

---

# Monitoring an ADSL link using a simple script

Aaron Hill  
Paul Hoadley

Copyright © 2002 Aaron HillPaul Hoadley

\$Date: 2003/03/31 10:30:33 \$

This article describes a simple but effective method for monitoring the state of an ADSL link (or, in fact, any PPP or PPPoE link using `/usr/sbin/ppp`). It has been observed that in at least one mode of operation (`-ddial`), the `/usr/sbin/ppp` program can become irretrievably wedged when the link goes down. The only solution seems to be killing and restarting the process. This article contains and describes the usage of a script that will periodically check the state of the link, and kill and restart `ppp` if required.

## 1. Operation of the script

The script obtains the default gateway from the `tun0` interface and attempts to ping it. If the ping fails it will ping a secondary IP address which you hard code in the script. If both these tests fail a certain number of times (2 by default) it restarts the `ppp` daemon. It will keep bouncing `ppp` until the ping test works again.

The script keeps a count of how many times the ping test failed so when the ping test works it can send you an email to let you know what happened and how many times the test failed. By default this email goes to `root`. Based on how often you're running the script from cron you'll be able to tell the length of the outage from this email.

## 2. Installing and customising the script

The following script should be installed as `pingmonitor.sh` in the `/usr/local/etc/` directory.

## Monitoring an ADSL link using a simple script

---

```
#!/bin/sh

#
# This script tests for network connectivity and restarts ppp if it is found
# to be down.
#
# --- User-modifiable variables ---

# Use this IP address if the primary address cannot be determined
secondaryaddr="139.134.2.2"

# Number of failed pings required to signify link failure
failedtrigger=2

# File to keep track of total number of failed pings
failedcountfile="/usr/local/etc/pingmonitor.missedping.count"

# Email address to send reports to
emailaccount="root"

# Set to the appropriate label in /etc/ppp/ppp.conf
isp="bigpond"

# These options will be given to /usr/sbin/ppp
ppp_opts="-quiet -ddial -nat"

# --- End of user-modifiable variables ---

# Load in system configuration.
if [ -f /etc/defaults/rc.conf ]; then
. /etc/defaults/rc.conf
source_rc_confs
elif [ -f /etc/rc.conf ]; then
. /etc/rc.conf
fi

# only continue if the ppp link should be up
if [ ! $ppp_enable ]; then
# PPP is not configured in the rc files.
exit 0
elif [ "$ppp_enable" = "NO" ] || [ "$ppp_enable" = "no" ]; then
# PPP is not wanted
exit 0
fi

# Set umask
umask 137

# Determine the default gateway for the ADSL link
primaryaddr=`ifconfig tun0 | grep 'inet ' | grep -v 255.255.255.255 | tail -1 | cut -f 2 -d '>'`

if [ "$primaryaddr" = "" ]; then
primaryaddr="0.0.0.0"
secondaryaddr="0.0.0.0"
fi

# Check if we've had any previous failures
if [ -f $failedcountfile ]; then
pingfailed=`head -n 1 $failedcountfile`
else
pingfailed=0
fi
```

## Monitoring an ADSL link using a simple script

---

```
# Run the ping for the primary address - our default gateway.
/sbin/ping -c 5 -t 5 -q -m 2 $primaryaddr > /dev/null 2> /dev/null

# If the ping failed. Check to see if the gateway is filtered.
if [ $? -ne 0 ]; then
# Try to pull a TTL EXCEEDED message from the gateway.
if [ ` /sbin/ping -c 1 -m 0 -n -t 1 $primaryaddr 2> /dev/null | grep -i "time to live exceeded"
ping_error=0
else
ping_error=1
fi
else
# No filtering. The default gateway responded to the initial ping.
ping_error=0
fi

#
# Ping returns a non-zero error condition if ALL the ECHO_REQUEST packets did
# not return a ECHO_REPLY. If we received just one answer then ping returns
# a zero error condition which is perfect for our tests.
#

# Check the ping status and try pinging the secondary address if it failed
if [ $ping_error -ne 0 ]; then
/sbin/ping -c 5 -t 5 -q -m 7 $secondaryaddr > /dev/null 2> /dev/null

if [ $? -eq 0 ]; then
ping_error=0
else
ping_error=1
fi
fi

# Test the error condition
if [ $ping_error -ne 0 ]; then
# Update and record the failure count
pingfailed=$(( $pingfailed + 1 ))
echo $pingfailed > $failedcountfile

# Test if we've hit our failure trigger
if [ $pingfailed -ge $failedtrigger ]; then
# time to restart ppp so kill it first
/usr/bin/killall ppp > /dev/null 2> /dev/null

# wait for it to die
sleep 5

# really ensure ppp is dead - we can't risk two running
/usr/bin/killall -9 ppp > /dev/null 2> /dev/null

# wait again
sleep 5

# start up ppp again
/usr/sbin/ppp $ppp_opts $isp > /dev/null 2> /dev/null

# our work here is done
fi
else
# the ping worked so check if we've just recovered from a failure
```

## Monitoring an ADSL link using a simple script

---

```
if [ $pingfailed -ge $failedtrigger ]; then
# we have just recovered so let the admin know
echo "PING test failed $pingfailed times" | /usr/bin/mail -s "PPP restart on `hostname -s`" `hostname -s`
fi

# all's well now so remove the failure count
rm -f $failedcountfile > /dev/null 2> /dev/null
fi

# that's it
```

Use the following steps to modify the script for use:

- Do a **traceroute** to anywhere over your ADSL connection to find the second hop IP address.  
  
Add this second hop IP address to the script in the variable `secondary-addr`. Alternatively you can use any external, reliable IP address like the Telstra DNS server on `139.130.4.4`.
- Set the `failedtrigger` variable to a value indicating the threshold for link failure. The script will accept that number of failures of consecutive ping tests before concluding that the link is down. A good default value is probably 2.
- Change the filename and path in the variable `failedcountfile` if there is a more appropriate place on your system.
- Change the variable `emailaccount` to an email address you'd like the error reports to go to.
- The variable `isp` should be set to correspond to a label in the file `/etc/ppp/ppp.conf`. This label name will be passed to `/usr/sbin/ppp`.
- Set the variable `ppp_opts` to contain any options that should be passed to `/usr/sbin/ppp`.
- Make sure you can manually ping the secondary IP address. You might have to modify your firewall rules depending on your setup.
- Make sure you can manually ping your default gateway and/or your firewall allows it to send TTL EXCEEDED messages to the ADSL interface. This is `icmptype 11` for **ipfw**.

## 3. Running and testing the script

The script should be run periodically using **cron**. Add the following entry to `/etc/crontab`:

```
# PING Monitor - check the ADSL connection every two minutes
0-59/2 * * * * root /usr/local/etc/pingmonitor.sh
```

Restart **cron** by running: **kill -HUP cron**.

Test the script. Try pulling the phone line from the back of the modem or such for enough time to trigger the script.

## Contacting the Author

The author of this document is **Paul A. Hoadley**. The author of the `ping-monitor.sh` script is **Aaron Hill**. Feel free to send details of any errors in this document by email.