

Configuring FreeDNS for Dual IPv4/IPv6 Hosts

This document describes how to set up a dual mode hostname that supports both a static IPv6 address as well as a dynamic IPv4 address.

1. Problem Statement

Everyone should be wanting to get onto the IPv6 bandwagon, we're rapidly running out of IPv4 addresses. For a countdown timer, check out Hurricane Electric's (<http://ipv6.he.net/statistics/>) stats page. The problem is a website can't afford to be IPv6-only, they need to support IPv4 since ISPs are slow to adopt IPv6.

One method of addressing IPv6 sites in use by Google (<http://ipv6.google.com>) and NetFlix (<http://ipv6.netflix.com>) is to create a separate hostname for IPv6 content. This is useful if you want to completely segregate IPv6 traffic, but it reduces a lot of your information gathering abilities. Even those with IPv6 (perhaps unknowingly) will likely only go to www.google.com (<http://www.google.com>) rather than the ipv6.google.com (<http://ipv6.google.com>) equivalent.

In my opinion, a much better implementation is the Kame.net (<http://www.kame.net>) model whereby a single hostname will serve up both an IPv4 A record (a specific IP address) and an IPv6 AAAA record (the IPv6 equivalent). Then the client can decide which to go to, usually IPv6 by default if it's enabled. Why don't NetFlix and Google do this? Because they use a Content Delivery Network (CDN) to help offload content. A CDN will take content and replicate it all over the Internet, providing both a better end user experience as well as allowing NetFlix to concentrate on getting more movies instead of just more bandwidth.

Since there isn't much draw for IPv6 content, the CDNs haven't seen the need to allow for IPv6 services. Of course, now we're in a chicken-and-egg situation, is there no draw for IPv6 content or do people simply not use it because they don't know it's there because the CDNs don't support it?

The root of the problem is that CDNs rely on DNS to map customers. When you go to a website it will generally return a CNAME - a pointer that says instead of something like movies.netflix.com you should go to movies.netflix.com.edgesuite.net. EdgeSuite is an Akamai service and essentially NetFlix is letting Akamai choose which IP address to hand out. Since Akamai doesn't (yet) support IPv6 NetFlix needs to serve up that content locally. However DNS doesn't support a dual mode "for IPv4 ask this guy about the IP address but here's the IPv6 address" so NetFlix is stuck.

And so are you. If you own your own domain, chances are you use a CNAME to forward requests for your hostname to someone like DynDNS who will record your current address. You need to do this because every time your home router reboots, or your ISP reboots their equipment, you run the risk of getting a different IP address. If you had a static IP address (which is getting expensive) you could just set up an A record and a AAAA record and be done with it. Even if you just use a domain that you get from DynDNS you're in trouble since they don't seem to support IPv6 either. But luckily we've got FreeDNS at Afraid.org (<http://afraid.org>).

2. Configuring Afraid.org

The first thing to do is to set up an account at Afraid.org (<http://afraid.org>). This is pretty straightforward and 100% free, but if you've got the means please consider a premium account to help support them.

Once you've logged in click on the Subdomains link on the left. You'll want to add a subdomain and there are plenty of free ones (<http://freedns.afraid.org/domain/registry/>) to choose from. You can do a search for a name you like, or just get a good generic sounding one. Don't ask my why mooo.com is so popular...

Once you've selected a domain, you'll want to create an A record. The subdomain is whatever you like, unless you own your own domain I'd stay away from common ones like www and instead use some variant of a username or your own domain name. I use the hostname on my own network followed by my username (www-username, for example). The Destination you can leave as is, we'll be making that dynamic soon enough and it defaults to the IP you're using right now.

Now you'll want to create a AAAA record. Use the same Domain and Subdomain you just used for the A record, but paste you IPv6 address in the Destination slot. Once this is done you're up and running! You can run **dig subdomain.domain aaaa** to see the AAAA record and **dig subdomain.domain a** to get the A record (note it may take a little while for the system to update, give it about an hour). Note that your ISP's nameserver may not give out IPv6 addresses, if that's the case you can try **dig @8.8.8.8 subdomain.domain aaaa** to use Google's public DNS (<http://code.google.com/speed/public-dns/>) which does handle IPv6 records (but not yet IPv6 transport).

3. Configuring Dynamic DNS

So you're up and running on both IPv4 and IPv6 - but only for now (assuming your server has the same public IP address as the computer you configured things from). To configure Dynamic DNS you have to click on the cleverly named Dynamic DNS link to the left of the Afraid.org screen. There you should see a list of your available A records, and you'll want to right click on the Direct URL link for the appropriate hostname and copy the link to your clipboard.

I'm going to assume you're using Linux here, but most of this should also work on OSX and if you've

got Windows I recommend looking into Cygwin (<http://www.cygwin.com/>). You can use a simple command called **wget** to update your IP address, just run **wget <http://freedns.afraid.org/dynamic/update.php?abcdefghi>** using the URL you just copied (which will not use abcdefghi). You could set that up in a cron job, but that can be hard on the Afraid.org servers and we like them so we want to be nice.

Rather than telling our DNS friends to update their DNS records every few minutes we'd rather ask what our IP address is every few minutes. As such we can use a simple script (dnsupdate.sh) that I put in `/usr/local/bin/dnsupdate.sh`:

Example 1. An example DNS update script

```
#!/bin/bash
logfile=/var/log/dnsupdate.log
ipfile=/var/log/currip
conffile=/etc/freedns.conf

# Lock down the files, they're important! I also run sudo despite being root
# so it prompts for a password if you run it as a normal user
sudo chown root $conffile
sudo chmod 600 $conffile
sudo chown root $0
sudo chmod 744 $0

# $conffile should be a simple list of the format:
# hostname direct_url
#
# Where hostname is the fully qualified hostname (www.ebower.com) of the host
# and the direct_url is the freedns.afraid.org URL for the host in
# question. One entry per line, as many lines as you need.
#
# Example:
# www.ebower.com http://freedns.afraid.org/dynamic/update.php?dskafjhawelirfuhawlkfjbaw

# Set the timestamp on the logfile to the last time we checked our IP
touch $logfile

errormode=none

currip=$(curl http://whatismyip.akamai.com 2> /dev/null)
ipresult=$?

# Did we get an IPv4 address?
isvalid=$(echo $currip | grep '[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}')

if [ ! $ipresult = 0 ]; then
    errormode="Error getting IP with result $ipresult"
else
    if [ "$isvalid" = "" ]; then
        errormode="Parse error: $currip"
    fi
fi
```

```
if [ ! "$errormode" = "none" ]; then
    echo $(date +%Y-%m-%d_%H-%M) - $errormode >> $logfile
    exit 2
fi

previp=$(cat $ipfile)

if [ ! "$currip" = "$previp" ]; then
    echo $currip > $ipfile
    updateresult=$?
    echo $(date +%Y-%m-%d_%H-%M) - IP changed from $previp to $currip with exit code of $updateresult
fi

currdns=$(dig +short $hostname | tail -n1)
dnsresult=$?

if [ ! $dnsresult = 0 ]; then
    currip="Error $dnsresult"
    echo $(date +%Y-%m-%d_%H-%M) - DNS error $dnsresult >> $logfile
    exit 2
fi

while read line
do
    updatehost=$(echo $line | awk '{print $1}')
    updateurl=$(echo $line | awk '{print $2}')

    currdns=$(dig +short $updatehost | tail -n1)
    dnsresult=$?

    if [ ! $dnsresult = 0 ]; then
        currip="Error looking up $updatehost with result $dnsresult"
    fi
    if [ ! "$currip" = "$currdns" ]; then
        wgetresult=$(wget -qo- --read-timeout=0.0 --waitretry=5 --tries=400 --background \
            $updateurl)
        echo $(date +%Y-%m-%d_%H-%M) - Updating DNS for $updatehost from $currdns to $currip with result $wgetresult
    fi
done < $conffile
```

You'll want to create a file called (by default) `/etc/freedns.conf` which is a simple list of hostnames and URLs. You can see an example listing in the printout above but for obvious reasons I don't include my own information.

The first thing I do is `touch /tmp/dnsupdate.log` - that's how you know it's working! I then make sure the file permissions are correct. The permissions on the script aren't all that important, but the permissions on the configuration file are - anyone who gets access to that file can compromise your DNS setup and hijack your domain. I use **wget** again to grab my IP address, I tend to prefer **curl** but it's not installed by default on Ubuntu systems and I wanted to make life easier on you. I then use **dig** to figure

out what the DNS server thinks my hostname is. If they're different, then I perform the update by going to the Afraid.org URL and then I tag the log file with an update.

Now you'll want to run this on a regular basis. You can do this by running **sudo crontab -e** and adding a line to the end of the file stating:

```
*/10 * * * * /usr/local/bin/dnsupdate.sh
```

This will run the script every ten minutes and you can check the logfile timestamp to make sure it's working.

4. Setting up Your Own Domain

I own all of ebower.com and, as such, I'd rather not need to remember the domain I get from Afraid.org or DynDNS or any other provider. So I have a CNAME set up. You'll have to look at your specific domain registrar to see what, if any, DNS tools they support. I use MyDomain.com and they have a fairly decent set of DNS features, but only IPv4.

You have a few choices in integrating your domain with your new Afraid.org account. You can use the safe route and configure a CNAME from your domain to your Afraid.org hosts - for example you can have www.ebower.com CNAME to www-ebower.linuxsecured.net.

A slightly more advanced technique is to change your DNS servers to point to Afraid.org nameservers (ns[1-4].afraid.org). If you use your registrar for anything other than DNS you'll want to be careful here, you may need to set up an MX (mail) record or make other changes. This is a bit of a cleaner solution but you'll want a little more DNS technical knowledge to accomplish this.

5. Setting up IPv6 Reverse Records

By now you should have both an A record and a AAAA record for your domain. But if you do a reverse DNS lookup you'll probably discover two things. First, running **dig -x <ip_address>** on your IPv4 address will probably result in a hostname based on your ISP and location. Mine is a verizon.net hostname and there isn't much you can do about it. If you've got a commercial account you can talk to your ISP and ask about changing the PTR record, but for home users that's probably not a good idea.

But your IPv6 address may not even have a PTR record. If you've gotten your IPv6 address from SixXS (<http://www.sixxs.net>) you can fix that. First, log into your SixXS account and click on the Home button. You should see a list of the Tunnels and Subnets on your account. Click on the Details link for your subnet (it should look like R12345). Under Reverse Delegations enter ns1.afraid.org and then repeat for ns2, ns3 and ns4.afraid.org. Each of these entries will cost you one credit, but if you intend to be a long-time user of SixXS this is nothing and if you're only playing around with IPv6 the credits mean nothing to you anyway - just leave your connection up for two weeks if you need more.

Now you'll need to log into your Afraid.org account. On the left there's an IPv6 Reverse link. Add a record for your subnet (should be something like 2001:4830:xxxx::/48). Note that you may need to wait for SixXS's database to update before you do this, if it doesn't work just try again in a few hours. Now you can add a record for each of your hosts, for example I have one where I type in :80 as the host IP and www.ebower.com as the hostname. This tells Afraid.org to return www.ebower.com when someone does a reverse lookup on 2001:4830:113c:0000:0000:0000:0000:0080.

A. About Me

My name is Jeff Bower, I'm a technology professional (<http://www.linkedin.com/in/jdbower>) with more years of experience in the telecommunications industry than I'd care to admit. I tend to post with the username jdbower on various forums, including Komodo Kamado (<http://komodokamado.com/forum/>), Android Central (<http://forum.androidcentral.com/>), VirtualBox (<http://forums.virtualbox.org/>), and MakeMKV (<http://www.makemkv.com/forum2/>). Writing these documents is a hobby of mine, I hope you find them useful and feel free to browse more at <https://www.ebower.com/docs>.

I also enjoy cooking, especially outdoors with my Komodo Kamado (<http://www.komodokamado.com>) and using my Stoker (<https://www.rocksbarbque.com>). Take a look at my recipes stored at <https://www.ebower.com/recipes>.

If you've got any questions or feedback please feel free to email me at docs@ebower.com (<mailto:docs@ebower.com>) or follow me on Google+ (<https://profiles.google.com/100268310848930740059>) or Twitter (<http://twitter.com/jdbower>).