OpenWRT - embedded Linux for wireless routers

Ted Faber
USC/ISI
USC Viterbi School of Engineering
22 Mar 2007
Disclaimer:
- Not an OpenWRT designer or developer
- There’s more than one way to do it

Outline

Big Picture
- OpenWRT
- Linksys WRT54GL
- networking

Practical Stuff
- buying
- flashing
- what you get
- how to get more

Example: my home network
Role of the Router

30,000 foot view

Default OpenWRT setup

- router does address translation for hosts
- can forward service requests to servers
- allocates local IP
- answers DNS queries
I've called you all here...

Embedded Linux for wireless routers
- Full command-line environment, but lean
- Appropriate device drivers
- WWW interface for simple configuration

Why screw around with flash on my router??
- Tweak the service configuration
  - firewall
  - local DNS
  - DHCP
- Get more out of the hardware
  - signal strength
  - afterburner
  - VLAN switch - DMZ
Let's talk about hardware...

**CPU/Filesystem resources**
- Broadcom 5352 200Mhz
- RAM 16 MB
- Flash 4 MB

**Networking**
- Wireless interface (including "afterburner")
- Ethernet bridge (bridges to wireless)
- VLAN switch

**Other Hardware:**
http://wiki.openwrt.org/TableOfHardware
Important Safety Tip

The WRT54G (not GL) is much less capable:
- 2 MB RAM 8 MB flash
- VxWorks OS
- very brickable

New from Amazon
- WRT54G: $49.99
- WRT54GL: $64.99
- WRT54GS: $69.99
- WRST54GS: $99.99 - USB!
The Innards of the WRT54GL

Mmmmm. Block diagram

Image from http://wiki.openwrt.org/OpenWrtDocs/NetworkInterfaces
What are VLANs for?

Image from http://wiki.openwrt.org/OpenWrtDocs/NetworkInterfaces
Network setup

don't burn any bridges...

Router lives on 192.168.1.1
Firewall rules to router open
Hooked to Internet for packages
Flashing the Firmware (WRT54GL)

Default webserver on 192.168.1.1:80
- user: admin password: admin

Download the right firmware
- http://downloads.openwrt.org/whiterussian/newest/
  - WRT54GL: default/openwrt-wrt54g-squashfs.bin
  - WRT54G: micro/openwrt-wrt54g-squashfs.bin
  - upgrade: openwrt-brcm-2.4-squashfs.trx

Choose "the update the firmware" option from the web server
- take a short walk - do not disturb

Other choices: TFTP, JTAG...
After the flash

Initial housekeeping

Telnet (!) in and set a root password
- initial install - no password
- later boots will disable telnet

Ssh server in place
- add ssh2 keys into /etc/dropbear/authorized hosts

Set boot_wait for safety

Can config a lot from the web server
- including all this
What comes with the default install out of the box

Linux 2.4.30 kernel

Utilities

- busybox
- telnet (but make sure it’s off)
- dropbear (ssh server)
- iptables (firewalling, NAT configuration)
- dnsmasq (DNS and DHCP on the 192.168.1.0 subnet, LAN port)
- udhcpc (busybox, WAN port)
Extensions

Extra packages:
- ipkg tool (std install)
- http://downloads.openwrt.org/whiterussian/packages/
- probably need http proxy

Compiling OpenWRT or packages
- http://wiki.openwrt.org/BuildingPackagesHowTo
- debian environment
- cross compile
- FreeBSD ports/Gentoo packages style
Linux on the OpenWRT

Filesystems

- /rom - read only files
- /tmp - memory file system
- /jffs2 - journalling flash file system
- mini_fo mitigates between jffs and rom (firstboot restores orig)

NVRAM (last 64K of flash)

- configuration options
  - scripts
  - commands
  - hardware config (VLANs & interfaces)
- nvram command manipulates this
Example: My Home Network

where the heart is

Server a choke point/my desktop
Wasted addresses in firewall
ISP believes we’re 1 happy subnet (proxy arp)

192.0.2.0 is the example subnet from RFC 3330, not my home net
Non-servers work if server down
Extra IP address
ISP believes we’re 1 happy subnet (proxy arp)

Plan - disable everything and build back up
Step 1: addresses and routing

getting packets in place

http://www.sjdjweis.com/linux/proxyarp/

Doing it

- Set addresses of WAN and LAN (WEP as well) using nvram
- Set proxy ARP on for ISP side (kernel feature!)
- iproute2
  - install with ipkg
  - set up routing with same address on 2 interfaces

Put it all (except nvram) into /etc/init.d/S80bridge
Edits to `/etc/user.firewall`

- disable host masquerading (NAT)
- add rules to allow local services on WRT54GL
  - NTP
  - DNS
- add rules for services on server
- default allows connections out
Step 3: DHCP and DNS

2 features, 1 program (dnsmasq)

http://wiki.openwrt.org/OpenWrtDocs/dnsmasq

DHCP
- override standard script completely
  - too helpful
- set to hand out my local addresses
  - timeouts, addresses, default routes, DNS servers, domain

DNS
- front end to my DNS and ISP DNS

Edits are to /etc/dnsmasq.conf
Step 4: Time Setting

http://wiki.openwrt.org/OpenWrtDocs/Configuration

Get ntpclient using ipkg

Write script to call
  - default setup failed due to my topology

Add cron call
That's it!
fun, fast, easy

http://wiki.openwrt.org/
http://www.openwrt.org/

Docs are good
Any time you touch firmware be patient
Fun little box that lets you explore networking