

Connecting to the IPv6 Internet with Linux

Ibrahim Haddad

In previous articles published on the O'Reilly Network, we explained the IPv6 protocol, demonstrated how to support IPv6 on your Linux machine, and how to connect your Linux machine to the IPv6 Internet. In this article, we go a bit further and demonstrate how you can connect your home or work network to the IPv6 Internet (also known as the 6bone) using the Freenet6 Tunnel Server Protocol (TSP) and a Linux router machine.

Preparatory Work

Our goal is to connect multiple machines sitting on the same network to the IPv6 Internet using the Freenet6 free service, as shown in Figure 1. In such a scenario, you can connect a full network to the IPv6 Internet using the Freenet6 service and a machine that acts as an IPv6 route to provide router advertisement for the other hosts on your network.

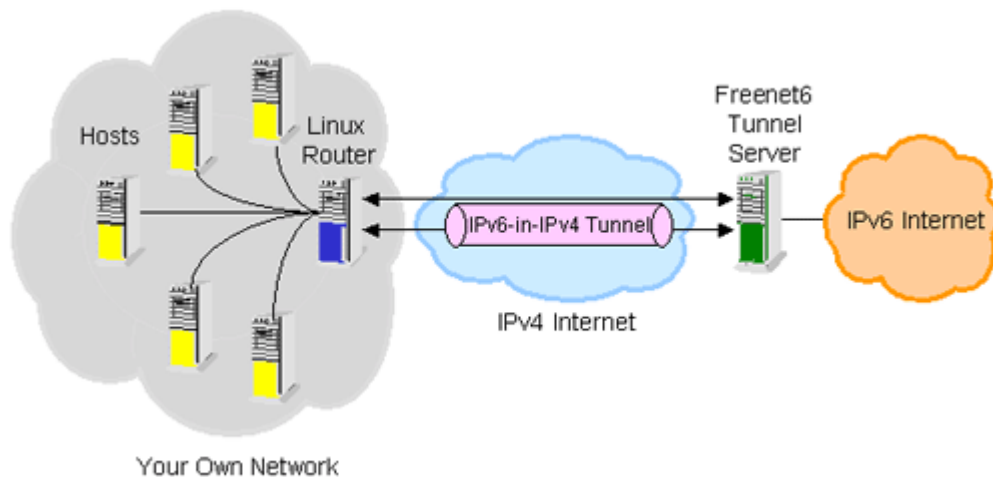


Figure 1. The Target Setup

For the purpose of this article, we will assume that the router is a Linux machine. The hosts can be running any operating system.

Checklist for the Linux Router

The Linux Router in Figure 1 is a Linux machine that will act as a router advertisement server that will allow your hosts to receive their IPv6 networking configurations and connect to the IPv6 Internet using tunneling techniques. This article will explain how to configure this machine to support IPv6, connect to the Freenet6 Tunnel server, and to run the router advertisement daemon that will allow automatic configurations for all hosts on your home or office network.

Checklist for the Hosts

The hosts sitting on your home or office network will receive their IPv6 network configurations from the Linux router. These machines can be running any operating system that supports IPv6 (make sure to enable IPv6 support on these machines). If these machines are Linux hosts, see the previous article that explains how to support IPv6 on Linux either as a loadable module or as a built-in kernel feature.

IPv6 Autoconfiguration

Network-protocol designers recognize the difficulty of installing and configuring TCP/IP networks. Over the years, they've come up with solutions to overcome these pitfalls. One of IPv6's useful features is its autoconfiguration ability. It does not require a stateful configuration protocol, such as DHCP or DHCPv6, which supports IPv6. Hosts can also use router discovery to determine the

Connecting to the IPv6 Internet with Linux

Ibrahim Haddad

addresses of routers and other configuration parameters. The router advertisement message also includes an indication of whether the host should use a stateful address configuration protocol.

There are two types of autoconfiguration. Stateless configuration requires the receipt of router advertisement messages. These messages include stateless address prefixes and preclude the use of a stateful address configuration protocol.

Stateful configuration uses a stateful address configuration protocol, such as DHCPv6, to obtain addresses and other configuration options. A host uses stateful address configuration when it receives router advertisement messages that do not include address prefixes and require that the host use a stateful address configuration protocol. A host will also use a stateful address configuration protocol when there are no routers present on the local link.

By default, an IPv6 host can configure a link-local address for each interface.

The main idea with autoconfiguration is that IPv6 has significant features that enhance the ability of a host to configure itself. The designers intended that a host should be able to discover automatically all of the information it needs to connect to the Internet without human intervention.

Autoconfiguration requires routers of the local network to run a program that answers the autoconfiguration requests of the hosts. On Linux, this program is radvd (Router ADVERTISEMENT Daemon). This daemon listens to router solicitations and answers with router advertisement. These router advertisements contain host configuration information such as address prefixes, the MTU of the link, and information about default routers. Furthermore, the daemon sends unsolicited router advertisements from time to time.

Using Autoconfiguration to Configure Multiple Hosts

To connect multiple hosts to the IPv6 Internet using the Freenet6 service, you need a Linux host to provide router advertisement for the rest of the hosts on your network. Freenet6 service will provide you with a /48 IPv6 prefix, which allows a site to deploy up to 2^{16} subnets. Each subnet can handle 2^{64} nodes. The number of Unicast IPv6 addresses you can use with only one /48 prefix is very large: 65,535 subnets * 18 446 744 073 709 551 616.

Configuring the Linux Router

To configure the Linux router:

1. Ensure that the router meets the required conditions.
2. Install and configure the TSP package.
3. Install and configure the router advertisement daemon.

The previous article ("Connecting to the IPv6 Internet") covered steps 1 and 2. Therefore, we will focus on installing and configuring the router advertisement daemon. Instead of repeating a lot of the common installation steps, I suggest you follow the tutorial and then continue with the instructions below.

The Linux router must have two Ethernet ports (since it will be acting as a router) and have IP forwarding enabled in the Linux kernel.

First, add special parameters to your tspec.conf file. After adding your userid and password, you need to add three other parameters needed to request a /48 prefix from the tunnel server:

Connecting to the IPv6 Internet with Linux

Ibrahim Haddad

```
host_type=router
prefixlen=48
if_prefix=YOUR_NETWORK_INTERFACE
```

The `host_type` indicates that the host is a router and not a standalone host. The `prefixlen` field indicates that you are requesting a /48 prefix. The `YOUR_NETWORK_INTERFACE` field identifies the network interface to use for IPv6 routing. In Linux, this takes the form of `ethx` where `x` is either 0 or 1 (in the case when you have two Ethernet ports).

This configuration will enable IPv6 forwarding between this interface (`ethx`) and the configured tunnel (IPv6-over-IPv4). It will also activate IPv6 router advertisements to allow hosts connected there to autoconfigure their IPv6 addresses.

Once you finish editing the TSP configuration file, you can run the TSP client.

```
#!/tspc -vf tspc.conf
```

Part of the output will include:

```
TSP_TUNNEL_PREFIXLEN    128
TSP_PREFIX              3ffe:0b80:1de0
TSP_PREFIXLEN          48
```

This information is important: it includes the prefix your Linux router will broadcast to the hosts on your network. Edit the `radvd.conf` file to reflect these values.

Please note that different Linux routers configured with Freenet6 TSP have different prefixes. For your own installation, replace the `TSP_PREFIX` value with the real value you receive from Freenet6.

After completing the tunnel configuration and running the tunnel client program, the Linux machine will be connected to the IPv6 Internet using a Freenet6 tunnel. However, it will be assigned with a router prefix and a block of addresses to broadcast to the hosts connected to the machine. That brings us to the next step.

Installing the Router Advertisement Daemon

To install the router advertisement daemon, first download the latest stable release from the `radvd` home page. If you prefer to download source and/or RPM packages, these are available too. For my test machine, I downloaded `radvd-0.7.2.tar.gz` into `/usr/src`.

```
# cd /usr/src
# tar -xzvf radvd-0.7.2.tar.gz
# cd radvd-0.7.2
# ./configure
# make
# make install
```

This will install `radvd` binary in `/usr/local/sbin` and `radvd.conf` in `/etc`. In addition, the installation will create a user called `radvd` with UID 75. (For more information, read `/etc/sysconfig/radvd`.)

Configuring the router advertisement daemon takes place through the `/etc/radvd.conf` file. You need to configure the interface section by adding the interface-specific and prefix options. Typically, you will

Connecting to the IPv6 Internet with Linux

Ibrahim Haddad

only need to use the AdvSendAdvert option, which enables sending of router advertisement messages.

Here's a sample /etc/radvd.conf file. Replace all italic variables with their real values, depending on which Ethernet interface you use to connect to the network where your hosts are, and also depending on the information you have received from the tunnel server upon establishing the tunnel.

```
interface ethn
{
    AdvSendAdvert on;

    # These settings cause advertisements to be sent
    # every 3-10 seconds.

    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 10;

    prefix TSP_PREFIX/TSP_PREFIXLEN
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr off;
    };
};
```

Once you complete editing the configuration file, run the router advertisement daemon:

```
# radvd -C /etc/radvd.conf
```

The daemon will start. The hosts on the routed network will now receive the router advertisement sent by this daemon. They should then autoconfigure their interfaces for IPv6 addresses.

Your home or office network is now connected to the IPv6 Internet.

Conclusion

In this article, we demonstrated how to connect a full network of machines to the IPv6 Internet using the Freenet6 service through a Linux machine acting as a router. In an upcoming article, we will present how to provide a web service over IPv6 using the setup we built in this article. Stay tuned!