

Boot me Boot me

Ogni elaboratore ha bisogno di un'informazione per lavorare (in rete).


Pochi elaboratori e nessuna mobilità -> si può procedere manualmente.

Molti elaboratori (mobili) -> il caos:

Cosa occorre per partire

Al minimo un indirizzo ip e una netmask
ma anche :

 1 default gateway

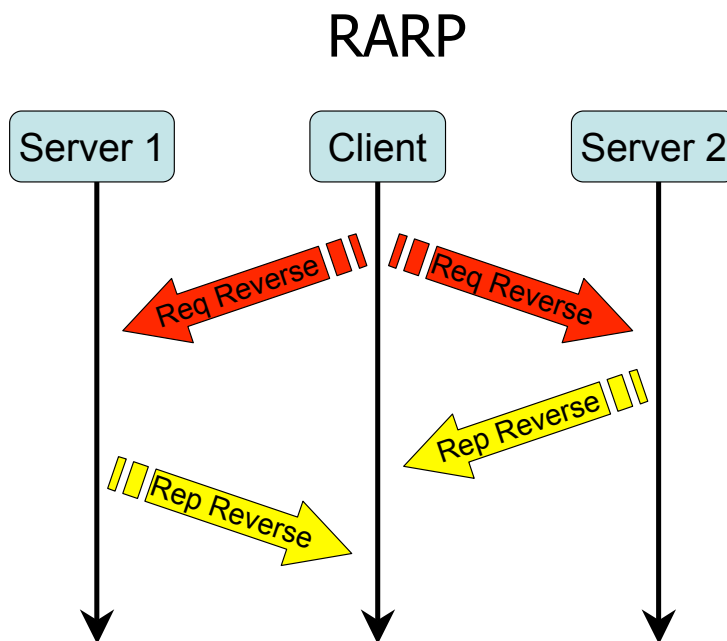
 1 dns

 1 server da cui caricare il sistema operativo

 etc...

primo tentativo RARP

- 🌻 Protocollo di livello 2
- 🌻 Stesso formato di pacchetto di ARP
- 🌻 Differente layer 2 protocol type
- 🌻 Multi server



Rarp: difetti

- 🌻 Pensato per ottenere solo 1 indirizzo IP
- 🌻 layer 2 -> forte dipendenza dallo HW/kernel
- 🌻 Nessuna negoziazione fra Client/Server
- 🌻 Multi server ma nessun coordinamento fra i server
- 🌻 funzionamento solo entro un dominio di broadcast

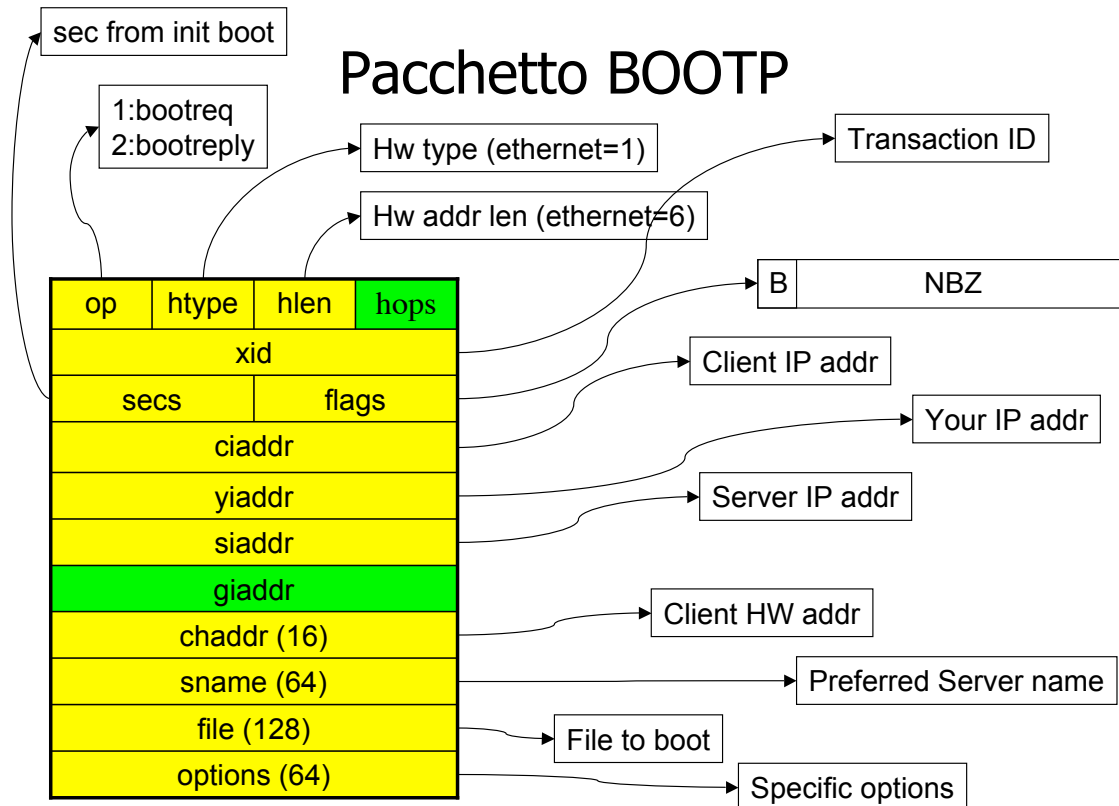
BOOTP

- 🌻 Protocollo di livello 3 (IP)
- 🌻 Meccanismi di negoziazione Client/Server
- 🌻 Meccanismi di assegnamento temporaneo di IP
- 🌻 Multi server
- 🌻 Possibilità di poter far arrivare al client TUTTE le informazioni necessarie per il funzionamento
- 🌻 Integrazione con altri protocolli (TFTP) per boot remoto

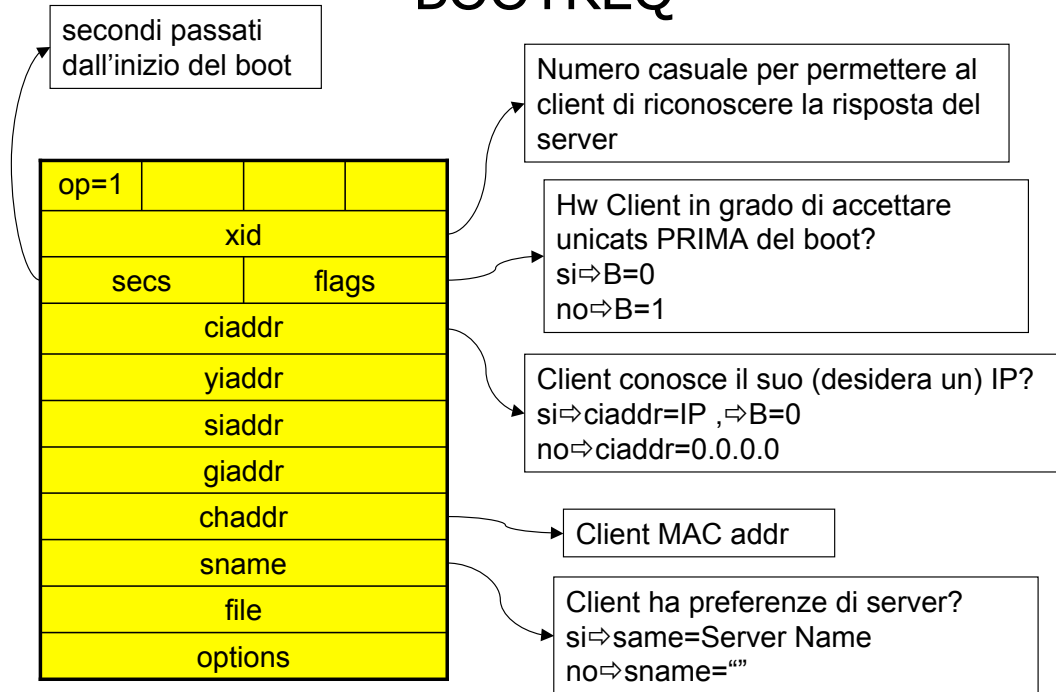
BOOTP: highlight

- 🌻 2 soli tipi di messaggi : BOOTREQUEST,BOOTREPLY
- 🌻 Il server ascolta sulla porta 67
- 🌻 Il client ascolta sulla porta 68
- 🌻 1 solo tipo di pacchetto
- 🌻 Funzionamento in più domini di broadcast

...Perché 2 porte distinte?

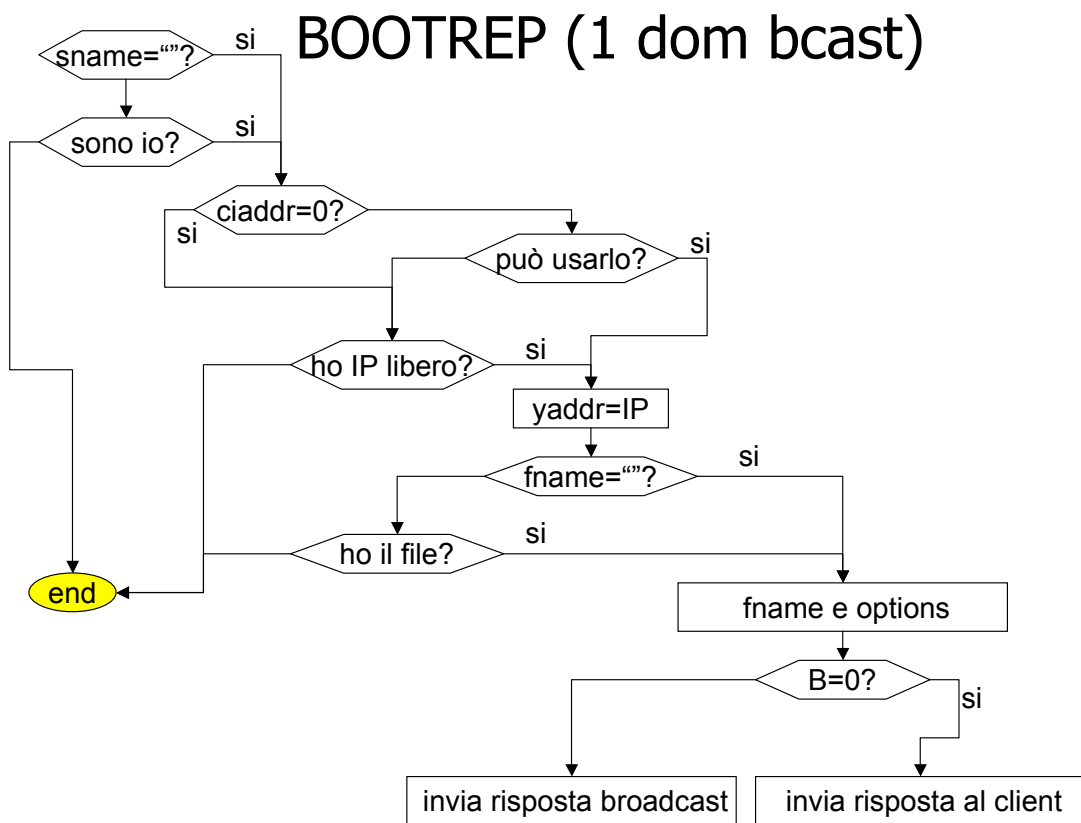


BOOTREQ



BOOTREQ: campo file

- 🔥 Identificativo per indicare al server che tipo di client è (es:sun oppure linux)
- 🔥 Nulla (""): client standard /non interessato al boot remoto ma solo a parametri.



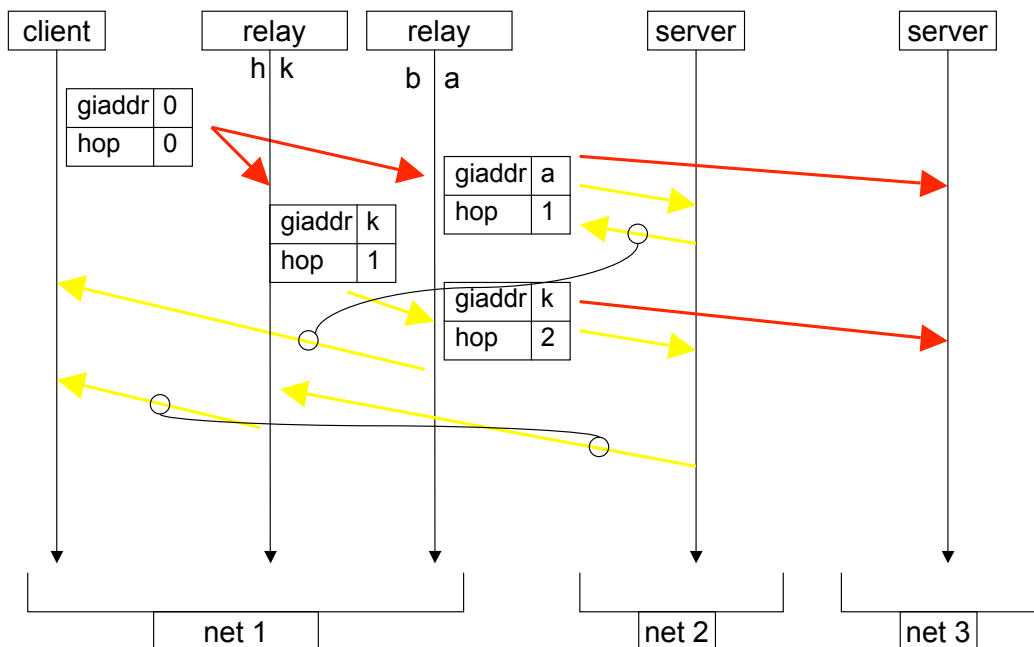
In the large

- 🌸 Al cresce del numero di host occorre creare sottoreti
- 🌸 bootp si basa su brodcast: funzionamento solo entro lo stesso dominio di broacast
- 🌸 Possibili soluzioni :
 - 🌸 Un server per ogni sottorete
 - 🌸 Un server unico e uso di proxy-agent (bootp relay agent)

BOOTP Relay Agent

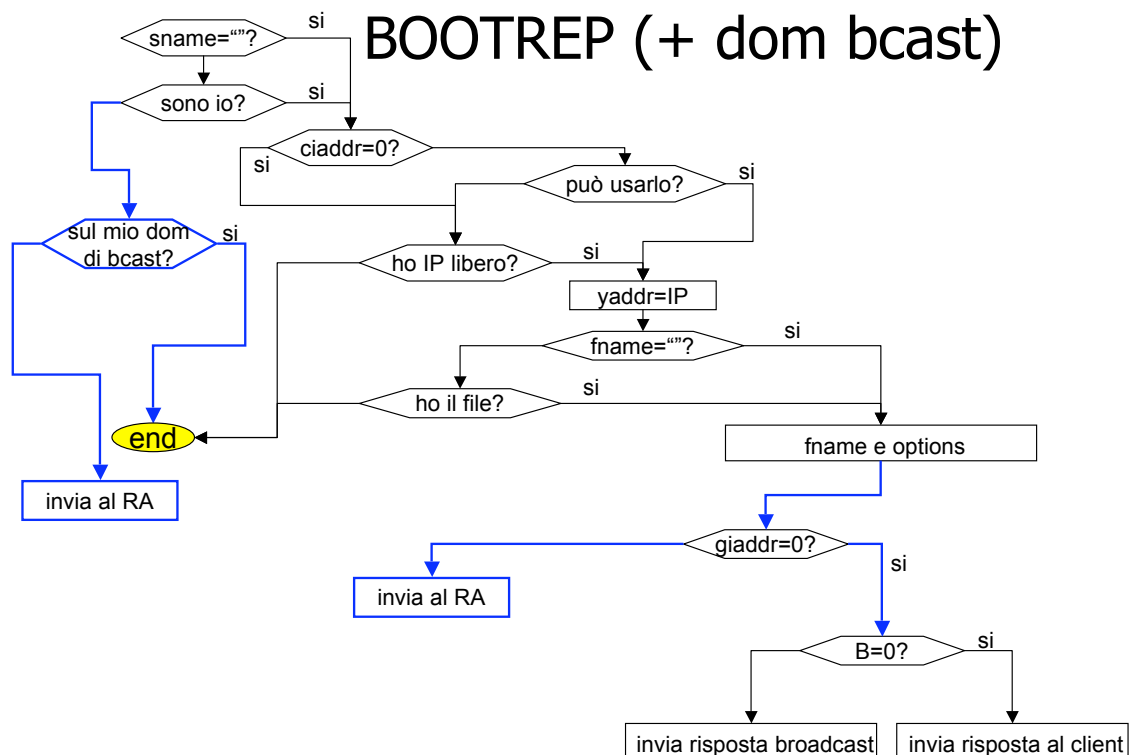
- ☀️ Si comporta come un proxy.
- ☀️ NON si comporta come un router IP
- ☀️ Accetta richieste da client in una sottorete ed emette richieste in altre sottoreti
- ☀️ Propaga all'indietro le risposte
- ☀️ Usualmente è integrato dentro router
- ☀️ Usa i campi giaddr e hop del pacchetto bootp

relay (esempio)



Relay Agent (RA):note

- 🔥 RA Ignora i pacchetti sulla porta bootpc (67)
- 🔥 RA Può usare il campo sec come 1 fattore per decidere sul relay
- 🔥 RA Ignora i BOOTPREQ con hop > soglia (max 16)
- 🔥 RA Incrementa sempre hop ad ogni forward
- 🔥 RA Modifica giaddr solo se e vuoto
- 🔥 Come (uni,multi,broad) e a chi fare il relay è una scelta dell'amministratore di rete
- 🔥 RA **deve** usare lo stesso insieme di destinatari per ogni determinato client
- 🔥 RA fa il replay di pacchetti BOOTREPLY solo a client e non ad altri RA



BOOTPREPLY: riassunto

| BOOTPREQ field | | | BOOTREPLY value | | |
|----------------|-----------|---|-----------------|-------------|-----------|
| ciaddr | giaddr | B | dst port | dst ip | dst l2 |
| ≠ 0.0.0.0 | X | X | bootpc(68) | ciaddr | std (arp) |
| 0.0.0.0 | ≠ 0.0.0.0 | X | bootps(67) | ciaddr | std (arp) |
| 0.0.0.0 | 0.0.0.0 | 0 | bootpc(68) | yiaddr | chaddr |
| 0.0.0.0 | 0.0.0.0 | 1 | bootpc(68) | ff.ff.ff.ff | broadcast |

BOOTREQ: campo options

- 🔥 Usato anche dopo la fase di boot
- 🔥 I primi 4 byte = 63 82 53 63 (magic cookie)
- 🔥 Serie libera di campi tagged (tag,len,value):
 - 🔥 Di lunghezza fissa
 - 🔥 Di lunghezza variabile

Fixed Len Subfield

| tag/len | nome | significato |
|---------|----------------------|---|
| 0/0 | pad | Usato per allineare i dati |
| 1/4 | subnet mask | netmask |
| 2/4 | time offset from UTC | differenza fra l'ora locale e l'ora UTC |
| 255/0 | end | fine campo opzioni |

Var Len Subfield

| tag | nome | significato |
|---------|----------------|-------------------------------|
| 3 | gateways | IP gateways |
| 4 | time servers | IP time servers |
| 5 | name serves | IP of Name Servers |
| 15 | domain name | domain name |
| 19-127 | reserved | (riservati: gestiti dai IANA) |
| 128-254 | specific local | Ad uso locale |

Ancor non basta

🌻 Al crescere della complessità sempre più informazioni sono necessarie per inizializzare le macchine

🌻 IP come risorsa: mancano meccanismi per restituirli (dopo la acquisizione)

Nascita di DHCP (Dynamic Host Configuration Protocol) come protocollo trasportato dentro i pacchetti bootp (nel campo option)

DHCP highlight

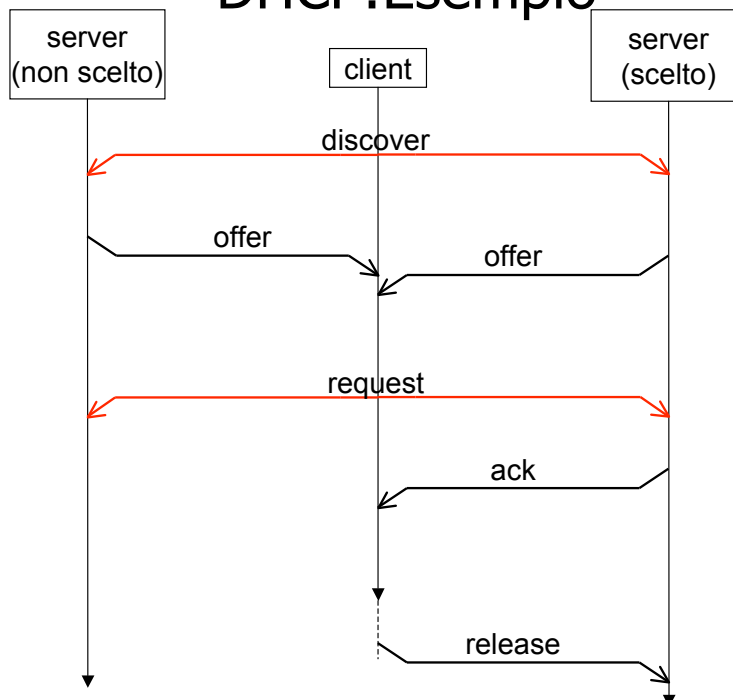
- 🌻 Compatibile con i client ed i relay agent bootp
- 🌻 IP dati a client per durate di tempo limitate
- 🌻 Possibilità per i client di chiedere proroghe della scadenza
- 🌻 Possibilità per i client di disdire l'uso dello IP
- 🌻 Stesso formato di pacchetto di bootp (il campo option ora di lunghezza variabile)
- 🌻 I messaggi da client ⇒ server imbustati dentro bootreq
- 🌻 I messaggi da server ⇒ client imbustati dentro bootreply

DHCP: Messaggi

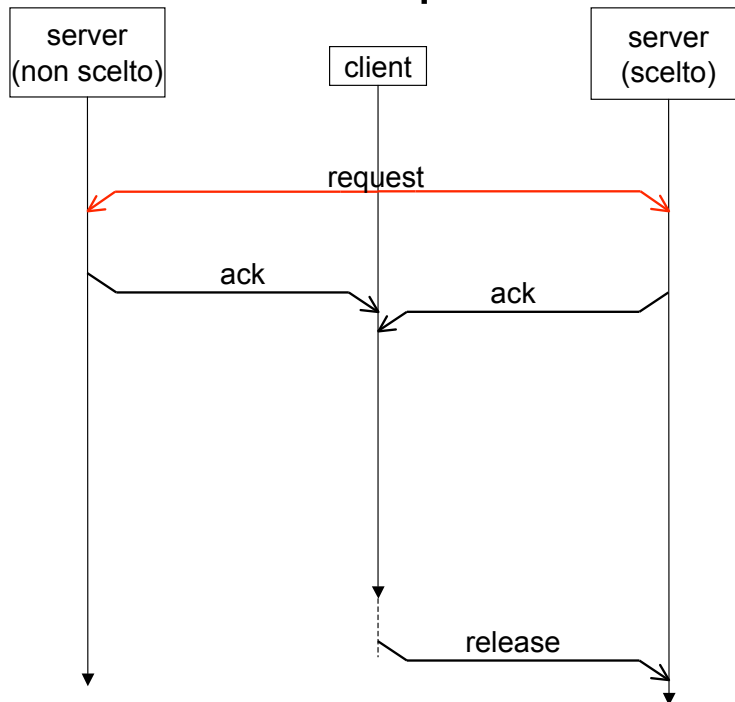
DHCP

| | |
|----------|--|
| discover | Client bcast to locate server |
| offer | Server offer parameter to client |
| request | Client request parameter offer by 1 server |
| ack | Server to Client: commit |
| nack | Server to Client: not commit |
| decline | Client to Server: ip already in use |
| release | Client to Server: release ip |
| inform | Client to Server: ask for other things |

DHCP: Esempio



DHCP: Esempio di riuso



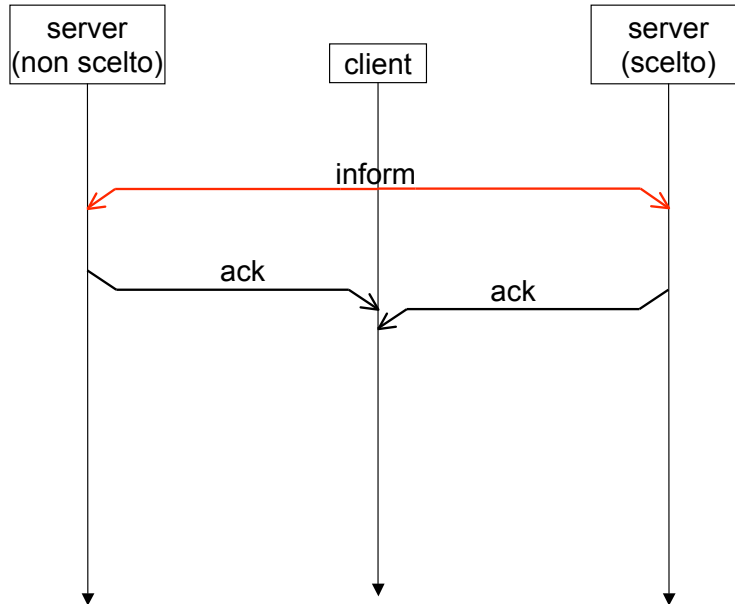
Sul riuso

- 🌸 Il client informa i server che intende usare un ip (specificato con una option)
- 🌸 I server che hanno conoscenza dei parametri usati dal client assentiscono SENZA alcuna verifica che IP sia libero (es ping): il client potrebbe già usarlo
- 🌸 I server possono negare il riuso (nack): es client ha cambiato sottorete
- 🌸 Se il client scopre IP occupato: decline

Altre informazioni

🌻 Se il client desidera altre informazioni: inform

🌻 Il server risponde con un ack contenente le informazioni



Altre opzioni

| tag | nome | significato |
|-----|-------------|---------------------------------|
| 12 | host name | client host name |
| 5 | name serves | IP of Name Servers |
| 50 | req ip | in discover a client ask for IP |
| 51 | lease time | use by server/client (in sec) |
| 53 | DCHP msg | type of message |
| 54 | server id | |
| 57 | max msg len | max len client can accept |
| 61 | client id | |

| tag | nome | significato |
|-----|--------------|--------------------|
| 9 | lpr server | printer servers |
| 12 | host name | client host name |
| 5 | name servers | IP of Name Servers |
| 60 | vendor class | Client hw type |
| 69 | smtp server | email servers |
| 70 | pop server | pop servers |

Problemi aperti

- 🌸 Client e server devono avere orologi che NON scorrono l'uno rispetto all'altro.
- 🌸 Il protocollo è del tutto insicuro!
- 🌸 Possibilità di deny of service
- 🌸 Difficoltà di load balancing
- 🌸 inconsistenza fra i servers

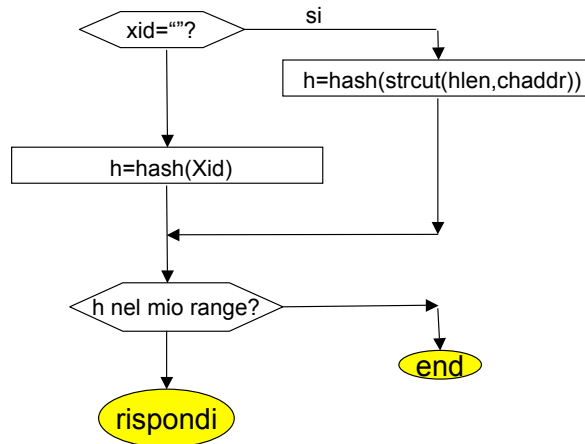
sicurezza

- 🌻 Cifrare tutto il traffico (es con RSA) computazionalmente troppo costo e incompatibile con attuale protocollo
- 🌻 Aggiunta di una opzione (tag=90) che contiene il necessario per l'autenticazione reciproca client/server e dei singoli messaggi

Load Balancing

- 🌻 Aggiungere un protocollo nuovo fra i server ?
- 🌻 Modificare il comportamento dei server senza variare i protocolli?

- 🌻 I server rispondono sulla base di un hash del Xid
- 🌻 Lo hash mappa i valori in un intervallo fra 0 e 255



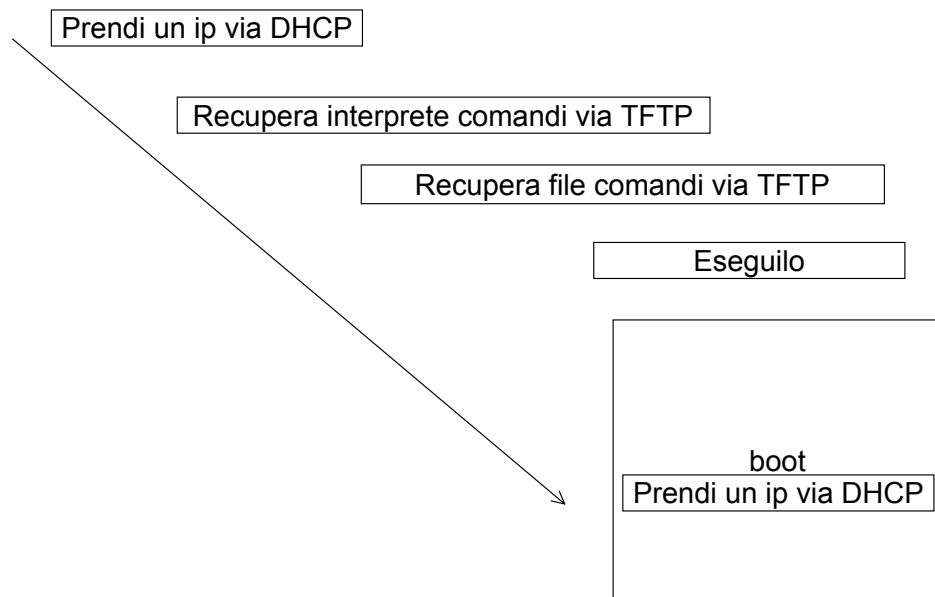
Fail Over

- 🌻 Se un server muore chi risponderà in sua vece ?
- 🌻 Modificare il comportamento dei server senza variare i protocolli/aggiungere protocolli di coordinamento fra i server?
- 🌻 Uso del campo secs per derogare all'uso della hash.
- 🌻 Cosa accade se il client bara sui tempi? e se un server morto torna in vita?
- 🌻 Necessità di un protocollo esplicito

| Regime | Main Server | Backup |
|---------|---|--------------------------------------|
| normal | Normale (da al Backup 1 pool per le emergenze e aggiorna il Backup su cio che dai ai client) | solo renew |
| no comm | Normale (ma non rialloca ip) | Risponde usando il pool di emergenza |

PXE

- 🌻 DHCP risolve non tutti i problemi:
- 🌻 Può esser necessario fare domande all'utente (es quale sistema operativo, farlo autenticare, ecc):
- 🌻 Preboot eXcution Enviroment
- 🌻 Boot con un programma (bpbata) in grado di interagire con l'utente e poi fare il boot vero e proprio.
- 🌻 Uso di dhcp/tftp per eseguire il tutto.
- 🌻 Es le macchina del Lab!



RFC

- 🌻 951
- 🌻 1497
- 🌻 1542
- 🌻 2131
- 🌻 2132
- 🌻 2322
- 🌻 3074
- 🌻 3118