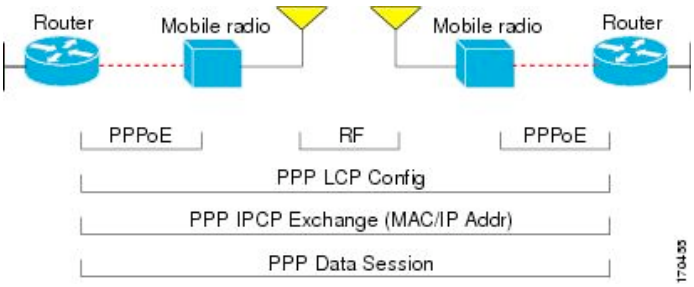
Description

This project provides an open source (GPLv2) pppoe client implementation of RFC4938, “PPP Over Ethernet (PPPoE) Extensions for Credit Flow and Link Metrics” and the IETF draft, draft-bberry-pppoe-scaled-credits-metrics-01.txt, “PPP Over Ethernet (PPPoE) Extensions for Scaled Credits and Link Metrics” based on Roaring Penguin's RP-PPPOE package.

This project provides a way to simulate the radio topologies described in RFC4938. These topologies, illustrated below, represent intra-nodal communications between a router and its partner radio using PPPoE to provide feedback to layer 3 routing protocols on the layer 2 characteristics of the radio link. Each radio initiates the PPPoE session as soon as the radio establishes a radio link to another radio. After the PPPoE sessions are active, a PPP session is established end-to-end (router-to-router). When a radio loses the radio link connection to a peer, the corresponding PPPoE connection is terminated.

This program simulates these radio connections through user input to establish, terminate, and inject metrics to the router partner.



Hardware and Software Needed

- Two or more routers which support RFC4938 will be required. On cisco.com, through feature navigator [http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp] search for the feature, “MANET: Enhancements to PPPoE for Router-to-Radio Links”. This will provide images which support rfc4938. To support credit and metric scaling, images are available under Early Field Trial license only. Please contact Rex Craig (recraig@cisco.com) for more information.
- Two or more linux platforms to run this project. Install instructions are provided for the Linksys wrtsl54gs [http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp] or wrt54gl [http://www.linksys.com/servlet/Satellite?c=L_Product_C2&childpagename=US%2FLayout&cid=1133202177241&pagename=Linksys%2FCommon%2FVisitorWrapper&lid=7724139789B06] running the WhiteRussian release of openwrt [http://www.linksys.com/servlet/Satellite?c=L_Product_C2&childpagename=US%2FLayout&cid=1133202177241&pagename=Linksys%2FCommon%2FVisitorWrapper&lid=7724139789B06].

References

- RFC4938 [http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp]
- draft-bberry-pppoe-scaled-credits-metrics-01.txt [http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp]
- RP-PPPoE [http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp]
- Mobile Ad Hoc Networks for Router-to-Radio Communications configuration guide on cisco.com [http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp]

Install instructions for the Linksys wrtsl54s running WhiteRussian openwrt

WRTSL54GS: Upgrade linksys image, then load openwrt image

Files Needed:

- latest linksys image [http://tinyurl.com/yq4atk] (WRTSL54GS_2.06.0_US_code.bin)
- openwrt image for wrtsl54gs [http://tinyurl.com/yq4atk](openwrt-wrtsl54gs-squashfs.bin)

The image that ships with the wrtsl54gs will not let you upgrade the firmware with an openwrt image. If you use the latest firmware, however, it does. After upgrading to the latest linksys image, you can upload the openwrt image. Connect to the LAN port on the SL and receive a dhcp address. Do not use the wireless interface. To do this go to the default webpage of 192.168.1.1 with a username blank and password 'admin'. Then go to the administration tab and “Firmware upgrade” tab.

WRT54GL: Load openwrt image directly

- openwrt image for wrt54gl [http://tinyurl.com/yq4atk] (openwrt-wrt54g-squashfs.bin)

Connect to the LAN port on the wrt54gl and receive a dhcp address. Do not use the wireless interface. To do this go to the default webpage of 192.168.1.1 with a username blank and password 'admin'. Then go to the administration tab and “Firmware upgrade” tab.

change root password

After the openwrt image is loaded, the machine will reboot into openwrt. Give it some time the first time it boots to set things up. When it comes up it will have the last lan ip settings that the linksys box had configured. The default is to give ip's out on the 192.168.1.1 network. Ping 192.168.1.1 to see when the openwrt box is up. Sometimes you may need to pull the power after a few minutes to force the reboot.

```
edward-paradises-computer:~ pdice$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=1.367 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.150 ms
^C
--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.150/1.258/1.367/0.108 ms
```

The first time you login, it must be over telnet. You can then set a password and, from then on, login using ssh and the 'root' username. There are instructions below for turning telnet back on if you choose.

```
edward-paradises-computer:~ pdice$ telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
 === IMPORTANT ========================
  Use 'passwd' to set your login password
  this will disable telnet and enable SSH
 ------------------------------------------


BusyBox v1.00 (2007.01.30-11:42+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 WHITE RUSSIAN (0.9) ------------------------------
  * 2 oz Vodka   Mix the Vodka and Kahlua together
  * 1 oz Kahlua  over ice, then float the cream or
  * 1/2oz cream  milk on the top.
 ------------------------------------------------
root@OpenWrt:/# passwd
Changing password for root
Enter the new password (minimum of 5 characters)
Please use a combination of upper and lower case letters and numbers.
Enter new password:
Bad password: too short.

Warning: weak password (continuing).
Re-enter new password:
Password changed.
root@OpenWrt:/# exit
Connection closed by foreign host.
```

How To Get Files and Install Packages and Scripts

During this configuration, you will need to download/copy certain files onto the openwrt device. This can be done in two ways:

1) You can use the ‘scp’ command to copy files from a PC to the WRTSL54GS

```
Example:
root@caladan-sl1:/# scp root@192.168.1.105:~/nvram-clean.sh .

Host '192.168.1.105' is not in the trusted hosts file.
(fingerprint md5 90:15:b5:73:9f:23:b1:70:3d:5a:ac:eb:6e:eb:5e:9c)
```

```
Do you want to continue connecting? (y/n) y
Password:
nvram-clean.sh                                100% 4702     4.6KB/s   00:00
root@caladan-sl1:/#
```

2) You can also use the 'scp' command in the opposite direction to copy a file from a PC to the WRTSL54GS using a client like WinSCP [http://winscp.net/eng/index.php]. More clients are listed on Wikipedia here [http://winscp.net/eng/index.php]

3) If you have access to an http server with the files on it, then you can use 'wget' command to download the files from the server.

```
Example:
root@caladan-sl1:/# wget http://192.168.1.105/nvram-clean.sh
Connecting to 192.168.1.105[192.168.1.105]:80
nvram-clean.sh       100% |****************************|  4702       00:00 ETA
root@caladan-sl1:/#
```

*NOTE: '192.168.1.105' in these examples corresponds to the IP address of the computer from which the files are being copied. Replace this IP address in accordance with your setup.

Clean up the configs

Files Needed:

- nvram-clean.sh [http://downloads.openwrt.org/people/kaloz/nvram-clean.sh]

———

When openwrt first boots up, it has a lot of variables from previous versions in nvram still present. We don't need them and it makes debugging more difficult if they are present. Use the nvram-clean.sh script to clean up the variables. It doesn't take effect until you save it and reboot.

run the clean script

```
root@OpenWrt:~# wget http://192.168.1.90/nvram-clean.sh
Connecting to 192.168.1.90[192.168.1.90]:80
nvram-clean.sh       100% |****************************************************************|  4702       00:00 ETA
root@OpenWrt:~# chmod +x nvram-clean.sh
root@OpenWrt:~# ./nvram-clean.sh
Before: size: 11197 bytes (21571 left)
After: size: 3610 bytes (29158 left)
root@OpenWrt:~# nvram commit
root@OpenWrt:~# reboot
```

Remove Packages

Remove the following packages, ppp and ppp-mod-pppoe

ppp

```
root@caladan-sl1:/etc/init.d# ipkg remove ppp
Removing package ppp from root...
Successfully terminated.
```

ppp-mod-pppoe

```
root@caladan-sl1:/# ipkg remove ppp-mod-pppoe
Removing package ppp-mod-pppoe from root...
Successfully terminated.
root@caladan-sl1:/#
```

Install Required Packages

Install the following packags. This package can either be installed from a webserver or locally.

- rfc4938_1.00-1_mipsel.ipk [http://sourceforge.net/project/showfiles.php?group_id=218542]

rfc4938

```
root@caladan-sl1:~# ipkg install http://192.168.1.188/openwrt/rfc4938_1.00-1_mipsel.ipk
Downloading http://192.168.1.188/openwrt/rfc4938_1.00-1_mipsel.ipk
Installing rfc4938 (1.00-1) to root...
Configuring rfc4938
Successfully terminated.
root@caladan-sl1:~#
```

Configure Wan Interface

The wan interface is used to connect to the router.

Set the hostname of the box.

```
nvram set wan_hostname=caladan-sl1
```

Configure the ip address to be a dummy address, this will not be used but it must be set.

```
nvram set wan_proto=static
nvram set wan_ipaddr=9.9.9.9
nvram set wan_netmask=255.255.255.0
```

lan config

Lan interface is used for neighbor communication. Set the IP to correspond the subnet you are going to use to communicate with the other rfc4938 neighbors and your controlling pc.

```
nvram set lan_ipaddr=192.168.1.101
```

Setup scripts in init.d

Files needed to modify:

- S35firewall
- S60dnsmasq

Files needed to optionally modify:

- S50telnet

———

We need to modify 2 scripts in /etc/init.d/ and optionally a third if we want to enable telnet

```
root@caladan-sl1:/# cd /etc/init.d/
```

S35firewall

First, disable the firewall.

```
root@caladan-sl1:/etc/init.d# chmod -x S35firewall
```

S60dnsmasq

Also remove the dhcp server and dns cache.

```
root@caladan-sl1:/etc/init.d# chmod -x S60dnsmasq
```

OPTIONAL - S50telnet

In /etc/init.d, remove the `-l /bin/login` from the S50telnet file to enable login via telnet.

```
root@caladan-sl1:/etc/init.d# cat S50telnet
#!/bin/sh
telnetd -l /bin/login
```

save your configs and reboot

```
nvram commit
reboot
```

Configure more boxes

Follow the same procedure above for the remaining openwrt devices in your setup, but change the lan_ipaddr to a different address on the same subnet.

Install instructions for other platforms

If you want to install these programs on another linux platform, you will need to build from source. First run `./configure`, then `make`, then `make install`. `Make install` will not install the programs in the correct locations, but rather it will place it in the `bin/` directory. From there you need to copy the `bin/pppoe` program into `/usr/sbin/`, the `bin/rfc4938` program into `/usr/sbin/` also, and the `bin/rfc4938ctl` program into `/usr/bin`. You will also need to copy `conf/rfc4938.conf` to `/etc`. Optionally, you can copy the init script, `src/S98rfc4938` into your init.d directory, most likely located in `/etc/rc.d/init.d` if you would like rfc4938 to start at bootup. Otherwise, you can just run it directly with root privileges.

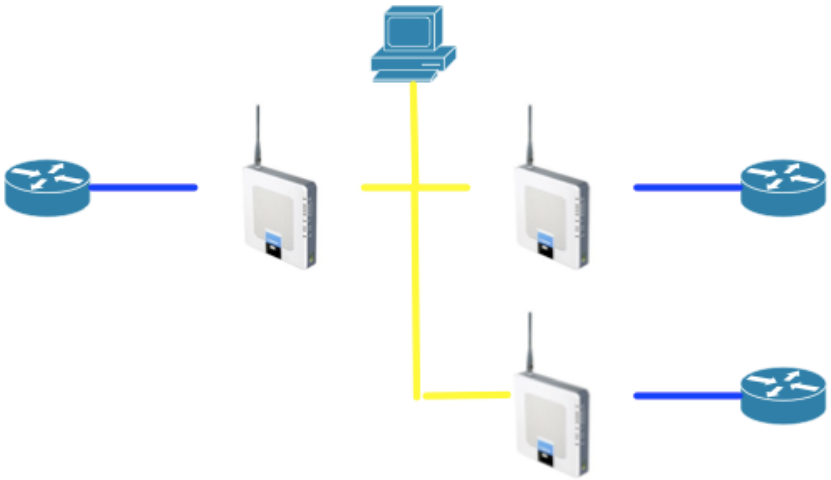
Using rfc4938 and rfc4938ctl

Setting up test network

Each device running rfc4938 will need two ethernet connections. One connection will be to the router and one connection will be to its neighbors also running rfc4938. The connection to the router will be needed in the rfc4938.conf file. It is strongly recommended that all neighbor connections will need to be on the same subnet, along with the pc you will use to control them.

wrtsl54gs and wrt54gl instructions

Plug each cisco router into the internet port of their partner linksys box using a crossover cable. Plug the neighbors together using the numbered lan ports to a switch with a crossover cable also. If you have a small number of devices, you can connect them together using crossover cables and their own lan ports. In addition, plug your pc into the switch or one of the lan ports of either linksys box and assign a static ip to your machine on the same subnet that you assigned for the lan ports on the other boxes.



rfc4938.conf Configuration File

A default configuration file is including in the rfc4938 ipkg. It is placed in /etc with the filename rfc4938.conf.

To simplify configuration and avoid errors, it is recommended that same configuration file be used on all rfc4938 boxes. When the configuration file is processed, the neighbor definition which matches the local machine will be ignored. For example, if your lan_ipaddr is 192.168.1.101, for the configuration below neighbor 1 would not be added. Modify this configuration file to match the ip addresses you assigned to each neighbor for each NEIGHBOR definition.

WRT54GL: Change default IFACE

On the wrt54gl, you must change the IFACE parameter to vlan1

Default Config File

```
# Interface to use to connect to the router
IFACE eth1

# Maximum number of neighbors that can be connected to.
# NOTE: this configuration directive must come before
# any neighbor definitions
MAX_NEIGHBORS 100

# Port the rfc4938ctl process will listen to.  This must
# be the same for all neighbors.
CTL_PORT 5010

# Port the rfc4938 process will listen to. This must
# also be the same for all neighbors.
PORT 5011

# Neighbor definitions
NEIGHBOR 1 192.168.1.101
NEIGHBOR 2 192.168.1.102
NEIGHBOR 3 192.168.1.103

# Service name to use in PPPoE PADI.  A prefix of manet_radio
# must be used for the router to identify it as a rfc4938
# compliant session
SERVICE_NAME manet_radio

# Debug level
#
# 0 - no output
# 1 - output from errors
# 2 - output from events
# 3 - output from packet events
DEBUG_LEVEL 2
```

Program Descriptions

The ipkg installs 3 programs, `rfc4938`, `rfc4938ctl`, and `pppoe`. The `rfc4938` program is the daemon which accepts connections from other neighbors. The `rfc4938ctl` program accepts user input and signals the `rfc4938` to create neighbors, terminate neighbors, inject metrics, and change the grant amount. The final program installed is the `pppoe` program which establishes the PPPoE connection to the router.

rfc4938ctl

The `rfc4938ctl` program has the following options:

```
  usage: rfc4938ctl [options]

     show
     padq neighbor <neighbor #> max-data-rate <rate> <scalar> cur-data-rate
           <rate> <scalar> latency <milliseconds> resources <percentage>
                rel-link-qual <percentage> [receive-only]
     padg neighbor <neighbor #> <credits>
     initiate { neighbor <neighbor #> | all } <scalar>
     terminate { neighbor <neighbor #> | all }
```

Show Status

The most important command in the `rfc4938ctl` program is the `show` command. With this command you can see the status of neighbors. There is no feedback reported from the rfc4938ctl command if there is a problem in starting a session, so you must use this `show` command to check status. You will see the list of all available neighbors, whether or not they are active.

```
root@caladan-sl1:~# rfc4938ctl show
show
Neighbor        IP              Active
3               192.168.1.103   INACTIVE
2               192.168.1.102   ACTIVE
```

Initiate Sessions

Sessions are initiated using the `initiate` option of the `rfc4938ctl` program. Sessions can be initiated for a single neighbor, or all neighbors from one of the neighbor's `rfc4938ctl` program. The initial credit scalar can also be specified. This scalar will be the same on both sides. A scalar of 0 specifies that no credit scalar should be sent, and the default of 64bytes will be used. This is equivalent to an implementation that has only implemented rfc4938 and not the credit and metric scaling draft. This will be refererred to as running an rfc4938-only session throughout the rest of this document.

Initiating for One Neighbor

Here a session is initiated for neighbor 2 without a credit scalar from caladan-sl1. The session is established on caladan-sl2 automatically.

caladan-sl1

```
root@caladan-sl1:~# rfc4938ctl show
show
Neighbor        IP              Active
3               192.168.1.103   INACTIVE
2               192.168.1.102   INACTIVE


root@caladan-sl1:~# rfc4938ctl initiate neighbor 2 0
initiate neighbor 2 scalar 0
root@caladan-sl1:~# rfc4938ctl show
show
Neighbor        IP              Active
3               192.168.1.103   INACTIVE
2               192.168.1.102   ACTIVE
```

caladan-sl2

```
root@caladan-sl2:~# rfc4938ctl show
show
Neighbor        IP              Active
3               192.168.1.103   INACTIVE
2               192.168.1.102   INACTIVE
```

The initiate command is now entered on caladan-sl1

```
root@caladan-sl2:~# rfc4938ctl show
show
Neighbor        IP              Active
3               192.168.1.103   INACTIVE
1               192.168.1.101   ACTIVE
```

Initiating for all neighbors

Here sessions are intitiated for all neighbors with the credit scalar of 100 bytes.

```
root@caladan-sl1:~# rfc4938ctl show
show
Neighbor        IP              Active
3               192.168.1.103   INACTIVE
```

```
2               192.168.1.102    INACTIVE

root@caladan-sl1:~# rfc4938ctl initiate all 100
initiate all scalar 100
root@caladan-sl1:~# rfc4938ctl show
show
Neighbor        IP               Active
3               192.168.1.103    ACTIVE
2               192.168.1.102    ACTIVE
```

Terminate Sessions

Sessions can be terminate using `rfc4938ctl` or from the router with a `clear pppoe all`. Sessions can be terminated from the `rfc4938ctl` program individually or all at once.

Terminate One Session

```
show
Neighbor        IP               Active
3               192.168.1.103    ACTIVE
2               192.168.1.102    ACTIVE

root@caladan-sl1:~# rfc4938ctl terminate neighbor 3
termiante neighbor 3
root@caladan-sl1:~# rfc4938ctl show
show
Neighbor        IP               Active
3               192.168.1.103    INACTIVE
2               192.168.1.102    ACTIVE
```

Terminate All Sessions

```
root@caladan-sl1:~# rfc4938ctl show
show
Neighbor        IP               Active
3               192.168.1.103    ACTIVE
2               192.168.1.102    ACTIVE


root@caladan-sl1:~# rfc4938ctl terminate all
terminate all
root@caladan-sl1:~# rfc4938ctl show
show
Neighbor        IP               Active
3               192.168.1.103    INACTIVE
2               192.168.1.102    INACTIVE
```

Injecting Metrics

Metrics can be injected using `rfc4938ctl` using the padq command. All parameters of this command must be entered with the exception of the receive only. Metrics are injected on a per neighbor basis and only for one side. For example, if I have two nodes I would need to enter the padq command twice, once on node 1 and once on node 2 for each router to receive the PADQ generated by the pppoe process. Also, scaling of the CDR and MDR values will only occur when a scaling session has been initiated. rfc4938-only sessions will ignore that input and use the kbps scalar specified in rfc4938.

The command:

```
padq neighbor <neighbor #> max-data-rate <rate> <scalar> cur-data-rate
           <rate> <scalar> latency <milliseconds> resources <percentage>
                rel-link-qual <percentage> [receive-only]
```

Parameter	Full Name	Minimum Value	Maximum Value
max-data-rate	Maximum Data Rate	0	65535
cur-data-rate	Current Data Rate	0	65535
latency	Latency	0	65535
resources	Resources	0	100

rel-link-qual	Relative Link Quality	0	100

The scalar is:

```
0 - kbps
1 - mbps
2 - gbps
3 - tbps
```

NOTE: CDR and MDR scalars are independent of each other.

PADQ injection example

```
root@caladan-sl1:~# rfc4938ctl padq neighbor 3 max-data-rate 2000 2 cur-data-rate 1000 1 latency 1 resources 100 rel-link-qual 90
```

Change Grant Amount

You can also change the 1-second grant interval for a neighbor. This grant amount is the maximum number of credits that will be granted in a 1 second period. With this setting, you are able to perform flow control at the rate you wish. For example, a setting of 1000 credits would result in a throughtput rate of 512kbps with the default scalar of 64 bytes (1000credits * 64bytes/credit * 8 bits/byte = 512kbps). The default setting is 1953 credits, which is 1Mbps for a 64byte scalar. This command, like the PADQ command, is on a per neighbor basis and only for one side. To get the same throughput from router to router, the credit value must be set the same on both pppoe clients through rfc4938ctl on each device running rfc4938. In this implementation, the credit granting was simplified to this 1-second window. In a more robust implementation, credits would be granted based on available buffer space in the radio.

```
root@caladan-sl1:~# rfc4938ctl padg neighbor 3 1000
padg neighbor 3 credits 1000
root@caladan-sl1:~#
```

How to use this tool to showcase the benefits of a system that incorporates rfc4938

RFC4938 is a simple messaging protocol between a radio and a router. The benefits are best observed as a solution that incorporates layer 3 routing protocols. Layer 2 feedback to the layer 3 routing protocols provides faster convergence, better route selection, and advanced queuing.

Per Session QoS

The radio creates an individual PPPoE session to the router for each radio peer it sees. Each of these sessions to the peers will appear as a different virtual access interfaces. The benefit of this, is that there will be a different queue per neighbor that QoS can be applied to. The radio will flow control these connections with credits differently, so a slow link to one neighbor will not effect the other. This could happen due to a variety of reasons, but one obvious situation is one where you have clear line-of-sight to one neighbor, but obstructed line-of-sight to another. The radio needs to use a lower datarate to send to the obstructed neighbor.

To demonstrate per session QoS using this tool, initiate two sessions from one node. On one session, leave the default grant window of 1953 credits. On the other session, cut the grant rate with the command, `rfc4938ctl padg neighbor 2 976`. This will cut the throughput of one neighbor in half while the other remains the same. If you are transmitting more that 500kbps of data, you can see the packets now queueing on the interface which you cut in half by using the `show queue virtual-access #` command in IOS and substituting the correct virtual-access number in.

Rapid Convergence based on Events and not Timers

The creation and termination of PPPoE sessions allow the layer 3 routing protocols to rapidly converge. Normally protocols like OSPF and EIGRP rely on hello and dead/hold timers to create and terminate layer 3 routing neighbors. They will periodically sent out multicast hellos to advertise themselves to other neighbors. After a relationship is formed, they then listen for these hellos from other neighbors, and if enough are missed, they declare that neighbor “dead”. In the case of OSPF, the default timers are 10 seconds for hello and 40 seconds for dead. With the addition of RFC4938, these timers are no longer the only way to establish and terminate neighbors in IOS. When a PPPoE session is created for a neighbor a hello is immediately sent. When a PPPoE session is terminated, the neighbor is declared dead. In the time that you would normally wait declare a neighbor dead, any traffic that was routed through that node would be lost. With the event, packet loss due to black hole routing is greatly minimized. This feature is currently supported in IOS with OSPFv3 and EIGRP.

Demonstrating this is fairly easy with this tool. Initiate two sessions with the `rfc4938ctl intitiate neighbor all 0` command. Then terminate one of those neighbors created with the `rfc4938ctl terminate neighbor 2` command. You'll see the layer 3 routing neighbor come up fully almost immediately, and then down immediately when the terminate command is issued.

Route Cost based on Radio Link Characteristics

The PADQ packet allows the radio to transmit the layer 2 radio link characteristics to the router through a set of six metrics, Maximum Data Rate, Current Data Rate, Latency, Resources, and Relative Link Quality. The radio is connected by an ethernet cable and OSPF and EIGRP would normally see the route cost for this link to be constant. However, this ethernet connection only connects it to the radio, the radio links charaterstics can be entirely different. On a per session basis, the radio can provide feedback in the PADQ packets to calculate different routing costs on a per neighbor basis. This feature is currently supported in IOS with OSPFv3 and EIGRP.

Consider the following trivial example that can be replicated with this tool. You have 3 nodes, node A, node B, and node C. Node A has 55Mbps links to Node B and Node C. Node B has a 55Mbps link to Node A, but only a 50kbps link to node C do to an obstructed view. With feedback from the radio, the radio can influence the route cost such that packets destined to node C from node B will take the 2-hop path of the two 100Mbps links instead of the single 55kbps link.

To accomplish this, you need to use the `rfc4938ctl padq` and inject padqs for each link at each node to simulate this scenario.

Sample Router Config

The following config shows an example on a 3845 that would support RFC4938 compliant sessions. You would plug your device into Gigabit 0/0. This same config could be used for your other routers, but you would need to change the ip address assigned to VMI1 to another address in the same subnet.

The IOS configuration guide for this feature is located here
[http://www.cisco.com/en/US/docs/ios/12_4t/ip_mobility/configuration/guide/ip_manet.html]

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 3845-example
!
boot-start-marker
boot-end-marker
!
logging buffered 3000000
enable password lab
!
no aaa new-model
no network-clock-participate slot 1
no network-clock-participate slot 2
no network-clock-participate slot 4
!
!
ip cef
!
!
no ip domain lookup
ip domain name yourdomain.com
subscriber authorization enable
!
subscriber profile 3845-example
 pppoe service manet_radio
!
!
multilink bundle-name authenticated
!
no virtual-template subinterface
!
!
!
archive
 log config
 hidekeys
!
!
!
policy-map FQ
 class class-default
  fair-queue
!
!
!
!
bba-group pppoe VMI1
```

```
 virtual-template 1
 service profile 3845-example
!
!
interface Loopback1
 ip address 1.1.1.1 255.255.255.255
 load-interval 30
!
interface GigabitEthernet0/0
 no ip address
 load-interval 30
 duplex full
 speed 100
 media-type rj45
 pppoe enable group VMI1
!
interface GigabitEthernet0/1
 no ip address
 load-interval 30
 duplex full
 speed 100
 media-type rj45
 no keepalive
!
interface FastEthernet0/0/0
!
interface FastEthernet0/0/1
!
interface FastEthernet0/0/2
!
interface FastEthernet0/0/3
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Virtual-Template1
 ip unnumbered vmi1
 load-interval 30
 no keepalive
 service-policy output FQ
!
interface Vlan1
 no ip address
!
interface vmi1
 ip address 4.3.3.1 255.255.255.0
 no ip redirects
 no ip split-horizon eigrp 1
 load-interval 30
 physical-interface GigabitEthernet0/0
!
router eigrp 1
 redistribute connected
 network 4.3.0.0 0.0.255.255
 auto-summary
!
!
!
ip http server
ip http authentication local
no ip http secure-server
ip http timeout-policy idle 5 life 86400 requests 10000
!
!
!
!
!
!
!
control-plane
!
!
!
!
!
```

```
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 privilege level 15
 no login
 transport input all
!
scheduler allocate 20000 1000
!
webvpn cef
!
end
```

Common Problems

Changes in grant amount have no effect

Problem

You've change the grant amount, but nothing different happens with your 2-way traffic.

Solution

Since there are two pppoe sessions, you need to change the grant amount on each connection from the SL. The lowest grant amount between the two will dictate what you throughput rate will be.

Scalars in PADQ aren't being sent

Problem

You've tried to send a scaled PADQ, but the scalar doesn't get transmitted.

Solution

You've most likely initiated a rfc4938-only session (scalar=0). The scalar values in the PADQ will not be sent if the session is rfc4938-only. Reestablish a scaled session and then send your PADQ.

Scalars in PADQ aren't being sent

Problem

The rfc4938ctl program is hung.

Solution

Most likely it is waiting for a response from rfc4938, but it hasn't received one. Make sure the rfc4938 daemon is started, and if not, ctrl-c the rfc4938ctl program that is hung, start the rfc4938 daemon, and re-enter your rfc4938ctl command.

Known Limitations

Dropped large packets

It has been observed when running on the wrtsl54gs hardware, that large packets will occasionly be dropped. In one overnight 1-hop test at 1Mbps, 1,492 out of 6,469,897 icmp packets were dropped, which is about .02% packet loss. As you increase the number of hops or the traffic rate, the packet loss will increase. However, that same test on a modest linux pc had 0% loss.

Troubleshooting

Verify Connectivity

First make sure that you can ping from linksys device to linksys device using their lan ip's.

Sniff linksys box to Router connection

Using a sniffer, make sure you see a successful PPPoE connection start between the linksys box and Router.

logread

use logread to view syslog in openwrt and cat /var/log/messages on others. prefixes will be “rfc4938:” and “pppoe:”. On other systems, cat /var/log/messages

Run rfc4938 directly to see output

Instead of having `rfc4938` run in the background, run it directly. First execut `killall rfc4938` kill any currently running rfc4938 process, then execute `/usr/sbin/rfc4938` with root priviledges. Then in another window execute `rfc4938ctl` like you would normally do from another window. Observe the messages from your rfc4938 window.

Increase debug level to 3

Increase the debug level output to 3 and observe the messages.

Support

Support will be provided on a best-effort basis through the sourceforge.net website. If you have compiled your own binaries for a platform other than the wrtsl54gs or wrt54gl running the WhiteRussian release of openwrt, please reproduce any bugs on that platform first before submitting.

- rfc4938 sourceforge support requests [https://sourceforge.net/tracker/?group_id=218542&atid=1043318]
- rfc4938 sourceforge help forum [https://sourceforge.net/tracker/?group_id=218542&atid=1043318]

License and Warranty

GPL License

```
PPPoE Version 1.0, Copyright (C) 2001-2006 Roaring Penguin Software Inc.
                   Copyright (C) 2007-2008 Cisco Systems, Inc.
PPPoE comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it under the terms
of the GNU General Public License, version 2 or any later version.
http://www.roaringpenguin.com
```

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.)  You can apply it to
your programs, too.
```

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such

```
    interactive use in the most ordinary way, to print or display an
    announcement including an appropriate copyright notice and a
    notice that there is no warranty (or else, saying that you provide
    a warranty) and that users may redistribute the program under
    these conditions, and telling the user how to view a copy of this
    License.  (Exception: if the Program itself is interactive but
    does not normally print such an announcement, your work based on
    the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole.  If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works.  But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

  3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

    a) Accompany it with the complete corresponding machine-readable
    source code, which must be distributed under the terms of Sections
    1 and 2 above on a medium customarily used for software interchange; or,

    b) Accompany it with a written offer, valid for at least three
    years, to give any third party, for a charge no more than your
    cost of physically performing source distribution, a complete
    machine-readable copy of the corresponding source code, to be
    distributed under the terms of Sections 1 and 2 above on a medium
    customarily used for software interchange; or,

    c) Accompany it with the information you received as to the offer
    to distribute corresponding source code.  (This alternative is
    allowed only for noncommercial distribution and only if you
    received the program in object code or executable form with such
    an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it.  For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable.  However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not
compelled to copy the source along with the object code.

  4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License.  Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

  5. You are not required to accept this License, since you have not
signed it.  However, nothing else grants you permission to modify or
distribute the Program or its derivative works.  These actions are
prohibited by law if you do not accept this License.  Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Program or works based on it.
```

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,

```
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

                     END OF TERMS AND CONDITIONS
```

Voiding Linksys Warranty

By loading openwrt on your Linksys wrtsl54gs or wrt54gl, you void your warranty provided by Linksys. No support will be provided by Linksys or Cisco for openwrt.