



# Configurazione IPv6 sulle diverse tipologie di host

## Linux

Andrea Di Peo - GARR NOC

9' Workshop GARR "Al servizio degli utenti" - Roma, 15 - 18 Giugno 2009



# Obiettivi del tutorial

---

- Considerazioni e Configurazioni di IPv6 su host Linux
- Controlli preliminari sulla compatibilità del proprio sistema con IPv6
- Descrizione delle tipologie di indirizzi IPv6 e come configurarli od ottenerli su Linux
- Linux in Rete con IPv6! (anche attraverso i Tunnel)

# Kernel

- Condizione indispensabile per poter utilizzare IPv6 su Linux è che il kernel sia compatibile con questo protocollo
- Il kernel è ufficialmente rilasciato da Linus Torvalds e dai progettisti Linux, quello senza alcuna patch aggiuntiva e' denominato vanilla
- La maggior parte delle distribuzioni usano un kernel generalmente patchato in base alle esigenze della distro stessa
- Il primo codice relativo a IPv6 fu aggiunto al kernel Linux 2.1.8 nel Novembre 1996 (si trattava di una patch)
- Il supporto per IPv6 nel kernel ha conosciuto uno sviluppo graduale. Per sopperire alle carenze iniziali prima, e per offrire funzionalita' supplementari in seguito, e' nato il progetto USAGI il cui intento è quello di implementare tutto il supporto IPv6 di cui il kernel deficiava



# Kernel

- Se il precedente comando fallisce il modulo IPv6 non è stato caricato
- Si può quindi provare quindi a caricare il modulo IPv6 eseguendo:

```
andrea@andrea-desktop:~$ modprobe ipv6
andrea@andrea-desktop:~$ lsmod |grep -w 'ipv6' && echo "IPv6 module successfully loaded"
ipv6                267908  10
IPv6 module successfully loaded
```

- È possibile caricare automaticamente il modulo IPv6 all'avvio editando il file */etc/rc.d/rc.local* aggiungendo:

```
#!/bin/sh
modprobe ipv6
```

- (oppure aggiungendo la riga `'modprobe ipv6'` in */etc/rc.d/rc.modules* o settando un alias in */etc/modules.conf*)

Se le operazioni precedenti avessero dato esito negativo e il kernel non avesse il supporto per IPv6, si hanno le seguenti possibilità:

- Aggiornare la propria distribuzione con una versione che supporti IPv6
- Utilizzare un nuovo kernel "vanilla"
- Ricompilare i sorgenti del kernel della propria distribuzione Linux
- Compilare un kernel con le estensioni USAGI

# Configurazione IPv6 – Linux Kernel

- **Compilare con supporto IPv6 nativo (incluso nel kernel) o modulare**

- **cd /usr/local/src/linux-2.6.29**  
**make menuconfig** (richiede ncurses e ncurses-dev)

**Networking support -> Networking options -> The IPv6 protocol ->**

```
x x      --- The IPv6 protocol                                x x
x x      [*] IPv6: Privacy Extensions support                x x
x x      [ ] IPv6: Router Preference (RFC 4191) support      x x
x x      [ ] IPv6: AH transformation                          x x
x x      [ ] IPv6: ESP transformation                         x x
x x      [ ] IPv6: IPComp transformation                     x x
x x      [ ] IPv6: IPsec transport mode                      x x
x x      [ ] IPv6: IPsec tunnel mode                         x x
x x      [ ] IPv6: IPsec BEET mode                           x x
x x      [*] IPv6: IPv6-in-IPv4 tunnel (SIT driver)           x x
x x      [*] IPv6: IP-in-IPv6 tunnel (RFC2473)
```

- **<http://tomoyo.sourceforge.jp/cgi-bin/lxr/source/net/ipv6/Kconfig>**

# Configurazione IPv6 – Linux Kernel

- altrimenti nel .config del kernel devono essere abilitate:

-supporto per IPv6:  
CONFIG\_IPV6=y

-Privacy Extensions for Stateless Address Autoconfiguration in IPv6  
CONFIG\_IPV6\_PRIVACY=y

-per router advertisement:  
CONFIG\_IPV6\_ROUTER\_PREF=y

-IPv4 in IPv6:  
CONFIG\_IPV6\_SIT=y  
CONFIG\_IPV6\_NDISC\_NODETYPE=y

-IPv6-in-IPv6 and IPv4-in-IPv6:  
CONFIG\_IPV6\_TUNNEL=y

-per IPsec v6:  
CONFIG\_INET6\_AH=y CONFIG\_INET6\_ESP=y  
CONFIG\_INET6\_IPCOMP=y CONFIG\_INET6\_XFRM\_MODE\_TRANSPORT=y  
CONFIG\_INET6\_XFRM\_MODE\_TUNNEL=y  
CONFIG\_INET6\_XFRM\_MODE\_BEET=y

- Caricare il modulo del kernel per il supporto IPv6
  - **modprobe ipv6**



## I differenti tipi di indirizzi IPv6:

IPv6 suddivide gli indirizzi in:

**Unicast**: identifica univocamente un'interfaccia

**Multicast**: indirizzi di gruppi di nodi

**Anycast**: indirizzi di servizi

Come ottenerli?

# Indirizzi Unicast

---

- Unspecified
- Loopback
- IPv4 Compatibili
- IPv4 Mapped
- Indirizzi Scoped:
  - Link-local
  - Site-local
- Aggregatable Global

## Unspecified:

- Indica l'assenza di indirizzo
- Il suo formato è 0:0:0:0:0:0:0:0 o semplicemente ::
- Come 0.0.0.0 in IPv4 (::/0 indica la rotta di default)
- Può essere usato nella richiesta iniziale DHCPv6 per ottenere un indirizzo
- Può essere usato nel controllo sull'eventuale duplicazione di indirizzo: Duplicate Address Detection (DAD)
- Non può essere assegnato ad alcuna interfaccia

# Indirizzi Unicast

## Loopback:

- E' rappresentato con 0:0:0:0:0:0:0:1 o semplicemente ::1
- Identifica il nodo stesso, come 127.0.0.1 in IPv4 (localhost)
- Per controllare se lo stack IPv6 funziona si può effettuare un ping6 all' ::1

```
andrea@andrea-desktop:~$ ping6 ::1
PING ::1(::1) 56 data bytes
64 bytes from ::1: icmp_seq=1 ttl=64 time=0.028 ms
64 bytes from ::1: icmp_seq=2 ttl=64 time=0.019 ms
64 bytes from ::1: icmp_seq=3 ttl=64 time=0.020 ms

--- ::1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.019/0.022/0.028/0.005 ms
```

## Link e Site local

- Lo scope (cioè l'ambito) in cui è confinato un indirizzo link-local è una Lan, una Vlan o un collegamento punto-punto (i nodi sullo stesso link sono detti neighbor)
  - Può essere usato solo fra nodi dello stesso link, non può essere routato
  - Sono stati concepiti per essere utilizzati per scopi di configurazione automatica della scheda o per la scoperta dei neighbor
- Lo scope di un site-local è invece un site, cioè gruppo di link gestiti da un'unica autorità (ad esempio il campus di un'università)
  - Può essere usato soltanto fra nodi dello stesso site, non può essere routato
  - Molto simile agli indirizzi privati IPv4
  - Non è configurato di default ed è deprecato

# Indirizzi Unicast

## Site-local address:

- Gli indirizzi site-local devono essere configurati esplicitamente, per farlo possiamo utilizzare il seguente comando:

```
ifconfig eth1 inet6 add fec0::1:2a0:24ff:fe77:4997/64
```

```
root@andrea-desktop:/home/andrea# ifconfig eth1 inet6 add fec0::1:2a0:24ff:fe77:4997/64
root@andrea-desktop:/home/andrea# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:16:35:79:41:ed
          inet addr:193.206.158.135  Bcast:193.206.158.255  Mask:255.255.255.0
          inet6 addr: 2001:760::216:35ff:fe79:41ed/64 Scope:Global
          inet6 addr: fe80::216:35ff:fe79:41ed/64 Scope:Link
          inet6 addr: fec0::1:2a0:24ff:fe77:4997/64 Scope:Site
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:20127402 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8131387 errors:0 dropped:0 overruns:0 carrier:0
          collision:0 txqueuelen:1000
          Byte RX:1981890145 (1.8 GB)  Byte TX:1535072460 (1.4 GB)
```

## Global address:

- Gli indirizzi Global aggregatable non hanno limitazioni nel loro campo d'azione e possono essere usati per comunicare ovunque
- Possono essere configurati manualmente
- Possono essere ottenuti automaticamente con l'aggiunta di altri parametri di rete quali Gateway o DNS (autoconfigurazione statefull), ad esempio con DHCPv6
- Possono essere ottenuti automaticamente via Stateless Address AutoConfiguration (SLAAC), possibile anche questa con DHCPv6 o con prefix-discovery
- Il formato utilizzato è: 2xxx: oppure 3xxx:

# Indirizzi Unicast

## Global address: Come ottenerlo

- Analogamente a quanto visto prima con il Site-local address possiamo configurare l'indirizzo IPv6 global con lo stesso metodo:

```
root@andrea-desktop:/home/andrea# ifconfig eth1 inet6 add 2001:760::216:35ff:fe79:41ed/64
root@andrea-desktop:/home/andrea# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:16:35:79:41:ed
          inet  addr:193.206.158.135  Bcast:193.206.158.255  Mask:255.255.255.0
          inet6 addr: 2001:760::216:35ff:fe79:41ed/64 Scope:Global
          inet6 addr: fe80::216:35ff:fe79:41ed/64 Scope:Link
          inet6 addr: fec0::1:2a0:24ff:fe77:4997/64 Scope:Site
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:20307118 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8140268 errors:0 dropped:0 overruns:0 carrier:0
          collisioni:0 txqueuelen:1000
          Byte RX:1995209060 (1.8 GB)  Byte TX:1538562915 (1.4 GB)
          Interrupt:16
```



# Indirizzi Multicast

- Identificano una serie di indirizzi assegnati ad un insieme di interfacce, un pacchetto inviato ad un indirizzo multicast viene consegnato a tutte le interfacce identificate da quell'indirizzo.
- In IPv6 il concetto di broadcast non esiste. Multicast e' usato al suo posto
- Il formato degli indirizzi Multicast è: FF00::/8:
  - All'indirizzo FF01::1 partecipano tutte le interfacce sullo stesso nodo
  - All'indirizzo FF02::1 partecipano tutte le interfacce sullo stesso link
  - All'indirizzo FF05::1 partecipano tutte le interfacce sullo stesso site
  - All'indirizzo FF0E::1 partecipano tutte le interfacce su internet
- Variando il Group-id si identifica un gruppo Multicast all'interno di un dato scope (ad esempio FF02::101 identifica il servizio NTP su tutte le interfacce dello stesso link link)

# Indirizzi Anycast

- Identificano un insieme di interfacce; un pacchetto inviato ad un anycast, viene recapitato ad una delle interfacce che fanno parte dell'insieme da esso individuato (tipicamente da quella più 'vicina');
- Gli indirizzi anycast sono indirizzi speciali e vengono utilizzati per trattare cose come il server DNS o DHCP più vicino o simili.
- Gli indirizzi Anycast non sono distinguibili dagli indirizzi unicast
  - Sono indirizzi unicast assegnati ad un gruppo di interfacce
  - Gli indirizzi anycast non possono essere usati come indirizzo del mittente, ma sono validi soltanto come indirizzi di destinazione
- Un esempio di un indirizzo anycast è il tipo subnet-router. Assumendo che un nodo abbia il seguente indirizzo globale IPv6:  
3ffe:ffff:100:f101:210:a4ff:fee3:9566/64 (host).L'indirizzo anycast subnet-router sarà creato svuotando completamente il suffisso (i 64 bit meno significativi): 3ffe:ffff:100:f101::/64

# In Rete

- Per eseguire tutte le configurazioni necessarie all'host per andare in Rete è opportuno effettuare un controllo sugli strumenti che ci permettono queste configurazioni
- Uno dei comandi più diffusi per la configurazione delle interfacce è ifconfig, un controllo sulla compatibilità di questo con IPv6 può essere effettuato con il comando:

```
/sbin/ifconfig -? 2>& 1|grep -qw 'inet6' &&  
echo "utility 'ifconfig' is IPv6-ready"
```

- Anche il comando con cui possiamo controllare e modificare gli instradamenti dell' host (route) dovrebbe essere testato con:

```
/sbin/route -? 2>& 1|grep -qw 'inet6' &&  
echo "utility 'route' is IPv6-ready"
```

```
andrea@andrea-desktop:~$ /sbin/ifconfig -? 2>& 1|grep -qw 'inet6' && echo "utility ifconfig is IPv6-ready"  
utility 'ifconfig' is IPv6-ready  
andrea@andrea-desktop:~$ /sbin/route -? 2>& 1|grep -qw 'inet6' && echo "utility route is IPv6-ready"  
utility 'route' is IPv6-ready
```

- Abilitare/disabilitare il supporto per l'autoconfiguration
  - `echo "0"["1"] > /proc/sys/net/ipv6/conf/eth0/accept_ra`
  - `echo "0"["1"] > /proc/sys/net/ipv6/conf/eth0/autoconf`

**more /etc/protocols | grep ipv6**

hopopt	0	HOPOPT	# hop-by-hop options for ipv6
ipv6	41	IPv6	# IPv6
ipv6-route	43	IPv6-Route	# Routing Header for IPv6
ipv6-frag	44	IPv6-Frag	# Fragment Header for IPv6
ipv6-icmp	58	IPv6-ICMP	# ICMP for IPv6
ipv6-nonxt	59	IPv6-NoNxt	# No Next Header for IPv6
ipv6-opts	60	IPv6-Opts	# Destination Options for IPv6

# Configurazione interfaccia

- Come visto in precedenza la sintassi della configurazione dell'indirizzo IPv6 e' simile a quella usata per l'indirizzo IPv4
- Nei nostri esempi utilizzeremo ifconfig:
  - Aggiungere un indirizzo IPv6 ad un'interfaccia:

```
# /sbin/ifconfig <interface> inet6 add ipv6addr/prefix  
# /sbin/ifconfig eth0 inet6 add 2001:760:ffff::126/64
```

- Rimuovere un indirizzo IPv6 da un'interfaccia:

```
# /sbin/ifconfig <interface> inet6 del ipv6addr/prefix  
# /sbin/ifconfig eth0 inet6 del 2001:760:ffff::126/64
```

# Configurazione IPv6 Permanente

- **RPM based** - Fedora/Redhat/CentOS/ScientificLinux
  - Abilitare IPv6
    - `/etc/sysconfig/network`
      - NETWORKING\_IPV6=yes
      - IPV6\_DEFAULTGW=2001:760:0:106::242
    - Assegnare l'indirizzo IPv6 alla macchina:
      - Vi `/etc/sysconfig/network-scripts/ifcfg-eth0`
        - IPV6ADDR=2001:760:0:159::212
        - IPV6INIT=yes
    - Default Gateway:
      - `/etc/sysconfig/network` file: IPV6\_DEFAULTGW=IPv6 address
    - Default route configuration:
      - `/etc/sysconfig/static-routes-ipv6` file:
        - eth0 2001::/3 3ffe:ffff:1234:0002:0:0:0:1
  - **DPKG based** – Debian/Ubuntu (<http://wiki.debian.org/DebianIPv6>)
    - Assegnare indirizzo IPv6, gw
      - Vi `/etc/network/interfaces`
        - auto eth0
        - iface eth0 inet6 static
          - address 2001:760:0:159::131
          - netmask 64
          - gateway 2001:760:0:159::242

# Strumenti per troubleshooting di rete

Il ping6 ed il traceroute6:

```
andrea@andrea-desktop:~$ ping6 ipv6.google.com
PING ipv6.google.com(fx-in-x68.google.com) 56 data bytes
64 bytes from fx-in-x68.google.com: icmp_seq=1 ttl=57 time=21.7 ms
64 bytes from fx-in-x68.google.com: icmp_seq=2 ttl=57 time=19.6 ms

--- ipv6.google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 19.618/20.678/21.738/1.060 ms
andrea@andrea-desktop:~$ traceroute6 ipv6.google.com
traceroute to ipv6.google.com (2001:4860:a003::68), 30 hops max, 40 byte packets
 1 2001:760::11 (2001:760::11)  2.428 ms  2.536 ms  2.640 ms
 2 ru-dir-rt-rm2.rm2.garr.net (2001:760:ffff:100::1)  0.931 ms  0.938 ms  0.947 ms
 3 rt-mi2-rt-rm2.rm2.garr.net (2001:760:ffff:ffff::5)  10.068 ms  10.136 ms  10.149 ms
 4 2001:760:ffff:104::9 (2001:760:ffff:104::9)  9.989 ms  2001:760:ffff:104::7 (2001:760:ffff:104::7)
 5 * * *
 6 * * *
 7 fx-in-x68.google.com (2001:4860:a003::68)  19.950 ms  19.791 ms  19.784 ms
```

# Strumenti per troubleshooting di rete

- Usando indirizzi di tipo link-local per un ping IPv6, il kernel non sa attraverso quale dispositivo (fisico o virtuale) deve inviare il pacchetto in quanto ogni dispositivo ha un indirizzo di tipo link-local. Il tentativo darà come risultato il seguente messaggio di errore:

```
andrea@andrea-desktop:~$ ping6 fe80::216:35ff:fe79:41ed  
connect: Invalid argument
```

- In questo caso bisogna specificare anche l'interfaccia come mostrato qui:

```
andrea@andrea-desktop:~$ ping6 -I eth1 -c 1 fe80::216:35ff:fe79:41ed  
PING fe80::216:35ff:fe79:41ed(fe80::216:35ff:fe79:41ed) from fe80::216:35ff:fe79:41ed eth1: 56 data bytes  
64 bytes from fe80::216:35ff:fe79:41ed: icmp_seq=1 ttl=64 time=0.032 ms  
  
--- fe80::216:35ff:fe79:41ed ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 0.032/0.032/0.032/0.000 ms
```



# Strumenti per troubleshooting di rete

## Ping6 verso indirizzi multicast

- Un meccanismo interessante per rilevare host attivi IPv6 su di un link consiste nell'effettuare ping6 verso l'indirizzo multicast link-local di tutti i nodi:

```
andrea@andrea-desktop:~$ ping6 -I eth1 ff02::1
PING ff02::1(ff02::1) from fe80::216:35ff:fe79:41ed eth1: 56 data bytes
64 bytes from fe80::216:35ff:fe79:41ed: icmp_seq=1 ttl=64 time=0.036 ms
64 bytes from fe80::21a:a0ff:fe51:e4ee: icmp_seq=1 ttl=64 time=0.192 ms (DUP!)
64 bytes from fe80::216:35ff:fe79:b9a9: icmp_seq=1 ttl=64 time=0.232 ms (DUP!)
64 bytes from fe80::202:a5ff:fe20:12d0: icmp_seq=1 ttl=64 time=0.238 ms (DUP!)
64 bytes from fe80::20c:29ff:fe5a:f5e7: icmp_seq=1 ttl=64 time=0.355 ms (DUP!)
64 bytes from fe80::212:79ff:fe67:6935: icmp_seq=1 ttl=64 time=0.442 ms (DUP!)
64 bytes from fe80::21a:a0ff:fe51:9a9a: icmp_seq=1 ttl=64 time=0.540 ms (DUP!)
64 bytes from fe80::280:7dff:fe98:c805: icmp_seq=1 ttl=64 time=0.599 ms (DUP!)
64 bytes from fe80::21a:a0ff:fe51:e8f1: icmp_seq=1 ttl=64 time=0.622 ms (DUP!)
64 bytes from fe80::213:60ff:fe21:bd89: icmp_seq=1 ttl=64 time=0.709 ms (DUP!)
64 bytes from fe80::20f:20ff:fef9:74dd: icmp_seq=1 ttl=64 time=0.732 ms (DUP!)
```

# Strumenti per troubleshooting di rete

## Controllo DNS

Nell'ottica di una connessione ad internet dovremmo verificare se il nostro DNS e' in grado di gestire la risoluzione dei nomi in IPv6, lo facciamo cosi` :

```
andrea@andrea-desktop:~$ host -t AAAA ns1.garr.net
ns1.garr.net has IPv6 address 2001:760:ffff:ffff::aa
```

Con lo stesso metodo si puo` controllare se un sito e` raggiungibile in IPv6:

```
andrea@andrea-desktop:~$ host -t AAAA ipv6.google.com
ipv6.google.com is an alias for ipv6.1.google.com.
ipv6.1.google.com has IPv6 address 2001:4860:a003::68
```

# Strumenti per troubleshooting di rete

## Visualizzare gli instradamenti IPv6

- Per controllare quali e quante rotte IPv6 sono già configurate si può usare:

route -6 (oppure -A inet6):

```
andrea@andrea-desktop:~$ route -6
Kernel IPv6 routing table
Destination          Next Hop              Flag Met Ref Use If
::1/128              ::                    Un   0   1  113 lo
2001:760::216:35ff:fe79:41ed/128 ::                    Un   0   1  1935 lo
2001:760::/64        ::                    UA   256 0 26692 eth1
fe80::216:35ff:fe79:41ed/128 ::                    Un   0   1  6088 lo
fe80::/64            ::                    U    256 0   0 eth1
fec0::1:2a0:24ff:fe77:4997/128 ::                    Un   0   1   0 lo
fec0:0:0:1::/64     ::                    U    256 0   0 eth1
ff00::/8            ::                    U    256 0  12 eth1
::/0                 fe80::213:7fff:fe74:91c9 UGDae 1024 0  8150 eth1
::/0                 ::                    !n   -1  1   5 lo
```

- In maniera del tutto analoga ad IPv4 possiamo effettuare varie operazioni sulle routing table, vediamo come...

# Strumenti per troubleshooting di rete

## Modificare gli instradamenti IPv6

- Potrebbe essere necessario configurare un instradamento per raggiungere l'esterno utilizzando un router abilitato IPv6 sul proprio link (gateway), ecco come:

```
root@andrea-desktop:/home/andrea# route -6 add default gw 2001:760::11
root@andrea-desktop:/home/andrea# route -6
Kernel IPv6 routing table
```

Destination	Next Hop	Flag	Met	Ref	Use	If
::1/128	::	Un	0	1	113	lo
2001:760::216:35ff:fe79:41ed/128	::	Un	0	1	1995	lo
2001:760::/64	::	UA	256	0	26712	eth1
fe80::216:35ff:fe79:41ed/128	::	Un	0	1	6282	lo
fe80::/64	::	U	256	0	0	eth1
fec0::1:2a0:24ff:fe77:4997/128	::	Un	0	1	0	lo
fec0:0:0:1::/64	::	U	256	0	0	eth1
ff00::/8	::	U	256	0	12	eth1
::/0	2001:760::11	UG	1	0	0	eth1
::/0	fe80::213:7fff:fe74:91c9	UGDAe	1024	0	8215	eth1
::/0	::	!n	-1	1	6	lo

# Strumenti per troubleshooting di rete

## Modificare gli instradamenti IPv6

- Potrebbe essere necessario anche configurare un instradamento per raggiungere una data rete passando per un'interfaccia specifica, lo facciamo così:

```
root@andrea-desktop:/home/andrea# route -6 add 2002:760::/64 dev eth0
root@andrea-desktop:/home/andrea# route -6
```

Kernel IPv6 routing table

Destination	Next Hop	Flag	Met	Ref	Use	If
::1/128	::	Un	0	1	113	lo
2001:760::216:35ff:fe79:41ed/128	::	Un	0	1	1995	lo
2001:760::/64	::	UA	256	0	26727	eth1
2002:760::/64	::	U	1	0	0	eth0
fe80::216:35ff:fe79:41ed/128	::	Un	0	1	6282	lo
fe80::/64	::	U	256	0	0	eth1
fec0::1:2a0:24ff:fe77:4997/128	::	Un	0	1	0	lo
fec0:0:0:1::/64	::	U	256	0	0	eth1
ff00::/8	::	U	256	0	12	eth1
::/0	fe80::213:7fff:fe74:91c9	UGDAe	1024	0	8215	eth1
::/0	::	!n	-1	1	6	lo

# Strumenti per troubleshooting di rete

## Visualizzare e manipolare la tabella dei neighbor

Con il seguente comando si possono visualizzare i vicini IPv6 memorizzati nell'apposita tabella (scoperti o impostati manualmente):

```
root@andrea-desktop:/home/andrea# ip -6 neigh show  
fc80::213:7fff:fc74:91c9 dev eth1 lladdr 00:13:7f:74:91:c9 router STALE
```

Con il seguente comando invece si puo' rimuovere manualmente una voce:

```
root@andrea-desktop:/home/andrea# ip -6 neigh del fec0::1 lladdr 02:01:02:03:04:05 dev eth1
```

# Tunnelling

## Configurare un Tunnel

Perche' dovremmo aver bisogno di configurare un tunnel?

E' possibile che l'infrastruttura che utilizziamo per l'accesso non supporti nativamente il protocollo IPv6 (le cause possono essere un'obsolescenza hardware o software).

Anche in questo caso e' possibile ottenere connettività IPv6 attraverso diversi metodi che possono essere considerati in base a quale livello e' coinvolto nella modifica:

- Implementati sugli host:
  - Host Dual Stack (host con 2 schede di rete: 1 x IPv4, 1 x IPv6)
  - Altri: BIS, BIA, ...
- Implementati a livello di rete:
  - Tunnel (Manuali, 6to4, automatici, Broker, ISATAP, Teredo)
  - Rete Dual Stack
- Basati su traduttori di protocollo:
  - Basati su traduttori di protocollo
  - Altri: TRT, ...

# Tunnelling

## Configurare un Tunnel

- Oggi parleremo di tunnel e della loro configurazione
- Tunnel: sono usati normalmente per trasportare pacchetti di un protocollo in una rete basata su un protocollo diverso
- Ad esempio IPv6-in-IPv4:  
Tecnicamente il pacchetto IPv6 viene incapsulato in un pacchetto IPv4
- I metodi di configurazione proposti di seguito fanno uso del comando `ifconfig` presente su tutte le distribuzioni del sistema operativo Linux. E' possibile ottenere gli stessi risultati usando il pacchetto `iproute2`, la cui presenza pero' non e' garantita su tutte le distribuzioni



# Tunnelling

## Configurare un Tunnel

- Configurazione di una Linuxbox utente:
  - 1 **ifconfig sit0 up**
  - 2 **ifconfig sit0 tunnel ::192.168.1.1**
  - 3 **ifconfig sit1 up**
  - 4 **ifconfig sit1 add 2001:760::2/126**
  - 5 **route -A inet6 add ::/0 gw 2001:760::1 dev sit1**
- La configurazione di un tunnel ipv6-in-ipv4 su Linux prevede l'uso di due interfacce
- sit0: usata per definire l'end-point IPv4 a cui saranno inviati i pacchetti incapsulati;
- sit1: usata per definire l'end-point IPv6 il cui traffico sarà veicolato attraverso il tunnel sit0.
- Alla riga 1 viene definita l'interfaccia tunnel sit0, specificando alla riga 2 che l'end-point IPv4 remoto ha indirizzo 192.168.1.1.
- Alla riga 3 si definisce l'interfaccia sit1 assegnandole un indirizzo IPv6 alla riga 4.
- In fine alla riga 5 viene impostata una default route verso la rete GARR raggiungibile mediante l'interfaccia sit1.

# Tunnelling

## Configurare un Tunnel

### Esempio di Tunnel tra un router Cisco e un host Linux

La Configurazione dell'interfaccia Tunnel del Router si presenta così:

```
Temporaneo6#sh run int tu0
Building configuration...

Current configuration : 212 bytes
!
interface Tunnel0
 description Tunnel IPv6 verso GARR
 no ip address
 ipv6 address 2001:760:FFFF:10C::55/127
 ipv6 enable
 tunnel source 192.168.100.1
 tunnel destination 192.168.100.2
 tunnel mode ipv6ip
end
```

# Tunnelling

## Configurare un Tunnel

### Esempio di Tunnel tra un router Cisco e un host Linux

La configurazione dell'host Linux invece e' così:

```
noc@noc-laptop:~$ sudo ifconfig sit0 up
noc@noc-laptop:~$ sudo ifconfig sit0 tunnel ::192.168.100.1
noc@noc-laptop:~$ sudo ifconfig sit1 up
```

La rete IPv6 e' configurata così:

```
noc@noc-laptop:~$ sudo ifconfig sit1 2001:760:FFFF:10C::54/127
```

Instradamento IPv6:

```
noc@noc-laptop:~$ sudo route -A inet6 add ::/0 gw 2001:760:FFFF:10C::55 dev sit1
```

# Tunnelling

## Configurare un Tunnel

### Esempio di Tunnel tra un router Cisco e un host Linux

- La condizione principale perche' funzioni il Tunnel e' che gli end-point IPv4 siano reciprocamente pingabili:

Dal Cisco:

```
Temporaneo6>ping 192.168.100.2  
  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 192.168.100.2, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

Dall'host Linux:

```
noc@noc-laptop:~$ ping 192.168.100.1  
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.  
64 bytes from 192.168.100.1: icmp_seq=1 ttl=255 time=6.65 ms  
64 bytes from 192.168.100.1: icmp_seq=2 ttl=255 time=0.935 ms
```

# Tunneling

## Configurare un Tunnel

Questo l'output della tabella di routing e del Tunnel su Linux

```
noc@noc-laptop:~$ route
Kernel IP routing table
Destination        Gateway           Genmask          Flags Metric Ref    Use Iface
192.168.100.0      *                255.255.255.0   U        0      0      0 eth0
noc@noc-laptop:~$ route -6
Kernel IPv6 routing table
Destination        Next Hop         Flag Met  Ref  Use If
::1/128           ::              Un   0   1    0 lo
::127.0.0.1/128   ::              Un   0   1    0 lo
::192.168.100.2/128 ::              Un   0   1    0 lo
::/96             ::              Un  256  0    0 sit0
2001:760:ffff:10c::54/128 ::              Un   0   1    0 lo
2001:760:ffff:10c::54/127 ::              Un  256  0    1 sit1
fe80::c0a8:6402/128 ::              Un   0   1    0 lo
fe80::21c:23ff:fe13:330c/128 ::              Un   0   1    0 lo
fe80::/64         ::              U    256  0    0 eth0
fe80::/64         ::              Un  256  0    0 sit1
ff00::/8          ::              U    256  0    0 eth0
ff00::/8          ::              U    256  0    0 sit1
::/0              2001:760:ffff:10c::55 UG   1   0    0 sit1
::/0              fe80::223:33ff:fe81:1fae UGDAe 1024 0    0 eth0
::/0              ::              !n   -1  1    1 lo
noc@noc-laptop:~$ ip tunnel
sit0: ipv6/ip remote any local any ttl 64 nopmtudisc
sit1: ipv6/ip remote 192.168.100.1 local any ttl 64
```

# Tunnelling

## Configurare un Tunnel

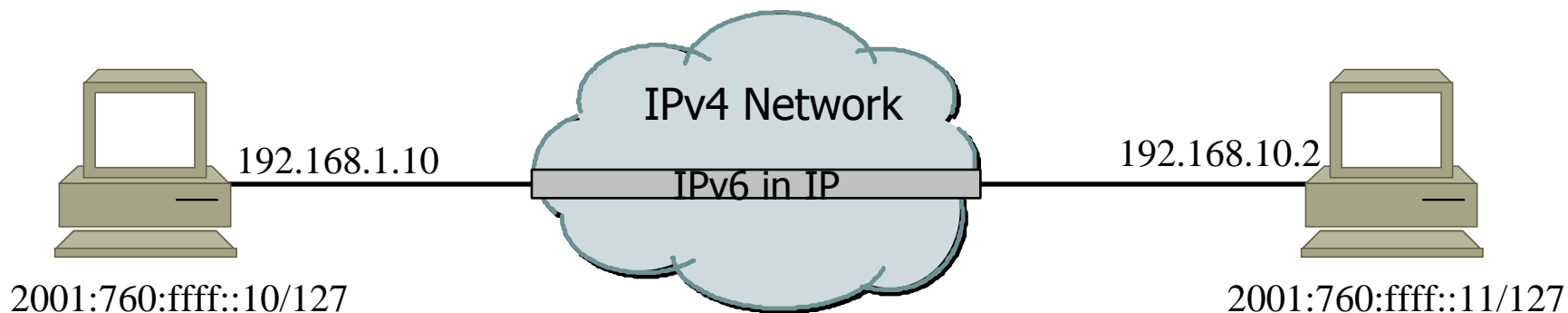
### Esempio di Tunnel tra un router Cisco e un host Linux

Il Tunnel e' funzionante:

```
noc@noc-laptop:~$ ping6 2001:760:FFFF:10C::55
PING 2001:760:FFFF:10C::55(2001:760:ffff:10c::55) 56 data bytes
64 bytes from 2001:760:ffff:10c::55: icmp_seq=1 ttl=64 time=3.87 ms
64 bytes from 2001:760:ffff:10c::55: icmp_seq=2 ttl=64 time=1.37 ms
64 bytes from 2001:760:ffff:10c::55: icmp_seq=3 ttl=64 time=1.32 ms
64 bytes from 2001:760:ffff:10c::55: icmp_seq=4 ttl=64 time=1.28 ms
64 bytes from 2001:760:ffff:10c::55: icmp_seq=5 ttl=64 time=1.35 ms
```

# Configurazione di un tunnel

Ipotesi: si vuole realizzare un collegamento punto-punto in IPv6 usando come trasporto IPv4



# Configurazione di un tunnel

Per la configurazione utilizziamo il comando *iptunnel*, questa utility consente la creazione, cancellazione, e modifica di tunnel ip-over-ip, gre, sit.

```
iptunnel {add|change|del|show} NAME mode  
{ipip|gre|sit} remote <endpointaddr> local  
<localaddr> [ ttl TTL ] [ tos TOS ] [ nopmtudisc ]  
[ dev PHYS_DEV ]
```



# Configurazione di un tunnel

Configurazione passo-passo:

- Configurazione del tunnel e creazione dell'interfaccia tunnel;  
`#iptunnel add sit1 remote 192.168.10.2 local 192.168.1.10 mode sit ttl 64`
- Configurazione dell'indirizzo IPv6 sull'interfaccia primaria;  
`#ifconfig sit1 inet6 add 2001:760:ffff::10/127`
- Attivazione dell'interfaccia tunnel  
`#ifconfig sit1 up`
- Creazione della statica verso il tunnel  
`#route --inet6 add default gw 2001:760:ffff::11`

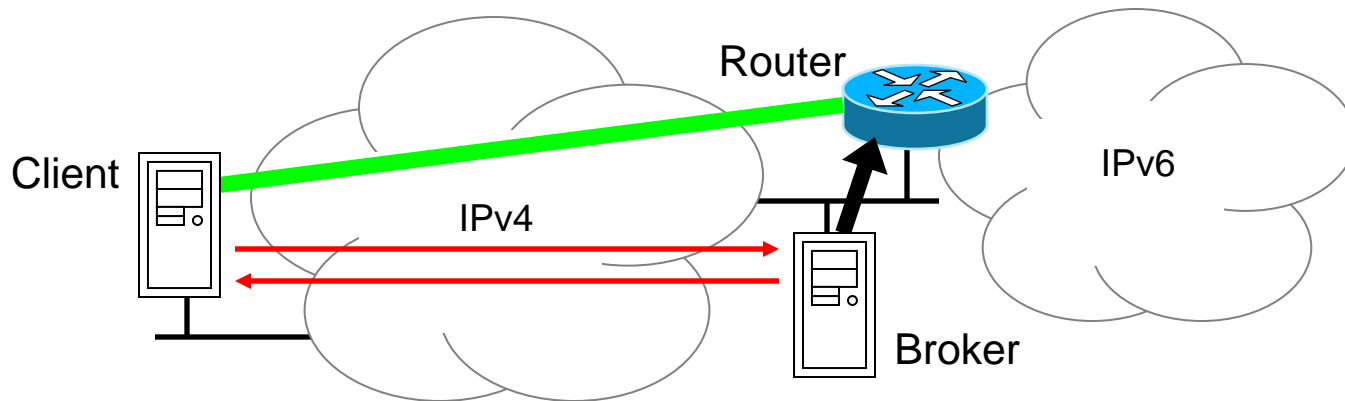
Con l'applicazione di questa rotta il traffico IPv6 viene routato, di default, attraverso il tunnel.

# Configurazione di un tunnel

```
#route --inet6
::/0          2001:760:fff::11      UG      1
0            0 sit1

#iptunnel show
sit1: ipv6/ip  remote 192.168.10.2  local any  ttl
64
```

# Tunnel Broker



- Applicazione web raggiungibile tramite IPv4
- Crea dinamicamente i tunnel configurati
  - Un utente che desidera stabilire un tunnel lo richiede al broker attraverso una pagina web
  - Il broker identifica l'utente
  - Il broker configura un router per creare il tunnel verso l'utente e ne comunica i parametri
- Molto utilizzati per utenti "occasionalisti"
- Esempio di tunnel Broker: Hurricane Electric

# Attivazione IPv6 in GARR

La procedura di attivazione si svolge secondo i seguenti passi:

L'APM invia una mail di richiesta a [noc@garr.it](mailto:noc@garr.it),

Il NOC chiede al gruppo LIR la riservazione di:

- una rete punto-punto (/126)

- una rete per lo spazio di indirizzamento utente (/48);

Il NOC invia una mail all'APM in cui comunica:

- Indirizzo sulla punto-punto

- Spazio di indirizzamento utente

- PoP di attestazione

Il NOC concorda con l'APM una data di attivazione

# Attivazione IPv6 in GARR

Una volta completata l'attivazione del servizio IPv6, è compito dell'APM inoltrare, a LIR, la richiesta di delega della zona di reverse DNS per la propria rete /48, inviando una mail a:

[dns-staff@garr.it](mailto:dns-staff@garr.it)

La mail, oltre contenere la richiesta, indicherà i server DNS autoritativi per rispondere alle query di reverse.

Il gruppo LIR provvederà a:

Configurare la delega della rete /48 ai Name Server dell'ente

# Riferimenti

- <http://www.bieringer.de/linux/IPv6/>
- <http://www.hs247.com>
- <http://linux-ipv6.org>
- <http://www.ipv6.ernet.in>
- <http://www.iitk.ernet.in/ipv6>
  - IPv6 Forum
    - <http://www.ipv6forum.com/>
  - Merit Networks Network Research and Technology
    - <http://www.merit.edu/nrd/projects/ipv6.html>
  - The NLANR Active Measurement Project
    - <http://watt.nlanr.net/IPv6/>
  - Moonv6
    - <http://www.moonv6.org/>
  - Internet Society
    - <http://www.isoc.org/>
- **IETF:**
  - IPv6 Working Group:
    - <http://www.ietf.org/html.charters/ipv6-charter.html>