

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

Contents

- [1 Introduction](#)
- [2 Definitions](#)
- [3 Diagnosis](#)
- [4 Solution](#)
 - ◆ [4.1 Increasing Max Connection count beyond limits of GUI](#)
- [5 Checking the Results](#)
- [6 Commonly mentioned out-dated solution & The modern equivalent](#)
 - ◆ [6.1 How It Works](#)
 - ◆ [6.2 Script](#)
- [7 AIM, ICO and other IM programs](#)
- [8 More help](#)
- [9 Bug in older WRT54G/S](#)
- [10 More help: Adjust your P2P applications](#)
- [11 More help: Heat-sinks](#)
- [12 More Help: Web Traffic Daemon on v24x](#)
- [13 Known issues with Applications/Operating Systems](#)
- [14 p2p protocoll info](#)

Introduction

Issues occur with many routers, *including routers running DD-WRT*, when using the router with heavy P2P applications. The router becomes 'slow' over time, and restarting helps for a short time. Some symptoms can be:

- Slow web-interface, or cannot connect at all to web interface
- Slowing transfer of data, e.g. browsing, after a reboot
- Not responding to ping
- Router Crash or even rebooting

Usually the culprits are heavy P2P software like Emule, Bittorrent, uTorrent, Azureus, Shareaza or something similar. These programs, by default, can require a lot of connections which could cause the routers' ip_contrack table to get full.

- Especially BitTorrent's DHT feature sends thousands of UDP packets that quickly overflow this table.

Routers affected with this issue are the most common types of routers running DD-WRT.

Definitions

P2P

Peer to Peer applications. Could mean eMule, uTorrent, Azureus. Within the context of this article does not mean Skype, or other 'lite' P2P applications.

Router_Slowdown

DD-WRT Device

A Router running third-party, DD-WRT firmware. Although configuration in other methods besides 'router' is possible, such a 'bridge', this article and phenomena only occurs in the default router configuration of DD-WRT.

Web Interface

The default management method, open a browser to <http://192.168.1.1> or your custom DD-WRT Device's LAN IP.

Diagnosis

When this 'slowdown' occurs and the router doesn't respond to pings nor Web Interface requests, you still can check what's going on:

- Close all your P2P/network applications and wait a few minutes for connections to be freed.
- Try to use the [CLI](#) to communicate with your router.
 - ◆ If that's not possible, reboot the DD-WRT Device.
 - ◆ Check to see if your problem is caused by TCP or UDP connections. From the CLI:

```
cat /proc/sys/net/ipv4/netfilter/ip_conntrack_max
4096 # Here the router displays the maximum number of possible connections.
grep -c ^udp /proc/net/ip_conntrack
3693 # Here UDP is using 3693 entries
grep -c ^tcp /proc/net/ip_conntrack
115 # and TCP uses only 115 entries
```

◇ You may also view syslog messages:

```
cat /var/log/messages
```

1. First, you'll see 'full, dropping packet.' messages followed by 'messages suppressed.'
2. Eventually, it will get verbose and you'll see logs similar to the following :

```
<4>kernel: ip_conntrack: table full, dropping packet.
<4>kernel: NET: 15 messages suppressed.
<4>kernel: ip_conntrack: table full, dropping packet.
<4>kernel: NET: 12 messages suppressed.
```

Solution

After you have rebooted and turned off any heavy P2P applications:

- Go to the Web Interface of the DD-WRT Device and log in
- Go to 'Administration'
- Go to 'Management'
- Enter the following values at 'IP Filter Settings'
 - ◆ **Maximum Ports:** 4096 (For an 8MB RAM model set it no higher than 1024)
 - ◆ **TCP Timeout (s):** between 300 to 900 (higher is safer, lower can forget connections too quickly, DO NOT EVER GO BELOW 300 (5 minutes)!!)
 - ◆ **UDP Timeout (s):** 30

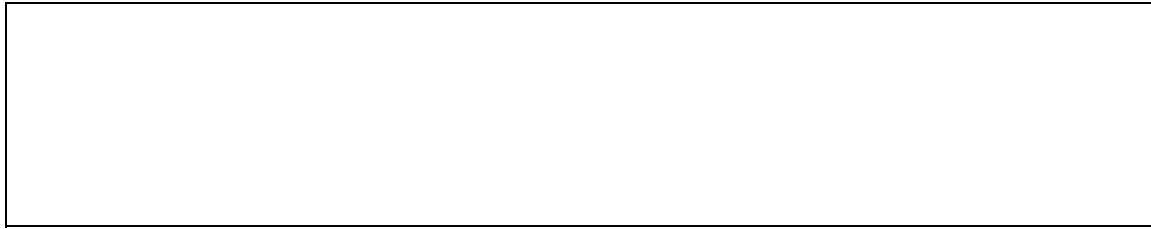


Image: IP Filter Settings on DD-WRT v24 svn 12548

- Save Settings and then Reboot Router

The above settings control how long it takes before inactive TCP & UDP connections are forgotten by the router. If you set them too low then the router will forget connections too quickly and active connections will be dropped.

- **If you start getting frequent disconnections in particular programs but not others (commonly Instant Messenger programs) then raise your TCP timeout to at least 600 or more.**

If you want about 15% more RAM available, which can help the router reach the theoretical maximum of 4096 ports:

- First, make sure the CLI is enabled. This way, it is possible to connect to the DD-WRT Device and turn the Web Interface back on when needed.
- Then turn off the Web Interface. This is done through the Web Interface. Apply your changes.
- When starting the Web Interface on an as-needed basis, use the command 'httpd'. To close it later, use the command 'killall httpd'.

At this point, you are finished. If you want to look at more technical information, and the technical results, more information is presented below.

Increasing Max Connection count beyond limits of GUI

Please note that this is more or less experimental and may compromise your router's stability.

From the thread: >>Here is a solution for increasing the maximum number of connections in Kernel 2.6: K2.6 is very different regarding ip_contrack_max (IP Filter Maximum Ports in the GUI) and it's hash table compared to the older kernel.

ip_contrack_max sets the maximum number of connections that can be kept at one time. Most people here seems to decrease the default timeout (3600s) for these connections (which results in other problems) instead of increasing the maximum value.

On routers with large amount of RAM (64 MB+ for example) one could increase this by 10 times from the default max of 4096 without any problem.

From what I understand you also have to increase the hash table where these are stored in order to benefit from this increase. In the older kernel this was not possible since one would have to set this value before booting (and there was no nvram setting for that?).

Router_Slowdown

Although in K2.6 one can change booth of these in realtime dynamically without restarting any process. A hashsize equally large to the conntrack_max, has the best performance as I understand. Also since K2.4.21 the hashsize performs best with a value that is a power of 2 ex. $2^{14} = 16384$. (I have also used this as an example)

Since K2.4.23 (and newer), to change ip_conntrack_max:

```
echo "16384" > /proc/sys/net/ipv4/netfilter/ip_conntrack_max
```

Since K2.6.20 (and newer), to change the hashsize:

```
echo "16384" > /sys/module/nf_conntrack/parameters/hashsize
```

But none of these will be stored after a reboot, since the values are stored in RAM.

Permanently change ip_conntrack_max, ether use the webgui or write:

```
nvramp set ip_conntrack_max=16384
nvramp commit
reboot
```

I haven't found any way to permanently change the hashsize parameter. Although since you now in K2.6 can change it in realtime, you can just add it to your startup script and it should do the job.

```
echo "16384" > /sys/module/nf_conntrack/parameters/hashsize
```

Checking the Results

Besides an obvious improvement in performance, there is a way to check via [CLI](#) the technical results:

```
cat /proc/sys/net/ipv4/netfilter/ip_conntrack_tcp_timeout_established
90
cat /proc/sys/net/ipv4/netfilter/ip_conntrack_udp_timeout_stream
90
```

Commonly mentioned out-dated solution & The modern equivalent

The following command is often recommended in forums, but only applies to older versions of the DD-WRT firmware up to V23SP1:

```
echo '600 1800 120 60 120 120 10 60 30 120' > /proc/sys/net/ipv4/ip_conntrack_tcp_timeouts
echo '30 60' > /proc/sys/net/ipv4/ip_conntrack_udp_timeouts
```

Router_Slowdown

Note: The method above, for resolving slowdown issues, does not apply for DD-WRT v23 SP2 firmware and beyond. Furthermore, this command only applies to TCP traffic issues only, not UDP issues.

The newer firmwares' (V23SP2 and newer) 2.4.23 kernel ignores both of these files. For this method to work in later kernels, the timeout values must be placed in individual files. The correct files are located in `/proc/sys/net/ipv4/netfilter/` as shown below.

How It Works

Few people know what this line actually does. It is a list for all possible TCP states and their timeouts in seconds. See the table below (they are in the correct order):

State	Timeout value
NONE	10 minutes
ESTABLISHED	30 minutes (default is 5 days)
SYN_SENT	2 minutes
SYN_RECV	60 seconds
FIN_WAIT	2 minutes
TIME_WAIT	2 minutes
CLOSE	10 seconds
CLOSE_WAIT	60 seconds (default is 12 hours)
LAST_ACK	30 seconds
LISTEN	2 minutes

The full list of files in the `/proc/sys/net/ipv4/netfilter/` directory that one can populate to get the same effect on v23 SP2 and later firmwares is:

```
/proc/sys/net/ipv4/netfilter/ip_conntrack_generic_timeout
/proc/sys/net/ipv4/netfilter/ip_conntrack_icmp_timeout
/proc/sys/net/ipv4/netfilter/ip_conntrack_tcp_timeout_close
/proc/sys/net/ipv4/netfilter/ip_conntrack_tcp_timeout_close_wait
/proc/sys/net/ipv4/netfilter/ip_conntrack_tcp_timeout_established
/proc/sys/net/ipv4/netfilter/ip_conntrack_tcp_timeout_syn_sent
/proc/sys/net/ipv4/netfilter/ip_conntrack_tcp_timeout_syn_recv
/proc/sys/net/ipv4/netfilter/ip_conntrack_tcp_timeout_fin_wait
/proc/sys/net/ipv4/netfilter/ip_conntrack_tcp_timeout_time_wait
/proc/sys/net/ipv4/netfilter/ip_conntrack_tcp_timeout_last_ack
/proc/sys/net/ipv4/netfilter/ip_conntrack_udp_timeout
/proc/sys/net/ipv4/netfilter/ip_conntrack_udp_timeout_stream
```

Script

The following script should set these values automatically. It can be installed by copying it into Administration -> Commands and clicking Save Startup.

If anyone's wondering, the 'prefix' var is there to save space in nvram.

```
prefix=/proc/sys/net/ipv4/netfilter/ip_conntrack

echo 600 > ${prefix}_generic_timeout
echo 30 > ${prefix}_udp_timeout
echo 60 > ${prefix}_udp_timeout_stream
echo 1800 > ${prefix}_tcp_timeout_established
```

Router_Slowdown

```
echo 120 > ${prefix}_tcp_timeout_syn_sent
echo 60 > ${prefix}_tcp_timeout_syn_recv
echo 120 > ${prefix}_tcp_timeout_fin_wait
echo 120 > ${prefix}_tcp_timeout_time_wait
echo 10 > ${prefix}_tcp_timeout_close
echo 60 > ${prefix}_tcp_timeout_close_wait
echo 30 > ${prefix}_tcp_timeout_last_ack
```

AIM, ICQ and other IM programs

If you're using certain IM clients such as ICQ or AIM (or other applications which do not send frequent keep-alive packets), you should set TCP timeout to 300 or 600 to help them stay connected. As higher values for TCP may cause hangups when you're dealing with very heavy P2P traffic, you might want to experiment with these values.

More help

DD-WRT has an inbuild proxy feature that allows rewriting of HTML content to filter ActiveX cookies, etc. As this is load-intensive, you may want to disable this feature, to give the DD-WRT Device more time to deal with open port requests. This should be done under Security->Firewall.

Unfortunately this function has 3 bugs:

- When unchecking all 4 checkboxes the rewriting proxy may still be running (you might have to reboot)
- The proxy drops connections if under heavy load
- One user notes: It probably runs haywire checking traffic on other ports than 80, hence the slowdown with P2P applications (unconfirmed)

A solution is to disable the firewall (which is *not* recommended): Security->Firewall. This can also fix slow-downs.

(This behavior has been seen in v22 and v23. Please report this to the maintainer and put it in the bug database.)

Bug in older WRT54G/S

Apparently there is a bug in the WRT54GS v2.2. The bug involves a fatal memory access error due to a difference in the CPU clock speed and the clock speed on the memory bus. This bug causes connections to be dropped when transferring large files. New versions of the Linksys firmware overclock the CPU to 216 MHz. This can be fixed by logging into the router via SSH or Telnet and executing the following commands:

```
nvramp set clkfreq=216
nvramp commit
reboot
```

Router_Slowdown

When running DD-WRT micro v23sp2 on a WRT54G v5 the clock speed can be adjusted by logging into the router via SSH or Telnet and executing the following commands:

```
nvramp set overclocking=216
nvramp commit
reboot
```

After rebooting the router you can confirm that the new clock speed has applied by checking the Status page of the DD-WRT GUI or by logging in via Telnet and executing the following:

```
nvramp get clkfreq
```

More help: Adjust your P2P applications

For P2P applications, and especially Bittorrent, the problem can be an overwhelming number of incoming connections. Try disabling DHT and/or setting a limit to the number of peers/incoming connections to around 100 or so. See the Azureus page regarding this.

More help: Heat-sinks

Consumer routers are not designed for high load applications, but instead for low cost. They often lack heatsinks on the main chips as a cost and space saving measure. Unfortunately, they can overheat when dealing with the load of continuous, heavy traffic.

Installing heat sinks, like those used for desktop system chipsets but smaller, does help this problem. A small amount of heat-transfer solution and some cyanoacrylate [the chemical name for SuperGlue-style glues, although much better quality versions can be found at hobby and electronics stores and online]. Clearance [distance to the inside-top of the router case] is a consideration.

Note that some Linksys routers in particular are intended to be assembled, not disassembled. You may need to use glue or other ugly ways to get the case back together after opening.

More Help: Web Traffic Daemon on v24x

If you are unable to load the WebUI and have already tried the TCP timeout tweaks, try turning off the WAN Traffic Daemon. This has been known to cause problems up to v24 for the D-Link DIR-300.

Known issues with Applications/Operating Systems

Applications

- When using Steam (of Valve) and when you refresh your server list, it might cause your router in the default setup, to reboot since it opens many connections. You can remedy this issue by following the steps on this Steam suport page:
https://support.steampowered.com/kb_article.php?ref=4487-YUOC-6922

p2p protocoll info

```
applejuice
audiogalaxy tcp
ares        tcp
bearshare   tcp
bittorrent  tcp/udp
directconnect tcp
edonkey     tcp/udp
fasttrak/kazaa tcp/udp
freenet
gnucleuslan
gnutella    tcp
goboogy
hotline
imesh
kugoo
mute        tcp
napster     tcp
openft      tcp
pando       tcp
soribada    tcp
soulseek    tcp
tesla
waste
winmx       tcp
xdcc
xunlei      tcp/udp
```