



## ANEXO B CONFIGURACIÓN SERVIDOR DE AUTOCONFIGURACIÓN Y PRUEBAS.

### B.1 CONFIGURACIÓN SERVIDOR DE AUTOCONFIGURACIÓN.

#### B.1.1 Configuración Stateless.

Dado que en este momento el enrutador IPv6 para la Universidad del Cauca es un equipo que cuenta con un sistema operativo Linux, concretamente Fedora Core 4, los anuncios de mensajes de enrutamiento se realizan mediante el demonio **radvd**, este demonio apropiadamente configurado permite la realización de la autoconfiguración sin control de estado (*stateless*).

La versión del paquete instalado en el equipo mencionado es la radvd-0.8-2.FC4. Este puede ser obtenido desde:

<http://rpm.pbone.net>  
<http://rpmfind.net>

La instalación de este paquete se realiza de la siguiente manera:

```
#rpm -i radvd-0.8-2.FC4.i386.rpm
```

Una vez el paquete ha sido configurado correctamente, el sistema generará un archivo llamado *radvd.conf.example* que tiene el aspecto mostrado en la figura B1.

Este es un ejemplo de un archivo de configuración, el cual describe un posible comportamiento de los anuncios de *router* en una interface específica.

Este archivo contiene una o más definiciones de interfaces, de prefijos o de *routers*.

El formato de las interfaces es el siguiente:

```
interface <Nombre de la interface> {  
    Lista de opciones específicas de la interfaz  
    Lista de definiciones del prefijo  
    Lista de definiciones del router  
};
```

Los prefijos deben ser definidos de la siguiente manera:

```
prefix prefijo/tamaño {  
    lista de opciones específicas al prefijo.  
};
```

Los *routers* deben ser definidos así:

```
route prefijo/tamaño {  
    Lista de opciones específicas al router. };
```



Donde el prefijo debe ser el mismo con el cual esta trabajando la red.

Cada opción debe ser terminada con un punto y coma (;).

```
interface eth0
{
    AdvSendAdvert on;
    # IgnoreIfMissing on;
    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 10;
    AdvHomeAgentFlag off;

    prefix 2001:db8:1:0::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr off;
    };

    route 2001:db0:fff::/48
    {
        AdvRoutePreference high;
        AdvRouteLifetime 3600;
    };

    prefix 0:0:0:1234::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr off;
        Base6to4Interface ppp0;
        AdvPreferredLifetime 120;
        AdvValidLifetime 300;
    };
};
```

Figura B1. Radvd.conf.example

#### B.1.1.1 Opciones específicas de la interfaz.

**IgnoreIfMissing (on|off):** Indica si debe o no ignorarse esta interfaz cuando no existe. Esta opción es usada cuando se utilizan interfaces dinámicas, las cuales no están activadas cuando radvd es inicializado, pero sin embargo han sido configuradas.

**MaxRtrAdvInterval (segundos):** El tiempo máximo permitido, en segundos, entre envíos de anuncios de *router* multicast no solicitados desde una interfaz.

Este tiempo no debe ser menor a 4 segundos ni mayor a 1800 segundos. Cuando se usa la extensión Mobile IPv6 el mínimo valor es 0.07 segundos. El valor por defecto es 600 segundos.

**MinRtrAdvInterval (segundos):** El tiempo mínimo permitido, en segundos, entre envíos



de anuncios de *router* multicast no solicitados desde una interfaz.

Este tiempo no debe ser menor a 3 segundos ni mayor a  $0.75 * \text{MaxRtrAdvInterval}$  segundos. Cuando se usa la extensión *Mobile IPv6* el mínimo valor es 0.03 segundos. El valor por defecto es  $0.33 * \text{MaxRtrAdvInterval}$  segundos.

**AdvHomeAgentFlag (on|off):** Cuando es activada esta opción, se indica que el *router* enviante, esta habilitado para servir como *Mobile IPv6 Home Agent*<sup>1</sup>. Cuando esta activo, son usados los límites mínimos para IPv6 Mobile en las opciones *MinRtrAdvInterval* y *MaxRtrAdvInterval*.

**AdvManagedFlag (on|off):** Al activarse, los *host* deben usar la autoconfiguración stateful, para obtener direcciones, además de cualquier dirección ya configurada usando stateless. Por defecto su valor es *off*.

**AdvOtherConfigFlag (on|off):** Al activarse los *host* usarán autoconfiguración stateful para información adicional de configuración (no direcciones). Por defecto su valor es *off*.

#### B.1.1.2 Lista de definiciones específicas del prefijo.

**AdvOnLink (on|off):** Cuando se encuentra activo indica que este prefijo puede ser usado para la determinación *on-link*. Cuando su valor es *off* los anuncios de *router* no hacen declaraciones acerca de las propiedades *on-link* u *off-link* del prefijo. Por ejemplo, el prefijo puede ser usado para la configuración de direcciones, con algunas de las direcciones pertenecientes al prefijo en *on-link* y otras en *off-link*.

**AdvAutonomous (on|off):** Cuando está activa, indica que el prefijo anunciado puede ser utilizado para la generación autónoma de direcciones (stateless).

**AdvRouterAddr (on|off):** Al activarse, indica que la dirección de la interfaz se envía en lugar del prefijo de Red, como es requerido por *Mobile IPv6*. Cuando toma el valor **on**, los límites específicos para *Mobile IPv6* son usados en las opciones *MinRtrAdvInterval* y *MaxRtrAdvInterval*.

**AdvValidLifetime (segundos | infinity):** Tiempo de vida válido, en segundos, que las direcciones generadas con este prefijo deben tomar. Por defecto esta opción toma el valor **infinity** (infinito = 0xffffffff).

**AdvPreferredLifetime (segundos|infinity):** Tiempo en segundos que una dirección generada con este prefijo permanecerá preferida. Por defecto 604800 segundos (7 días).

#### B.1.1.3 Lista de definiciones específicas del router.

**AdvRouteLifetime (segundos|infinity):** El tiempo de vida en segundos asociado a una ruta. Por defecto  $3 * \text{MaxRtrAdvInterval}$ .

**AdvRoutePreference (low|medium|high):** La preferencia asociada a este *router*, por defecto "*medium*".

---

<sup>1</sup> Un encaminador de la red propia que gestiona la localización del *host* móvil.



## B.1.2 Configuración Stateful

Las dos implementaciones con las cuales se realizaron las pruebas de autoconfiguración con control de estado fueron Dnsmasq y Sourceforce DHCPv6, debido a que cumplen con todos los estándares, están plenamente disponibles para uso libre, y cuentan con una amplia documentación.

El sistema operativo sobre el cual corrieron estas aplicaciones como Servidores DHCPv6 fue Fedora Core 4.

### B.1.2.1 Configuración Sourceforce DHCPv6.

El paquete utilizado para las pruebas realizadas fue el **dhcpv6-0.10-13**, el cual puede ser obtenido desde:

<http://rpmfind.net/linux/RPM/fedora/4/i386/dhcpv6-0.10-13.i386.html>  
<http://rpm.pbone.net>

O también esta disponible en los CD's de instalación de la distribución linux Fedora Core 4. Este paquete requiere de las siguientes librerías para su correcto funcionamiento: libcrypto.so.5, libresolv.so.2.

Una vez el paquete haya sido obtenido, ejecutamos el comando:

```
[root@zeus ~]# rpm -i dhcpv6-0.10.13
```

Cuando la operación haya terminado exitosamente se generaran unos archivos de configuración en `/usr/share/doc/dhcpv6-10/`, que ejemplifican una configuración estándar, estos archivos controlan el comportamiento de cliente, servidor y agente relay.

#### B.1.2.1.1 Archivo de Configuración del Servidor.

El archivo de configuración del servidor dhcpv6 es `dhcp6s.conf`, y debe ser copiado en `/etc/`, para posteriormente ser editado a conveniencia del sistema donde funcionará. El aspecto del mencionado archivo que se genera en la instalación es el mostrado en la figura B2.

`dhcp6s.conf` contiene la información de configuración de direcciones, prefijos y parámetros de configuración de red para cada interfaz del servidor. El orden específico de declaración es: Interfaz, *link*, *host*. Dentro de un *link* puede haber una declaración más específica llamada pool.

Cada declaración y sus respectivas expresiones, deben ser terminadas con punto y coma (;), y mediante el signo numeral (#) se hacen comentarios dentro del archivo.



```
interface eth0 {
    server-preference 255;
    renew-time 60;
    rebind-time 90;
    prefer-life-time 130;
    valid-life-time 200;
    allow rapid-commit;
    option dns_servers 2000::2 ibm.com;
    link AAA {
        pool{
            range 3ffe:501:ffff:0::1 to 3ffe:501:ffff:0::10/64;
            prefix 3ffe:501:ffff:0001::/48;
        };
        # range 3ffe:501:ffff:1::1 to 3ffe:501:ffff:1::10/64;
    };
    host BBB {
        duid 00:01:00:01:05:c8:8c:7e:00:10:a4:8d:30:7f;
        iaidinfo {
            iaid 8335501;
            renew-time 30;
            rebind-time 40;
        };
        prefix {
            fecc::/24;
            prefer-life-time 90;
            valid-life-time 120;
        };
        prefix {
            3000::/64;
            prefer-life-time 100;
            valid-life-time 120;
        };
        address {
            fecc::1/34;
            prefer-life-time 60;
            valid-life-time 90;
        };
    };
};
```

Figura B2. Dhcp6s.conf

Declaraciones:

**interface <nombre de la interfaz>**

```
{
    [declaraciones];
};
```

La declaración de interface es usada para informarle al servidor DHCPv6 que los *links* y *hosts* declarados dentro de esta, están conectados al mismo segmento de red. Los parámetros de red comunes a este segmento pueden ser definidos dentro de la declaración de interfaz.

**link <nombre del link>**

```
{
    [declaraciones];
};
```

Esta declaración permite proveer al servidor DHCPv6 los rangos de direcciones, prefijos y *hosts*. Los parámetros de red comunes a este link, pueden ser declarados dentro de ésta.



```
group  
{  
  [declaraciones];  
};
```

Esta declaración es usada para agrupar expresiones para definir parámetros de configuración de red comunes, los cuales solo son válidos dentro de la declaración. Grup{} puede ser uno de los siguientes tipos: interface, link, pool, o host.

```
pool  
{  
  [declaraciones];  
};
```

Esta instrucción es usada para declarar un banco de direcciones IPv6, desde el cual puedan ser obtenidas. Dentro del pool{}, se pueden hacer declaraciones propietarias de configuraciones relacionada con parámetros de red.

```
host <nombre del host>  
{  
  [duid <DUID>];  
  [iaidinfo {}];  
  [address {}];  
  [prefix <prefijo>/<tamaño del prefijo>];  
  [declaraciones];  
};
```

Esta declaración permite a los administradores describir las direcciones, prefijos, y parámetros de configuración de red, específicos a un host. Para esta declaración deben ser especificados un identificador único DHCP (DUID), un identificador de identidad de asociación (IAID), y una dirección. Esta declaración es usada para asignación estática de direcciones para un cliente DHCP en particular.

```
address  
{  
  [<direccion IPv6>/<tamaño del prefijo>];  
  [prefer-life-time <tiempo de vida preferido>];  
  [valid-life-time <tiempo de vida válido>];  
};
```

Esta declaración permite especificar direcciones, tiempo de vida preferido, tiempo de vida válido, para un host. Esta es válida únicamente dentro de una declaración host{}.

```
prefix <prefijo>/<tamaño del prefijo>;
```

Mediante esta declaración se especifica un prefijo.



## **iaidinfo**

```
{  
  [iaid <IAID number>];  
  [renew-time <tiempo renew>];  
  [rebind-time <tiempo rebind>];  
};
```

iaidinfo permite describir la información para un IAID específico. También opcionalmente se pueden definir tiempos para los mensajes *renew* y *rebind*. Esta declaración es válida solamente dentro de *host{}*.

## **range <direccion IPv6 > to <direccion IPv6 >/<tamaño prefijo>;**

Esta expresión permite anunciar un rango y el tamaño del prefijo. En cada rango anunciado no más de una dirección, no otorgada, será asignada ante la petición de un cliente.

## **option dns\_server <direcciones IPv6 dominio>;**

Esta declaración permite anunciar un servidor DNS a ser configurado en los potenciales clientes.

## **send <OPTION>;**

Esta declaración habilita el envío de opciones específicas en los mensajes DHCPv6 enviados por parte del servidor:

### **send unicast;**

Habilita el envío de opciones servidor unicast. Usualmente los clientes utilizan direcciones multicast para contactarse con servidores DHCP y agentes *Relay*. Esta declaración especifica que dhcp6s aceptará mensajes unicast desde los clientes DHCPv6 si estos incluyen opciones servidor unicast.

### **send server-preference <Valor de preferencia>;**

Esta declaración incluye un valor de preferencia para los mensajes DHCPv6 enviados. El cliente optará por los mensajes con mayor valor de preferencia. El máximo valor que puede tomar esta opción es 255.

### **send temp-address;**

Esta declaración asigna una dirección temporal a una petición por parte de un cliente.

### **send information-only;**

Esta opción habilita a dhcp6s a enviar únicamente información de configuración, como por ejemplo información referente a servidores DNS.

## **allow <OPTION>;**

Esta declaración permite al cliente DHCP especificar una opción en los mensajes recibidos.



### **allow rapid-commit;**

Esta opción posibilita un intercambio de 2 mensajes entre cliente y servidor. [20]

#### **B.1.2.1.2 Archivo de Configuración del Cliente.**

El archivo de configuración del cliente DHCPv6 es *dhcp6c.conf*, debe estar ubicado en */etc/* por defecto. El archivo *dhcp6c.conf* es generado en la instalación de la aplicación y sirve como guía para realizar configuraciones particulares a cada sistema, su aspecto es el siguiente:

```
interface eth0 {  
#     information-only;  
    send rapid-commit;  
    request prefix-delegation;  
#  
    request temp-address;  
};
```

Figura B3. Archivo dhcp6c.conf

##### **B.1.2.1.2.1 Declaraciones**

Las declaraciones se deben realizar bajo el siguiente formato:

```
interface <nombre de la interfase>  
{  
    [Declaraciones]  
};
```

Las declaraciones pueden ser de los siguientes tipos:

#### **send [opciones dhcp];**

Por medio de *send*, *dhcp6c* envía opciones específicas, normalmente la opción enviada es *rapid-commit*, o sea la petición de autorización de establecimiento de comunicaciones dhcp con solo dos mensajes.

#### **request [opciones dhcp];**

Por medio de *request*, *dhcp6c* envía peticiones específicas al servidor DHCP6, generalmente las opciones definidas son: *temp-address*, *domain-name-servers*, y *prefix-delegation*.

*request domain-name-servers;*

Esta declaración permite a *dhcp6c* hacer peticiones de un listado de Servidores para ser configurados por ejemplo servidores DNS, o NTP.

*request prefix-delegation;*

Mediante esta petición se solicita un prefijo al servidor DHCP6.

*request temp-address;*





Por defecto un cliente solicita direcciones no temporales, mediante esta opción el cliente solicitará direcciones temporales.

#### **information-only;**

Mediante esta opción un *host* solicitará información de configuración, exceptuando direcciones IPv6.

```
address {  
  [<IPv6 address>/<prefix length>;  
  [prefer-life-time <preferred-lifetime>;  
  [valid-life-time <valid-lifetime>;  
};
```

Mediante la expresión *address* se define un conjunto de direcciones preferidas para un cliente DHCPv6, dentro de esta declaración se pueden declarar a su vez un tiempo de vida preferido y un tiempo de vida válido para dichas direcciones.

```
prefix {  
  [<IPv6 prefix>/<prefix length>;  
  [prefer-life-time <preferred-lifetime>;  
  [valid-life-time <valid-lifetime>;  
};
```

Esta expresión define un prefijo preferido para un cliente DHCPv6, dentro de esta declaración se pueden declarar a su vez un tiempo de vida preferido y un tiempo de vida válido para dicho prefijo.

#### **renew-time <tiempo en segundos>;**

Tiempo de renovación de direcciones, con el servidor que las entregó originalmente (T1).

#### **rebind-time <rebind-time>;**

Tiempo de renovación de direcciones, con cualquier servidor (T2).

### **B.1.2.2 Configuración Dnsmasq.**

Cliente, servidor y agente *Relay* deben ser instalados de la misma manera. Sin embargo dado que esta es una versión portable se debe realizar una diferenciación entre la instalación en sistemas Linux y Windows. En cualquier caso los instaladores pueden ser obtenidos de:

<http://klub.com.pl/dhcpv6/>

#### **B.1.2.2.1 Instalación Linux.**

Una vez el paquete `dnsmasq-0.4.1-1.i386.rpm` ha sido obtenido procedemos a instalarlo de la siguiente manera:



```
#rpm -i dibbler-0.4.1-1.i386.rpm
```

Si se desea instalar Dibbler desde las fuentes, se debe descargar el paquete con extensión tar.gz. , extraerlo, y luego realizar la instalación por medio del comando make install. Por ejemplo al instalar servidor y agente Relay:

```
tar zxvf dibbler-0.1.0-src.tar.gz  
make server relay  
make install
```

#### **B.1.2.2.2 Instalación Windows.**

A partir de la versión 0.2.1 Dibbler tiene soporte para Windows XP y 2003. En la versión 0.4.1 se cuenta con una versión experimental para Windows NT y 2000. Una vez descargada la versión adecuada al sistema en el cual será instalada, hacer doble click sobre el instalador. Y seguir las instrucciones que aparecerán en pantalla. Al ser instalado Dibbler estará disponible en el menú Inicio.

#### **B.1.2.2.3 Archivos de Configuración.**

Los archivos de configuración para sistemas Linux deben estar ubicados en un el directorio */etc/dibbler/*, o en el directorio de instalación en sistemas Windows. Dentro de estos archivos de configuración todos los tipos de comentarios son permitidos:

- Estilo de comentarios C++ : // Esto es un comentario
- Multilínea comentada estilo C: /\* Esta es una multilínea comentada\*/
- Comentarios con numeral: # Comentario

En este archivo existen cuatro tipos de alcances o dominios de acción para las opciones: global, interface, IA y direcciones. El alcance global es el mayor, este cubre la totalidad del archivo de configuración y aplica a todas las interfaces, IA's, y direcciones, a menos que una opción con alcance más bajo lo sobrescriba. Después se presenta el alcance interface, donde las opciones definidas son válidas para dicha interface, todas las IA's y direcciones dentro de esta. Luego se tiene el alcance IA, las opciones ahí definidas aplican a un IA específico y a un grupo de direcciones pertenecientes a dicho IA. El alcance menos significativo es direcciones.

Una opción puede ser usada en más de un alcance, por ejemplo, un T1 Puede ser usado por los alcances global o interface. En el caso interface esto significaría: este es el T1 específico para la interfaz XX. Las opciones pueden ser usadas en diferentes tiempos, en este caso el último valor definido es usado.

#### **B.1.2.2.4 Archivo de configuración del Servidor.**

El archivo de configuración del servidor Dibbler es el *server.conf*. Una vez el servicio es inicializado las versiones antiguas de este archivo son almacenadas en *server.conf-old*.



#### B.1.2.2.4.1 Alcance Global.

Toda opción puede ser declarada con un alcance global de la siguiente manera:

**Declaración de interfaz, o, opciones globales, o, opciones interfaz**

**Opciones class**

#### B.1.2.2.4.2 Alcance Interface

Una declaración de interface puede tener los siguientes formatos:

**Nombre de la interfaz**

```
{  
    Opciones interface, o, opciones class  
}
```

O

**Numero de interface**

```
{  
    Opciones interface, o, opciones class  
}
```

#### B.1.2.2.4.3 Alcance class (IA)

Este alcance puede ser declarado así:

**Class**

```
{  
    Class, o, address pool  
}
```

address pool puede ser definido de cualquiera de las siguientes formas:

**Pool dirección mínima – dirección máxima**  
**Pool dirección/prefijo**

#### B.1.2.2.4.3 Opciones

La tabla B1 resume las opciones configurables en *server.conf*:

Tabla B1. Opciones del servidor Dibbler Dhcpv6

<b>Nombre</b>	<b>Alcance</b>	<b>Tipo Valor</b>	<b>Valor por defecto</b>	<b>Descripción</b>
work-dir	global	String	vacío	Directorio de trabajo.



<b>Nombre</b>	<b>Alcance</b>	<b>Tipo Valor</b>	<b>Valor por defecto</b>	<b>Descripción</b>
Log-level	global	1-8	8	Nivel de los logs (8 el mas prolijo).
Log-name	global	String	Client	Nombre que aparecerá en el archivo log.
Log-mode	global	Corto o entero	full	Modo del log. Tipo corto o full.
Preference	interface	0-255	0	Valor de preferencia del servidor.
Unicast	interface	Dirección	vacío	Especifica que dirección debe ser usada.
iface-max-lease	interface	Entero	4294967296	Número máximo de direcciones entregadas a todos los clientes.
lient-max-lease	interface	Entero	4294967296	Número máximo de direcciones entregadas a un cliente.
rapid-commit	interface	0, o, 1	0	Permitir intercambio de dos mensajes.
Relay	interface	String	No definido	Nombre de la interfaz física usada para alcanzar el agente <i>Relay</i> .
interface-id	interface	Entero	No definido	ID de la interfaz <i>relay</i> . Debe ser UNICA.
valid-lifetime	class	Entero	4294967296	Tiempo válido para las direcciones (especificado en segundos).
prefered-lifetime	class	Entero	4294967296	Después de este tiempo (en segundos) la dirección será despreciada.
T1	class	Entero	4294967296	Tiempo de renovación de direcciones, con el servidor que las



<b>Nombre</b>	<b>Alcance</b>	<b>Tipo Valor</b>	<b>Valor por defecto</b>	<b>Descripción</b>
				entrego originalmente.
T2	class	Entero	4294967296	Tiempo de renovación de direcciones, con cualquier servidor.
reject-clients	class	Direcciones o lista de DUID's	vacío	Lista de clientes descartados, para esta configuración de esta IA.
Accept-only	class	Direcciones o lista de DUID's	vacío	Clientes con permiso para usar esta class.
class-max-lease	class	entero	4294967296	Número máximo de direcciones asignadas a esta clase.

#### **B.1.2.2.4.3 Opciones Adicionales**

El servidor soporta opciones no especificadas en el RFC 3315, estas presentan el siguiente formato genérico:

##### **option Nombre-Opción Valor-Opción**

La tabla B2 presenta, todas las posibles opciones configurables:

Tabla B2. Opciones Adicionales Servidor Dibbler Dhcpv6

<b>Nombre de la opción</b>	<b>Tipo de Valor</b>	<b>Valor por defecto</b>	<b>Descripción</b>
dns-server	Lista de direcciones	vacío	Lista de servidores DNS.
Domain	Lista de cadenas de caracteres	vacío	Lista de nombres de Dominio.
ntp-server	Lista de direcciones	vacío	Lista de servidores NTP.
time-zone	Valor de tiempo	vacío	Valor de tiempo.
sip-server	Lista de direcciones	vacío	Lista de servidores SIP.



<b>Nombre de la opción</b>	<b>Tipo de Valor</b>	<b>Valor por defecto</b>	<b>Descripción</b>
sip-domain	Lista de cadenas de caracteres	vacío	Lista de nombres de Dominio.
nis-server	Lista de direcciones	vacío	Lista de servidores NIS.
nis-domain	Cadena de caracteres	vacío	Nombre de dominio.
Nisplus-server	Lista de direcciones	vacío	Lista de servidores NIS+
nis-domain	Cadena de caracteres	vacío	Nombre de dominio.
Lifetime *	Entero	vacío	Tiempo de renovación de opciones.

\* El *lifetime*, indica a los clientes por cuanto tiempo la información obtenida es válida.

#### **B.1.2.2.5 Archivo de configuración del Cliente.**

Este archivo lleva por nombre *client.conf*. Después de la inicialización de esta aplicación las versiones antiguas de su archivo de configuración serán almacenadas en *client.conf-old*.

##### **B.1.2.2.5.1 Alcance Global.**

Una opción puede ser declarada con un alcance global de la siguiente forma:

Declaración de interface  
Opciones globales  
Opciones de interface  
Opciones IA  
Opciones de direcciones.

##### **B.1.2.2.5.1 Declaración de interface**

La declaración de interface puede hacerse de una de las dos siguientes maneras:

```
iface <nombre de la interface>
{
    Opciones de la interface
    Opciones IA
    Opciones de de direcciones
}
o
```

```
iface <numero de la interface>
{
    Opciones de la interface
```



## Opciones IA Opciones de direcciones

}

### B.1.2.2.5.2 Declaración IA

Las IA necesarias para el funcionamiento del DHCPv6, se declaran de la siguiente manera:

IA <número>

{

Declaración de direcciones  
Opciones IA  
Opciones de direcciones

}

### B.1.2.2.5.3 Declaración de direcciones

La declaración de direcciones se realiza de la siguiente manera:

Número de direcciones

{

Opciones de direcciones  
Direcciones IPv6

}

Donde número de direcciones denota el número que serán requeridas.

### B.1.2.2.5.4 Opciones estándar

La tabla B3 ilustra las opciones que el estándar RFC 3315, propone para una implementación cliente DHCPv6.

Tabla B3. Opciones cliente Dibbler Dhcpv6.

<b>Nombre</b>	<b>Alcance</b>	<b>Tipo de Valor</b>	<b>Valor por defecto</b>	<b>Descripción</b>
valid-lifetime	direcciones	Enteros	4294967296	Tiempo de vida válido para las direcciones (especificado en segundos).
Prefered-lifetime	direcciones	Enteros	4294967296	Después de este tiempo (en segundos) la dirección será despreciada.
T1	IA	Enteros	4294967296	Tiempo de renovación de direcciones, con el servidor que las entrego originalmente.



<b>Nombre</b>	<b>Alcance</b>	<b>Tipo de Valor</b>	<b>Valor por defecto</b>	<b>Descripción</b>
T2	IA	Enteros	4294967296	Tiempo de renovación de direcciones, con cualquier servidor.
Reject-servers	IA	Direcciones o lista de DUID's	vacio	Lista de Servidores que deberían ser descartados en una configuración IA.
Prefered-servers	IA	Direcciones o lista de DUID's	vacío	Lista de Servidores preferidos para una configuración IA.
rapid-commit	Interface	0, o, 1	0	Solicitud de permiso para intercambio de dos mensajes.
Unicast	Interface	0, o, 1	0	El cliente pregunta si la Comunicación unicast es permitida.
work-dir	Global	Cadena de caracteres	vacío	Directorio de trabajo.
log-level	Global	1-8	8	Nivel de los logs (8 el mas prolijo).
log-name	Global	Cadena de caracteres	Client	Nombre que aparecerá como nombre del archivo log.
log-mode	Global	Corto o entero	full	Modo del log. Tipo corto o full.

## B.2 PRUEBAS SERVIDOR STATELESS

El archivo de configuración del demonio **radvd**, no debe ser modificado para su uso sobre clientes Linux o Windows, ya que como se expuso en el capítulo 3 el cliente solo requiere el prefijo anunciado por el *router* y una dirección hardware para formar una dirección, por tanto la configuración en nodos cliente es nula. En la figura B4 se puede apreciar el archivo utilizado para las pruebas realizadas.





```
interface eth0
{
    AdvSendAdvert on ;
    MinRtrAdvInterval 30 ;
    MaxRtrAdvInterval 100 ;
    AdvManagedFlag off ;
    AdvOtherConfigFlag off ;

    prefix 2001:448:1024:1::/64
    {
        AdvOnLink on ;
        AdvAutonomous on ;
        AdvRouterAddr off;
        AdvValidLifetime 2592001 ;
        AdvPreferredLifetime 604800 ;
    };
};
```

Figura B4. radvd.conf

Este archivo anuncia el prefijo 2001:448:1024:1::/64, además mediante valor *on* en la bandera *AdvAutonomous* se autoriza a los clientes a realizar autoconfiguración *stateless*.

Con permisos de administrador se ejecuta el siguiente comando para iniciar el envío de mensajes de *router*:

```
#radvd -u radvd
```

**prueba 1** : Configuración de un cliente Linux.

Sobre un *host* con soporte IPV6, se procede a verificar si la autoconfiguración *stateless* tiene efecto, ejecutando el comando */sbin/ifconfig*

```
eth0      Link encap:Ethernet  HWaddr 00:12:3F:6F:9D:5B
          inet addr:172.16.30.117  Bcast:172.16.255.255  Mask:255.255.0.0
          inet6 addr: 2001:448:1024:1::15/128 Scope:Global
          inet6 addr: fe80::212:3fff:fe6f:9d5b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8885 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6794 errors:0 dropped:0 overruns:0 carrier:0
          collisions:5 txqueuelen:1000
          RX bytes:1337898 (1.2 MiB)  TX bytes:698713 (682.3 KiB)
```

Figura B5. Configuración *stateless* sistemas Linux

**Prueba 2** : Configuración de un cliente Windows.

Para que este tipo de configuración tenga efecto, el cliente debe tener habilitado el soporte para IPv6. En una ventana de DOS, ejecutamos el comando *ipconfig*, el resultado se puede observar en la figura B6.



```
Adaptador Ethernet Conexión de área local :
Sufijo de conexión específica DNS :
Dirección IP. . . . . : 172.16.30.112
Máscara de subred . . . . . : 255.255.0.0
Dirección IP. . . . . : 2001:448:1024:1:142:42a5:afa4:eb07
Dirección IP. . . . . : fe80::212:3fff:fe6f:a466%4
Puerta de enlace predeterminada : 172.16.255.254
                                fe80::203:ceff:fe89:2fc2%4
```

Figura B6 Configuración stateless en sistemas Windows.

Se puede observar que en los dos escenarios planteados la autoconfiguración fué exitosa.

### B.3 PRUEBAS SERVIDOR STATEFUL.

El objetivo que se persigue en estas pruebas es lograr la interacción entre clientes y servidor DHCP, este último deberá asignar direcciones e información de configuración relevante a los *hosts* cliente. Por otro lado el Servidor DHCP no entregará información de enrutamiento, por tanto el demonio *radvd* debe mantenerse en funcionamiento, con la salvedad de que la bandera *AdvAutonomus* debe tomar el valor **off**, para lograr un stateful puro.

#### Prueba 1 – Servidor Sourceforce , cliente Sourceforce, (linux) :

El archivo de configuración del servidor DHCPv6 Sourceforce es el siguiente:

```
interface eth0{
    server-preference 255 ;
    renew-time 60 ;
    rebind-time 90 ;
    prefer-life-time 120 ;
    valid-life-time 200 ;
    allow rapid-commit ;
    option dns_servers 2001:448:1024:1::10 ipv6.unicauca.edu.co ;
    link AAA {
        pool{
            range 2001:448:1024:1::11 to 2001:448:1024:1::20/64 ;
        };
    };
};
```

Figura B7. dhcp6s.conf

Mediante el comando `#dhcp6s -dDf eth0 -c /etc/dhcp6s.conf`, el servidor se pondrá en marcha, y si no se detectan problemas, se desplegara en pantalla lo expuesto en la figura B8:

```
[root@zeus ~]# dhcp6s -dDf eth0 -c /etc/dhcp6s.conf
Sep/14/2006 15:20:10 extracted an existing DUID from /var/lib/dhcpv6/dhcp6s_duid: 00:01:00:01
Sep/14/2006 15:20:10 set timer for syncing file ...
Sep/14/2006 15:20:10 add domain name zeus.ipv6.unicauca.edu.co
Sep/14/2006 15:20:10 add domain name ipv6.unicauca.edu.co
Sep/14/2006 15:20:10 interface definition for eth0 is ok
```

Figura B8. Inicialización servidor DHCPv6 Sourceforce

En este momento el servidor esta listo para suministrar parámetros de configuración de



red. Cabe anotar que toda la información IPv6 que el servidor entregue a los clientes, será almacenada en el archivo `/var/lib/dhcpv6/server6.leases`.

Por otro lado el cliente DHCPv6, tiene el archivo de configuración mostrado en la figura B9:

```
interface eth0 {
#       information-only;
       send rapid-commit;
       request domain-name-servers;
};
```

Figura B9. dhcp6c.conf

Para iniciar el cliente DHCPv6 sourceforce, se debe ingresar el comando:  
`# dhcp6c -dDf eth0`.

```
Sep/14/2006 15:35:18 get DHCP option status code, len 2
Sep/14/2006 15:35:18   this message status code: success
Sep/14/2006 15:35:18 reply message XID is (e4ebd6)
Sep/14/2006 15:35:18 ifp 0x9849008 event 0x984d2d8 id is e4ebd6
Sep/14/2006 15:35:18 serverID is 00:01:00:01:0b:c6:4b:c9:00:03:ce:89:2f:c2 len i
s 14
Sep/14/2006 15:35:18 status code: success
Sep/14/2006 15:35:18 assigned address 2001:448:1024:1::1 is not in any RAs prefi
x length using 64 bit instead
Sep/14/2006 15:35:18 get address is 2001:448:1024:1::1/64
Sep/14/2006 15:35:18 lease address is 2001:448:1024:1::1/64
Sep/14/2006 15:35:18 renew time 60, rebind time 90
Sep/14/2006 15:35:18 removing an event 0x984d2d8 on eth0, state=4, xid=e4ebd6
Sep/14/2006 15:35:18 DUID is 00:01:00:01:0b:c6:4b:c9:00:03:ce:89:2f:c2, DUID_LEN
is 14
Sep/14/2006 15:35:18 removing ID (ID: 00:01:00:01:0b:c6:4b:c9:00:03:ce:89:2f:c2)
Sep/14/2006 15:35:18 got an expected reply, sleeping.
```

Figura B10. Recepción de información de configuración de un cliente Sourceforce.

Si el código de estado es SUCCESS como se puede observar en la figura anterior, entonces la configuración de este *host* es exitosa, situación que se puede verificar al ejecutar el comando `/sbin/ifconfig` para observa las direcciones configuradas, como se aprecia en la figura B11. En la figura B12, se observa el archivo `resolv.conf`, donde se escribirá información complementaria: DNS y Dominio.



```
eth0      Link encap:Ethernet  HWaddr 00:12:3F:6F:9D:5B
          inet addr:172.16.30.117  Bcast:172.16.255.255  Mask:255.255.0.0
          inet6 addr: 2001:448:1024:1::1/64 Scope:Global
          inet6 addr: fe80::212:3fff:fe6f:9d5b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3566 errors:0 dropped:0 overruns:0 frame:0
          TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:298236 (291.2 KiB)  TX bytes:3001 (2.9 KiB)
          Interrupt:11

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:1358 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1358 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1869195 (1.7 MiB)  TX bytes:1869195 (1.7 MiB)
```

Figura B11. Asignación de direcciones cliente Sourceforce Dhcpv6

```
nameserver 2001:448:1024:1::10
search ipv6.unicauca.edu.co
```

Figura B12. resolv.conf cliente Sourceforce Dhcpv6.

Finalmente para comprobar que con los parámetros asignados se tiene conectividad, se hace un ping6, a un sitio con soporte IPv6.

```
[root@atlas ~]# ping6 www6.tahi.org
PING www6.tahi.org(pickles.tahi.org) 56 data bytes
64 bytes from pickles.tahi.org: icmp_seq=0 ttl=51 time=351 ms
64 bytes from pickles.tahi.org: icmp_seq=1 ttl=51 time=350 ms
64 bytes from pickles.tahi.org: icmp_seq=2 ttl=51 time=394 ms
64 bytes from pickles.tahi.org: icmp_seq=3 ttl=51 time=367 ms
```

Figura B13. Prueba de conectividad.

## Prueba 2 – Servidor Sourceforce, cliente Dibbler (Windows y Linux):

En este caso el archivo de configuración para el servidor Sourceforce, es el mismo que el utilizado en la prueba anterior, figura B7.

### Prueba 2.1 – Servidor Sourceforce, cliente Dibbler Windows:

El archivo de configuración del cliente Dibbler puede observarse en la figura B14.

```
#log-mode short

# 7 = omit debug messages
#log-level 7

iface "Conexión de área local" {
    option dns-server
    option domain
    ia
}
```

Figura B14. Archivo de configuración cliente Dibbler Windows.



Al iniciar el cliente se puede observar la secuencia de intercambio de mensajes DHCP, mediante el cual se asigna al *host* cliente una dirección dentro del rango predefinido, lo cual se puede verificar, al ejecutar el comando *ipconfig*, en una ventana DOS, figura B15.

```

Adaptador Ethernet Conexión de área local          :
Sufijo de conexión específica DNS                 :
Dirección IP. . . . .                             : 172.16.30.117
Máscara de subred . . . . .                       : 255.255.0.0
Dirección IP. . . . .                             : 2001:448:1024:1::12
Dirección IP. . . . .                             : fe80::212:3fff:fe6f:9d5b%6
Puerta de enlace predeterminada                   : 10.200.2.254
                                                    fe80::203:ceff:fe89:2fc2%6
  
```

Figura B15. Configuración de una dirección IPv6 en un cliente Dibbler.

Sin embargo en la figura B16, se aprecia como la asignación de otra información de configuración (servidor DNS, Dominio) no se realiza, debido a que Dibbler no reconoce el formato de las opciones enviadas por el Servidor Sourceforce.

```

11:54:05 Debug      Received 97 bytes on interface Conexión de área local/6 (socket=1712, addr=fe80::203:ceff:fe89:2fc2.).
11:54:05 Warning    Unknown option: 25, length=16. Ignored.
11:54:05 Warning    Unknown option: 26, length=22. Ignored.
11:54:05 Notice    Received REPLY on Conexión de área local/6,TransID=0x5796, 4
opts: 1 2 7 13
11:54:05 Notice    Status SUCCESS :
11:54:05 Notice    Not all options were assigned (23 24 ). Sending new INFORMATION-REQUEST.
11:54:05 Debug      Creating INF-REQUEST on the Conexión de área local/6.
11:54:05 Info       Processing msg (INF-REQUEST,transID=0x35ff,opts: 1 8 6)
11:54:05 Debug      Sending INF-REQUEST on Conexión de área local/6 to multicast.
  
```

Figura B16. Secuencia de mensajes DHCPv6.

En la figura B17, se observa que dicha información de configuración, efectivamente no fue configurada:

```

Servidores DNS . . . . . : fec0:0:0:fff::1%1
                          fec0:0:0:fff::2%1
                          fec0:0:0:fff::3%1
  
```

Figura B17. Configuración Servidor DNS cliente Dibbler Windows.

Por tanto como resultado de las pruebas realizadas, se concluye que hay interoperabilidad parcial entre un servidor DHCPv6 Sourceforce y un cliente Dibbler Windows.

**Prueba 2.2 – Servidor Sourceforce, cliente Dibbler Linux:**

La figura B18, ilustra el archivo de configuración del cliente Dibbler Linux.



```
# 7 = omit debug messages
#log-level 7

iface eth0 {
    option dns-server
    option domain
    ia
}
```

Figura B18. Archivo de configuración cliente Dibbler Linux.

El comando de ejecución del cliente Dibbler Linux es: `/usr/sbin/dibbler-client run`. El resultado de la interacción fue idéntico al que se presentó en la prueba previa, una dirección IPv6 es entregada al cliente Dibbler, pero la información de configuración adicional no. Figuras B19 y B20.

```
eth0    Link encap:Ethernet  HWaddr 00:12:3F:6F:9D:5B
        inet addr:172.16.30.117  Bcast:172.16.255.255  Mask:255.255.0.0
        inet6 addr: 2001:448:1024:1::15/128 Scope:Global
        inet6 addr: fe80::212:3fff:fe6f:9d5b/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:8885 errors:0 dropped:0 overruns:0 frame:0
        TX packets:6794 errors:0 dropped:0 overruns:0 carrier:0
        collisions:5 txqueuelen:1000
        RX bytes:1337898 (1.2 MiB)  TX bytes:698713 (682.3 KiB)
        Interrupt:11
```

Figura B19. Configuración de una dirección IPv6 en un cliente Dibbler linux.

```
37:19 Debug   Received 97 bytes on interface eth0/2 (socket=6, addr=fe80::203:
ceff:fe89:2fc2.).
37:19 Warning Unknown option: 25, length=16. Ignored.
37:19 Warning Unknown option: 26, length=22. Ignored.
37:19 Notice  Received REPLY on eth0/2,TransID=0x9160f1, 4 opts: 1 2 7 13
37:19 Notice  Status SUCCESS :
37:19 Notice  Not all options were assigned (23 24 ). Sending new INFORMATION-
```

Figura 20. Intento de configuración de información adicional en un cliente Dibbler Linux.

### Prueba 3: -- Servidor Dibbler, cliente Dibbler (Windows y Linux):

El comando de ejecución del servidor Dibbler en Linux es: `/usr/sbin/dibbler-server run` .en la figura B21, se puede observar el archivo de configuración del servidor Dibbler:



```
log-level 7
log-mode short

iface eth0 {
    T1 1800
    T2 2000
    preferred-lifetime 3600
    valid-lifetime 7200

    class {
        pool 2001:448:1024:1::/64
    }

    option dns-server 2001:448:1024:1::10
    option domain ipv6.unicauca.edu.co
}
```

Figura B21. Archivo de configuración servidor Dibbler.

### Prueba 3.1 – Servidor Dibbler, cliente Dibbler Windows:

El archivo de configuración del cliente Dibbler Windows es idéntico al expuesto en la prueba 2.1 Figura B14.

Para correr este cliente desde el menú *inicio/programas/Dibbler* hacer click sobre la opción *Client Run in the console*, o en una ventana DOS, ubicarse dentro del directorio de instalación de Dibbler y ejecutar el comando *dibbler-client.exe run*. Como resultado aparecerá en pantalla lo que se puede observar en la figura B22.

```
11:22:32 Debug      Sending REQUEST on Conexión de Brea local 2/5 to multicast.
11:22:32 Notice     Sleeping for 1 second(s).
11:22:32 Debug      Received 164 bytes on interface Conexión de Brea local 2/5 (socket=1752, addr=fe80::203:ceff:fe89:2fc2.).
11:22:32 Notice     Received REPLY on Conexión de Brea local 2/5,TransID=0x73e8, 5 opts: 1 3 2 23 24
11:22:32 Notice     Address 2001:448:1024:1:b7ab:ae87:ee46:5a0b added to Conexión de Brea local 2/5 interface.
11:22:32 Notice     Setting up DNS server 2001:448:1024:1::10 on interface Conexión de Brea local 2/5.
11:22:32 Notice     Setting up Domain ipv6.unicauca.edu.co on interface Conexión de Brea local 2/5.
11:22:33 Notice     Sleeping for 1 second(s).
11:22:34 Error      DAD inconclusive. Unable to dermine 2001:448:1024:1:b7ab:ae87:ee46:5a0b address state. Assuming NOT TENTATIVE.
11:22:34 Notice     Sleeping for 1798 second(s).
```

Figura B22. Comportamiento de un cliente Dibbler Windows, frente a un Servidor Dibbler.

Como se puede apreciar, se asignó, toda la información requerida por el cliente: dirección IPv6, dominio y servidor DNS.

Por otro lado en el archivo de almacenamiento de información de direcciones del servidor, ubicado en */var/lib/dibbler/server-AddrMgr.xml*, se pude comprobar la asignación de la dirección, figura B23.



```
<AddrClient>
  <duid length="14">0x00:01:00:06:45:0c:0b:11:00:12:3f:6f:a3:88</duid>
  <AddrIA unicast="fe80::212:3fff:fe6f:a388"
    T1="1800" T2="2000" IAID="1" state="NOTCONFIGURED" iface="2">
    <duid length="14">0x00:01:00:06:45:0c:0b:11:00:12:3f:6f:a3:88</duid>
    <AddrAddr timestamp="1158405461" pref="3600" valid="7200">2001:448:1024:1:b7ab:ae87:ee46:5a0b</AddrAddr>
  </AddrIA>
</AddrClient>
```

Figura B23. Archivo server-AddrMgr.xml

El resultado de la ejecución de ipconfig, se ilustra en la figura B24, en donde se observa todos los parámetros de configuración asignados:

```
Dirección IP. . . . . : 2001:448:1024:1:b7ab:ae87:ee46:5a0b
Dirección IP. . . . . : fe80::212:3fff:fe6f:a388%5
Puerta de enlace predeterminada . . : 10.200.2.254
                                       fe80::203:ceff:fe89:2fc2%5
Servidores DNS . . . . . : 2001:448:1024:1::10
```

Figura B24. Parámetros de configuración Adicional cliente Dibbler Windows.

Cabe resaltar que aunque el servidor DNS, aparece configurado, Windows XP, no puede resolver peticiones a través de este, ya que solo trabaja bien, con DNS preferidos IPV4, evitando así el uso del DNS IPv6. Razón por la cual no resolverá peticiones IPv6 internas, como por ejemplo www.ipv6.unicauca.edu.co, pero si dará respuesta ante peticiones externas, figura B25:

```
C:\Documents and Settings\Unicauca>ping6 www.kame.net
Haciendo ping www.kame.net [2001:200:0:8002:203:47ff:fea5:3085]
de 2001:448:1024:1:b7ab:ae87:ee46:5a0b con 32 bytes de datos:
Respuesta desde 2001:200:0:8002:203:47ff:fea5:3085: bytes=32 tiempo=519ms
Respuesta desde 2001:200:0:8002:203:47ff:fea5:3085: bytes=32 tiempo=507ms
Respuesta desde 2001:200:0:8002:203:47ff:fea5:3085: bytes=32 tiempo=431ms
Respuesta desde 2001:200:0:8002:203:47ff:fea5:3085: bytes=32 tiempo=469ms
```

Figura B25. Prueba de alcanzabilidad externa.

### Prueba 3.2 – Servidor Dibbler, cliente Dibbler Linux:

En el cliente se ejecuta el comando `# usr/sbin/dibbler-client run`, para inicializar el servicio, si el arranque es exitoso se podrá apreciar un comportamiento semejante al mostrado en la Figura B26.

```
41:51 Debug      Sending REQUEST on eth0/2 to multicast.
41:51 Notice     Sleeping for 1 second(s).
41:51 Debug      Received 164 bytes on interface eth0/2 (socket=6, addr=fe80::203:ceff:fe89:2fc2.).
41:51 Notice     Received REPLY on eth0/2,TransID=0x6039eb, 5 opts: 1 3 2 23 24
41:51 Notice     Address 2001:448:1024:1:d38b:bc0d:caef:a1b added to eth0/2 inter
face.
41:51 Notice     Setting up DNS server 2001:448:1024:1::10 on interface eth0/2.
41:51 Notice     Setting up Domain ipv6.unicauca.edu.co on interface eth0/2.
41:51 Notice     Sleeping for 1 second(s).
```

Figura B26. Inicialización cliente Dibbler Linux.





Nótese que todos los parámetros de configuración fueron aceptados por el cliente, tal como lo confirman las figuras B27 (comando *ifconfig*) y 28 (archivo *resolv.conf*):

```
[invitado@atlas ~]$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:12:3F:6F:9D:5B
          inet addr:172.16.30.117  Bcast:172.16.255.255  Mask:255.255.0.0
          inet6 addr: 2001:448:1024:1:d38b:bc0d:caef:alb/128 Scope:Global
          inet6 addr: fe80::212:3fff:fe6f:9d5b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:33669 errors:0 dropped:0 overruns:0 frame:0
          TX packets:59 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5693844 (5.4 MiB)  TX bytes:4657 (4.5 KiB)
          Interrupt:11
```

Figura B27. Asignación de una dirección IPv6 a un cliente Dibbler Linux

```
nameserver 2001:448:1024:1::10
search ipv6.unicauca.edu.co
```

Figura B28. Archivo *resolv.conf*, cliente Dibbler Linux

Por último se comprueba el acceso del *host*, a la red mundial IPv6, mediante un *ping6* a una pagina IPv6 pura, figura B29.

```
[invitado@atlas ~]$ ping6 www6.tahi.org
PING www6.tahi.org(pickles.tahi.org) 56 data bytes
64 bytes from pickles.tahi.org: icmp_seq=0 ttl=48 time=363 ms
64 bytes from pickles.tahi.org: icmp_seq=1 ttl=48 time=377 ms
64 bytes from pickles.tahi.org: icmp_seq=2 ttl=48 time=379 ms
64 bytes from pickles.tahi.org: icmp_seq=3 ttl=48 time=373 ms
64 bytes from pickles.tahi.org: icmp_seq=4 ttl=48 time=365 ms
[invitado@atlas ~]$ ping6 www.ipv6.unicauca.edu.co
PING www.ipv6.unicauca.edu.co(www.ipv6.unicauca.edu.co) 56 data bytes
64 bytes from correo6.ipv6.unicauca.edu.co: icmp_seq=0 ttl=64 time=0.507 ms
64 bytes from zeus.ipv6.unicauca.edu.co: icmp_seq=1 ttl=64 time=0.519 ms
64 bytes from dns6.ipv6.unicauca.edu.co: icmp_seq=2 ttl=64 time=0.515 ms
64 bytes from www.ipv6.unicauca.edu.co: icmp_seq=3 ttl=64 time=0.518 ms
```

Figura B29. Prueba de alcanzabilidad cliente Dibbler.

#### Prueba 4: -- Servidor Dibbler, cliente Sourceforce, (linux):

El archivo de configuración que determina el comportamiento del servidor Dibbler, fue el mismo que el utilizado en la prueba anterior B21, mientras que el archivo de configuración del cliente Sourceforce puede apreciarse en la figura B9, prueba 1.

Al realizar la prueba, se registró una interacción parcial entre los sistemas Dibbler y Sourceforce. Al ser inicializados se puede observar un intercambio de mensajes DHCP, mediante los cuales se realiza la configuración de una dirección IPv6 para el cliente, como se observa en las figuras B30 y B31:



```
07:07:07 Notice    Accepting connections. Next event in 2147483647 second(s).
07:09:24 Notice    Received SOLICIT on eth0/2,TransID=0xcceb5c7, 5 opts: 1 8 14 3
6, 0 relay(s).
07:09:24 Info      SOLICIT with RAPID-COMMIT received, but RAPID-COMMIT is disab
led on eth0 interface.
07:09:24 Notice    Generating ADVERTISE message, RAPID COMMIT option ignored.
07:09:24 Warning   IA option (with IAADDR suboptions missing) received. Assignin
g one address.
07:09:24 Info      Client requested ::, got 2001:448:1024:1:9116:7f78:2e0b:a143
(IAID=1866404352, pref=3600,valid=7200).
07:09:24 Notice    Sending ADVERTISE on eth0/2,transID=0xcceb5c7, opts: 1 3 2 7,
0 relay(s).
07:09:24 Notice    Accepting connections. Next event in 2147483647 second(s).
```

Figura B30. Interacción de un servidor Dnsmasq con un cliente Sourceforce.

```
Sep/16/2006 07:12:55 IA address option: opt_5, len 24
Sep/16/2006 07:12:55 get IAADDR address information: 2001:448:1024:1:62e0:b7db:
4cb0:900 preferlifetime 3600 validlifetime 7200
Sep/16/2006 07:12:55 IA address option: status code, len 64
Sep/16/2006 07:12:55 status code for this address is: success
```

Figura B31. Interacción de un cliente Sourceforce con un servidor Dnsmasq.

Al ejecutar el comando `/sbin/ifconfig`, podemos observar que la dirección fue, efectivamente, configurada en el cliente, figura B32.

Pero sin embargo al revisar la configuración del sistema, en el archivo `/etc/resolv.conf`, para verificar la configuración automática del DNS, se observó que no fue exitoso, por tanto la navegabilidad no será completamente funcional. Lo anterior debido al formato dispar de los mensajes de opciones de configuración adicional, que manejan las dos implementaciones.

```
eth0    Link encap:Ethernet  HWaddr 00:12:3F:6F:9D:5B
        inet addr:172.16.30.117  Bcast:172.16.255.255  Mask:255.255.0.0
        inet6 addr: 2001:448:1024:1:62e0:b7db:4cb0:900/64 Scope:Global
        inet6 addr: fe80::212:3fff:fe6f:9d5b/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:4064 errors:0 dropped:0 overruns:0 frame:0
        TX packets:26 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:751028 (733.4 KiB)  TX bytes:2489 (2.4 KiB)
        Interrupt:11
```

Figura B32. Configuración de dirección IPv6 en un cliente Sourceforce.

## B.4 PRUEBAS SERVIDOR HÍBRIDO STATELESS-STATEFUL.

### Prueba 1. Servidor Dnsmasq – cliente Dnsmasq (Linux y Windows).

#### Prueba 1.1 Servidor Dnsmasq + Radvd – cliente Dnsmasq Windows.

En este caso, el tipo de autoconfiguración que se busca, se puede lograr adhiriendo la opción *stateless* con un alcance adecuado, en los dos archivos de configuración (cliente o servidor), como se muestra en las figuras B33 y B34.



```
log-level 8
log-mode short
stateless

iface eth0 {

    option dns-server 2001:448:1024:1::10
    option domain ipv6.unicauca.edu.co
    option lifetime 1800

}
```

Figura B33. Archivo de configuración Server.conf en modo stateless

```
log-level 8
log-mode short

iface eth0
{
    stateless

    option dns-server
    option domain
}
```

Figura B34. Archivo de configuración client.conf modo stateless

Además en el archivo de configuración *radvd.conf*, la bandera *AdvAutonomous* debe tener por valor *on*.

Es de notar que existen ciertos condicionamientos en esta configuración:

- Si la opción *stateless* no es incluida en el archivo de configuración del servidor, y tampoco se incluye ninguna definición IA con direcciones para otorgar a potenciales clientes, se generará un error como se muestra en la figura B35.

```
04:30:22 Critical Config problem: Interface eth0/2: No class definitions present, but stateless mode not set.
04:30:22 Critical Fatal error during CfgMgr initialization.
04:30:22 Notice Bye bye.
```

Figura B35. Error de configuración en Server.conf

- Si la opción *stateless* esta configurada en el servidor y no así en el cliente, se presentará el error mostrado en la figura B36, por el cual no se logrará la configuración de las opciones adicionales.

```
05:00:50 Notice Received SOLICIT on eth0/2,TransID=0x562c, 4 opts: 1 3 8 6, 0 relay(s).
05:00:50 Warning Stateful configuration related message received while running in the stateless mode. Message ignored.
```

Figura B36. Error producido por una configuración en client.conf

Otra forma de lograr configurar las opciones adicionales sin involucrar la opción *stateless*,



es que el cliente solicite únicamente las opciones adicionales de configuración, como lo muestra la figura B37:

```
log-level 8
log-node short

iface eth0
{
  option dns-server
  option domain
}
```

Figura B37. Archivo staeless cliente Dibbler.

Mientras que el servidor muestra un archivo de configuración igual al utilizado en pruebas anteriores, figura B21.

Una vez cliente y servidor están corriendo apropiadamente, el cliente debería presentar una configuración completa, con una dirección construida en base a su dirección hardware y el prefijo enviado por el enrutador. Mientras que la información correspondiente a DNS y dominio será entregada por el servidor DHCPv6, figura B38 y B39.

```
10:32:53 Info      Creating INFORMATION-REQUEST message on Conexión de área local
1/6 interface.
10:32:53 Debug    Creating INF-REQUEST on the Conexión de área local/6.
10:32:53 Notice   Sleeping for 1 second(s).
10:32:54 Info      Processing msg <INF-REQUEST,transID=0x6ba7,opts: 1 8 6>
10:32:54 Debug    Sending INF-REQUEST on Conexión de área local/6 to multicast.

10:32:54 Notice   Sleeping for 1 second(s).
10:32:54 Debug    Received 94 bytes on interface Conexión de área local/6 (sock
et=1712, addr=fe80::203:ceff:fe89:2fc2.).
10:32:54 Notice   Received REPLY on Conexión de área local/6,TransID=0x6ba7, 5
opts: 1 23 24 42 2
10:32:54 Notice   Setting up DNS server 2001:448:1024:1::10 on interface Conexi
ón de área local/6.
10:32:54 Notice   Setting up Domain ipv6.unicauca.edu.co on interface Conexión
de área local/6.
10:32:54 Info      Next option renewal in 1800 seconds .
10:32:54 Debug    All requested options were assigned.
```

Figura B38. Asignación a un cliente Dibbler de información de configuración adicional.

```
Dirección IP. . . . . : 172.16.30.117
Máscara de subred . . . . . : 255.255.0.0
Dirección IP. . . . . : 2001:448:1024:1:212:3fff:fe6f:9d5b4
Dirección IP. . . . . : fe80::212:3fff:fe6f:9d5b26
Puerta de enlace predeterminada : 10.200.2.254
                               fe80::203:ceff:fe89:2fc2x6
Servidores DNS . . . . . : 2001:448:1024:1::10
```

Figura B39. Respuesta al comando ipconfig en un cliente Windows.

La información entregada al cliente se registra en el archivo *server-CfgMgr.xml* ubicado en */var/lib/dibbler*, figura B40.



```
<SrvCfgMgr>
  <workDir>/var/lib/dibbler</workDir>
  <LogName>Server</LogName>
  <LogLevel>8</LogLevel>
  <duid length="14">0x00:01:00:00:44:47:a2:c6:00:12:3f:6f:9f:42</duid>
  <stateless/>
  <SrvCfgIface name="eth0" ifindex="2">
    <!-- <relay/> -->
    <preference>0</preference>
    <ifaceMaxLease>4294967295</ifaceMaxLease>
    <clntMaxLease>4294967295</clntMaxLease>
    <!-- <unicast/> -->
    <!-- <rapid-commit/> -->
    <!-- IPv6 addr class count: 0-->
    <!-- options -->
    <!-- <dns-servers count="1"> -->
    <dns-server>2001:448:1024:1::10</dns-server>
    <!-- <domains count="1"> -->
    <domain>ipv6.unicauca.edu.co</domain>
    <!-- <ntp-servers count="0"> -->
    <!-- <timezone/> -->
    <!-- <sip-servers count="0"> -->
    <!-- <sip-domains count="0"> -->
    <!-- <fqdn/> -->
    <!-- <nis-servers count="0"> -->
    <!-- <nis-domain/> -->
    <!-- <nisplus-servers count="0"> -->
    <!-- <nisplus-domain/> -->
    <lifetime>1800</lifetime>
  </SrvCfgIface>
</SrvCfgMgr>
```

Figura B40. Archivo server-CfgMgr.xml

Como se mencionó anteriormente, aunque el servidor DNS aparece configurado, Windows XP, no puede resolver peticiones a través de este, ya que solo trabaja bien, con DNS preferidos IPV4, evitando así el uso del DNS IPv6. Razón por la cual no resolverá peticiones IPv6 internas, como por ejemplo www.ipv6.unicauca.edu.co

### Prueba 1.2 Servidor Dibbler + Radvd – cliente Dibbler Linux.

Los archivos de configuración en esta prueba, son exactamente iguales a los utilizados para la experiencia anterior. Las figuras B41, B42 y B43. Ilustran la configuración de un cliente Linux, por parte del servidor DCHPv6 Dibbler y de Radvd.

```
50:40 Debug    Creating INF-REQUEST on the eth0/2.
50:40 Notice   Sleeping for 1 second(s).
50:41 Info     Processing msg (INF-REQUEST,transID=0x3778cf,opts: 1 8 6)
50:41 Debug    Sending INF-REQUEST on eth0/2 to multicast.
50:41 Notice   Sleeping for 1 second(s).
50:41 Debug    Received 94 bytes on interface eth0/2 (socket=6, addr=fe80::203:ceff:fe89:2fc2.).
50:41 Notice   Received REPLY on eth0/2,TransID=0x3778cf, 5 opts: 1 23 24 42 2
50:41 Notice   Setting up DNS server 2001:448:1024:1::10 on interface eth0/2.
50:41 Notice   Setting up Domain ipv6.unicauca.edu.co on interface eth0/2.
50:41 Info     Next option renewal in 1800 seconds .
50:41 Debug    All requested options were assigned.
```

Figura B41. Asignación de información de configuración a un Cliente Dibbler Linux



```
eth0      Link encap:Ethernet  HWaddr 00:12:3F:6F:9D:5B
          inet addr:172.16.30.117  Bcast:172.16.255.255  Mask:255.255.0.0
          inet6 addr: 2001:448:1024:1:212:3fff:fe6f:9d5b/64 Scope:Global
          inet6 addr: fe80::212:3fff:fe6f:9d5b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:100333 errors:0 dropped:0 overruns:0 frame:0
          TX packets:393 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13332632 (12.7 MiB)  TX bytes:42700 (41.6 KiB)
          Interrupt:11
```

Figura B42. Configuración de una de una dirección IPv6, en un cliente Dibbler Linux.

```
nameserver 2001:448:1024:1::10
search ipv6.unicauca.edu.co
```

Figura B43. Dominio y Servidor DNS configurados en el archivo resolv.conf en un cliente Dibbler Linux.

De esta manera se obtiene plena funcionalidad y conectividad para el cliente hacia la red IPv6. Figuras B44 y B45.

```
PING www.ipv6.unicauca.edu.co(www.ipv6.unicauca.edu.co) 56 data bytes
64 bytes from correo6.ipv6.unicauca.edu.co: icmp_seq=0 ttl=64 time=0.527 ms
64 bytes from zeus.ipv6.unicauca.edu.co: icmp_seq=1 ttl=64 time=0.519 ms
64 bytes from dns6.ipv6.unicauca.edu.co: icmp_seq=2 ttl=64 time=0.515 ms
```

Figura B44. Ping6 comprobando conectividad y resolución interna.

```
PING www6.tah1.org(pickles.tah1.org) 56 data bytes
64 bytes from pickles.tah1.org: icmp_seq=8 ttl=54 time=950 ms
64 bytes from pickles.tah1.org: icmp_seq=9 ttl=54 time=773 ms
```

Figura B45. Ping6 comprobando conectividad y resolución externa.

## Prueba 2. Servidores Sourceforce + Radvd – cliente Sourceforce.

La figura B46. Ejemplifica un archivo de configuración para un cliente Sourceforce Linux, que solicita únicamente información de configuración adicional.

```
interface eth0 {
    information-only;
    send rapid-commit;
    request prefix-delegation;
};
```

Figura B46. Archivo de configuración de un cliente Sourceforce solicitando información de configuración adicional.

Al iniciar cliente, servidor y el demonio *radvd*, apropiadamente configurados, se obtiene una completa configuración, tanto de los parámetros adicionales de configuración como también de una dirección IPv6, obsérvese figura B47.



```
Sep/20/2006 07:10:19 get DHCP option DNS_RESOLVERS, len 16
Sep/20/2006 07:10:19 get DHCP option opt_26, len 22
Sep/20/2006 07:10:19 expand domain name ipv6.unicauca.edu.co, size 20
Sep/20/2006 07:10:19 reply message XID is (606c24)
Sep/20/2006 07:10:19 ifp 0x9e44008 event 0x9e482b8 id is 606c24
Sep/20/2006 07:10:19 serverID is 00:01:00:01:0b:c6:4b:c9:00:03:ce:89:2f:c2 len i
s 14
Sep/20/2006 07:10:19 received nameserver[0] 2001:448:1024:1::10
Sep/20/2006 07:10:19 received domainname[0] ipv6.unicauca.edu.co
Sep/20/2006 07:10:19 domain name ipv6.unicauca.edu.co added in resolv.conf
Sep/20/2006 07:10:19 status code: success
```

Figura B47. Obtención por parte del cliente Sourceforce e información de configuración adicional.

Los parámetros configurados son exactamente iguales a los que se pueden observar en las figuras B42 y B43 del apartado anterior. De esta manera se obtiene plena funcionalidad y conectividad para el cliente hacia la red IPv6.

## B5. PRUEBAS REALIZADAS CON ETHEREAL.

A pensar que en pantalla se puede observar el intercambio de mensajes DHCPv6 al adherir al comando de inicialización de Servidor y cliente Sourceforce la directiva *-Ddf*, o en Dibbler ejecutando la opción run, se puede dificultar identificar una secuencia completa de mensajes entre cliente y servidor, mas aún al aumentar el numero de clientes. Por tal razón se realizo un análisis de las herramientas mediante el software Ethereal. Es importante destacar que las dos herramientas presentaron resultados idénticos, lo cual resulta lógico ya que ambas implementaciones basan su comportamiento en el RFC 3315.

La prueba se realizó sobre un cliente que DHCPv6 que había sido configurado anteriormente con la dirección 2001:448:1024:1::11, por tanto este cliente debería intentar configurar esta dirección otra vez al ser inicializado nuevamente . En la figura B48 se observa como el nodo, primero que todo envía un mensaje multicast para alertar al enrutador de su presencia.



No. -	Time	Source	Destination	Protocol	Info
129	1.725793	fe80::208:54ff:fe1f:37f0	ff02::16	ICMPv6	Multicast Listener Report Message v2
139	1.845773	::	ff02::1:ff00:11	ICMPv6	Neighbor solicitation
197	2.527545	fe80::208:54ff:fe1f:37f0	ff02::1:2	DHCPv6	Confirm
198	2.527753	fe80::203:ceff:fe89:2fc2	fe80::208:54ff:fe1f	DHCPv6	Reply
516	7.633342	fe80::208:54ff:fe1f:37f0	ff02::16	ICMPv6	Multicast Listener Report Message v2
567	8.617096	fe80::208:54ff:fe1f:37f0	ff02::16	ICMPv6	Multicast Listener Report Message v2
606	8.989013	::	ff02::1:ff00:11	ICMPv6	Neighbor solicitation
617	9.146175	fe80::203:ceff:fe89:2fc2	ff02::1	ICMPv6	Router advertisement
636	9.500099	fe80::208:54ff:fe1f:37f0	ff02::1:2	DHCPv6	Confirm
637	9.500323	fe80::203:ceff:fe89:2fc2	fe80::208:54ff:fe1f	DHCPv6	Reply

```

> Ethernet II, Src: 00:08:54:1f:37:f0, Dst: 33:33:00:00:00:16
< Internet Protocol Version 6
  Version: 6
  Traffic class: 0x00
  Flowlabel: 0x00000
  Payload length: 56
  Next header: IPv6 hop-by-hop option (0x00)
  Hop limit: 1
  Source address: fe80::208:54ff:fe1f:37f0 (fe80::208:54ff:fe1f:37f0)
  Destination address: ff02::16 (ff02::16)
  
```

Figura B48. Reporte del cliente a todos los enrutadores de la red.

En la figura B49, se aprecia como el cliente envía un mensaje Confirm con la intención de confirmar si la dirección asignada anteriormente aún es válida para el enlace, para ello utiliza la dirección muticast para todos los servidores y Agentes Relay dentro del link. Es de notar el uso del puerto 547 para esta operación.

No. -	Time	Source	Destination	Protocol	Info
129	1.725793	fe80::208:54ff:fe1f:37f0	ff02::16	ICMPv6	Multicast Listener Report Message v2
139	1.845773	::	ff02::1:ff00:11	ICMPv6	Neighbor solicitation
197	2.527545	fe80::208:54ff:fe1f:37f0	ff02::1:2	DHCPv6	Confirm
198	2.527753	fe80::203:ceff:fe89:2fc2	fe80::208:54ff:fe1f	DHCPv6	Reply
516	7.633342	fe80::208:54ff:fe1f:37f0	ff02::16	ICMPv6	Multicast Listener Report Message v2
567	8.617096	fe80::208:54ff:fe1f:37f0	ff02::16	ICMPv6	Multicast Listener Report Message v2
606	8.989013	::	ff02::1:ff00:11	ICMPv6	Neighbor solicitation
617	9.146175	fe80::203:ceff:fe89:2fc2	ff02::1	ICMPv6	Router advertisement
636	9.500099	fe80::208:54ff:fe1f:37f0	ff02::1:2	DHCPv6	Confirm
637	9.500323	fe80::203:ceff:fe89:2fc2	fe80::208:54ff:fe1f	DHCPv6	Reply

```

> Frame 197 (134 bytes on wire, 134 bytes captured)
> Ethernet II, Src: 00:08:54:1f:37:f0, Dst: 33:33:00:01:00:02
> Internet Protocol Version 6
> User Datagram Protocol, Src Port: dhcpv6-server (547), Dst Port: dhcpv6-server (547)
< DHCPv6
  Message type: Confirm (4)
  Transaction-ID: 0x00265233
  > Client Identifier
  > Elapsed time
  > Identity Association for Non-temporary Address
  
```

Figura B49. Mensaje Confirm.

El siguiente paso registrado es la replica enviada por el servidor DHCP, hacia el cliente mediante la cual confirma que la dirección solicitada es adecuada para el enlace actual, esta dirección es reconfirmada en el campo IA del mensaje Reply. Nótese que en todo el proceso el cliente utiliza su dirección link-local generada al momento de la inicialización para realizar esta negociación. Figura B50.





No. .	Time	Source	Destination	Protocol	Info
129	1.725793	fe80::208:54ff:fe1f:37f0	ff02::16	ICMPv6	Multicast Listener Report Message v2
139	1.845773	::	ff02::1:ff00:11	ICMPv6	Neighbor solicitation
197	2.527545	fe80::208:54ff:fe1f:37f0	ff02::1:2	DHCPv6	Confirm
198	2.527753	fe80::203:ceff:fe89:2fc2	fe80::208:54ff:fe1f	DHCPv6	Reply
516	7.633342	fe80::208:54ff:fe1f:37f0	ff02::16	ICMPv6	Multicast Listener Report Message v2
567	8.617096	fe80::208:54ff:fe1f:37f0	ff02::16	ICMPv6	Multicast Listener Report Message v2
606	8.989013	::	ff02::1:ff00:11	ICMPv6	Neighbor solicitation
617	9.146175	fe80::203:ceff:fe89:2fc2	ff02::1	ICMPv6	Router advertisement
636	9.500099	fe80::208:54ff:fe1f:37f0	ff02::1:2	DHCPv6	Confirm
637	9.500323	fe80::203:ceff:fe89:2fc2	fe80::208:54ff:fe1f	DHCPv6	Reply

```
T2: 0
  ▾ IA Address
    option type: 5
    option length: 24
    IPv6 address: 2001:448:1024:1::11
    Preferred lifetime: 0
    Valid lifetime: 0
  ▷ Preference
  ▷ Status code
  ▷ Identity Association for Prefix Delegation
```

Figura B50. Mensaje Reply.

Por último en la figura B51, se aprecia un mensaje RA enviado desde Radvd, este mensaje contiene la información referente a enrutamiento y el prefijo usado en la red.

No. .	Time	Source	Destination	Protocol	Info
129	1.725793	fe80::208:54ff:fe1f:37f0	ff02::16	ICMPv6	Multicast Listener Report Message v2
139	1.845773	::	ff02::1:ff00:11	ICMPv6	Neighbor solicitation
197	2.527545	fe80::208:54ff:fe1f:37f0	ff02::1:2	DHCPv6	Confirm
198	2.527753	fe80::203:ceff:fe89:2fc2	fe80::208:54ff:fe1f	DHCPv6	Reply
516	7.633342	fe80::208:54ff:fe1f:37f0	ff02::16	ICMPv6	Multicast Listener Report Message v2
567	8.617096	fe80::208:54ff:fe1f:37f0	ff02::16	ICMPv6	Multicast Listener Report Message v2
606	8.989013	::	ff02::1:ff00:11	ICMPv6	Neighbor solicitation
617	9.146175	fe80::203:ceff:fe89:2fc2	ff02::1	ICMPv6	Router advertisement
636	9.500099	fe80::208:54ff:fe1f:37f0	ff02::1:2	DHCPv6	Confirm
637	9.500323	fe80::203:ceff:fe89:2fc2	fe80::208:54ff:fe1f	DHCPv6	Reply

```
Retrans time: 0
  ▾ ICMPv6 options
    Type: 3 (Prefix information)
    Length: 32 bytes (4)
    Prefix length: 64
  ▷ Flags: 0x80
    Valid lifetime: 0x00278d01
    Preferred lifetime: 0x00093a80
    Prefix: 2001:448:1024:1::
  ▷ ICMPv6 options
```

Figura B51. RA.