Introduction

(copied from forum-posting by user aldoir Dual-WAN forum posting

You'll need to create a new VLAN, called vlan2.

Enter in the web interface, Settings/VLANs. Set Port 4 to a new VLAN.

Enter in the console, and do the following commands

commands to set VLANs

```
nvram set vlan0ports="1 2 3 5*"
nvram set vlan2ports="4 5"
nvram set vlan2hwname=et0
nvram commit
```

Here are the scripts I used to startup the interface. You need to call them in rc_firewall

wan2.firewall

//edit - dont like this bit, it doesnt seem to be for DHCP connection and looks like its for a static IP connection

```
#!/bin/sh
WAN2_IFNAME=vlan2
WAN2_IPADDR=10.0.0.2
WAN2_GATEWAY=10.0.0.1
WAN2_NETMASK=255.0.0.0
if [ "$(nvram get wan2_ipaddr)" != "$WAN2_IPADDR" ]; then
        nvram set wan2_ifname=$WAN2_IFNAME
   nvram set wan2_ipaddr=$WAN2_IPADDR
   nvram set wan2_gateway=$WAN2_GATEWAY
   nvram set wan2_netmask=$WAN2_NETMASK
   nvram commit
fi
ifconfig $(nvram get wan2_ifname) up $(nvram get wan2_ipaddr) netmask $(nvram get wan2_netmask)
```

routes.firewall

```
#!/bin/sh
ip rule flush
ip rule add lookup main prio 32766
ip rule add lookup default prio 32767
ip rule add from $(nvram get wan_ipaddr) table 100 prio 100
ip rule add fwmark 0x100 table 100 prio 101
ip rule add from $(nvram get wan2_ipaddr) table 200 prio 200
ip rule add fwmark 0x200 table 200 prio 201
ip route flush table 100
ip route flush table 200
for TABLE in 100 200
do
   ip route | grep link | while read ROUTE
   do
      ip route add table $TABLE to $ROUTE
   done
done
ip route add table 100 default via $(nvram get wan_gateway)
```

```
ip route add table 200 default via $(nvram get wan2_gateway)
```

firewall.firewall

```
#!/bin/sh
#DD-WRT firewall rules
#BEGIN
#apply simple forward rules
for RULE in $(nvram get forward_spec)
do
   FROM=`echo $RULE | cut -d '>' -f 1`
   TO=`echo $RULE | cut -d '>' -f 2`
   STATE=`echo $FROM | cut -d ':' -f 2`
   PROTO=`echo $FROM | cut -d ':' -f 3`
   SPORT=`echo $FROM | cut -d ':' -f 4`
   DEST=`echo $TO | cut -d ':' -f 1`
   DPORT=`echo $TO | cut -d ':' -f 2`
   if [ "$STATE" = "on" ]; then
      if [ "$PROTO" = "both" ]; then
         #udp
         #iptables -A FORWARD -d $(nvram get wan2_ipaddr) -p udp --dport $SPORT -j ACCEPT
         iptables -A PREROUTING -t nat -p udp -d $(nvram get wan2_ipaddr) --dport $SPORT -j DNAT
         #tcp
         #iptables -A FORWARD -d $(nvram get wan2_ipaddr) -p tcp --dport $SPORT -j ACCEPT
         iptables -A PREROUTING -t nat -p tcp -d $(nvram get wan2_ipaddr) --dport $SPORT -j DNAT
      else
         #iptables -A FORWARD -d $(nvram get wan2_ipaddr) -p $PROTO --dport $SPORT -j ACCEPT
         iptables -A PREROUTING -t nat -p $PROTO -d $(nvram get wan2_ipaddr) --dport $SPORT -j DN
      fi
   fi
done
#apply range forward rules
for RULE in $(nvram get forward_port)
do
   FROM=`echo $RULE | cut -d '>' -f 1`
   TO=`echo $RULE | cut -d '>' -f 2`
   STATE=`echo $FROM | cut -d ':' -f 2`
   PROTO=`echo $FROM | cut -d ':' -f 3`
   SPORT=`echo $FROM | cut -d ':' -f 4`
   EPORT=`echo $FROM | cut -d ':' -f 5`

   if [ "$STATE" = "on" ]; then
      if [ "$PROTO" = "both" ]; then
         #udp
         #iptables -A FORWARD -d $(nvram get wan2_ipaddr) -p udp --dport $SPORT:$EPORT -j ACCEPT
         iptables -A PREROUTING -t nat -p udp -d $(nvram get wan2_ipaddr) --dport $SPORT:$EPORT -
         #tcp
         #iptables -A FORWARD -d $(nvram get wan2_ipaddr) -p tcp --dport $SPORT:$EPORT -j ACCEPT
         iptables -A PREROUTING -t nat -p tcp -d $(nvram get wan2_ipaddr) --dport $SPORT:$EPORT -
      else
         #iptables -A FORWARD -d $(nvram get wan2_ipaddr) -p $PROTO --dport $SPORT:$EPORT -j ACCE
         iptables -A PREROUTING -t nat -p $PROTO -d $(nvram get wan2_ipaddr) --dport $SPORT:$EPOR
      fi
   fi
done
iptables -A PREROUTING -t nat -p icmp -d $(nvram get wan2_ipaddr) -j DNAT --to $(nvram get lan_ip
if [ $(nvram get remote_management) -eq 1 ]; then
      iptables -A PREROUTING -t nat -p tcp -d $(nvram get wan2_ipaddr) \
            --dport $(nvram get http_wanport) -j DNAT --to $(nvram get lan_ipaddr):$(nvram get ht
fi
if [ $(nvram get dmz_enable) -eq 1 ]; then
      DMZ_IP=$(nvram get lan_ipaddr | sed -r 's/[0-9]+$//')$(nvram get dmz_ipaddr)
```

```
      iptables -A PREROUTING -t nat -d $(nvram get wan2_ipaddr) -j DNAT --to $DMZ_IP
fi
iptables -A PREROUTING -t nat --dest $(nvram get wan2_ipaddr) -j TRIGGER --trigger-type dnat
iptables -A FORWARD -i $(nvram get wan2_ifname) -o $(nvram get lan_ifname) -j TRIGGER --trigger-t
iptables -A PREROUTING -t mangle -i $(nvram get wan2_ifname) -j IMQ --todev 0
iptables -A PREROUTING -t mangle -i $(nvram get wan2_ifname) -j SVQOS_IN
iptables -A POSTROUTING -t mangle -o $(nvram get wan2_ifname) -j SVQOS_OUT
#DD-WRT END
#Save the gateway in the connection mark for new incoming connections
iptables -t mangle -A PREROUTING -i $(nvram get wan_ifname) -m conntrack --ctstate NEW -j CONNMAR
iptables -t mangle -A PREROUTING -i $(nvram get wan2_ifname) -m conntrack --ctstate NEW -j CONNMA
# Save the gateway in the connection mark for new outgoing connections
iptables -t mangle -A POSTROUTING -o $(nvram get wan_ifname) -m conntrack --ctstate NEW -j CONNMA
iptables -t mangle -A POSTROUTING -o $(nvram get wan2_ifname) -m conntrack --ctstate NEW -j CONNM
# Use the correct gateway for reply packets from the LAN
iptables -t mangle -A PREROUTING -i $(nvram get lan_ifname) -m conntrack --ctstate ESTABLISHED,RE
# Use the correct gateway for reply packets from local connections
iptables -t mangle -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j CONNMARK --restore-mar
#mask known packets to its source address
iptables -A POSTROUTING -t nat -m mark --mark 0x100 -j SNAT --to-source $(nvram get wan_ipaddr)
iptables -A POSTROUTING -t nat -m mark --mark 0x200 -j SNAT --to-source $(nvram get wan2_ipaddr)
#permit access to wan2
iptables -A POSTROUTING -t nat -j MASQUERADE -o $(nvram get wan2_ifname)
#restore-mark is done in PREROUTING. If restored again, will loose the outgoing marks
iptables -t mangle -D SVQOS_OUT -j CONNMARK --restore-mark 2> /dev/null
#iptables -t mangle -A PREROUTING -i $(nvram get lan_ifname) -m multiport -p tcp --dport 22,25,80
#use WAN1 for common navigation, WAN2 for P2P traffic and others
#iptables -t mangle -I PREROUTING -i $(nvram get lan_ifname) -s 192.168.1.100 -j MARK --set-mark
#iptables -t mangle -I PREROUTING -i $(nvram get lan_ifname) -s 192.168.1.100 -j LOG --log-prefix
RP_PATH=/proc/sys/net/ipv4/conf
for IFACE in `ls $RP_PATH`; do
   echo 0 > $RP_PATH/$IFACE/rp_filter
done
```

Once everything works, you should use the command ' ip route equalize' (search with Google for the syntax) to balance the load, or do a -j MARK --set-mark 0xN00 via iptables to redirect your traffig to WAN1 (0x100) or WAN2 (0x200)

If you need further assistance, post in forums or PM-me

The firewall scripts does not yet implement the full DD-WRT firewall, only the basic port/range forwarding.

This is script works for both outgoing AND incoming connections (most scripts only treats outgoing), so you should be able to accept connections in both links simultaneous (eg. web in one and FTP in another).

External links

http://roadrunnerguide.com/dualwan.html