Benedikt Stockebrand

# IPv6 in Practice

## A Unixer's Guide to the Next Generation Internet

## **Fedora Core 6 Supplement**

Version 1.0 (2007-02-03)

137 **p. 352**

***Fedora Core 6*** *Like Solaris 10, Fedora Core 6 doesn't provide an initial* `named.conf`*. The boot scripts expect it in* `/etc`*, so we can proceed as with Solaris 10 and copy the sample configuration from page 351 without the* `pid-file` *line there and create the directory* `/etc/named/zonedata`*. Fedora Core 6 runs the* `named` *daemon as user* `named` *and group* `named`*, so we should* `chown` *the directory to* `named:named`*.*

*Finally we run*

```
# chkconfig named on
```

*to enable the boot script.*

138 **p. 352/353**

***Fedora Core 6*** *If we don't want to reboot the name server, then running*

```
# /etc/init.d/named start
```

*will start the name server right away.*

139 **p. 354**

***Fedora Core 6*** *According to the* `directory` *statement in our configuration file, all zone files are expected to reside in* `/var/named/zonedata`*.*

140 **p. 355**

***Fedora Core 6*** *Simply running*

```
# /etc/init.d/named restart
```

*will restart the name server.*

123 **p. 289–291**

***Fedora Core 6*** *The distribution includes two packages* `dhcpv6` *and* `dhcpv6_client`*; they hold the entire implementation and the client part only of the other DHCPv6 implementation available for Linux, respectively.*

*Besides the limited documentation the packages shipping with Fedora Core 6 exhibit a crippling series of shortcomings: They are based on drafts dating back as far as 2003, rather than the official RFCs released since then; the implementation is apparently abandoned and surely not up to current specs; finally, the DHCPv6 server part doesn't even start with the sample configuration from the man page.*

*The obvious fallback option, building and installing Dibbler, doesn't work either: Dibbler doesn't compile on Fedora Core 6.*

*In short, using DHCPv6 with Fedora Core 6 is effectively infeasible.*

133 **p. 304**

***Fedora Core 6*** *The core installation already contains everything we need.*

134 **p. 317**

***Fedora Core 6*** *The* `ip6tables` *version shipping with Fedora Core 6 supports the* `ah` *and* `esp` *extensions that allow filtering by SPI.*

135 **p. 324**

***Fedora Core 6*** *The MIPL project mentioned with Debian Sarge can also be used with Fedora Core 6.*

136 **p. 351**

***Fedora Core 6*** *The name server proper is available from the* `bind` *package on the installation medium. Client-side tools are already installed with the base installation.*



1 **p. VIII**

***Fedora Core 6*** *This Linux distribution is similar to Debian Sarge, but it uses different configuration files than Debian.*

2 **p. X**

***Fedora Core 6*** *is the community-maintained branch of RedHat Linux.*

3 **p. 11**

***Fedora Core 6*** *This distribution installs kernel version 2.6.18, which works fine.*

4 **p. 11**

***Fedora Core 6*** *Even a minimal installation already brings everything we need.*

5 **p. 11/12**

***Fedora Core 6*** *The command* `makewhatis` *without any arguments rebuilds the Whatis index.*

6 **p. 14**

***Fedora Core 6*** *IPv6 support is enabled by default, so* `ifconfig -a` *should show an IPv6 address "*`::1`*" on the loopback interface.*

7 **p. 14**

***Fedora Core 6*** *Again, IPv6 support is already enabled by default.*

9 p. 15

***Fedora Core 6*** *As with Debian,*

```
sysctl -a | egrep ^net.ipv6
```

*does the job.*

10 p. 16/17

***Fedora Core 6*** *uses the same* `ip6tables` *filter as Debian Sarge.*

11 p. 17–19

***Fedora Core 6*** *Different than Debian, Fedora Core 6 already installs a basic* `ip6tables` *configuration by default. The command*

```
# chkconfig ip6tables off
```

*disables it again; similarly,*

```
# chkconfig ip6tables on
```

*re-enables it. The default filter configuration is troublesome, so it is generally best to turn the filter off for now.*

12 p. 24

***Fedora Core 6*** *The distribution includes an RPM package that contains* `ipv6calc`*, version 0.61.*

15 p. 26

***Fedora Core 6*** *Like Debian Sarge, the* `ping6` *command supports the* `-I` *option.*



117 p. 278

***Fedora Core 6*** *The primary development platform for* `mrd` *is Debian, so we could get the sources from* `http://hng.av.it.pt/mrd6/download/mrd6-*.tar.gz` *and compile them ourselves. Unfortunately, at the time of this writing (February 2007, referring to* `mrd6-0.9.5`*) it seems like* `mrd` *uses some deprecated header file macros that have already been removed from the files shipped with Fedora Core 6. It doesn't compile, so Fedora Core 6 is effectively useless for multicast routing purposes.*

118 p. 279

***Fedora Core 6*** *If* `mrd` *compiled on Fedora Core 6, then the same considerations as for Debian Sarge should apply.*

119 p. 279/280

***Fedora Core 6*** *If* `mrd` *compiled on Fedora Core 6, then the same considerations as for Debian Sarge should apply.*

120 p. 280/281

***Fedora Core 6*** *If* `mrd` *compiled on Fedora Core 6, then the same considerations as for Debian Sarge should apply.*

121 p. 281

***Fedora Core 6*** *If* `mrd` *compiled on Fedora Core 6, then the same considerations as for Debian Sarge should apply.*

122 p. 281

***Fedora Core 6*** *If* `mrd` *compiled on Fedora Core 6, then the same considerations as for Debian Sarge should apply.*

```
# chkconfig --add zebra
# chkconfig --add ripngd
# chkconfig zebra on
# chkconfig ripngd on
```

Finally, if we use the standard `ip6tables` filter rules, then we need to enable access to the administrative interface in the filter rules.

110 p. 248

***Fedora Core 6*** *We enable the OSPFv3 and disable the RIPng daemon with the commands*

```
# chkconfig --add ospf6d
# chkconfig ospf6d on
# chkconfig ripngd off
```

112 p. 264/265

***Fedora Core 6*** *Both* `netstat` *and* `ip` *behave no different on Fedora Core 6 than on Debian Sarge.*

113 p. 266

***Fedora Core 6*** *Different than the Debian Sarge* `ping6` *command, the* `ping6` *command shipping with Fedora Core 6 uses a TTL value of 64.*

115 p. 267

***Fedora Core 6*** *The standard kernel here also uses MLDv2 packets.*

116 p. 271/272

***Fedora Core 6*** *According to updated information on the* `mcast-tools` *web page, its Linux support is broken. Again we may consider using* `ecmh` *as a workaround.*



16 p. 36

***Fedora Core 6*** *Again, Fedora Core 6 and Debian Sarge don't differ; both use the same* `ifconfig` *and* `ip` *commands.*

17 p. 37/38

***Fedora Core 6*** *Again, Fedora Core 6 and Debian Sarge don't differ; both use the same* `ifconfig` *and* `ip` *commands.*

18 p. 38

***Fedora Core 6*** *Again, Fedora Core 6 and Debian Sarge don't differ; both use the same* `ifconfig` *and* `ip` *commands.*

19 p.39

***Fedora Core 6*** *There is no manpage available to explain how to configure IPv6 on Fedora Core 6, but according to the documentation hidden in* `/usr/share/doc/initscripts*/sysconfig.txt` *we must first add a line*

/etc/sysconfig/network

```
NETWORKING_IPV6=yes
```

*in* `/etc/sysconfig/network` *to make the boot scripts configure IPv6. In* `/etc/sysconfig/network-scripts/ifcfg-eth0` *we can then pass the actual configuration following the pattern*

/etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
IPV6INIT=yes
IPV6ADDR=2001:db8:fedc:cdef::4
IPV6ADDR_SECONDARIES="2001:db8:fedc:abcd::4"
```

*The* `BOOTPROTO` *line tells the boot scripts that the interface will be configured statically, the* `IPV6INIT` *line to configure IPv6 on it, the* `IPV6ADDR`

*line the "primary" IPv6 address (even though "primary" addresses are realy an IPv4-only concept) and the* `IPV6_SECONDARIES` *all additional addresses. To enable the configuration it takes either a quick reboot or running*

```
# ifup eth0
```

*for all affected interfaces.*

20 p. 41

***Fedora Core 6*** `ip -6 neigh show` *(as with Debian Sarge).*

21 p. 47/48

***Fedora Core 6*** *First we must set up the router as a router, i.e. a device that forwards packets. To do so we add the lines*

`//etc/sysconfig/network`

```
NETWORKING_IPV6=yes          || Should already be there
IPV6FORWARDING=yes           || Enable forwarding
IPV6_ROUTER=yes              || Disable autoconfiguration
```

*The somewhat inconsistent variable naming tends to cause trouble because this file is simply read as a shell script; if the configuration doesn't work, then check that there are no missing or superfluous underscores in the variable names.*

*Next we need to install the* `radvd` *package from the distribution. Then we must write a* `/etc/radvd.conf` *configuration file; we can copy the Debian Sarge example here. Now*

```
# chkconfig radvd on
```

*enables the router advertisement daemon. Finally a quick reboot is the easiest way to bring the router up.*

22 p. 49

***Fedora Core 6*** *If we just enable IPv6 as in section 4.1.2, but don't assign any static addresses, then the host comes up correctly, configuring its addresses using autoconfiguration. So again we need a line*



102 p. 227/228

***Fedora Core 6*** *Yet again, since Fedora Core 6 uses the same* `radvd` *router advertisement daemon as Debian Sarge, the same considerations as for Debian Sarge apply.*

103 p. 228-230

***Fedora Core 6*** *Still Fedora Core 6 uses the same* `radvd` *as Debian Sarge does, so no surprise, the Debian Sarge configuration also works with Fedora Core 6.*

104 p. 236

***Fedora Core 6*** *As we have already seen in section 7.4 (p. 108ff), Fedora Core 6 ships with a* `quagga` *package.*

105 p. 236/237

***Fedora Core 6*** *Like Debian Sarge, Fedora Core 6 expects all Quagga configuration files in* `/etc/quagga`.

106 p. 237/238

***Fedora Core 6*** *Again, we must enable IPv6 packet forwarding by adding the lines*

`/etc/sysconfig/network`

```
IPV6FORWARDING=yes           || Enable forwarding
IPV6_ROUTER=yes              || Disable autoconfiguration
```

*in* `/etc/sysconfig/network`*, making sure that we get the inconsistent use of underscores right. Then we edit* `/etc/sysconfig/quagga` *and replace all occurences of "*`-A 127.0.0.1`*" with "*`-A ::1`*"; this ensures that we can use the administrative interface through the IPv6 loopback interface. Next we enable the* `zebra` *and* `ripngd` *daemons with the commands*

/sbin/ifup-local

```
#! /bin/sh

/sbin/sysctl -w net.ipv6.conf.$1.use_tempaddr=1
/sbin/sysctl -w net.ipv6.conf.$1.temp_prefered_lft=600
/sbin/sysctl -w net.ipv6.conf.$1.temp_valid_lft=7200
```

*exists, then it configures all interfaces accordingly whenever they are brought up. If we want to use different settings for different interfaces, then we can fairly easily extend the script, taking the `$1` argument into account.*

98 p. 220

***Fedora Core 6*** *If we change the first `sysctl` invocation in our custom `/sbin/ifup-local` to set the variable to `2` instead of `1`, then we get the same results as with Debian Sarge.*

99 p. 223

***Fedora Core 6*** *As with Debian Sarge, there is no documented way to manipulate the default policy table.*

100 p. 224/225

***Fedora Core 6*** *Since Fedora Core 6 uses the same `radvd` router advertisement daemon as Debian Sarge, the same considerations as for Debian Sarge apply.*

101 p. 225/226

***Fedora Core 6*** *Again, since Fedora Core 6 uses the same `radvd` router advertisement daemon as Debian Sarge, the same considerations as for Debian Sarge apply.*



/etc/sysconfig/network

```
NETWORKING_IPV6=yes
```

*in `/etc/sysconfig/network` and the lines the lines*

/etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
IPV6INIT=yes
```

*in `/etc/sysconfig/network-scripts/ifcfg-eth0` for every interface we want to use.*

23 p. 50/51

***Fedora Core 6*** *By default Fedora Core 6 mixes static and autoconfigured addresses. Setting the variable `IPV6_AUTOCONF` to `no` either globally in `/etc/sysconfig/network` or on a per-interface basis in `/etc/sysconfig/network-scripts/ifcfg-eth0` turns off autoconfiguration.*

24 p. 52

***Fedora Core 6*** *As with Debian Sarge, the command to use here is `ip addr show`.*

25 p. 55

***Fedora Core 6*** *Even though Fedora Core 6 uses a later kernel than Debian Sarge, the basic limitations are the same; only the `REJECT` target seems to work as expected.*

26 p. 56

***Fedora Core 6*** *Again, since Fedora Core 6 and Debian Sarge use the same packet filter implementation, they also share its limitations.*

27 p. 56/57

***Fedora Core 6*** *Yet again, since Fedora Core 6 and Debian Sarge use the same packet filter implementation, they also share its limitations.*

28 p. 57–63

***Fedora Core 6*** *We can mostly use the Debian Sarge script for the packet filter configuration. There are however differences:*

*Fedora Core 6 supports the* `REJECT` *target, so we should substitute it for the* `DROP` *target. Even better, we might write a* `reject ()` *function and use it for those* `drop` *invocations that "deserve" a proper error notification.*

*Something Fedora Core 6 (at least the x64 version) doesn't support are the* `ipv6header` *extensions, so we have to remove the* `SANITIZE` *chain for now.*

*Finally, Fedora Core 6 handles filters differently than Debian Sarge: It assumes that the packet filter configuration to load at boot is stored in* `/etc/sysconfig/ip6tables` *using the* `ip6tables-save` *command, so it is easiest to first run our script manually to set up the filter rules and afterwards invoke*

```
# /etc/init.d/ip6tables save
```

*to save the rules for the next reboot.*

*As a matter of personal preference, I have a dislike for this approach because the configuration saved with* `ip6tables-save` *doesn't preserve any comments. Packet filter configurations are notoriously difficult to understand; I consider comments an invaluable tool to help keeping them correct even when things change.*

30 p. 68

***Fedora Core 6*** *The* `telnet` *client here doesn't support the* `-4` *or* `-6` *option. It does show the same misbehaviour as Debian Sarge: It only tries the first IPv6 address it finds, not all of them.*



`IPV6_RADVD_TRIGGER_ACTION` *to either* `reload` *or* `restart` *will run the* `radvd` *init script with this parameter.*

*If we use our own init scripts for PPP, then we need to provide this logic ourselves or use the* `radvd` *configuration shown for Debian Sarge.*

93 p. 207

***Fedora Core 6*** *Since Fedora Core 6 uses the same kernel PPP implementation as Debian Sarge and Solaris 10, it also suffers the same limitation.*

94 p. 207

***Fedora Core 6*** *Apparently there are no hooks to update the packet filter configuration from the* `sysconfig` *scripts. This is yet another reason to ignore them and use handcrafted scripts that put the necessary packet filter configuration logic in* `/etc/ppp/ipv6-up` *and* `-down`*.*

95 p. 213

***Fedora Core 6*** *Most unsurprisingly, the same Quagga configuration that works with Debian Sarge also applies to Fedora Core 6.*

96 p. 215/216

***Fedora Core 6*** *Unsurprisingly, Fedora Core 6 uses the same default behaviour and* `sysctl` *variable to control the use of mapped addresses.*

97 p. 218/219

***Fedora Core 6*** *We can use the same* `sysctl` *commands shown for Debian Sarge to handle temporary addresses. The* `sysconfig` *framework offers a different hooks to run scripts after an interface has come up, though: If a command* `/sbin/ifup-local` *exists, then it is run whenever an interface is brought up, the name of the interface being passed as its sole argument. So the script*

**89 p. 200–202**

***Fedora Core 6*** *By default, Fedora Core 6 expects the PPP configuration in* `/etc/ppp/options` *rather than* `/etc/ppp/options.server`*. So if we copy the configuration from the Debian Sarge example there, PPP works as expected.*

*Fedora Core 6 doesn't show the problem with the* `persist` *option. So we can omit the last* `nodetach` *line and add the* `persist` *option instead to make the PPP daemon run forever, re-establishing a link every time it fails for whatever reason.*

*We can start the PPP daemon manually as with Debian Sarge. Making the boot scripts start the PPP daemon automatically turns out to be tedious: The standard* `sysconfig` *interface makes a number of assumptions that don't apply to our scenario, namely the use of a modem, ISDN line or ADSL/PPP over Ethernet connection. In summary, for our scenario it is easiest to use the* `persist` *option and write a short* `init` *script that brings up the interface and shuts it down again properly.*

*Alternatively, leaving* `/etc/ppp/options` *alone and configuring PPP through the* `sysutils` *configurations is reasonable if we actually want to use a modem, ISDN line or ADSL connection. In that case we just set up PPP as we would for IPv4 and then add the IPv6-specific configuration parameters as we would for an ordinary Ethernet interface.*

**90 p. 202–204**

***Fedora Core 6*** *Since Fedora Core 6 puts some distribution-specific logic into the standard* `/etc/ppp/ipv6-up` *and* `-down` *scripts, again we have two options: Either do it the Fedora Way, configuring the interface through* `sysconfig` *configuraiton files or by replacing the scripts with our own. In the latter case the Solaris* `/etc/ppp/ipv6-up` *and* `-down` *scripts are the preferred starting point since the Debian Sarge scripts are also distribution specific.*

**92 p. 205/206**

***Fedora Core 6*** *The* `sysconfig` *scripts provide a hook to signal the* `radvd` *if an interface has changed. Setting the configuration variable*



**31 p. 68**

***Fedora Core 6*** *As with Debian Sarge, both* `ping6` *and* `traceroute6` *do a reverse lookup.*

**32 p. 69**

***Fedora Core 6*** *Depending on the initial installation configuration the* `/etc/nsswitch.conf` *file should contain the necessary*

`/etc/nsswitch.conf`

```
hosts: files dns
```

*line to do DNS lookups.*

**33 p. 78**

***Fedora Core 6*** *As with Debian Sarge, the* `ip6tables` *framework supports a separate* `FORWARD` *chain for packets that are forwarded.*

**34 p. 82**

***Fedora Core 6*** *The only Inetd supported is the* `xinetd`*, which we must install from the installation media.*

**35 p. 83–85**

***Fedora Core 6*** *The configuration files differ from Debian Sarge in two ways: While Debian Sarge configures both the TCP and UDP echo services in a single file, Fedora Core 6 splits them into two files* `/etc/xinetd.d/echo-stream` *and* `/etc/xinetd.d/echo-dgram`*. Additionally, the Fedora Core 6 files contain a plethora of comments, because they are meant to serve as templates for other services. But other than that, we only have to set the* `disable` *and* `flags` *parameters to* `no` *and* `IPv6`*, respectively.*

36 p. 86/87

***Fedora Core 6*** *The* `netstat` *command here doesn't support the* `-4` *and* `-6` *options. Both* `netstat` *and* `lsof` *show the dangerous behaviour explained in the warning on page 87.*

*The* `netcat` *package is capable of port scanning, so running*

```
# nc -z 2001:db8:fedc:abcd::4 1-65535
```

*does a port scan to find all open IPv6 ports.*

*The* `nmap` *command needs an additional* `-6` *option to scan IPv6 for open ports; so we can resort to it, too.*

37 p. 88

***Fedora Core 6*** *Even a minimal installation already installs the SSH daemon and enables it, including IPv6 support.*

38 p. 90

***Fedora Core 6*** *First we install the* `ntp` *package, then set up the configuration in* `/etc/ntp.conf` *and finally enable the service using* `chkconfig`*.*

*Apparently Fedora Core 6 doesn't have the problem mentioned with Debian Sarge on page 90.*

39 p. 91

***Fedora Core 6*** *As with Debian Sarge, Fedora Core 6 doesn't support Syslog over IPv6.*

41 p. 92

***Fedora Core 6*** *First we must install the* `sendmail-cf` *package; with it we can create a custom sendmail configuration. Then we must edit the file* `/etc/mail/sendmail.mc`*. It is an* `m4` *file, so comments start with*



82 p. 186

***Fedora Core 6*** *The same considerations as for Debian Sarge also apply here.*

83 p. 192/193

***Fedora Core 6*** *Again, as long as we configure our packet filter with the Debian-style script, we can also use the rules shown there.*

84 p. 193

***Fedora Core 6*** *Yet again, the same considerations as for Debian Sarge also apply here.*

86 p. 196

***Fedora Core 6*** *No surprise, the* `ip` *command behaves the same as with Debian Sarge.*

87 p. 196/197

***Fedora Core 6*** *And again, the* `ip` *command is the same as with Debian Sarge.*

88 p. 199

***Fedora Core 6*** *Like Debian Sarge, Fedora Core 6 includes Paul Mackerras' kernel PPP in its core installation.*

**80 p. 181/182**

***Fedora Core 6*** *Using the same configuration mechanisms as with Debian Sarge we can configure the tunnel interface temporarily. We may need to load the* `ip_gre` *kernel module first, though. In detail, the commands*

```
# modprobe ip_gre          || May be unnecessary; depends on kernel
# ip -6 tunnel add gre1 mode gre remote 192.0.2.2
# ip link set dev gre1 up
# ip -6 addr add fe80::192.0.2.130 dev gre1
```

*configure the tunnel temporarily. To make the configuration permanent we need a file* `/etc/sysconfig/network-scripts/ifcfg-gre1` *with the contents*

/etc/sysconfig/network-scripts/ifcfg-gre1

```
DEVICE=gre1
BOOTPROTO=none
ONBOOT=yes
IPV6INIT=yes
MY_OUTER_IPADDR=192.0.2.130
MY_INNER_IPADDR=fe80::192.0.2.130
PEER_OUTER_IPADDR=192.0.2.2
TYPE=GRE
```

*So the configuration here takes different configuration variables is slightly more complex than for a simple configured tunnel.*

**81 p. 184**

***Fedora Core 6*** *There are no binary RPMs for OpenVPN and the LZP library shipped with Fedora Core 6. Both compile from the sources without any problems, though.*

*To start the daemon, we must pass it the location of the configuration file, like*

```
# openvpn --config /etc/openvpn.conf
```

*if we keep the configuration in* `/etc/openvpn.conf`*.*



`dnl`*, which stands for "delete to newline". We must comment out the line*

/etc/mail/sendmail.mc

```
DAEMON_OPTIONS(‘Port=smtp,Addr=127.0.0.1, Name=MTA’)dnl
```

*by prepending a* `dnl`*, so* `sendmail` *won't open a socket for the loopback interface only and comment in the line*

/etc/mail/sendmail.mc

```
DAEMON_OPTIONS(‘Name=MTA-v4, Family=inet, Name=MTA-v6, \
    Family=inet6’)
```

*to make it listen on both IPv4 and IPv6 for incoming connections. Afterwards we build the configuration file (*`sendmail.cf`*) from* `sendmail.mc` *by running*

```
# make -C /etc/mail
```

*Running*

```
# /etc/init.d/sendmail restart
```

*should restart the service, making it open a single socket for both IPv4 and IPv6.*

**42 p. 94**

***Fedora Core 6*** *All browsers and Wget support IPv6 here; lynx again doesn't support URLs with directly specified IPv6 addresses, but accepts URLs with DNS names that resolve to IPv6 addresses.*

**43 p. 94**

***Fedora Core 6*** *The Apache2 package is called* `httpd-2.*` *here. The* `httpd.conf` *file resides in* `/etc/httpd/conf/`*.*

**44 p. 95**

***Fedora Core 6*** *Like Debian Sarge, the* `httpd` *daemon is built with IPv4-mapped IPv6 addresses enabled.*

45 **p. 96**

***Fedora Core 6*** *The proxy module works out of the box.*

47 **p. 96**

***Fedora Core 6*** *There is no RPM for* `ffproxy` *available, so we need to build it from sources ourselves. Additionally, we need to write a simple init script; the* `ffproxy` *command needs the additional option* `-b` *when started.*

48 **p. 97**

***Fedora Core 6*** *As with Debian Sarge, there is no IPv6 support in the Linux portmapper and NFS over IPv6 doesn't work.*

52 **p. 100**

***Fedora Core 6*** *If we use the configuration script from Debian Sarge, then we can also apply the updates shown here to our Fedora script.*

53 **p. 101**

***Fedora Core 6*** *If we use the configuration script from Debian Sarge, then we can also apply the updates shown here to our Fedora script.*

54 **p. 101**

***Fedora Core 6*** *The* `multiport` *extension mentioned for Debian Sarge is also available with Fedora Core 6.*



70 **p. 164/165**

***Fedora Core 6*** *As usual, using the same command as for Debian Sare to set up a temporary route also works with Fedora Core 6. To make the route permanent, we add it to* `/etc/sysconfig/static-routes-ipv6` *as an interface route as explained before. As with Debian Sarge we must specify the IPv4-compatible address of the relay router instead of its 6to4 address.*

72 **p. 166/167**

***Fedora Core 6*** *With the lines*

`/etc/sysconfig/network`

```
NETWORKING_IPV6=yes
IPV6_DEFAULTDEV=tun6to4
```

*in* `/etc/sysconfig/network` *we set up a 6to4 node to use the next public default router. If we wanted to use specific (non-public) relay router we could set* `IPV6TO4_RELAY` *to its address to override the default setting* `::192.88.99.1`*.*

73 **p. 170/171**

***Fedora Core 6*** *As with Debian Sarge there are no 4in6 tunnels available for Linux yet.*

75 **p. 173/174**

***Fedora Core 6*** *As with Debian Sarge there is no 6to6 support available yet.*

79 **p. 179**

***Fedora Core 6*** *The same considerations as with Debian Sarge apply with Fedora Core 6, too.*

*The optional last line configures a routeable address on the tunnel interface.*

*The related documentation is hidden in a separate file that can be found at* `/usr/share/doc/initscripts*/ipv6-tunnel.howto`.

**67 p. 156/157**

***Fedora Core 6*** *To configure routes temporarily we can use the same commands as with Debian Sarge. For a permanent configuration we can set the* `IPV6_DEFAULTDEV` *variable in* `/etc/sysconfig/network` *to the tunnel interface name. As explained above, more specific routes go to* `/etc/sysconfig/static-routes-ipv6`*; but since these are interface routes we omit the last item in each line there.*

**68 p. 158**

***Fedora Core 6*** *As with Debian Sarge, the* `sit0` *interface is used for automatic tunneling.*

**69 p. 160/161**

***Fedora Core 6*** *We can configure the tunnel interface temporarily using the same commands as for Debian Sarge. The init scripts assume however that there will be only a single 6to4 interface which is called* `tun6to4` *rather than* `sit1` *as in the Debian Sarge example.*

*The file* `/usr/share/doc/initscripts-8.45.3/ipv6-6to4.howto` *explains in detail how Fedora Core 6 expects 6to4 support to be configured permanently. For a 6to4 tunnel host, all we need are the lines*

`/etc/sysconfig/network-scripts/ifcfg-eth0`

```
IPV6INIT=yes
IPV6TO4INIT=yes
```

*in* `/etc/sysconfig/network-scripts/ifcfg-eth0`*; they tell the init scripts to use this interface and its IPv4 address to set up the 6to4 interface.*



**55 p. 106–108**

***Fedora Core 6*** *The commands* `netstat`, `route` *and* `ip` *explained for Debian Sarge also work with Fedora Core 6. Configuring static routes permanently differs between Debian Sarge and Fedora Core 6, though. They are kept in a file* `/etc/sysconfig/static-routes-ipv6` *that looks like this in our example:*

`/etc/sysconfig/static-routes-ipv6`

```
#If    Network to route                      Gateway
eth0   2001:db8:fedc:aaaa:ff:ff:ff:ff/128   2001:db8:fedc:1::1
eth0   2001:db8:fedc:bbbb::/64              2001:db8:fedc:1::2
eth0   2001:db8::/16                        2001:db8:fedc:1::3
```

*If the interface is a tunnel device, then the address of the next-hop gateway is omitted.*

*The default route is set differently: In* `/etc/sysconfig/network` *we can set either the* `IPV6_DEFAULTDEV` *variable to the name of a point-to-point interface or the* `IPV6_DEFAULTGW` *variable to the address of the next-hop router, possibly followed by a percent sign and an interface name.*

`/etc/sysconfig/network`

```
# Pick any single one...
IPV6_DEFAULTDEV=sit1
IPV6_DEFAULTGW=2001:db8:fedc:1::4
IPV6_DEFAULTGW=2001:db8:fedc:1::4%eth0
```

*Adding the interface explicitly will be necessary if the next-hop router is specified by its link-local address. According to the documentation in* `/usr/share/doc/initscripts*/sysconfig.txt` *it is generally considered good practice to specify it even if the next-hop router is identified by a routeable address.*

**56 p. 109**

***Fedora Core 6*** *First we need to install the* `quagga` *package from the distribution media. Next we create a file* `/etc/quagga/ripngd.conf` *like we would for Debian Sarge. Different than Debian Sarge there is*

*no file* `/etc/quagga/daemons` *but instead individual init scripts for the daemons we need. To enable them we must run the commands*

```
# chkconfig --add zebra
# chkconfig --add ripngd
# chkconfig zebra on
# chkconfig ripngd on
```

*and either reboot afterwards or start the daemons through their init scripts by hand.*

57 p. 110/111

***Fedora Core 6*** *We can apply the steps shown for Debian Sarge without any changes to a Fedora Core 6 setup.*

58 p. 119

***Fedora Core 6*** *Again, we can apply the steps shown for Debian Sarge without any changes to a Fedora Core 6 setup.*

59 p. 121/122

***Fedora Core 6*** *If we use the* `/etc/ip6tables.sh` *script from Debian Sarge to configure our* `ip6tables` *filter rules, then we can apply the steps shown for Debian Sarge without any changes to a Fedora Core 6 setup.*

60 p. 122/123

***Fedora Core 6*** *Again, if we use the* `/etc/ip6tables.sh` *script from Debian Sarge to configure our* `ip6tables` *filter rules, then we can apply the steps shown for Debian Sarge without any changes to a Fedora Core 6 setup.*



62 p. 129

***Fedora Core 6*** *Since* `ip6tables` *is the IPv6 packet filter used with all current Linux distribution, the same considerations as for Debian Sarge apply to Fedora Core 6.*

63 p. 137–139

***Fedora Core 6*** *There is no TRT or NAT-PT implementation available for Fedora Core 6 either.*

64 p. 140

***Fedora Core 6*** *Since there is no TRT or NAT-PT implementation, there are no packet filter considerations to consider.*

65 p. 151

***Fedora Core 6*** *As with Debian Sarge, the implementation follows the stricter RFC 2893 requirements but doesn't assume its peers to do the same, so it conforms to RFC 4213 as well.*

66 p. 152–155

***Fedora Core 6*** *To configure a tunnel temporarily we can apply the same* `ip` *invocations as with Debian Sarge. To configure a tunnel permanently we create a file* `/etc/sysconfig/network-scripts/ifcfg-sit1` *with the contents*

`/etc/sysconfig/network-scripts/ifcfg-sit1`

```
DEVICE=sit1
BOOTPROTO=none
ONBOOT=yes
IPV6INIT=yes
IPV6TUNNELIPV4=192.0.2.2
IPV6ADDR=2001:db8:fedc:4646::1/64       || Optional
```