

Getting started with ipv6

It's actually easier than you might think. :-)

Theory

The problem we face is that our ISP's don't generally have IPV6 routing set up. If they won't carry our packets, just how are we going to carry on a conversation with an IPv6 machine somewhere?

The approach we'll take is to *tunnel* our ipv6 traffic inside ipv4 packets to a site on the Internet that *does* have the ability to move our ipv6 traffic for us. Our ISP just sees normal ipv4 traffic, nothing more. We see a full IPV6 connections for all our machines.

Requirements

For this example, I'll show you how to work with www.freenet6.net, a *tunnel broker*. They have kindly offered to accept connections from people like us, people whose ISP's don't know how to talk ipv6. When we send them an ipv6 packet stuffed inside an ipv4 header, they unwrap that packet and send it on. When the response comes back from the ipv6 server, they wrap that response back up in an ipv4 header and ship the response back to the real ipv4 address of our gateway.

For this to work, your ipv6 gateway box needs at least one real ipv4 address. It also needs sufficient access through any firewalls on the way out to the Internet to allow outbound connections to tcp ports 3653 and 4343, and both outbound and inbound protocol 41 packets. The former handle authentication with the freenet6 site, the latter actually carry the ipv6 traffic inside ipv4 headers. The traffic on these ports and protocol 41 will only be to machines on the 206.123.31.0/24 network if you want to lock down traffic a bit on your firewall.

Once this is set up, the gateway box will provide a huge number of addresses to the machines behind it. Those machines can run any operating system that supports IPV6. In fact, the gateway itself can run a range of operating systems, although the step-by-step instructions I'll provide show the setup on a linux box. The [Freenet6 download page](#) lists support for Windows, FreeBSD, Linux, Solaris, Openbsd, Netbsd, and Cisco routers.

Software install

In addition to providing the tunnel service, www.freenet6.net also provides a free program that authenticates you to freenet6 and sets up the routing. First, install the radvd routing daemon. If your distribution doesn't have it, check rpmfind.net or the [software's home site](#). This program will hand out addresses to your client machines, much like a dhcp server did on ipv4. It only needs to be installed on your gateway. You don't need to do anything about configuring it or running it; this will be handled automatically.

The other piece of software to install (only on the gateway) is the freenet6 client. This can be found at the [Freenet6 download area](#). I installed the i386 rpm by pulling it down to my hard drive, then running:

```
rpm -Uvh freenet6-0.9.7-1.i386.rpm
```

Don't worry too much if the versions seem a bit whacky; they advertise version 1.0.0 on the front page of the site, but it doesn't seem to be actually available for download. No problem, 0.9.7 seems to work just fine.

The next step is to [apply for an account](#). Pick a username of at least 8 characters, and enter it and your contact email address. The password to use will be mailed to that account.

Configuration

Now you need to edit the file `/usr/local/tsp/bin/tspc.conf` . This file holds your network settings. Here are the lines you need to change, and a quick description.

`client_v4=ip.address`

The IPv4 address to use here is the real IPV4 address assigned to the outside interface of your gateway, for example, 12.13.14.15 . If this gateway box changes IPV4 addresses from time to time, try using the value "auto" instead.

`userid=TheUserIdYouPickedAtFreenet6`

`passwd=ThePasswordTheyEmailedBack`

`server=tsps2.freenet6.net`

If this doesn't work, try `tsps1.freenet6.net` .

`if_tunnel=sit2`

This should be a sit device that does not yet exist (do a `less /proc/net/dev` and make sure it's not there. If it already exists, the software setup will come back with odd errors.

`host_type=router`

This box will be the ipv6 router for your clients, so we literally use the word "router".

`prefixlen=48`

Freenet6 will provide you with a "/48" block of addresses. That's 2^{80} addresses. Care to retrieve your jaw from the floor now? Yes, that's right – you get far, far, far more addresses than everything that exists in the ipv4 address space.

`if_prefix=eth2`

In place of eth2, put the network interface on which your client machines can be found.

Start and test the tunnel

Now start up the ipv6 tunnel with:

```
/etc/rc.d/init.d/freenet start
```

If it gives you errors, see if you can figure out what the problem is. Otherwise, check that this machine has a live ipv6 connection with the command:

```
ping6 -n www.ipv6.org
```

It should provide output similar to:

```
[root@gateway bin]# ping6 -n www.ipv6.org
PING www.ipv6.org(2001:6b0:1:ea:a00:20ff:fe8f:708f) from zzzzzzz : 56 data bytes
64 bytes from 2001:6b0:1:ea:a00:20ff:fe8f:708f: icmp_seq=1 ttl=249 time=616 ms
64 bytes from 2001:6b0:1:ea:a00:20ff:fe8f:708f: icmp_seq=2 ttl=249 time=507 ms
64 bytes from 2001:6b0:1:ea:a00:20ff:fe8f:708f: icmp_seq=3 ttl=249 time=650 ms
64 bytes from 2001:6b0:1:ea:a00:20ff:fe8f:708f: icmp_seq=4 ttl=249 time=561 ms
```

While that's running, you can watch the tunneled packets with:

```
tcpdump -i eth1 -qtnp 'net 206.123.31'
```

which will show the packets heading to and coming from the tunnel servers at 206.123.31.114 and 206.123.31.115 . Substitute the outside network interface of your gateway for eth1.

Getting started with ipv6

To see what network block you received, try:

```
ip -f inet6 route show
```

Which should return, among others, lines similar to:

```
[root@gateway bin]# ip -f inet6 route show
3ffe:bc0:zzzz:1::/64 dev tap1 metric 256 mtu 1500 advmss 1440
3ffe:bc0:zzzz::/48 dev tap1 metric 1 mtu 1500 advmss 1440
[snip]
default dev sit2 metric 1 mtu 1480 advmss 1420
```

That /48 is the block you now have for your machines. `ifconfig` will show what address is assigned to your local interfaces; you're interested in the lines with "Scope:Global" as these are the addresses you can actually use for IPV6 Internet conversations. "Scope:Link" addresses are only for local use on your network cable, similar to RFC1918 addresses in IPV4.

Client setup and testing

As long as the client computer has the ability to autoconfigure an ipv6 address and routing, you're done! The same:

```
ping6 -n www.ipv6.org
```

test should show a live ping to the remote end. The same `tcpdump` above should show the tunneled packets going back and forth.

Additional things to try

tracpath6 www.ipv6.org

traceroute6 www.ipv6.org

Check the route packets take to the destination

lynx <http://www.ipv6.org>

This text-based web browser knows how to talk ipv6. The web server at that site will report back whether you've come in on an IPV6 or IPV4 address, and what address that is – another good way to tell if your IPV6 connection works.

Additional IPV6 applications.

The Linux IPV6 HOWTO.

ping6 -I eth0 ff02::1

Ping all hosts on the eth1 cable, no matter what IPV6 address they would normally use.

Credits

Thanks to all the people who have worked to make IPV6 a reality and the people that implemented IPV6 in Linux. Also, special thanks to the Freenet6 team at Hexago for providing this public service at no cost. Thanks also to David Coulson for putting up with some rather silly questions.

Bill Stearns wrote this text.

William is an Open-Source developer, enthusiast, and advocate from New Hampshire, USA. His day job at SANS pays him to work on network security and Linux projects.

Getting started with ipv6

This document is Copyright 2003, William Stearns <wstearns@pobox.com>.

Last updated 12/2/2003.