Chapter 8

DHCP & ARP

The ability to network devices quickly and easily is critical in a hyper-connected world, and although it has been around for decades, DHCP remains an essential method to ensure that devices are able to join networks and are configured correctly. DHCP greatly reduces the errors that are made when IP addresses are assigned manually, and can stretch IP addresses by limiting how long a device can keep an individual IP address.

8.1 DHCP Definition

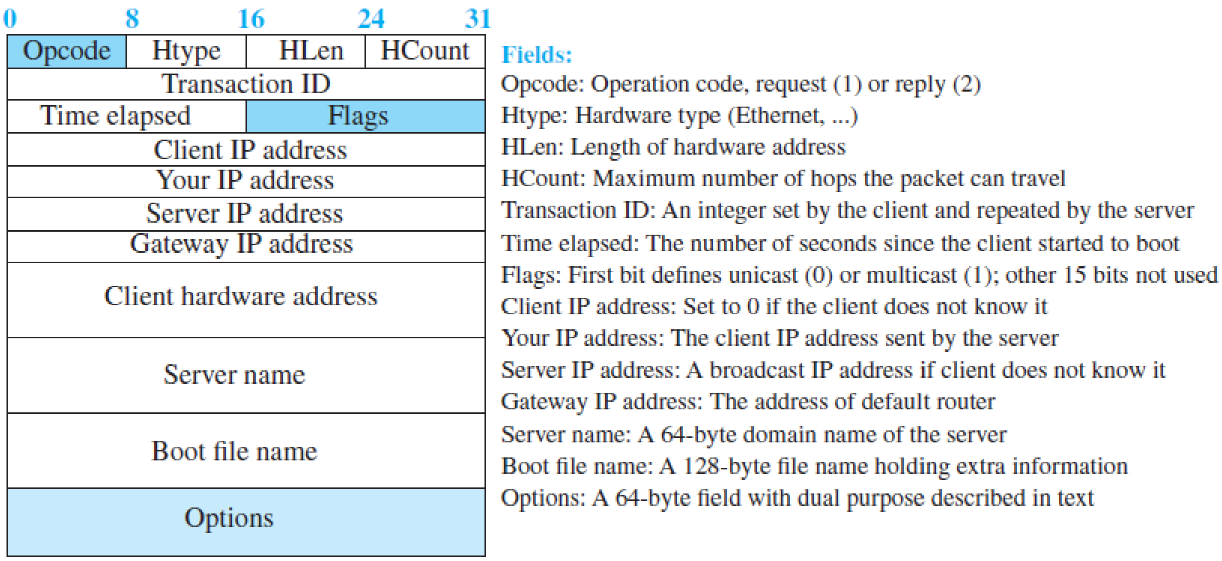
DHCP stands for dynamic host configuration protocol and is a network protocol used on IP networks where a DHCP server automatically assigns an IP address and other information to each host on the network so they can communicate efficiently with other endpoints.

In addition to the IP address, DHCP also assigns the subnet mask, default gateway address, domain name server (DNS) address and other pertinent configuration parameters. Request for comments (RFC) 2131 and 2132 define DHCP as an Internet Engineering Task Force (IETF)-defined standard based on the BOOTP protocol.

8.2 DHCP Message Format

DHCP is a client-server protocol in which the client sends a request message and the server returns a response message. Before we discuss the operation of DHCP, let us show the general format of the DHCP message in Figure 18.25. Most of the fields are explained in the figure, but we need to discuss the option field, which plays a very important role in DHCP.

The 64-byte option field has a dual purpose. It can carry either additional information or some specific vendor information. The server uses a number, called a **magic cookie,** in the format of an IP address with the value of 99.130.83.99. When the client finishes reading the message, it looks for this magic cookie. If present, the next 60 bytes are options. An option is composed of three fields: a 1-byte tag field, a 1-byte length field, and a variable-length value field. There are several tag fields that are mostly used by vendors. If the tag field is 53, the value field defines one of the 8 message types shown in the given figure. We show how these message types are used by DHCP.



Fields:
Opcode: Operation code, request (1) or reply (2)
Htype: Hardware type (Ethernet, ...)
HLen: Length of hardware address
HCount: Maximum number of hops the packet can travel
Transaction ID: An integer set by the client and repeated by the server
Time elapsed: The number of seconds since the client started to boot
Flags: First bit defines unicast (0) or multicast (1); other 15 bits not used
Client IP address: Set to 0 if the client does not know it
Your IP address: The client IP address sent by the server
Server IP address: A broadcast IP address if client does not know it
Gateway IP address: The address of default router
Server name: A 64-byte domain name of the server
Boot file name: A 128-byte file name holding extra information
Options: A 64-byte field with dual purpose described in text

Figure 8.1. DHCP Message Format

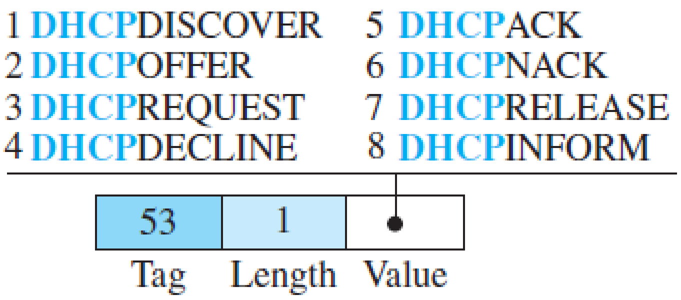
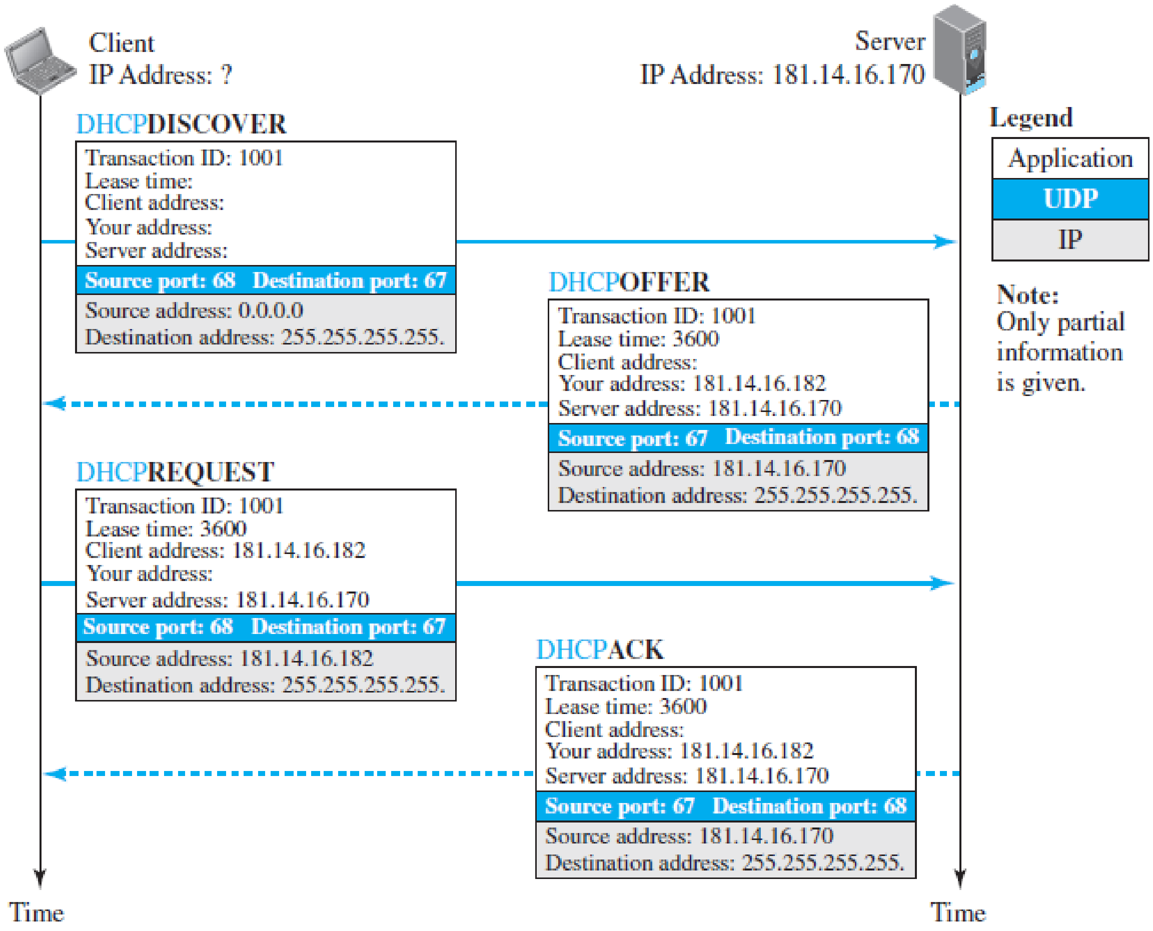


Figure 8.2. Shows a Simple Scenario

8.3 Operation of DHCP



DHCPDISCOVER
Transaction ID: 1001
Lease time:
Client address:
Your address:
Server address:
Source port: 68 Destination port: 67
Source address: 0.0.0.0
Destination address: 255.255.255.255.

DHCPOFFER
Transaction ID: 1001
Lease time: 3600
Client address:
Your address: 181.14.16.182
Server address: 181.14.16.170
Source port: 67 Destination port: 68
Source address: 181.14.16.170
Destination address: 255.255.255.255.

DHCPREQUEST
Transaction ID: 1001
Lease time: 3600
Client address: 181.14.16.182
Your address:
Server address: 181.14.16.170
Source port: 68 Destination port: 67
Source address: 181.14.16.182
Destination address: 255.255.255.255.

DHCPACK
Transaction ID: 1001
Lease time: 3600
Client address:
Your address: 181.14.16.182
Server address: 181.14.16.170
Source port: 67 Destination port: 68
Source address: 181.14.16.170
Destination address: 255.255.255.255.

Figure 8.3. Operation of DHCP

1. DHCP Discover
When a client (PC) is booted, it broadcasts a DHCP Discover message over the Ethernet network to locate all available DHCP servers on the same subnet network (by setting the destination MAC address in the Ethernet header as Broadcast MAC=FF:FF:FF:FF:FF:FF), reaching all the DHCP servers on the same subnet network.

2. DHCP Offer
When a DHCP server receives the DHCP Discover message from the client, it also broadcasts a DHCP Offer message over the Ethernet network (because the client IP address has not been allocated yet), informing the client that it is available. This message contains the network information, such as client IP address, subnet mask, default gateway IP address, DNS IP address, IP lease time and DHCP server IP address. The DHCP Offer message broadcasted is delivered to all the clients on the same subnet network, including the one that sent the DHCP Discover message.

3. DHCP Request

The client, having received the DHCP Offer message, recognizes there is a DHCP server available on the same subnet. Then it broadcasts a DHCP Request message to the server over the Ethernet network, requesting network configuration data including an IP address for itself. If more than one DHCP server responds on the same subnet and hence the client receives multiple DHCP Offer messages, it selects one of the DHCP servers, and enters the IP address of the selected DHCP server in the DHCP Server Identifier (option 54) field of the DHCP Request message. Then it informs all the DHCP servers on the subnet network about such selection by broadcasting the DHCP Request message. Typically, all DHCP servers internally store the network configuration data (i.e. IP address for the client and other information) when they send a DHCP Offer message. So, the client broadcasts the DHCP Request message to all the DHCP servers, so that those not selected can also receive the message and delete the stored network configuration data from their memory.

4. DHCP Ack

The DHCP server which received the DHCP Request message from the client checks if the IP address shown in the DHCP Server Identifier (option 54) field matches its own. If it does, it broadcasts a DHCP Ack message ensuring the client can receive the message (Note: the client has NOT been allocated an IP address yet).

8.4 Transition States

The previous scenarios we discussed for the operation of the DHCP were very simple. To provide dynamic address allocation, the DHCP client acts as a state machine that performs transitions from one state to another depending on the messages it receives or sends. Figure 8.3 shows the transition diagram with the main states.

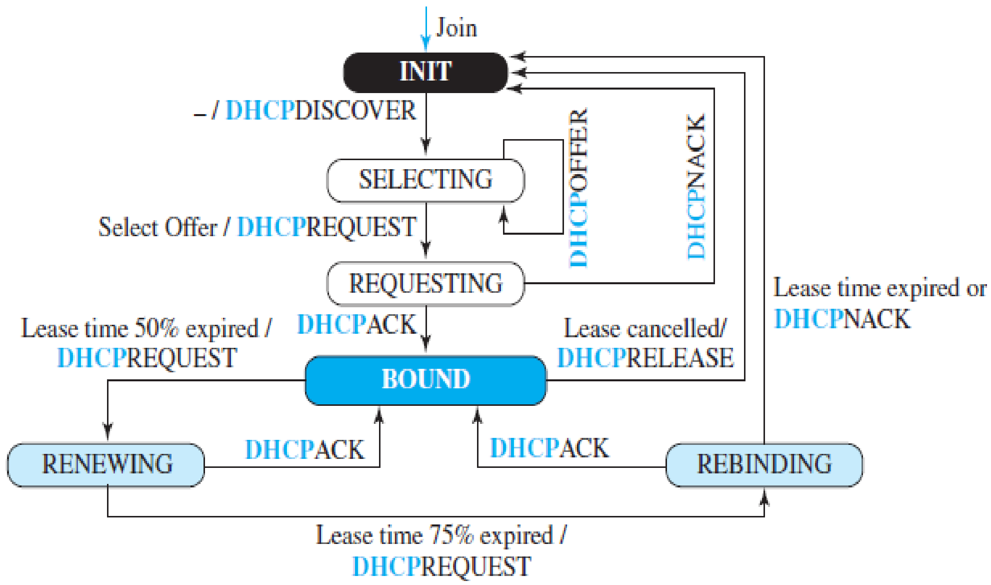


Figure 8.3. Operation of DHCP

- **INIT State:** When the DHCP client first starts, it is in the INIT state (initializing state). The client broadcasts a DHCPDISCOVER message (a request message with the DHCPDISCOVER option), using port 67.
- **SELECTING State:** After sending the DHCPDISCOVER message, the client goes to the selecting state. Those servers that can provide this type of service respond with a DHCPOFFER message. In these messages, the servers offer an IP address. They can also offer the lease duration. The default is 1 hour. The server that sends a DHCPOFFER locks the offered IP address so that it is not available to any other clients. The client chooses one of the offers and sends a DHCPREQUEST message to the selected server. It then goes to the requesting state. However, if the client receives no DHCPOFFER message, it tries four more times, each with a span of 2 seconds. If there is no reply to any of these DHCPDISCOVERs, the client sleeps for 5 minutes before trying again.
- **REQUESTING State:** The client remains in the requesting state until it receives a DHCPACK message from the server that creates the binding between the client physical address and its IP address. After receipt of the DHCPACK, the client goes to the bound state.
- **BOUND State:** In this state, the client can use the IP address until the lease expires. When 50 percent of the lease period is reached, the client sends another DHCPREQUEST to ask for renewal. It then goes to the renewing state. When in the bound state, the client can also cancel the lease and go to the initializing state.
- **RENEWING State:** The client remains in the renewing state until one of two events happens. It can receive a DHCPACK, which renews the lease agreement. In this case, the client resets its timer and goes back to the bound state. Or, if a DHCPACK is not received, and 87.5 percent of the lease time expires, the client goes to the rebinding state.

- **REBINDING State:** The client remains in the rebinding state until one of three events happens. If the client receives a DHCPNACK or the lease expires, it goes back to the initializing state and tries to get another IP address. If the client receives a DHCPACK, it goes to the bound state and resets the timer.

8.5 ARP Definition

ARP (Address Resolution Protocol) is a network protocol used to find out the hardware (MAC) address of a device from an IP address. It is used when a device wants to communicate with some other device on a local network (for example on an Ethernet network that requires physical addresses to be known before sending packets). The sending device uses ARP to translate IP addresses to MAC addresses. The device sends an ARP request message containing the IP address of the receiving device. All devices on a local network segment see the message, but only the device that has that IP address responds with the ARP reply message containing its MAC address. The sending device now has enough information to send the packet to the receiving device.

ARP request packets are sent to the broadcast addresses (FF:FF:FF:FF:FF:FF for the Ethernet broadcasts and 255.255.255.255 for the IP broadcast).

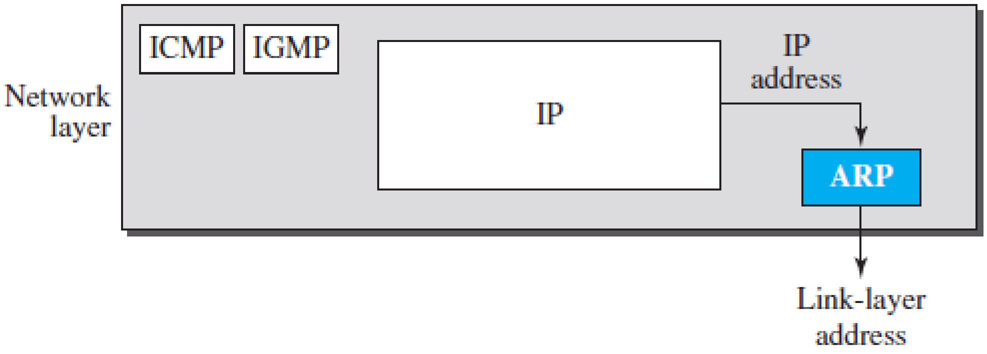
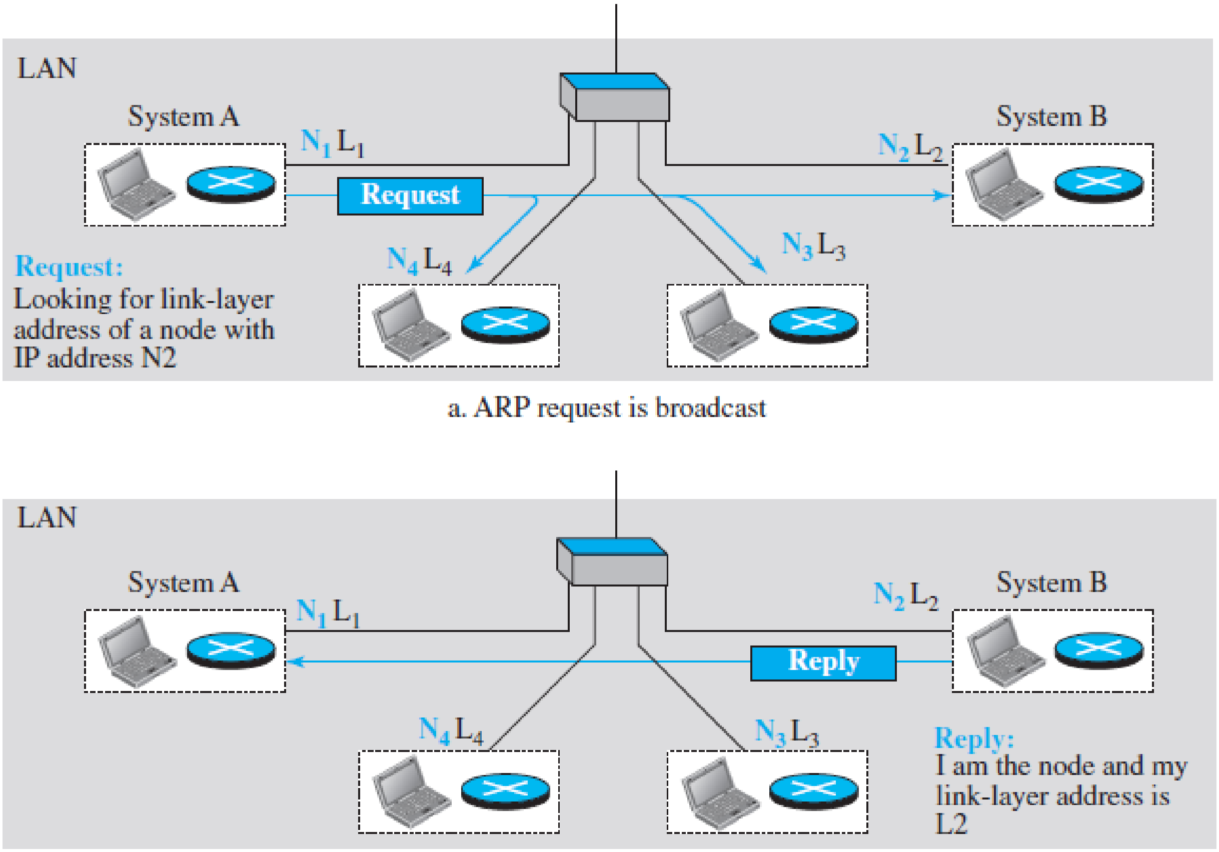


Figure 8.4. Position of ARP in TCP/IP Protocol Suit

8.6 ARP Operation

Anytime a host or a router needs to find the link-layer address of another host or router in its network, it sends an ARP request packet. The packet includes the link-layer and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the link-layer address of the receiver, the query is broadcast over the link using the link-layer broadcast address, which we discuss for each protocol later (see Figure 8.5).



Request:
Looking for link-layer address of a node with IP address N2

a. ARP request is broadcast

Reply:
I am the node and my link-layer address is L2

b. ARP reply is unicast

Figure 8.5. ARP Operation

Every host or router on the network receives and processes the ARP request packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet. The response packet contains the recipient's IP and link-layer addresses. The packet is unicast directly to the node that sent the request packet.

In Figure 8.5a, the system on the left (A) has a packet that needs to be delivered to another system (B) with IP address **N2**. System A needs to pass the packet to its data-link layer for the actual delivery, but it does not know the physical address of the recipient. It uses the services of ARP by asking the ARP protocol to send a broadcast ARP request packet to ask for the physical address of a system with an IP address of **N2**.

This packet is received by every system on the physical network, but only system B will answer it, as shown in Figure 8.5b. System B sends an ARP reply packet that includes its physical address. Now system A can send all the packets it has for this destination using the physical address it received.

8.7 Caching

An Address Resolution Protocol cache (ARP cache) is a repository for data that is used to connect an IP address to a Media Access Control (MAC) address for a physical machine or device in a local network. The ARP cache can hold data for both wireless and Ethernet routing, and helps to route packets to the right endpoint. One main role of an ARP cache is to accommodate ARP requests where a gateway has to deal with where to send packets within a local network. For consumers, the gateway is often part of the Internet Service Provider infrastructure. The gateway may generate an ARP request, where the system will use information in the ARP cache to find the right connected device for a given address.

Some issues with an ARP cache relate to "resolving" an IP address to a MAC address. To this end, dynamic ARP cache setups have been created, where a registered address will be kept for reference for a specific length of time. This helps to limit problems with ARP address resolution.

A question that is often asked is this: If system A can broadcast a frame to find the link layer address of system B, why can’t system A send the datagram for system B using a broadcast frame? In other words, instead of sending one broadcast frame (ARP request), one unicast frame (ARP response), and another unicast frame (for sending the datagram), system A can encapsulate the datagram and send it to the network. System B receives it and keep it; other systems discard it.

To answer the question, we need to think about the efficiency. It is probable that system A has more than one datagram to send to system B in a short period of time. For example, if system B is supposed to receive a long e-mail or a long file, the data do not fit in one datagram.

Let us assume that there are 20 systems connected to the network (link): system A, system B, and 18 other systems. We also assume that system A has 10 datagrams to send to system B in one second.

a. Without using ARP, system A needs to send 10 broadcast frames. Each of the 18 other systems need to receive the frames, decapsulate the frames, remove the datagram and pass it to their network-layer to find out the datagrams do not belong to them.This means processing and discarding 180 broadcast frames.

b. Using ARP, system A needs to send only one broadcast frame. Each of the 18 other systems need to receive the frames, decapsulate the frames, remove the ARP message and pass the message to their ARP protocol to find that the frame must be discarded. This means processing and discarding only 18 (instead of180) broadcast frames. After system B responds with its own data-link address, system A can store the link-layer address in its cache memory. The rest of the nine frames are only unicast. Since processing broadcast frames is expensive (time consuming), the first method is preferable.

8.8 An Example of Communication

To show how communication is done at the data-link layer and how link-layer addresses are found, let us go through a simple example. Assume Alice needs to send a datagram to Bob, who is three nodes away in the Internet. How Alice finds the network-layer address of Bob. For the moment, assume that Alice knows the network-layer (IP) address of Bob. In other words, Alice's host is given the data to be sent, the IP address of Bob, and the IP address of Alice's host (each host needs to know its IP address). Figure 8.6 shows the part of the internet for our example.

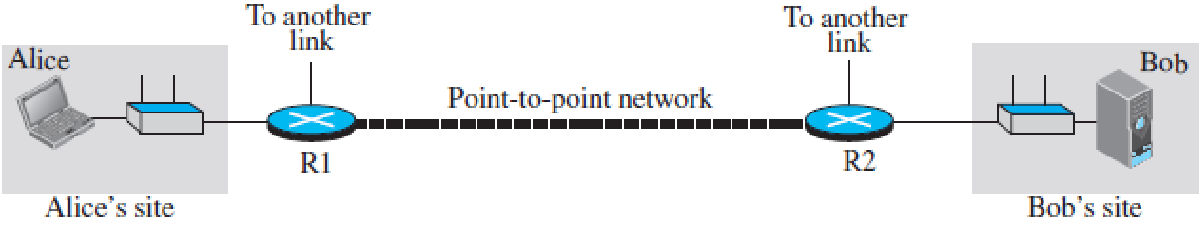


Figure 8.6. The Internet for our example

Activities at Alice's Site

We will use symbolic addresses to make the figures more readable. Figure 8.7 shows what happens at Alice's site.

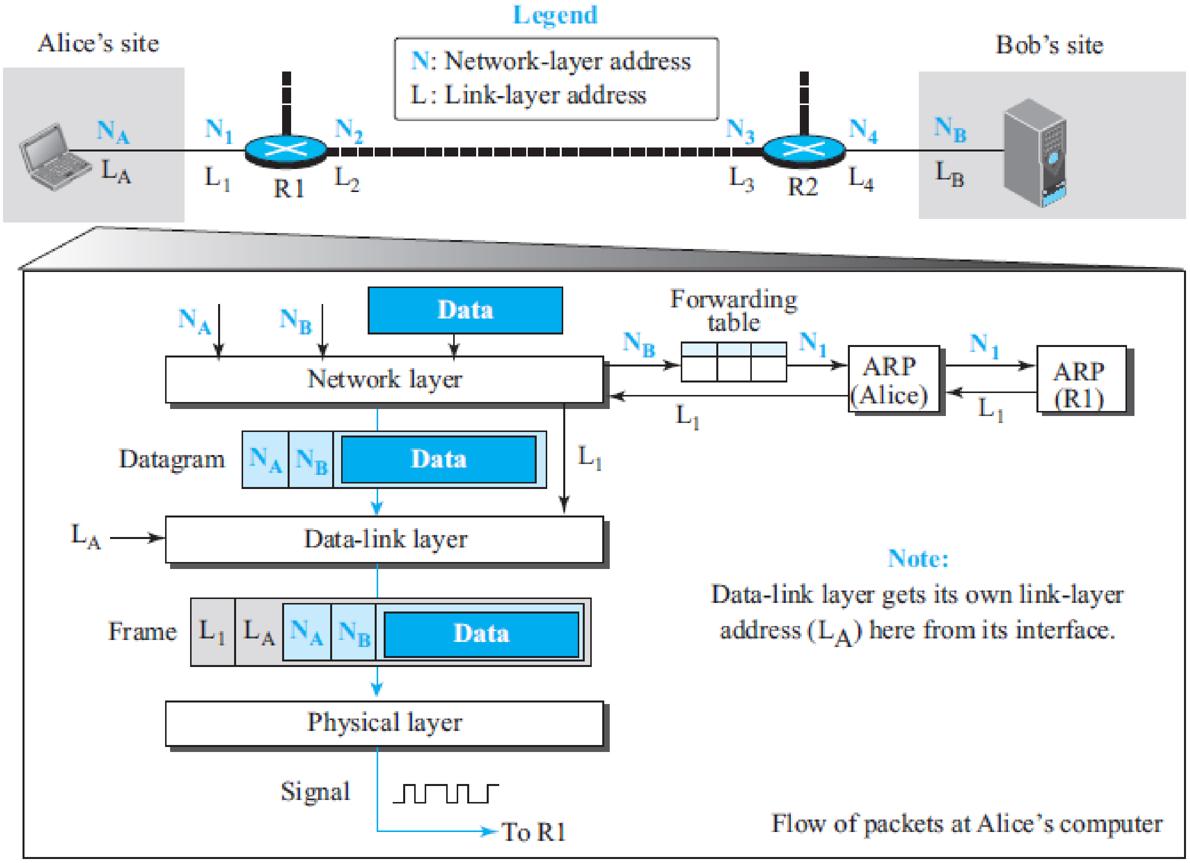


Figure 8.7. Flow of packets at Alice's computer

The network layer knows it's given **NA**, **NB**, and the packet, but it needs to find the link-layer address of the next node. The network layer consults its routing table and tries to find which router is next (the default router in this case) for the destination **NB**. The routing table gives **N1**, but the network layer needs to find the link-layer address of router R1. It uses its ARP to find the link-layer address **L1**. The network layer can now pass the datagram with the link-layer address to the data-link layer.

The data-link layer knows its own link-layer address, **LA**. It creates the frame and passes it to the physical layer, where the address is converted to signals and sent through the media.

Activities at Router R1

Now let us see what happens at Router R1. Router R1, as we know, has only three lower layers. The packet received needs to go up through these three layers and come down. Figure 8.8 shows the activities.

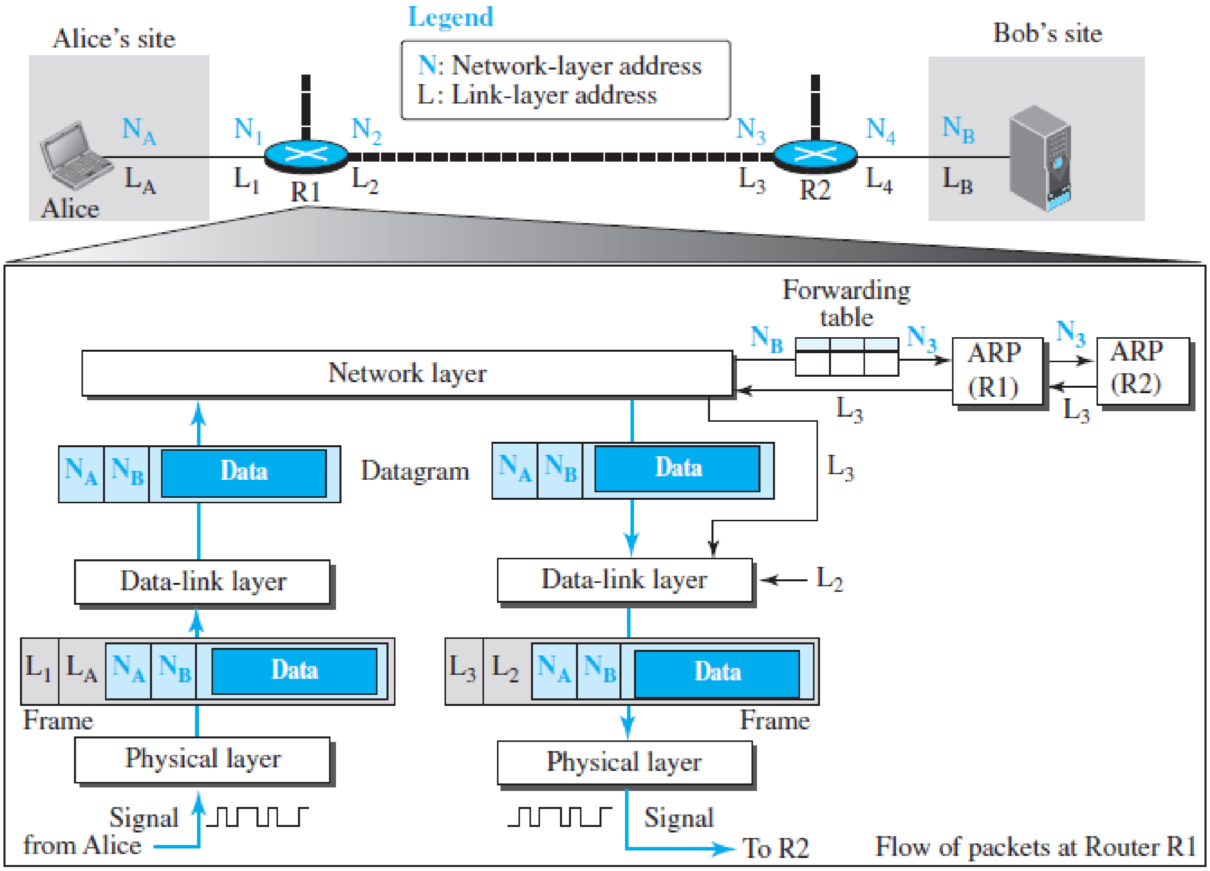


Figure 8.8. Flow of activities at Router R1

At arrival, the physical layer of the left link creates the frame and passes it to the data-link layer. The data-link layer decapsulates the datagram and passes it to the network layer. The network layer examines the network-layer address of the datagram and finds that the datagram needs to be delivered to the device with IP address **NB**. The network layer consults its routing table to find out which is the next node (router) in the path to **NB**. The forwarding table returns **N3**. The IP address of router R2 is in the same link with R1. The network layer now uses the ARP to find the link-layer address of this router, which comes up as **L3.** The network layer passes the datagram and **L3** to the data-link layer belonging to the link at the right side. The link layer encapsulates the datagram, adds **L3** and **L2** (its own link-layer address), and passes the frame to the physical layer. The physical layer encodes the bits to signals and sends them through the medium to R2.

Activities at Router R2

Activities at router R2 are almost the same as in R1, as shown in Figure 8.9

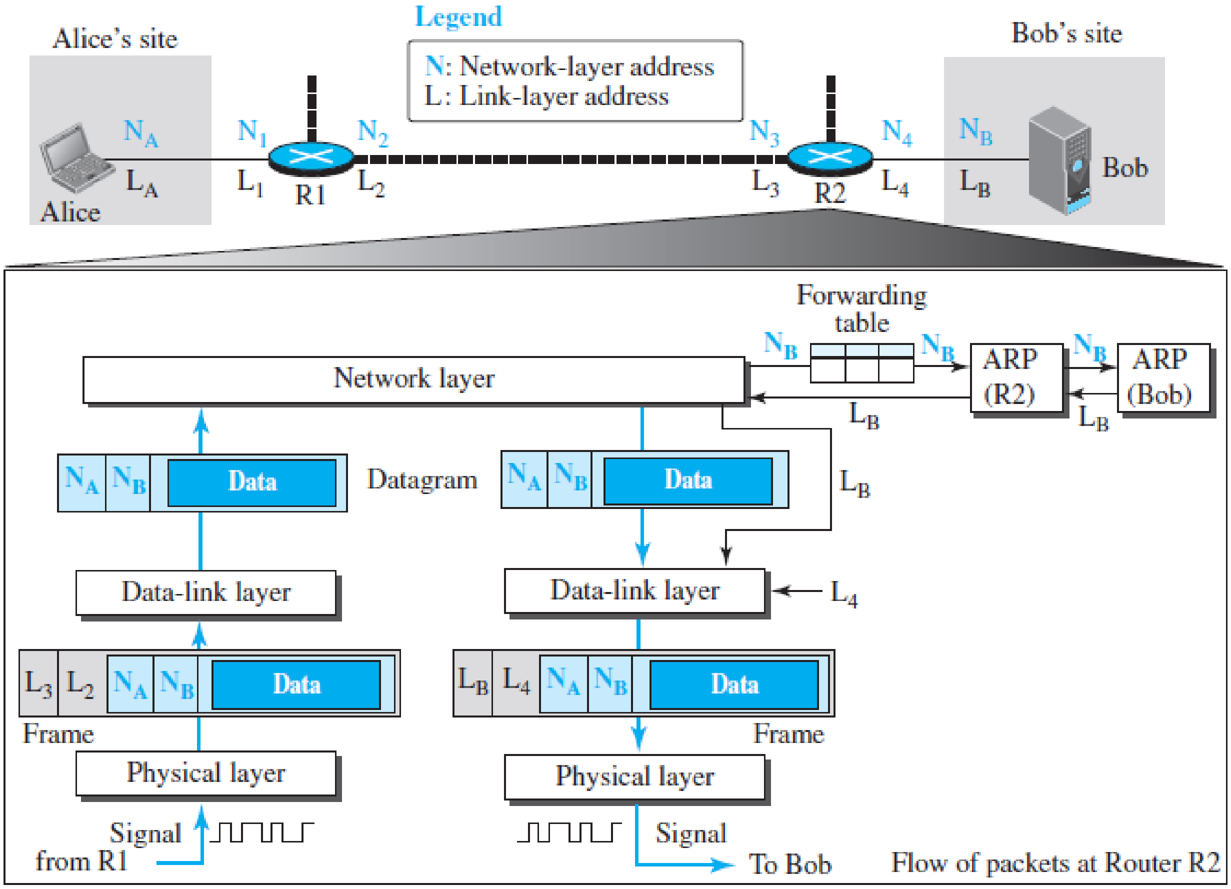


Figure 8.9. Activities at Router R2

Activities at Bob's Site

Now let us see what happens at Bob's site. Figure 8.10 shows how the signals at Bob's site are changed to a message. At Bob's site there are no more addresses or mapping needed. The signal received from the link is changed to a frame. The frame is passed to the data-link layer, which decapsulates the datagram and passes it to the network layer. The network layer decapsulates the message and passes it to the transport layer.

Changes in Addresses

This example shows that the source and destination network-layer addresses, NA and NB, have not been changed during the whole journey. However, all four network-layer addresses of routers R1 and R2 (N1, N2, N3, and N4) are needed to transfer a datagram from Alice's computer to Bob's computer.

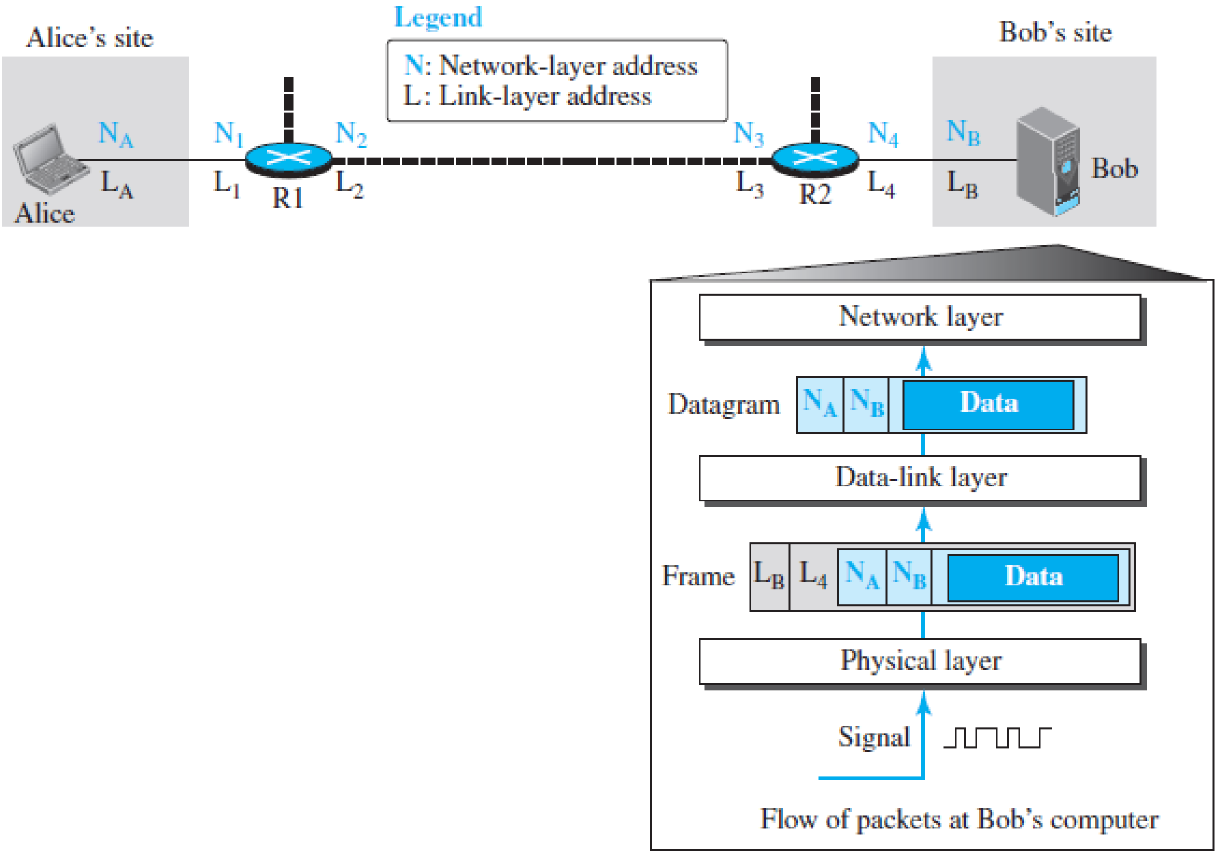


Figure 8.10. Activities at Bob's site