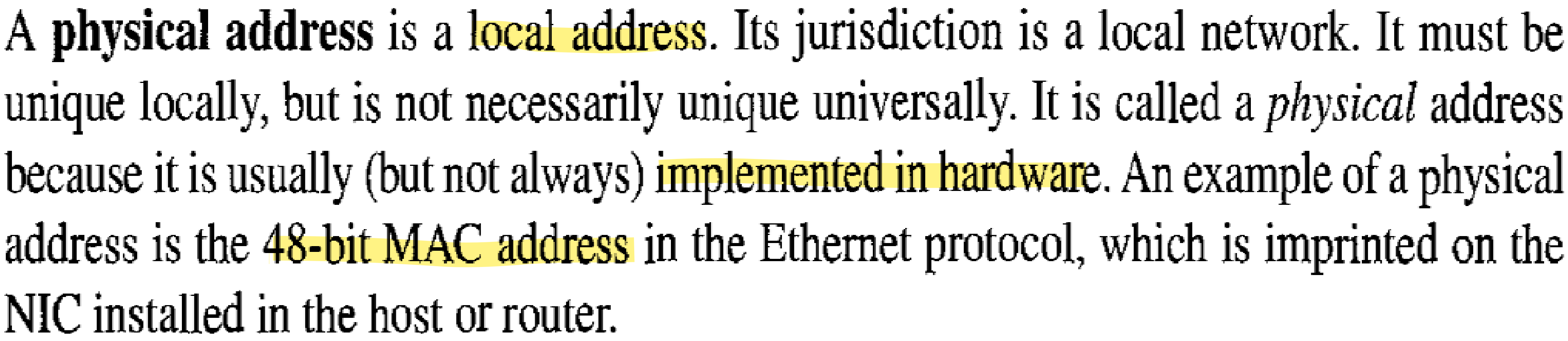
# Mapping Logical address to Physical address

By

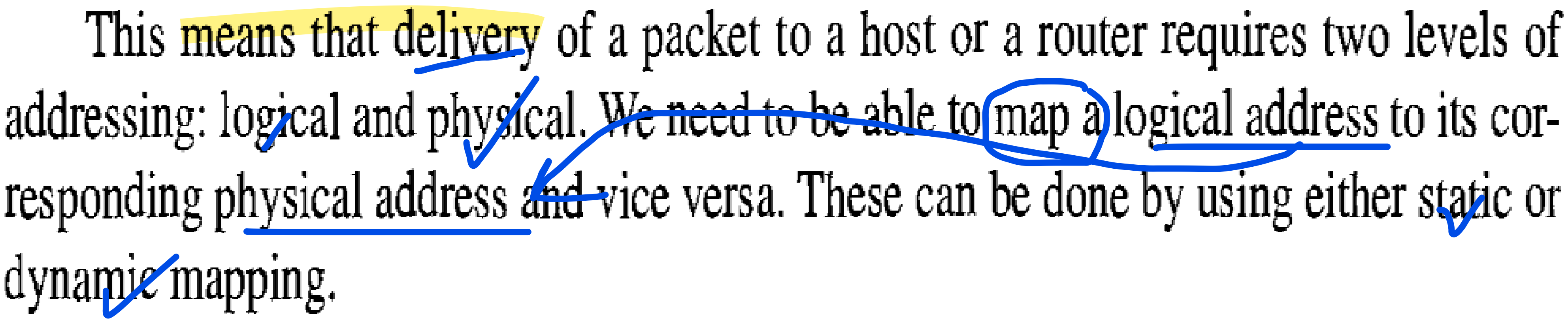
K raghuveer

# Physical Address



A **physical address** is a local address. Its jurisdiction is a local network. It must be unique locally, but is not necessarily unique universally. It is called a *physical* address because it is usually (but not always) implemented in hardware. An example of a physical address is the 48-bit MAC address in the Ethernet protocol, which is imprinted on the NIC installed in the host or router.

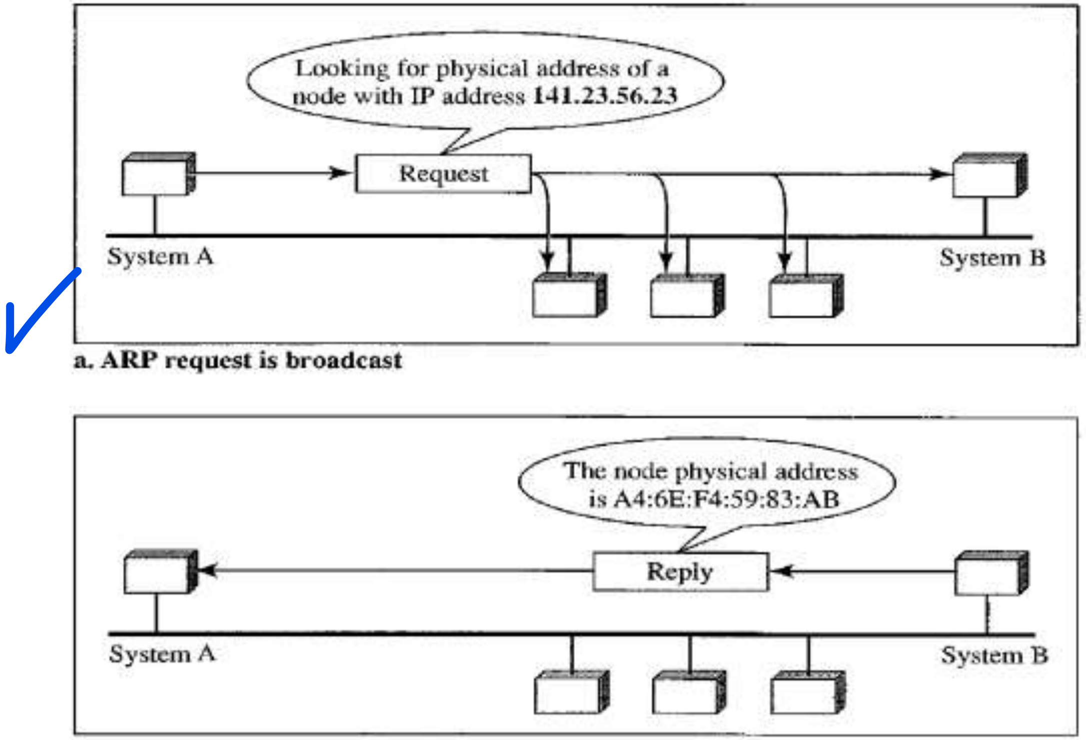
# Logical address



This means that delivery of a packet to a host or a router requires two levels of addressing: logical and physical. We need to be able to map a logical address to its corresponding physical address and vice versa. These can be done by using either static or dynamic mapping.

# Mapping Logical to Physical Address - ARP

**Figure 21.1** *ARP operation*

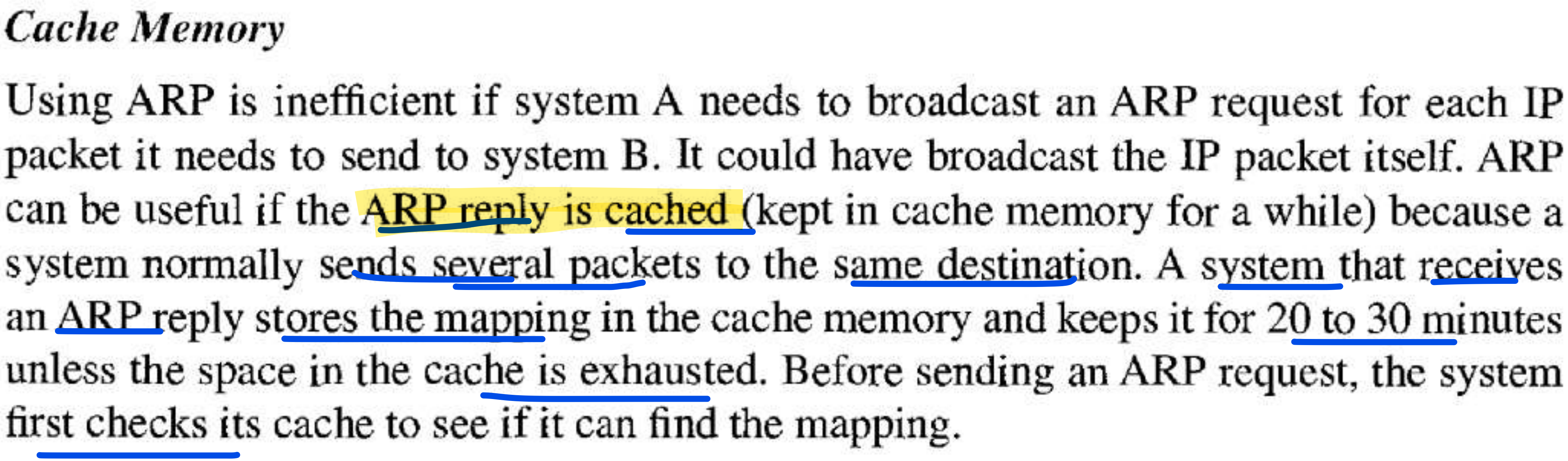

Looking for physical address of a node with IP address 141.23.56.23
Request
System A
System B
The node physical address is A4:6E:F4:59:83:AB
Reply
System A
System B

✓

**a. ARP request is broadcast**

**b. ARP reply is unicast**

# ARP with cache memory

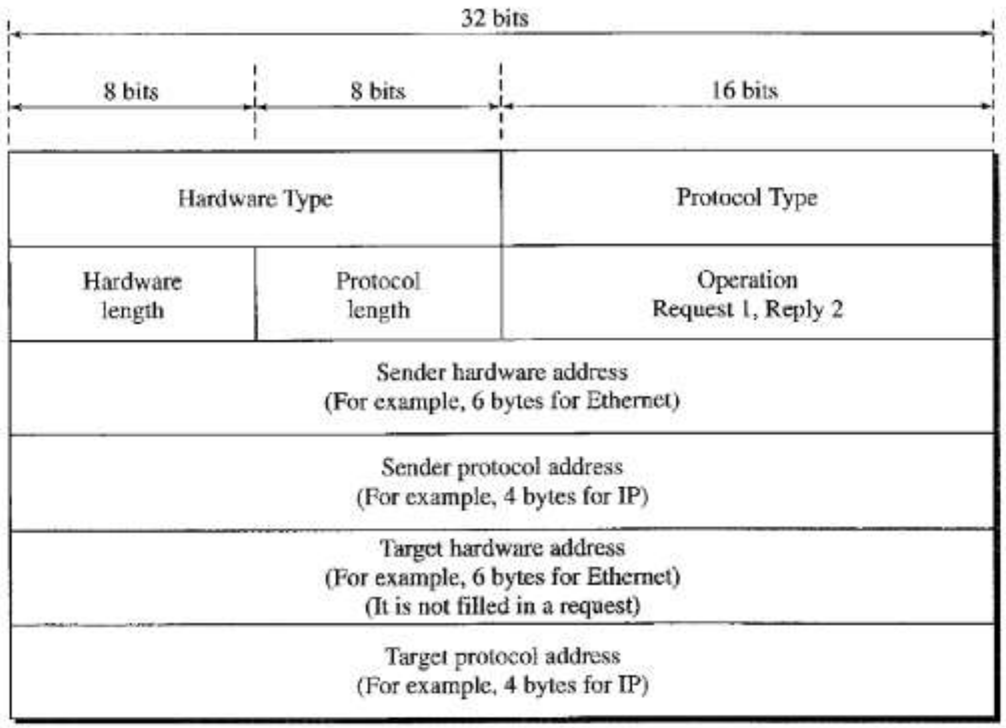


### *Cache Memory*

Using ARP is inefficient if system A needs to broadcast an ARP request for each IP packet it needs to send to system B. It could have broadcast the IP packet itself. ARP can be useful if the ARP reply is cached (kept in cache memory for a while) because a system normally sends several packets to the same destination. A system that receives an ARP reply stores the mapping in the cache memory and keeps it for 20 to 30 minutes unless the space in the cache is exhausted. Before sending an ARP request, the system first checks its cache to see if it can find the mapping.

# ARP packet

**Figure 21.2** *ARP packet*



Hardware Type

Protocol Type

Hardware length

Protocol length

Operation
Request 1, Reply 2

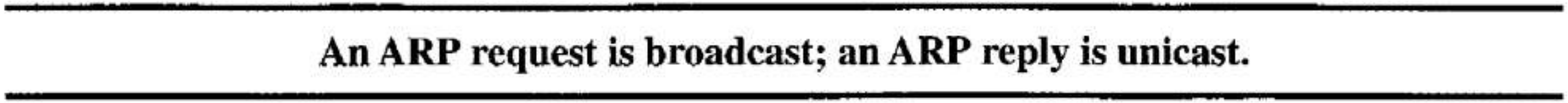
Sender hardware address
(For example, 6 bytes for Ethernet)

Sender protocol address
(For example, 4 bytes for IP)

Target hardware address
(For example, 6 bytes for Ethernet)
(It is not filled in a request)

Target protocol address
(For example, 4 bytes for IP)

# Feature



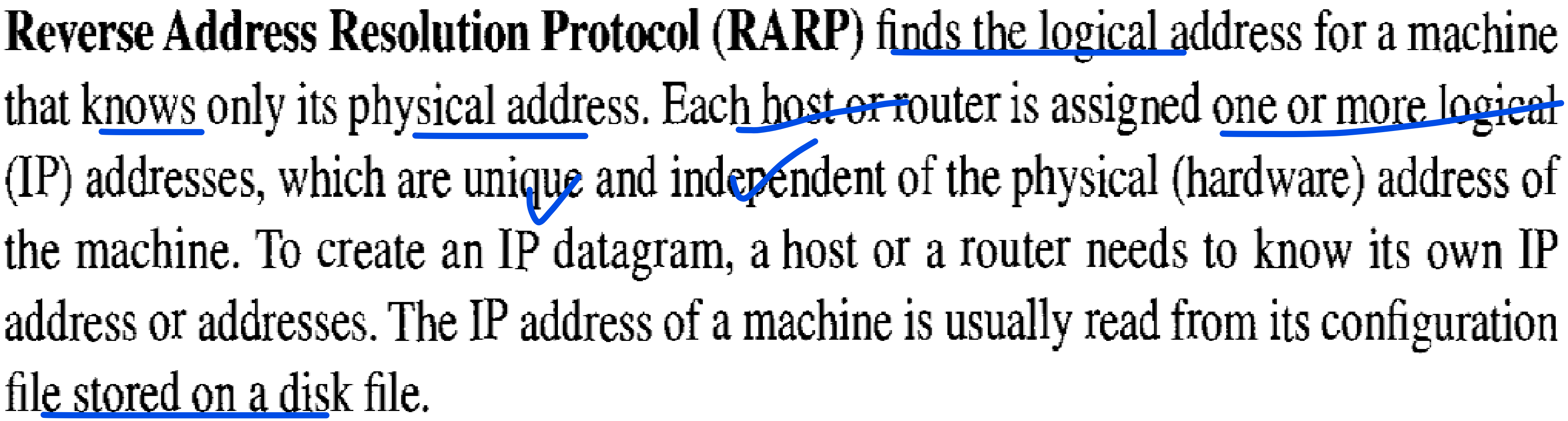
**An ARP request is broadcast; an ARP reply is unicast.**

# Mapping physical to logical address: RARP, Bootp and DHCP

There are occasions in which a host knows its physical address, but needs to know its logical address. This may happen in two cases:

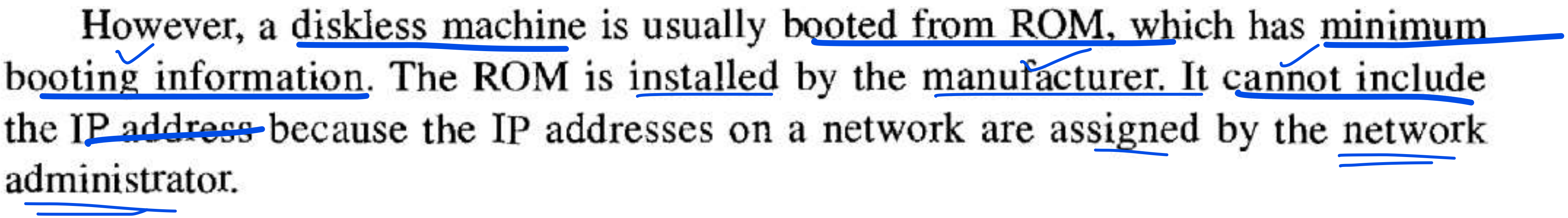
1. A diskless station is just booted. The station can find its physical address by checking its interface, but it does not know its IP address.
2. An organization does not have enough IP addresses to assign to each station; it needs to assign IP addresses on demand. The station can send its physical address and ask for a short time lease.

# RARP



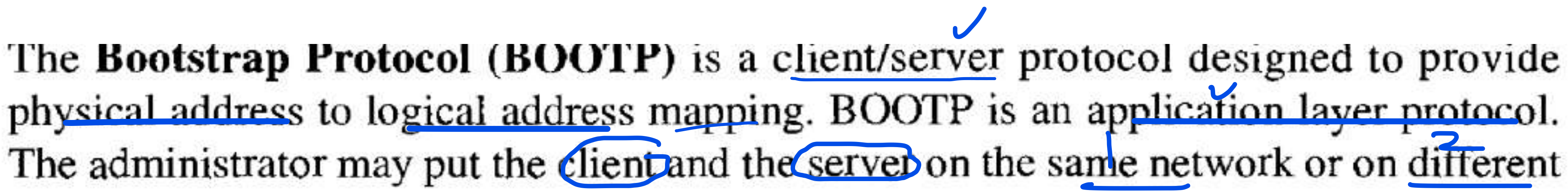
**Reverse Address Resolution Protocol (RARP)** finds the logical address for a machine that knows only its physical address. Each host or router is assigned one or more logical (IP) addresses, which are unique and independent of the physical (hardware) address of the machine. To create an IP datagram, a host or a router needs to know its own IP address or addresses. The IP address of a machine is usually read from its configuration file stored on a disk file.

# RARP



However, a diskless machine is usually booted from ROM, which has minimum booting information. The ROM is installed by the manufacturer. It cannot include the IP address because the IP addresses on a network are assigned by the network administrator.

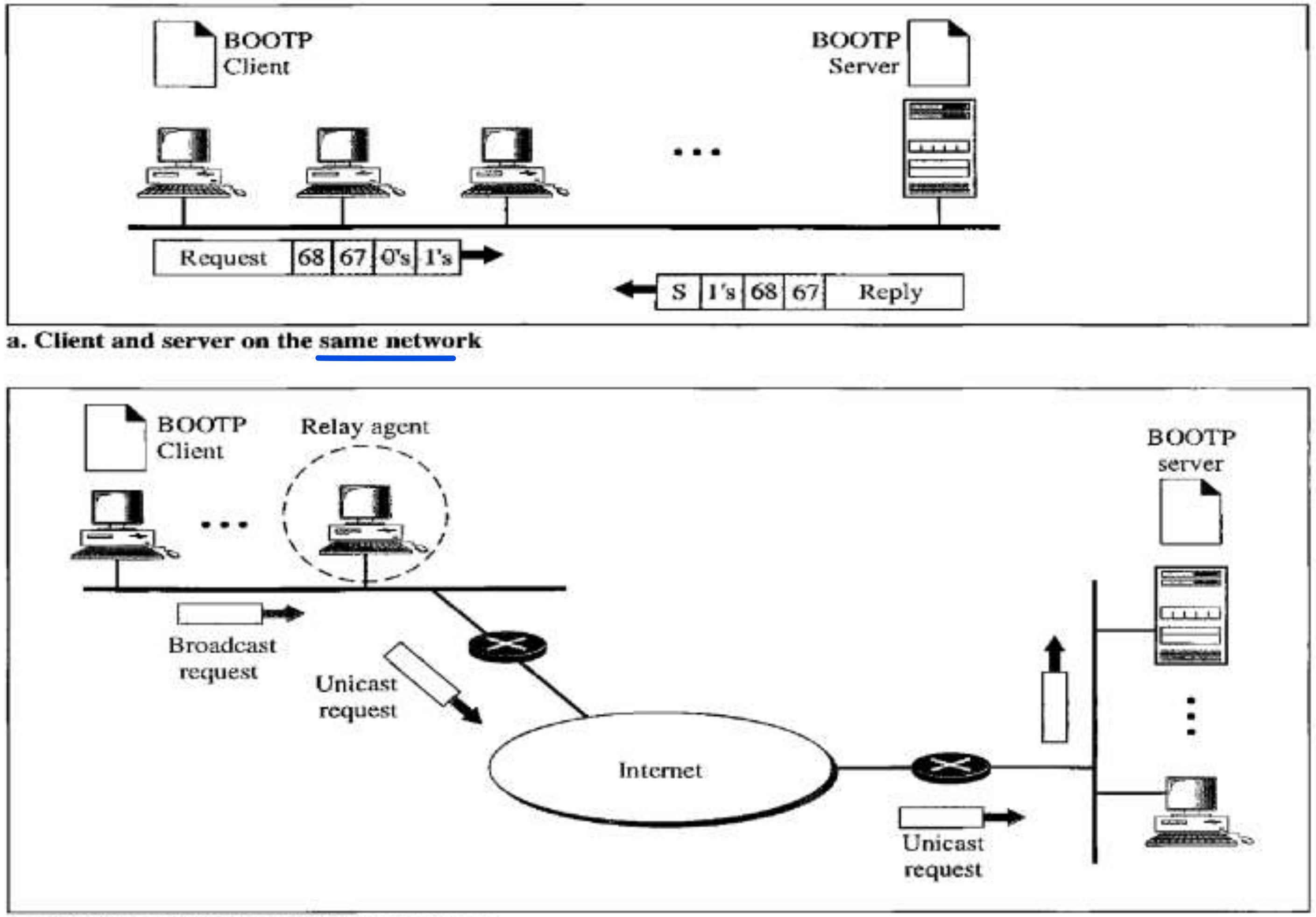
# BOOTP



The **Bootstrap Protocol (BOOTP)** is a client/server protocol designed to provide physical address to logical address mapping. BOOTP is an application layer protocol. The administrator may put the client and the server on the same network or on different

# BootP

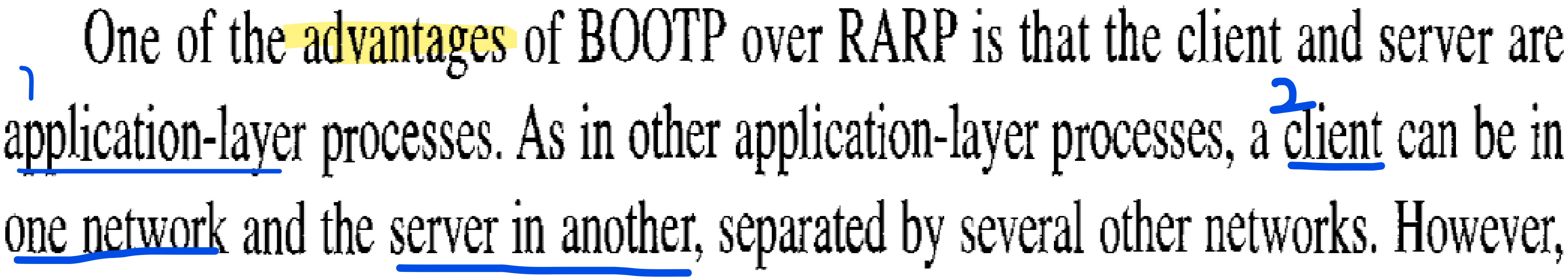
**Figure 21.7** *BOOTP client and server on the same and different network*



**a. Client and server on the same network**

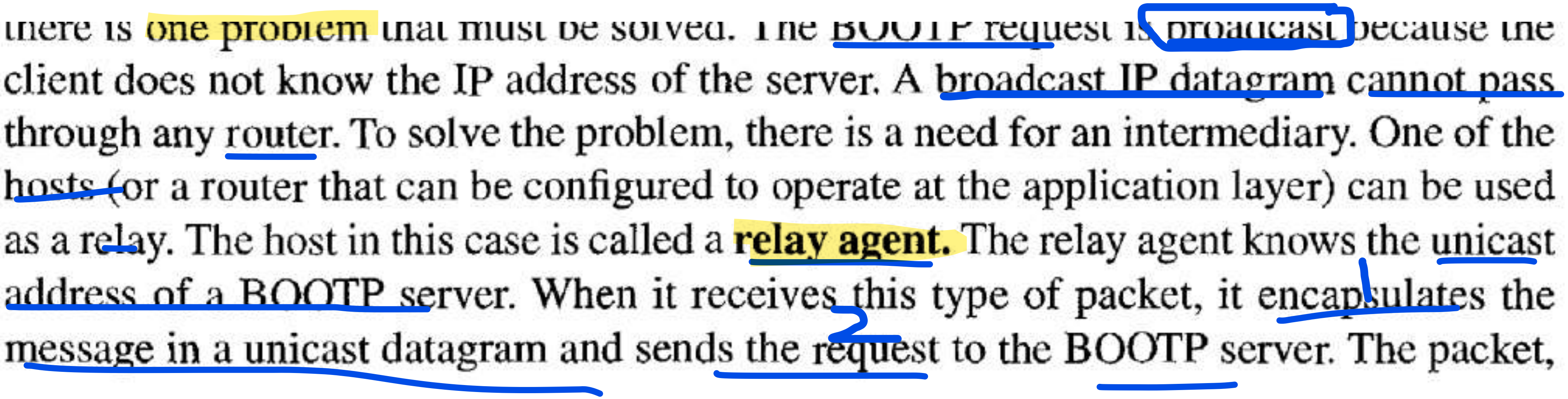
**b. Client and server on different networks**

# BootP



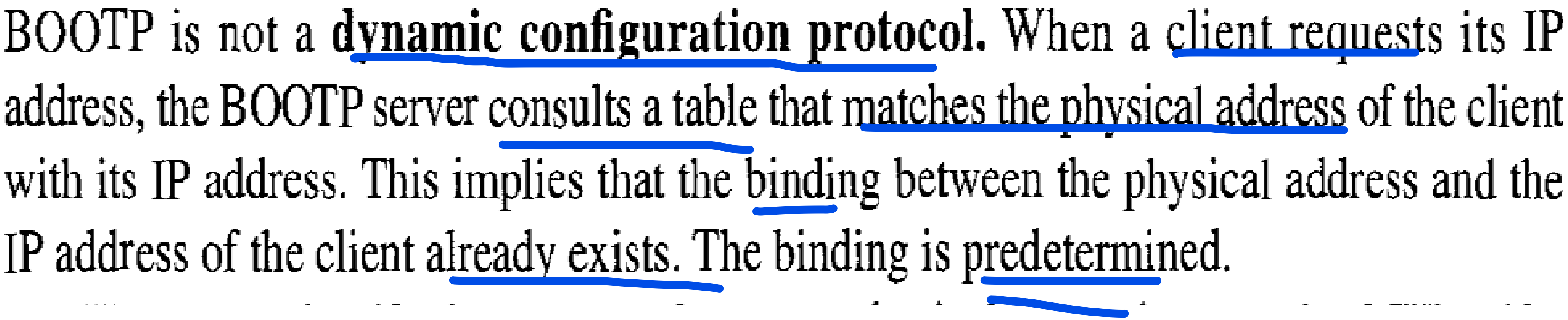
One of the advantages of BOOTP over RARP is that the client and server are 1 application-layer processes. As in other application-layer processes, a 2 client can be in one network and the server in another, separated by several other networks. However,

# BOOTP



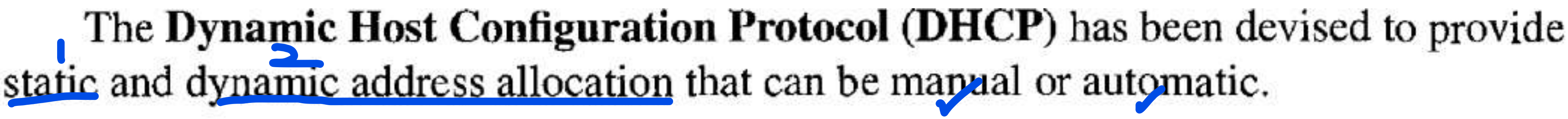
there is one problem that must be solved. The BOOTP request is broadcast because the client does not know the IP address of the server. A broadcast IP datagram cannot pass through any router. To solve the problem, there is a need for an intermediary. One of the hosts (or a router that can be configured to operate at the application layer) can be used as a relay. The host in this case is called a **relay agent.** The relay agent knows the unicast address of a BOOTP server. When it receives this type of packet, it encapsulates 1 the message in a unicast datagram and sends the 2 request to the BOOTP server. The packet,

# DHCP



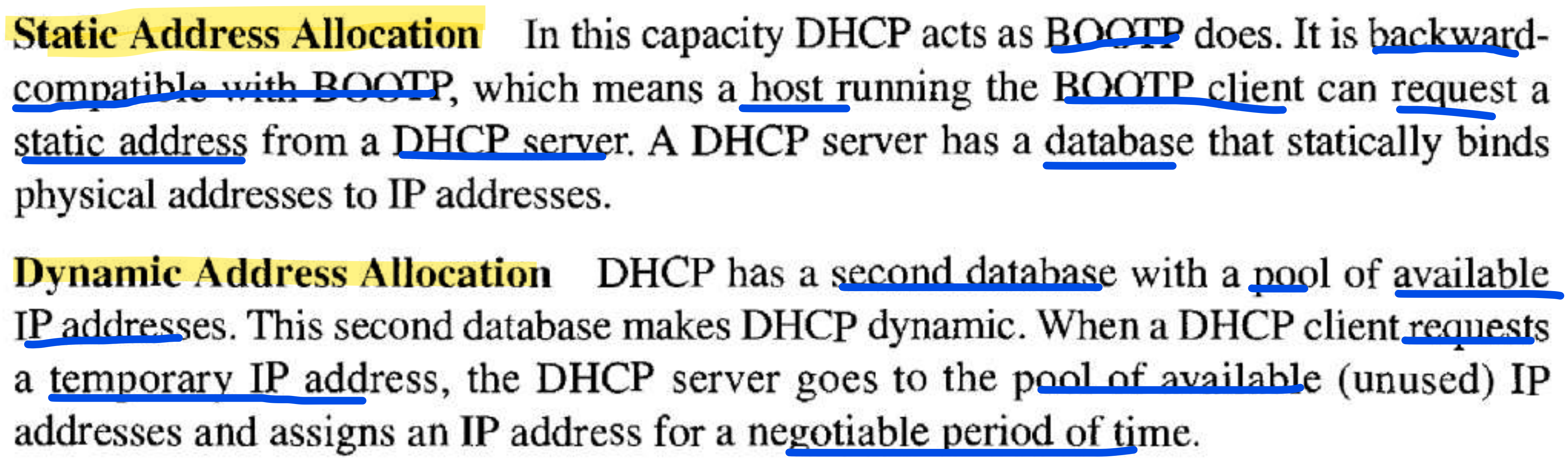
BOOTP is not a **dynamic configuration protocol.** When a client requests its IP address, the BOOTP server consults a table that matches the physical address of the client with its IP address. This implies that the binding between the physical address and the IP address of the client already exists. The binding is predetermined.

# DHCP



The **Dynamic Host Configuration Protocol (DHCP)** has been devised to provide 1 static and 2 dynamic address allocation that can be manual ✓ or automatic ✓.

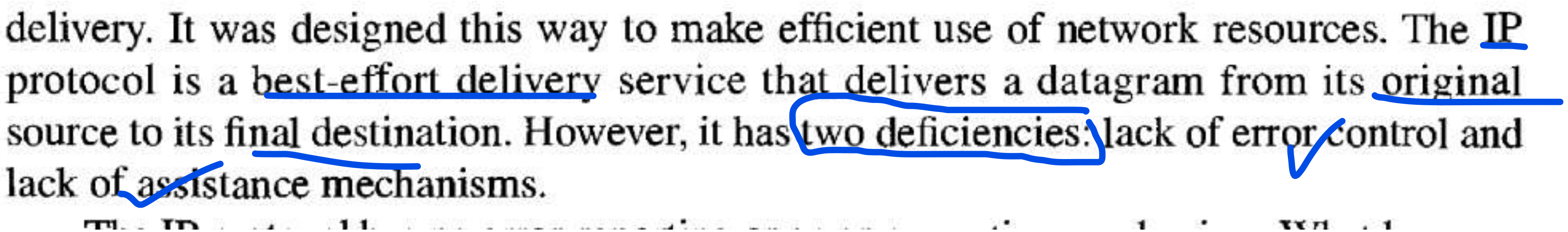
# DHCP



**Static Address Allocation** In this capacity DHCP acts as BOOTP does. It is backward-compatible with BOOTP, which means a host running the BOOTP client can request a static address from a DHCP server. A DHCP server has a database that statically binds physical addresses to IP addresses.

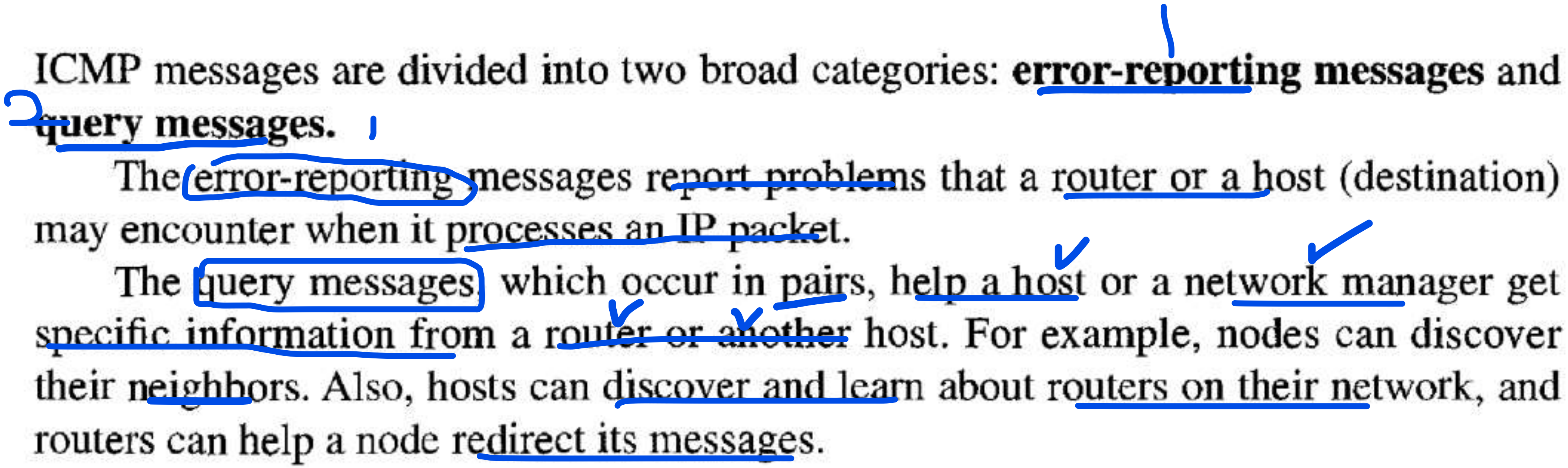
**Dynamic Address Allocation** DHCP has a second database with a pool of available IP addresses. This second database makes DHCP dynamic. When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available (unused) IP addresses and assigns an IP address for a negotiable period of time.

# ICMP



delivery. It was designed this way to make efficient use of network resources. The IP protocol is a best-effort delivery service that delivers a datagram from its original source to its final destination. However, it has two deficiencies: lack of error control and lack of assistance mechanisms.

# ICMP (Types of Messages)

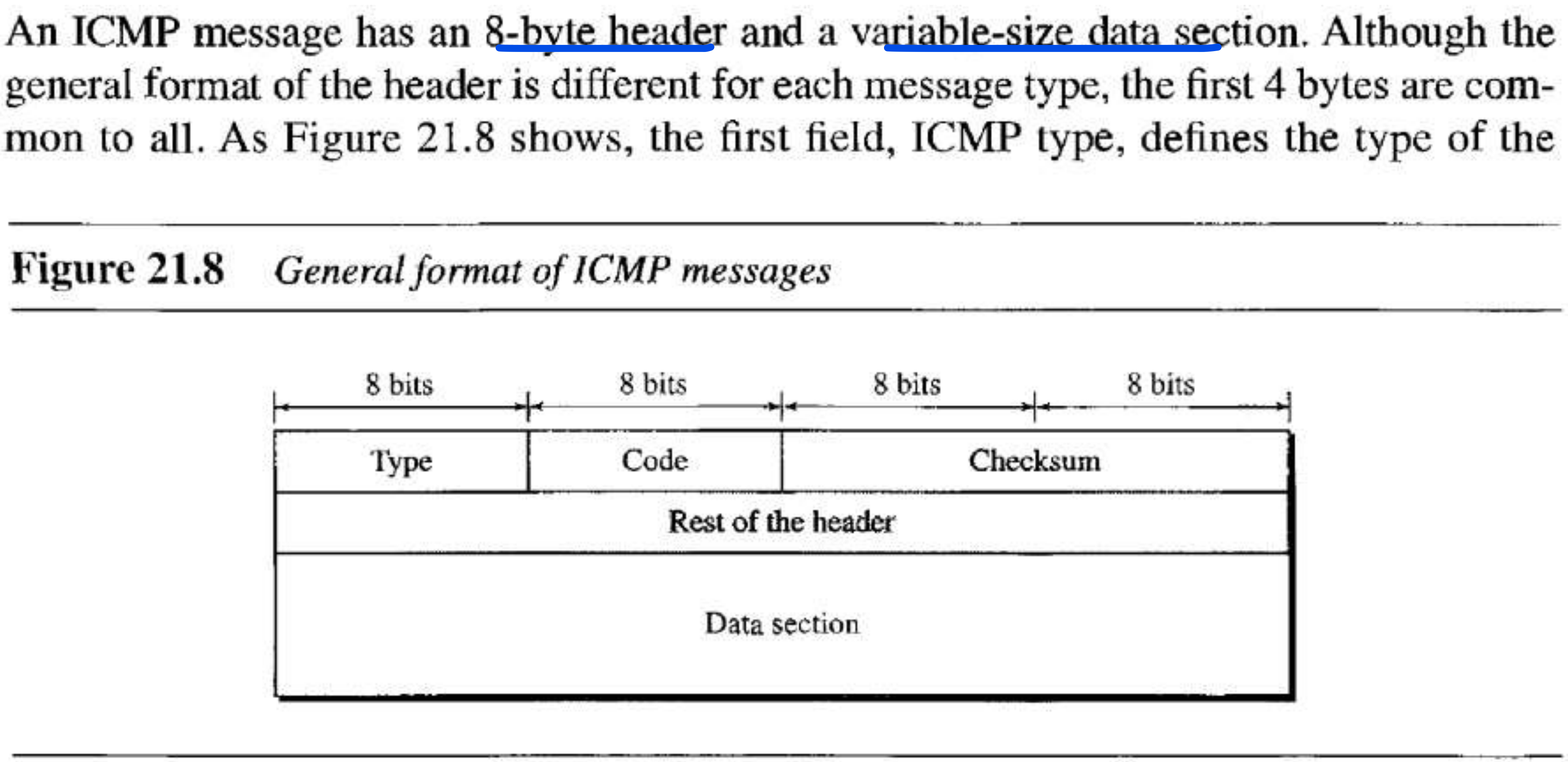


ICMP messages are divided into two broad categories: 1 **error-reporting messages** and 2 **query messages.**

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. Also, hosts can discover and learn about routers on their network, and routers can help a node redirect its messages.

# ICMP (Message format)

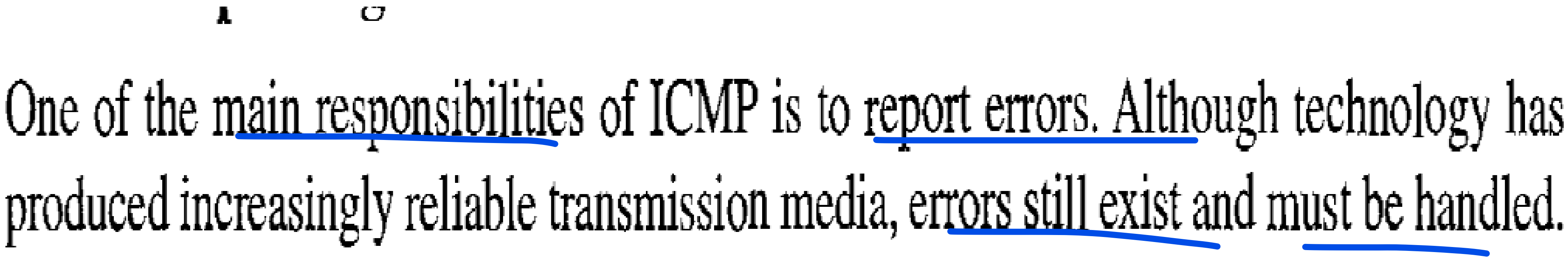


An ICMP message has an 8-byte header and a variable-size data section. Although the general format of the header is different for each message type, the first 4 bytes are common to all. As Figure 21.8 shows, the first field, ICMP type, defines the type of the

**Figure 21.8** *General format of ICMP messages*

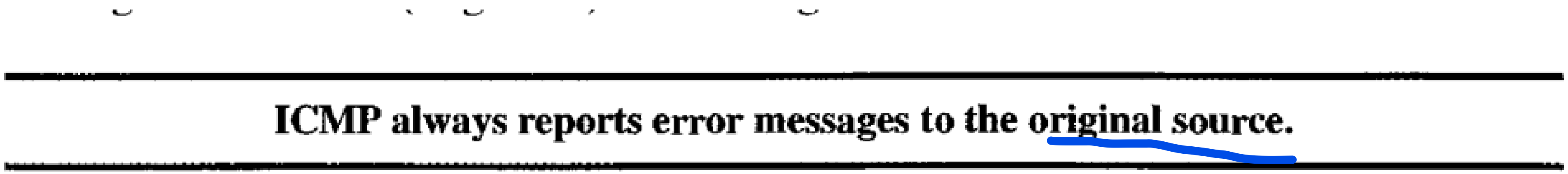
| 8 bits | 8 bits | 8 bits | 8 bits |
|---|---|---|---|
| Type | Code | Checksum | |
| Rest of the header | | | |
| Data section | | | |

# ICMP (Error Reporting)



One of the main responsibilities of ICMP is to report errors. Although technology has produced increasingly reliable transmission media, errors still exist and must be handled.

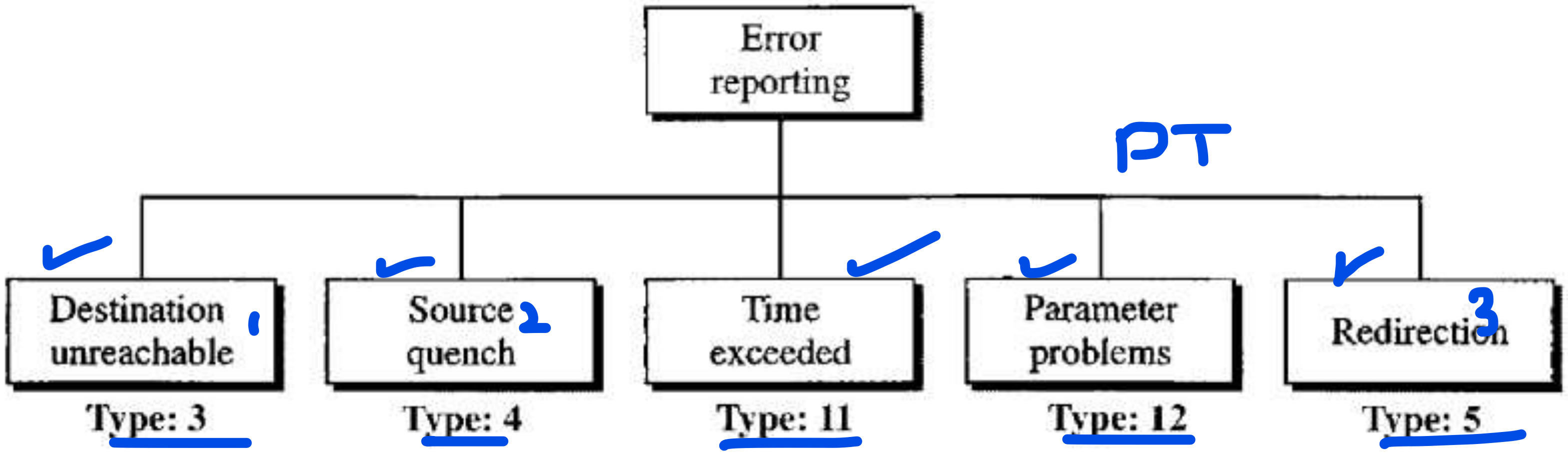
# ICMP (Error Reporting)



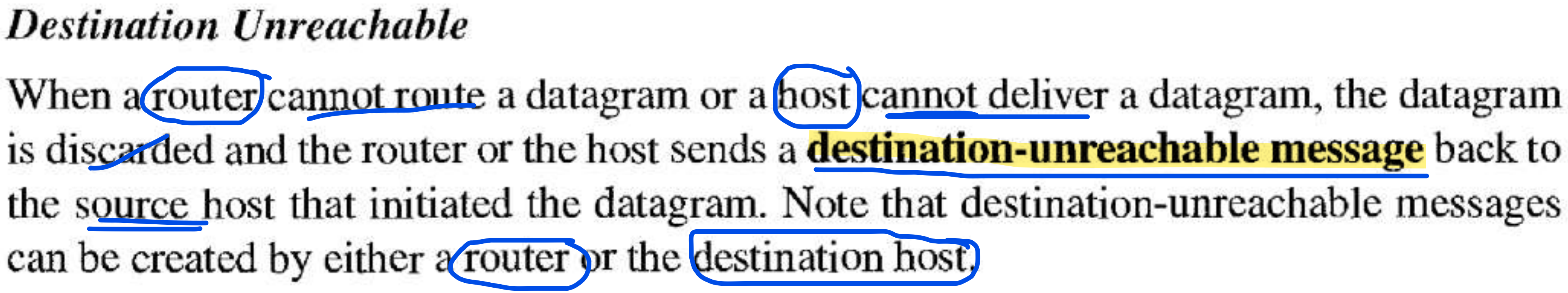
**ICMP always reports error messages to the original source.**

# ICMP (Error reporting messages)

**Figure 21.9** *Error-reporting messages*



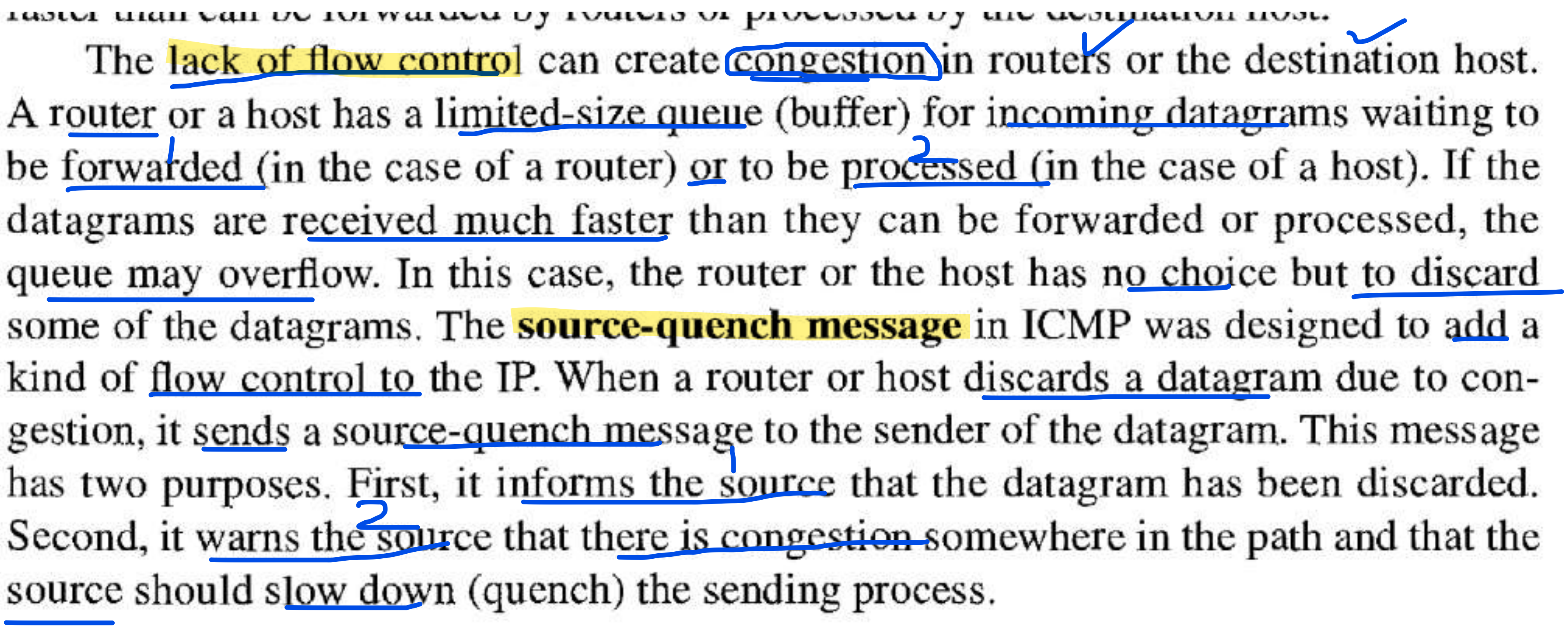
# ICMP message



### *Destination Unreachable*

When a router cannot route a datagram or a host cannot deliver a datagram, the datagram is discarded and the router or the host sends a **destination-unreachable message** back to the source host that initiated the datagram. Note that destination-unreachable messages can be created by either a router or the destination host.

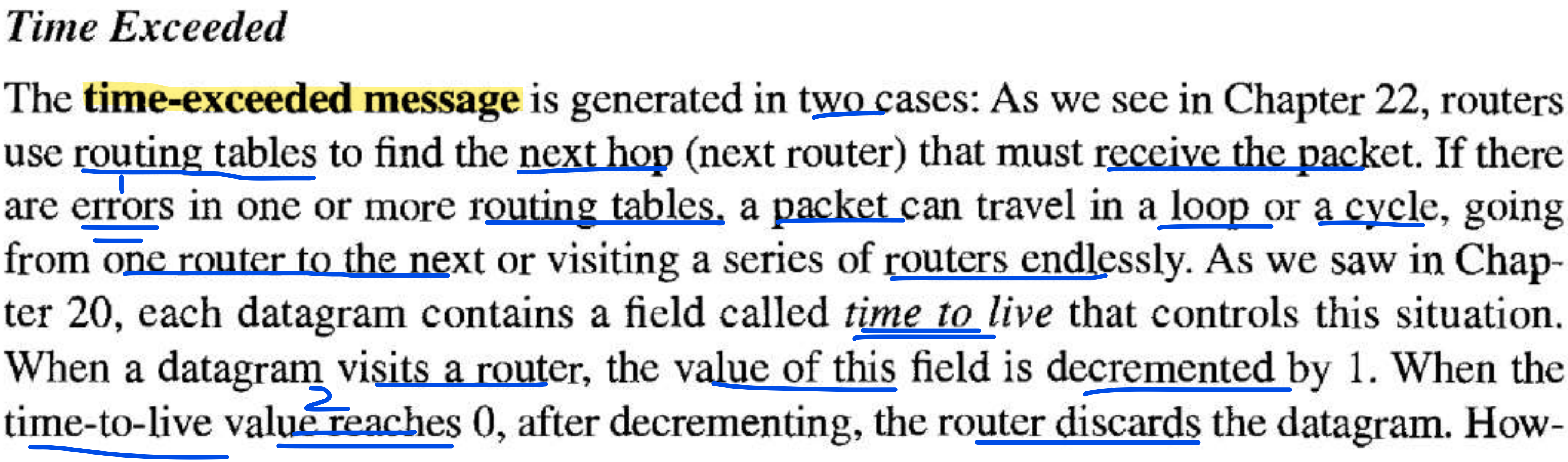
# ICMP message



faster than can be forwarded by routers or processed by the destination host.

The lack of flow control can create congestion in routers or the destination host. A router or a host has a limited-size queue (buffer) for incoming datagrams waiting to be forwarded (in the case of a router) or to be processed (in the case of a host). If the datagrams are received much faster than they can be forwarded or processed, the queue may overflow. In this case, the router or the host has no choice but to discard some of the datagrams. The **source-quench message** in ICMP was designed to add a kind of flow control to the IP. When a router or host discards a datagram due to congestion, it sends a source-quench message to the sender of the datagram. This message has two purposes. First, it informs the source that the datagram has been discarded. Second, it warns the source that there is congestion somewhere in the path and that the source should slow down (quench) the sending process.

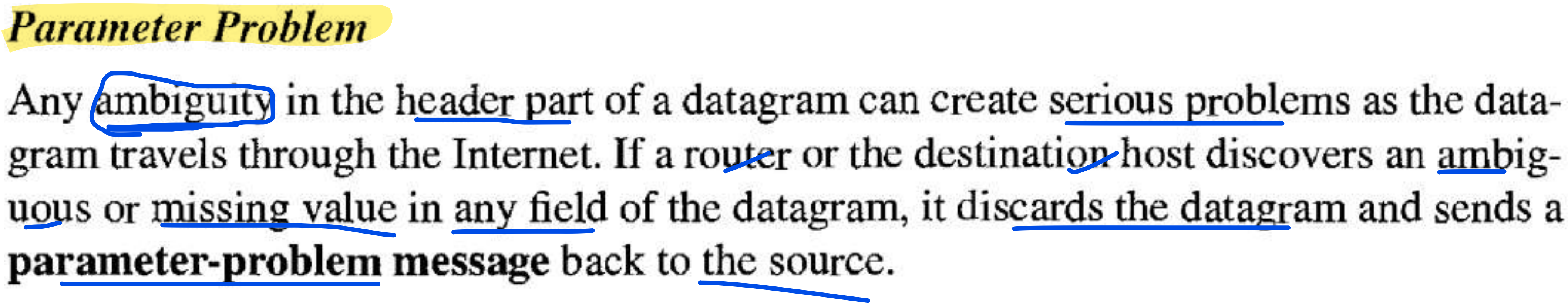
# ICMP message



### *Time Exceeded*

The **time-exceeded message** is generated in two cases: As we see in Chapter 22, routers use routing tables to find the next hop (next router) that must receive the packet. If there are errors in one or more routing tables, a packet can travel in a loop or a cycle, going from one router to the next or visiting a series of routers endlessly. As we saw in Chapter 20, each datagram contains a field called *time to live* that controls this situation. When a datagram visits a router, the value of this field is decremented by 1. When the time-to-live value reaches 0, after decrementing, the router discards the datagram. How-

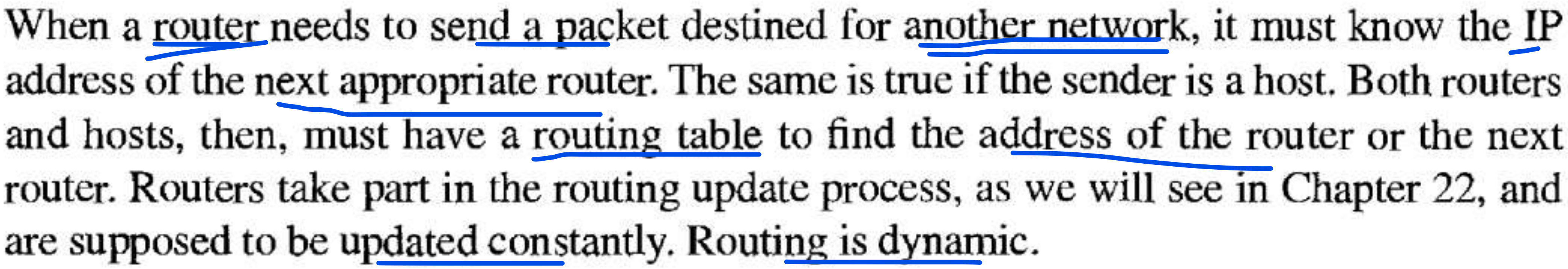
# ICMP message



***Parameter Problem***

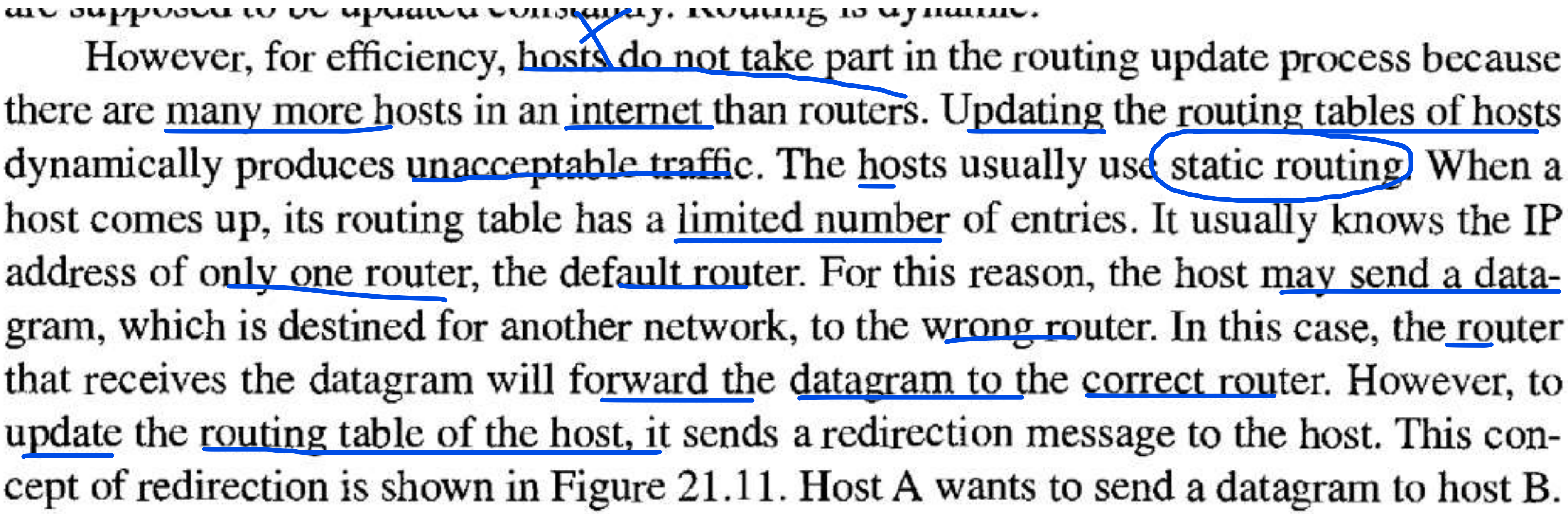
Any ambiguity in the header part of a datagram can create serious problems as the datagram travels through the Internet. If a router or the destination host discovers an ambiguous or missing value in any field of the datagram, it discards the datagram and sends a **parameter-problem message** back to the source.

# ICMP message(Redirection)



When a router needs to send a packet destined for another network, it must know the IP address of the next appropriate router. The same is true if the sender is a host. Both routers and hosts, then, must have a routing table to find the address of the router or the next router. Routers take part in the routing update process, as we will see in Chapter 22, and are supposed to be updated constantly. Routing is dynamic.

# ICMP message(Redirection)

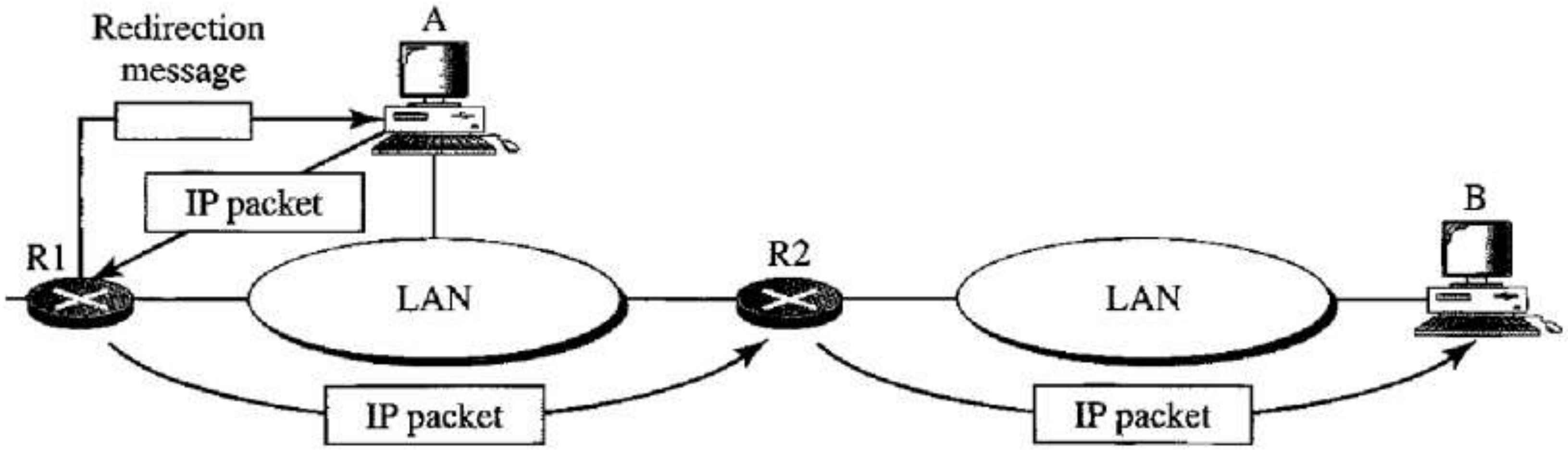


are supposed to be updated constantly. Routing is dynamic.

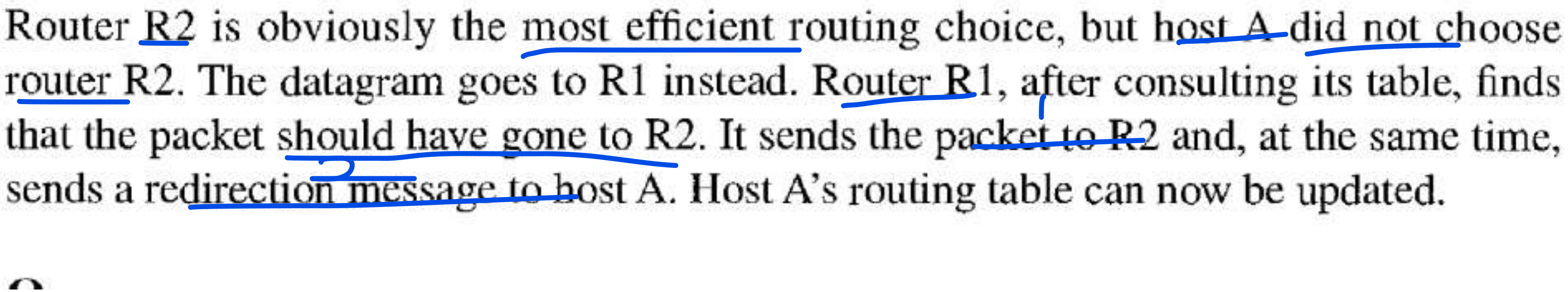
However, for efficiency, hosts do not take part in the routing update process because there are many more hosts in an internet than routers. Updating the routing tables of hosts dynamically produces unacceptable traffic. The hosts usually use static routing. When a host comes up, its routing table has a limited number of entries. It usually knows the IP address of only one router, the default router. For this reason, the host may send a datagram, which is destined for another network, to the wrong router. In this case, the router that receives the datagram will forward the datagram to the correct router. However, to update the routing table of the host, it sends a redirection message to the host. This concept of redirection is shown in Figure 21.11. Host A wants to send a datagram to host B.

# ICMP message(Redirection)

**Figure 21.11** *Redirection concept*

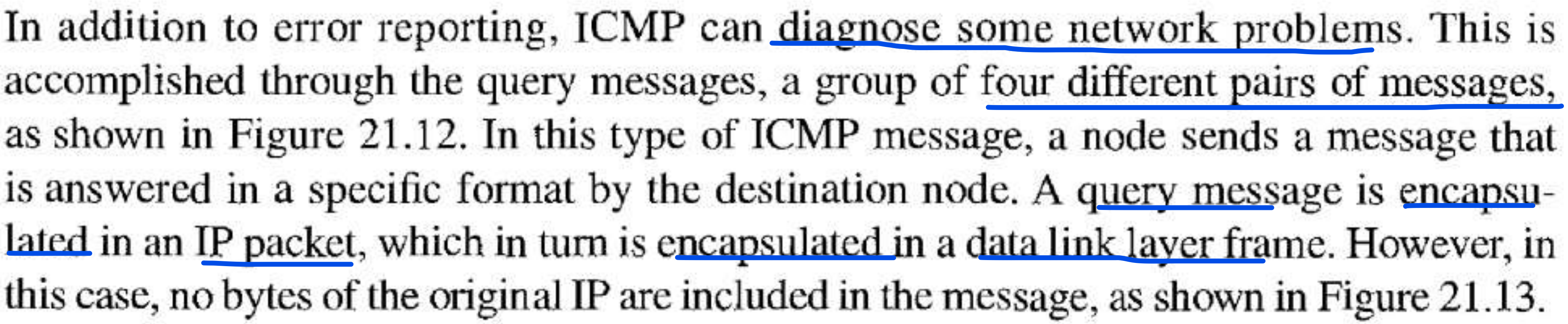


# ICMP message(Redirection)



Router R2 is obviously the most efficient routing choice, but host A did not choose router R2. The datagram goes to R1 instead. Router R1, after consulting its table, finds that the packet should have gone to R2. It sends the packet to R2 and, at the same time, sends a redirection message to host A. Host A's routing table can now be updated.

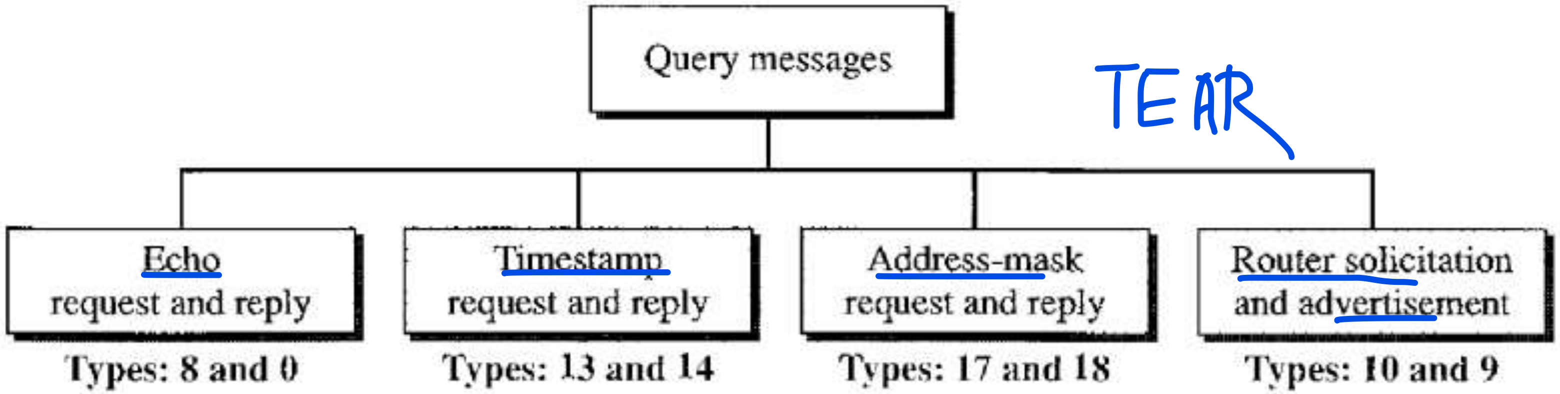
# Query Messages



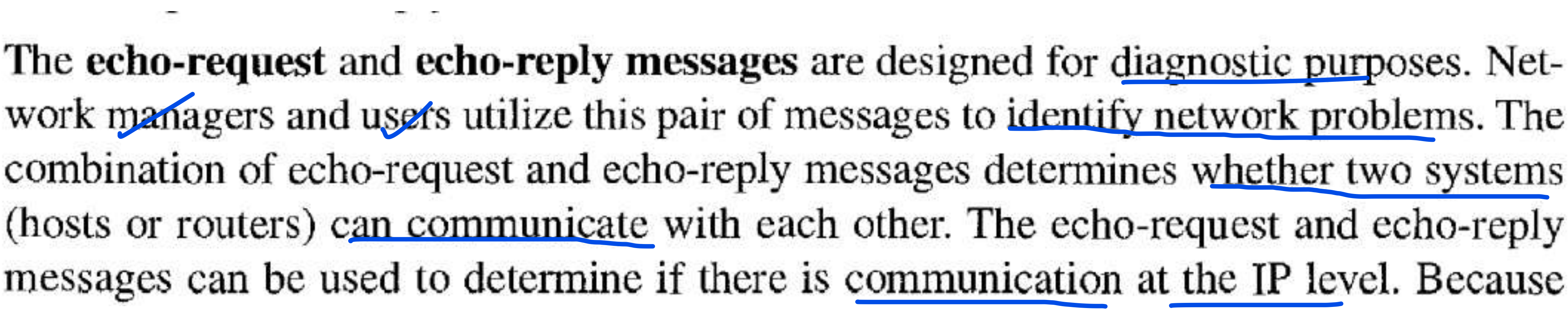
In addition to error reporting, ICMP can diagnose some network problems. This is accomplished through the query messages, a group of four different pairs of messages, as shown in Figure 21.12. In this type of ICMP message, a node sends a message that is answered in a specific format by the destination node. A query message is encapsulated in an IP packet, which in turn is encapsulated in a data link layer frame. However, in this case, no bytes of the original IP are included in the message, as shown in Figure 21.13.

# Query messages

**Figure 21.12** *Query messages*

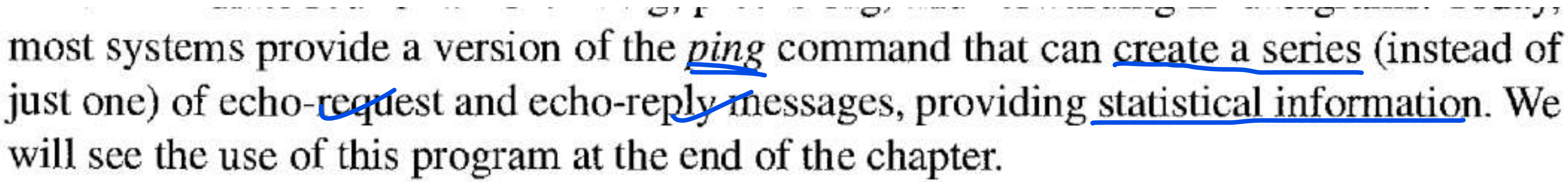


# Echo Requests and reply



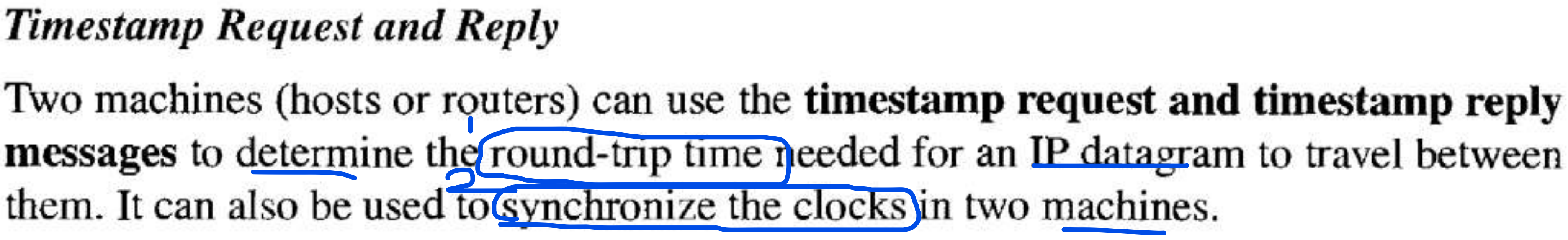
The **echo-request** and **echo-reply messages** are designed for diagnostic purposes. Network managers and users utilize this pair of messages to identify network problems. The combination of echo-request and echo-reply messages determines whether two systems (hosts or routers) can communicate with each other. The echo-request and echo-reply messages can be used to determine if there is communication at the IP level. Because

# Echo Requests and reply



most systems provide a version of the *ping* command that can create a series (instead of just one) of echo-request and echo-reply messages, providing statistical information. We will see the use of this program at the end of the chapter.

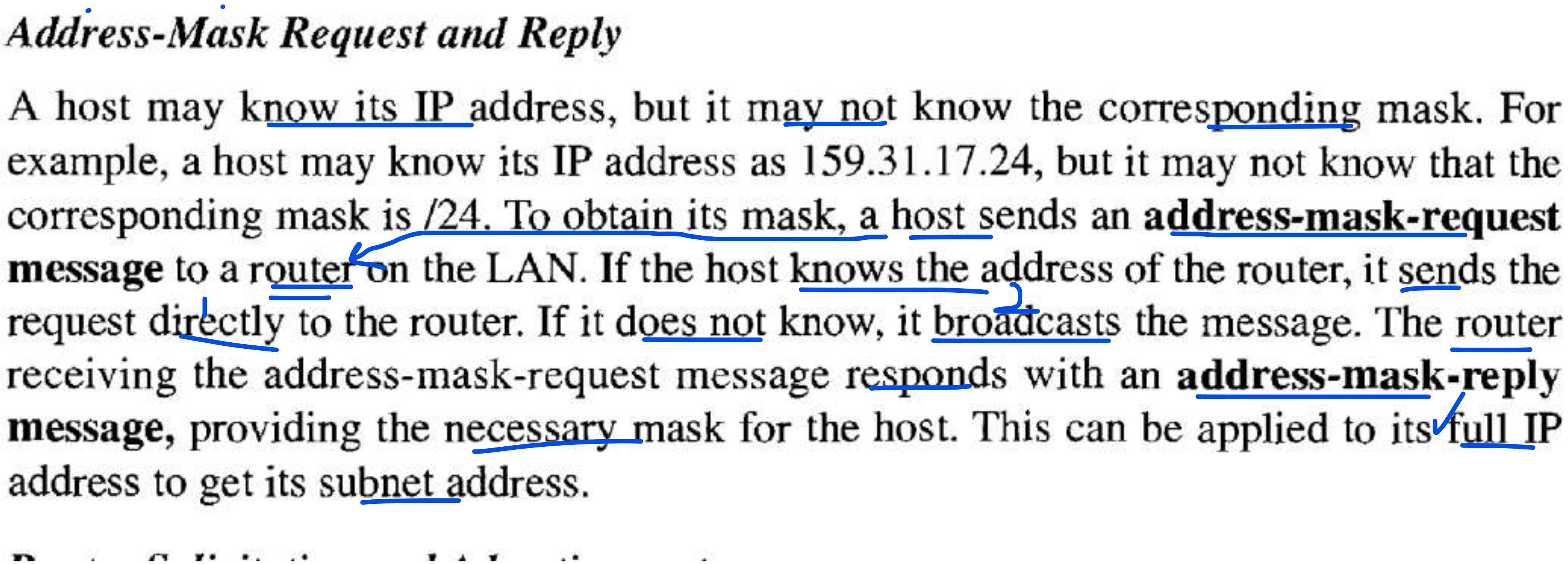
# Timestamp and Reply



### *Timestamp Request and Reply*

Two machines (hosts or routers) can use the **timestamp request and timestamp reply messages** to determine the round-trip time needed for an IP datagram to travel between them. It can also be used to synchronize the clocks in two machines.

# Address-Mask Request and Reply



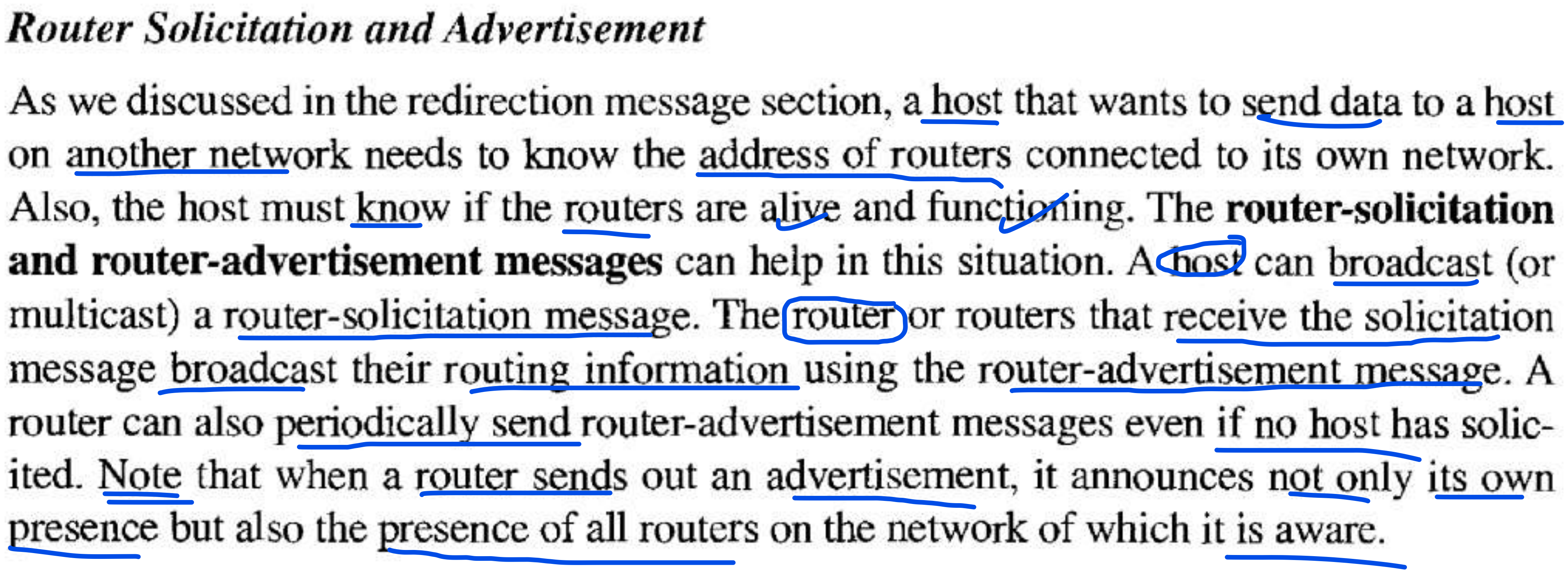
### *Address-Mask Request and Reply*

A host may know its IP address, but it may not know the corresponding mask. For example, a host may know its IP address as 159.31.17.24, but it may not know that the corresponding mask is /24. To obtain its mask, a host sends an **address-mask-request message** to a router on the LAN. If the host knows the address of the router, it sends the request directly to the router. If it does not know, it broadcasts the message. The router receiving the address-mask-request message responds with an **address-mask-reply message,** providing the necessary mask for the host. This can be applied to its full IP address to get its subnet address.

*Router Solicitation and Advertisement*

# Router solicitation and Advertisement

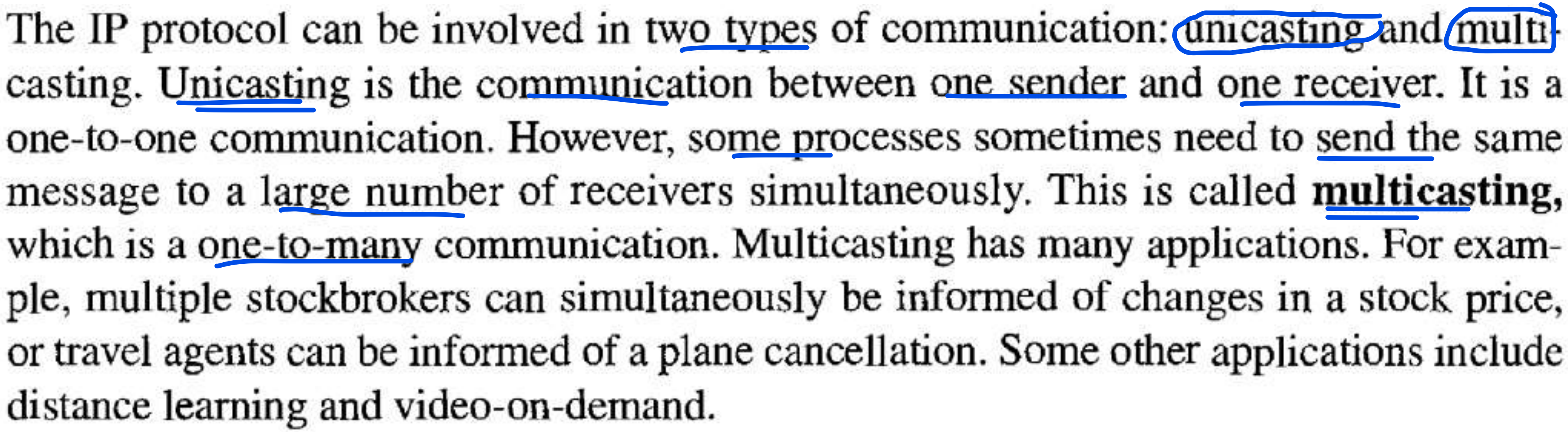
tchr
Stud



### *Router Solicitation and Advertisement*

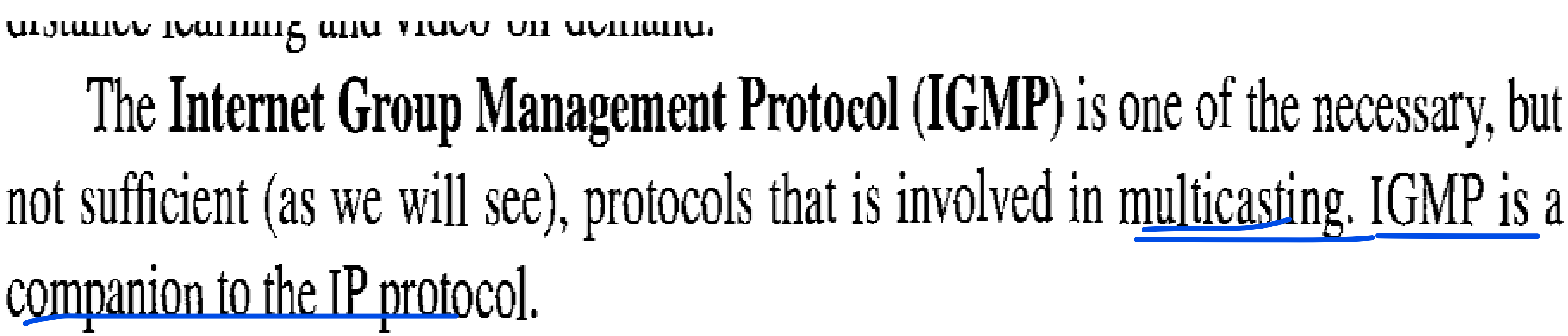
As we discussed in the redirection message section, a host that wants to send data to a host on another network needs to know the address of routers connected to its own network. Also, the host must know if the routers are alive and functioning. The **router-solicitation and router-advertisement messages** can help in this situation. A host can broadcast (or multicast) a router-solicitation message. The router or routers that receive the solicitation message broadcast their routing information using the router-advertisement message. A router can also periodically send router-advertisement messages even if no host has solicited. Note that when a router sends out an advertisement, it announces not only its own presence but also the presence of all routers on the network of which it is aware.

# IGMP



The IP protocol can be involved in two types of communication: unicasting and multicasting. Unicasting is the communication between one sender and one receiver. It is a one-to-one communication. However, some processes sometimes need to send the same message to a large number of receivers simultaneously. This is called **multicasting,** which is a one-to-many communication. Multicasting has many applications. For example, multiple stockbrokers can simultaneously be informed of changes in a stock price, or travel agents can be informed of a plane cancellation. Some other applications include distance learning and video-on-demand.

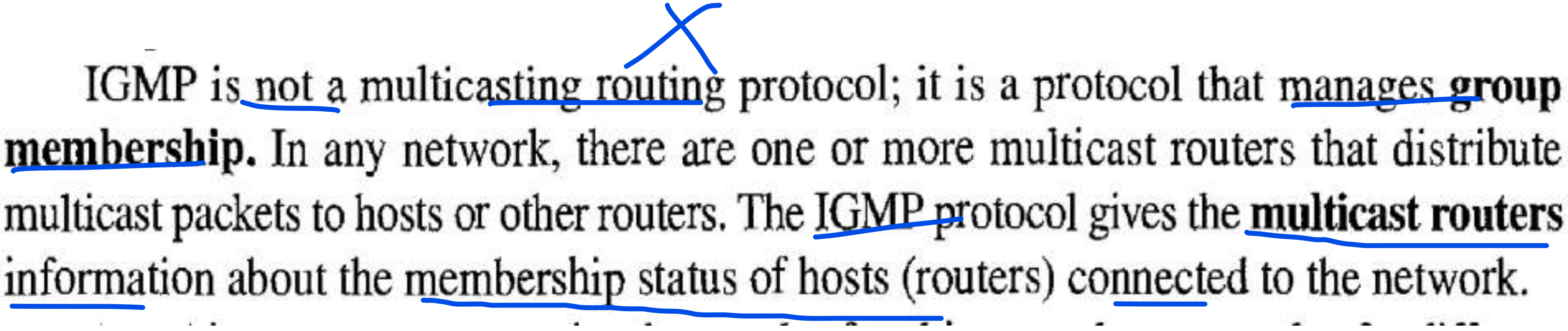
# IGMP



distance learning and video on demand.

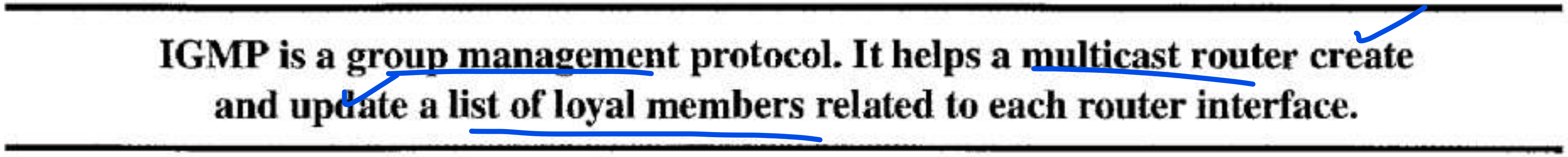
The **Internet Group Management Protocol (IGMP)** is one of the necessary, but not sufficient (as we will see), protocols that is involved in multicasting. IGMP is a companion to the IP protocol.

# IGMP



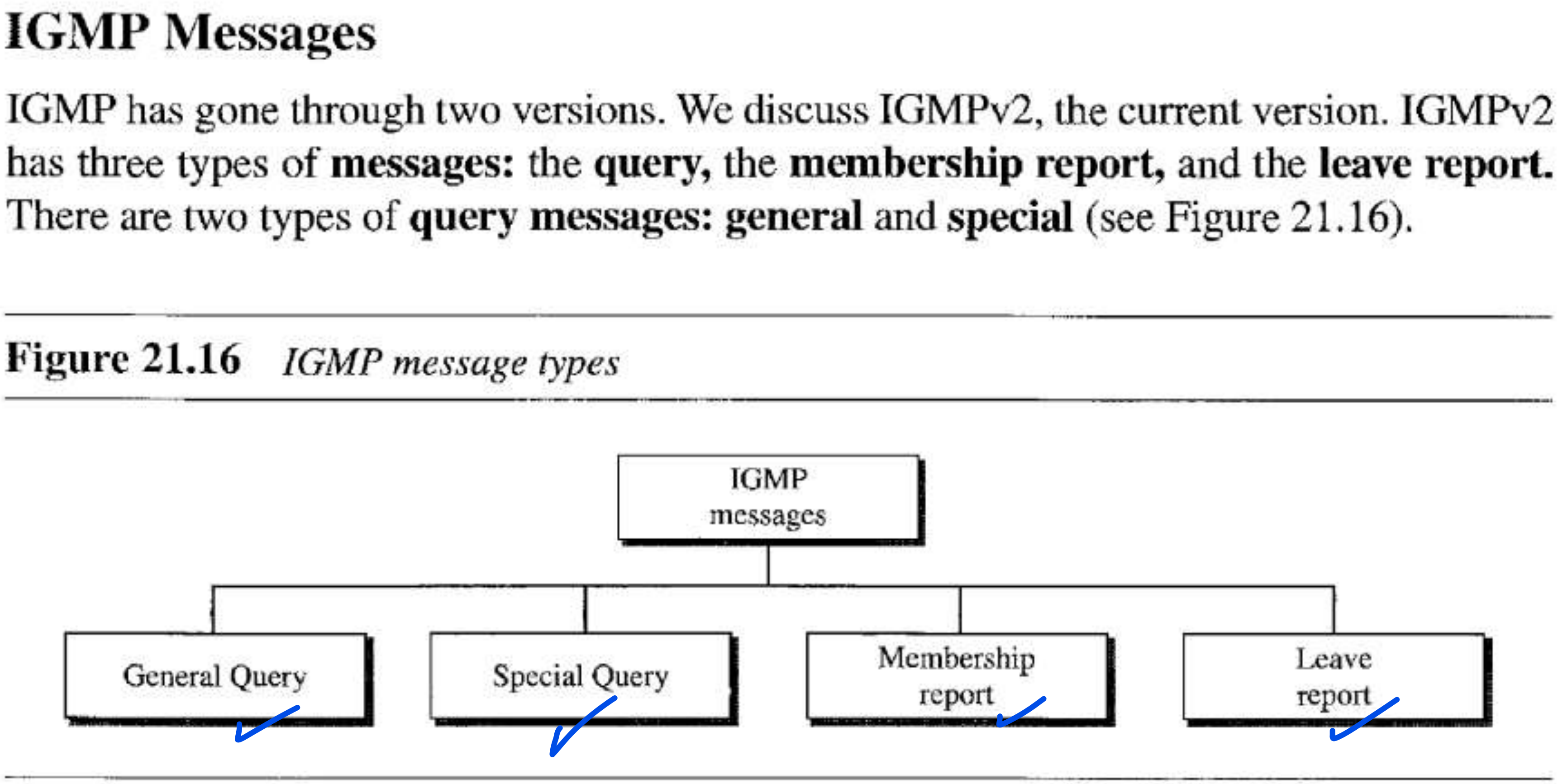
IGMP is not a multicasting routing protocol; it is a protocol that manages **group membership.** In any network, there are one or more multicast routers that distribute multicast packets to hosts or other routers. The IGMP protocol gives the **multicast routers** information about the membership status of hosts (routers) connected to the network.

# IGMP



**IGMP is a group management protocol. It helps a multicast router create and update a list of loyal members related to each router interface.**

# IGMP



## IGMP Messages

IGMP has gone through two versions. We discuss IGMPv2, the current version. IGMPv2 has three types of **messages:** the **query,** the **membership report,** and the **leave report.** There are two types of **query messages: general** and **special** (see Figure 21.16).

**Figure 21.16** *IGMP message types*

IGMP messages

General Query ✓

Special Query ✓

Membership report ✓

Leave report ✓

# IGMP message format

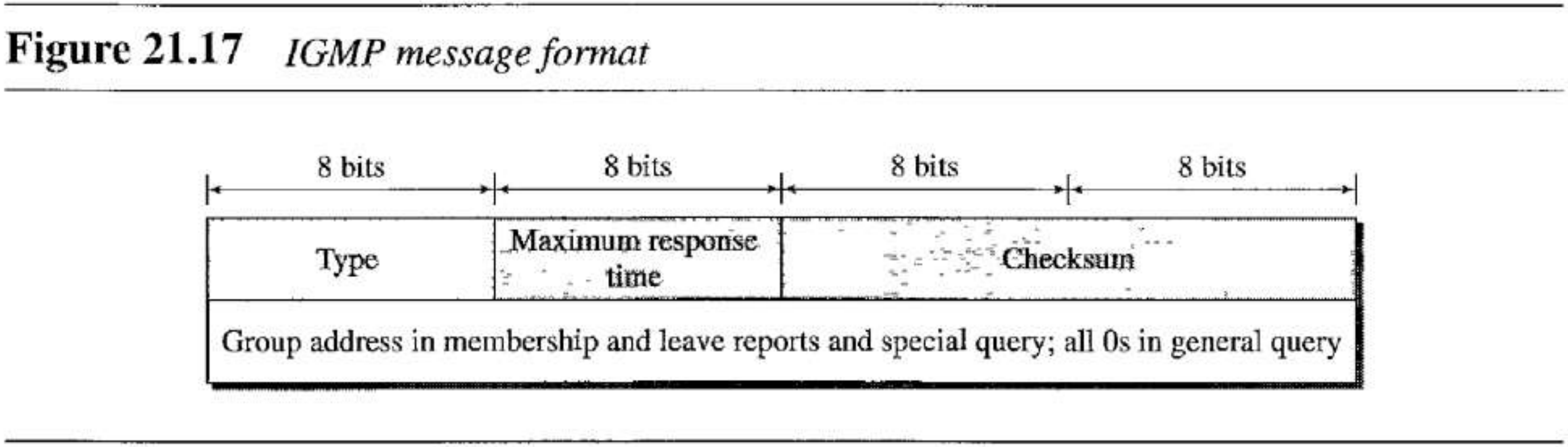
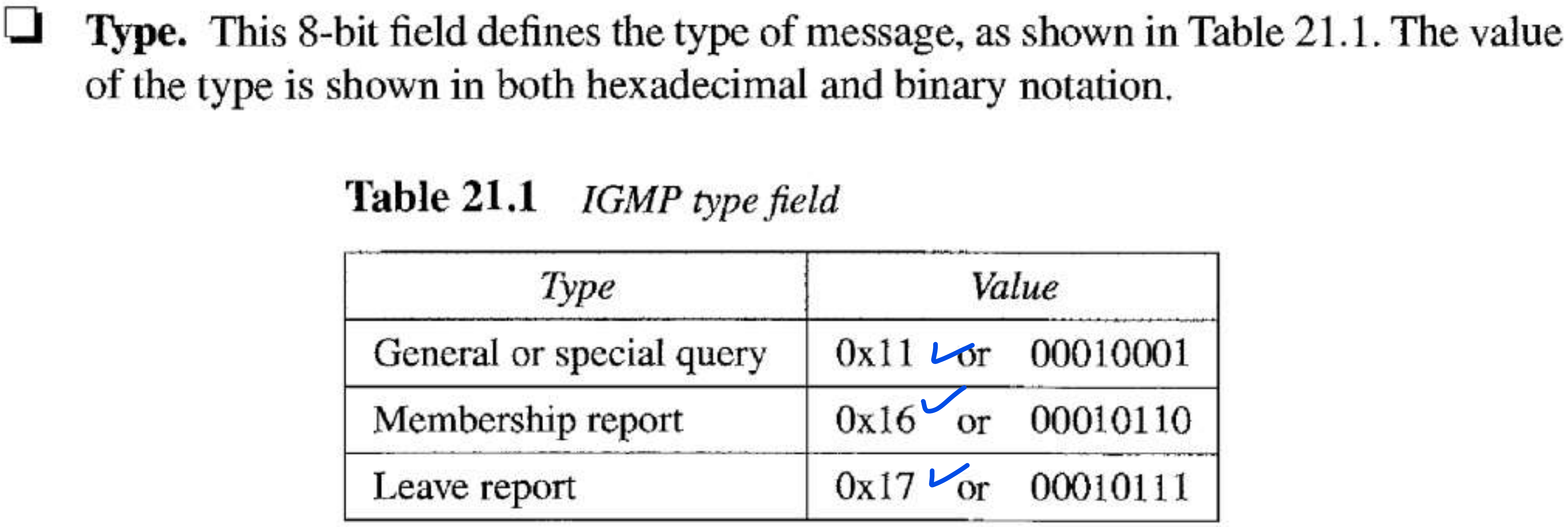


**Figure 21.17** *IGMP message format*

| 8 bits | 8 bits | 8 bits | 8 bits |
|---|---|---|---|
| Type | Maximum response time | Checksum | |
| Group address in membership and leave reports and special query; all 0s in general query | | | |

# IGMP message format

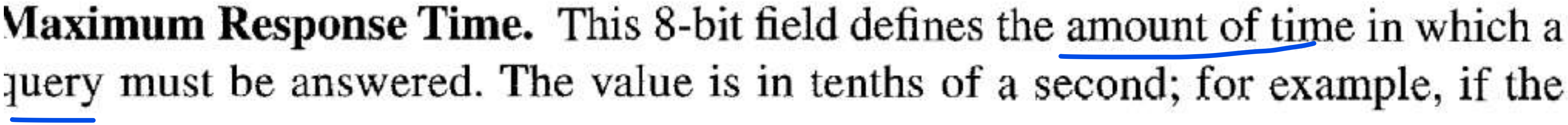


❑ **Type.** This 8-bit field defines the type of message, as shown in Table 21.1. The value of the type is shown in both hexadecimal and binary notation.

**Table 21.1** *IGMP type field*

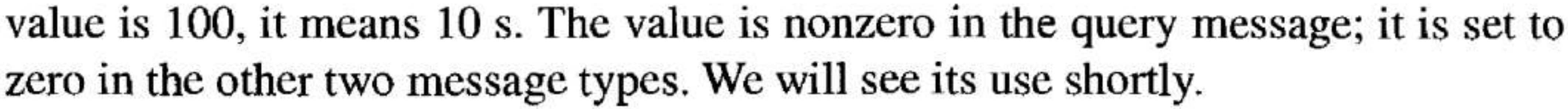
| *Type* | *Value* |
|---|---|
| General or special query | 0x11 ✓ or 00010001 |
| Membership report | 0x16 ✓ or 00010110 |
| Leave report | 0x17 ✓ or 00010111 |

# IGMP message format



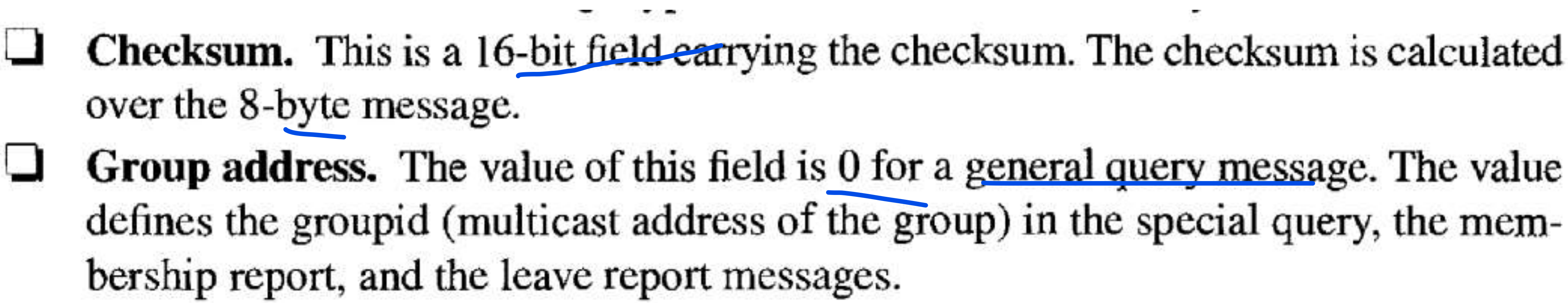
**Maximum Response Time.** This 8-bit field defines the amount of time in which a query must be answered. The value is in tenths of a second; for example, if the

# IGMP message format



value is 100, it means 10 s. The value is nonzero in the query message; it is set to zero in the other two message types. We will see its use shortly.

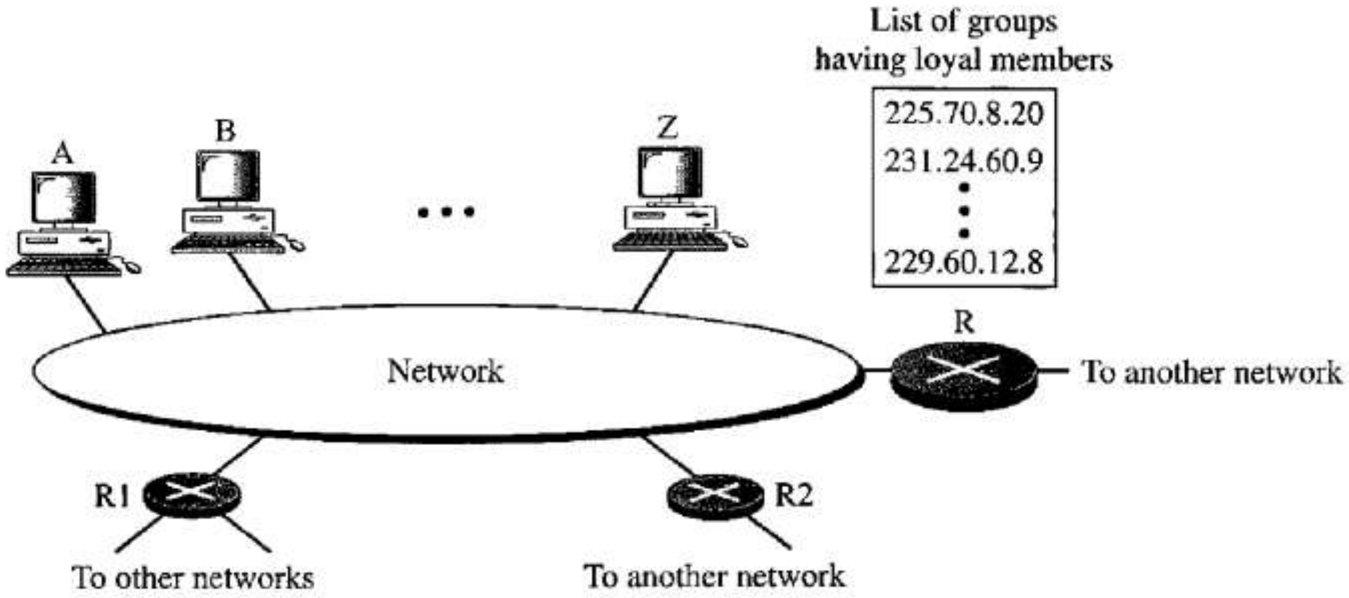
# IGMP message format



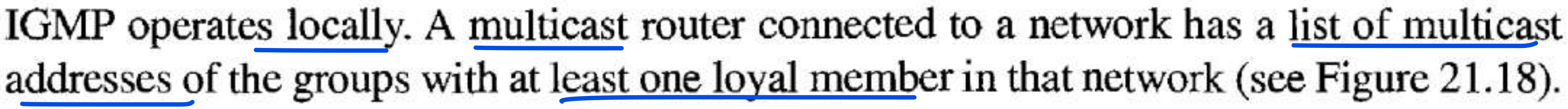
- ❑ **Checksum.** This is a 16-bit field carrying the checksum. The checksum is calculated over the 8-byte message.
- ❑ **Group address.** The value of this field is 0 for a general query message. The value defines the groupid (multicast address of the group) in the special query, the membership report, and the leave report messages.

# IGMP

**Figure 21.18** *IGMP operation*

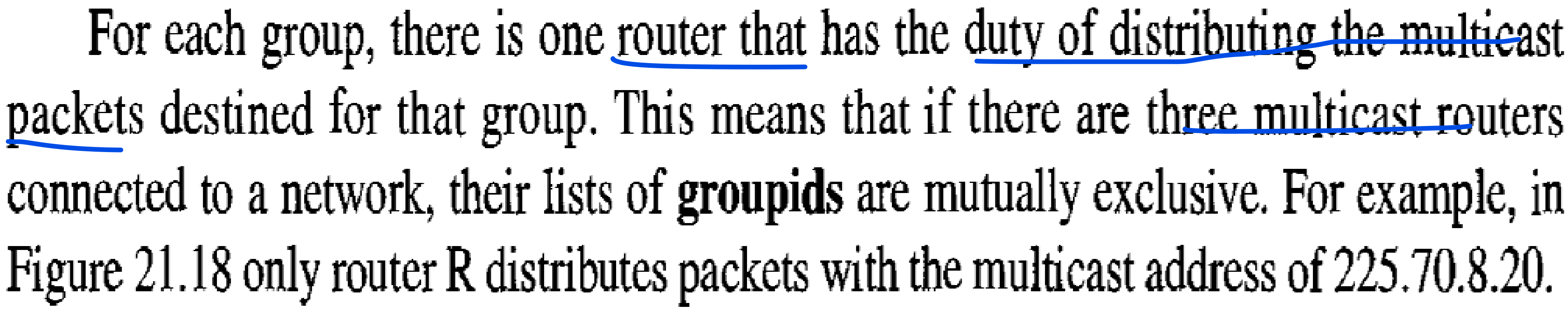


# IGMP (Operation)



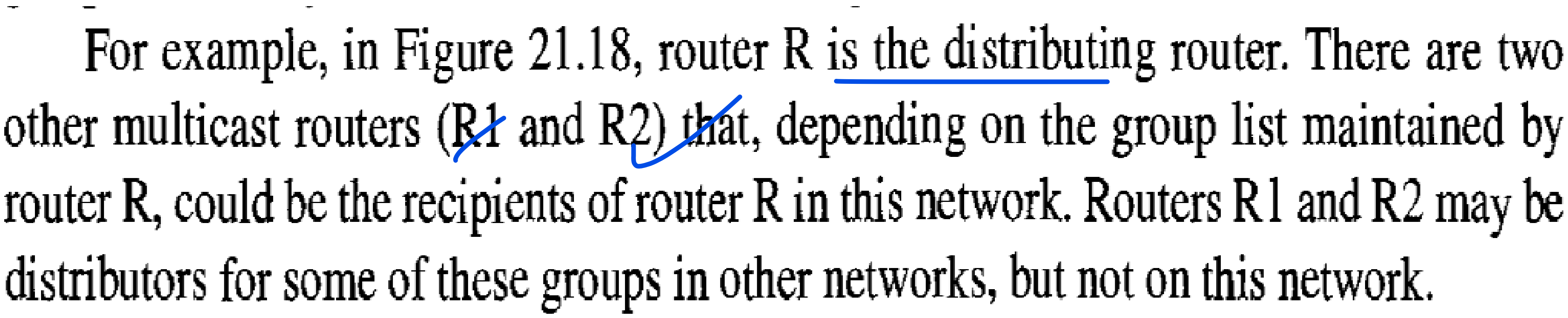
IGMP operates locally. A multicast router connected to a network has a list of multicast addresses of the groups with at least one loyal member in that network (see Figure 21.18).

# IGMP(Operation)



For each group, there is one router that has the duty of distributing the multicast packets destined for that group. This means that if there are three multicast routers connected to a network, their lists of **groupids** are mutually exclusive. For example, in Figure 21.18 only router R distributes packets with the multicast address of 225.70.8.20.

# IGMP(Operation)



For example, in Figure 21.18, router R is the distributing router. There are two other multicast routers (R1 and R2) that, depending on the group list maintained by router R, could be the recipients of router R in this network. Routers R1 and R2 may be distributors for some of these groups in other networks, but not on this network.

THANK YOU