Connection Troubleshooting

This guide is intended for older hardware with custom mesh software that we no longer use. It is not applicable to the RouterOS software that is currently in use by the majority of nodes.

This page should be updated to cover troubleshooting and watchdog configuration for RouterOS

Wireless networks have a bit of a reputation for instability. Modern hardware has fixed most hardware problems, but there is work that needs to be done to make the firmware reliable. You can do this with "watchdog" scripts. I haven't had to reboot a router that is running our watchdog script.

Our firmware image (based on qMp) comes with a "bmx6health" script that checks whether the mesh software is running correctly and restarts it if necessary. This script by default runs once per day. I've found it better to run this every 5 minutes. You can do this by editing the crontab-

ssh into the router and in the terminal-

```
crontab -e
```

This opens a vi editor and you can change or add different scripts to run at different times. (The vi commands you need are "i" to insert, "esc" to stop editing, and ":x" to save and eXit.)

For some nodes, their main purpose is to be an internet gateway. To ensure that they always try to be online, you can add a watchdog script that pings a known website and calls "network restart" if it fails. These kind of scripts often ping 8.8.8.8, which is Google's DNS server.

I've discovered 3 ways to recover a qMp mesh router that has functioning wifi but has lost internet- `network restart`, `bmx6 restart` and restarting dnsmasq-`killall dnsmasq; dnsmasq start`. Sometimes the dns forwarder, dnsmasq will stop working correctly letting you ping some things and not others. dnsmasq will then forward bad dns info to the other routers too so it needs to be fixed quickly! `killall dnsmasq; dnsmasq start` will fix it.

gwck is a qMp utility that is restarted after network restart.

Another problem I've had occasionally is that the wifi will lose connections. Even though the radio is on and the router lights are normal you can't connect. I've written a simple script to restart wifi if both the ad-hoc and access point interfaces have no connections. It is a bit of a hack since the interface may be ok, but since nothing is connected via wifi it doesn't hurt too much to restart it. I've also found that a network restart is necessary to make the wifi stable.

By default wlan0 is the ad-hoc interface that is used to mesh the routers and wlan0ap is the access point. This script checks to see the number of wireless interfaces so it works with dual-band routers and routers that are only ad-hoc or ap.

I'm using "Signal: unknown" to show there is no connection. It seems to work reliably. You could also try iwinfo wlan0 assoclist.

"sleep 5" is usual between "wifi down" and "wifi up". I've found it not necessary when there are no connections, but I'll leave it there in case.

You can download the watchdog here

in the terminal-

```
vi /root/mesh-watchdog.sh
```

and paste this:

```
#!/bin/sh
# mesh-watchdog v1.1.1, NYC Mesh, Brian Hall

restartWifi()
{
  wifi down
  sleep 5
  wifi up
}

restartNetwork()
{
  /etc/init.d/network restart
  if /etc/init.d/gwck enabled; then
    /etc/init.d/gwck restart
  fi
  /etc/init.d/bmx6 restart
  sleep 4
  killall dnsmasq
  /etc/init.d/dnsmasq start
}
```

```
#gets date-time from log and exit if recently run. date-time is first two words of last line
exitIfRecentRestart()
{
if [ -e $LOG ]; then
  set -- `tail -1 $LOG`
  LASTRUN=`date --date="$1 $2" +%s`
  if [ "$?" = "0" ]; then
    #don't run for 1200s (20 minutes)
    NEXTRUN=$(($LASTRUN + 1200))
    NOW=`date +%s`
    es=$(($NOW - $LASTRUN))
    printf "time since last restartNetwork: "
    printf '%dd %dh:%dm:%ds\n' $(($es/86400)) $(($es%86400/3600)) $(($es%3600/60)) $(($es%60))
    if [ $NOW -lt $NEXTRUN ]; then
      echo "waiting $(($NEXTRUN - $NOW)) seconds, use option -f to force"
      exit 1
    else
      echo "run tests-"
    fi
  else
    echo "invalid date from log, run tests-"
  fi
else
 echo "no log, run tests-"
fi
}

LOG="/tmp/log/mesh-watchdog.log"
FORCE=0

if [ "$1" = "-n" ]; then
  echo "restartNetwork"
  restartNetwork
  exit 1
elif [ "$1" = "-f" ]; then
  echo "force tests-"
  FORCE=1
elif [ "$1" = "-w" ]; then
  echo "restartWifi"
  restartWifi
  exit 1
elif [ "$1" = "-b" ]; then
  echo "restart wifi, wait, restart network"
  restartWifi; wait 60; restartNetwork
  exit 1
elif [ "$1" != "" ]; then
  echo -e "Usage: `basename $0` [OPTION]\n\nTests wifi and internet connections and restarts if necessary (default)\n\n\t-f\tforce test\n\t-n\trestart network\n\t-w\trestart wifi\n\t-b\trestart both wifi and network\n\t-h\toptions\n"
  exit 1
fi

if [ $FORCE != 1 ]; then
  exitIfRecentRestart
fi

DATE=`date +%Y-%m-%d\ %H:%M:%S`
IWINFO=`iwinfo`

# find lines containing "ESSID"|get name (previous word)|replace return with ","
WI=`echo "$IWINFO" | grep ESSID | grep -Eo '^[^ ]+' | sed ':a;N;$!ba;s/\n/, /g`
# count the number of wlan interfaces, and number of wlans with 'no signal'
WLAN=`echo "$WI" | wc -w`
NOSIGNAL=`echo "$IWINFO" | grep 'Signal: unknown' | wc -l`

if [ $WLAN -eq 0 ]; then
  echo "no wlan interfaces, wifi is probably disabled"
elif [ $WLAN -eq $NOSIGNAL ]; then
  # all wlan interfaces are down, so restart wifi
  echo "$DATE restart wifi- wlans:$WLAN no-signal:$NOSIGNAL interfaces:$WI" | tee -a $LOG
  restartWifi
  sleep 60
  restartNetwork
  exit 1
else
  echo "wifi:ok wlans:$WLAN no-signal:$NOSIGNAL interfaces:$WI"
fi

# restart network if ping google.com && 8.8.8.8 fails 4 times
```

```
count=1
while [ "$count" -le 4 ]
 do
   if /bin/ping -c 1 google.com >/dev/null && /bin/ping -c 1 8.8.8.8 >/dev/null; then
      echo "wan:ok  ping-count:$count"
      exit 0
   fi
 let count++
done
echo "$DATE network restart" | tee -a $LOG
restartNetwork
```

Make it executable-

```
chmod +x /root/mesh-watchdog.sh
```

Afterwards, add the following entry with crontab -e

```
* * * * * /root/mesh-watchdog.sh
```

It can run once a minute as it detects whether a network restart has just occurred and will wait 20 minutes before restarting again. I added the 20 minute delay so the router is still functional without an internet gateway.

Thanks to Nitin for help with the wifi problem and Zach for help with dnsmasq.

Email me if you have any questions or suggestions.

Revision #4
Created 9 December 2023 04:40:04 by Willard Nilges
Updated 8 September 2024 00:16:01 by Lydon Thorpe