Rutgers University - WINLAB

Luis R. Rodriguez <mcgrof@winlab.rutgers.edu>

Accomplishments from December, 2006 to January 31, 2008

Revision: 3.14

I was hired at WINLAB to work on enhancing the drivers used for research and providing mechanisms to help enrich such research. WINLAB uses GNU/Linux and their main chipset on the 40x40 grid is the Atheros AR5212 802.11a/b/g chipset. The AR5212 chipset is what we call a SoftMAC chipset [1] in the industry, one which requires the MLME [2] state management to be handled in software. The driver used to support the AR5212 in Linux was the MadWifi [3] driver. This is the driver that we used throughout most of 2006. The MadWifi driver has three parts to it, the SoftMAC layer (net80211), the Atheros driver based on the SoftMAC layer provided (ath/ directory), and the proprietary Hardware Abstraction Layer (HAL, and provided by the hal/ directory of the driver). While patching MadWifi for WINLAB I was found incapacitated by the lack of extensions which could be added due to the nature of Linux wireless extensions [4] and the lack of access to finer grainer control of the hardware because of the proprietary HAL so I set out to look for alternative drivers for us. This search ended up with in a large unexpected tedious legal uphill with the help of the Software Freedom Law Center (SFLC) [5] helping us by evaluating the development history of Reyk Floeter's OpenBSD reverse-engineered Atheros HAL, ar5k [6] (November, 2006), then doing a code audit with the help of Atheros of the official proprietary Atheros HAL against our own (MadWifi) Linux port of ar5k called OpenHAL [7] (published July 31, 2007), and finally reviewing line-by-line changes and attributions to this code base to ensure we can help OpenBSD reap benefits of our changes and that we could finally include this into the Linux kernel [8] (published September 27, 2007). In the end this all created certainty in the possibility of including an Atheros FOSS driver in the Linux kernel which prompted a change of focus in development to MadWifi team [9] (announced on September 21, 2007), declaring MadWifi legacy and ath5k the new development focus. The ath5k Linux driver is scheduled for inclusion in the next release of the Linux kernel, for 2.6.25. There is still a lot of work ahead though, I help advance the driver it and maintain it along with Nick Kossifidis, who had originally ported ar5k to Linux (OpenHAL) and Jiri Slaby, who completed the port of MadWifi and the OpenHAL to mac80211.

During the beginning of 2007 I was challenged to come up with a framework which would enable intelligent Access Points to come up with a way to measure the local environment (temperature, signal strengths, currently used channels, modulations and so on) and communicate to stations a set of rules they could abide by to enhance communication. The idea behind this was to pave the way for usage of Software Defined Radios (SDRs) and enhance communication. In addition to enhancing communication one must also be very aware of local regulatory laws and abide by them. I set out first to compile in a database all known regulatory laws and their restrictions for frequency band usage, power limits (maximum IR for PtP and PtMP links), maximum allowed antenna gain, and map them to countries. Much of this work, except power maps, was done by Reyk Floeter's ar5k effort. As it turns out this database can be used to centralize regulatory domain control for an Operating System's wireless subsystem, essentially providing regulatory domain control for SoftMAC drivers that don't have such code implemented and to eliminate redundancy in code for such efforts across separate drivers. I set out to implement first then a Central Regulatory Domain Agent (CRDA for short) as a way to introduce at a later point more finer grained control SDR rules for interoperability. Patches have been sent upstream and have been reviewed by the community [10] (posted September 21, 2007). Upon review it has been determined this database will go into userspace and only one regulatory domain definition uploaded to the kernel at a time. The third revision of this work is now underway.

A CRDA should become a necessary component of wireless subsystems to ensure regulatory compliance of wireless devices where the regulatory control is left to the host to implement and to avoid driver redundancy. Some wireless devices use firmware to ensure regulatory control but if their firmware is reversed engineered vendors and developers will be faced with the same problem and redundancy, specially in FOSS platforms where FOSS solutions are preferred over binary solutions in firmware and simply not possible for integration due to license incompatibilities on driver code. Having a CRDA can provide additional effort of regulatory integrity checks for those wireless devices that do not have such restrictions handled in hardware and provide complete regulatory control for devices which left regulatory control implementation up to the host. Embracing a strong CRDA, using a public database, and allowing for these attributes to be further expanded, along with other mechanisms such as a frequency broker [11] (proposed May 2007) to enable smart host-awareness of wireless capabilities and current communication, can give way for future more complex intelligent and efficient wireless networks.

When certifying wireless products for sale in the United States a wireless product vendor (Dell, IBM, HP) has two options   seek certification under part 15 rules of the FCC or seek SDR certification. Up until now most wireless vendors have certified their devices under part 15 rules. Certification under part 15 rules is software agnostic -- only the hardware is tested. SDR certification is a relatively new option and laws governing SDR devices are still not very clear. There have been a few companies who have obtained SDR certification for their devices though, amongst them Vanu in November of 2004 for a GSM base station [12] and later in September of 2005 Cisco for an 802.11a Access Point [13] which became the industry's first 802.11 SDR product. There has been fear of supporting FOSS due concerns that FCC's new clarification may outlaw FOSS SDRs but in fact the opposite is true. The FCC now clearly acknowledges usage and importance of FOSS in the wireless devices and defines strict requirements which would need to be met should certification be made using FOSS [14] (published July 6, 2007). Despite this clarification there is still fear of the impact that SDR certification will have on wireless businesses and some vendors are reluctant to pursue this path.

Lifting the veil over the obfuscation over Atheros' proprietary HAL has set in motion a slew of discussions about Software Defined Radios (SDRs), how they are regulated and what the implications are for FOSS. The industry itself is wary as to how local regulatory laws can apply to their devices with more concern over the release of their next wireless product. With the reverse engineering of the Atheros HAL and with the development of a replacement driver underway for inclusion in the Linux kernel it has been proven that security through obscurity is not a strong enough method which can be relied upon to ensure regulatory domain control. Some vendors use firmware as a way to support FOSS driver development in hopes that that itself can ensure their state under part 15 rules of the FCC for certification as it can be argued that firmware might be more difficult to reverse engineer. Some vendors are afraid what to do next, and those that have followed the SDR path for 802.11 have asked the FCC for clarification of SDR rules in hopes that other vendors with similar hardware configuration would follow the same path. The FCC has provided such clarification, "the FCC clarified the scope of the rules to require certification under the new process of any device that uses software to comply with the Part 15 regulations if such software is 'designed or expected to be modified by a party other than the manufacturer.'" but also clarified usage of FOSS under SDR rule FCC by stating that "a SDR device which uses FOSS to build the 'security measures' protecting the software against modification would face a 'high burden' during the certification process 'to demonstrate that it is sufficiently secure'" [15] (interpretation by SFLC published on July 6, 2007). While the regulatory bodies advance certification requirements and the industry works on what to do with respect to these new rules the work I have done at WINLAB has allowed us to pave the way for embracing SDR in the FOSS community through ideas which have been reviewed and improved upon by the community.

Apart from work with ath5k, mac80211, and a CRDA I also maintain the baseline images (used during experimentation) and PXE boot images (used internally during imaging) used on the 40x40 wireless grid. We currently use a Debian based Linux distribution as base for our images. To enhance the experience of the experimenter we seek to reduce the speed it takes to image the grid. We accomplish this by optimizing the size of our images. It is also very desirable to support as many wireless drivers as possible with one standalone image. To help with size and performance enhancements I have set out a guideline to move slowly towards the smallest possible PXE boot images with added performance enhancements [16]. Interestingly it might be possible to use this work as evidence of big performance degradation on the Linux kernel from 2.6.9 to 2.6.23 as kernels with similar configurations and the same initramfs yield better performance on 2.6.9. The original PXE boot image consists of 5.2 MB and the 2nd phase PXE boot image I produced consists of 3.1 MB (788 KB for the kernel and 2.3 MB for the initramfs) for total savings of 40% in size. There is a 3rd phase proposed for the PXE boot images but this work requires an internal overhaul of utilities used for imagine. Kernel performance issues are still being worked on and will have to be addressed within the community. The baseline images are now streamlined, supporting as much hardware as possible except for specialized hardware such as the GNU Radios which require specialized software and a large development environment which merits its own separate image [17]. The baseline image has also been optimized for size, its size has been reduced by 48% from the original image size (from 357 MB down to 187 MB). Drivers for new wireless hardware have been added using the same kernel supported by the distribution image used (2.6.22) using a backward-compatibility package to avoid separate complex images. I provided this backward-compatibility package and maintain it for the community [18] (initial release on November 7, 2007). This compatibility package allows users or experimenters to use, test and modify any of latest Linux wireless drivers and the mac80211 stack itself on kernels >= 2.6.22. One advantage for example is experimenters can now patch the bleeding edge 802.11s advancements without much effort.

Finally to help increase the amount of documentation available for experimenters wishing to tweak wireless drivers, the internals of the SoftMAC stack we use, and status of support for new wireless cards I have worked with the FOSS community to help document all new Linux wireless advances and provide Linux wireless development documentation on a central wiki [19]. The de-facto standard for 802.11 frame injection and reception, which Linux and all the BSD families have embraced is now being documented in a central wiki as well [20].

[1] http://linuxwireless.org/en/developers/Documentation/Glossary#SoftMAC
[2] http://linuxwireless.org/en/developers/Documentation/Glossary#MLME
[3] http://madwifi.org/
[4] http://linuxwireless.org/en/developers/Documentation/Wireless-Extensions
[5] http://www.softwarefreedom.org/
[6] http://lwn.net/Articles/209472/
[7] http://www.softwarefreedom.org/news/2007/jul/31/openhal/
[8] http://www.softwarefreedom.org/news/2007/sep/27/wireless-review/
[9] http://kerneltrap.org/Linux/MadWifi_Switches_Focus_to_ath5k
[10] http://www.kernel.org/pub/linux/kernel/people/mcgrof/v2-regdomain-patches/
[11] http://linuxwireless.org/en/developers/FrequencyBroker
[12] http://www.allbusiness.com/electronics/computer-electronics-manufacturing/5584326-1.html
[13] http://www.unstrung.com/document.asp?doc_id=81017
[14] http://www.softwarefreedom.org/news/2007/jul/06/sdr-paper/
[15] http://www.softwarefreedom.org/resources/2007/fcc-sdr-whitepaper.html
[16] http://orbit-lab.org/wiki/Documentation/orbit-pxe/specs
[17] http://orbit-lab.org/wiki/Documentation/SupportedImages
[18] http://linuxwireless.org/en/users/Download
[19] http://linuxwireless.org/
[20] http://www.radiotap.org/