

Contents

- 1 Configuring stateless DHCPv6 server and client
 - ◆ 1.1 Configuring stateless DHCPv6 server
 - ◆ 1.2 Configuring stateless DHCPv6 client
 - ◇ 1.2.1 Configuring interface for DHCPv6 client requests
 - ◇ 1.2.2 Configuring DHCPv6 client
- 2 Configuring state-full DHCPv6 server and client
 - ◆ 2.1 Configuring state-full DHCPv6 server
 - ◆ 2.2 Configuring state-full DHCPv6 client
- 3 Static address allocation to client using DHCPv6
 - ◆ 3.1 Server Configuration
 - ◆ 3.2 Client configuration
- 4 DHCPv6 firewall considerations
 - ◆ 4.1 DHCPv6 server firewall configuration
 - ◆ 4.2 DHCPv6 client firewall configuration
- 5 Miscellaneous

Configuring stateless DHCPv6 server and client

Configuring stateless DHCPv6 server

In IPv4 networks DHCP servers not only provide IP address information, but also provide other important information like DNS servers, search domains, etc. Since in IPv6 we have option of auto-configuration, where through router advertisement messages hosts can learn which prefixes are being used on the link and configure an address for all the prefixes. Thus eliminating need of DHCP servers to distribute unique addresses. But we still need central mechanism for distributing information like domain names, DNS servers, etc.

Hence we have concept of stateless DHCPv6 server. Stateless DHCP servers only provide information like DNS servers, search domains, etc. and they do not assign IP addresses to hosts.

To configure DHCPv6 server, we use '/etc/dhcp6s.conf' configuration file. Sample stateless DHCPv6 server configuration is

```
option dns_servers <IPv6_addr1> <IPv6_addr2> <Domain_name>
#send information-only;
```

Configuring_DHCPv6_server_and_client

```
interface eth0
{
    link eth0_link1
    {
    };
};
```

Note:

- Server can understand which values are IPv6 addresses and which value is for domain name, we need not indicate it explicitly while using 'option dns_servers'
- It is not necessary to specify 'send information-only;' in stateless server which uses above configuration as any way server is not configured for prefix/range to be distributed to client. Hence it can send only DNS information anyway. Even if we specify 'send information-only;' then also server would work fine.
- It is necessary to declare an interface and a link, even if we are not going to distribute IPs. If we do not do this clients get error 'Not on link' when they request DHCP information from server.
- We can test server configuration by running server using 'dhcp6s -D -d -f' to run it in foreground.

Configuring stateless DHCPv6 client

To configure DHCPv6 client that can take DNS information using DHCPv6 server we need to first configure interface such that during its initialization DHCPv6 protocol is also used. Then we need to configure DHCPv6 client to indicate that we are interested only in stateless information.

Configuring interface for DHCPv6 client requests

To configure interface so that it uses DHCPv6 protocol during initialization, we can add line:

```
DHCPV6C=yes
```

in file '/etc/sysconfig/network-scripts/ifcfg-eth<n>'.

Configuring DHCPv6 client

Then we need to create entry for interface in DHCPv6 client configuration file, which is '/etc/dhcp6c.conf'. For stateless client we can use:

```
interface eth0
{
    request domain-name-servers;
    request domain-search-list;
    information-only; #Very important for stateless
```

}

Note:

- Specifying parameter 'information-only' is very important while configuring stateless client, otherwise client will fail to obtain even DNS entries. DNS server does send DNS servers and search domain information, if we do not specify 'information-only' but for some reason client rejects it saying 'Unexpected reply'. Hence to troubleshoot 'Unexpected Reply' messages we need to specify 'information-only' option in stateless clients.
- We can test the client configuration using 'dhcp6c -Df eth0' command. If we want to test information-only request then we can also use 'dhcp6c -DfI eth0'. If command 'dhcp6c -Df eth0' fails to update '/etc/resolv.conf' then there is some problem with 'dhcp6c.conf' file or with DHCPv6 server. To narrow down we can use command 'dhcp6c -DfI eth0' which will fail only if there is problem with DHCPv6 server. We should ensure another copy of 'dhcp6c' is not running using 'ps -C dhcp6c' and kill it using 'killall dhcp6c', if required, before trying to run 'dhcp6c' manually in foreground with debugging enabled.

Configuring state-full DHCPv6 server and client

Configuring state-full DHCPv6 server

For state-full server we also need to specify various time-outs, IP ranges through which IPs can be allotted to client and even prefixes that can be allotted to routers and in-turn clients will end up auto-configuring IPs for those prefixes. Note that routers won't use auto-configuration, hence we need to configure IPs manually in routers for prefixes that we try to advertise.

Sample '/etc/dhcp6s.conf' configuration is:

```
option dns_servers fd57:1d29:4f94:1:0216:36ff:fe00:1 fd57:1d29:4f94:a:0216:36ff:fe00:1 ipv6test.iiit

interface eth0
{
    renew-time 60;
    rebind-time 90;
    server-preference 20;
    allow rapid-commit;
    prefer-life-time 36000;
    valid-life-time 72000;

    link AAA
    {
        range fd57:1d29:4f94:33:0216:36ff:fe00:10 to fd57:1d29:4f94:33:0216:36ff:fe00:80/64;
        prefix fd57:1d29:4f94:44:0216:36ff:fe00:0/64;
    };
};
```

Note that:

- For prefix delegation to work, we need to configure option 'request prefix-delegation' on IPv6 router and then restart 'dhcp6c' on router. Once we do this the client will learn prefix, store this information in client.leases file and also modify '/etc/radvd.conf' so that this new prefix is also advertized. Soon all the nodes in network will learn about this new prefix and configure IP in this prefix.
- Routers will not configure IPs in range specified by DHCPv6 server as routers do not learn IPs from DHCPv6 servers.
- Nodes will configure IPs in above range, just like they do in case of IPv4 DHCP. If we kill dhcp6c service then all learned DHCPv6 addresses get removed. Hence we should send 'dhcp6c' to background, if we want to verify whether we have received IP or not.

Configuring state-full DHCPv6 client

On DHCPv6 client we can use following '/etc/dhcp6c.conf' configuration to receive DHCP IPs from server.

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

On DHCPv6 routers using DHCPv6 we can use following '/etc/dhcp6c.conf' configuration so that router can also learn about prefixes from DHCPv6 server

```
interface eth0
{
    send rapid-commit;
    request prefix-delegation;
    request domain-name-servers;
    request domain-search-list;
};
```

Static address allocation to client using DHCPv6

Server Configuration

We can assign static addresses to client using `host{ }` option in `'dhcp6s.conf'` file. Sample file which assigns static address to client is like:

```
option dns_servers fd57:1d29:4f94:1:0216:36ff:fe00:1 fd57:1d29:4f94:a:0216:36ff:fe00:1 ipv6test.iiit
#send information-only;

interface eth0
{
    renew-time 60;
    rebind-time 90;
    server-preference 20;
    allow rapid-commit;
    prefer-life-time 36000;
    valid-life-time 72000;

    link AAA
    {
        pool
        {
#           range fd57:1d29:4f94:ee:0216:36ff:fe00:10 to fd57:1d29:4f94:ee:0216:36ff:fe00:10;
#           range fd57:1d29:4f94:22:0216:36ff:fe00:10 to fd57:1d29:4f94:22:0216:36ff:fe00:10;
            prefix fd57:1d29:4f94:123:0216:36ff:fe00:0/64;
            prefix fd57:1d29:4f94:456:216:36ff:fe00:aa5/64;
            prefix fd57:1d29:4f94:789:216:36ff:fe00:aa5/64;
        };

        host vm4eth0
        {
            duid 00:01:00:01:14:66:5a:fb:00:00:00:00:00:00;
            iauserinfo
            {
                iauid 83886134;
                renew-time 8000;
                rebind-time 8000;
            };
            address
            {
                fd57:1d29:4f94:22:216:36ff:fe00:1a5/64;
                prefer-life-time 8000;
                valid-life-time 8000;
            };
            address
            {
                fd57:1d29:4f94:abcd:216:36ff:fe00:1b5/64;
                prefer-life-time 8000;
                valid-life-time 8000;
            };
            address
            {
                fd57:1d29:4f94:44:216:36ff:fe00:1c5/64;
                prefer-life-time 8000;
                valid-life-time 8000;
            };
        };
    };
};
```

Note:

- We cannot use 'range' option if we want to use host{} to do static address assignment. If we try to use both than 'Not-On-Link' error is sent from server to client.
- To find server or client duid we can use 'xxd /var/lib/dhcpv6/dhcp6s_duid' on server and 'xxd /var/lib/dhcpv6/dhcp6c_duid' on clients. Only the last 14 bytes should be used and first two bytes should be ignored.
- IAID can be taken from previous lease files or can be generated randomly and configured both in client and server. Note that both iaid and duid should match between client and server for address allocation to work.
- We can give any number of static address allocation to same client by using multiple address{} blocks within single host{} block.

Client configuration

Specify iaid in '/etc/dhcp6c.conf' so that server can recognize the client.

```
interface eth0
{
    send rapid-commit;
    iaid 83886134;
    request domain-name-servers;
    request domain-search-list;
};
```

DHCPv6 firewall considerations

DHCPv6 server firewall configuration

Client send DHCPv6 requests on destination UDP port 547. Hence we should add following line to '/etc/sysconfig/ip6tables' file so that DHCPv6 client requests do not get filtered

```
-A INPUT -p udp -m udp --dport 547 -j ACCEPT
```

DHCPv6 client firewall configuration

DHCPv6 client receives response from DHCPv6 server on destination UDP port 546. Hence we can add following lines to '/etc/sysconfig/ip6tables' file so that DHCPv6 server response does not get filtered.

```
-A INPUT -p udp -m udp --dport 546 -j ACCEPT
```

Miscellaneous

- One can refer to 'man dhcp6s', 'man dhcp6c', 'man dhcp6s.conf' and 'man dhcp6c.conf' to learn about DHCPv6 server and clients
- Both server and client support debugging and running in foreground. These features can be very helpful in debugging server issues.
- Before trying a new configuration
 1. Delete server.leases file from server
 2. Stop DHCPv6 server
 3. Kill dhcp6c process on client
 4. Delete client leases file
 5. Remove any addresses that remain due to previous experiments
 6. Now start server with new configuration
 7. Run 'dhcp6c -Df eth0' on client. *Do not use Ctrl+C to kill this process or learned addresses would get removed. Use 'Ctrl+Z' and 'bg' to send process to background, if you want to run 'ifconfig' or 'ip addr show' commands for verification.*