Internet Architecture
Second Stage
Asst. Lect. Noor Razaq

COLLEGE OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SECURITY

Lecture 6

Host Configuration:

Dynamic Host Configuration Protocol

DHCP

Lecture Outline

1. *Introduction*
2. *DHCP Operation*
3. *Configuration*

1. INTRODUCTION

- ❑ Each computer that uses the TCP/IP protocol suite needs to know its IP address. If the computer uses classless addressing or is a member of a subnet, it also needs to know its subnet mask.
- ❑ Most computers today need two other pieces of information: the address of a default router to be able to communicate with other networks and the address of a name server to be able to use names instead of addresses. In other words, four pieces of information are normally needed.

1. INTRODUCTION

The **Dynamic Host Configuration Protocol (DHCP)** is a client/server protocol designed to provide the four pieces of information for a computer that is booted for the first time:

1. **The IP address of the computer**
2. **The subnet mask of the computer**
3. **The IP address of a router**
4. **The IP address of a name server**

Previous Protocols

Before **DHCP** became the formal protocol for host configuration, some other protocols were used for this propose:

- **Reverse Address Resolution Protocol (RARP).**
- **Bootstrap Protocol (BOOTP).**

Reverse Address Resolution Protocol (RARP)

- At the beginning of the Internet era, a protocol called Reverse Address Resolution Protocol (**RARP**) was designed to provide the IP address for a booted computer. **RARP** maps a physical address to an IP address. However, **RARP** is deprecated today for two reasons:
  - **First:** RARP used the broadcast service of the data link layer, which means that a RARP server must be present in each network.
  - **Second:** RARP can provide only the IP address of the computer, but a computer today needs all four pieces of information mentioned above.

Bootstrap Protocol (BOOTP)

- ❑ The Bootstrap Protocol (**BOOTP**) is **the pre-runner of DHCP**. It is **a client/server protocol** designed to overcome the two deficiencies of the RARP protocol:
  - ✓ **First:** since it is a client/server program, the BOOTP server can be anywhere in the Internet.
  - ✓ **Second:** it can provide all pieces of information we mentioned above.
- ❑ **BOOTP** is **a static 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.

Bootstrap Protocol (BOOTP)

- ❑ There are some situations in which we need a dynamic configuration protocol in some situations:
  - ✓ When a host moves from one physical network to another.
  - ✓ When a host wants a temporary IP address to be used for a period of time.
- ❑ BOOTP cannot handle these situations because the binding between the physical and IP addresses is **static** and **fixed** in **a table** until changed by the administrator. Therefore, DHCP has been devised to handle these shortcomings.

2. DHCP OPERATION

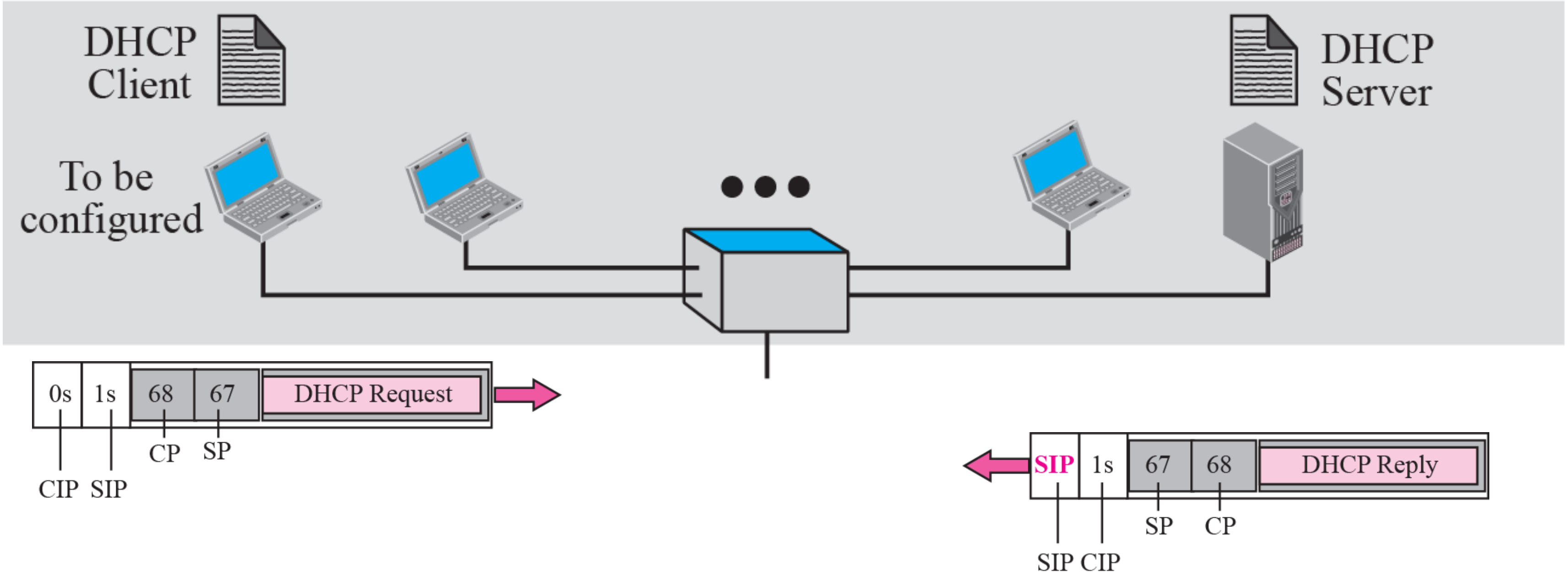
The DHCP client and server can either be on the same network or on different networks.

Client and server on the same network

The administrator may put the client and the server on the same network.

Legend

CP: Client Port Number
SP: Server Port Number
CIP: Client IP Address
SIP: Server IP Address



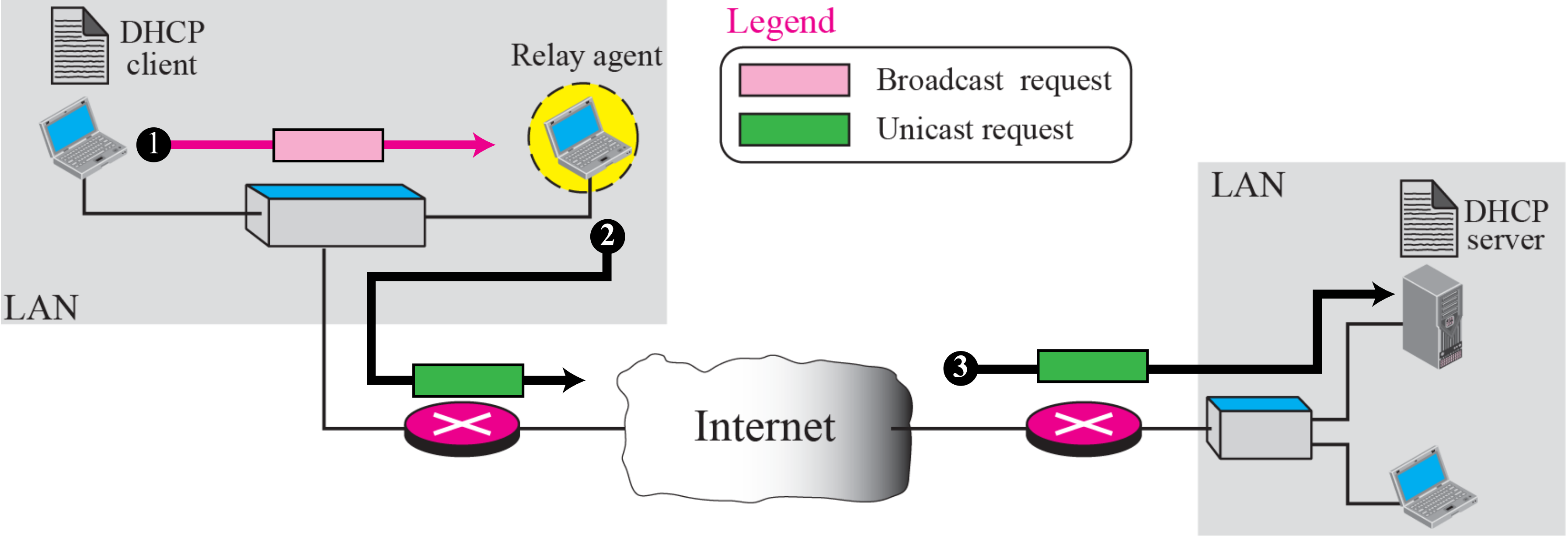
Client and Server on the same network

The operation can be described as follows:

1. **The DHCP server** issues a passive open command on **UDP port number 67** and waits for a client.
2. **A booted client** issues an active open command on port number. The message is encapsulated in a UDP user datagram, using **the destination port number 67** and **the source port number 68**. The UDP user datagram, in turn, is encapsulated in an IP datagram. The client uses all 0s as the source address and all 1s as the destination address.
3. **The server** responds with either a broadcast or a unicast message using UDP source port number 67 and destination port number 68.

Client and Server on two different networks

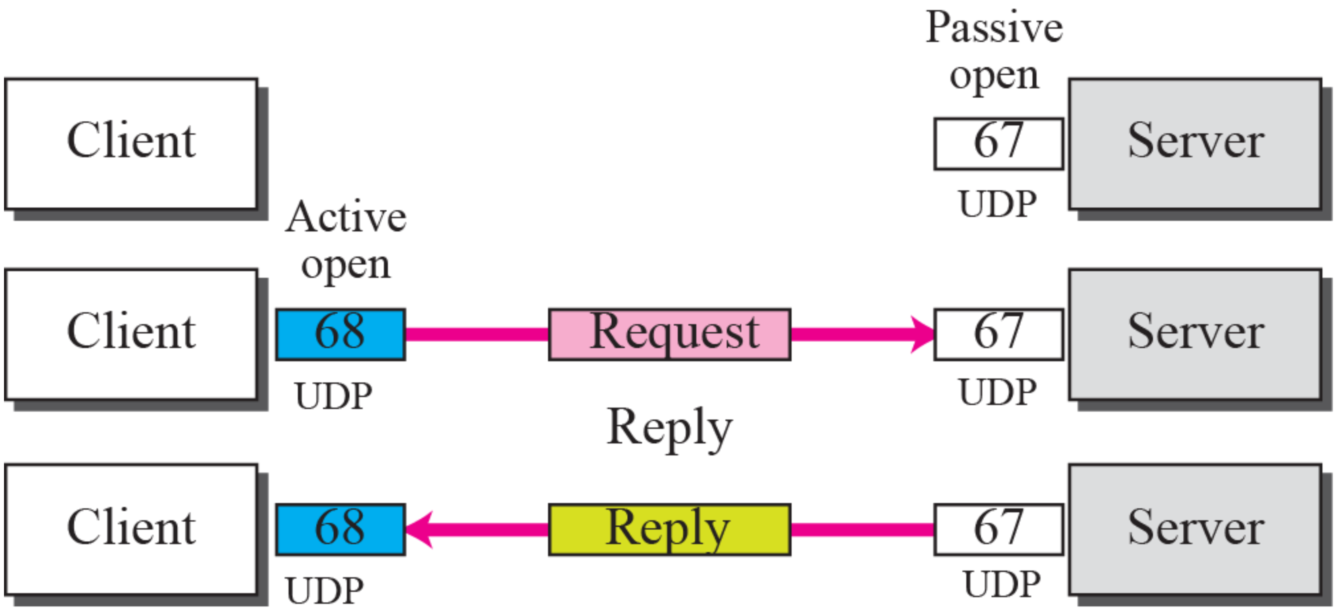
- As in other application-layer processes, a client can be in one network and the server in another, separated by several other networks.
- **There is one problem that must be solved**. The DHCP request is broadcast because the client does not know the IP address of the server. A broadcast IP datagram cannot pass through any router



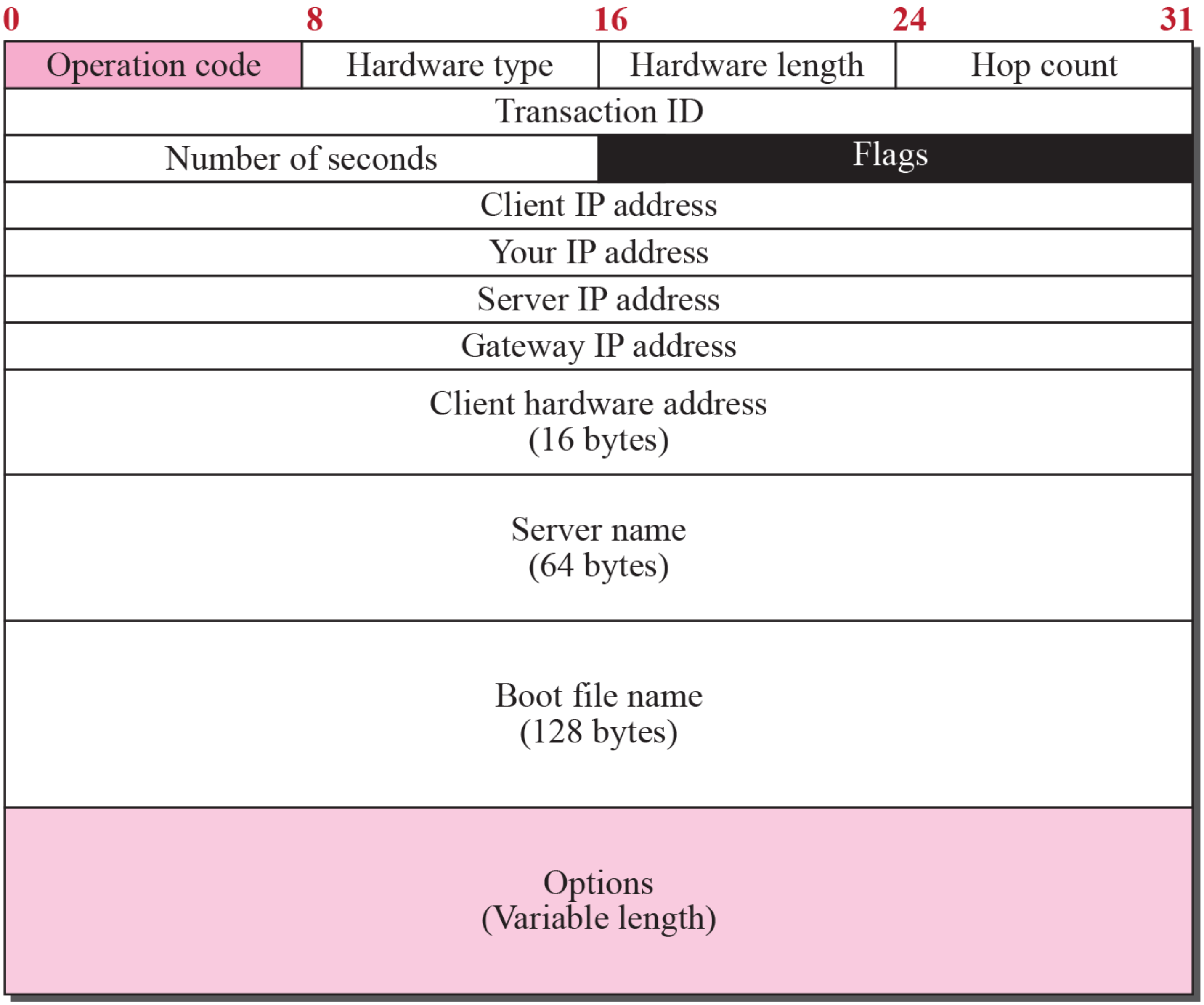
Client and Server on two different networks

- ❑ To solve the problem, there is a need for an intermediary. One of the hosts 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 DHCP server and listens for broadcast messages on port 67.
- ❑ When it receives this type of packet, it encapsulates the message in a unicast datagram and sends the request to the DHCP server.
- ❑ The DHCP server knows the message comes from a relay agent because one of the fields in the request message defines the IP address of the relay agent.
- ❑ The relay agent, after receiving the reply, sends it to the DHCP client.

Use of UDP ports



DHCP Packet Format



0 8 16 24 31

Operation code | Hardware type | Hardware length | Hop count

Transaction ID

Number of seconds | Flags

Client IP address

Your IP address

Server IP address

Gateway IP address

Client hardware address
(16 bytes)

Server name
(64 bytes)

Boot file name
(128 bytes)

Options
(Variable length)

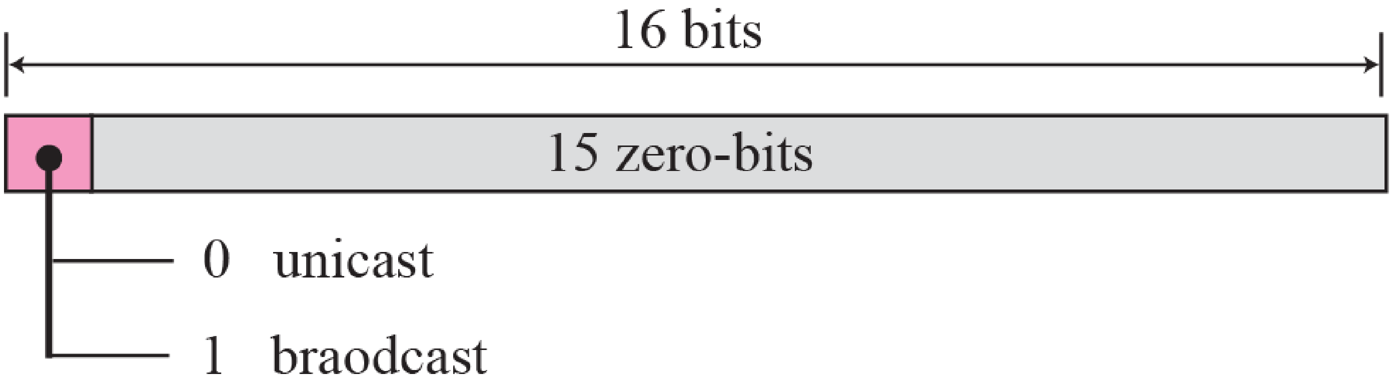
DHCP Packet Format

The following briefly describes each field:

- **Operation code.** This 8-bit field defines the type of DHCP packet: request (1) or reply (2).
- **Hardware type.** This is an 8-bit field defining the type of physical network. Each type of network has been assigned an integer. For example, for Ethernet the value is 1.
- **Hardware length.** This is an 8-bit field defining the length of the physical address in bytes. For example, for Ethernet the value is 6.
- **Hop count.** This is an 8-bit field defining the maximum number of hops the packet can travel.
- **Transaction ID.** This is a 4-byte field carrying an integer. The transaction identification is set by the client and is used to match a reply with the request. The server returns the same value in its reply.
- **Number of seconds.** This is a 16-bit field that indicates the number of seconds elapsed since the time the client started to boot.

DHCP Packet Format

- **Flag.** This is a 16-bit field in which only the leftmost bit is used and the rest of the bits should be set to 0s. A leftmost bit specifies a forced broadcast reply (instead of unicast) from the server. If the reply were to be unicast to the client, the destination IP address of the IP packet is the address assigned to the client. Since the client does not know its IP address, it may discard the packet. However, if the IP datagram is broadcast, every host will receive and process the broadcast message.



DHCP Packet Format

- **Client IP address.** This is a 4-byte field that contains the client IP address. If the client does not have this information, this field has a value of 0.

- **Your IP address.** This is a 4-byte field that contains the client IP address. It is filled by the server (in the reply message) at the request of the client.

- **Server IP address.** This is a 4-byte field containing the server IP address. It is filled by the server in a reply message.

- **Gateway IP address.** This is a 4-byte field containing the IP address of a router. It is filled by the server in a reply message.

- **Client hardware address.** This is the physical address of the client. Although the server can retrieve this address from the frame sent by the client, it is more efficient if the address is supplied explicitly by the client in the request message.

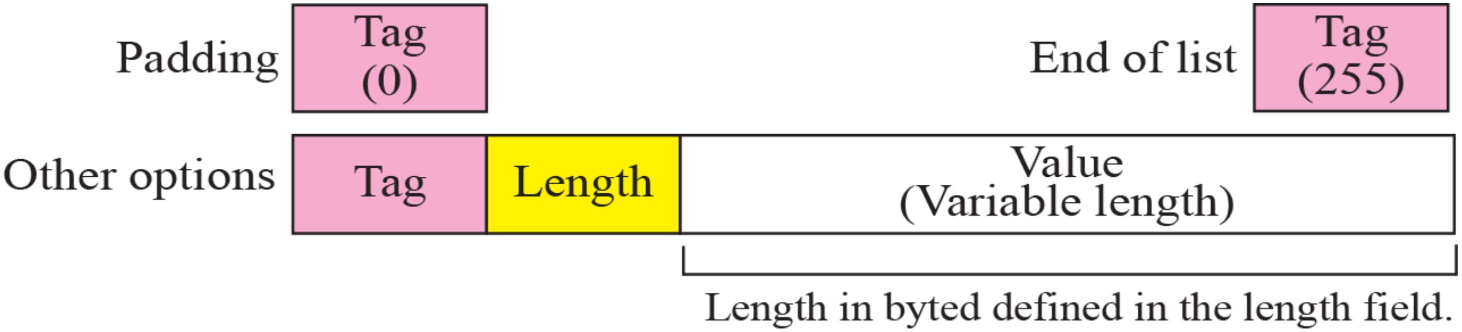
- **Server name.** This is a 64-byte field that is optionally filled by the server in a reply packet. It contains a null-terminated string consisting of the domain name of the server. If the server does not want to fill this field with data, the server must fill it with all 0s.

DHCP Packet Format

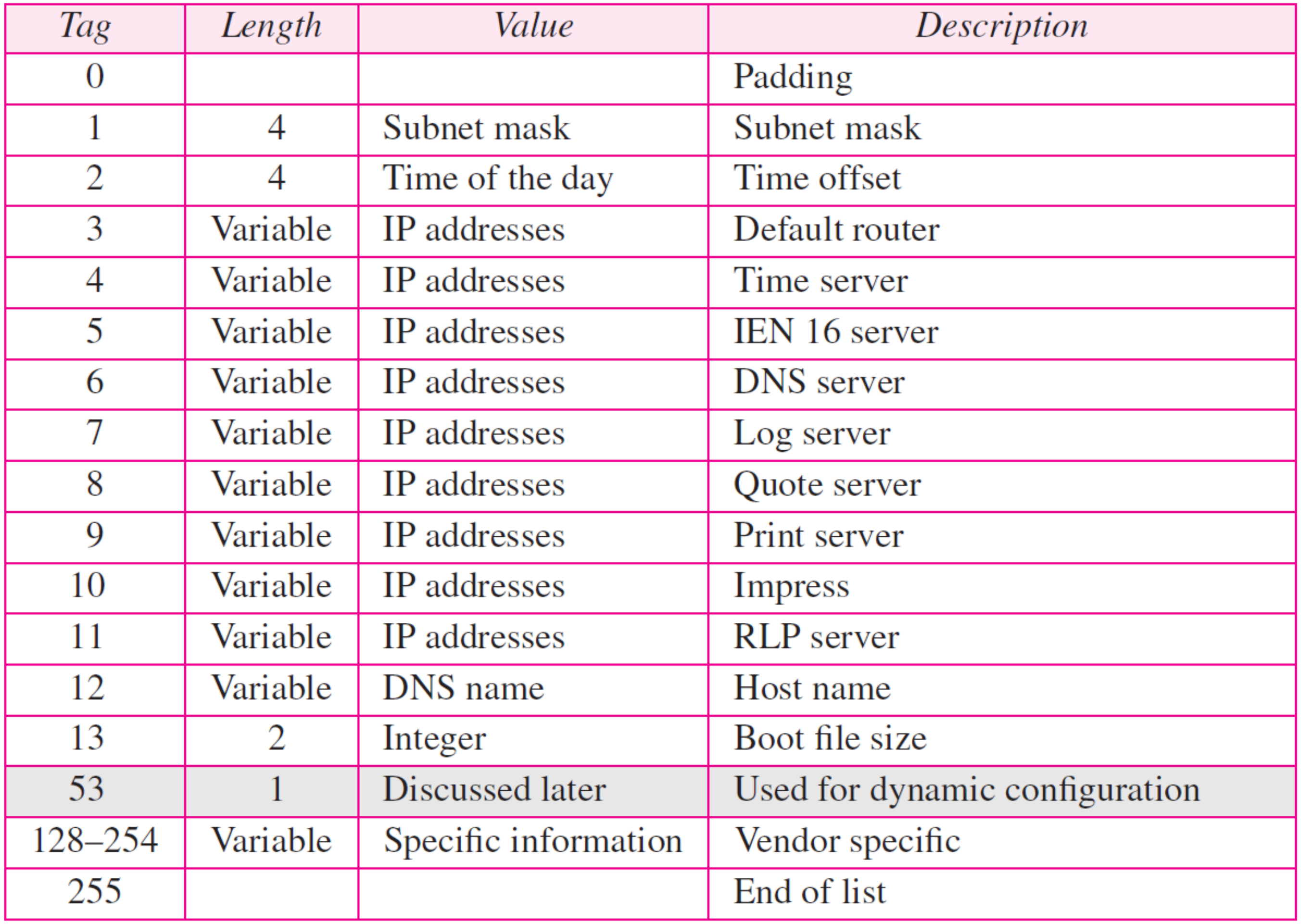
- **Boot filename.** This is a 128-byte field that can be optionally filled by the server in a reply packet. It contains a null-terminated string consisting of the full pathname of the boot file. The client can use this path to retrieve other booting information. If the server does not want to fill this field with data, the server must fill it with all 0s.

- **Options.** This is a 64-byte field with a dual purpose. **It can carry either additional information** (such as the network mask or default router address) or some specific vendor information. **The field is used only in a reply message**.
  - ✓ 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.** The length field defines the length of the value field, not the whole option.

Option Format



Options for DHCP



Tag	*Length*	*Value*	*Description*
0			Padding
1	4	Subnet mask	Subnet mask
2	4	Time of the day	Time offset
3	Variable	IP addresses	Default router
4	Variable	IP addresses	Time server
5	Variable	IP addresses	IEN 16 server
6	Variable	IP addresses	DNS server
7	Variable	IP addresses	Log server
8	Variable	IP addresses	Quote server
9	Variable	IP addresses	Print server
10	Variable	IP addresses	Impress
11	Variable	IP addresses	RLP server
12	Variable	DNS name	Host name
13	2	Integer	Boot file size
53	1	Discussed later	Used for dynamic configuration
128–254	Variable	Specific information	Vendor specific
255			End of list

3. CONFIGURATION

The DHCP has been devised to provide static and dynamic address allocation.

Static Address Allocation

- DHCP server has a database that statically binds physical addresses to IP addresses. When working in this way, DHCP is backward compatible with the deprecated protocol BOOTP, which we discussed before.

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.
- ❑ When a DHCP client sends a request to a DHCP server, the server first checks its static database. If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned. On the other hand, if the entry does not exist in the static database, the server selects an IP address from the available pool, assigns the address to the client, and adds the entry to the dynamic database.

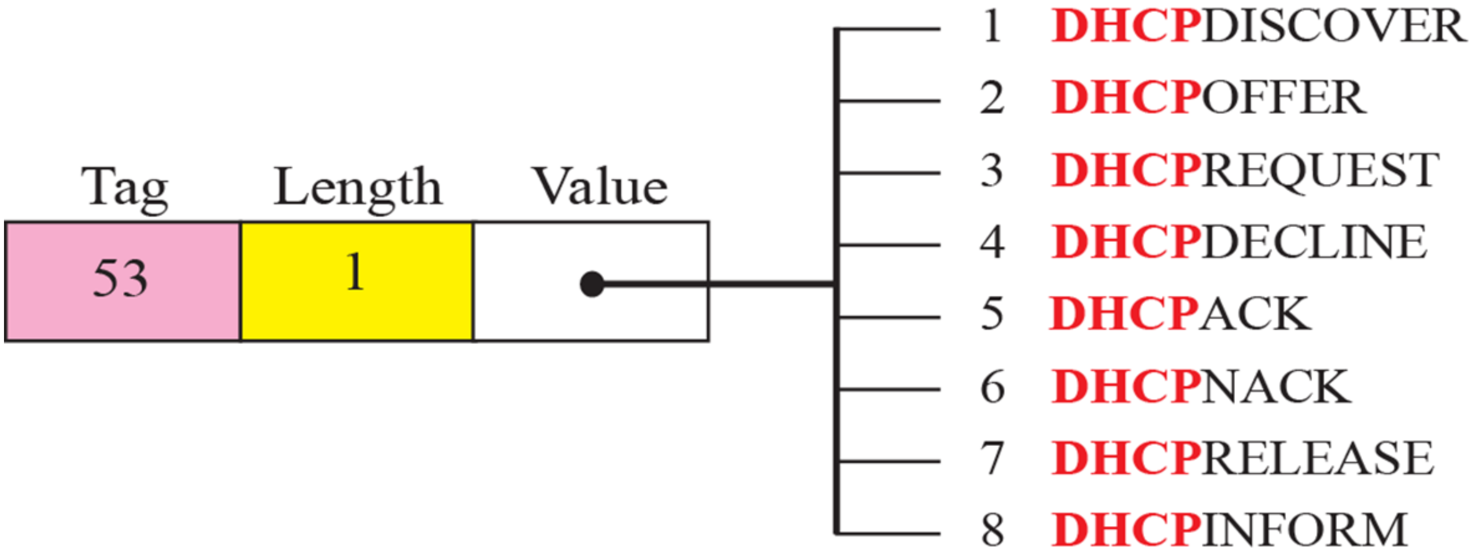
Dynamic Address Allocation

- The dynamic aspect of **DHCP** is needed when a host moves from network to network or is connected and disconnected from a network. DHCP provides temporary IP addresses for a limited period of time.

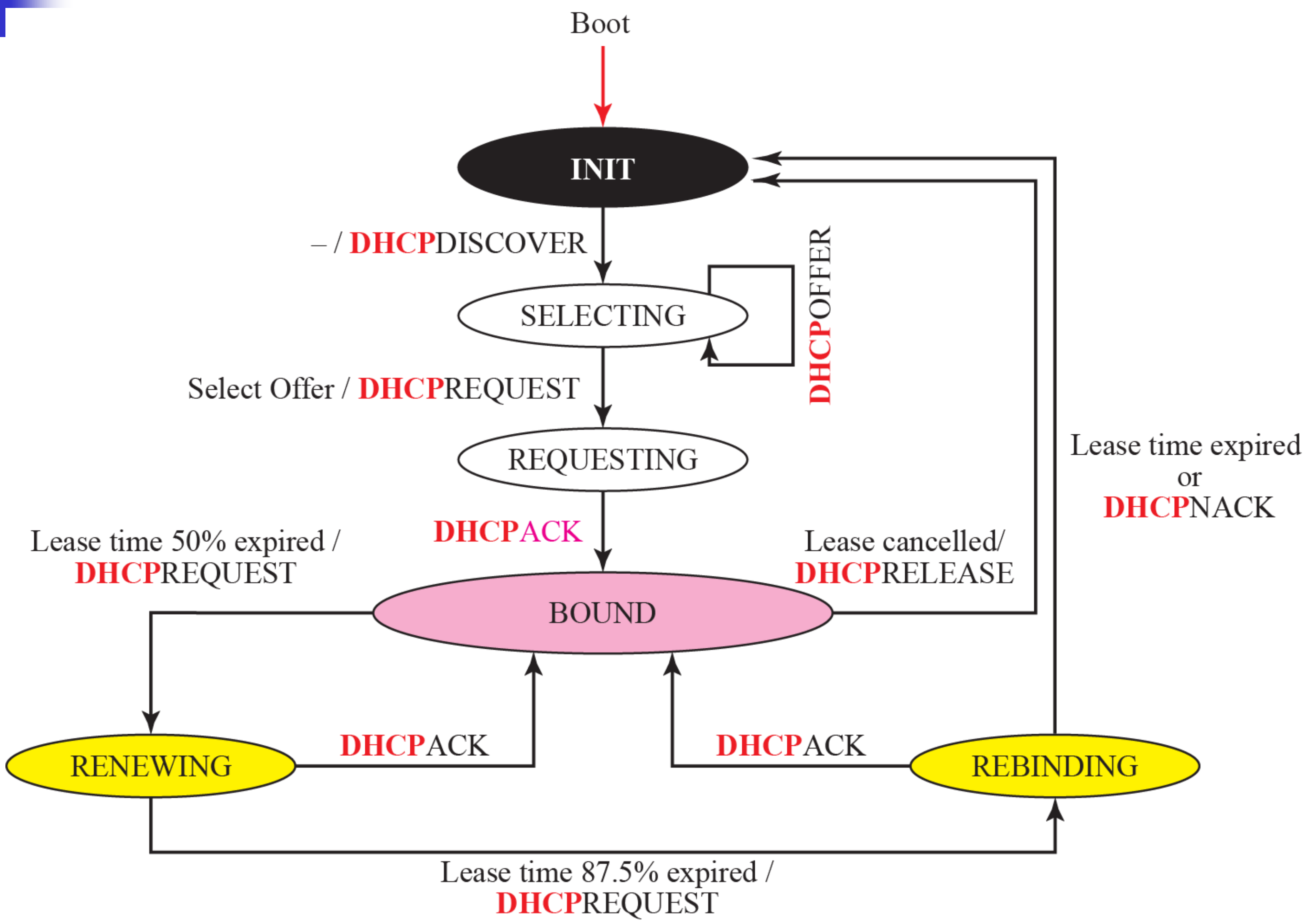
- The addresses assigned from **the pool** are temporary addresses. The DHCP server issues a lease for a specific period of time. When the lease expires, the client must either stop using the IP address or renew the lease. The server has the choice to agree or disagree with the renewal. If the server disagrees, the client stops using the address.

Transition States

- 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. The type of the message in this case is defined by the option with tag 53 that is included in the DHCP packet.



DHCP client transition diagram



Transition States

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**.

Transition States

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.

Transition States

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.

Transition States

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

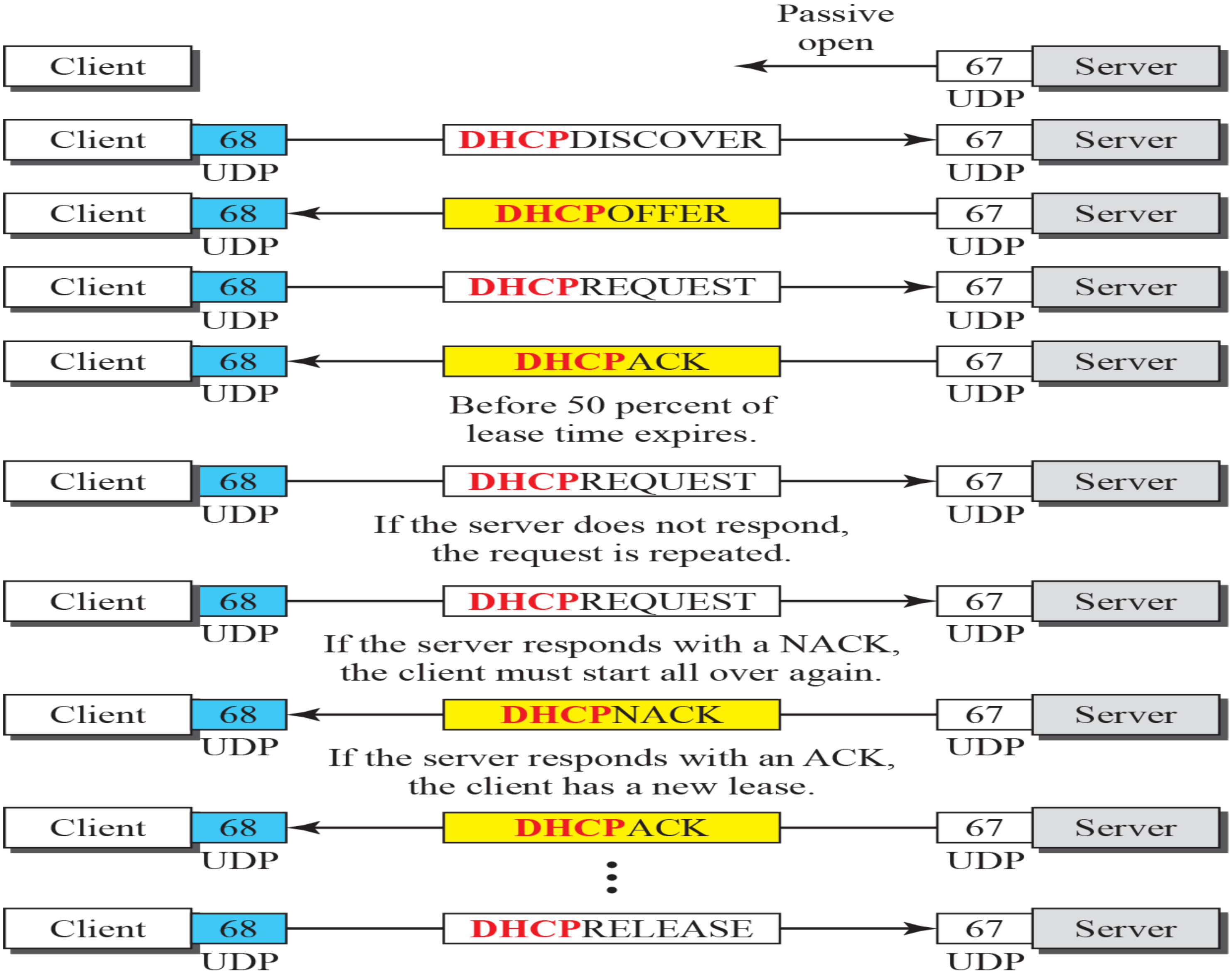
The client remains in the rebinding state until one of three events happens. If the client receives a **DHCPNACK** or the lease expires, it goes back to the initializing state and tries to get another IP address. If the client receives a **DHCPACK**, it goes to the bound state and resets the timer.

Early Release

- A DHCP client that has been assigned an address for a period of time may release the address before the expiration time.

- The client may send a **DHCPRELEASE** message to tell the server that the address is no longer needed. This helps the server to assign the address to another client waiting for it.

Exchanging Messages


Passive open
Client
67
UDP
Server
Client
68
UDP
DHCPDISCOVER
67
UDP
Server
Client
68
UDP
DHCPOFFER
67
UDP
Server
Client
68
UDP
DHCPREQUEST
67
UDP
Server
Client
68
UDP
DHCPACK
67
UDP
Server
Before 50 percent of lease time expires.
Client
68
UDP
DHCPREQUEST
67
UDP
Server
If the server does not respond, the request is repeated.
Client
68
UDP
DHCPREQUEST
67
UDP
Server
If the server responds with a NACK, the client must start all over again.
Client
68
UDP
DHCPNACK
67
UDP
Server
If the server responds with an ACK, the client has a new lease.
Client
68
UDP
DHCPACK
67
UDP
Server
Client
68
UDP
DHCPRELEASE
67
UDP
Server