Settima Esercitazione

Sommario

- **Configurazione dell'interfaccia di rete**
  - **comando** `ifconfig`
    - ⇨ **individuazione interfacce**
      - **attivazione** (`up`)
      - **disattivazione** (`down`)
    - ⇨ **impostazioni:**
      - **indirizzo IP**
      - **maschera di rete** (`netmask`)
      - **indirizzo di broadcast** (`broadcast`)
      - **MTU** (`mtu`)
  - **file di configurazione per l'interfaccia di rete**
    - ⇨ **file** `/etc/rc.conf` **e** `/etc/defaults/rc.conf`
    - ⇨ **configurazione con DHCP** (`dhclient`)

Sommario

- **Nomi di host e risoluzione degli indirizzi**
  - **visualizzazione e impostazione nome host**
    - ⇨ `hostname`
    - ⇨ **database locale nomi host**
    - ⇨ **file** `/etc/hosts` **(formato)**
  - **modalita' di risoluzione dei nomi**
    - ⇨ **file** `/etc/host.conf` **(formato)**
  - **risoluzione indirizzi tramite DNS**
    - ⇨ **file** `/etc/nsswitch.conf` **(formato)**
    - ⇨ **comandi** `host` **e** `nslookup`
  - `tcpdump`

Sommario

- Connessione a Internet tramite gateway
  - tabella di routing dell'host
    - ⇨ `route`
      - opzioni
      - output
    - ⇨ impostazione manuale del gateway
  - individuazione problemi (troubleshooting)
    - ⇨ `ping`
      - opzioni e output
    - ⇨ `traceroute`
      - opzioni e output
    - ⇨ `arp`
      - opzioni e output
    - ⇨ `netstat`
      - opzioni e output

Introduzione

- La connessione in una rete basata su IP necessita inizialmente dell'assegnazione di indirizzi IP e quindi di un instradamento per determinare quale strada devono prendere i pacchetti per raggiungere la destinazione.
- Generalmente valgono queste regole:
  - ogni interfaccia di rete ha un proprio indirizzo IP
  - un'interfaccia di rete di un elaboratore può comunicare con un'interfaccia di un altro elaboratore solo se queste sono fisicamente connesse alla stessa rete
  - un'interfaccia di rete di un elaboratore può comunicare con un'interfaccia di un altro elaboratore solo se gli indirizzi di queste interfacce appartengono alla stessa rete.

Configurazione dell'interfaccia di rete

Configurazione di un'interfaccia

- La configurazione di un'interfaccia implica essenzialmente l'attribuzione di un indirizzo IP.
- Quando si assegna un indirizzo a un'interfaccia, occorre anche stabilire la rete a cui questo appartiene, attraverso la maschera di rete
- Il risultato di

  indirizzo_di_interfaccia AND *maschera_di_rete*

  genera l'indirizzo della rete.

ifconfig

- `ifconfig` (***Interface configuration***)
  - **serve per la gestione delle interfacce di rete (network interfaces)**

```
ifconfig <interfaccia> <indirizzo_ip>
  netmask <subnet_mask> broadcast
  <indirizzo_broadcast> up
```

`ifconfig`: opzioni

- `-a`
  - **Mostra tutte le interfacce di rete con le relative caratteristiche, anche se sono impostate down (non attive)**
- `-v`
  - **Modalità verbose**
- `up`
  - **Attiva un'interfaccia di rete**
- `down`
  - **Disattiva un'interfaccia di rete**
- `netmask` *`address`*
  - **Imposta l'indirizzo della netmask**
- `broadcast` *`address`*
  - **Imposta l'indirizzo di broadcast**

ifconfig: output (1 di 5)


```
eth0 Link encap:Ethernet HWaddr 00:05:04:22:3A:50
  inet addr:192.168.0.1 Bcast:192.168.0.255
  Mask:255.255.255.0
  UP BROADCAST NOTRAILERS RUNNING MULTICAST MTU:1500
  Metric:1
  RX packets:260589 errors:0 dropped:0 overruns:0
  frame:0
  TX packets:42355 errors:0 dropped:0 overruns:0
  carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:33172704 (31.6 MiB) TX bytes:2709641 (2.5
  MiB)
  Interrupt:9 Base address:0xfc00
lo Link encap:Local Loopback
  inet addr:127.0.0.1 Mask:255.0.0.0
  UP LOOPBACK RUNNING MTU:16436 Metric:1
  RX packets:43 errors:0 dropped:0 overruns:0 frame:0
  TX packets:43 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:0
  RX bytes:3176 (3.1 KiB) TX bytes:3176 (3.1 KiB)
```

`ifconfig`: output (2 di 5)

- `Link encap:Ethernet`
  - **Indica che l'interfaccia è di tipo ethernet.**
- `Link encap:Local Loopback`
  - **Indica l'interfaccia virtuale di LoopBack.**
- `HWaddr 00:05:04:22:3A:50`
  - **Rappresenta l'indirizzo MAC, cioè l'identificativo unico associato all'interfaccia.**
- `inet addr: 192.168.0.1`
  - **È l'indirizzo IP associato all'interfaccia.**
  - **Nel caso dell'esempio per** `eth0` **è** `192.168.0.1`**, mentre per l'interfaccia di** `loopback` **è** `127.0.0.1`**.**

`ifconfig`: output (3 di 5)

- `Bcast:`
  - **È l'indirizzo di Broadcast dell'interfaccia**
  - **Nel caso di `eth0` è `192.168.0.255`**
- `Mask:`
  - **È la maschera di rete associata all'interfaccia**
  - **Nel caso di `eth0` è `255.255.255.0`**
- `UP`
  - **indica che per l'interfaccia di rete è caricato il relativo modulo/driver e che è attiva.**

`ifconfig`: output (4 di 5)

- `BROADCAST, NOTRAILERS, RUNNIG, MULTICAST`
  - Supporta il broadcast
  - L'encapsulation trailer è disabilitato
  - È pronta per accettare dati
  - Supporta il multicast.
- `MTU:`
  - Acronimo di Max Transmission Unit.
  - Indica la grandezza massima di ogni pacchetto di rete.
- `Metric:`
  - `>= 0`
  - utile solo per i pacchetti in routing
  - più basso è il numero maggiore è il peso.

`ifconfig`: output (5 di 5)

- `RX Packets, TX Packets:`
  - **numero totale di pacchetti**
    - ⇨ **Numero di errori**
    - ⇨ **Numero pacchetti scartati**
    - ⇨ **Numero di collisioni**
    - ⇨ **Lunghezza di trasmissione della coda della periferica.**
- `RX Bytes, TX Bytes:`
  - **Numero totale di byte ricevuti ed inviati che sono passati dall'interfaccia**
- `Interrupt, Base Address`
  - **indirizzi fisici associati dal sistema**

Loopback

- Un elaboratore connesso o meno a una rete fisica vera e propria, **deve** avere una connessione virtuale a una rete immaginaria interna allo stesso elaboratore.
- A questa rete virtuale inesistente si accede per mezzo di un'interfaccia immaginaria denominata `lo`
- L'indirizzo utilizzato è sempre lo stesso, `127.0.0.1`, ma ugualmente deve essere indicato esplicitamente.
- Si definisce il nodo di rete locale con:

```
ifconfig lo 127.0.0.1
```

/etc/rc.conf

- Controlla cosa viene eseguito in automatico all'avvio del sistema
- le impostazioni di default sono definite nel file `/etc/defaults/rc.conf`
- `/etc/rc.conf` contiene solamente le opzioni modificate dall'utente rispetto ai default.

/etc/rc.conf

- Al suo interno è possibile impostare alcuni dei settaggi più comuni
  - timezone
  - keymap
  - moduli del kernel
  - demoni da caricare all'avvio
  - etc.
- Il sistema sara' cosi' pulito e facilmente configurabile da un unico file principale di configurazione.

DHCP

- DHCP (*Protocollo di Configurazione Host Dinamico*) descrive i passi attraverso i quali un sistema si può connettere ad una rete ed ottenere l'informazione necessaria per comunicare attraverso quella rete.
- Quando eseguito sulla macchina client questa inizia a fare broadcasting di richieste per informazioni di configurazione.
- La richiesta viene fatta in broadcasting perché il sistema non conosce a priori l'indirizzo del server DHCP

DHCP

- Di default queste richieste sono sulla `porta UDP 68`.
- Il server risponde sulla `porta UDP 67`, dando al client un indirizzo IP ed altre informazioni rilevanti di rete come la netmask, il router ed il DNS server.
- Tutte queste informazioni sono valide solo per un certo periodo di tempo (configurato dall'amministratore del server DHCP).
  - In questo modo, gli indirizzi IP bloccati da client che non sono più connessi alla rete possono essere riutilizzati automaticamente.

dhclient: opzioni

- **Sintassi**

```
dhclient [-bdqu] [-c file] [-l file] interface
```

- **Deve essere specificato il nome dell'interfaccia da configurare**
- `-b`
  - **Forza il processo a lavorare in backgroud**
- `-c file`
  - **Specifica il file da cui prendere la configurazione**
- `-d`
  - **Forza il processo a lavorare in foreground (default)**

```
dhclient rl0
```

Nomi di host e risoluzione degli indirizzi

Indirizzi e nomi

- La gestione diretta degli indirizzi IP in forma numerica può essere utile in fase di progetto di una rete, ma a livello di utente è una pretesa praticamente inaccettabile.
- Per questo, agli indirizzi IP numerici si affiancano quasi sempre dei nomi che teoricamente potrebbero anche essere puramente fantastici e senza alcuna logica.
- Ogni volta che si fa riferimento a un nome, il sistema è (o dovrebbe essere) in grado di convertirlo nel numero IP corrispondente
- Ci sono due metodi per trasformare un nome in un indirizzo IP e viceversa:
  - un elenco contenuto nel file `/etc/hosts`
  - server DNS

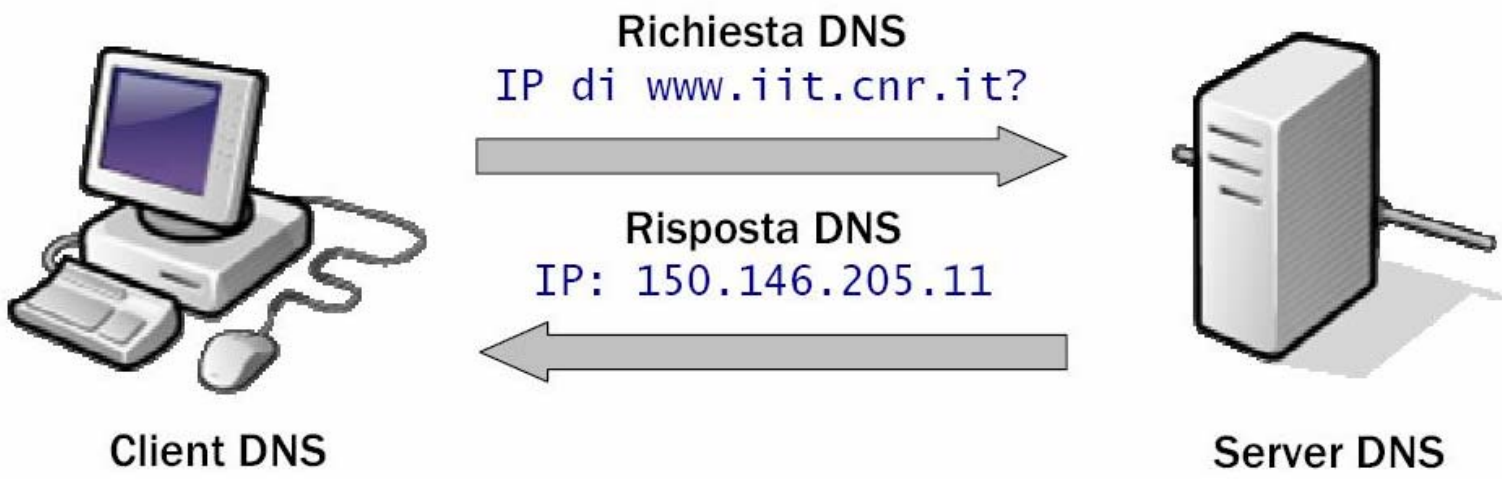
hostname

- Comando che permette di visualizzare o impostare l'hostname ovvero il nome del computer su cui esso viene lanciato.
- Se il comando viene lanciato senza passare nessun argomento verrà solamente visualizzato il nome dell'host.
- Sintassi:

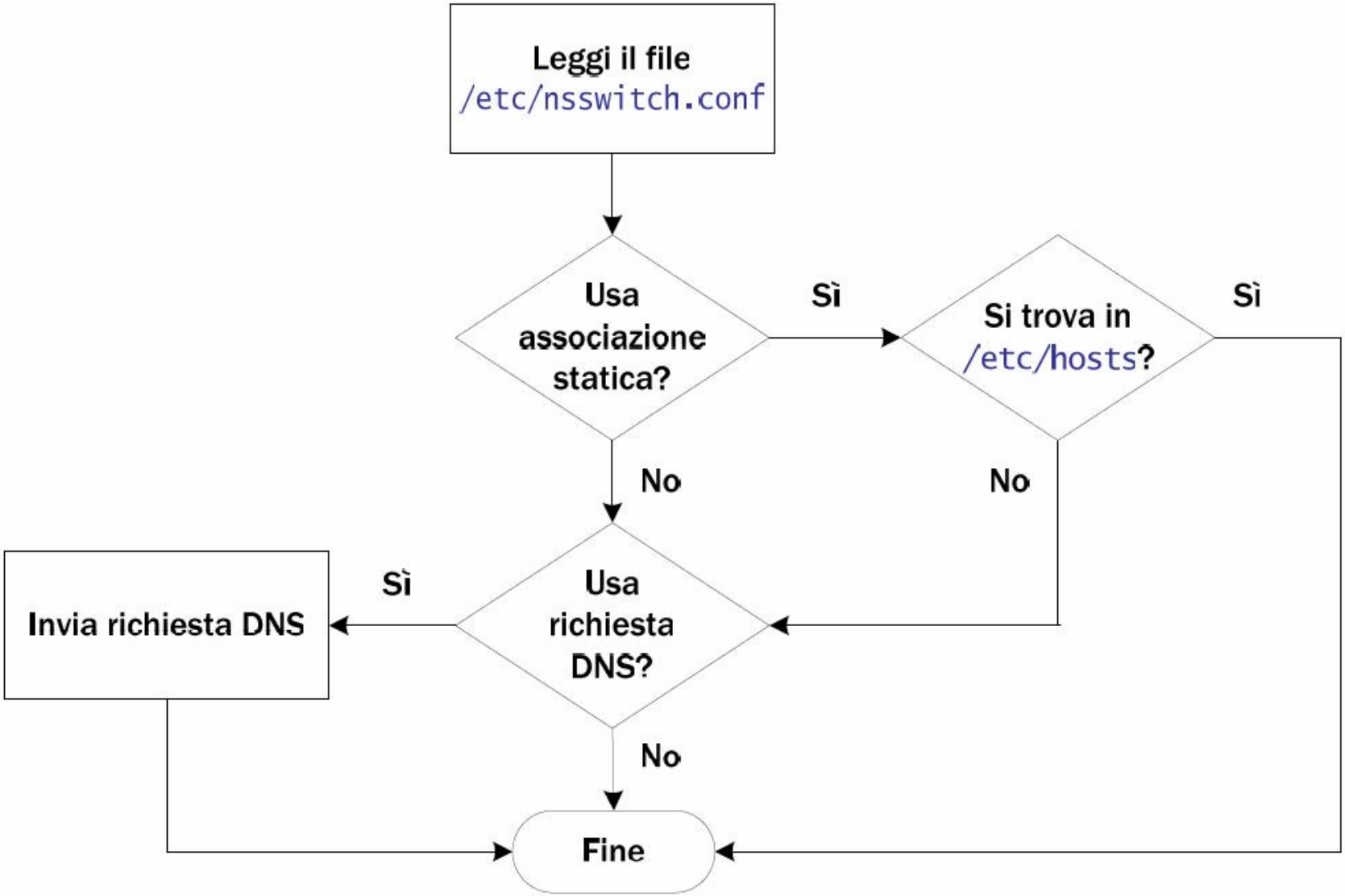
```
hostname [opzioni] [argomenti]
```

Servizio di risoluzione dei nomi

- **Architettura di riferimento**
  - **client/server**
    - ⇨ **sistema gerarchico**
- **Interazione richiesta/risposta**



Risoluzione di una richiesta DNS



/etc/nsswitch.conf

- **Il file del `Network Services Switch` determina l'ordine delle ricerche effettuate quando viene richiesta una certa informazione**

/etc/nsswitch.conf

```
hosts: files nis dns
```

- specifica che le funzioni di ricerca degli host dovrebbero prima guardare
  - nel file locale `/etc/hosts`,
  - di seguito fare una ricerca `NIS`
    - ⇨ Network Information Service: servizio che fornisce informazioni che devono essere conosciute in ogni parte della rete, a tutte le macchine che vi prendono parte.
  - infine utilizzare il servizio dei nomi di dominio `/etc/resolv.conf`
- A quel punto, se nessuna corrispondenza è stata trovata, viene riportato un errore.

/etc/hosts.conf

- Viene usato per determinare quali servizi usare per risolvere i nomi di dominio.
- Ogni riga rappresenta un'opzione di funzionamento
- Solitamente vengono specificate solo due direttive:

```
order hosts,bind
multi on
```

- `order`
  - indica l'ordine dei servizi.
  - In questo caso si utilizza prima il file `/etc/hosts` e quindi si interpella il servizio di risoluzione dei nomi.
- `multi`
  - abilita la possibilità di trovare all'interno del file `/etc/hosts` l'indicazione di più indirizzi IP per lo stesso nome.
  - Un evento del genere può verificarsi quando uno stesso elaboratore ha due o più connessioni per la rete e per ognuna di queste ha un indirizzo IP diverso.

/etc/hosts

- Viene usato per convertire i nomi degli elaboratori in numeri IP e viceversa.
- `/etc/hosts` accetta il formato:

```
[Indirizzo Internet ] [nome host ufficiale] [alias1] [alias2] ...
```

- Esempio:

```
10.0.0.1 myRealHostname.example.com myRealHostname foobar1 foobar2
```

/etc/hosts (1 di 2)


```
#
# Necessario per il "loopback" IPv4.
#
127.0.0.1       localhost.localdomain   localhost
#
# Indirizzi IPv4.
#
192.168.1.1     dinkel.brot.dg              dinkel
192.168.1.2     roggen.brot.dg              roggen
#
192.168.2.1     weizen.mehl.dg              weizen
```

/etc/hosts (2 di 2)


```
#
# Necessario per il loopback IPv6.
#
::1                                     ip6-localhost      ip6-loopback
#
# Necessari per il multicast IPv6.
#
fe00::0                                 ip6-localnet
ff00::0                                 ip6-mcastprefix
ff02::1                                 ip6-allnodes
ff02::2                                 ip6-allrouters
ff02::3                                 ip6-allhosts
#
# Indirizzi IPv6.
#
fec0::1:2a0:24ff:fe77:4997              dinkel.brot.dg         dinkel
fec0::1:280:5fff:fea6:6d3d              roggen.brot.dg         roggen
#
fec0::2:280:adff:fec8:a981              weizen.mehl.dg         weizen
```

nslookup

- `Name Server Lookup` è uno strumento presente in tutti i sistemi operativi che utilizzano il protocollo TCP/IP
- Consente di effettuare delle query ad un server DNS per la risoluzione di indirizzi IP o hostname
- Si usa per poter ottenere da un dominio il relativo indirizzo IP o nome host e viceversa.
- Si può utilizzare in due modi: interattivo e non interattivo.
- SYNOPSIS

```
nslookup [opzioni] [host_da_indagare | -DNSserver]
```

nslookup interattivo

- Questo modo permette di effettuare più query e visualizza i singoli risultati.
- Viene abilitato in modo automatico quando:
  - il comando non è seguito da argomenti
  - se il primo argomento è un trattino (-) seguito dal secondo argomento che corrisponde all'host name o all'ip del name server.
- `exit`: permette di uscire dalla shell interattiva

```
$ nslookup
> www.ing.unipi.it
Server:         131.114.21.15
Address:        131.114.21.15#53
www.ing.unipi.it  canonical name =www.web.ing.unipi.it.
Name:   www.web.ing.unipi.it
Address: 131.114.28.27
>
```

nslookup non interattivo

- **Permette di effettuare una sola query e ovviamente visualizza il risultato della singola query.**
- **Abilitato se si specifica l'*host-to-find*.**

```
$ nslookup www.ing.unipi.it
Server:         131.114.21.15
Address:        131.114.21.15#53
www.ing.unipi.it        canonical name =www.web.ing.unipi.it.
Name:   www.web.ing.unipi.it
Address: 131.114.28.27
$
```

host

- `host` è **considerato insieme a** `dig` **il sostituto ufficiale di** `nslookup`.
- **Viene utilizzato per risolvere i nomi in indirizzi numerici e viceversa.**
- **Se lanciato senza parametri ci presenta una lista delle opzioni possibili e una breve spiegazione della loro funzione.**

```
host [opzioni] host [server]
```

`host`: opzioni

- `-c` *`classe`*
  - **Ricerca il record di risorsa specificato da *classe*.**
    - ⇨ **IN= INTERNET (default)**
- `-f` *`nome_del_file`*
  - **Invia l'output ad un file specificato oltre che allo standard output.**
- `-l` *`zona`*
  - **Elenca tutte le macchine appatenenti alla zona specificata.**
- `-T`
  - **Utilizza TCP anzichè UDP.**

`host`: opzioni

- `-t` *`tipo`*
  - Permette di specificare il tipo di query per i record di risorse.
  - I tipi possono essere `A`, `NS`, `PTR`, `ANY`, o `*` (tutti).
- `-v` e `-vv`
  - Rispettivamente *verbose* e *very verbose*.
  - Il primo include tutti i campi del record delle risorse, compresi Time-to-Live e classe, oltre ai campi "additional information" e "authoritative nameservers".
  - Il secondo aggiunge anche informazioni riguardo l'host richiesto (CPU, sistema operativo) ovviamente se presenti nel database DNS.

tcpdump

- Strumento per l'analisi del traffico che avviene nella rete fisica a cui si è collegati.
- È uno strumento di sniffing particolarmente flessibile
- Si setta in modalità promiscua cioè non vede solo i pacchetti diretti a lui ma tutto il traffico
- NON visualizza il contenuto dei pacchetti ma solo le loro intestazioni (protocollo, IP sorgente, destinazione, porte ecc.) per cui si presta bene alla diagnostica di problemi di networking
- Sintassi

```
tcpdump [opzioni] [espressioni]
```

`tcpdump`: opzioni

- `-a`
  - Permette di convertire indirizzi broadcast e ip in nomi
- `-c`
  - Esce dopo aver ricevuto un tot di pacchetti (es. `c100`)
- `-e`
  - Stampa il MAC address in ogni pacchetto catturato
- `-i`
  - Definisce l'interfaccia di rete. (es. `-i eth0` o `-i lo`)
- `-p`
  - L'interfaccia locale non viene settata in modalità promiscua
- `-q`
  - Visualizza il minor numero di informazioni possibili

`tcpdump`: opzioni

- `-r`
  - Utilizza il file specificato come input per i dati da filtrare
- `-s`
  - Indica la quantità in byte di un pacchetto catturato
- `-t`
  - Non stampa il contrassegno temporale per ogni pacchetto catturato
- `-tt`
  - Stampa il contrassegno temporale (timestamp) non formattato su ogni linea
- `-v`
  - Verbose
- `-w` *`file`*
  - Scrive su file il risultato dello sniffing
- `-X`
  - Stampa i pacchetti in formato HEX e ASCII

tcpdump: espressioni

```
tcpdump [opzioni] [espressioni]
```

- Composte da primitive che possono essere raggruppate per mezzo delle parentesi tonde
- Connesse attraverso operatori booleani
- Solo i pacchetti che soddisfano la condizione espressa vengono presi in considerazione
- Se l'espressione manca, vengono catturati tutti i pacchetti.

Operatori booleani

- `!` o `not`
  - negazione logica
- `&&` o `and`
  - and logico
- `||` o `or`
  - OR logico
- `tcpdump` accetta anche operatori come:
  - `<`
    - ⇨ es. `< 40`
  - `>`
    - ⇨ es. `> 50`

tcpdump: entità

- `type`
  - **Dice a cosa l'ID (numerico o nome) si riferisce.**
    - ⇨ `host`
    - ⇨ `net`
    - ⇨ `port`
- `dir`
  - **entità che specifica una particolare direzione di trasferimento per/da un ID.**
    - ⇨ `src`
    - ⇨ `dst`
    - ⇨ **Default** `src or dst`
    - ⇨ **es.** '`src` *foo*', '`src or dst port ftp-data`'
- `proto`
  - **entità che restringe la cattura ad un particolare protocollo.**
    - ⇨ `ether, fddi, ip, arp, rarp, decnet, lat, moprc, mopdl, tcp e udp`
    - ⇨ **es.** '`ether src` *foo*', '`tcp port 21`')
  - **Se il protocollo non è selezionato vengono considerati tutti i protocolli**

tcpdump: esempio

tcpdump –vvv –i lo

```
15:46:49.086648 localhost.1029 > localhost.ftp: P
  3015740888:3015740901(13) ack 3004264398 win 31072
  <nop,nop,timestamp 578305 549639> (DF) [tos 0x10]
15:46:49.086648 localhost.1029 > localhost.ftp: P
  0:13(13) ack 1 win 31072 <nop,nop,timestamp 578305
  549639> (DF) [tos 0x10]
15:46:49.086739 localhost.ftp > localhost.1029: . ack
  13 win 31059 <nop,nop,timestamp 578305 578305> (DF)
  [tos 0x10]
15:46:49.086739 localhost.ftp > localhost.1029: . ack
  13 win 31059 <nop,nop,timestamp 578305 578305> (DF)
  [tos 0x10]
```

tcpdump: esempio

```
15:46:49.086648 localhost.1029 > localhost.ftp: P
  3015740888:3015740901(13) ack 3004264398 win 31072
  <nop,nop,timestamp 578305 549639> (DF) [tos 0x10]
```

- `15:46:49.086648`
  - **tempo in cui il pacchetto è stato catturato.**
- `localhost.1029 > localhost.ftp`
  - **il pacchetto è stato spedito da un'applicazione su `localhost` associata alla porta `1029` ad un'applicazione ancora su `localhost` associata alla porta riservata ai servizi `ftp`.**
  - **sniffing dal loopback**
- **Ci sono altri dettagli di TCP come i numeri di sequenza della connessione (`3015740888`) o il valore di particolari bit che non ci interessano.**
- **Con queste opzioni `tcpdump` *non* prende in considerazione la parte dati.**

tcpdump: esempio

```
tcpdump -i lo -s 200 -x -q
```

- `-s 200` **specifica di catturare 200 byte di ogni pacchetto**
- `-x` **specifica di stampare (in esadecimale) l'inizio di ogni pacchetto**
- `-q` **fornisce una stampa meno invadente dell'output (mancano i numeri di sequenza e altri dettagli)**

```
15:59:35.695801 localhost.1031 > localhost.ftp: tcp 13
  (DF) [tos 0x10]
       4510 0041 014e 4000 4006 3b57 7f00 0001
       7f00 0001 0407 0015 f47b f62d f429 54a3
       8018 7960 c764 0000 0101 080a 0009 fe76
       0009 f7c5 5553 4552 2067 6975 6d6f 6e0d
       0a
15:59:35.695801 localhost.1031 > localhost.ftp: tcp 13
  (DF) [tos 0x10]
       4510 0041 014e 4000 4006 3b57 7f00 0001
       7f00 0001 0407 0015 f47b f62d f429 54a3
       8018 7960 c764 0000 0101 080a 0009 fe76
       0009 f7c5 5553 4552 2067 6975 6d6f 6e0d
       0a
```

`tcpdump`: esempi

```
tcpdump -i eth0 -w file tcp port 23
```

- **cattura solo i pacchetti** `tcp`
- **interfaccia di rete** `eth0`
- **destinati alla** `porta 23(telnet)`
- **salva tutto l'output su un file**

```
tcpdump -r file
```

- **visualizza l'output in maniera standard**

tcpdump: esempi

```
tcpdump port 80
```

- **Visualizza solo i pacchetti che hanno come sorgente o destinazione la** `porta 80`

```
tcpdump host 192.168.0.150
```

- **Visualizza solo i pacchetti che hanno come IP sorgente o destinazione** `192.168.0.150`**.**

```
tcpdump host 10.0.0.150 and not port 22
```

- **Visualizza solo i pacchetti relativi all'host** `10.0.0.150` **che non usino la porta** `ssh` **(**`and not port 22`**).**

```
tcpdump net 10.0.0.0/24 and port 22
```

- **Visualizza tutti i pacchetti per la rete** `10.0.0.0/24` **relativi al protocollo** `ssh` **(and** `port 22`**)**

`tcpdump`: considerazioni

- `tcpdump` è molto potente e versatile
- Interpretare direttamente il suo output non è pratico.
- Se si è interessati a visualizzare e raggruppare i dati sniffati si possono usare dei tool che facilitano queste operazioni.

Connessione a Internet tramite gateway

route

```
route -n get www.google.it
```

```
route to: 209.85.135.99
destination: default
mask: default
gateway: 131.114.31.254
interface: rl0
flags: <UP,GATEWAY,DONE,STATIC>
recvpipe sendpipe ssthresh rtt,msec  rttvar  hopcount    mtu   expire
       0        0        0        0       0          0  1500        0
```

route

- **Sintassi**

```
route [-dnqtv] command [[modifiers] args]
```

- command
  - add
    - ⇨ Aggiunge una route.
  - flush
    - ⇨ Rimuove tutte le route
  - delete
    - ⇨ Elimina una specifica route
  - Change
    - ⇨ Cambia le caratteristiche di una route, ad esempio il gatway
  - get Lookup and display the route for a destination.
    - ⇨ Cerca e mostra una route per una destinazione
  - monitor
    - ⇨ Riporta continuamente i cambiamenti alla tabella di routing

route: campi

- destination
  - indica l'host o la rete che si vuole raggiungere
- mask
  - indica la maschera di rete
- gateway
  - indica, dove presente, il gateway, ovvero il computer della rete locale tramite cui connettersi ad un'altra rete
- interface
  - l'interfaccia di rete tramite cui connettersi

route: opzioni

- -n
  - **Per indicare il valore numerico degli host nell'output**
- -v
  - **Verbose**
- -q
  - **Mostra un output piu' scarno per** add, change, delete **e** flush

route: GNU/LINUX

- **Permette di vedere e modificare la tabella di routing.**

`route -n`

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.114.228.145  0.0.0.0         255.255.255.255 UH    0      0        0 tun0
10.114.228.128  10.114.228.145  255.255.255.128 UG    0      0        0 tun0
131.114.9.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.114.0.0      10.114.228.145  255.255.0.0     UG    0      0        0 tun0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth0
0.0.0.0         131.114.9.29    0.0.0.0         UG    100    0        0 eth0
```

- **Il computer dove è stato lanciato il comando si connette tramite l'interfaccia `eth0` direttamente alle reti `131.114.9.0` e `169.254.0.0` (il campo gateway è indicato con `0.0.0.0`)**
- **per tutte le altre destinazioni (`0.0.0.0`) si connette mediante il gateway `131.114.9.29`**

route: esempi

```
route add -net 192.168.101.0 netmask
  255.255.255.0 gw 192.168.101.102
```

- **Aggiunge la route** `192.168.101` **alla routing table indicando la maschera di rete** `255.255.255.0` **e il gateway** `192.168.101.102`

```
route add default gw 192.168.12.1
```

- **Aggiunge l'host** `192.168.12.1` **come gateway di default**

```
route add 10.41.0.0 mask 255.255.0.0 10.27.0.1
                    if 0x3
```

- **Aggiunge una route alla destinazione** `10.41.0.0` **con la subnet mask** `255.255.0.0`**, l'indirizzo di hop successivo** `10.27.0.1` **e utilizza l'indice di interfaccia** `0x3`

ping

- Invia una successione di pacchetti ad una stazione per verificarne la raggiungibilità
- Ricorre al protocollo ICMP.

ICMP

- *Internet Control Message Protocol* (Protocollo per i Messaggi di Controllo in Internet).
- Il protocollo IP fornisce un meccanismo di trasferimento dei pacchetti dal mittente al destinatario secondo un approccio *best-effort.*
- Questo vuol dire che l'IP non è in grado di garantire la consegna dei pacchetti al destinatario, ma esegue dei tentativi di consegna.
- ICMP segnala solamente errori e malfunzionamenti, ma non esegue alcuna correzione

Esempio

- Supponiamo che dalla **stazione A** si voglia controllare l'integrità della connessione fino alla **stazione B**.
- Si esegue il comando `ping`, passandogli come argomento l'indirizzo della **stazione B**.
- Il programma manda una serie di messaggi ICMP `ECHO_REQUEST` (generalmente uno al secondo) dalla **stazione A** verso la **stazione B**.
- Quando la **stazione B** riceve un pacchetto `ECHO_REQUEST`, risponde con un nuovo datagramma `ECHO_REPLY`, che viene mandato indietro alla **stazione A**.
- Il programma `ping` userà le informazioni così collezionate (esistenza dei pacchetti di ritorno, tempo intercorso per ogni pacchetto, etc.) per calcolare dei valori statistici sulla bontà della connessione e presentarli all'utente.

ping: output

```
ping www.google.it
PING www.l.google.com (64.233.183.103) 56(84) bytes of data.
64 bytes from nf-in-f103.google.com (64.233.183.103):
   icmp_seq=1 ttl=232 time=91.6 ms
64 bytes from nf-in-f103.google.com (64.233.183.103):
   icmp_seq=2 ttl=232 time=94.6 ms

--- www.l.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time
   4001ms
rtt min/avg/max/mdev = 69.754/87.351/101.647/11.455 ms
```

- **dimensione del pacchetto** ECHO_REPLAY
- **indirizzo IP di DEST**
- **numero di sequenza della risposta**
- **“time-to-live”** (TTL)
- **“round-trip time”** (RTT)
- **risultati statistici**

ping: opzioni

- `-c count`
  - Stop l'invio (e ricezione) di pacchetti count `ECHO_RESPONSE`
- `-i wait`
  - Aspetta wait secondi tra gli invii dei pacchetti.
  - Il default è di aspettare per un secondo tra ciascun pacchetto.
- `-n`
  - Solo output numerico. Non verrà fatto nessun tentativo di cercare nomi simbolici per gli indirizzi dell'host.
- `-q`
  - Output silenzioso. Non è visualizzato nulla tranne le linee di sommario all'avvio e quando termina.
- `-s dimensione pacchetto`
  - Specifica il numero di byte di dati da inviare.
  - Il default è 56, che si traduce in 64 byte di dati ICMP quando combinato con gli 8 byte dei dati di intestazione di ICMP.

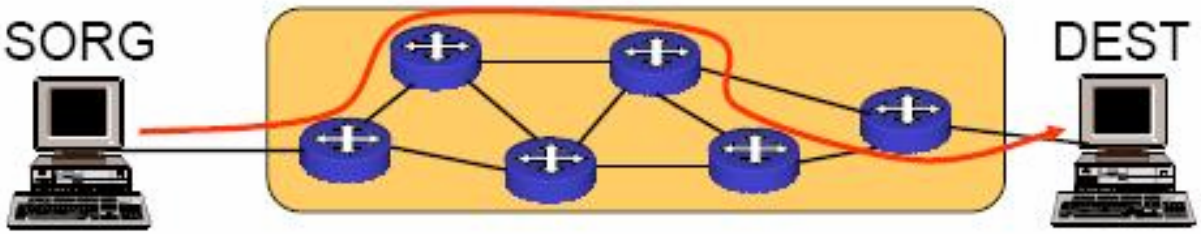
traceroute

- Permette di conoscere il percorso seguito dai pacchetti inviati da una sorgente e diretti verso una destinazione

```
traceroute [<opzioni>] <destinazione> [<lunghezza>]
```

- `traceroute` inizia la trasmissione di pacchetti (utilizzando il protocollo ICMP) con un TTL molto basso.
- Si aspetta di ricevere un messaggio di errore dall'host in cui il TTL raggiunge lo zero.
- Incrementando lentamente il valore del TTL, `traceroute` riesce a conoscere gli indirizzi dei nodi attraversati, purché tutto funzioni come previsto (cioè che i vari nodi generino correttamente i pacchetti ICMP di errore).
- Quando tutto funziona come previsto, `traceroute` genera un elenco di host a partire da primo nodo attraversato, fino all'ultimo che rappresenta la destinazione richiesta.
- Se in alcuni punti non si ottiene risposta, i nodi ipotizzati vengono segnalati con degli asterischi.

traceroute: esempio



- SORG **invia a** DEST **una serie di pacchetti ICMP di tipo** ECHO **con un** TIME-TO-LIVE (TTL) **progressivo da 1 a 30 (per default)**
- **Ciascun nodo intermedio decrementa** TTL
- **Il nodo che decrementa** TTL **a** 0 **invia a** SORG **un pacchetto ICMP di tipo** TIME_EXCEEDED
- SORG **costruisce una lista dei nodi attraversati fino a** DEST

traceroute: output

```
traceroute www.ing.unipi.it
traceroute to www.ing.unipi.it (131.114.28.27), 30 hops max,
   40 byte packets
 1  man-gate.iet.unipi.it (131.114.9.29)  0.706 ms  0.688 ms
   0.929ms
 2  131.114.186.9 (131.114.186.9)  3.929 ms  4.494 ms  13.325
   ms
 3  www.web.ing.unipi.it (131.114.28.27)  3.908 ms !H  3.905
   ms !H  5.469 ms !H
```

- **TTL**
- **nome DNS**
- **indirizzo IP dei nodi intermedi**
- **ROUND-TRIP TIME (RTT)**

ARP

- `Address Resolution Protocol (ARP)` **è un protocollo che fornisce la "mappatura" tra l'indirizzo IP e il suo MAC address**
- **ARP viene utilizzato per ottenere l'indirizzo MAC quando un pacchetto deve essere inviato ad un calcolatore nella stessa sottorete**
- **Se il pacchetto deve essere inviato ad un calcolatore di un'altra sottorete, ARP viene utilizzato per scoprire il MAC address del gateway (non può attraversare router).**
- **Il protocollo ARP tiene traccia delle risposte ottenute in una apposita cache, per evitare di utilizzare ARP prima di inviare ciascun pacchetto.**
- **Le voci della cache ARP vengono cancellate dopo un certo periodo dall'ultima occorrenza, tipicamente dopo 5 minuti.**

Meccanismo ARP

- L'host che vuole conoscere il mac address di un altro host, di cui conosce l'indirizzo IP, invia in broadcast una richiesta ARP_REQUEST contenente l'indirizzo IP dell'host di destinazione ed il proprio indirizzo MAC.
- Tutti i calcolatori della sottorete ricevono la richiesta.
- In ciascuno di essi il protocollo ARP verifica se viene richiesto il proprio indirizzo IP.
- L'host di destinazione che riconoscerà il proprio IP nel pacchetto di ARP_REQUEST, provvederà ad inviare una risposta ARP_REPLAY in unicast all'indirizzo MAC sorgente, contenente il proprio MAC.
- Ogni host può scoprire l'indirizzo fisico degli altri host sulla stessa sottorete.

arp

- **Permette di visualizzare e manipolare le voci di `arp` nella cache del sistema.**

```
arp [opzioni]  -a [hostname]
```

- **Visualizzazione del contenuto di tutta la cache, oppure specificando l'host solo l'`arp` del suddetto host**

```
arp [opzioni]  -d hostname
```

- **Cancellazzione dell'`arp` di uno specifico host**

```
arp  [opzioni] -s hostname hw_addr [opzioni]
```

- **Creazione manuale di uno specifico arp di un host**

- **Trova l'indirizzo MAC della macchina da raggiungere attraverso una tabella di cache**

`arp`: opzioni

- `-v`
  - Abilita il verbose mode
- `-n`
  - Non esegue il DNS lookup degli indirizzi ip
- `-i`
  - Specifica l'interfaccia

arp: esempio

arp –a

```
ns.ing.unipi.it (131.114.28.5) at
  00:0e:2e:02:c7:3a on rl0 [ethernet]
docenti.ing.unipi.it (131.114.28.20) at
  00:04:75:77:5f:ae on rl0 [ethernet]
studenti.ing.unipi.it (131.114.29.9) at
  00:e0:4c:39:21:15 on rl0 [ethernet]
e-ing.ing.unipi.it (131.114.29.11) at
  00:04:96:34:7e:d6 on rl0 [ethernet]
? (131.114.31.0) at 00:48:54:01:1b:40 on rl0
  [ethernet]
juniper.ing.unipi.it (131.114.31.254) at
  00:19:e2:98:04:21 on rl0 [ethernet]
```

netstat

- Permette di vedere lo stato delle connessioni instaurate sul computer locale.
- Eseguito senza dare opzioni visualizza lo stato dei socket attivi
- Sintassi

```
netstat [information-type] [options]
```

`netstat`: information-type

- `-r`
  - Mostra il contenuto delle tabelle di routing
  - Restituisce lo stesso output del comando `route`
- `-g`
  - Visualizza informazioni riguardanti i multicast group membership (ipv4 e ipv6)
- `-i`
  - Visualizza le statistiche di tutte le interfacce o della singola interfaccia specificata
- `-s`
  - Statistiche sui protocolli di rete

`netstat`: opzioni

- `-a`
  - permette di vedere anche lo stato dei socket non attivi
- `-f`
  - Limita ad una famiglia di protocolli
    - ⇨ unix, inet
- `-w N`
  - Ogni quanto vengono fornite informazioni (in secondi)
- `-n`
  - Mostra gli indirizzi numerici (invece di quelli simbolici) degli host e delle porte
  - risparmia i tempi per query DNS.

`netstat`: output

```
Active Internet connections
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0     48  wind2.ing.unipi..ssh   host84-33-dynami.39255 ESTABLISHED
udp4       0      0  wind2.ing.unipi..978   131.114.31.0.nfsd
Active UNIX domain sockets
Address  Type   Recv-Q Send-Q    Inode     Conn   Refs  Nextref Addr
c491a240 stream      0      0        0 c491bbd0   0        0
c491b000 stream      0      0 c46e5550        0   0        0 /var/run/devd.pipe
c491a480 dgram       0      0        0 c491ae10   0 c491acf0
```

- `netstat` **classifica i tipi di socket:**
  - **Active Internet connection** (`tcp` **e** `udp`)
  - **Active UNIX domain sockets**

`netstat`: output

- `Proto`
  - **nome del protocollo**
- `Local Address`
  - **indirizzo IP del computer locale insieme al numero di porta usato**
- `Foreign Address`
  - **indirizzo IP e porta al quale il socket è connesso**
- `State`
  - **stato della connessione**
- `Type`
  - **Tipo del socket:**
    - ⇨ `DGRAM`: **udp**
    - ⇨ `STREAM`: **tcp**

Esericizi

- Identificare i principali parametri delle interfacce di rete:
  - indirizzo ip
  - netmask
  - broadcast
  - maximum transmission unit (MTU)
- Determinare il nome della propria macchina
  - Determinare il dominio di appartenenza della propria macchina
- Chiedere al server DNS avente indirizzo `131.114.28.5` (`131.114.21.15`) l'indirizzo IP della macchina `www.ing.unipi.it`
- Visualizzare la cache ARP e svuotarla
- Effettuare un `ping` ad una macchina vicina utilizzando 3 richieste tra loro separate con intervalli di 2 secondi
- Visualizzare di nuovo la cache ARP
- Individuare il gateway default e fare un `ping` verso di esso utilizzando indirizzi numerici

Soluzione

- `ifconfig`
  - **(la scheda di rete è quella nella cui sezione compare la riga ether)**
- `hostname`
  - `cat /etc/resolv.conf | grep domain`
- `host www.ing.unipi.it 131.114.28.5`
  - `(host www.ing.unipi.it 131.114.21.15)`
- `arp -an` **e** `arp -ad`
- `ping -c 3 -i 2 <ip_vicino>`
- `arp –an`
  - **(nella cache ARP è apparsa una riga relativa al vicino)**
- `route -n get default` **e** `ping <ip_gateway>`