Experimenting Software Radio with the Sora Platform

Jiansong Zhang Kun Tan Sen Xiang Qiufeng Yin Qi Luo Yong He
Ji Fang Yongguang Zhang
Microsoft Research Asia
Beijing, China

ABSTRACT

Sora is a fully programmable, high performance software radio platform based on commodity general-purpose PC. In this demonstration, we illustrate the main features of the Sora platform that provide researchers flexible and powerful means to conduct wireless experiments at different levels with various goals. Specifically, the demonstrator will show four useful applications for wireless research that are built based on the Sora platform: 1) A capture tool that allows one to take a snapshot on a wireless channel; 2) a signal generation tool that allows one to transmit arbitrary baseband wave-form over the air, from a monophonic tone to a complex modulated frame; 3) an on-line real-time receiving application that uses the Sora User-Mode Extension; and 4) a fully featured Software radio WiFi driver (SoftWiFi) that can seamlessly inter-operate with commercial WiFi cards.

Categories and Subject Descriptors

C.2.1 [**COMPUTER-COMMUNICATION NETWORKS**]: Network Architecture and Design—*Wireless communication*

General Terms

Algorithms, Design, Experimentation

Keywords

Software radio, Sora, Wireless experiment

1. INTRODUCTION

Wireless networking is ubiquitous and has become an essential part of global Internet. Recently, there shows a clear trend that wireless networking research has been crossing from higher layer ((*i.e.* MAC and above) into the low layer primitives (*i.e.* PHY). However, it remains a challenge to experiment with such research work, especially with high-speed modulations and wide spectrum band. It is because that low layers of wireless are typically implemented in hardware, which is either based on ASIC that has very limited reconfigurability (*e.g.* Atheros chips with MadWifi driver [2]), or is based on FPGA and requires complicated programming (*e.g.* WAPR [4]).

To address this challenge, we develop Sora, a fully programmable, high performance software radio platform based on commodity general-purpose PC [3, 5]. Compared to hardware solutions, Sora allows researchers to implement high-speed wireless protocols entirely in software with familiar programming environment, *e.g.* C/C++ language, and conduct real over-the-air experiments. Compared with other GPP-based SDR solutions (*e.g.* GNURadio/USRP [1]), Sora can achieve the full fidelity of state-of-the-art wireless protocols in real high-speed, wide-band environments, while using standard operating systems and applications.

In this demonstration, we will illustrate the main features of the Sora platform that provide researchers flexible and powerful means to conduct wireless experiments at different levels with various goals. Specifically, the demonstrator will show four useful applications for wireless research that are built based on Sora:

1. A capture tool that allows one to take a snapshot on a wireless channel. The captured signals are stored locally in a hard-drive file. This file can be later fed to other off-line processing tools for analysis or display.

2. A signal generation tool that allows one to transmit arbitrary baseband wave-form over the air, from a monophonic tone to a complex modulated frame. This generated wireless signal can work as a stimulus source to test receivers or measure wireless channels.

3. An on-line real-time receiving application that uses the Sora User-Mode Extension. This receiving application runs in the User Mode, and is able to receive digital signals from radio front-end in real-time manner with very low latency. Moreover, the user-mode application can still enjoy the *core dedication* service provided by Sora platform. Thus, researchers can write high-performance user-mode process programs with minimal disruptions.

4. A fully featured Software radio WiFi driver (SoftWiFi). The driver exposes a normal Ethernet interface to the upper TCP/IP layer with fully software implemented IEEE 802.11a/b/g PHY and MAC. SoftWiFi can communicate seamlessly with commercial 802.11 NICs at various data rates up to 54Mbps. Implemented as Kernel-mode drivers, software radios will have less overhead and enjoy near-metal short responsive time. Further, it also enables researchers to deploy and experiment with rich existing network applications on their software radio test-beds.

2. SORA OVERVIEW

Architecturally, Sora platform contains both hardware and software components, as shown in Figure 1.

The hardware components in the Sora architecture are a new radio control board (RCB) with an interchangeable radio front-end

SIGCOMM'10, August 30–September 3, 2010, New Delhi, India.
ACM 978-1-4503-0201-2/10/08.

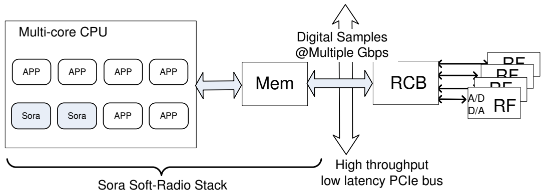

Figure 1: The Sora system architecture.

(RF front-end). The radio frontend is a hardware module that receives and/or transmits radio signals through an antenna. Multiple wireless technologies defined on the same frequency band can use the same RF front-end hardware, and the RCB can connect to different RF front-ends designed for different frequency bands.

The RCB is a new PC interface board for establishing a high-throughput, low-latency path for transferring high-fidelity digital signals between the RF frontend and PC memory, as shown in Figure 3. To achieve the required system throughput, the RCB uses a high-speed, low-latency PCIe bus. With a maximum throughput of 16Gbps (PCIe x8) and submicrosecond latency, it is well-suited for supporting multiple gigabit data rates for wireless signals over a very wide band or over many MIMO channels.

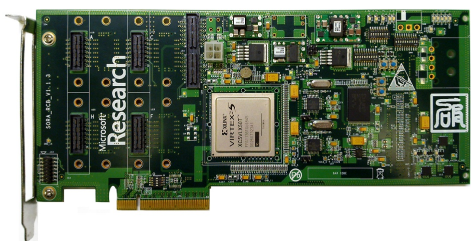

Figure 2: Sora radio control board.

Figure 3 illustrates Sora's software architecture. The software components in Sora provide necessary system services and programming support for implementing various wireless PHY and MAC protocols in a general-purpose operating system. In addition to facilitating the interaction with the RCB, the Sora platform provides a set of techniques to greatly improve the performance of PHY and MAC processing on GPPs. To meet the processing and real-time requirements, these techniques make full use of various common features in existing multi-core CPU architectures, including the extensive use of lookup tables (LUTs), the efficient partitioning of streamlined processing over multiple cores, exclusive dedication of cores for software radio tasks and substantial data-parallelism with CPU SIMD extensions. Sora provides an high-level abstraction of SIMD operations, named *Vector1*, that allows developers to exploit SIMD capability of modern processors in an architecture-independent manner.

Developers can write both user-mode processing applications or kernel network drivers for software radio processing. The Sora User-Mode Extension provides a convenient way for user-mode applications to access digital samples received from RF front-end in real-time with low latency.

3. DEMONSTRATION

Our demonstrations takes four sample applications that are built on the Sora platform.

Wireless channel capture tool. Sora provides API for applications to capture a snapshot on a wireless channel and store the signals in a local file. The snapshot is 16M bytes, which is translated to a period of 100 ms for a 40MHz wireless channel. The capture

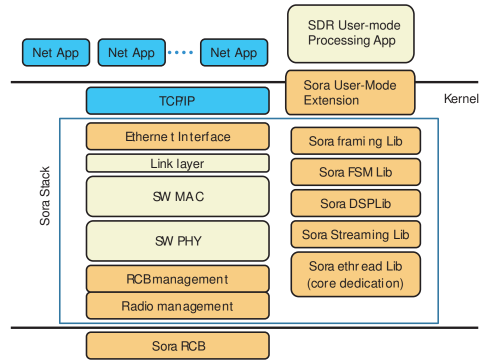

Figure 3: Sora software architecture.

file contains the raw I/Q signals and can be further processed by other off-line analysis tools. In this demonstration, we will show a graphic-based spectrum analyzer that parses the store signals and visualize the results on the screen.

Wireless signal generator. An application program can further instruct Sora platform to transmit an arbitrary waveform over the air. The waveform can be a simple monophonic tone or a complex modulated signal. In this demonstration, we will use the tool to generate a signature signal to precisely measure the wireless channel coefficients, and transmit a pre-modulated frame to test our receiver.

On-line real-time SDR receiving program. This is a user-mode program that receives and processes real-time digital signals using the Sora User-Mode Extension. A user-mode program is easy to write, debug and deploy. With the Sora platform, it is also possible for a user-mode program to use the *core dedication* service. Thus, efficient DSP algorithms can be implemented in a multi-core system with minimal disruption. We demonstrate two receivers, one ZigBee and one 802.11. These two receivers can work simultaneously on the same Sora hardware.

Fully featured SoftWiFi Driver. The driver implements a fully featured 802.11a/b/g PHY and MAC that can seamlessly inter-operate with commercial NICs at all data rates up to 54Mbps. It exposes a virtual Ethernet interface to the operating system. Thus, existing network applications can run unmodified over a software radio network.

These sample applications cover several common useful scenarios of experimenting wireless. We note that part of these samples are included in Microsoft Software Radio Kit for academic [3] and they are extensible and can be easily customized for future applications.

4. REFERENCES

[1] Gnu radio. *http://www.gnu.org/software/gnuradio/*.

[2] Madwifi. *http://sourceforge.net/projects/madwifi*.

[3] Microsoft research software radio platform. *http://research.microsoft.com/en-us/projects/sora/academickit.aspx*.

[4] WARP: Wireless open access research platform. *http://warp.rice.edu/trac*.

[5] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W. Wang, and G. M. Voelker. Sora: High performance software radio using general purpose multi-core processors. In *NSDI 2009*.