

[English](#) • [Deutsch](#) • [Español](#) • [Français](#) • [Italiano](#) • [???](#) • [Polski](#) • [Português](#) • [???????](#) • [Svenska](#) • [???\(????\)?](#) • [???\(??\)?](#) •

**Internet Protocol version 6 (IPv6)** is a network layer IP standard used by electronic devices to exchange data across a packet-switched network. It follows IPv4 as the second version of the Internet Protocol to be formally adopted for general use.

There are too many devices in the world for 32-bit ( $2^{32}$ , or 4,294,967,296) IP addresses to be sufficient. The next version of the TCP/IP protocol, IPv6 offers 128-bit address space, or  $2^{128}$  ( $\approx 3.4 \times 10^{38}$ ) IP addresses; this is approximately  $5 \times 10^{28}$  addresses for each of the roughly 6.5 billion people alive today, or about 57 billion addresses for each gram of matter in the Earth.

Keep in mind it can be dangerous to enable IPv6 without also having a firewall on each client that **handles IPv6 packets**, or having ip6tables on your router to filter incoming connections. ip6tables is NOT included by default with DD-WRT, which means your clients will be directly exposed to the Internet once you have enabled IPv6.

## Contents

- [1 Prerequisites](#)
- [2 IPv6 on your LAN](#)
  - ◆ [2.1 Overview](#)
  - ◆ [2.2 Enabling IPv6 Support](#)
  - ◆ [2.3 Configuring Radvd \(if your DD-WRT box is NOT your IPv6-router\)](#)
  - ◆ [2.4 ip6tables for K26 big images](#)
    - ◇ [2.4.1 Kernel modules](#)
    - ◇ [2.4.2 Userland](#)
- [3 6to4 Setup](#)
  - ◆ [3.1 Overview](#)
  - ◆ [3.2 Enabling IPv6 Support](#)
  - ◆ [3.3 Configuring Radvd](#)
  - ◆ [3.4 Startup Script](#)
  - ◆ [3.5 6to4 on current builds \(after v24 sp1\)](#)
  - ◆ [3.6 6to4 on k2.6 builds](#)
  - ◆ [3.7 6rd](#)
  - ◆ [3.8 Performance Concerns](#)
  - ◆ [3.9 6to4 IPv6 as default protocol](#)
  - ◆ [3.10 6to4 on WAN connections with dynamically assigned IPv4 address](#)
- [4 SixXS Tunnel Broker](#)
  - ◆ [4.1 Requesting a Tunnel/Subnet](#)
  - ◆ [4.2 Static Tunnel](#)
  - ◆ [4.3 Dynamic Tunnel](#)
    - ◇ [4.3.1 Install AICCU](#)
    - ◇ [4.3.2 Announce your own subnet with radvd](#)
- [5 Hurricane Electric's Tunnelbroker.net](#)
  - ◆ [5.1 Registration](#)

- ◆ [5.2 Features](#)
- ◆ [5.3 Startup Script](#)
- ◆ [5.4 Endpoint updater](#)
- ◆ [5.5 Another approach](#)
- ◆ [5.6 Additional Links](#)
- [6 Notes](#)
  - ◆ [6.1 Anonymous IPv6 at Microsoft Windows](#)
  - ◆ [6.2 ping6 at dd-wrt v23](#)
  - ◆ [6.3 ping6 at dd-wrt v24-sp2](#)
- [7 Troubleshooting](#)
  - ◆ [7.1 Tunnel issues](#)
    - ◇ [7.1.1 6to4-specific issues](#)
    - ◇ [7.1.2 Hurricane Electric specific issues](#)
  - ◆ [7.2 Radvd issues](#)
  - ◆ [7.3 Working configuration](#)
- [8 External Links](#)

## Prerequisites

- [DD-WRT compatible router](#)
- Knowledge about the [Command Line](#)
- Knowledge about the [ip command](#)
- A current recommended build of the firmware (see your hardware forum for recommendations) using [a version that includes IPv6 support](#).
- IPv6 enabled through the GUI section Administration/Management/Router Management/Ipv6
- Enabling [jffs](#) would be helpful (optional)

## IPv6 on your LAN

### Overview

If you already have a working IPv6 router or are not interested in routing IPv6 traffic out of your LAN, your main concern is to allow your wireless clients traffic pass the to the wired side.

When your DD-WRT box is your IPv6-router, see other solutions below for 6to4 tunneling solutions.

### Enabling IPv6 Support

On the **Administration>Management** page, enable **IPv6** and **radvd**.

### Configuring Radvd (if your DD-WRT box is NOT your IPv6-router)

## IPv6\_(tutorial)

Radvd is similar to a DHCP server. It advertises your IPv6 prefix to the local network, so that a new client can generate its own IP address. This config tells radvd to get an IPv6-address from the wired side, and properly announce it to the LAN (br0).

Copy the following into your radvd settings box:

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

## ip6tables for K26 big images

The currently recommended K2.6 big images ship with basic IPv6 support. That means that you're able to activate the IPv6 IP stack for routing and static tunnels. What's missing are the ip6tables kernel modules and the userland tools needed. It's possible to add the missing parts but that requires some basic Linux system knowledge. It's assumed here that you have /jffs mounted and a few megs of space available and you know how to use ipkg. **Please note:** What you've to download still depends a lot on the hardware you're running on!

## Kernel modules

The currently recommended build r14929 ships with a patched Linux kernel with a version number set to 2.6.24.111. Since the kernel modules from OpenWRT have the magic version information set you can't use the kmod packages from OpenWRT. Because of the version mismatch insmod won't load those modules.

This leaves you with two options:

1. Trust lazytom and download his set of modules from the forum  
<http://www.dd-wrt.com/phpBB2/download.php?id=16285>
2. If you don't trust random people on the internet (which is a good thing) you'll find an introduction on how to build these modules yourself in this blog post  
<http://blog.dest-unreach.be/2010/12/01/compiling-custom-dd-wrt-kernel-modules>

Whatever you do I assume that you've now a bunch of *.ko* files at your hand. Those are your kernel modules. The three most relevant might be *ip6\_tables.ko*, *ip6table\_filter.ko* and *nf\_conntrack\_ipv6.ko*.

On your router create an appropriate directory, I use */jffs/lib/modules/2.6.24.111* and scp all or a subset of the modules to this directory. Afterwards you can change to that directory and try to load them by executing e.g. *insmod ip6tables.ko*. With *lsmod* you can check if they're loaded correctly.

Now that you've the required kernel modules loaded we can move on to the userland tools.

## Userland

System wise OpenWRTs Kamikaze release is very close to the r14929 build, so you should pick at least the following packages from <http://downloads.openwrt.org/kamikaze/8.09.2/>:

- ip6tables
- iptables (important, contains basic shared objects used by ip6tables aswell)
- iptables-mod-contrack
- iptables-mod-filter

Furthermore for debugging efforts it might be helpful to install the *iputils-ping6* and *iputils-traceroute6* packages aswell.

Download all those packages and install them on your router with e.g. *ipkg -d root install \*.ipkg*. Now *ip6tables -vxL* should give you the basic output with the *INPUT*, *FORWARD* and *OUTPUT* default tables.

Now face the fact that the OpenWRT userland is organized slightly different which results in *ip6tables* being unable to find its shared object files. You can workaroud that problem by setting the *IP6TABLES\_LIB\_DIR* enviroment variable.

```
export IP6TABLES_LIB_DIR=/jffs/usr/lib/iptables
```

To make it a permanent setting you can put the following in a *ipv6.startup* file.

```
echo "export IP6TABLES_LIB_DIR=/jffs/usr/lib/iptables" >> /tmp/root/.profile
```

You should now finally be able to create a bunch of regular rules with the *ip6tables* command. Your average */jffs/etc/config/ipv6.startup* file (you've to create one to make this persistent) might start with something like this:

```
#!/bin/sh
export PATH=$PATH:/jffs/usr/sbin
export IP6TABLES_LIB_DIR=/jffs/usr/lib/iptables
echo "export IP6TABLES_LIB_DIR=/jffs/usr/lib/iptables" >> /tmp/root/.profile
MODPATH='/jffs/lib/modules/2.6.24.111'
KMODS='ip6_tables.ko ip6table_filter.ko nf_contrack_ipv6.ko'
for x in $KMODS; do
    insmod $MODPATH/$x
done
```

I hope this helps to get you started without digging hours through the various forum posts.--[Fnord42](#) 20:06, 14 January 2011 (CET)

## 6to4 Setup

### Overview

6to4 is a mechanism which assigns a large block of IPv6 addresses to every IPv4 address on the Internet. You can use 6to4 with DD-WRT to give every computer on your network a globally-accessible IPv6 address

without the need to configure explicit tunneling. This should be the simplest method to add IPv6 to your entire network.

Important Reminders :

- In order to use ipv6, you need the **Standard** or **VoIP** version of DD-WRT, as these are currently the only ones which support both IPv6 and radvd.
- The detailed configuration steps are targeted toward users with a basic DHCP connection for the WAN part. So, if using PPPoE will require replacing vlan1 with ppp0 in each instance. Other connection types will vary.
- When using DD-WRT standard on a router with 4MB Flash, there is no space available for jffs.
- This guide only relies on nvram variables, so that **jffs is not needed**.
- The configuration shown below has been verified to work properly on v24 sp1 stable (standard)

## Enabling IPv6 Support

On the **Administration>Management** page, enable **IPv6** and **radvd**.

## Configuring Radvd

Radvd is similar to a DHCP server. It advertises your IPv6 prefix to the local network, so that a new client can generate its own IP address. This config tells radvd to get a 6to4 prefix from the WAN side (vlan1) and announce it to the LAN (br0)

Copy the following into your radvd settings box:

```
interface br0 {
  MinRtrAdvInterval 3;
  MaxRtrAdvInterval 10;
  AdvLinkMTU 1280;
  AdvSendAdvert on;
  prefix 0:0:0:1::/64 {
    AdvOnLink on;
    AdvAutonomous on;
    AdvValidLifetime 86400;
    AdvPreferredLifetime 86400;
    Base6to4Interface vlan1;
    AdvRouterAddr on;
  };
};
```

## Startup Script

This startup script acquires your current IPv4 address and uses it to calculate a 6to4 IPv6 address. It then creates the tunnel interface, configures a route, and assigns the address. Afterwards, it sends radvd a SIGHUP so that it will become aware that the prefix has changed.

## IPv6\_(tutorial)

Go to **Administration>Diagnostics** or **Administration>Commands** in v24 sp1, copy the following into the big box, and hit "Save Startup".

```
insmod /lib/modules/`uname -r`/ipv6.o
sleep 5
radvd -C /tmp/radvd.conf start
sleep 5
WANIP="$(ifconfig vlan2 | sed -n '/inet /{s/.*addr://;s/ .*;p}')"
if [ -n "$WANIP" ]
then
  V6PREFIX=$(printf '2002:%02x%02x:%02x%02x' $(echo $WANIP | tr . ' '))
  ip tunnel add tun6to4 mode sit ttl 255 remote any local $WANIP
  ip link set tun6to4 mtu 1280
  ip link set tun6to4 up
  ip addr add $V6PREFIX:0::1/16 dev tun6to4
  ip addr add $V6PREFIX:1::1/64 dev br0
  ip -6 route add 2000::/3 via ::192.88.99.1 dev tun6to4
  kill -HUP $(cat /var/run/radvd.pid)
fi
```

## 6to4 on current builds (after v24 sp1)

Pandora-Box got a script that works on current builds of v24. This has been successfully implemented on Eko 13309 big (which does not come with ipv6, so you must use "ipkg install kmod-ipv6" to get it to work.

The scripts are as follows:

in Administration, under IPv6/RaDVD config:

```
interface br0 {
  MinRtrAdvInterval 3;
  MaxRtrAdvInterval 10;
  AdvLinkMTU 1280;
  AdvSendAdvert on;
  prefix 0:0:0:1::/64 {
    AdvOnLink on;
    AdvAutonomous on;
    AdvValidLifetime 86400;
    AdvPreferredLifetime 86400;
    Base6to4Interface vlan2;
  };
};
```

in Administration > Commands, Startup Script:

```
insmod /lib/modules/`uname -r`/ipv6.o
sleep 5
radvd -C /tmp/radvd.conf start
sleep 5
WANIP="$(ifconfig vlan2 | sed -n '/inet /{s/.*addr://;s/ .*;p}')"
if [ -n "$WANIP" ]
then
  V6PREFIX=$(printf '2002:%02x%02x:%02x%02x' $(echo $WANIP | tr . ' '))
  ip tunnel add tun6to4 mode sit ttl 255 remote any local $WANIP
  ip link set tun6to4 mtu 1280
  ip link set tun6to4 up
```

Startup Script

## IPv6\_(tutorial)

```
ip addr add $V6PREFIX:0::1/16 dev tun6to4
ip addr add $V6PREFIX:1::1/64 dev br0
ip -6 route add 2000::/3 via ::192.88.99.1 dev tun6to4
kill -HUP $(cat /var/run/radvd.pid)
fi
radvd -C /tmp/radvd.conf start
```

## 6to4 on k2.6 builds

The above scripts can be modified to use with 2.6 kernel builds, as follows:

### Radvd config:

```
interface br0 {
MinRtrAdvInterval 3;
MaxRtrAdvInterval 10;
AdvLinkMTU 1480;
AdvSendAdvert on;
prefix 0:0:0:1::/64 {
AdvOnLink on;
AdvAutonomous on;
AdvValidLifetime 86400;
AdvPreferredLifetime 86400;
Base6to4Interface vlan2;
};
};
```

### Startup script:

```
insmod /lib/modules/`uname -r`/kernel/net/ipv6/sit.ko
sleep 5
radvd -C /tmp/radvd.conf start
sleep 5
WANIP="$(ifconfig vlan2 | sed -n '/inet /{s/.*addr://;s/ .*;p}')"
if [ -n "$WANIP" ]
then
V6PREFIX=$(printf '2002:%02x%02x:%02x%02x' $(echo $WANIP | tr . ' '))
ip tunnel add tun6to4 mode sit ttl 255 remote any local $WANIP
ip link set tun6to4 mtu 1480
ip link set tun6to4 up
ip addr add $V6PREFIX:0::1/16 dev tun6to4
ip addr add $V6PREFIX:1::1/64 dev br0
ip -6 route add 2000::/3 via ::192.88.99.1 dev tun6to4
kill -HUP $(cat /var/run/radvd.pid)
fi
sleep 10
radvd -C /tmp/radvd.conf start
```

**Note** that this script is specifically for the WRT610N v1, where vlan2 is the WAN. On other routers, you may have to substitute vlan1, eth1, etc. To find your WAN interface, telnet to your router and run

```
ip -4 addr show
```

and find the line with your WAN address. The interface listed there is your WAN interface (mine was eth1 on a DLink DIR-825).

## 6rd

Comcast (and other providers) are using IPV6 over 6rd. See: [Comcast 6RD Configuration Instructions for IPV6](#)

Here is an example dd-wrt config for use with the 2.4 pre-sp2 with a k2.6 build (14584M NEWD-2 K2.6 Eko)

You can disable radvd from the management tab since this script creates the conf file and starts radvd. Place the following in your startup commands section:

```
insmod /lib/modules/`uname -r`/kernel/net/ipv6/sit.ko
sleep 5
HOST6RD=$(nslookup 6rd.comcast.net |grep "Address"|awk '{ print $3 }'|grep -v 192.168.1.1 -m1)
WANIP="$(ifconfig vlan2 | sed -n '/inet /{s/.*addr://;s/ .*;p}')"
if [ -n "$WANIP" ]
then
V6PREFIX=$(printf ' 2001:55c:%02x%02x:%02x%02x' $(echo $WANIP | tr . ' '))
ip tunnel add tun6rd mode sit ttl 255 remote any local $WANIP
ip link set tun6rd mtu 1280
ip link set tun6rd up
ip addr add $V6PREFIX:0::1/32 dev tun6rd
ip addr add $V6PREFIX:1::1/64 dev br0
ip -6 route add 2000::/3 via ::$HOST6RD dev tun6rd
kill -HUP $(cat /var/run/radvd.pid)
fi
echo "interface br0 { \
MinRtrAdvInterval 3; MaxRtrAdvInterval 10; AdvLinkMTU 1280; AdvSendAdvert on; \
prefix $V6PREFIX::/64 { AdvOnLink on; AdvAutonomous on; AdvValidLifetime 86400; \
AdvPreferredLifetime 86400; }; };" \
> /tmp/radvd.conf
radvd -C /tmp/radvd.conf start
```

If you use a local lan gateway other than 192.168.1.1, you will need to update the script. Right now 6rd.comcast.net only resolves to one IP address, but the script does take multiple A records into account and uses the dns resolver on the router to load balance between addresses.

## Performance Concerns

When you send a packet from a 6to4 address to a native IPv6 address, it has to pass through a 6to4 gateway. [RFC3068](#) established the special anycast address **192.88.99.1** which is supposed to lead you to the nearest gateway. However, some ISPs don't handle this properly, so you may find that all of your packets are crossing an ocean or two.

If this affects you, then you may need to do some detective work to locate a closer gateway. One possible approach is to use [IPv4 Looking Glass](#) tools to traceroute to 192.88.99.1 from various locations on the Internet and see where you end up.

Or you could bug your ISP about it, but they'll probably be like "IPvWHAT?"



## 6to4 IPv6 as default protocol

As per [RFC3484](#) §2.1 default policy table, having a 6to4 enabled LAN (using a 6to4 anycast tunnel on a DD-WRT enabled box, say) will mean that 6to4 IPv6 will only be used to communicate to a server if the server is having a 6to4 address too (ie: 2002: prefix on its address) or if there is no IPv4 address assigned to the server name (ie: no A but an AAAA).

In short the following makes IPv4 default on IPv6 enabled Windows boxes using 6to4 prefixes.

1. Start -> Run -> "cmd" -> "netsh" -> "interface" -> "ipv6"

2. To enable IPv6 preference on Microsoft Windows...

```
set prefix ::1/128 50 0
set prefix ::/0 40 1
set prefix 2002::/16 30 1
set prefix ::/96 20 3
set prefix ::ffff:0/96 10 4
set prefix 2001::/32 5 5
```

Only the 2002::/16 line (6to4 prefix policy) is changed.

3. To make things goes back to original mode (IPv4 preference)...

```
set prefix ::1/128 50 0
set prefix ::/0 40 1
set prefix 2002::/16 30 2
set prefix ::/96 20 3
set prefix ::ffff:0/96 10 4
set prefix 2001::/32 5 5
```

And Voilà ;-)

Notes: ::1/128 == localhost, ::/0 == default, 2002::/16 == 6to4, ::/96 + ::ffff::0/96 == IPv4 compatibility, 2001::/32 == Teredo, 2000::/3 would be All IPv6 Global Unicast address space

## 6to4 on WAN connections with dynamically assigned IPv4 address

If you have a dial-up connection with a dynamically assigned IPv4 address, the corresponding IPv6 address block changes every time a new WAN connection is established. You can put the 6to4 startup script into `/jffs/etc/config/6to4.wanup` to get the correct 6to4 IPv6 block every time when you get a new IPv4. Please note that this means that all your network's global IPv6 addresses change, too. For more info, see [\[1\]](#)

## SixXS Tunnel Broker

## Requesting a Tunnel/Subnet

1. [Sign up with a RIPE/ARIN/etc handle](#) or by using the [direct signup](#)
2. After your login at the [SixXS Home](#) you can request a tunnel and later a subnet.

Note that providing a link to a XING or LinkedIn profile or providing links to projects you work on can provide extra ISK credits. ISK credit is used to 'purchase' a tunnel, subnet or other services.

## Static Tunnel

There are instructions at [Remoteroot.net](#) with example scripts.

This wiki also offers an example [IPv6 startup script](#)

Notice that radvd doesn't work with mini version but requires the full version of DD-WRT. Also if you have problems with pinging and SixXS, notice their [FAQ message](#) related to this.

## Dynamic Tunnel

### Install AICCU

1. Install AICCU using ipkg.

```
# ipkg -force-depends install http://downloads.openwrt.org/kamikaze/8.09.2/brcm47xx/packages/aiccu
```

2. Configure AICCU.

```
# vi /jffs/etc/aiccu.conf

# AICCU Configuration

# Login information
username <your nichandle/username>
password <your password>

# Protocol and server listed on your tunnel
protocol tic
server tic.sixxs.net

# Interface names to use
ipv6_interface sixxs

# The tunnel_id to use
# (only required when there are multiple tunnels in the list)
tunnel_id <your tunnel id>

# Be verbose?
verbose false

# Daemonize?
```

## IPv6\_(tutorial)

```
daemonize true

# Require TLS?
requiretls true

# Set default route?
defaultroute true
```

Note that if you can't write to `/jffs/etc/`, you can always store the script in another directory (that of course is persistent otherwise you loose it :) and use eg `'aiccu start /tmp/aiccu.conf'` to pick a different configuration file location.

To actually start aiccu you've to add an *ipup* script, e.g. `/jffs/etc/config/sixxs.ipup`.

```
#!/bin/sh
export PATH=$PATH:/jffs/usr/sbin

# wait until time is synced
while [ `date +%Y` -eq 1970 ]; do
    sleep 5
done

# check if aiccu is already running
if [ -n "`ps|grep etc/aiccu|grep -v grep`" ]; then
    aiccu stop
    sleep 1
    killall aiccu
fi

# start aiccu
sleep 3
aiccu start /jffs/etc/aiccu.conf
```

## Announce your own subnet with radvd

*radvd* is already part of the IPv6 enabled images so you just have to bring up your own IPv6 address and start *radvd* with a configuration. You can enter the contents for your *radvd.conf* in the webfrontend so that it will end up in `/tmp/radvd.conf` but you've to start *radvd* yourself with a *startup* file **after** you assigned an IPv6 address from the subnet to one of your interfaces. In practise your *radvd.conf* can be as simple as this:

```
interface br0
{
    AdvSendAdvert on;
    prefix 2001:DB8::/64
    {
        };
    };
};
```

The *startup* file, e.g. `/jffs/config/ip6.startup` to go with this configuration will depends a bit on how many addresses/subnets you'll actually use. In the usual cases it will be very similar to the following.

```
#!/bin/sh
# Assign an IPv6 address and route for br0
ip -6 addr add 2001:DB8::/64 dev br0
ip -6 route add 2001:DB8::/64 dev br0
```

## IPv6\_(tutorial)

```
# route the rest of the /48 to loopback
ip -6 route add 2001:DB8::/48 dev lo

# Execute a script with some iptables rules
#/jffs/etc/filters6.sh

# Start radvd
/usr/sbin/radvd -C /tmp/radvd.conf
```

**Please note: 2001:DB8::/32 is a subnet dedicated for use in documentation. You've to replace that with the subnet you'll get assigned by SixXS.**

Reboot the router! When the router comes back up, you should be able to use IPv6. Congratulations!

# Hurricane Electric's Tunnelbroker.net

## Registration

- 1) Sign up for an account with our web based registration at [Tunnelbroker.net](http://Tunnelbroker.net)
- 2) After creating your account, provide your IPv4 endpoint, and pick the closest tunnel-server to your location:

### Asia

- Hong Kong, HK
- Singapore, SG
- Tokyo, JP

### Europe

- Amsterdam, NL
- Paris, FR
- Stockholm, SE
- Zurich, CH

### North America

- Ashburn, VA, US
- Chicago, IL, US
- Dallas, TX, US
- Fremont, CA, US
- Los Angeles, CA, US
- Miami, FL, US
- New York, NY, US
- Seattle, WA, US
- Toronto, ON, CA

## Features

By default you already get a /64 allocation routed to your side of the tunnel.

If you need more than a single /64, with the click of a button you can allocate a single /48 routed to your side of the tunnel.

A large variety of configuration examples are provided for various platforms covering: Linux, \*BSD, Windows as well as router platforms.

Tunnelbroker.net is operated and maintained by a business and not individuals, and provided FREE to anyone interested in learning more about IPv6.

## Startup Script

This works for me on v24 SP2:

```
#edited by timenetworks (7-6-2011)
#to fix the ipv6 route table issue

insmod ipv6
sleep 5
WANIP=$(ifconfig vlan2 | sed -n '/inet /{s/.*/addr://;s/ .*;p}')
MY_TUNNEL_ADDR="2001:470:1f0e:5cd::2" # change me
MY_ROUTED_ADDR="2001:470:1f0f:5cd::1" # change me
echo "External IP:" $WANIP > /jffs/startup.debug
if [ -n $WANIP ]
then
echo "configuring tunnel" >> /jffs/startup.debug

# The following commands are straight from HE's website
ip tunnel add he-ipv6 mode sit remote 216.218.224.42 local $WANIP ttl 255
ip link set he-ipv6 up
ip addr add $MY_TUNNEL_ADDR/64 dev he-ipv6
ip route add ::/0 dev he-ipv6

# These commands aren't on HE's website, but they're necessary for the tunnel to work
#WARNING: Adding two times the same address seems to be WRONG, please someone with
# HE access update this information to something decent.
# See also http://www.dd-wrt.com/phpBB2/viewtopic.php?t=137874
#ip -6 addr add $MY_ROUTED_ADDR/64 dev he-ipv6 #this no longer require as he-ipv6 interface only
ip -6 addr add $MY_ROUTED_ADDR/64 dev br0
#ip route add 2000::/3 dev he-ipv6 #default route is enough (23-6-2011)

echo "starting radvd" >> /jffs/startup.debug
radvd -C /jffs/radvd.conf &
fi
```

- The tunnel address and routed address are on your Tunnel Details page on HE's website. For MY\_TUNNEL\_ADDR, use the value for "Client IPv6 Address", and for MY\_ROUTER\_ADDR, use the value for "Routed /64".
- Many people have working configurations using only the tunnel address for everything, but you need to use the routed address for your network if you want certain things like reverse DNS to work.

## IPv6\_(tutorial)

- You have to have your radvd config saved in /jffs/radvd.conf. Running radvd from the web interface has never worked for me; if it works for you, you can drop the radvd line from the script.
- This script doesn't handle when your IP address changes; you'll need to manually update it on HE's website, or use the endpoint updater script below.
- I occasionally have issues with the tunnel dying randomly. Pinging the router's IPv6 address fixes it for some reason, I have no idea why. :( -- update 2009.12.14 by calraith: Try adding metric 1 as an argument to the ip route add directives. ip route add ::/0 dev he-ipv6 metric 1
- When you put your radvd config into the webgui (on administration tab) it is available at /tmp/radvd.conf. You can use this instead of creating a /jffs/radvd.conf file (I have issues with my jffs on my WRT350N). I have my startup.debug file in /tmp/startup.debug as well. ~methaneb

## Endpoint updater

If your router gets a dynamic ip, you need to update your endpoint in your tunnelbroker account. This can be done on DDWRT with this busybox-only script: [Tunnelbroker.net endpoint address updater](#)

Download it into tmp or jffs, edit the file to set up your credentials or just run it from commandline:

```
updateipv4.sh username password tunnelid
```

A successful update means either your current endpoint IP is the same as your previous one, or it has been updated to your current one.

Edit: zracer For this script to work you need to deselect Security->Firewall->"Block Anonymous WAN Requests (ping) "

Because tunnelbroker.net uses a ping to verify the new ip address.

## Another approach

The above approach did not work for me. The general problem with DD-WRT is there are no tools to really test what is wrong with the configuration. IPv6 to IPv4 did work for me. So I decide to try a method that would allow me to easily toggle back and forth. That way I could tell the difference between a problem on my router v.s. a problem in my end client.

First off since we need a way to recycle the tunnel when the router's IPV4 address change, it makes sense to put the main logic in a separate script. The following is my /jffs/updateipv4.sh script:

```
#!/bin/sh -x
#SAMPLE USERID="29812e32f424324324234"
USERID="<insert value here>"
#SAMPLE PASSWORD="CatsAndDogs"
PASSWORD="<insert value here>"
#SAMPLE TUNNEL_ID="96782"
TUNNEL_ID="<insert value here>"
#SAMPLE CLIENT_IPV6_ADDRESS="2001:271:18:2c7::2/64"
CLIENT_IPV6_ADDRESS="<insert value here>"
#SAMPLE SERVER_IPV4_ADDRESS="218.65.27.48"
SERVER_IPV4_ADDRESS="<insert value here>"
echo >> /tmp/wanip
```

## IPv6\_(tutorial)

```
WANIP_OLD=`cat /tmp/wanip`
WANIP="$(ifconfig vlan2 | sed -n '/inet /{s/. *addr://;s/ .*;p}')"
if [ -n "$WANIP" ]
then
  if [ "$WANIP" != "$WANIP_OLD" ]
  then
    echo "External IP: $WANIP" 1>&2
    ROUTED_ADDRESS=`sed -n -e 's,^ *prefix *\([^ ]*\) *{,\1,p' /tmp/radvd.conf`
    if [ $ROUTED_ADDRESS = "0:0:0:1::/64" ]
    then
      CLIENT_IPV6_ADDRESS=""
    fi
    ip tunnel del tun6to4 2>>/dev/null
    ip tunnel del he-ipv6 2>>/dev/null
    if [ -n "$CLIENT_IPV6_ADDRESS" ]
    then
      echo "Configure he-ipv6 tunnel" 1>&2
      MD5PASSWORD=`echo -n "$PASSWORD"|md5sum|sed -e 's/ *-//g`
      /usr/bin/wget 'http://ipv4.tunnelbroker.net/ipv4_end.php?ipv4b='AUTO'&pass='$MD5PASSWORD
      ip tunnel add he-ipv6 mode sit ttl 255 remote $SERVER_IPV4_ADDRESS local $WANIP
      ip link set he-ipv6 up
      ip addr add $CLIENT_IPV6_ADDRESS dev he-ipv6
      ip route add ::/0 dev he-ipv6

      BR0_MAC=$(ifconfig br0 |sed -n -e 's,.*HWaddr \(..\):\(..\):\(..\):\(..\):\(..\):\(..\):
      # These commands aren't on HE's website, but they're necessary for the tunnel to work
      ip -6 addr add $(echo "$ROUTED_ADDRESS"|sed "s,::/..,:::$BR0_MAC/64,") dev br0
      ip -6 route add 2000::/3 dev he-ipv6
    else
      echo "Configure tun6to4 tunnel" 1>&2
      V6PREFIX=$(printf '2002:%02x%02x:%02x%02x' $(echo $WANIP | tr . ' '))
      ip tunnel add tun6to4 mode sit ttl 255 remote any local $WANIP
      ip link set tun6to4 mtu 1480
      ip link set tun6to4 up
      ip -6 addr add $V6PREFIX:0::1/16 dev tun6to4
      ip -6 addr add $V6PREFIX:1::1/64 dev br0
      ip -6 route add 2000::/3 via ::192.88.99.1 dev tun6to4
    fi
    echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
  fi
fi
```

Be sure to run 'chmod ugo+rx /jffs/updateipv4.sh'.

Next the startup script. Set this in your startup under Administration->Commands->Startup:

```
#Enable IPV6
insmod /lib/modules/`uname -r`/kernel/net/ipv6/sit.ko
sleep 5
radvd -C /tmp/radvd.conf start
sleep 5
rm -f /tmp/wanip
/jffs/updateipv6.sh 2> /jffs/startup.debug
kill -HUP $(cat /var/run/radvd.pid)
sleep 10
echo "starting radvd" >> /jffs/startup.debug
radvd -C /jffs/radvd.conf start &
```

Note: The sleep commands are just to give time for the IPV4 network to come-up, and for the updates from the script to finish.

Another approach

## IPv6\_(tutorial)

Finally, Radv5 config. Set this under Administration->Management->IPV6 Support->Radvd config:

```
interface br0 {
MinRtrAdvInterval 3;
MaxRtrAdvInterval 10;
AdvLinkMTU 1480;
AdvSendAdvert on;
prefix <insert value here> {
AdvOnLink on;
AdvAutonomous on;
AdvValidLifetime 86400;
AdvPreferredLifetime 86400;
# Base6to4Interface vlan2;
};
};
```

The prefix value should be your Route 64 address. e.g. 2001:270:1c:1d8::/64

If instead you wish to use an IPV6 to IPV4 bridge instead use an address of 0:0:0:1::/64 and uncomment the Base6to4Interface line like the following:

```
interface br0 {
MinRtrAdvInterval 3;
MaxRtrAdvInterval 10;
AdvLinkMTU 1480;
AdvSendAdvert on;
prefix 0:0:0:1::/64 {
AdvOnLink on;
AdvAutonomous on;
AdvValidLifetime 86400;
AdvPreferredLifetime 86400;
Base6to4Interface vlan2;
};
};
```

Walla. Once everything is saved you are ready to reboot your router. If things don't work try running the updateipv6.sh script interactively. Note: I've designed the updateipv6.sh script so it can be rerun regularly to pickup a new wan ip address. So if the wanip is the same as the last time it ran, it does nothing. If you want to force it to run anyway (for debugging purposes), the simply remove the /tmp/wanip file first. e.g.

```
$ rm /tmp/wanip;/jffs/updateipv6.sh
```

## Additional Links

[Tunnelbroker.net's user forums](#)

[IPv6 Setup with SAMBA](#)

[IPv6 Setup using NVRAM](#)



# Notes

## Anonymous IPv6 at Microsoft Windows

By default on WinXP, Win2003 or WinVista, IPv6 addresses built from IPv6 prefix (stateless configuration for instance) are using the EUI-64 (cf MAC address) of your network card as a suffix. This can lead to some privacy issues, hence another IPv6 address is dynamically computed and used by default. Such an anonymous address changes (according to a computation algorithm) each time the IPv6 stack starts. This can cause some trouble under certain conditions where the server needs a unique and static IPv6 (containing the EUI-64). The solution is to deactivate the creation of those "privacy addresses". But be carefull, this will then lead to leaking the EUI-64 and also enables people to track you (logs, google, etc). So, it's a tradeoff.

If you want to disable temporary IPv6 addresses, try the command :

```
netsh interface ipv6 set privacy state=disable
```

or

```
ipv6 -p gpu UseTemporaryAddresses no
```

## ping6 at dd-wrt v23

Try to install busybox from OpenWRT: <http://downloads.openwrt.org/people/florian/bin/packages/>

```
~ # ipkg install http://downloads.openwrt.org/people/florian/bin/packages/busybox\_1.1.3-1\_mipsel
```

Okay now you have to install uclibc (with -force-depends):

<http://downloads.openwrt.org/people/florian/bin/packages/>

```
~ # ipkg -force-depends install http://downloads.openwrt.org/people/florian/bin/packages/uclibc
```

After installing these packages you have to change the priority of the /lib path in your shell environment:

```
~ # export LD_LIBRARY_PATH=/usr/lib:/jffs/lib:/jffs/usr/lib:/jffs/usr/local/lib:/lib
```

For the last step you have to change the ping6 symlink:

```
~ # rm /jffs/bin/ping6
~ # cd /jffs/bin
~ # ln -s /jffs/bin/busybox ping6
```

Now ping6 should work for the current xterm session.

## ping6 at dd-wrt v24-sp2

Install iputils-ping6 from OpenWRT

```
root@DD-WRT:~# ipkg -force-depends install http://downloads.openwrt.org/kamikaze/8.09.2/brcm47xx/
```

## Troubleshooting

So you've tried to set up IPv6, but it didn't work! Don't panic, it's probably still fixable.

### Tunnel issues

The first question to ask is, can you ping the router from the internet? Head over to [2] and use the ping test with your external IPv6 IP. (For HE and SixXS tunnels, this will be your client tunnel IP; for 6to4, it'll be generated from your public IPv4 address.) If the ping goes through, congratulations! Your tunnel isn't the problem.

### 6to4-specific issues

6to4 uses a special IP address (192.88.99.1) to configure the tunnel. Try pinging this IP; if it doesn't work, then 6to4 probably won't work for you. You could bother your ISP about it, but as likely as not they'll just ignore you completely.

6to4 also requires that protocol 41 (different from port 41!) be accessible. It's rare for ISPs to block this, but if they really feel like jerks it's not impossible. proto41 blocking is difficult to detect, unfortunately.

### Hurricane Electric specific issues

Make sure your IPv4 address is set properly on the HE website. If not, your tunnel won't work. Similarly, make sure the IPv6 tunnel addresses you use are correct.

### Radvd issues

If you're using radvd and it's not working somehow, then your router will have a working IPv6 connection, but it won't be passed on to your other computers. If your computers aren't receiving IPv6 addresses, then this is the problem.

The best tool for checking this is the radvdump program. Run it as root, and it'll show you any router advertisements your computer sends or receives. If you don't see any after a while, radvd probably isn't running. Try running "ps | grep radvd" on your router to see if it is. If it's not running, you can save your config file to jffs, and then run radvd manually:

```
radvd -C /jffs/radvd.conf
```

Beyond making sure IPv6 support is enabled on your client computers, you shouldn't have to configure anything else when using radvd.

## Working configuration

For comparison, here's the output of a few commands on my router (which has a working HE.net tunnel):

```

root@DD-WRT:~# ip -6 addr show
1: lo: <LOOPBACK,MULTICAST,UP>
   inet6 ::1/128 scope host
3: eth0: <BROADCAST,MULTICAST,PROMISC,UP>
   inet6 fe80::21c:10ff:fe5b:7978/64 scope link
4: eth1: <BROADCAST,MULTICAST,PROMISC,UP>
   inet6 fe80::21c:10ff:fe5b:797a/64 scope link
5: vlan0: <BROADCAST,MULTICAST,PROMISC,UP>
   inet6 fe80::21c:10ff:fe5b:7978/64 scope link
6: vlan1: <BROADCAST,MULTICAST,UP>
   inet6 fe80::21c:10ff:fe5b:7979/64 scope link
7: br0: <BROADCAST,MULTICAST,UP>
   inet6 2001:470:1f0f:5cd::1/64 scope global
   inet6 fe80::21c:10ff:fe5b:7978/64 scope link
10: he-ipv6: <POINTOPOINT,NOARP,UP>
    inet6 2001:470:1f0f:5cd::1/64 scope global
    inet6 2001:470:1f0e:5cd::2/64 scope global
    inet6 fe80::48b3:3572/128 scope link

root@DD-WRT:~# ip -6 route show
2001:470:1f0e:5cd::/64 via :: dev he-ipv6 metric 256 mtu 1480 advmss 1420
2001:470:1f0f:5cd::/64 via :: dev he-ipv6 metric 256 mtu 1480 advmss 1420
2001:470:1f0f:5cd::/64 dev br0 metric 256 mtu 1500 advmss 1440
2000::/3 dev he-ipv6 metric 1024 mtu 1480 advmss 1420
fe80::/64 dev eth0 metric 256 mtu 1500 advmss 1440
fe80::/64 dev eth1 metric 256 mtu 1500 advmss 1440
fe80::/64 dev vlan0 metric 256 mtu 1500 advmss 1440
fe80::/64 dev vlan1 metric 256 mtu 1500 advmss 1440
fe80::/64 dev br0 metric 256 mtu 1500 advmss 1440
fe80::/64 via :: dev he-ipv6 metric 256 mtu 1480 advmss 1420
ff00::/8 dev eth0 metric 256 mtu 1500 advmss 1440
ff00::/8 dev eth1 metric 256 mtu 1500 advmss 1440
ff00::/8 dev vlan0 metric 256 mtu 1500 advmss 1440
ff00::/8 dev vlan1 metric 256 mtu 1500 advmss 1440
ff00::/8 dev br0 metric 256 mtu 1500 advmss 1440
ff00::/8 dev he-ipv6 metric 256 mtu 1480 advmss 1420
default dev he-ipv6 metric 1024 mtu 1480 advmss 1420

```

## External Links

- [IPv6 at Wikipedia](#)
- [IPv6 Information Page](#)
- ["IPv6: What, Why, How" - a presentation/tutorial to introduce those familiar with IPv4 to IPv6](#)
- [Setting up IPv6 using Hurricane Electric on DD-WRT v23 - SAMBA](#)
- [Setting up IPv6 using Hurricane Electric on DD-WRT v23 - NVRAM](#)
- [Setting up ipv6 using MyBSD/MANISV6 on DD-WRT v23](#)