Latency and Cost Requirements of Systems for Teaching MAC Protocols

Abhay Samant*, Venkataramana Badarla*, Sandeep Yadav*, Mythili Vutukuru†, Paul Khanna‡ and Erik Luther‡
*Indian Institute of Technology, Jodhpur
Email: abhay, ramana, sy@iitj.ac.in
†Department of Computer Science and Engineering, Indian Institute of Technology, Bombay,
Email: mythili@cse.iitb.ac.in
‡National Instruments, Austin, TX-78759
Email: paul.khanna, erik.luther@ni.com

***Abstract*—Over the past decade, the physical (PHY) layer of communication systems has evolved with the addition of techniques such as orthogonal frequency division multiplexing (OFDM) and multi-carrier aggregation. This has resulted in significant performance improvements, but it has come at the cost of increased power consumption and system complexity. To overcome this problem, a wide range of new Medium Access Control (MAC) protocols have been proposed for wireless networks. However, the speed of innovation in MAC protocols has not been able to keep up with the fast pace of PHY layer research; the latter being fueled by the availability of a variety of Software Defined Radio (SDR) platforms. These systems have eventually made their way into the classrooms and labs, thus giving communication engineers an experiential learning opportunity. They have provided students cost effective options to acquire real-world signals and analyze them using digital signal processing techniques. In essence, this has done for communications engineering students, what the sound card did for students learning audio signal processing. On the other hand, computer science students have been left with the option of learning about MAC protocols only through text books or by using software simulations. This is because most SDR systems do not meet the stringent latency and performance requirements required for creating real-world communication links; and the few that do are priced out of reach for classroom sizes typically found in Indian engineering colleges. In this paper, we analyze this situation at hand and discuss the emergence of a new design space for MAC layer prototyping systems. This paper discusses the key requirements, namely latency, processing speed, and cost, of systems in this design space. Finally, this paper describes how availability of commercial technology and careful trade-off with other requirements, such as throughput and frequency agility, is making it feasible to design a system that meets these key requirements.**

I. Introduction

The concept of Software Defined Radios (SDRs) was first introduced in a research paper by Joseph Mitola in 1991 [1]. In this paper, SDR is defined as a radio which can be reprogrammed in the field to instantly change its communication protocol to a different standard. This allows the radio to quickly adapt to changing requirements such as new channel conditions. While this concept has been around for more than 20 years, there has been a recent acceleration in the adoption of SDRs. This has been driven by advancements in software, processing technologies, and radio transceivers.

978-1-4799-6619-6/15/$31.00

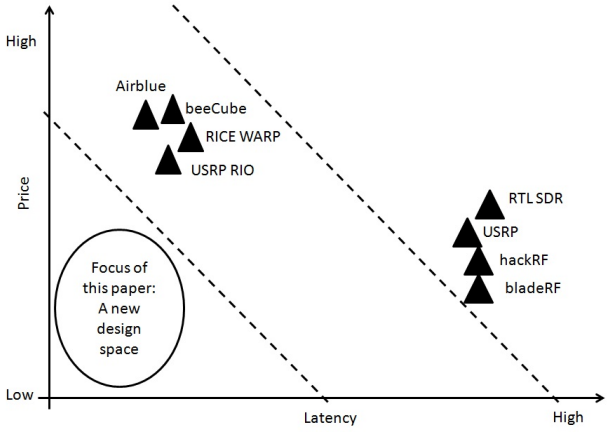

Fig. 1. Emergence of a new design space driven by low cost and low latency requirements.

It has resulted in a number of commercial SDR platforms such as the Ettus Research USRP [2], beeCube systems [3], RICE WARP [4], Airblue [5], and hobbyist platforms such as bladeRF [6] and hackRF [7]. These systems, and many others described in [8], have eventually made their way into classrooms and labs, thus giving students the opportunity to learn about PHY layer concepts with real-world signals. It has allowed students to rapidly prototype new algorithms such as carrier frequency offset estimation, synchronization, and multi-carrier aggregation, and validate these prototypes with real-world signals captured over the air. This has impacted the communication engineering curriculum in the same way that the sound cards have impacted the audio signal processing curriculum.

The key word in Mitola's definition of SDR, which is left open for creative interpretation is "instantly". All the systems described above meet the "instant reprogramming" requirement for physical layer algorithms. However, these platforms are not suitable for meeting the low latency and real-time performance needs required to ensure that radios can establish and sustain communication links. Figure 1 shows how the existing SDR systems map onto the product landscape defined by latency and price requirements. USRP and other host-based systems offer good choices for lower cost of development while trading off latency. On the other hand, RICE WARP

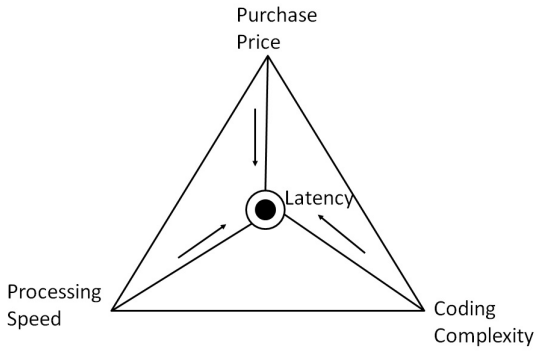

Fig. 2. Key Design Considerations for MAC Protocol Prototyping Systems

and other platforms offer choices for Field Programmable Gate Array (FPGA) implementation which enable low-latency applications, but it comes at the cost of increased coding complexity. This current situation has led to the emergence of a new design space as shown in Figure 1. The focus of this paper is to describe the requirements of this new space and explore the technology and business feasibility of designing systems that can play in this space. Rest of the paper is organized as follows. Section II outlines some of the key requirements for MAC layer protocols, centered around the ideas of latency, processing speed, and cost. Section III presents experimental results which prove why traditional SDR platforms are not suitable for meeting the stringent latency requirements of MAC protocols. Section IV discusses how advances in commercial off-the-shelf technology and the right design trade-offs can make the design of such low-cost latency sensitive systems a reality. Section V concludes the paper and describes scope for future work.

II. MAC Layer Requirements

A generalized framework for evaluating experimentation systems for cross layer design has been described in [9]. This framework contains six metrics, mainly cost, latency, throughput, hardware agility, software portability, and extensibility. Attempts to build a system that scores high on all of these metrics will result in a system that ends up being insufficient for any use-case. Hence, it is very important that we clearly understand the key care-abouts of the application at hand. With this objective in mind, this paper analyzes the requirements specifically for MAC layer prototyping systems. In this application space, we believe that the most important criteria is low latency, which in-turn is driven by processing speed, purchase price, and coding complexity, as shown in Figure 2.

A MAC layer prototyping system should have deterministic, also referred to as real-time, behavior. It should ideally be able to sustain latency requirements within few tens of microseconds, and in the best effort case, to within hundreds of microseconds. Most MAC protocols use techniques such as Code Division Multiple Access (CDMA), Time Division Multiple Access (TDMA), Carrier Sense Multiple Access (CSMA), and variants thereof [10]. These techniques are used in many commercial wireless networks such as Bluetooth, Zigbee and Wireless Local Area Network (WLAN). TMDA-based protocols specifically require precise scheduling of time to ensure that transmissions occur during the prescribed time slots. Modern contention-based protocols also require the ability to precisely control timing requirements to implement techniques such as back-off periods, contention windows, and inter-frame spacing. However, it is important to note that not all MAC protocols have low latency requirements. For example, the MAC protocols used in the licensed cellular bands assume that a particular time or frequency slot is available for a specific user and hence do not need extremely fast transmit-receive turnaround times. Research in this application space is mainly being driven by a handful of organizations who have the license to operate in these bands and can afford to use higher cost prototyping systems. Most of the contributions to the 3GPP standards, as an example, are being driven by these organizations. On the other hand, research in the unlicensed band is being driven by researchers in universities and entrepreneurs all over the world. Over the years, they have been making recommendations to the IEEE standards using pure theoretical and simulation-based findings. It is getting increasingly important that these recommendations are validated using real-world prototyping systems in order to increase the likelihood of being ratified by the standards bodies. This is driving the need for a low cost experimentation platform which can truly democratize the evolution of these standards.

Processing Speed: In a contention-based protocol, a back-off mechanism is used to reschedule the transmission assuming that the packet loss was due to collision. While the concept of backoff is related to precise scheduling, it also requires the ability to reschedule the transmission quickly without requiring a full packet transmission. A good experimentation system should support fast carrier sense techniques. A simple, yet elegant way, to detect the presence of a carrier is to measure the power of the received signal. More complicated techniques involve full demodulation to get to the decoded bits. The latter is more time-consuming and hence generally avoided. A related functionality requirement is the ability to do a fast recognition to detect the incoming packets. An experimentation system in the new design space should have processing capability which allows the MAC to quickly identify the incoming packet without decoding the entire packet. Such a system should also be able to generate dependent packets quickly and transmit them with precise timing relative to the previous packets. Examples of such packets are ACKs for error control and RTS/CTS for identifying channel access.

Purchase Price: In order to provide true hands-on experience, it is important that students have full access to an experimentation system. To make this financially viable for classroom sizes typically found in Indian engineering colleges, we believe that the cost of each lab station should be less than USD 100. If we were to translate all these requirements into a single metric such as the Bill of Materials (BoM), then our design goal should be to have a BoM of less than USD 75. To drive the manufacturing costs down, it is also important that the board schematic be open source so that it can be manufactured locally.

Coding Complexity Students would ideally prefer one language to write both MAC and PHY layer algorithms. Experimentation systems should have well-documented and well-supported build tools. Ideally a single programming language

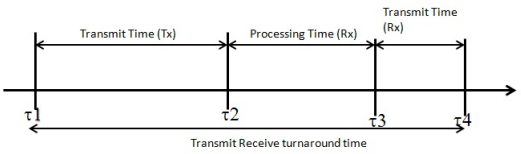

Fig. 3. Definition of precise time stamps required for MAC protocols

should be able to target different processing elements on the system. Platform should support a reconfigurable Zigbee-like stack which students can easily modify. Programming languages should be able to support seamless distribution of algorithms across compute nodes. Ideally, platform should support C/C++ or Phython programming since most computer science students learn this language. Reconfigurability of the radio front end is not as important and can be traded off to make programming easier. One of the benefits of host-based programming is that the operating system has sufficiently matured to give elements such as multi-threading and application development, at zero cost. We lose some of these benefits as we move to a FPGA-based system. Some of this risk can be mitigated by providing a higher layer API, which can be provided as open source code, along with various lab modules and algorithm building blocks.

To summarize, in this section, we have described the key requirements of a MAC layer prototyping system. In Section III, we will discuss some of the challenges of meeting these requirements with the current SDR systems.

III. Challenges with SDR for MAC Prototyping

New physical layer algorithms and their experimental results using SDR platforms have been widely published in the literature [14], [15], [16], [17]. Availability of ample processing power on general purpose processors and lesser stringent requirements for latency are two of the key factors driving this proliferation. Likewise, the use of open source software, such as GNU radio by USRP, cannot be emphasized enough in driving this rapid adoption. As described earlier, these systems have also made their way into the classrooms and labs, where students have the opportunity to learn about PHY layer techniques with real-world signals. However, it has been a challenge to replicate this success for teaching MAC layer protocols as current SDR systems do not meet many of the key requirements described in Section II. For example, in order to support the desired processing speeds, MAC protocols typically require computational partitioning of data and control blocks among parallel heterogeneous computational engines. This can be a difficult task on traditional SDR systems. Likewise, these systems support a buffered model of computation, in which a block of data is transferred from the acquisition subsystem to the processing subsystem. Each of these steps require a distinct amount of time as shown in Figure 3. The sum of these times is generally referred to as transmit-receive turnaround time. Note that, the receive time and transmit time are a function of the size of the packets; which are often asymmetric across transmit and receive functions.

Timing control refers to the ability to prescribe and measure the amount of time between events of interest. Precise timing control is important in digital communication systems

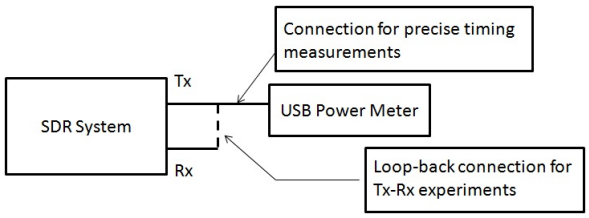

Fig. 4. Novel experimentation system using power meter for fast power measurements

because it can affect the ability to maintain communication links. Such applications often require response-time guarantees, which are measured as the time between the receipt of packet by the system and when the system responds with a message. Figure 3 shows these different time stamps. Let $\tau 1$ denote the time instant when the first sample is sent from the transmitter to the receiver. Let $\tau 2$ denote the time instant when the n^{th} sample is transmitted, where n denotes the block size. Let $\tau 3$ denote the time instant when the receiver finishes processing the incoming packet, which includes decoding the received packet and creating a response packet, such as an ACK. At this time instant, the receiver is ready to transmit the first sample. Let $\tau 4$ denote the time instant when the n^{th} sample is finished transmitting. Let τ_t denote the transmit time at the transmitter, τ_p denote the process time, and τ_r denote the transmit time at the receiver, as defined by the following simple equations.

$$\tau_t = |\tau 2 - \tau 1| \tag{1}$$

$$\tau_p = |\tau 3 - \tau 2| \tag{2}$$

$$\tau_r = |\tau 4 - \tau 3| \tag{3}$$

When we talk about latency requirements and precise time scheduling, we are referring to a tight control over these measurements. State of the art literature currently lacks a well-defined test procedure to accurately measure these times. We overcome this problem by describing a test procedure that allows us to make precise measurements of the three quantities described in the above equations. Measurements of τ_p have been done in [10] purely from the software perspective, using ping commands. However, these measurements do not take the overall system effects of both hardware and software into account. Additionally, the process described in [10] does not allow separation of the overall time into the three components as shown in Figure 3. The method described in this paper is fast and is able to make time measurements at finer granularity.

Figure 4 shows our proposed experimentation system which uses a power meter to measure power at the RF transmit (Tx) and receive (Rx) output ports of the SDR hardware. Power meters are capable of doing fast and accurate measurements of power by allowing users to trade off measurement accuracy over time by carefully controlling the number of averages in the measurement.

Figure 5 shows the measurements recorded by the power at the RF output ports. Region A indicates the time period when the transmitter is sending a message and Region B indicates the time period when the receiver is sending an acknowledgement. Our method introduces a GPS time-stamped marker in the

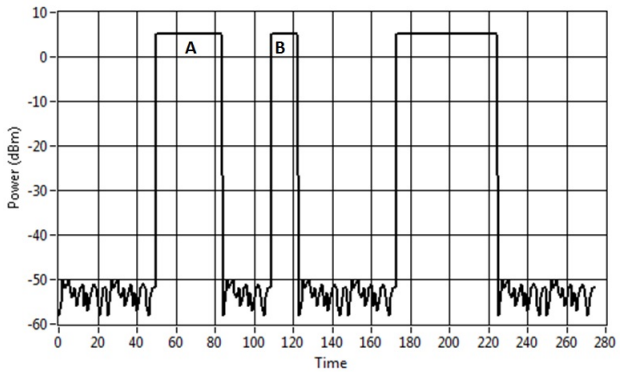

Fig. 5. Precise timing achieved using power meters

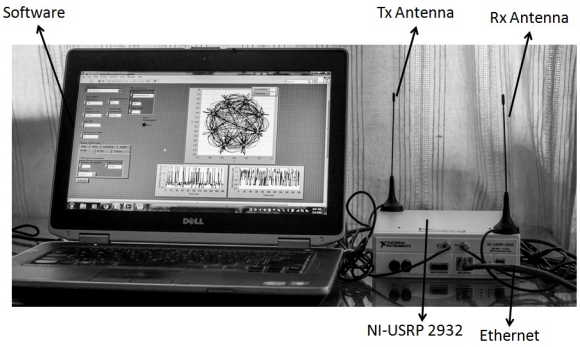

Fig. 6. Experimentation system used to validate new test procedure using power meters

first sample of the transmitted packet. We then measure the time instant when the sample arrives at the RF port, indicated by a rise in the measured power level, and the time instant after the nth sample arrives at the RF port, indicated by a drop in the measured power. Our experiments show that the power measured by the meter settled to a value within +-1dB in less than 80 microseconds. Using this test procedure, we can accurately measure both τ_t and τ_r by minimizing any uncertainty in the measurement.

To prove the validity of our test procedure, we have setup an experimentation system as shown in Figure 6, which contains a laptop connected through an Ethernet cable to National Instruments Universal Software Radio Peripheral (NI-USRP 2932). NI-USRP 2932, which serves as a traditional SDR system for our experiment, contains a radio front end which can be tuned to frequencies from 400 MHz to 4,400 MHz. It offers support for 20 MHz baseband I/Q bandwidth and support for streaming data at a rate of 25 MS/s. The key parameters for this custom scheme are listed in Table I.

TABLE I. KEY PARAMETERS OF THE EXPERIMENTATION SYSTEM

Parameter	Value
Center Frequency	2400 MHz
Modulation Scheme	M-ary Phase Shift Keying (PSK)
MAC Protocol	TDMA
Symbol Rate	1 MS/s

Table II shows the results of measuring τ_t as a function of packet size, averaged over 1000 iterations. τ_t is the time taken by the transmitter to send the packet. This data shows two characteristics. First of all, the latency is pretty high with traditional SDR architecture. Secondly, it is also interesting to note the high standard deviation and how this value scales as a function of the packet size.

TABLE II. TRANSMITTER LATENCY VS PACKET SIZE

Packet Size	Average Latency (ms)	Standard Deviation
100	10.3	0.23
200	13.0	0.34
500	17.4	0.45
1000	19.8	0.65
2000	20.1	0.8
5000	23.5	0.9

The second set of experimentation shows measurement results of τ_r as a function of packet size, averaged over 100 iterations. This is the time taken by the receiver to transmit the created packet. Once again, we notice a wide statistical distribution in the time measurements. It is also important to note that, for same packet sizes, the transmit time and receive times are not equal, which shows that the principle of reciprocity should not be assumed for transmit and receive channels.

TABLE III. RECEIVER LATENCY VS PACKET SIZE

Packet Size	Average Latency (ms)	Standard Deviation
100	11.0	0.21
200	12.3	0.31
500	16.8	0.425
1000	17.1	0.65
2000	18.7	0.78
5000	20.1	0.8

IV. DESIGN FEASIBILITY

In this section, we discuss the feasibility aspects of designing a system that meets the key requirements discussed in Section II. These key requirements are latency and cost. In order to design a system that focuses on one of these key requirements, it is important that we have the flexibility to trade-off on other dimensions. Since the focus of this paper is on a teaching platform for MAC protocols, we have given lower priority to requirements such as throughput and hardware agility. Our market research shows that commercial technology has and will continue to evolve so that FPGA and real-time processing units can be put on a single silicon fabric. This makes it feasible to build systems which students can use to implement and experiment with latency optimized algorithms. Likewise, state of the art allows ADC/DAC techniques to be merged with RF front ends on a single chip, making it feasible to lower the cost of the system. As part of our on-going research work, we plan to provide open source MAC/PHY algorithms which can be run on the FPGA and real-time processors. We believe that this will significantly reduce the cost of ownership, allowing students to leverage the community ecosystem, similar to what the GNU Radio has done for USRP. Thus, by using commercially available technology, trading throughput, and by building open source PHY/MAC FPGA IP, it is possible to build latency and cost optimized systems which can be used for teaching. In the rest of this section, we will provide a very high level view and feasibility analysis of our proposed new design.

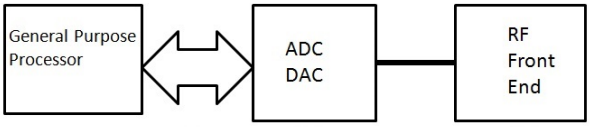

Fig. 7. Architecture of current PC-based SDR platforms

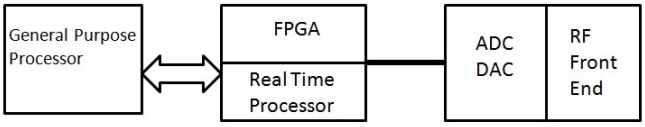

Fig. 8. New design capable of meeting latency and cost requirements for MAC layer protocols

TABLE IV. COST ANALYSIS OF PROPOSED DESIGN

Block	Traditional SDR	Proposed Design
General Purpose Processor	0	0
Bus	$	0
FPGA	0	$$
ADC	$$	0
RF	$$	$

An architectural diagram of a traditional, PC-based SDR platform is shown in Figure 7. The block diagram of our proposed design is shown in Figure 8. Table IV compares the relative normalized costs of these two systems along various vectors. SDR systems typically feature a general purpose processor, typically a desktop or a laptop computer, as the processing block. We assume that that most students will have access to a laptop or desktop and hence we set the cost of this component to zero. Traditional SDR systems generally use a higher bandwidth bus such as, GigE and PCIExpress, between the processor and the analog to digital converters, as all the data processing is done on the general purpose processor. We set the normalized cost of this block to 1 unit (or a single $ sign). On the other hand, our proposed design uses a USB bus technology, which has become a commodity using current technology. Hence, we set the normalized cost of this block to zero.

Next block in a traditional SDR is the analog to digital converter (ADC). Traditional SDRs have generally used higher resolution on their ADCs. In industrial data-acquisition projects, designers need to digitize an input signal that extends over a very wide dynamic range. Other examples where a wide dynamic range is needed is when one needs to accommodate signals from different sources that exhibit quite different signal ranges or to resolve small changes around a certain value. Usage of higher resolution ADC results in a higher dynamic range in the resulting signal, but it comes at the expense of higher system cost, as shown in Table IV. We have set the normalized cost of ADc to zero for our proposed design as it uses integrated ADC and RF subsystem, as discussed next.

The RF portions used in traditional SDR systems are largely comprised of direct up/direct down topology also known as homodyne. They are generally designed to cover a wide frequency range from DC to 6GHz or higher, support a wide instantaneous bandwidth and have good spurious free dynamic range. While traditional SDRs can boast of an RF front end which is best in its class, it has driven the cost of the hardware higher and thus out of reach of a wide population

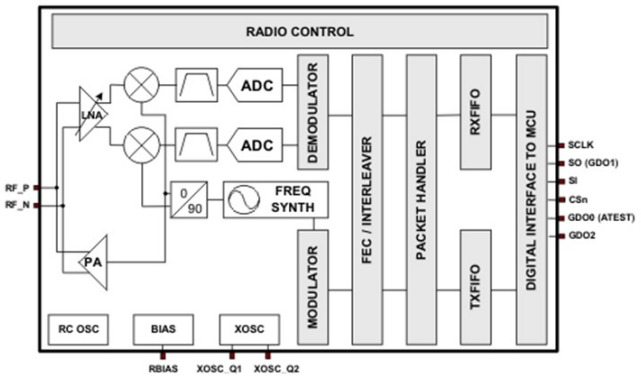

Fig. 9. Texas Instruments RF Transceiver chip

of teachers and researchers. Hence, we assign the normalized cost of this component to 2 units. Our proposal is to combine the ADC and RF capabilities into a single transceiver chip which are commercially available in the market place today. One example of such a chip is the AD9361 which is a high performance and highly integrated RF agile transceiver [11]. This device combines an RF front end with a flexible mixed-signal baseband section and integrated frequency synthesizers. Since this device incorporates both the RF front end and the ADC/DACs, its cost relative to the cost in traditional systems is low.

Another example is the Texas Instruments TI CC2500, which is a low-cost 2.4 GHz transceiver designed for very low-power wireless applications. It is available in volume for prices as low as $10 [13]. Due to the availability of such commercial RF transceivers, we have assigned a normalized cost of 1 unit to this component while performing cost analysis of our proposed design.

One of the key requirements listed in this paper is latency. Latency can be defined as the delay between decoding a particular packet and the generation of a message in response to the packet. FPGAs are generally suited to meet the latency requirements of MAC protocols as they contain specialized circuitry that can perform multiple, sequential, and parallel operations within a single clock cycle. FPGAs are also made up of different types of resources, such as logic, signal processing, and memory blocks, which can be used programmed to extract optimal performance. Hence, we recommend the usage of FPGAs for our proposed design. One of the challenges with FPGA is its price and coding complexity. Hence, we have given this component a normalized cost of 2 units in Table IV. GNU radio provided an ecosystem of open source library for traditional SDR systems which has significantly driven the costs associated with coding complexity to zero. As part of our research, we plan to create a GNU-Radio like environment for our proposed system to help overcome the coding challenges. We plan to publish PHY/MAC layer algorithms that students can use as a starting point to conduct their experiments. Once implemented, this initiative would significantly offset the increased cost of ownership introduced by the use of FPGAs on our proposed design and encourage it's adoption by labs all over the country.

V. Conclusion and Future Work

PC-based SDR systems have made their way into the classrooms and labs, thus providing PHY layer engineers with an experiential learning environment. On the other hand, students are still learning about MAC layer protocols in theory or with the aid of software simulations. In this paper, we analyze this current situation and discuss the emergence of a new design space for MAC layer prototyping systems. This paper discusses the key requirements of such systems, namely latency, processing speed, and cost. Finally, this paper explains how availability of commercial technology and careful trade-off with other requirements, such as throughput and frequency agility, is making it feasible to build a system that meets our design objectives.

MAC for multi-user MIMO, meeting latency requirements for deterministic control applications, dealing with increasingly wide bandwidth channels, and hybrid communications that can hand-off between WLAN to 5G standards are some of the many aspects of MAC protocols that require active research. Solving these complex challenges will require a future generation of computer scientists who fully understand the core fundamentals of their work. The purpose of the system proposed in this paper is to achieve this goal. Our scope of future work includes prototyping the system described in this paper, validating its performance with real-world signals, and exploring creative ways to further lower the cost of the system. Scope of future work also includes creating an open source software library of MAC layer protocols and teaching aids which will help lower the cost of ownership and drive widespread adoption of this system.

Acknowledgment

The authors would like to thank Mr. Balint Sabeer and Mr. Matt Ettus from Ettus Research for providing their insights into the feasibility analysis of proposed system design.

References

[1] Mitola, Joe. "The software radio architecture." Communications Magazine, IEEE 33, no. 5 (1995): 26-38.

[2] Ettus, Matt. "Universal software radio peripheral (USRP)." Ettus Research LLC http://www. ettus. com (2008).

[3] Rothman, Joseph, and Chen Chang. "BEE technology overview." In Embedded Computer Systems (SAMOS), 2012 International Conference on, pp. 277-277. IEEE, 2012.

[4] Hunter, Chris, Joseph Camp, Patrick Murphy, Ashutosh Sabharwal, and Chris Dick. "A flexible framework for wireless medium access protocols." In Signals, Systems and Computers, 2006. ACSSC'06. Fortieth Asilomar Conference on, pp. 2046-2050. IEEE, 2006.

[5] Ng, Man Cheuk, Kermin Elliott Fleming, Mythili Vutukuru, Samuel Gross, and Hari Balakrishnan. "Airblue: A system for cross-layer wireless protocol development." In Proceedings of the 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, p. 4. ACM, 2010.

[6] https://www.kickstarter.com/projects/1085541682/bladerf-usb-30-software-defined-radio

[7] http://ossmann.blogspot.com.au/2012/06/introducing-hackrf.html

[8] Cass, Stephen. "Tools and Toys: Hardware for your Software Radio." Spectrum, IEEE 43, no. 10 (2006): 51-54.

[9] Abhay Samant; Sandeep Yadav; V. Badarla; M. Vutukuru, "A Framework for Comparing Experimentation Systems for Cross Layer Design", submitted to EURASIP Journal on Wireless Communications and Networking, May 2014.

[10] Nychis, George, Thibaud Hottelier, Zhuocheng Yang, Srinivasan Seshan, and Peter Steenkiste. "Enabling MAC Protocol Implementations on Software-Defined Radios." In NSDI, vol. 9, pp. 91-105. 2009.

[11] http://www.analog.com/en/rfif-components/rfif-transceivers/ad9361/products/product.html

[12] http://www.xilinx.com/products/silicon-devices/soc/zynq-7000/

[13] http://www.ti.com/product/cc2500

[14] Hunter, Chris, Joseph Camp, Patrick Murphy, Ashutosh Sabharwal, and Chris Dick. "A flexible framework for wireless medium access protocols." In Signals, Systems and Computers, 2006. ACSSC'06. Fortieth Asilomar Conference on, pp. 2046-2050. IEEE, 2006.

[15] Shepard, Clayton, Hang Yu, and Lin Zhong. "ArgosV2: a flexible many-antenna research platform." In Proceedings of the 19th annual international conference on Mobile computing and networking, pp. 163-166. ACM, 2013.

[16] Everett, Sahai, Sabharwal, On the Impact of Phase Noise on Active Cancellation in Wireless Full-Duplex, Submitted to IEEE Transactions on Wireless Communication, January 2013

[17] Sahai, Achaleshwar, Gaurav Patel, Chris Dick, and Ashutosh Sabharwal. "Understanding the impact of phase noise on active cancellation in wireless full-duplex." In Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on, pp. 29-33. IEEE, 2012.