Mission 5: Real-Time project on a "naked" computer

Tommaso Marinelli Matteo Di Pirro

December 20, 2017

1 User manual

2 Documentation for system engineers

3 Documentation for programmers

3.1 Specification

The program implements a DHCP relay. Figure 1 depicts how it works.

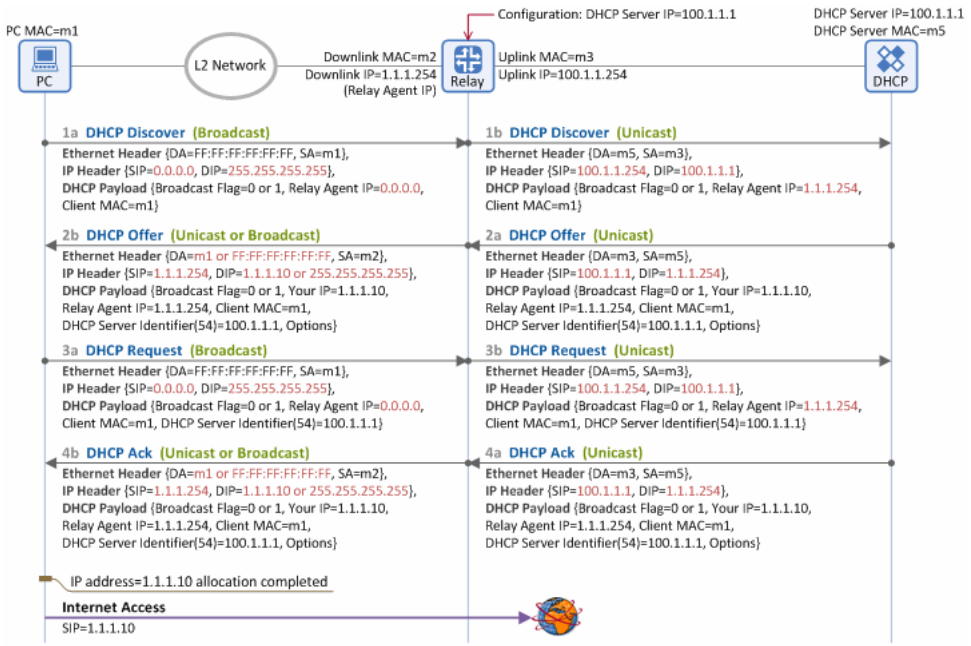

Figure 1: DHCP relay[1]

Basically, a DHCP relay is a "man in the middle" between the client and the DHCP server. It helps the client to contact the server in order to obtain an IP address. It does so receiving the broadcast packets sent by clients and forwarding them in an unicast connection to one (or more) DHCP servers, and vice versa, forwarding the server responses in broadcast to the clients.

While modeling its behavior, we refer to the above figure. We suppose to be able to detect when a message comes from a client and when it comes from a server by means of two different interruptions, without looking at the `OPCode` field (which is 1 for a request and 2 for a response). In other words, two *different events* occur. This is reasonable, since Figure 1 depicts two different links (*Downlink* and *Uplink*), with two different MAC and IP addresses.

Our ASG diagram is depicted in Figure 2.

[1] `https://www.netmanias.com/en/?m=view&id=techdocs&no=6000`

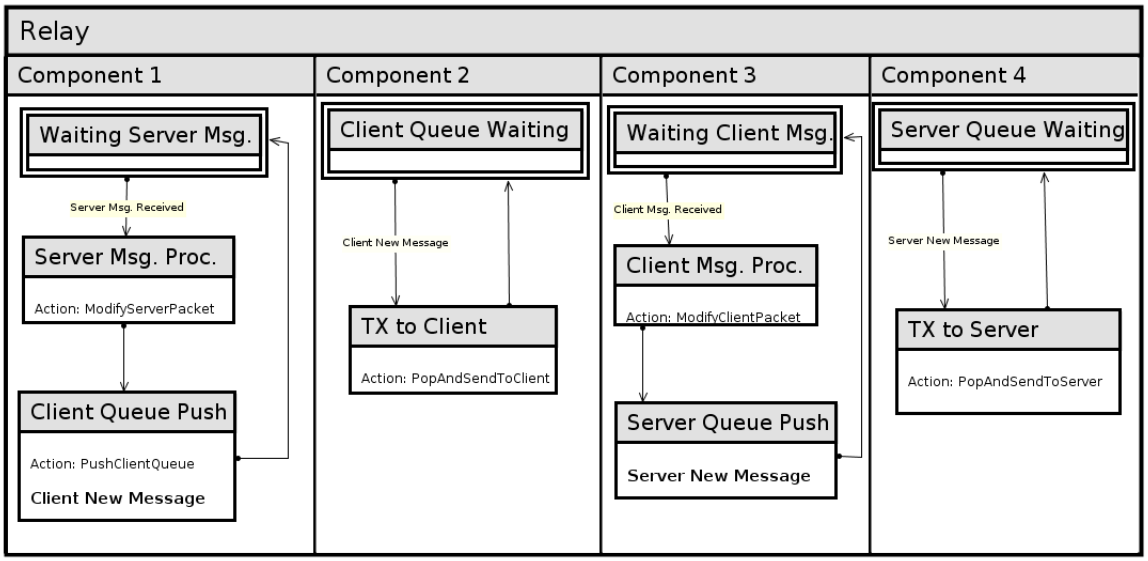

Figure 2: ASG diagram modeling a DHCP relay

A DHCP relay consists of four parallel components following a producer-consumer pattern. Two components, 1 and 3, wait for a packet, either from a client (broadcast) or a server (unicast). They then modify this packet as follows:

- **Component 1** - `ModifyServerPacket`:
  - **Ethernet Payload**
    - Destination MAC Address: DHCP Relay Uplink MAC → Broadcast
    - Source MAC Address: DHCP Server MAC → DHCP Relay MAC Address
  - **IP Payload**
    - Source IP Address: DHCP Server IP Address → DHCP Relay Downlink IP
    - Destination IP Address: DHCP Relay Downlink IP → Broadcast
- **Component 2** - `ModifyClientPacket`:
  - **Ethernet Payload**
    - Destination MAC Address: Broadcast → DHCP Server MAC
    - Source MAC Address: PC MAC Address → DHCP Relay Uplink MAC
  - **IP Payload**
    - Source IP Address: 0.0.0.0 (no IP address) → DHCP Relay Uplink IP
    - Destination IP Address: Broadcast → DHCP Server IP
  - **DHCP Payload**
    - Gateway IP Address (GIADDR): 0.0.0.0 → DHCP Relay Downlink IP

The two components then push the modified packet in a queue (two different queues are present, respectively for server and client) and set a rendez-vous (respectively **Client New Message** and **Server New Message**) to awake the corresponding consumer.

Component 2 and 4 represent the consumers. If they are waiting, they are awoke by a push of the corresponding producer. Once a message is received, they enter the transmission state, and leave it only when the queue is empty. Here they continuously pop an element from the queue and send it in broadcast (Component 2) or in unicast to one (or more) DHCP servers (Component 4).

This separation in two producers-consumers allow us to separate two logical different operations: handling the traffic incoming client-side and the one incoming server-side.

4 Program listings