Sistemas & Telemática
ISSN: 1692-5238
EditorSyT@icesi.edu.co
Universidad ICESI
Colombia

Martinez Alayón, Carlos Andrés; Ferro Escobar, Roberto; Arrieta Zambrano, Víctor José
Implementation of transition and coexistence mechanisms for IPV4-IPV6 protocols in computer centers on supported high performance academic networks
Sistemas & Telemática, vol. 13, núm. 34, junio, 2015, pp. 83-106
Universidad ICESI
Cali, Colombia

Original Research / Articulo original - Tipo 1

Implementation of transition and coexistence mechanisms for IPV4-IPV6 protocols in computer centers on supported high performance academic networks

Carlos Andrés Martinez Alayón, M.Sc. (c) / camartineza@correo.udistrital.edu.co

Roberto Ferro Escobar, Ph.D. / rferro@udistrital.edu.co

Víctor José Arrieta Zambrano / vjarrietaz@correo.udistrital.edu.co

Grupo de Investigación L.I.D.E.R - Universidad Distrital Francisco José de Caldas

ABSTRACT This document aims to contextualize the reader about some of the mechanisms that currently exist for IPv4-IPv6 transition and evidence some aspects that must be taken into account when evaluating and implementing some of them, specifically in centers of high performance computing and academic networks to support research projects. It also aims to show the implementation and support of IPv6 in e-learning technology platforms.

KEYWORDS IPv6 transition; Dual Stack; tunneling; ISATAP; 6to4; Teredo; SIIT.

Implementación de mecanismos de transición y coexistencia para los protocolos IPV4-IPV6 en centros de computación de alto desempeño soportados sobre redes académicas

RESUMEN El presente documento pretende contextualizar al lector sobre algunos de los mecanismos que existen para la transición de IPv4-IPv6 y evidencia algunos aspectos que se deben tener en cuenta al momento de evaluar e implementar algunos de ellos, específicamente en centros de computación de alto desempeño y en redes académicas para el apoyo de proyectos de investigación. También se pretende mostrar la implementación y soporte de IPv6 en plataformas tecnológicas e-learning

PALABRAS CLAVE Transición IPv6, Doble Pila, túneles, ISATAP, 6to4, Teredo, SIIT

Implementação de mecanismos de transição e coexistência dos protocolos IPV4-IPV6 nos centros de computação de alto desempenho suportados pelas redes acadêmicas

RESUMO Este documento tem como objetivo contextualizar o leitor sobre alguns dos mecanismos que existem para a transição do IPv4 para o IPv6 e evidenciar alguns aspectos que devem ser considerados na avaliação e implementação de qualquer um deles, especificamente nos centros de computação de alto desempenho e redes acadêmicas para apoiar projetos de pesquisa. Ainda se pretende mostrar a implementação e o suporte de IPv6 em plataformas tecnológicas e-learning.

PALAVRAS-CHAVE Transição IPV6, Pilha Dupla, túneis, ISATAP, 6to4, Teredo, SIIT.

Received / Recepción: Abril 22, 2015 - Accepted / Aceptación: Junio 17, 2015 *doi: 10.18046/syt.v13i34.2094*

I. Introduction

The first version of the Internet protocol widely used, searching for unique plus global addressing and guaranteeing identification between two network devices, was IPv4. Due to fast growth in devices accessing the Internet, an enormous quantity of addresses are required. Since February 2011, the IPv4 central addresses stock, managed by the Internet Assigned Numbers Authority [IANA], has depleted. At that time, when only five /8 blocks were available, were delivered to each one of the five Regional Internet Registry [RIR] around the world. In June of 2014, the Latin America & Caribbean Network Information Centre [LACNIC], responsible for assigning resources in this region, announced depletion of IPv4 addresses and restrictive policies for delivering Internet resources in the region started to be applied (LACNIC, 2014).

In order to surpass the Internet Protocol [IP] current limitations related with number of addresses; routing; and security, a new version of IP was designed by the Internet Engineering Task Force [IETF]: IPv6 (Deering & Hinden, 1998).

The main purpose in designing this new Internet Protocol was to increment a number of addresses. IPv6 addresses were designed using a 128 bits (16 bytes) addressing scheme; compared with IPv4 data (32 bits or 4 bytes), the increase is considerable. This means that, in IPv6, near $3.4x10^{38}$ addresses are possible; compared with $4x10^{9}$ addresses in IPv4 (Deering & Hinden, 1998). Making an analogy, the total surface area on Earth, including oceans, is 510,072,000 km^2; i.e. in squared millimeters this equals approximately $5.1x10^{20}$ mm^2 (Pidwirny, 2006). Hence, an interesting conclusion of this fact is that, from each squared millimeter on Earth's surface, it is possible to "assign" an approximately value of $6.66x10^{17}$ IPv6 addresses, enough to proportionate an address to each connectable device (cell phone, computer, tablet, mp3 player, car, etc.) on the planet. This eliminates the prerequisite of using Network Address Translators [NAT] (Egevang & Francis, 1994), one of actual mechanisms used to share addresses. Additionally, IPv6 is designed to support IPsec, offer scalability and robust multimedia transmissions. In general terms, IPv6 was carefully conceived and designed thinking of future applications.

It is predicted that the transition between IPv4 towards IPv6 will not occur immediately; conversely, this transi-

I. Introducción

La primera versión del protocolo de Internet que fue ampliamente utilizado con el fin de proveer un direccionamiento único global y asegurar que dos dispositivos de red se identifiquen entre ellos fue el IPv4. Debido al rápido crecimiento de la red, así como de los dispositivos con acceso a Internet, actualmente es necesaria una gran cantidad de direcciones. Desde febrero de 2011 el stock central de direcciones IPv4 administrado por la Internet Assigned Numbers Authority [IANA] quedó finalmente agotado. En ese momento, al quedar disponibles sólo cinco bloques /8, se hizo entrega de ellos a cada uno de los cinco Regional Internet Registry [RIR] en el mundo. En junio de 2014 el Registro de Direcciones de Internet para América Latina y el Caribe [LACNIC], responsable de la asignación de recursos para esta región, anunció el agotamiento del stock de direcciones IPv4 y empezó a regir políticas restrictivas para la entrega de recursos de Internet en el continente (LACNIC, 2014).

Con el fin de superar las limitaciones del protocolo Internet [IP] actual, relacionadas con la cantidad de direcciones, el enrutamiento y la seguridad, la Internet Engineering Task Force [IETF] diseño una nueva versión, el IPv6 (Deering & Hinden, 1998). El propósito principal del diseño del nuevo IP fue incrementar el número de direcciones. La dirección IPv6 ha sido diseñada con un esquema de direccionamiento de 128 bits (16 bytes), en lugar de los 32 bits (4 bytes) en IPv4. Lo anterior significa que en IPv6 se pueden tener cerca de $3.4x10^{38}$ posibles direcciones diferentes, un número significativamente mayor que el $4.3x10^{9}$ de IPv4 (Deering & Hinden, 1998). El área de la superficie terrestre, incluyendo los océanos, es de 510 072 000 km^2, en milímetros aproximadamente $5.1x10^{20}$ mm^2 (Pidwirny, 2006). De lo anterior se puede concluir que a cada milímetro cuadrado de superficie terrestre le corresponderían cerca de $6.66x10^{17}$ (666 mil billones) direcciones, suficiente para darle a cada dispositivo (teléfono móvil, computador, reproductor mp3, automóvil, etc.) en la superficie de la tierra su propia dirección IP, lo cual elimina la necesidad del uso de *Network Address Translators* [NAT] (Egevang & Francis, 1994), uno de los mecanismos que actualmente se utiliza para compartir direcciones. Adicionalmente, IPv6 está diseñado para soportar seguridad (IPSec), escalabilidad y transmisiones multimedia. En general, IPv6 fue cuidadosamente concebido y diseñado pensando en futuras aplicaciones.

Se prevé que el cambio de IPv4 a IPv6 no suceda de manera rápida sino que por el contrario tome un largo tiempo. El hecho de que IPv4 sea hoy el protocolo predominante y que Internet se haya convertido en algo imprescindible en el planeta hace difícil –por no decir imposible– realizar la sustitución de los protocolos de una manera rápida. Esta operación involucra a muchas organizaciones y empresas que tendrían que trabajar conjuntamente y de manera sincronizada en el cambio a IPv6.

Debido a los retos mencionados en el proceso de cambio a IPv6, la IETF diseño, junto con el mismo protocolo IPv6, unos mecanismos llamados de transición y coexistencia, con el fin de manejar el paso de IPv4 a IPv6.

Así que, ambos Protocolos de Internet (IPv4 e IPv6) deberán coexistir durante un periodo de tiempo en el que poco a poco habrá más contenidos disponibles en IPv6, y por consiguiente más trafico IPv6; y al mismo tiempo IPv4 debe tender a desaparecer, al menos en un gran porcentaje de la red.

Los mecanismos de transición y coexistencia que han sido desarrollados se dividen en tres grupos *dual stack*, túneles y traducción (Gilligan & Nordmark, 2000).

El proceso de prueba e implementación de los mecanismos de transición mostrados en este documento son el resultado de un proyecto de investigación del grupo Laboratorio de Investigación y Desarrollo en Electrónica y Redes [LIDER] de la Universidad Distrital, el cual trata del análisis de los mecanismo de transición para la coexistencia y/o migración IPv4-IPv6 en el Centro de Computación de Alto Desempeño y la Red de Investigación de Tecnología Avanzada de la Universidad Distrital [RITA-UD] para el soporte IPv6 de distintos servicios y plataformas académicas para e-learning.

II. Mecanismos de transición y coexistencia

El IETF es un grupo auto-organizado de personas que contribuye a la ingeniería y evolución de las tecnologías de Internet. Es el principal órgano involucrado en el desarrollo de las especificaciones de nuevos estándares de Internet (Hoffman & Harris, 2006). Dentro del IETF han existido grupos de trabajo como IP Next Generation Working Group [IPNG], IP Version 6 Working Group [IPv6] y Next Generation Transition Working Group [ngtrans]; actualmente existe IPv6 Operations Working Group [v6ops], cuyo objetivo es establecer lineamientos para la operación de Internet IPv4/IPv6 y proveer una orientación de cómo implementar IPv6 en redes con IPv4 ya implementado y en nuevas redes con IPv6 nativo (IETF, 2012). IPv6 fue diseñado de tal forma que se facilite la transición y coexistencia con IPv4, por lo que se han diseñado diferentes estrategias para la coexistencia con redes y nodos IPv4.

Las herramientas de transición disponibles actualmente se pueden clasificar en tres categorías de acuerdo con la técnica que se utiliza: dual-stack, túneles y traducción.

A. Dual-stack

Es el método propuesto originalmente para tener una transición suave hacia IPv6. La RFC 2893 introdujo el mecanismo *dual-stack*, en el que el sistema operativo de un host o un enrutador está equipado con las dos pilas de protocolos (Gilligan & Nordmark, 2000). De esta manera, el nodo estará en la capacidad de enviar y recibir paquetes IPv4 e IPv6; de esta forma, cuando se establece una conexión ha-

tion potentially requires a long time. The fact that IPv4 is the predominant protocol in our time and, given the actual position of the Internet on the planet (making it indispensable for mankind), it is extremely difficult to carry out this transition in a fast way. This operation involves many organizations and companies, which ought to work together in a synchronized way.

Due to these mentioned challenges in the change process, IETF designed, together and in parallel with IPv6 development, some transition and coexistence mechanisms, responsible for handling the pass from IPv4 to IPv6.

Consequently, both Internet Protocols (IPv4 and IPv6) must coexist during a few occasions, where IPv6 content is expected to grow constantly increasing IPv6 traffic. In contrast, IPv4 traffic should have a tendency to reduce, at least in the majority of Internet sites.

Developed transition and coexistence mechanisms present three main subdivisions: dual stack, tunneling, and translation (Gilligan & Nordmark, 2000).

The test and implementation process of transition mechanisms shown in this document are a result of a research project, carried out at the Laboratory of Research and Development in Electronics and Networks [LIDER, *Laboratorio de Investigación y Desarrollo en Electrónica y Redes*] from the Universidad Distrital. This project is focused on analysis of transition mechanisms for coexistence and IPv4-IPv6 migration in the High Performance Computing Center and the Advanced Technology Research Network [RITA, *Red de Investigación de Tecnología Avanzada*] for IPv6 support of several services and e-learning academic platforms in the university.

II. Transition and coexistence mechanisms

IETF is a self-organized group of people that contributes to engineering and evolution of Internet technologies. It is the main entity involved in development of specifications for new Internet standards (Hoffman & Harris, 2006). Inside IETF, several work groups have existed, like IP Next Generation Working Group [IPNG], IP Version 6 Working Group [IPv6], Next Generation Transition Working Group [ngtrans], and currently the IPv6 Operations Working Group [v6ops] survives, which objective is to *establish lineaments for IPv4/IPv6 operation* and *provide orientation on how to implement IPv6 in networks functioning*

over IPv4 rules and over native IPv6 networks (IETF, 2012). IPv6 was designed to facilitate coexistence with IPv4 and transition towards this new standard, this is why several strategies for IPv4 networks and nodes are designed.

Transition tools currently available are classified in 3 categories related with the used technique:

A. Dual stack

It is the originally proposed method to provide soft transition on the way to IPv6. The Request for Comments [RFC] document number 2893 introduced dual stack mechanism, where operating system in hosts or routers is equipped with the two protocol stacks (Gilligan & Nordmark, 2000); subsequently, nodes are capable to send and receive IPv4 and IPv6 packets. When connections for IPv4 destinations are established, IPv4 connectivity is utilized, and if it is an IPv6 address, this protocol is used. This is, maybe, the simplest coexistence mechanism between these two protocols.

IPv4/IPv6 nodes process IPv4 applications using its corresponding stack, whilst IPv6 applications use IPv6 stack. In case the destination has both protocols, normally IPv6 has priority and nodes will try to connect first using this stack, followed by IPv4 if first connection is not possible. Flow decisions are based on the IP header, specifically on its *version* and *destination address* fields. **Figure 1** shows the way dual stack operates, where, from left to right, we present an IPv4 machine, a dual stack machine, and an IPv6 machine.

In transition, it is important not to affect IPv4; hence, it is convenient using a dual stack. However, a dual stack does not necessarily implies the need of public IPv4 addresses, i.e. when these addresses are run out, dual stack has the possibility to maintain itself.

Many times, it is considered that maintaining dual stacks causes negative impacts over machines and it is not true. For example, an IPv4 stack can take up to 50KB and the impact of this quantity of data in modern computers is insignificant. In contrast, the value used by an IPv6 stack is not double the one needed in IPv4, given similarities between Transmission Control Protocol [TCP] and User Datagram Protocol [UDP]. These similarities result in an increase of 10% or 15% more compared with IPv4. This means modern computers, routers, switches, and other network devices are not affected in their performance by implementing a dual stack.

Today, many commercial systems operate with a dual stack. So, this makes it a very used mechanism in the transition

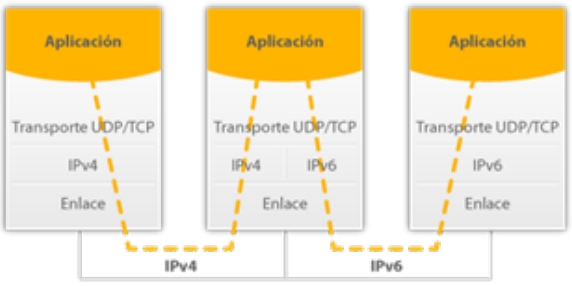

Figure 1. IPv4, Dual stack, IPv6 (Cicileo, 2014) / IPv4, Dual Stack, IPv6 (Cicileo, 2014)

cia un destino sólo IPv4, se utilizará la conectividad IPv4 y si es hacia una dirección IPv6, se utilizará IPv6. Ésta es tal vez la manera más simple de coexistencia de IPv4 e IPv6.

Los nodos IPv4/IPv6 procesan las aplicaciones IPv4 utilizando la pila IPv4, mientras que para las aplicaciones IPv6 utilizan la pila IPv6. En caso de que el destino tenga ambos protocolos, normalmente se preferirá intentar conectar por IPv6 y en segunda instancia por IPv4. Las decisiones de flujo se basan en el encabezado IP, en su campo versión para recibir y en la dirección destino para enviar. La **Figura 1** muestra la forma en que funciona la doble pila, donde se tiene de izquierda a derecha una maquina IPv4, una maquina doble pila y una maquina solo IPv6.

Es importante en la transición no afectar IPv4, por lo que conviene usar la doble pila, lo que no significa necesariamente que se debe tener direcciones IPv4 públicas; es decir, cuando exista un agotamiento total de direcciones IPv4 públicas, aun así podemos mantener la doble pila.

Muchas veces se cree que tener doble pila causa un impacto grande sobre las máquinas y no es así. Una pila IPv4 puede ocupar como mucho 50 kb, el impacto de 50kb en un ordenador actual es insignificante; la doble pila no implica tener 100kb, sino que normalmente, dado que TCP y UDP son los mismos y que hay muchas cosas en común entre IPv6 e IPv4, el impacto al final es de un 10% a un 15% más. Lo anterior quiere decir que en un ordenador de mesa, un portátil, incluso en un dispositivo más pequeño, la doble pila no causa ninguna implicación ni de coste ni de prestaciones, mucho menos cuando hablamos de un router.

Actualmente muchos sistemas comerciales operantes ya cuentan con la doble pila. En consecuencia, esto lo hace un mecanismo bastante utilizado en el proceso de transición.

Existen dispositivos que solo tiene pila IPv6 y dispositivos que solo tienen pila IPv4, por lo que es ideal tener doble pila. Para los dos casos la maquina con doble pila, que puede ser un servidor o un host cualquiera, puede hablar tanto con los dispositivos que solo tienen IPv4, como con los que solo tienen IPv6.

La doble pila se puede implementar en todos los dispositivos en la LAN de una organización para permitir la conectividad interna, pero para la conectividad externa puede suceder que en el camino de la comunicación algún nodo

o red no soporte IPv6, debido a que son redes externas a la organización, como las del ISP; si es así, la comunicación falla y no se puede realizar, por tal razón el mecanismo *dual-stack* no resuelve todo en la transición (la pila dual puede o no utilizarse en conjunto con las técnicas de *tunnelling*).

1) Implementación de dual stack

Implementar dual stack en los equipos que lo soporten, significa instalar y configurar IPv6 teniendo IPv4 ya configurado, lo cual es lo más común. Por ejemplo, en los routers que soporten los dos protocolos, es decir que sean dual stack, se les configura el direccionamiento IPv6, el enrutamiento IPv6, los servicios, etc., y estos brindan servicios y reenvían el trafico, tanto de IPv4, como de IPv6. De igual manera los host dual stack pueden acceder a los recursos tanto de IPv4 como de IPv6.

A continuación se muestra la instalación y configuración básica de IPv6 en un router cisco, en un host Linux y host Windows.

a) Host Windows

Sin duda, una de las más completas pilas IPv6 es la existente en las plataformas Windows más recientes, se tiene soporte completo en las siguientes plataformas: Windows XP SP1 y posteriores, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7 y Windows 8 (Microsoft, 2014). Existen dos formas de activar IPv6 en estas plataformas:

En Windows XP IPv6 viene incluido, y se puede activar o habilitar de dos maneras, por la línea de comandos o con la interfaz gráfica. Con la línea de comandos en una ventana MS-DOS se debe ejecutar el comando:

```
ipv6 install
```

Después de unos segundos un mensaje indicara la correcta instalación. También se podría utilizar, dependiendo de la versión:

```
netsh interface ipv6 install
```

A través del entorno gráfico o panel de control, nos situamos en "Conexiones de red", seleccionar la "red de área local" o "red inalámbrica", "Propiedades" con el pulsador derecho del ratón y a continuación pulsar sobre "instalar", "protocolo" y seleccionar "Microsoft TCP/IP versión 6".

Desde el lanzamiento de los sistemas operativos Windows Vista/2008/7/8 estos sistemas operativos incluyen soporte de IPv6 instalado y habilitado por defecto, por lo tanto no es necesario hacer ninguna configuración adicional. En caso de que se hubiera desactivado, se podría utilizar el procedimiento descrito.

Para realizar la configuración de IPv6 también se puede realizar de dos maneras, por la línea de comandos o con la interfaz gráfica.

Con la consola de comandos se puede configurar manualmente una dirección IPv6 utilizando el comando *netsh* de la siguiente manera:

```
netsh interface ipv6 add address
[interface] [address/prefixe]
```

sition process. Nevertheless, there are devices that only have one stack (either IPv6 or IPv4); for both cases, a machine with a dual stack might be a server or a common host to allow communication with these "one-stacked" devices.

A dual stack can be implemented in every device on Local Area Networks [LAN] of companies, permitting internal connectivity, but for external communications it might happen that a single device or network does not support IPv6, since they are external networks like the ones provided by Internet Service Providers [ISP]. For this reason, the dual stack technique *does not solve* everything in transition (dual stack might be used with tunneling techniques to confront this issue).

1. Dual stack implementation

Implementation of a dual stack on supported devices means installing and configuring IPv6 while having IPv4 already configured, which is a common task. For instance, in routers supporting both protocols (i.e. dual stack devices), IPv6 addressing, routing, and services are configured. These elements provide services and resend both IPv6 and IPv4 traffic. Likewise, dual stack hosts can access either IPv4 or IPv6 resources.

In the following sections, we show basic installation and configuration of IPv6 in a Cisco® router over Linux and Windows® hosts.

a) Windows® host

Undoubtedly, one of the most complete IPv6 stacks is the one in recent Windows® operating systems. Complete support for this protocol is present in the following platforms: Windows XP® SP1 and later, Windows Server® 2003, Windows Vista®, Windows Server® 2008, Windows® 7, Windows® 8 and the recent Windows® 10 (Microsoft, 2014). There are two ways to activate IPv6 in these platforms:

In Windows® XP, IPv6 comes included but factory disabled; therefore, its enabling can be done in two ways: either from the command line or using the graphical interface. With the command line, in a MS-DOS window, the following line must should be executed:

```
ipv6 install
```

After a short time, the system will inform you of correct installation. Also, depending on the version, the following command can be used:

```
netsh interface ipv6 install
```

Through graphic interface, in "Control panel" and under "Network connections", select "Local area network" or "Wireless network"; then select "Internet Protocol (TCP/IP)" and click "Properties" button. Next, select "Install", then "Protocol" and finally select "Microsoft TCP/IP version 6".

From Windows® Vista and posterior versions, IPv6 comes preinstalled and factory enabled; hence, it is not necessary to make additional configurations. In case the protocol is disabled, the previously described procedure can be used.

In order to configure the protocol, a reader might infer it is possible via a command line or graphical interface.

Using the command window, an IPv6 address can be manually configured using the *netsh* command as follows:

```
netsh interface ipv6 add address
[interface] [address/prefixe]
```

Example:

```
netsh interface ipv6 add address 5
2001:db8::65
```

For modification of IPv6 addresses, we use *set* instead of *add*, and elimination of addresses is carried out via the *delete* command. Adding a Domain Name Server [DNS] is possible using the following line:

```
netsh interface ipv6 add dnsserver [name=]
[address=] [index=]
```

Where "Index" means order in the DNS servers list. Example:

```
netsh interface ipv6 add dnsserver "Local
area network" 2001:db8::53 1
```

In actual Windows® clients, the most common procedure is using a graphical interface to configure IPv6 parameters. In order to access this graphical interface, from the "Control panel" select "Network and sharing center", then click on "Change adapter settings" and select, depending on the communication type "Local area connection" or "Wireless network". To finish, select "Properties" of IPv6 protocol. After configuring these parameters, in the command window and using the "ipconfig" line, the user can observe the dual stack feature in the client (**Figure 2**) with options and parameters of IPv4 and IPv6.

Ejemplo:

```
netsh interface ipv6 add address 5
2001:db8::65
```

Para modificarla la dirección IPv6 en lugar de *add* se utiliza *set* y para eliminarla se utiliza *delete*. Para agregar un servidor DNS se utiliza el comando:

```
netsh interface ipv6 add dnsserver [name=]
[address=] [index=]
```

Ejemplo:

```
netsh interface ipv6 add dnsserver "Local
área network" 2001:db8::53 1
```

Index significa el orden en la lista de servidores DNS.

Lo más común en los clientes Windows actuales es utilizar la interface gráfica para configurar los distintos parámetros de IPv6. Para acceder a la interface gráfica en los clientes Windows vista/7/8 desde el panel de control, se selecciona "Redes y recursos compartidos", Luego "Cambiar configuración del adaptador" se selecciona "Conexión de área local" o "red inalámbrica", donde se selecciona las Propiedades del protocolo de Internet versión 6. Despues de configurar los distintos parametros, con el comando "ipconfig" en la consola de comandos se puede observar claramente la caracteristica dual stack del cliente (**Figura 2**) con opciones y parametros tanto de IPv4 como de IPv6.

b) Host Linux

IPv6 está soportado a partir de versión del kernel 2.4.x y 2.2.x en las distribuciones Linux.

Para instalar el módulo IPv6 se ejecuta el siguiente comando:

```
#modprobe ipv6
```

Para comprobar si el módulo IPv6 está instalado, se debe utilizar el siguiente comando:

```
#test -f /proc/net/if_inet6 && echo
"Kernel actual soporta IPv6"
```

Para realizar la configuración de IPv6 en Linux se utilizan una serie de comandos especiales que pueden variar con la distribución de Linux. Para realizar la configuración manual de direcciones se utilizan los siguientes comandos:

```
# /sbin/ip -6 addr add
<ipv6address>/<prefixlength> dev
<interface>
```

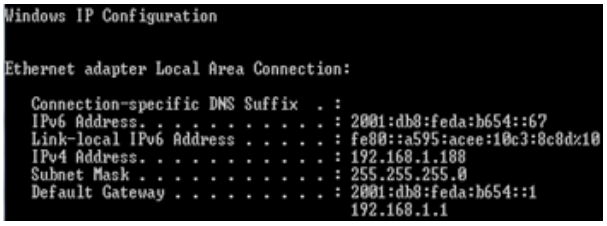

Figure 2. Dual stack feature in a Windows® client / Característica Dual Stack en cliente Windows

```
# /sbin/ifconfig <interface> inet6 add
<ipv6address>/<prefixlength>
```

Ejemplos:

```
# /sbin/ip -6 addr add 2001:db8::10/64 dev
eth1

# /sbin/ifconfig eth1 inet6 add
2001:db8::10/64
```

Para eliminar una dirección se reemplaza la palabra *add* por la palabra *del*. Para añadir una puerta de enlace se utilizan los siguientes comandos:

```
# /sbin/ip -6 route add
<ipv6network>/<prefixlength> via
<ipv6address> [dev <device>]

#/sbin/route -A inet6 add
<ipv6network>/<prefixlength> gw
<ipv6address> [dev<device>]
```

Ejemplos:

```
# /sbin/ip -6 route add 2801:13:77::/48
via 2001:db8::10 eth1

#/sbin/route -A inet6 add add
2801:13:77::/48 gw 2001:db8::10 eth1
```

c) Router Cisco

Las características IPv6 son soportadas desde las versiones 12.0 S, 12.2T, 12.2S, 12.3, 12.4, 12.4T, 12.3T y posteriores (Cisco, 2014). A continuación se muestran las principales características a configurar en un router cisco para trabajar con el protocolo IPv6 son los siguientes.

Antes que los routers puedan enrutar los paquetes IPv6, el enrutamiento IPv6 debe estar habilitado. En los routers Cisco, el enrutamiento IPv4 está activado por defecto, pero el enrutamiento IPv6 no está activado por defecto. La solución tiene un solo comando:

```
ipv6 unicast-routing
```

A menudo pasado por alto, un importante paso al configurar IPv6 en los routers Cisco. Este comando habilita el enrutamiento IPv6 en el router. Se debe tener en cuenta que se debe habilitar el enrutamiento en el modo de configuración global, se debe hacer mínimo este paso y asignar una dirección IPv6 en la interfaz antes de que el router intente encaminar los paquetes de entrada y salida de una interfaz. Si se omite el comando ipv6-unicast-routing, todavía se puede configurar direcciones IPv6 en la interfaz, pero el router actuará como un host IPv6, y no enrutará paquetes IPv6.

Un parámetro imprescindible es configurar una dirección IPv6 en la interface de red. En la configuración de cada interface se asigna la respectiva dirección IPv6 como se muestra a continuación:

```
ipv6 address 2001:db8:0:0::1/64
```

En la mayoría de los casos, se configuran las direcciones IPv6 de las interfaces de los routers de manera estática.

b) Linux host

IPv6 is supported from kernel version 2.4 in Linux distributions. Installation of the IPv6 module is carried out executing the following command:

```
#modprobe ipv6
```

After installation, verification of IPv6 operation is completed in the following line:

```
#test -f /proc/net/if_inet6 && echo "Actual
kernel supports IPv6"
```

In order to configure IPv6 protocol in Linux, several commands ought to be used; these commands might be different depending on the distribution. We execute the following commands to manually configure addresses:

```
# /sbin/ip -6 addr add
<ipv6address>/<prefixlength> dev <interface>
# /sbin/ifconfig <interface> inet6 add
<ipv6address>/<prefixlength>
```

Some examples are:

```
# /sbin/ip -6 addr add 2001:db8::10/64 dev
eth1

# /sbin/ifconfig eth1 inet6 add
2001:db8::10/64
```

Elimination of a simple address is carried out replacing *add* word for *del* word. To add a gateway, we used the following lines:

```
# /sbin/ip -6 route add
<ipv6network>/<prefixlength> via
<ipv6address> [dev <device>]
#/sbin/route -A inet6 add
<ipv6network>/<prefixlength> gw <ipv6address>
[dev<device>]
```

Examples:

```
# /sbin/ip -6 route add 2801:13:77::/48 via
2001:db8::10 eth1

#/sbin/route -A inet6 add 2801:13:77::/48
gw 2001:db8::10 eth1
```

c) Cisco® router

IPv6 features are supported from Cisco IOS versions 12.0 S, 12.2T, 12.2S, 12.3, 12.4, 12.4T, 12.3T and posterior (Cisco, 2014). In the following paragraphs, we present the main features to be configured in a Cisco® router in order to work with IPv6.

Before devices can process IPv6 packets, IPv6 routing must be enabled. In Cisco® routers, IPv4 routing is fac-

tory-enabled, but IPv6 routing is not. The solution for this issue consists only in execution of the following line:

```
ipv6 unicast-routing
```

The previous command is important to configure IPv6 in Cisco® routers and, commonly, the programmer forgets to execute it. This enabling must be done in global configuration mode; after that, the programmer must assign an IPv6 address to the interface *before* the router starts to route packets in and out of the interface. If this command is not executed, IPv6 addresses might be configured still, but the router will act as an IPv6 host and it *will not* route IPv6 packets.

An essential parameter is configuration of IPv6 addresses in network interfaces. In each interface setting, addresses are assigned executing the following command:

```
ipv6 address 2001:db8:0:0::1/64
```

In almost every case, this address configuration is carried out in a static way. Nevertheless, routers might be configured to use dynamic IPv6 addresses (e.g. DHCPv6):

```
ipv6 address dhcp
```

Through self-configuration of stateless addresses:

```
ipv6 address autoconfig
```

B. Tunnels

It is one of the most long-serving mechanisms to access networks that do not support certain protocols natively. In general terms, tunnels are used for encapsulating IPv6 inside IPv4, allowing to pass non-IPv6 networks; this situation is also presented vice versa. Packets are transported until a point in the network in its original configuration, then they are encapsulated to surpass an unsupported part of network and finally, decapsulated in the other extreme to send them in a native way to the destination.

The main objective of tunneling tools is to simplify communication between IPv6 sites. These tools are very useful for network administrators, since they can use them to test IPv6 before total migration to it. They are also useful to connect other IPv6 sites through IPv4 infrastructure (i.e. tools allow IPv6 transit over ISP offering only IPv4). There are two subdivisions in tunneling mechanisms: *manual* and *automatic* ones.

Sin embargo, los routers se pueden configurar para utilizar direcciones IPv6 aprendidas dinámicamente, por ejemplo, por DHCPv6:

```
ipv6 address dhcp
```

Por medio de la autoconfiguración de direcciones sin estado:

```
ipv6 address autoconfig
```

B. Túneles

Es uno de los mecanismos más antiguos para poder atravesar redes que no tienen soporte nativo del protocolo que se está utilizando. En general se utilizan túneles encapsulando IPv6 dentro de IPv4, permitiendo de esta forma atravesar redes que no manejan IPv6, aunque también se encuentra la situación inversa. Los paquetes originales son transportados hasta un punto de la red por medio del protocolo original, luego encapsulados para atravesar la porción de red que no lo soporta y luego des-encapsulados en el otro extremo para ser enviados al destino final en forma nativa.

Las herramientas de tunelizado tienen como objetivo simplificar la comunicación entre sitios IPv6. Estas herramientas son muy importantes porque los administradores de red pueden usarlas para probar IPv6 antes de una migración total y para conectar otros sitios IPv6 a través de la infraestructura de Internet IPv4, es decir permite por ejemplo el tránsito de IPv6 a través de ISPs que solo ofrecen servicio IPv4. Dentro de los mecanismos de túneles existen dos grandes grupos, los que se tienen que configurar o manuales y los túneles automáticos.

Los túneles manuales son la configuración estática en los túneles, en palabras sencillas utilizara una relación de direcciones IPv4 con IPv6 de forma estática y solamente podrá transportar paquetes de IPv6 a redes previamente establecidas, es decir hace una conexión punto a punto con una configuración previamente establecida en los dos extremos. En los túneles automáticos solo un extremo se tiene que configurar, por lo tanto es ideal para los usuarios residenciales. Estos permiten que diferentes redes IPv6 estén interconectadas sobre una red IPv4. La diferencia clave con los túneles manuales es que el túnel automático no es punto a punto, sino que se crean de manera dinámica, punto – multipunto.

1) 6to4

La herramienta 6to4 es un mecanismo que permite a sitios IPv6 comunicarse a través de Internet IPv4 (Carpenter & Moore, 2001). 6to4 asume de manera efectiva a Internet IPv4 como una capa de enlace unicast punto a punto, especificando un mecanismo de encapsulación para transmitir paquetes IPv6 en Internet asignando un prefijo de dirección IPv6 único a cualquier sitio con al menos una dirección IPv4 pública. El mecanismo construye un prefijo de 48 bits usando el prefijo 6to4 2002::/16 y la dirección IPv4 del sitio. Entre sus beneficios, esta técnica no introduce nuevos

registros en las tablas de enrutamiento IPv4 y sólo un nuevo registro con máscara /16 en la tabla de enrutamiento global IPv6. 6to4 es uno de los túneles automáticos. Para ilustrar como trabaja 6to4, supongamos que se tiene un router conectado a una red corporativa cuyo ISP no ofrece acceso nativo a IPv6, el cual tiene una dirección IPv4 estática, este router será el enrutador de límite del sistema en la red IPv6 y se configurara como un punto final del túnel automático 6to4, lo que hace 6to4 de forma automática es generar una dirección IPv6 especial que no depende del ISP, el router 6to4 puede participar en un túnel hasta otra ubicación 6to4, o si es necesario hasta una ubicación IPv6 nativa no 6to4, es decir 6to4 permite una comunicación *peer-to-peer* en máquinas que estén detrás de 6to4 (Carpenter, 2011), esto se ilustra en la **Figura 3**.

A simple vista 6to4 es una gran herramienta, porque es una comunicación *peer-to-peer* que si se configura adecuadamente, siempre funciona. El problema con 6to4 sucede cuando los clientes que están utilizando 6to4, necesitan acceder a redes IPv6 que no utilizan 6to4. Si un cliente 6to4 quiere acceder una página web, pero esta no ha configurado 6to4 en su red, en este punto es donde se hace necesario lo que se conoce como Relay 6to4, este relé solo se usa cuando se trata de comunicar un nodo 6to4 con otro que no sea 6to4, y funcionan como enrutadores IPv6 entre el servicio de túneles 6to4 y el Internet IPv6 nativo. Lo más recomendable en estos casos es tener un Relay 6to4 propio para nuestra red. 6to4 tiene otra desventaja, como se mencionó anteriormente necesita una dirección IPv4 publica en las máquinas para funcionar.

2) Tunnel bróker

El agente de túnel gestiona automáticamente los túneles IPv6 y las solicitudes de túnel de sitios IPv6 aislados en nombre de uno o más servidores dedicados (Durand, 2001). Si se tienen por ejemplo miles de clientes, y se tienen que configurar todos estos túneles manualmente desde luego no es una buena opción, entonces los *tunnel broker* reducen la carga de administración para administradores de redes, quienes de otra manera tendrían que configurar y mantener cada túnel. Los *tunnel broker* se gestionan mediante una interfaz, que muchas veces es una interfaz gráfica, como una página web u otra aplicación parecida, en la cual los usuarios se conectan y se registran para pedir un túnel. De alguna manera esta es una configuración de túneles bastante automática. El *tunnel broker* funciona mejor para sitios IPv6 aislados y para hosts Internet IPv4 que quieren conectarse a una red IPv6.

3) ISATAP

El Protocolo de direccionamiento con túneles automáticos entre sitios, ISATAP [*Intrasite Automatic Tunnel Addressing Protocol*] está diseñado para conectar hosts y enrutadores IPv6 aislados con un sitio IPv4 (Mackay, Edwards, Dunmore, Chown, & Carvalho, 2003), ISATAP especifica un formato de dirección compatible IPv6-IPv4, así como un medio para el descubrimiento de enrutadores de sitios fron-

Manual tunnels equal static configuration in tunnels. In simple words, they use an address relation between IPv4 and IPv6 in a static way and they *only* can transport IPv6 packets to pre-established networks. This means manual tunnels make a point-to-point connection with a previously established connection. Contrariwise, in automatic tunnels only one end has to be configured; hence, they are ideal for residential users. These tunnels allow interconnection of several IPv6 networks over *one* IPv4 network. The key difference with manual tunnels is that automatic ones *are not* point-to-point, they are dynamically created with a point-multipoint configuration.

1) 6to4

The 6to4 tool is a mechanism that allows IPv6 sites to communicate on the Internet through IPv4 (Carpenter & Moore, 2001). 6to4 effectively assumes Internet IPv4 as a unicast data-link layer, specifying encapsulation methods to transmit IPv6 packets over the Internet by assigning a unique IPv6 address prefix with, at least, one public IPv4 address. This tool builds a 48-bit prefix using 6to4 prefix 2002::/16 and IPv4 site address. One of its benefits is *non-introduction* of new registers in IPv4 routing tables: it only creates a new register with /16 subnet mask in global IPv6 routing table. Consequently, 6to4 is an automatic tunnel. To illustrate operation of 6to4, the reader is intended to suppose a router connected to a corporative network, where ISP does not offer native IPv6 access. The router has a static IPv4 address and it is the limit of the system in IPv6 network; thus, it is configured as an end-point of 6to4 automatic tunnel. The function of 6to4 is to automatically generate a special IPv6 address that does not depend on the ISP. The 6to4 router can participate in a tunneling activity until the other 6to4 location or, if necessary, until a non-6to4 native IPv6 location. Namely, 6to4 allows peer-to-peer communication in machines behind it (Carpenter, 2011). This is illustrated in **Figure** 3.

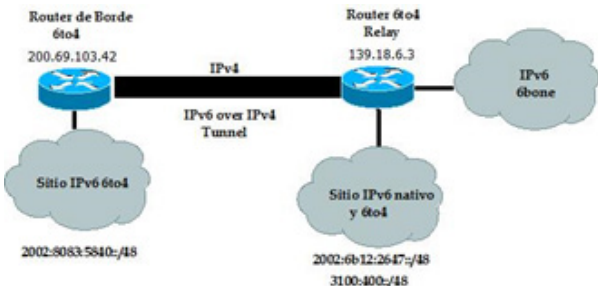

Figure 3. 6to4 tunneling / Tunelizado 6to4

At a glance, 6to4 is a very useful tool, since well configured peer-to-peer communication almost always works. The problem with this method is when clients using it need access to IPv6 networks with no 6to4 support. If a 6to4 client needs access to web pages but the method is not configured on its network, a 6to4 relay is required. This relay is used *only* when communication between 6to4 capable node and 6to4 non-capable node is required. The most recommendable option is having a proper 6to4 relay in our network. Another disadvantage of this method is the necessity of a public IPv4 address in machines to operate.

2) Tunnel broker

The tunnel manager automatically manages IPv6 tunnels and tunnel requests of isolated IPv6 sites in the name of one or more dedicated servers (Durand, 2001). If, for example, thousands of clients are present in the network, manual configuration of every tunnel is not a convenient option. Therefore, tunnel brokers reduce the management load of network administrators avoiding manually configuring tunnels in clients. These are managed through an interface (generally graphical as web pages or applications), where users connect and register to ask for a tunnel. Somehow, this tunnel configuration is automatic. Tunnel brokers work better for isolated IPv6 sites and IPv4 hosts connecting to IPv6 networks.

3) ISATAP

Intra-site Automatic Tunnel Addressing Protocol [ISATAP] is designed to connect isolated IPv6 hosts and routers with an IPv4 site (Mackay, Edwards, Dunmore, Chown, & Carvalho, 2003). ISATAP specifies an IPv6-IPv4 compatible address format and a means to discover routers in border sites. It facilitates IPv6 implementation because it assumes the IPv4 infrastructure of sites with a *Non Broadcast Multiple Access* [NBMA]. ISATAP uses a new interface identifier format for IPv6 that enables automatic IPv6 tunneling in IPv4 within the site, no matter if site uses either public or private IPv4 addresses. This new format can be used with both local and global IPv6 unicast prefixes, in order to allow global and local IPv6 routing. ISATAP mechanisms do not impact size of routing tables and do not require IPv4 special services.

4) Teredo

Teredo proposes a tunneling mechanism of packets over UDP to provide IPv6 connectivity to IPv6 nodes located behind NAT (Templin, 2008). Consequently, Teterizos. Facilita la implementación de IPv6 porque asume la infraestructura IPv4 de los sitios como una capa de enlace no difundida de acceso múltiple [NBMA, *Non Broadcast Multiple Access*]. ISATAP utiliza un nuevo formato de identificador de interfaz IPv6 que habilita el tunelizado automático IPv6 en IPv4 dentro del sitio, así el sitio utilice direcciones IPv4 públicas o privadas. El nuevo formato de identificador de interfaz puede ser utilizado con prefijos unicast IPv6 locales y globales para permitir enrutamiento IPv6 local y global. Los mecanismos ISATAP no impactan el tamaño de la tabla de enrutamiento y no requieren servicios IPv4 especiales.

4) Teredo

Teredo propone un mecanismo que tuneliza paquetes sobre UDP [*User Datagram Protocol*] para brindar conectividad IPv6 a nodos IPv6 ubicados detrás de NATs (Templin, 2008), por lo tanto no necesita una dirección IPv4 publica en las maquinas, sino que utiliza la dirección IPv4 publica que tiene el NAT. Teredo funciona de forma similar a 6to4 ya que utiliza el prefijo 2001::/32 y la dirección IPv4 del sitio, la diferencia se observa en la conectividad a través de NATs, la cual con 6to4 no siempre es posible, Teredo encapsula el tráfico en UDP sobre IPv4, para realizar comunicaciones *peer-to-peer* al igual que lo realiza 6to4. Para ejecutar el servicio, una red necesita servidores Teredo, que administran únicamente una fracción del tráfico entre clientes Teredo y los relés Teredo al igual que en 6to4 actúan como enrutadores IPv6 entre el servicio de túneles Teredo y el Internet IPv6 nativo. Teredo preferiblemente debe usarse únicamente como último recurso donde los dispositivos NATs IPv4 restrinjan el funcionamiento de otros mecanismos.

5) 6in4

6in4 es un mecanismo de transición de protocolo IPv4 a IPv6, este mecanismo funciona de la misma manera que los mecanismos de tunelizado anteriormente mencionados, utiliza un túnel que encapsula los paquetes de IPv6 en IPv4 y se envían a través de la infraestructura de Internet IPv4, 6in4 crea una conexión *peer-to-peer* tal como si fuera una VPN (Huitema, 2006). La acción de encapsular paquetes IPv6 dentro de paquetes IPv4 se conoce como el protocolo 41, de hecho a 6in4 también se le llama a veces protocolo 41. Existen métodos de configuración automática pero normalmente se hace de forma manual. El túnel 6in4 se configura de forma manual para pequeñas redes cuando el proveedor no brinda soporte IPv6, como una medida temporal. Para grandes empresas o ISPs con muchos clientes este tipo de túnel no es operativo, porque se tendrían que configurar y mantener demasiados túneles de forma manual, 6in4 es lógico para utilizar en cantidades pequeñas.

6) Implementación de Túneles

A continuación se muestra la implementación de algunos de los servicios de túneles más populares, como lo es 6to4 y túnel bróker.

a) Implementación del servicio Tunnel broker

A continuación se va a mostrar como configurar este servicio en un cliente residencial sin soporte IPv6 por parte del ISP. Se va a utilizar el servicio de túnel bróker que brinda la empresa de servicios de Internet Hurricane Electric. Hurricane Electric utiliza 6in4 como mecanismo de túnel, el cual también se le llama protocolo-41, 6in4 es un túnel de configuración manual, pero el servicio túnel bróker hace la configuración en el otro extremo del túnel automáticamente. Para utilizar este servicio se necesitan los siguientes requisitos:

- la dirección IPv4 pública del cliente debe ser accesible vía ICMP; y
- si se utiliza NAT, debe permitir y reenviar el protocolo 41.

El Protocolo 41 es uno de los números de Protocolo de Internet. Dentro de la cabecera IPv4, el campo protocolo se establece en 41 para indicar un paquete IPv6 encapsulado.

A continuación se muestran los pasos para utilizar el servicio túnel bróker de Hurricane Electric:

- Se ingresa a la página web https://tunnelbroker.net donde después de crear una cuenta de usuario y acceder, se elige la función *Create Regular Tunnel*.
- Para la creación del túnel se solicita la dirección pública IPv4 a la cual se tiene conexión, si la dirección cumple los requisitos para la implementación del túnel se muestra un mensaje satisfactorio como: *IP is a potential tunnel endpoint*.
- Se debe un elegir un servidor de túnel de los que ofrece el servicio y después se debe escoger la opción *Create Tunnel*.
- Después de crear el túnel se selecciona el enlace del túnel para ver los detalles del mismo en la pestaña *IPv6 Tunnel*, como se observa en la **Figura 4**.
- En la pestaña *Example Configurations* se pueden ver ejemplos de configuración para múltiples sistemas operativos
- Para realizar la configuración en un host Linux, se escoge el sistema operativo *Linux-net-tools* y se debe realizar la configuración del túnel en la consola de comandos tal cual como lo muestra el ejemplo de configuración de la **Figura 5**.

Despues de realizar la configuración en el host se tiene conectividad a Internet IPv6, como se observa en la **Figura 6**.

b) Implementación de 6to4

6to4 es un túnel automático que evita la necesidad de configurar túneles manualmente. Fue diseñado para permitir conectividad IPv6 sin la cooperación de los proveedores de Internet y puede funcionar en un router, permitiendo conectividad a toda una red, o en un host de usuario final. En cualquiera de los casos 6to4 necesita una dirección IPv4 pública para funcionar tal como se mencionó anteriormente. 6to4 utiliza el prefijo 2002::/16 para asignar una dirección Ipv6 que contiene embebida la dirección pública IPv4.

redo does not need a public IPv4 address in machines, it uses public address present in NAT. Teredo works similar to 6to4, since it uses the 2001::/32 prefix and IPv4 site address; the difference relies on NAT connectivity, because it is not always possible using 6to4. Teredo encapsulates traffic in UDP over IPv4 to enable peer-to-peer communications, just as with 6to4. To execute the service, networks need Teredo servers managing only a fraction of the traffic between Teredo clients and relays. Teredo must be used preferably as a last resource, where NAT devices restrict operation of other devices.

5) 6in4

6in4 is a transition mechanism from IPv4 to IPv6 and it works in a similar way to previous tunneling mechanisms. It uses tunnels, which encapsulate IPv6 packets in IPv4. The latter are sent over the Internet IPv4 infrastructure creating a peer-to-peer connection, just as with Virtual Private Networks [VPN] (Huitema, 2006). Encapsulating IPv6 packets into IPv4 is also known as *protocol 41*, the name is also used sometimes for 6in4. There are automatic configuration methods, but usually, this method is manually set. A 6in4 tunnel is manually configured for small-scale networks when ISP does not support IPv6. For big companies or ISP with thousands of clients, this type of tunneling is not operative, given the necessity to set and maintain many tunnels manually; consequently, 6in4 is useful in small quantities.

6) Tunnels implementation

In the following paragraphs, we present implementation mechanisms of some of tunneling services previously presented.

a) Implementation of tunnel broker service

We show how to configure this service in a residential client without IPv6 support. We used the tunnel broker service provided by ISP Hurricane Electric. Hurricane Electric uses 6in4 as a manual tunneling mechanism in one end, but the tunnel broker is automatically configured in the other end. In order to use this service, the following requirements are needed:

- IPv4 public client address must be accessed via Internet Control Message Protocol [ICMP].
- If NAT is used, it must allow resending of protocol 41.

Protocol 41 is one of the numbers in Internet Protocol. Within IPv4 header, *protocol* field is established in "41" to indicate an encapsulated IPv6 packet.

Figure 4. Tunnel details / Detalles del Túnel

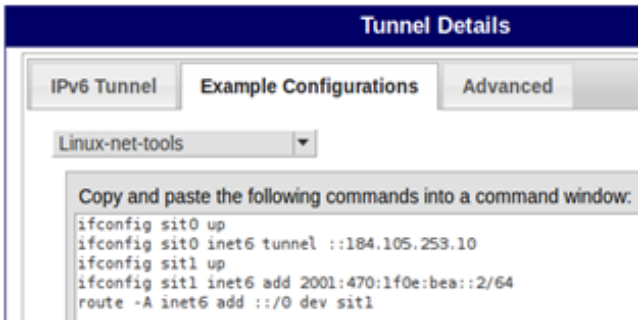

Figure 5. Tunnel broker configuration example / Ejemplo de configuración de Tunnel broker

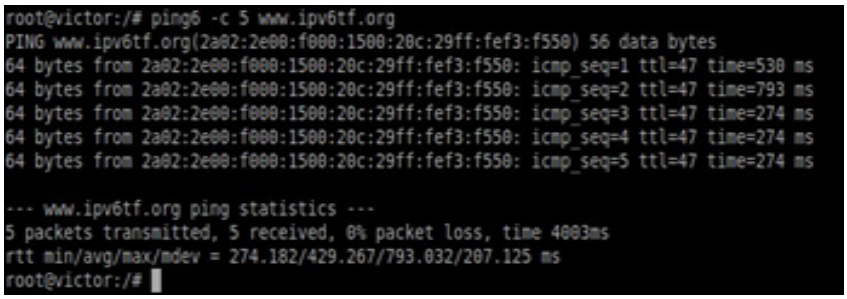

Figure 6. IPv6 connectivity with tunnel broker service / Conectividad IPv6 con el servicio Túnel Bróker

We show the steps to use the tunnel broker service in Hurricane Electric ISP:

- Go to webpage https://tunnelbroker.net, where, after creating an account and gain access, choose option *Create Regular Tunnel*.
- For tunnel creation, the system requires a public IPv4 address to connect; if this address complies with requirements for tunnel implementation, a confirm message as the following is shown: *IP is a potential tunnel endpoint*.
- Choose one tunnel server from the ones offered by the service, after choose *Create Tunnel* option.
- After tunnel creation, select tunnel link to see details of the same in tab *IPv6 Tunnel*, as **Figure 4** shows.
- In the *Example Configurations* tab, the user can see examples of configuration in several operating systems.
- To configure the Linux host, choose the *Linux-net-tools* option and proceed to configure the tunnel in the command window as the example in **Figure 5** shows.

After configuration is done in the host, IPv6 connectivity is enabled, as **Figure 6** illustrates.

b) 6to4 implementation

6to4 is an automatic tunneling method that avoids the necessity of manually configuring tunnels. It was

Para implementar 6to4 se puede hacer uso de algún servicio de relay 6to4 ofrecido públicamente. La universidad politécnica de Cataluña ofrece este servicio del cual se hará uso para realizar la implementación. En un host Linux Debian/Ubuntu se realiza la configuración de la siguiente manera:

En el fichero /etc/network/interfaces se debe escribir las siguientes líneas:

```
auto tun6to4
iface tun6to4 inet6 v4tunnel
address <<dirección 6to4 del host>>
netmask 16
gateway ::<<IPv4 del router relay>>
endpoint any
local <<Dirección IPv4 del host>>
mtu 1472
```

Donde la dirección 6to4 del host se debe calcular a partir del prefijo 2002::/16 y la dirección IPv4 publica del host. Si la dirección pública IPv4 es 200.69.103.10 la dirección 6to4 se calcula así: se convierte a hexadecimal cada uno de los octetos de la dirección IPv4: 200 = C8, 69 = 45, 103 = 67, 10 = A; se añaden al prefijo IPv6 y se completa la dirección arbitrariamente como se muestra a continuación:

```
2002:c845:670a::1
```

La dirección IPv4 del router relay 6to4 lo debe proporcionar quien brinda el servicio del relay router en este caso la Universidad Politécnica de Cataluña ofrece este servicio proporcionando la siguiente dirección del relay router:

```
81.88.81.19
```

Con estos datos la configuración queda de la siguiente manera:

```
auto tun6to4
iface tun6to4 inet6 v4tunnel
address 2002:c845:670a::1
netmask 16
gateway :: 81.88.81.19
endpoint any
local 200.69.103.10
mtu 1472
```

Después de realizar está configuración y guardarla ya se tiene conectividad a sitios IPv6.

C. Herramientas de traducción

Esta técnica consiste en utilizar algún dispositivo en la red que convierta los paquetes de IPv4 a IPv6 y viceversa. Ese dispositivo tiene que ser capaz de realizar la traducción en los dos sentidos para poder permitir la comunicación

Ni los mecanismos *dual-stack* ni los mecanismos de tunelizado funcionan para comunicaciones entre un nodo sólo IPv6 y un nodo sólo IPv4. Esas comunicaciones requieren un mecanismo de traducción ya sea en la capa de red, transporte o aplicación. Este mecanismo fue pensado inicialmente para plataformas que solo tuvieran soporte IPv4 y tuvieran que comunicarse con plataformas que solo tuvieran soporte IPv6. Originalmente se pensó que los servidores web tendrían una velocidad de adopción mucho más lenta que los clientes, pero en realidad no ha sido así porque la mayoría de los sistemas operativos tienen su versión cliente y servidor, con soporte IPv4 e IPv6, por ello y por otras razones propias de la traducción, la IETF decidió descatalogar los mecanismos de traducción.

1) SIIT

El Protocolo de Traducción de IP/ICMP sin estado [*Stateless IP/ICMP Translation*] especifica un algoritmo de traducción clave para habilitar la interoperabilidad entre hosts IPv6 exclusivos y hosts IPv4 exclusivos (Nordmark & Gilligan, 2005). En SIIT, direcciones IPv4 asignadas temporalmente se utilizan para direccionar direcciones IPv6 mapeadas desde IPv4. Los paquetes pasan a través de un traductor SIIT, que convierte el encabezado de los paquetes a IPv4 o IPv6 y traduce la dirección del encabezado en IPv4 a un lado e IPv6 al otro. El término de traducción se refiere a la conversión directa de protocolos entre IPv4 e IPv6, de manera bidireccional. SIIT define un rango especial de direcciones IPv6 llamadas IPv4 traducidas o direcciones IPv6 mapeadas desde Ipv4, una dirección IPv4 traducida es de la forma 0::FFFF:0:W.X.Y.Z que indica un nodo IPv6. En la **Figura 7** se muestra el modelo de funcionamiento de la traducción.

a) Traducción de IPv4 a IPv6

Cuando el traductor recibe un datagrama IPv4 que contiene una dirección destino que está fuera de la red IPv4, entonces traduce el encabezado de ese datagrama por uno IPv6 y lo reenvía basándose en la dirección IPv6 destino. Una descripción básica y rápida de esta traducción consiste en que el encabezado IPv4 del paquete es removido y reemplazado por uno IPv6.

designed to allow IPv6 connectivity without cooperation of ISP and it can operate in a router, allowing connectivity to the whole network or a single host. In any case, 6to4 needs a public IPv4 address to operate (as mentioned before). 6to4 uses the 2002::/16 prefix to assign an IPv6 address that contains the public IPv4 address embedded.

To implement 6to4, the network administrator can use some public relay 6to4 service. The *Universidad Politécnica de Cataluña* offers this service; so, we used this service in our implementation. In a Linux/Debian host, configuration is carried out as follows:

Open the terminal and in the directory "/etc/network/interfaces", write the following lines:

```
auto tun6to4
iface tun6to4 inet6 v4tunnel
address <<host 6to4 address>>
netmask 16
gateway ::<<relay IPv4 router>>
endpoint any
local <<host IPv4 address>>
mtu 1472
```

Where 6to4 host address must be calculated from 2002::/16 prefix and host IPv4 public address. If the IPv4 public address is, for example, 200.69.103.10, the-6to4 address is calculated as follows: each one of the octets in the IPv4 address are converted to hexadecimal: 200 = C8, 69 = 45, 103 = 67, and 10 = A. Then, these values are added to IPv6 prefix and address is arbitrary completed as the following line shows:

```
2002:c845:670a::1
```

The IPv4 address of the 6to4 relay router is provided by the company allowing the relay router service, i.e. the Universidad Politécnica de Cataluña in this case. The address provided by this company is:

```
81.88.81.19
```

Consequently, the configuration is as follows:

```
auto tun6to4
iface tun6to4 inet6 v4tunnel
address 2002:c845:670a::1
netmask 16
gateway :: 81.88.81.19
endpoint any
local 200.69.103.10
mtu 1472
```

After configuring the host, IPv6 connectivity is presented and fully operational.

C. Translating tools

This technique consists in using a network device that converts IPv6 packets into IPv4 packets and vice versa. This device needs to be able to perform translation in both ways to allow communication.

Neither dual stack nor tunneling mechanisms work for communications between only one IPv6 and one IPv4 node. That kind of communication requires translating mechanisms in the network, transport, or application layer. This translating mechanism was originally designed for platforms that support only IPv4 requiring communication with platforms supporting only IPv6. Originally, it was estimated that web servers would have lower adaptation speeds than clients, but this is not true, since most operating systems have their client and server versions with support of IPv4 and IPv6. For this and other reasons, IETF decided to discontinue translating mechanisms.

1) SIIT

Stateless IP/ICMP Translation [SIIT] specifies a key translating algorithm to enable interoperability between IPv6 and IPv4 exclusive hosts (Nordmark & Gilligan, 2005). Here, temporarily assigned IPv4 addresses are used to redirect IPv6 addresses mapped from IPv4. Packets pass through the SIIT translator, which converts the packets header to IPv4 or IPv6 and translates the header address in IPv4 on one side, and in IPv6 in the other. The translating term refers to the protocol of the direct conversion between IPv4 and IPv6 in a bidirectional way. SIIT defines a special range of IPv6 addresses called *translated IPv4* or *IPv6 mapped from IPv4*. A translated IPv4 address has the form 0::FFFF:0:W.X.Y.Z; in **Figure 7**, we show the translating functioning model.

a) IPv4 to IPv6 translation

When a translator receives an IPv4 datagram that contains a destination address outside IPv4 network, it translates the header of this datagram into an IPv6 one and resends it based on its IPv6 destination address. A quick and basic description of this translation consists of a change in the packet header from IPv4 to IPv6.

Figure 7. SIIT translating / Traducción SIIT

b) Traducción de IPv6 a IPv4

Cuando el traductor recibe un datagrama IPv6 destinado a una dirección IPv4-mapeada, éste traduce el encabezado IPv6 a un encabezado IPv4. Nuevamente, el encabezado original es removido y sustituido, en este caso, por un encabezado IPv4. Existen algunas diferencias entre la fragmentación IPv6 e IPv4 y el mínimo MTU por enlace que efectúa la traducción. Un enlace IPv6 tiene un MTU mínimo de 1280 bytes. El límite correspondiente para IPv4 es de 68 bytes (Li, Bao, & Baker, 2011). Para una mayor referencia y explicación consultar el RFC 2765.

2) NAT-PT

Traducción de direcciones de red/traducción del protocolo NAT-PT, definida en la RFC2766, permite la comunicación entre host solo IPv6 y host solo IPv4. La comunicación se realiza mediante el uso de un dispositivo dedicado que hace la traducción entre direcciones IPv4 e IPv6 y mantiene el estado durante el tiempo de la sesión. Traducción de direcciones de red [NAT] es muy similar al NAT de IPv4 descrito en el RFC 1631, pero no es idéntico. El NAT de IPv4 traduce una dirección IPv4 en otra dirección IPv4. Aquí NAT-PT se refiere a la traducción de una dirección IPv4 en una dirección IPv6 y viceversa.

Traducción del protocolo [PT] utiliza SIIT y se refiere a la traducción de un paquete IPv4 en un paquete IPv6 semánticamente equivalente y viceversa. El dispositivo NAT-PT también incluye un Nivel de pasarela de aplicaciones [ALG] para hacer posible la traducción entre las peticiones y respuestas DNS IPv4 e IPv6.

3) NAT64/DNS64

Si una red es IPv6 nativa para llegar a sitios que son sólo IPv4 se realiza una traducción al estilo NAT, mediante un mapeo entre los paquetes IPv6 e IPv4. Se utiliza un prefijo especial para mapear direcciones IPv4 a IPv6: 64:ff9b::/96. Es necesario también utilizar una modificación al DNS, llamada DNS64, que permite generar un registro AAAA aun cuando el destino no tenga dirección IPv6, es decir, cuando el DNS responde sólo con registros de tipo A (Nordmark, 2000).

4) Implementación de la traducción

NAT-PT ofrece una solución al problema de interconectividad mediante el uso de enrutamiento transparente y traducción de direcciones y protocolos. Para el siguiente ejemplo de implementación se mostrara la configuración de NAT-PT en un enrutador cisco que hará las veces de traductor NAT-PT y comunicará una red IPv4 con una red IPv6:

```
ipv6 nat v4v6 source 192.168.30.9
2000::960B:202

ipv6 nat v6v4 source 3001:11:0:1::1
150.11.3.1

ipv6 nat prefix 2000::/96
```

El prefijo 2000::/96 traduce cualquier entrada IPv4 para ser vista por la red IPv6. La red 192.168.30.0 es la red IPv4 que se desea comunicar con la red IPv6. Las redes 3001:11:0:: y 150.11.3.0 se configuran en las interfaces *loopback* del host en la red IPv6 y el host en la red IPv4 respectivamente.

Los mecanismos de traducción fueron pensados inicialmente para plataformas que solo tuvieran soporte IPV4 y tuvieran que comunicarse con plataformas que solo tuvieran soporte IPv6. Originalmente se pensó que los servidores web tendrían una velocidad de adopción mucho más lenta que los clientes, pero en realidad no ha sido así porque la mayoría de los sistemas operativos tienen su versión cliente y servidor, con soporte IPv4 e IPv6. La traducción de direcciones no es muy recomendada como mecanismo de transición, ya que tiene varias limitaciones, como que muchos protocolos de seguridad no pueden ser utilizados a través de dispositivos de traducción. Personalmente en las fuentes consultadas, los mecanismos de traducción son poco utilizados e implementados en la transición IPv4-IPv6. Por lo anterior y por otras razones propias de la traducción, la IETF decidió descatalogar los mecanismos de traducción (Bagnulo, Matthews, & van Beilnum, 2011).

III. Criterios de evaluación a tener en cuenta de las herramientas de transición

Es esencial que cada mecanismo pueda considerarse como "adecuado para sus fines", es decir, apto para una determinada función cuando se aplica a un escenario de red particular, como criterios a tener en cuenta se puede mencionar:

A. Escalabilidad

Tal vez la más importante consideración es cómo un mecanismo en particular podrá ser escalado. La técnica de túneles manuales tiene la desventaja de que si se anexa una nueva red IPv6 todo los enrutadores frontera de cada red deben actualizar su configuración de túneles, esto es claramente un problema de escalabilidad.

Por ejemplo, NAT-PT quedó obsoleto porque podía manejar bien pocas conexiones, pero, así como el NAT Ipv4, cuando las conexiones empezaban a crecer también empezaba a crecer el procesamiento y las cargas de mantenimiento, lo que causaba degradación y fallas en el servicio.

B. Requerimientos de direcciones IPv4 e IPv6

Diferentes mecanismos tienen diferentes requerimientos de direcciones IP para su funcionamiento. 6to4 por ejemplo, requiere una dirección Ipv4 global para su configuración, mientras que teredo y el servicio de túnel bróker se pueden configurar detrás de dispositivos NAT sin necesidad de tener una dirección pública Ipv4 en cada host.

C. Funcionalidad

En ciertos escenarios de transición algunas de las nuevas características de Ipv6 no pueden ser aprovechadas. Muchos mecanismos tienen dificultades traduciendo direcciones. Dual Stack es el mecanismo de transición más funcio-

b) IPv6 to IPv4 translation

When the translator receives an IPv6 datagram destined to an IPv4 mapped address, it translates the IPv6 header into an IPv4 header. Again, the original header is removed and replaced, in this case, for an IPv4 header. There are some differences between IPv6 and IPv4 fragmentation and the smallest value of Maximum Transfer Unit [MTU] per link that is needed in translation. An IPv6 link has a minimal MTU of 1280 bytes. The corresponding limit for IPv4 is 68 bytes (Li, Bao, & Baker, 2011). For further reference and other information, we suggest the reader to consult RFC 2765.

2) NAT-PT

Network Address Translation – Protocol Translation [NAT-PT] is defined in RFC 2766 and allows communication between only-IPv4 nodes with only-IPv6 nodes. Communication is carried out through use of a dedicated device, which translates addresses and maintains the state during session time. NAT-PT is almost identical to the original NAT described in RFC 1631, but they have slight differences. IPv4 NAT translates IPv4 addresses into IPv4 addresses. On the other hand, NAT-PT refers to translation of an IPv4 address into an IPv6 address and vice versa.

Protocol translation uses SIIT and refers to translation of an IPv4 packet into an IPv6 packet semantically equivalent and vice versa. NAT-PT also includes an Application Level Gateway [ALG] to make possible translation between DNS requests and responses in both Internet protocols.

3) NAT64/DNS64

If a network is native-IPv6, to achieve sites only-IPv4 a NAT translation is carried out through mapping between IPv4 and IPv6 packets. A special prefix to map these addresses is used: "64:ff9b::/96". Also, it is necessary to use a modification to the DNS, called DNS64, which allows generation of an AAAA records even when destination does not have IPv6 address, –i.e. when DNS *only* operates with type A records– (Nordmark, 2000).

4) Translation implementation

NAT-PT offers a solution to the problem of interconnectivity by using transparent routing and addresses/protocols translation. In the following implementation example, we show configuration of NAT-PT in a Cisco® router, which handles role of NAT-PT translator and communicates an IPv4 network with an IPv6 network:

```
ipv6 nat v4v6 source 192.168.30.9
2000::960B:202
```

```
ipv6 nat v6v4 source 3001:11:0:1::1
150.11.3.1

ipv6 nat prefix 2000::/96
```

Prefix 2000::/96 translates any IPv4 input, in order to be understandable for the IPv6 network. Network 192.168.30.0 is the IPv4 one to communicate with the IPv6 one. Networks 3001:11:0:: and 150.11.3.0 are configured in host loopback interface in IPv6 and IPv4, respectively.

Translating mechanisms were initially designed for platforms with support only for IPv4 with communication necessities with only-IPv6 networks. Addresses translation is not very recommended as a transition mechanism, since many security protocols cannot be used through translating devices. In consulted literature, translation mechanisms have low usage and implementation in IPv4-IPv6 transition. Hence, IETF decided to discontinue these mechanisms (Bagnulo, Matthews, & van Beilnum, 2011).

III. Evaluation criteria to be considered in transition tools

It is essential that each mechanism can be considered as "suitable for its goal", i.e. suitable for a certain function when it is implemented in a particular network scenario. As general criteria to consider, we can find:

A. Scalability

Maybe the most important consideration is *how a particular mechanism will be scalable*. The manual tunnels technique has the disadvantage that, given its absence of scalability when new IPv6 networks are added, every border router in each network must update its tunneling configuration.

To demonstrate this, NAT-PT is now obsolete because although it worked flawlessly with low connections, but when the network grows exponentially, processing time and maintain loads also grow, affecting system performance by causing degradation and failures.

B. IPv4 and IPv6 addresses request

Several mechanisms have different IP addresses requirements for their operation. For example, 6to4 requires a global IP4 address for its configuration, whilst Teredo and tunnel broker can be set *behind* NAT devices, eliminating need of a public IPv4 address in each host.

C. Functionality

In certain transition scenarios, some of the new features of IPv6 cannot be implemented as many mechanisnal por excelencia porque permite la inmersión de IPv6 sin afectar IPv4.

D. Facilidad de uso

La configuración de una herramienta de transición debería ser transparente al usuario final; si IPv6 se implementa satisfactoriamente es poco probable que los usuarios lo noten. El mecanismo Dual Stack es el más sencillo de utilizar ya que se trata de la utilización de cada protocolo independientemente sin interferir en el funcionamiento del otro. El servicio de túnel bróker también es bastante sencillo de utilizar e implementar para el usuario final.

E. Facilidad de administración

Se refiere al esfuerzo de implementación requerido y al esfuerzo en la administración de la red de transición. Los túneles manuales por razones obvias tienen una carga de administración mayor que los túneles configurados automáticamente, como 6to4. Nuevamente se debe mencionar el mecanismo dual stack debido a que es muy sencillo al separar la administración de los dos protocolos.

IV. Análisis del soporte e implementación de ipv6 para plataformas académicas y servicios de red

El Centro de Computación de Alto Desempeño de la Universidad Distrital ofrece servicios a la RITA, la cual se encarga de trabajar por la investigación en la Universidad Distrital mediante la conexión a las redes académicas, la dotación y administración de infraestructura de red para la investigación, y en la implementación y apropiación de servicios. En su enfoque académico la red de investigación cuenta con plataformas e-learning para uso académico e investigativo como aulas virtuales Moodle, servidor de matemáticas SAGE, repositorio de software libre mediante un servidor FTP para descargas desde enlaces de alta velocidad, entre otros. Estas plataformas e-learning están alojadas en un servidor HTTP Apache instalado en una distribución Debian y el repositorio de medios utiliza el servidor FTP pure-FTPd. Como tal para revisar el soporte IPv6 en las plataformas e-learning vasta revisar y configurar el soporte IPv6 en el servidor HTTP donde se encuentran tales aplicaciones.

A. Análisis del soporte y configuración de IPv6 en el servidor HTTP Apache donde se encuentran alojadas las plataformas e-learning

El proyecto servidor HTTP Apache es un software desarrollado con esfuerzo colaborativo, cuyo objetivo es crear un servidor Web HTTP robusto, de grado comercial, con muchas características y de libre disposición e implementación del código fuente. Apache desde la versión 2.0 tiene soporte para IPv6 de forma predeterminada. Para verificar que el servidor apache está escuchando por el puerto 80 en todas sus interfaces se utiliza el siguiente comando:

```
# netstat –antup
```

El resultado obtenido se muestra en la Figura 8.

La línea donde se muestra :::80 indica la utilización del puerto 80 de la dirección IPv6 ::/128, que es utilizado para mostrar que el servicio no está asociado a ninguna de las direcciones IPv6 del dispositivo, es decir el servicio puede ser accedido en cualquiera de las interfaces de red de la máquina. Esto también se puede observar en el archivo de configuración ports. conf que se encuentra en el directorio /etc/apache2, en las siguientes líneas:

```
NameVirtualHost   *:80

Listen   80
```

Si se desea especificar una dirección IPv6, lo que es bastante común en servidores de producción, para el caso de la plataforma Moodle con la dirección IPv6 2801:14:0:CE5E::70, se debe modificar de la siguiente manera:

```
NameVirtualHost   [2801:14:0:CE5E::70]:80

Listen   [2801:14:0:CE5E::70]:80
```

Y en el archivo /etc/apache2/sites-available/default se debe modificar la línea:

```
<VirtualHost *:80>
```

Por la línea:

```
<VirtualHost [2801:14:0:CE5E::70]:80>
```

De esta manera las plataformas virtuales e-learning como Moodle, el servidor de matemáticas SAGE, etc. ya tienen conectividad global mediante el protocolo IPv6, en la **Figura 9** se muestra el acceso a la plataforma Moodle mediante IPv6:

B. Análisis del soporte IPv6 en el servidor FTP Pure-FTPd donde se encuentra el repositorio de software libre para descargas mediante enlaces de alta velocidad

El servidor FTP Pure-FTP es un servidor muy seguro, fácil de usar, eficiente y con muchas características. Además es totalmente compatible con IPv6 por defecto en la versión 1.0.35, la cual se instaló en una distribución Debian para la RITA, y cada componente de opción de configuración y registro funciona con IPv4 e IPv6. Se comprueba que se puede acceder al servidor vía IPv6 mediante un cliente FTP, se debe ingresar la dirección IPv6 del servidor entre corchetes, el nombre del usuario y la clave, de igual manera también se puede tener acceso mediante un navegador web, como se muestra en la **Figura 10**.

V. Análisis del rendimiento de la red utilizando la herramienta túnel bróker, el protocolo IPv6 y el protocolo IPv4

Con el propósito de analizar el rendimiento de la red del centro de computación de alto desempeño de la Universidad Distrital se realizaron distintas mediciones de parámetros comunes utilizando la herramienta túnel bróker la cual se comparó con las mismas mediciones realizadas utilizando los protocolos IPv4 e IPv6. Para realizar estas ms present problems translating addresses. Dual stack is the most functional transition mechanism, since it permits IPv6 full immersion without affecting IPv4.

D. Ease of use

Setting a transition tool should be transparent to end users; if IPv6 is satisfactorily implemented, users have low probability of noticing. The dual stack mechanism is the simplest to use given its feature of using each protocol independently. The tunnel broker service is also easy to use and implement for end users.

E. Easiness in management

This refers to the required implementation effort and transition network management effort. Manual tunnels have, as the reader might infer, higher management loads compared with automatically configured tunnels (as 6to4). Again, the dual stack mechanism stands out for its simplicity in management.

IV. Analysis of support and implementation of ipv6 for academic platforms and network services

The high performance computing center of the Universidad Distrital offers services to RITA, which safeguards research in the university through connections with academic networks, endowment and management of network infrastructure for research, and services implementation plus appropriation. Given its academic focus, the research network has e-learning platforms for academic and research use; providing resources like Moodle virtual classrooms, SageMath servers, and free software repositories through FTP, among others. These e-learning platforms are allocated in an Apache HTTP [Hypertext Transfer Protocol] server installed in a Debian distribution and media repositories use pure-FTPd FTP server. Summarizing, checking IPv6 support in e-learning platforms requires inspecting and configuring IPv6 support in HTTP server where these applications are allocated.

A. Analysis of support and configuration of IPv6 in Apache HTTP server where e-learning applications are allocated

HTTP Apache server project is software with collaborative development and its objective is to create robust, commercial grade, with several features, and freely licensed HTTP web servers. Since version 2.0, Apache has native IPv6 support. To verify that the server is listening in port 80 in each interface, we use the following command:

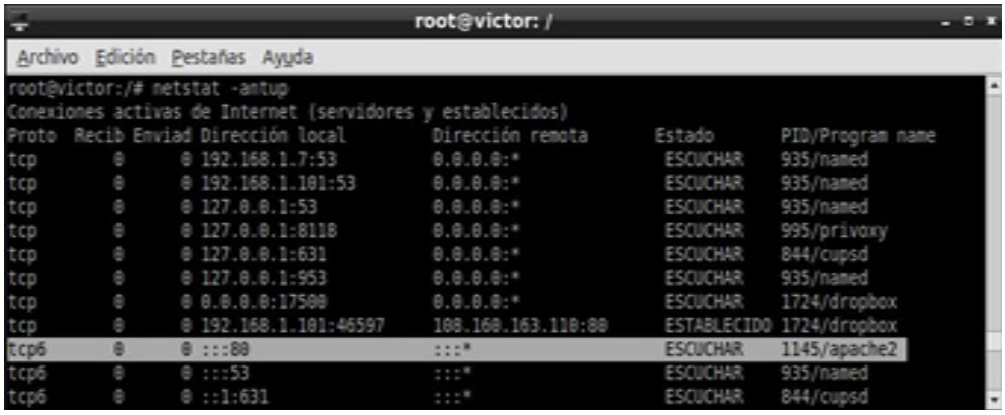

Figure 8. Apache2 port verification / Verificación puerto Apache2

```
# netstat –antup
```

The obtained result is shown in **Figure 8**.

The line where ":::80" is shown indicates using port 80 of the "::/128" IPv6 address. This utilization shows that the service is not associated with any IPv6 device address; i.e. the service can be accessed in any of machine network interfaces. This feature is also observed in the *ports.conf* configuration file, located in the "/etc/apache2" directory by executing the following lines:

```
NameVirtualHost   *:80
Listen   80
```

If the user wants to specify an IPv6 address (common task in production servers), modification must be carried on in the following way if, for example, the service to modify is the Moodle platform with "2801:14:0:CE5E::70" IPv6 address:

```
NameVirtualHost   [2801:14:0:CE5E::70]:80
Listen   [2801:14:0:CE5E::70]:80
```

And in the file "/etc/apache2/sites-available/default, the following line must be modified:

```
<VirtualHost *:80>
```

By this:

```
<VirtualHost [2801:14:0:CE5E::70]:80>
```

By executing this, e-learning platforms like Moodle and SageMath achieve global connectivity through IPv6. We show access to Moodle using IPv6 in **Figure 9**.

B. Analysis of IPv6 support in Pure-FTPd FTP server where free software repository is allocated

The Pure-FTPd FTP server is safe, easy to use, efficient, and it has other notable features. In addition, it is

mediciones se utilizaron herramientas comunes como la utilidad ping que tienen los sistemas operativos la cual permites observar los paquetes enviados y recibidos y el tiempo de entrega de cada uno, para medir el *throughput* se utilizó la herramienta de descarga de los navegadores webs que indican la velocidad a la cual se descargan los archivos y la herramienta de gestión de red de código abierto Zennos. Inicialmente se midió el *throughput* en la red a una determinada hora de día haciendo uso únicamente del protocolo IPv6 y desactivando el protocolo IPv4, luego con el protocolo IPv4 únicamente y por último utilizando la herramienta túnel bróker.

Se realizaron varias mediciones desde un sitio específico a la misma hora para los tres casos, de todas las mediciones se graficó el promedio de los resultados obtenidos. El servidor de prueba fue debían.org cuya ubicación está en Australia. Se realizaron diez descargas y se observó el trhoughput desde el navegador, una de las descargas para los tres casos se observa en la **Figura 11**, de esta manera se hicieron las diez descargas midiéndose el tiempo de cada una y posteriormente se calcula el promedio del *throughput*.

El trhoughput medido con el navegador en las diez descargas se promedió con el *throughput* calculado con la herramienta de gestión de red Zennos la cual tiene herramientas graficas de visualización del trhoughput en tiempo real como se observa en la **Figura 12**.

Las 10 mediciones de las descargas en el navegador promediadas con la medición observada en la herramienta Zennos se muestran en la **Tabla 1**.

Como se puede observar en la **Figura 13** el *throughput* es mucho mayor cuando se utiliza únicamente el protocolo IPv6, esto se debe a la mayor eficiencia de IPv6 en cuanto a el tamaño de la MTU por defecto, la simplicidad de la cabecera, entre otros aspectos que pueden ayudar a una mayor eficiencia del protocolo IPv6 en el proceso de transferencia de paquetes y por consiguiente en la fragmentación y el reensamblado. El *throughput* con el túnel configurado es muy similar a IPv4.

La segunda medición que se realizó está relacionada con la perdida de paquetes. Para esta medición se enviaron cien

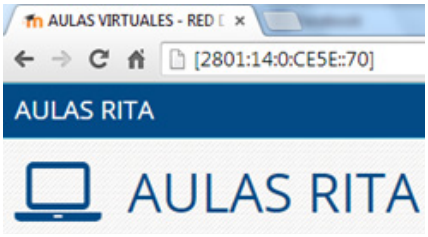

Figure 9. Access to Moodle platform using IPv6 / Acceso a la plataforma Moodle mediante IPv6

Figure 10. Access to FTP server through IPv6 / Acceso servidor FTP mediante IPv6

Figure 11. Downloads in web browser / Descargas con el navegador

Figure 12. Real time measurement of throughput with Zennos / Medición en tiempo real del throughput con Zennos

fully compatible with IPv6 in version 1.0.35, which we installed on a Debian distribution for RITA; confirming correct operation of IPv4 and IPv6. We established that server access via IPv6 is possible through the FTP client; for this, user must enter the IPv6 server address in square brackets, then introduce the username and password. Similarly, access from web browsers is also possible, as **Figure 10** shows.

V. Analysis of network performance using tunnel broker tool and IPv6/IPv4 protocols

In order to analyze the performance of the network in the high performance computing center of the Universidad Distrital, we carried out several measurements of common parameters using a tunnel broker tool. These measurements were compared by using both IPv4 and IPv6. We used common tools like *ping*, download tools in browsers, and Zennos network management tool. Ping allows observation of sent and received packets together with their delivery time and throughput measurement was possible with download tools in browsers. Initially, we measured throughput in the network at a given time using *only* IPv6 (i.e. disabling IPv4), then using only IPv4, and finally using a tunnel broker tool.

Several measurements were carried out from a specific site at the same time for the three previous cases; we processed and plotted the average of the obtained results. The test server was "debian.org", located in Australia. We requested a total of 10 downloads and we focused on throughput from the browser; one of these downloads is shown in **Figure 11**. The average throughput was calculated measuring the time of each download.

The throughput measured with the web browser in the 10 downloads was averaged with the calculated one from Zennos tool. This tool has throughput graphical visualization tools, as **Figure 12** shows.

We averaged measurements of downloads in web browser with observed data in Zennos. These results are shown in **Table 1.**

As the reader can see in **Figure 13**, the throughput is higher when only IPv6 is used. This because its greater efficiency is related to the MTU size and simplicity in the header, among other features. IPv6 attains better efficiency in packet transfer and, subsequently, in pac-

	IPv6	IPv4	Tunnel / *Túnel*
Navigator / Navegador	19,20	12,56	10,58
	21,23	11,01	9,20
	22,54	10,50	11,2
	22,60	13,20	8,44
	23,40	9,84	8,94
	24,28	9,27	12,4
	21,30	8,43	13,1
	24,02	12,90	11,2
	22,57	11,6	8,40
	23,10	12,7	7,80
Zennos	23,00	13,00	11,00
Average / Promedio	22.47	11.36	10.20

Table 1. Measurements of throughput - first instance / Mediciones de throughput - primer caso (Mbps)

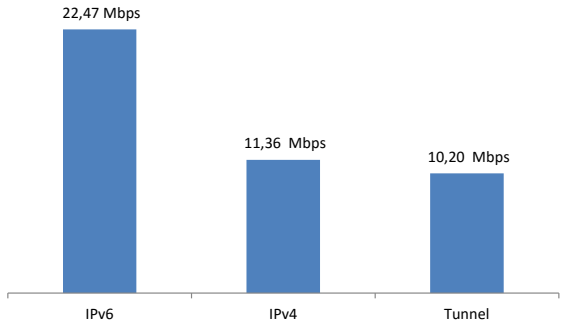

Figure 13. Other throughput measurements / Medidas de throughput

ket fragmentation and assembling. The throughput with configured tunnel is similar to the one presented in IPv4.

The second measurement was related to packet loss. For this, we sent 100 packets to server "ipv6tf.org" using ping utility as follows:

```
C:\Users\admin>ping -n 100 www.ipv6ft.org
```

The results obtained by using IPv4, IPv6, and tunnels are shown in **Figure 14, Figure 15, and Figure 16**, respectively.

A summary of results is in displayed in **Figure 17**, where, for better visualization, we plotted received packets. From this, the reader can infer there is no packet loss with any of used tools, which represents a good network quality and ICMP plus ICMPv6 protocols.

From this ping test, the third measurement is derived. It is related to response time, obtaining the average time of sent packets, summarized in **Table 2** and plotted in **Figure 18**.

The best response time was obtained using the IPv6 protocol and corresponded to half of the time presented

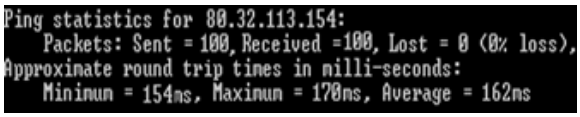

Figure 14. Obtained results - IPv4 protocol / Resultado obtenido - protocolo IPv4

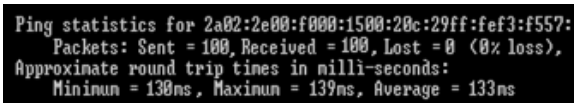

Figure 15. Obtained results - IPv6 protocol / Resultado obtenido - protocolo IPv6

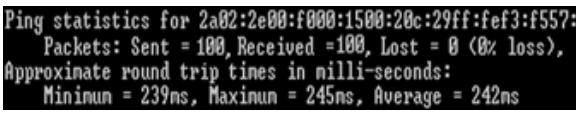

Figure 16. Obtained results using tunnels / Resultado obtenido utilizando el túnel

paquetes al servidor ipv6tf.org utilizando la utilidad ping como de la siguiente manera:

```
C:\Users\admin>ping -n 100 www.ipv6ft.org
```

Los resultados obtenidos con el protocolo IPv4, con el protocolo IPv6 y utilizando el túnel, se muestran, respectivamente, en las **Figuras 14 a 16**.

El resumen de los resultados se grafica en la **Figura 17**, en la cual, para una mejor visualización, se graficaron los paquetes recibidos, se observa que no existe perdida de paquetes con ninguna de las herramientas utilizadas, lo cual habla muy bien de la calidad de la red y del protocolo ICMP e ICMPv6.

De la anterior prueba de ping, en la cual se enviaron cien paquetes al servidor www.ipv6tf.org, se obtiene tambien la tercera medición realizada, el tiempo de respuesta; de esta prueba se obtuvo el tiempo promedio de los cien paquetes enviados, cuyos resultados se muestran en la **Tabla 2** y se grafican en la **Figura 18.**

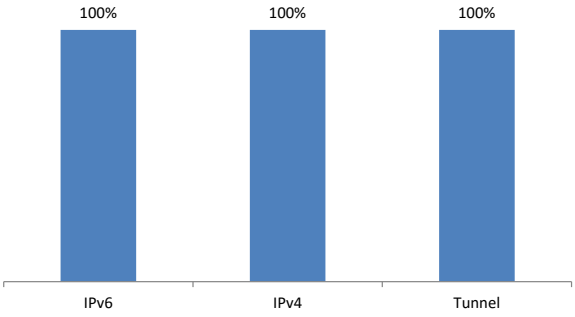

Figure 17. Received packets / Paquetes recibidos

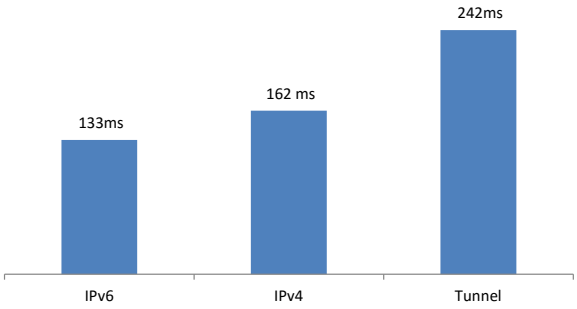

Figure 18. Response time / Tiempo de respuesta

IPv6	IPv4	Tunnel / *túnel*
133	162	242

Table 2. Average response time / Tiempo promedio de respuesta (ms)

El mejor tiempo de respuesta se obtuvo utilizando el protocolo IPv6, que mostró aproximadamente la mitad del tiempo de respuesta del túnel, siendo con este mecanismo el tiempo más largo de respuesta como se muestra en la Figura 18, con IPv4 se tuvo el segundo mejor tiempo 29 ms más lento que con IPv6. Se debe tener en cuenta que el servidor de pruebas en este caso fue ipv6tf.org que se encuentra en Europa, y al estar el servidor de túneles en estados unidos es comprensible que el mecanismo de túneles tenga el tiempo más grande.

VI. Conclusiones

La IETF tuvo en cuenta que debido a que muchas aplicaciones están funcionando con el protocolo IPv4, no se puede reemplazar IPv4 con IPv6 de manera inmediata. Se trata entonces de que los dos protocolos coexistan, es por esto que no se puede hablar de migración en toda la infraestructura de Internet, si no de transición y coexistencia de los dos protocolos durante un periodo de tiempo que es dificil predecir. En este documento se ha presentado una breve introducción acerca de los mecanismos creados por la IETF para IPv6 pensando en facilitar la transición y coexistencia con IPv4.

Se mencionaron los tres tipos de mecanismos, la doble pila, los túneles, y la traducción. La doble pila es el mecanismo ideal para dejar IPv4 funcionando tal como está y no afectarlo. el mecanismo de la doble pila o dual stack es el principal y más utilizado mecanismo de transición, es casi obligatorio en el proceso de transición de cualquier red, ya que permite una transición suave y una convivencia de los dos protocolos, dando la gran ventaja de implementar IPv6 sin interferir IPv4.

El otro mecanismo ampliamente utilizado son los túneles, en el que se encuentran muchas opciones, para muchos casos en particular. Acerca de los Relay necesarios para un buen funcionamiento de los túneles lo lógico es que estos sean desplegados por los proveedores, porque si un proveedor no ha podido desplegar IPv6 nativo en su red, por lo menos es mejor que el relay esté en su red antes que los usuarios lo tengan que buscar afuera, el servicio para el cliente va a ser mejor a diferencia si por ejemplo el relay está del otro lado del mundo.

in tunneling; this last presented the longest response time as **Figure 18** shows. Using IPv4, the results were 29ms slower than IPv6. One detail to consider is the location of the test server: Europe. Therefore, it is comprehensible that a tunnel server located in United States presented higher response times.

VI. Conclusions

IETF considered that, due to many applications still working with the IPv4 protocol, replacement of them for IPv6 cannot be done immediately. Hence, the main idea is coexistence of both protocols; for this reason, migration of *all* Internet infrastructure is, nowadays, illogical. Instead, IETF proposes transition and coexistence of both IPv4 and IPv6 but in a time period difficult to predict. In this document, we presented a brief introduction about mechanisms created by this organization for IPv6, designed to ease these tasks between both protocols.

We mentioned three mechanisms: dual stack, tunnels, and translation. Dual stack is the most used transition mechanism; it is almost mandatory in transition networks because it provides soft moving and coexistence of IPv4 and IPv6 without interferences.

Another widely used mechanism is tunnels; they offer several options and the network administrator should choose carefully depending on its needs. One particular detail about relays, necessary to a correct operation of tunnels, is that logic implies that these relays are implemented by ISP. This is because, in case the ISP does not have this service, users will search it outside its network, affecting general performance in tunneling.

The best results related with measurements carried out at the high performance computing center of the Universidad Distrital were obtained using IPv6. With tunnels mechanism, it is well known that a tunnel server

is needed; this server receives the 6in4 packet, decapsulates it and sends to 6bone. The location and general performance of this server affect packet transfer, generating delays, just as measurements showed. During throughput measurements in first server, an average value of 22.47 Mbps was obtained using IPv6; in contrast, using IPv4, system reached 50.55% of first value (11.36 Mbps) and using tunnels, value was 45.5% of obtained with IPv6 (10.22 Mbps). Regarding the packet loss measurement, neither of the studied techniques presented lost. Finally, in response time measurements, IPv6 presented best results with 133ms, whilst IPv4 had 21.8% slower response time and tunnel results were 62.9% slower than IPv4 and 92.4% slower than IPv6. It is well known that these measured parameters are directly affected by other network aspects, such as physical, data link, and application layer parameters. Regardless of this, differences between these three protocols related to quality and performance were observed, with best results in every aspect to IPv6.

Our study presented in this document is an important effort to push the global implementation of IPv6. Some of the authorities behind pursuing this goal are: IETF, RIPE NCC, SurfNET, Hurricane Electric, LACNIC, NIC.br, NIC.mx, CLARA, RENATA, and Ministries of ICT in several countries, among others. This paper also makes an important academic contribution in IPv4-IPv6 transition and migration process, since implementation of IPv6 acquires today an urgency state for performance, customer satisfaction, and scalability of ISP. Nowadays, it is almost impossible to think about implementation of new networks without IPv6. ST

De las mediciones de parámetros que se hicieron para evaluar el rendimiento de la red en el centro de computación de alto desempeño de la Universidad Distrital, con el protocolo IPv6 se obtuvieron los mejores resultados. Con el mecanismo de túneles se sabe que se utiliza un servidor de túneles que se encarga de recibir el paquete 6in4, el cual lo desencapsula y lo envía a la 6bone, la ubicación y rendimiento de este servidor de túneles afecta en la transferencia y en los retardos de los paquetes, tal como se observó en las mediciones realizas. En la medición del throughput al primer servidor, con el protocolo IPv6 se obtuvo el mayor valor con 22.47 Mbps de valor promedio, con el protocolo IPv4 se obtuvo un 50.55% de este valor, y con el túnel un 45.5%. En la medición de pérdidas, con ningún protocolo existieron perdidas. Por último en la medición del tiempo de respuesta el protocolo IPv6 tuvo el mejor tiempo con 133 ms, el protocolo IPv4 tuvo un tiempo de respuesta 21,8% más lento, y la respuesta con el túnel fue 92,4% más lento que con IPv6 y 62,9% más lento que con IPv4. Se sabe que todos estos parámetros que se midieron son afectados directamente también por otros aspectos de la red de la capa fisica, la capa de enlace de datos, la capa de aplicación, pero a pesar de esto se evidenciaron grandes diferencias de calidad y rendimiento comparando estos protocolos, resultando mejor en la comparación de estos parámetros el protocolo IPv6.

El estudio que se hace en este documento hace un gran aporte a esfuerzos que se están haciendo en el mundo para impulsar la implementación del protocolo IPv6. Entre las autoridades en el área que impulsan la implementación del protocolo IPv6 están: IETF, RIPE NCC, SurfNET, Gobierno Español, Hurricane Electric, LACNIC, NIC.br, NIC.mx, Red CLARA, RENATA, MinTIC de Colombia, gobiernos de muchos países, etc., también hace un aporte académico importante como todo artículo de investigación, en el proceso de transición y migración IPv4-IPv6, ya que desplegar el protocolo IPv6 adquiere hoy más que nunca un sentido de urgencia, volviéndose inevitable e inaplazable si los proveedores de conectividad desean satisfacer la demanda de sus clientes y de nuevos usuarios. En la actualidad es inaceptable implementar nuevas redes y servicios sin tener en cuenta el protocolo IPv6. ST

References / Referencias

Bagnulo, M., Matthews, P., & van Beilnum. (2011, Apr.). Stateful NAT64: network address and protocol translation from IPv6 clients to IPv4 servers [RFC 6146 / on-line]. Retrieved from: http://tools.ietf.org/html/rfc6146

Carpenter, B. & Moore, K. (2001, Feb.). *Connection of IPv6 domains via IPv4 clouds* [RFC 3056]. Retrieved from: http://www.ietf.org/rfc/rfc3056.txt

Carpenter, B. (2011, Aug.). *Advisory guidelines for 6to4 deployment* [RFC 6343]. Retrieved from: http://tools.ietf.org/html/rfc6343

Cicileo G. (2014). *Mecanismos de transición*. Retrieved from: http://portalipv6.lacnic.net/mecanismos-de-transicion/

Cisco (2014). *Cisco IOS IPv6 feature mapping*. Retrieved from: http://docwiki.cisco.com/wiki/Cisco_IOS_IPv6_Feature_Mapping

Deering, S & Hinden, R. (1998). *Internet Protocol, version 6 (IPv6) specification* [RFC 2460]. Retrieved form: http://www.ietf.org/rfc/rfc2460.txt

Durand, A. (2001, Jan.). *IPv6 tunnel broker* [RFC 3053]. Retrieved from: http://datatracker.ietf.org/doc/rfc3053/

Egevang, K. & Francis, P. (1994, May). *The IP network address translator (NAT)* [RFC 1631]. Retrieved from: http://www.faqs.org/rfcs/rfc1631.html

Gilligan, R. & Nordmark, E. (2000, Aug.) *Transition Mechanisms for IPv6 Hosts and Routers* [RFC 2893]. Retrieved from: http://www.ietf.org/rfc/rfc2893.txt

Hoffman, P. & Harris, S. (2006, Sep.) *The Tao of IETF: A novice's guide to the Internet Engineering Task Force* [RFC 4677]. Retrieved from: http://www.ietf.org/rfc/rfc4677.txt

Huitema, C. (2006, Feb.) *Teredo: tunneling IPv6 over UDP through network address translations (NATs)* [RFC 4380]. Retrieved from: http://datatracker.ietf.org/doc/rfc4380/

Internet Engineering Task Force [IETF]. (2012, Jun.). *IPv6 Operations (v6ops). Description of working group*. Retrieved from: http://datatracker.ietf.org/wg/v6ops/charter/

LACNIC. (2014, june 10th). *No hay más direcciones IPv4 en América Latina y Caribe*. Retrieved from: http://www.lacnic.net/web/anuncios/2014-no-hay-mas-direcciones-ipv4-en-lac

Li, X., Bao, C., & Baker, F. (2011, Apr.). *IP/ICMP translation algorithm* [RFC 6145]. Retrieved from: http://datatracker.ietf.org/doc/rfc6145/

Mackay, M., Edwards, C., Dunmore, M., Chown, T., & Carvalho, G. (2003). A scenario-based review of IPv6 transition tools. *IEEE Internet Computing, 7*(3), 27-35.

Microsoft (2014). *Networking and access technologies TCP/IP v4 and v6*. Retrieved from: http://technet.microsoft.com/en-s/network/bb530961.aspx

Nordmark, E. & Gilligan, R. (2005, Oct.). *Basic transition mechanisms for IPv6 hosts and routers* [RFC 4213]. Retrieved From: http:// tools.ietf.org/html/rfc4213

Nordmark, E. (2000, Feb.) *Stateless IP/ICMP translation algorithm (SIIT)* [RFC 2765]. Retrieved from: http://www.ietf.org/rfc/rfc2765.txt

Pidwirny, M. (2006). Introduction to the Oceans. In *Fundamentals of physical geography [2nd ed.]*. Retrieved from: http://www.physicalgeography.net/fundamentals/8o.html

Templin, F. (2008, March). *Intra-site automatic tunnel addressing protocol (ISATAP)* [RFC 5214]. Retrieved from: http://datatracker.ietf.org/doc/rfc5214/

CURRICULUM VITAE

Carlos Andrés Martínez Alayón Electronic Engineer, Specialist in Network Security. Candidate to Master of Science in Computer Science and Communications. Professor in Telematics, Informatics, and Networks in the Universidad Escuela Colombiana de Carreras Industriales [ECCI] and in the Universidad Distrital Francisco José de Caldas. Coordinator of the Advanced Technology Research Network in the Universidad Distrital [RITA-UD]. Scientific vice-director of the Laboratory of Research and Development in Electronics and Networks [LIDER], also at the Universidad Distrital. / Ingeniero Electrónico, Especialista en Seguridad de Redes. Candidato al título de Maestría en Ciencias de la Información y las Comunicaciones con énfasis en Teleinformática. Docente en Teleinformática y Redes de la Universidad Escuela Colombiana de Carreras Industriales (ECCI) y la Universidad Distrital Francisco José de Caldas. Coordinador de la Red de Investigaciones de la Red de Investigaciones en Tecnológica Avanzada de la Universidad Distrital [RITA-UD]. Codirector Científico del Grupo de investigación Laboratorio de Investigación y Desarrollo en Electrónica y Redes [LIDER] de la Universidad Distrital.

Roberto Ferro Escobar Electronic Engineer, Master of Science in Telematics, Informatics, and Networks from the Universidad Distrital Francisco José de Caldas. PhD in Engineering Informatics, Information Society, and Knowledge Management from the Universidad Pontificia de Salamanca. He is currently director of PhD in Engineering program, Dean of Engineering Faculty at the Universidad Distrital Francisco José de Caldas, and associate professor. Director of the Advanced Technology Research Network in the Universidad Distrital [RITA-UD] and scientific director of the Laboratory of Research and Development in Electronics and Networks [LIDER]. / Ingeniero Electrónico, Magister en Teleinformática de la Universidad Distrital "Francisco José de Caldas". Doctor en Ingeniería Informática, Sociedad de la Información y Gestión del Conocimiento de Universidad Pontificia de Salamanca. Actualmente es Director del Programa de Doctorado en Ingeniería y Decano de la Facultad de Ingeniería de la Universidad Distrital "Francisco José de Caldas". Docente de Planta en la misma Universidad. Director de la Red de Investigaciones de la Red de Investigaciones en Tecnológica Avanzada de la Universidad Distrital [RITA-UD]. Director Científico del Grupo de investigación Laboratorio de Investigación y Desarrollo en Electrónica y Redes [LIDER] de la Universidad Distrital.

Victor José Arrieta Zambrano Electronic Engineer from the Universidad Distrital Francisco José de Caldas. Research assistant at the Laboratory of Research and Development in Electronics and Networks [LIDER]. He has worked in projects related with addressing and routing in IPv4 and IPv6. Network and Support technician at the Advanced Technology Research Network in the Universidad Distrital [RITA-UD]. / Ingeniero Electrónico de la Universidad Distrital "Francisco José de Caldas". Monitor de Investigaciones en el grupo de investigación Laboratorio de Investigación y Desarrollo en Electrónica y Redes [LIDER] de la Universidad Distrital. Ha realizado trabajos de Investigación, principalmente en protocolos de direccionamiento y enrutamiento sobre IPv4 e IPv6. Técnico de Soporte y Redes en la Red de Investigaciones de la Red de Investigaciones en Tecnológica Avanzada de la Universidad Distrital [RITA-UD].