

EzPlanet - Router - MultiHomedLB

[EzPlanet firmware multi homed configuration \(dual WAN\)](#)

Configuring EzPlanet WRT54GS firmware for dual WAN (multi homed)

There are a number of routers supported by **EzPlanet** running Linux that have a base WAN (Ethernet) port and one or more LAN Ethernet ports.

These routers could also be configured to load balance two (or more) WAN (ADSL) Internet links.

The instructions in this article are gathered from DD-WRT message board and, with some modifications, can be applied to set up any Linux box to load balance between two or more WAN links.

EzPlanet WRT firmware configuration

First of all you'll need to create a new vlan, called **vlan2**

Enter in the web interface, Settings / VLANs. Set the PORT4 to a new VLAN. Then enter in the console through a telnet or ssh session, and do the following commands:

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

here, the scripts I used to startup the interface. You need to call them in `rc_firewall`

wan2.firewall Code:

```
#!/bin/shWAN2_IFNAME=vlan2 WAN2_IPADDR=10.0.0.2 WAN2_GATEWAY=10.0.0.1 WAN2_NETMASK=255.0.0.0if [ "$(nvramp get wan2_ipaddr)" != "$WAN2_IPADDR" ]; then nvramp set wan2_ifname=$WAN2_IFNAME nvramp set wan2_ipaddr=$WAN2_IPADDR nvramp set wan2_gateway=$WAN2_GATEWAY nvramp set wan2_netmask=$WAN2_NETMASK nvramp commit fiifconfig $(nvramp get wan2_ifname) up $(nvramp get wan2_ipaddr) netmask $(nvramp get wan2_netmask)
```

routes.firewall Code:

```
#!/bin/ship rule fluship rule add lookup main prio 32766 ip rule add lookup default prio 32767ip rule add from $(nvramp get wan_ipaddr) table 100 prio 100 ip rule add fwmark 0x100 table 100 prio 101ip rule add from $(nvramp get wan2_ipaddr) table 200 prio 200 ip rule add fwmark 0x200 table 200 prio 201ip route flush table 100 ip route flush table 200for TABLE in 100 200 do ip route | grep link | while read ROUTE do ip route add table $TABLE to $ROUTE done doneip route add table 100 default via $(nvramp get wan_gateway) ip route add table 200 default via $(nvramp get wan2_gateway)
```

firewall.firewall Code:

```
#!/bin/sh#DD-WRT firewall rules #BEGIN #apply simple forward rules for RULE in $(nvramp 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 $(nvramp get wan2_ipaddr) -p udp --dport $SPORT -j ACCEPT iptables -A PREROUTING -t nat -p udp -d $(nvramp get wan2_ipaddr) --dport $SPORT -j DNAT --to $DEST:$DPORT #tcp #iptables -A FORWARD -d $(nvramp get wan2_ipaddr) -p tcp --dport $SPORT -j ACCEPT iptables -A PREROUTING -t nat -p tcp -d $(nvramp get wan2_ipaddr) --dport $SPORT -j DNAT --to $DEST:$DPORT else #iptables -A FORWARD -d $(nvramp get wan2_ipaddr) -p $PROTO --dport $SPORT -j ACCEPT iptables -A PREROUTING -t nat -p $PROTO -d $(nvramp get wan2_ipaddr) --dport $SPORT -j DNAT --to $DEST:$DPORT fi fi done#apply range forward rules for RULE in $(nvramp 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 $(nvramp get wan2_ipaddr) -p udp --dport $SPORT:$EPORT -j ACCEPT iptables -A PREROUTING -t nat -p udp -d $(nvramp get wan2_ipaddr) --dport $SPORT:$EPORT -j DNAT --to $TO #tcp #iptables -A FORWARD -d $(nvramp get wan2_ipaddr) -p tcp --dport $SPORT:$EPORT -j ACCEPT iptables -A PREROUTING -t nat -p tcp -d $(nvramp get wan2_ipaddr) --dport $SPORT:$EPORT -j DNAT --to $TO else #iptables -A FORWARD -d $(nvramp get wan2_ipaddr) -p $PROTO --dport $SPORT:$EPORT -j ACCEPT iptables -A PREROUTING -t nat -p $PROTO -d $(nvramp get wan2_ipaddr) --dport $SPORT:$EPORT -j DNAT --to $TO fi fi done iptables -A PREROUTING -t nat -p icmp -d $(nvramp get wan2_ipaddr) -j DNAT --to $(nvramp get lan_ipaddr)if [ $(nvramp get remote_management) -eq 1 ]; then iptables -A PREROUTING -t nat -p tcp -d $(nvramp get wan2_ipaddr) --dport $(nvramp get http_wanport) -j DNAT --to $(nvramp get lan_ipaddr):$(nvramp get http_lanport) fiif [ $(nvramp get dmz_enable) -eq 1 ]; then DMZ_IP=$(nvramp get lan_ipaddr | sed -r 's/[0-9]+$//')$(nvramp get dmz_ipaddr) iptables -A PREROUTING -t nat -d $(nvramp get wan2_ipaddr) -j DNAT --to $DMZ_IP fiiptables -A PREROUTING -t nat --dest $(nvramp get wan2_ipaddr) -j TRIGGER --trigger-type dnat iptables -A FORWARD -i
```

EzPlanet - Router - MultiHomedLB

```
$(nvram get wan2_ifname) -o $(nvram get lan_ifname) -j TRIGGER --trigger-type 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 CONNMARK --set-mark 0x100 iptables -t mangle -A PREROUTING -i $(nvram get wan2_ifname) -m conntrack --ctstate NEW -j CONNMARK --set-mark 0x200
```

1. 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 CONNMARK --set-mark 0x100 iptables -t mangle -A POSTROUTING -o $(nvram get wan2_ifname) -m conntrack --ctstate NEW -j CONNMARK --set-mark 0x200
```

1. Use the correct gateway for reply packets from the LAN

```
iptables -t mangle -A PREROUTING -i $(nvram get lan_ifname) -m conntrack --ctstate ESTABLISHED,RELATED -j CONNMARK --restore-mark
```

1. Use the correct gateway for reply packets from local connections

```
iptables -t mangle -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j CONNMARK --restore-mark#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 22,25,80,110,119,143,443,993,3389 -j MARK --set-mark 0x100#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 0x200#iptables -t mangle -I PREROUTING -i $(nvram get lan_ifname) -s 192.168.1.100 -j LOG --log-prefix 'P2P: '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 ip route equalize to balance the load, or do a **-j MARK --set-mark 0xN00** via iptables to redirect your traffic to **WAN1 (0x100)** or **WAN2 (0x200)**

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

This script works for both outgoing AND incoming connections (most scripts only handle outgoing), so you should be able to accept connections in both links simultaneously (WEB in one and FTP in another, for example).

Sample **ip route equalize** command:

```
ip route add default scope global equalize nexthop via 12.34.52.1 dev eth1 nexthop via 10.43.93.1 dev eth2
```

Remarks

This simple round-robin splitting of connection table entries, often doesn't "increase speed" as most people mean it. If you start up a single download it will still only be as fast as one of your connection. According to rules of round-robin this problematic situation is frequent:

1. HTTP/FTP download -> WAN A
2. DNS request -> WAN B
3. HTTP/FTP download -> WAN A

So you end up with WAN A crushed with load and B idle. I used this 4-5 years ago at a business. ON AVERAGE it works well. However the first thing most n00bs do when you hand them a recipe like this is run a speed test and complain that it hasn't doubled their speeds. Also, when you lose the round-robin coin-toss so to speak it can be a drag. I suppose with BitTorrent it could help since it's frequently opening connections.

If one connection goes down, all connections in progress over that link are just stuck. There is no failover mode.

EzPlanet - Router - MultiHomedLB

This method it may be argued, is not technically a "load balancing" method since there is no intelligent handling of traffic load, no rebalancing of connections if it becomes unbalanced, and no failover.

[EzPlanet firmware multi homed configuration \(dual WAN\)](#) (en)

Creator: XWiki.haydude Date: 2007/07/24 13:51

Last Author: XWiki.haydude Date: 2007/07/24 13:54

Copyright EzPlanet One