Demo Abstract: Open RT-WiFi Platform on Software-Defined Radio

Zelin Yun†, Peng Wu†, Shengli Zhou†, Aloysius K. Mok‡, Mark Nixon§, Song Han†
†University of Connecticut {zelin.yun, peng.wu, shengli.zhou, song.han}@uconn.edu
‡University of Texas at Austin mok@cs.utexas.edu
§Emerson Automation Solutions mark.nixon@emerson.com

2022 IEEE 28th Real-Time and Embedded Technology and Applications Symposium (RTAS) | 978-1-6654-9998-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/RTAS54340.2022.00038

I. Introduction

Smart factory automation has an ongoing trend to employ high-speed real-time wireless technologies to interconnect heterogeneous industrial assets to perform various sensing and control services, and support mobile equipment to conduct designated tasks in a collaborative fashion. Applications in automation industries usually have stringent requirements on both high data rates and deterministic real-time performance. Existing efforts, however, either cannot meet the performance requirements or are based on commercial-off-the-shelf (COTS) hardware and cannot provide full-stack configurability [1].

In this paper, we present a software-defined radio (SDR) [3] based RT-WiFi solution, namely SRT-WiFi [4], to address the above limitations of existing solutions. SDR is a radio communication system where components traditionally implemented in hardware are instead implemented by means of software on a PC or an embedded system. We design SRT-WiFi based on an advanced SDR platform (Zynq-7000 and AD9364) where the radio functions are programmed on FPGA. Thus, the physical layer (PHY) and data link layer (DLL) of SRT-WiFi can run in real time since the radio functions are achieved by the logic blocks in FPGA driven by an oscillator. With such a programmable real-time radio system, SRT-WiFi achieves the key functions required to support high-speed real-time wireless communications, and also provides an open-source platform to support ever-evolving IEEE 802.11 standards.

II. System Architecture of SRT-WiFi

SRT-WiFi is based on the Openwifi project [2] which is Linux mac80211 compatible IEEE802.11 design. SRT-WiFi has two major components as shown in Fig. 1: the Processing System (PS) running on an ARM processor and the Programmable Logic (PL) running on an FPGA. PS is a Linux OS running the non-real-time part of the DLL and all the other higher layers. PL runs the real-time part of the DLL and the PHY. In PL, the TXI and OFDM TX modules handle the packet transmissions and modulations, respectively, while the RXI and OFDM RX modules handle packet demodulations and receptions, respectively. The XPU (application-specific processing unit) module runs the state machine of IEEE 802.11 channel access methods to control the transmissions and receptions. On the other hand, PS runs the major part of DLL which is the MAC80211 subsystem interfacing with PL through the MAC80211 driver. The MAC80211 driver relies on the sub-drivers as APIs to exchange data with PL.

The key design goal of SRT-WiFi is to support precise time synchronization and real-time communications with effective rate adaptation at run time. For this purpose, in the following we present the SRT-WiFi architecture by modifying PL and PS in Openwifi to add the desired functions.

SRT-WiFi PL: The PL component of SRT-WiFi is designed to 1) achieve the real-time packet transmissions with high precision of time synchronization, and 2) measure the reception SNR of the links precisely in order to provide reference for effective rate adaptation. To achieve real-time packet transmissions, we design a TDMA block (see Fig. 1) in XPU to supplement the existing CSMA block. The TDMA block triggers the PHY and DLL activities with high time precision. It either runs according to the local timer or synchronizes with another device in the SRT-WiFi network. The synchronization time error and standard deviation in the SRT-WiFi are as low as 0.7 µs and 0.1 µs, respectively. The TDMA block triggers the packet transmissions according to a schedule maintained by the network manager in PS through a TDMA driver. To achieve the second design goal, we leverage the capability of SRT-WiFi to have direct access to the received signals, and develop methods to measure the SNR precisely and implement it in the OFDM RX module.

SRT-WiFi PS Kernel: As shown in Fig. 1, we add the TDMA driver and modify the MAC80211, TX and OFDM RX drivers to provide an interface for PS and PL to exchange the schedule, data rate and SNR information in the TDMA mode. The TDMA driver is registered in the kernel as a miscellaneous driver (MSIC) to provide APIs for the network manager. The network manager configures the schedule and data rate in PL through the TDMA driver. The OFDM RX driver is enhanced to support reading the SNR values measured in PL.

SRT-WiFi Network Management: SRT-WiFi employs a hierarchical network management framework, consisting of network managers running on the APs and device managers running on the stations. These network managers are designed for two purposes: 1) to exchange information at the application layer among the APs and devices, including the schedule, data rates and SNR of links, 2) to manage the TDMA DLL on each device such as configuring the schedule for the TDMA block and reading the SNR measurement from PL.

III. Testbed Setup and Demonstration

Fig. 2 presents the hardware devices used in our SRT-WiFi testbed. ZC706 consists of Z7045 SoC and AD9364 radio chip. It is used as the hardware for both AP and stations. ADRV9364-Z7020 consists of Z7020 SoC and AD9364 radio chip. It is only used for some stations due to its limited

978-1-6654-9998-9/22/$31.00 ©2022 IEEE
DOI 10.1109/RTAS54340.2022.00038

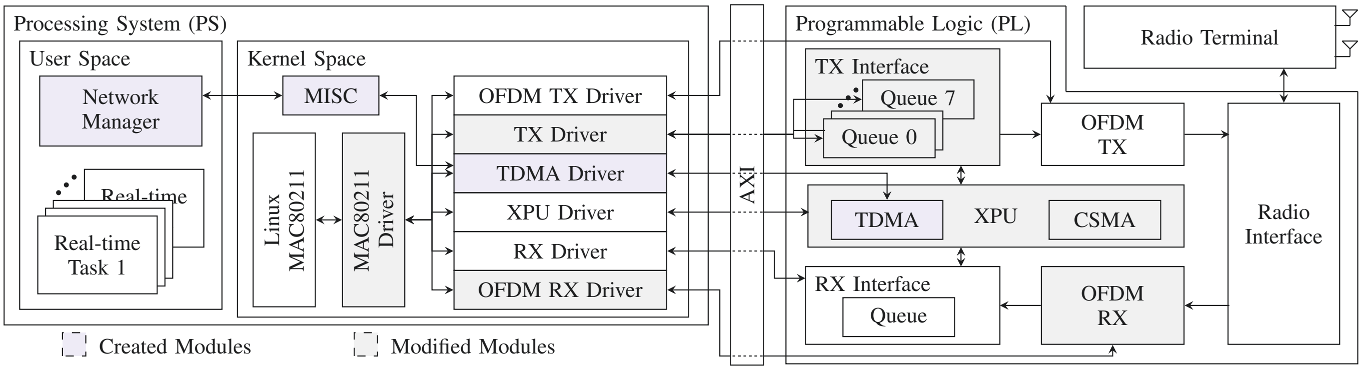

Fig. 1: Overview of the SRT-WiFi system architecture design based on the Openwifi project [2].

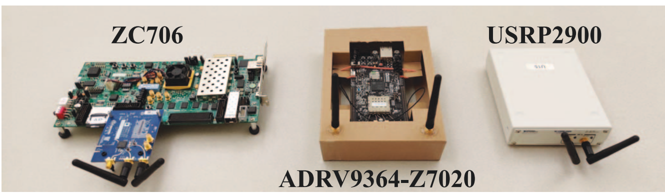

Fig. 2: SDR hardware used in the SRT-WiFi testbeds.

FPGA resources. USRP2900 is a traditional SDR device used for testing purpose such as signal analysis and interference generation. Fig. 3 shows the testbed of our proposed SRT-WiFi network where multiple APs are synchronized and connected to a backbone network (the router). A central network manager in the backbone network manages all the network resources and allocates them to individual APs. In each SRT-WiFi network, high-speed real-time point-to-point wireless communications with rate adaptation are supported.

We plan to demonstrate the key features of SRT-WiFi using the following three demos.

Demo 1: SRT-WiFi network setup. This demo shows a typical setup of a SRT-WiFi network, including the device joining process. We will deploy one SRT-WiFi network with one AP and two SRT-WiFi stations. The demo will show how the AP starts and connects to the central network manager to obtain the basic information of the network. Next, the stations will be powered on and then connect to the AP with given SSID. We will use a COTS WiFi adaptor to serve as the sniffer to catch the packets with timestamps to show the schedule of the joining process. After joining the SRT-WiFi network, the stations will obtain the assigned schedule and communicate with the AP accordingly. The new communication patterns will also be captured and demonstrated through the sniffer.

Demo 2: SRT-WiFi synchronization. In this demo, we show the time synchronization performance of SRT-WiFi. We will set up two APs (AP1 and AP2) and let AP2 synchronize with AP1 through beacon packets. In this demo, we use USRP2900 to measure the synchronization error. The USRP2900 device will capture the baseband signal of AP1 and AP2 so that we can measure the exact physical layer synchronization performance. At the same time, we will also use the COTS

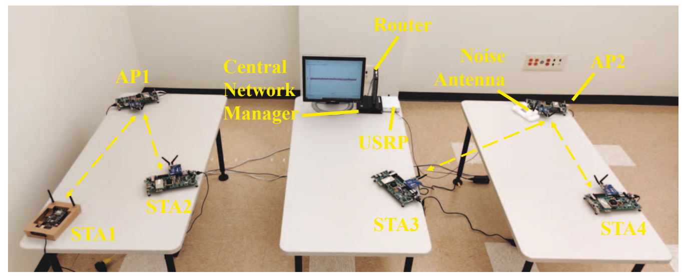

Fig. 3: An overview of the multi-cluster SRT-WiFi testbeds.

WiFi sniffer to capture the timestamps and compare them with the USRP2900 results to show that our SRT-WiFi has better timing performance than the COTS hardware.

Demo 3: SRT-WiFi rate adaptation. In this demo, we will show how the rate adaptation mechanism of SRT-WiFi works. We will set up one AP and one station in this demo and use USRP2900 to add interference to the network. The network manager on both devices will send the measured link quality information to the central network manager. The central network manager will decide the data rates of the links and return the latest rate configuration along with the schedule information to the devices. We will use the COTS WiFi sniffer to capture the packets to show the data rate changes against the measured signal-to-noise ratio of each link.

IV. Acknowledgement

This work was partly supported by the National Science Foundation under Award CNS-2008463 and TI-1919229.

References

[1] Y.-H. Wei, Q. Leng, S. Han, A. K. Mok, W. Zhang, and M. Tomizuka, “RT-WiFi: Real-time high-speed communication protocol for wireless cyber-physical control applications,” in *2013 IEEE 34th Real-Time Systems Symposium*. IEEE, 2013, pp. 140–149.

[2] J. Xianjun, L. Wei, and M. Michael. (2019) open-source IEEE802.11/Wi-Fi baseband chip/FPGA design. [Online]. Available: https://github.com/open-sdr/openwifi

[3] M. Dillinger, K. Madani, and N. Alonistioti, *Software defined radio: Architectures, systems and functions*. John Wiley & Sons, 2005.

[4] Z. Yun, W. Peng, Z. Shengli, M. Aloysius, K., N. Mark, and S. Han, “RT-WiFi on Software-Defined Radio: Design and Implementation,” in *28th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2022.