

## MODULE 4

### Network Layer Protocols and Unicast Routing Network Layer Protocols

**In this chapter, we show how the network layer is implemented in the TCP/IP protocol suite. The protocols in the network layer have gone through a few versions; in this chapter, we concentrate on the current version v4.**

Communication at the network layer is host-to-host (computer-to-computer); a computer somewhere in the world needs to communicate with another computer somewhere else in the world through the Internet.

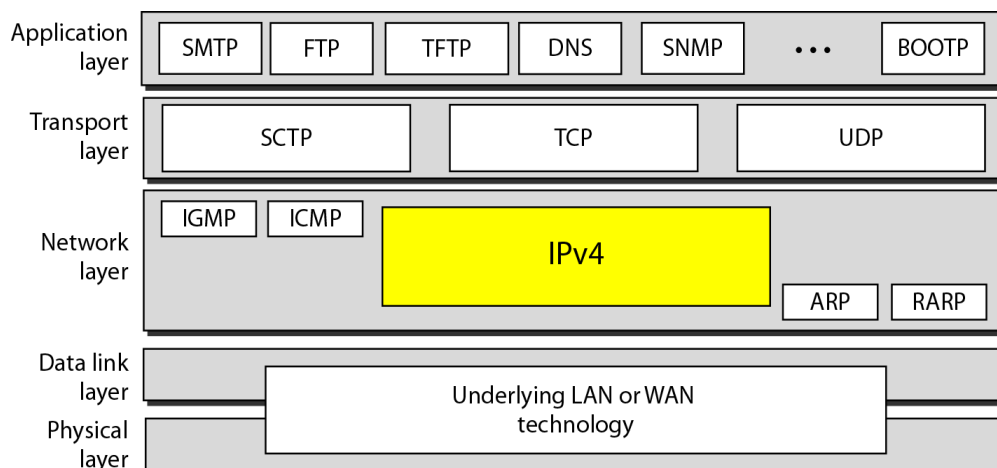
The packet transmitted by the sending computer may pass through several LANs or WANs before reaching the destination computer. A global addressing scheme called logical addressing is required for this communication. The term IP address refers to the logical address in the network layer of the TCP/IP protocol suite.

**Communication at the network layer in the Internet is connectionless. If reliability is important, IPv4 must be paired with a reliable transport-layer protocol such as TCP.**

#### 4.1 Position of IPv4 and other network protocols in TCP/IP protocol suite

The network layer in version 4 can be thought of as one main protocol and three auxiliary ones as shown in Figure 4.1.

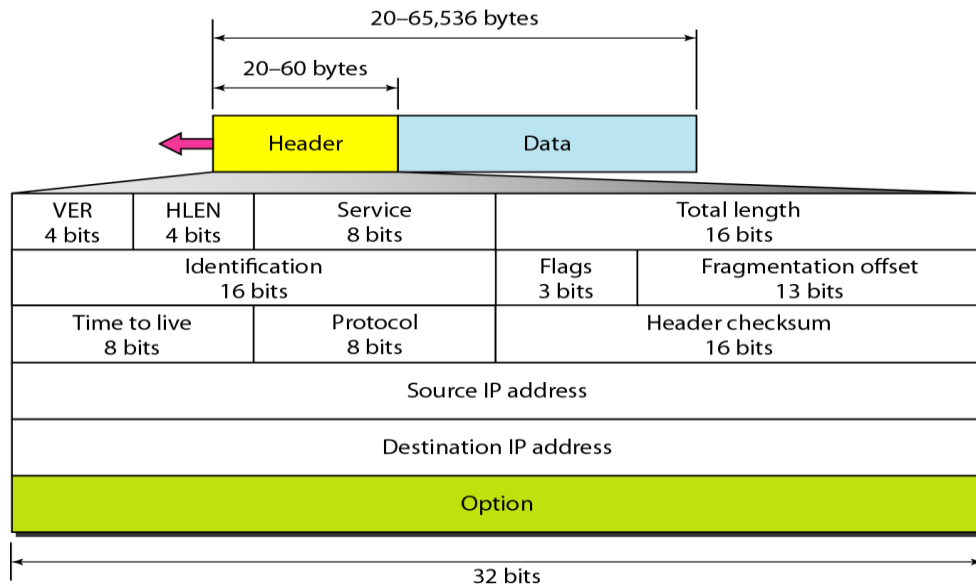
- The main protocol, Internet Protocol version 4 (IPv4), is responsible for packetizing, forwarding, and delivery of a packet at the network layer.
- The Internet Control Message Protocol version 4 (ICMPv4) helps IPv4 to handle some errors that may occur in the network-layer delivery.
- The Internet Group Management Protocol (IGMP) is used to help IPv4 in multicasting.
- The Address Resolution Protocol (ARP) is used to glue the network and data-link layers in mapping network-layer addresses to link-layer addresses



**Figure 4.1: Position of IP and other network-layer protocols in TCP/IP protocol suite**

#### Datagram Format

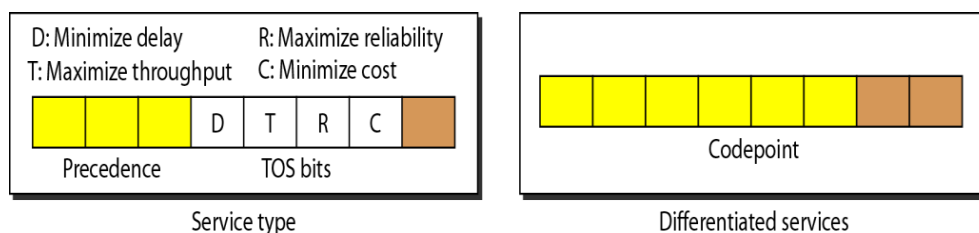
The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols. Packets used by the IP are called datagrams. A datagram is a variable-length packet consisting of two parts: header and payload (data). The header is 20 to 60 bytes in length and contains information essential to routing and delivery. It is customary in TCP/IP to show the header in 4-byte sections.



**Figure 4.2: IPv4 Datagram**

A brief description of each field is in order:

1. **Version Number:** The 4-bit version number (VER) field defines the version of the IPv4 protocol, which, obviously, has the value of 4.
2. **Header Length:** The 4-bit header length (HLEN) field defines the total length of the datagram header in 4-byte words. The IPv4 datagram has a variable-length header. When a device receives a datagram, it needs to know when the header stops and the data, which is encapsulated in the packet, starts. The total length is divided by 4 and the value is inserted in the field. The receiver needs to multiply the value of this field by 4 to find the total length.
3. **Service Type:** In the original design of the IP header shown in Fig 4.3, this field was referred to as type of service (TOS), which defined how the datagram should be handled. In the late 1990s, IETF redefined the field to provide differentiated services (DiffServ).



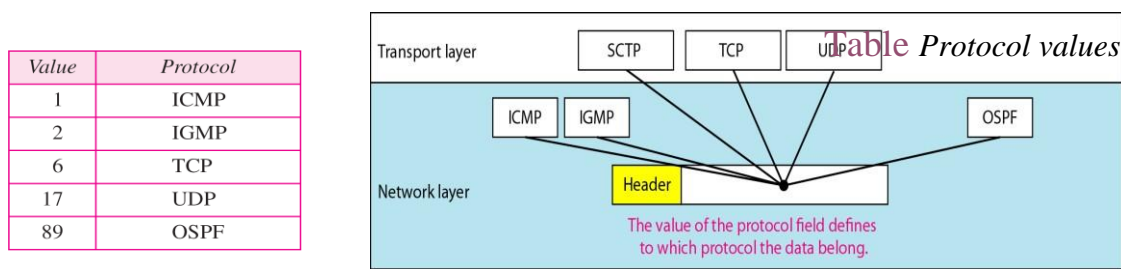
**Figure 4.3: Service Type**

**Note: The precedence subfield was part of version 4, but never used**

4. **Total Length:** This 16-bit field defines the total length (header plus data) of the IP datagram in bytes. A 16-bit number can define a total length of up to 65,535 (when all bits are 1s). However, the size of the datagram is normally much less than this. This field helps the receiving device to know when the packet has completely arrived. To find the length of the data coming from the upper layer, subtract the header length from the total length. The header length can be found by multiplying the value in the HLEN field by 4.

**Note: The total length field defines the total length of the datagram including the header.**

5. **Identification, Flags, and Fragmentation Offset:** These three fields are related to the fragmentation of the IP datagram when the size of the datagram is larger than the underlying network can carry.
6. **Time-to-live:** Due to some malfunctioning of routing protocols (discussed later) a datagram may be circulating in the Internet, visiting some networks over and over without reaching the destination. This may create extra traffic in the Internet. The time-to-live (TTL) field is used to control the maximum number of hops (routers) visited by the datagram. When a source host sends the datagram, it stores a number in this field. This value is approximately two times the maximum number of routers between any two hosts. Each router that processes the datagram decrements this number by one. If this value, after being decremented, is zero, the router discards the datagram.
7. **Protocol:** In TCP/IP, the data section of a packet, called the payload, carries the whole packet from another protocol. A datagram, for example, can carry a packet belonging to any transport-layer protocol such as UDP or TCP. A datagram can also carry a packet from other protocols that directly use the service of the IP, such as some routing protocols or some auxiliary protocols. The Internet authority has given any protocol that uses the service of IP a unique 8-bit number which is inserted in the protocol field. When the payload is encapsulated in a datagram at the source IP, the corresponding protocol number is inserted in this field; when the datagram arrives at the destination, the value of this field helps to define to which protocol the payload should be delivered. In other words, this field provides multiplexing at the source and demultiplexing at the destination



**Figure 4.4: Multiplexing and demultiplexing using the value of the protocol field**

8. **Header checksum:** IP is not a reliable protocol; it does not check whether the payload carried by a datagram is corrupted during the transmission. IP puts the burden of error checking of the payload on the protocol that owns the payload, such as UDP or TCP. The datagram header, however, is added by IP, and its error-checking is the responsibility of IP. Errors in the IP header can be a disaster. For example, if the destination IP address is corrupted, the packet can be delivered to the wrong host. If the protocol field is corrupted, the payload may be delivered to the wrong protocol. If the fields related to the fragmentation are corrupted, the datagram cannot be reassembled correctly at the destination, and so on. For these reasons, IP adds a header checksum field to check the header, but not the payload. We need to remember that, since the value of some fields, such as TTL, which are related to fragmentation and options, may change from router to router, the checksum needs to be recalculated at each router. Checksum in the Internet normally uses a 16-bit field, which is the complement of the sum of other fields calculated using 1s complement arithmetic.
9. **Source and Destination Addresses:** These 32-bit source and destination address fields define the IP address of the source and destination respectively. The source host should know its IP address. The destination IP address is either known by the protocol that uses the service of IP or is provided by the DNS. Note that the value of these fields must remain unchanged during the time the IP datagram travels from the source host to the destination host.

10. **Options:** A datagram header can have up to 40 bytes of options. Options can be used for network testing and debugging. Although options are not a required part of the IP header, option processing is required of the IP software. This means that all implementations must be able to handle options if they are present in the header. The existence of options in a header creates some burden on the datagram handling; some options can be changed by routers, which forces each router to recalculate the header checksum. There are one-byte and multi-byte options
11. **Payload:** Payload, or data, is the main reason for creating a datagram. Payload is the packet coming from other protocols that use the service of IP. Comparing a datagram to a postal package, payload is the content of the package; the header is only the information written on the package

### 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 ( $2 \times 4 = 8$ ). 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  $8 \times 4$ , or 32 bytes. The first 20 bytes are the base header, the next 12 bytes are the options.*

### Example 3

**Solution**

*The HLEN value is 5, which means the total number of bytes in the header is  $5 \times 4$ , 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.*

### Example 6

*An IPv4 packet has arrived with the header decimal digits as shown below. Calculate the checksum for this header*

	4	5	0	28	
	49.153		0	0	
	4	17	0		
Departm	10.12.14.5				
	12.6.7.9				

4, 5, and 0 →	4	5	0	0	
28 →	0	0	1	C	
1 →	C	0	0	1	
0 and 0 →	0	0	0	0	
4 and 17 →	0	4	1	1	
0 →	0	0	0	0	
10.12 →	0	A	0	C	
14.5 →	0	E	0	5	
12.6 →	0	C	0	6	
7.9 →	0	7	0	9	
Sum →	1	3	4	4	E
Wrapped sum →	3	4	4	F	
Checksum →	C	B	B	0	

Replaces 0  
 je 4

## Fragmentation

A datagram can travel through different networks. Each router decapsulates the IP datagram from the frame it receives, processes it, and then encapsulates it in another frame. The format and size of the received( or sent) frames depend on the protocol used by the physical network through which the frame has just travelled( or going to travel). For example, if a router connects a LAN to a WAN, it receives a frame in the LAN format and sends a frame in the WAN format.

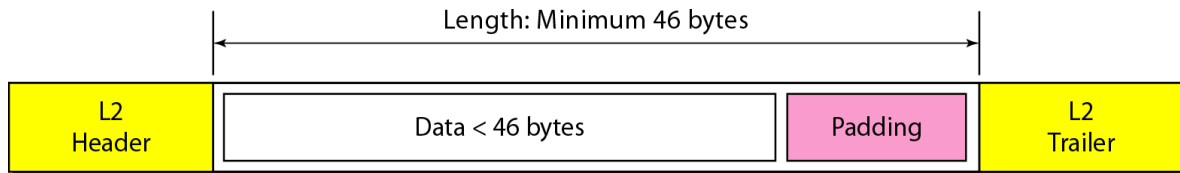


Figure 4.5: Encapsulation of a small datagram in an Ethernet frame

## Maximum transfer unit (MTU)

Each link-layer protocol has its own frame format. One of the features of each format is the maximum size of the payload that can be encapsulated. In other words, when a datagram is encapsulated in a frame, the total size of the datagram must be less than this maximum size, which is defined by the restrictions imposed by the hardware and software used in the network

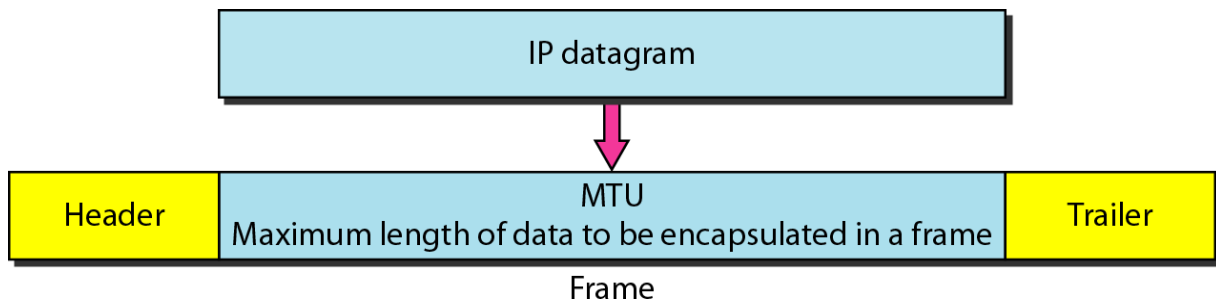


Figure 4.6: Maximum transfer unit (MTU)

In order to make the IP protocol independent of the physical network, the designers decided to make the maximum length of the IP datagram equal to 65,535 bytes. This makes transmission more efficient if one day we use a link-layer protocol with an MTU of this size. However, for other physical networks, we must divide the datagram to make it possible for it to pass through these networks. This is called fragmentation. When a datagram is fragmented, each fragment has its own header with most of the fields repeated, but some have been changed.

A fragmented datagram may itself be fragmented if it encounters a network with an even smaller MTU. In other words, a datagram may be fragmented several times before it reaches the final destination. A datagram can be fragmented by the source host or any router in the path. The reassembly of the datagram, however, is done only by the destination host, because each fragment becomes an independent datagram

Table 4.1 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

## Fields Related to Fragmentation

**Three fields in an IP datagram are related to fragmentation: identification, flags, and fragmentation offset**

When a datagram is fragmented, the value in the identification field is copied into all fragments. In other words, all fragments have the same identification number, which is also the same as the original datagram. The identification number helps the destination in reassembling the datagram. It knows that all fragments having the same identification value should be assembled into one datagram

## Flags used in fragmentation

When the payload of the IP datagram is fragmented, most parts of the header, with the exception of some options, must be copied by all fragments. The host or router that fragments a datagram must change the values of three fields: flags, fragmentation offset, and total length. The value of the checksum must be recalculated regardless of fragmentation.



Figure 4.7 Flags in fragmentation

The leftmost bit is reserved (not used). The second bit (D bit) is called the do not fragment bit. If its value is 1, the machine must not fragment the datagram. If it cannot pass the datagram through any available physical network, it discards the datagram and sends an ICMP error message to the source host (discussed later). If its value is 0, the datagram can be fragmented if necessary. The third bit (M bit) is called the more fragment bit. If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment.

## Fragmentation offset

The 13-bit fragmentation offset field shows the relative position of this fragment with respect to the whole datagram. It is the offset of the data in the original datagram measured in units of 8 bytes.

## Fragmentation example

Figure 4.8 shows a datagram with a data size of 4000 bytes fragmented into three fragments. The bytes in the original datagram are numbered 0 to 3999. The first fragment carries bytes 0 to 1399. The offset for this datagram is  $0/8 = 0$ . The second fragment carries bytes 1400 to 2799; the offset value for this fragment is  $1400/8 = 175$ . Finally, the third fragment carries bytes 2800 to 3999. The offset value for this fragment is  $2800/8 = 350$ .

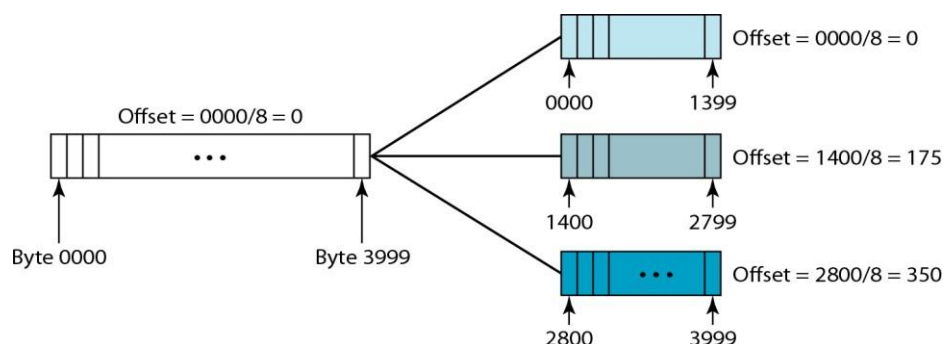


Figure 4.8 A fragmentation example

## Detailed fragmentation example

The value of the offset is measured in units of 8 bytes. This is done because the length of the offset field is only 13 bits long and cannot represent a sequence of bytes greater than 8191. This forces hosts or routers that fragment datagrams to choose the size of each fragment so that the first byte number is divisible by 8. Figure 4.9 shows an expanded view of the fragments in the previous figure. The original packet starts at the client; the fragments are reassembled at the server. The value of the identification field is the same in all fragments, as is the value of the flags field with the more bit set for all fragments except the last. Also, the value of the offset field for each fragment is shown. Note that although the fragments arrived out of order at the destination, they can be correctly reassembled.

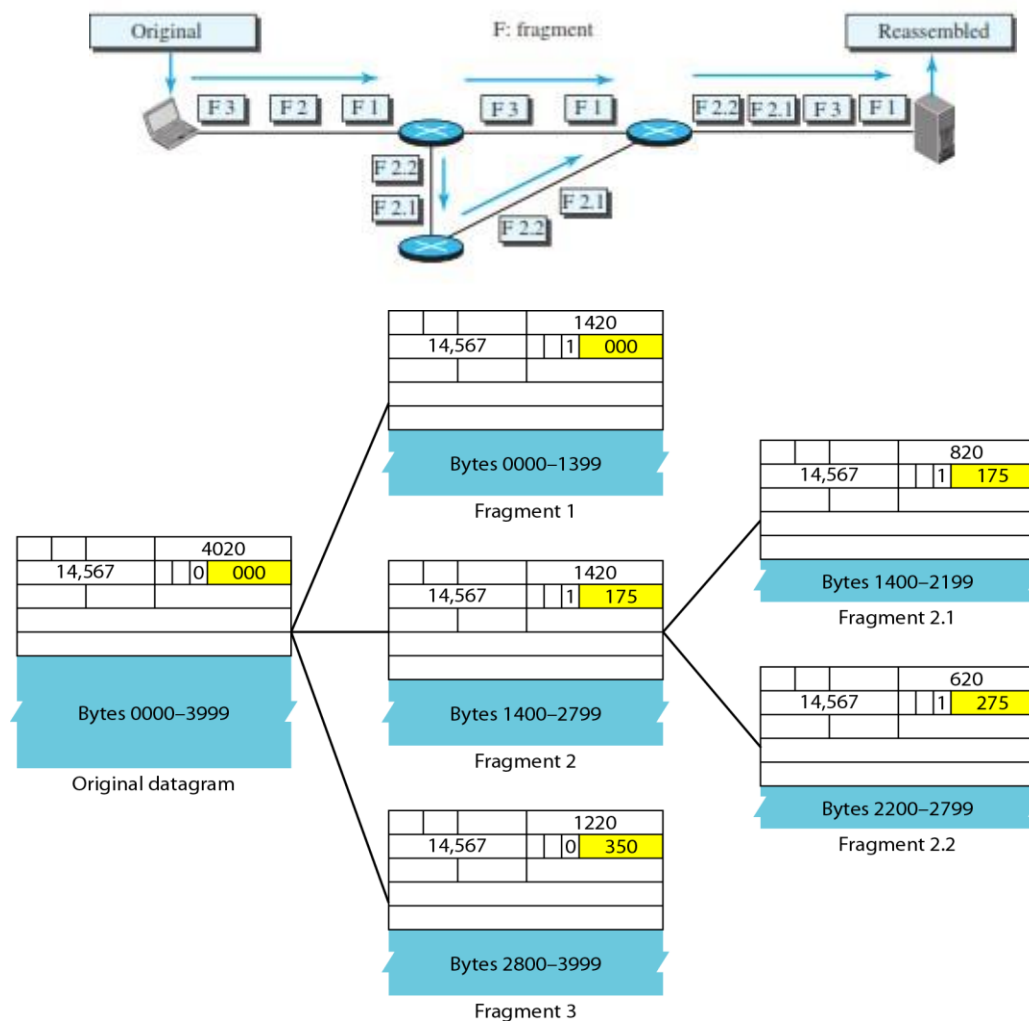


Figure 4.9 Detailed fragmentation examples - Expanded view

### Example

*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 :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: 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 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 of the data.

*Example - 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  $100 \times 8 = 800$ . The total length is 100 bytes, and the header length is 20 bytes ( $5 \times 4$ ), which means that there are 80 bytes in this datagram. If the first byte number is 800, the last byte number must be 879.

### Options

The header of the IPv4 datagram is made of two parts: a fixed part and a variable part. The fixed part is 20 bytes long and was discussed in the previous section. The variable part comprises the options that can be a maximum of 40 bytes (in multiples of 4-bytes) to preserve the boundary of the header. Options, as the name implies, are not required for a datagram. They can be used for network testing and debugging.

### Taxonomy of options in IPv4

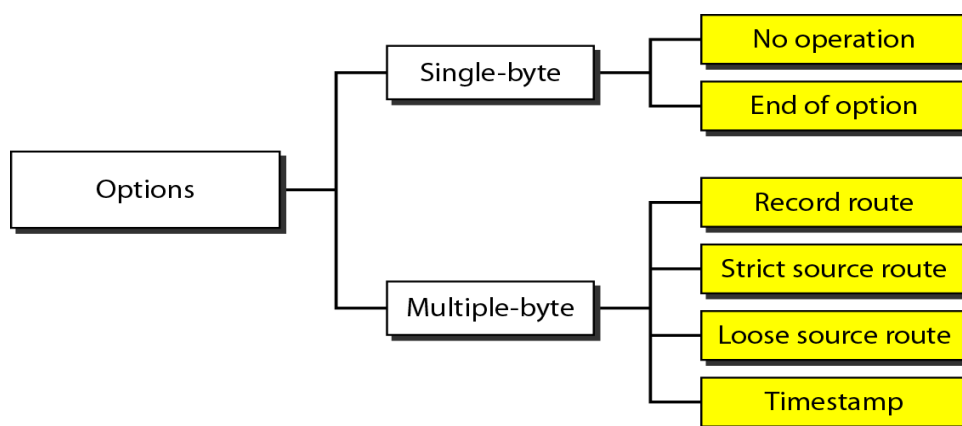


Figure 4.10: Classification of Options

**I Single-Byte Options:** There are two single-byte options. No Operation A no-operation option is a 1-byte option used as a filler between options. End of Option An end-of-option option is a 1-byte option used for padding at the end of the option field. It, however, can only be used as the last option.

**II Multiple-Byte Options:** There are four multiple-byte options.



- a) **Record Route** A record route option is used to record the Internet routers that handle the datagram. It can list up to nine router addresses. It can be used for debugging and management purposes.
- b) **Strict Source Route** A strict source route option is used by the source to predetermine a route for the datagram as it travels through the Internet. Here, the sender can choose a route with a specific type of service, such as minimum delay or maximum throughput. Alternatively, it may choose a route that is safer or more reliable for the sender's purpose. For example, a sender can choose a route so that its datagram does not travel through a competitor's network. If a datagram specifies a strict source route, all the routers defined in the option must be visited by the datagram
- c) **Loose Source Route:** A loose source route option is similar to the strict source route, but it is less rigid. Each router in the list must be visited, but the datagram can visit other routers as well.
- d) **Timestamp:** A timestamp option is used to record the time of datagram processing by a router. The time is expressed in milliseconds from midnight, Universal time or Greenwich mean time. Knowing the time a datagram is processed can help users and managers track the behavior of the routers in the Internet

### Security of IPv4 Datagrams

The IPv4 protocol, as well as the whole Internet, was started when the Internet users trusted each other. No security was provided for the IPv4 protocol. Today, however, the situation is different; the Internet is not secure anymore. There are three security issues that are particularly applicable to the IP protocol: packet sniffing, packet modification, and IP spoofing.

**Packet Sniffing:** An intruder may intercept an IP packet and make a copy of it. Packet sniffing is a passive attack, in which the attacker does not change the contents of the packet. This type of attack is very difficult to detect because the sender and the receiver may never know that the packet has been copied. Although packet sniffing cannot be stopped, encryption of the packet can make the attacker's effort useless. The attacker may still sniff the packet, but the content is not detectable.

**Packet Modification:** The second type of attack is to modify the packet. The attacker intercepts the packet, changes its contents, and sends the new packet to the receiver. The receiver believes that the packet is coming from the original sender. This type of attack can be detected using a data integrity mechanism. The receiver, before opening and using the contents of the message, can use this mechanism to make sure that the packet has not been changed during the transmission

**IP Spoofing:** An attacker can masquerade as somebody else and create an IP packet that carries the source address of another computer. An attacker can send an IP packet to a bank pretending that it is coming from one of the customers

**IPSec :** The IP packets today can be protected from the previously mentioned attacks using a protocol called IPSec (IP Security).

### ICMPv4

The IPv4 has no error-reporting or error-correcting mechanism. The IP protocol also lacks a mechanism for host and management queries. A host sometimes needs to determine if a router or another host is alive. And sometimes a network manager needs information from another host or router.

The Internet Control Message Protocol version 4 (ICMPv4) has been designed to compensate for the above two deficiencies. It is a companion to the IP protocol. ICMP itself is a network-layer protocol. However, its messages are not passed directly to the data-link layer as would be expected. Instead, the messages are first encapsulated inside IP datagrams before going to the lower layer. When an IP

datagram encapsulates an ICMP message, the value of the protocol field in the IP datagram is set to 1 to indicate that the IP payload is an ICMP message.

### MESSAGES

ICMP messages are divided into two broad categories: error-reporting messages and query messages. The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet. The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host. For example, nodes can discover their neighbors

An ICMP message has an 8-byte header and a variable-size data section. Although the general format of the header is different for each message type, the first 4 bytes are common to all. As figure 4 shows, the first field, ICMP type, defines the type of the message. The code field specifies the reason for the particular message type. The last common field is the checksum field (to be discussed later in the chapter). The rest of the header is specific for each message type.

The data section in error messages carries information for finding the original packet that had the error. In query messages, the data section carries extra information based on the type of query.

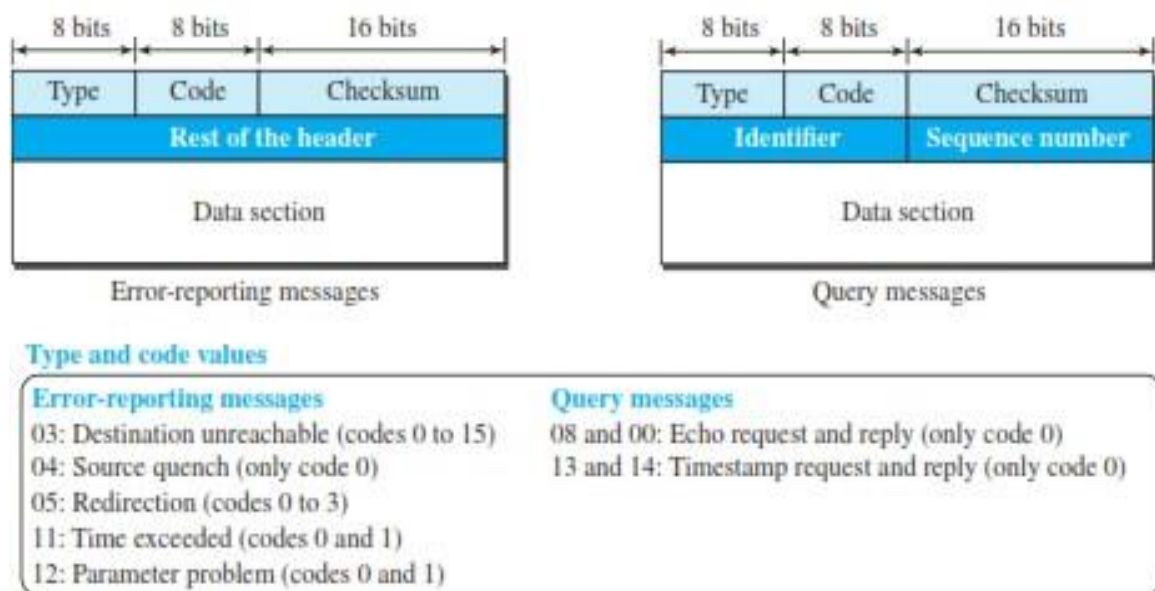


Figure 4.11 - General format of ICMP messages

### Error Reporting Messages

Since IP is an unreliable protocol, one of the main responsibilities of ICMP is to report some errors that may occur during the processing of the IP datagram. ICMP does not correct errors, it simply reports them. Error correction is left to the higher-level protocols. Error messages are always sent to the original source because the only information available in the datagram about the route is the source and destination IP addresses. ICMP uses the source IP address to send the error message to the source (originator) of the datagram. To make the error-reporting process simple, ICMP follows some rules in reporting messages.

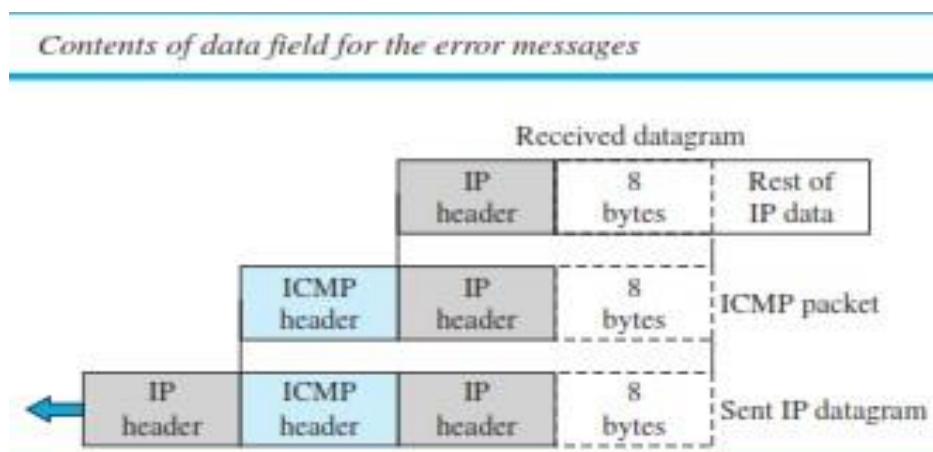
1. No error message will be generated for a datagram having a multicast address or special address (such as this host or loopback).
2. No ICMP error message will be generated in response to a datagram carrying an ICMP error message.

3. No ICMP error message will be generated for a fragmented datagram that is not the first fragment.

Note that all error messages contain a data section that includes the IP header of the original datagram plus the first 8 bytes of data in that datagram. The original datagram header is added to give the original source, which receives the error message, information about the datagram itself. The 8 bytes of data are included because the first 8 bytes provide information about the port numbers (UDP and TCP) and sequence number (TCP). This information is needed so the source can inform the protocols (TCP or UDP) about the error.

The following are important points about ICMP error messages:

- ☐ No ICMP error message will be generated in response to a datagram carrying an ICMP error message.
- ☐ No ICMP error message will be generated for a fragmented datagram that is not the first fragment.
- ☐ No ICMP error message will be generated for a datagram having a multicast address.
- ☐ No ICMP error message will be generated for a datagram having a special address such as 127.0.0.0 or 0.0.0.0.



### Error Message Types

- **Destination Unreachable** - The most widely used error message is the destination unreachable (type 3). This message uses different codes (0 to 15) to define the type of error message and the reason why a datagram has not reached its final destination
- **Source Quench** - Another error message is called the source quench (type 4) message, which informs the sender that the network has encountered congestion and the datagram has been dropped; the source needs to slow down sending more datagrams. In other words, ICMP adds a kind of congestion control mechanism to the IP protocol by using this type of message.
- **Redirection Message** - The redirection message (type 5) is used when the source uses a wrong router to send out its message. The router redirects the message to the appropriate router, but informs the source that it needs to change its default router in the future. The IP address of the default router is sent in the message.
- **Parameter Problem** - A parameter problem message (type 12) can be sent when either there is a problem in the header of a datagram (code 0) or some options are missing or cannot be interpreted (code 1).
- **Query Messages** - Query messages are used to probe or test the liveliness of hosts or routers in the Internet, find the one-way or the round-trip time for an IP datagram between two devices,

or even find out whether the clocks in two devices are synchronized. Query messages in ICMP can be used independently without relation to an IP datagram. But a query message needs to be encapsulated in a datagram, as a carrier. Naturally, query messages come in pairs: request and reply. The echo request (type 8) and the echo reply (type 0) pair of messages are used by a host or a router to test the liveliness of another host or router

### Deprecated Messages

Three pairs of messages are declared obsolete by IETF:

1. Information request and replay messages are not used today because their duties are done by the Address Resolution Protocol (ARP).
2. Address mask request and reply messages are not used today because their duties are done by the Dynamic Host Configuration Protocol (DHCP)
3. Router solicitation and advertisement messages are not used today because their duties are done by the Dynamic Host Configuration Protocol (DHCP)

### Debugging Tools

There are several tools that can be used in the Internet for debugging. We can determine the viability of a host or router. We can trace the route of a packet. We introduce two tools that use ICMP for debugging: ping and trace route.

**Ping** - We can use the ping program to find if a host is alive and responding. We use ping here to see how it uses ICMP packets. The source host sends ICMP echo-request messages; the destination, if alive, responds with ICMP echo-reply messages. The ping program sets the identifier field in the echo-request and echo-reply message and starts the sequence number from 0; this number is incremented by 1 each time a new message is sent. Note that ping can calculate the round-trip time. It inserts the sending time in the data section of the message. When the packet arrives, it subtracts the arrival time from the departure time to get the round-trip time (RTT).

#### *Example:*

*The following shows how we send a ping message to the auniversity.edu site. We set the identifier field in the echo request and reply message and start the sequence number from 0; this number is incremented by one each time a new message is sent. Note that ping can calculate the round-trip time. It inserts the sending time in the data section of the message. When the packet arrives, it subtracts the arrival time from the departure time to get the round-trip time (rtt).*

```
$ ping auniversity.edu  PING auniversity.edu (152.181.8.3)  56 (84) bytes of data.
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=0    ttl=62    time=1.91 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=1    ttl=62    time=2.04 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=2    ttl=62    time=1.90 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=3    ttl=62    time=1.97 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=4    ttl=62    time=1.93 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=5    ttl=62    time=2.00 ms
--- auniversity.edu statistics -- 6 packets transmitted, 6 received, 0% packet loss rtt min/avg/max =
1.90/1.95/2.04 ms
```

**Traceroute or Tracert** - The trace route program in UNIX or tracert in Windows can be used to trace the path of a packet from a source to the destination. It can find the IP addresses of all the routers that are visited along the path. The program is usually set to check for the maximum of 30 hops (routers) to be visited. The number of hops in the Internet is normally less than this.

**Traceroute** - The trace route program is different from the ping program. The ping program gets help from two query messages; the trace route program gets help from two error-reporting messages: time-exceeded and destination-unreachable. The trace route is an application layer program, but only the client program is needed, because, as we can see, the client program never reaches the application layer in the destination host. In other words, there is no trace route server program. The trace route application program is encapsulated in a UDP user datagram, but trace route intentionally uses a port number that is not available at the destination. If there are  $n$  routers in the path, the trace route program sends  $(n + 1)$  messages. The first  $n$  messages are discarded by the  $n$  routers, one by each router; the last message is discarded by the destination host. The trace route client program uses the  $(n + 1)$  ICMP error-reporting messages received to find the path between the routers.

The first trace route message is sent with time-to-live (TTL) value set to 1; the message is discarded at the first router and a time-exceeded ICMP error message is sent, from which the trace route program can find the IP address of the first router (the source IP address of the error message) and the router name (in the data section of the message). The second trace route message is sent with TTL set to 2, which can find the IP address and the name of the second router. Similarly, the third message can find the information about router 3. The fourth message, however, reaches the destination host.

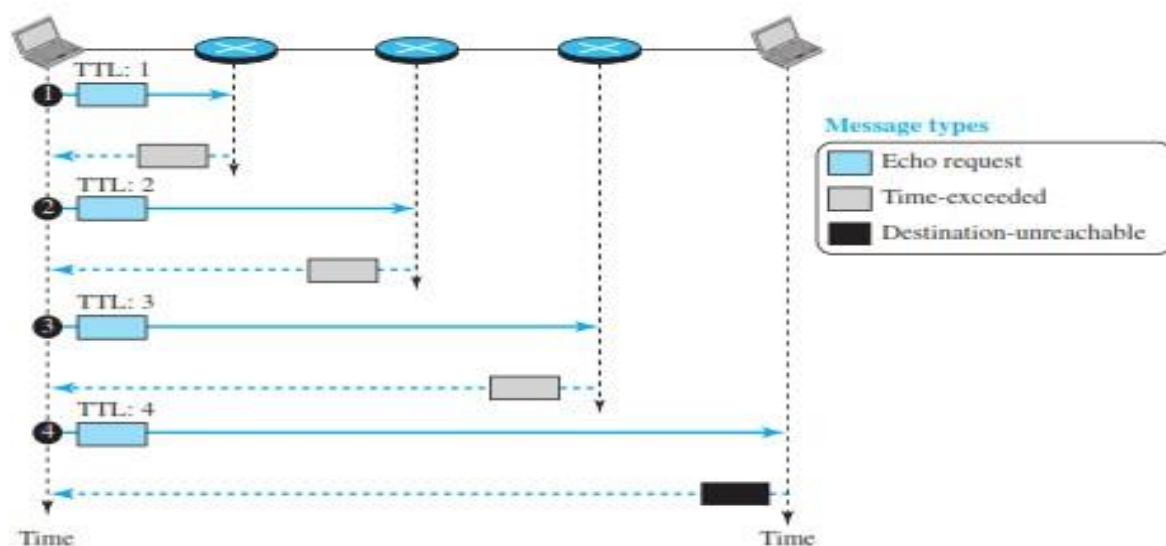


Figure 4.12: Use of ICMPv4 in trace route

The trace route program also sets a timer to find the round-trip time for each router and the destination. Most trace route programs send three messages to each device, with the same TTL value, to be able to find a better estimate for the round-trip time. The following shows an example of a trace route program, which uses three probes for each device and gets three RTTs.

```
4 trace route printers.com trace route to printers.com (13.1.69.93), 30 hops max, 38-byte
packets 1 route.front.edu (153.18.31.254) 0.622 ms 0.891 ms 0.875 ms
2 ceneric.net (137.164.32.140) 3.069 ms 2.875 ms 2.930 ms
3 satire.net (132.16.132.20) 3.071 ms 2.876 ms 2.929 ms
4 alpha.printers.com (13.1.69.93) 5.922 ms 5.048 ms 4.922 ms
```

**Tracert** - The tracert program in windows behaves differently. The tracert messages are encapsulated directly in IP datagrams. The tracert, like trace route, sends echo-request messages. However, when the last echo request reaches the destination host, an echo replay message is issued



## ICMP Checksum

In ICMP the checksum is calculated over the entire message (header and data).

*Example: Figure 4.13 below shows an example of checksum calculation for a simple echo-request message. We randomly chose the identifier to be 1 and the sequence number to be 9. The message is divided into 16-bit (2-byte) words. The words are added and the sum is complemented. Now the sender can put this value in the checksum field.*

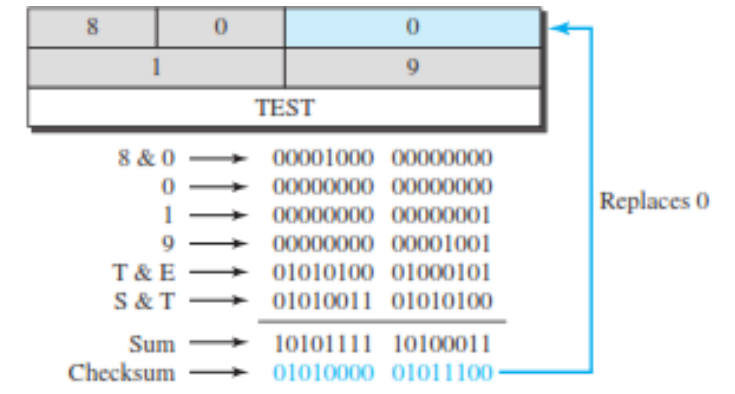


Figure 4.13 – Calculation of ICMP checksum

## MOBILE IP

As mobile and personal computers such as notebooks become increasingly popular, we need to think about mobile IP, the extension of IP protocol that allows mobile computers to be connected to the Internet at any location where the connection is possible.

### Addressing

The main problem that must be solved in providing mobile communication using the IP protocol is addressing.

**Stationary Hosts** - The original IP addressing was based on the assumption that a host is stationary, attached to one specific network. A router uses an IP address to route an IP datagram. An IP address has two parts: a prefix and a suffix. The prefix associates a host with a network. For example, the IP address 10.3.4.24/8 defines a host attached to the network 10.0.0.0/8.

The IP addresses are designed to work with stationary hosts because part of the address defines the network to which the host is attached.

**Mobile Hosts** - When a host moves from one network to another, the IP addressing structure needs to be modified. Several solutions have been proposed.

1. **Changing the Address** - One simple solution is to let the mobile host change its address as it goes to the new network. The host can use DHCP to obtain a new address to associate it with the new network. This approach has several drawbacks. First, the configuration files would need to be changed. Second, each time the computer moves from one network to another, it must be rebooted. Third, the DNS tables need to be revised so that every other host in the Internet is aware of the change. Fourth, if the host roams from one network to another during a transmission, the data exchange will be interrupted. This is because the ports and IP addresses of the client and the server must remain constant for the duration of the connection.
2. **Two Addresses** - The approach that is more feasible is the use of two addresses. The host has its original address, called the home address, and a temporary address, called the care-of

address. The home address is permanent; it associates the host with its home network, the network that is the permanent home of the host. The care-of address is temporary. When a host moves from one network to another, the care-of address changes; it is associated with the foreign network, the network to which the host moves. When a mobile host visits a foreign network, it receives its care-of address during the agent discovery and registration phase

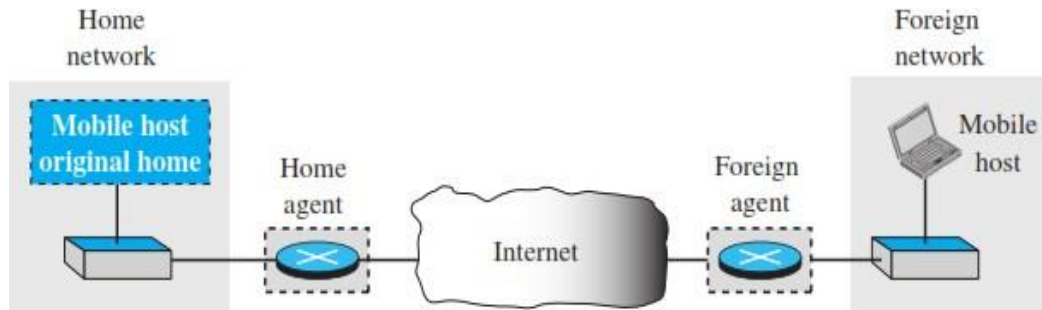


Figure 4.14: Home address and care-of address

### Agents

To make the change of address transparent to the rest of the Internet requires a home agent and a foreign agent. Figure 4.14 shows the position of a home agent relative to the home network and a foreign agent relative to the foreign network. We have shown the home and the foreign agents as routers, but we need to emphasize that their specific function as an agent is performed in the application layer. In other words, they are both routers and hosts.

**Home Agent** - The home agent is usually a router attached to the home network of the mobile host. The home agent acts on behalf of the mobile host when a remote host sends a packet to the mobile host. The home agent receives the packet and sends it to the foreign agent.

**Foreign Agent** - The foreign agent is usually a router attached to the foreign network. The foreign agent receives and delivers packets sent by the home agent to the mobile host. The mobile host can also act as a foreign agent. In other words, the mobile host and the foreign agent can be the same. However, to do this, a mobile host must be able to receive a care-of address by itself, which can be done through the use of DHCP. In addition, the mobile host needs the necessary software to allow it to communicate with the home agent and to have two addresses: its home address and its care-of address. This dual addressing must be transparent to the application programs. When the mobile host acts as a foreign agent, the care-of address is called a collocated care-of address.

The advantage of using a collocated care-of address is that the mobile host can move to any network without worrying about the availability of a foreign agent. The disadvantage is that the mobile host needs extra software to act as its own foreign agent

**Three Phases** - To communicate with a remote host, a mobile host goes through three phases: agent discovery, registration, and data transfer, as shown in Figure 4.15.



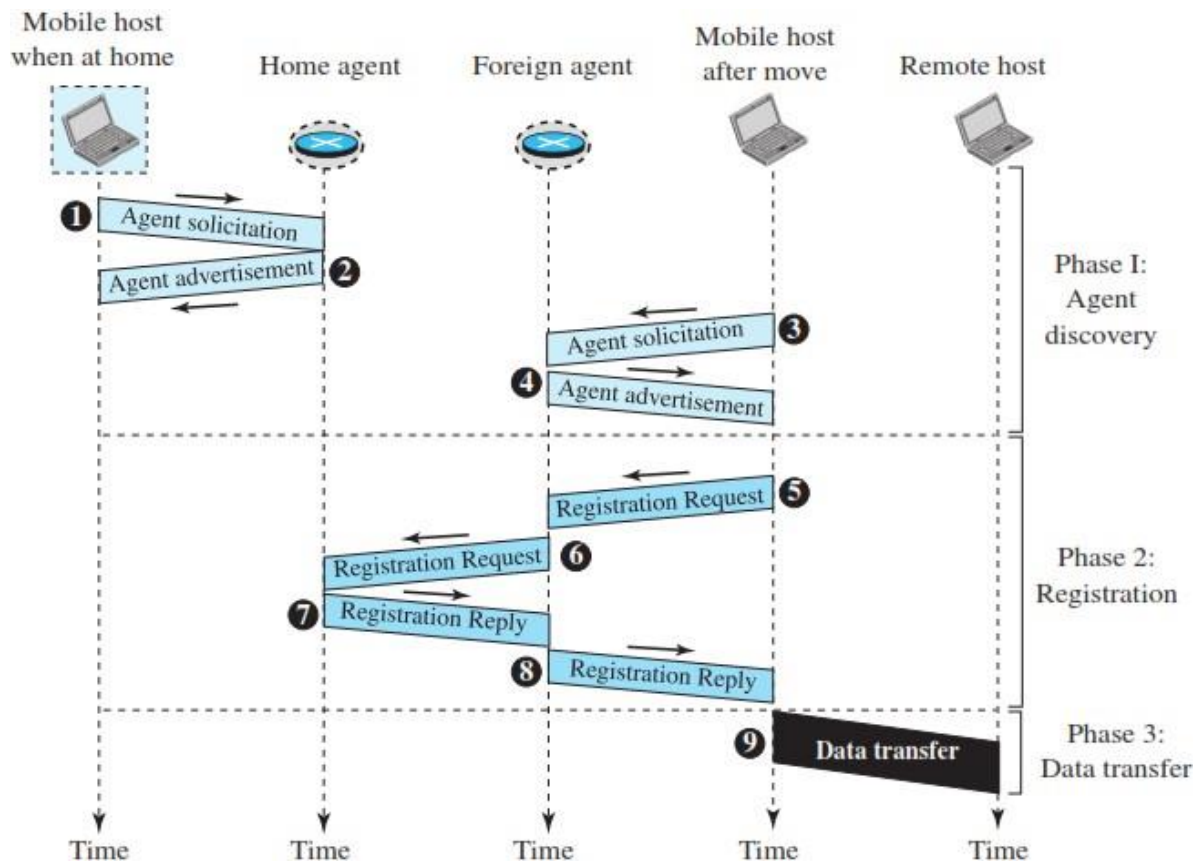


Figure 4.15 : Remote host and mobile host communication

The first phase, agent discovery, involves the mobile host, the foreign agent, and the home agent. The second phase, registration, also involves the mobile host and the two agents. Finally, in the third phase, the remote host is also involved. We discuss each phase separately.

**Agent Discovery** - The first phase in mobile communication, agent discovery, consists of two sub phases. A mobile host must discover (learn the address of) a home agent before it leaves its home network. A mobile host must also discover a foreign agent after it has moved to a foreign network. This discovery consists of learning the care-of address as well as the foreign agent's address. The discovery involves two types of messages: advertisement and solicitation

**Agent Advertisement** - When a router advertises its presence on a network using an ICMP router advertisement, it can append an agent advertisement to the packet if it acts as an agent. Mobile IP does not use a new packet type for agent advertisement; it uses the router advertisement packet of ICMP, and appends an agent advertisement message

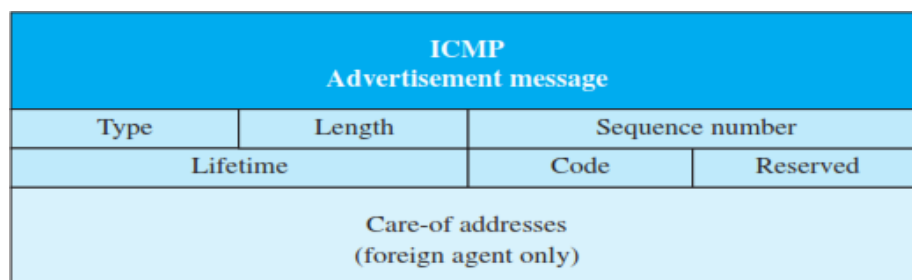


Figure 4.16 : Agent advertisement

The field descriptions are as follows:

- ❑ **Type.** The 8-bit type field is set to 16.
- ❑ **Length.** The 8-bit length field defines the total length of the extension message (not the length of the ICMP advertisement message).
- ❑ **Sequence number.** The 16-bit sequence number field holds the message number. The recipient can use the sequence number to determine if a message is lost.
- ❑ **Lifetime.** The lifetime field defines the number of seconds that the agent will accept requests. If the value is a string of 1s, the lifetime is infinite.
- ❑ **Code.** The code field is an 8-bit flag in which each bit is set (1) or unset (0). The meanings of the bits are shown in Table 4.2

Table 4.2 :Code bits

<i>Bit</i>	<i>Meaning</i>
0	Registration required. No collocated care-of address.
1	Agent is busy and does not accept registration at this moment.
2	Agent acts as a home agent.
3	Agent acts as a foreign agent.
4	Agent uses minimal encapsulation.
5	Agent uses generic routing encapsulation (GRE).
6	Agent supports header compression.
7	Unused (0).

❑ **Care-of Addresses.** This field contains a list of addresses available for use as careof addresses. The mobile host can choose one of these addresses. The selection of this care-of address is announced in the registration request. Note that this field is used only by a foreign agent.

### Agent Solicitation

When a mobile host has moved to a new network and has not received agent advertisements, it can initiate an agent solicitation. It can use the ICMP solicitation message to inform an agent that it needs assistance. Mobile IP does not use a new packet type for agent solicitation; it uses the router solicitation packet of ICMP.

**Registration** - The second phase in mobile communication is registration. After a mobile host has moved to a foreign network and discovered the foreign agent, it must register. There are four aspects of registration:

1. The mobile host must register itself with the foreign agent.
2. The mobile host must register itself with its home agent. This is normally done by the foreign agent on behalf of the mobile host.
3. The mobile host must renew registration if it has expired.
4. The mobile host must cancel its registration (deregistration) when it returns home. Request and Reply To register with the foreign agent and the home agent, the mobile host uses a registration request and a registration reply as shown in Figure 4.17.

**Registration Request** - A registration request is sent from the mobile host to the foreign agent to register its care-of address and also to announce its home address and home agent address. The foreign agent, after receiving and registering the request, relays the message to the home agent.

Type	Flag	Lifetime
Home address		
Home agent address		
Care-of address		
Identification		
Extensions ...		

Figure 4.17 : Registration request format

The field descriptions are as follows:

- ❑ **Type.** The 8-bit type field defines the type of message. For a request message the value of this field is 1.
- ❑ **Flag.** The 8-bit flag field defines forwarding information. The value of each bit can be set or unset. The meaning of each bit is given in Table 4.3.

Table 4.3 : Registration request flag field bits

Bit	Meaning
0	Mobile host requests that home agent retain its prior care-of address.
1	Mobile host requests that home agent tunnel any broadcast message.
2	Mobile host is using collocated care-of address.
3	Mobile host requests that home agent use minimal encapsulation.
4	Mobile host requests generic routing encapsulation (GRE).
5	Mobile host requests header compression.
6–7	Reserved bits.

- ❑ **Lifetime.** This field defines the number of seconds the registration is valid. If the field is a string of 0s, the request message is asking for deregistration. If the field is a string of 1s, the lifetime is infinite.
- ❑ **Home address.** This field contains the permanent (first) address of the mobile host.
- ❑ **Home agent address.** This field contains the address of the home agent.
- ❑ **Care-of address.** This field is the temporary (second) address of the mobile host. ❑ **Identification.** This field contains a 64-bit number that is inserted into the request by the mobile host and repeated in the reply message. It matches a request with a reply.
- ❑ **Extensions.** Variable length extensions are used for authentication. They allow a home agent to authenticate the mobile agent

**Registration Reply** - A registration reply is sent from the home agent to the foreign agent and then relayed to the mobile host. The reply confirms or denies the registration request. Figure 4.18 shows the

format of the registration reply. The fields are similar to those of the registration request with the following exceptions. The value of the type field is 3. The code field replaces the flag field and shows the result of the registration request (acceptance or denial). The care-of address field is not needed.

Type	Code	Lifetime
Home address		
Home agent address		
Identification		
Extensions ...		

Figure 4.18: Registration reply format

**Encapsulation** - Registration messages are encapsulated in a UDP user datagram. An agent uses the well-known port 434; a mobile host uses an ephemeral port.

**Data Transfer** - After agent discovery and registration, a mobile host can communicate with a remote host. Figure 4.19 shows the idea.

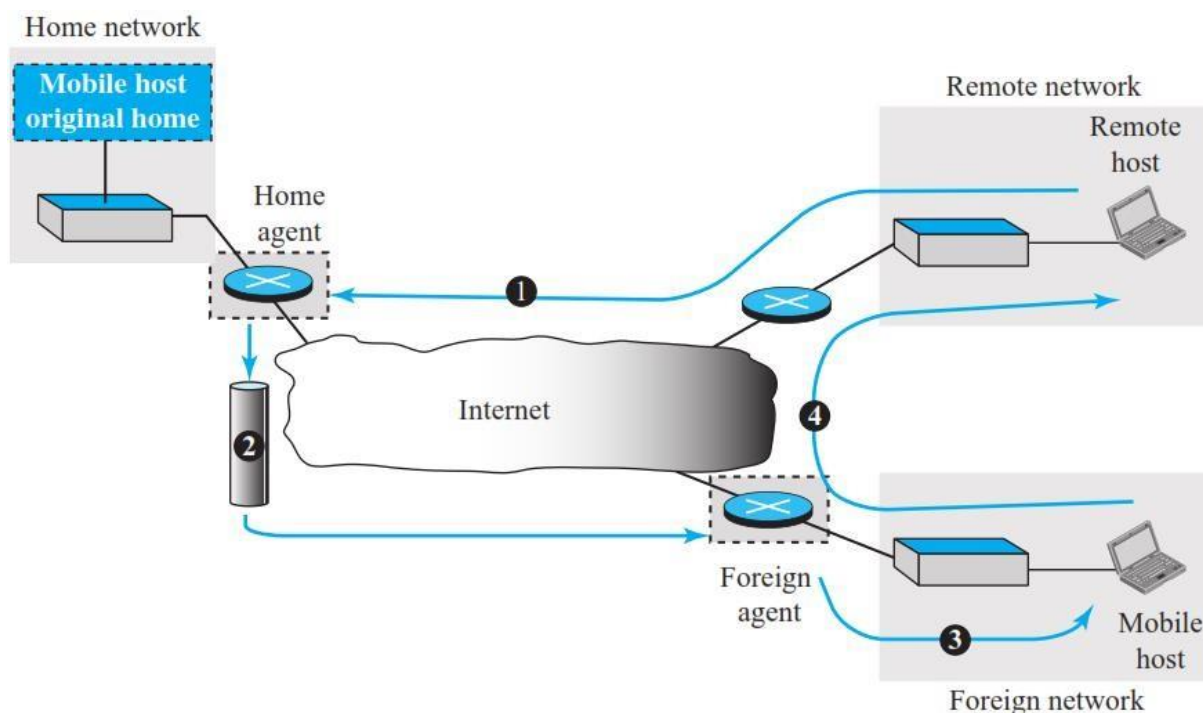


Figure 4.19: Data transfer

**From Remote Host to Home Agent** - When a remote host wants to send a packet to the mobile host, it uses its address as the source address and the home address of the mobile host as the destination address. In other words, the remote host sends a packet as though the mobile host is at its home network. The packet, however, is intercepted by the home agent, which pretends it is the mobile host. This is done using the proxy ARP technique. Path 1 of Figure 4.19 shows this step.

**From Home Agent to Foreign Agent** - After receiving the packet, the home agent sends the packet to the foreign agent, using the tunneling concept. The home agent encapsulates the whole IP packet inside another IP packet using its address as the source and the foreign agent's address as the destination. Path 2 of Figure 4.19 shows this step.

**From Foreign Agent to Mobile Host** When the foreign agent receives the packet, it removes the original packet. However, since the destination address is the home address of the mobile host, the foreign agent consults a registry table to find the care-of address of the mobile host. (Otherwise, the packet would just be sent back to the home network.) The packet is then sent to the care-of address. Path 3 of Figure 4.19 shows this step.

**From Mobile Host to Remote Host** When a mobile host wants to send a packet to a remote host (for example, a response to the packet it has received), it sends as it does normally. The mobile host prepares a packet with its home address as the source, and the address of the remote host as the destination. Although the packet comes from the foreign network, it has the home address of the mobile host. Path 4 of Figure 4.19 shows this step.

**Transparency** - In this data transfer process, the remote host is unaware of any movement by the mobile host. The remote host sends packets using the home address of the mobile host as the destination address; it receives packets that have the home address of the mobile host as the source address. The movement is totally transparent. The rest of the Internet is not aware of the movement of the mobile host.

### Inefficiency in Mobile IP

Communication involving mobile IP can be inefficient. The inefficiency can be severe or moderate. The severe case is called double crossing or 2X. The moderate case is called triangle routing or dog-leg routing.

**Double Crossing** - Double crossing occurs when a remote host communicates with a mobile host that has moved to the same network (or site) as the remote host (see Figure 4.20). When the mobile host sends a packet to the remote host, there is no inefficiency; the communication is local. However, when the remote host sends a packet to the mobile host, the packet crosses the Internet twice. Since a computer usually communicates with other local computers (principle of locality), the inefficiency from double crossing is significant.

**Triangle Routing** - Triangle routing, the less severe case, occurs when the remote host communicates with a mobile host that is not attached to the same network (or site) as the mobile host. When the mobile host sends a packet to the remote host, there is no inefficiency

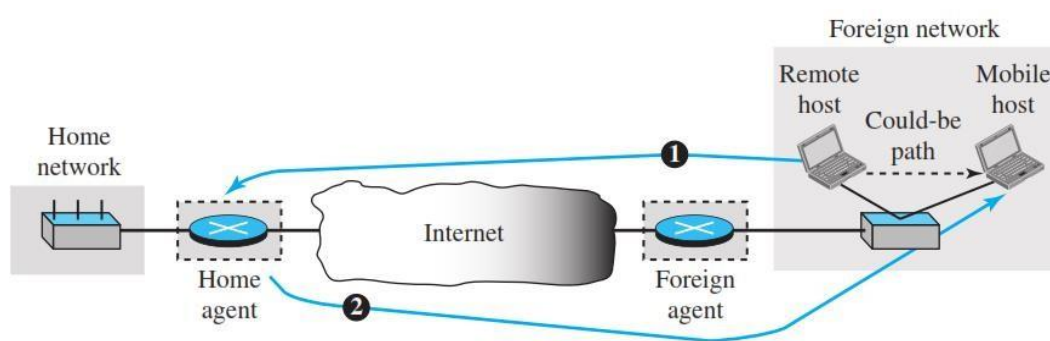


Figure 4. 20: Double crossing

However, when the remote host sends a packet to the mobile host, the packet goes from the remote host to the home agent and then to the mobile host. The packet travels the two sides of a triangle, instead of just one side

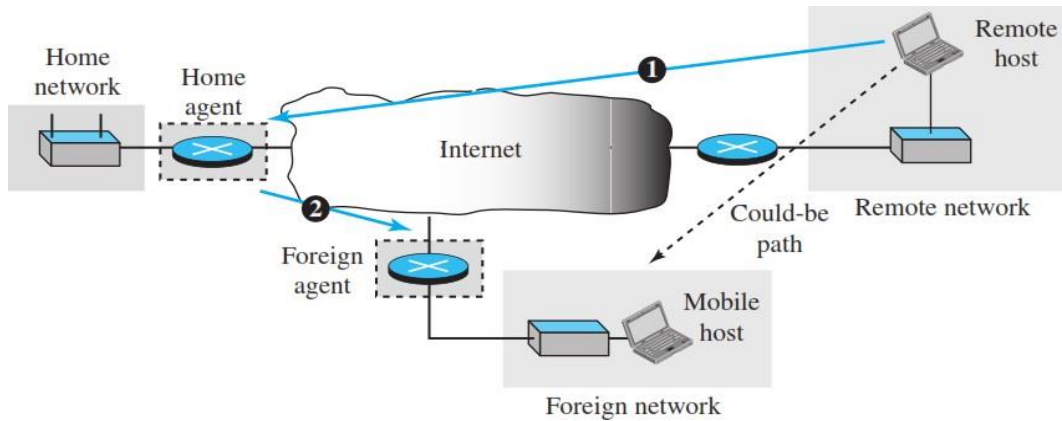


Figure 4.21: Triangle routing

**Solution** - One solution to inefficiency is for the remote host to bind the care-of address to the home address of a mobile host. For example, when a home agent receives the first packet for a mobile host, it forwards the packet to the foreign agent; it could also send an update binding packet to the remote host so that future packets to this host could be sent to the care-of address. The remote host can keep this information in a cache. The problem with this strategy is that the cache entry becomes outdated once the mobile host moves. In this case the home agent needs to send a warning packet to the remote host to inform it of the change.