Dual, Triple (and probably quad) WAN with multiple active WAN links and source routing

This tutorial shows how to implement multiple active WAN links, source routing, and fail-over with DD-WRT. This has been tested in dual and triple WAN configurations. Scripts are list-based, so they should support a quad WAN configuration.

Multi-WAN Goals and Requirements:

• Modular design for scalability and testing;
• All WAN interfaces usable concurrently for traffic inbound from the Internet.
• Fault detection parameters configurable for each WAN interface;
• Detect failure of local loop/first hop, ISP routing, and excessive packet loss;
• Software fault detection and correction;
• Configuration through a single file;
• Easily reconfigurable routing;
• Dynamic DNS support for DHCP WAN addresses; and
• Seamless response to WAN IP address changes by DHCP.

Theory of Operation

Two scripts form the core of Multi-WAN: linkd and changewan. Each WAN interface is monitored by an instance of linkd. Each instance is independently configurable. A status file for each interface is maintained by linkd in /tmp/wan/status.*. When linkd detects a change of WAN status, changewan is called to evaluate the change. A status word, composed of two bits per interface, is assembled by changewan. This value is processed through a case statement to chose the appropriate routing configuration.

Remaining package files are described below.
My Configuration

My current configuration is a Netgear R7000 with two high speed connections (200/20 and 20/1.5 Mbps), plus a 2 Mbps backup. In normal (default) mode, the 200 Mbps connection handles entertainment streaming and most devices. The 20 Mbps connection is used for VoIP and trading ? the top 64 IP addresses in the LAN subnet. Should either high speed connection fail, all traffic is routed to the surviving high speed connection. The 2 Mbps connection is a final backup, used if both high speed connections are unavailable.

My prior configuration was a Linksys e3000 with two high speed connections (50/5 and 20/1.5 Mbps), plus a 2 Mbps backup. The e3000 could only sustain 80 to 90 Mbps, leading to the R7000 upgrade.

Installing and Configuring the Multi-WAN Package

First, complete single WAN configuration of your router. Configuration steps are well documented elsewhere, so details are not included here.

Single WAN configuration checklist:

- Install an appropriate DD-WRT version, this configuration has been tested on r14929, r15962 and 24345M;
- Internet interface on WAN port;
- Time server;
- DHCP server;
- Configure static DNS servers ? DNS servers must be accessible through all WAN interfaces.;
- DDNS if applicable;
- Wireless configuration;
- syslog;
- ssh management;
- Firewall and configure port mapping;
- Web access;
- Remote access;
- Enable cron;
- JFFS; and
- SNMP.

Set the following NVRAM configuration variables through the nvram set and commit interface:

- ddns?_hostname: Dynamic DNS service hostname for each WAN interface, e.g., ddns2, ddns3;
- ddns?_username: Dynamic DNS service user name;
- ddns?_passwd: Dynamic DNS service password;
- email_admin: E-mail address(es) for administrative messages, e.g., logwatch;
- email_dest: E-mail address(es) for event notifications;
- email_passwd: E-mail server password;
- email_server: E-mail server hostname;
- email_user: E-mail server user name;
- route_ips: List of IP addresses for Internet routing tests by linkd, addresses should be on separate networks and outside of your ISPs, a null list will inhibit routing tests;
- wan?_ipaddr: WAN IP address ? static addresses are set by user, dynamic addresses are set by udhcpc, e.g., wan2, wan3;
Dual, Triple (and probably quad) WAN with multiple active WAN links and source routing

- `wan?_netmask`: WAN interface netmask; and
- `wan?_ifname`: WAN interface name, e.g., vlan2, vlan3.

Set the following `/jffs/scripts/wan/global` environment variables with a text editor:

- `WANLIST`: List of monitored WAN interfaces, e.g., wan, wan2, wan3;
- `STATICWANLIST`: List of WAN interfaces with static IP addresses—exclude `wan`;
- `DHCPLIST`: List of WAN interfaces configured by DHCP; and
- `BASEDHCP`: Name of the VLAN configured by the DD-WRT base configuration, usually `vlan2`.

Multi-WAN package configuration:

- Copy Multi-WAN package tarball to `/jffs/scripts/wan` and extract contents, located here: https://www.dropbox.com/sh/ji5fgci1z0i4ggq4/AACRGIfFJzw7mw7zouHEv8dta?dl=0
- Remove port(s) from `vlan1` for new WAN link(s)
  - Dual: `nvram set vlan1ports="2 3 4 8*"`
  - Triple: `nvram set vlan1ports="3 4 8*"`
- Create VLANs for new WAN connection:
  - Dual: `nvram set vlan3ports="1 8" && nvram set vlan3hwname=et0`
  - Triple: `nvram set vlan4ports="2 8" && nvram set vlan4hwname=et0`
  - `nvram commit`

For a DHCP-only configuration, add to Startup in Control Panel at Administration->Commands->Startup:

```bash
sleep 5
udhcpc -i vlan3 -p /var/run/udhcpc-vlan3.pid -s /jffs/scripts/wan/udhcpc-vlan3
#'''Note:''' The next two lines are only for a triple wan configuration
sleep 5
udhcpc -i vlan4 -p /var/run/udhcpc-vlan4.pid -s /jffs/scripts/wan/udhcpc-vlan4
sleep 20
/jffs/scripts/wan/clockupdate
/jffs/scripts/dnsupdate
/jffs/scripts/wan/monitord
```

For a static IP-only configuration, verify `wan?_ipaddr`, `wan?_netmask`, and `wan?_ifname` in `nvram` for each interface, then add to Startup in Control Panel at Administration->Commands->Startup:

```bash
sleep 5
/jffs/scripts/wan/configstatic
sleep 20
/jffs/scripts/wan/clockupdate
/jffs/scripts/dnsupdate
/jffs/scripts/wan/monitord
```

Add to Firewall in Control Panel at Administration->Commands->Firewall:

```
/jffs/scripts/wan/firewall
/jffs/scripts/wan/route
```

Add the following cron jobs in Control Panel at Administration->Management->Additional cron jobs:

```
* * * * * root /jffs/scripts/wan/procmon
5 0,6,12,18 * * * root /jffs/scripts/wan/ddnsupdateall  # delete if not using dynamic DNS
0 10 * * * root /jffs/scripts/wan/clockupdate
0 11 * * * root /jffs/scripts/wan/logwatch
```
Link linkdupdate to /jffs/etc/config/linkdupdate.ipup to trigger update when the vlan2 DHCP address changes, command:

```
l /jffs/scripts/wan/linkdupdate /jffs/etc/config/linkdupdate.ipup
```

Reboot router.

Time to test the configuration:

- Verify that all WAN interfaces are up and have public IP addresses. This may require reconfiguration of your ISP's equipment to provide your dd-wrt router a public IP address;
- Test that configroute properly routes for each state: default, wan, wan2, and wan3;
- Check local dns configuration: `cat /etc/resolv.conf` should reflect proper domain and servers;
- Test sendstatus.new and sendstatus.old: `./sendstatus test test` ? configured e-mail address should receive messages, copy the appropriate version to sendstatus;
- Verify the following processes are running:
  - monitord;
  - linkd: one for each WAN interface; and
  - udhcpc: one for each DHCP-configured WAN interface;
- Check status files: cat /tmp/wan/status* -- route should be default, wan interfaces up;
- In a separate window, watch the wanlogger output: `tail -f /tmp/wan/wan.log` and execute the following commands:
  - `ifconfig vlan2 down`: wan goes to down state; all traffic flows to wan2; test that traffic flows as anticipated;
  - `ifconfig vlan2 up`: wan goes to link status;
  - `./configroutes wan2`: adds route test routing entries for wan; wan will go into recovering state; wan will remain in recovering state through the configured restore delay period, or you can force the link up immediately by executing `./linkup wan`; routing restored to default; test that traffic flows as anticipated;
  - `ifconfig vlan3 down`: wan2 goes to down state and all traffic is routed to wan; test that traffic flows as anticipated;
  - `ifconfig vlan3 up`: wan2 goes to link status;
  - `./configroutes wan3`: adds route test entries for wan and wan2; wan and wan2 will go into recovering state; wan will remain in recovering state through the configured restore delay period, or you can force the link up early by executing `./linkup wan2`; routing restored to default; test that traffic flows as anticipated;
- For a triple WAN configuration:
  - `ifconfig vlan2 down && ifconfig vlan3 down`: wan and wan2 go to down state and all traffic is routed to wan3; test that traffic flows as anticipated;
  - `ifconfig vlan2 up && ifconfig vlan3 up`: wan and wan2 go to link status;
  - `./configroutes wan3`: adds route test entries for wan and wan2; wan and wan2 will go into recovering state; wan and wan2 will remain in recovering state through the configured restore delay period, or you can force links up early by executing `./linkup wan2 && ./linkup wan2`; routing restored to default; test that traffic flows as anticipated.
Troubleshooting

Two troubleshooting mechanisms are included. Wanlogger levels can be set to \$LOG_DEBUG to be more verbose. These levels can be set globally in the global configuration file or locally overridden in each script. Additionally, setting a DEBUG environment variable prior to script execution will configure 'set -x' in scripts executed from the command line.

Customizing Your Configuration

My configuration is probably only a starting point. Configuration changes are made in these files:

- global: tuning parameters, file locations, and log levels;
- changewan: mapping wan states to wan routing configurations;
- configroutes: multi-wan routing configuration, and;
- route: source route subnet definitions.

Multi-WAN Files

apprestart: Restarts httpd to stop its 100 percent CPU utilization

changewan: Changes routes based on current WAN interface status. The Multi-WAN States spreadsheet is provided to help route planning.

clockupdate: Updates the router system clock from ntp_server.

configroutes: Configures routes as selected by changewan.

ddnsupdate: Updates DynDNS for a specific DHCP WAN interface.

ddnsupdateall: Updates DynDNS for all DHCP WAN interfaces.

dnsupdate: Updates resolv.conf with wan_domain and wan_dns.

firewall: Configures the router firewall.

global: Global configuration file.

linkd: Monitors a WAN link to detect faults and track status ? one linkd for each WAN interface.

linkdupdate: Triggers re-read of linkd global configuration and network parameters for running linkd instances. Link to /jffs/etc/config/linkdupdate.ipup to trigger update when the vlan2 DHCP address changes.

linkup: Forces a WAN link in Recovering state to Up status.

monitord: Monitors the status of vital processes (linkd, udhcpc, and cron) and relaunches processes when necessary.
monitordupdate: Triggers re-read of monitord configuration from global.

procmon: Launched by cron to check that monitord is running and to relaunch monitord when necessary.

route: Configures base IP routes.

sendstatus.old, sendstatus.new: Sends status messages, old and new versions are included to support multiple DD-WRT versions.

/tmp/wan/logwatch.*: logwatch output files

/tmp/wan/status*: Current route selection and WAN interface status. WAN interface status values maintained by linkd:

- up: Interface is successfully passing traffic to the Internet.
- recovering: Interface can successfully pass traffic to the Internet, but has experienced a failure within the configured restore delay.
- link: Interface is passing traffic to its first-hop gateway, but not to the Internet.
- down: Interface is not passing traffic to its gateway.

Route interface status values are user configurable in changewan and configroute:

- default: Traffic from the /26 LAN subnet is routed to wan2. Other traffic routed to wan.
- wan: All traffic is routed to wan.
- wan2: All traffic is routed to wan2.
- wan3: All traffic is routed to wan3.

/tmp/wan/wan.log: active wanlogger output file.

/tmp/wan/wan.log.*: archived wanlogger output files.

udhcpc-vlan3: udhcpc configuration file for wan2/vlan3. The target WAN interface is configured within this file.

udhcpc-vlan4: udhcpc configuration file for wan3/vlan4. The target WAN interface is configured within this file.

/tmp/wan/wan.log: Output of wanlogger, use 'tail -f' to monitor continuously.

**Multi-WAN files located here**

https://www.dropbox.com/sh/ji5fgcj1z0i4gg4/AACRGIIIFJzw7mw7zouHEv8dta?dl=0