

Optimización de la Plataforma de Desarrollo Click para la Auto-configuración sin Estado en IPv6

Investigación

¹C.Dr. Marco A. Hernández, ¹Dr. José A. Calderón, ²C.M.C. Juan A. Nungaray, ²Ing. Iván Castillo, ³C. Dr. Omaira Parada

¹ Departamento de Sistemas y Computación y Eléctrica y Electrónica

Miembros del Cuerpo Académico del Sistemas Distribuidos

Instituto Tecnológico de Aguascalientes, Av. Adolfo Lopez Mateos 1801 Ote. Fracc. Ojocaliente, Aguascalientes, Ags. C.P. 20256, Tel: 01 (449) 9105002, Fax: 01 (449) 9700423

{marco_hernandez_vargas, jac012000}@hotmail.com

Departamento de Sistemas y Computación

²Instituto Tecnológico del Llano Km. 18 Carr. Ags.-SLP., C.P. 20330, El llano, Ags. Tel: 01 (449) 9161251, Fax: 01 (449)9162094

{a.catillo.z, aenr25}@hotmail.com

³Universidad Cuauhtémoc Plantel Aguascalientes, Av. Independencia No. 100 Trojes de Alonso, C.P. 20116 – Aguascalientes, Ags., Tel 01(449) 9731122 ext.116.

Coord.posgrados@ucuauhtemoc.edu.mx

Resumen

Actualmente estamos experimentando una evolución continua de la Internet. Propuestas sobre nuevos protocolos y nuevas arquitecturas están siendo analizadas para solucionar algunos de los problemas presentes: adecuada calidad de servicio, movilidad, seguridad, entre otros. Por otro lado, para un mejor entendimiento de estas propuestas es necesario disponer de nuevas herramientas de prototipado rápido. Existen diversos sistemas para la implementación de redes con los protocolos y arquitecturas actuales. Uno de ellos es la plataforma Click, que permite diseñar, configurar y emular componentes internos de routers y así poder soportar los problemas actuales en la red. La plataforma Click tiene un gran soporte para desarrollar procesadores de paquetes bajo IPv4 pero desafortunadamente no es así para IPv6. Este trabajo presenta el diseño, implementación y evaluación de un elemento para la generación de un mensaje de anunciación de router para llevar a cabo el proceso de auto-configuración sin estado en IPv6 desarrollado sobre la arquitectura de software Click y poder así, contribuir al desarrollo de elementos para el soporte de IPv6.

Palabras clave: IPv6, ICMPv6, Click, IPv4.

Abstract

Actually we are experimented a continue evolution of Internet. Proposals about new protocols and architectures are being analyzed to solve some problems: Quality of Service, mobility, security and others. In other hand, acquire a better understanding about these proposals is necessary to get new fast prototyping tools. Exist different systems for network implementation with actual protocols and architectures. One of these is the Click platform, it allows design, configure and emulate internal router components and this way can supports the actual network problems. Click Platform have a great support to development packet processors under IPv4 but unfortunately there is no support for IPv6. This research presents the design, implementation and evaluation of an element to generate a router advertisement message to achieve the process of IPv6 stateless auto-configuration developed on Click platform software and this way can contribute to the elements development to support IPv6

Keywords: Ipv6, ICMPv6, Click, IPv4.

Introducción

Desde su aparición, la Internet ha sufrido una evolución vertiginosa y constante que va más allá de

las expectativas iniciales con las que fue creada, esto se refleja con el imparable desarrollo de aplicaciones y dispositivos con cada vez más exigencias (Videoconferencia, Voz sobre IP, etc), provocando una serie de problemas y dificultades a las limitaciones de la actual arquitectura basada en IPv4 tales como el reducido espacio de direccionamiento, sin soporte para la movilidad, pobre calidad de servicio, la gran dimensión de las tablas de enrutamiento en el troncal de Internet, entre otros.

A partir de estos problemas, se han presentado varias soluciones que están revolucionando la capa de red, entre ellas se encuentra el protocolo IPv6 que plantea un cambio de arquitectura en dicha capa comenzando con un nuevo esquema de direccionamiento, encaminamiento y un mejor procesamiento de su cabecera. Sin embargo, esto requiere la actualización de varios protocolos para permitir su funcionamiento y compatibilidad bajo la arquitectura de IPv6, tal es el caso de protocolo de mensajes de control de Internet versión 6 (Internet Control Message Protocol version 6, ICMPv6) el protocolo de descubrimiento de vecinos (Neighbor Discovery Protocol, ND), protocolo de configuración dinámica de host versión 6 (Dynamic Host Configuration Protocol versión 6, DHCPv6), etc.

Así mismo se han creado varias estrategias o mecanismos de transición que permiten el uso paralelo entre IPv4 e IPv6 para lograr una “evolución transparente” hacia IPv6. No obstante, para poder aprovechar las arquitecturas y protocolos actuales y fomentar la integración hacia IPv6 se requieren de herramientas que permitan su rápido prototipado e implantación. Una de las herramientas utilizada para el diseño, desarrollo y emulación de procesadores de paquetes (routers) es la plataforma Click. Este artículo presenta el diseño, desarrollo e implementación de un elemento para la generación automática y transparente de mensajes de anunciación de router (Router Advertisement, RA), que apoyan al proceso de auto-configuración sin estado sobre un router dual IPv4/IPv6 sobre la plataforma Click y así contribuir al desarrollo de elementos bajo el protocolo IPv6 en Click.

Tecnologías relacionadas con la propuesta

En esta sección se incluye una descripción general de las tecnologías o protocolos involucrados en esta propuesta.

Protocolo de Internet versión 6 (IPv6). El protocolo de Internet en su versión 6 (Internet Protocol versión 6, IPv6) [1] propone un mayor espacio de direccionamiento y un mejor

procesamiento de los paquetes. La cabecera de IPv6 se muestra en la figura 1.

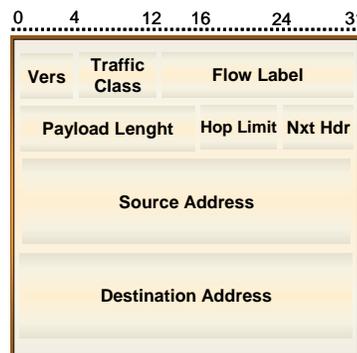


Fig. 1. Cabecera IPv6

La cabecera IPv6 tiene una longitud fija de 40 bytes dividida en 8 campos: los 6 primeros (8 bytes) proporcionan información de control y los 2 restantes (32 bytes) se reservan para las direcciones (source address, destination address). Esta reducción de campos aunado a la longitud fija de la cabecera implica su procesamiento rápido en los routers y por consecuencia representa una optimización del flujo de los paquetes en el camino de reenvío.

Para que un paquete IPv6 pueda llegar a su destino basta con que tenga la cabecera básica. Sin embargo, en ocasiones se requiere agregar información adicional en el paquete IPv6. IPv6 permite el uso de cabeceras adicionales a través del campo Next Header. Estas cabeceras son colocadas inmediatamente después de la cabecera básica IPv6 y son contadas como parte de la longitud de carga útil. La longitud de las cabeceras de extensión es variable y depende del tipo de la que se trata, pero siempre será un múltiplo de 8 bytes. Entre las cabeceras de extensión se encuentran las siguientes: salto a salto (Hop-by-Hop Header), de enrutamiento (IPv6 Routing Header), de fragmentación (IPv6 Fragment Header), de autenticación (IPv6 Authentication Header) y opciones de destino (Destination Options Header).

El esquema de direccionamiento de IPv6 [2] es una de sus principales características y ventajas este nos ofrece un espacio de 2¹²⁸ direcciones IP equivalente a 340,282,366,920,938,463,374,607,431,768,211,456 o lo que es lo mismo a 50,000 direcciones por metro cuadrado sobre la tierra. Una dirección IPv6 esta representada por 8 números hexadecimales de 4 dígitos cada uno divididos por “:”. A continuación se muestra un ejemplo de una dirección IPv6.

fe80:0000:0000:0000::0210:b5ff:fe79:3565

El nuevo esquema de direccionamiento que brinda IPv6 junto con las demás características ya

mencionadas anteriormente, nos conduce a una de sus innovaciones principales, “La Auto-configuración de Direcciones”. Esta característica permite que los nodos decidan como auto-configurar sus interfaces en IPv6 dependiendo de su contexto en la red y que así se puedan comunicar. Esta función, entre otras, es llevada a cabo por dos protocolos esenciales en IPv6: ICMPv6 y ND. El protocolo ICMPv6 [3], se encarga del reporte de errores, descubrimiento de ruta y diagnósticos. Un nodo recién incorporado a la red, auxiliándose del protocolo ND [4], detecta la presencia de otros nodos en el mismo enlace, routers locales, prefijos de red, auto-configuración y resolución de direcciones.

Protocolo de Mensajes de Control de Internet versión 6 (ICMPv6)

La notificación de errores, descubrimiento de ruta y diagnósticos son tareas fundamentales para que IPv6 lleve a cabo sus objetivos. Es por esto que el Protocolo de Mensajes de Error (ICMP) ha sido actualizado para su operación bajo IPv6 y se encuentra descrito en [3]. Los mensajes ICMPv6 pueden ir precedidos por una cabecera básica IPv6 o por una o más cabeceras de extensión, cualquiera que sea el caso, la cabecera ICMPv6 debe ser identificada con el valor 58 en el campo Next Header de la cabecera inmediata anterior. El formato definido para la cabecera ICMPv6 se muestra en la figura 2.

Existen dos clases de mensajes ICMPv6 que son identificados por un rango de valores correspondientes al campo “Tipo”: Mensajes de Error y Mensajes Informativos. Los Mensajes de error tienen un valor que va de 0 a 127. Los Mensajes Informativos comprenden un rango de valores entre 128 y 255.

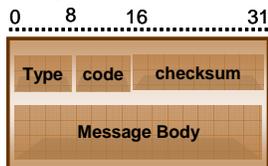


Fig. 2. Cabecera ICMPv6

Cálculo del “Checksum”. Para la detección de errores en los mensajes ICMPv6, este protocolo hace uso de la suma de verificación (checksum) de 16 bits [3]. El procedimiento por el cual se obtiene dicho valor es el siguiente:

1. Se crea una pseudo-cabecera (pseudo-header) a partir de los campos de la cabecera IPv6 (figura 3). Esta pseudo-cabecera se coloca encima del mensaje ICMPv6 y es usada en vez de la cabecera IPv6.

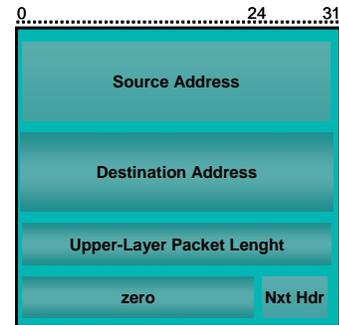


Fig. 3. Pseudo-cabecera ICMPv6

- *Campos de direcciones.* La dirección origen y dirección destino se obtienen de la cabecera IPv6.
 - *Longitud del paquete de capa superior (Upper-Layer Packet Length).* ICMPv6 no incluye información acerca de la longitud de los mensajes, por lo tanto debe usar el valor especificado por el campo de “longitud de carga útil” (payload length) de la cabecera IPv6. Si existe una o más cabeceras de extensión entre el mensaje ICMPv6 y la cabecera IPv6 estas deben restarse al valor de “longitud de carga útil”.
 - *Cero.* Este campo es llenado con ceros.
 - *Siguiente cabecera (Next Header).* Como ya se ha mencionado anteriormente el valor designado para ICMPv6 es el 58 y debe especificarse en este campo.
2. Para realizar el cálculo, el campo de checksum del mensaje ICMPv6 debe rellenarse con ceros.
 3. Se deben de ir sumando grupos de 16 bits comenzando por la pseudo-cabecera y siguiendo con el mensaje ICMPv6. Después se debe obtener el “complemento a uno” del resultado de cada suma y sumar dichos complementos. El resultado de la suma de los complementos a uno será el valor del checksum para el mensaje ICMPv6.

ICMPv6, a diferencia de su versión empleada en IPv4, incluye una pseudo-cabecera. Este cambio es necesario para proteger los mensajes ICMPv6 en caso de que los campos de la cabecera IPv6, de los cuales depende, sufran inconsistencia o sean corrompidos durante su trayecto.

Protocolo de Descubrimiento de Vecinos (Neighbor Discovery). El protocolo de descubrimiento de vecinos (Neighbor Discovery, ND) [4], corresponde a la combinación de los protocolos ARP en IPv4, ICMP Router Discovery [5], e ICMP Redirect [6]. ND permite que un nodo IPv6, al ser incorporado a una red, detecte la presencia de otros nodos sobre el mismo enlace y determine sus direcciones de capa dos, también permite identificar a los routers pertenecientes

al mismo segmento de red y mantener información de conectividad para detectar los cambios en la red.

Este protocolo define varios mecanismos para solucionar problemas relacionados con la interacción entre nodos conectados sobre un mismo enlace, tales como: descubrimiento de routers, prefijos, parámetros, auto-configuración y resolución de direcciones, determinación del siguiente salto, detección de nodos no alcanzables y de direcciones duplicadas, y re-direccionamiento.

ND hace uso de los servicios ofrecidos por ICMPv6 para llevar a cabo las tareas mencionadas. ND hace uso de cinco tipos de mensajes:

- Router Solicitation: Solicitud de router, cuando una interfase es activada, los nodos pueden enviar este tipo de mensaje para solicitar que los routers se anuncien inmediatamente en vez de esperar el tiempo programado para hacerlo.
- Router Advertisement: Anunciación de router, los routers informan de su presencia junto con parámetros acerca del enlace y de Internet. Este tipo de mensajes se generan periódicamente o en respuesta a una "solicitud de router".
- Neighbor Solicitation: Solicitud de vecino, es generado por un nodo para determinar la dirección de capa de enlace de un vecino, o para verificar que el vecino sigue activo, así mismo se usa para detectar direcciones duplicadas.
- Neighbor Advertisement: Anunciación de vecino, se genera como una respuesta a una "solicitud de vecino", o para informar cambios en la dirección de capa de enlace del nodo.
- Redirect: Redirección, es empleado por los routers para informar a los nodos de un mejor salto para un determinado destino.

Una de las principales ventajas que presenta el protocolo ND, frente a los mecanismos empleados en IPv4, es la auto-configuración de direcciones [7] otorgando una gran potencialidad y flexibilidad a IPv6. Existen dos mecanismos que utiliza IPv6 para la auto-configuración de nodos: auto-configuración sin intervención o automática (stateless) y la predeterminada o manual (stateful).

Plataforma Click. Click es una arquitectura de software cuyo objetivo es la construcción y emulación de procesadores de paquetes abiertos, flexibles y configurables. De este modo, Click nos permite diseñar de manera modular routers utilizando protocolos y arquitecturas de red actuales. Además, las configuraciones hechas en Click son escalables posibilitando incorporar nuevas funciones, como es

el caso de IPv6. Para el manejo de la plataforma Click, se debe de tener conocimiento de dos partes esenciales, su arquitectura y el lenguaje.

Arquitectura. La arquitectura de Click se basa en los *elementos*, donde cada elemento es un componente de software que realiza una operación simple dentro del router. Estos elementos se conectan entre si formando un grafo direccionado a través del cual viajarán los paquetes. De esta manera, la configuración de un Router Click se trata de un grafo direccionado de elementos que representa su *trayectoria de reenvío* (forwarding path). Los principales componentes de la arquitectura de Click son: los elementos, los paquetes, las conexiones, y el procesamiento. La figura 4 muestra un ejemplo de una configuración común hecha en Click.

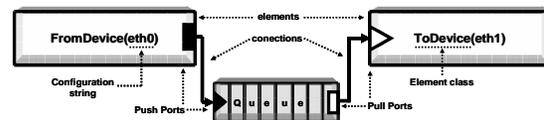


Fig. 4. Configuración Click sencilla

Lenguaje. Click define su lenguaje propio que describe textualmente una configuración (grafo direccionado) de un router a través de una *declaración* y una *conexión* que crean y conectan un elemento respectivamente. A continuación se presenta un ejemplo simple de una declaración y conexión en lenguaje Click.

```
name :: class(config-string);
// declaración
name1[port1] -> [port2]name2;
// conexión
```

Click además provee de estructuras que permiten crear elementos compuestos. El diseño del lenguaje persigue dos objetivos: la facilidad de lectura y conveniencia entre herramientas, de estos se desprenden los siguientes principios específicos:

- Se trata de un lenguaje declarativo
- Es simple
- Existe equivalencia entre los programas y los grafos de configuración
- Las herramientas no requieren conocer completamente la semántica de Click

Diseño y desarrollo de un elemento para el proceso de auto-configuración en IPv6

En esta investigación, nosotros presentamos el diseño y desarrollo de un elemento para la arquitectura de desarrollo de procesadores Click, que implementa las características básicas de un mensaje de Anunciación de Router (RA) y que contribuye al proceso de auto-configuración sin estado en IPv6.

Este elemento permite la generación multidifusión periódica de mensajes RAs y también, como respuesta al mensaje de Solicitud de Router (RS), genera el respectivo mensaje RA unidifusión, ambos casos para apoyar el proceso de auto-configuración a nivel de sitio, de una manera dinámica y transparente en una red IPv6.

Nosotros hemos considerado la trayectoria de reenvío implementada en el Router Dual IPv4/IPv6 desarrollado en [8]. Esta trayectoria no contempla el proceso de auto-configuración dinámica a nivel de sitio; generación automática de mensajes RAs y procesamiento de mensajes RSs. Esta limitación no permite la incorporación dinámica y transparente de nuevos hosts a la red IPv6.

El diseño de la configuración del Router Dual contiene las funciones más importantes del protocolo ND de IPv6. Sin embargo, para poder implementar el protocolo ND en la plataforma Click, fue necesario, además de diseñar la trayectoria de reenvío, extender esta plataforma creando un nuevo elemento a nivel de código de C++ que permita su integración transparente dentro de la plataforma.

La arquitectura Click ofrece una gran variedad de elementos que conforman su lenguaje: 409 elementos para el soporte de IPv4 y 19 elementos para el soporte de IPv6, aproximadamente (hasta la fecha de haber escrito este artículo). Los elementos de Click definen el aspecto de las configuraciones y sus funciones. Con base en lo anterior, los elementos definidos en Click para el soporte de IPv6 realizan sólo las operaciones básicas del protocolo. Aunque esto podría ser una limitación, no lo es, ya que también ofrece la posibilidad de extender su código para crear nuevos elementos que implementen funciones específicas.

Click no ofrece un elemento para llevar a cabo la función de Anunciación de Router (RA), por lo tanto, impide la realización del proceso de auto-configuración de los nodos. Por lo anterior, se procedió al diseño y desarrollo de un nuevo elemento para apoyar al proceso de auto-configuración a través de la generación multi-difusión de mensajes RAs y la respuesta a la solicitud de mensajes RSs como se especifica en la norma del protocolo ND [4]. El nombre que se le ha asignado al nuevo elemento es *CIP6RouterAdv*.

Diseño del Elemento CIP6RouterAdv. El nuevo elemento creado para llevar a cabo la *Anunciación de Router* se ha llamado *CIP6RouterAdv* y su declaración se muestra en la figura 5.



Fig. 5. Elemento CIP6RouterAdv

El elemento *CIP6RouterAdv* esta formado por los siguientes componentes: un puerto de entrada y un puerto de salida, ambos de procesamiento push y una cadena de configuración que recibe los siguientes parámetros:

- IP6Address1: Dirección IPv6 de la interfase del router asociada con el enlace local.
- IP6Address2: Dirección IPv6 que representa el prefijo publicado en el enlace local.
- EtherAddr: Dirección física de la interfase del router asociada con el enlace local.
- tInterval: Especifica el intervalo de tiempo válido entre las anunciaciones del RA del router.

Se puede observar que los datos proporcionados en la cadena de configuración del elemento constituyen la información básica de la interfase del router y del prefijo de subred deseado para el enlace. Esta información es esencial para la auto-configuración exitosa en los hosts y cualquier omisión de alguno de los parámetros incurriría en un fallo al momento de inicializar el router. Así mismo, proporcionar parámetros erróneos traería como consecuencia el fracaso del proceso de la auto-configuración. Cabe mencionar que este elemento ha sido desarrollado en lenguaje C++ y presenta un tamaño de 200 líneas de código, aproximadamente.

Uso del elemento CIP6RouterAdv. Para contribuir al proceso de auto-configuración en los hosts, el Router Dual ofrece una respuesta a dos eventos importantes: Inicialización del Router y Respuesta a una Solicitud de Router. En la figura 6 se muestra el estado inicial del host, es decir, antes de recibir un mensaje RA. Como se puede observar, el host sólo cuenta con una dirección con alcance local.

```

C:\>ip6v6 Tf
Interfaz 5: Pseudo-interfaz de protocolo de túnel Teredo
GUID {510c5238-99d3-49d6-90d7-cd3a8341d1f2}
zonas: link 5 site 2
cable desconectado
usa descubrimiento de vecinos
usa descubrimiento de enrutador
preferencia de enrutamiento 2
dirección de capa de enlace: 0.0.0.0:0
preferred link-local fe80::ffff:ffff:ffff, duración infinite
multidifusión interface-local ff02::1, 1 referencias, no reportable
multidifusión link-local ff02::1, 1 referencias, no reportable
enlace MTU 1280 (enlace MTU 1280)
límite de saltos actual128
tiempo alcanzable 37000ms (base 30000ms)
intervalo de retransmisión 1000ms
transmisiones DAD 0
longitud de prefijo de sitio predeterminada 48
Interfaz 4c: Ethernet: Conexión de Área Local
GUID {895cf07-6d9e-4ef3-a5c7-42fa80c954c2}
usa descubrimiento de vecinos
usa descubrimiento de enrutador
dirección de capa de enlace: 00-16-76-49-a6-ee
preferred link-local fe80::216:76ff:fe49:a6ee, duración infinite
multidifusión interface-local ff02::1, 1 referencias, no reportable
multidifusión link-local ff02::1, 1 referencias, no reportable
multidifusión link-local ff02::1:ff49:a6ee, 1 referencias, último informado
enlace MTU 1500 (enlace MTU 1500)
límite de saltos actual128
tiempo alcanzable 27500ms (base 30000ms)
intervalo de retransmisión 1000ms
transmisiones DAD 1
longitud de prefijo de sitio predeterminada 48
Interfaz 5: Pseudo-interfaz de protocolo de túnel 6to4
GUID {A995346E-9F3E-2ED8-47D1-9cC78A01c073}
no usa descubrimiento de vecinos
no usa descubrimiento de enrutador
preferencia de enrutamiento 1
    
```

Fig. 6. Estado inicial del host antes de recibir un mensaje RA.

- Inicialización del router. Cuando se inicializa el router, a través del elemento CIP6RouterAdv, se envían anunciaciones de router periódicas multi-difusión a todos los nodos en el enlace local; según el tiempo indicado en el parámetro Interval. El RA resultante es producto de la combinación exacta de la información proporcionada en la cadena de configuración con la información indicada por defecto en el comportamiento interno del elemento, es decir, en su código. La figura 7 muestra el host con su información de configuración en IPv6 con un prefijo con alcance de sitio.
- Respuesta a un RS. Este evento se presenta cuando se conecta un nuevo host a la red. El host generará automáticamente mensajes de RS que llegarán a la interfase de entrada del Router Dual y que posteriormente serán enviados al puerto de entrada del elemento CIP6RouterAdv. Una vez que recibe el mensaje RS, el elemento CIP6RouterAdv procederá a realizar un análisis completo de su contenido, tanto de su cabecera básica IPv6 como de la extensión ICMPv6 y determinará los datos que deberá contener el mensaje unidifusión RA a ser enviado al host que se ha conectado a la red.

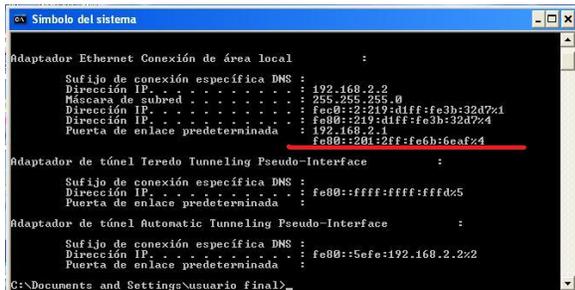


Fig. 7. Pantalla del host que muestra la auto-configuración después de haber recibido un RA.

La figura 8 muestra la pantalla del router en el momento que emite un mensaje RA a petición de un RS.

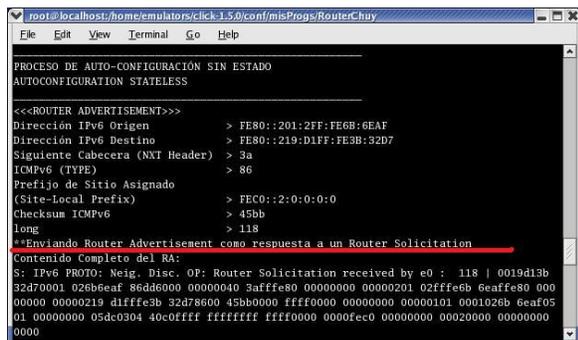


Fig. 8. Pantalla del Router Dual que muestra la transmisión de un RA a solicitud de un RS.

Uno de los procesos más importante que se debe considerar en la generación de los mensajes RAs es el cálculo de la suma de verificación (checksum), ya que si este proceso no es exitoso, el router no podría proceder a la publicación de dichos mensajes y, por lo tanto, la contribución al proceso de auto-configuración sería nula.

En cualquiera de los dos casos, existen procesos internos constantes, uno de los más importantes es el cómputo del checksum para mensajes ICMPv6. Siguiendo el algoritmo marcado por [3], se implementó una función para llevar a cabo el cálculo automático del checksum en el elemento CIP6RouterAdv, la cual recibe como parámetros los campos de la pseudo-cabecera y el inicio y longitud de la cabecera ICMPv6. Posteriormente, la función realiza las sumas y complementos a "1" correspondientes y el valor resultante se anexa en el campo correspondiente al checksum del mensaje RA.

Trabajos Futuros

Este es un trabajo en progreso y el objetivo final será agregar las capacidades de MPLS al router Click. También este elemento CIP6RouterAdv se puede extender para implementar algunas funcionalidades en los agentes de movilidad; modificación de los mensajes de extensión de encaminamiento tipo II y de opciones de destino.

Conclusiones

El proceso de auto-configuración es de vital importancia en la emulación de escenarios con IPv6 y es básico para la implementación de otras funcionalidades o protocolos, como es el caso de movilidad o multi-alojamiento. El diseño y desarrollo del nuevo elemento CIP6RouterAdv en Click proporciona soporte transparente en la auto-configuración sin estado en los hosts en un entorno de red IPv6 con un alcance de sitio. El elemento CIP6RouterAdv ha sido desarrollado estrictamente siguiendo las normas internacionales emitidas por la fuerza de trabajo de Internet (IETF) lo que permite su uso en cualquier red bajo IPv6. La experiencia adquirida en la extensión de la plataforma Click ha sido muy enriquecedora gracias a la escalabilidad, flexibilidad y su característica de software abierto.

Agradecimientos

Agradecemos al Cuerpo Académico de Sistemas Distribuidos del Departamento de Sistemas y Computación por las facilidades brindadas para la realización de este trabajo de investigación.

Referencias

- [1] Deering, S., and R. Hinden, (December 1998). "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460.
- [2] Hinden, R., (July 1998). "IP Version 6 Addressing Architecture", RFC 2373.
- [3] Conta, A. and S. Deering, (December 1998). "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 2463.
- [4] Narten, T., Nordmark, E. and W. Simpson, (1998) "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.
- [5] Deering, S., (1991), "ICMP Router Discovery Messages", RFC 1256, September 1991.
- [6] Postel, J., (1981), "Internet Control Message Protocol", RFC 792, September 1981.
- [7] Thomson, S. and T. Narten, (1998), "IPv6 Address Autoconfiguration", RFC 2462, December 1998.
- [8] Hernández M. et al, (2006), "An IPv4-IPv6/MPLS Router Design Based on Click Platform," *13th. International Congress on Computer Science Research*, Tamaulipas, México, Noviembre de 2006.

Artículo recibido: 28 de enero de 2008

Aceptado para publicación: 19 de junio de 2008