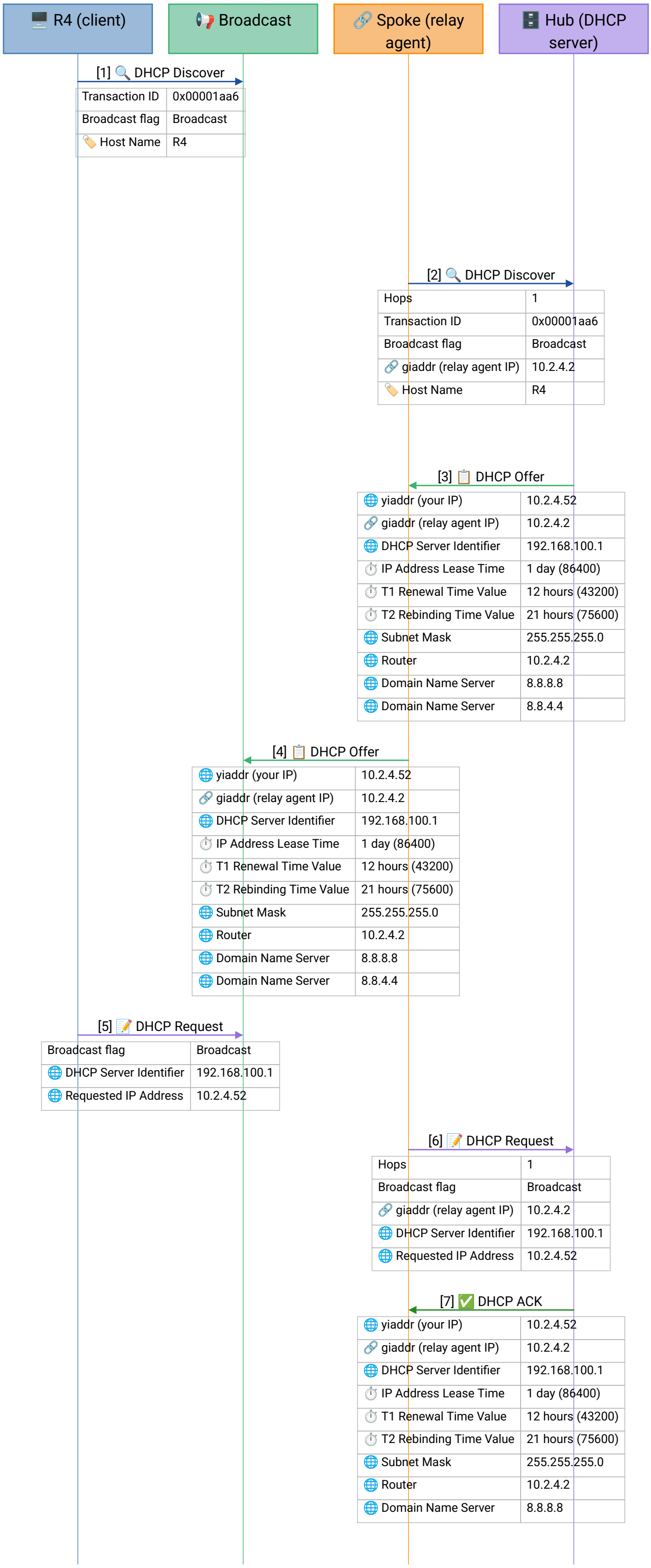


[Frame 1] Discover on the client's LAN.

R4 broadcasts a Discover from `0.0.0.0`, exactly as a client does with a server on the same segment. It doesn't know, or need to know, that no server is on this LAN.

- Transaction ID `0x1aa6`, broadcast flag set, host name `R4`.
- `Hops` is 0 and `giaddr` is empty: no relay has touched this packet yet.

Broadcasts stop at the router, so without help this Discover would go nowhere.

[Frame 2] The relay agent forwards it to the server.

The spoke router is configured as a DHCP relay agent (on Cisco IOS, `ip helper-address`). It turns the broadcast into a unicast to the server at `10.0.0.1`, through the DMVPN tunnel to the hub. Two header fields change:

- `giaddr` (gateway IP) `10.2.4.2`: the relay's own address on the client's LAN. The server uses it to choose the address pool, and sends its reply back to it.
- `Hops` goes from 0 to 1.

Note the ports: relay to server is UDP 67 to 67, not 68 to 67.

[Frame 3] Offer, back to the relay.

The hub's server answers about 2 seconds later. It picked `10.2.4.52` from the pool that matches `giaddr` `10.2.4.2`, and sets the default router to `10.2.4.2`, the relay's LAN interface.

The reply goes to the relay (UDP 67 to 67), not to the client. Its source, and the `Server Identifier`, is `192.168.100.1`: the server's own identity, not the tunnel address `10.0.0.1` the relay sent to.

The 2-second wait fits a server that checks the address is free before offering it; RFC 2131 suggests probing with an ICMP echo. That check happens at the hub and isn't in this capture.

[Frame 4] The relay delivers the Offer to the client.

The spoke sends the Offer onto R4's LAN from UDP 67 to 68. Because R4 set the broadcast flag, the relay broadcasts it to `255.255.255.255`. `giaddr` stays `10.2.4.2` in the copy.

[Frame 5] Request, broadcast on the LAN.

R4 accepts with `Requested IP Address` `10.2.4.52` and `Server Identifier` `192.168.100.1`. It names the server by the identifier from the Offer; it never sees the relay's role.

[Frame 6] The relay forwards the Request.

Same treatment as the Discover: unicast through the tunnel to `10.0.0.1`, `giaddr` `10.2.4.2`, `Hops` `1`.

[Frame 7] ACK to the relay.

No delay this time: the ACK follows the Request by half a millisecond. The lease is one day (T1 12 hours, T2 21 hours), on `10.2.4.52/24`, default router `10.2.4.2`.



[Frame 8] The relay delivers the ACK.

Broadcast on R4's LAN. R4 now owns `10.2.4.52`. Right after this, outside the diagram, R4 sends a gratuitous ARP for its new address and pings the hub at `10.0.0.1` through the tunnel; all five echoes are answered.