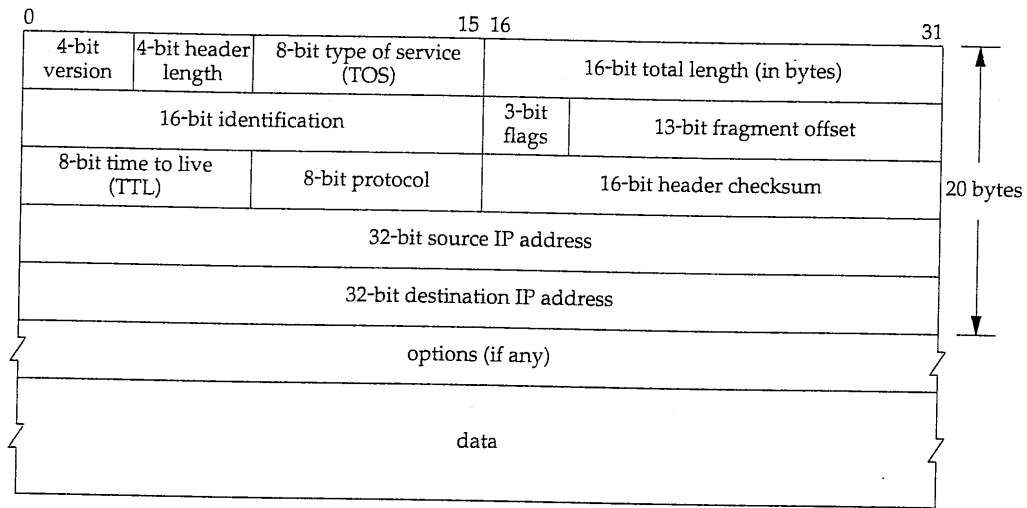
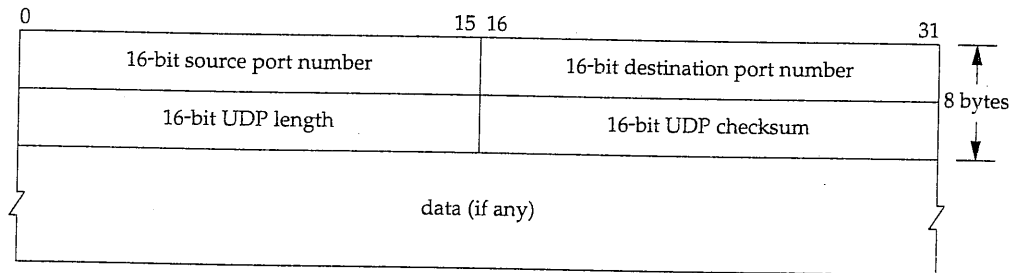


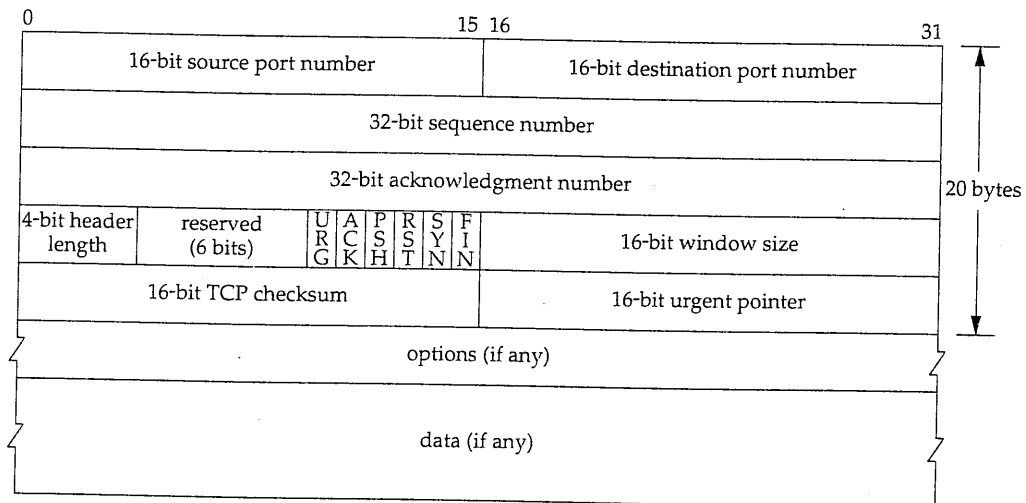
IP Header



UDP Header



TCP Header



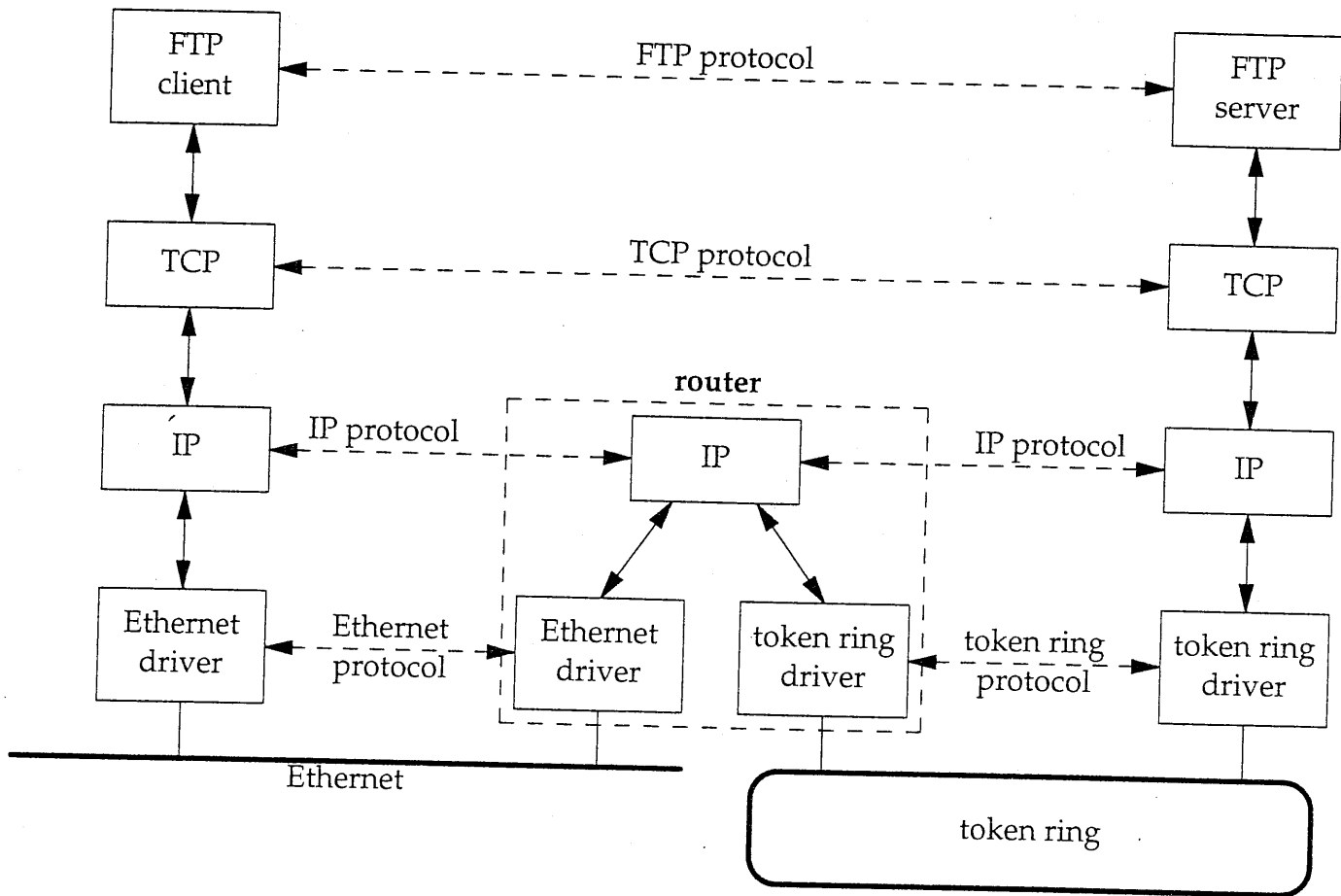


Figure 1.3 Two networks connected with a router.

In the TCP/IP protocol suite the network layer, IP, provides an unreliable service. That is, it does its best job of moving a packet from its source to its final destination, but there are no guarantees. TCP, on the other hand, provides a reliable transport layer using the unreliable service of IP. To provide this service, TCP performs timeout and retransmission, sends and receives end-to-end acknowledgments, and so on. The transport layer and the network layer have distinct responsibilities.

A router, by definition, has two or more network interface layers (since it connects two or more networks). Any system with multiple interfaces is called *multihomed*. A host can also be multihomed but unless it specifically forwards packets from one interface to another, it is not called a router. Also, routers need not be special hardware

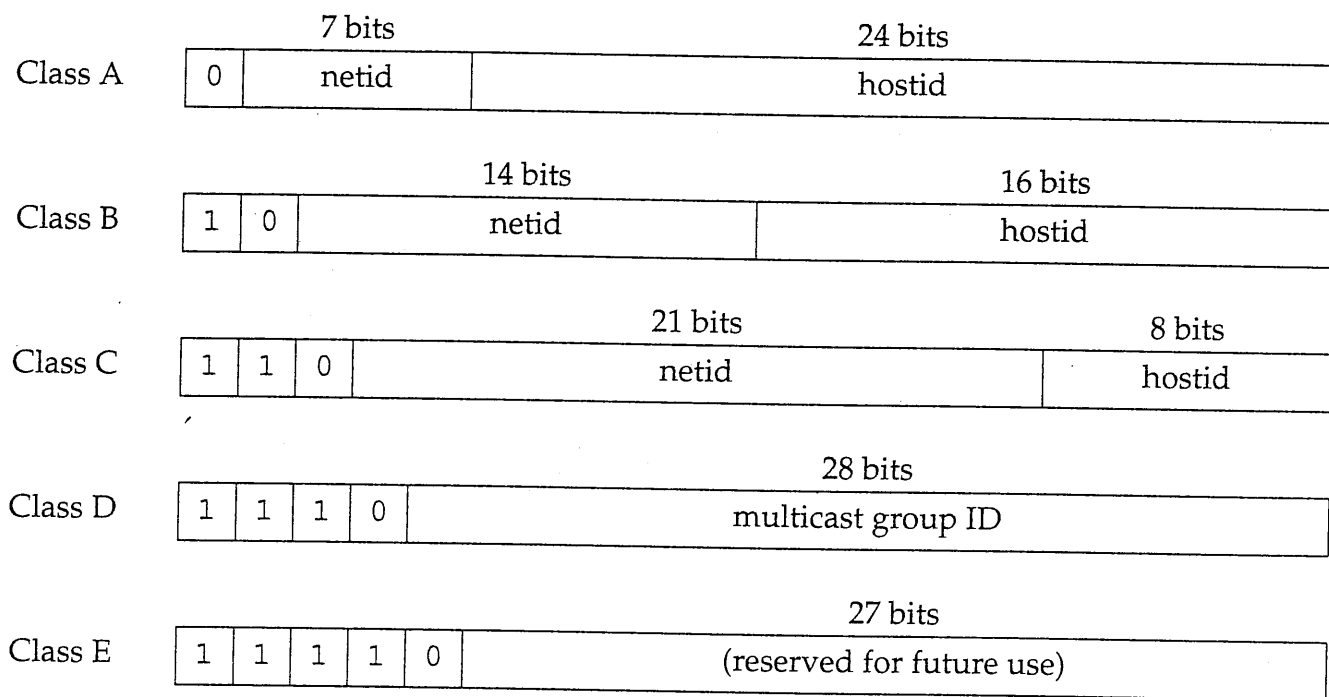


Figure 1.5 The five different classes of Internet addresses.

Class	Range
A	0.0.0.0 to 127.255.255.255
B	128.0.0.0 to 191.255.255.255
C	192.0.0.0 to 223.255.255.255
D	224.0.0.0 to 239.255.255.255
E	240.0.0.0 to 247.255.255.255

Figure 1.6 Ranges for different classes of IP addresses.

worth reiterating that a multihomed host will have multiple IP addresses: one per
 ar
 nce every interface on an internet must have a unique IP address, there must be
 central authority for allocating these addresses for networks connected to the
 wide Internet. That authority is the *Internet Network Information Center*, called the
 IIC. The IIC assigns IP addresses to all IP addresses on the Internet.

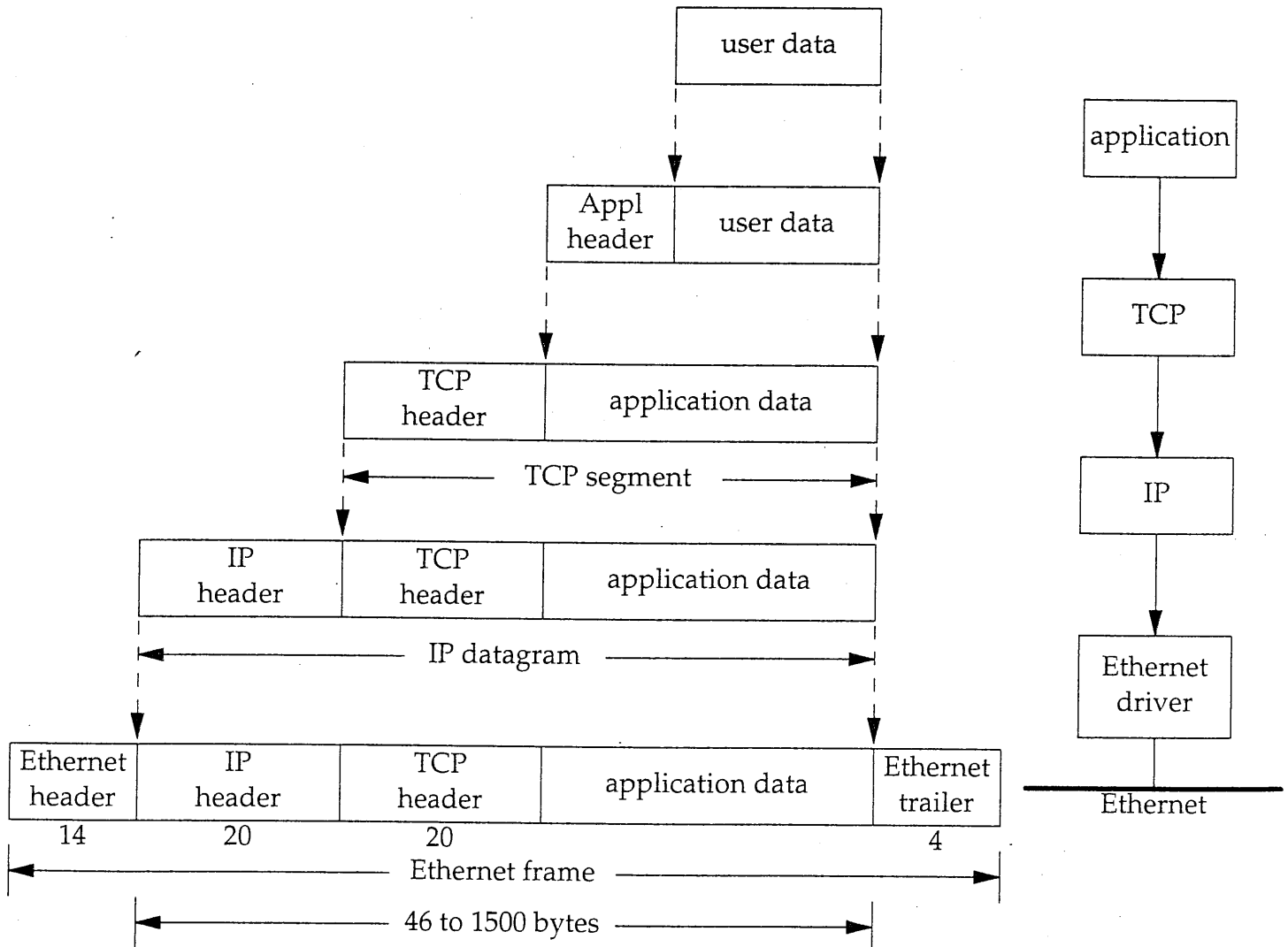


Figure 1.7 Encapsulation of data as it goes down the protocol stack.

Recall from Figure 1.4 (p. 6) that TCP, UDP, ICMP, and IGMP all send data to IP. IP must add some type of identifier to the IP header that it generates, to indicate the layer to which the data belongs. IP handles this by storing an 8-bit value in its header called the *protocol* field. A value of 1 is for ICMP, 2 is for IGMP, 6 indicates TCP, and 17 is for

Demultiplexing

When an Ethernet frame is received at the destination host it starts its way up the protocol stack and all the headers are removed by the appropriate protocol box. Each protocol box looks at certain identifiers in its header to determine which box in the next upper layer receives the data. This is called *demultiplexing*. Figure 1.8 shows how this takes place.

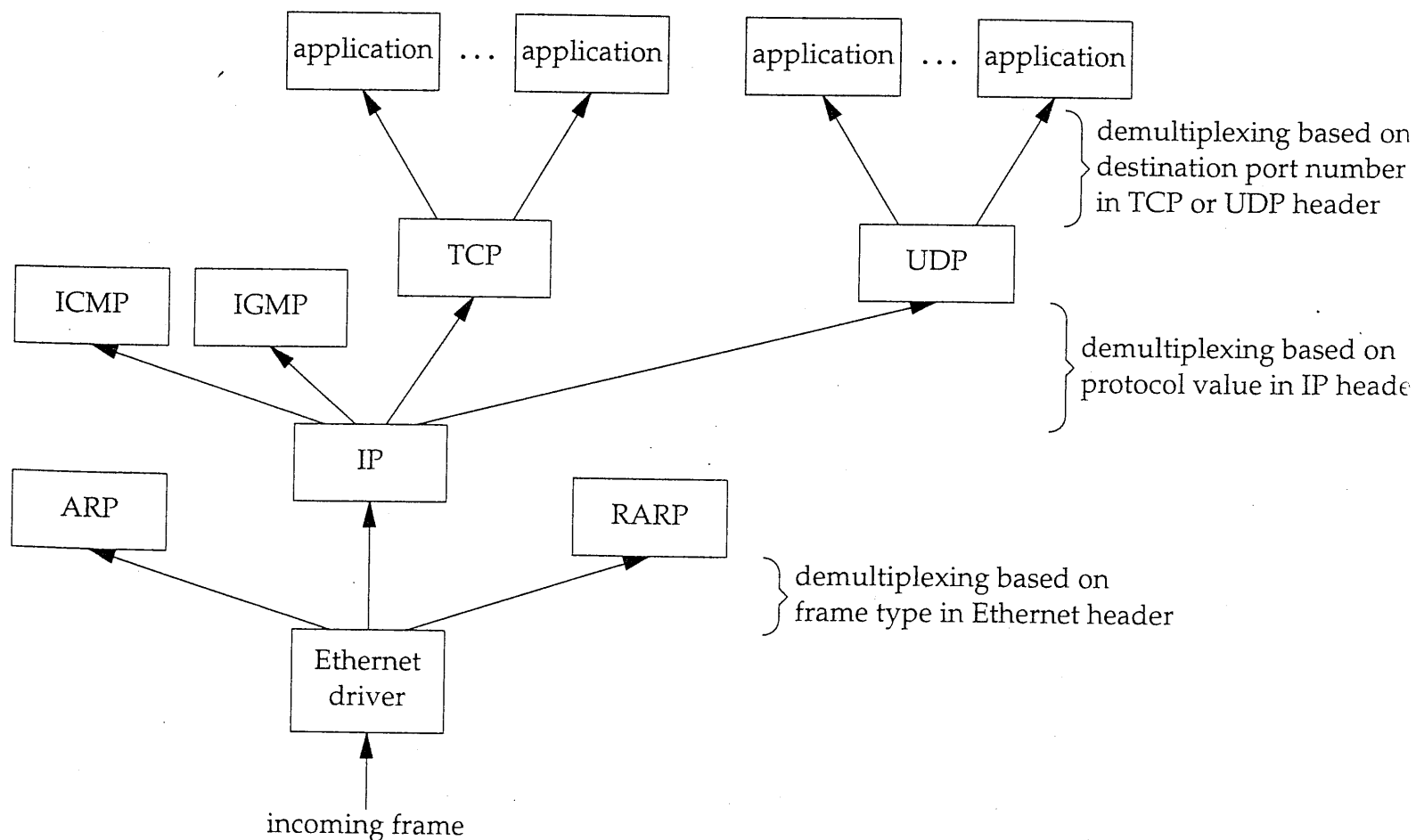


Figure 1.8 The demultiplexing of a received Ethernet frame.

Positioning the protocol boxes labeled "ICMP" and "IGMP" is always a challenge. In Figure 1.4 we showed them at the same layer as IP, because they really are adjuncts to IP. But here we show them above IP to reiterate that ICMP is not a transport protocol and IGMP is not a network protocol.

ARP provides a dynamic mapping from an IP address to the corresponding hardware address. We use the term *dynamic* since it happens automatically and is normally not a concern of either the application user or the system administrator.

RARP is used by systems without a disk drive (normally diskless workstations or X terminals) but requires manual configuration by the system administrator. We describe it in Chapter 5.

2 An Example

Whenever we type a command of the form

```
% ftp bsdi
```

the following steps take place. These numbered steps are shown in Figure 4.2.

1. The application, the FTP client, calls the function `gethostbyname(3)` to convert the hostname (`bsdi`) into its 32-bit IP address. This function is called a *resolver* in the DNS (Domain Name System), which we describe in Chapter 14. This conversion is done using the DNS, or on smaller networks, a static hosts file (`/etc/hosts`).
2. The FTP client asks its TCP to establish a connection with that IP address.
3. TCP sends a connection request segment to the remote host by sending an IP datagram to its IP address. (We'll see the details of how this is done in Chapter 18.)
4. If the destination host is on a locally attached network (e.g., Ethernet, token ring, or the other end of a point-to-point link), the IP datagram can be sent directly to that host. If the destination host is on a remote network, the IP routing function determines the Internet address of a locally attached next-hop router to send the IP datagram to. In either case the IP datagram is sent to a host or router on a locally attached network.
5. Assuming an Ethernet, the sending host must convert the 32-bit IP address into a 48-bit Ethernet address. A translation is required from the *logical* Internet address to its corresponding *physical* hardware address. This is the function of ARP.

ARP is intended for broadcast networks where many hosts or routers are connected to a single network.

6. ARP sends an Ethernet frame called an *ARP request* to every host on the network. This is called a *broadcast*. We show the broadcast in Figure 4.2 with dashed lines. The ARP request contains the IP address of the destination host (whose name is `bsdi`) and is the request "if you are the owner of this IP address, please respond to me with your hardware address."

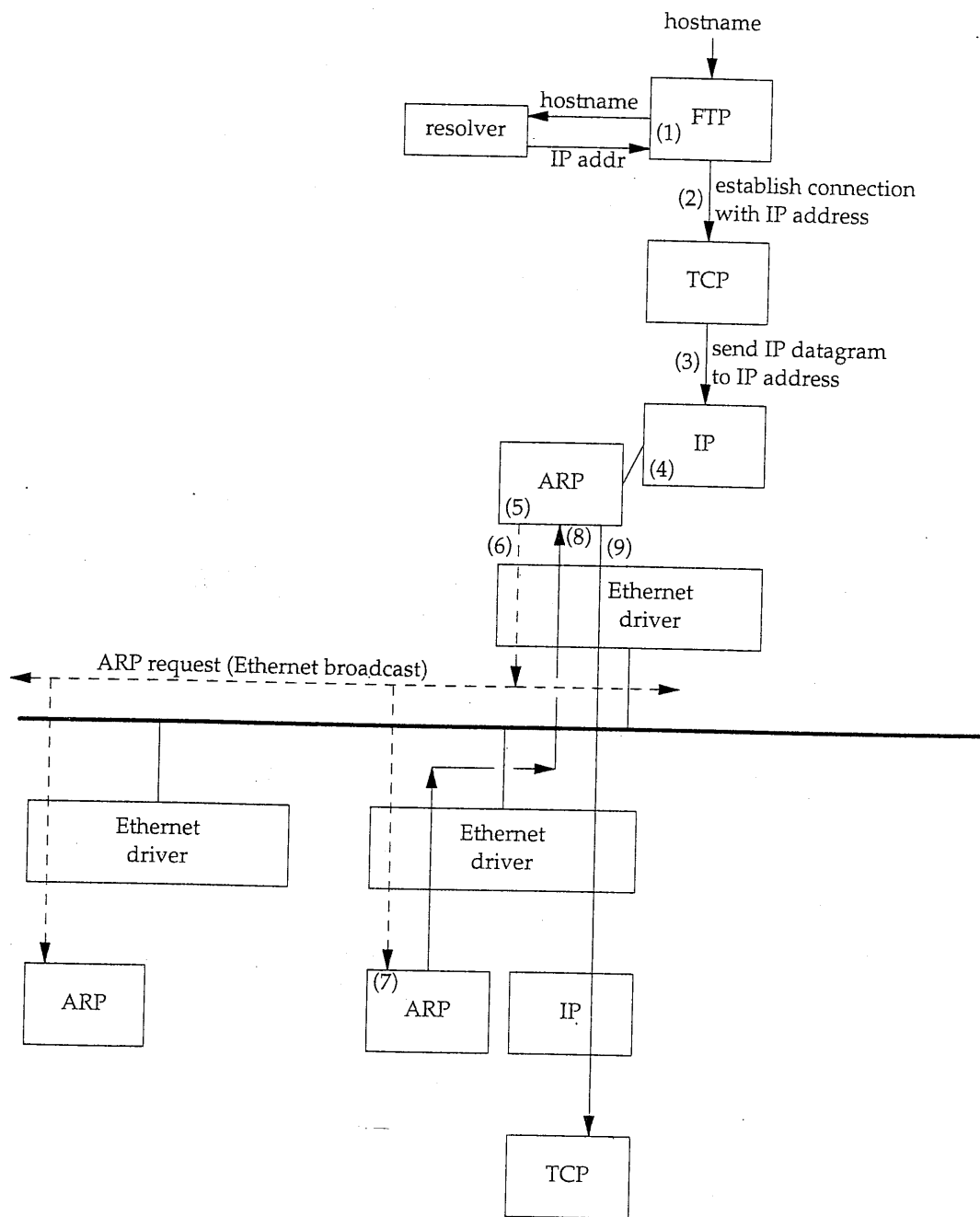


Figure 4.2 Operation of ARP when user types "ftp hostname".

7. The destination host's ARP layer receives this broadcast, recognizes that the sender is asking for its IP address, and replies with an *ARP reply*. This reply contains the IP address and the corresponding hardware address.
8. The ARP reply is received and the IP datagram that forced the ARP request-reply to be exchanged can now be sent.
9. The IP datagram is sent to the destination host.

Figure 18.13 shows the normal TCP connection establishment and termination detailing the different states through which the client and server pass. It is a redo of Figure 18.3 showing only the states.

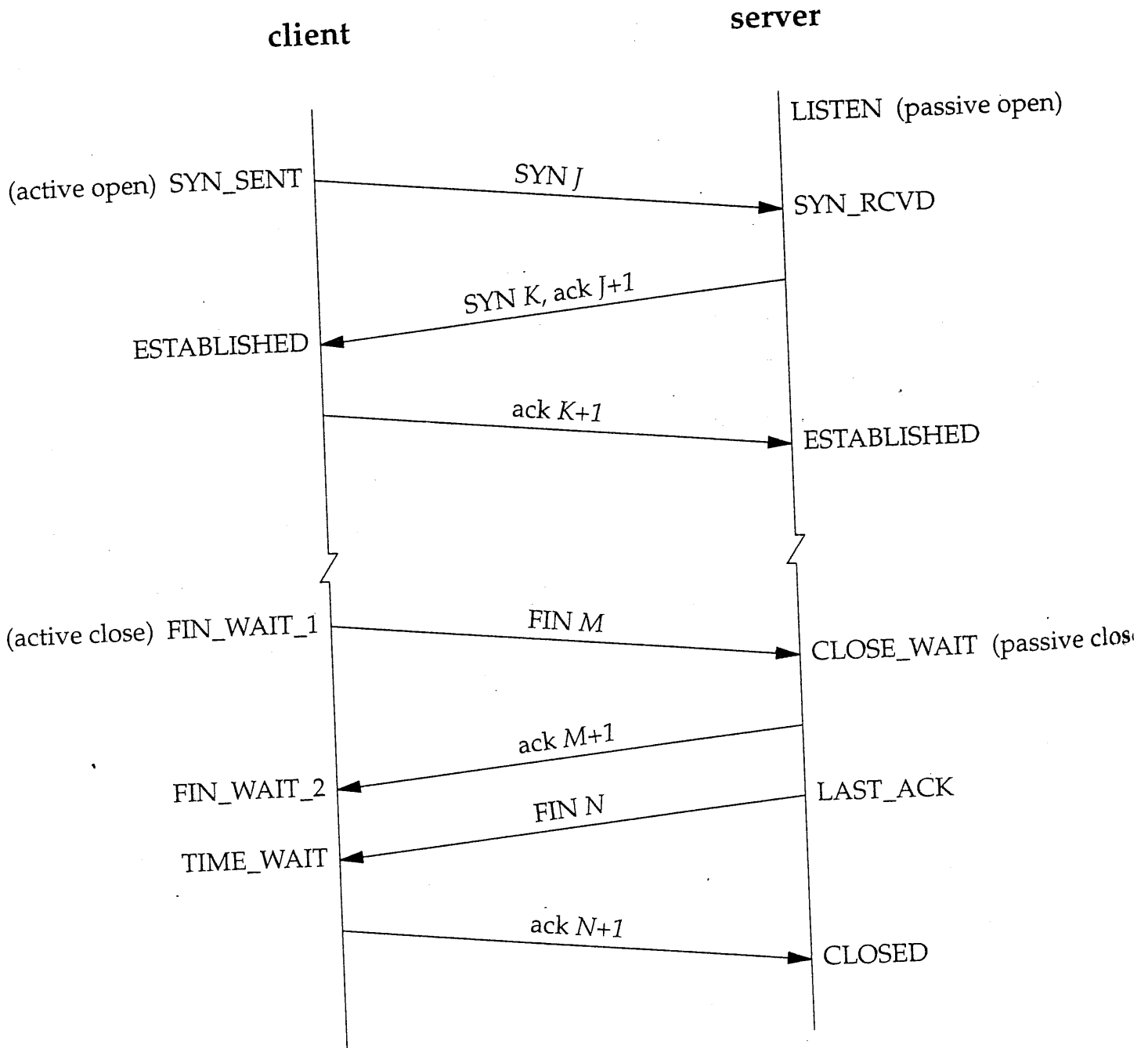


Figure 18.13 TCP states corresponding to normal connection establishment and termination.

We assume in Figure 18.13 that the client on the left side does an active open, and server on the right side does a passive open. Although we show the client doing active close, as we mentioned earlier, either side can do the active close.

You should follow through the state changes in Figure 18.13 using the state transition diagram in Figure 18.12, making certain you understand why each state change takes place.