HEISON CHAK

Asterisk appliance

Heison Chak is a system and network administrator at SOMA Networks. He focuses on network management and performance analysis of data and voice networks. Heison has been an active member of the Asterisk community since 2003.

heison@chak.ca

WITH COMMUNITY EFFORT TO enhance Linux and other UNIX-based embedded systems, the creativity and ingenuity of these users are driving hardware manufacturers to deliver more robust and ever improved products. Both the Linksys WRT54G and the NSLU2 (a.k.a. “slug”) have attracted developers and users to modify firmware and hardware to suit their custom needs and gratification. This article will focus on running the Asterisk PBX on Linksys NSLU2 with the Unslung firmware.

What Is NSLU2?

Linksys NSLU2 is a NAS (Network Attached Storage) unit designed to share USB storage via the SMB protocol. NSLU2 stands for Network Storage Link for USB 2.0 disk drives. The NSLU2 features an Intel IXP420 (ARM) CPU, 32 MB of RAM and 8 MB of flash, an Intel IXP425 Ethernet interface, and a dual-port USB controller. Under the hood, it has a Linux-based embedded OS driven by a Web front end for users to manage and share the two USB-connected devices.

The devices can be USB memory sticks, USB card readers with a removable medium (CF, SD, etc.), or USB hard drives. Although NSLU2 features a 266-MHz ARM processor, it is limited to operating at half the speed of a 133-MHz processor, perhaps because of heat dissipation requirements in its small footprint. Despite the slow response and low throughput, the NSLU2 is an inexpensive way to add a hard drive or two to a network.

With community efforts, the NSLU2 can be flashed with different firmware depending on your interests:

- Linksys (original firmware, driven by Web interface, based on Linux 2.4.22)
- Unslung (modified firmware for beginners with little Linux knowledge, based on Linux 2.4.22)
- SlugOS (for experienced Linux users who need software from different repositories, based on Linux 2.6)
- OpenSlug
  - OpenDebianSlug
  - SlugOS/LE
  - GentooSlug
  - UcSlugC
- Debian/NSLU2 (based on SlugOS kernel patches, Linux 2.6)

NSLU2 Firmware Replacement

The original Linksys firmware is based on a 2.4.22 Linux kernel and uses RedBoot as its bootloader. RedBoot expects to find the kernel in the 8-MB flash at address 0x50060000 (/dev/mtdblock2).

Flash memory is partitioned into four "/dev/mtdblock" devices, as shown in the following table:

Linux device	Type	Start address	Length	Description
/dev/mtdblock0	RedBoot	0x50000000	256 KB	Contains code from which the IXP420 boots
/dev/mtdblock1	System	0x50040000	128 KB	NSLU2 configuration Config, e.g., IP address
/dev/mtdblock2	Kernel	0x50060000	1 MB	Linux kernel
/dev/mtdblock3	RAMdisk	0x50160000	6.625 MB	Ramdisk image for /

Unslung is one of the many firmware replacements for NSLU2; it is based on the same firmware that the manufacturer ships except that it enables support for telnet and unslinging. Unslinging refers to the procedure of copying the filesystem to an attached USB device (e.g., a 2-GB USB stick is used for testing) and booting off the external device. Additional space on the root device allows installation of extra software and other custom packages.

The Unslung 6.8 firmware, along with instructions on how to flash the firmware, can be found at http://www.slug-firmware.net/u-dls.php. It is crucial to note that the firmware *must* be flashed when *no* USB device is connected. Once the firmware is updated, telnet can be enabled and then one can run the unsling script (/sbin/unsling). This divides the target USB device into three partitions (ext3 root, ext3 conf, and swap).

Although using a flash drive is often an attractive way to run an Unslung NSLU2, the low-power, low-cost, and quiet appliance may be exposed to the danger of failing owing to wearing of the flash drive. To extend the life of flash devices, it is best to disable swap and mount the ext3 filesystems so that access time is not updated.

If a file named /.ext3flash exists, the Unslung boot scripts will remount the root and conf partitions with the appropriate options (-o noatime for both and sync for conf) and disable swapping to the root drive by running swapoff.

Installing Packages on NSLU2

Similar to Debian apt-get, an Unslung NSLU2 can install ported packages onto the external USB flash by using the ipkg command. There are 900+ packages available for the Unslung firmware. For those who are more adventurous, OpenSlug is a another firmware to explore, as it has more ported packages and is better suited for compiling and building packages.

The update and upgrade options of ipkg work in similar fashion as in Debian; it looks for package lists in the archives found in /etc/ipkg/[cross|native]-feed.conf and performs upgrades to installed packages and their dependencies:

```
# ipkg update ; ipkg upgrade
```

For the purposes of this article, the NSLU2 is intended to run as an Asterisk server. Thus, the following packages are installed:

```
# ipkg install openssh ntp asterisk14 asterisk14-core-sounds-en-ulaw
        asterisk14-extra-sounds-en-gsm asterisk14-extra-sounds-en-ulaw
        asterisk14-gui
```

Packages installed by ipkg live under /opt; otherwise they reside in their normal path. In the case of Asterisk, the configuration files can be found under /opt/etc/asterisk and the sound files are located in /opt/var/lib/asterisk/sounds. Running Asterisk on the NSLU2 isn't much different from running it on x86 or AMD CPUs.

Connecting NSLU2 to the PSTN

Once Asterisk registers itself to a SIP or IAX provider, VoIP handsets and soft clients provisioned to use the NSLU2 can accept calls from the PSTN through the VoIP service provider:

```
SIP.conf: register => username:password@voip.service.provider
```

This is probably good enough for those who are planning to run Asterisk as a pure VoIP application. For those who want to connect Asterisk to the PSTN via FXO (Foreign Exchange Office) or prefer to connect an analog phone to Asterisk, NSLU2 doesn't quite meet the bar.

One can go about writing one's own driver for USB-based FXO/FXS interfaces and worry about fitting all those drivers in only 32 MB of RAM. But why reinvent the wheel when there is an inexpensive and quick solution around the corner?

The Grandstream GS-488 ATA and the Linksys 3102 ATA (analog telephone adapters) are examples of SIP-based media gateways that feature both FXO and FXS interfaces. Both are affordable low-power, low-cost, and quiet VoIP appliances. (The GS-488 has been reported as unable to pass inbound callerID to SIP. Other than this small glitch, the two pieces of equipment are comparable.) The ATAs will take the burden of POTS line and analog devices (phone or fax) out of the NSLU2. Incoming calls from the PSTN to the POTS line will come through the ATA, then get converted to SIP UDP packets destined for Asterisk (e.g., an IVR menu). If the PSTN incoming call ends up going to an analog phone (as a result of the IVR menu), it can be sent as SIP UDP packets from Asterisk to the ATA and ring on its analog port.

If you terminate inbound VoIP calls from the service provider and originate VoIP-based calls to the PSTN via the service provider, you now have the ability to do this on either the NSLU2 or the ATA. You can also use the ATA as the interface for other Asterisk installations, such as the one I described in my April 2007 column which runs within a virtual machine without needing to install device drivers.