

## Installing OpenWrt via TFTP

[Go back to generic.flashing](#)

TFTP is a very simple protocol; simple enough to be implemented in small boot loaders. The basic idea is as follows:

1. Router is powered on
2. Bootloader prepares startup of firmware code
3. **For a few seconds** it initializes the wired lan ports
  - This doesn't happen instantly but a short time during startup/boot
  - The network settings are not always the same as OpenWRT's defaults
4. It listens to TFTP requests for transferring a flash image
5. Either
  - a. It receives an image to flash and flashes it
  - b. or continues booting the device normally if the TFTP communication timed out
    - Provided the currently loaded firmware is valid
    - If not valid behavior varies based on bootloader. Some wait for a flash via tftp while others just show an error diag light and wait for a reboot.
    - See [Bootloader functionality](#) for more details about the bootloader




### Warning!

This section describes actions that might damage your device or firmware. Proceed with care!

You must determine whether your hardware's bootloader has a TFTP client or server. Consult your specific model's OpenWRT Wiki page for details on necessary settings and the TFTP type offered if any.

This documentation will use example IP addresses according to RFC5737. Please consult your model's wiki documentation for actual IP addresses specific to your device.

 **Fix Me!** Cleanup IP addresses as 192.168.1.x is common, but not universal. Use of RFC5737 might be more appropriate or just italic *ip*v4.x.y.z indicators.

## Bootloader contains TFTP client

### Example1

For example the [bootloader implementation of the DIR-300 redboot](#) contains a TFTP client. Two steps:

1. first you install and start a TFTP server (or daemon) on your host and place the image(s) to be flashed in the <root directory> of this software (you may be required to do this as root). Example:

```
sudo apt-get install tftpd-hpa tftp
sudo cp ~/uboot/arch/arm/boot/uboot.img /var/lib/tftpboot
```

2. Test the server:

```
tftp localhost
tftp> get uboot.img
tftp> quit
cmp /var/lib/tftpboot/uboot.img uboot.img
# no output other then a prompt means it worked correctly
```

3. connect to the bootloader, and tell it to get the images on your harddisk via its TFTP client (in case of the DIR-300 you obtain a connection via telnet [\[https://en.wikipedia.org/wiki/telnet\]](https://en.wikipedia.org/wiki/telnet) on the non-default port 9000). Example:

```
telnet 192.168.20.81 9000
Redboot> load uboot.img
go
```

4. after successful installation of OpenWrt, do not forget to deactivate the TFTP server again!

### Example2

In case of the xxx Step 3 from Example 1 above is not applicable. There is no console to login to, the bootloader will automatically try to get a firmware over TFTP from a pre-configured IP address at every boot.

 **Fix Me!** TODO

## tftpd server on Mac OS X Lion

*Note: TftpServer.app places a pleasing GUI on top of the native OSX tftpd. There's a writeup of using TftpServer.app at <https://lede-project.org/docs/howto/tftpserver> [https://lede-project.org/docs/howto/tftpserver]. If you prefer to use the command-line, read on...*

OS X Lion comes with a tftpd but its disabled by default. Like most services in OS X, tftpd is controlled by launchctl. The configuration with which the daemon is launched is in **/System/Library/LaunchDaemons/tftp.plist** and the the identifier is **com.apple.tftpd**

before you make changes to the config run:

```
sudo launchctl unload -F /System/Library/LaunchDaemons/tftp.plist
```

then:

```
sudo launchctl load -F /System/Library/LaunchDaemons/tftp.plist
```

to stop tftpd run:

```
sudo launchctl stop com.apple.tftpd
```

to start tftpd run:

```
sudo launchctl start com.apple.tftpd
```

Here is an example config file that will work:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Label</key>
  <string>com.apple.tftpd</string>
  <key>ProgramArguments</key>
  <array>
    <string>/usr/libexec/tftpd</string>
    <string>-l</string>
    <string>-s</string>
    <string>/private/tftpboot</string>
  </array>
  <key>inetdCompatibility</key>
  <dict>
    <key>Wait</key>
    <true/>
  </dict>
  <key>InitGroups</key>
  <true/>
  <key>Sockets</key>
  <dict>
    <key>Listeners</key>
    <dict>
      <key>SockServiceName</key>
      <string>tftp</string>
      <key>SockType</key>
      <string>dgram</string>
    </dict>
  </dict>
</dict>
</plist>
```

Differences from the default include removing this, to enable the service:

```
<key>Disabled</key>
<true/>
```

Add this to the ProgramArguments array to make it log to **/var/log/syslog.log**

```
<string>-l</string>
```

Place the openwrt image file you want to serve in:

```
/private/tftpboot
```

Notice that even after running **launchctl start com.apple.tftpd** you will not see tftpd running when executing **ps aux | grep tftpd** because of the way launchctl works. tftpd is in fact **not** running but launchctl will launch it as soon as it is required.

In some cases, when the output on the serial console is grabbed you can still act on faith and execute the following commands, which will work in most cases:

```
setenv ipaddr 192.168.1.1

setenv serverip 192.168.1.100

tftpboot 0x80000000 openwrt-xxx-generic-xxx-squashfs-factory.bin

erase 0x9f020000 +0x332004

cp.b 0x80000000 0x9f020000 0x332004

boot.m 0x9f020000
```

---

## Bootloader contains TFTP server

The basic procedure of using a tftp client to upload a new firmware to your router:

1. Unplug the power to your router
2. Plug the Ethernet connection from your computer (acting as tftp client) into a LAN port on your model. See your model's OpenWRT wiki for details on which port to use
3. Start your tftp client on your computer
4. Give it the router's address (specific to model and bootloader, see your model's wiki page)
5. Set mode to octet/binary
6. Tell the client to resend the file until it succeeds
7. "put" the file
8. Plug-in your router while having the tftp client running and constantly probing for a connection
9. The tftp client will receive an ack from the bootloader and starts sending the firmware

### Tips

- **⚠ Please be patient**, the reflashing occurs *after* the firmware has been transferred. In most cases the router will automatically reboot itself. Some models do not reboot so wait at least 15 minutes before power cycling them.
- **⚠ Note that the bootloader usually does not use the IP address or MAC address stored in nvram**, it will revert to a bootloader default instead. See your model's wiki documentation for specifics.
- **⚠ Put a hub or switch between the router and the computer**, this will make sure that the local computer link is up before the *boot\_wait* period is passed. This is a requirement to make TFTP work on computers where the local link is brought up too late and is usually simpler than trying to force the link to stay up instead.
- On routers with a DMZ LED, OpenWrt will light the DMZ LED while booting, after the bootup scripts are finished it will turn off the DMZ LED.

---

The TFTP commands vary across different implementations. Here are some examples:

### Linux/BSD

The network link must be up and established during power up. One way to ensure this happens is to use a switch or hub inbetween your computer and the device you are flashing as this will leave the link established when you power off the device.

Another option is to disable network manager in Linux (or use a distro/LiveCD that doesn't have it). Some commands that may disable it (depends on the distribution of Linux used):

- `/etc/init.d/networking stop`
- `/etc/init.d/network stop`
- `/etc/init.d/NetworkManager stop`
- `service networking stop`
- `service network stop`
- `service NetworkManager stop`

### Preparation Steps

1. Configure a static IP to match your \*bootloaders\* network
  - `ifconfig eth0 ipv4.x.y.z netmask 255.255.255.0`
  - See your device model's OpenWRT wiki page for specific settings
2. Sometimes it helps to have a preconfigured arp entry to increase your chances of catching the tftp window
  - `arp -s ipv4.x.y.1 20:aa:bb:cc:dd:00`
    - Check your device model's OpenWRT wiki page for correct IP and MAC addresses

### Using atftp

Advanced TFTP [<http://freecode.com/projects/atftp>]

Source Code care of OpenWRT [<http://downloads.openwrt.org/sources/atftp-0.7.tar.gz>]

As a single command-line:

```
atftp --trace --option "timeout 1" --option "mode octet" --put --local-file openwrt-xxx-x.x-xxx.bin IPv4.x.y.z
```

Step by step:

```
atftp
connect IPv4.x.y.z
mode octet
trace
timeout 1
put openwrt-xxx-x.x-xxx.bin
```

#### Using netkit's tftp

As a single command-line:

```
echo -e "binary\nrexmt 1\ntimeout 60\ntrace\nput openwrt-xxx-x.x-xxx.bin\n" | tftp IPv4.x.y.z
```

Step by step:

```
tftp IPv4.x.y.z
binary
rexmt 1
timeout 60
trace
Packet tracing on.
tftp> put openwrt-xxx-x.x-xxx.bin
```

Setting "rexmt 1" will cause the tftp client to constantly retry to send the file to the given address. As advised above, plug in your box after typing the commands, and as soon as the bootloader starts to listen, your client will successfully connect and send the firmware.

Some devices will also respond to ping while others do not.

Note: for some versions of the CFE bootloader, the last line may need to be "put openwrt-xxx-x.x-xxx.bin code.bin". If this does not work try other variations instead of code.bin - e.g. openwrt-g-code.bin or openwrt-gs-code.bin.

One CFE version only worked after renaming the '....bin' file to 'code.bin'. From Linux Ubuntu I then used the command 'tftp -m binary 192.168.1.1 -c put code.bin' and the transfer process came to life.

#### Using curl

```
curl -T openwrt-xxx-x.x-xxx.bin tftp://IPv4.x.y.z
```

## MacOS X

On Mac OS X, you should be able to flash the router with the command line tftp client, which behaves identically to netkit's tftp above.

Some people have had problems with the command line tftp client, however, and recommend using MacTFTP Client [<http://www.mactechnologies.com/index.php?page=downloads#mactftpclient>] instead:

- Download, install, and open MacTFTP
- Choose Send
- Address: *Your bootloaders IP address*
- Choose the openwrt-xxx-x.x-xxx.bin file
- Click on start while applying power to the WRT54G

Many Macs will disable the Ethernet card when the router is powered off and will take too long to re-enable the card, causing the TFTP transfer to fail with an "Invalid Password" error. Many people have had success if they manually configure their network card (in the "Ethernet" tab of "Built-in Ethernet" in System Preferences' Network panel) to:

- Configure: Manual (Advanced)
- Speed: 10 BaseT/UTP
- Duplex: full-duplex

Alternatively, you can connect the router to the Mac via a hub or switch; see below for more information.

## Windows

There are multiple tftp clients that you can choose from. See the [What TFTP client should I use to flash my device?](#) section for options.

1. Plug in your Windows network interface into the appropriate port on the device you will be flashing
2. Static IP your wired Ethernet interface to an appropriate IP address on the same network as your **bootloaders** network

- Example: Your bootloader has an IP of 192.0.2.1 (netmask of 255.255.255.0) so your Windows network configuration would use 192.0.2.2/24
  - See your device's OpenWRT wiki page for specific settings
3. Open an elevated command prompt
    - Start, Run, "cmd" on Windows 2000,XP,2003
    - Start, search on "cmd", ctrl+shift+enter on Vista, 7, 2008, etc.
  4. `arp -s <bootloaders ip address> <bootloaders mac address> <your Windows static IP address>`
    - Assists in avoiding delay in reaching tftp server on device
    - Windows MAC addresses use dashes instead of colons (ex: 00-00-00-00-00-00)
    - Windows 7 and other similar versions may require
      - `netsh interface ipv4 add neighbors "Local Area Connection" <bootloaders ip address> <bootloaders mac address>`
  5. Disable Windows media sensing (shouldn't be necessary if you have a switch in the middle)
    - `netsh interface ipv4 set global dhcpmediasense=disabled`
    - `netsh interface ipv6 set global dhcpmediasense=disabled`
  6. Disable Windows firewall and any other firewalls on your client machine
  7. Unplug your router
  8. Run your tftp program
    - See [What TFTP client should I use to flash my device?](#) section for general usage instructions
  9. Plug in router immediately after tftp program begins put attempts
  10. Flash usually takes a few minutes. See your device's OpenWRT wiki page for specifics
  11. You will probably want to re-enable Windows media sensing and revert your other network changes

## Troubleshooting

Don't forget about your firewall settings, if you use one. It is best to run the "put" command and then immediately apply power to the router, since the upload window is extremely short and very early in boot.


TFTP Error	Reason
Code pattern is incorrect	The firmware image you're uploading was intended for a different model.
Invalid Password	The firmware has booted and you're connected to a password protected tftp server contained in the firmware, not the bootloader's tftp server.
Timeout	Ping to verify the router is online Try a different tftp client (some are known not to work properly)
Timeout	Ping to the router works NetworkManager (Linux) may still be running causing autosense. Try again with manual configuration.


Some machines will disable the ethernet when the router is powered off and not enable it until after the router has been powered on for a few seconds. If you're consistently getting "Invalid Password" failures try connecting your computer and the router to a hub or switch. Doing so will keep the link up and prevent the computer from disabling its interface while the router is off.

Before you go searching for a hub to keep your link live, try setting your TCP/IP setting to a static IP (192.168.1.10; 255.255.255.0; 192.168.1.1 [gateway]) method instead of DHCP.

**! If you can flash your router and after that it says "Boot program checksum is invalid" or "Invalid boot block on disk" on serial console try a different tftp client - *atftp* works well. This occurs with some *netkit* tftp packages and big firmwares.**

### Example

 would this be better to just exist in specific model's wiki pages?

 should we create a page to list models with tftp support, noting which ones need the reset button trick?

On many routers, including the Asus WL-500g Premium v1 that I use, you flash an image by disconnecting power, press and hold down the reset button, and connect the power again. Wait a few seconds and the PWR LED will start to blink. Release the reset button. The device will now have a TFTP server running on 192.168.1.1.

**Note that many TP-Link models** are reported to support the same trick, including the TL-WR740Nv4, TL-WDR4300v1, TL-WDR3600v1, TL-WR842NDv1, TL-WR841NDv8, TL-MR3020v1, TL-MR3220v2, TL-MR3420v2, TL-WR940Nv2, TL-WR941NDv5, TL-WR1042NDv1 and possibly any other TP-Link model that has a recent firmware upgrade from the manufacturer. For a summary and ongoing experiments, see: <http://bkil.blogspot.com/2014/12/hidden-tftp-of-tp-link-routers.html> [<http://bkil.blogspot.com/2014/12/hidden-tftp-of-tp-link-routers.html>]

You'll have to use a Ethernet cable at this point. Connect it to LAN1-LAN4, not WAN. Configure your local machine on the 192.168.1.x/24 network, for example as 192.168.1.42. The router will use 192.168.1.1.

```
$ tftp 192.168.1.1
tftp> trace
Packet tracing on.
```

```
tftp> binary
tftp> put openwrt-brcm-2.4-squashfs.trx
sent WRQ <file=openwrt-brcm-2.4-squashfs.trx, mode=octet>
received ACK <block=0>
sent DATA <block=1, 512 bytes>
received ACK <block=1>
sent DATA <block=2, 512 bytes>
received ACK <block=2>
sent DATA <block=3, 512 bytes>
received ACK <block=3>
sent DATA <block=4, 512 bytes>
...
received ACK <block=4742>
sent DATA <block=4743, 512 bytes>
received ACK <block=4743>
sent DATA <block=4744, 512 bytes>
received ACK <block=4744>
sent DATA <block=4745, 0 bytes>
received ACK <block=4745>
Sent 2428928 bytes in 6.2 seconds
tftp> quit
$
```

Wait one minute and restart the box by disconnecting and reconnecting power. Some documentations claim that the device should restart by itself but I have never seen this happen, no matter how long I wait.

## What TFTP client should I use to flash my device?

Which ever you want! Some suggestions are given below:

### GNU/Linux

- Some people suggest `atftp` with GNU/Linux

### Windows

- DD-WRT tftp GUI [<ftp://ftp.dd-wrt.com/others/tornado/Windows-TFTP/tftp2.exe>]
  1. Server is the IP address of your **bootloader** tftp server
  2. Password is typically blank
  3. Select the firmware file
  4. Set retries to 20 or more (most of the time you get it in 3)
  5. Click on Upgrade and it will constantly retry until it gets it
- Windows 2000 and Windows XP have a built-in TFTP client and it can be used [<http://martybugs.net/wireless/openwrt/flash.cgi>] to flash with OpenWrt firmware.
- `tftpd32` (use **client mode!**) from <http://perso.wanadoo.fr/philippe.jounin/tftpd32.html> [<http://perso.wanadoo.fr/philippe.jounin/tftpd32.html>]

### TFTP command line client short Instructions

1. Open a command window (`cmd.exe`)
2. `"tftp -i <bootloader IP tftp server address> PUT OpenWrt-gs-code.bin"`
3. Now you may plug in the router (unplug it first if it was plugged).

Note that some bootloaders do not respond to ping.

### MAC OS X

## Getting tftp to work in MacOSX "Lion"

If you get "tftp: timeout", use below

below from <http://forums.creativecow.net/thread/180/857349> [<http://forums.creativecow.net/thread/180/857349>]

```
sudo launchctl load -F /System/Library/LaunchDaemons/tftp.plist
```

```
sudo launchctl start com.apple.tftpd
```

You will get errors like this

below from <http://www.linuxsmiths.com/blog/?p=427> [<http://www.linuxsmiths.com/blog/?p=427>]

```
tftp: server says: File not found
```

to fix it use

cd /var/tftpboot touch (the file in your tmp dir that you are going to send)

chmod 777 (the file in your tmp dir that you are going to send)

then do the tftp command again (ie: # tftp -p -l /tmp/mtd1 192.168.2.2)

TODO

## tftpd on MacOS 10.4 "Tiger"

tftpd works out-of-the-box also on the old 10.4. Maybe the tftp dir is not yet created but this is just a mkdir. Get a root shell and issue these commands:

```
bash-4.2$ sudo bash
Password:
bash-4.2# mkdir -p /private/tftpboot/
bash-4.2# cp /path/to/openwrt-image /private/tftpboot/
bash-4.2# launchctl load -F /System/Library/LaunchDaemons/tftp.plist
bash-4.2# ps aux|grep ftp
root      23494   0.0  0.0      27696    152  ??  Ss   4:34PM    0:00.00 launchctl load -F /System/Library/LaunchDaemons/tftp.plist
root      23496   0.0  0.0      38604      4  p3  R+   4:34PM    0:00.00 grep ftp
bash-4.2# launchctl start com.apple.tftpd
bash-4.2# ps aux|grep ftp
root      23494   0.0  0.0      27696    152  ??  Ss   4:34PM    0:00.00 launchctl load -F /System/Library/LaunchDaemons/tftp.plist
root      23498   0.0  0.0      27244    464  ??  Ss   4:34PM    0:00.01 /usr/libexec/launchproxy /usr/libexec/tftpd -i /private/tftpboot
root      23500   0.0  0.0      38604      4  p3  R+   4:34PM    0:00.00 grep ftp
bash-4.2# tftp 192.168.100.72  ### just testing
tftp> get openwrt-ar71xx-generic-hornet-ub-squashfs-sysupgrade.bin
Received 7270950 bytes in 2.7 seconds
tftp>
```

## File Permissions

Check if your TFTP Server has sufficient access rights to files or directories. U-Boots TFTP Client / tftpboot can complain with:

```
## Error: 'Access violation' (2), starting again!
```