"CODAREC6: AN IPV6 TEST BED" – LABORATORIO DE ESTUDIO, DISEÑO, DESARROLLO, IMPLEMENTACIÓN, ENSAYO Y CAPACITACIÓN DEL PROTOCOLO DE INTERNET VERSIÓN 6

Carlos Taffernaberry, Alejandro Dantiacq Picolella, Gustavo Mercado y Adrián Francisconi

CODAREC
Universidad Tecnológica Nacional
Facultad Regional Mendoza
Mendoza, 5500, Argentina
carlos_taffe@frm.utn.edu.ar, alejandrod@frm.utn.edu.ar,
gmercado@frm.utn.edu.ar

Abstract

Despite the fact that the current Internet Protocol -known as IPv4- has successfully served for more than 20 years, it is nowadays reaching its own design limits, showing itself unable to provide an adequate response to actual desirable features.
Internet Engineering Task Force (IETF) started developing a new Internet protocol, call IPV6, to replace the older one, that contemplates improvements in addressing, security, quality of service, etc.
To achieve an appropriate transition between IPV4 and IPV6 they should be kept in mind the training, implementation and diffusion of the new protocol; in an appropriate development environment.
This work has as objective to design and to develop the first IPV6 test laboratory for the region.
A IPV6 6bone addresses range was gotten and a dual stack node was mounted with IPV6 basic development services in linux platform. The node was connected to Internet and links (tunneling) were configured with the utn.frlp and the 6bone. Additionally IPV6 client's machines were settled with diverse operating systems, conforming an IPV6 native local area network (LAN). IPV6 Address space was delegated to other regional institutions to achieve the project goals.
The results and experience obtained, from CODAREC6 test bed, will advance the IPV6 protocol understanding and will help in its transition and deployment.

Keywords: Internet, IP Protocol, Advanced Networks, Internet2

Resumen

El protocolo de internet actual, IPV4, ha vinculado al mundo por más de 20 años, aunque se encuentra ya al límite de su diseño y no puede dar respuestas adecuadas a las necesidades actuales.
El IETF desarrolló uno nuevo llamado IPV6, que contempla mejoras en direccionamiento, seguridad, calidad de servicio, etc.
Para lograr una apropiada transición entre IPV4 e IPV6 se deben tener en cuenta la capacitación, implementación y difusión del nuevo protocolo; en un ambiente de desarrollo adecuado.
Este trabajo tiene como objetivo diseñar e implementar uno de los primeros laboratorios de ensayo basado en IPV6 para la región.
Se consiguió un rango de direcciones IPV6 del 6bone. Se montó un nodo con dual stack y servicios básicos de desarrollo en IPV6 sobre plataforma linux. Se ubicó el nodo en internet y se configuraron enlaces (túneles) con la utn.frlp y el 6bone.
Adicionalmente se instalaron máquinas clientes con IPV6 con diversos sistemas operativos, creando una red local (LAN) nativa IPV6. Finalmente se delegó numeración IPV6 a otras entidades

regionales desde el CODAREC6.
El laboratorio experimental CODAREC6 ha permitido lograr resultados y experiencias que incrementan el conocimiento regional del protocolo IPV6 y ayudará en su transición y despliegue.

Palabras claves: Internet, Protocolo IP, Redes Avanzadas, Internet 2

1 INTRODUCCIÓN

EI protocolo IPV4 comienza a dar señales de debilidad. Después de 20 años, la versión 4 del protocolo de Internet (IP) ya no puede seguir brindando respuestas adecuadas, sobretodo en cuanto al paulatino agotamiento de las direcciones IP disponibles, un proceso que culminará en unos pocos años, al ritmo actual de crecimiento de Internet. Formadas por cuatro grupos de tres números cada uno, las direcciones IP identifican a cada uno de los dispositivos (PC, servidor, PDA, teléfono móvil, electrodomésticos, automóviles...) conectados a la red mundial que forma Internet y permiten que la información enviada llegue efectivamente al destino deseado. Ante el enorme crecimiento de usuarios de Internet, que hoy tienen exigencias distintas a las de hace unos años, las poco más de cuatro mil millones de direcciones en todo el mundo que posibilita el IPV4 se han vuelto insuficientes. [1]
El tiempo de vida de IPV4 se ha extendido gracias a técnicas tales como reutilización de direcciones y uso temporal de asignaciones. Si bien estas técnicas parecen incrementar el espacio de direcciones y satisfacer el crecimiento de la red, fallan en atender las necesidades de las nuevas aplicaciones. La necesidad de ambientes siempre funcionando (“always-on environments”, como Internet residencial a través de accesos de banda ancha, cable módem, o Ethernet-to-the-Home) para ser contactables, excluye estas conversiones de direcciones IP. Técnicas de pooling y asignación temporaria y "plug and play", que requieren los dispositivos para el hogar con conexión a Internet, ampliarán los requerimientos de direcciones. [2].
Finalmente en 1992 la Internet Engineering Task Force (IEFT), convocó a la comunidad de investigadores para estudiar alternativas superadoras para el IPV4. El resultado llegó en 1995 y se llamó IPV6 (Internet Protocol versión 6) [3]. Si bien por estos días IPV6 es especialmente atractivo para los pioneros en los sectores de redes inalámbricas, de juegos, de uso doméstico, redes de investigación nacional conectadas a nivel mundial, organismos militares y gobierno, una vez estandarizada, entre 2005 y 2008, ofrecerá:

- Direccionamiento extendido: Con 128 bits para la dirección IP, hará innecesario el uso de NAT y direccionamiento privado.
- Calidad de Servicio (Qos): IPV6 puede diferenciar los paquetes de datos como pertenecientes a un flujo particular, y así otorgar un ancho de banda en función de cada necesidad, ya sea para correo electrónico, comunicaciones de voz o videoconferencia.
- Capacidades de autentificación y privacidad: IPV6 emplea como parte integral el entorno de seguridad IPSec, que no está implementado en los hosts del IPV4 en forma nativa.
- Autoconfigurable: en IPV6 los nodos no necesitan ser configurados manualmente.
- End to end: IPV6 no usa NAT ya que tiene direcciones globales para todos los nodos. Así, éstos pueden reenviar cada paquete sin alterar su contenido.
- Simplificación del formato del encabezamiento: Es más sencillo y su tamaño es fijo. Se han suprimido campos como el checksum, tos y fragmentación, y agregado uno para identificar flujos de datos. Las funciones de los campos eliminados se logran con encabezados de extensión, que permiten incorporar nuevas características al protocolo, como IPSec o movilidad.

La tecnología IPV6 además produce el consecuencia de afectar a toda la Internet, es decir a todas las redes conectadas mundialmente.
Existen varias iniciativas de redes experimentales en IPv6, siendo la más importante y conocida el 6bone [4], el cual después de un exitoso camino ha dejado de existir. También en América Latina instituciones como CLARA [5], REUNA de Chile [6] y CUDI de México [7] han comenzado ha implementar redes Académicas Avanzadas con tecnología IPV6. En Argentina existen varia

instituciones con redes experimentales como, la UTN Facultad Regional La Plata [8], la incipiente RUT2 de la UTN y proyecto IPV6 de la red RETINA [9].

Una de las tareas más importantes es la transición del viejo protocolo al nuevo IPV6. Esta afirmación define un conjunto de mecanismos que los hosts y routers IPV6 deben implementar para ser compatibles con los hosts y router IPV4. Estos mecanismos permitirán usar infraestructuras IPV4 par IPV6 y viceversa. Después de varios intentos, algunos de ellos fallidos, para llevar adelante la migración de sistemas, la IETF ha formado el grupo de operaciones, el v6ops [10]. El objetivo del v6ops es llevar adelante y fomentar la transición a IPV6 con aportes de la comunidad de Internet y poniendo el foco en la aspectos operacionales y de seguridad.

En este sentido del abordaje de las nuevas tecnologías, el "CODAREC6: A IPV6 Test Bed" pretende ser un ambiente de trabajo y desarrollo que permita:

- El ESTUDIO del protocolo, comprendiendo cabalmente sus funciones, objetivos y alcances:
- El DISEÑO y actualización de aplicaciones para que puedan operar tanto con el nuevo protocolo como con el antiguo.
- El DESARROLLO e IMPLEMENTACIÓN de escenarios de trabajo para montar funciones, mecanismos y aplicaciones.
- El ENSAYO de la funcionalidad del protocolo y de las aplicaciones para comprobar la validez de los procesos y su aproximación a las normas, sobre diferentes plataformas operativas.
- La CAPACITACIÓN y la DIFUSIÓN para ayudar a la comunidad regional informática a comprender e implementar las características del protocolo Ipv6 [11].

Para cumplir con estas metas se construyó un Laboratorio informático único en la región, con conexión permanente a Internet y con la capacidad suficiente para operar como nodo universitario de IPV6. En dicho laboratorio se montaron las funciones corrientes y operativamente probadas, sobre varios sistemas operativos y en modalidad de "dual stack" para ensayar mecanismos de transición. Una de las características fundamentales del CODAREC6 es ser un ambiente propicio para que otros investigadores y desarrolladores lleven a cabo proyectos sobre el protocolo IPV6; poniendo especial énfasis a la formación de recursos humanos por medio de becas, tesinas de grado y tesis de postgrado.
En el siguiente capitulo se desarrollará la metodología y los mecanismos para cumplir estos requerimientos.

2 COMPONENTES DEL "TEST BED"

La figura 1 muestra el esquema el laboratorio de pruebas, que está formado por varias redes locales nativas ipv6 distribuidas geográficamente y conectadas físicamente a través enlaces WAN (Wide Area Network) a Internet.
Para comunicar estas "islas" ipv6 en redes que solo soportan ipv4, es necesario implementar una técnica denominada "túnel" [12] [13].

2.1 Elementos

En la tabla 1 se muestran los elementos que componen el CODAREC6. Denotándose las características del hardware utilizado, el sistema operativo empleado y los servicios implementados. Las Redes L.A.N (local area network) fueron desarrolladas usando tecnología fast-ethernet CSMA/CD (Carrier Sense Multiple Access with Collision Detection), (100baseTX) y se utilizaron enlaces WAN (wide area network) con anchos de banda no dedicados.

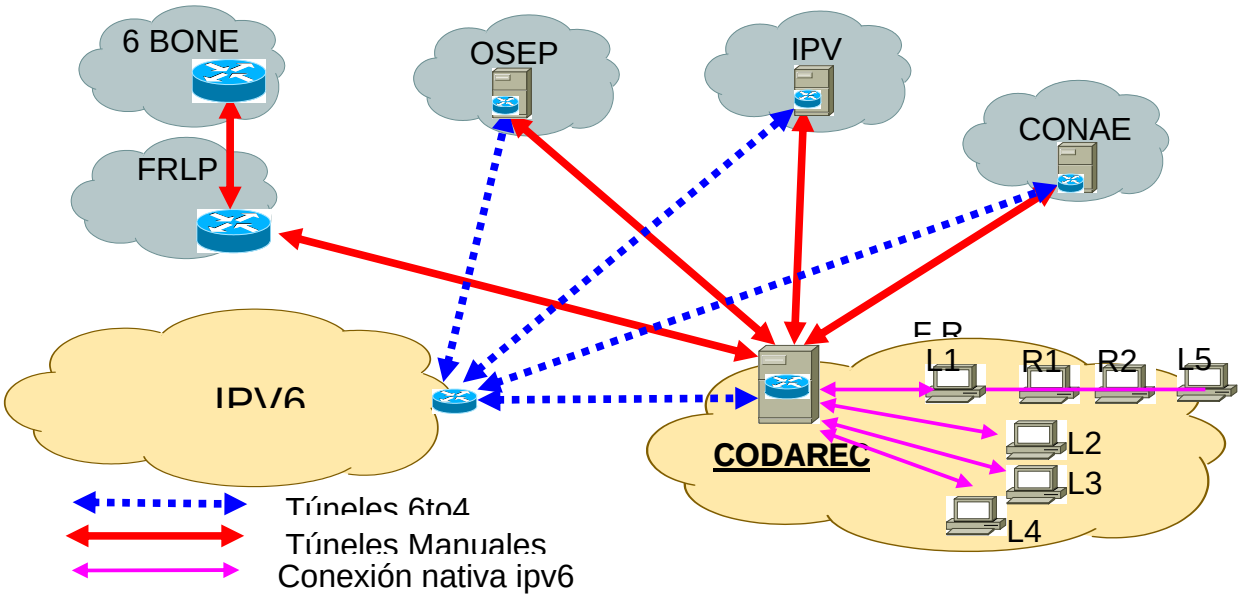

Figura 1: CODAREC6 Arquitectura

Nombre	***Marca-Procesador***	***Memoria R.A.M***	***Sistema operativo con doble stack [14]***	***Servicios***	***Direcciones globales unicast***
CODAREC6	Hewlett-Packard, "Proliant dl 40", procesador Intel-Xeon (dual core) de 2.8 Ghz.	1 Gbyte	Linux "kernel 2.6.15-1.2054_FC5smp Fedora core 5 [15] (ipv4 e ipv6)	dns [16], dhcp6 [17], http, ssh6, ftp6, webmail, radvd (Router Advertisement Daemon) [18] , ip6tables, Quagga Routing [19]	Túneles manuales ipv: 3ffe:38e1:600::1:ff19/126 conae:3ffe:38e1:600::1:ff05/126 osep: 3ffe:38e1:600::1:ff09/126 frlp: 3ffe:38e1::600:2:ff19/124 Túnel 6to4: 2002:aad2:ee19::1/48
Host "L1"	AMD athlon XP 2.2 Ghz.	128 MB	Windows XP server	dhcp6 y http clientes	2002:aad2:ee19:c0da::11/48
Host "L2"	Intel P III de 550 MHz,	128 MB	Unix Solaris 9.0	dhcp6 y http clientes	2002:aad2:ee19:c0da::12/48
Host "L3"	Intel Pentium III de 1.6 Ghz	256 MB	Linux kernel 2.6.11-1.1369_FC4 distribución "Fedora core 4"	Dhcp6, ssh6, ftp6 y http clientes	2002:aad2:ee19:c0da::13/48
Host "L4"	Intel Pentium III	64 MB	free BSD	http cliente	2002:aad2:ee19:c0da::14/48
"OSEP"	Marca IBM , modelo "netfinity PCServer 325", procesador Intel Pentium II 332 MHz	192 MB	Linux, kernel 2.6.16-1.2122_FC5 distribución Fedora core 5 .	Túneles tipo (manual y 6to4), router, ftp6, ssh6 y http.	Túnel 6to4 2002:c805:744a::1/48 Túnel manual 3ffe:38e1:600::1:ff08/126
"IPV"	Intel Celeron 2,13 Ghz	1.128 GB	Linux con 2.6.16-1.2122_FC5 distribución Fedora core 5.	Túneles tipo (manual y 6to4), router, servicios radvd, http, ftp6, ssh6 y webmail.	Túnel 6to4 2002:be30:245a::1/48 Túnel manual 3ffe:38e1:600::1:ff18/126
"CONAE"	Pentium III (Coppermine)	262 MB	Linux, kernel 2.4.22-1.2115.nptl distribución Fedora core 1.	túneles de tipo (manual y 6to4), y servicios radvd y ftp6	Túnel 6to4 2002:c810:534c::1/48 Túnel manual 3ffe:38e1:600::1:ff06/126
Host "R1"	Intel Celeron 1,7Ghz	256MB	Linux con 2.6.16-1.2122 distribución Fedora core 5,	túnel manual y routing ipv4 e Ipv6.	Direcciones de ensayo ipv6 3ffe::1:1/120 ; 3ffe::2:1/120 túnel : 3ffe::10:1/120
Host "R2"	Pentium III	128 MB	Linux kernel 2.6.11-1.1369_FC4 distribución "Fedora core 4"	túnel manual y routing ipv4 e Ipv6.	Direcciones de ensayo ipv6 3ffe::2:fe/120; 3ffe::3:1/120 túnel : 3ffe::10:2/120
Host "L5"	Pentium III	128MB	Linux, kernel 2.4.22-1.2115.nptl distribución Fedora core 1	Ipv4 ipv6	Direcciones de ensayo ipv6 3ffe::3:fe/120

Tabla 1: Detalle de Hardware usado

3 SERVICIOS IMPLEMENTADOS Y DESEMPEÑO

En esta sección se muestran los ensayos de las distintas funcionalidades ipv6 y el cumplimiento de las RFCs correspondientes.
Se implementó el ruteo nativo en IPV6, para poder verificar la mejora en cuanto al desempeño del ruteo en IPV4. Se ensayaron métodos de transición y se evaluó su desempeño.
Se verificó el cumplimiento de las RFC respecto de auto-configuración Statefull [20] y Stateless [21] por medio de DHCPv6 y RADV.
También, y para cumplir con la generación de un ambiente propicio para nuevas investigaciones, se implementaron otros servicios necesarios para el CODAREC6, como DNS con soporte de direcciones IPV6, RIPv2, BGP4, etc., [26] verificando el cumplimiento de las RFCs.

3.1 Encaminamiento nativo Ipv6

Se configuraron los routers R1 y R2, para vincular 3 redes internas ubicadas en el CODAREC6 test bed, de acuerdo a la topología del Figura 2.

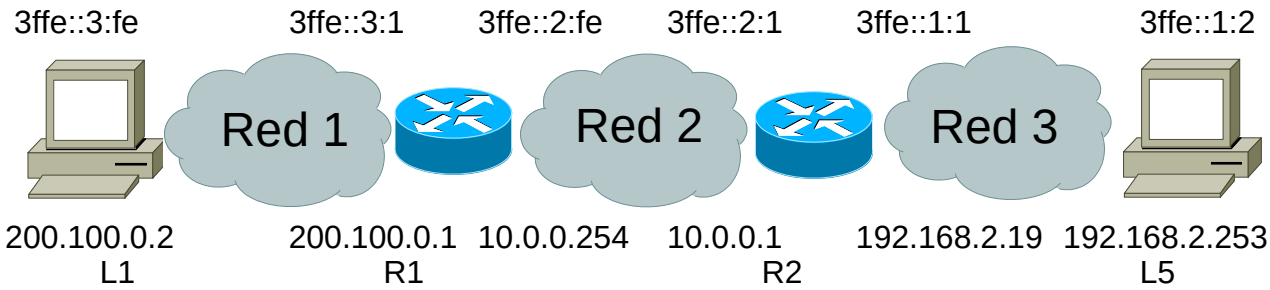

Figura 2: Topología de red

Todos los nodos se implementaron con sistema operativo gnu/linux. Se configuraron en forma idéntica con soporte para doble stacks Ipv4 e Ipv6. Las direcciones se asignaron manualmente, tomando los valores que indica el Figura 2. En R1 y R2 adicionalmente se habilitó el encaminamiento para ambos protocolos.
A continuación para comprobar el correcto funcionamiento para IPv4, se realizaron pruebas de conectividad. Se enviaron paquetes ICMP [22] echo request desde la red 3 a la red 1 y se recibieron las respuestas. Se aumentó la carga útil del paquete icmp a 2000 bytes, para producir fragmentación de datagramas IP [23]. Se muestra el comportamiento en la Figura 3 donde se verificó la fragmentación de datagramas en origen, pues la red subyacente era Ethernet con MTU de 1500 bytes. [2].

```
11:10:23.858053 IP 192.168.2.253 > 200.100.0.2: ICMP echo request, id 59675, seq 1, length
1480
11:10:23.858105 IP 192.168.2.253 > 200.100.0.2: icmp
11:10:23.862917 IP 200.100.0.2 > 192.168.2.253: icmp
11:10:23.865101 IP 200.100.0.2 > 192.168.2.253: ICMP echo reply, id 59675, seq 1, length
1480
11:10:24.857772 IP 192.168.2.253 > 200.100.0.2: ICMP echo request, id 59675, seq 2, length
1480
11:10:24.858212 IP 192.168.2.253 > 200.100.0.2: icmp
11:10:24.863111 IP 200.100.0.2 > 192.168.2.253: icmp
11:10:24.865354 IP 200.100.0.2 > 192.168.2.253: ICMP echo reply, id 59675, seq 2, length
1480
```

Figura 3: Fragmentación de datagramas IPV4

3.1.1 Comparación ruteo Ipv4 e Ipv6

Se comparó el comportamiento del protocolo Ipv6 para iguales estímulos, enviando paquetes icmpv6 tipo echo request [24] desde la red 1 a la 3 y la respuesta. Se aumentó la carga útil del paquete icmpv6 a 2000 bytes. Se representa el tráfico generado en el Figura 4.

```
14:31:01.296304 IP6 3ffe::1:2 > 3ffe::3:fe: frag (0|1448) ICMP6, echo request, seq 10, length
1448
```

```
14:31:01.296354 IP6 3ffe::1:2 > 3ffe::3:fe: frag (1448|560)
14:31:01.296371 IP6 3ffe::3:fe > 3ffe::1:2: frag (0|1448) ICMP6, echo reply, seq 10, length
1448
14:31:01.296388 IP6 3ffe::3:fe > 3ffe::1:2: frag (1448|560)
14:31:02.283468 IP6 3ffe::1:2 > 3ffe::3:fe: frag (0|1448) ICMP6, echo request, seq 11, length
1448
14:31:02.283943 IP6 3ffe::1:2 > 3ffe::3:fe: frag (1448|560)
14:31:02.290691 IP6 3ffe::3:fe > 3ffe::1:2: frag (0|1448) ICMP6, echo reply, seq 11, length
1448
14:31:02.291116 IP6 3ffe::3:fe > 3ffe::1:2: frag (1448|560)
```

Figura 4: Fragmentación datagramas IPV6

Se verificó que los paquetes IPv6 se enviaron fragmentados desde el host origen, constatándose que en cada cabecera se agregó un Fragmentation header [3].
Se concluye al comparar los time stamps [2] de los Figuras 3 y 4 que hubo más retardo utilizando la versión 6 del protocolo IP que la 4, a diferencia de lo que se suponía que ocurriría.

3.1.2 Desempeño en el ruteo

Para corroborar este comportamiento se realizó un ensayo más completo, enviando paquetes icmp tan rápido como la interfase de red pueda despacharlos, sin esperar la respuesta correspondiente. Con esta medida se exigió a los routers, aumentando sustancialmente el tráfico. Se tomaron los tiempos de RoundTrip [2] y se repitió el ensayo variando la carga útil entre 54 bytes y 10000 bytes. La cantidad con las que se obtuvo el promedio fue de 5000 paquetes. En la Tabla 2 podemos ver el resultado de los ensayos realizados.

Carga útil	***IPv4(miliseg)***			***Ipv6 (miliseg)***		
	Mínimo	Promedio	Máximo	Mínimo	Promedio	Máximo
56 bytes	0.963	1.165	8.113	1.071	1.193	2.836
500 bytes	3.606	3.823	4.612	3.740	3.947	5.318
2000 bytes	10.423	10.770	11.894	10.392	10.974	14.080
5000 bytes	15.671	16.730	24.836	15.761	17.735	30.079
8000 bytes	23.228	1146.369	1912.195	26.691	846.590	1571.177
10000bytes *	48.145	2777.031	4402.791	55.085	2598.171	4122.365

Tabla 2:Round Trip time

* Se pierden paquetes tanto para IPV4 como para IPV6. El porcentaje fue significativamente superior para el stack IPV4.

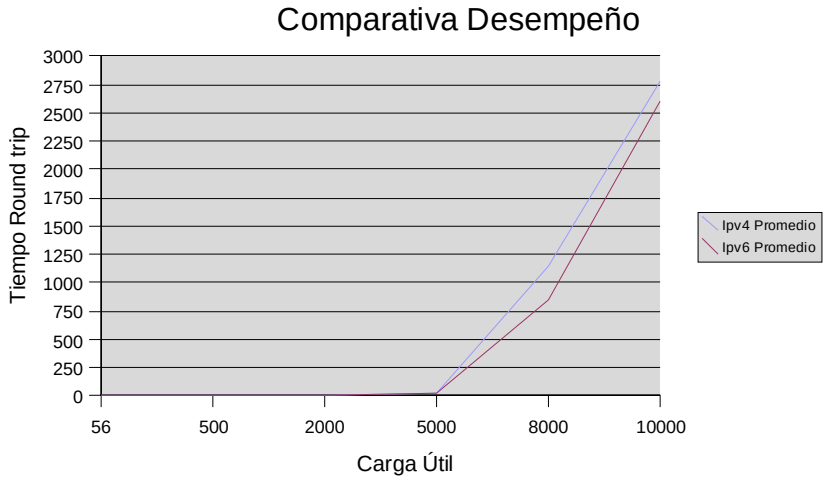

Figura 5: Comparación desempeño

El Figura 5 demostró que para tamaños de carga útil bajos, el tiempo de Roud Trip es menor para

Ipv4. Se justifica por que prevalece el aumento de tamaño del encabezado ipv6 sobre la optimización en los algoritmos de ruteo. A medida que el tamaño de carga útil aumenta, los routers Ipv4 se ven mas exigidos. Esta demora prevalece sobre un aumento del encabezado Ipv6.

3.1.3 Comportamiento frente a variaciones de MTU

Se comprobó el comportamiento de ambos protocolos cuando la MTU del camino era menor que el tamaño del datagrama enviado

Utilizando la misma topología del Figura 2, se inundó con paquetes Icmp para la versión Ipv4 con carga útil de 1472 bytes. Como las tres redes subyacentes eran ethernet, los paquetes icmp no sufrieron fragmentación, como era de esperar.

Posteriormente se disminuyó la MTU en las interfaces que vinculan R1 y R2. Dejando su valor en 1280 [25]. Se tomaron los tiempos Round Trip. La tabla 3 muestra la comparación, y un factor de degradación de desempeño. La cantidad con las que se obtuvo el promedio fue de 5000 paquetes.

Round Trip(ms)	***MTU 1500***	***MTU 1280***	***Degradación %***
Mínimo	9.376	9.907	5.66
Promedio	9.606	13.566	41.22
Máximo	10.694	28.393	165.5

Tabla 3: Degradación por fragmentación

Se realizó el mismo ensayo sobre el stack ipv6 para ver la degradación en el Round Trip cuando la MTU del canal es menor al tamaño de los paquetes enviados.

Se inundó de paquetes Icmpv6 con una carga útil de 1440 bytes. Como era de esperar los paquetes icmpv6 no sufrieron fragmentación en todo el camino. Se colocaron los tiempos involucrados en una tabla. Se disminuyó la MTU en las interfaces que vinculan R1 y R2. Dejando su valor en 1280.

Se inundó de paquetes Icmpv6, con una carga útil de 1440 bytes. Se logró verificar el mecanismo por el cual el origen fragmenta [26], como se observa en el Figura 6

```
17:57:20.237204 IP6 3ffe::1:2 > 3ffe::3:fe: ICMP6, echo request, seq 1, length 1448
17:57:20.240848 IP6 3ffe::1:1 > ff02::1:ff01:2: ICMP6, neighbor solicitation, who has 3ffe::1:2,
length 32
17:57:20.241282 IP6 3ffe::1:2 > 3ffe::1:1: ICMP6, neighbor advertisement, tgt is 3ffe::1:2,
length 32
17:57:20.241342 IP6 3ffe::1:1 > 3ffe::1:2: ICMP6, packet too big, mtu 1280, length 1240
17:57:21.237980 IP6 3ffe::1:2 > 3ffe::3:fe: frag (0|1232) ICMP6, echo request, seq 2, length
1232
17:57:21.238054 IP6 3ffe::1:2 > 3ffe::3:fe: frag (1232|216)
17:57:22.237829 IP6 3ffe::1:2 > 3ffe::3:fe: frag (0|1232) ICMP6, echo request, seq 3, length
1232
17:57:22.237904 IP6 3ffe::1:2 > 3ffe::3:fe: frag (1232|216)
17:57:22.244294 IP6 3ffe::3:fe > 3ffe::1:2: frag (0|1232) ICMP6, echo reply, seq 3, length 1232
17:57:22.244554 IP6 3ffe::3:fe > 3ffe::1:2: frag (1232|216)
```

Figura 6: Fragmentación en origen

La tabla 4 muestra los tiempos Round Trip y un factor de degradación de desempeño. La cantidad con las que se obtuvo el promedio fue de 5000 paquetes.

Round Trip(ms)	MTU 1500	MTU 1280	Degradación %
Mínimo	9.286	9.395	1.17
Promedio	9.527	9.571	0.46
Máximo	10.782	17.936	66.35

Tabla 4

De la tabla 4 pudimos concluir que ante una MTU del canal menor que el tamaño del paquete, la degradación para el protocolo Ipv6 es casi inexistente a diferencia de lo experimentado para Ipv4.

3.2 Establecimiento de túneles

Debido a la imposibilidad de actualizar simultáneamente a Ipv6 a todos los host conectados a

Internet, los grupos de trabajo del IETF plantearon mecanismos de migración graduales
El objetivo de este apartado es poder verificar unas alternativas disponibles para hacer la transición.

3.2.1 Túnel manual
Se encapsulan paquetes ipv6 (header y datos) dentro del payload de paquetes ipv4.
Utilizamos la topología del codarec6 test bed como se muestra en la figura 7.

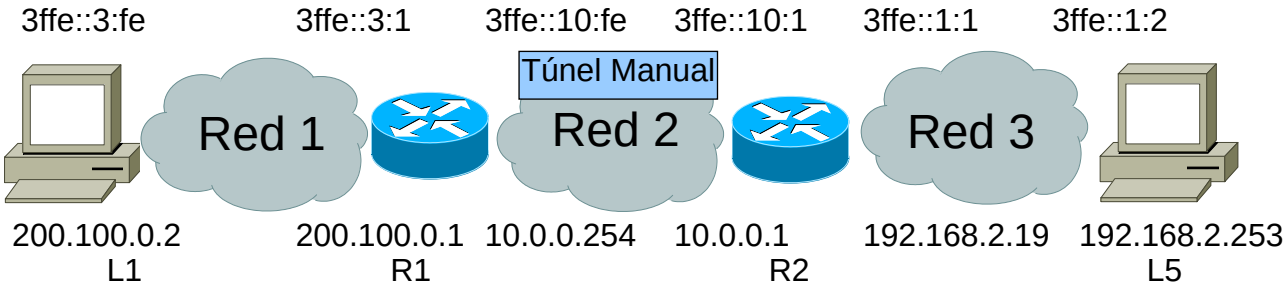

Figura 7: Topología para test de túnel

Los nodos se implementaron con sistema operativo gnu/linux.
Se configuraron en forma idéntica con soporte doble stacks Ipv4 e Ipv6 , a excepción de las interfaces de la red 10.0.0.0.Las direcciones se asignaron manualmente de acuerdo a la Figura 7.
Para R1 y R2 se configuró un túnel manual. Si bien el túnel se estableció en la misma red, en la práctica no es necesaria esta condición. Solo se necesita conectividad Ipv4 entre R1 y R2 y que los routers IPV4 intermedios encaminen el protocolo 0x29 [12].
Se observó el tráfico Ipv6 encapsulado en paquetes Ipv4 en el Figura 8 cuando se enviaron paquetes Icmpv6 desde la red 1 a la 3 y la respuesta. Esto permitió comprobar la transparencia para transferir datos en la implementación de mecanismos de transición.

```
16:32:24.870862 IP 10.0.0.1 > 10.0.0.254: IP6 3ffe::1:2 > 3ffe::3:fe: ICMP6, echo request, seq
1, length 64
16:32:24.871950 IP 10.0.0.254 > 10.0.0.1: IP6 3ffe::3:fe > 3ffe::1:2: ICMP6, echo reply, seq 1,
length 64
16:32:25.871406 IP 10.0.0.1 > 10.0.0.254: IP6 3ffe::1:2 > 3ffe::3:fe: ICMP6, echo request, seq
2, length 64
16:32:25.872318 IP 10.0.0.254 > 10.0.0.1: IP6 3ffe::3:fe > 3ffe::1:2: ICMP6, echo reply, seq 2,
length 64
```
Figura 8

Luego se aumentó la carga útil del paquete Icmpv6 a 2000 bytes. Se verificó nuevamente la transparencia de los túneles. Se observa que, al igual que en el Figura 4, fue agregado en cada cabecera IPv6 un Fragmentation header, cumpliendo con lo estipulado en el RFC [26].

3.2.2 Desempeño en el ruteo
Si bien se comprobó que la estrategia es viable en términos de transparencia, el overhead utilizado sin lugar a dudas tendría influencia en el desempeño del enlace. Para ponderar esta influencia, se realizó el mismo ensayo que en el punto 3.1.2.
Se tomaron los tiempos de Round Trip y se repitió la prueba variando la carga útil entre 54 bytes y 10000 bytes. La cantidad con las que se obtuvo el promedio fue de 5000 paquetes. La Figura 9 muestra que el tiempo de Roud Trip fue menor para Ipv4 que para Ipv6. Esto es por que al utilizar Ipv6, en la parte del camino que hay configurado un túnel, el overhead generado sobre un paquete Ipv4, en ningún caso será menor que el paquete Ipv4 nativo.

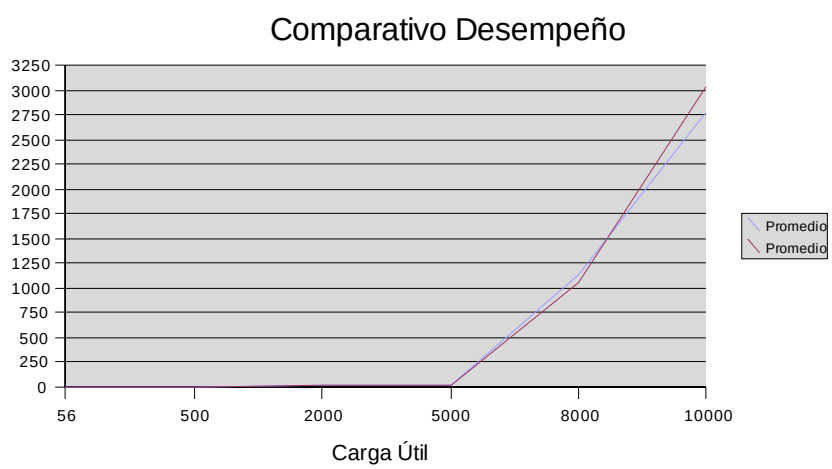

Figura 9: Comparación desempeño

3.3 Implementación de autoconfiguración

Al igual que IPV4, IPV6 posee protocolos que permiten la autoconfiguración de las interfaces de red. Un caso de esto es el protocolo llamado DHCPv6. Este tipo de protocolo se denomina "statefull" [26], debido a que es mantenido por algún servidor dedicado y la información de la configuración asignada a cada cliente queda almacenada en tablas dentro del mismo.
Por otro lado, existe en IPV6 otro protocolo de autoconfiguración sin equivalentes en la versión anterior de IP. Este protocolo es del tipo "stateless" [26], debido a que no existe estado alguno para almacenar. No es necesario instalar servidores para que los nodos puedan configurarse a si mismos.

3.3.1 Implementación de autoconfiguración Statefull usando el protocolo DHCPv6

El protocolo DHCPv6, a diferencia de la versión anterior, no utiliza direcciones broadcast. Usa un rango de direcciones multicast dedicadas exclusivamente para brindar este servicio.
La dirección FF02::1:2 es usada para que un cliente se comunique con sus vecinos dentro de un alcance local. Todos los servidores DHCPv6 o RELAYs [26] deben ser miembros de este grupo multicast. La dirección FF02::5:3 tiene un alcance de Site y es utilizada por lo RELAYs para comunicarse con los servidores DHCPv6 a lo largo de todas las subredes del sitio. Todos los servidores DHCPv6 deben ser miembros de este grupo multicast.
El procedimiento se verificó en el Codarec6, el tráfico generado se muestra en la figura 10

```
16:49:38.359479 IP6 fe80::200:21ff:fec5:3b3b > ff02::16: HBH ICMP6, multicast listener report
v2, 2 group record(s), length 48
16:49:45.611954 IP6 fe80::211:5bff:feb3:7f56.dhcpv6-server > ff02::1:2.dhcpv6-server: dhcp6
solicit
16:49:45.619960 IP6 fe80::200:21ff:fec5:3b3b > ff02::1:ffb3:7f56: ICMP6, neighbor solicitation,
who has fe80::211:5bff:feb3:7f56, length 32
16:49:45.620489 IP6 fe80::211:5bff:feb3:7f56 > fe80::200:21ff:fec5:3b3b: ICMP6, neighbor
advertisement, tgt is fe80::211:5bff:feb3:7f56, length 32
16:49:45.620553 IP6 fe80::200:21ff:fec5:3b3b.dhcpv6-client > fe80::211:5bff:feb3:7f56.dhcpv6-
client: dhcp6 advertise
16:49:45.621415 IP6 fe80::211:5bff:feb3:7f56.dhcpv6-server > ff02::1:2.dhcpv6-server: dhcp6
request
16:49:45.634240 IP6 fe80::200:21ff:fec5:3b3b.dhcpv6-client > fe80::211:5bff:feb3:7f56.dhcpv6-
client: dhcp6 reply
16:49:45.638742 IP6 fe80::211:5bff:feb3:7f56 > ff02::16: HBH ICMP6, multicast listener report
v2, 2 group record(s), length 48
```

Figura 10: Autoconfiguración Statefull

También se observaron mensajes periódicos RENEW desde el cliente, solicitando la renovación de la dirección asignada, a las cuales el servidor DHCPv6 respondió con mensajes REPLY.
Se deshabilitó el servidor, para comprobar el comportamiento del cliente. El mismo continuó enviando periódicamente RENEW. Ante la primer falta de respuesta, envió un REBIND, cumpliendo con lo especificado en el RFC 3397 [27]. Como se muestra en la figura 11.

```
16:50:36.432605 IP6 fe80::211:5bff:feb3:7f56.dhcpv6-server > ff02::1:2.dhcpv6-server: dhcp6
confirm
16:50:36.433202 IP6 fe80::200:21ff:fec5:3b3b.dhcpv6-client > fe80::211:5bff:feb3:7f56.dhcpv6-
client: dhcp6 reply
16:50:46.431284 IP6 fe80::211:5bff:feb3:7f56 > fe80::200:21ff:fec5:3b3b: ICMP6, neighbor
solicitation, who has fe80::200:21ff:fec5:3b3b, length 32
16:50:46.431381 IP6 fe80::200:21ff:fec5:3b3b > fe80::211:5bff:feb3:7f56: ICMP6, neighbor
advertisement, tgt is fe80::200:21ff:fec5:3b3b, length 24
16:50:46.436539 IP6 fe80::211:5bff:feb3:7f56.dhcpv6-server > ff02::1:2.dhcpv6-server: dhcp6
renew
16:50:52.532104 IP6 fe80::200:21ff:fec5:3b3b > ff02::16: HBH ICMP6, multicast listener report
```

```
v2, 2 group record(s), length 48
16:50:56.652421 IP6 fe80::211:5bff:feb3:7f56.dhcpv6-server > ff02::1:2.dhcpv6-server: dhcp6
renew
16:51:16.441329 IP6 fe80::211:5bff:feb3:7f56.dhcpv6-server > ff02::1:2.dhcpv6-server: dhcp6
rebind
16:51:17.288302 IP6 fe80::211:5bff:feb3:7f56.dhcpv6-server > ff02::1:2.dhcpv6-server: dhcp6
renew
16:51:27.350185 IP6 fe80::211:5bff:feb3:7f56.dhcpv6-server > ff02::1:2.dhcpv6-server: dhcp6
rebind
```

Figura 11: Deshabilitación servidor DHCPv6

3.3.2 Implementación de autoconfiguración Stateless

La autoconfiguración stateless se basa en los alcances de las direcciones, un identificador local a partir de una dirección de enlace, y la utilización de dos subprotocolos de ICMPv6, Neighbor Discovery[28] y Multicast Listener Discovery[29].

Cada nodo, cuando se inicia, crea una dirección de link local por cada interfase con que cuenta.

Para construir la dirección se utiliza un prefijo “bien conocido”, el FE80::/64 y un identificador local para los 64 bits restantes. Este último está basado en la dirección MAC de cada interfase, utilizando el algoritmo standard IEEE EUI-64. Antes de asignarla a la interfase, se debe verificar la unicidad de la dirección construida y se envía un mensaje tipo Neighbor Solicitation. La dirección fuente del requerimiento es indefinida (::), y la destino es la que el nodo construyó e intenta utilizar.

Si algún otro nodo ya está usando esa dirección tentativa, responderá con un mensaje Neighbor Advertisement. En ese caso, la dirección no debe asignarse al nodo y se interrumpe el proceso de autoconfiguración. Si ninguno contesta el Neighbor Solicitation, se asigna la dirección a la interfase. Finalmente el nodo intentará autoconfigurar una dirección global, enviando mensajes Router Solicitation. Si existe algún router en el enlace local que publique su prefijo, le contestará con un mensaje Router Advertisement y de esa manera el nodo también tendrá una dirección global.

Para verificar el mecanismo, y utilizando la topología de la figura 2.Se activó el protocolo Router Advertisement en R2 y se capturó el tráfico generado por R1 .Se verificó la autoconfiguración de direcciones local y global en R1.

```
6:23:01.647940 IP6 fe80::200:21ff:fecb:9903 > ff02::1: ICMP6, router advertisement, length 56
16:23:36.774047 IP6 fe80::200:21ff:fecb:9903 > ff02::1: ICMP6, router advertisement, length 56
16:24:15.192180 IP6 fe80::200:21ff:fecb:9903 > ff02::1: ICMP6, router advertisement, length 56
16:24:23.636245 IP6 :: > ff02::16: HBH ICMP6, multicast listener report v2, 1 group record,
length 28
16:24:24.016509 IP6 :: > ff02::1:ff01:e4aa: ICMP6, neighbor solicitation, who has
fe80::2d0:9ff:fe01:e4aa, length 24
16:24:25.015997 IP6 fe80::2d0:9ff:fe01:e4aa > ff02::2: ICMP6, router solicitation, length 16
16:24:25.035488 IP6 fe80::200:21ff:fecb:9903 > ff02::1: ICMP6, router advertisement, length 56
16:24:25.783977 IP6 :: > ff02::1:ff01:e4aa: ICMP6, neighbor solicitation, who has
2002:aad2:ee19:c0da:2d0:9ff:fe01:e4aa, length 24
16:24:41.953914 IP6 fe80::200:21ff:fecb:9903 > ff02::1: ICMP6, router advertisement, length 56
16:25:09.775755 IP6 fe80::200:21ff:fecb:9903 > ff02::1: ICMP6, router advertisement, length 56
```

Figura 12: Autoconfiguración Stateless

Finalmente se configuró en R2 la misma dirección de enlace local de R1, con el objeto de simular una repetición de direcciones de enlace en la misma red local. Se constató que si bien R1 asignó la dirección a su interfase, el nodo queda inalcanzable desde el resto de la red.

3.4 Firewalling y filtrado de paquetes

Como medida de seguridad se configuró en el Servidor “Codarec6” el software “ip6tables” [30] esta herramienta esta integrada al kernel de Linux. Las reglas son heredadas de la versión para ipv4 con varias modificaciones, la desaparición de la tabla NAT (Network Address Translation) y la sentencia “masquerade”, se agregaron las reglas necesarias para las cabeceras opcionales, y el protocolo multipropósito icmpv6.

Ingresando el comando “ip6tables -nv -L” podemos observar las reglas activas y la cantidad de paquetes que cumplen con ellas.

```
[root@codarec6 ~]# ip6tables -nv -L
pkts bytes target prot opt in out source destination
```

```
103K 10M LOG all * * ::/0 ::/0 LOG flags 0 level 7 prefix `INPUT:`
40 2816 ACCEPT all eth+ * ::/0 ::/0 eui64
813 84456 ACCEPT all icmpv6 * * ::/0 ::/0 ipv6-icmp type 128 limit: avg 30/min burst 5
97032 9315K ACCEPT icmpv6 * *  ::/0 ::/0 ipv6-icmp type 134
7 336  ACCEPT icmpv6 * *  ::/0 ::/0 ipv6-icmp type 135
11 760 ACCEPT icmpv6 * *  ::/0 ::/0 ipv6-icmp type 136
0 0 ACCEPT icmpv6 * *  ::/0 ::/0 ipv6-icmp type 4 code 0
0 0 ACCEPT icmpv6 eth+ *  ::/0 ::/0 ipv6-icmp
 812 91880  ACCEPT icmpv6 * *  ::/0 ::/0
3134 333K  ACCEPT tcp * *  ::/0 ::/0 tcp dpt:22
105 11916 ACCEPT tcp * *  ::/0 ::/0 tcp dpt:80
0 0  ACCEPT udp * *  ::/0 ::/0 udp dpt:53
1776 202K DROP all * * ::/0 ::/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
218 37382 ACCEPT all * * 2002:aad2:ee19::/48 ::/0
0 0 ACCEPT all * * 3ffe:38e1:600::/48 ::/0
203 138K ACCEPT all * * ::/0 2002:aad2:ee19::/48
0 0  ACCEPT all * * ::/0 3ffe:38e1:600::/48
0 0 DROP  all * * ::/0 ::/0

Chain OUTPUT (policy ACCEPT 123K packets, 12M bytes)
```

Figura 12: Reglas activas de filtrado

Se pudo comprobar el funcionamiento de las reglas en forma eficiente. Queda a posterior un estudio profundo de cada una de ellas.

3.5 Implementación de diversos servicios

En este apartado se enumeran brevemente el resto de los servicios implementados en el CODAREC6 Test-bed para cumplir con el objetivo planteado y de esta manera generar un ambiente propicio para futuras investigaciones en el área.

Una vez obtenido un pool de direcciones globales IPV6, fue necesario montar un servicio DNS que pudiera mapear direcciones de 128 bits a nombres. Se llevó a cabo utilizando la aplicación BIND versión 9.3 en Codarec6.

Una vez interconectadas las distintas redes mostradas en la Figura 5, se utilizó el protocolo de ruteo RIPv2 para facilitar el mantenimiento de las rutas. El mismo fue implementado en los distintos nodos por medio de Quagga versión 0.98.

Para interactuar con el grupo autónomo de fi-unlp se configuró el protocolo de ruteo BGP4, también utilizando la aplicación Quagga.

Se montaron servicios de HTTP, FTP, SSHD en el servidor Codarec6, para poder brindar servicios a través de Internet, con sus direcciones globales IPV6.

4 CONCLUSIONES

Este trabajo ha mostrado la implementación y puesta en marcha de un laboratorio experimental de IPV6 denominado CODAREC6. Se ha descrito la arquitectura y los elementos constituyentes del "testbed". Se montaron y ensayaron distintas funcionalidades ipv6 con la finalidad de verificar el cumplimiento de las normativas del IETF a través de sus correspondientes RFC. Así también se montó y probó técnicas de ruteo nativo y métodos de transición IPV4 a IPV6. También se ensayaron métodos de autoconfiguración Statefull y Stateless. Se montó un servicio de DNS con soporte a IPV6. También se implementaron protocolos de ruteo RIPv2 y BGP4. Se montaron servicios de HTTP, FTP y SSHD. En todos los casos se realizaron pruebas de verificación y validación y en algunos se realizaron pruebas de desempeño.

En este trabajo no se mostraron algunas características del CODAREC6, por no estar totalmente implementadas y probadas. Nos referimos a funcionalidades de seguridad intrínseca IPSEC, movilidad y calidad de servicio. Estas tareas estarán presentes en próximas mejoras del testbed. Tampoco se ha mencionado en detalle el capacidad del sistema operativo Windows para manejar IPV6, ya que si bien fue probado en modo cliente, aún no se han obtenido resultados definitivos de desempeño cuando se montan servicios o se implementan túneles.

Por último creemos que el CODAREC6 contribuirá en el estudio, el diseño, el desarrollo e

implementación, el ensayo y la capacitación y difusión del protocolo de Internet versión 6.

AGRADECIMIENTOS

Agradecemos la inestimable colaboración de las instituciones y de su personal técnico/científico que permitieron la concreción del CODAREC6; nos referimos a la UTN Fac Regional La Plata, Instituto Padre Vázquez, CONAE Mendoza, OSEP y UTN Fac. Regional Mendoza.

REFERENCIAS

[1] Robert L. Fink, IPv6—What and Where It Is. *The Internet Protocol Journal*, Volume 2, Number 1, March 1999
[2] Douglas E. Comer. *Redes Globales de Información con Internet y TCP/IP*. Pearson PH, Tercera Edición, 1996, ISBN 0-13-219687
[3] S. Deering y R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. *RFC 2460*, December 1998.
[4] 6bone *http://www.6bone.net/old_6bone_home_page.html*
[5] Cooperación Latino Americana de Redes Avanzadas (RED CLARA) *http://www.redclara.net/*
[6] Red Universitaria Nacional de Chile (REUNA) *http://aplicaciones-avanzadas.reuna.cl/*
[7] Corporación Universitaria para el Desarrollo de Internet (CUDI) *http://www.cudi.edu.mx/*
[8] Laboratorio de Ingeniería de Sistemas de Información UTN FRLP *http://www.lines.edu/*
[9] REd TeleINformática Académica.IPv6 (RETINA) *http://www.ipv6.retina.ar/*
[10] IPv6 Operations (v6ops) *http://www.ietf.org/html.charters/v6ops-charter.html*
[11] Mercado, G. Codarec6: An IPv6 Test Bed- Project Plan. PID 25/J042 EIINME317H *UTN FRM* Oct 2004
[12] R.Gilliga, E. Nordmark. Transition Mechanisms for IPv6 Hosts and Routers. *RFC 2893*, 2000.
[13] B.Carpenter, K. Moore Connection of IPv6 Domains via IPv4 Clouds *RFC 3056*, Feb 2001.
[14] R. Gilligan, E. Nordmar. Transition Mechanisms for IPv6 Hosts and Routers. *RFC 1933*. Abril (1996).
[15] RedHat Fedora OS *http://www.redhat.com/fedora/*
[16] S. Thomson, C. Huitema, V. Ksinant, M. Souissi. DNS Extensions to Support IP Version 6. *RFC 3596*, Oct 2003.
[17] R. Droms, Ed. DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6). *RFC 3646*, Dec 2003.
[18] S. Deering. ICMP Router Discovery Messages. *RFC 1256*, Sep 1991.
[19] Quagga Routing Software Suite *http://www.quagga.net.*
[20] R. Droms, Ed. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). *RFC 3315* Jul 2003
[21] S. Thomson, T. Narten. IPv6 Stateless Address Autoconfiguration. *RFC 1791* Ago 1996
[22] J. Postel. Internet Protocol. *RFC 791* Sep 1981.
[23] J. Postel. Internet Control Message Protocol. *RFC 792* Sep 1981.
[24] A. Conta, S. Deering. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. *RFC 2463*, Dec1998.
[25] J. Reynolds, J. Postel. ASSIGNED NUMBERS. *RFC 1700*, Oct 1994
[26] Pete Loshin *Ipv6: Theory, protocol and Practice* Morgan Kaufmann, Segunda Edición, 2004, ISBN 1-55860-810-9
[27] B. Aboba, S. Cheshire. Dynamic Host Configuration Protocol Domain Search Option *RFC 3397*, Nov 2002.
[28] T.Narten, E.Nordmark, W.Simpson. Neighbor Discovery for IP Version 6 *RFC 2462*, Dec 1998
[29] S. Deering, W. Fenner, B. Haberman. Multicast Listener Discovery (MLD) for Ipv6. *RFC 2710*, Oct 1999.
[30] NetFilter *http://www.netfilter.org/projects/iptables/index.html*