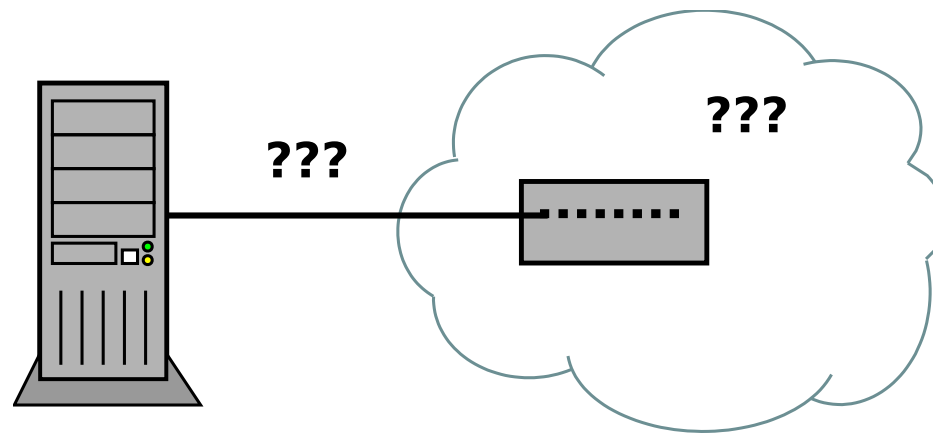


DHCP

DHCP (Dynamic Host Configuration Protocol)

è un protocollo estremamente diffuso per la
configurazione automatica
degli host che si connettono a una rete



DHCP: caso di studio

È interessante da studiare perché molto (molto) diffuso e perché si basa su protocolli che agiscono contemporaneamente a diversi livelli dello stack TCP/IP.

- È un *protocollo applicativo* (basato su *UDP*)
- Serve a *configurare il livello di rete* (IP)
- La maggior parte delle informazioni vengono trasmesse *direttamente tramite protocolli H2N*

DHCP: Goal

DHCP is designed to supply DHCP clients with the configuration parameters defined in the Host Requirements RFCs. ***After obtaining parameters via DHCP, a DHCP client should be able to exchange packets with any other host in the Internet.***

(RFC 2131, §1.3)

Not all of these parameters are required for a newly initialized client.

...

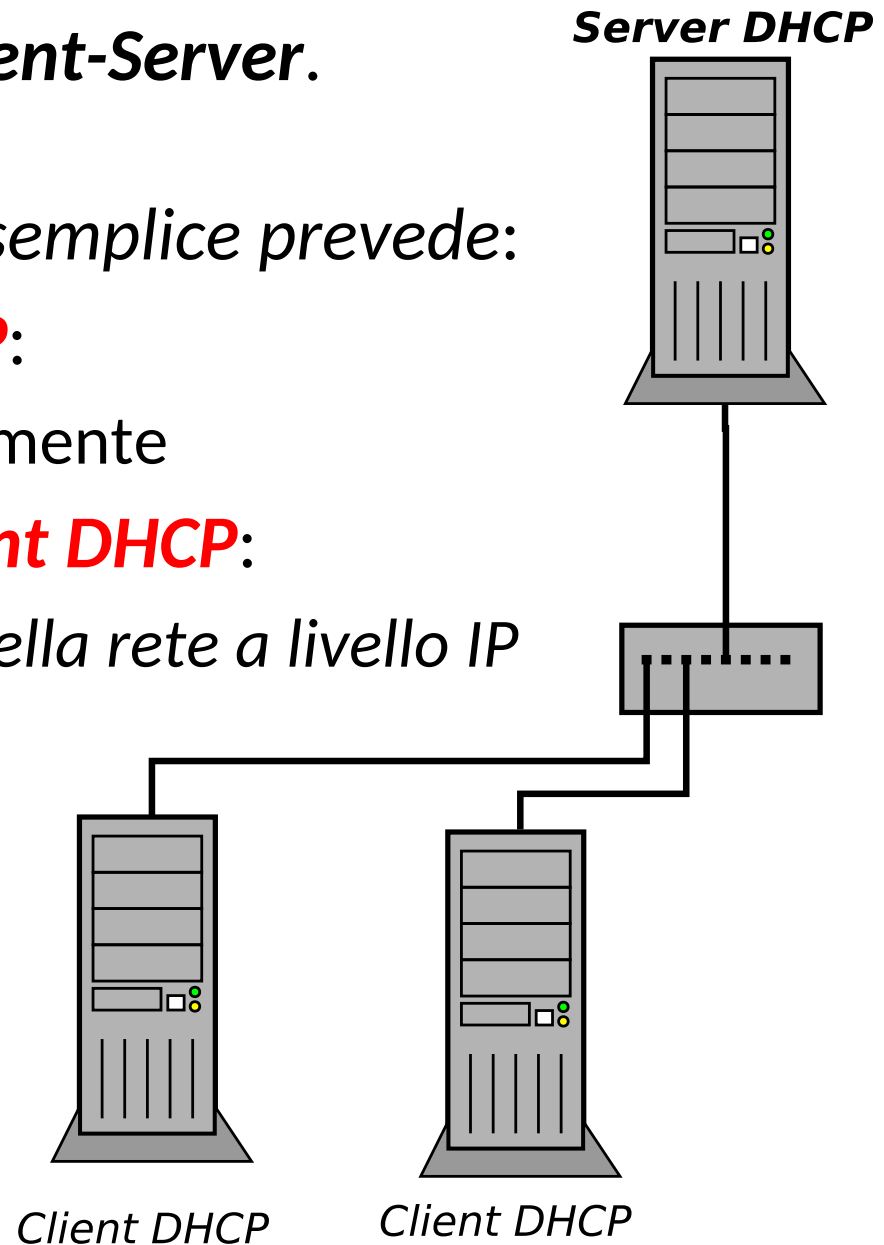
DHCP is not intended for use in configuring routers.

DHCP: Architettura di partenza

DHCP è basato su protocolli *Client-Server*.

La soluzione architetturale più semplice prevede:

- un host esegue **il server DHCP**:
 - ◊ l'host è preconfigurato staticamente
- gli altri host eseguono un **client DHCP**:
 - ◊ *gli host non conoscono nulla della rete a livello IP*



DHCP: Obiettivi progettuali

- 1) Fornire i meccanismi attraverso i quali un amministratore può implementare delle **policy di rete**:
 - gestione di tutte le informazioni di configurazione della rete
- 2) Gli host che si connettono alla rete devono configurarsi **automaticamente**, senza alcuna informazione preliminare:
 - sviluppo di protocolli di supporto per TCP/IP, mantenendo l'interoperabilità con gli host configurati staticamente.
- 3) Implementabile su di una **architettura scalabile**:
 - *distribuzione/replicazione* delle informazioni di configurazione su più host;
 - funzionante su *una o più subnet di rete* (ovvero, il protocollo passare attraverso dei router).

(1) Policy di rete

1) Fornire i meccanismi attraverso i quali un amministratore può implementare delle **policy di rete**:

- policy di **assegnazione degli indirizzi IP** agli host della rete
- **regole aggiuntive** (e.g., routing statico, nomi degli host)
- sono **persistenti** al riavvio dei server e dei client DHCP

*Tutte le informazioni sono memorizzate in uno o più **database** gestiti dai server DHCP.*

Policy di rete - Gestione degli indirizzi

Dati uno o più *range di indirizzi (subnet)* al server:

- **Assegnazione manuale:**

il binding fra host e indirizzi IP viene definito manualmente sul database

→ la configurazione della rete è **statica**, ma la rete è più facile da configurare (centralizzazione della configurazione)

- **Assegnazione dinamica** (*permanente/temporanea*):

il server decide autonomamente la distribuzione degli indirizzi. Gli indirizzi IP vengono riutilizzati dopo un certo periodo di tempo (**lease**)

→ ottimizzazione del numero degli indirizzi a disposizione

Policy di rete - Database DHCP

DHCP memorizza le informazioni di rete tramite un *database chiave-valore*, in cui:

- la chiave *identifica univocamente un host*. È una coppia composta dal NetID della subnet più un identificativo dell'host. Quello predefinito è l'HW address (i.e. indirizzo **MAC** in Ethernet) ma possono essere utilizzati altri valori (e.g., **hostname**, **client id**)
- il valore contiene l'indirizzo IP da assegnare all'host ed eventuali opzioni di configurazione (e.g., **gateway**, **hostname**, **DNS server**)

Ad esempio:

Key: (IP-subnet-number, HW-Address)

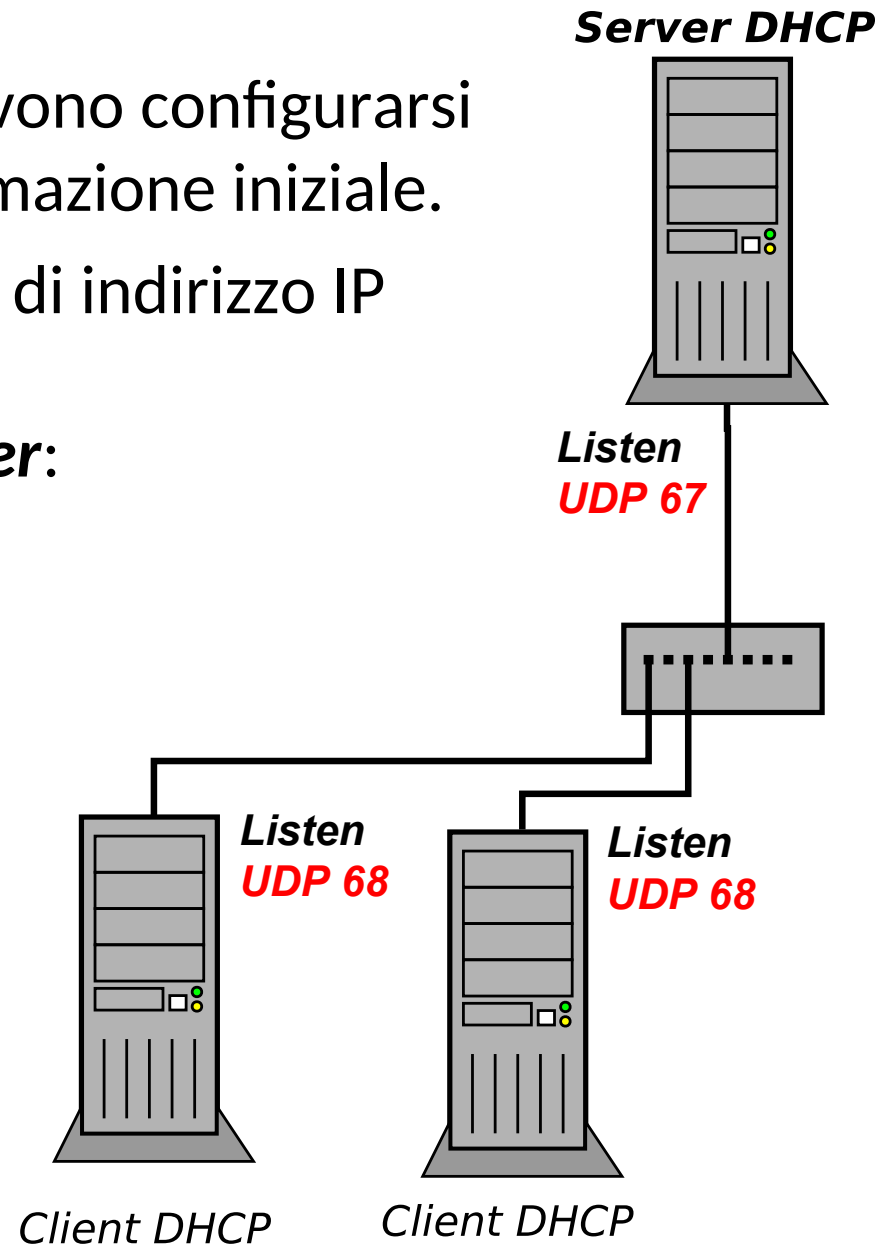
Value: { IP-Address, hostname, default gw }

(2) DHCP: Protocolli di comunicazione

- 2) Gli host che si connettono alla rete devono configurarsi **automaticamente**, senza alcuna informazione iniziale.
- ♦ Inizialmente il client non è provvisto di indirizzo IP

DHCP è basato su protocolli **Client-Server**:

- Il server DHCP rimane in ascolto sulla porta **UDP 67**
- Il client DHCP rimane in ascolto sulla porta **UDP 68**



Protocolli: Pacchetto DHCP

0

31

<i>op</i> (1)	htype (1)	hlen (1)	hops (1)
<i>xid</i> (4)			
secs (2)		flags (2)	
<i>ciaddr</i> (4)			
<i>yiaddr</i> (4)			
<i>siaddr</i> (4)			
giaddr (4)			
<i>chaddr</i> (16)			
sname (64)			
file (128)			
<i>options</i> (variable)			

op: *msg type*

xid: *transaction ID*

ciaddr: client IP address

yiaddr: your IP address

siaddr: next server IP

chaddr: client HW addr

options: additional info

Tipologia dei Messaggi: Client → Server

OP=1

- **DHCPDISCOVER**: un client non *configurato* vuole contattare per la prima volta il server DHCP tramite comunicazione **broadcast**
- **DHCPREQUEST**: un client richiede *conferma* per un indirizzo IP
 - ♦ risoluzione di conflitti in presenza di più server DHCP
 - ♦ conferma di un indirizzo precedentemente ricevuto
 - ♦ rinnovo di un indirizzo IP ricevuto in lease
- **DHCPDECLINE**: rifiuto di un indirizzo già allocato (e.g., duplicato)
- **DHCPRELEASE**: rilascio esplicito di un indirizzo IP (e.g., prima della disconnessione dell'host)
- **DHCPINFORM**: richiesta dei soli parametri di connessione, non dell'indirizzo IP (e.g., host con IP configurato staticamente)

Tipologia dei Messaggi: Server → Client

OP=2

- **DHCPOFFER**: offerta di un indirizzo DHCP per un client
 - ♦ non riserva l'indirizzo lato server
- **DHCPACK**: accettazione dell'indirizzo proposto dal client e invio informazioni di configurazione
- **DHCPNAK**: rifiuto dell'indirizzo proposto dal client

Workflow: caso più comune (1)

Client

Server

DHCPDISCOVER (op=1)

generazione transaction id random (**xid**)

[suggested *IP Address* e una *lease duration*]

broadcast dH2N e dIP, sIP 0.0.0.0, sUDP68, dUDP 67 →

DHCPOFFER (op=2)

Scelta/lookup indirizzo IP (**yaddr**)

[test IP]

← invio sIP personale, dIP yaddr, sUDP 67, dUDP 68

Workflow: caso più comune (2)

DHCPREQUEST

inserimento del server destinatario come server identifier (**siaddr**),
conferma esplicita dell'indirizzo accettato (**yiaddr**)



DHCPACK : OK

conferma il binding (**client id + addr**)

← conferma esplicita dell'indirizzo assegnato (**yiaddr**)

← **DHCPNAK : KO**

Test delle impostazioni

If test fail:

KO: DHCPDECLINE →

else: **OK:** memorizzazione locale, [nota tempi lease]

Workflow: note (1)

- Il transaction ID (xid) generato nel primo messaggio dal client viene inserito in ogni messaggio successivo. Se il client riceve risposte corrette ma con txid diverso scarta i pacchetti senza rispondere.
- Come fa il client a ricevere pacchetti destinati a un indirizzo IP specifico (DHCPRESPONSE, DHCPACK, DHCPNAK) se l'host non ha ancora un indirizzo IP?

→ Da rfc, i pacchetti in ingresso da un'interfaccia non configurata non dovrebbero mai essere scartati automaticamente

Workflow: note (2)

- Come verificare l'assenza di conflitti nell'assegnazione (server) e nell'impostazione (client) degli indirizzi IP?
 - Impiego di protocolli standard dello stack TCP/IP (e.g., ICMP echo request, ARP).
- Perché un server identifier nel campo applicativo e l'impiego di indirizzo broadcast anche nella DHCPREQUEST? Perché l'indicazione esplicita degli indirizzi HW in campi applicativi?
 - Più server DHCP e relay agent possono convivere sulla stessa rete: il client può decidere quali offerte seguire.

DHCP client con NetworkManager

- Utilizziamo il client DHCP integrato con il servizio **NetworkManager**, disponibile come un servizio all'interno del sistema Linux in uso

systemctl {start,stop,restart,status} NetworkManager

- Possiamo interagire con la configurazione attuale da command line tramite il tool **nmcli**

nmcli connection {show,edit} [ifname]

- Esempio:

```
# nmcli connection show
```

NAME	UUID	TYPE	DEVICE
Wired connection 1	82943526-eca9-...	ethernet	eth0
lo	ada97ca1-c660-...	loopback	lo

DHCP client con ifupdown

- Le configurazioni create con NetworkManager sono già persistenti (a meno di non specificare esplicitamente che non lo sono)
- Visto che di default NetworkManager attiva un client dhcp, per attivare in modo persistente il client dhcp è possibile avviare NetworkManager all'avvio con il predicato **enable** di systemctl
- Altrimenti, se non vogliamo usare NetworkManager, possiamo usare l'opzione **dhcp** al posto di **static** nel file interfaces
- Se vogliamo utilizzare entrambi i sistemi contemporaneamente (e.g., eth0 configurata con ifupdown, eth1 configurata con NetworkManager), per evitare conflitti è necessario esplicitare di lasciare fuori dalla configurazione alcune interfacce
 - Vedere ad esempio il file **/etc/NetworkManager/conf.d/99-unmanaged-devices.conf** nelle macchine marionnet

DHCP server con DNSmasq [1]

- DNSmasq è un servizio molto popolare in contesti di reti non troppo complesse per aver name server DNS e DHCP server
- Presente come un servizio gestibile con `systemctl` (come appena osservato per NetworkManager)
- La configurazione si basa sul file **`/etc/dnsmasq.conf`**

DHCP server con DNSmasq [2]

- Nel file `/etc/dnsmasq.conf` sono presenti le varie opzioni di configurazione per le diverse informazioni di configurazione
- Esempio con opzioni principali:
 - `dhcp-range=<ip-min>,<ip-max>,<lease-time>`
 - `dhcp-option=option:netmask,<local-net-ext-netmask>`
 - `dhcp-option=option:router,<default-gw>`
 - `dhcp-option=option:classless-static-route, \`
`<network-#address>/<cidr-nm>,<router>`
 - `dhcp-option=option:dns-server,<local-nm-ip>`
 - `dhcp-host=<mac-address>,<ip-address>`
- Nota: si dovrebbe usare l'opzione **router** o l'opzione **classless-static-route**, perchè client che ricevono quest'ultima possono ignorare la prima

DHCP relay agent

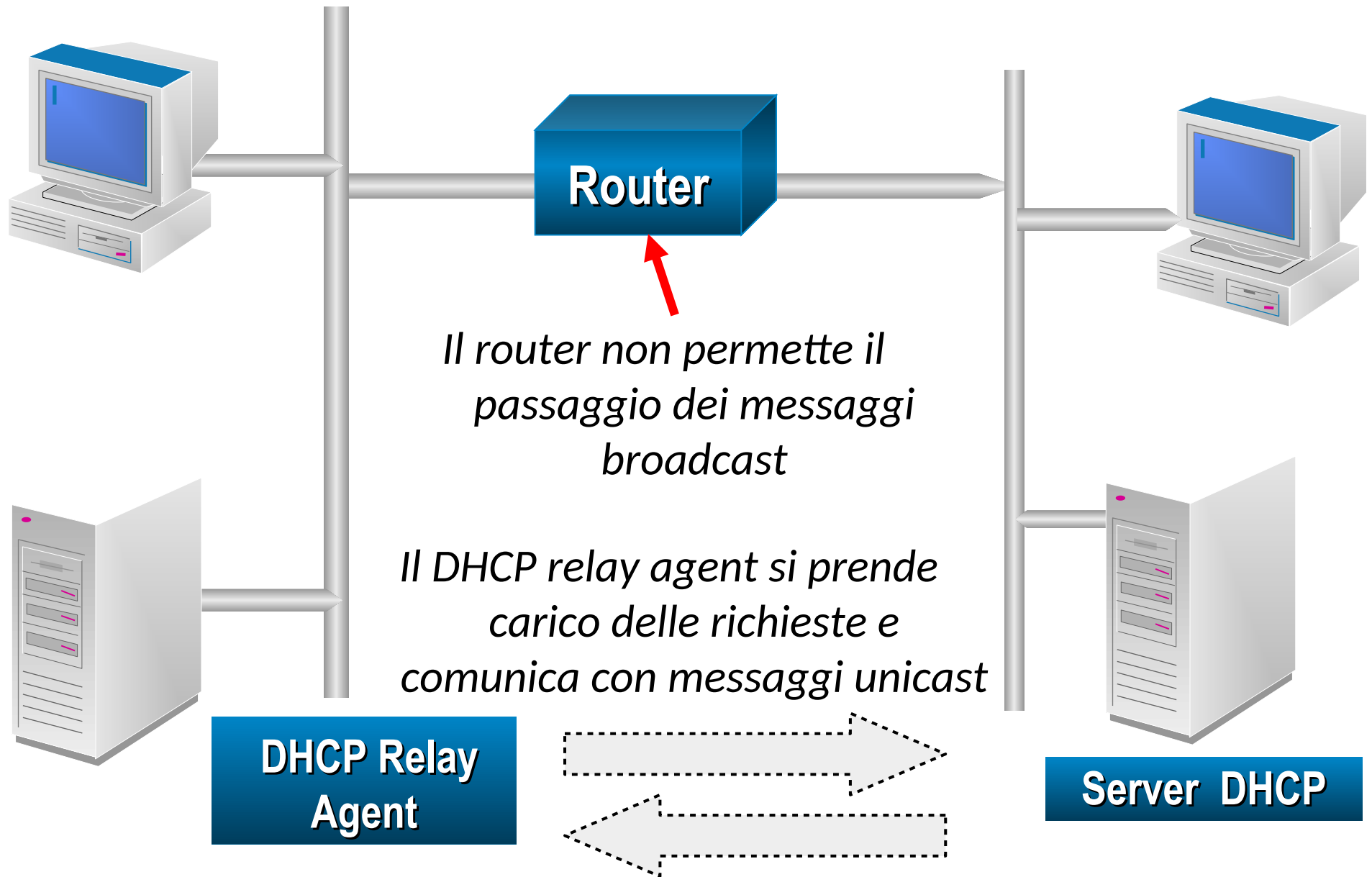
Il protocollo DHCP visto fin'ora si basa su pacchetti **broadcast**, che però non vengono inoltrati dai router.

Risultato: il server DHCP può configurare solo gli host collegati a livello H2N.

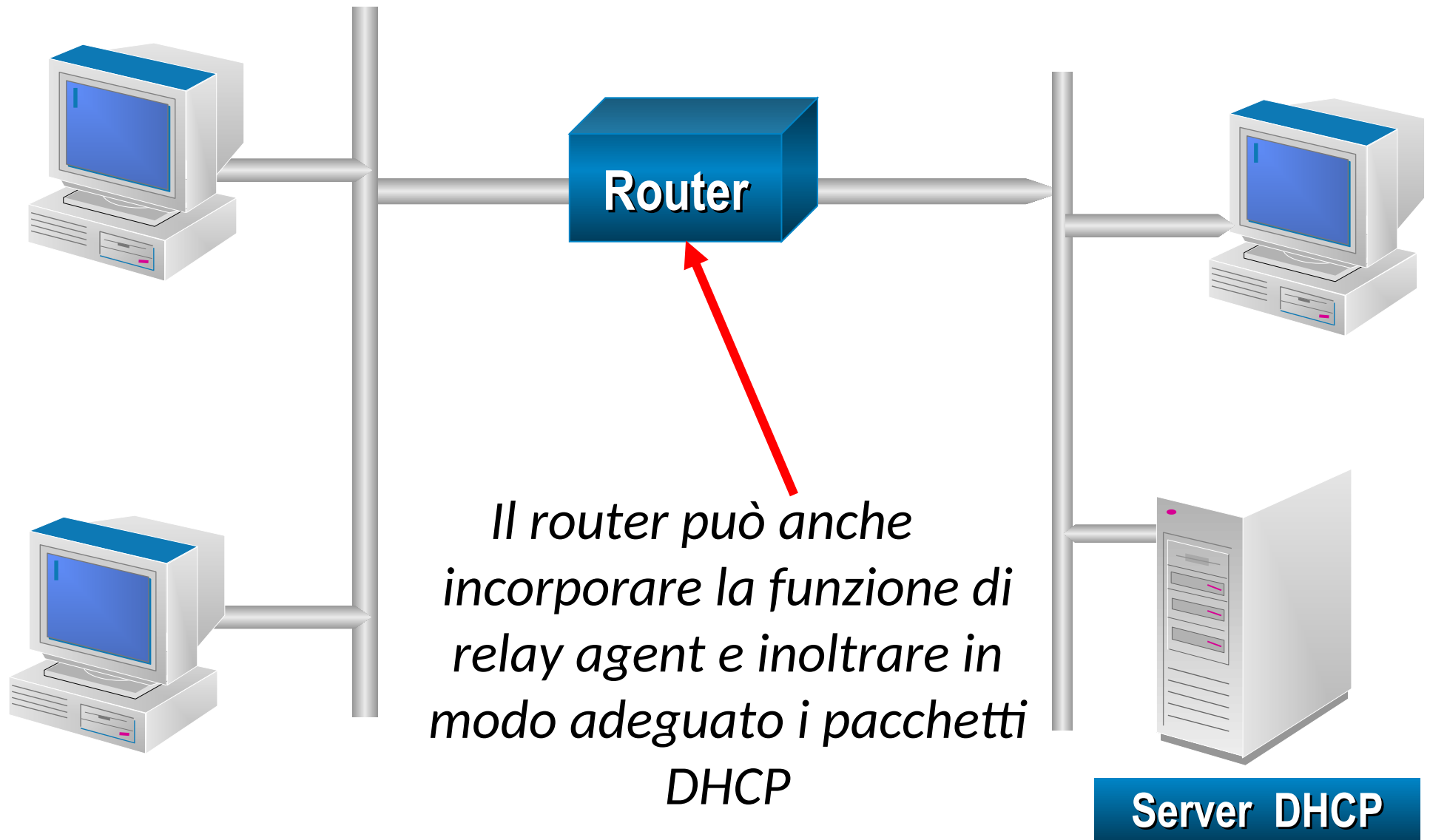
Per configurare host di diverse subnet senza installare diversi server DHCP si ricorre ai **Relay agent**. Il relay agent è un intermediario: è collegato in H2N e inoltra le richieste al server DHCP di rete.

Vantaggio: la configurazione del relay agent è molto semplice, e tutta la complessità di configurazione della è mantenuta sul server DHCP

DHCP relay agent (2)



DHCP relay agent (3)



DHCP relay agent - DNSmasq

- Possiamo usare DNSmasq anche per funzionare con relay agent tramite l'opzione

dhcp-relay=<local-ip>,<dhcp-server-ip>

Nota 1: <local-ip> riguarda la rete in cui il dhcp relay deve servire dei client (possibili ambiguità in relay agent eseguiti su router)

Nota 2: sul dhcp server è necessario inserire configurazioni per le diverse reti da servire. È possibile disambiguare le opzioni da servire tramite il campo **tag:<name>** all'inizio del valore dell'opzione

- Esempio: **dhcp-option=tag:lan1,option:...**

Esercizio - Schema

Configurare la rete in figura:

LAN1: 192.168.0.0/24

LAN2: 192.168.1.0/24

dhcp: 192.168.0.252/24

g12:

eth0= 192.168.0.253/24,

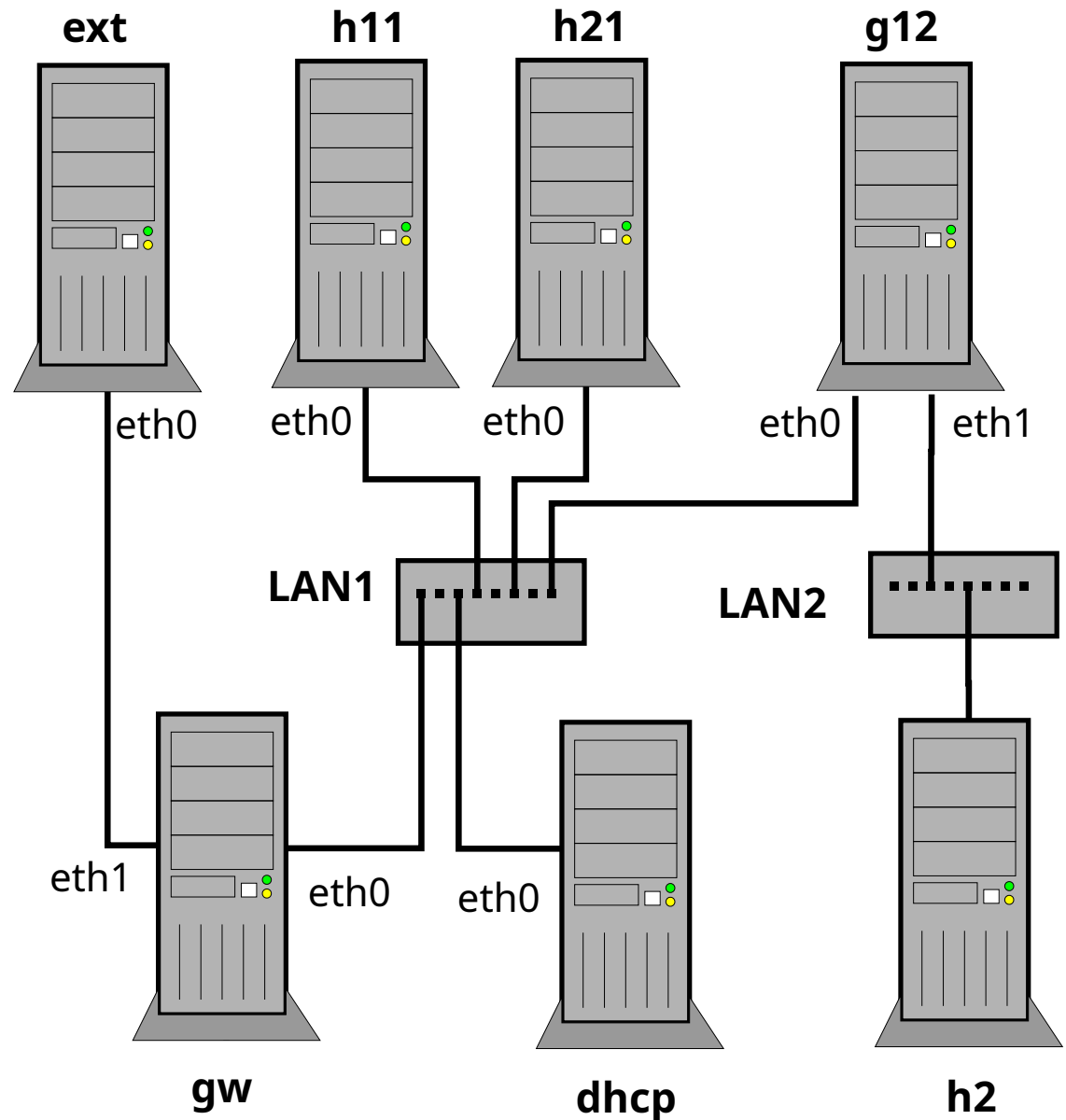
eth1= 192.168.1.254/24

gw:

eth0= 192.168.0.254/24,

eth1= 1.1.1.1/32

ext: 2.2.2.2/32



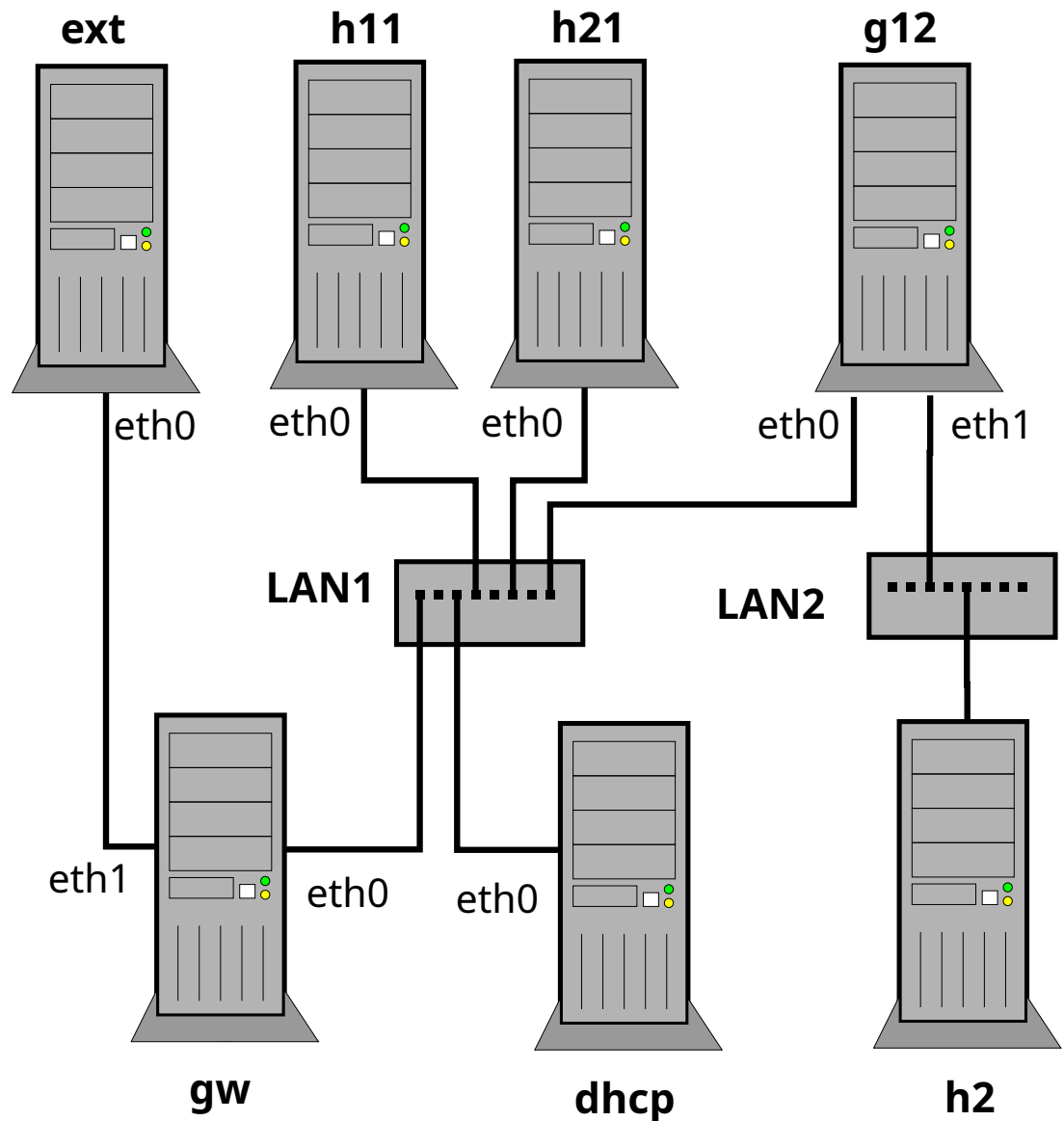
Esercizio - Scenario 1

*Configurare gli host della rete
nelle seguenti modalità:*

h11: dhcp, dinamico

h21: dhcp, 192.168.0.10

h2: statico, 192.168.1.1



Esercizio - Scenario 2

*Configurare gli host della rete
nelle seguenti modalità:*

h11: dhcp, dinamico

h21: dhcp, 192.168.0.10

h2: dhcp, dinamico

