El Protocolo Ipv6
Una Introducción

Horacio Castellini.

Motivos de un nuevo protocolo

- Agotamiento del espacio de direciones disponibles en el protocolo Ipv4 (Se estima que para el 2012 llegará al agotamiento, ya lo predijo Nostradamus)
- Grandes tablas de enrutamiento en troncales Ipv4.
- Mejorar la calidad en el transporte de los nuevos protocolos en tiempo real como VoIP.

Motivos de un nuevo protocolo

IPv4 posibilita 4.294.967.296 direcciones de red diferentes, un número inadecuado para dar una dirección a cada persona del planeta, y mucho menos a cada vehículo, teléfono, PDA, etcétera. En cambio, IPv6 admite 340.282.366.920.938.463.463.374.607.431.768.211.456 de direcciones (670 mil billones de direcciones/mm^2)

Otra vía para la popularización del protocolo es la adopción de este por parte de instituciones. El gobierno de los Estados Unidos ordenó el despliegue de IPv6 por todas sus agencias federales en el año 2008.

Ventajas del IPv6

- **Escalabilidad**: Ipv6 tiene un campo de direcciones de 128 bits frente a los 32 bits de Ipv4.
- **Seguridad**: Ipv6 incluye seguridad en sus especificaciones como ser la encriptación y la autenticación (IPsec).
- **Aplicaciones a tiempo real**: Ipv6 incluye campos para etiquetar los flujos de datos, con esto los router pueden establecer prioridades (QoS).

Ventajas del IPv6

- **Plug and Play**: IPv6 incluye en su mecanismo el *plug and play* lo cual facilita a los usuarios la conección con los equipos de la red.
- **Direccionamiento estructurado**: La formación de una dirección sigue criterios establecidos.

Por qué Ipv6 y no IPv5?

Se decidió asignar como IPv5 (RFC 1819) al obsoleto protocolo experimental llamado ST-II ó *stream protocol version 2*. Diseñado para el control de flujo de datos. Este fue abandonado por la IETF en 1995. Por eso el siguiente protocolo debía llamarse IPv6.

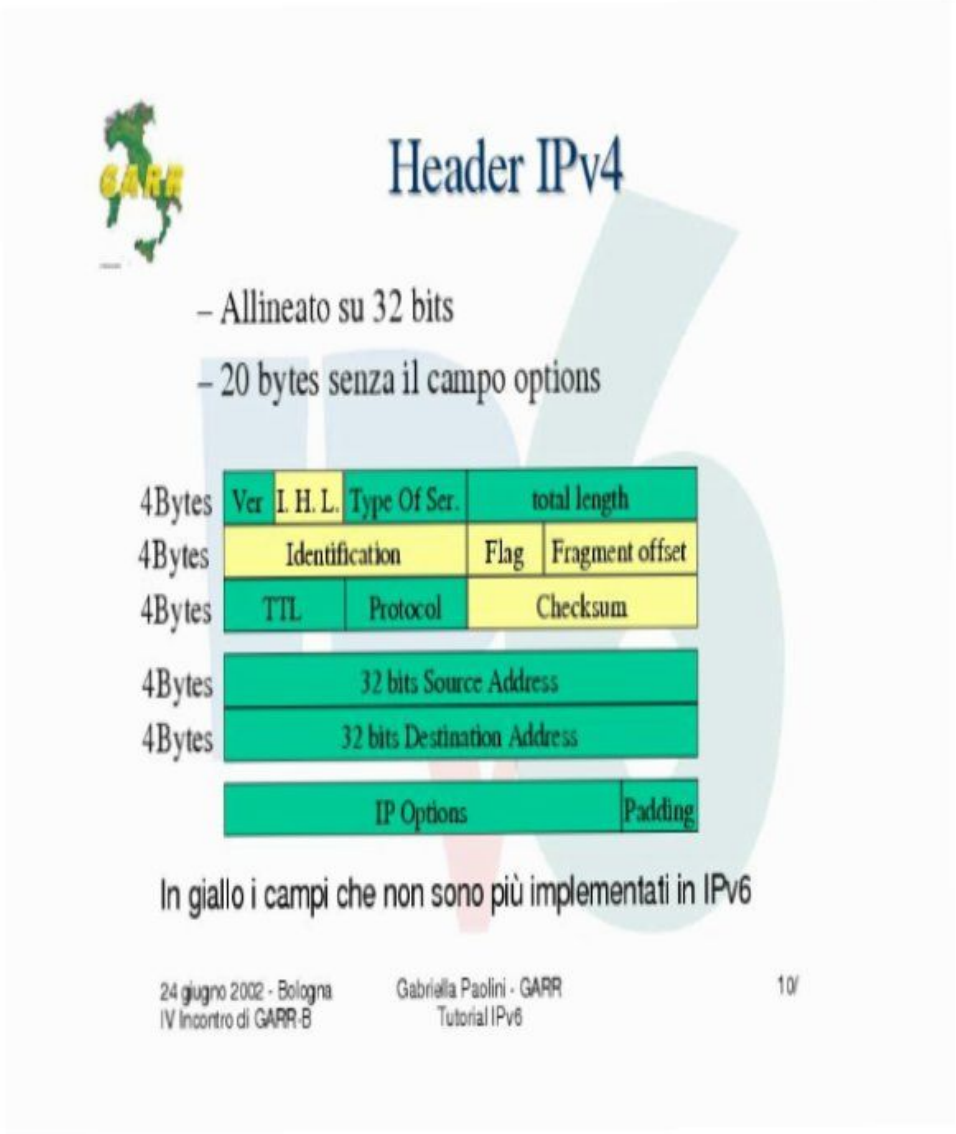
Terminología IPv6

- **Nodo**: Cualquier dispositivo que implementa Ipv6.
- **Router**: Un nodo que re-envía paquetes de datos no explícitamente direccionados por él.
- **Host**: Cualquier nodo que no sea un router.
- **Capa superior**: Un protocolo superior al Ipv6 en la torre OSI. (TCP, UDP, etc.)

Terminología IPv6

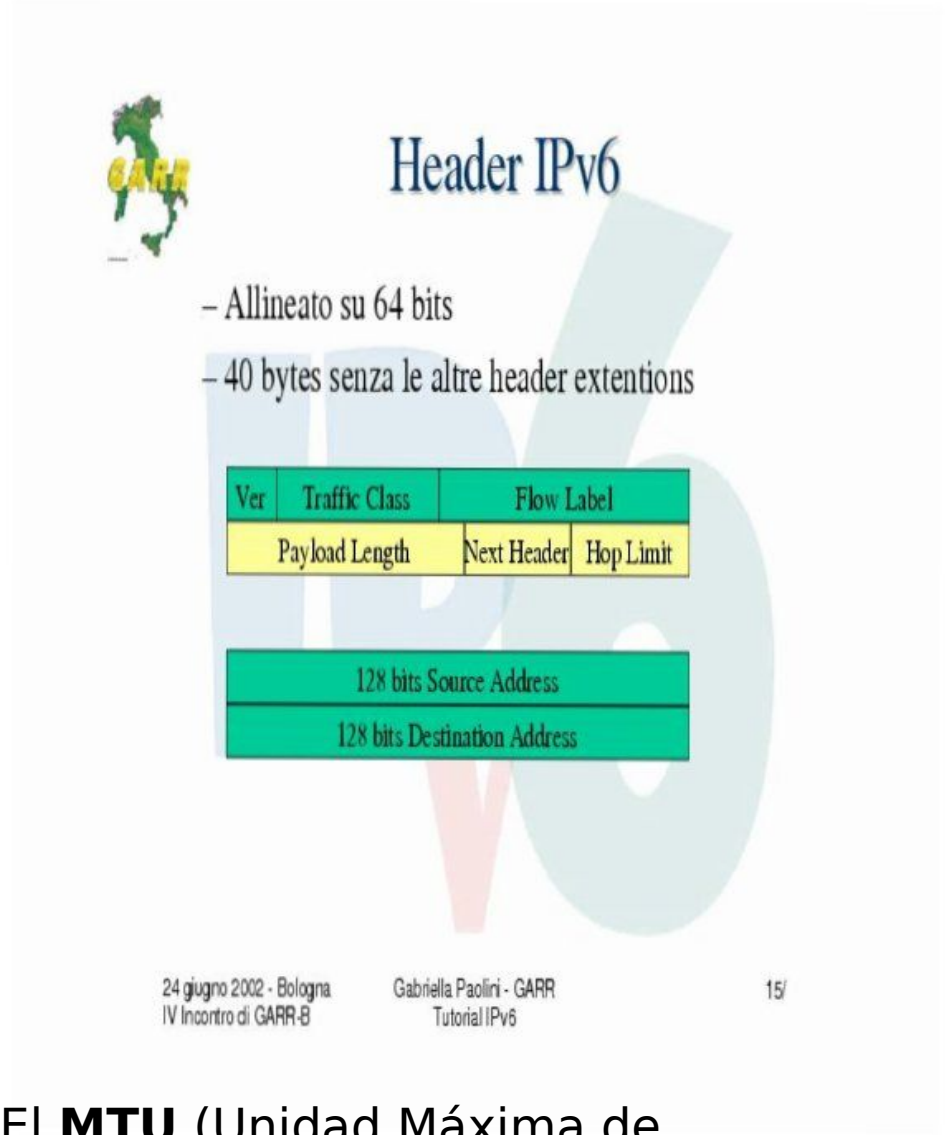
- **Enlace** o *link*: Una facilidad o medio sobre el cual los nodos pueden comunicarse, es una capa inferior al Ipv6 según la torre OSI. (Ethernet, ATM, etc.)
- **Vecinos**: Nodos anexados al mismo enlace.
- **Dirección**: Un identificador de la capa 3 del la torre OSI para identificar una interfaz o conjunto de interfaces.

Cabecera Ipv4 (RFC 791)



- Ver Versión (4 bits)
- I.H.L. Cabecera (4 bits)
- T.O.S. Tipo de servicio (1 byte)
- Total Lenght Longitud total (2 bytes)
- Identificación (2 bytes)
- Flag Identificador (4 bits)
- Fragment Offset Desplazamiento de Fragmentación (12 bits)
- TTL tiempo de vida (1 byte)
- Protocol Protocolo (1 byte)
- Checksum Verificación (2 bytes)

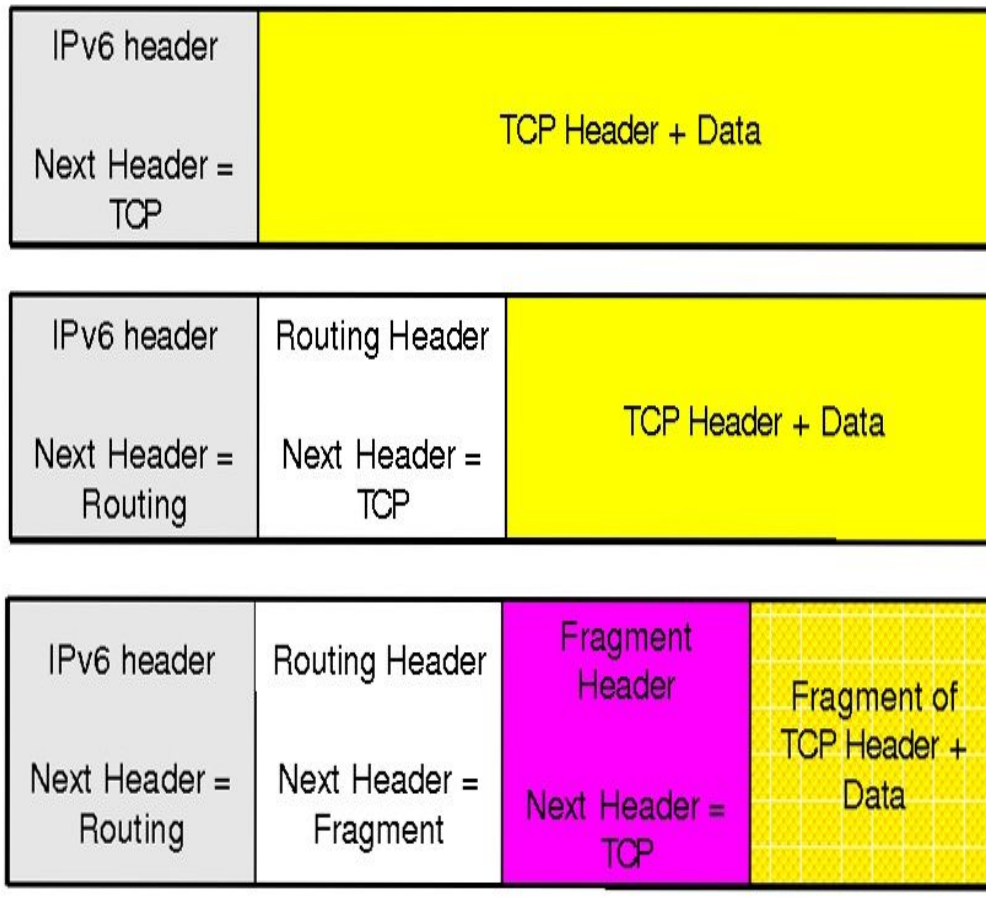
Cabecera IPv6 (RFC 2460)



- Ver Versión (4 bits)
- Traffic Class TOS en ipv4 (1 byte)
- Flow Label Permite tráficos con requisitos de tiempo real (20 bits)
- Payload lenght longitud toal de datos (2 bytes)
- Hop Limit TTL de ipv4 (1 byte)
- Next Header (próximo protocolo) (1 byte)

El **MTU** (Unidad Máxima de Transmisión) debe ser como mínimo de 1280 bits

Next Header, encapsulación de cabeceras

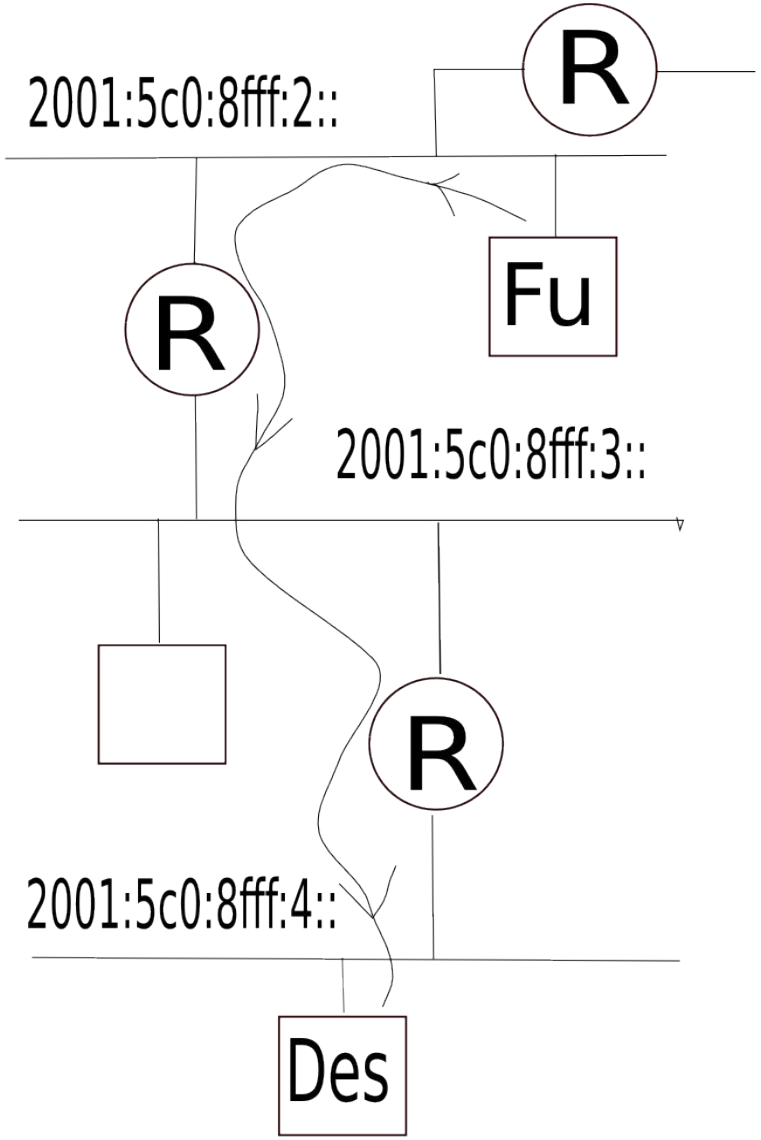


- 00 Opción Salto por Salto
- 43 Enrutado
- 44 Fragmentado
- 50 Seguridad en datos
- 51 Autenticado
- 58 ICMP ipv6
- 59 Fin de transmición
- 60 Opciones de Destino
- xx Protocolo capa superior

Modelo de direccionamiento (RFC 2373)

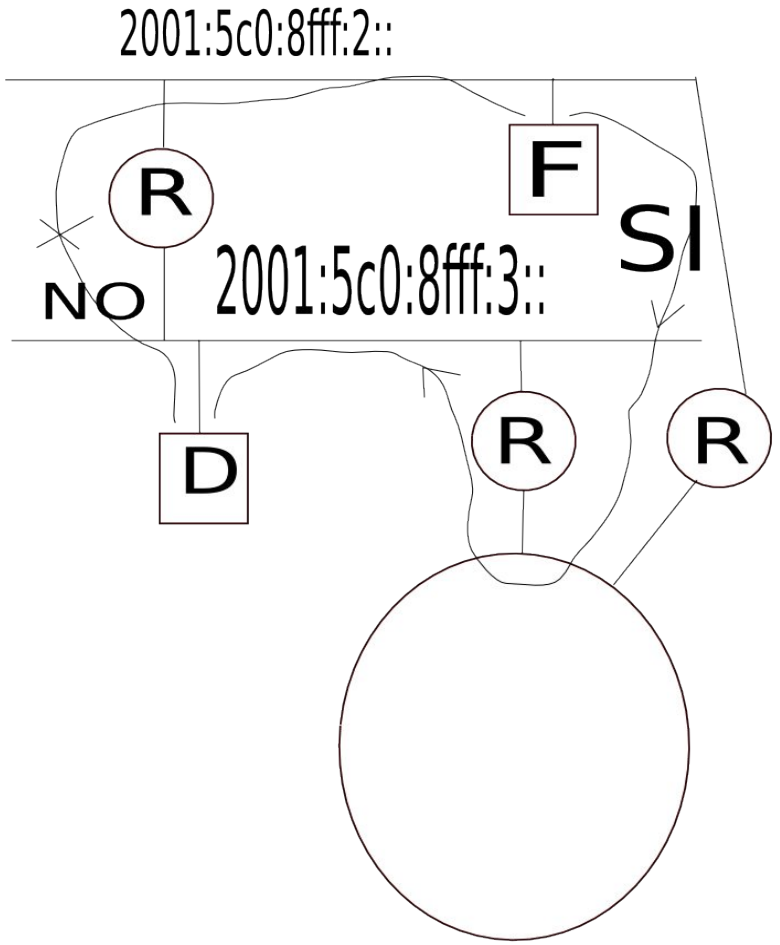
- Una dirección Ipv6 de cualquier tipo está asociada a una interfaz y no a un nodo.
- *Unicast* Identificador para una única interfaz. Es similar a la dirección IPv4.
- *Anycast* Identificador para un conjunto de interfaces pertenecientes o no al mismo nodo. Se entrega a la interfaz de tiempo menor.
- *Multicast* Identificador para un conjunto de nodos, similar al broadcast.

Tipos de direccionamiento



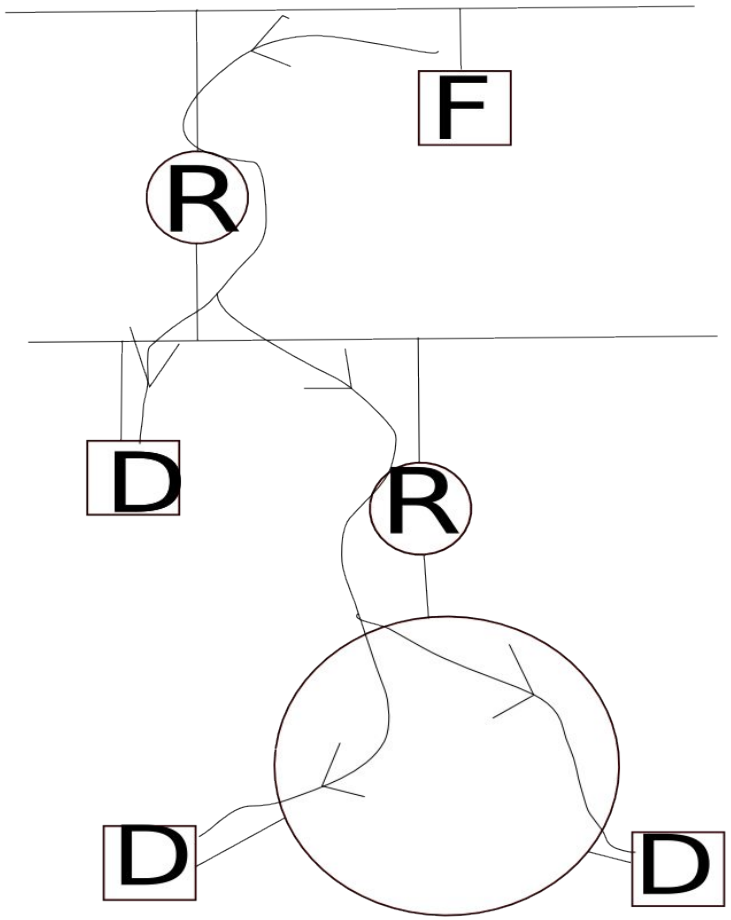
- **Unicast**: Un identificador para cada interfaz. Un paquete enviado a una dirección unicast es remitido al identificador de interfaz asociado a dicha dirección.

Tipos de direccionamiento



- **Anycast**:Un identificador para un conjunto de interfaces. Un paquete enviado a una dirección anycast es remitido al identificador mas cercano (tiempo) asociado a dicha dirección.

Tipos de direccionamiento



- **Multicast**: Un identificador para un conjunto de interfaces. Un paquete enviado a una dirección multicast es remitido a todas las interfaces asociadas con dicha dirección.

Tipos de direccionamiento

- Nota: No existe direccionamiento *boadcast* en ipv6.
- Los campos de direcciones reciben nombres específicos denominados **prefijos**.
- Los prefijos permiten saber la ruta de encaminamiento.
- Todas las interfaces han de tener al menos una dirección unicast.
- Al igual que en Ipv4, se asocia un prefijo de subred.

Representación de direcciones

- La forma canónica es xx:xx:xx:xx:xx:xx:xx:xx, donde xx es una cantidad hexadecimal de dos dígitos, por ejemplo: 3ffe:0b90:0003:0f17:0000:0000:000:0002
- Los ceros consecutivos se los puede resumir con “::” una sola vez, los ceros a la izquierda se los puede omitir. El ejemplo anterior queda: 3ffe:b90:3:f17::2

Representación de direcciones

- La dirección unicast ::1 es llamada *loopback* y es interna al nodo. Y no puede usarse como dirección de destino fuera del nodo.
- la dirección :: especifica aucencia de dirección, y no puede usarse como dirección de destino, se la usa para conocer la propia dirección del host.

Representación de direcciones

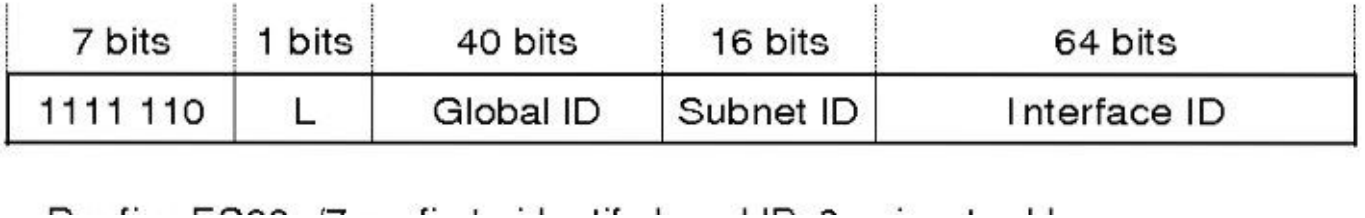
- **Túneles dinámicos** IPv6 sobre IPv4 ::ddd.ddd.ddd.ddd permite el tráfico de IPv6 compatibles con IPv4, por ejemplo ::177.15.20.12
- **Representación automática** de IPv4 sobre IPv6, :ffff:ddd.ddd.ddd.ddd. Permite que los nodos que solo soportan IPv4 operen en IPv6. ::ffff:129.14.52.88

Representación de direcciones

Las direcciones de enlace local o *link-local* son direciones unicast usadas para direccionar sobre un enlace para descubrir vecinos cuando no hay routes. Los routers no pueden remitir estos paquetes. El formato es **fe80::ID** donde **ID** es el identificador de la interfaz de 64 bits. Antepuesto de 54 bits a cero.

Por ejemplo: fe80::21f:d0ff:feb2:cf51/64

Representación de direcciones



7 bits	1 bits	40 bits	16 bits	64 bits
1111 110	L	Global ID	Subnet ID	Interface ID

Las direciones de sitio o *site-local* son direcciones unicast usadas para direccionar una subred interna sin usar prefijos globales, los routers no pueden remitirlas. Su formato es **fc00::/7** L=1 para indicar identificadores locales, L=0 no locales.

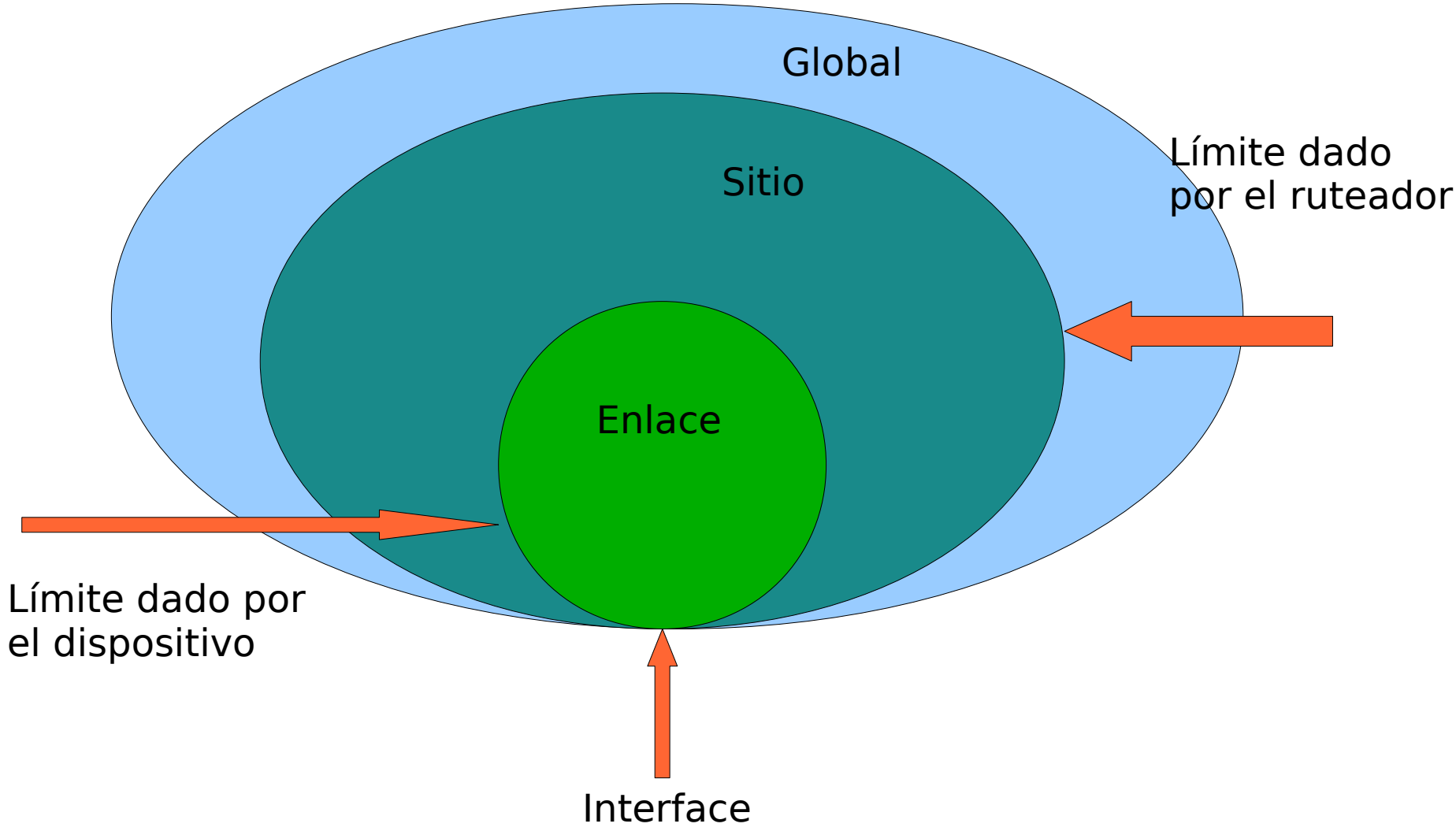
Tambien se la conoce como ULA *Unique Local Adress*.

3ffe:b90:3::2 (globla) ---> fc00:b90:3::2 (ULA)

Representación de Direcciones

- Una dirección unicast *global* con prefijo se la especifica como ipv6-dir/prefijo donde prefijo es la cantidad de bits más significativos (a la izquierda) que lo forman. Es la única remitida por los ruteadores.
- Por ejemplo 2001:5c0:8fff:2::3/64 se puede expandir como
- **red** 2001:5c0:8fff:2::
- **Prefijo** de red ffff:ffff:ffff:ffff::
- **ID** ::3

Representación de Direcciones



Direcciones globales

- Interface 3ffe:b90:3::2/128 (Unicast)
- Subred 3ffe:b90:3::/48 (Anycast)
- Interface + Subred 3ffe:b90:3::2/48
- En Formato Hexadecimal \ [x3ffe0b900030000000000000002/48] que se usa en los DNS. (RFC 2874)

Prefijos Reservados

Estado	Prefijo en binario	Fraación del Espacio	Ejemplo
Reservado	0000 0000	1/256	00xx::/8
No Asignado	0000 0001	1/256	01xx::/8
Reservado NSAP	0000 001	1/128	02xx::/7; 03xx::/7
Reservado IPX	0000 010	1/128	04xx::/7; 05xx::/7
No Asignado	0000 011	1/128	06xx::/7; 07xx::/7
No Asignado	0000 1	1/128	08xx::/5; ... 0fxx::/5
No Asignado	0001	1/16	1xxx::/4;
Unicast Globales	001	1/8	2xxx::/3; 3xxx::/3
Unicast link	1111 1110 10	1/1024	fe8x::/10; fe9x::/10;
Unicat Sitio	1111 1110 11	1/1024	fecx::/10; fedx::/10;
Multicast	1111 1111		ffxx::/8

Broker Unicast Globales

- Los que empiezan con 2xxx:: y 3XX:: son conocidos como unicast globales.
- 3ffe::/16 Troncales 6Bone (fuera de servicio público desde el 2006).
- 2002::/16 Espacio de direcciones asignado para troncales 6to4 (RFC 2529, no se dará aquí).
- 2001::/16 Dirección delegada a tuneles, en general el Broker es privado y da el servicio gratuito (por ahora), una empresa es http://gogonet.gogo6.com cuya ipv6 es 2001:5c0:1000:f::3

Brokers disponibles

Static= ip estática
RDNS= delegación inversa de nombre
TSP= Tunnel Setup Protocol
BGP= Ruteo BGP (Border Gateway Protocol)
AYIYA=Tunel “Anything In Anything”

Provider	Coverage	Subnet	Protocols	Static	RDNS	BGP	Registration	Configuration	Language
Hurricane Electric[4]	United States, Canada, Europe (6 Countries), Hong Kong, Tokyo	/64 and /48 subnet	6in4	Yes	Yes	Yes	Signup required	Website	English
SixXS	United States, Brazil, Europe (13 countries), New Zealand[5]	/64 tunnel and /48 subnet	6in4, AYIYA [6]	Yes	Yes		RIPE, ARIN, APNIC, LACNIC, AFRINIC, or direct signup[7]	TIC/AICCU, manual, website	English
gogo6/Freenet6[2]	Montreal, Amsterdam, Taipei, Sydney	/56 subnet	6in4, 4in6, TSP	Yes	Yes		Anonymous, or signup (via registration on gogoNET[2])	TSP, website	English

Asia-Pacific

[edit]

Tunnel brokers that provide tunnels to users in the Asia-Pacific region.

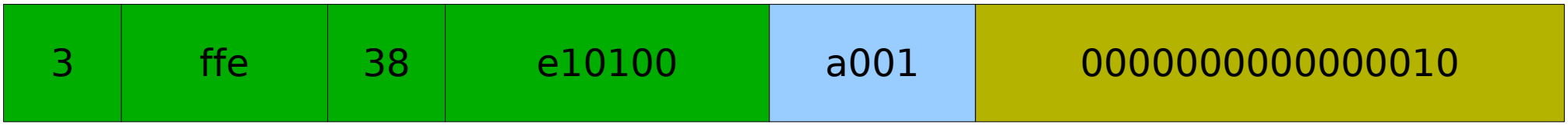
Provider	Coverage	Subnet	Protocols	Static	RDNS	BGP	Registration	Configuration	Language
Aarnet[2]	Australia		6in4, TSP			No	Anonymous, or signup	TSP, website	English
IIJ	Japan		6in4				Signup required		English, Japanese
6fei	China		6in4, 4in6	Yes			Signup required		Chinese
Internode[2]	Australia		6in4, TSP				Anonymous, or signup	TSP, website	English
IPv6Now[2]	Australia	/64, /48	6in4, TSP	Yes	Yes		Signup	TSP	English
Mytbs	Malaysia	/64 subnet	6in4		Yes		Signup	website	English

Unicast Globales ejemplo de estructura agregable

- caspa.ipv6.frlp.utn.edu.ar has IPv6 address 3ffe:38e1:0100:a001::10 (**hasta el 6/6/2006**)
- 3ffe::/16 (prefijo para el 6bone)
- 3ffe:3800::/24 (prefijo para el pTLA FIBERTEL, como la longitud del prefijo es de 24 bits la parte significativa es 3ffe:38xx).
- 3ffe:38e1::/32 (prefijo para el pNLA UTN).
- 3ffe:38e1:0100::/48 (prefijo para UTN-FRLP)
- 3ffe:38e1:0100:a001::/64 (prefijo para la subred a001 dentro de UTN-FRLP).
- 3ffe:38e1:0100:a001::10/128 (dirección IP completa, los últimos 64 bits corresponden al *interface identifier* ::10, antes conocido como número de host).

Unicast Globales Agregables (RFC 2374) (ejemplo anterior)

- FP: Prefijo de formato global (3 bits)
- TLA: Identificador de agregación nivel superior (13 bits)
- Res: Reservado para el gestionador (8 bits) (era FIBERTEL)
- NLA: Identificador agregable del siguiete nivel (24 bits)
- SLA: Identificador agregable del sitio (16 bits)
- ID: Identificador de interfaz (64 bits)
- color identifica la topologías: Públicas y de Sitios.



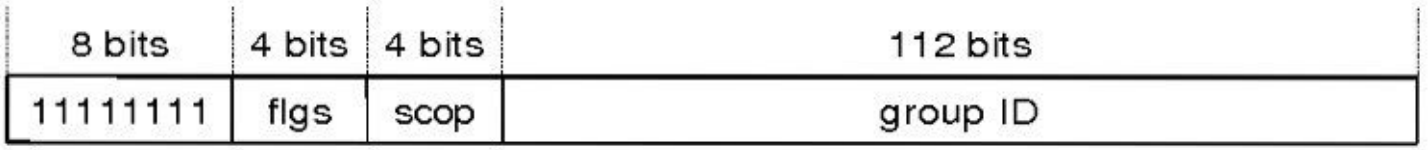
Unicast Globales agregables prefijos

- /3 Registro Global
- /16 6Bone
- /24 Registros globales de Internet
- /48 Topología pública
- /64 subred del sitio + Topología

UNT-FRLP Hoy

- Las direcciones IPv6 tienen 128 bits. La dirección 2001:1318:1001:4802:0000:0000:0000:0010 corresponde a un router IPv6 en la FRLP. Ésta dirección puede descomponerse de la siguiente forma:
- 2001:1318:1001::/48 (prefijo para UTN)
- 2001:1318:1001:4800::/56 (prefijo para UTN-FRLP).
- 2001:1318:1001:4802::/64 (Prefijo asignado a la subred 2 dentro de la UTN FRLP).
- 2001:1318:1001:4802:4802::10 (dirección IP completa, los últimos 64 bits corresponden al interface identifier 10, antes conocido como número de host).

Direcciones multicast (RFC 2375)



- Bandera=$(000T)_2$, si T=0 multicast permanente asignada por la autoridad de numeración global de Internet, T=1 multicast temporal
- Los bits del ámbito (B) pueden tomar los siguientes valores hexadecimales con sentido, B=0 reservado (no sé para que), B=1 ámbito local, B=2 ámbito de enlace, B=5 ámbito de sitio, B=8 ámbito local de organización, B=E ámbito global, B=F reservado.
- Una dirección multicast solo puede aparecer como dirección de destino y nunca de fuente.
- Por ejemplo ff02::101 todos los servers NTP en el enlace.

Direcciones multicast permanentes

- ff01:: nodo-local, los paquetes nunca dejan un nodo.
- ff02:: link-local, los paquetes nunca serán remitidos por un ruteador.
- ff05:: site-local, los paquetes nunca dejan un sitio
- ff08:: organización-local, dificil de implementar
- ff0e:: global

Multicast de sondeo (ID reservados)

- El ID de los host es 0x1 (hexadecimal)
- El ID de los enrutadores es 0x2
- ff02::1 todos los nodos sobre el link
- ff02::2 todos los enrutadores sobre el link
- ff02::5 todos los RIP enrutadores sobre el link
- ff02::1:2 todos los DHCP sobre el link.
- ff02::1:ff00:0000/104 solicitud dirección de nodo. Por ejemplo ff02::1:ff12:415d → 2001:5c0:1000:b::12:415d

Ejemplo de multicast (ICMPv6)

Descubrir los nodos sobre una interfaz.

```
hcaste@azul:~$ ping6 -Ieth1 ff02::1
PING ff02::1(ff02::1) from fe80::21f:d0ff:feb2:cf51 eth1: 56 data bytes
64 bytes from fe80::21f:d0ff:feb2:cf51: icmp_seq=1 ttl=64 time=0.094 ms
64 bytes from fe80::20b:6aff:fe7d:53c: icmp_seq=1 ttl=64 time=1.59 ms
(DUP!)
```

Hay dos computadoras.
fe80::21f:d0ff:feb2:cf51 es la direción link-local de la interfaz eth1 de azul
fe80::20b:6aff:fe7d:53c es la direción link-local de la interfaz eth1 de clara

Descubrir los enrutadores sobre una interfaz.

```
hcaste@azul:~$ ping6 -Ieth1 ff02::2
PING ff02::2(ff02::2) from fe80::21f:d0ff:feb2:cf51 eth1: 56 data bytes
^C
--- ff02::2 ping statistics ---
11 packets transmitted, 0 received, 100% packet loss, time 10000ms
```

Era de esperar ya que no tengo router ipv6 en eth1...

Ejemplo de multicast (ICMPv6)

Descubrir los nodos sobre una interfaz.

```
hcaste@clara:~$ ping6 -Itun ff02::1
64 bytes from 2001:5c0:1000:b::415d: icmp_seq=1 ttl=64 time=0.112 ms
64 bytes from 2001:5c0:1000:b::415c: icmp_seq=1 ttl=64 time=269 ms
(DUP!)
```

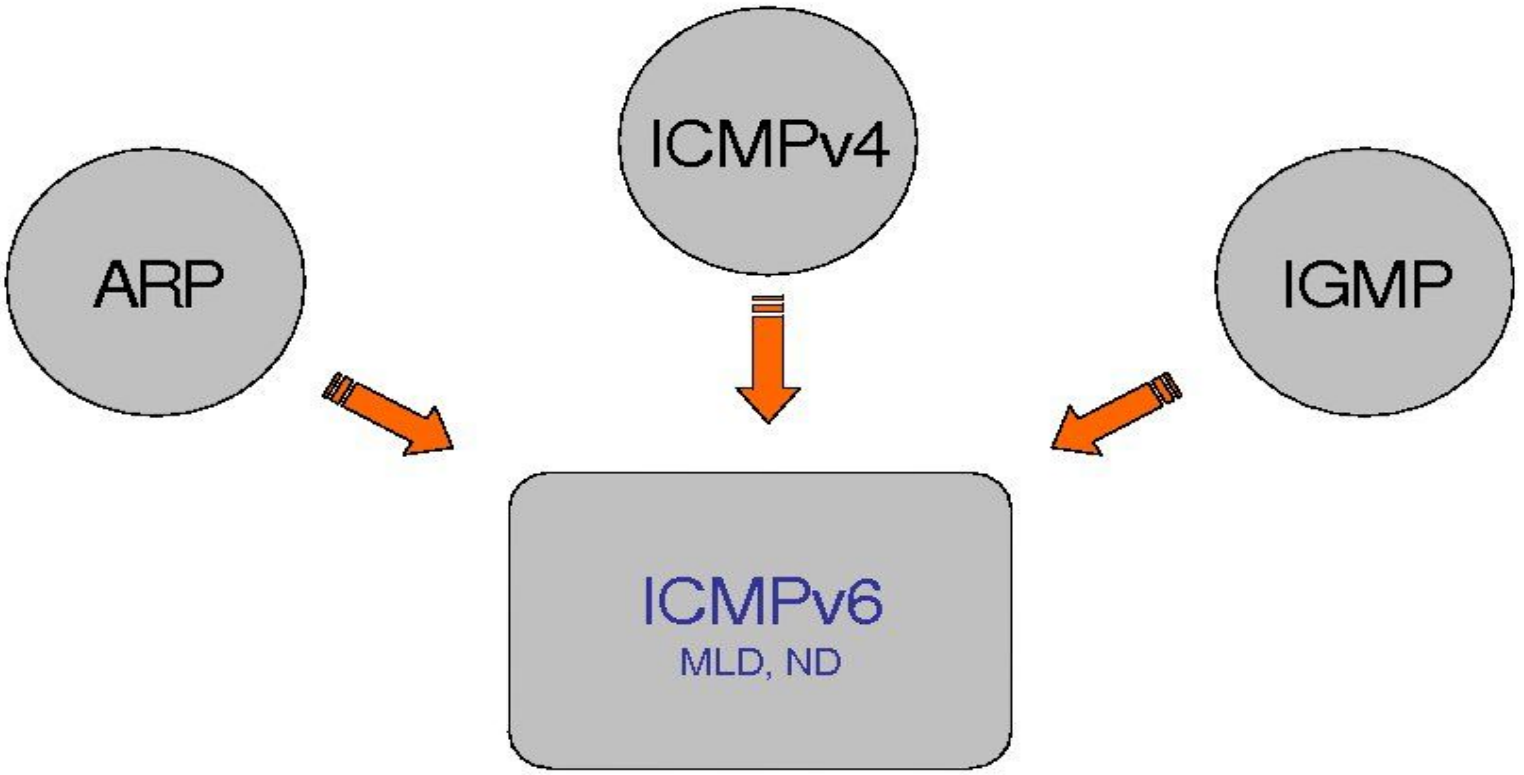
Sobre 2001:5c0:1000:b::/64 hay dos interfaces. Las dos corresponden al tunel cautivo. 2001:5c0:1000:b::415d corresponde a mi computadora.

Descubrir los enrutadores sobre una interfaz.

```
hcaste@clara:~$ ping6 -Itun ff02::2
PING ff02::2(ff02::2) from 2001:5c0:1000:b::415d tun: 56 data bytes
^C
--- ff02::2 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2015ms
```

Tampoco hay enrutadores.

ICMPv6



ICMPv6 (RFC 2463)

El protocolo de Mensajes para Control de Internet descripto en el RFC 792 fue adaptado para el uso en IPv6.

Se le ha asignado el valor 58 a Next Header.

El formato genérico de los ICMPv6 es la siguinte: 8 bits de tipo + 8 bits de código + 16 bits de checksum. El campo tipo indica el tipo de mensaje, y su valor determina el formato del resto del mensaje. El campo código depende del campo tipo y se usa para elaborar una jerarquía.

Hay dos tipos de mensajes: mensajes de error (tipo=0..127) y mensajes informativos (tipo=128..256) por ejemplo tipo=128 *Echo Request*, tipo=129 *Echo Reply*.

Por razones de seguridad las cabecers ICMPv6 pueden estar encriptadas.

ICMPv6 Mensajes de error

Tipo=1 destino no alcanzado, Código=0 sin ruta, Código=1 comunicación prohibida, Código=4 puerto no alcanzado.
Tipo=2 Paquete demasiado grande
Tipo=3 tiempo de vida excedido, Código=0 límite de saltos, Códogo=1 tiempo de desfragmentación largo.
Tipo=4 Parámtros erroneos, Código=0 Cabecera con errores, Código=1 cabecera siguiente desconocida, Código=2 error en paquete.

ICMPv6 Mensajes de información

Aquí solo una referencia en el aspecto de descubrimiento y autoconfiguración (RFC 2462):

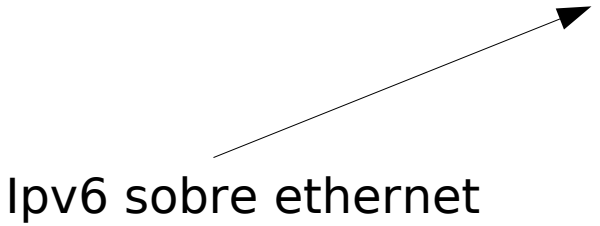
Tipo=133..137 Neighbor Discovery (RFC 2461) protocolo equivalente al ARP.
133 solicitud de enrutador
134 anuncio de enrutador
135 solicitud de vecino
136 anuncio de vecino
137 rediderccionar (*ICMP redirect*)

Direcciones Anycast

- Son direciones especiales usadas para solicitudes tipo ruta (ver ejemplo), etc.
- Las direcciones anycast no pueden ser usadas como dirección origen, solo como dirección destino.
- 2001:5c0:8fff:1::1/68 dirección del nodo, 2001:5c0:8fff:1::/68 dirección sub-red anycast que especifica una ruta.

Identificador de interfaz (ID)

- Es usado para definir una interfaz en un enlace
- Debe ser único en el enlace
- En algunos casos se usa como ID al identificador MAC especialemte en las *link-local*. (ipv6 sobre ethernet)
- Red+ID-MAC
  2001:05c0:8fff:fffe:2c0:dfff:fef9:d8d7



IPv6 sobre Ethernet (RFC 2464)

- El ID de la interface es la dirección MAC (Media Access Control IEEE 802)
- Se toma los 3 primeros octetos más significativos y se les intercala :fffe:
- luego se complementa a uno el segundo bit menos significativo del octeto más significativo

IPv6 sobre Ethernet (link)

- 00:C0:DF:F9:D8:D7 MAC (48 bits)
- 00c0:dfff:fef9:d8d7 se le intercaló :fffe:
- 2c0:dfff:fef9:d8d7 se completó a 1
- fe80::2c0:dfff:fef9:d8d7 se le añade el prefijo de link-local (128 bits)

Mostrando IPv6 existentes

```
hcaste@azul:~$ /sbin/ifconfig eth1|grep "inet6"
   Dirección inet6: 2001:5c0:8fff:a::11/64 Alcance:Global
   Dirección inet6: fe80::21f:d0ff:feb2:cf51/64 Alcance:Enlace

hcaste@azul:~$ /sbin/ifconfig tun|grep "inet6"
   Dirección inet6: 2001:5c0:1000:b::415d/128 Alcance:Global
```

Mostrando IPv6 rutas

```
hcaste@azul:~$ /sbin/route -A inet6 |grep -w "eth1"
Destino                    Next Hop      Flag  Metric Ref use interface
2001:5c0:8fff:a::/64       ::            U     256    0     0    eth1
fe80::/64                  ::            U     256    0     0    eth1
ff00::/8                   ::            U     256    0     0    eth1

hcaste@azul:~$ /sbin/route -A inet6 |grep -w "tun"
2001:5c0:1000:b::415d/128  ::            U     256 0     0 tun
2000::/3                   ::            U     1   0     0 tun
fe80::/64                  ::            U     256 0     0 tun
::/0                       ::            U     1   0     0 tun
ff00::/8                   ::            U     256 0     0 tun
```

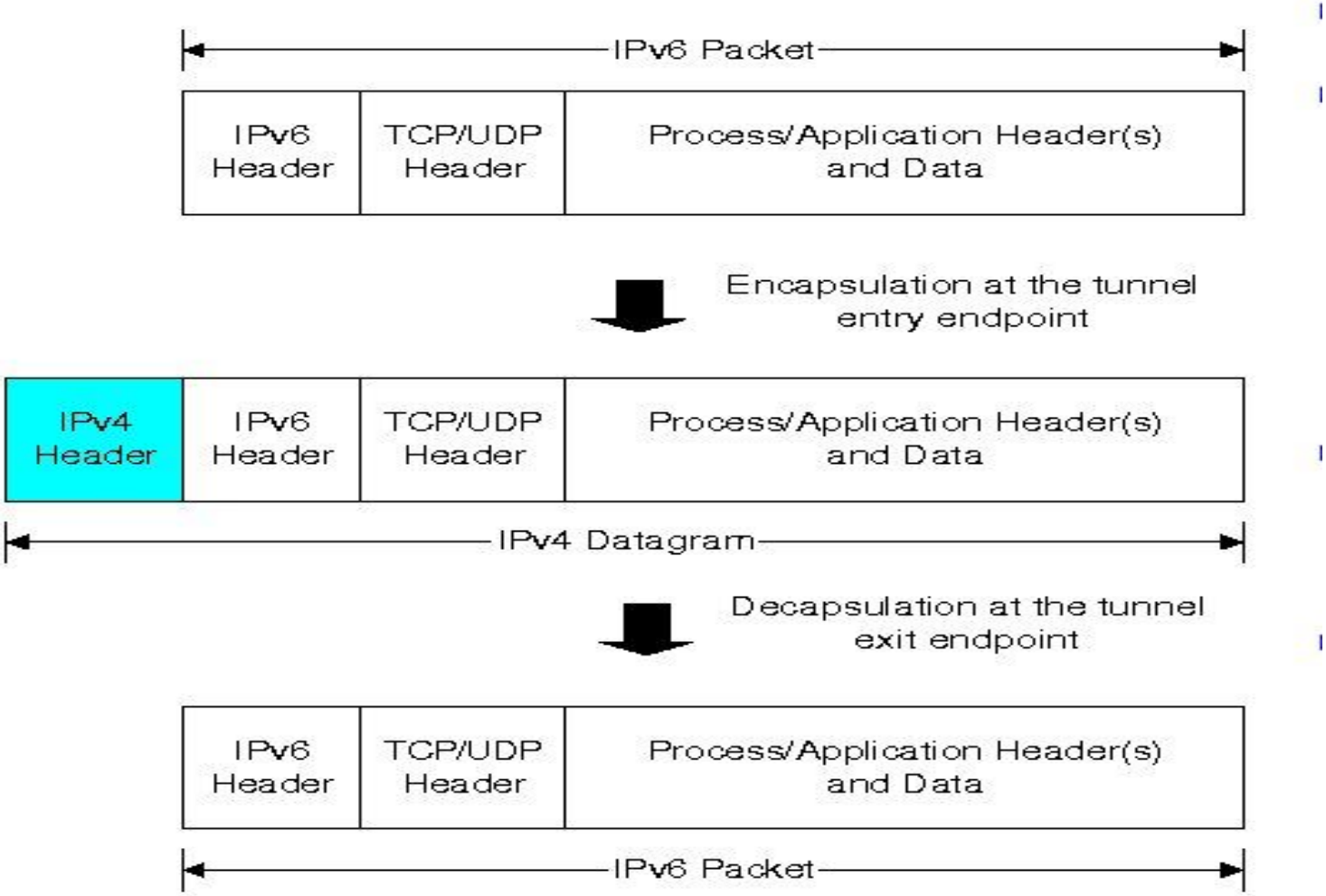
Red en IPv6 por tunel IPv4 privado

Esto será una breve descripción de como poner un servicio ipv6 en una máquina con Debian y Ubuntu. No será una explicación técnica sino paradigmática.

Advertencia!

No se debe bloquear el puerto 3653 en UDP pues es usado por el tunel. En freenet era el puerto 41 en TCP.

Red en IPv6 por tunel IPv4 privado



Políticas de Kernel

Asumo que la salida a Internet está en eth1
Primero definimos la política de conecciones.

```
sysctl -w net.ipv6.conf.all.forwarding=1
sysctl -w net.ipv6.conf.all.autoconf=0
sysctl -w net.ipv6.conf.all.accept_ra=0
sysctl -w net.ipv6.conf.all.accept_redirects=0
sysctl -w net.ipv6.conf.all.router_solicitations=0
```

Activar dipositivos

En el kernel 2.6.xx con xx>20:

/sbin/ifconfig eth1 inet6 add 2001:5c0:8fff:b::11/64

Advertencia en los kernels 2.6.xx con xx<21:

sit0 no puede usarse para definir nuestro dispositivo tunel, pues este es usado como dispositivo especial.

/sbin/ifconfig sit0 up (solo para kernels viejos)
/sbin/ifconfig eth1 inet6 add 2001:5c0:8fff:b::11/64

Activado en arranque

en /etc/rc.local basta colocar en los kernels nuevos.

ifconfig eth0 inet6 add 2001:5c0:8fffa:b::11/64
route -A inet6 add 2001:5c0:8fff:b::/64 dev eth1

Nota: Si la ruta (Anycast) no se lenvata automáticamente (Debian y Ubuntu) hay que ponerlo en el rc.local

Resultado (solo para kernels viejos)

```
sit0      Link encap:IPv6-in-IPv4
          inet6 addr: ::127.0.0.1/96 Scope:Unknown
          inet6 addr: ::192.168.1.4/96 Scope:Compat
          inet6 addr: ::10.0.0.4/96 Scope:Compat
          UP RUNNING NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Resultado (nuevos y viejos)

```
hcaste@azul:~$ ifconfig eth0
eth0      Link encap:Ethernet  direcciónHW 00:e0:7d:d8:08:2f
          Direc. inet:192.168.1.11  Difus.:192.168.1.255
Másc:255.255.255.0
          Dirección inet6: 2001:5c0:8fff:b::11/128 Alcance:Global
          Dirección inet6: fe80::2e0:7dff:fed8:82f/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500
Métrica:1
          Paquetes RX:2602 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:2969 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:42 long.colaTX:1000
          Bytes RX:447695 (447.6 KB)  TX bytes:276787 (276.7 KB)
          Interrupción:16 Dirección base: 0xb000

hcaste@azul:~$ /sbin/route -A inet6 |grep -w "eth0"
2001:5c0:8fff:b::/64       ::                  U     1    0        0 eth0
fe80::/64                  ::                  U     256  0        0 eth0
ff00::/8                   ::                  U     256  0        0 eth0
```

Tunel

El tunel ipv6-ipv4 es prestado por la empresa GOGO6 (ex Hexago-Go6) de forma gratuita por ahora. Se lo conoce como FreeNet6. Hay dos formas de usar el tunel:

- Anónima: subred 2001:5c0:1000:a::/64
- Registrada: subred 2001:5c0:1000:b::/64

Tunel registrado

Se lo puede instalar de la siguiente forma:

1- Se pide un par de enlace (*pear*) a http://gogonet.gogo6.com/ que nos devolverá un nombre de usuario y correo por correo-e.
2- Se loguea en la cuenta asignada, se llena lo que se pide (no piden la targeta de crédito!! por ahora) en http://gogonet.gogo6.com/page/freenet6-services
3- En http://gogonet.gogo6.com/page/freenet6-home-access se baja el archivo gogoc-1_2-RELEASE.tar.gz (está bajo licencia BSD)
4- Se debe compilar y instalar el cliente pero con las bibliotecas OpenSSL, libcrypto y libpthread.

Tunel registrado

No requiere configuración previa

5- Para compilarlo hay que ejecutar make platform=linux all

6- Para instalarlo hay que ejecutar make platform=linux installdir=/usr/local/gogoc install
Entonces instala no solo los ejecutables sino el script de configuración y rutas de la interfaz en:
"/usr/local/etc/gogoc/template/linux.sh"

7- El ejecutable está en /usr/local/gogoc/bin y se llama gogoc

8- Antes de todo hay que configurar nuestro cliente

Tunel Registrado

En el archivo gogoc.conf debe aparecer lo siguiente:

```
userid=hcaste
passwd=*******
server=authenticated.freenet6.net
host_type=router # solo para GW
prefixlen=48 #solo para GW
if_prefix=eth1 #solo para GW
```

Nota: Los túneles anónimos solo pueden configurarse como host y no se pueden usar como gateway.

Tunel registrado

Una vez instalado el cliente, si todo salió bien debe aparecer:

2010/05/13 06:59:39 I gogoc: Tunnel negotiation successful. Accepted offer.
2010/05/13 06:59:39 I gogoc: Checking for Linux IPv6 support...
2010/05/13 06:59:39 I gogoc: IPv6 support found.
2010/05/13 06:59:39 I gogoc: TSP_OPERATION=TSP_TUNNEL_CREATION
2010/05/13 06:59:39 I gogoc: TSP_VERBOSE=3
2010/05/13 06:59:39 I gogoc: TSP_HOME_DIR=/usr/local/etc/gogoc
2010/05/13 06:59:39 I gogoc: TSP_TUNNEL_MODE=v6udpv4
2010/05/13 06:59:39 I gogoc: TSP_HOST_TYPE=host
2010/05/13 06:59:39 I gogoc: TSP_TUNNEL_INTERFACE=tun
2010/05/13 06:59:39 I gogoc: TSP_HOME_INTERFACE=
2010/05/13 06:59:39 I gogoc: TSP_CLIENT_ADDRESS_IPV4=186.124.140.180
2010/05/13 06:59:39 I gogoc:
TSP_CLIENT_ADDRESS_IPV6=2001:05c0:1000:000b:0000:0000:0000:415d
2010/05/13 06:59:39 I gogoc:
TSP_CLIENT_DNS_ADDRESS_IPV6=2001:05c0:1000:0011:0000:0000:0000:0002
2010/05/13 06:59:39 I gogoc: TSP_CLIENT_DNS_NAME=hcaste.broker.freenet6.net
2010/05/13 06:59:39 I gogoc: TSP_SERVER_ADDRESS_IPV4=64.86.88.116
2010/05/13 06:59:39 I gogoc:
TSP_SERVER_ADDRESS_IPV6=2001:05c0:1000:000b:0000:0000:0000:415c
2010/05/13 06:59:39 I gogoc: TSP_TUNNEL_PREFIXLEN=128
2010/05/13 06:59:39 I gogoc: Executing interface configuration script: /bin/sh
"/usr/local/etc/gogoc/template/linux.sh".
2010/05/13 06:59:39 I gogoc: Waiting for setup script to complete.

Tunel Registrado

2010/05/13 06:59:39 I gogoc: Script: linux.sh
2010/05/13 06:59:39 I gogoc: tun setup
2010/05/13 06:59:39 I gogoc: /sbin/ifconfig tun up
2010/05/13 06:59:39 I gogoc: This host is: 2001:05c0:1000:000b:0000:0000:0000:415d/128
2010/05/13 06:59:39 I gogoc: /sbin/ifconfig tun add 2001:05c0:1000:000b:0000:0000:0000:415d/128
2010/05/13 06:59:39 I gogoc: /sbin/ifconfig tun mtu 1280
2010/05/13 06:59:39 I gogoc: Adding default route
2010/05/13 06:59:39 I gogoc: /sbin/route -A inet6 del ::/0
2010/05/13 06:59:39 I gogoc: /sbin/route -A inet6 del 2000::/3
2010/05/13 06:59:39 I gogoc: /sbin/route -A inet6 add ::/0 dev tun
2010/05/13 06:59:39 I gogoc: /sbin/route -A inet6 add 2000::/3 dev tun
2010/05/13 06:59:39 I gogoc: Adding DNS server
2010/05/13 06:59:39 I gogoc: NOTE: Adjust template script to perform actions
2010/05/13 06:59:39 I gogoc: --- End of configuration script. ---
2010/05/13 06:59:39 I gogoc: Interface configuration script completed successfully.
2010/05/13 06:59:39 I gogoc: The host type is 'host'.
2010/05/13 06:59:39 I gogoc: The tunnel type is v6udpv4.
2010/05/13 06:59:39 I gogoc: Client proxying is disabled.
2010/05/13 06:59:39 I gogoc: Your IPv6 address is 2001:05c0:1000:000b:0000:0000:0000:415d.
2010/05/13 06:59:39 I gogoc: Your IPv6 DNS address is 2001:05c0:1000:0011:0000:0000:0000:0002.
2010/05/13 06:59:39 I gogoc: Keepalive initialized with peer
2001:05c0:1000:000b:0000:0000:0000:415c. Interval=30000ms. Timeout=5000ms. General timeout
at 3 consecutive timeouts.

Si el tunel levantó

tun Link encap:UNSPEC dirección HW 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00

Dirección inet6: 2001:5c0:1000:b::415d/128 Alcance:Global

ACTIVO PUNTO A PUNTO FUNCIONANDO NOARP MULTICAST MTU:1280 Métrica:1

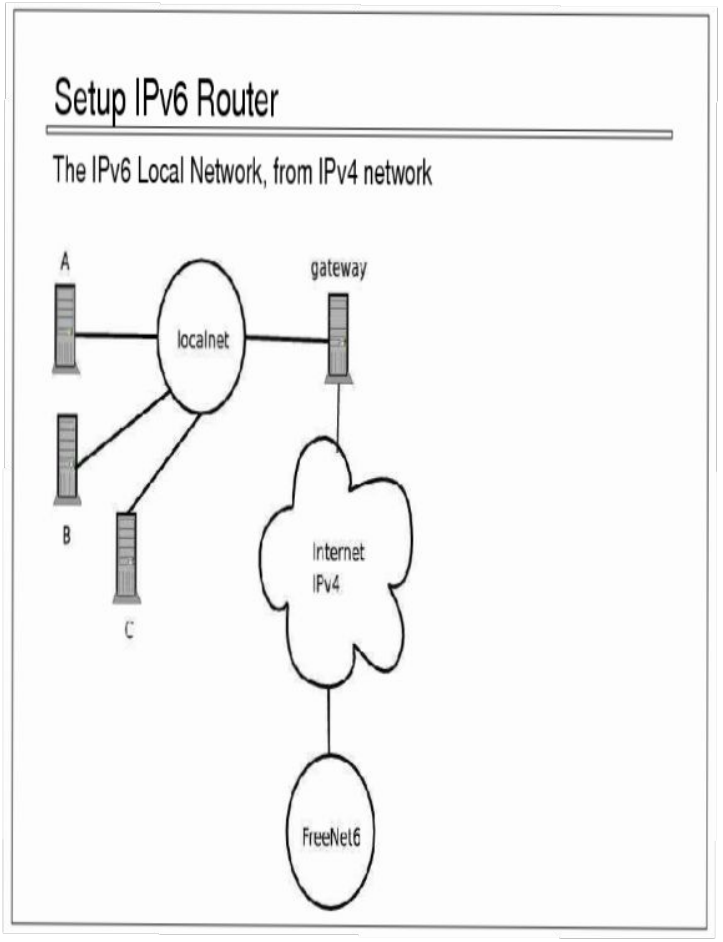
Paquetes RX:13378 errores:0 perdidos:0 overruns:0 frame:0

Paquetes TX:16662 errores:0 perdidos:0 overruns:0 carrier:0

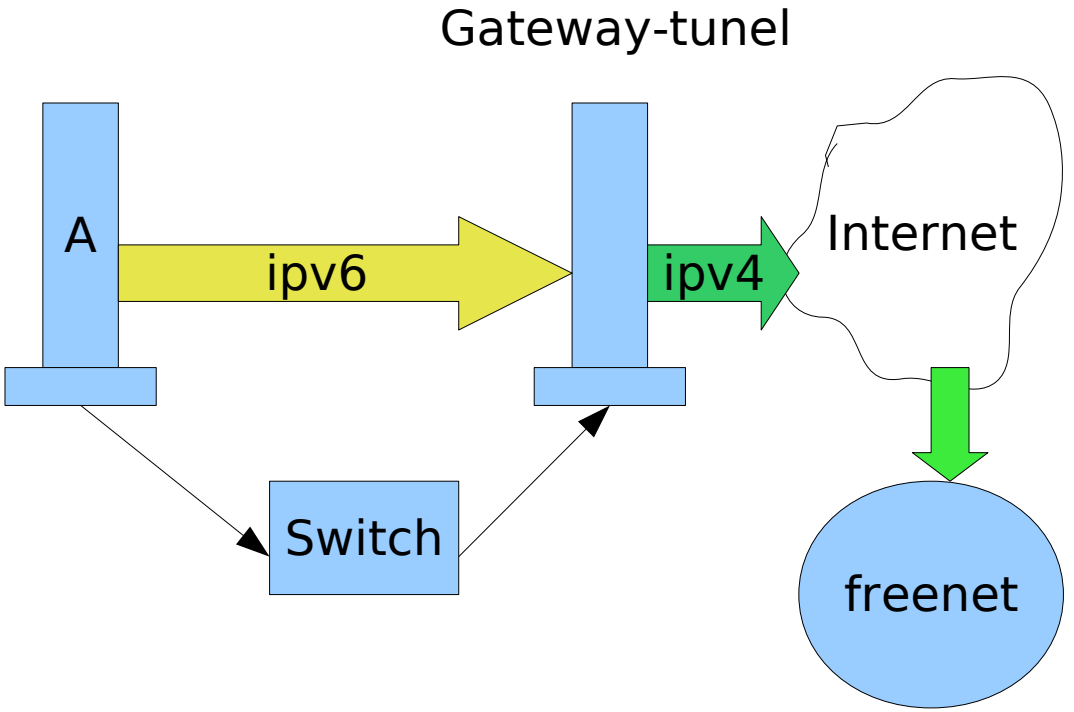
colisiones:0 long.colaTX:500

Bytes RX:3773169 (3.7 MB) TX bytes:1261001 (1.2 MB)

Tunel Registrado



- Los Hub y Switch enrutan sin problema los paquetes ipv6, pues usan la MAC!!



Tunel Registrado enrutado

También hay que decirle al nucleo que debe re-enviar los paquetes

net.ipv6.conf.default.forwarding = 1
net.ipv6.conf.all.forwarding = 1

Obsoleto en gogoc.

y debe colocarse en /etc/radvd.conf

```
interface eth1
{
    AdvSendAdvert on;
prefix 2001:5c0:1000::/64;
    {
    };
}
```

Tunel Registrado DNS

En /etc/bind/named.conf.local agregamos las zonas ipv6, lo suigiente es un ejemplo...

```
zone "ipv6.ar" {
	type master;
	file "extranet6.host";
	allow-query { any; };
	allow-transfer{any; };
};

zone "f.f.f.f.8.c.5.0.1.0.0.2.IP6.ARPA"
{
	type  master;
	file  "ipv6.arpa";
};
```

Archivo ipv6.ar

```
$TTL    86400
@       IN      SOA     oscura6.ipv6.ar. root.oscura.ipv6.ar. (
                2009032403      ;serie
                3600    ;refresco por hora
                1800    ;reintento por 1/2 horas
                604800  ;expira en 42 dias
                43200   ) ;minimo ttl
;
                IN      AAAA    2001:5c0:8fff:a::11
                IN      NS      ns.ipv6.ar.
$ORIGIN ipv6.ar.
oscura6         IN      AAAA    2001:5c0:8fff:a::11
clara6          IN      AAAA    2001:5c0:8fff:a::4
ns              IN      AAAA    2001:5c0:8fff:a::11
www.ipv6        IN      CNAME   oscura6
@       TXT     "oscura.ipv6.ar Servidor de Nombres"
@       HINFO   "Linux Debian" "amd64"
```

Archivo ipv6.arpa

```
$TTL 604800
@       IN SOA  oscura6.ipv6.ar. root.oscura6.ipv66.ar. (
                      2005032401 ;serie
                      36000   ;refresco por 100 horas
                      7200    ;reintento por 2 horas
                      3600000 ;expira en 1000 horas
                      2419200 );minimo 1 mes
        IN      NS      oscura6.ipv6.ar.
        IN      NS      ns.ipv66.ar.
$ORIGIN f.f.f.f.8.c.5.0.1.0.0.2.ip6.arpa.
a.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.4  in ptr  clara6.ipv6.ar.
a.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.1  in ptr  oscura6.ipv6.ar.
```

ip6tables

Este es un ejemplo de filtrado de paquetes para que ningún oportunista nos moleste.

```
#!/bin/sh
# FireWall IPv6
#Variables
fw6="/sbin/ip6tables"
#limpia relglas
$fw6 -F
$fw6 -X
#politicas por defecto
$fw6 -P OUTPUT ACCEPT
$fw6 -P INPUT ACCEPT
$fw6 -P FORWAR ACCEPT
# aceptamos ping6 y www
$fw6 -A FORWARD -p ipv6-icmp -j ACCEPT
$fw6 -A INPUT -p ipv6-icmp -j ACCEPT
$fw6 -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
#Bloquear
$fw6 -A INPUT  -i tun+ -p udp --dport 1:1024 -j DROP
$fw6 -A INPUT  -i tun+ -p tcp --dport 1:1024 -j DROP
```

Comprobación que uno está registrado

hcaste@azul:~/tmp$ host hcaste.broker.freenet6.net
hcaste.broker.freenet6.net has address 64.86.88.116
hcaste.broker.freenet6.net has IPv6 address 2001:5c0:1000:b::415d

hcaste@azul:~/tmp$ host 2001:5c0:1000:b::415d
d.5.1.4.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.b.0.0.0.0.0.0.1.0.c.5.0.1.0.0.2.ip6.arpa
domain name pointer hcaste.broker.freenet6.net.

Aparece la resolución inversa de nombre que la mantiene la empresa.
Tengo ip fija!! (pero ipv6)

Desde otra máquina con ipv6+tunel

```
tracepath6  www.ipv6style.jp
1?: [LOCALHOST]                                                pmtu 1280
1:  2001:5c0:1000:b::415c                                       238.899ms
1:  2001:5c0:1000:b::415c                                       240.845ms
2:  ix-0-0.224.mcore4.MTT-Montreal.ipv6.as6453.net              258.246ms
3:  if-12-3.mcore3.MTT-Montreal.ipv6.as6453.net                 273.121ms
4:  POS13-0.mcore4.NYY-NewYork.ipv6.as6453.net                  255.752ms
5:  if-3-0-0.mcore5.NYY-NewYork.ipv6.as6453.net                 254.843ms
6:  if-15-0-0.829.core2.NTO-NewYork.ipv6.as6453.net             259.119ms
7:  if-1-0-0.core3.NTO-NewYork.ipv6.as6453.net                  260.364ms
8:  xe-7-2.r01.nycmny01.us.bb.gin.ntt.net                       269.606ms
9:  ae-3.r20.nycmny01.us.bb.gin.ntt.net                         254.235ms
10:  as-2.r21.sttlwa01.us.bb.gin.ntt.net                         333.347ms
11:  as-2.r21.osakjp01.jp.bb.gin.ntt.net                         446.541ms
12:  ae-4.r21.tokyjp01.jp.bb.gin.ntt.net                         541.129ms
13:  2001:218:0:6000::62                                         462.757ms
14:  2001:218:2001:117::2                                        505.761ms
15:  2001:218:2001:3005::87                                      448.359ms reached
    Resume: pmtu 1280 hops 15 back 53
```

Registros Apache2

http://hcaste.broker.freenet6.net
Funciona!!! el servido apache. (Host virtual ipv4 on ipv6)

En la red interna 2001:5c0:8fff:a::/64

2001:5c0:8fff:a::4 - - [22/Aug/2010:16:11:32 -0300] "GET /horacio9573/yo2.png HTTP/1.1" 200 261296 "http://oscura.ipv6.ar/horacio9573/principal.html" "Mozilla/5.0 (X11; U; Linux i686; es-AR; rv:1.9.2.8) Gecko/20100723 Ubuntu/9.04 (jaunty) Firefox/3.6.8"

En el tunel llamado por otra computadora (Red Claro), 2001:5c0:1000:b::415c es la ip del par del tunel.

2001:5c0:1000:b::415c - - [23/Aug/2010:07:06:15 -0300] "GET / HTTP/1.1" 200 325 "http://www.google.com.ar/url?sa=t&source=web&cd=1&ved=0CBQQFjAA&url=http%3A%2F%2Fhcaste.broker.freenet6.net%2F&rct=j&q=hcaste.broker.freenet6.net&ei=_0dyTPaPOoOglAeX0rjODw&usg=AFQjCNGZKpeLUfYM2OyuR8cl9bAmWVaUww" "Mozilla/5.0 (X11; U; Linux i686; es-AR; rv:1.9.2.8) Gecko/20100723 Ubuntu/9.04 (jaunty) Firefox/3.6.8"

Inconveniente del tunel

No se pude llegar a la red 6to4 ó 2002::/16 a partir del 2009 en adelante:

hcaste@azul:~/tmp$ ping6 caspa.ipv6.frlp.utn.edu.ar
PING caspa.ipv6.frlp.utn.edu.ar(2002:aad2:1004:480a::2) 56 data bytes

Obs sobre 6to4
Ipv4=170.194.16.4 → aad2:1004 y $(480a)_{16}$=UTN-FRLP subred a_{16} que es el SLA-ID

From ix-5-0-1.6bb1.MTT-Montreal.ipv6.as6453.net icmp_seq=3 Destination unreachable: No route
From ix-0-0.224.mcore4.MTT-Montreal.ipv6.as6453.net icmp_seq=5 Destination unreachable: No route

El log da la siguinte información:

Aug 11 18:11:08 azul named[2366]: unexpected RCODE (SERVFAIL) resolving 'ns1.riu.edu.ar/A/IN': 200.16.98.2#53
Aug 11 18:11:08 azul named[2366]: unexpected RCODE (SERVFAIL) resolving 'ns1.riu.edu.ar/AAAA/IN': 200.16.98.2#53

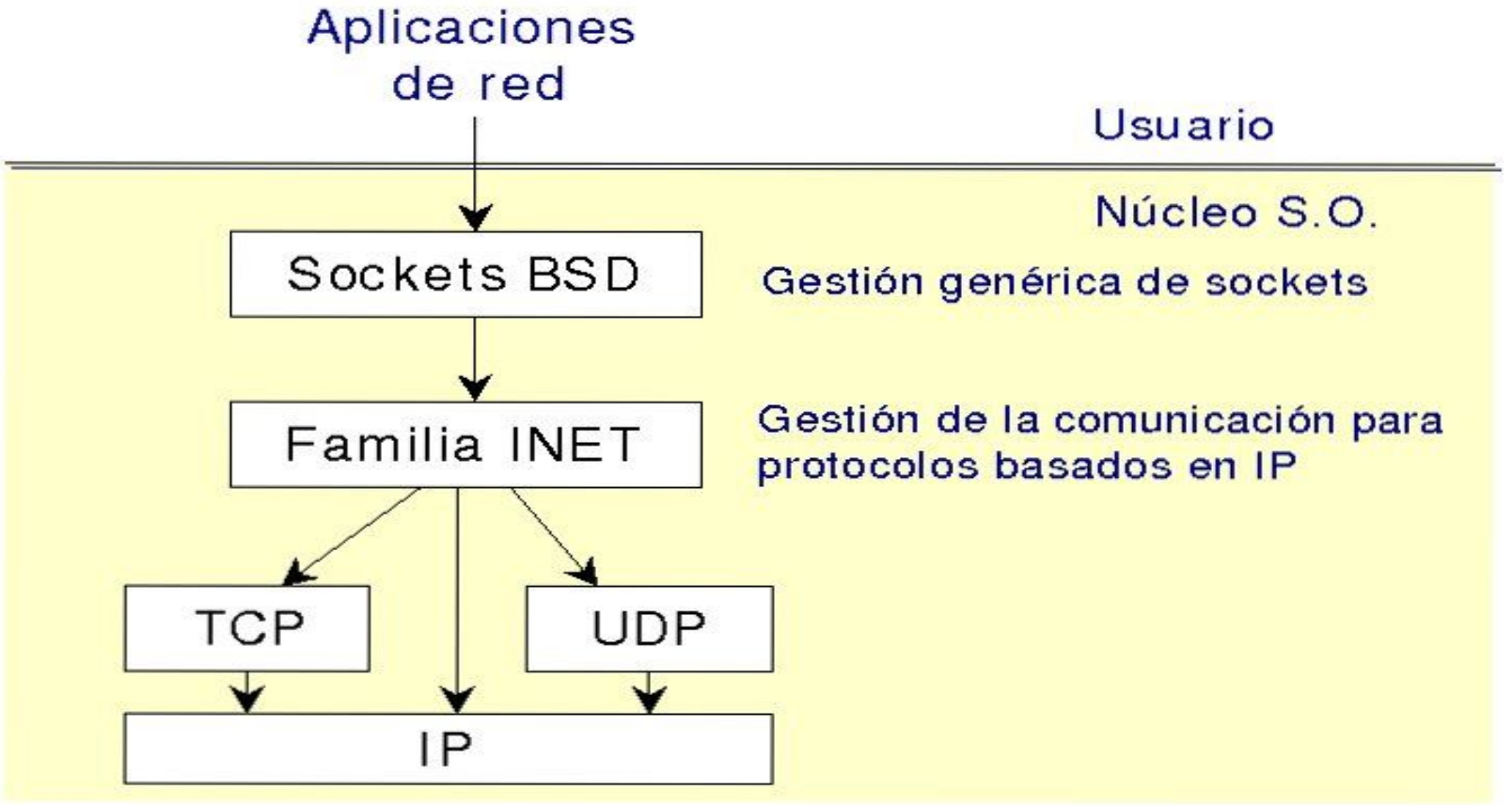
XCHAT en IPv6

XCHAT hecho en Python soporta IPv6, como lo muestra este log

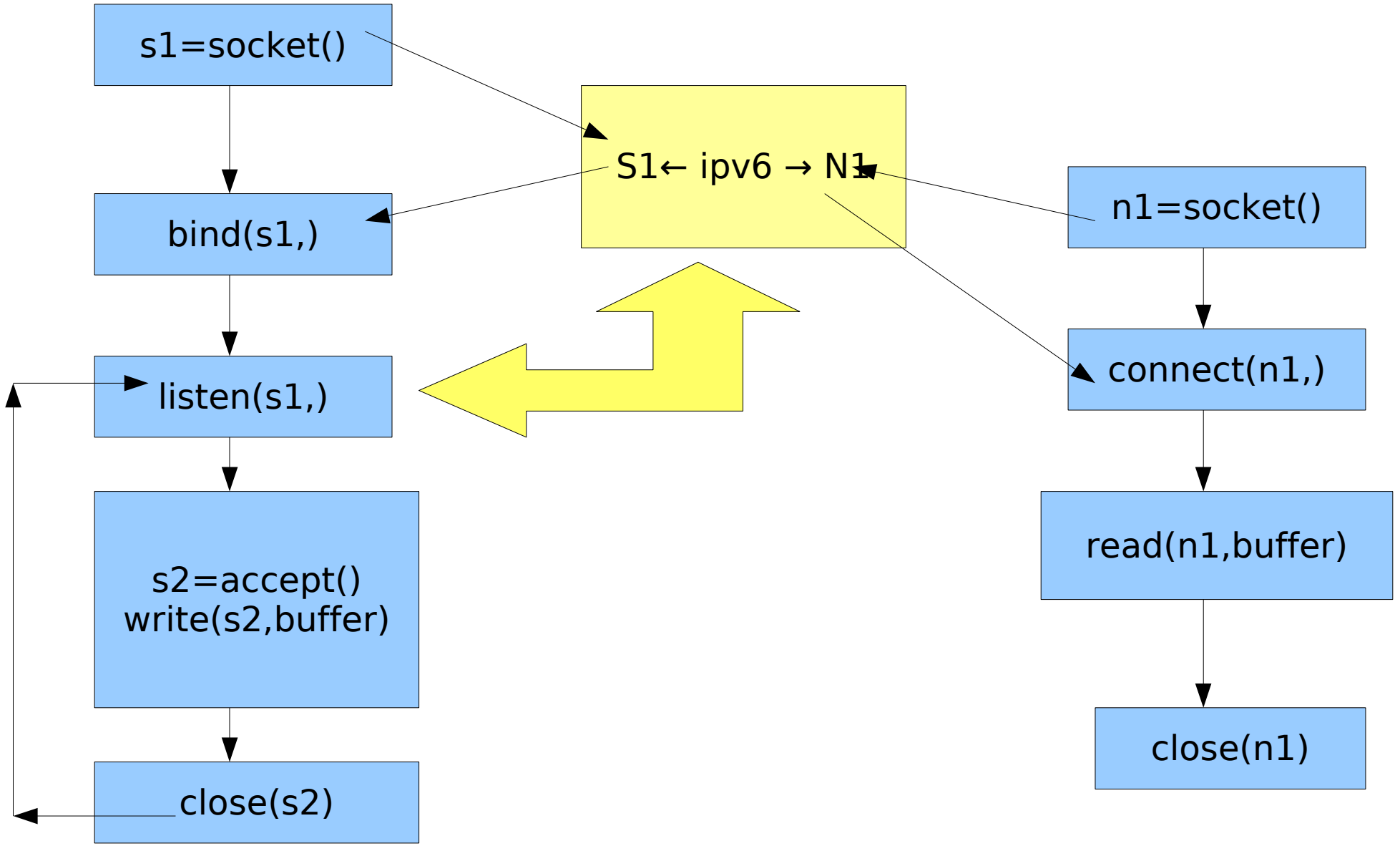
Python interface loaded
 Perl interface loaded
 Tcl plugin for XChat - Version 1.63
 Copyright 2002-2005 Daniel P. Stasinski
 http://www.scriptkitties.com/tclplugin/
 Tcl interface loaded
* Buscando irc.freenode.net
* Conectando a chat.freenode.net (2001:6b0:5:1688::10) puerto 6667...
* Connected. Now logging in...
* *** Looking up your hostname...
* *** Checking Ident
* *** No Ident response
* *** Found your hostname
* Welcome to the freenode Internet Relay Chat Network Dementor9573
* Your host is lindbohm.freenode.net[2001:6b0:5:1688::10/6667], running version ircd-seven-1.0.3
* This server was created Wed Feb 24 2010 at 00:01:31 CET

Socket IPv6

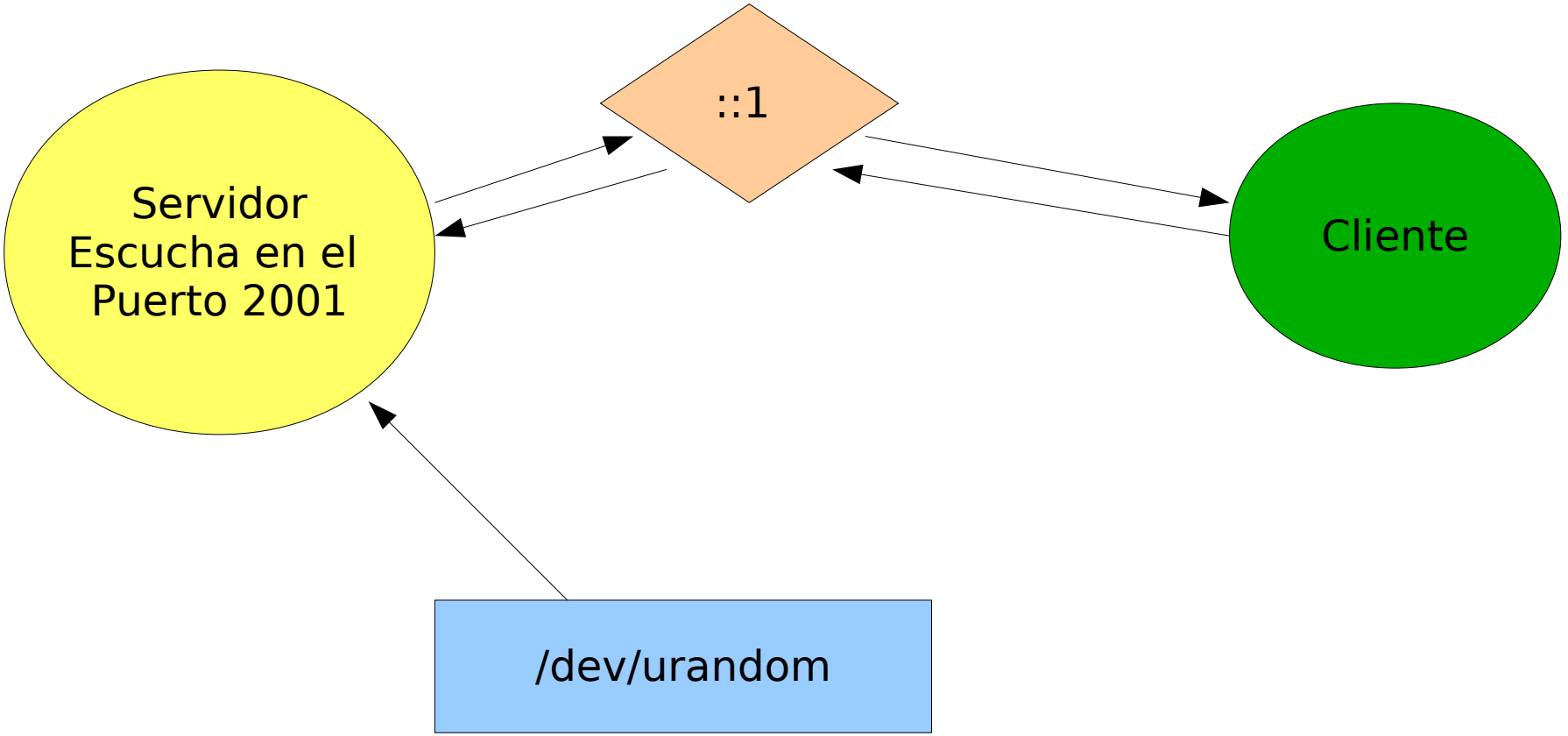
Modelo de Socket BSD



Esquema operativo cliente-servidor TCP en IPv6



Paradigma, un servidor de números aleatórios



Código del cliente

```
.........
#define PORT 2001 /* una odisea en el espacio */
#define MAXMENS 1024
char buf[256];
int main(){
   int s1, s2, s3;
   int x;
   struct sockaddr_in6 cliente;
   /*------*/
   x=0;
   cliente.sin6_family=AF_INET6;
   cliente.sin6_port=htons(PORT);
   inet_pton(AF_INET6, "::1", &(cliente.sin6_addr)); /* diferente a IPv4 */
   s1=socket(PF_INET6,SOCK_STREAM,0);
   if (s1<0){
       perror("Error en enchufe\n");
       exit(1);
   }
.........
```

Código del cliente

```
/* iguál que en ipv4... */

   s2=connect(s1,(struct sockaddr *)&cliente,sizeof(cliente));
   if (s2<0){
       perror("Conexion no permitida\n");
       exit(1);
   }
   read(s1,buf,sizeof(buf));
   close(s1);
   x=atoi(buf);
   printf("Escuche %d\n",x);
   return 0;
}
```

Código del servidor

```
#define PORT 2001
#define CLIENTES 5
int main(){
   char Buffer[256];
   int pid,s1,s2,d1;
   int Buf1;
   FILE *flujo;
   struct sockaddr_in6 servidor;
   /* con esto se genera el demoino servidor */
   pid=fork();
   if(pid<0){
       perror("No se clon\'o\n");
       exit(1);
   }
   if(pid!=0){
      printf("Demonio creado su pid= %d\n",pid);
      exit(0);
   }
   /*-------------------------*/
```

Código del servidor

```
servidor.sin6_family=AF_INET;
servidor.sin6_port=htons(PORT);
inet_pton(AF_INET6, "::1", &(servidor.sin6_addr));
s1=socket(PF_INET6,SOCK_STREAM,0);
if (s1<0){
    perror("Error en enchufe\n");
    exit(1);
}
d1=bind(s1,(struct sockaddr *)&servidor,sizeof(servidor));
if (d1<0){
    perror("No se pude enganchar\n");
    exit(1);
}
```

Código del servidor

```
  while(1){
    listen(s1,CLIENTES);
    d1=sizeof(servidor);
    s2=accept(s1,INADDR_ANY,&d1);
    if(s2<0){
      perror("Error al aceptar\n");
      exit(1);
    }
    flujo=fdopen(s2,"w");
    if(flujo==NULL){
      perror("Error flujo no creado\n");
      exit(1);
    }
    d1=open("/dev/urandom", O_RDONLY);
    if(d1<0){
      perror("Error no leo urandom\n");
      exit(1);
    }
    read(d1,&Buf1,sizeof(int));
    close(d1);
    sprintf(Info,"Mande: %d\n",Buf1);
    syslog(LOG_INFO|LOG_USER, Info);
    fprintf(flujo,"%d",Buf1);
    fclose(flujo);
    close(s2);
  }
  close(s1);
  return 0;
}
```

Ensayos

hcaste@azul:~./cliente6
Aug 24 19:07:01 azul server6-2: Mande: 1640948724 ← syslog()
Escuche 1640948724 <--- recibe el cliente

$ls -l /proc/23029/fd
total 0
lrwx------ 1 hcaste hcaste 64 2010-08-24 08:29 0 -> /dev/pts/0
lrwx------ 1 hcaste hcaste 64 2010-08-24 08:29 1 -> /dev/pts/0
lrwx------ 1 hcaste hcaste 64 2010-08-24 08:29 2 -> /dev/pts/0
lrwx------ 1 hcaste hcaste 64 2010-08-24 08:29 3 -> socket:[353030]
lrwx------ 1 hcaste hcaste 64 2010-08-24 19:08 5 -> socket:[476036]

3--> TCP/::1, 5--> UDP/syslog()

$netstat -ta

tcp6	0	0 [::]:www	[::]:*	ESCUCHAR
tcp6	0	0 localhost:2001	[::]:*	ESCUCHAR
tcp6	0	0 [::]:domain	[::]:*	ESCUCHAR

Una alternativa más académica

```
.........................
char host[MAXMENS] = "localhost";
struct sockaddr_in6 server;
struct hostent *servidor;
.............
.............
server.sin6_family = AF_INET6;
server.sin6_port = htons(puerto);
if (servidor = gethostbyname2(host,AF_INET6)) {
      memcpy(&server.sin6_addr, servidor->h_addr, servidor->h_length);
    } else {
      printf("No se encontro %s (%s)\n",host,strerror(errno));
      printf("Ejecute: %s [servidor [puerto]]\n", argv[0]);
      exit(1);
}
```

Cliente en Python

```
import socket
import string
c=socket.socket(socket.AF_INET6,socket.SOCK_STREAM)
c.connect(("::1",2001))
recibido=c.recv(10)
cosa=string.atoi(recibido)
print "Escuche: ",cosa
c.close()
```

Fin, ...