Demo: Multi-Radio Access Technology IoT Gateway

Saptarshi Hazra, Thiemo Voigt, Bengt Ahlgren, Chenguang Lu,* Daniel Cederholm,* Gyanesh Patra*

RISE SICS, Sweden. *Ericsson Research, Sweden.

saptarshi.hazra@ri.se,bengt.ahlgren@ri.se, thiemo.voigt@ri.se,chenguang.lu@ericsson.com
daniel.cederholm@ericsson.com,gyanesh.patra@ericsson.com

1 Introduction

Internet of Things (IoT) solutions have been enacted in different application domains independently targeting particular use cases resulting in a plethora of different Radio Access Technologies (RATs) and application architectures. Upcoming challenges particularly in smart cities and industries require a combination of these wireless technologies to address different use cases. The lack of integrated infrastructure for convergent access is further exacerbated by the ever increasing number of RATs and the lack of a single solution fitting the requirements of all use cases.

Traditional approaches have used dedicated radio hardware for different communication technologies. This approach is hard to scale and lacks flexibility. In our previous work, we have introduced VGATE [1], a virtual gateway. VGATE uses Software Defined Radios (SDRs) for designing the core baseband functionality as software functions. The functions are implemented as containers on an edge and fog computing infrastructure. This approach makes the baseband functions easy to upgrade to future standard revisions. It also enables the possibility of deploying the baseband resources based on network load leading to better resource utilization with better energy efficiency.

In this demo, we present a basic implementation of our VGATE architecture incorporating three baseband functions, namely IEEE 802.15.4, LoRa and NB-IoT in a single testbed. We showcase the following:

- Simultaneous multi-channel access for IEEE 802.15.4.
- Feasibility of our VGATE architecture for providing convergent access.
- Feasibility of simultaneously running multiple RATs without loss of performance on the same shared infrastructure.

2 VGATE Design and Implementation

Our VGATE architecture has two main infrastructure layers: edge and radiohead. The radiohead manages access and transfers radio samples to and from the wireless medium using an USRP SDR platform. The edge hosts the baseband function for each RAT. Both the radiohead and edge functions are virtualized as self-contained containers. In this design, multiple RATs can be simultaneously supported at one shared Edge infrastructure.

We design the baseband functions to explore necessary methods needed to support the simultaneous operation of multiple RATs on a single shared infrastructure. A cluster of these functions and methods are formed into a container which is deployed on the edge and radiohead. We discuss our software components below.

2.1 Radiohead

The radiohead functions are designed in the GNU Radio framework. The main functionalities are the following:

- **USB-Ethernet conversion**: Performs the interface adaptation from the USB-connected USRP to Ethernet using both TCP and UDP as transport protocols.
- **IQ-sample filtering**: In order to reduce the traffic on the Ethernet, we implement preamble detection on the radiohead. This enables us to suppress unnecessary radio samples and transmit only radio samples that belong to a packet to the edge.
- **Multi-channel multiplexing/de-multiplexing**: This is implemented for IEEE 802.15.4. It multiplexes multiple channels into a single wideband signal on the downlink. For the uplink, it demultiplexes the wideband signal into radio samples for each individual channel. This is implemented using polyphase filters.

2.2 Edge

The edge hosts the core signal processing functions for each RAT and also the networking stack. The main functionality of the Edge container for each of our RATs are:

- **IEEE 802.15.4**: Contiki-NG provides the network stack for the IEEE 802.15.4 function through a UDP-based radio driver. The CSMA Medium Access Control (MAC) layer is modified to support the increased round trip times introduced by the SDR setup. The physical layer (PHY) is adpated from [] to support our architecture and provides multi channel support. For our

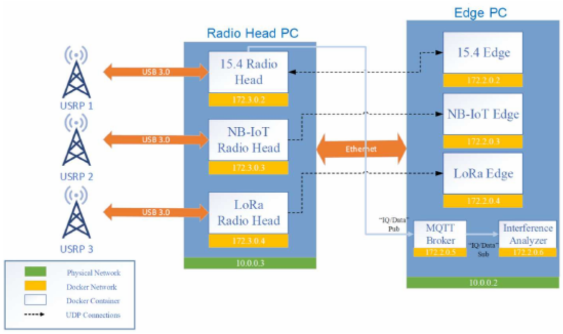

Figure 1: Evaluation Setup

demonstration we use a Light Weight Machine to Machine Server (LWM2M) as our application layer.

- **LoRa**: We adapt the GNU Radio-based LoRa setup to our architecture and also introduce multi channel support. Our LoRa function supports L1 and simple application layer sending and receiving messages with the Pycom Fife platform.
- **NB-IoT**: We implement some key components of the OFDM transmitter for NB-IoT in GNU Radio. A convolutional coding rate of 1/2 is used. We have developed a chat application that supports sending of Telegram messages from a mobile phone to the receiver.

3 Evaluation

Setup. As shown in Figure 1, our evaluation setup includes three USRPs with the corresponding radioheads that perform the radiohead functions described above. USRP 1 is configured at 2.4 GHz band with a sample rate of 15 Msps to cover three 5 MHz channels for the IEEE 802.15.4 multi-channel implementation. USRP 2 handles NB-IoT; also at 2.4 GHz but non-overlapped with the IEEE 802.15.4 carriers. The choice of 2.4 GHz is only for test and demonstration purpose to be able to transmit the RF signal over the air. The sample rate is set to 1.92 Msps. Finally, USRP 3 is configured at the EU 863-870 MHz unlicensed band for LoRa. The sample rate is set to 1Msps to support up to 2 channels.

The radio heads communicate with the edge over Ethernet using UDP. The edge hosts the networking stack including the physical layer. As IoT devices we use a Zolertia Firefly (802.15.4), a LoRa Pycom FiPy and for NB-IoT, we develop our own NB-IoT receiver emulator using an USRP and a laptop or mini PC, on which some key components of the NB-IoT receiver are implemented using GNU Radio.

Experimental results. The goal of this experiment is to show that the performance of each IoT RAT should not suffer when running with other IoT RATs simultaneously.

Figure 2 shows the ping results as a CDF of the round-trip times (RTT) values and the packet error (PER) in each RAT combination. The figure shows increase in PER value when including additional RATs. As the execution of IEEE 802.15.4 is highly compute intensive, the packet error rate (PER) increases slightly due to the resource contention at the radio head. The MAC retransmission timeout for Acknowledgements of successfully transmitted packets is set to 100ms, which explains the knee in the curves around that 100ms point in Figure 2. In real scenarios, the radio head functions will be implemented in hardware, e.g. FPGA, DSP etc. which would avoid this problem. Therefore, the results show that IEEE 802.15.4 can be realized with same performance in presence of other RATs in our Multi-RAT IoT scenario.

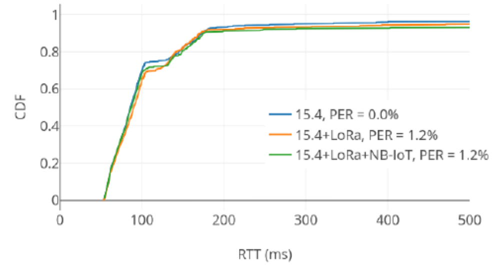

Figure 2: Comparison CDF and RTT of Ping and PER for IEEE 802.15.4 in multi-RAT scenario

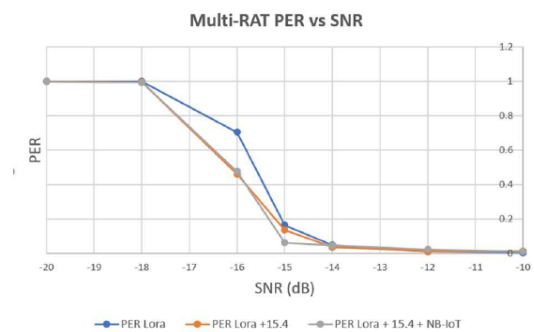

Figure 3: Comparison PER vs SNR for LoRa in multi-rat scenario, SF = 7

Figure 3 shows the PER vs SNR results for LoRa in a multi-RAT scenario. In this experiment, the LoRa received signal power is kept constant at -28dBm and noise is added in the radio head software to for SNR configuration to obtain a specific value of SNR. The Pycom FiPy continuously transmits 500 data packets with SF = 7 and at the edge we count the number of received packets for different SNRs. Figure 5-6 shows that the results are very similar among three scenarios for single, two and three RATs running simultaneously, respectively. The results indicate that the performance of our LoRa receiver is not affected by the presence of other RATs with all the combinations showing close to 0% PER when SNR is set greater than -10dB.

Conclusions. We have presented VGATE, a multi-RAT IoT gateway. Our results indicate that with VGATE, multiple RATs can be simultaneously supported at one shared edge infrastructure with little impact on protocol performance.

4 References

[1] S. Hazra, S. Duquennoy, P. Wang, T. Voigt, C. Lu, and D. Cederholm. Handling inherent delays in virtual iot gateways. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 58–65. IEEE, 2019.

5 Demo setup

Figure 4: Demo hardware setup

The setup of the demo is illustrated in Figure 4. In addition to the equipment in the picture, one Ethernet switch and at least one laptop or one screen for demo display is used.

The needs of the demo are:

- A table at least about 90cm X 120cm, preferrably longer.
- Power strip with 6 outlets.
- WiFi connectivity, which mostly is needed for adminstrative tasks, not the actual demo.
- If possible, borrow one or two 24” computer screens and a keyboard.