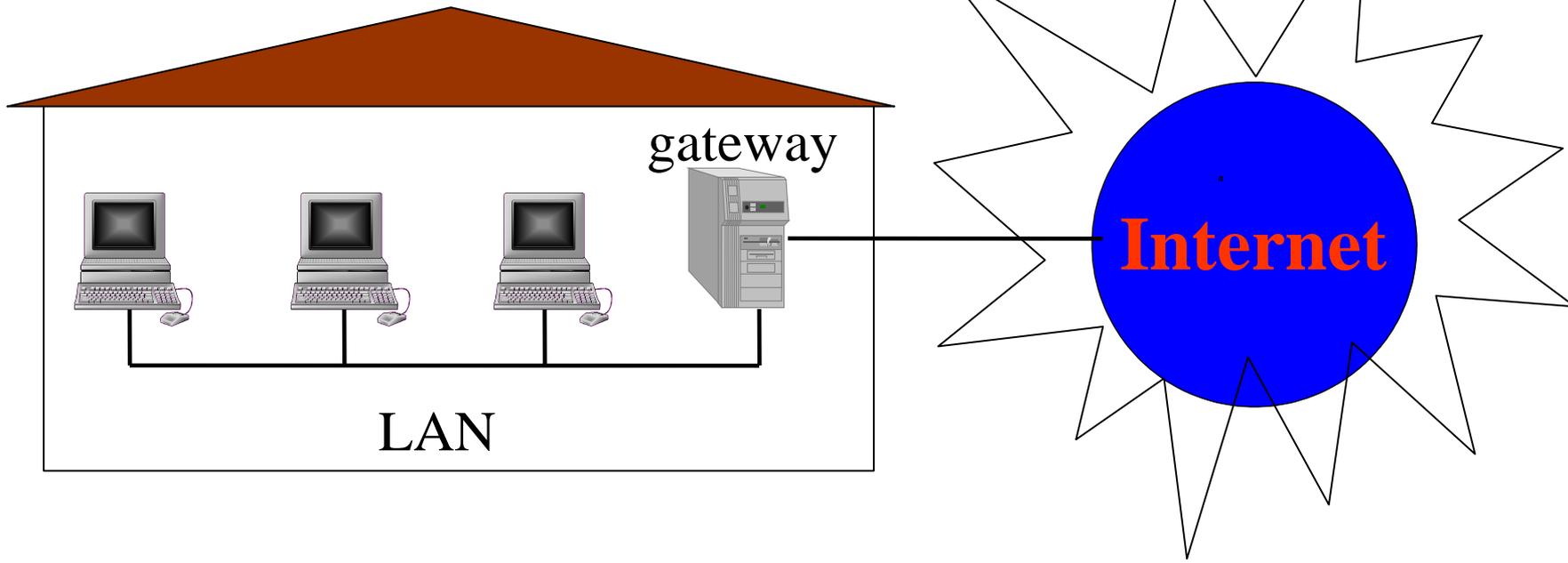


# Reti di Computers

# Rete di Computers

insieme di computer collegati tra loro da una struttura di interconnessione

- locale (Local Area Network): edificio
- metropolitana (Metropolitan Area Network): città
- geografica (Wide Area Network): nazione
- internet: mondo



# Rete di Computers

- Esistono vari tipi di rete
  - ethernet (più diffusa)
  - token ring
  - DQDB
  - .....
- ogni computer ha una scheda di rete (NIC)
- tra un computer e l'altro vi è un cavo di rete
  - coassiale (vecchio) 500m max
  - doppino (10baseT) 100m max
  - ottico (10baseF) 2000m max

# Rete di Computers

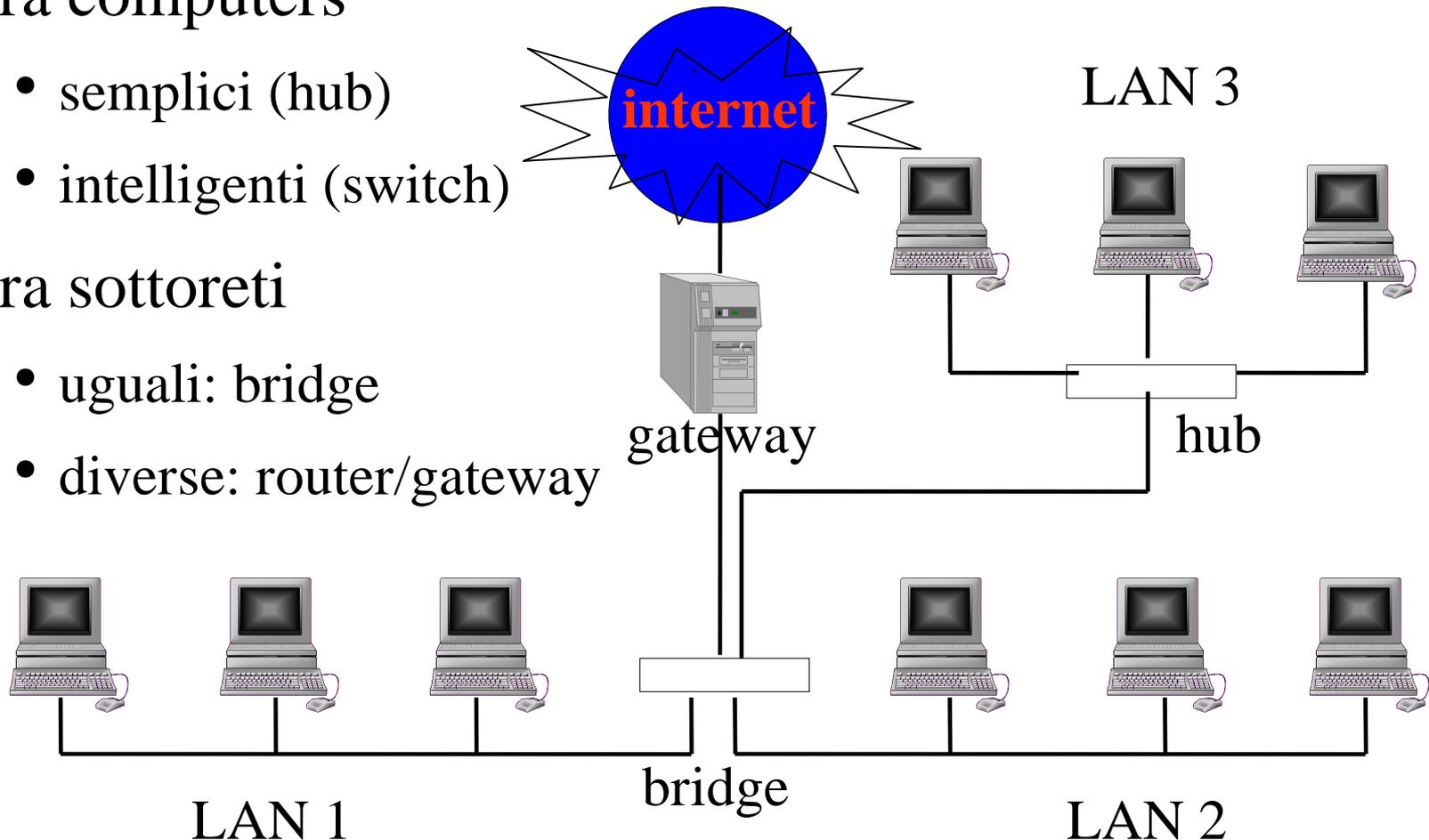
nella rete ci sono dei ripetitori

– tra computers

- semplici (hub)
- intelligenti (switch)

– tra sottoreti

- uguali: bridge
- diverse: router/gateway



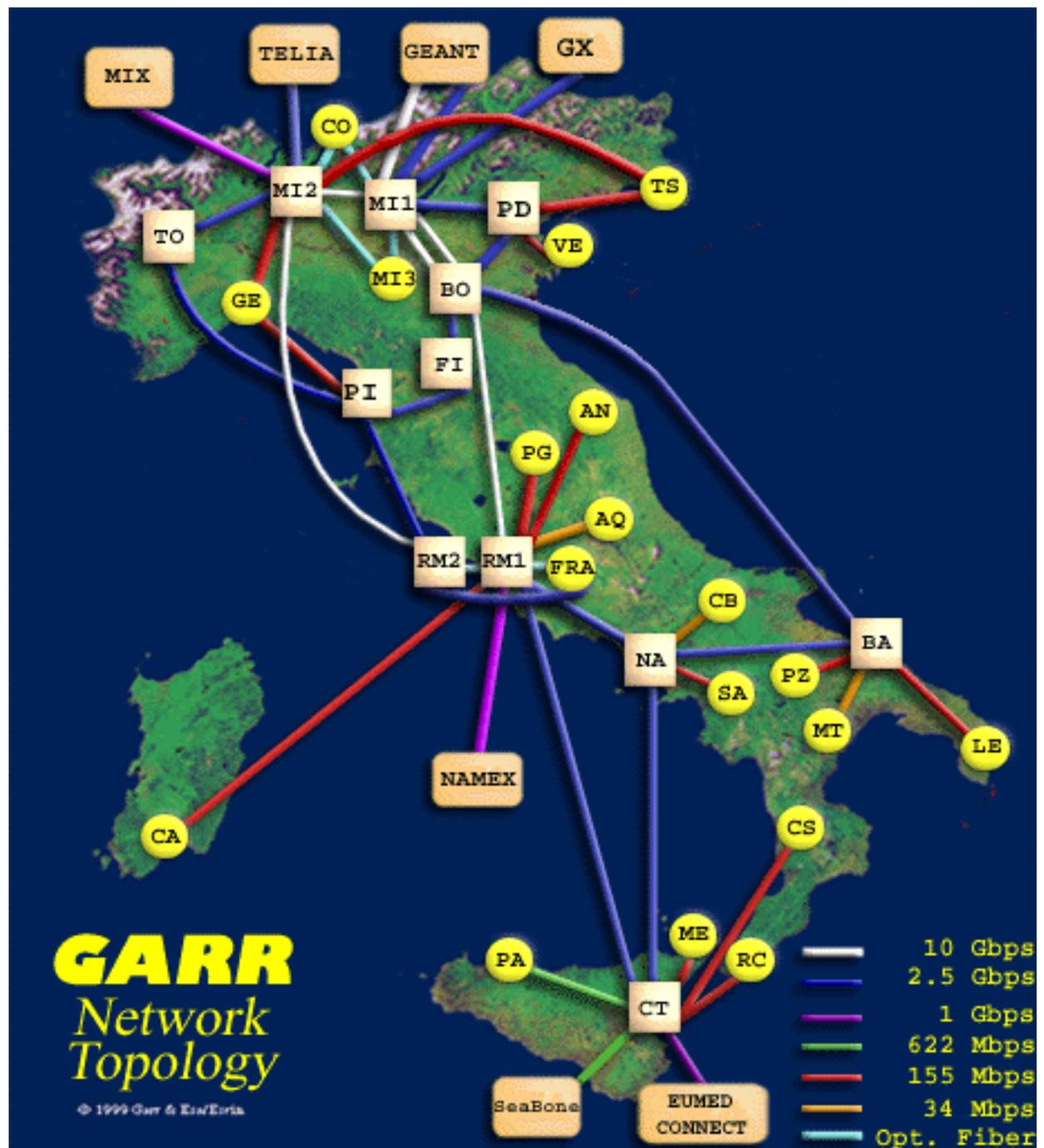
# Internet

- 1969: ARPANET rete sperimentale con 4 nodi della Advanced Research Projects Agency
- 1983: ARPANET conta centinaia di nodi
  - passa alla Defence Communication Agency
  - viene separata MILNET la rete militare
- 1984: NSFNET gestita da National Science Foundation (1,5 Mbps)
  - dorsale americana collegata a reti regionali
  - collegata con ARPANET

# Internet

- 1990: ARPANET viene smantellata
- 1990: Advanced Network and Services rileva NSFNET e crea ANSNET (45 Mbps)
- moltissime altre reti sono nate in seguito
- INTERconnected NETworks:
  - è costituita dall'insieme di tutte le reti collegate
  - non esiste una data di creazione vera e propria
  - 1990: 200.000 computer collegati e 3000 reti
  - 1992: 1.000.000 di computer collegati
  - 1995: milioni di computer, decine di migliaia di reti

# GARR



# Servizi di rete

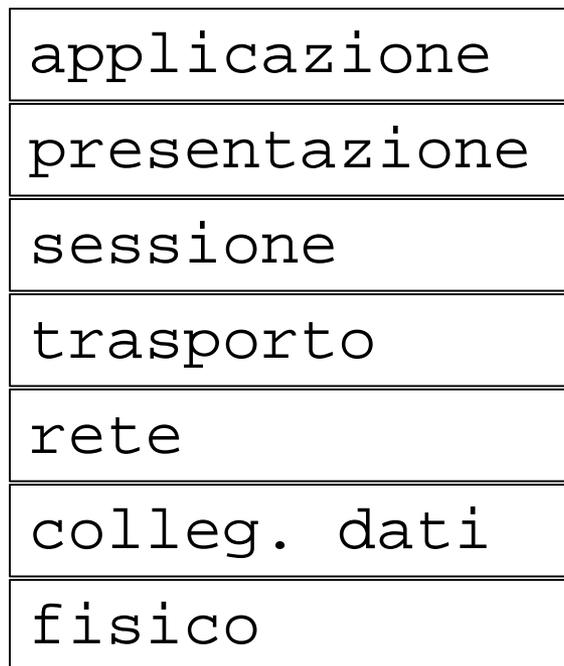
- posta elettronica (e-mail)
- trasferimento files (ftp, scp)
- utilizzo computer remoti (telnet, ssh,....)
- gruppi di interesse (newsgroups)
- navigazione in world wide web (www)

# Pacchetti

- le informazioni viaggiano sulla rete sotto forma di pacchetti
- la costruzione del pacchetto che viaggia sulla rete dipende dal tipo di rete utilizzato
- tra il messaggio che si vuole spedire e il pacchetto che viaggia sulla rete ci sono vari livelli di impacchettamento

# Modello OSI/ISO

- stabilisce uno standard dei livelli per la costruzione dei pacchetti



# Modello TCP/IP

- standard alternativo dei livelli per la costruzione dei pacchetti

applicazione
trasporto
rete
fisico

# Descrizione livelli TCP/IP

- fisico: hardware e protocollo scheda di rete
- rete: protocollo IP
- **trasporto: protocolli TCP e UDP**
- applicazione: protocolli telnet, ssh, ftp, http...

# Protocollo TCP

- **Transmission Control Protocol**
  - affidabile (3way handshake)
  - ordinamento pacchetti
  - orientato alla connessione
  - punto punto (no broadcast/multicast)
  - full duplex

# Protocollo UDP

- **User Data Protocol**
  - non affidabile (può perdere pacchetti)
  - ricezione pacchetti non ordinati
  - senza connessione
  - punto punto o multicast
  - molto piu' efficiente di TCP

# Comunicare in Rete

Per comunicare con un processo su un host in rete (remoto) occorre identificare:

- Indirizzo IP

- Rete
- Host

- Porta

# Indirizzi IPv4

- gli indirizzi IPv4 si rappresentano come una quadrupla di numeri decimali a 8 bit (0-255)
- ogni rete ha
  - un indirizzo IP unico (ad es. 131.114.0.0)
  - un nome unico (ad es. di.unipi.it)
- ogni computer nella rete ha:
  - un indirizzo IP unico: (ad es. 131.114.5.123)
  - un nome unico: nomecomputer.dominio

# Indirizzi IPv6

- gli indirizzi IPv6 si rappresentano come una ottupla di numeri esadecimali a 16 bit (0-FFFF)

ad es. 1080:0:0:0:8:800:200C:417A

# Classi di indirizzi IPv4

esistono 5 classi di indirizzi IPv4

rete

host

– classe A

• 0-----.-----.-----.-----

il primo bit  
ha sempre  
valore 0

• da 1.0.0.0 a 126.255.255.255

Rappresentazione binaria indirizzo IP

• il primo numero indica la rete, gli altri identificano le sottoreti e gli hosts

– classe B

• 10-----.-----.-----.-----

i primi due  
bit hanno  
sempre  
valore 10

• da 128.0.0.0 a 191.255.255.255

• i primi due numeri indicano la rete....

# Classi di indirizzi IPv4

## – classe C

- 110----- . ----- . ----- . -----
- da 192.0.0.0 a 223.255.255.255
- i primi tre numeri indicano la rete...

## – classe D: multicast

- 1110----- . ----- . ----- . -----
- da 224.0.0.0 a 239.255.255.255

## – classe E: usi speciali

- 11110----- . ----- . ----- . -----

# Indirizzo di rete e broadcast

- il primo indirizzo (0) di una rete individua la rete stessa

10.0.0.0

- l'ultimo indirizzo individua tutti gli host di quella rete (broadcast)

10.255.255.255

# Tipi di Indirizzi IPv4

- Pubblico
  - unico in internet
  - assegnato dall' autorità IANA
- Privato
  - non utilizzato in internet
  - può essere utilizzato in reti non collegate direttamente ad internet

# Indirizzi IPv4 per reti private

- Classe A da 10.-.-.- a 10.-.-.-
- Classe B da 172.16.-.- a 172.31.-.-
- Classe C da 192.168.0.- a 192.168.255.-

# Maschera di rete

- individua la parte di un indirizzo IP che rappresenta la rete
- AND bit a bit con indirizzo IP in binario

indirizzo IP host = 10.1.1.23

maschera = 255.0.0.0

indirizzo IP rete = 10.0.0.0

# Assegnamento Indirizzi IP all' host

- statico
  - l'indirizzo IP è fissato quando la rete viene configurata
- dinamico
  - l'indirizzo viene assegnato ad ogni avvio dal servizio DHCP

# Porta

- Numero intero da 1 a 65535
- Individua il processo con cui comunicare
- Le porte da 1 a 256 sono riservate a servizi noti
  - 22 ssh
  - 80 http
- Il file `/etc/services` elenca le porte dei servizi noti
- Le porte da 257 a 1023 sono riservate a root
- Le porte TCP e UDP sono distinte

# /etc/services

```
ftp-data      20/tcp
ftp-data      20/udp # 21 is registered to ftp,
but also used by fsp
ftp           21/tcp
ftp           21/udp          fsp fspd

ssh           22/tcp # SSH Remote Login Protocol
ssh           22/udp # SSH Remote Login Protocol

telnet        23/tcp
telnet        23/udp

smtp          25/tcp          mail
smtp          25/udp          mail
```

# Controllo di instradamento

- per richiedere un risposta ad un host  
`ping <host>`
- opzioni
  - `-c` termina dopo aver ricevuto `c` risposte
  - `-f` invia più pacchetti possibile
  - `-s` dimensione del pacchetto

# ping

```
ping -s 300 -c 10 qua.di.unipi.it
```

```
PING qua.di.unipi.it (131.114.4.118): 300 data bytes  
308 bytes from 131.114.4.118: icmp_seq=1 ttl=52 time=290.0 ms  
308 bytes from 131.114.4.118: icmp_seq=4 ttl=52 time=280.0 ms  
308 bytes from 131.114.4.118: icmp_seq=7 ttl=52 time=280.0 ms  
308 bytes from 131.114.4.118: icmp_seq=8 ttl=52 time=300.0 ms
```

```
--- qua.di.unipi.it ping statistics ---
```

```
10 packets transmitted, 4 packets received, 60% packet loss  
round-trip min/avg/max = 280.0/287.5/300.0 ms
```

# Controllo di instradamento

- per verificare il percorso fino ad un host  
`traceroute <host>`
- opzioni
  - `-n` non traduce gli indirizzi IP in nomi
  - `-m <num>` `num` è il massimo numero di nodi da attraversare

# traceroute

```
traceroute qua.di.unipi.it
1  10.10.10.1 (10.10.10.1)  136.498 ms  129.732 ms  130.006 ms
2  151.5.148.97 (151.5.148.97)  139.850 ms  129.852 ms
3  151.5.206.1 (151.5.206.1)  129.887 ms  129.966 ms  129.828 ms
4  gw3.iunet.it (192.106.1.145)  139.957 ms  129.846 ms
5  192.94.212.74 (192.94.212.74)  149.905 ms
6  garrb.mix.aiip.it (194.177.110.39)  139.558 ms * 209.992 ms
7  * milano-mix.garr.net (193.206.134.177)  140.305 ms *
8  * bo-mi-1.garr.net (193.206.134.2)  220.002 ms *
9  pi-bo-2.garr.net (193.206.134.86)  160.147 ms * *
10 * * unipi-rc.pi.garr.net (193.206.136.18)  320.038 ms
11 * eth04-gw.unipi.it (131.114.188.50)  219.943 ms *
12 * di-gw.unipi.it (131.114.191.5)  240.121 ms *
13 * qua.di.unipi.it (131.114.4.118)  180.062 ms  159.855 ms
```

# Interfaccia di rete

- ad ogni interfaccia (scheda) di rete deve essere attribuito un indirizzo IP
- per vedere lo stato di un'interfaccia di rete:

```
ifconfig <inter>
```

# ifconfig

```
ifconfig eth0
```

```
Link encap:Ethernet HWaddr 52:54:00:EB:EC:C6
```

```
inet addr:192.168.0.1 Bcast:192.168.0.255
```

```
Mask:255.255.255.0
```

```
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

```
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
```

```
TX packets:127 errors:0 dropped:0 overruns:0 carrier:0
```

```
collisions:0 txqueuelen:100
```

```
Interrupt:10 Base address:0xb800
```

# Indirizzi IP e nomi

- l'utilizzo degli indirizzi IP non è agevole per gli utenti
- agli indirizzi IP vengono associati dei nomi
- la conversione nome $\leftrightarrow$ indirizzo IP può essere fatta:
  - tramite files locali
  - tramite il servizio DNS (Domain Name Server)

# Conversione nomi

- per la conversione nome $\leftrightarrow$ indirizzo IP occorre predisporre i files:
  - `/etc/hosts.conf`: determina quali servizi di conversione usare
  - `/etc/hosts`: contiene le coppie nome-indirizzo IP
  - `/etc/resolv.conf`: contiene l'indirizzo IP del server dei nomi (DNS)

# /etc/hosts

formato= <indirizzo\_IP> <nome\_host>

```
127.0.0.1    localhost localhost.localdomain
192.168.0.1  pippo.di.unipi.it
192.168.0.2  pluto.di.unipi.it
```

# /etc/resolv.conf

```
domain di.unipi.it
```

dominio

```
search di.unipi.it
```

dominio di default da dare ai nomi senza

```
nameserver 131.114.4.6
```

```
nameserver 131.114.4.7
```

indirizzi IP dei server dei nomi

# Conversione nomi

- per convertire un nome di host in indirizzo IP  
`host <host_name>`
- per converire un indirizzo IP in nome di host  
`host <ip_addr>`
- per visualizzare il nome dell'host locale  
`hostname`

# Connessioni attive

- per elencare le connessioni attive

`netstat`

- opzioni

- `-t` connessioni TCP

- `-u` connessioni UDP

- `-a` visualizza anche socket in attesa di connessione

# netstat

```
netstat -ta
```

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	*:sunrpc	::*	LISTEN
tcp	0	0	apollo:ipp	::*	LISTEN
tcp	0	0	apollo:5335	::*	LISTEN
tcp	1	0	apollo:52912	apollo:ipp	CLOSE_WAIT
tcp	0	0	*:ssh	::*	LISTEN

# InetAddress

```
import java.net.*;

public class ProvaAddress {

    public static void main(String[] args) {

        try {
            InetAddress h1 = InetAddress.getByName("giglio.di.unipi.it");
            InetAddress h2 = InetAddress.getByName("131.114.2.67");

            System.out.println("host 1: nome " + h1.getHostName() +
                " indirizzo " + h1.getHostAddress());
            System.out.println("host 2: nome " + h2.getHostName() +
                " indirizzo " + h2.getHostAddress());
        }
        catch(UnknownHostException e) {
            System.err.println("errore nella risoluzione");
        }
    }
}
```

# Output

```
host 1: nome giglio.di.unipi.it indirizzo 131.114.2.67  
host 2: nome giglio.di.unipi.it indirizzo 131.114.2.67
```

# InetAddress: host locale

```
import java.net.*;

public class ProvaAddress {

    public static void main(String[] args) {

        try {
            InetAddress lhost = InetAddress.getLocalHost();

            System.out.println("local host: " + lhost.toString());

            System.out.println("lhost e' un indirizzo di loopback? " +
                lhost.isLoopbackAddress());
        }
        catch(UnknownHostException e) {
            System.err.println("errore nella risoluzione");
        }
    }
}
```

# InetAddress

```
import java.net.*;

public class ProvaAddress {

    public static void main(String[] args) {

        try {
            InetAddress[] hosts=InetAddress.getAllByName("giglio.di.unipi.it");

            System.out.println("l'host ha " + hosts.length + " interfacce");

            for (int i=0; i<hosts.length; i++)
                System.out.println("indirizzo " + hosts[i].getHostAddress());

        }
        catch(UnknownHostException e) {
            System.err.println("errore nella risoluzione");
        }
    }
}
```

# InetAddress

```
import java.net.*;

public class ProvaAddress {

    public static void main(String[] args) {

        try {
            InetAddress h1 = InetAddress.getByName("giglio.di.unipi.it");
            InetAddress h2 = InetAddress.getByName("131.114.2.67");

            System.out.println("host 1 e host 2 sono lo stesso host? " +
                h1.equals(h2));

        }
        catch(UnknownHostException e) {
            System.err.println("errore nella risoluzione");
        }
    }
}
```