

Contents

- 1 Configuration commands
 - ◆ 1.1 Enabling IPv6
 - ◆ 1.2 Configuring temporary static IPv6 addresses
 - ◆ 1.3 Removing IPv6 address assigned to a interface
 - ◆ 1.4 Configuring permanent static IPv6 addresses
 - ◆ 1.5 Disabling auto-configuration of addresses via router discovery/advertisements
 - ◇ 1.5.1 Disabling auto-configuration on all interfaces
 - ◇ 1.5.2 Disabling auto-configuration on particular interface
- 2 Basic network service test commands
 - ◆ 2.1 Pinging hosts using their link-local addresses
 - ◆ 2.2 Pinging all nodes or all routers multicast address
- 3 Informational commands
 - ◆ 3.1 Seeing neighbour hardware address cache
 - ◆ 3.2 Seeing IPv6 address valid and preferred lifetimes
 - ◆ 3.3 Seeing IPv6 routing table
 - ◆ 3.4 Resolving IPv6 addresses using nslookup

Configuration commands

Enabling IPv6

To enable IPv6 networking on a machine:

- Edit file '/etc/sysconfig/network' and add line
NETWORKING_IPV6=yes
- Edit file '/etc/sysconfig/network-scripts/ifcfg-eth<n>' and add line
IPv6INIT=yes

Note: Once you enable IPv6 on node it is automatically configured for auto-configuration. To disable auto-configuration on IPv6 enabled node, we need to use 'IPV6_AUTOCONF=no' option in either '/etc/sysconfig/network' (global) or in interface configuration file (per interface).

Configuring temporary static IPv6 addresses

To configure temporary static IPv6 addresses which will cease to exist if machine or network is restarted, use:

```
ifconfig eth0 add <ipv6_address>/<netmask>
```

command. Note that we can add any number of IP addresses using this technique as in IPv6 same interface can be assigned more than one IP address.

Removing IPv6 address assigned to a interface

Address prefixes learned via router will slowly get deferred and their valid lifetime would decrease, and eventually they will get removed automatically. However if we have added some address (temporary or permanent) statically then to remove it we can use:

```
ifconfig eth0 del <ipv6_address>/<netmask>
```

- Note that if we remove some address that was learned via router, then it is possible that the address gets added again on next router advertisement message. To really ensure that address does not get added again we have to disable auto-configuration ('`sysctl net.ipv6.conf.{all|<interface>.autoconf = 0`') or ignore all router advertisements ('`sysctl net.ipv6.conf.{all|<interface>.accept_ra = 0`').

Configuring permanent static IPv6 addresses

To configure permanent static IPv6 addresses on a machine which would remain even if network service is restarted:

- First enable IPv6 on machine
- Then edit file '`/etc/sysconfig/network-scripts/ifcfg-eth<n>`' and add lines
IPV6ADDR=`<ipv6_address1_without_quotes>`
IPV6ADDR_SECONDARIES="`<ipv6_address2> <ipv6_address3> <ipv6_address4>`"

Note that multiple IPv6 addresses can be defined in single IPV6ADDR_SECONDARIES option each separated from one another using space.

Disabling auto-configuration of addresses via router discovery/advertisements

General IPv6 node configuration and commands

By default nodes learn prefixes through router solicitation and advertisement messages and configure an address for each prefix learned via such messages. This also allows hosts to configure routes through which they can send packets to nodes which are not on same link.

Disabling auto-configuration on all interfaces

To disable auto-configuration on complete node (all interfaces), edit file `/etc/sysconfig/network` and add option

```
IPV6_AUTOCONF=no
```

We can also use `sysctl` to disable auto-configuration using:

```
sysctl net.ipv6.conf.all.autoconf=0
```

Disabling auto-configuration on particular interface

To disable auto-configuration on particular interface, edit interface configuration file `/etc/sysconfig/network-scripts/ifcfg-eth<n>` and add line:

```
IPV6_AUTOCONF=no
```

We can also disable auto-configuration on particular interface using `sysctl` as:

```
sysctl net.ipv6.conf.<interface_name>.autoconf=0
```

Basic network service test commands

Pinging hosts using their link-local addresses

To ping a host using its link-local address we can use:

```
ping6 -I eth<n> <link_local_address_of_remote_node>
```

Note that we need to specify interface to be used as node will have one IPv6 link local address (`fe80::/10`) on each interface. Hence there is no way for node to know which interface should be used to send ping requests as it has

IP address in that range on all interfaces.

Pinging all nodes or all routers multicast address

To ping all nodes on a link we can use:

```
ping6 -I eth<n> ff02::1
```

To ping all nodes on a routers we can use:

```
ping6 -I eth<n> ff02::2
```

Note that same as in case of pinging link local addresses, here also we need to explicitly specify which interface should be used to send the ping request to multicast addresses as multicast requests can be sent via any interface and there wont be any default route configured to send multicasts.

Informational commands

Seeing neighbour hardware address cache

We can see learned MAC addresses through ICMPv6 using command:

```
ip neigh show
```

This command will show MAC addresses learned via both IPv4 and IPv6. To just see mac addresses learned via IPv4 we can use:

```
arp -a -n
```

and to see MAC addresses learned via IPv6 we can use

```
ip -6 neigh show
```

Seeing IPv6 address valid and preferred lifetimes

We can see list of IP addresses assigned to a node (or to a particular interface of a node) using

```
ip addr show [<interface>]
```

instead of using 'ifconfig [<interface>]'. The advantage of using 'ip addr show' command over 'ifconfig' command is that it will also show valid and preferred lifetime of all IPv6 addresses assigned to the node (or particular interface of node) which is not available in 'ifconfig' output.

Seeing IPv6 routing table

We can use command

```
netstat -vnr -A inet6
```

to see IPv6 routing table. If we want to see both IPv4 and IPv6 routing tables then we can use

```
netstat -nvr -A inet6,inet
```

command.

- Note that 'route -n' command does not shows IPv6 routes.

Resolving IPv6 addresses using nslookup

By default 'nslookup' only sends and accepts IPv4 DNS queries and replies. In order to make 'nslookup' also request and process IPv6 queries and responses we can use:

```
nslookup -type=any <domain_name> [<DNS_server_IP>]
```

format so that even IPv6 responses are displayed.