Revista Educación en Ingeniería
ISSN 1900-8260
Enero a Junio de 2014, Vol. 9, N°. 17, pp. 132-142 • © 2014 ACOFI • http://www.educacioneningenieria.org
Recibido: 10/04/2014 • Aprobado: 02/02/2014

SISTEMA DIDÁCTICO PARA EL APRENDIZAJE DE IPV6 BASADO EN NS-3

TRAINING SYSTEM FOR IPV6'S LEARNING BASED ON NS-3

Oscar Polanco Sarmiento
Universidad del Valle (Cali, Colombia)

Resumen

En este artículo se presenta el diseño e implementación de un sistema didáctico para el aprendizaje de IPv6, basado en el simulador de red NS-3. Incluye inicialmente cuatro prácticas que abordan los siguientes temas: configuración de direcciones estáticas, configuración automática de direcciones sin estado mediante el anuncio del prefijo de red por parte de los encaminadores, resolución de direcciones, detección de direcciones duplicadas, detección de inaccesibilidad del vecino, fragmentación de datagramas IPv6 en el sistema final emisor con base en el mecanismo de descubrimiento de la unidad máxima de transferencia de la ruta, la operación del mensaje ICMPv6 de corrección de ruta y el uso de la cabecera de extensión para encaminamiento. El sistema utiliza las funciones básicas del protocolo de descubrimiento de vecinos.

Palabras claves: aprendizaje de IPv6; simulador de red NS-3; ICMPv6; protocolo de descubrimiento de vecinos

Abstract

This paper reports both the design and implementation of an training system for IPv6's learning, based on the NS-3 network simulator. The system initially includes four practices that address the following topics: static addresses configuration, stateless address autoconfiguration through the router advertisements of a network prefix, the address resolution, duplicated address detection, neighbor unreachability detection, IPv6 datagram's fragmentation in the sending end system based on the path maximum transfer unit discovery mechanism, the ICMPv6's redirect message operation and, the use of the routing extension header. The system uses the core functions of the neighbor discovery protocol.

Keywords*:* IPv6's learning; NS-3 Network Simulator; ICMPv6; Neighbor Discovery Protocol

Introducción

Algunos de los operadores de red que forman parte de la internet global, se han interesado en adoptar el protocolo IPv6 (Deering y Hinden, 1998) en los servidores y encaminadores de su propiedad, los cuales conforman la infraestructura de red que les permite proporcionar diversos tipos de servicios; esto lo evidencia el incremento gradual en el número de redes y sitios web que soportan IPv6 (http://www.worldipv6launch.org/measurements/). En sintonía con lo anterior, los temas y conceptos nuevos alrededor de IPv6 han empezado a ser considerados en las nuevas ediciones de algunos libros de texto, los cuales desarrollan los contenidos del dominio o disciplina de *Internetworking* (Comer, 2013) o interconexión de redes de computadores, que algunos simplemente la denominan TCP/IP (Stevens y Fall, 2011).

Lo anterior plantea la necesidad de incorporar los conceptos subyacentes a IPv6 en el aula de clase, específicamente en aquellas asignaturas en las que la interconexión de redes es el tema central. Debido a la complejidad de la disciplina, algunos libros de texto (Peterson y Davie, 2011) sugieren acompañar y complementar algunos de los tópicos con un manual de experimentos en simulación de redes (Aboelela, 2011), el cual usa la herramienta Opnet (http://www.opnet.com/). No obstante, aún no se abordan simulaciones que permitan reforzar los conceptos fundamentales del protocolo IPv6.

En el año 2006 nació la iniciativa que buscaba reemplazar el *software* de simulación de redes NS-2 (http://www.isi.edu/nsnam/ns), que generó como resultando el desarrollo de NS-3, cuya arquitectura y características se encuentran documentadas detalladamente en el sitio web del producto (http://www.nsnam.org). NS-3 tiene términos de licencia GNU GPLv2, que alienta a la comunidad *open source* a participar en su desarrollo de código abierto. Una de las metas fundamentales del diseño de NS-3 es mejorar el realismo de los modelos (Riley y Henderson, 2010). Además, permite que los usuarios sean libres de desarrollar sus simulaciones mediante C++ main() o Phython. NS-3 puede generar resultados de la simulación mediante archivos en formato de texto o en formato pcap (*packet capture*); el formato pcap lo usan las herramientas *tcpdump* y *Wireshark* (Combs, 1998) para representar las tramas capturadas en una red real, lo cual resulta útil en el aprendizaje de los protocolos de red a partir de un escenario simulado.

En 2008 se propuso la implementación básica de la pila IPv6 para el simulador de redes NS-3 (Vincent et al., 2008), la cual soporta las siguientes funciones: el protocolo para descubrir vecinos, en adelante NDP o *Neighbor Discovery Protocol for IP version 6* (Narten et al., 2007); el protocolo ICMPv6 (Conta et al., 2006); y el protocolo que permite la configuración automática de direcciones sin estado mediante el anuncio del prefijo de red por parte de los encaminadores, en adelante SLAAC o StateLess Address AutoConfiguration (Thomson et al., 2007).

Puesto que NS-3 permite experimentar con los conceptos subyacentes a los protocolos de redes en general, y con los conceptos de IPv6 en particular, en este trabajo se desarrolló un programa en C++ main(), utilizando las librerías para IPv6 de NS-3 versión 3.19. Dicho programa está orientado específicamente a facilitar la interacción de los estudiantes con algunos de los conceptos del protocolo IPv6 y al utilizarlo se pueden invocar las cuatro funciones denominadas *lifetime*, *fragment*, *redirect* y *source*, que tienen los siguientes objetivos:

a. Entender y contrastar los procesos involucrados en la configuración de la dirección IPv6 en un nodo, específicamente, cuando se utiliza el método estático y el método automático basado en SLAAC; en este último se puede controlar el tiempo de vida o Valid Life Time del prefijo de red adquirido por el nodo. Se ilustran los mecanismos de resolución de direcciones; de detección de direcciones duplicadas o Duplicated Address Detection, en adelante DAD; y de detección de inaccesibilidad del vecino o Neighbor Unreachability Detection, en adelante NUD. Esto es realizado por la función *lifetime* del programa desarrollado.
b. Realizar el fragmentado de datagramas IPv6 desde el sistema final emisor con base en el mecanismo de descubrimiento de la unidad máxima

de transferencia de la ruta o *Path Maximum Transmission Unit*, en adelante PMTU; dicho mecanismo utiliza el protocolo ICMPv6. Esto lo realiza la función *fragment* del programa desarrollado.

c. Ilustrar la operación del mensaje ICMPv6 de corrección de ruta, en adelante denominado mensaje de *redirec*t, que lo realiza la función de este nombre del programa desarrollado.
d. Usar la cabecera de extensión para encaminamiento de tipo RH0, que tiene valor pedagógico puesto que permite ilustrar el concepto de encaminamiento basado en la ruta definida por el origen. Esto lo ejecuta la función *source* del programa desarrollado.

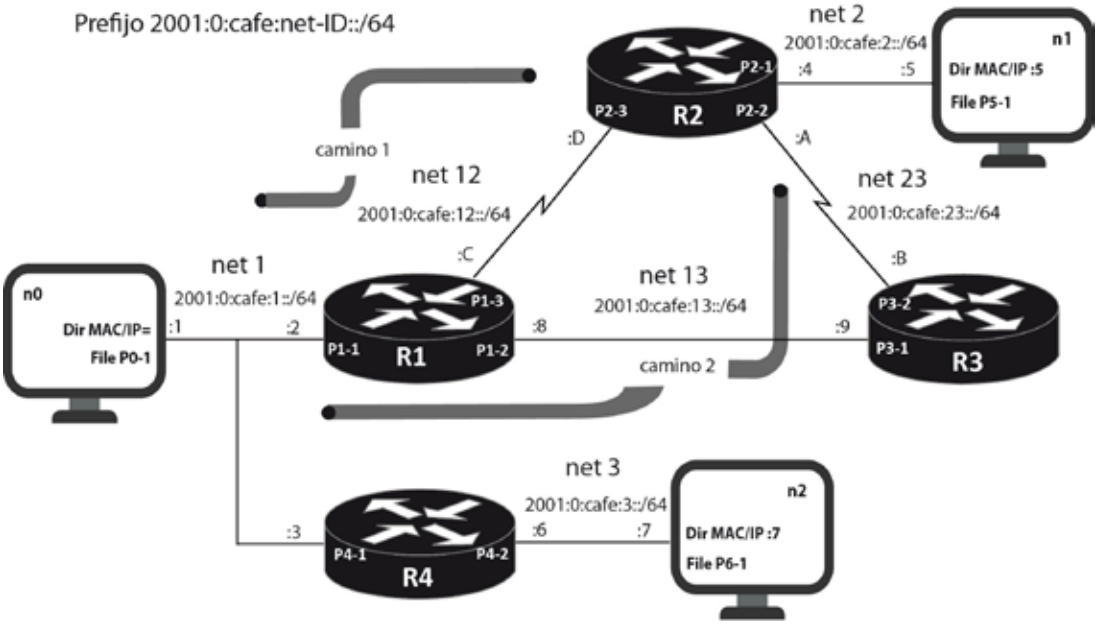
Metodología

La figura 1 representa la red IPv6 que se modeló usando el lenguaje de programación C++ main() del simulador de redes NS-3. En términos generales, la red está constituida por los siguientes elementos: cinco redes CSMA alambradas (net 1, net 2, net 3, net 13 y net 23) y una red punto a punto que usa el protocolo PPP (net 12); un sistema final emisor (nodo n0) que envía tráfico de datos hacia uno de los posibles sistemas finales de destino (nodo n1 o nodo n2), teniendo presente que el destino seleccionado dependerá del objetivo que se desee abordar durante la simulación; cuatro sistemas intermedios o encaminadores de red (R1, R2, R3 y R4); y finalmente, un prefijo de encaminamiento global (2001:0:cafe::/48) asignado a un sitio, dentro del cual se adicionan 16 bits de subred, cuyo valor permite identificar a cada una de las subredes componentes del sitio, por ejemplo, a la subred 13 (net 13) le corresponde el prefijo 2001:0:cafe:13::/64.

Cada interfaz de red de los nodos tiene asignado en los últimos 64 bits de su dirección IPv6, un valor único (Interfaz ID), el cual permite diferenciarla de las demás. En los extremos de cada enlace de la figura 1, se indican los últimos 16 bits de dicho valor, y se omite el valor común 200:00ff:fe00 de los primeros 48 bits; esto debido a que el valor 200:00ff:fe00 se repite en todas las direcciones IPv6 de dichas interfaces. Además, por convención, el valor de los primeros 44 bits de la dirección MAC de cada interfaz es 00:00:00:00:00:0, mientras que el valor de los últimos 4 bits es igual al valor de los últimos 4 bits de su dirección IPv6. De hecho, el valor de los últimos 24 bits de la dirección MAC e IPv6 coinciden, puesto que NS-3, por defecto, opera con el identificador único extendido *EUI-64™*. Todas las redes o enlaces que conforman el sitio tienen una MTU (*Maximum Transfer Unit*) de 1500 bytes, excepto la red net 1, la cual tiene una MTU de 5000 bytes.

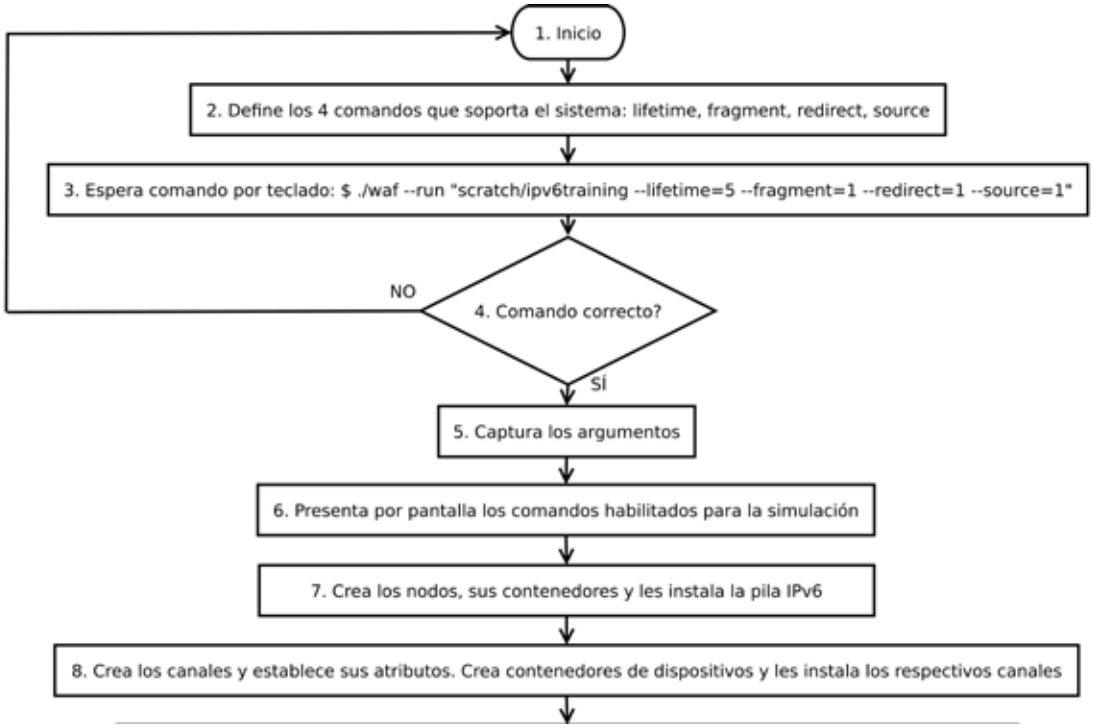
Para la comunicación entre los nodos n0 y n1, por defecto se utiliza el camino 1, tanto en el viaje de ida como en el de regreso, excepto para el escenario en el que se aborda la extensión de cabecera IPv6 para el encaminamiento de tipo RH0, en cuyo caso el origen (n0) determina que la ruta para el viaje de ida sea a través del camino 2. Siempre el nodo destino es el n1, excepto en el escenario de corrección de ruta (*redirect*), en cuyo caso el destino es el nodo n2. Una vez se ejecuta la simulación en NS-3, como parte del resultado se generan 13 archivos de salida en formato pcap, los cuales posteriormente se analizan con *Wireshark*. En la figura 1 se indica el nombre del archivo asignado a cada interfaz. Por ejemplo, el sistema final n0 tiene asociado el archivo P0-1 a su única interfaz (nomenclatura usada para representar: pcap, nodo 0, interfaz 1), mientras que el encaminador R1 tiene asociados los tres archivos P1-1, P1-2 y P1-3, uno por cada interfaz (R1 es el nodo 1 dentro de NS-3). Note la diferencia entre las expresiones nodo 1 y nodo n1, la primera expresión hace alusión al nombre interno que NS-3 le asigna a R1, cuando lo crea como objeto, mientras que la segunda expresión se usa para asignar un nombre al sistema final n1 representado en la red de la figura 1.

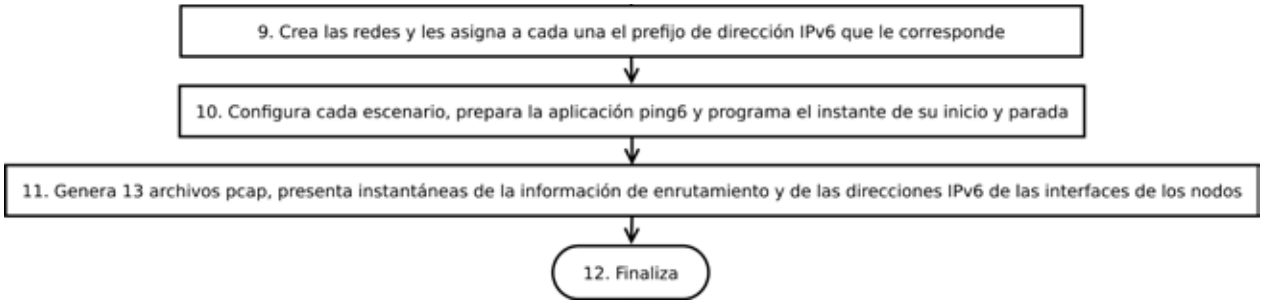
Figura 1. Red IPv6 modelada en NS-3 para el aprendizaje de IPv6.



La figura 2 presenta el diagrama de flujo del "Sistema didáctico para el aprendizaje de IPv6 basado en NS-3", en el que se indican los bloques del sistema desarrollado. En el bloque 2 se definen los nombres de los comandos que soportan las cuatro funciones, mencionadas en la sección de introducción. Dichas funciones se pueden activar de manera exclusiva o combinada. El bloque 3 muestra la línea de comando que se espera por parte del usuario para la ejecución del programa; se utiliza *Waf* para compilar o ejecutar el programa cuyo nombre es *ipv6training*, acompañado de la función que se desee realizar (*lifetime*, *fragment*, *redirect o source*).

Figura 2. Diagrama de flujo del sistema didáctico para el aprendizaje de IPv6 basado en NS-3.





En el bloque 4 se valida el comando que el usuario introduce de acuerdo con la función o funciones que desee ejecutar. En caso de que el comando sea correcto, el bloque 5 captura en una o más variables la función o funciones invocadas. Dichas variables se utilizan para la posterior ejecución de la función solicitada. Cuando el usuario introduce valores por fuera del rango requerido por una función en particular, por ejemplo, valores negativos para la función *lifetime*, el programa de todas maneras se ejecuta pero utiliza los valores establecidos por defecto (de 5 segundos), lo cual se le indica al usuario mediante el bloque 6. El bloque 8 instala las capas físicas y de enlace en cada una de las interfaces de los nodos, mientras que el bloque 7 instala la pila de IPv6 en las mismas. El bloque 9 utiliza el API Ipv6AddressHelper para crear los prefijos de cada subred IPv6 dentro del sitio. Las líneas 13 y 14 de la figura 3 ilustran la creación del prefijo 2001:0:cafe:1::/64, el cual le corresponde a la subred net 1; las líneas 16 a 18 ilustran el código que hace que el nodo n0 carezca inicialmente de una dirección IPv6, y que para generar su dirección de manera automática, tome como base el anuncio del prefijo 2001:0:cafe:1::/64 (y del segundo prefijo, 2001:beca::/64), difundido por el encaminador R1; las líneas 20 a 22 presentan el código que permite asignar de manera estática la dirección 2001:0:cafe:1::2/64 a la interfaz que R1 tiene en el lado de net 1; la línea 24 habilita al nodo R1 para que haga las funciones de encaminador; la línea 25 permite que R1 sea la puerta de enlace por defecto de los equipos finales que están conectados en la net 1, por ejemplo el nodo n0. En el sistema desarrollado, las direcciones IPv6 de los nodos n0 y n1, se configuran automáticamente de acuerdo con los anuncios de encaminador (en adelante RA, *Router Advertisements*) difundidos por los encaminadores R1 y R2, respectivamente. En contraste, la dirección IPv6 del nodo n2 se configura de manera estática. Las líneas 28 a 35 presentan parte del código que hace posible que R1 difunda el prefijo 2001:0:cafe:1::/64 con el valor del atributo "tiempo de vida" igual al capturado en la variable *validlifetime*, dentro del programa.

Las líneas 37 a 39 establecen una ruta estática en R1, definiendo a R4 como próximo salto cuando el destino sea el nodo n2, esto da soporte a la función *redirect*. Las líneas 41 a 46 definen un vector con cinco ranuras de direcciones IPv6 para dar soporte a la función *source* que demanda el uso del camino 2. En dicho caso, los valores del vector especifican parte de la ruta que se debe seguir para ir desde el nodo n0 hasta el n1. En las líneas 44, 45 y 46 se repite la dirección 2001:0:cafe:2:200:ff:fe00:5 del nodo n1 de destino. Esto se requiere para completar las cinco ranuras y evitar que la implementación IPv6 de NS-3 genere mensajes de error. Es decir, las líneas 45 y 46 hacen la función de comodín. Con la línea 48 se controla el tamaño del mensaje de solicitud de eco de acuerdo con el parámetro *packetSize*, lo que permite que con la función *fragment* se pueda solicitar "hacer fragmentado" en el origen.

Los anuncios RA del prefijo de red por parte de R1 y R2 se planifican o programan para que inicien en el instante t=1 de la simulación, y para que terminen en el instante t=2; los valores del tiempo están dados en segundos. El tiempo de vida de las direcciones IPv6 de los nodos n0 y n1, queda determinado por el valor del *validlifetime* anunciado en el mensaje RA, y por lo tanto, expiran en el instante t=*validlifetime*+1. A partir de ese momento, dichos nodos no pueden usar su dirección IPv6 global para enviar o recibir mensajes de solicitud de eco. La aplicación ping6 se programó para iniciar en el instante t=2 y para detenerse en el instante t=12, haciendo un envío máximo de 10 paquetes a una tasa de un paquete por segundo. Durante la simulación se registra la información acerca de las direcciones IPv6 que tiene el nodo n0 en los instantes t=2 (líneas 50 y 51) y t=10.

También se registra la información correspondiente a las entradas en las tablas de encaminamiento de los nodos, y se captura el tráfico enviado y recibido por cada interfaz de los siete nodos que conforman la red.

Figura 3. Fracción de código del programa C++ main() del sistema didáctico.

```
13  Ipv6AddressHelper ipv6;
14  ipv6.SetBase (Ipv6Address ("2001:0:CAFE:1::"), Ipv6Prefix (64));
15
16  NetDeviceContainer tmp0;
17  tmp0.Add (d1.Get (0)); /* interfaz en net1 del nodo n0 */
18  Ipv6InterfaceContainer iic1 = ipv6.AssignWithoutAddress (tmp0);
19
20  NetDeviceContainer tmp1;
21  tmp1.Add (d1.Get (1)); /* interfaz en net1 del nodo R1 */
22  Ipv6InterfaceContainer iicr1 = ipv6.Assign (tmp1);
23
24  iicr1.SetForwarding (0, true);
25  iicr1.SetDefaultRouteInAllNodes (0);
26  iic1.Add (iicr1);
27
28  RadvdHelper radvdHelper1;
29  radvdHelper1.AddAnnouncedPrefix(iic1.GetInterfaceIndex (1), Ipv6Address("2001:0:CAFE:1::"),
    64);
30  RadvdInterface::RadvdPrefixList prefixList1 = routerInterface1->GetPrefixes ();
31  for (RadvdInterface::RadvdPrefixListI iter = prefixList1.begin(); iter != prefixList1.end(); iter++)
32  {
33  ...
34  (*iter)->SetValidLifeTime (int (validlifetime));
35  }
36  ...
37  Ipv6StaticRoutingHelper routingHelper;
38  Ptr<Ipv6StaticRouting> routing1 = routingHelper.GetStaticRouting (r1->GetObject<Ipv6> ());
39  routing1->AddHostRouteTo (iic3.GetAddress (1, 1), iic1.GetAddress (2, 0), iic1.GetInterfaceIndex
    (1));
40  ...
41  std::vector<Ipv6Address> routersAddress;
42  routersAddress.push_back (iic13.GetAddress (1, 1));
43  routersAddress.push_back (iic23.GetAddress (0, 1));
44  routersAddress.push_back (Ipv6Address ("2001:0:CAFE:2:200:ff:fe00:5"));
45  routersAddress.push_back (Ipv6Address ("2001:0:CAFE:2:200:ff:fe00:5"));
46  routersAddress.push_back (Ipv6Address ("2001:0:CAFE:2:200:ff:fe00:5"));
47  ...
48  ping6.SetAttribute ("PacketSize", UintegerValue (packetSize));
49  ...
50  IpAddressHelper ipAddressHelper;
51  Simulator::Schedule (Seconds (2.0), &IpAddressHelper::PrintIpAddresses, &ipAddressHelper, n0);
52  csma.EnablePcapAll (std::string ("ipv6training"), true);
53  ...
```

Resultados

Métodos y procedimientos en la configuración de las direcciones IPv6

La configuración automática de las direcciones IPv6 por medio de SLAAC para los nodos n0 y n1 se activa con el comando indicado en la línea 54 de la figura 4; en este caso, el tiempo de vida de los dos prefijos anunciados por R1 se establece en 5 segundos (a partir de t=1). Las líneas 57 a 64 y 73 a 76 presentan la tabla de encaminamiento del nodo n0, para los instantes t=2 y t=10, respectivamente. De manera similar, las líneas 66 a 71 y 78 a 81 presentan las direcciones IPv6 de la interfaz de red del nodo n0, para los instantes t=2 y t=10, en su orden. Finalmente, las líneas 83 a 87 presentan las direcciones IPv6 del nodo n2. Como se puede observar, el nodo n2 mantiene su dirección global con el paso del tiempo, en contraste con el nodo n0, que pierde sus direcciones globales tan pronto como expira el tiempo de vida válido del prefijo adquirido.

Figura 4. Direcciones IPv6 de los nodos n0 (SLAAC) y n2 (estática).

```
54  $ ./waf --run “scratch/ipv6training --lifetime=5” > display.out 2>&1
55  Run Simulation.
56
57  Node: 0 Time: 2s Ipv6ListRouting table
58  Destination Next Hop Flag Met Ref Use If
59  ::1/128 :: UH 0 - - 0
60  fe80::/64 :: U 0 - - 1
61  2001:0:cafe:1::/64 :: U 0 - - 1
62  ::/0 fe80::200:ff:fe00:2 UG 0 - - 1
63  2001:beca::/64 :: U 0 - - 1
64  ::/0 fe80::200:ff:fe00:2 UG 0 - - 1
65
66  Node: 0 Time: 2s IPv6 addresses
67  (Interface index, Address index)      IPv6 Address
68  (0,0)       address: ::1/128; scope: HOST
69  (1,0)       address: fe80::200:ff:fe00:1/64; scope: LINK-LOCAL
70  (1,1)       address: 2001:0:cafe:1:200:ff:fe00:1/64; scope: GLOBAL
71  (1,2)       address: 2001:beca::200:ff:fe00:1/64; scope: GLOBAL
72
73  Node: 0 Time: 10s Ipv6ListRouting table
74  Destination Next Hop Flag Met Ref Use If
75  ::1/128 :: UH 0 - - 0
76  fe80::/64 :: U 0 - - 1
77
78  Node: 0 Time: 10s IPv6 addresses
79  (Interface index, Address index)      IPv6 Address
80  (0,0)       address: ::1/128; scope: HOST
81  (1,0)       address: fe80::200:ff:fe00:1/64; scope: LINK-LOCAL
82
83  Node: 6 Time: 10s IPv6 addresses
84  (Interface index, Address index)      IPv6 Address
85  (0,0)       address: ::1/128; scope: HOST
86  (1,0)       address: fe80::200:ff:fe00:7/64; scope: LINK-LOCAL
87  (1,1)       address: 2001:0:cafe:3:200:ff:fe00:7/64; scope: GLOBAL
88  Done.
```

La figura 5 corresponde a una parte del vestigio del archivo pcap P1-1. La línea 90 indica la operación DAD que hace n0; en este caso, n0 verifica que su dirección IPv6 de enlace local (fe80::200:ff:fe00:1) no esté duplicada, antes de usarla. La línea 91 indica el anuncio del prefijo 2001:0:cafe:1::/64 por parte de R1, el cual sirve para la configuración SLAAC de n0; el tiempo de vida válido es de 5 segundos. En la línea 92, el nodo n0 valida con éxito la dirección que éste generó (2001:0:cafe:1:200:ff:fe00:1) a partir del prefijo anunciado por R1; en este momento, con la información del RA transmitido por R1, el nodo n0 también conoce la puerta de enlace por defecto (fe80::200:ff:fe00:2) y la dirección MAC respectiva (00:00:00:00:00:02). La línea 93 indica el mensaje de ping enviado por n0 con destino a n1. Las líneas 94 y 95 muestran que cuando llega la respuesta del ping a R1, éste tiene que hacer resolución de direcciones para obtener la dirección MAC del nodo n0. La línea 96 muestra la respuesta del ping de n1 hacia n0, enviada con éxito por R1. Finalmente, puesto que el programa envía un mensaje ICMP de ping cada segundo; la línea 97 muestra el quinto y último mensaje enviado (con secuencia 4) en t=6; debido a que el prefijo expira cinco segundos después de anunciado (en t=1). Puesto que en el instante t=7, n0 realiza la detección de inaccesibilidad del vecino (NUD), mediante una solicitud de vecino (en adelante NS, *Neighbor Solicitation*) dirigida a R1, a la cual este último responde con un anuncio de vecino (en adelante NA, *Neighbor Advertisement*), es necesario que el comando de la línea 54 tenga un valor del *lifetime* mayor a 7, para poder apreciar dicho intercambio.

Figura 5. Trazas de Wireshark del protocolo NDP.

89	No	Time	Source	Destination	Info
90	4	0.009	::	ff02::1	DAD: NS, for fe80::200:ff:fe00:1 from 00:00:00:00:00:01
91	7	1.002	fe80::200:ff:fe00:2	fff02::1 mac 33:33:00:00:00:01	RA: router advertisement from R1. Flags M=0,O=0. Prefix information 2001:0:cafe:1::/64, Valid Lifetime=5, Preferred Lifetime=3, Flag A=1 Link-layer address 00:00:00:00:00:02
92	10	1.010	::	ff02::1: mac 33:33:ff:00:00:01	DAD: NS, for 2001:0:cafe:1:200:ff:fe00:1 from 00:00:00:00:00:01
93	11	2.000	2001:0:cafe:1:200:ff:fe00:1	2001:0:cafe:2:200:ff:fe00:5	Ping n0 → n1: ICMP6, echo request, seq 0, id=0xbeef
94	12	2.039	2001:0:cafe:1:200:ff:fe00:2	ff02::1:ff00:1	Address Resolution: NS, for 2001:0:cafe:1:200:ff:fe00:1 from 00:00:00:00:00:02
95	13	2.039	2001:0:cafe:1:200:ff:fe00:1	2001:0:cafe:1:200:ff:fe00:2	Address Resolution: NA 2001:0:cafe:1:200:ff:fe00:1 (sol, ovr) is at 00:00:00:00:00:01
96	14	2.044	2001:0:cafe:2:200:ff:fe00:5	2001:0:cafe:1:200:ff:fe00:1	n1 → n0: ICMP6, echo reply, seq 0, id=0xbeef
97	21	6.000	2001:0:cafe:1:200:ff:fe00:1	2001:0:cafe:2:200:ff:fe00:5	Ping n0 → n1: ICMP6, echo request, seq 4, id=0xbeef

Funciones de fragmentado, corrección de ruta y ruta en el origen

Las líneas 99 a 102 de la figura 6 representan parte del archivo pcap P1-1. Cuando se usa la función de fragmentado, en la línea 99 se ilustra que R1, después de descartar el primer mensaje con secuencia cero, le indica la MTU apropiada a n0, y en la línea 100 se muestra que n0 le envía a R1 el primer fragmento con 1448 bytes del mensaje, cuyo tamaño total es de 4096 bytes. Cuando se usa la función de corrección de ruta (*redirect*), la línea 101 indica el mensaje ICMPv6 de *redirect*, que R1 le envía a n0, para que instale una ruta que le permita llegar a n2 directamente a través de R4. Cuando se usa la función de ruta en el origen, la línea 103, que representa parte del archivo pcap P3-2, ilustra un mensaje ICMPv6 de solicitud de eco, que viaja

por el camino 2 con destino a n1. La ruta ha sido previamente definida en el origen n0; el mensaje, capturado en la salida de R3, ha usado las dos primeras ranuras de direcciones (2001:0:cafe:13:200:ff:fe00:9 y 2001:0:cafe:23:200:ff:fe00:a) de la cabecera de extensión de ruta en el origen.

Figura 6. Trazas de las funciones fragment, redirect y source.

98	Time	Source	Destination	Info
99	2.022	2001:0:cafe:1:200:ff:fe00:2	2001:0:cafe:1:200:ff:fe00:1	packet too big, mtu 1500, (seq 0 lost)
100	3.000	2001:0:cafe:1:200:ff:fe00:1	2001:0:cafe:2:200:ff:fe00:5	frag (0|1448) ICMP6, echo request, seq 1, length 1448 (first fragment)
101	2.022	fe80::200:ff:fe00:2	2001:0:cafe:1:200:ff:fe00:1	ICMP6, redirect, 2001:0:cafe:3:200:ff:fe00:7 to fe80::200:ff:fe00:3
102	3.004	2001:0:cafe:1:200:ff:fe00:1	2001:0:cafe:3:200:ff:fe00:7	ICMP6, echo request, seq 1 (after NS from n0 to R4, and NA from R4 to n0)
103	2.021	2001:0:cafe:1:200:ff:fe00:1	2001:0:cafe:23:200:ff:fe00:a	echo request, seq 0. Next header IPv6 routing (0x2b), segleft=3,

Discusión

Cuando se ejecuta la función *lifetime,* se observa que la configuración de la dirección IPv6 del nodo n2 (de manera estática), de los nodos n0 y n1 (a través de SLAAC), las operaciones DAD, de resolución de direcciones y NUD, en general, funcionan de acuerdo con lo esperado. Las direcciones IPv6 de alcance global que generan n0 y n1, por medio de SLAAC, expiran de acuerdo al tiempo de vida válido de los prefijos anunciados por R1 y R2, respectivamente. Los equipos de la red, sin excepción, usan la dirección *multicast* ff02::1 de destino, para validar la unicidad de sus direcciones de alcance local y global, antes de permitir que éstas pasen del estado tentativo a los estados preferido, en desuso e inválido. En IPv6 se usa NDP para realizar la operación de resolución de direcciones, lo cual equivale al ARP de IPv4. En NDP, los nodos emplean un intercambio de dos mensajes, el NS y el NA. En el mensaje NS, el nodo solicitante usa como dirección destino de capa 3, la dirección *multicast* de nodo solicitado, la cual genera completando los últimos 24 bits del prefijo ff02::1:ff00:0000/104, con los últimos 24 bits de la dirección IPv6 del nodo destino, como se indica en la línea 94. Además, para generar la dirección *multicast* de nodo solicitado de capa 2, del nodo destino, el valor 33:33:ff, se completa con los últimos 24 bits de la dirección IPv6 del nodo destino; el nodo destino se ha unido previamente a dicha dirección *multicast* de capa 2, desde del momento en que validó y aceptó el uso de la dirección IPv6 que le fue configurada. En relación con la operación NUD, podemos afirmar que las trazas pcap efectivamente muestran que cada nodo verifica periódicamente la existencia de los otros que comparten su mismo enlace; esto lo hace mediante el intercambio de mensajes NS y NA.

Cuando se ejecutan de manera independiente, las funciones *fragment*, *redirec*t y *source*, se observa que las operaciones funcionan según lo esperado. La operación *fragment* permite que el nodo n0 ajuste la MTU a 1500 bytes, en relación con lo indicado en el mensaje ICMPv6 *Too Big*, enviado por R1. Por lo tanto, el mensaje de 4096 bytes que necesita enviar el nodo n0, es fragmentado por éste en dos mensajes de 1428 bytes y uno de 1200 bytes, logrando obtener respuesta de parte del nodo n1. La información capturada en el instante t=3 muestra que para dicho momento la función *redirect* ya ha conseguido instalar en el nodo n0 la ruta óptima que conduce al nodo n2 (2001:0:ca fe:3:200:ff:fe00:7/128), dicha ruta indica que se debe pasar por R4 (fe80::200:ff:fe00:3) para llegar a n2. Con respecto a la función *source*, las trazas de los archivos P1-2, P3-1, P3-2, P2-2, P2-1 y P5-1 indican las ranuras que utilizaron R1, R3 y R2 para hacer el reenvío de los datagramas IPv6, las cuales coinciden con las direcciones de la ruta definida en la aplicación del nodo n0. También se observa que en cada salto el campo *hop limit* disminuye en uno, y que aunque el mensaje de ida viaja por el camino 2, su respuesta regresa por el camino 1. A pesar de que la opción de ruta tipo RH0 está en desuso, permite explicar la opción RH2, la cual está vigente (Stevens y Fall, 2011). Cuando se combina la función *lifetime* con *redirect*, se observa que, a pesar de vencerse el tiempo de vida de

la dirección global del nodo n0, éste continúa enviando mensajes de ping, los cuales quedan confinados en la red net 1; esto se debe a que n0 empieza a utilizar su dirección de enlace local para llegar al destino n2, lo cual supone una operación incorrecta, y es un indicio de un posible problema en los API de la implementación IPv6 en NS-3. Cuando se combina la función *fragment* con *redirect*, se pierde el primer mensaje de ping, debido a que el primer mensaje ICMPv6 *Too Big,* enviado por R4, le llega a R1 y no al nodo n0 (al cual le llega el segundo mensaje *Too Big*). Esto se interpreta como una operación correcta, que simplemente causa la pérdida del primer datagrama, mientras n0 logra ajustar adecuadamente la PMTU. Cuando se combina la función *source* con *fragment*, el nodo n0 hace caso omiso del anuncio del mensaje ICMPv6 *Too Big,* enviado por R1, razón por la cual el nodo n0 continúa enviando paquetes de 4096 bytes que no van más allá de R1, esto denota una limitación en el uso de la extensión de cabecera de ruta en el origen, debido a que n0 no se ajusta al cambio dinámico de la PMTU. Finalmente, otros autores han sugerido el uso de NS-3 en el aula de clase (Wang y Jiang, 2009), para abordar el aprendizaje de conceptos diferentes a IPv6.

Conclusiones

NS-3 es una plataforma de simulación de rápido desarrollo, pertinente para aplicarlo tanto en la investigación como en la educación. A pesar de las limitaciones que tiene la pila IPv6 de NS-3, por ejemplo, la de no soportar los protocolos dinámicos de encaminamiento, podemos afirmar, de acuerdo con nuestra experiencia en el aula de clase, que el sistema didáctico para el aprendizaje de IPv6, desarrollado en C++ main() de NS-3, permite ilustrar de manera simple, directa, y sin requerir recursos diferentes a un computador, los conceptos fundamentales de IPv6, cubriendo con éxito los objetivos mencionados en la introducción. NS-3 brinda un gran potencial para trabajos futuros en el desarrollo de API que soporten diversos protocolos en cada una de las capas de la arquitectura TCP/IP.

Referencias

Aboelela, E. (2011). *Network Simulation Experiments Manual*. (5th Edition). Burlington, Massachusetts: Morgan Kaufmann.

Combs, G. (1998). *Wireshark*. Recuperado el 1.º de febrero de 2014, de http://www.wireshark.org.

Comer, D. E. (2013). *Internetworking with TCP/IP Volume One*. (6th Edition). New Jersey: Addison-Wesley.

Conta, A., Deering, S. y Gupta, M. (2006). *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, Internet Engineering Task Force Request for Comments (RFC) 4443. Recuperado el 1.º de febrero de 2014 de http://www.ietf.org/rfc/rfc4443.txt.

Deering, S. y Hinden, R. (1998). *Internet Protocol, Version 6 (IPv6) Specification*. In IETF (The Internet Engineering Task Force) Request for Comments 2460. Recuperado el 28 de enero de 2014 de http://www.ietf.org/rfc/rfc2460.txt.

Internet Society (2014). *The future is forever, world IPv6 launch – Meachurements*. Recuperado el 28 de enero de 2014 de http://www.worldipv6launch.org/measurements.

Narten, T., Nordmark, E., Simpson, W., y Soliman. H. (2007). *Neighbor Discovery for IP version 6 (IPv6)*, Internet Engineering Task Force Request for Comments (RFC) 4861. Recuperado el 1.º de febrero de 2014 de http://www.ietf.org/rfc/rfc4861.txt.

NS-2. *The Network Simulator – ns-2*. Recuperado el 1.º de febrero de 2014 de http://www.isi.edu/nsnam/ns/.

NS-3. *The ns-3 Project*. Recuperado el 1.º de febrero de 2014 de http://www.nsnam.org.

Opnet Technologies. *Application and network performance*. Recuperado el 28 de enero de 2014 de http://www.opnet.com.

Peterson, L. L. y Davie, B. S. (2011). *Computer Networks: A Systems Approach*. (5th Edition). Burlington, Massachusetts: Morgan Kaufmann.

Riley, G. F. y Henderson T. R. (2010). *Modeling and Tools for Network Simulation*. Berlin Heidelberg: Springer-Verlag.

Stevens, W. R. y Fall, K. R. (2011). *TCP/IP Illustrated, Volume 1: The Protocols*. (2nd Edition). New Jersey: Addison-Wesley Professional.

Thomson, S., Narten, T. y Jinmei, T. (2007). *IPv6 Stateless Address Autoconfiguration*, Internet Engineering

Task Force Request for Comments (RFC) 4862, Recuperado el 1.º de febrero de 2014 de http://www.ietf.org/rfc/rfc4862.txt.

Vincent, S., Montavont, J. y Montavont, N. (2008). Implementation of an IPv6 stack for ns-3, *2nd International Workshop on NS-2* (WNS2 2008).

Wang, A. y Jiang, W. (2009). Research of Teaching on Network Course Based on NS-3. *Education technology and computer Science*. First International Workshop on Wuhan, Hubei.