Implementación y Evaluación de métodos de Traslación de Protocolos para la transición IPv4-IPv6.

Gustavo Mercado, Cristian Pérez Monte, Carlos Taffernaberry, María Inés Robles, Marcela Orbiscay, Sebastián Tobar, Raúl Moralejo, y Santiago Pérez

Grupo UTN Gridtics - Departamento de Electrónica,
Universidad Tecnológica Nacional – Facultad Regional Mendoza
Mendoza, 5500, Argentina
{gustavo.mercado, cristian.perez, carlos.taffernaberry, ines.robles, marcela.orbiscay, sebastian.tobar, raul.moralejo}@gridtics.frm.utn.edu.ar
http://www.gridtics.frm.utn.edu.ar

Resumen. En la actualidad millones de computadores están interconectados usando Internet Protocol version 4 (IPv4) y es imposible cambiar a la nueva versión, IPv6, en forma simultánea cada uno de ellos. Por tal motivo la IETF ha definido una serie de mecanismos para hacer una transición paulatina y controlada hacia el nuevo protocolo. Por otro lado, en un futuro cercano los proveedores de acceso a Internet (ISP) ya no tendrán nuevas direcciones globales IPv4 que ofrecer a los usuarios finales debido al agotamiento de este tipo de direcciones [1]. Una alternativa muy interesante con la que cuentan los ISP es usar direcciones globales IPv6 y luego, por algún método de transición, acceder al backbone actual de IPv4. El objetivo del presente trabajo es comparar, dos métodos de acceso transparente al backbone de Internet IPv4, desde redes "Solo IPv6". Para hacer la comparativa de ambos mecanismos se desarrolló un software que implementa un ALG (Application Layer Gateway), y se utilizó Ecdysis para implementar NAT64. Ambos ensayos utilizaron una red Test Bed IPv6. En el documento se detallan los principios de diseño y los aspectos fundamentales de la implementación del ALG, como así también la implementación del NAT64. Finalmente se presentan los test realizados y las conclusiones obtenidas sobre la plataforma de prueba.

Palabras Clave: Internet, Protocolo IPv6, Métodos de Transición, ALG, NAT64, ISP.

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, sobre todo en cuanto al paulatino agotamiento de las direcciones IP disponibles, que según las mediciones, en nuestra región, sucederá a mitad de 2014 [1]. 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 disponibles en IPv4 se han vuelto insuficientes [2]. La necesidad de ambientes siempre funcionando ("always-on environments"), ampliarán los requerimientos de direcciones. 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].
Uno de los pasos más importantes en la adopción de IPv6 es la denominada "Transición" del protocolo IPv4 al IPv6. Jordi Palet dice [4] *"Dado que IPv6 es un protocolo nuevo, no es compatible con IPv4, y por ello IPv6 ha sido diseñado previendo un largo período de transición y co-existencia entre ambos".* Si bien para una transición total es necesario que el backbone actual se conecte a IPv6, también es cierto que los usuarios finales y los ISP pueden comenzar a implementar el protocolo. En este trabajo se desarrolla este último concepto, permitiendo que una red final "solo IPv6" se conecte a Internet IPv4 e IPv6 utilizando técnicas de transición. Esto permitirá obtener experiencia y capacitación en la transición IPv4 a IPv6 y por lo tanto estar preparados para tal acontecimiento.
En la siguiente sección se detallan algunos de los mecanismos de transición más utilizados. La sección 3 presenta el escenario y el problema que se intenta resolver por medio de los mecanismos mencionados. La siguiente sección analiza cuál de ellos es el más adecuado para este escenario. A lo largo de la sección 5 se desarrolla y ensaya un ALG. La sección 6 muestra detalles de la implementación de un NAT64, mientras que la siguiente sección hace una evaluación y comparación de ambos métodos. Finalmente con la sección 8 se obtienen valiosas conclusiones.

2 Descripción de mecanismos de transición

El soporte de IPv6 está ahora extensamente disponible para la mayoría de los sistemas operativos de hosts y de routers. Si se desea tener comunicación con otros sistemas IPv6, es vital acceder a la Internet global IPv6 pero los despliegues de redes solo-IPv6 no son muy comunes. La realidad práctica muestra un estado de transición intermedio dónde co-existen IPv4 e IPv6. La expansión de la funcionalidad de IPv6 desde una pequeña infraestructura a una red grande puede ser una aventura compleja y difícil. Para un sitio grande diferentes requisitos y condiciones hacen necesario emplear varios mecanismos de transición según las peculiaridades de, por ejemplo, una subred dada, ambiente inalámbrico o móvil, una tecnología dial-in, etc. [5].
Dos métodos ampliamente usados son el "mecanismo de pila dual" y las "técnicas de tunelling", pero en este trabajo implementaremos y evaluaremos los métodos de "traslación" y de ellos haremos una somera introducción en los párrafos siguientes.

2.1 Traslación de protocolos

Los métodos de traslación fueron desarrollados para lograr la comunicación entre hosts solo IPv4 y hosts solo IPv6. Entre ellos encontramos:
- SIIT (Stateless Ip/Icmp Traslator) [6] y NAT-PT (Network Address Traslation – Protocol Traslation) [7] son mecanismos, a diferencia de los túneles, que traducen encabezados de IPv4 a IPv6 y viceversa. Estas técnicas corpanten los problemas de NAT y deben lidiar con la semántica de convertir correctamente los campos. En algunos casos en el proceso de conversión se pierde información de encabezado. Por esta razón el IETF recomienda estos métodos solo como último recurso.
- BIS (Bump In the stack) [8] es una aproximación similar a la anterior SIIT, pero implementada directamente en el sistema operativo de cada host (entre el módulo de TCP/IP y el driver de placa de red), aunque solo disponible para aplicaciones IPv4 y redes IPv6. Es complejo implementarlo y muy poco utilizado.
- BIA (Bump in the API) [9] agrega una API de traslación entre la API de Socket y el

stack TCP/IP, permitiendo una mejora al método BIS en cuanto a la dependencia del driver de red, pero tiene las mismas limitaciones que BIS.

- TRT (Transport Relay Traslator) [10] es una conversión de protocolos en la capa de transporte que usa como pieza fundamental un DNS proxy. Este recibe consultas de los hosts IPv6 y si el nombre requerido está asociado a direcciones IPv4, devuelve una dirección IPv6 con el formato *prefijoIPv6 (64 bits) + ceros (32 bits) + direcciónIPv4 (32 bits)*. Este método fue reemplazado por NAT64.

- NAT 64 [11] consta de un servidor con al menos una IPv4 global y un segmento de direcciones IPv6 con un prefijo /32 (por ejemplo 64:ff9b::/96). El cliente IPv6 construye la dirección IPv6 destino utilizando el rango anterior de 96 bits más los 32 bits de la dirección IPv4 con la que desea comunicarse, enviando los paquetes a la dirección resultante. El servidor NAT64 crea entonces un mapeo de NAT entre la dirección IPv6 y la dirección IPv4, permitiendo la comunicación. También es necesario el uso de DNS 64.

- DNS 64: [12] Cuando un servidor DNS con funcionalidad DNS 64 recibe una petición de dominio por registro AAAA, pero sólo dispone registros A, crea registros AAAA a partir de los registros A. La primera porción de la dirección IPv6 creada apunta a un traductor IPv6/IPv4, y la segunda incluye la dirección IPv4 del registro A. El traductor en cuestión suele ser un servidor NAT64.

- ALG (Application Layer Gateway) es una traslación realizada en la capa de aplicación. No hay RFC específicos a seguir, pues su implementación depende del protocolo de capa aplicación al que se dará soporte.

3 Escenario planteado

Se muestra en la figura 1 el escenario creado para evaluar las alternativas de transición disponibles.

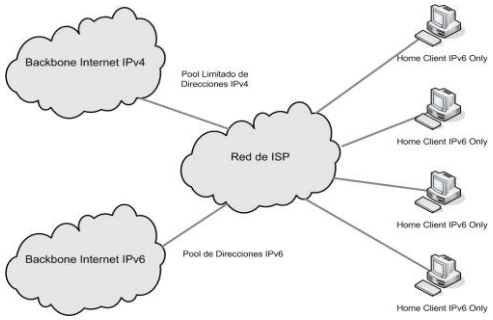

Fig. 1. Escenario común para los ISP en la actualidad

La topología propuesta está compuesta por hosts “home clients” que constituyen una red de clientes de un ISP configurados utilizando IPv6 nativo. El ISP tiene conectividad tanto IPv4 como IPv6. El objetivo es permitir que los home clients puedan acceder a servidores y servicios disponibles en InternetV4 sin que sea necesario hacer modificaciones, ya sea instalando pila dual, configurando túneles o haciendo traslación de protocolos. Se aclara que el ISP no tiene nueva numeración IPv4 por lo que solo puede instalar en direcciones IPv6 en sus clientes.

4 Evaluación de distintos métodos de transición para el escenario dado

Para alcanzar el objetivo se debe implementar alguna de las técnicas enumeradas en el apartado 2.3 pues la comunicación es exclusivamente entre hosts solo IPv4 y hosts solo IPv6, descartándose por esta causa las técnicas de pila dual o tunneling.
A continuación se analizan las alternativas existentes para realizar Traslación de protocolos:

- La aplicación de SIIT y NAT-PT es descartada, debido a los problemas normales de NAT y la posible pérdida de información de encabezado [13].
- Para usar BIS o BIA es necesario modificar los sistemas operativos de los hosts clientes. Se encontrarán problemas para los sistemas operativos que no dispongan del código fuente.
- La alternativa de un TRT es factible, pero es obsoleta.
- NAT64 es bastante usada, encontrando inclusive alguna implementación libre disponible para su ensayo. Como inconveniente requiere un DNS Proxy (DNS64) especialmente configurado para funcionar correctamente.
- La implementación de un ALG también es viable, si no se tiene en cuenta la disminución de performance por hacer toda la conversión en la capa de aplicación.

De los aspectos considerados, se optó por seleccionar NAT64 y ALG para hacer la evaluación de funcionalidades y performance [14].

5 Application Layer Gateway

Se decidió implementar un ALG para protocolo HTTP/HTTPS. Justifica esta decisión la facilidad de implementación y no ser necesarios elementos adicionales, como un DNS Proxy o código fuente del Sistema Operativo. Debido a que casi no existe diferencia entre un ALG y un proxy de aplicación, se intentó inicialmente utilizar el conocido y utilizado proxy HTTP/HTTPS Squid. Pero al momento no tiene soporte IPv6, por lo que finalmente se decidió realizar una aplicación propia para cumplir el objetivo.

5.1 Propuesta de funcionamiento

Se propone implementar la alternativa de los métodos anteriores como se muestra en la Figura 2.

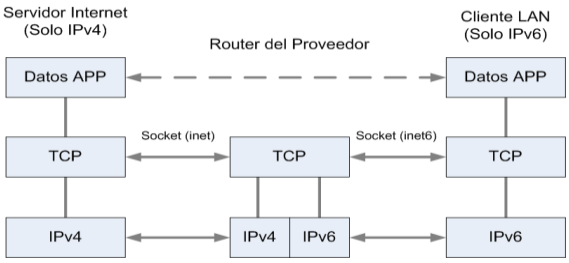

Fig. 2. Diagrama que detalla la técnica de ALG .

La idea básica de la aplicación ALG es permitir que el router del proveedor sea el encargado de intercambiar la información entre los dos extremos. Para ello es necesario que el mismo tenga dual stack y ejecute la aplicación ALG.

Los clientes locales iniciarán la comunicación utilizando un socket INET6, y hacen una solicitud al ALG, que será almacenada en un buffer. El ALG, utilizando un socket INET, iniciará una nueva conexión como cliente al servicio InternetV4 solicitado por el cliente local y se reenvían los datos del requerimiento que almacenó previamente. La respuesta del servicio solicitado, será reenviado por el ALG al cliente local, utilizando IPv6. La aplicación debe resolver el nombre del dominio solicitado, antes de enviar el requerimiento hacia InternetV4.

5.2 Implementación

Se realizó un prototipo, para poder evaluar el correcto funcionamiento de este mecanismo. La programación fue hecha en Python. Se muestran a continuación las porciones de código más relevantes:

```
  #Bucle principal
def listen (self):
    escucha = socket(AF_INET6,SOCK_STREAM)#IPv6 Only
    escucha.bind(self.ADDR6,self.PORT)
    escucha.listen(10)        #hasta 10 a la espera
    while True:
        interno,cliente = escucha.accept()
        pid = os.fork()
        if pid != 0 :         #proceso hijo
            self.servicio()
        else:                 #proceso padre
            interno.close()

def servicio (self):
    Pedido = interno.recv(self.buffer)
    externo = socket(AF_INET,SOCK_STREAM)# a InternetV4
    externo.connect(res[0][4][:2])
    externo.send(Pedido) #reenvio requerimiento
    RespInternet = ''
    while RespInternet <> ''  #lee IPv4 -> escribe IPv6
        RespInternet = externo.recv(self.buffer)
        interno.send(RespInternet)
    interno.close()           #Termino el envio de IPV4
    externo.close()
    sys.exit()
```

La función “listen”, crea un socket de la familia AF_INET6 (IPv6) y espera que un home client se conecte mediante el método escucha.accept(). Una vez conectado, mediante la llamada os.fork(), crea un hijo que atiende a cada uno de los clientes, usando el método self.servicio(). La función “servicio” guarda en una variable local, Pedido, el requerimiento original del cliente. Luego, usando la API de socket, crea un socket de la familia AF_INET (IPv4) y se conecta como cliente con el servidor al que tiene que hacer el requerimiento, mediante la llamada externo.send(Pedido). Luego obtiene la respuesta con el método externo.recv(self.buffer) y lo reenvía usando el socket IPv6 al home client original. El router en el que se ensayó el método fue host con GNU/Linux distribución Ubuntu 9.04. Se eligió a Windows Xp como home client con soporte solo IPv6, por ser el sistema operativo mas difundido, sin embargo se puede usar otros sistemas operativos como GNU/Linux, Solaris, Mac Os o Win Vista. También se hicieron pruebas exitosas con un teléfono celular Nokia modelo N95 con sistema operativo Symbian. En todos los casos se desactivó el stack IPv4. Se configuró la aplicación cliente HTTP (navegador) para que utilice como proxy la dirección IPv6 del router local donde se ejecutó el ALG.

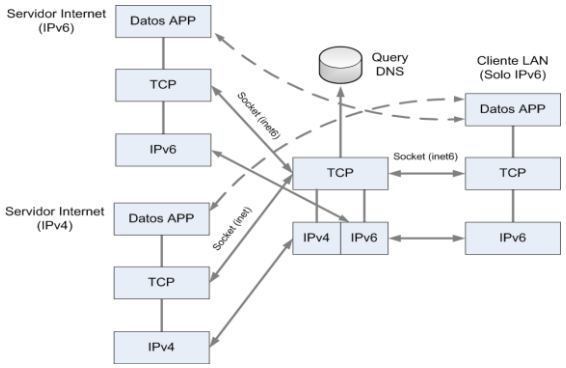

Fig. 3. Diagrama final del ALG

6 Implementación de NAT64

Para la implementación de NAT64 se instaló en el router de proveedor dual stack con un enlace WAN que permitió acceder a Internetv4 e Internetv6. Además se configuraron dos enlaces LAN, el primero con conectividad IPv4 e IPv6 hacia el servidor que realiza la tarea NAT64+DNS64. El segundo enlace LAN hacia una red IPv6 only, en la cual se ubicaron los home clients "solo IPv6". Se configuró una ruta por defecto al servidor NAT64+DNS64 para la red /96 asignada al NAT64.

En el servidor NAT64+DNS64 se utilizó como sistema operativo Linux Fedora 14 al cual se le instaló el paquete ecdysis-nf-nat64 [15] correspondiente a NAT64 y ecdysis-unbound que implementa un servidor DNS64. Por problemas en el servidor NAT64+DNS64, se deshabilitó la configuración automática y se asignaron direcciones IPv4 e IPv6 estáticas, las ruta por defecto del router y la ruta por defecto de la interfaz nat64 para la red asignada al NAT64. Finalmente, a los clientes se les asignó por autoconfiguración sus direcciones IPv6 y ruta por defecto y por medio de DHCPv6, el servidor DNS que corresponde a la IPv6 del servidor NAT64+DNS64. En la figura 4, se muestra la topología de la implementación realizada.

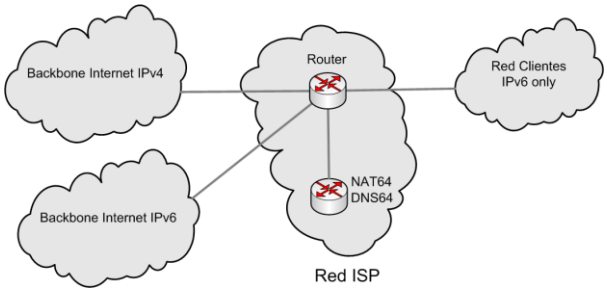

Fig. 4. Topología de NAT64

7 Evaluación de métodos de translación

Lo importante, para los clientes, es la capacidad de que los servicios de Internet, mediados por el ISP, sean accesibles, transparentes y seguros. Es evidente que los clientes "sólo IPv6" deben tener, al menos, la misma funcionalidad que los IPv4. La evaluación, de los mecanismos examinados en este trabajo, está orientada a verificar si la mayoría de los servicios presentan una correcta funcionalidad y performance.

7.1 Evalución de ALG

En primer lugar se verificó la validez del método capturando el tráfico. En la Figura 5 se muestran las trazas capturadas tanto en la red LAN IPv6 como en el acceso a IPv4.

```
 1  fe80::16fc:eeff:fe7f:a2ff -> ff02::1      ICMPv6 Router advertisement
 2  2001:1938:110:23:213:d3ff:fe78:c33d -> 2001:1938:110:23:21b:9eff:fe2d:668 TCP 1093 > 8080 [SYN] Seq=0 Win=16384 Len=0 MSS=1440
 3  2001:1938:110:23:21b:9eff:fe2d:668 -> ff02::1:ff78:c33d ICMPv6 Neighbor solicitation
 4  2001:1938:110:23:213:d3ff:fe78:c33d -> 2001:1938:110:23:21b:9eff:fe2d:668 ICMPv6 Neighbor advertisement
 5  2001:1938:110:23:21b:9eff:fe2d:668 -> 2001:1938:110:23:213:d3ff:fe78:c33d TCP 8080 > 1093 [SYN, ACK] Seq=0 Ack=1 Win=5760 Len=0
 6  2001:1938:110:23:213:d3ff:fe78:c33d -> 2001:1938:110:23:21b:9eff:fe2d:668 TCP 1093 > 8080 [ACK] Seq=1 Ack=1 Win=17280 Len=0
 7  2001:1938:110:23:213:d3ff:fe78:c33d -> 2001:1938:110:23:21b:9eff:fe2d:668 HTTP GET
http://sitecheck2.opera.com/?host=www.altavista.com&hdn=nubrKnkzLB7qxAS86abtMw== HTTP/1.0
 8  2001:1938:110:23:21b:9eff:fe2d:668 -> 2001:1938:110:23:213:d3ff:fe78:c33d TCP 8080 > 1093 [ACK] Seq=1 Ack=498 Win=6432 Len=0
 9  2001:1938:110:23:213:d3ff:fe78:c33d -> 2001:1938:110:23:21b:9eff:fe2d:668 HTTP GET http://www.altavista.com/ HTTP/1.0
10  2001:1938:110:23:21b:9eff:fe2d:668 -> 2001:1938:110:23:213:d3ff:fe78:c33d TCP 8080 > 1094 [ACK] Seq=1 Ack=539 Win=6456 Len=0
11  192.168.1.223 -> 192.168.1.1  DNS Standard query AAAA www.altavista.com
12  192.168.1.223 -> 192.168.1.1  DNS Standard query AAAA sitecheck2.opera.com
13  192.168.1.1 -> 192.168.1.223 DNS Standard query response CNAME avatw.search.a00.yahoodns.net
14  192.168.1.223 -> 192.168.1.1  DNS Standard query A www.altavista.com
15  192.168.1.1 -> 192.168.1.223 DNS Standard query response CNAME avatw.search.a00.yahoodns.net A 72.30.186.25
16  192.168.1.223 -> 72.30.186.25 TCP 43019 > 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=2203659 TSER=0 WS=6
17  1.731010 72.30.186.25 -> 192.168.1.223 TCP 80 > 43019 [SYN, ACK] Seq=0 Ack=1 Win=8712 Len=0 MSS=1452 WS=0 TSER=2203659
18  1.731031 192.168.1.223 -> 72.30.186.25 TCP 43019 > 80 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSV=2203661 TSER=3240479415
19  1.731092 192.168.1.223 -> 72.30.186.25 HTTP GET http://www.altavista.com/ HTTP/1.0
20  1.749107 72.30.186.25 -> 192.168.1.223 TCP 80 > 43019 [ACK] Seq=1 Ack=539 Win=15846 Len=0 TSV=3240479417 TSER=2203661
21  2.188310 72.30.186.25 -> 192.168.1.223 HTTP HTTP/1.0 200 OK  (text/html)
22  2.188401 192.168.1.223 -> 72.30.186.25 TCP 43019 > 80 [ACK] Seq=539 Ack=1441 Win=8768 Len=0 TSV=2203775 TSER=3240479460
23  2.188462 2001:1938:110:23:21b:9eff:fe2d:668 -> 2001:1938:110:23:213:d3ff:fe78:c33d HTTP HTTP/1.0 200 OK  (text/html)
24  2.198668 2001:1938:110:23:213:d3ff:fe78:c33d -> 2001:1938:110:23:21b:9eff:fe2d:668 TCP 1096 > 8080 [SYN] Seq=0 Win=16384 Len=0
25  2.198693 2001:1938:110:23:21b:9eff:fe2d:668 -> 2001:1938:110:23:213:d3ff:fe78:c33d TCP 8080 > 1096 [SYN, ACK] Seq=0 Ack=1 Win=5760
```

Fig. 5. Captura de tráfico existente mediante el uso del ALG

Posteriormente se realizaron mediciones de tiempos de conexión y acceso total a distintos sitios IPv4 por medio de Apache Benchmark [16]. Finalmente se hicieron evaluaciones funcionales, teniendo presente que este método sólo permite hacer traslación de protocolo HTTP/HTTPS. Se logró utilizar exitosamente incluso en sistemas operativos relativamente antiguos, como el caso de Windows XP. Para ello se configuró el nombre del router como proxy en el cliente HTTP (navegador) y se agregó en el archivo hosts de Win XP la dirección IPv6 del router. También, ALG, operó correctamente en el sistema operativo para celulares Symbian y en todos los sistemas operativos que prefieren IPv4 a IPv6 en la navegación.

7.2 Evaluación de NAT64

El desempeño de NAT64 fue satisfactorio como solución para la conectividad IPv4 a redes "IPv6 only" siempre y cuando el dispositivo NAT64 se encuentre ubicado en redes próximas a las redes de servicio. Se puede observar una compatibilidad prácticamente total con todos los protocolos de capa de aplicación basados en TCP, UDP o ICMP. Para evaluar el desempeño se realizaron mediciones de tiempos de conexión y acceso total a distintos sitios IPv4. Sin embargo, respecto a la evaluación funcional, al analizar el NAT64 con un rango de dirección de traslación público y al utilizarlo remotamente a través de Internet IPv6 pudo observarse numerosos problemas a la hora de acceder a determinados servidores HTTP IPv4. En el caso específico del acceso a otros servicios como por ejemplo SSH no se observó ningún

problema. Pudo encontrarse algunas incompatibilidades en los hosts de las redes "IPv6 only" que impidieron su uso cuando los mismos poseían sistemas operativos relativamente antiguos, como el caso de Win XP. El problema se manifiesta debido a la incapacidad de Win XP para realizar consultas DNS a través de IPv6. Pudo observarse en sistemas operativos nuevos tales como Windows 7 y últimas versiones de Linux una completa compatibilidad y una configuración automática y transparente para el usuario final. En el caso de versiones de Linux menos actualizadas se requiere únicamente la configuración del servidor DNS64 en el archivo de configuración.
En cuanto a la comunicación con sitios IPv6, la misma es directa sin la intervención de ningún dispositivo intermedio lo cual le impone una ventaja con otros métodos. En la Figura 6 se muestran las trazas capturadas con trafico NAT64 con acceso a IPv4.

```
  1   0.000000 2001:1938:110:23:32::4 -> 2001:1291:217:23:250:56ff:feae:27 DNS Standard query AAAA
www.yahoo.com
  2   0.999437 2001:1938:110:23:32::4 -> 2001:1291:217:23:250:56ff:feae:27 DNS Standard query AAAA
www.yahoo.com
  3   1.532224 2001:1938:110:23:32::4 -> 2001:1291:217:64:9b:0:43c3:a04c TCP 54826 > 80 [SYN] Seq=0
Win=8192 Len=0 MSS=1440 WS=2
  4   2.346770 2001:1291:217:64:9b:0:43c3:a04c -> 2001:1938:110:23:32::4 TCP 80 > 54826 [SYN, ACK] Seq=0
Ack=1 Win=5840 Len=0 MSS=1440 WS=8
  5   2.346898 2001:1938:110:23:32::4 -> 2001:1291:217:64:9b:0:43c3:a04c TCP 54826 > 80 [ACK] Seq=1
Ack=1 Win=17280 Len=0
  6   2.347022 2001:1938:110:23:32::4 -> 2001:1291:217:64:9b:0:43c3:a04c HTTP GET / HTTP/1.1
  7   3.151339 2001:1291:217:64:9b:0:43c3:a04c -> 2001:1938:110:23:32::4 TCP 80 > 54826 [ACK] Seq=1
Ack=602 Win=7168 Len=0
  8   3.159450 2001:1291:217:64:9b:0:43c3:a04c -> 2001:1938:110:23:32::4 HTTP HTTP/1.1 302 Found
(text/html)
  9   3.162587 2001:1938:110:23:32::4 -> 2001:1291:217:23:250:56ff:feae:27 DNS Standard query AAAA
ar.yahoo.com
 10   3.359405 2001:1938:110:23:32::4 -> 2001:1291:217:64:9b:0:43c3:a04c TCP 54826 > 80 [ACK] Seq=602
Ack=666 Win=16612 Len=0
 11   3.993892 2001:1938:110:23:32::4 -> 2001:1291:217:64:9b:0:43c3:a04c TCP 54827 > 80 [SYN] Seq=0
Win=8192 Len=0 MSS=1440 WS=2
 12   4.958166 2001:1291:217:64:9b:0:43c3:a04c -> 2001:1938:110:23:32::4 TCP 80 > 54827 [SYN, ACK] Seq=0
Ack=1 Win=5840 Len=0 MSS=1440 WS=8
 13   4.958303 2001:1938:110:23:32::4 -> 2001:1291:217:64:9b:0:43c3:a04c TCP 54827 > 80 [ACK] Seq=1
Ack=1 Win=17280 Len=0
 14   4.958430 2001:1938:110:23:32::4 -> 2001:1291:217:64:9b:0:43c3:a04c HTTP GET /?p=us HTTP/1.1
 15   5.654285 2001:1291:217:64:9b:0:43c3:a04c -> 2001:1938:110:23:32::4 TCP 80 > 54827 [ACK] Seq=1
Ack=774 Win=7424 Len=0
 16   6.286348 2001:1291:217:64:9b:0:43c3:a04c -> 2001:1938:110:23:32::4 TCP [TCP segment of a
reassembled PDU]
 17   6.485549 2001:1938:110:23:32::4 -> 2001:1291:217:64:9b:0:43c3:a04c TCP 54827 > 80 [ACK] Seq=774
Ack=537 Win=16744 Len=0
 18   6.528961 2001:1938:110:23:32::4 -> 2001:1938:110:23::1 ICMPv6 Neighboradvertisement
```

Fig. 6. Captura de tráfico existente al utilizar el NAT64

7.3 Comparación ALG- NAT64 en acceso a servidores IPv4

7.3.1 Comparación funcional

En la siguiente tabla, se muestran los resultados de los parámetros evaluados aplicados a ALG y NAT64/DNS64, en una escala de cuatro niveles:

- Inexistente - Bajo - Medio –Alto

Parámetro Evaluado	ALG	NAT64/DNS64
Complejidad Instalación del servicio	Bajo	Media
Mantenibilidad	Media	Media
Performance en tiempo de respuesta	Alta	Media
Problemas acceso	Inexistente	Bajo
Cantidad de protocolos	Bajo	Alto
Escalabilidad	Media	Alta
Integración de Seguridad	No probada	No probada

Latencia	Baja	Media
Complejidad instalación de los hosts	Media	Inexistente o Baja
Compatibilidad de los sistemas operativos de los nodos clientes	Alta	Media

7.3.2 Comparación en el desempeño

Para las pruebas de desempeño se realizaron mediciones de tiempos de conexión y acceso total a distintos sitios Ipv4 (www.google.com, www.mit.edu), por medio de Apache Benchmark, haciendo 100 requerimientos con una concurrencia de 10 a cada uno de ellos. Ver Figuras 7 y 8. De la pruebas de desempeño de ambos métodos se presentan gráficos indicando los tiempos mínimos, medios y máximos para establecer la conexión (ALG conn y NAT64 conn) y para completar el requerimiento (ALG total y NAT64 total). Cabe aclarar que solo se compararon los protocolos HTTP/HTTPS y únicamente hacia servidores IPv4. Realizar las pruebas hacia servidores IPv6 escapa al objetivo del presente trabajo, ya que NAT64 no interviene en el mismo. ALG si lo hace por lo que la performance sería levemente inferior en el segundo caso debido al agregado del software intermedio.

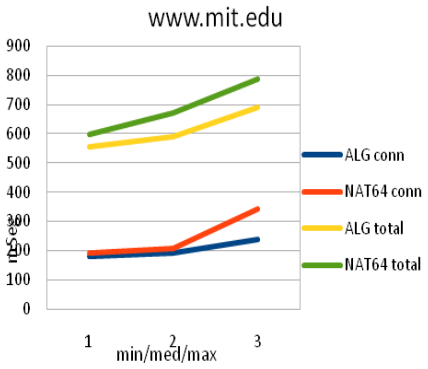

Fig. 7

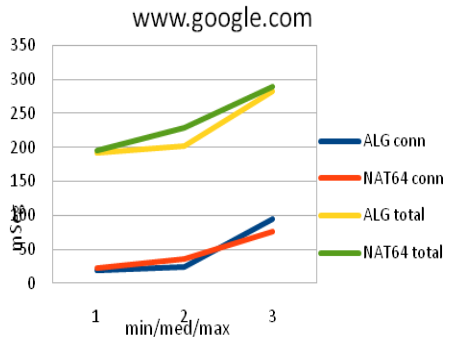

Fig. 8

8 Conclusiones

El presente trabajo pretende ser una herramienta adicional para que los ISPs evalúen las distintas alternativas a la hora de hacer la transición.

A medida que los ISP no obtengan nuevas direcciones IPv4, ambas técnicas se pueden comenzar a implantar gradualmente en un grupo reducido de sus clientes, a modo de evaluación. Esto será transparente para el resto de sus clientes y permitirá hacer todos los ajustes necesarios para un correcto despliegue.

De las comparativas funcionales realizadas, se determina que el método ALG es indicado cuando los "home clients" solo acceden a la Internet para usar HTTP/HTTPS. NAT64+DNS64 lo aventaja en cuanto a la cantidad de protocolos soportados. Por otro lado, se observó que ALG es un complemento perfecto de

NAT64+DNS64 en los casos que los hosts posean sistemas operativos como Windows XP o Symbian, pues éstos priorizan para las consultas DNS el registro A sobre el AAAA. Al utilizar ALG para la navegación en servidores IPv4 desde redes "IPv6 only", se salva la imposibilidad de resolver nombres utilizando IPv6.
Se prevé realizar futuros trabajos de implementación y comparación de modelos de transición IPv4/IPv6.
Por último creemos que el presente trabajo y el trabajo que lleva a cabo el grupo GridTics, contribuye a la capacitación, difusión y formación de recursos humanos para afrontar el inminente cambio al protocolo de Internet versión 6 en la región.

Referencias

1 Informe LACNIC, "Distribuciones/Asignaciones IPv4, espacio disponible y pronósticos (Informe Marzo 2011 - Actualizado Abril 2011)", http://www.lacnic.net/sp/registro/espacio-disponible-ipv4.html, [última visita: 14/07/2011]

2 Robert L. Fink "IPv6—What and Where It Is", The Internet Protocol Journal, Volume 2, Number 1, March 1999

3 S. Deering y R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, December 1998

4 J. Palet, "Cómo es la transición?", http://portalipv6.lacnic.net/es/ipv6/novedades/ como-es-la-transición [última visita: 14/07/2011]

5 T. Chown, "IPv6 Campus Transition Scenario Description and Analysis", Internet-Draft, March 2007

6 E. Nordmark Stateless IP/ICMP Translation Algorithm (SIIT) RFC 2765, February 2000

7 G. Tsirtsis, P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)",RFC 2766, February 2000

8 K. Tsuchiya, H. Higuchi, Y. Atarashi, "Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS)", RFC 2767, February 2000

9 S. Lee, M-K. Shin, Y-J. Kim, E. Nordmark, A. Duran, "Dual Stack Hosts Using "Bump-in-the-API" (BIA)", RFC 3338, October 2002

10 J. Hagino, K. Yamamoto, "An IPv6-to-IPv4 Transport Relay Translator", RFC 3142, June 2001

11 M. Bagnulo, P. Matthews, I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011

12 M. Bagnulo, A. Sullivan, P. Matthews, I. van Beijnum," DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011

13 C. Aoun, E. Davies, "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status", RFC 4966, July 2007

14 C. Bao, C. Huitema, M. Bagnulo, M. Boucadair, X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010

15 Ecdysis: open-source implementation of a NAT64 gateway - http://ecdysis.viagenie.ca/[última visita: 14/07/2011]

16 Apache HTTP server benchmarking tool - http://httpd.apache.org/docs/2.0/ programs/ab.html [última visita: 14/07/2011]