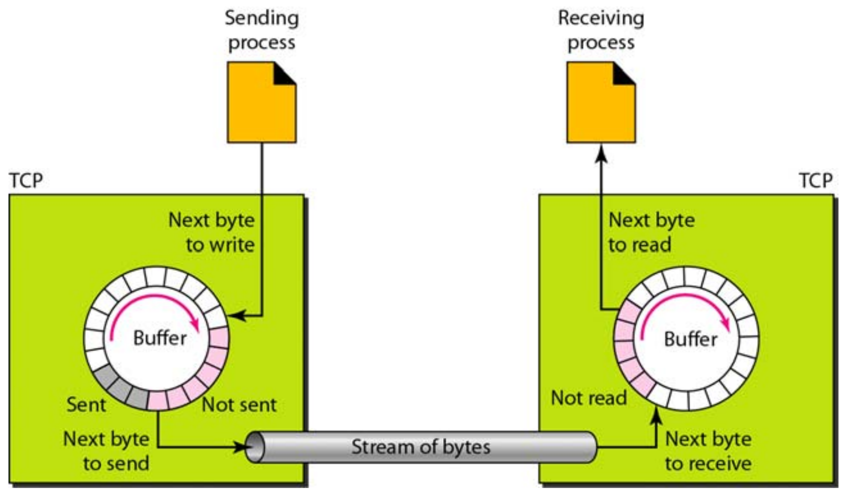
Well-known ports used by TCP

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call

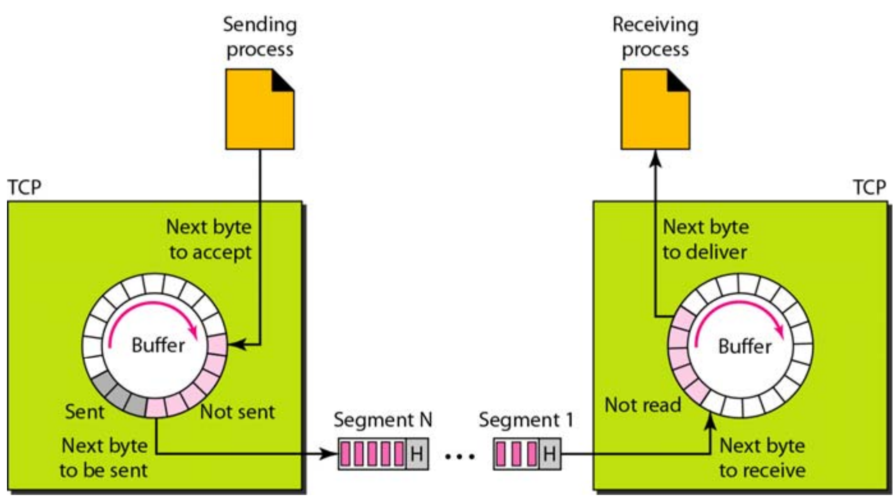
Stream delivery



Sending and receiving buffers



TCP segments



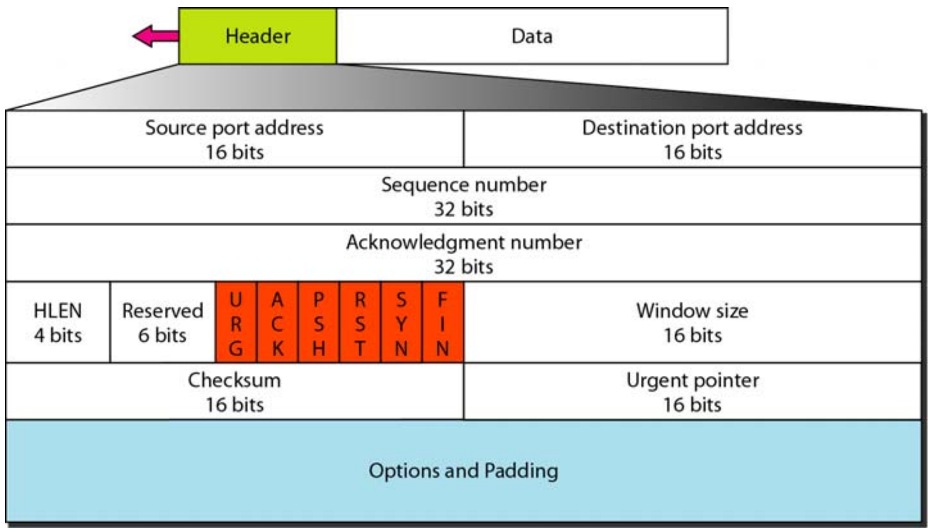
The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with a randomly generated number.

Example

The following shows the sequence number for each segment:

Segment 1	→	Sequence Number: 10,001 (range: 10,001 to 11,000)
Segment 2	→	Sequence Number: 11,001 (range: 11,001 to 12,000)
Segment 3	→	Sequence Number: 12,001 (range: 12,001 to 13,000)
Segment 4	→	Sequence Number: 13,001 (range: 13,001 to 14,000)
Segment 5	→	Sequence Number: 14,001 (range: 14,001 to 15,000)

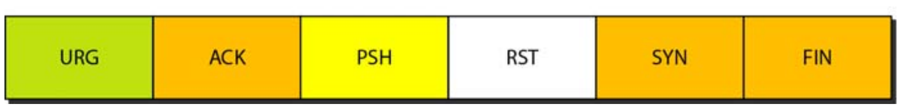
TCP segment format



Control field

URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push

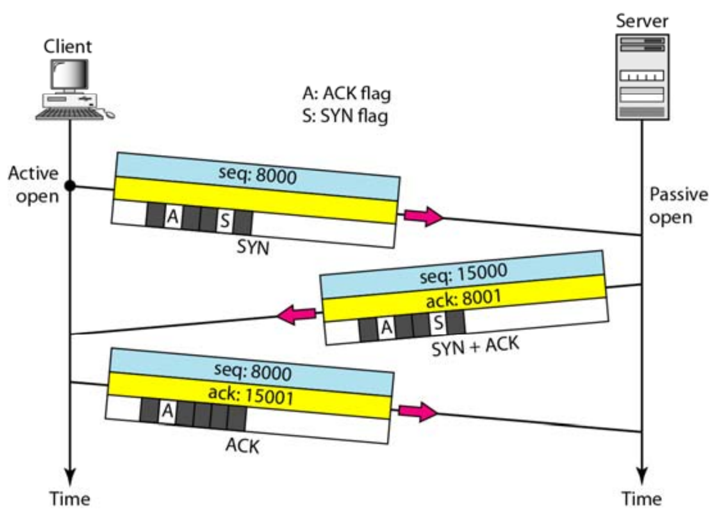
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection



Description of flags in the control field

Flag	*Description*
URG	The value of the urgent pointer field is valid.
ACK	The value of the acknowledgment field is valid.
PSH	Push the data.
RST	Reset the connection.
SYN	Synchronize sequence numbers during connection.
FIN	Terminate the connection.

Connection establishment using three-way handshaking

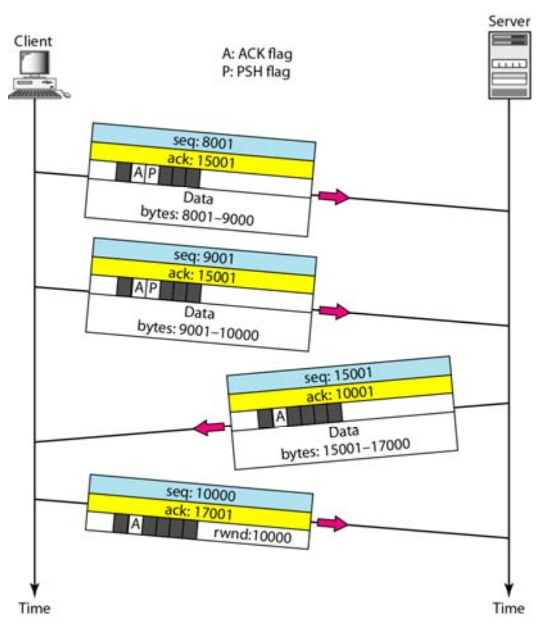


A SYN segment cannot carry data, but it consumes one sequence number.

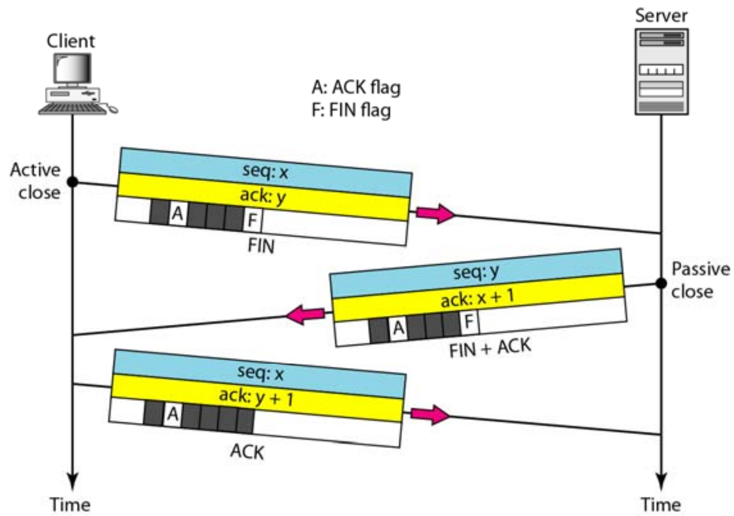
A SYN + ACK segment cannot carry data, but does consume one sequence number.

An ACK segment, if carrying no data, consumes no sequence number.

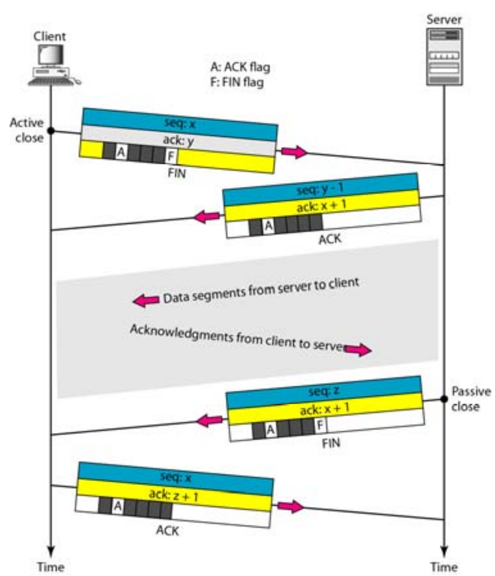
Data transfer



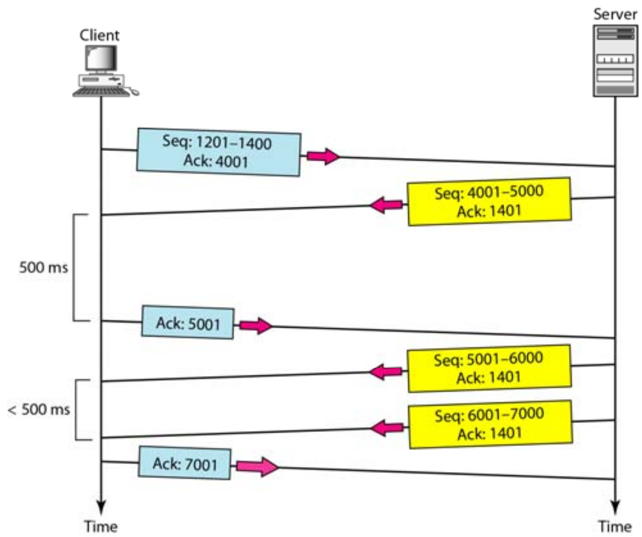
Connection termination using three-way handshaking



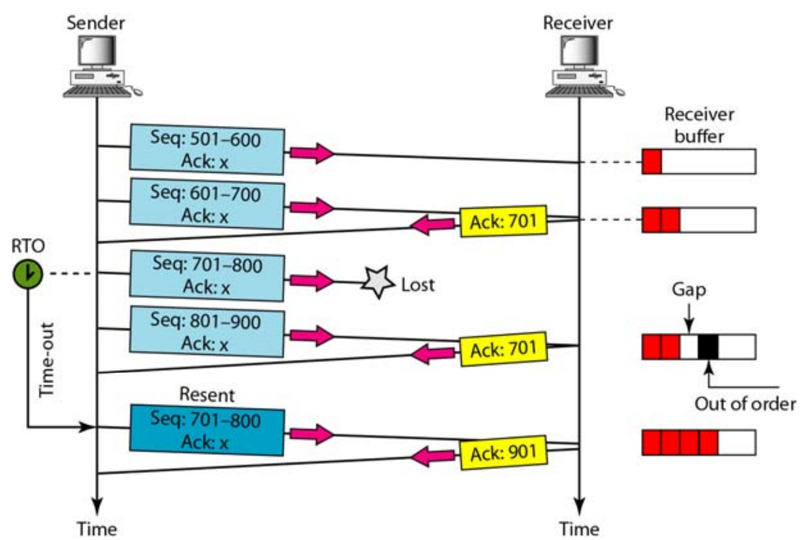
Half-close



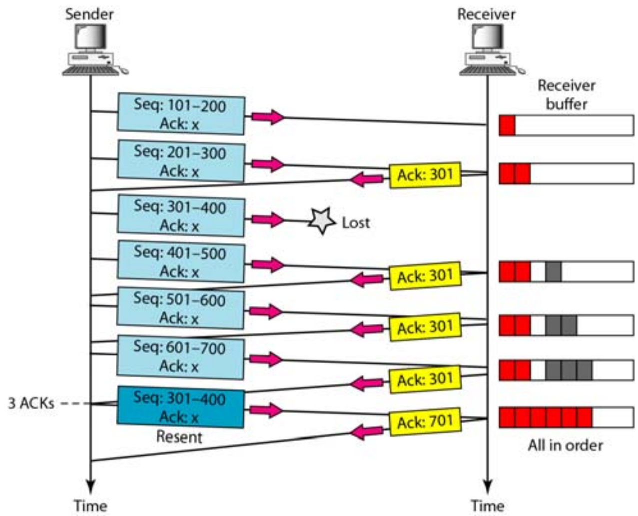
Normal operation



Lost segment



Fast retransmission



SCTP

Stream Control Transmission Protocol (SCTP) is a new reliable, message-oriented transport layer protocol.

SCTP, however, is mostly designed for Internet applications that have recently been introduced.

These new applications need a more sophisticated service than TCP can provide.

SCTP is a message-oriented, reliable protocol that combines the best features of UDP and TCP.

Some SCTP applications

Protocol	*Port Number*	*Description*
IUA	9990	ISDN over IP
M2UA	2904	SS7 telephony signaling
M3UA	2905	SS7 telephony signaling
H.248	2945	Media gateway control
H.323	1718, 1719, 1720, 11720	IP telephony
SIP	5060	IP telephony