



**UNIMINUTO**  
Corporación Universitaria Minuto de Dios

# **John R. Correa**

## **Administrador de Seguridad de la Información.**



# Agenda

2. Implementando un servidor DHCP6 en IPv6.
  - 2.1. Introducción.
  - 2.2. ICMPv6
  - 2.3. Neighbor Discovery (ND).
  - 2.4. Router Advertisements (RA).
  - 2.5. Autoconfiguración.
    - 2.5.1. Autoconfiguración Stateless o Serverless (RFC4862)
  - 2.6. Ventajas/Beneficios de la Autoconfiguración Stateless.
  - 2.7. Autoconfiguración Stateful o DHCPv6 (RFC3315).
  - 2.8. Servidores DHCP con soporte Ipv6.
  - 2.9. DHCPv6.
    - 2.9.1. DHCPv6 Ventajas.
    - 2.9.2. DHCPv6 Desventajas.
    - 2.9.3. Cambios en DHCPv6.
  - 2.10. Instalación de paquetes necesarios para el servidor DHCP6.
  - 2.11. Configuración de ficheros.



# Agenda

3. Implementando un servidor FIREWALL en IPv6.
  - 3.1. Introducción.
  - 3.2. Instalación de paquetes necesarios para el servidor Iptables6.



## 2. Implementando un servidor DHCP6 en IPv6.

### 2.1. Introducción.

**El servicio DHCP (Dynamic Host Configuration Protocol) o (Protocolo de configuración dinámica de servidores), es un protocolo que permite a los dispositivos y equipos determinar cierta información en cuanto a la configuración de red, por ejemplo: (Dirección IP, máscara de sub-red, puerta de enlace, servidor DNS principal y secundario, etc).**

**El protocolo DHCPv6, trabaja de forma conjunta con el mecanismo “stateless”. Por lo cual el encargado de la administración de la red debe determinar los procesos que se van a emplear o enviar a través de los mensajes “RA” de ICMPv6. Otra posibilidad que ofrece el protocolo es la de permitir a los clientes solicitar múltiples direcciones IPv6, algo que no era posible en la antigua estructura del protocolo IPv4.**

**DHCPv6 funciona sobre el protocolo de transporte UDP, de tal forma que el cliente utiliza una dirección link-local u otra determinada a través de otros mecanismos para transmitir y recibir los mensajes generados por el protocolo DHCPv6.**



## 2.2. ICMPv6

**El protocolo ICMPv6 (Internet Control Message Protocol), es fundamental y parte integral de IPv6 y debe ser completamente implementado por cada nodo IPv6. El protocolo ICMP se encuentra definido para IPv4 en la RFC792, y para la versión de IPv6 se han realizado algunos cambios de importancia los cuales se han documentado en la RFC4443**

**El protocolo ICMPv6 genera dos tipos de mensajes que se clasifican en “Mensajes de error y Mensajes informativos”.**

**Los mensajes de error para IPv6 se dividen en cuatro categorías (destino inaccesible, paquete demasiado grande, tiempo excedido y problemas de parámetros). Tienen un cero en el bit de mayor orden del valor del campo Type. El valor del campo Type es de 0 a 127.**



## 2.2. ICMPv6

- 1 Destination Unreachable (Destino Inalcanzable)**
- 2 Packet Too Big (Paquete Demasiado Grande)**
- 3 Time Exceeded (Tiempo Agotado)**
- 4 Parameter Problem (Problema de Parámetros)**

Mientras que los mensajes informativos tienen valores para el campo Type de 128 a 255. Estos mensajes se subdividen en tres grupos (mensajes de diagnóstico, mensajes para la administración de grupos multicast y mensajes de Neighbor Discovery).

- 128 Echo Request (Solicitud de Eco)**
- 129 Echo Reply (Respuesta de Eco)**



## 2.3. Neighbor Discovery (ND).

Es un protocolo de IPv6 equivalente al protocolo utilizado en IPv4 ARP (Address Resolution Protocol), el cual incorpora además de las funcionalidades definidas en ARP otras de otros protocolos de esta versión.

Neighbor Discovery es un mecanismo con el cual un nodo que se incorpora a una red, descubre la presencia de otros nodos en el mismo enlace, obteniendo las direcciones IPv6 asignadas a cada uno de ellos. Otra funcionalidad de este protocolo es la de mantener limpios los caches donde es almacenada la información relativa del contexto de red a la cual se está conectando un nuevo nodo. De esta forma cuando una ruta definida hacia cierto nodo falla, el equipo enrutador buscará las rutas alternativas, para ello emplea los mensajes de ICMPv6, por lo cual es base fundamental para permitir la autoconfiguración en IPv6 para los dispositivos.



## 2.3. Neighbor Discovery (ND).

Los hosts también utilizan Neighbor Discovery para encontrar enrutadores vecinos que retransmitirán los paquetes que se les envíen.

Por lo tanto los nodos usan el protocolo para tener conocimiento de los vecinos que son alcanzables y los que no, para detectar cambios de sus direcciones en la capa de enlace.

Neighbor Discovery define cinco tipos de paquetes:

1. Router Solicitation (RS): 133
2. Router Advertisement (RA): 134
3. Neighbor Solicitation (NS): 135
4. Neighbor Advertisement (NA): 136
5. Redirect: 137





## 2.4. Router Advertisements (RA).

Para los enlaces multicast y enlaces punto a punto, cada uno de los enrutadores difunde periódicamente un paquete de anuncio de rutas en el cual se establece su disponibilidad.

Los hosts reciben estos anuncios y construyen a partir de esta información listas de routers por defecto.

Los RAs y los “Flags” asociados a cada prefijo permiten a los enrutadores indicar a los hosts como deben realizar la autoconfiguración la cual puede ser (stateless o DHCPv6).



## 2.5. Autoconfiguración.

El estándar especifica los pasos que un equipo final host debe seguir para decidir cómo auto-configurar sus interfaces de red para IPv6.

IPv6 define el mecanismo de auto-configuración de las direcciones tipo “stateful” como “stateless”.

El proceso de auto-configuración incluye la creación de una dirección IPv6 de ámbito local llamada (link-local) y además se realiza una verificación para determinar que la dirección no esta duplicada en el mismo segmento de red.



## 2.5.1. Autoconfiguración Stateless o Serverless (RFC4862)

Por lo tanto el mecanismo “stateless” permite a un host generar su propia dirección IPv6, usando una combinación de cierta información localmente disponible y de información proporcionada por los routers.

Por lo tanto los routers anuncian los prefijos de red que identifican la subred asociada a un determinado segmento de red

Los hosts generan un identificador de interfaz que lo identifica de manera única en la subred. Este identificador se genera localmente, a partir de la dirección física del dispositivo conocida como dirección MAC.

Por lo tanto la dirección IPV6 se forma mediante la combinación de ambas informaciones obtenidas.



## 2.6. Ventajas/Beneficios de la Autoconfiguración Stateless.

- La configuración manual de cada host antes de ser conectada a la red no se necesita.
- Los sitios pequeños donde se tienen pocos equipos conectados al mismo segmento no necesitan de un servidor DHCPv6 ni de un equipo enrutador para comunicarse, ya que estos usarían direcciones locales o link-local.
- Un sitio grande con varias subredes no necesita de un servidor DHCPv6 para la configuración de direcciones.



## 2.7. Autoconfiguración Stateful o DHCPv6 (RFC3315).

Cuando se utiliza el método Stateful o DHCPv6, los hosts obtienen las direcciones de la interfaz de red y/o la información de configuración desde un servidor configurado para tal fin.

Por lo tanto los servidores mantienen una base de datos que actualizan con las direcciones IPv6 que han asignado y con la información de los hosts que se han asignado.

Las auto-configuraciones “stateless” y la “stateful” se complementan una a la otra, ya que ambos tipos se pueden usar de forma simultánea sin que se presenten errores de asignación de direcciones.

Sin embargo es recomendable que el encargado de administrar la red, especifique que tipo de auto-configuración se usa, por medio de la configuración de los campos adecuados de los mensajes Ras.



## 2.8. Servidores DHCP con soporte Ipv6.

Para la implementación de un servidor DHCP con soporte para IPv6, existen algunos proyectos que pueden ser utilizados en sistemas operativos, ya sean de carácter propietario como libre, estos se pueden clasificar según el tipo de sistema operativo que se desee utilizar, por ejemplo:

- Dnsmasq (<http://klub.com.pl/dhcpv6/>)
- Wide.dhcpv6 (<http://wide-dhcpv6.sourceforge.net/>)
- Dnsmasq (<https://fedorahosted.org/dhcpv6>)
- ISC DHCP4.x (<http://www.isc.org/software/dhcp>)

## 2.9. DHCPv6.

**El protocolo DHCPv6 tiene como funcionamiento básico la forma cliente servidor, se encuentra definido en la RF3315 de la IETF.**

**Presenta las siguientes características.**

- Resuelve la auto configuración de direcciones.
- Brinda más control al administrador de la red sobre las asignaciones.
- Da mayor amplitud en la configuración de servicios de red.
- puede trabajar de forma conjunta con el mecanismo “stateless”.
- Permite a los clientes la solicitud de múltiples direcciones IPv6.
- Funciona sobre el protocolo de transporte UDP.
- El cliente utiliza una dirección link-local u otra determinada a través de otros mecanismos para transmitir y recibir los mensajes DHCPv6.
- Los servidores DHCPv6 reciben mensajes de los clientes utilizando una dirección reservada multicast.
- Para solicitar una o más direcciones IPv6 así como información de configuración, un cliente primero tiene que localizar al servidor DHCPv6 y luego hace la petición al servidor para la asignación de la dirección y otra información de configuración.



## 2.9.1. DHCPv6.

### Ventajas.

- Trabaja como complemento de los RAs (Router Advertisements).
- Puede publicar dinámica-mente información de DNS, SNTP, nombres de dominio, etc.
- Se puede integrar con el DNS para generar dinámica-mente registros PTR.



## 2.9.2. DHCPv6

### Desventajas.

- No esta disponible en todos los Sistemas Operativos.
- No hay garantía de que el cliente vaya a utilizar la dirección asignada al iniciar nuevas conexiones.
- No se puede asociar una dirección fija IPv6 con una MAC. (para ese procedimiento se utiliza el DUID).
- No asigna el default Gateway (Se requiere un RA).



## 2.9.3. Cambios en DHCPv6.

- Si existe un enrutador en la red y este envía un mensaje RA con los bits 0 (solicitud de opciones) y/o M (Direccionamiento Administrado, indica que debe solicitar la IP a un servidor DHCPv6).
- Los clientes envían sus solicitudes a la dirección multicast reservada: FF02::1:2 (All\_DHCP\_Relay\_Agents\_and\_Servers).
- Si el servidor no está en la misma subred, un relay reenvía la solicitud a la dirección IPv6 Unicast del servidor, si está se conoce.
- Si la dirección Unicast del servidor es conocida, podría reenviar a la otra dirección multicast reservada: FF05::1:3 (All\_DHCP\_servers)
- Para el servidor se utiliza el puerto 547 en UDP.
- Para el cliente se utiliza el puerto 546 en UDP.



# INTERCAMBIO DE CONFIGURACIÓN DHCPv6 INICIADO POR EL CLIENTE.

- Los agentes DHCPv6 envían a la dirección multicast FF02::1 mensajes DHCPv6 Advertise, anunciándose como Servidor DHCPv6.
- Los hosts interesados en obtener la configuración envían un mensaje REQUEST a la dirección de uno de estos Servidores.
- El servidor Envía la configuración al host con los parámetros solicitados.



## 2.10. Instalación de paquetes necesarios para el servidor DHCP6.

Para realizar la instalación del paquete DHCP se utiliza el siguiente comando: (yum -y install dhcp)

```
[root@Curso_IPv6 ~]# yum -y install dhcp
* base: mirror.edatel.net.co
* epel: mirror.globo.com
* epel-debuginfo: mirror.globo.com
* epel-source: mirror.globo.com
* extras: mirror.edatel.net.co
* rpmfusion-free-updates: mirror.us.leaseweb.net
* updates: mirror.edatel.net.co
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete dhcp.x86_64 12:4.2.5-36.el7.centos debe ser instalado
--> Procesando dependencias: dhcp-common = 12:4.2.5-36.el7.centos para el paquete: 12:dhcp-4.2.5-36.el7.centos.x86_64
--> Ejecutando prueba de transacción
---> Paquete dhcp-common.x86_64 12:4.2.5-27.el7.centos debe ser actualizado
--> Procesando dependencias: dhcp-common = 12:4.2.5-27.el7.centos para el paquete: 12:dhclient-4.2.5-27.el7.centos.x86_64
---> Paquete dhcp-common.x86_64 12:4.2.5-36.el7.centos debe ser una actualización
---> Paquete dhcp-libs.x86_64 12:4.2.5-27.el7.centos debe ser actualizado
---> Paquete dhcp-libs.x86_64 12:4.2.5-36.el7.centos debe ser una actualización
--> Ejecutando prueba de transacción
---> Paquete dhclient.x86_64 12:4.2.5-27.el7.centos debe ser actualizado
---> Paquete dhclient.x86_64 12:4.2.5-36.el7.centos debe ser una actualización
--> Resolución de dependencias finalizada
```



## 2.11. Configuración de ficheros.

El servidor DHCPv6 debe ser ejecutado de forma independiente para IPv4 e IPv6, por lo cual se requiere definir dos scripts diferentes para dar inicio a cada proceso.

- Para el servidor DHCP versión 4 se activaran los puertos de escucha 67 y 68 en UDP.
- para el servidor DHCPv6 se activara el puerto 547 en UDP.

La ubicación de los ficheros de configuración se ubican en el directorio `/etc/dhcp/` y llevan por nombre: `(dhcpd.conf)` para IPv4 y `(dhcpd6.conf)` para IPv6.

El archivo `/etc/dhcp/dhcpd.conf` para IPv4 se mantendrá igual, el script de inicio que permite invocar el proceso DHCP con los siguientes parámetros que definen la interfaz por la cual atenderá las solicitudes.

```
#dhcpd -q -cf /etc/dhcp/dhcpd.conf -lf /var/lib/dhcpd/dhcpd.leases eth0
```



## 2.11. Configuración de ficheros.

En el archivo `/etc/dhcpd6.conf` que define la configuración para el protocolo IPv6, el script de inicio invoca el proceso DHCP con los siguientes parámetros para que atienda por la interfaz seleccionada.

```
# dhcpd -6 -q -cf /etc/dhcp/dhcpd6.conf -lf /var/lib/dhcpd/dhcpd6.leases eth0
```



## Opciones de configuración global:

- **dhcpv6-lease-file-name <FILE>;**

Permite indicar el archivo de almacenamiento de las direcciones asignadas.

- **option dhcp6.sntp-servers <IPv6>;**

Permite especificar los servidores de tiempo.

- **option dhcp6.domain-search "mydomain.com";**

permite especificar el dominio de búsques poe defecto.

- **option dhcp6.info-refresh-time <SEGUNDOS>;**

Tiempo de refresco de la información proporcionada.

- **option dhcp6.preference <NUMERO>;**

Orden de importancia del servidor en la red.



# Configuración en modo simple para el Servidor DHCPv6:

```
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 2001:db8:1:f101::1 rede.edu;  
    link AAA {  
        range 2001:db8:1:f101::1000 to 2001:db8:1:f101::ffff/64;  
        prefix 2001:db8:1:f101::/64;  
    };  
};
```





## Configuración para el Cliente DHCP6C:

**Se requiere editar el fichero `/etc/dhcp/dhcp6c.conf`**

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

**Inicio del servicio DHCPv6 en el servidor CentOS-7.**

```
# systemctl start dhcp6s.service
```



# Configuración para el Cliente DHCP6C:

**Iniciar cliente en primer plano.**

```
# dhcp6c -f eth0
```

**Consulta de servidores DHCPv6 disponibles desde un Cliente:**

```
[root@localhost dhcpd]# dhclient -6 -d ens1
Internet Systems Consortium DHCP Client 4.2.5
Copyright 2004-2013 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on Socket/ens1
Sending on Socket/ens1
PRC: Soliciting for leases (INIT).
XMT: Forming Solicit, 0 ms elapsed.
XMT: X-- IA_NA a9:e9:0d:9c
XMT: | X-- Request renew in +3600
```



## Como definir una subred:

El rango puede especificar una subred completa, por ejemplo:

```
# Toda la subred
subnet6 2001:db8:1::/48 {
range6 2001:db8:1::/48;
}
```

```
# los últimos 32 bits de la subred.
subnet6 2001:db8:1::/48 {
range6 2001:db8:1::0 2001:db8:1::ffff:ffff;
}
```



## Definición de host:

Para definir un nuevo host en el fichero de configuración DHCPD6.CONF ya no se utiliza la dirección MAN como se tenía por costumbre para el protocolo IPv4, ahora se requiere indicar el (DUID) o identificador del host.

```
Host mipc {  
# No se utiliza la MAC, en su lugar se indica el DUID  
host-identifier option dhcp6.client-id 00:01:00:01:0D:EF:64:ED:00:21:25:5D:5F:18;  
fixed-address6 2001:db8:1::a201:  
option hostname "mipc";  
}
```



## Como obtener el DUID para un host Linux:

En linux se requiere del cliente dhcp (dhcpv6-client), este crea el fichero de datos `/var/lib/dhcp6c_duid`.

Para ver el client-id que se genera para la maquina, se usa el comando:

```
#hexdump -e ""%07.7_ax " 1/2 "%04x" " " 14/1 "%02x:" "\n" /var/lib/dhcpv6/dhcp6c_duid
```



**UNIMINUTO**  
Corporación Universitaria Minuto de Dios

**GRACIAS**