

**UPDATE: If you are just trying to get IPv6 working (6to4) on v24 sp1, please view this page:**  
[http://www.dd-wrt.com/wiki/index.php/IPv6#6to4\\_Setup](http://www.dd-wrt.com/wiki/index.php/IPv6#6to4_Setup)

The following information was provided by forum member **crushedhat**:  
Further info on this subject can be found [here](#) and [here](#)

**UPDATE:** For Eko's v24 TNG build 14583 you will need 2.4.37 ip6tables modules. Get those from [here](#) and read more [here](#)

Since "working for me" can mean anything, I'm posting exactly what I needed to do to get IPv6 working for me. More detail about many of these steps can be found in the wiki and other docs.

I'm using Eko's v24 TNG build 9856 on my Buffalo WHR-HP-G54 (4MB flash). I'm running this variant so that I'll have some JFFS space for packages:

[http://www.dd-wrt.com/dd-wrtv2/downloads/others/eko/V24\\_TNG/svn9856/dd-wrt.v24-9856\\_NEWD\\_nokaid\\_nohots](http://www.dd-wrt.com/dd-wrtv2/downloads/others/eko/V24_TNG/svn9856/dd-wrt.v24-9856_NEWD_nokaid_nohots)

I see that the IPv6 kernel module and the **radvd** daemon are in the build, and that the Administration/Management panel provides a few IPv6 options. I've set "IPv6" to "Enable", although this doesn't seem to load the IPv6 module. Also, I've set "Radvd enabled" to "Enable", even though this doesn't guarantee that **radvd** will start at the proper time, but it does create /tmp/radvd.conf from the entered config, which is useful.

I have a static IPv4 address, so I've configured DD-WRT to connect a tunnel to Hurricane Electric. In the Administration/Commands panel I've set the following Startup script (these aren't my real addresses, but in the pattern of an HE tunnel):

1. insmod ipv6
2. insmod /jffs/lib/modules/2.4.34/ip6\_tables.o
3. insmod /jffs/lib/modules/2.4.34/ip6table\_filter.o
4. insmod /jffs/lib/modules/2.4.34/ip6t\_multiport.o
5. ip tunnel add he-ipv6 mode sit remote 72.52.104.74 local 1.2.3.4 ttl 64
6. ip link set he-ipv6 up
7. ip addr add 2001:470:1f00:0::2/64 dev he-ipv6
8. ip route add ::/0 dev he-ipv6
9. ip addr add 2001:470:1f01:0:200:00ff:fe00:0000/64 dev br0
10. echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
11. radvd

- ◆ Line 1 loads the critical IPv6 kernel module, followed by the IPv6 filter modules, discussed below.
- ◆ Lines 5 and 6 configure the IPv4 tunnel between the appropriate server at HE and the local static IP address.
- ◆ Line 7 sets the local IPv6 address on the small tunnel network supplied by HE.
- ◆ Line 8 sets a default route to the tunnel for IPv6 traffic.
- ◆ Line 9 sets the IPv6 address on the local LAN/WLAN bridge. This is based on the /64 prefix allocated by HE for the local net plus the MAC address of br0, encoded in the standard IPv6 way. This could instead be in a /64 subnet of a /48 allocated by HE.
- ◆ Line 10 enables packet forwarding between IPv6 networks.
- ◆ Line 11 guarantees that radvd is run after this configuration is complete. It seems that the radvd enabled by the GUI option starts in a different thread from the startup script, so that it

## IPv6\_on\_v24

exits immediately upon discovering IPv6 missing in the kernel or an incomplete config. The second run will simply exit, so there should be correct behavior (one running radvd) regardless of the race.

I've entered the following in ?Radvd config? on the Administration/Management panel. The same network prefix is used as above on line 9:

```
interface br0 {
    AdvSendAdvert on;
    prefix 2001:470:1f01:0::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
};
```

One more important thing is necessary. In the ?Firewall? section of Administration/Commands I've added this line, which allows incoming packets using the IPv6-over-IPv4 protocol to flow to my tunnel:

```
iptables -I INPUT 2 -p ipv6 -i vlan1 -j ACCEPT
```

These steps were sufficient to get DD-WRT routing IPv6 to my local net. Machines on the local net now autoconfigure using Stateless Address Autoconfiguration. However, this leaves IPv6 without a firewall, which is critical especially for Windows which listens to MSRPC (port 135) on IPv6. To configure an IPv6 firewall, I've installed the following packages to /jffs:

[http://downloads.openwrt.org/kamikaze/7.09/brcm-2.4/packages/kmod-ip6tables\\_2.4.34-brcm-1\\_mipsel.ipk](http://downloads.openwrt.org/kamikaze/7.09/brcm-2.4/packages/kmod-ip6tables_2.4.34-brcm-1_mipsel.ipk)  
[http://downloads.openwrt.org/kamikaze/7.09/brcm-2.4/packages/ip6tables\\_1.3.7-1\\_mipsel.ipk](http://downloads.openwrt.org/kamikaze/7.09/brcm-2.4/packages/ip6tables_1.3.7-1_mipsel.ipk)

For some reason the **ipkg** in v24 won't download from URLs, so I was forced to download them elsewhere and copy the packages in using **scp**. After installing these packages, lines 2 thru 4 in the startup config above will load appropriate kernel modules (there are other **ip6tables** modules you might also want to load).

Next, I added the following to my firewall commands:

```
export IP6TABLES_LIB_DIR=/jffs/usr/lib/iptables
PATH="$PATH":/jffs/usr/sbin
ip6tables -F
ip6tables -A FORWARD -p tcp -i he-ipv6 --syn -m multiport --dports ftp-data,ftp,ssh,smtp,http,https
ip6tables -A FORWARD -p tcp -i he-ipv6 --syn -j DROP
ip6tables -A FORWARD -p udp -i he-ipv6 -m multiport --dports ntp,domain -j ACCEPT
ip6tables -A FORWARD -p udp -i he-ipv6 -j DROP
```

The first two lines set up the environment to run the **ip6tables** command. Next the tables are flushed, since this is not done automatically before calling the firewall script as it is with the IPv4 tables. My testing reveals that booting a router causes the firewall script to be run no less than four times! I'm guessing this might be because it is tied to the **wanup** condition. The remaining lines allow through certain TCP and UDP ports, reject all other ports, and ultimately let through all other protocols, such as ICMPv6. If you are more paranoid you could write more specific rules for other protocols.

What could be done to improve this and make IPv6 configuration easier? First of all, the necessary IPv6 firewall modules could be in a standard build. NEWD\_nokaoid isn't small enough for any free JFFS space, but I believe that these few modules should fit (I also miss the **ip6t\_REJECT** module, which is missing from the

## IPv6\_on\_v24

OpenWRT package). Next, it would be really nice to have **busybox** built with the **ping6** command and the **inet6** option to **netstat**, and to add the **traceroute6** command. It's really frustrating not having a **ping6** command, and installing an alternate **busybox** from OpenWRT just for this is rather large. Finally, some well-integrated GUI configuration for IPv6, so that it doesn't look like a second-class protocol, would be welcome.

If I get some time, I'll create my own custom build with some of these issues fixed and release it if anyone is interested.

Overall, I'm happy to see it finally possible to make IPv6 work again in some form on a 4MB router, after the (apparently) failed promise to have it integrated before the v24 release. I understand there are space issues, but this is no excuse for omission from the mega build. I've always found it annoying that most external mention of DD-WRT has included IPv6 in the feature list because it's supported in the ancient v23, but that reality doesn't fit people's expectations that it is also supported in current development.