2022 International Conference on Unmanned Aircraft Systems (ICUAS)
Dubrovnik, Croatia. June 21-24, 2022

2022 International Conference on Unmanned Aircraft Systems (ICUAS) | 978-1-6654-0593-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICUAS54217.2022.9836117

An SDR-based LTE System for Unmanned Aerial Systems

Jinran Zhang*, Baoqian Wang†, Dong Wang‡, Shengli Fu*, Kejie Lu‡, Yan Wan§, and Junfei Xie†
*Department of Electrical Engineering, University of North Texas
†Department of Electrical and Computer Engineering, San Diego State University
‡Department of Computer Science and Engineering, University of Puerto Rico at Mayagüez
§Department of Electrical Engineering, University of Texas at Arlington

***Abstract*—In the communication community, software-defined radio (SDR) has been widely adopted for its flexibility in different communication technologies. In this paper, we explore the application of SDR over unmanned aerial systems (UAS). More specifically, we develop an SDR-based long-term evolution (LTE) communication system with features of compact size, low power consumption, and high computing capability to enable its deployment over UAS. The system is implemented by integrating (1) Jetson TX2, which is a classical high-performance embedded computing module, (2) LimeSDR, which is a light-weighted SDR hardware module, and (3) srsLTE, which is an open-source software module that provides LTE implementations. Using this prototype, we conduct comprehensive experiments to evaluate different aspects of the system performance, such as Round Trip Time, Throughput, Signal to Noise Ratio, Computing Resource Consumption, etc. Moreover, we compare the performance of the LTE system built on Jetson TX2 with another LTE system built on a personal computer with a general-purpose processor (GPP). Our study provides important observation and design guidelines for future wireless communication systems that can be carried by UAS.**

***Index Terms*—SDR, LTE, Embedded Computing Platform, Communication Performance, Resource Consumption.**

I. Introduction

For unmanned aerial systems (UAS), a major challenge is how to provide robust and high data rate communication to enable collaborations among the UAS [1]. While traditional communication techniques such as Wi-Fi, Bluetooth, and the cellular system could be used by UAS, it is important to evaluate their performance over UAS and tailor the communication system design based on the characteristics of UAS.

Due to the environmental constraints on UAS, such as limited space, payload, and power supply, communication systems on UAS have to consider extra airborne requirements beyond the fundamental radio features. To evaluate different wireless communication technologies in UAS, a viable solution is to use software-defined radio (SDR) because of its flexibility and programmability over the traditional embedded systems. Essentially, the design of an SDR-based system is based on the concept of using software to process signals in radio technology instead of using a purely hardware-based method [2]. Consequently, many SDR-based prototypes have been developed in recent years, such as Wi-Fi [3], satellite [4], Bluetooth [5], etc. Because of these advantages, in this paper, we explore the application of SDR over UAS. More specifically, we develop an SDR-based long-term evolution (LTE) communication system with features of compact size, low power consumption, and high computing capability to enable its deployment over UAS.

Despite the advantages of SDR, using software to process signals implies that the computing hardware must have sufficient computing capability. Due to the compatibility issues and performance concerns, most existing SDR-based prototypes use regular computers with *general-purpose processors* (GPPs) as the computing platform. However, the GPP-based solutions may not be suitable for UAS (e.g., size and power).

In this paper, we aim to tackle the aforementioned challenges for applications over UAS. Firstly, we implement the system over GPP. Then, we migrate the open-source platform onto the UAS using a high-performance mobile computing platform: Jetson TX2 [6]. Specifically, we conduct development and experiments to explore the communication capability of an SDR-based LTE prototype that utilizes Jetson TX2 as the computing platform. This methodology qualifies Jetson TX2 to be a viable platform to support an important wireless communication standard. We then investigate application performance on the networked Jetson TX2 while the system is in a static indoor environment to avoid dynamic interference from mobility and external forces. The deployment confirms the feasibility of developing an LTE system using Jetson TX2 but also discovers important issues that are still challenging to address. Finally, we explore the feasibility of deploying the LTE system on the UAS.

The main contributions of this paper include the follows.

a) *Implementing an LTE cellular network*: We implement an LTE system, in which we use Jetson TX2 and SDR board to set up a base station on a UAS and we test the connectivity using both commercial smartphones and SDR-based user devices. The communication in the network conforms to the 4G LTE standard.
b) *Investigating interference in LTE cellular network*: In the experimental LTE system, we investigate varied performance metrics under different end-to-end distances (illustrated in Section IV(B–D)).
c) *Comparing communication performance between GPP*

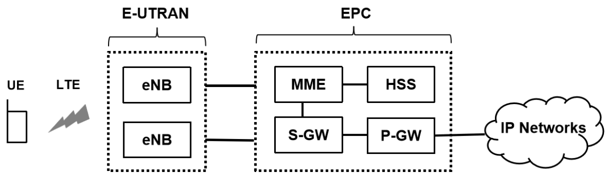


Fig. 1. An overview for LTE systems that shows main components related to this paper.

and Jetson TX2: We compare the communication performance on two types of platforms (illustrated in Section IV(B–D)).

d) *Investigating the computing resource consumption*: When we run applications on a machine, there can be varied resource consumptions based on our formulated execution, which indicates that applications could promisingly achieve optimal performance based on the regulated computing resources (illustrated in Section IV(E)).

The rest of this paper is organized as follows. In Section II, we first discuss the background and architecture of LTE. Then, in Section III, we elaborate on the hardware and software aspects of the SDR-based LTE implementation over Jetson TX2. Finally, in Section IV, we present comprehensive experimental results, before concluding the paper in Section VI.

II. Background of The LTE System

LTE is a wireless communication standard designed by the Third Generation Partnership Project (3GPP) [7]. The specifications and foundation features for LTE are in release 8 of 3GPP. It was completed in December 2008 and this is the basis for the following specifications of LTE. LTE supports a wide range of frequency bands from 700 MHz up to 2.7GHz as well as bandwidths from 1.4 MHz up to 20 MHz [7]. It supports multiple services such as mobile broadband (MBB), Internet of Things (IoT), and vehicle services.

As shown in Fig. 1, the major components of LTE include User Equipment (UE), Evolved Universal Terrestrial Access Network (E-UTRAN), Evolved Packet Core (EPC), and IP Networks. The core network consists of many logical nodes (i.e., EPCs) where each EPC is composed of multiple network nodes, including Mobility Management Entity (MME), Home Subscriber Server (HSS), Serving Gateway (SGW), and Packet Data Network (PDN) Gateway (P-GW). The EPC is responsible for authenticating subscribers and helping locate the UE. The access network E-UTRAN consists of multiple Evolved NodeBs (eNB) that connect to the UEs through wireless channels. The basic functionalities of an eNB include radio access and session management.

The LTE system has achieved tremendous success, especially with applications through smartphones. It has also attracted many research interests to study the performance of LTE through SDR-based testbeds. In [8], an open-source platform for LTE experimentation is introduced to support various tests and experiments. The open platform includes both LTE libraray (srsLTE) and LTE UE (srsUE) with full compliance of LTE Relase 8. In [9], an SDR physical layer extension was developed for the ns-3 simulator such that researchers could change the transmission parameters to implement the network function virtulization. In [10], the authors studied the LTE and license-assisted access (LAA) coexistence systems, and they developed an SDR-based testbed to validate performance such as throughput.

III. The LTE System Design

In this section, we describe the proposed SDR-based LTE system (E-UTRAN and EPC in Fig. 1) over high-performance embedded computing platform, which facilitates the deployment over UAS in terms of system features, weight, dimension, and power supply.

A. SDR Hardware

The concept of SDR is introduced to offer the opportunity for performing software stack on re-configurable boards to prototype wireless communication [2]. SDR provides solutions to radio spectrum solely based on customized software rather than specialized hardware. The typical SDR platform consists of two parts: the programmable RF front-end and application-oriented software. This technology has a lot of benefits with respect to its flexibility, cost effectiveness, easy replication and life cycle reduction. For example, SDR platform enables one RF front-end to meet lots of standards like FM, LTE [11], [12]. We can reuse software modules for various frequency and modulation techniques according to the different design requirements.

Table I compares three SDR hardware platforms that have been used widely. The trade-offs we should keep in mind while choosing the SDR platform include the cost, frequency, bandwidth, and dimension. The goal of our design is to develop a compact SDR system that can be deployed over autonomous systems like UAS which may work as edge computing nodes. We choose LimeSDR USB as it provides a good trade-off between the performance and physical parameters. In addition to the compact size (100mmx60mm with weight of 118 gram), LimeSDR USB is equipped with an Altera Cyclone IV FPGA with high computing power. It covers the frequency range between 100 kHz and 3.8 GHz with bandwidth of 61.44 MHz, which satisfy most communication requirements.

B. Mobile Computing Platform: Jetson TX2

Jetson TX2 is a computing module that adopts the NVIDIA Pascal architecture. The advantages of Jetson TX2, such as energy-saving, compact size and powerful computing capability, have resulted in a dramatic migration of applications onto this platform. With a size of a credit card, the power

TABLE I
COMPARSION OF SDR PLATFORMS

	USRP B210	LimeSDR Mini	LimeSDR USB
Frequency	70 MHz – 6 GHz	100 kHz – 3.8 GHz	100 kHz – 3.8 GHz
Bandwidth	56 MHz	30.72 MHz	61.44 MHz
Cost	$1,472.00	$186.95	$349.95
Weight	350 g	20 g	118 g
Dimension	97mm x 15.5mm	69mm x 31.4mm	100mm x 60mm
Power	USB 3.0, DC 6V	USB 3.0, DC 6V	USB 3.0, DC 6V

TABLE II
COMPUTING HARDWARE CHARACTERS

	Laptop	Jetson TX2
CPU	Intel® Core™ i7-6700HQ CPU @ 2.60GHz × 8	Dual-Core NVIDIA Denver 2 64-Bit CPU Quad-Core ARM® Cortex®-A57 MPCore
GPU	N/A	256-core NVIDIA Pascal™ GPU architecture with 256 NVIDIA CUDA cores
Memory	32GB	8GB
OS	Ubuntu 18.04 LTS	Ubuntu 18.04 LTS

consumption of Jetson TX2, which is around 7.5 watts, is far less than that required by traditional GPP and old NVIDIA Jetson modules.

In our LTE system, Jetson TX2 is chosen as the computing platform according to our previous studies [6]. eNB and EPC work as an edge computing server running on Jetson TX2. To demonstrate the performance of Jetson TX2, a Dell Laptop based LTE system is also evaluated in the experiments for comparison purpose. The characteristics of these two computing platforms are summarized in Table II.

We use an Apple iPhone 7, standard Edition, as the User Equipment (UE in Fig. 1). A programmable SIM Card from the Open Cell project [13] is inserted into the UE to help access the network,which can be configured with IMSI and authentication information. The resulting UE is able to connect to the base station where we simulate a commercial scenario.

C. System Software

The software to enable SDR over Jetson TX2 include: system kernel, LimeSuite (version 20.01) for SDR driver, SoapySDR (version 0.7.1), and srsLTE (version release 19.12).

System Kernel: During our research, we find that Stream Control Transmission Protocol (SCTP), a protocol used between eNB and EPC, is not enabled in the default Jetson TX2 kernel. For the purpose of our experiment, we have to compile the kernel from the source code to support the essential SCTP. We choose to locally build kernel based on the downloaded source from the website. To generate the Linux kernel for SCTP on Jetson TX2, we need to edit the configuration files for two purposes. The first one is to define a local version for the kernel which is the identity (ID) for all the modules in it. The second one is to turn on the SCTP option to enable the protocol at run time. Based on the configuration, modules are built and output into a specified directory. After these edits, the new kernel file is copied to the boot directory with an optional kernel added into the boot configuration. The new kernel with LTE capacity is then available after reboot of the system.

SDR driver: In our research, LimeSDR USB board works as the RF front-end for a radio frequency transceiver. Except for the physical board, the driver for LimeSDR USB named LimeSuite should be installed. We install LimeSuite by locally compiling from the source.

Middleware: In addtion to the LTE software, SoapySDR is also needed. In the microsystem of SDRs, hardware and driver are provided by the manufactures. The upper-level software cannot use lower-level drivers directly because of its program language and professional interface. SoapySDR is a type of wrapper which can transfer interfaces from C language to Python language and encapsulate the lower-level interfaces into APIs to support the upper-level software. Then, the upper-level software can configure and monitor the SDR device regardless of the environments it runs in and their interfaces. In our experiments, we install SoapySDR by locally compiling from source.

Software for eNB and EPC: In this study, we select the srsLTE project as eNB and EPC based on our investigation of several types of open-source LTE platforms. Considering its bandwidth compatibility, flexible interconnection to core network (IP Networks in Fig. 1) and system stability [14], srsLTE is suitable for our study. In our experiments, we install srsLTE by locally compiling from source. The design of srsLTE project can be flexibly exploited. eNB and EPC can run on a single machine or two separated machines. In this paper, we run the two components on a single machine.

IV. EXPERIMENTAL SETUP AND RESULTS

A. Experimental Setup

1) Frequency and Bandwidth: In our experiments, we select Band 3 for radio frequency which is not used in local commercial LTE networks. The unique frequency will avoid interference from commercial networks in the test area. Since many existing LTE networks use frequency division duplexing (FDD), we implement the FDD and we set 1787.4MHz as the uplink carrier and 1878.4MHz as the downlink carrier. Finally, we use 3MHz bandwidth for our cellular network.

2) SDR Configuration: LimeSDR supports both single antenna and multiple antennas for a channel. As the first step for our research, we use a single antenna on LimeSDR.

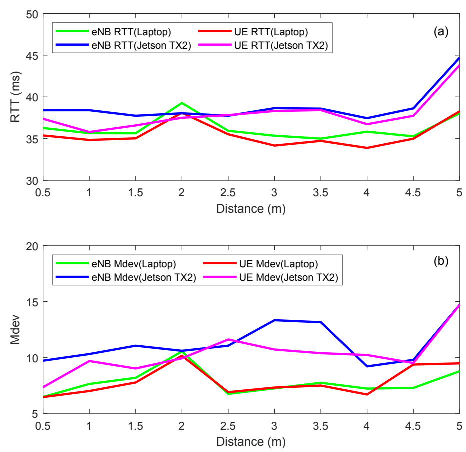


Fig. 2. Round Trip Time (a) and Mean Deviation (b) on two computing platforms (Jetson TX2, Laptop) versus distances between UE and eNB, EPC.

According to the previous band selection, we use the RX1-L port on the SDR board as the receive channel, and we use the TX1-1 port as the transmission channel.

3) Environments: Since our purpose is to emulate an LTE network in practical environments, we perform the tests with different distances between a UE device and an eNB. In this manner, we can analyze performance metrics versus the distances. We can also measure the computing resources used by applications on two platforms at run time, which can illustrate the impact of Jetson TX2 on the software performance.

B. End-to-end delay

We begin our study by evaluating the end-to-end delay. In communication systems, packet latency is a key performance metric regularly measured by practitioners, researchers, and vendors. In our tests, UE will search wireless signals when it is turned on or returns to the wireless environment from the airplane mode. When a UE attaches to the network created by our prototype, it must get an IP address (IPv4 or IPv6 address). In our system, we set the EPC to the IPv4 mode allocating IPv4 address to the UE.

To measure the latency versus distance, we execute the ping command on UE and EPC separately to get Round Trip Time (RTT) when the connection is established. The packets are sent once every second for a total of 100 seconds at each location. We repeat this experiment five times at a location and use the average delay as the experimental data, which is shown in Fig. 2. Then, we repeat the measurement at different locations with distances varying from 0.5 meters to 5 meters.

Since we implement two LTE systems, one with Jetson TX2 and another with a laptop computer, there are a total of four curves in Fig. 2. From Fig. 2 (a), we can observe that the latency increases slightly with the increase of distance for both implementations. The results also show that the latency of the LTE system with Jetson TX2 is greater than that of the LTE system with the laptop. A similar trend can also be observed when we investigate the mean deviation of delay in Fig. 2 (b), which is a benchmark for connection stability.

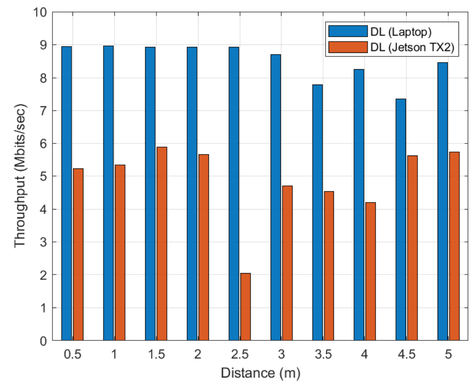


Fig. 3. Throughput on two computing platforms (Jetson TX2 and laptop) versus distances between UE and eNB, EPC.

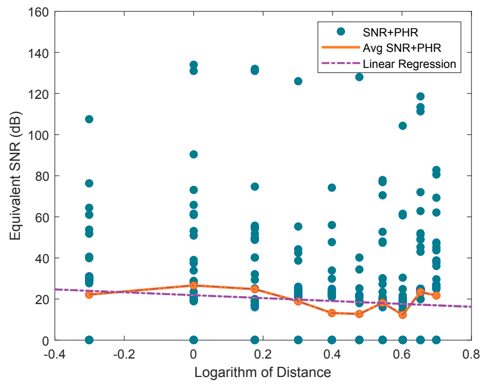
C. Throughput

Next, we study the throughput of the cellular network by sending User Datagram Protocol (UDP) packets between UE and EPC. Iperf is the tool we use to generate UDP packets transmitted from client to server. To evaluate downlink performance, UE is set up as a server, and EPC is set up as a client. We measure the average rate by sending a specified amount of UDP packets every second for 15 seconds in a test. Then the throughput performance is evaluated by averaging the measurements of three tests at a location.

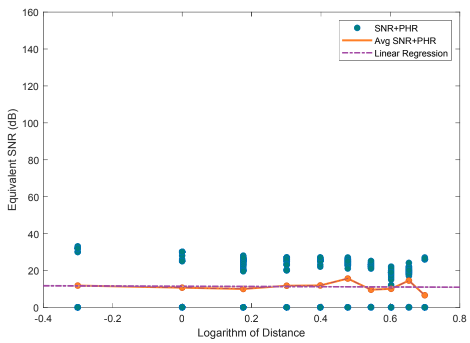
Fig. 3 illustrates the downlink throughput performance for two systems. This figure clearly demonstrates that the LTE system using Jetson TX2 achieves a smaller throughput, about two-thirds of the throughput achieved by the LTE system using laptop PC, most likely due to the lower computing capability. Another remarkable result from the figure is that, when the distance increases from 0.5 meters to 5 meters, the throughput in both systems drop only sightly.

D. Signal to Noise Ratio

The process of measuring Signal to Noise Ratio (SNR) is very similar to the process of measuring throughput. Specifically, we use iperf on eNB to generate UDP packets. When a large number of UDP packets are sent from eNB to the UE, we open the monitor on eNB to inspect the uplink and downlink parameters such as uplink SNR, uplink Power Headroom Report (PHR). In an LTE cellular network, UE is required to adapt to the complex network, report its situations as well as maintain its performance, where it balances the resource consumption for applications. PHR is an indicator for the extra power of a UE apart from the power spent on the current transmission. PHR implies that UE has extra



(a) SNR over Jetson TX2



(b) SNR over Laptop

Fig. 4. Signal to Noise Ratio (SNR) on two computing platforms ((a) Jetson TX2 and (b) Laptop) versus distances, including SNR + PHR at a fixed location, Average SNR + PHR vs. Logarithm of Distance and Linear Regression Relation Between Average SNR + PHR and logarithm of Distance.

power that can be used if needed. We propose a parameter named equivalent SNR to represent SNR by adding PHR to SNR. The SNR mentioned below refers to equivalent SNR. Through the srsLTE software, we can get these parameters in the resolution of a second. The process of inspecting SNR will last 15 seconds each time. For a given distance, we collect the measurement three times and get the average SNR.

Fig. 4(a) summarizes the experimental results of SNR over Jetson TX2. From the SNR plots, we can observe the SNRs of the LTE network tend to display small-scale fading at a fixed distance. In addition to the average SNR vs. distance, the curve varies significantly when the distance increases from 0.5 meters to 5 meters. Also, the trend of SNR does not demonstrate a significant decrease with the increase of distance. One possible reason is that the experiments are conducted in an indoor environment with heavy multiple-path effects. To further investigate the SNR performance, we may need a signal amplifier to increase the transmission range (far more than 5 meters of current setting) and conduct the experiments in an open space with less multi-path impacts. In Fig. 4(a), we also obtain the Linear Regression Relation

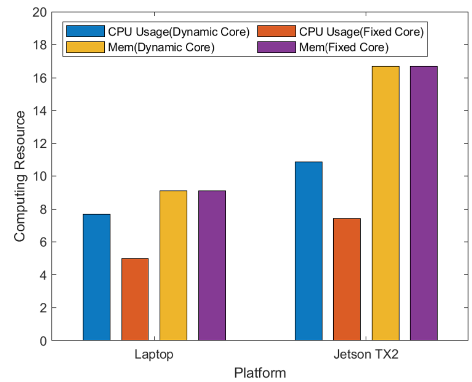


Fig. 5. Computing Resources Consumption on two computing platforms (Jetson TX2 and Laptop).

between SNR and logarithm of Distance. The curve shows a small decrease on SNR (less then 10dB) when the distance increases from 0.5 meters to 5 meters. Fig. 4(b) shows similar results for the LTE system with laptop. In this figure, we can observe that the SNR fluctuation is much smaller at each location, representing a more stable SNR performance. Also, the linear regression of SNR seem to be the same at different locations. The results in Fig. 4 (a) and (b) may be due to a variety of aspects, including the LTE protocol, the mechanism for UE to allocate power resources, and the algorithm used by srsLTE to generate and display results.

E. Resource Consumption

Due to the recent progress in semiconductor technologies, most of the current computing systems are equipped with multiple CPU cores. A default scheduling mechanism is available to allocate the computing tasks cross the multiple cores. The goal of this experiment is to investigate the utilization of computing resource including CPU and memory usage. Specifically, we compare the performance between a laptop and Jetson TX2.

We start the eNB and EPC programs on Jetson TX2 by default command which will assign the processor dynamically. The CPU usage is obtained through the "top" command to monitor the process state every second. The data collected is the average of ten minutes when no UE is attached to the network. Note that Jetson TX2 has six CPU cores and each CPU may be allocated to execute threads of eNB and EPC during the test duration. The operations of eNB, EPC and other system programs on this platform are all following the system's default scheduling rules. In order to evaluate the impact of the number of CPU cores on our application, we invoke the "taskset" command to limit the application to run on one specific CPU. Through the same test duration and method for data acquisition, we get the second set of average CPU and memory usage on Jetson TX2. Similarly we obtain the measurement when a laptop is used. Fig. 5 summarizes the experimental results, where the y-axis is the consumption of computing resource in terms of percentage.

The figure illustrates that there is notable variation in terms of CPU usage and memory usage between fixed CPU scheduling and dynamic CPU scheduling. The trend can be observed in both Jetson TX2 and laptop. It is interesting to note that fixed CPU mode consumes less resource than the dynamic mode: 2% for a laptop and 3% for the Jetson TX2. A possible reason for this is that the scheduling and task dispatch require extra computing resource when multiple CPU cores are used. This also suggests that the current GNU Radio has not fully explored the multi-CPU resource, which indicates the need for future development.

F. Discussions

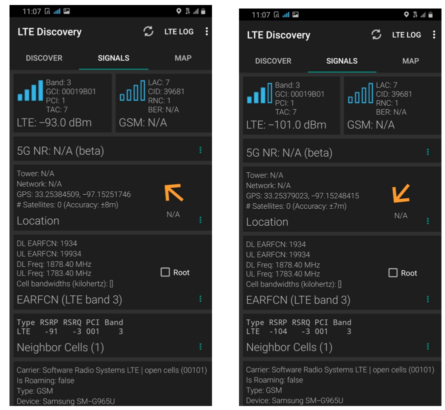
The above comparative studies show the benchmarks cross computing platforms and distances. Firstly, the end-to-end delay enlightens that the system structure of Jetson TX2 may affect the latency more than laptop does. The network connection on Jetson TX2 is more unstable than that on laptop as well as network latency, which is reflected by mdev (the standard deviation of RTT). Secondly, indoor tests about throughput are influenced the same as the performance of time delay. The throughput on Jetson TX2 is greatly reduced where the results on Jetson TX2 obtained at almost all test points are two-thirds of that on the laptop. In addition, the indoor test results show that the wireless signal is possibly affected by the external environment, and the wireless channel has many unpredictable uncertainties. Such characteristics of the wireless connection challenge the software when they are migrated from GPP onto Jetson TX2, which can be inspected from the result of SNR. The tests on resource consumption indicate that the resource allocation possibly impacts the upper-level software on its performance. Applications may behave differently depending on assigned hardware resources. The scheduling approach provided by the platform is of significant influence for achieved performance of applications. Moreover, upper-level applications can adopt resource methodologies at the program level to maximize the production of the hardware resource.

V. Migration onto the Drone

A. Experimental Setup

We justify the feasible LTE system when moving the LTE system onto the drone. Due to the compact size of the computing platform (Jetson TX2) and SDR (LimeSDR USB), we can migrate the LTE system onto the drone without changing any system module and their configurations. To monitor the LTE system remotely, we build a wireless connection from the test laptop to the LTE system to send commands and inspect results. Due to the mobility and a large scale of distance for the drone to fly in the outdoor environment, the test is controlled and measured under a larger distance interval than that indoor, demonstrating noticeable performance

Fig. 6. Experiments over UAS



(a) Display at 3 meters (b) Display at 5 meters

Fig. 7. Signal parameters are detected on the UE: (a) UE display at 3 meters; (b) UE display at 5 meters.

differences. We investigate the system performance on the drone at two points where UE is far from the system, 3 meters and 5 meters.

B. System Characteristics

1) Signal parameters: We use an app on the cell phone to identify the wireless features of the LTE system we simulate. From the app display shown in Fig. 7, we verify the established signals. The main characteristics of the LTE system are shown, such as band 3, DL frequency, UL frequency (the EARFCN part in Fig. 7), and open-source project of LTE (srsLTE) (the carrier part in Fig. 7), discussed in previous section (Section IV). We also detect the signal power that are -93.0 dBM at 3 meters and -101.0 dBM at 5 meters.

2) End-to-end delay: We explore the system performance at two distances between the UE and the LTE system, 3 meters and 5 meters. We repeat the experiments three times at a point and average the experimental results in Fig. 8.

3) Throughput: The process for measuring the experimental throughput is similar with that for the end-to-end delay. At the two points (3 meters and 5 meters), we repeated iperf process for ten seconds. The experimental throughput is shown in Fig. 9.

4) Signal-to-Noise Ratio: We calculate the equivalent SNR by adding the SNR and PHR in the system output. We

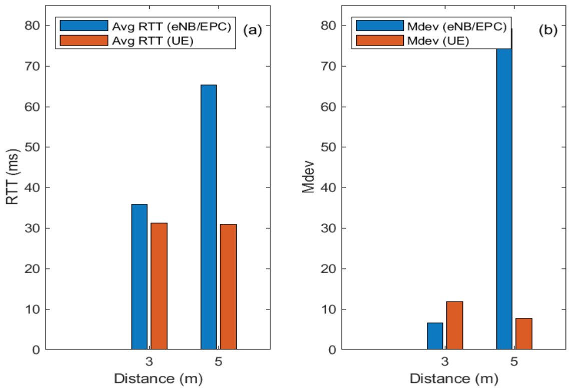
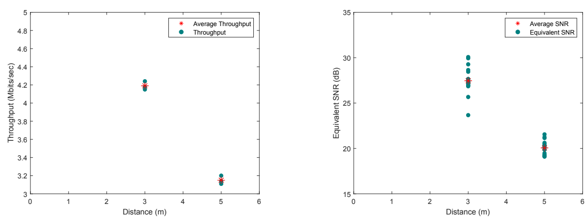


Fig. 8. Round Trip Time (a) and Mean Deviation (b) measured on eNB/EPC and UE sides respectively.



(a) Throughput (Downlink)

(b) Signal-to-Noise Ratio

Fig. 9. Experiments over UAS: (a) Downlink Throughput at 3 meters and 5 meters; (b) Equivalent SNR (SNR + PHR) at 3 meters and 5 meters.

repeat the experiments twice where each experiment lasts ten seconds. The results are shown in Fig. 9.

C. Discussion

Due to the consideration of the size, payload and power, we validate the migration of the LTE system onto UAS without replacing any module. We detect the wireless signals and investigate the system performance in a scale, verifying the feasible UAS application by employing SDR (LimeSDR USB) and the embedded computing platform (Jetson TX2).

VI. Conclusions

In this paper, we have systematically investigated how to implement an experimental LTE system using a light-weighted software-defined radio (LimeSDR), a high-performance mobile computing platform (Jetson TX2), and an open-source LTE software platform (srsLTE). Using the prototype, we have conducted extensive experiments to evaluate the performance of the experimental LTE system, such as throughput, delay, signal-to-noise ratio, resource consumption, etc. Furthermore, we also compared the computing and communication performance of the aforementioned LTE system with the one to use general-purpose processors (GPPs) as the computing platform. This performance comparison confirms feasibility to use high-performance mobile computing platform to implement an LTE system. The experimental results also suggest that the performance of such an LTE system can be further improved if the software platform can fully utilize the computing capability, such as multiple cores, GPU, etc.

Acknowledgement

We would like to thank National Science Foundation under Grants CI-1953048/1730675/1730570/1730325, CNS-1522458, CAREER-2048266, and CAREER-1714519, for the support of this work.

References

[1] K. Lu, J. Xie, Y. Wan, and S. Fu, "Toward uav-based airborne computing," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 172–179, 2019.

[2] A. F. Selva, A. L. Reis, K. G. Lenzi, L. G. Meloni, and S. E. Barbin, "Introduction to the software-defined radio approach," *IEEE Latin America Transactions*, vol. 10, no. 1, pp. 1156–1161, 2012.

[3] M. Rahman, S. Latif, T. Ahad, P. K. Dey, M. F. Ur Rashid, B. Rashid, and S. Ahmed, "The study of OFDM ICI cancellation schemes in 2.4 GHz frequency band using Software Defined Radio," *7th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2011*, 2011.

[4] R. Pi, H. Wijanarko, and A. Ahmad, "Low Cost Nano Satellite Communication System," *2018 International Conference on Applied Engineering (ICAE)*, pp. 1–4, 2018.

[5] A. A. Tabassam and S. Heiss, "Bluetooth clock recovery and hop sequence synchronization using software defined radios," *2008 IEEE Region 5 Conference*, pp. 23–27, 2008.

[6] B. Wang, J. Xie, S. Li, Y. Wan, S. Fu, and K. Lu, "Enabling high-performance onboard computing with virtualization for unmanned aerial systems," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, pp. 202–211.

[7] "3gpp." [Online]. Available: https://www.3gpp.org/technologies/keywords-acronyms/98-lte

[8] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. J. Leith, "srslte: An open-source platform for lte evolution and experimentation," in *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*, 2016, pp. 25–32.

[9] M. Miozzo, N. Bartzoudis, M. Requena, O. Font-Bach, P. Harbanau, D. López-Bueno, M. Payaró, and J. Mangues, "Sdr and nfv extensions in the ns-3 lte module for 5g rapid prototyping," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018, pp. 1–6.

[10] Y. Ma, R. Jacobs, D. G. Kuester, J. Coder, and W. Young, "Sdr-based experiments for lte-laa based coexistence systems with improved design," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[11] T. Juhana and S. Girianto, "An SDR-based multistation FM broadcasting monitoring system," in *Proceeding of 2017 11th International Conference on Telecommunication Systems Services and Applications, TSSA 2017*, vol. 2018-Janua, 2018, pp. 1–4.

[12] M. Moradi, K. Sundaresan, E. Chai, S. Rangarajan, and Z. M. Mao, "SkyCore: Moving Core to the Edge for Untethered and Reliable UAV-Based LTE Networks," *Communications of the ACM*, vol. 64, no. 1, pp. 116–124, 2018.

[13] "Open cell." [Online]. Available: https://open-cells.com/index.php/sim-cards/

[14] F. Gringoli, P. Patras, C. Donato, P. Serrano, and Y. Grunenberger, "Performance Assessment of Open Software Platforms for 5G Prototyping," in *IEEE Wireless Communications*, vol. 25, no. 5. IEEE, 2018, pp. 10–15.