

Linux Networking Tools e Esercizi su TCP

Esercitazione di Laboratorio di Programmazione di Rete A

Daniele Sgandurra

Università di Pisa

12/11/2008

Ifconfig

- Mostra la **configurazione** delle **interfacce di rete**:

```
fujim10 [~] -> ifconfig
eth0  Link encap:Ethernet  HWaddr 00:0D:61:C1:52:84
      inet addr:131.114.11.210  Bcast:131.114.11.255  Mask:255.255.255.0
      inet6 addr: fe80::20d:61ff:fec1:5284/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:161949 errors:0 dropped:0 overruns:0 frame:0
      TX packets:82302 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:25724163 (24.5 MiB)  TX bytes:18102833 (17.2 MiB)
      Interrupt:169 Base address:0xdc00

lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:2652 errors:0 dropped:0 overruns:0 frame:0
      TX packets:2652 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:348028 (339.8 KiB)  TX bytes:348028 (339.8 KiB)
```

Ifconfig

- Per mostrare **tutte** le interfacce di rete (anche quelle inattive), usare l'opzione `-a`:

```
fujim10 [~] -> ifconfig -a
```

- Da **root** è possibile **configurare** staticamente le interfacce di rete:

```
# ifconfig eth0 192.168.10.1 netmask 255.255.255.0
```

Assegna all'interfaccia di rete `eth0` l'IP `192.168.10.1` e maschera `255.255.255.0`.

- Per **attivare/disattivare** le interfacce di rete (da root):

```
# ifup eth0  
# ifdown eth0
```

Ping

- Rileva se un host remoto e' **raggiungibile**:

```
fujim10 [~] -> ping fujih1 -c 5
PING fujih1.cli.di.unipi.it (131.114.11.151) 56(84) bytes of data.
64 bytes from fujih1.cli.di.unipi.it (131.114.11.151): icmp_seq=1 ttl=64 time=0.156 ms
64 bytes from fujih1.cli.di.unipi.it (131.114.11.151): icmp_seq=2 ttl=64 time=0.159 ms
64 bytes from fujih1.cli.di.unipi.it (131.114.11.151): icmp_seq=3 ttl=64 time=0.148 ms
64 bytes from fujih1.cli.di.unipi.it (131.114.11.151): icmp_seq=4 ttl=64 time=0.165 ms
64 bytes from fujih1.cli.di.unipi.it (131.114.11.151): icmp_seq=5 ttl=64 time=0.164 ms

-- fujih1.cli.di.unipi.it ping statistics --
5 packets transmitted, 5 received, 0% packet loss, time 4064ms
rtt min/avg/max/mdev = 0.148/0.158/0.165/0.012 ms
```

Ping: opzioni

Opzioni più utilizzate:

```
ping [ -LRUbdfnqrvVaAB] [-c count] [-i interval] [-s packetsize]  
    [-t ttl] [-w deadline] [-I interface] destination
```

- **-n**: non risolve gli IP in nomi di host;
- **-c count**: invia “count” pacchetti;
- **-i interval**: attende “interval” secondi tra l’invio di un pacchetto e l’altro;
- **-s packetsize**: specifica il numero di bytes da inviare. Di default sono 56 + 8 di header ICMP;
- **-t ttl**: specifica il campo IP Time To Live (numero massimo di hop);
- **-w deadline**: numero massimo in secondi prima che il ping termini;
- **-I interface**: setta l’indirizzo sorgente a quello associato all’interfaccia “interface” (nome dell’interfaccia o indirizzo IP).

Traceroute

- **Traccia la rotta** percorsa per raggiungere l'host remoto:

```
fujim10 [~] -> traceroute www.repubblica.it
traceroute to www.repubblica.it (213.92.16.171), 30 hops max, 40 byte packets
 1 di-gw.di.unipi.it (131.114.3.2)  0.514 ms  0.454 ms  0.680 ms
 2 131.114.191.126 (131.114.191.126)  1.302 ms  0.978 ms  1.527 ms
 3 fib-ser.unipi.it (131.114.191.49)  1.494 ms  1.090 ms  1.344 ms
 4 ru-unipi-rt-pil-1.pil.garr.net (193.206.136.13)  1.726 ms  1.419 ms  1.352 ms
 5 rt-pil-rt-tol.tol.garr.net (193.206.134.73)  8.091 ms  7.424 ms  6.729 ms
 6 rt-tol-rt-mi2.mi2.garr.net (193.206.134.41)  11.243 ms  8.661 ms  7.965 ms
 7 inet.mix-it.net (217.29.66.2)  9.175 ms  8.206 ms  8.095 ms
 8 ge5-1.wf1-gsr3.net.inet.it (212.239.97.113)  9.858 ms  9.904 ms  9.287 ms
 9 ge6-0-4.wf1-kwcore.wf.inet.it (212.239.110.14)  9.245 ms  9.328 ms  9.149 ms
10 * * *
...
```

Nestat

- Mostra lo **stato** e **statistiche** sulle connessioni TCP/IP attive.

```
fujim10 [~] -> netstat -antu
```

```
Active Internet connections (servers and established)
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	0.0.0.0:515	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:680	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:111	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:880	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:689	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:113	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:45810	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:47155	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:7741	0.0.0.0:*	LISTEN
tcp	0	0	131.114.11.210:1022	131.114.11.65:2049	ESTABLISHED
tcp6	0	0	:::22	:::*	LISTEN
tcp6	0	0	::ffff:131.114.11.21:22	::ffff:131.114.3.:39889	ESTABLISHED
udp	0	0	0.0.0.0:32768	0.0.0.0:*	
udp	0	0	0.0.0.0:32769	0.0.0.0:*	
udp	0	0	0.0.0.0:517	0.0.0.0:*	
udp	0	0	0.0.0.0:518	0.0.0.0:*	
udp	0	0	0.0.0.0:676	0.0.0.0:*	
udp	0	0	0.0.0.0:690	0.0.0.0:*	
udp	0	0	0.0.0.0:957	0.0.0.0:*	
udp	0	0	0.0.0.0:7741	0.0.0.0:*	
udp	0	0	0.0.0.0:877	0.0.0.0:*	
udp	0	0	0.0.0.0:111	0.0.0.0:*	
udp	0	0	0.0.0.0:1021	0.0.0.0:*	
udp	0	0	0.0.0.0:1022	0.0.0.0:*	
udp	0	0	0.0.0.0:1023	0.0.0.0:*	



Netstat: opzioni

Opzioni più utilizzate:

- **-a**: (all) mostra tutte le connessioni e le porte aperte;
- **-n**: (numeric) non risolve gli IP in hostname;
- **-p**: (pid) mostra il PID del processo che utilizza il socket;
- **-t**: (tcp) mostra le connessioni TCP;
- **-u**: (udp) mostra le sessioni UDP;
- **-r**: (route) mostra la tabella di routing dell'host.

```
fujim10 [~] -> netstat -r
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	MSS	Window	irrt	Iface
localnet	fire-int.cli.di	255.255.255.224	UG	0	0	0	eth0
131.114.120.0	*	255.255.255.128	U	0	0	0	eth0
131.114.11.0	*	255.255.255.0	U	0	0	0	eth0
default	fire-int.cli.di	0.0.0.0	UG	0	0	0	eth0



Arp

- Mostra la tabella delle **associazioni** tra indirizzo **hardware** e indirizzo **IP**.

```
fujim10 [~] -> arp
```

Address	HWtype	HWaddress	Flags Mask	Iface
giada.cli.di.unipi.it	ether	00:A0:C9:B5:50:14	C	eth0
raid.cli.di.unipi.it	ether	00:10:5A:F7:65:25	C	eth0
fire-int.cli.di.unipi.i	ether	00:A0:C9:97:59:2B	C	eth0

- L'opzione **-n** mostra gli indirizzi IP invece che gli hostname:

```
fujim10 [~] -> arp -n
```

Address	HWtype	HWaddress	Flags Mask	Iface
131.114.11.73	ether	00:A0:C9:B5:50:14	C	eth0
131.114.11.65	ether	00:10:5A:F7:65:25	C	eth0
131.114.11.64	ether	00:A0:C9:97:59:2B	C	eth0



Dig

- Dig (Domain Information Groper) è un tool per interrogare un name server e eseguire query di **DNS lookup**.

```
fujim10 [~] -> dig www.repubblica.it

; «» DiG 9.3.2 «» www.repubblica.it
;; global options: printcmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 60933
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
www.repubblica.it.          IN      A

;; ANSWER SECTION:
www.repubblica.it.        300     IN      A      213.92.16.171
www.repubblica.it.        300     IN      A      213.92.16.191

;; AUTHORITY SECTION:
repubblica.it.             7712    IN      NS     venere.inet.it.
repubblica.it.             7712    IN      NS     ns2.inet.it.

;; ADDITIONAL SECTION:
ns2.inet.it.                7672    IN      A      194.20.8.1
venere.inet.it.             7672    IN      A      194.20.8.4

;; Query time: 53 msec
;; SERVER: 131.114.11.73#53(131.114.11.73)
;; WHEN: Sat Nov 17 11:35:40 2007
;; MSG SIZE rcvd: 143
```

Nslookup

- Nslookup permette di fare query DNS **iterativamente**.

```
fujim10 [~] -> nslookup
> www.repubblica.it
Server:      131.114.11.73
Address:    131.114.11.73#53
```

Non-authoritative answer:

```
Name:  www.repubblica.it
Address: 213.92.16.171
Name:  www.repubblica.it
Address: 213.92.16.191
```

```
> www.corriere.it
Server:      131.114.11.73
Address:    131.114.11.73#53
```

Non-authoritative answer:

```
Name:  www.corriere.it
Address: 194.20.158.101
> www.di.unipi.it
Server:      131.114.11.73
Address:    131.114.11.73#53
```

Non-authoritative answer:

```
Name:  www.di.unipi.it
Address: 131.114.3.18
```



Route

- Mostra la **routing table**:

```
fujii7 [~] -> route
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
localnet	fire-int.cli.di	255.255.255.224	UG	0	0	0	eth0
131.114.120.0	*	255.255.255.128	U	0	0	0	eth0
131.114.11.0	*	255.255.255.0	U	0	0	0	eth0
default	fire-int.cli.di	0.0.0.0	UG	0	0	0	eth0

- L'opzione **-n** non risolve gli indirizzi IP in nome:

```
fujii7 [~] -> route -n
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
131.114.11.32	131.114.11.64	255.255.255.224	UG	0	0	0	eth0
131.114.120.0	0.0.0.0	255.255.255.128	U	0	0	0	eth0
131.114.11.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
0.0.0.0	131.114.11.64	0.0.0.0	UG	0	0	0	eth0

Route

- Per aggiungere regole di routing (da root):

```
route add default gw {IP-ADDRESS} {INTERFACE-NAME}
```

Dove:

- IP-ADDRESS: specifica l'indirizzo IP del router;
 - INTERFACE-NAME: specifica l'interfaccia di rete per la rotta.
- Esempi:

```
# route add -net 192.168.201.0 netmask 255.255.255.0 gw 192.168.200.254
```

Aggiunge la rotta verso la rete 192.168.201.0 alla routing table, con maschera di rete 255.255.255.0, tramite il gateway 192.168.200.254.

```
# route add default gw 192.168.1.254
```

Ad es., se il router ha l'IP 192.168.1.254, questa regola setta il default gateway.

Telnet

- Telnet può essere utilizzato per **diagnosticare** problemi di rete (ad es., porte non raggiungibili), o **testare** applicazioni di cui si conosce la sintassi utilizzata dal protocollo (SMTP, HTTP, FTP):

```
fujim10 [~] -> telnet fujim9 22
Trying 131.114.11.209...
Connected to fujim9.cli.di.unipi.it.
Escape character is '^]'.
SSH-2.0-OpenSSH_4.3p2 Debian-2
```

```
fujim10 [~] -> telnet fujim8 22
Trying 131.114.11.208...
telnet: Unable to connect to remote host: Connection refused
```

```
fujim10 [~] -> telnet www.di.unipi.it 80
Trying 131.114.3.18...
Connected to www.di.unipi.it.
Escape character is '^]'.
GET / HTTP/1.0
```

```
HTTP/1.1 200 OK
Date: Sat, 17 Nov 2007 11:06:24 GMT
Server: Apache
Content-Location: index.html.en
Vary: negotiate,accept-language
TCN: choice
Last-Modified: Mon, 12 Nov 2007 19:08:03 GMT
ETag: c3c072-2091-d340ac0
Accept-Ranges: bytes
Content-Length: 8337
...
```

File di Configurazione per il Networking

Questi file sono di solito modificabili solo da root:

- **/etc/hosts**: contiene il mapping statico fra indirizzi IP e hostname oltre agli alias.

```
127.0.0.1      localhost

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
ff02::3     ip6-allhosts
```

- **/etc/resolv.conf**: contiene gli indirizzi dei server DNS, il dominio dell'host e l'ordine di ricerca:

```
fujim10 [~] -> cat /etc/resolv.conf
search cli.di.unipi.it
nameserver 131.114.11.73
```

File di Configurazione per il Networking

- **/etc/services**: contiene il mapping tra i numeri di porta/protocollo e i nomi dei servizi.

```
tcpmux          1/tcp          # TCP port service multiplexer
echo            7/tcp
echo            7/udp
discard        9/tcp          sink null
discard        9/udp          sink null
systat         11/tcp         users
daytime        13/tcp
daytime        13/udp
netstat        15/tcp
gotd           17/tcp         quote
msp            18/tcp         # message send protocol
msp            18/udp
chargen        19/tcp         ttytst source
chargen        19/udp         ttytst source
ftp-data       20/tcp
ftp            21/tcp
fsp            21/udp         fspd
ssh            22/tcp         # SSH Remote Login Protocol
ssh            22/udp
telnet         23/tcp
smtp           25/tcp         mail
time           37/tcp         timserver
time           37/udp         timserver
rlp            39/udp         resource       # resource location
nameserver     42/tcp         name           # IEN 116
whois          43/tcp         nickname
...
```


Esercizio 1

- Progettare un'applicazione client/server in cui il server fornisca un **servizio di compressione** di dati.
- Il **client** `ZipClient` legge chunk di byte da un file e li spedisce al server che provvede alla loro compressione.
- Il **server** `ZipServer` restituisce i byte in formato compresso al client che provvede a creare un file con lo stesso nome del file originario e con estensione `.gz`, che contiene i dati ricevuti dal server.
- La comunicazione tra client e server utilizza il protocollo **TCP**. Per la compressione si può utilizzare la classe JAVA `GZIPOutputStream`.
- **Attenzione:** `ZipServer` non deve aspettare di ricevere tutto il file prima di rispondere al client, ma ogni volta che riceve un chunk di byte lo invia immediatamente al client tramite il `GZIPOutputStream`.

Esercizio 1

- All'avvio del client, specificare **IP e porta del server remoto** e il **nome del file** da inviare.
- Requisito: `ZipClient` e `ZipServer` devono essere eseguiti su due **host diversi**.
- In generale, è preferibile non eseguire i processi **client** e **server** sulla stessa macchina utilizzando un'interfaccia locale per le connessioni.
 - Il sistema operativo può mettere a disposizione delle interfacce di **loopback** (interfacce virtuali: non sono associate ad alcun dispositivo fisico).
 - Su ogni PC l'hostname `localhost` identifica l'interfaccia di loopback con l'IP `127.0.0.1`.

Come Eseguire l'Applicazione Distribuita?

- Soluzione: usare **ssh** (secure shell), un protocollo utilizzato per stabilire una sessione remota cifrata con un host remoto
- Es.: `ZipClient` è eseguito localmente sull'host **fujim9** e tramite `ssh` `ZipServer` è avviato su **fujim10**.
- Es: il percorso completo dei due file è (usare due directory diverse): `~/java/client/ZipClient.java`,
`~/java/server/ZipServer.java`.
 - Il simbolo `~` (tilde) identifica l'home directory.

Esempio

1. Sul computer **fujim9** viene eseguito il client localmente.
2. Da **fujim9** si effettua il comando `ssh fujim10` per loggarsi remotamente su **fujim10**.
3. Su **fujim10** si digita `cd java/server/` per spostarsi nella directory del server.
4. Si avvia `java ZipServer porta` su **fujim10** (dopo averlo compilato con `javac`).
5. Si avvia `java ZipClient fujim10 porta file` su **fujim9** (dopo averlo compilato con `javac`).
6. Per uscire dalla sessione remota, ed effettuare il logout da **fujim10**, digitare `exit`.

Esempio

Sul **server**:

```
fujim10 [~/java/server] -> javac ZipServer.java
fujim10 [~/java/server] -> java ZipServer 10000
Ricevuta richiesta da 131.114.11.209 porta 48583
```

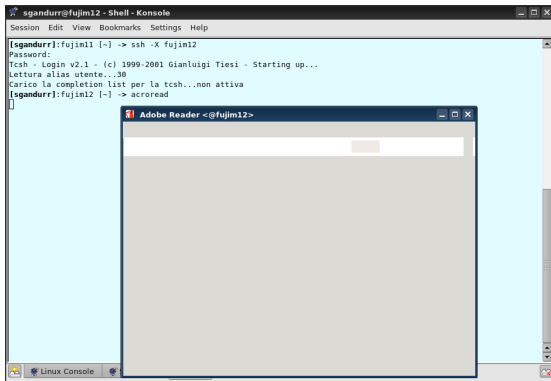
Sul **client**:

```
fujim9 [~/java/client] -> javac ZipClient.java
fujim9 [~/java/client] -> java ZipClient fujim10 10000 prova.txt
salvato prova.txt.gz
fujim9 [~/java/client] -> mv prova.txt prova2.txt
fujim9 [~/java/client] -> gunzip -d prova.txt.gz
fujim9 [~/java/client] -> diff prova.txt prova2.txt
fujim9 [~/java/client]
```



SSH e X11 Forwarding

Nel caso di applicazioni client/server Java che fanno uso di AWT/SWING, per abilitare la **redirezione** del display remoto in locale, usare l'opzione `-X` (X maiuscola): `ssh -X user@hostname`.



File System Distribuito

- Al polo, l'**home directory** viene “montata” sul File System locale di ogni host da uno storage remoto, tramite protocollo NFS (Network File System).
- Effettuando una modifica ad un file, la modifica è **globale** e quindi visibile da tutti i PC.
- Es.: modificando da **fujim9** il file `ZipServer.java`, le modifiche sono visibili anche da **fujim10**.
- Questo permette di modificare e compilare i file da un **unico** host.

Gli Hostname e SFTP

- L'**hostname** completo è `fujim10.cli.di.unipi.it`.
- Poiché il **dominio** su cui risiedono i computer è `cli.di.unipi.it`, per connettersi dall'interno del polo si può semplicemente scrivere `ssh fujim10`.
- Per connettersi da remoto tramite **ssh**, il dominio va inserito, per cui il comando completo è:
`ssh username@fujim10.cli.di.unipi.it`.
- Analogamente, per trasferire file si può usare **sftp**. Il comando è:
`sftp username@fujim10.cli.di.unipi.it`
 - Per fare l'**upload** di un file: `put nomefile`.
 - Per fare il **download** di un file: `get nomefile`.

Soluzioni

Inviare la soluzione degli esercizi (solo i file .java) a :

`ricci@di.unipi.it`

`sgandurra@di.unipi.it`

Tra due settimane saranno disponibili le soluzioni.