Informe técnico CESGA-2005-001

Estudio de la problemática de la implantación de IPv6 en la RECETGA

Andrés Gómez, Jose Carlos Pérez, Juan Villasuso, Natalia Costas

Fundación CESGA
Avenida de Vigo, s/n
Campus Sur
15705 Santiago de Compostela
A Coruña – ESPAÑA

E-mail: {agomez, jcarlos, jvilla, natalia}@cesga.es

28 de enero del 2005

Palabras clave: IPv6, QoS, redes

Resumen

Dado que el protocolo IPv6 está alcanzando un nivel de madurez suficiente, la Fundación CESGA ha analizado el estado del arte del mismo y las consecuencias y estrategias de migración desde la versión actual a la prevista. Este documento resume los resultados de dicho análisis y propone una estrategia de migración de la RECETGA a IPv6.

Tabla de contenidos

1. Introducción. Objetivos del documento

La Fundación Centro Tecnolóxico de Supercomputación de Galicia (Fundación CESGA) en lo que sigue) tiene encomendada la gestión de la Red de Ciencie e Tecnoloxía de Galicia (RECETGA) que conecta todos los centros de investigación de Galicia y Universidades, además de proveer el acceso a redes externas. Esta red de comunicaciones está basada en el protocolo Ipv4 como protocolo de transporte. Dado que existen problemas importantes de escalabilidad de este protocolo debido al crecimiento exponencial de dispositivos conectados que momentáneamente están siendo resueltos con soluciones parciales, se ha definido un nuevo estándar (Ipv6) que intenta paliar estos problemas a la vez que añade nuevas funcionalidades que se han visto necesarias para un correcto funcionamiento de las redes de comunicaciones.

Dado que este nuevo protocolo está alcanzando un nivel de madurez suficiente, la Fundación CESGA se propuso analizar el estado del arte del mismo y analizar las consecuencias de la migración y estrategias de migración desde la versión actual a la prevista. Este documento resume los resultados de dicho proyecto y propone una estrategia de migración de la RECETGA a Ipv6.

El informe se estructura en seis partes fundamentales. Primero se analiza la situación de los protocolos involucrados y las ventajas comparativas de Ipv6 frente a Ipv4. En una segunda sección se analiza el posible soporte de Ipv6 en la infraestructura de red del CESGA, para a continuación explicar el test-bed desarrollado para probar las funcionalidades del nuevo protocolo. Finalmente, se analizan las posibles estrategias de migración y se hacen las recomendaciones para migrar la RECETGA teniendo en cuenta éstas y la infraestructura disponible.

2. Descripción de IPv6. Ventajas IPv6 frente a IPv4

2.1. Introducción

IP versión 6 (IPv6 – *Internet Protocol version 6*) es una nueva versión del protocolo de Internet (IP – *Internet Protocol*), diseñada para sustituir a la versión 4 (IPv4) utilizada en redes de datos a nivel mundial.

IPv6 afronta los cambios que sufrirá el tráfico de las redes IP de forma global. Se prevé la intensificación en el uso de transacciones en tiempo real al tiempo que evolucionan las intranets e Internet de las redes actuales a sistemas de transmisión que transportan una vasta riqueza de datos, entretenimiento y otros servicios.

La razón fundamental detrás de la necesidad de un nuevo protocolo de Internet radica en el intenso y continuado crecimiento de la red. El espacio de direcciones de 32 bits de IPv4 está agotándose. Cada *host* en Internet necesita una dirección única y, aunque determinadas técnicas permiten la compartición de direcciones (como NAT, Network Address Translation, o los servidores web virtuales), éstas son sólo un paliativo para un sistema exhausto.

Un motivo adicional además de la enorme capacidad de direccionamiento IPv6 es la simplificación del encaminamiento. No es excesivamente costosa la reserva de un espacio de direcciones elevado, en términos de recursos, y nos reporta la posibilidad de asignación de direcciones en jerarquías multi-nivel. Esta asignación simplifica los algoritmos de enrutamiento y la cantidad de espacio necesaria en las tablas de encaminamiento. La explosión del tamaño de las tablas de los *routers* es un fenómeno bien conocido y su minimización compensa el coste (http://www.potaroo.net/ispcolumn/2001-03-bgp.html). Esta aproximación jerárquica hará la configuración automática del *router* mucho más viable. A medida que Internet crece, y el número de direcciones se incrementa, también se incrementa el número de caminos extremo a extremo, mientras el número de rutas intermedias se expande por un factor incluso mayor.

2.2. Adaptabilidad al crecimiento

De la experiencia con IPv4 se deriva que las funcionalidades de direccionamiento y enrutado deben ser válidas ante previsiones de crecimiento razonables.

El crecimiento del número de equipos que se sirven del protocolo IP ha aumentado de forma exponencial, conectando ordenadores en grandes negocios, administración, universidad, etc. El tipo de terminales finales en comunicaciones a través de Internet varía desde PC's a superordenadores, la mayor parte de ellos se conectan a redes de área local y el número de ellos que son móviles se incrementa continuamente.

La necesidad de proporcionar conectividad a estos dispositivos móviles impondrá al protocolo de Internet ciertas restricciones de diseño, de forma que

supongan una baja carga de ancho de banda y se soporte la movilidad y la autoconfiguración como elemento fundamental, así como confidencialidad y autenticación.

Otro mercado que hará uso de Internet es el del entretenimiento. Cada equipo de televisión podrá ser un *host* en Internet. Este grupo de clientes necesitarán que el protocolo de Internet soporte direccionamiento y enrutamiento a gran escala y autoconfiguración con un mínimo *overhead*.

Los dispositivos de control también harán uso de las características aventajadas del nuevo protocolo de Internet. Estos dispositivos permiten el control de elementos tales como iluminación, calefacción y aire acondicionado, motores y otro equipamiento controlado por conmutadores analógicos, consumiendo durante su funcionamiento grandes cantidades de energía eléctrica. Las soluciones de red que necesita este grupo son sencillez, robustez, facilidad de uso y bajo coste.

2.3. Mejoras aportadas por IPv6

IPv6 no fue diseñado para cambiar de forma radical las características de IPv4; funciones existentes se han mantenido y otras que no se utilizaban han sido eliminadas. Los cambios aportados por IPv6 se pueden clasificar en las siguientes categorías:

2.3.1. Mayores capacidades de encaminamiento y direccionamiento

IPv6 incrementa el tamaño de las direcciones IP de 32 a 128 bits lo que permite:

- Incremento del número de niveles de jerarquías de direccionamiento.
- Incremento del número de nodos direccionables.
- Autoconfiguración de direcciones mucho más sencilla.
- Mejora de la escalabilidad de enrutamiento *multicast* mediante la adición del campo *scope* en las direcciones de dicho protocolo.

2.3.2. Anycast

Definición del nuevo tipo de dirección *anycast*, ésta identifica conjuntos de nodos. Se utiliza para enviar paquetes a cualquier integrante del grupo.

2.3.3. Simplificación del formato de cabecera

Algunos campos contenidos en las cabeceras IPv4 se han eliminado o hecho opcionales para reducir el coste del procesado realizado con frecuencia y, de esta forma, limitar el coste de ancho de banda de la cabecera IPv6 aún habiendo incrementado el tamaño de las direcciones.

Aunque las direcciones IPv6 son cuatro veces más largas que las direcciones IPv4, la cabecera IPv6 es solamente dos veces mayor que la cabecera IPv4.

2.3.4. Soporte mejorado de opciones y extensiones

Cambios en la codificación de las opciones de la cabecera de forma que permita:

- Reenvío de paquetes más eficiente.
- Límites menos estrictos en la longitud de las opciones.
- Mayor flexibilidad para la introducción de opciones nuevas en un futuro.

2.3.5. Adición de capacidad para ofrecer calidad de servicio

Se añade capacidad de etiquetado de paquetes pertenecientes a flujos de tráfico particulares para los cuales el emisor solicita tratamiento especial, tal como calidad de servicio que no sea por defecto o en tiempo real.

2.3.6. Capacidad de privacidad y autenticación

Se especifican extensiones que dan soporte de:

- Autenticación
- Integridad de datos y
- Confidencialidad de datos

Se incluyen como elemento básico de IPv6 y forma parte de todas las implementaciones.

2.4. Características notables de IPv6

Existen cierto número de razones por las cuales IPv6 es apropiado como futuro protocolo de Internet.

- Resuelve el problema de escalabilidad de Internet.
- Proporciona mecanismos de transición flexibles.
- Ha sido diseñado para satisfacer las necesidades de los nuevos mercados tales como los dispositivos móviles, de entretenimiento y/o control.

La facilidad de transición es un punto clave en la concepción del diseño de IPv6. Fue diseñado para interoperar con IPv4. Se han añadido mecanismos específicos (direcciones IPv4 embebidas, etc.) para soportar la transición y compatibilidad hacia atrás.

IPv6 soporta direcciones jerárquicas de gran tamaño, lo que permitirá a Internet continuar su crecimiento y proporcionar nuevas capacidades de encaminamiento no presentes en IPv4. Dispone de direcciones *anycast* que pueden ser utilizadas para rutas seleccionadas en función de políticas y tiene *multicast* de ámbito que proporciona mayor escalabilidad que el *multicast* IPv4. También dispone de mecanismos de utilización de direcciones locales que permiten la autoconfiguración de los nodos.

La estructura de direcciones IPv6 ha sido diseñada para transportar información enviada mediante otros protocolos como IPX o NSAP. Esto facilitará la migración de estos protocolos de internet a IPv6.

IPv6 proporciona una plataforma para nueva funcionalidad de Internet. Lo cual incluye:

- Soporte de flujos de tiempo real
- Selección de proveedor
- Movilidad de *hosts*
- Seguridad extremo a extremo
- Auto-configuración
- Auto-`reconfiguración`

En resumen, IPv6 es la nueva versión del protocolo IP interoperable con IPv4. Ha sido diseñado para tener un buen comportamiento en redes de gran ancho de banda (ATM por ejemplo) y al mismo tiempo ser eficiente en redes de bajo ancho de banda (inalámbricas por ejemplo).

2.5. Aspectos técnicos del protocolo IPv6

2.5.1. Tipos de direcciones IPv6

- **Direcciones *unicast* globales o direcciones para proveedores**

  Este tipo de direcciones posee una estructura jerárquica muy sencilla:

3 bits	N bits	M bits	O bits	125-n-m-o bits
010	ID registro	ID proveedor	ID cliente	Dentro del cliente

Tabla 1: Formato de las direcciones *unicast* globales

- **Direcciones locales de *site***

  Muchas empresas y organizaciones utilizan TCP-IP sin estar conectados realmente a Internet, bien sea porque temen los riesgos relativos a la seguridad que vienen derivados de la conexión directa o porque quieren

configurar y probar su red interna sin realizar el procedimiento de conexión completa. Para este propósito existen las direcciones locales de *site*, reservándose para ellas el prefijo 11111110 11. Las direcciones locales de *site* no se pueden utilizar para encaminar a través de Internet. Su unicidad está garantizada únicamente dentro del *site*. Sólo se pueden utilizar para comunicación entre dos estaciones dentro de un *site*.

10 bits	38 bits	16 bits	64 bits
1111111011	00 ... 00	ID subred	Dirección única para la tecnología del enlace

Tabla 2: Formato de las direcciones locales de *site*

- **Direcciones de enlace local**

Los nodos que no estén configurados ni con direcciones globales ni con direcciones de *site* pueden configurarse con direcciones de enlace local. Éstas se componen del prefijo de enlace local, 1111111010, un conjunto de ceros y una interfaz de red.

10 bits	54 bits	64 bits
1111111011	00 ... 00	Dirección única para la tecnología del enlace

Tabla 3: Formato de las direcciones de enlace local

Estas direcciones están definidas dentro de un enlace y sólo pueden ser utilizadas para compartir información entre estaciones pertenecientes al mismo enlace.

Estos paquetes no podrán ser retransmitidos por un *router*.

- **Direcciones *multicast***

Existen muchos tipos diferentes de direcciones *multicast*. Todas ellas siguen el formato indicado en la tabla siguiente:

8 bits	4 bits	4 bits	112 bits
11111111	000T	Ámbito	ID de grupo

Tabla 4: Formato de las direcciones *multicast*

T = 0 para una dirección permanente *multicast*, pública
T = 1 para una dirección *multicast* temporal.

- **Direcciones *anycast***

Es un tipo de direcciones experimental. Una dirección *anycast* es asignada a más de una interfaz, en vez de enviarse los datos a un servidor específico, se envía a una dirección genérica que será reconocida por servidores de un tipo concreto. El sistema entregará el paquete a aquel servidor que cumpla mejor ciertas condiciones impuestas, por ej. el servidor más cercano.

n bits	128-n bits
Prefijo de subred	00 ... 00

Tabla 5: Formato de las direcciones *anycast*

- **Direcciones especiales**

  - Dirección sin especificar

    Es una dirección con todos sus bytes a cero. A veces se utiliza como origen durante la inicialización cuando el sistema todavía no conoce su propia dirección.

    0:0:0:0:0:0:0:0

  - Dirección de bucle interno (*loopback*)

    En IPv6 se ha establecido que la dirección de bucle interno tendrá la forma siguiente:

    0:0:0:0:0:0:0:1

  - Direcciones IPv4

    Las direcciones de la versión 4 del protocolo IP que no admitan la conversión a la versión 6 podrán ser traducidas de la siguiente forma:

    0:0:0:0:0:FFFF:a.b.c.d

    donde a.b.c.d es la dirección IP original.

2.5.2. Formato de direcciones IPv6

- **Formato de cabecera IPv6**

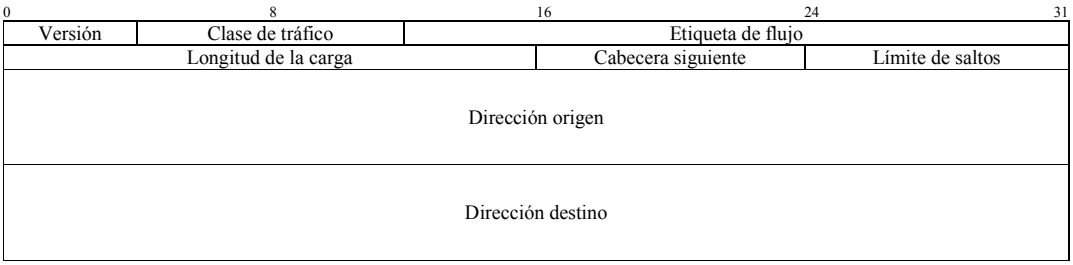


Figura 1: Cabecera IPv6

- Versión: versión del protocolo de Internet. 4 bits. Valor = 6
- Clase de tráfico: Campo de la clase de tráfico. 8 bits.
- Etiqueta de flujo: 20 bits.
- Longitud de la carga: Longitud restante a continuación de esta cabecera en octetos. 16 bits.
- Cabecera siguiente: Identifica el tipo de cabecera que sigue a la cabecera IPv6.
- Límite de saltos: Es decrementado en una unidad por cada nodo que reenvía el paquete. El paquete es descartado en cuanto llega a cero.
- Dirección origen: dirección de 128 bits del emisor el paquete.
- Dirección destino: Dirección de 128 bits del receptor del paquete (probablemente no el destinatario final si se incluye una cabecera de encaminamiento).

- **Cabeceras de extensión IPv6**

En IPv6 es posible codificar la información opcional del nivel internet mediante las cabeceras de extensión. Éstas se ubican entre la cabecera IPv6 y la cabecera del nivel superior. Existen un número reducido de tales cabeceras de extensión, cada una identificada por un valor "Cabecera siguiente" distinto. Un paquete IPv6 puede llevar cero, una o más cabeceras de extensión.

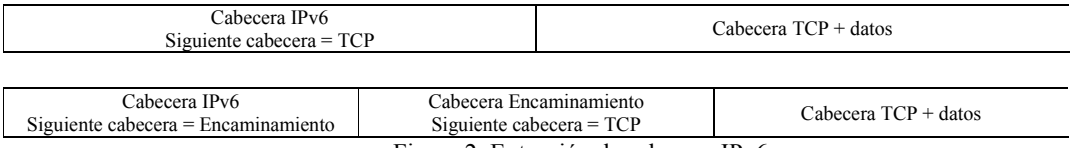


Figura 2: Extensión de cabecera IPv6

Por lo general las cabeceras de extensión no son examinadas o procesadas por ningún nodo a lo largo de la ruta (la cabecera de opciones *Hop by Hop* es una excepción), hasta que el paquete llega al nodo de destino (o a un

grupo de nodos en caso de *multicast*), estando éste identificado por el campo de Dirección de Destino de la cabecera IPv6.

Una implementación completa de IPv6 incluye la implementación de las siguientes cabeceras de extensión:

- Opciones *Hop-by-Hop*

  Esta cabecera se utiliza para llevar información opcional que debe ser examinada por todos los nodos que se encuentren en la ruta a seguir por el paquete.

- Encaminamiento

  La utiliza un nodo de origen para listar uno o más nodos intermedios por los cuales el paquete debe pasar.

- Fragmentación

  Esta cabecera es utilizada por el nodo emisor del paquete para enviar un paquete de mayor longitud del que encajaría en el MTU (*Maximum Transmission Unit*) del camino al destino (a diferencia de IPv4, la fragmentación IPv6 se realiza únicamente en origen, no en los *routers* que se encuentran a lo largo del camino).

- Opciones de destino

  Esta cabecera se utiliza para transportar información opcional que debe ser examinada sólo por el nodo de destino del paquete.

- Autenticación

- *Encapsulating Security Payload*

  Garantiza integridad de datos y confidencialidad.

- **Etiquetas de flujo**

Éstas pueden utilizarse en origen para etiquetar secuencias de paquetes, estas etiquetas indican a los *routers* IPv6 que deben tratarlos de forma especial.

- **Clases de tráfico**

  Los nodos origen o *routers* utilizan este campo para distinguir entre diferentes clases o prioridades de los paquetes IPv6.

2.5.3. Encaminamiento IPv6

El encaminamiento en IPv6 es casi idéntico a IPv4 bajo CIDR (Encaminamiento Inter Dominio Sin Clases) a excepción del tamaño de las direcciones (128 bits en lugar de 32). Con algunas extensiones sencillas los algoritmos de encaminamiento IPv4 (OSPF, RIP, IDRP, ISIS, etc.) se pueden utilizar para encaminar IPv6.

IPv6 también incluye extensiones de encaminamiento que soportan nuevas funcionalidades. Éstas incluyen:

- Selección de proveedor (basado en políticas, rendimiento, coste, etc.)
- Movilidad de terminales (encaminamiento a la localización actual)
- Auto-redireccionamiento (ruta a la nueva dirección)

La nueva funcionalidad de encaminamiento se obtiene creando secuencias de direcciones IPv6 utilizando la opción de encaminamiento IPv6. Un nodo origen IPv6 utiliza la opción de encaminamiento para listar uno o más nodos intermedios a "ser visitados" a lo largo del camino hacia el destino de un paquete.

2.5.4. Capacidad de calidad de servicio IPv6

Los campos de "Etiqueta de Flujo" y "Prioridad" de la cabecera IPv6 se pueden utilizar para identificar los paquetes para los que se solicita un tratamiento especial por parte de los *routers*, tales como calidad de servicio "no por defecto" o servicio en tiempo real. Estas capacidades son importantes para aplicaciones con restricciones en cuanto al retardo, *jitter* o *throughput*. Estas aplicaciones se describen comúnmente como "multimedia" o "de tiempo real".

2.5.5. Seguridad IPv6

El protocolo de Internet actual muestra cierto número de problemas de seguridad y carece de mecanismos de privacidad y autenticación por debajo de la capa de aplicación. IPv6 ofrece dos opciones integradas que proporcionan servicios de seguridad. Estas dos opciones pueden utilizarse independientemente o de forma conjunta para proporcionar diferentes niveles de seguridad a diferentes usuarios.

- El primer mecanismo, la "Cabecera de Autenticación IPv6", es una cabecera de extensión que proporciona autenticación e integridad a los datagramas IPv6. La extensión es independiente del algoritmo y soportará diferentes técnicas de autenticación, se propone el uso de MD5 para ayudar a asegurar interoperabilidad en Internet a nivel mundial.

- La segunda cabecera de extensión de seguridad es la *Encapsulating Security Header*. Este mecanismo proporciona integridad y confidencialidad a los datagramas IPv6. Es más sencilla que otros protocolos de seguridad similares (ISO, NLSP, SP3D…) pero es flexible e independiente del algoritmo. Para conseguir interoperabilidad en el Internet global, se utiliza DES CBC como algoritmo estándar en esta cabecera.

2.5.6. Referencias

- http://playground.sun.com/pub/ipng/html/INET-IPng-Paper.html
- http://playground.sun.com/pub/ipng/html/ipng-main.html
- http://www.ipv6.org
- *Understanding IPv6*" – David Morton
- *IP Next Generation Overview*. Robert M. Hinden
- *IPv6 What and Where It is*. Robert L. Fink. IP Journal.

  Descripción general de IPv6 con amplias referencias e implementaciones

3. Estado del IPv6 y soporte en REGETGA

3.1. En infraestructuras (de red y servidores):

3.1.1. *Routers*

Una de las características que posee RECETGA, es la variedad de equipamiento de *routing*. Así nos encontramos con *routers* Cisco (3725, 7206VXR, 1600, 2500, 1700), Juniper (M10) y Enterasys (XP2400).

En la mayoría de los casos, Cisco y Juniper, el soporte de las diferentes funcionalidades vienen dadas por la versión de software que tengan bien sea IOS para los equipos Cisco o JunOS para los equipos Juniper. En el caso de los Enterasys para tener soporte IPv6 es necesaria la instalación de un módulo hardware (http://www.enterasys.com/products/routing/ipv6/).

Vemos en la siguiente tabla la versión de software necesaria en ambas marcas de *router* para obtener las diversas características IPv6:

Capacidades	Cisco IOS (min. versión)	Juniper JunOS (min. versión)
IPv6 (Internet Protocol Version 6)	12.3(7)T	6.2
Servicios de conmutación IPv6	12.3(11)T	
IPv6 *routing*	12.3(7)T	
IPv6 servicios y gestión	12.3(11)T	
IPv6 multicast	12.3(11)T	
Traducción de protocolo NAT	12.3(2)T	
Servicios de túnel IPv6	12.3(7)T	
QoS IPv6 (Calidad de servicio)	12.3(2)T	
Capa de enlace de datos IPv6	12.3(2)T	

Tabla 6: Necesidades Software de los *routers*

Para los *routers* Cisco tenemos funcionalidad básica con la versión 12.3(7)T, sin embargo, teniendo en cuenta la necesidad de *multicast*, *netflow* y otras características utilizadas en los *routers*, se concluye que la versión mínima de software necesaria para el soporte IPv6 es la 12.3(11)T en el caso de *routers* Cisco y la 6.2R3 en el caso de Juniper.

3.1.2. *Switches* (Conmutadores Ethernet) y *bridges* (puentes)

- **Equipamiento *wired***

  En principio cualquier conmutador *ethernet* valdría para interconectar equipos que usen IPv6, dado que los *switches* son equipamiento de nivel 2 con lo que le da lo mismo el protocolo usado en la capa superior. Sin embargo surge una doble problemática:

- Gestión de los equipos: Para poder configurar remotamente el equipo y poder monitorizarlo usando IPv6 los equipos deberían permitir la configuración de una dirección IPv6 y esto, a día de hoy, no está disponible en todos los modelos (p. ej., el equipamiento 3com no lo soporta).

- MLD *snooping*: Esta es una característica "prescindible" y que todavía esta en fase de aprobación. Es la homóloga del IGMP *snooping*, usado por los conmutadores para aprender los puertos por los que tienen receptores de tráfico *multicast*. De este modo se consigue que, cuando un equipo final solicite recibir un determinado grupo *multicast*, el *switch* solamente encamine el tráfico *multicast* por aquellos puertos que tienen máquinas subscritas al mismo grupo *multicast*. Al usar *multicast* en IPv6 todos los puertos pertenecientes a una misma VLAN verán el tráfico *multicast,* tanto si están suscritos a dicho grupo, como si no lo están; esto es, para el tráfico *multicast* IPv6 el *switch* se comporta como un *hub* para una determinada VLAN

- **Equipamiento *wireless***

  Al igual que con el equipamiento fijo, al ser de nivel 2, no hemos detectado ningún tipo de problema en cuanto al paso transparente de paquetes IPv6. Los equipos usados han sido

  - Cisco AccessPoint 350 Series
  - Tarjetas Wireless 802.11b (Cisco y Compaq WL110)

  Sin embargo, no es posible configurar en los equipos *Access Point* direcciones IPv6 con lo que tampoco es posible gestionarlos mediante IPv6.

3.1.3. QoS[1]

Las técnicas de QoS utilizadas en IPv4 son válidas también para IPv6. Comentaremos, además, las características que el propio protocolo tiene y alguno de los intentos y propuestas para especificar y conseguir calidad de servicio a través de los distintos campos de la cabecera IPv6. La propia especificación del protocolo IPv6 provee de un campo "etiqueta de flujo" (*Flow Label*) para la provisión de QoS, sin embargo, dicha funcionalidad no ha sido estandarizada y, consecuentemente, no se encuentra disponible en el equipamiento existente.

Con el objetivo de conseguir calidad de servicio, al menos dos mecanismos diferentes han sido propuestos. El primero de ellos haciendo uso de las extensiones de cabecera salto a salto modificada, como mecanismo de transición hasta que el campo *Flow Label* fuese estandarizado. El otro, representa una posible especificación del

[1] Basado en el proyecto fin de carrera con título "Mecanismos de QoS en redes IPv4/IPv6 y su implantación en RECETGA" de Juan Villasuso Barreiro, 2002/2003

propio campo *Flow Label* enviado como borrador al IETF para su revisión y posible adopción (aunque a día de hoy dicho borrador ha expirado sin haberse estandarizado).

3.1.3.1. Extensión de Cabecera Salto a Salto modificada

Esta propuesta no emplea para nada el mencionado campo de etiqueta de flujo para proporcionar calidad de servicio. Esta basada en el modelo *IntServ* y se propuso como posible solución temporal hasta que la especificación del campo etiqueta de flujo estuviese aceptablemente desarrollado.

En la implementación de cualquier modelo de QoS, los recursos requeridos han de ser solicitados a todos los *routers* a lo largo del camino y es importante que estos den su visto bueno. Atendiendo a la especificación del protocolo IPv6, la extensión de cabecera salto a salto será procesada por todos los *routers* en el camino hacia el destino. Así todos los *routers* del recorrido verán cualquier información embebida en esta cabecera.

Las opciones tipo-longitud-valor (TLV – *type-length-value*) en la extensión de cabecera salto a salto no han sido explotadas en su totalidad. Así pues aprovechando estas opciones según nos convenga, sería posible especificar la información de los requisitos para cada flujo (el tipo y los recursos necesarios) a todos los *routers* intermedios. Los *routers* de forma individual podrían enviar determinados mensajes a la fuente si estos no fuesen quien de cumplir los requerimientos de recursos.

Atendiendo al RFC 2460, la especificación formal de IPv6, la extensión de cabecera salto a salto, está definida con un valor de cabecera siguiente (*Next Header*) igual a 0 dentro de la cabecera IPv6 y tiene el siguiente formato.

Figura 3: Cabecera *Next Header*

donde:

- ***Next Header***: es un campo de 8 bits que identifica el tipo de cabecera inmediatamente a continuación de la cabecera salto a salto.

- ***Hdr Extensión Length***: es un campo de 8 bits (*unsigned integer*) el cual especifica la longitud de las opciones de la cabecera salto a salto en unidades de 8 octetos, sin incluir los primeros 8 octetos.

- ***Options***: Es un campo de longitud variable, de longitud tal que la cabecera completa de opciones salto a salto es un entero múltiplo de 8 octetos.

A su vez la cabecera de opciones (*Options*) está compuesta por:

Option Type	Option Data Length	Option Data

- ***Option Type***: identificador del tipo de opción (8 bits)

- ***Option Data Length***: entero sin signo de 8 bits que representa la longitud del campo *Option Data* en octetos

- ***Option Data***: campo de longitud variable, que contiene datos específicos según el tipo de opción.

En la definición del protocolo fueron estandarizados algunos tipos de opciones. Usando valores de tipos de opciones no definidos esta aproximación divide el campo *option data* en un identificador de QoS (8 bits) un identificador de recursos (4 bits) que se emplea para describir el tipo de recursos solicitado (tasa constante, tasa media constante, retardo medio, retardo mínimo) y una lista de recursos requeridos.

3.1.3.2. Usando el campo *Flow Label*

Como ya se mencionó anteriormente, cuando el protocolo IPv6 fue diseñado se tuvieron muy en cuenta las carencias de su predecesor IPv4 en materia de QoS. Así en la cabecera IPv6 aparece un nuevo campo *Flow Label* (etiqueta de flujo) de 20 bits que será usado para proporcionar calidad de servicio. La especificación de dicho campo no está sujeto a ningún estándar a día de hoy aunque diferentes borradores han sido enviados a revisión al IETF para su posible estandarización.

Estos *drafts* difieren básicamente en la forma en la que el campo *Flow Label* de 20 bits es dividido. Por ejemplo `[draft-banerjee-flowlabel-ipv6-qos-03]` basándose en `[draft-conta-ipv6-flow-label-02]` propone una estructura del campo *Flow Label* en la que se divide el campo en 3 y 17 bits. Los 3 primeros bits son usados como selector del "modelo" y otros 17 bits tendrán un significado acorde al "modelo". Así,

- 000: Por defecto
- 001: Número aleatorio o pseudo-aleatorio
- 010: Es usada la extensión de cabecera salto a salto sin importar los 17 bits restantes.
- 011: Clasificador Multicampo. Sugiere usar los 17 bits restantes del campo como código identificador del comportamiento por salto (*Per Hop Behavior Ident. Code*). Posibilidad de mapeo de rangos de etiquetas.
- 100: Formato que incluye el número de puerto y el protocolo.
- 101: Nueva definición. draft-banerjee
- 110 y 111: Reservados para futuro uso.

Atendiendo al código selector 101, Banerjee propone una estructura mejorada de los restantes 17 bits de forma que cualquier aplicación podría solicitar los distintos parámetros de QoS a través de la etiqueta de flujo. Así como la pérdida de paquetes y el *jitter* se desea siempre que sea lo mínimo posible los posibles parámetros serían:

- Ancho de Banda (BW): 5 bits representando múltiplos de 32 Kbps.
- Requerimientos de Búfer: 5 bits representando múltiplos de 512 bytes.
- Retardo: 5 bits representando múltiplos de 4 nanosegundos.

Donde para todos los casos la fórmula a seguir sería valor = 2^B * unidad_base, siendo B el valor decimal correspondiente a los 5 bits. Además el bit 17 diferencia entre *soft real time applications* y *hard real time applications* y el bit 16 representa si los parámetros especifican valores máximos o mínimos.

Finalmente, cabría preguntarse ¿qué debería hacer un *router* con las etiquetas de flujo para las que no tiene estado o cuál sería la acción por defecto para etiquetas de flujo distintas de cero?. La especificación de IPv6 sugiere las siguientes posibles soluciones:

- El *router* puede ignorar la etiqueta de flujo
- Los datagramas IPv6 pueden llevar información de inicialización de flujo en sus opciones

El RFC-1809 propone que la acción por defecto debería ser tal que si un *router* recibe un datagrama con la etiqueta de flujo distinta de cero, éste debe el datagrama como si la etiqueta fuese cero. Como parte del encaminamiento, el *router* examinará cualquier opción salto a salto y aprenderá si el datagrama requiere un tratamiento especial. Las opciones podrían variar desde la indicación de que el datagrama va a ser descartado a la información de estado que el *router* debería tener.

3.2. En software

3.2.1. Sistemas operativos

Dada la cantidad de sistemas operativos que pueden estar disponibles en la RECETGA, es imposible hacer un estudio detallado de todos ellos. Por eso, en esta sección se analizarán aquellos sistemas operativos utilizados en la prestación de servicios del CESGA.

3.2.1.1. Windows XP

- **Soporte**

Todas las versiones de XP incluyen IPv6 preinstalado, pero es preciso habilitarlo.

- **Instalación**

  Para habilitarlo es necesario ejecutar con privilegios de administrador, lo siguiente:

```
prompt> ipv6 install
```

Para comprobar que se ha instalado correctamente podemos realizar:

```
prompt> ipv6 if

Interfaz 4: Ethernet: conexión de área local
   usa unidad de detección de equipos cernamos (neighbor discovery)
   utiliza descubrimiento de enrutador
   dirección de capa de vínculo: 00-03-47-cc-30-6d
   preferred link-local fe80::203:47ff:fecc:306d, duración infinite
   multidifusión interface-local ff01::1, 1 referencias, no se puede
      informar
   multidifusión link-local ff02::1, 1 referencias, no se puede informar
   multidifusión link-local ff02::1:ffcc:306d, 1 referencias, último
      informe
   vínculo MTU 1500 (vínculo MTU verdadero 1500)
   límite de saltos actual 128
   tiempo accesible 41000 ms (base 30000 ms)
   intervalo de retransmisión 1000 ms
   transmisiones DAD 1
   (…)
```

Mostrando la configuración y las direcciones IPv6 autoconfiguradas para cada interfaz existente.

- **Configuración**

  El comando `IPv6.exe` se puede utilizar para comprobar y configurar manualmente interfaces, direcciones y rutas.

```
prompt> ipv6 help

Uso:   ipv6 [-v] if [ifindex]
       ipv6 [-p] ifcr v6v4 v4scr v4dst [nd] [pmld]
       ipv6 [-p] ifcr 6over4 v4src
       ipv6 [-p] ifc ifindex [forwards] [-forwards] [advertises] [-
         advertises] [mtu #bytes] [identificador-sitio] [preference P]
       ipv6 rlu ifindex v4dst
       ipv6 [-p] ifd ifindex
       ipv6 [-p] adu ifindex/address [life validlifetime[/preflifetime]]
         [anycast] [unicast]
       ipv6 nc [ifindex [address]]
       ipv6 ncf [ifindex [address]]
       ipv6 rc [ifindex [address]]
       ipv6 rcf [ifindex [address]]
       ipv6 bc
       ipv6 [-v] rt
       ipv6 [-p] rtu prefix ifindex[/address] [life valid[/pref] [preference
         p] [publish] [age] [spl] [SitePrefixLength]
       ipv6 spt
       ipv6 spu prefix ifindex [life L]
       ipv6 gp
       ipv6 [-p] gpu [valor de parámetro] … (vea -?)
       ipv6 renew [ifindex]
       ipv6 ppt
       ipv6 [-p] ppu prefix precedente P srclabel SL [dstlabel DL]
       ipv6 [-p] ppd prefix
       ipv6 [-p] reset
       ipv6 install
       ipv6 uninstall
```

3.2.1.2. Windows 2000

- **Soporte**

  El software *Microsoft IPv6 Technology Preview* es un derivado de la implementación de IPv6 del grupo de investigación de Microsoft. Puede utilizarse para aprender y experimentar en aquellos casos en los que no se pueden utilizar las implementaciones actuales por algún que otro motivo. Microsoft no proporciona soporte IPv6 con calidad de producción para Windows 2000 (si existen para Windows Server 2003, Windows XP SP1, Windows CE .NET[2]).

  IPv6 está soportado en sus funciones básicas por este sistema operativo, pero dicho soporte sólo es completo (no de producción) en los sistemas operativos actualizados con la última versión del *service pack*.

- **Instalación**

  - Windows 2000 con *service pack* 1

    Esta instalación es válida para cualquier versión comercial de Windows 2000 que tenga instalado el *Service Pack* 1.

[2] http://www.microsoft.com/windowsserver2003/technologies/ipv6/default.mspx#implementations

1. Ejecutar el fichero `tpipv6-001205.exe` desde:

   http://msdn.microsoft.com/downloads/sdks/platform/tpipv6/download.asp

   Probablemente sea preciso reiniciar el ordenador si así lo indica.

2. Desde el escritorio, pulsar el botón derecho de Entorno de Red, pulsar propiedades, y de nuevo, con el botón derecho sobre la tarjeta de red en la que se quiere instalar IPv6, haciendo clic en propiedades.

3. Hacer clic sobre `Instalar`, y seleccionar protocolo y añadir. Escoja `Microsoft IPv6` y pulsar sobre OK. Pulsar sobre cerrar. Para comprobar que ha sido correctamente instalado, usar:

```
prompt> ipv6 if

Interfaz 4: Ethernet: conexión de área local
   usa unidad de detección de equipos cernamos (neighbor discovery)
   utiliza descubrimiento de enrutador
   dirección de capa de vínculo: 00-03-47-cc-30-6d
   preferred link-local fe80::203:47ff:fecc:306d, duración infinite
   multidifusión interface-local ff01::1, 1 referencias, no se puede
      informar
   multidifusión link-local ff02::1, 1 referencias, no se puede informar
   multidifusión link-local ff02::1:ffcc:306d, 1 referencias, último
      informe
   vínculo MTU 1500 (vínculo MTU verdadero 1500)
   límite de saltos actual 128
   tiempo accesible 41000 ms (base 30000 ms)
   intervalo de retransmisión 1000 ms
   transmisiones DAD 1
   (…)
```

Se muestran la configuración y las direcciones IPv6 adquiridas (auto-configuradas) para cada interfaz de red existente.

- Windows 2000 con *Service Pack* 2, 3 o 4

  Para instalar *Microsoft IPv6 Technology Preview* en Windows 2000 con SP2:

  1. Descargar el archivo `tpipv6-001205.exe` del siguiente URL:

     http://msdn.microsoft.com/downloads/sdks/platform/tpipv6/download.asp

  2. Salvarlo en un directorio local (`C:\IPv6TP`).
  3. Ejecutar `Tpipv6-001205.exe` y extraer los ficheros a dicha ubicación.

4. Ejecutar `setup.exe -x` y extraer los ficheros a un subdirectorio (`C:\IPv6TP\files` por ejemplo).
5. Dentro del subdirectorio se encuentra un archivo con nombre `Hotfix.inf`, abrirlo con un editor de texto.
6. En la sección `[Version]` de dicho archivo cambiar la línea (según se trate del *service pack* 2, 3 o 4):

   SP2: `NTServicePackVersion=256` a `NTServicePackVersion=512`
   SP3: `NTServicePackVersion=256` a `NTServicePackVersion=768`
   SP4: `NTServicePackVersion=256` a `TServicePackVersion=1024`

7. Guardar los cambios.
8. Desde el subdirectorio, ejecutar `Hotfix.exe`.
9. Reiniciar el ordenador
10. Una vez que el ordenador ha reiniciado, ya se puede añadir el protocolo IPv6 al interfaz de red deseado.

- **Configuración**

  Los cambios de configuración no son permanentes y se pierden con un reinicio del *host* o del protocolo IPv6. Se pueden guardar dichos cambios colocándolos como líneas de comando en un fichero de comandos (.cmd) que se ejecuta después del reinicio del protocolo IPv6. Para ejecutarlo automáticamente después del reinicio del ordenador, utilizar la herramienta "Tareas Programadas" para ejecutar el fichero de comandos después del inicio del ordenador.

  `IPv6.exe` se puede utilizar para comprobar y configurar manualmente interfaces, direcciones y rutas (usuarios avanzados).

```
prompt> ipv6 help

Uso:   ipv6 [-v] if [ifindex]
       ipv6 [-p] ifcr v6v4 v4scr v4dst [nd] [pmld]
       ipv6 [-p] ifcr 6over4 v4src
       ipv6 [-p] ifc ifindex [forwards] [-forwards] [advertises] [-
         advertises] [mtu #bytes] [identificador-sitio] [preference P]
       ipv6 rlu ifindex v4dst
       ipv6 [-p] ifd ifindex
       ipv6 [-p] adu ifindex/address [life validlifetime[/preflifetime]]
         [anycast] [unicast]
       ipv6 nc [ifindex [address]]
       ipv6 ncf [ifindex [address]]
       ipv6 rc [ifindex [address]]
       ipv6 rcf [ifindex [address]]
       ipv6 bc
       ipv6 [-v] rt
       ipv6 [-p] rtu prefix ifindex[/address] [life valid[/pref] [preference
         p] [publish] [age] [spl] [SitePrefixLength]
       ipv6 spt
       ipv6 spu prefix ifindex [life L]
       ipv6 gp
       ipv6 [-p] gpu [valor de parámetro] … (vea -?)
       ipv6 renew [ifindex]
       ipv6 ppt
       ipv6 [-p] ppu prefix precedente P srclabel SL [dstlabel DL]
       ipv6 [-p] ppd prefix
       ipv6 [-p] reset
       ipv6 install
       ipv6 uninstall
```

Se puede comprobar el correcto funcionamiento de la pila IPv6 con:

```
prompt> ping6 ::1

Haciendo ping ::1
de ::1 con 32 bytes de datos:

Respuesta desde ::1: bytes=32 tiempo<1m
Respuesta desde ::1: bytes=32 tiempo<1m
Respuesta desde ::1: bytes=32 tiempo<1m
Respuesta desde ::1: bytes=32 tiempo<1m

Estadísticas de ping para ::1:
       Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
       Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

::1 es la dirección de *loopback* en IPv6.

Existen otros comandos y utilidades como `tracert6`, `telnet6`, `ftp6`, `ipsec6`, `ttcp6` y `6to4-cfg`.

La aplicación `net.exe` se puede utilizar para iniciar o parar el protocolo IPv6. Reiniciar el protocolo IPv6 causa el mismo efecto que un reinicio del PC, lo que podría cambiar los números de interfaz.

`Net.exe` dispone de muchos subcomandos, con sus conjuntos de argumentos y opciones. Son de interés para el proyecto los siguientes:

`net stop tcpip6`
Para el protocolo IPv6 y lo descarga de memoria.

`net start tcpip6`
Inicia el protocolo IPv6 en caso de que esté parado.

3.2.1.3. Linux

Las máquinas empleadas en el proyecto con S.O. Linux fueron instaladas con las distribuciones RedHat 9.0 y posteriormente Fedora Core 2.

```
RedHat 9.0:       kernel 2.4.20-8
Fedora Core 2:    kernel 2.6.6
```

Ambas distribuciones incluyen de "serie" soporte IPv6, como un módulo *kernel*, con lo que resulta necesaria la compilación del *kernel* partiendo de las fuentes (es necesario, simplemente, compilar el soporte IPv6 como módulo, eso sí, partiendo de los fuentes correspondientes a cada distribución).

Ambas distribuciones son "gemelas" en el sentido que la 2ª es la evolución de la 1ª, con lo que, tanto la estructura de directorios (ubicación de los ficheros de configuración) como los nombres de los mismo, se han mantenido.

Ninguna de las máquinas Linux actuaba como *router* con lo que no fue necesaria la instalación ni la configuración de ningún software adicional. "`radvd`"

Por tanto los únicos pasos necesarios para la configuración de IPv6 serían:

1. Añadir al fichero "`/etc/sysconfig/network`"

   ```
   NETWORKING_IPV6=yes
   ```

2. Añadir a `/etc/sysconfig/network-scripts/ifcfg-eth0`

   ```
   IPV6INIT=yes
   ```

Salida del comando "`ifconfig -a`" (Aquí vemos como el interfaz `eth0` no tiene habilitado IPv6):

```
eth0      Link encap:Ethernet  HWaddr 00:B0:D0:F3:5E:40
          inet addr:193.144.34.136  Bcast:193.144.34.255
Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:70681549 errors:0 dropped:0 overruns:20051 frame:0
          TX packets:76979404 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:88645778 (84.5 Mb)  TX bytes:2715091311 (2589.3 Mb)
          Interrupt:16 Base address:0xecc0 Memory:fe2ff000-fe2ff038

eth1      Link encap:Ethernet  HWaddr 00:B0:D0:F3:5E:41
          inet addr:193.144.36.137  Bcast:193.144.36.143
Mask:255.255.255.240
          inet6 addr: fe80::2b0:d0ff:fef3:5e41/64 Scope:Link
          inet6 addr: 2001:720:1210:e:2b0:d0ff:fef3:5e41/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16731438 errors:2 dropped:0 overruns:0 frame:2
          TX packets:3812373 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:2628372523 (2506.6 Mb)  TX bytes:1682028974 (1604.1 Mb)
          Interrupt:17 Base address:0xec80 Memory:fe2fe000-fe2fe038

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:139450032 errors:0 dropped:0 overruns:0 frame:0
          TX packets:139450032 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:402722033 (384.0 Mb)  TX bytes:402722033 (384.0 Mb)

sit0      Link encap:IPv6-in-IPv4
          NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

3.2.1.4. Solaris

La versión usada en este proyecto es la `Solaris 2.8` (`SunOS 5.8`) que viene de "serie" con soporte para IPv6 integrado. Obviamente las versiones posteriores también soportan IPv6 de forma nativa. Como se explica en la primera parte de este informe existen varias posibilidades a la hora de configurar el direccionamiento IPv6 en las maquinas corriendo el SO Solaris. La usada para el proyecto fue la denominada auto-configuración.

Pasos a seguir:

- Determinar el nombre de los interfaces en los cuales queremos tener direccionamiento IPv6. Así podemos tener `lo0`, `hme0`, `elx10` ... (si dudamos "`#ifconfig -a`" proporciona el nombre que usa la maquina para referirse a los interfaces de red).

- Crear un fichero vacío bajo el directorio `/etc` llamado `hostname6.nombre_del_interface` (así tendríamos por ejemplo

`/etc/hostname6.hme0`). Una forma sencilla de hacer esto es usando el comando "`touch`" (#`touch /etc/hostname6.hme0`)

- Reiniciar, el demonio "`ndpd`" (*`Network Discovery Protocol Daemon`*) cogerá el prefijo IPv6 correspondiente del *router* y usará el formato *64-bit Extended Unique Identifier* (`EUI-64`) para asignar la correspondiente dirección IPv6 al interfaz seleccionado.

Nota: La ruta por defecto quedará también establecida de forma automática.

Un paso más necesario para que nuestra máquina use el `DNS` para buscar máquinas con direcciones IPv6 será añadir a la entrada "`ipnodes`" y "*hosts*" del fichero de configuración "`/etc/nsswitch.conf`" la clave "`dns`"

Así, tendremos en `/etc/nsswitch.conf`

```
hosts:       files  dns
ipnodes:     files  dns
```

Salida del comando "`ifconfig -a`"

```
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
       inet 127.0.0.1 netmask ff000000
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
       inet 193.144.44.10 netmask ffffff00 broadcast 193.144.44.255
lo0: flags=2000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
       inet6 ::1/128
hme0: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
       inet6 fe80::a00:20ff:fea2:e27d/10
hme0:1: flags=2080841<UP,RUNNING,MULTICAST,ADDRCONF,IPv6> mtu 1500 index 2
       inet6 2001:720:1210:c:a00:20ff:fea2:e27d/64
```

3.2.1.5. HP-UX

HP-UX 11i v.2 proporciona soporte nativo IPv6 con las siguientes características:

- Doble pila IPv4/IPv6 la cual permite que las aplicaciones IPv4 actuales no necesiten ser modificadas.
- *Tunneling* IPv6, que facilita a los *hosts* IPv4/IPv6 conectar con otros *hosts* IPv4/IPv6 sobre redes IPv4 existentes.
- Autoconfiguración sin estado de direcciones IPv6
- Procedimiento de Descubrimiento de Vecino (*Neighbor Discovery*) IPv6 (que incluye descubrimiento de *router* y detección de direcciones duplicadas).
- TCP/UDP sobre IPv6, PMTUv6, ICMPv6, IPv6 MIBs y APIs de *sockets*.
- Utilidades de configuración y resolución de problemas de red para IPv4 e IPv6: `ifconfig`, `netstat`, `ping`, `route`, `ndd`, `ndp` (el comando `neighbor-discovery` sólo para IPv6).

- El fichero `netconf-ipv6` almacena los parámetros de configuración IPv6. El fichero de configuración `/etc/rc.config.d/netconf-ipv6` almacena información de configuración de una forma similar a la que se utilizaba en el fichero `/etc/rc.config.d/netconf` para IPv4.
- El fichero `/etc/hosts` soporta direcciones IPv6 e IPv4. Este fichero contiene direcciones IP con sus correspondientes nombres de *host*. El archivo puede contener direcciones IPv4 e IPv6 para el mismo *host*. Por ejemplo:

```
15.15.15.15       hpindon
3ffe:1111::1234   hpindon     hpindon6
```

- El "*Name Service Switch*": `/etc/nsswitch.conf` es un fichero de configuración para el "*name service switch*". Una nueva entidad, los `ipnodes`, especifica que servicios de nombre resuelven los nombres y direcciones de *host*.

Deben tenerse en cuenta las siguientes limitaciones:

- `setparms` no ha sido mejorado para la configuración IPv6
- Limitación de *multihome*: En ausencia de un *router* que anuncie prefijos, ningún interfaz más se configurará con direcciones IPv6 en un *host* con interfaces de red físicos múltiples.
- NIS+, NIS y NFS no están soportados actualmente en IPv6.

3.2.1.6. *BSD (FreeBSD, NetBSD, OpenBSD

Todos estos sistemas operativos poseen una configuración muy similar para habilitar el soporte de IPv6. Las opciones para habilitarlo así como el modo de funcionamiento y la forma de configurar los interfaces (automática o manual) se lleva a cabo usando un único fichero de configuración "`/etc/rc.conf`". La única diferencia entre ellos es en nombre de las opciones dentro de dicho fichero y el número de ellas.

Así, independientemente de la forma de operar de la máquina, bien como *router* o como estación, será necesario habilitar (poner a "YES" la correspondiente directiva dentro del fichero de configuración). Esto es IMPRESCINDIBLE.

```
----------------------/etc/rc.conf----------------------------
      ipv6_enable="YES"
            # by default it's "NO"
--------------------------------------------------------------
```

Dado que no se han usado equipos *BSD durante en la realización del proyecto no comentaremos los pasos necesarios para la correcta configuración de cada uno de los SO's.

3.2.2. Aplicaciones

Al igual que en la sección de sistemas operativos, no es posible analizar en detalle todas y cada una de las aplicaciones existentes que utilicen comunicaciones. Por ello, el estudio se centra exclusivamente en las aplicaciones necesarias para prestar servicios de red.

3.2.2.1. Servidores

- **DNS**

  - Descripción

    El servicio de nombres es un servicio cuya finalidad es el mapeo entre los nombres de las máquinas y su/s correspondientes direcciones IP. Dicho servicio, se vuelve con el paso a IPv6 más imprescindible si cabe dado que al ser direcciones de 128 bits no resultan nada fáciles de recordar, aunque se use un direccionamiento "particularizado".

  - Soporte IPv6. Problemas detectados y soluciones.

    Uno de los servidores de nombres de uso más extendido es el `bind` (www.isc.org) que a partir de la versión 9 tiene pleno soporte para su uso con direcciones IPv6. Para las versiones anteriores es necesario aplicar un parche para añadir el soporte de direcciones IPv6. Notar que aunque el DNS no tenga soporte para IPv6, no implica que no sea capaz de proporcionar la resolución de direcciones a los clientes.

    Uno de las peculiaridades detectadas al usar direccionamiento IPv6 es que el DNS devuelve en primera instancia direcciones IPv6 en el caso de que ambos mapeados estén disponibles. Así pues para un nombre de máquina "`ejemplo.cesga.es`", que tenga dirección IPv4 y dirección IPv6, el DNS devolverá la correspondiente dirección IPv6.

  - Ejemplos de configuración

    Configuración – Nuevos parámetros y entradas.

    a) `named.conf`: Es el fichero donde se especifican las opciones globales del servidor y los ficheros que contienen las diferentes zonas (dominios) y su correspondiente resolución inversa. El único parámetro que es necesario habilitar para que `bind` pueda soportar consultas usando IPv6 como transporte es "*`listen-on-v6 { any;};`*" dentro de *`options`*

```
options {
       //#ident "@(#)cmd-inet:common/cmd/cmd-inet/etc/named.boot 1.1.3.1"
       //#ident "$header: named.boot 1.2 91/06/26 $"
       //
       //boot file for name server
       //
       //type domain source file or host
       //
       directory     "/etc/named";
       //
       //
       host-statistics yes;
       multiple-cnames yes;
       statistics-file "/var/log/named.stats";
       pid-file "/var/run/named.pid";
       listen-on-v6 { any;}; // Escucha peticiones por la dirección IPv6
       notify yes;
};
logging {
       category "default" { "default_syslog"; "default_debug";
                     };
       channel queries {
              file "/tmp/query.log";
       };
       category queries { queries; };
       };
zone "." {
       type hint;
       file "root.hint";
};
zone "localhost" {
       type master;
       file "db.localhost";
};
zone "0.0.127.in-addr.arpa" {
       type master;
       file "db.127.0.0";
       notify no;
};
zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.int" {
       type master;
       file "db.0000:0000:0000:0000.ip6.int";
       notify no;
};
zone "\[x0000000000000000/64].ip6.arpa" {
       type master;
       file "db.0000:0000:0000:0000.ip6.arpa";
       notify no;
};
zone "ipv6.cesga.es" {
       type master;
       file "db.ipv6.cesga.es";
       notify yes;
};
zone "c.0.0.0.0.1.2.1.0.2.7.0.1.0.0.2.ip6.int" {
       type master;
       file "db.2001:0720:1210:000c.ip6.int";
       notify yes;
};
zone "\[x200107201210000c/64].ip6.arpa" {
       type master;
       file "db.2001:0720:1210:000c.ip6.arpa";
       notify yes;
};
```

La configuración de las zonas "directas es exactamente la misma tanto para IPv4 como para IPv6". Lo único que cambia es la definición de las zonas de la resolución inversa. Aquí hemos usado tanto el formato `Bitlabels` o `Bitstrings` (*ipv6.arpa*) y el formato `Nibble bit` (*ipv6.int*) por compatibilidad que sería el equivalente en IPv4 de *in-addr.arpa*. El formato *Bitlabels* se ha desechado con lo que la delegación de resolución inversa se proporciona en formato `Nibble bit` tanto para *ipv6.int* como *ipv6.arpa*

b) Ficheros de zonas: Contienen las entradas en su forma directa o inversa de cada uno de los dominios o subdominios. Se usan tanto AAAA o A6 (este último relegado a uso experimental) para especificar una dirección IPv6

Así el fichero "`db.ipv6.cesga.es`" contiene las entradas correspondientes para el subdominio "`ipv6.cesga.es`". El fichero tiene el siguiente formato.

```
$ORIGIN ipv6.cesga.es.
@       4h      IN      SOA     atmos.cesga.es. comunicaciones.cesga.es. (
                2003061101      ; serial
                86400           ; refresh
                7200            ; retry
                2592000         ; expire
                172800          ; minimum
                )
;
                IN      NS      atmos.ipv6.cesga.es.
                IN      MX      10 atmos.ipv6.cesga.es.
;
$TTL 1h
;
atmos           IN      A               193.144.44.43
                IN      AAAA            2001:720:1210:c:a00:20ff:fe89:62b6
                IN      A6      0       2001:720:1210:c:a00:20ff:fe89:62b6
router1                 IN      AAAA            2001:720:1210:c::1
                IN      A6      0       2001:720:1210:c::1
router2                 IN      AAAA            2001:720:1210:a::1
                IN      A6      0       2001:720:1210:a::1
trevize                 IN      A               193.144.44.26
                IN      AAAA            2001:720:1210:c:203:baff:fe02:8359
                IN      A6      0       2001:720:1210:c:203:baff:fe02:8359
medulio                 IN      A               193.144.44.12
                IN      AAAA            2001:720:1210:c:200:e2ff:fe19:f963
                IN      A6      0       2001:720:1210:c:200:e2ff:fe19:f963
stream          IN      A               193.144.44.19
                IN      AAAA            2001:720:1210:c:230:5ff:fe1a:4ca7
                IN      A6      0       2001:720:1210:c:230:5ff:fe1a:4ca7
sonda           IN      A               193.144.44.17
                IN      AAAA            2001:720:1210:c:a00:20ff:fe9f:9cc0
                IN      A6      0       2001:720:1210:c:a00:20ff:fe9f:9cc0
tambre          IN      A               193.144.44.11
                IN      AAAA            2001:720:1210:c:290:27ff:fed3:6490
                IN      A6      0       2001:720:1210:c:290:27ff:fed3:6490
ftp             IN      AAAA            2001:720:1210:e:2b0:d0ff:fef3:5ec8
                IN      A6      0       2001:720:1210:e:2b0:d0ff:fef3:5ec8
                IN      A               193.144.36.138
windowsmedia    IN      AAAA            2001:720:1210:e:230:5ff:fe19:719d
                IN      A6      0       2001:720:1210:e:230:5ff:fe19:719d
                IN      A               193.144.36.135
bream           IN      A               193.144.44.37
                IN      AAAA            2001:720:1210:c:204:76ff:fe92:feba
                IN      A6      0       2001:720:1210:c:204:76ff:fe92:feba
;
```

En cuanto a la resolución inversa tendremos los ficheros "*db.2001:0720:1210:000c.ip6.arpa*"

```
$ORIGIN \[x200107201210000c/64].ip6.arpa.
@      4h     IN     SOA    atmos.cesga.es. comunicaciones.cesga.es. (
              2003061101   ; serial
              86400        ; refresh
              7200         ; retry
              2592000            ; expire
              172800       ; minimum
              )
;
              IN     NS     atmos.cesga.es.
;
$TTL 1h
;
\[x0a0020fffe8962b6/64]          IN     PTR    atmos.ipv6.cesga.es.
\[x0000000000000001/64]          IN     PTR    router1.ipv6.cesga.es.
\[x0203bafffe028359/64]          IN     PTR    trevize.ipv6.cesga.es.
\[x0200e2fffe19f963/64]          IN     PTR    medulio.ipv6.cesga.es.
\[x023005fffe1a4ca7/64]          IN     PTR    stream.ipv6.cesga.es.
\[x0a0020fffe9f9cc0/64]          IN     PTR    sonda.ipv6.cesga.es.
\[x029027fffed36490/64]          IN     PTR    tambre.ipv6.cesga.es.
\[x020476fffe92feba/64]          IN     PTR    bream.ipv6.cesga.es.
```

y "*db.2001:0720:1210:000c.ip6.int*"

```
$ORIGIN c.0.0.0.0.1.2.1.0.2.7.0.1.0.0.2.ip6.int.
@      4h     IN     SOA    atmos.cesga.es. comunicaciones.cesga.es. (
              2003061101   ; serial
              86400        ; refresh
              7200         ; retry
              2592000            ; expire
              172800       ; minimum
              )
;
              IN     NS     atmos.cesga.es.
;
$TTL 1h
;
6.b.2.6.9.8.e.f.f.f.0.2.0.0.a.0        IN     PTR    atmos.ipv6.cesga.es.
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0        IN     PTR    router1.ipv6.cesga.es.
9.5.3.8.2.0.e.f.f.f.a.b.3.0.2.0        IN     PTR    trevize.ipv6.cesga.es.
3.6.9.f.9.1.e.f.f.f.2.e.0.0.2.0        IN     PTR    medulio.ipv6.cesga.es.
7.a.c.4.a.1.e.f.f.f.5.0.0.3.2.0        IN     PTR    stream.ipv6.cesga.es.
0.c.c.9.f.9.e.f.f.f.0.2.0.0.a.0        IN     PTR    sonda.ipv6.cesga.es.
0.9.4.6.3.d.e.f.f.f.7.2.0.9.2.0        IN     PTR    tambre.ipv6.cesga.es.
a.b.e.f.2.9.e.f.f.f.6.7.4.0.2.0        IN     PTR    bream.ipv6.cesga.es.
```

Resta indicar que usando esta notación, las direcciones se leen al revés tal y como se hace en IPv4. Así pues la máquina "`atmos.cesga.es`" tendría una dirección IPv6 *"2001:0720:1210:000c:0a00:20ff:fe89:62b6"* que en su forma compacta sería *"2001:720:1210:c:a00:20ff:fe89:62b6"*.

- **FTP**

Han sido varios los servidores de `ftp` probados a lo largo del proyecto.

- `Pureftpd`: Soporte nativo de IPv6 http://www.pureftpd.org

- `Proftpd`: Para conseguir que soporte IPv6 es necesario aplicar un parche a los fuentes de software. Sin embargo a partir de la versión 1.9.2rc2 proporciona soporte nativo de IPv6 http://proftpd.linux.co.uk/

- `Vsftpd`: Es un servidor de `ftp` seguro, estable y extremadamente rápido. Soporta IPv6 de forma nativa desde la versión 0.2.0 http://vsftpd.beasts.org/

Se ha escogido este último por sus características y por tener un fichero de configuración muy "amigable". Opciones como servidor anónimo, con directorio de *upload*, o "*chrooting*" selectivo de usuarios son simplemente habilitados bajo una directiva de configuración.

A continuación se muestra un fichero de configuración del servidor `dovecot` ("*standalone*").

```
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are very paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
#
# Allow anonymous FTP?
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you
will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
#
# Make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES
#
# If you want, you can arrange for uploaded anonymous files to be owned by
# a different user. Note! Using "root" for uploaded files is not
# recommended!
#chown_uploads=YES
#chown_username=whoever
#
# You may override where the log file goes if you like. The default is
shown
# below.
#xferlog_file=/var/log/vsftpd.log
#
# If you want, you can have your log file in standard ftpd xferlog format
xferlog_std_format=YES
#
# You may change the default value for timing out an idle session.
#idle_session_timeout=600
#
# You may change the default value for timing out a data connection.
#data_connection_timeout=120
```

… (continua en página siguiente)

```
… (viene de página anterior)

#
# It is recommended that you define on your system a unique user which the
# ftp server can use as a totally isolated and unprivileged user.
#nopriv_user=ftpsecure
#
# Enable this and the server will recognise asynchronous ABOR requests. Not
# recommended for security (the code is non-trivial). Not enabling it,
# however, may confuse older FTP clients.
#async_abor_enable=YES
#
# By default the server will pretend to allow ASCII mode but in fact ignore
# the request. Turn on the below options to have the server actually do
ASCII
# mangling on files when in ASCII mode.
# Beware that turning on ascii_download_enable enables malicious remote
parties
# to consume your I/O resources, by issuing the command "SIZE /big/file" in
# ASCII mode.
# These ASCII options are split into upload and download because you may
wish
# to enable ASCII uploads (to prevent uploaded scripts etc. from breaking),
# without the DoS risk of SIZE and ASCII downloads. ASCII mangling should
be
# on the client anyway..
#ascii_upload_enable=YES
#ascii_download_enable=YES
#
# You may fully customise the login banner string:
#ftpd_banner=Welcome to blah FTP service.
#
# You may specify a file of disallowed anonymous e-mail addresses.
Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd.banned_emails
#
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
#chroot_list_enable=YES
# (default follows)
#chroot_list_file=/etc/vsftpd.chroot_list
#
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror"
assume
# the presence of the "-R" option, so there is a strong case for enabling
it.
#ls_recurse_enable=YES

pam_service_name=vsftpd
userlist_enable=YES
#enable for standalone mode
listen_ipv6=YES
tcp_wrappers=YES
#chroot_local_user=YES
```

Destacar que para que funcione como demonio "*standalone*" y en IPv6 es necesario añadir la directiva

```
listen_ipv6=YES
```

- **HTTP (*HyperText Transfer Protocol*)**

  Los servidores web usados han sido dos:

  - `Apache 1.3.x`: Soporte de IPv6 a través de un parche. En el caso de su uso con IPv6 sería recomendable el salto a las versiones 2.0.x

  - `Apache 2.0.x`: Soporte IPv6 nativo (proporcionado por la propia aplicación). Recomendado.

  Ambos están disponibles en http://httpd.apache.org/ En cuanto al fichero de configuración no hay grandes cambios salvo los relacionados con el "*binding*" del servicio a una o todas las IP's que el servidor tenga y en cuanto a sí el servidor es accesible vía IPv6 o no.

  Dado que en las versiones de Linux usadas esta habilitado por defecto `--enable-v4-mapped` no sería necesario ningún cambio en la configuración del fichero "`httpd.conf`" del servidor `apache`. Sin embargo en sistemas `*BSD` por defecto no fue compilado con esta directiva, salvo se instale usando los *ports*.

  ```
  Listen [::]:80
  ```

  Si queremos que el servidor use *sockets* diferentes para IPv4 e IPv6 lo haríamos deshabilitando las direcciones `v4-mapped` y en la configuración usamos

  ```
  Listen [::]:80
  Listen 0.0.0.0:80
  ```

- **SSH (*Secure Shell*)**

  En este caso el software usado ha sido `Openssh` ftp://ftp.openbsd.org/pub/OpenBSD/OpenSSH/portable/. Este software provee soporte IPv6 nativo desde el año 2000, con lo que cualquier versión relativamente reciente lo soporta sin problema alguno.

  Cabe destacar, y ajeno al soporte IPv6 que, siendo `ssh` un servicio sobre el que se van a basar otros, para conferir seguridad, sería muy recomendable tener instalada siempre una versión lo más reciente posible dado que usa las librerías proporcionadas por `openssl` http://www.openssl.org/ y existen fallos de seguridad recientes relacionados con ella.

- **X-Windows**

  Es un protocolo inseguro y que pocos usuarios ahora mismo usan de forma encaminada. Lo que se suele hacer es tunelizar los "*displays*" gráficos en una sesión `ssh` con lo que el soporte IPv6 es proporcionado por `ssh`. Esto, a parte de conferir seguridad, añade la ventaja de poder añadir compresión con el consiguiente ahora de ancho de banda. El CESGA recomienda a sus usuarios proceder de esta última forma.

  Ejemplo del comando "`netsat -na`" muestra los puertos a la escucha y las sesiones establecidas. Aquí vemos como sesiones establecidas de `ssh` usando direcciones "`v4-mapped`" tunelizan los *displays* gráficos.

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:6010          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:6011          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:47956         127.0.0.1:6010          ESTABLISHED
tcp        0      0 127.0.0.1:6010          127.0.0.1:47956         ESTABLISHED
tcp        0      0 127.0.0.1:48993         127.0.0.1:6010          ESTABLISHED
tcp        0      0 127.0.0.1:48616         127.0.0.1:6010          ESTABLISHED
tcp        0      0 127.0.0.1:60085         127.0.0.1:6010          ESTABLISHED
tcp        0      0 127.0.0.1:6010          127.0.0.1:32838         ESTABLISHED
tcp        0      0 127.0.0.1:6010          127.0.0.1:60085         ESTABLISHED
tcp        0      0 127.0.0.1:6010          127.0.0.1:32824         ESTABLISHED
tcp        0      0 127.0.0.1:6010          127.0.0.1:48993         ESTABLISHED
tcp        0      0 127.0.0.1:32838         127.0.0.1:6010          ESTABLISHED
tcp        0      0 127.0.0.1:32824         127.0.0.1:6010          ESTABLISHED
tcp        0      0 127.0.0.1:6010          127.0.0.1:48616         ESTABLISHED
tcp        0      0 :::22                   :::*                    LISTEN
tcp        0      0 :::80                   :::*                    LISTEN
tcp        0      0 ::1:6010                :::*                    LISTEN
tcp        0      0 ::1:6011                :::*                    LISTEN
tcp        0     36 ::ffff:193.144.34.13:22 ::ffff:193.144.44.:1023 ESTABLISHED
tcp        0      0 ::ffff:193.144.34.13:22 ::ffff:193.144.44.:2432 ESTABLISHED
tcp        0      0 ::ffff:193.144.34.13:22 ::ffff:193.144.44.1:965 ESTABLISHED
```

- **SMTP (*Simple Mail Transfer Protocol*)**

  Dentro de los posibles servidores de correo se han probado dos diferentes:

  - `Sendmail`: Este es uno de los más conocidos servidores de correo. Proporciona soporte nativo IPv6. Esta disponible en http://www.sendmail.org/

  - `Postfix`: El soporte de IPv6 se consigue parcheando los fuentes del software. http://www.postfix.org/ . El parche esta disponible en http://www.ipnet6.org/postfix/ . Como añadido el parche también proporciona soporte TLS (*Tranport Layer Secure*)

  Son necesarios algunos cambios en la configuración de ambos servidores para poder servir peticiones que usen IPv6 como transporte, ya que es

posible deshabilitar o habilitar de forma selectiva el tipo de transporte usado IPv4/IPv6 en las conexiones hechas al servidor.

`Postfix`:

- `Mynetworks`

  Los rangos IPv6 se suministran en formato `[ipv6:addr:range]/plen`

- `smtp_bind_address6`

  Dirección origen usada en las conexiones `STMP` salientes.

- `lmtp_bind_address6`

  Dirección de origen usada en las conexiones `LMTP` como cliente.

`Sendmail`:

```
DAEMON_OPTIONS(`Name=MTA-v4, Family=inet')
DAEMON_OPTIONS(`Name=MTA-v6, Family=inet6')
```

Si usamos el fichero `sendmail.mc`, o bien

```
# SMTP daemon options
O DaemonPortOptions=Name=MTA-v4, Family=inet
O DaemonPortOptions=Name=MTA-v6, Family=inet6
```

- **POP3/IMAP (*Mailbox Daemons*)**

  Dentro del proyecto se han probado software distinto, en este caso dos demonios diferentes:

  - `Courier-imap`: Bien integrado como parte de su hermano mayor la "*suite*" de demonios `Courier` http://courier.sf.net/ o, si se descarga el demonio por separado, provee un soporte nativo de IPv6 con todas las funcionalidades.

  - `Cyrus-imapd`: La rama de desarrollo 2.2 del `cyrus-imapd` proporciona dominios virtuales y soporte nativo IPv6. Sin embargo dicha versión esta considerada de calidad beta. A la versión estable 2.1.15 es necesario aplicarle un parche para que pueda funcionar con IPv6

  - `Dovecot`: De reciente aparición (versión 0.99.x) soporta IPv6 también de forma nativa. Esta considerado un demonio rápido y seguro con gran variedad de mecanismos de autenticación, característica muy deseable dado que se trata de acceder a los buzones de correo de los distintos usuarios. Se puede descargar en http://dovecot.procontrol.fi/ .

  A continuación se muestra la parte representativa del fichero de configuración para el software `dovecot`, "`dovecot.conf`".

```
## Dovecot 1.0 configuration file

# '#' character and everything after it is treated as comments. Extra spaces
# and tabs are ignored. If you want to use either of these explicitly, put the
# value inside quotes, eg.: key = "# char and trailing whitespace  "

# Default values are shown after each value, it's not required to uncomment
# any of the lines. Exception to this are paths, they're just examples
# with real defaults being based on configure options. The paths listed here
# are for configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var
# --with-ssldir=/etc/ssl

# Base directory where to store runtime data.
#base_dir = /var/run/dovecot/

# Protocols we want to be serving:
#  imap imaps pop3 pop3s
protocols = imap imaps pop3 pop3s

# IP or host address where to listen in for connections. It's not currently
# possible to specify multiple addresses. "*" listens in all IPv4 interfaces.
# "[::]" listens in all IPv6 interfaces, but may also listen in all IPv4
# interfaces depending on the operating system. You can specify ports with
# "host:port", although with multiple protocols you probably want to move this
# setting inside protocol imap/pop3 { ... } section, so you can specify
# different ports for IMAP/POP3.
#listen = *
listen = [::]

# IP or host address where to listen in for SSL connections. Defaults
# to above if not specified.
#ssl_listen =
```

..... (el resto del fichero es hace relación a los certificados y a los modos de autenticación a usar).

- **LDAP (*Lightweight Directory Access Protocol)***

  `OpenLDAP` es una implementación *open source* del `LDAP` (*Lightweight Directory Access Protocol*).

  Posee soporte nativo IPv6 y por lo tanto no requiere de aplicar ningún parche a los fuentes del software. Puede descargarse de http://www.openldap.org/.

3.2.2.2. Clientes

En este apartado se describen una serie de aplicaciones de uso común por parte de los usuarios. Así podemos considerarlas como aplicaciones cliente. Esta es una relación del software empleado

- Clientes de terminal: `Putty` (cliente `telnet/ssh` para Windows) – `telnet/ssh` (clientes para `Linux`)

- Clientes FTP: `Filezilla`, `Iexplorer`, `Firefox`, `Konqueror`, `ncftp`, `ftp`

- Navegadores: `Iexplorer`, `Firefox`, `Konqueror`, `Mozilla`
- Clientes de correo: `Outlook`, `Thunderbird`, `Kmail`, `Evolution`, `Messenger` (`Mozilla`)
- *Streaming* de video: `VLC` (http://www.videolan.org), `Windows Media Server 9`
- Videoconferencia: `VIC/RAT`, `gnomeeting`
- `SNMP`: Para la gestión de los equipos de encaminamiento (*routers*). `Net-snmp` (v5, aunque el soporte es sólo parcial)

3.2.2.3. Adaptación de aplicaciones (*porting*)

Destacar que la adaptación de software, tanto de servidores como de clientes, es un proceso relativamente sencillo y mecánico. Con esto lo que se pretende resaltar es que es fácil hacer que una aplicación IPv4 también funcione en IPv6. Sin embargo, si quisiéramos hacer uso de las funcionalidades avanzadas de IPv6 en la aplicación sería necesario reescribirla teniendo IPv6 en mente. (QoS en etiquetas de flujo, *anycast*, movilidad, *multihoming*)

En http://www.deepspace6.net/docs/ipv6_status_page_apps.html se recoge una extensa lista de aplicaciones y su grado de soporte de IPv6.

3.2.3. Problemas detectados y soluciones

Aunque en la mayoría de los casos analizados, las aplicaciones fundamentales tenían un soporte básico para IPv6, se han detectado diversos problemas en dichas aplicaciones que se detallan a continuación:

- **`Sendmail` (versión 12.x):**

  Se puede usar `FEATURE(`msp', `[127.0.0.1]')` en `submit.mc` para evitar problemas con la resolución de nombre de `localhost`, el cual puede resolverse como 127.0.0.1 (o ::1 en IPv6). Si solamente queremos usar IPv6 entonces deberíamos usar `FEATURE (`msp', `[IPv6:::1]')`.

  Otro problema cuya causa no es en sí mismo el `sendmail`, es cuando hay que tratar con servidores de nombre corruptos que ignoran los mensajes `SERVFAIL` de error devueltos por el `DNS` en las peticiones de tipo `T_AAAA` (IPv6). La solución a esto es fijar

  ```
  ResolverOptions=WorkAroundBrokenAAAA
  ```

- **`DNS` (bind 9.2):**

Una de las peculiaridades detectadas al usar direccionamiento IPv6 es que el DNS devuelve en primera instancia direcciones IPv6 en el caso de que ambos mapeados estén disponibles. Así, para un nombre de máquina ejemplo.cesga.es, que tenga dirección IPv4 y dirección IPv6, el DNS devolverá la correspondiente dirección IPv6.

- ***VideoLan Client*** **(VLC):**

  Este software hasta su versión 0.7.2, escuchaba por defecto los anuncios SAP de emisiones *multicast* en la dirección ff08::2:7ffe (esto es con un ámbito o "*scope*" de 8), con lo que en nuestras pruebas iniciales los anuncios de emisiones *multicast* en IPv6 no se recibían. Una vez detectado, existe una opción para poder fijar el ámbito, con lo que los anuncios de las sesiones *multicast* no tienen problemas

  `vlc –extraintf sap –sap-ipv6 –sap-ipv6-scope "e"` ➔ (escuchamos en ff0e::2:7ffe)

3.2.4. Necesidades no resueltas.

El CESGA dispone de equipamiento mayormente multimedia cuya adaptación a IPv6 no es directa. Así nos encontramos con que no es posible actualizar el software (a día de hoy) para que soporten IPv6 en los siguientes equipos:

- MCU: Radvision MCU-323
- *Gateway*: Radvision GW-323
- Polycom ViaVideo (varios modelos , SP128, MP, FX)
- Servidor de *Streaming* Minerva
- Equipamiento LAN (no es posible configurar direcciones IPv6 en los propios equipos para su gestión).

Dicho equipamiento tendrá que seguir usando el protocolo Ipv4, lo qu ha de tenerse en cuenta en el momento de diseñar la migración de la red.

4. *Test-bed*

4.1. Descripción

El planteamiento del proyecto estableció como objetivo el diseño e implementación de un piloto que sirviese para la evaluación de la implantación del protocolo IPv6 en diversas plataformas y equipamientos seleccionados junto con otros existentes en el mercado. Este piloto debía proporcionar, además, la funcionalidad necesaria para el testeo de sistemas inalámbricos con mecanismos de calidad de servicio y para la realización de pruebas de integración con redes europeas y de *routing* IPv6 a nivel europeo.

Con tal finalidad se montó una red piloto IPv6 que conectaba el CESGA con el Departamento de Ingeniería Telemática de la Universidad de Vigo a través de la infraestructura actual. Este piloto permitió, según lo previsto:

- Analizar la problemática de implantación de IPv6 y de las funciones descritas sobre éste.
- Estudiar la interrelación existente entre IPv6 y las tecnologías *wireless*.
- Estudiar el comportamiento de determinadas aplicaciones sobre estas tecnologías (teleconferencia para usuarios móviles, transmisiones *multicast*, trabajos en sistemas remotos a través de clientes inalámbricos con garantía de calidad de servicio, etc.

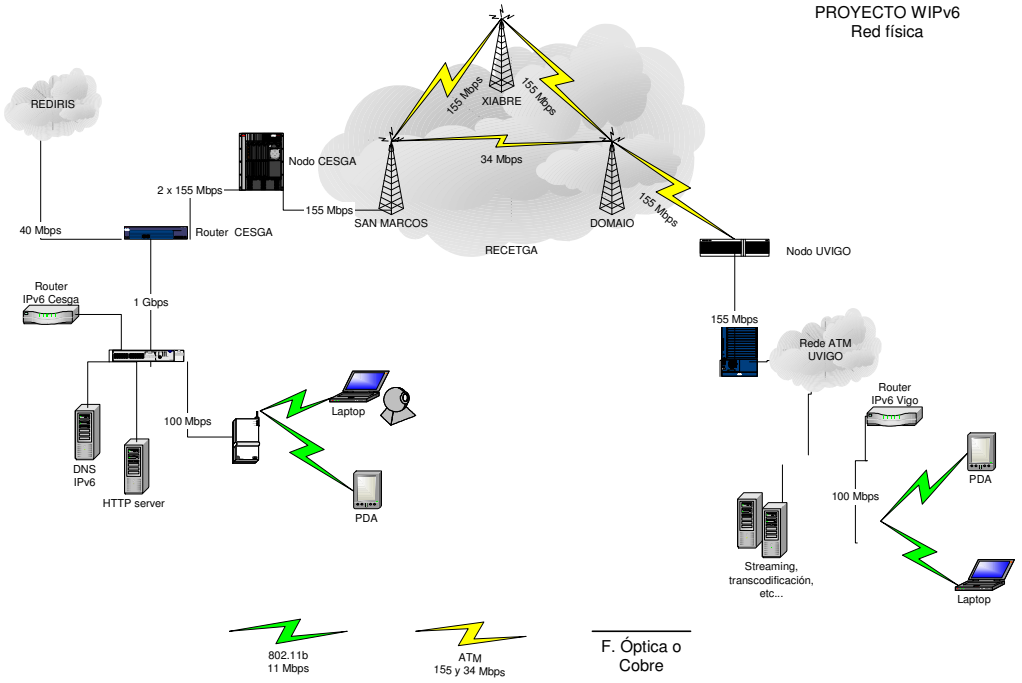
Se analizaron y probaron las características técnicas del sistema *wireless*, prestando especial atención a las referentes a la variación de ancho de banda disponible, tanto en transmisiones *unicast* como *multicast*.

Para la creación del prototipo se seleccionó una plataforma que permitiese servir video con diferentes calidades, a fin de controlar la QoS a nivel de aplicación. Para ello, se diseñó un sistema de transcodificación. Este sistema no fue utilizado hasta hoy en día para generar contenidos para redes móviles. Como fuente de información se empleó una transmisión DVB vía satélite a 15Mbps, correspondiente a un canal de televisión en abierto. Una vez transcodificada la información, el flujo de salida se introdujo en una red inalámbrica como tráfico multicast IEEE 802.11b y se visualiza en PDAs tipo iPAQ. La utilización de IEEE 802.11b se debió a que es el primer estándar legalizado en la UE y porque la comercialización del IEEE 802.11a sufrirá retraso por la normativa legal europea. La elección de satélite como fuente de video se debió, fundamentalmente, a que proporciona una fuente de video digital de alta calidad de la que se pueden extraer distintas tasas de IEEE 802.11 en función de la calidad de servicio: 1, 2, 5.5 y 11 Mbps. El procedimiento de transcodificación se puede realizar en tiempo real en un PIII debido al tamaño de la pantalla de la PDA y a que solo se utiliza una canal de TV.

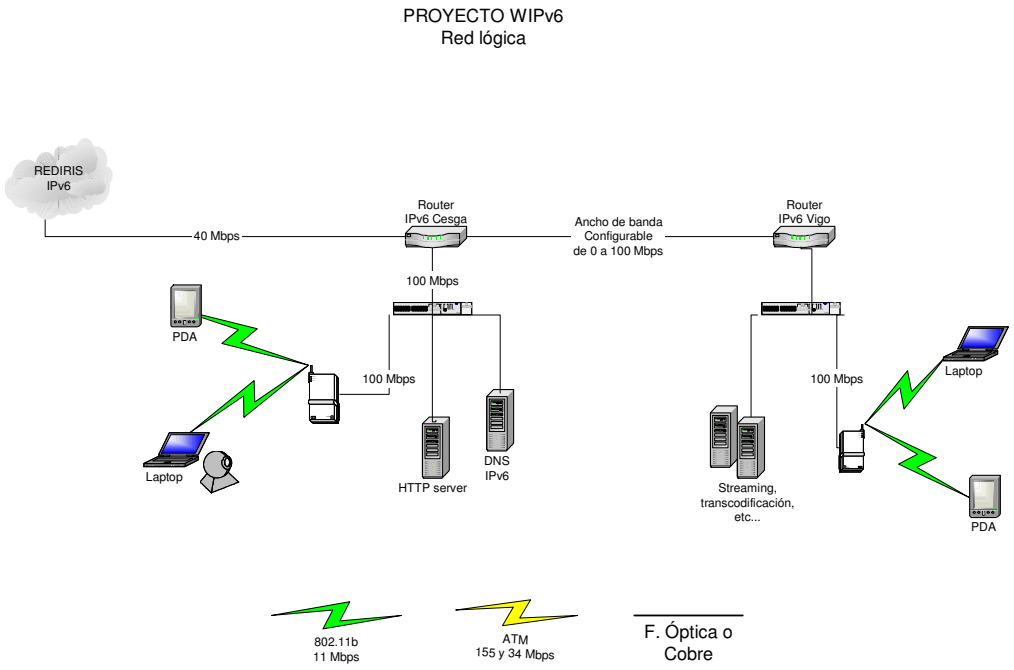
4.2. Implementación del prototipo

En el esquema siguiente se observa la infraestructura de RECETGA (Red de Ciencia e Tecnología) que se utilizó para el establecimiento de la conexión de la red piloto de IPv6 y *wireless* sobre la que se realizaron las diversas pruebas. Cabe resaltar la

gran disponibilidad de la troncal, con doble camino entre Santiago y Vigo, lo que hace posible disponer de ancho de banda reservado para el establecimiento de la red piloto.



En el esquema siguiente se aprecia la red establecida entre los participantes en el proyecto. Toda la infraestructura soporta IPv6 de forma nativa, como el caso de los *routers*, ordenadores y terminales empleados. Los equipos de nivel 2 utilizados, básicamente los conmutadores FastEthernet y los equipos *wireless,* no necesitan soporte IPv6 dado que no intervienen a nivel 3, sólo lo dejan pasar de forma transparente.



Resultó práctica la posibilidad que ofrece la infraestructura ATM de RECETGA de variar el ancho de banda y las características de la interconexión entre los dos *routers* IPv6. De esta manera fue posible simular diversos escenarios: Poco ancho de banda, mucho, mucho ancho de banda pero saturado con otro tráfico, etc. Lo que permite una evaluación más sencilla de los mecanismos de Calidad de Servicio (QoS).

4.3. Resultados obtenidos

La elaboración del *testbed* y sus pruebas posteriores permitieron evaluar el protocolo IPv6 en una WAN de interconexión y en las redes de área local situadas en ambos extremos. Los técnicos de red y usuarios involucrados en las pruebas aumentaron sus conocimientos sobre este protocolo desde sus respectivos puntos de vista.

Se descubrieron problemas en la implementación de `Videolan`, el programa de *streaming* de video sobre IPv6 utilizado. El programa por defecto escuchaba los anuncios SDP en la dirección ff08::2:7ffe cuando el rango utilizado en MBONE es el ff0e::2:2ffe. Esto provocó en un principio que se creyera que se trataba de un problema con el IPv6, pero tras realizar diversas pruebas se comprobó que se trataba del problema antes descrito. Tras reportar este problema a la organización que desarrolla el `Videolan`, esta introdujo cambios en el programa para solventarlo.

Se realizaron *test* de la implementación para IPv6, concretamente para SSM, de herramientas como el `vic` y el `rat`, comprobándose que dichas implementaciones no eran correctas.

Se han verificado y, cuando era necesario y ha sido posible actualizado, las versiones del software de los *routers* principales presentes en la red. El *router* Juniper que nos interconecta con RedIRIS es perfectamente operativo a nivel IPv6 en estos momentos. Así mismo los nuevos *routers* instalados en Vigo, Pontevedra y Ourense ya

han sido instalados con todas las opciones necesarias para funcionar como *dualstack* y con un soporte pleno de IPv6.

Destacamos además los siguientes hechos:

- Dentro de las pruebas de implantación de IPv6 en RECETGA (Rede Ciencia e Tecnoloxía de Galicia) se realizaron transmisiones de datos utilizando esta tecnología, siendo las primeras de este nivel realizadas en el marco de GEANT (red europea de investigación), las cuales fueron recogidas en diversos medios de comunicación.

- Se produjo una intensificación y/o consolidación de vínculos de trabajo con otros grupos o instituciones con intereses comunes. En las pruebas realizadas colaboraron con la red Gallega de Ciencia y Tecnología (RECETGA), la red española (REDIRIS), la red francesa (RENATER) y la portuguesa (FCCN). Esta colaboración contribuye al intercambio de experiencias y conocimientos pudiendo derivar en la creación de nuevas líneas de trabajo comunes o participación conjunta en otras ya existentes.

- Colaboración con RedIRIS en las pruebas e implantación de un sistema jerárquico de RPS para permitir la interconexión de la red española de investigación con M6BONE. Participación en la resolución de problemas planteados.

4.4. Problemas encontrados y soluciones aportadas

Los problemas que se han detectado eran los previstos en la dirección del proyecto. Éstos están relacionados con las implementaciones de IPv6 en las plataformas tanto de *routers* como ordenadores disponibles en el mercado.

Dado que detectar estos problemas de implantación era uno de los objetivos del proyecto no se consideró que el proyecto se hubiese visto perjudicado.

5. Estrategias de migración de IPv4 a IPv6

5.1. Necesidad de transición

Tarde o temprano debe afrontarse el despliegue del nuevo protocolo de Internet. Dos factores nos llevan a esto: el encaminamiento y el direccionamiento. Como ya se dijo anteriormente, el encaminamiento global basado en direcciones de 32 bits se agota. Las direcciones IPv4 no proporcionan suficiente flexibilidad para construir jerarquías eficientes que puedan ser agregadas. Incluso aunque IPv4 pueda escalar para soportar un Internet completamente IPv4, tarde o temprano se agotarán las direcciones de red.

El desafío radica en realizar la transición a la nueva generación del protocolo IP antes de que se produzca el agotamiento de IPv4. Esta transición será mucho más sencilla si las direcciones IPv4 son todavía únicas. Los dos requisitos de transición más relevantes son:

- Flexibilidad de despliegue
- Que los *hosts* que sólo entienden IPv6 se puedan comunicar con *hosts* que sólo entienden IPv4.

La estrategia de despliegue de IPv6 debe ser tan flexible como sea posible. Internet es excesivamente grande para poder realizar un despliegue muy coordinado.

5.2. Problemática de la transición

Las características principales de la migración de IPv4 a IPv6 se basan en los siguientes puntos:

- IPv4 e IPv6 son incompatibles a nivel de paquete, es decir:
  - Los nodos finales actuales de Internet no generan ni reconocen IPv6
  - Los *routers* IP actuales de Internet descartan los paquetes IPv6.
- La principal dificultad es migrar la red Internet:
  - Durante la etapa de transición convivirán Internet IPv4 e IPv6 a nivel lógico.

5.2.1. Procedimiento de transición

Los objetivos fundamentales de la etapa de transición son los siguientes:

- Permitir a *hosts* IPv4 e IPv6 interoperar.
- Permitir a *hosts* y *routers* IPv6 desplegarse en Internet de forma incremental.
- La transición debe de ser tan sencilla como sea posible para usuarios finales, administradores de sistemas y operadores de red. Tanto como para comprenderla como para llevarla a cabo.

Los mecanismos de transición son un conjunto de mecanismos de protocolo implementados en *hosts* y *routers* junto con algunas indicaciones operacionales para direccionamiento y despliegue, diseñados de forma que se realice la transición con la menor disrupción posible.

Estos mecanismos de transición incluyen:

- Actualización y despliegue incremental. *Hosts* y *routers* IPv4 individuales pueden actualizarse sin necesidad de hacerlo simultáneamente.
- Dependencias de actualización mínimas. El único prerrequisito para actualizar *hosts* a IPv6 es que el servidor de DNS se actualice en primer lugar para poder manejar entradas con direcciones IPv6. No hay prerrequisitos para actualizar *routers*.
- Fácil direccionamiento. Cuando *hosts* o *routers* IPv4 existentes se actualicen a IPv6 pueden continuar utilizando su dirección existente. No necesitan que se les asignen nuevas direcciones. Los administradores no tienen que idear nuevos planes de direccionamiento.
- Bajo coste inicial. Es necesaria poca preparación para actualizar sistemas IPv4 a IPv6, o para desplegar nuevos sistemas IPv6. Los mecanismos empleados para la transición incluyen:

  - Una estructura de direccionamiento IPv6 que incluye direcciones IPv4 embebidas en direcciones IPv6, y codifica otra información utilizada por los mecanismos de transición.
  - Un modelo de despliegue en el cual los *hosts* y *routers* actualizados a IPv6 en una etapa inicial tienen capacidad "dual" (implementan pilas IPv4 e IPv6 completas).
  - La técnica de encapsulamiento de paquetes IPv6 con cabeceras IPv4 para transportarlos sobre segmentos del camino en los cuales los *routers* todavía no han sido actualizados con IPv6.
  - Técnicas de traducción de cabecera para permitir la introducción eventual de topologías de encaminamiento que encaminen sólo el tráfico IPv6, y el despliegue que *hosts* que soporten IPv6. La utilización de esta técnica es opcional y podría ser utilizada en las últimas fases o no ser utilizada en absoluto.

Los mecanismos de transición de IPv6 aseguran que los *hosts* de ambos protocolos pueden interoperar en cualquier punto de Internet hasta que se agoten las

direcciones IPv4, y permite a los hosts IPv6 e IPv4 de un ámbito limitado interoperar indefinidamente después de esto. Esta característica garantiza la enorme inversión realizada en IPv4 y asegura que IPv6 no convierte a IPv4 en un protocolo obsoleto.

5.3. Descripción de los mecanismos de transición:

Las posibles transiciones pueden clasificarse de la siguiente forma:

- Doble capa IP (*dual stack*)
- Mecanismos de tipo túnel: se basan en encapsular un protocolo sobre otro. Están enfocados a unir dos islas IPvX a través de un océano IPvY.
  - Túneles manuales
  - Túneles automáticos
  - Túneles 6to4
  - Túneles 6over4
- Mecanismos de traducción: se basan en traducir, en un elemento de red, los paquetes de un formato a otro.
  - NAT-PT
  - SOCKSv5
  - BIS (*Bump in the stack*)

En los siguientes apartados comentaremos dichos procedimientos.

5.3.1. Doble pila IP (*dual stack*)

Mediante esta técnica se proporciona soporte IP completo para ambos protocolos de Internet tanto en *hosts* como en *routers*.

Es la forma más sencilla de que los nodos IPv6 sean compatibles con los nodos que disponen únicamente de IPv4. Los nodos que poseen pila dual tienen la capacidad de enviar y recibir paquetes IPv6 e IPv4. Pueden interoperar directamente con nodos IPv4 utilizando paquetes IPv4 y con nodos IPv6 utilizando paquetes IPv6.

La técnica de pila dual puede utilizarse en conjunto con las técnicas de *tunneling* IPv6 sobre IPv4.

- **Configuración de direcciones**

  Dado que se soportan ambos protocolos, debe asignarse a los nodos con pila dual ambos tipos de direcciones. Pueden utilizar para ello las técnicas

típicas de uno u otro protocolo (p. ej. DHCP para IPv4 o autoconfiguración de direcciones sin estado para IPv6).

- **DNS**

  Los nodos con capacidad dual deben de proporcionar librerías de resolución con capacidad para tratar con registros "A" IPv4 así como con registros "AAAA" y "A6".

5.3.2. Mecanismos de *tunneling*

Proporcionan una forma de utilizar la estructura de encaminamiento IPv4 para transportar tráfico IPv6.

Los *hosts* y *routers* pueden encapsular datagramas IPv6 en regiones con topología de encaminamiento IPv4. El *tunneling* puede utilizarse en diferentes escenarios:

- *Router* a *router*
- *Host* a *router*
- *Host* a *router*
- *Router* a *host*

En cualquiera de los casos se crea un túnel que comprende parte o la totalidad de la extensión del camino que toman los paquetes IPv6.

Las técnicas de encapsulamiento se clasifican generalmente en función del mecanismo de determinación de la dirección del nodo del final del túnel.

En el caso del escenario *router* a *router* y *host* a *router*, el paquete IPv6 se envía mediante un túnel a un *router*. En este caso el fin del túnel es diferente del destinatario del paquete IPv6 que va por él, de forma que las direcciones del paquete transmitido mediante el túnel no proporcionan la dirección de destino del paquete, ésta debe ser determinada mediante información de configuración proporcionada en el nodo que crea el túnel. Se utiliza el término "*tunneling* configurado o manual" para describir el tipo de *tunneling* en el cual el punto final se configura de forma explícita.

En los dos últimos casos de *tunneling*, *host* a *host* y *router* a *host*, el paquete IPv6 se transmite mediante un túnel toda la extensión hasta su destino final. En este caso, la dirección de destino tanto del paquete IPv6 como de la cabecera de encapsulación IPv4 identifica al mismo nodo. Este hecho permite que se pueda obtener la dirección IPv4 de destino de forma automática. En este hecho se basa la técnica de *tunneling* automático, la cual utiliza un formato de direcciones IPv6 con direcciones IPv4 embebidas que permiten a los nodos que realizan el *tunneling* obtener automáticamente la dirección IPv4 de destino, eliminando la necesidad de configurar de forma explícita la dirección del punto final del túnel, simplificando así la configuración.

NOTA:

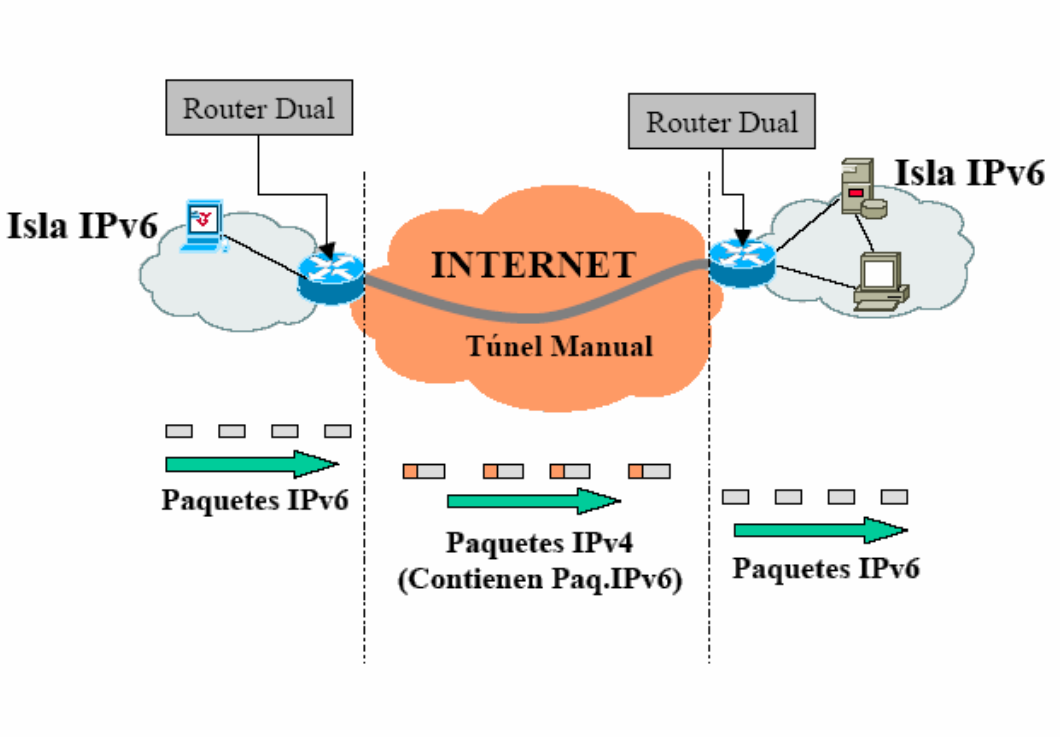
Cuando un *host* o *router* recibe un datagrama IPv4 con destino una de sus direcciones IP, y su valor del campo de protocolo es 41 (correspondiente a tipo de carga IPv6), reensambla el paquete (en caso de que haya sido fragmentado), elimina la cabecera IPv4 y envía el datagrama al código de nivel IPv6.

- **Configuración de encapsulamiento manual:**

  Consiste en un túnel IPv6 sobre IPv4 en el cual la dirección del punto final IPv4 está determinada por información de configuración del nodo que realiza el encapsulamiento.

  Características:

  - Su funcionalidad principal radica en interconectar islas IPv6 a través de un océano IPv4.
  - Cada extremo es un nodo dual y en ellos se configuran las direcciones IPv4 e IPv6 tanto local como remotas.



En este caso la dirección del extremo del túnel viene determinada por la información de configuración del nodo que realiza la encapsulación, éste debe de almacenar la dirección extremo para cada túnel. Esto se realiza generalmente mediante la tabla de encaminamiento, la cual direcciona los paquetes basándose en su dirección de destino y utilizando la máscara de prefijo.

Hosts IPv6/IPv4 conectados a enlaces que no disponen de un *router* IPv6 pueden utilizar un túnel manual para alcanzar a un *router* IPv6. Este túnel permite a los *hosts* comunicarse con el resto de Internet IPv6. (nodos con direcciones nativas IPv6). Si se conoce la dirección IPv4 de un router de borde IPv6/IPv4 del *backbone* IPv6, puede utilizarse como dirección extremo del túnel.

Como ventajas de este método podemos citar las siguientes:

- Se utiliza con frecuencia en el acceso al `6-bone` y está disponible en multitud de plataformas (Cisco, Linux, Solaris, Windows, etc.).
- Es un método transparente al nivel IPv6 y superiores, con lo cual no afecta a las aplicaciones.
- No consume excesivos recursos (la MTU se reducen en 20 bytes de la cabecera IPv4 típica).
- Su aplicación principal es la conexión con ISP IPv6 remotos a través de Internet.

Como desventajas podríamos citar las siguientes:

- No son dinámicos, deben establecerse manualmente o de forma semiautomática.
- Si se unen N islas y no se considera un nodo central que realice el intercambio, el número de túneles a establecer en cada sitio asciende a N-1, no siendo escalable.

Existe una herramienta de gestión del establecimiento de túneles manuales: Túnel Broker.

- **Configuración de encapsulamiento automático**

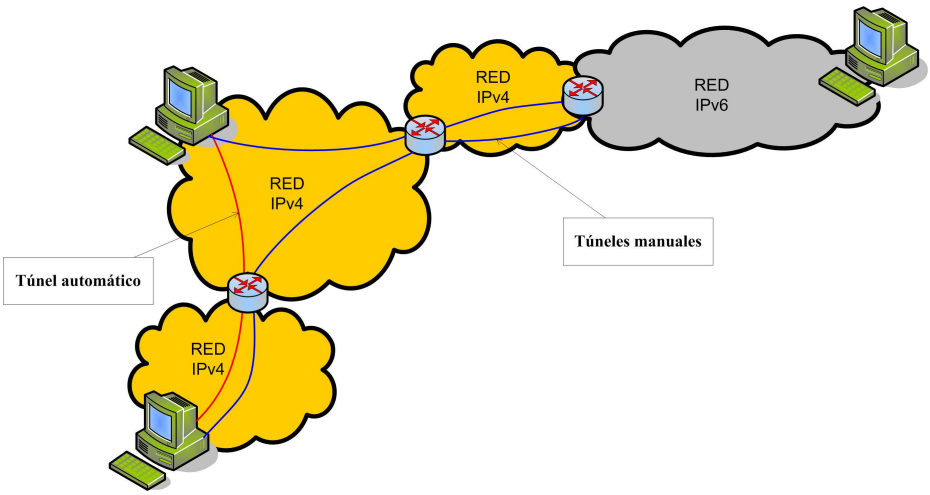
Consiste en un túnel IPv6 sobre IPv4 en el cual la dirección del punto final IPv4 está determinada por la dirección IPv4 embebida en la dirección de destino compatible IPv4 del paquete IPv6 enviado al túnel.

Como características principales de este método de encapsulación detallamos las siguientes:

- Permite a nodos duales comunicarse a través de una infraestructura IPv4
- Hace uso de direcciones IPv6 compatibles con IPv4.
- Los paquetes destinados a direcciones compatibles IPv4 se envían por el túnel automático.
- Se define una interfaz virtual para la dirección compatible IPv4.
- Los paquetes destinados a direcciones compatibles IPv4 se envían por el túnel automático con las siguientes reglas:
  - La dirección origen IPv6 es una dirección compatible IPv4 local.

- La dirección de destino IPv4 se obtiene de la dirección compatible IPv4.

Pueden utilizarse tanto túneles automáticos como manuales en *hosts* aislados (sin *routers* IPv6 en el enlace).



En *tunneling* automático, la dirección del extremo del túnel se determina del paquete a ser enviado. Si la dirección IPv6 es compatible IPv4, el paquete puede enviarse mediante *tunneling* automático, si se trata de una dirección nativa IPv6, el paquete no puede enviarse mediante *tunneling* IPv6.

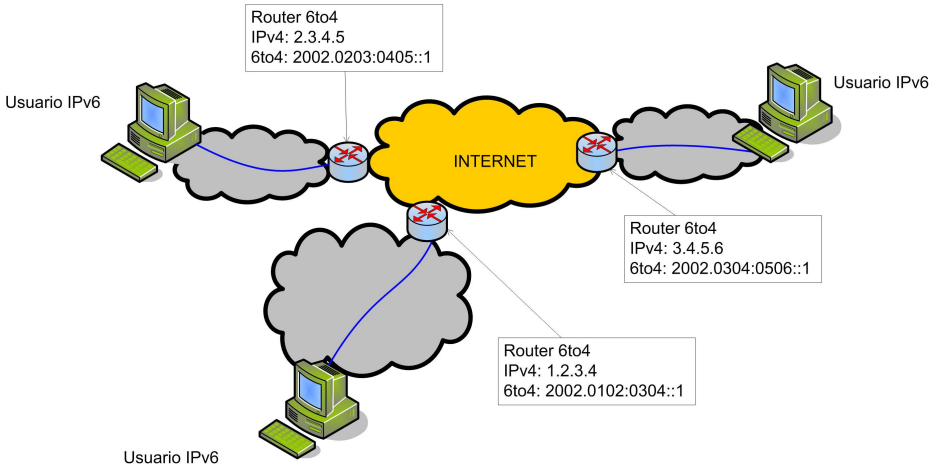
En los túneles automáticos la dirección del extremo del túnel viene determinada por la dirección de destino compatible IPv4 del paquete a IPv6 a enviar por el túnel. Esta forma de encapsulamiento permite a los nodos IPv6/IPv4 comunicarse sobre infraestructuras de encaminamiento sin túneles preconfigurados.

- **Configuración de encapsulamiento 6to4:**

Su aplicación principal es unir islas IPv6 dispersas en un océano IPv4.

A cada isla se le asigna un prefijo IPv6: 2002::/16+Dir IP del *router* frontera. El siguiente salto IPv4 está contenido en la dirección IPv6. El encaminamiento entre las distintas islas se apoya en el encaminamiento IPv4 subyacente.

Existen implementaciones de este mecanismo en Windows NT y Proyecto KAME: Linux y FreeBSD



Como ventajas citaremos las siguientes:

- Al igual que los túneles manuales, son transparentes a nivel IPv6, no afectando a las aplicaciones.
- Se trata de túneles establecidos dinámicamente y sin configuración previa.
- Dadas N islas IPv6, sólo se establecen los túneles necesarios para las conexiones activas en cada momento.

El inconveniente principal radica en que las organizaciones que se conecten a un ISP IPv6 remoto no necesitan más de un túnel (quizás dos por redundancia con otro ISP IPv6), por lo que se puede emplear un mecanismo de túneles manuales, que se haya más extendido.

- **Configuración de encapsulamiento 6over4:**

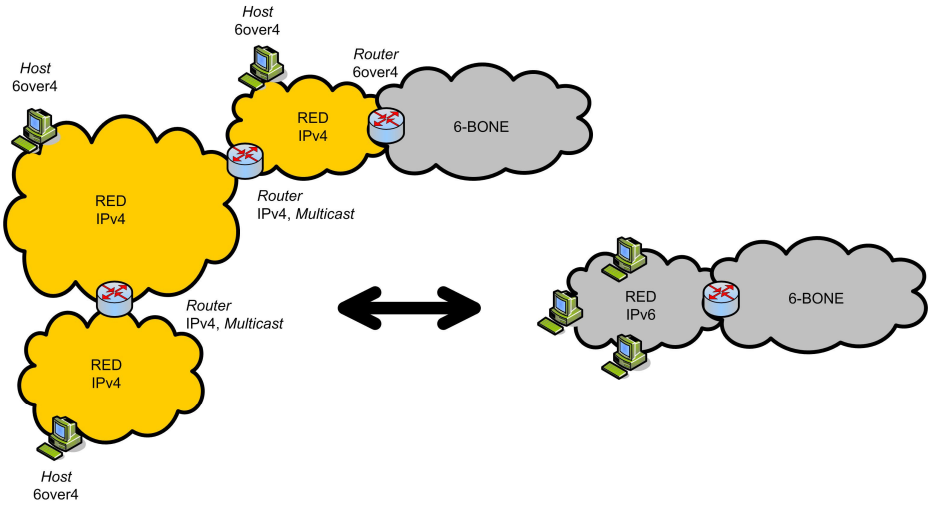
  Consiste en encapsular paquetes IPv6 en cabeceras IPv4 de forma que se puedan encaminar a través de estructuras IPv4.

  Se trata de un mecanismo utilizado por las direcciones compatibles IPv4 para encapsular automáticamente paquetes IPv6 sobre redes IPv4.

  Se trata de túneles punto a punto hechos encapsulando paquetes IPv6 con cabeceras IPv4, que les permitan ser encaminados sobre infraestructuras IPv4.

  Sus características principales son las siguientes:

  - Conectan nodos IPv6 dispersos en subredes IPv4, se forma una "LAN virtual" IPv6.
  - El tráfico IPv6 entre nodos es encapsulado en IPv4.
  - Los procesos de descubrimiento de vecino y *router* se realizan empleando *multicast*.
  - Si se dispone de un *router* 6over4 con acceso al `6-bone` tendremos que todos los nodos pueden acceder al `6-bone`.



Como ventajas destacamos las siguientes:

- Son transparentes al nivel IPv6, no afectando a las aplicaciones.
- Son túneles establecidos dinámicamente y sin configuración previa.
- Permiten probar IPv6 en nodos de una red corporativa IPv4 sin instalar IPv6 en los *routers* internos.
- Instalando en un único *router* la pila IPv6 y conectándolo al `6-bone` se proporciona acceso a la red al resto de nodos IPv6.

Como inconvenientes cabe destacar:

- Es un mecanismo adecuado para redes finales únicamente
- No está ampliamente implementado.

- ***Tunneling multicast* IPv4**

Tunneling IPv6 sobre IPv4 en el cual la dirección del punto final IPv4 está determinada mediante el procedimiento Descubrimiento de Vecino (*Neighbor Discovery*). A diferencia del *tunneling* configurado no necesita configuración de direcciones y a diferencia del *tunneling* automático no requiere del uso de direcciones compatibles con IPv4. Sin embargo, el mecanismo asume que la infraestructura IPv4 soporta *multicast* IPv4.

5.3.3. Mecanismos de traducción

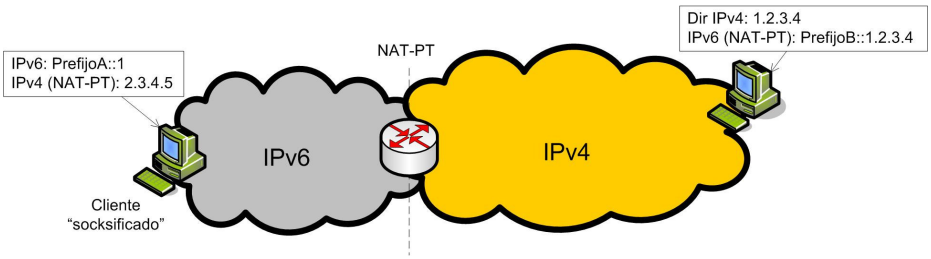
- **NAT-PT**

Características principales:

En los mecanismos de NAT tradicionales se realiza la traducción de direcciones (conexión de redes con direccionamiento IPv4 privado). En el caso de NAT-TP se realiza además la traducción de protocolo. Esta está basada en el algoritmo SIIT (RFC 2765).

Debe tenerse en cuenta que no es transparente al nivel de aplicación, precisando de algunas extensiones.

- DNS-ALG: Transforma peticiones DNS "A" a peticiones "AAAA"
- FTP-ALG: Las conexiones con FTP son problemáticas pues abren dos conexiones TCP intercambiando direcciones IP al nivel de aplicación.



Como ventajas cabe citar las siguientes:

- Muchas redes corporativas poseen experiencia en la gestión/administración de NATs
- Implementado en la mayor parte de los routers y en algunas plataformas habituales en nodos finales (Windows 2000).
- Si la comunicación extremo a extremo es heterogénea (IPvX-IPvY) NAT-PT resulta adecuado (teniendo en cuenta siempre la carga de trabajo prevista).

Como inconvenientes resaltamos:

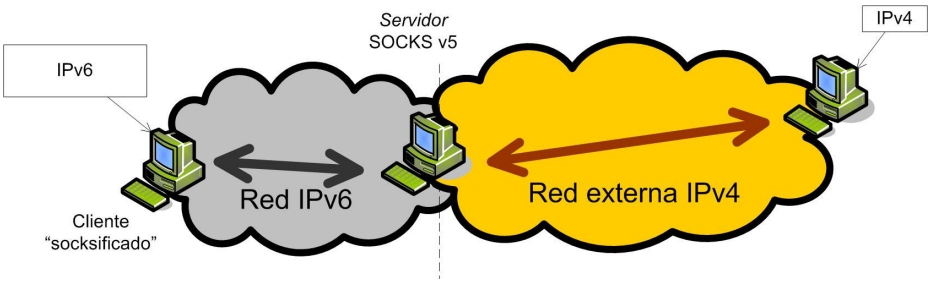
- Los NATs poseen un alto coste de gestión/administración
- El proceso de traducción es más costoso en recursos que el de encapsulamiento.
- Si la comunicación extremo a extremo es homogénea (IPvX-IPvX) siempre es preferible emplear túneles a dos sistemas de traducción consecutivos.
- Si en un protocolo de aplicación se intercambian direcciones IP (DNS, FTP, etc.), es necesario una extensión o módulo que incluya un algoritmo para su tratamiento específico (DNS-ALG, FTP-ALG).

- **SOCKSv5**

El uso tradicional de SOCKSv5 es proporcionar conectividad IP directa a Internet en redes con *firewall* en determinados *hosts*. En este caso se hace

uso de un servidor SOCKSv5 dual, que realiza además la función de traductor de protocolos (Algoritmo SIIT):

- Traducción IPv4-IPv6 y viceversa. Conexiones siempre iniciadas por el cliente.
- Dos componentes: Servidor SOCKSv5+Librería SOCKSv5 (cliente)



Funcionamiento detallado: (Red IPv4 = Red interna)

- Una aplicación en el nodo cliente inicia una conexión TCP o UDP con un nodo externo empleando el nombre completo (FQDN).
- La librería SOCKSv5 en el cliente intercepta la resolución del nombre ("`gethostbyname`") e inicia una conexión TCP al puerto 1080 del servidor SOCKSv5.
- El servidor SOCKSv5 devuelve al cliente una dirección IPv4 falsa ("*fake IPv4 address*")
- El servidor SOCKSv5 inicia la conexión TCP o UDP con el nodo remoto y hace de *proxy* entre el cliente y el nodo externo. Si el nodo externo es IPv6, aplica además el algoritmo de traducción SIIT.
- En el cliente, los paquetes con la dirección IPv4 falsa como origen o destino son interceptados y tratados por la librería SOCKSv5 que los recibe o envía respectivamente al servidor SOCKSv5.

Como ventajas cabe destacar las siguientes:

- Se trata de un sistema apto para empresas que deseen dar acceso a determinados nodos internos a servicios IPv6 sin probar exhaustivamente el protocolo.
- Provee sistemas de autenticación adecuados para evitar usos indeseados.

Presenta los siguientes inconvenientes:

- Necesidad de instalación de las librerías SOCKSv5 en todos los clientes a los que se desee dar acceso.
- El proceso de traducción es costoso en cuanto a consumo de recursos en el servidor, por lo que un factor limitante es la carga de tráfico prevista.

- Las conexiones sólo pueden ser iniciadas por los nodos internos, con lo cual no es posible ofrecer servicios al exterior mediante este método.
- Como todos los mecanismos de traducción debe incorporar algoritmos específicos para aquellos protocolos de aplicación que intercambien direcciones IP (FTP).

5.3.4. Estrategias de migración

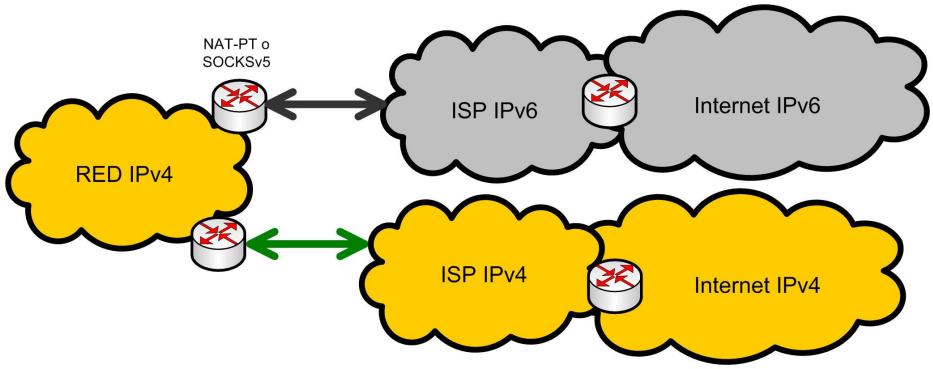
En general, debe estudiarse la migración de las redes finales primero y, según aumente el tráfico IPv6 realizar la migración de ISPs y *backbones* principales.

Para redes finales pueden seguirse las siguientes recomendaciones:

- Servidores: "doble pila", para atender peticiones IPv4 e IPv6
- Clientes: "doble pila", conectividad con servidores IPv4 e IPv6.

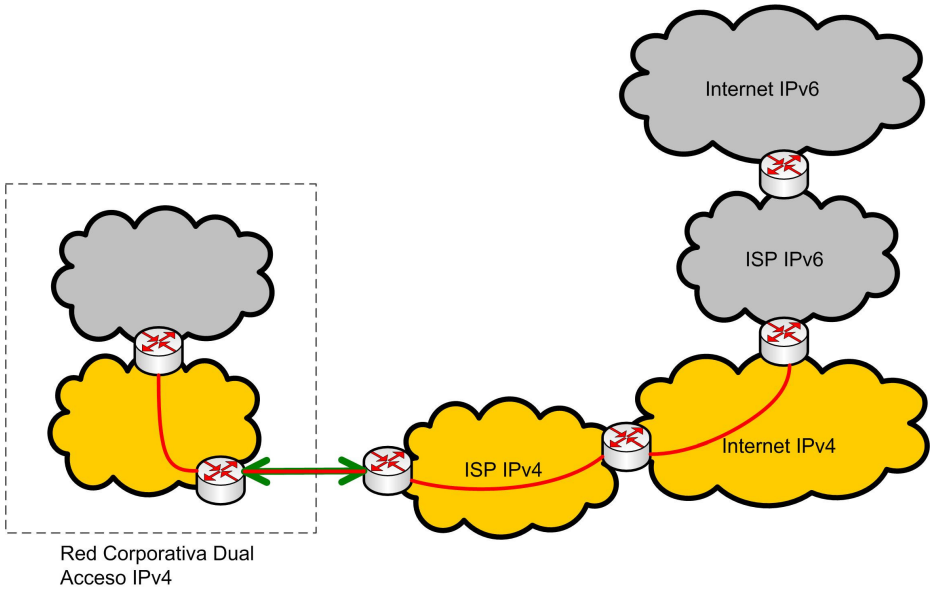
- **Mecanismos de migración de redes finales (clientes y servidores)**

- Migración mediante mecanismos de traducción



- Migración mediante mecanismos de *tunneling*

  Primera fase: Conexión IPv4 al ISP y enviar el tráfico IPv6 mediante un túnel sobre IPv4, hasta que el ISP ofrezca conexión con IPv6 nativo.



Segunda fase: Conexión IPv6 al ISP y túnel IPv4 sobre IPv6 para conectar IPv4 (caso complementario)

- **Estrategias de migración para ISPs**

  - Conexión nativa a *backbones* IPv4 e IPv6, sin emplear túneles.
  - Modos de acceso:
    - ISP IPv4 tradicionales: Acceso IPv4 y tratar de ofrecer acceso a Internet IPv6 mediante un traductor.
    - Nuevos ISP IPv6: Acceso IPv6 y mediante túnel a través de Internet. Ofrecer conectividad a Internet IPv4 mediante traductores.

- **Estrategias de migración de backbones**

  - Mantener configuración actual y migrar cuando el tráfico entunelado sea mayor que el tráfico IPv4.
  - Debido a los problemas del número de rutas existente, recomendar y colaborar con los ISP y otros *backbones* para evitar una migración forzosa.

5.4. Referencias:

- RFC 2893 - "*Transition Mechanisms for IPv6 Hosts and Routers*". R. Gilligan, E. Nordmark, Agosto 2000.

  Especificación de los mecanismos de compatibilidad con IPv4 que pueden ser implementados en *hosts* y *routers*.

- "IPv6: Mecanismos de Transición IPv4-IPv6". Carlos Ralli Ucendo. Telefónica I+D.

6. Estrategia de migración de RECETGA

6.1. Descripción RECETGA

6.1.1. Introducción

Con la instalación en 1993 del Centro de Supercomputación de Galicia (CESGA) en Santiago de Compostela se pone en funcionamiento en Galicia una red de interconexión entre los 7 campus universitarios presentes en la comunidad. El CESGA será la entidad encargada de velar por la gestión de esta red y el responsable último de su expansión en Galicia. Esta red, que desde su nacimiento no ha dejado de crecer tanto en ancho de banda como en número de centros conectados, se denomina Red de Ciencia y Tecnología de Galicia (RECETGA).

RECETGA es hoy una infraestructura de comunicaciones propiamente gallega que no depende de operadoras externas. Hoy, las operadoras, proveedoras de conectividad y servicios de comunicaciones, no disponen de alternativas a RECETGA en cuanto a relación precio/prestaciones.

6.1.2. Objetivos

- Proveer servicios de comunicaciones a la comunidad académica y de investigación en Galicia.
- Proveer un entorno tecnológico que posibilite la Investigación, Desarrollo e Innovación en el campo de las comunicaciones en nuestra comunidad.
- Favorecer el desarrollo de la Sociedad de la Información y el Conocimiento en Galicia.

6.1.3. Usuarios

Actualmente RECETGA da servicio de comunicaciones a los siete Campus Universitarios gallegos, Centros Tecnológicos y de Investigación dependientes de la Xunta de Galicia, Consejo Superior de Investigaciones Científicas (CSIC), Instituto Español de Oceanografía, Laboratorios de Investigación de Complejos Hospitalarios y más de veinte instituciones y empresas que trabajan en I+D+I. El número de usuarios de la red se cifra entorno a los 100.000, incluyendo, docentes, investigadores, estudiantes, etc.

6.1.4. Estructura

RECETGA es una red ATM soportada sobre fibra óptica y radio enlaces, con un ancho de banda en la troncal de hasta 622 Mbps. La topología es mixta, presentando una troncal mallada en su mayor parte y el resto en estrella, con un nodo central situado en Santiago de Compostela.

CENTROS CONECTADOS A RECETGA	
Universidade da Coruña	
Campus da Coruña	155 Mbps
Campus de Ferrol	155 Mbps
Universidade de Santiago de Compostela	
Campus de Santiago	2 x 155 Mbps 1 Gbps
Campus de Lugo	155 Mbps
Universidade de Vigo	
Campus Vigo	155 Mbps
Campus de Pontevedra	155 Mbps
Campus de Ourense	155 Mbps
Centros Tecnolóxicos e de Investigación	
ANFACO - Vigo	10 Mbps
Aula de Productos Lácteos	2 Mbps
CEIDA - Centro de Divulgación Ambiental de Galicia	11 Mbps
Centro de Investigacións Forestais de Lourizán	155 Mbps
Centro Superior Bibliográfico de Galicia	155 Mbps
Centro de Investigacións Lingüísticas "Ramón Piñeiro"	155 Mbps
Centro de Experimentación en Acuicultura	64 Kbps
Estación de Viticultura e Enoloxía de Leiro	64 Kbps
Centro de Cultivos Mariños (Ribadeo)	64 Kbps
Centro de Información e Tecnoloxía Ambiental	2 Mbps
Centro de Investigacións Agrarias de Mabegondo	2 Mbps
Centro de Control de Calidade do Medio Mariño	155 Mbps
Centro de Investigacións Mariñas	2 Mbps
Centro de Supercomputación de Galicia	622 Mbps
Consorcio de Bibliotecas Universitarias de Galicia	155 Mbps
Centro Tecnológico del Mar (CETMAR)	11 Mbps
Centro de Innovacións e Servicios en Ferrol	155 Mbps
Fundación Empresa-Universidade de Galicia	155 Mbps
Novo Seminario de Estudos Galegos	2 Mbps
Dirección Xeral de Investigación e Desenvolvemento	155 Mbps
Hospitais	
Complexo Hospitalario "Cristal Piñor"	64 Mbps
Hospital Clínico Universitario de Santiago de Compostela	155 Mbps
Complexo Hospitalario "Xeral-Cies"	155 Mbps
Complejo Hospitalario Universitario "Juan Canalejo"	155 Mbps
Unidade de Investigación do Hospital do Meixoeiro	155 Mbps
IEO	
Instituto Español de Oceanografía - A Coruña	256 Kbps
Instituto Español de Oceanografía - Vigo	256 Kbps
CSIC	
Misión Biolóxica de Galicia- CSIC	2 Mbps
Instituto de Investigacións Agrobiolóxicas (CSIC)	1Gbps
Instituto de Investigacións Mariñas (CSIC)	155 Mbps
Instituto de Estudios Galegos "Padre Sarmiento"	2 Mbps
Delegación do CSIC en Galicia	64 Kbps
Outros Centros	
Palacio de Exposicións e Congresos de Galicia	155 Mbps
Parque Tecnolóxico de Galicia	155 Mbps
Autopista Galega da Información (AGI)	155 Mbps

Esquema de la Red de Ciencia y Tecnología de Galicia en 2003

6.1.5. Nodos de Acceso

La red cuenta con 44 nodos de acceso de distintas capacidades:

- 3 conexiones a 1 Gbps
- 24 conexiones a 155 Mbps
- 2 conexiones a 11 Mbps
- 1 conexión a 10 Mbps
- 7 conexiones a 2 Mbps
- 2 conexiones a 256 Kbps
- 5 conexiones a 64 Kbps

6.1.6. Relación de centros actualmente conectados a RECETGA

Centro	Dirección	Capacidad
Universidad de la Coruña		
UDC Coruña	Coruña	155 Mbps
UDC Ferrol	Ferrol	155 Mbps
Universidad de Vigo		
UVIGO Ourense	Ourense	155 Mbps
UVIGO Vigo	Vigo	155 Mbps
UVIGO Pontevedra	Pontevedra	155 Mbps
Centros tecnológicos y de investigación		
ANFACO	Vigo	10 Mbps
Aula Productos Lacteos	Lugo	2 Mbps
Centro de Investigaciós Forestais e Ambientais de Lourizan	Pontevedra	155 Mbps
Centro Superior Bibliográfico de Galicia		155 Mbps
Centro de Investigaciones Lingüísticas "Ramón Piñeiro"	Vigo	155 Mbps
Centro de Experimentación en Acuicultura		2 Mbps
Estación de Viticultura y Enología de Leiro		64 kbps
Centro de Cultivos Marinos	Ribadeo	64 kbps
Centro de Investigaciones Agrarias de Mabegondo		2 Mbps
Centro de Control de Calidad del Medio Marino		155 Mbps
Centro de Investigaciones Mariñas		155 Mbps
CESGA		1 Gbps
Consorcio de Bibliotecas Universitarias de Galicia		155 Mbps
CETMAR	Vigo	11 Mbps
CIS FERROL	Ferrol	155 Mbps
Fundación Empresa - Universidad de Galicia	Vigo	155 Mbps
Fundación Empresa - Universidad de Galicia	Santiago	155 Mbps
Seminario de Estudos Galegos	Santiago	2 Mbps
Dirección Xeneral de Investigación y Desenvolvemento		155 Mbps
CTAG	Porriño	2 Mbps
Hospitales		
Complejo hospitalario "Cristal Piñor"		64 Mbps
Hospital Clínico Universitario de Santiago de Compostela		155 Mbps
Hospital Cies	Vigo	155 Mbps
Hospital Juan Canalejo	Coruña	155 Mbps
Hospital Meixoeiro	Vigo	155 Mbps
IEO		
Instituto Español de Oceanografía	A Coruña	256 Kbps
Instituto Español de Oceanografía	Vigo	256 Kbps
CSIC		
Misión Biológica do CSIC		2 Mbps
Instituto de Investigación Agrobiolóxicas (CSIC)		1 Gbps
Instituto de Investigacións Mariñas (CSIC)		155 Mbps
Instituto de Estudos Galegos "Padre Sarmiento"		2 Mbps
Delegación do CSIC en Galicia		64 Kbps
Otros centros		
Palacio de Exposiciones y Congresos de Galicia		155 Mbps
Parque Tecnológico de Galicia		155 Mbps
Autopista Gallega de la Información		155 Mbps

6.1.7. Gestión de Red

El CESGA gestiona el tráfico en todos los tramos de RECETGA haciendo uso de herramientas de gestión de red avanzadas, tanto comerciales como desarrolladas y adaptadas por el CESGA.

6.1.8. Mantenimiento de Equipos

El CESGA mantiene un convenio con la empresa RETEGAL, S.A. dependiente de la Dirección Xeral de Comunicación e Audiovisual de la Consellería de Cultura, Comunicación Social e Turismo de la Xunta de Galicia. En virtud de este convenio, RETEGAL, S.A. lleva a cabo las tareas de mantenimiento del equipamiento físico que conforma la red y que se encuentra distribuido en múltiples puntos de la geografía gallega (radio enlaces, fibras ópticas, *switches*, *routers*, conmutadores, etc.).

6.1.9. Conexión con otras redes científico-académicas y de investigación

El CESGA aloja el nodo de RedIRIS en Galicia, la red de comunicaciones de la comunidad científica española. Este nodo está conectado mediante 3 líneas de 2'5 Gbps y 1 línea de 622 Mbps. A través de RedIRIS, los usuarios de RECETGA acceden a la red paneuropea GÈANT y a otras redes internacionales de I+D como Internet 2, Abilene, Ca*Net, CLARA, etc.

6.1.10. Conexión con redes comerciales

Los usuarios de RECETGA intercambian su tráfico de red con usuarios de redes comerciales a través de diversos puntos. El CESGA aloja el Punto Neutro de Intercambio de Tráfico de Internet en Galicia (GALNIX) que facilita el intercambio de tráfico con origen y destino en la comunidad gallega. Los investigadores gallegos, a través de RedIRIS, también acceden a las redes comerciales a través del punto neutro estatal de intercambio de tráfico de Internet, ESPANIX.

6.1.11. Servicios específicos CESGA a través de la Red

Además de los servicios descritos en el apartado anterior, el CESGA, a través de la red, pone a disposición de sus usuarios otros servicios, entre los que cabe destacar: acceso a servidores de computación de altas prestaciones, acceso a aplicaciones de cálculo y simulación, acceso a *Grids* computacionales, almacenamiento masivo de datos, visualización científica, acceso a la red de aulas de tele-enseñanza (1 por campus), acceso a plataformas software de tele-enseñanza o acceso a cartografía digital.

La existencia de RECETGA posibilita que distintas instituciones usuarias puedan participar en proyectos e iniciativas de investigación nacionales e internacionales tales como IRISGRID, CROSSGRID, LHCb o MISION PLANCK. La red también ha fomentado el uso de Internet como medio de intercambio de información especializada. La alta capacidad de transferencia de datos ha favorecido la creación de BUGALICIA, una de las mayores bibliotecas universitarias de publicaciones electrónicas on-line de Europa, así como la constitución de redes temáticas en Galicia como la Rede Galega de Bioinformática o la Rede Galega de Computación Paralela, Distribuida e de Tecnoloxías GRID

6.1.12. Especificaciones Técnicas

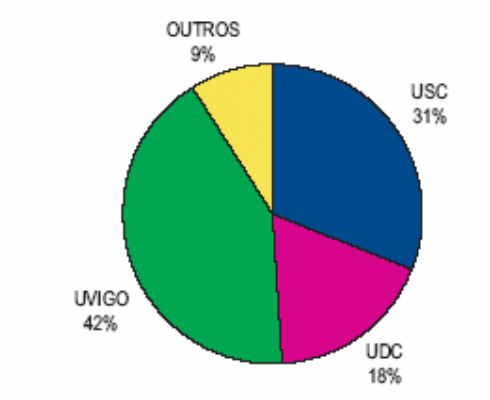
Red Troncal	• Basada en Radio enlaces SDH a 155 Mbps y Fibra Óptica. • Conmutadores ARM de FORE ASX-200 y ASX1000 • Conmutadores ATM de CISCO LS1010
Red de acceso	• Fibra óptica, radioenlaces SDH de 155 Mbps • Radio enlaces 4x2 Mbps. Enlaces RDSI, *wireless*. • Conmutadores FORE, *routers* CISCO, ENTERASYS
Gestión de red	• Basada en SPECTRUM de Aprisma y desarrollos propios.
Conexión a RedIris	• Tres líneas a 2,5 Gbps e 1 a 622 Mbps
Red Interna CESGA	• Gigabit Ethernet – Fast Ethernet – Red ATM • Conmutadores ATM de FORE ASX 200, ASX 1000 • Conmutadores Fast Ethernet 3COM y 2810. • Conmutadores Gigabit FORE ESX 4800 y 3COM • ATM y Gigabit hasta el puesto en los servicios que lo requieran
Tráfico	• CBR, ABR, VBR, UBR

6.1.13. Estado actual

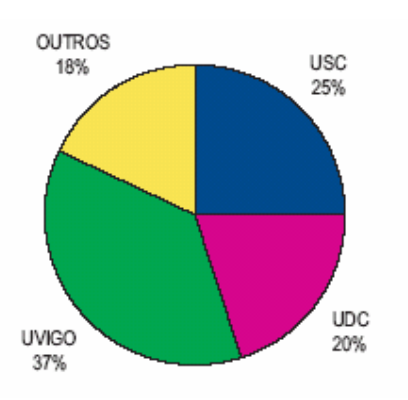
El tráfico registrado en la Red de Ciencia y Tecnología (RECETGA) se incrementó durante el año 2003 en un 100 % respecto al año anterior.

En los siguientes diagramas se muestra la distribución del tráfico de entrada/salida a RECETGA por Instituciones.

Porcentaje de bytes de entrada



Porcentaje de bytes de salida



Además, durante este período se amplió la conexión de RECETGA con la red de investigación estatal RedIRIS2, y se puso en producción el Punto Neutro de Intercambio de Datos de Internet en Galicia (galNIX). La nueva conexión de RECETGA con RedIRIS2 comenzó su actividad en febrero del 2003. Actualmente, la

Red de Ciencia y Tecnología de Galicia está conectada con la red estatal por tres enlaces con una capacidad de 2.5 gigabits por segundo (Gbps) que unen esta comunidad con el País Vasco, Madrid y Cataluña, y uno de 622 megabits por segundo (Mbps) que la conecta con Asturias. Antes de la ampliación, RECETGA disponía de una única conexión con Madrid que apenas alcanzaba los 40 Mbps de entrada y 20 Mbps de salida. Estas ampliaciones se enmarcan en la nueva infraestructura prevista por RedIRIS2, que contempla la transformación de su anterior estructura en estrella, en la que todos los puntos de la red están conectados única y exclusivamente con Madrid, por un nuevo diseño en forma de malla, que incluye también conexiones entre los nodos periféricos.

Por otro lado, en mayo del 2003 comenzó a funcionar el Punto Neutro de Intercambio de Datos de Internet en Galicia (galNIX), situado en las instalaciones del CESGA y gestionado por esta institución.

6.2. Migración

Dada la estructura y tipo de servicios de la RECETGA, se considera que, de los posibles caminos de migración posibles descritos anteriormente, el más aconsejable es de doble pila, en donde se tramitan las dos versiones IP de forma independiente. Para ello es necesario adaptar el hardware y software de los routers para soportar el protocopo Ipv6 admás del Ipv4, teniendo en cuenta que las tablas de enrutamiento se duplican ya que habrá una para Ipv4 y otra para Ipv6.

Para la realización del cambio es necesario una estrategia de migración paulatina en donde se habilitan por tramos el direccionamiento de los diferentes interfases partiendo desde las líneas troncales hacia los diferentes enrutadores de los centros, con lo cual no es necesario una estrategia de tunel, conviviendo ambos protocolos sin problemas de forma nativa. El cambio deberá ser coordinado con los diferentes gestores de red de los centros para que evalúen e instalen el hardware o software adicional que sea necesario como firewalls adaptados a Ipv6.

Esta migración presenta diversas cuestiones que deben de ser tomadas en cuenta y que se analizan a continuación. En concreto, la gestión de los rangos de nuevas direcciones de tal forma que se minimice la dispersión de direcciones para optimizar al máximo las tablas de enrutamiento, la migración software y hardware de los diferentes componentes de red y la modificación del sistema de captura de información estadística. El resto de los componentes de los servicios tienen una migración sencilla que puede ser realizada sin complicaciones tanto antes (como es el caso del DNS) como posteriormente.

6.2.1. Gestión administrativa y técnica

- **Gestión de la asignación de direcciones IPv6**

  Debe analizarse la línea a seguir en la definición de políticas de gestión del direccionamiento IPv6. En la actualidad el CESGA obtiene bloques de

direcciones de la entidad Red Iris, al igual que las universidades gallegas. Este planteamiento deriva en el análisis de los siguientes puntos:

- Se observa un desaprovechamiento de las posibilidades de agregación de direcciones. El CESGA o las universidades obtienen bloques de direcciones de Red Iris de forma independiente, siendo dichos bloques discontinuos en general. Esto provoca un incremento de las rutas anunciadas hacia y desde dicha entidad. La minimización del tamaño de las tablas de encaminamiento globales es de relevancia para reducir la memoria y la carga de procesado de los *routers* de *backbone* Internet, procurando el objetivo de máxima agregación de direcciones, lo cual reduce el tamaño de las tablas de encaminamiento.

- Debe buscarse el conseguir una mayor facilidad de gestión y configuración del direccionamiento en RECETGA ante situaciones de *multihoming*. En el caso de poseer una conexión a Internet alternativa a RedIris es más sencillo el planteamiento de políticas de encaminamiento si las direcciones manejadas son propias y no cedidas por otro organismo.

- La asignación de direcciones en RECETGA de forma autónoma (independiente de RedIris) simplifica la gestión del tráfico a nivel de seguridad, calidad de servicio y otro tipo de políticas.

- El CESGA es *Local Internet Register* y, por lo tanto, puede solicitar direcciones directamente a RIPE sin necesidad de intermediación de RedIris.

Por los condicionantes presentados se considera que una solución de direccionamiento de RECETGA gestionada y administrada por el CESGA, en la cual se soliciten bloques continuos a la entidad RIPE europea, es más conveniente que la situación actual.

En la actualidad no existe una solución de *multihoming*, es decir, se dispone de un único punto de salida a Internet, por lo que por el momento, y de forma temporal, puede hacerse uso de los rangos asignados por RedIris como se ha hecho hasta el momento con el direccionamiento IPv4.

6.2.2. Análisis de cambios en la infraestructura

- **Actualizaciones en los *routers***

Para proporcionar soporte IPv6 es necesario realizar algunas actualizaciones hardware. Como ya se comentó en el punto 3.1.1 en RECETGA se utilizan tres marcas distintas de *routers*: Cisco, Juniper y Enterasys. Comentamos las modificaciones que debemos realizar en cada uno de estos:

- Cisco: Es necesario incrementar la memoria de los equipamientos de *routing*. El incremento necesario dependerá de la serie del dispositivo.

- Juniper: En este caso el equipo concreto es un Juniper M10 que se prevé llegue a comunicar rutas mediante BGP, se recomienda por tanto un incremento de la memoria interna hasta 1 GB.

Mostramos a continuación una tabla de la memoria mínima que deben poseer una vez se hayan actualizado según el modelo de *router*:

+

Modelo	**Memoria interna**	**Memoria externa**
Cisco 1600, 1700, 2500	128 MB	64 MB
Cisco 3700	256 MB	128 MB
Cisco 7200	512 MB	128 MB
Juniper M10	1 GB	--

- Enterasys: En RECETGA se dispone de *routers* de la serie 2400 de esta marca, los cuales son modelos de bajas prestaciones. La actualización a IPv6 de dichos dispositivos no es posible, dado que se necesita un módulo hardware que está sólo disponible para la gama superior (la serie 2800). Consultar http://www.enterasys.com/products/routing/ipv6/.

6.2.3. Análisis de cambios en la monitorización

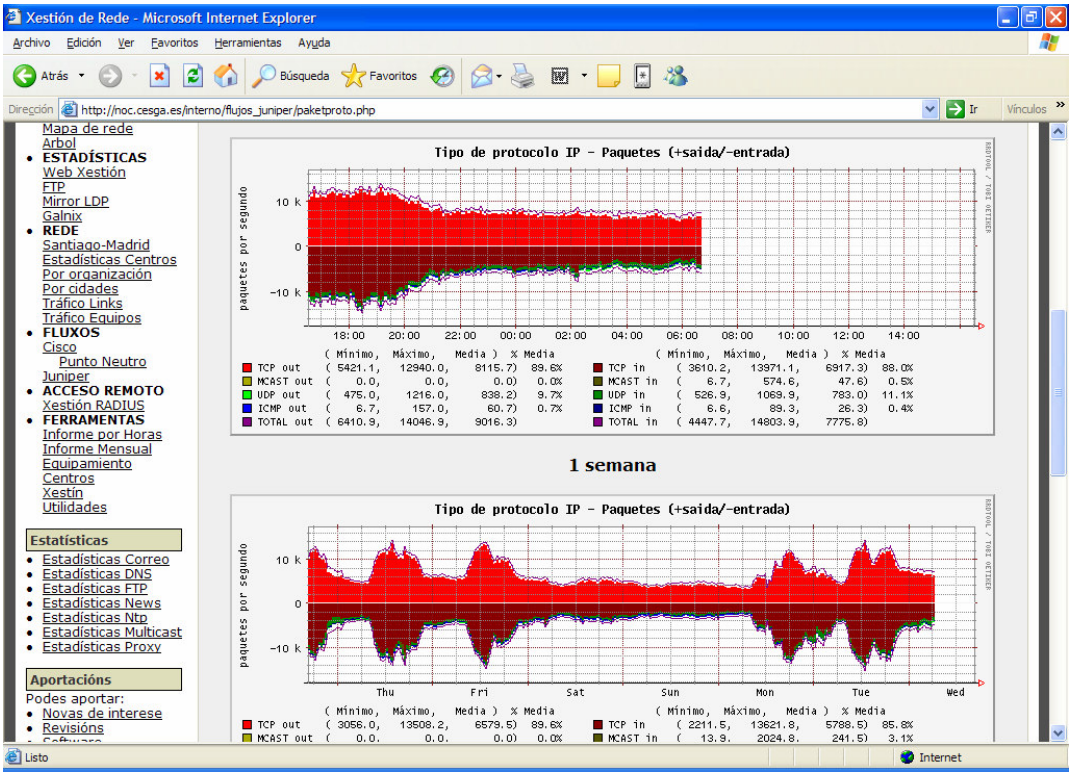
El CESGA dispone actualmente de un sistema de captura de estadísticas que utiliza la exportación de los flujos en formato NetFlow V5. Este sistema es utilizado por los difentes centros para conocer el grado de sus conexiones así como por los técnicos del CESGA para el control y gestión de la red, fundamentalmente en la detección de intrusos y malas utilizaciones: detección de tráficos anormales, P2P, escaneos, etc.

Para poder controlar y distinguir el tráfico Ipv6 frente al Ipv4 es necesario habilitar la nueva versión del NetFlow (la versión 9[3]) que incluye plantillas de forma que es posible adecuar la información al protocolo. Para ello son necesarios los siguientes pasos:

6.2.3.1. Adaptar la aplicación de gestión actual para obtener los paquetes de la nueva versión y poder procesarlos adecuadamente. Dicha adaptación incluirá la definición de nuevas estadísticas que permitan medir el grado de uso de cada uno de los protocolos de tal forma que se pueda decidir cuando eliminar o realizar el siguiente paso de la migración.
6.2.3.2. Habilitar en cada uno de los enrutadores la exportación de la información en este nuevo formato de forma gradual.

[3] http://www.cisco.com/application/pdf/en/us/guest/tech/tk362/c1550/ccmigration_09186a00800a3db9.pdf

En cualquier caso hay que tener en cuenta que la nueva versión del NetFlow no es estándar en este momento y, de hecho, existen otras alternativas como IPFix, que difieren en el formato de exportación. En caso de que no esté disponible esta solución para todos los equipamientos de red afectados, también es posible utilizar SNMPv3 para la contabilidad y desagregación de las estadísticas de tráfico.



Aplicación de estadísticas de tráfico de la RECETGA

7. Conclusiones y recomendaciones.

A través de las experiencias descritas en este informe, se ha llegado a la conclusión de que Ipv6 se encuentra en un estado de madurez suficiente en cuanto a especificaciones y soporte por los diferentes fabricantes de software y hardware de prestación de servicios de red, así como en la mayor parte de los sistemas operativos de uso habitual. Asimismo, la implantación de Ipv6 en la red de investigación gallega podría impulsar la utilización de las mejores capacidades que presenta Ipv6 frente a la versión Ipv4 en los aspectos de seguridad, calidad de servicio, aprovechamiento del ancho de banda (por ejemplo, con los *jumbo packets*), etc, necesarias por las nuevos métodos de investigación existentes en la comunidad internacional y la explosión de transmisión de datos científicos actual.

Dentro de las posibles estrategias de migración existentes, se considera que es preferible optar por la denominada de doble pila, manteniendo la coexistencia temporalmente entre las dos versiones del protocolo IP. Esto implica importantes cambios en el hardware y software de los sistemas de red existentes, que se han de realizar previamente a la inclusión del nuevo protocolo. Ésta ha de ser gradual, partiendo de los enlaces troncales hacia los diferentes centros y coordinada con los diferentes gestores de red de las instituciones involucradas. En cualquier caso, aunque el CESGA habilite este nuevo protocolo, no implica que los diferentes centros tengan que habilitarlo, siendo una posible opción que podrán habilitar según sus necesidades. Sí es importante el que la gestión de los rangos de direcciones utilizadas por todas las instituciones conectadas a la RECETGA tengan la mínima dispersión para mantener controladas las tablas de enrutamiento.