Poster Abstract: Taming Link-layer Heterogeneity in IoT through Interleaving Multiple Link-Layers over a Single Radio

Hassan Iqbal
Computer Science Department, LUMS
16060023@lums.edu.pk

Muhammad Hamad Alizai
Computer Science Department, LUMS
hamad.alizai@lums.edu.pk

Zartash Afzal Uzmi
Computer Science Department, LUMS
zartash@lums.edu.pk

Olaf Landsiedel
Chalmers University of Technology
olafl@chalmers.se

ABSTRACT

We propose dynamic reconfiguration of the radio interface in IoT platforms to support multiple link layers simultaneously. This allows us to tackle the increasing link layer heterogeneity in IoT devices, thus bringing a multitude of benefits: extensible and vender agnostic multihop deployments, rapid integration of the new "things" into an existing network, as well as seamless integration of the IoT with the traditional wireless Internet.

CCS CONCEPTS

•**Computer systems organization** → **Embedded systems;** *Redundancy;* Robotics; •**Networks** → Network reliability;

KEYWORDS

Link layer heterogeneity, multihop IoT, dynamic reconfiguration

ACM Reference format:
Hassan Iqbal, Muhammad Hamad Alizai, Zartash Afzal Uzmi, and Olaf Landsiedel. 2017. Poster Abstract: Taming Link-layer Heterogeneity in IoT through Interleaving Multiple Link-Layers over a Single Radio. In *Proceedings of SenSys '17, Delft, Netherlands, November 6–8, 2017,* 2 pages.
DOI: 10.1145/3131672.3136966

1 INTRODUCTION

The growing IoT application myriad poses diverse communication requirements in terms of reliability, latency, throughput, and energy consumption. This has led to the development of a wide range of communication standards — Zigbee, Bluetooth LE (BLE), Z-Wave, LoRaWAN, WirelessHART, and IEEE 802.11 ah to name a few. Although this wider choice of wireless protocols is welcomed by application specific deployments based on homogenous IoT platforms, it is complicating internetworking in others; we see that most deployments are characterized by the burgeoning heterogeneity of IoT devices and their communication interfaces [2]. This heterogeneity stems from the evolving nature of IoT (adding new or replacing faulty "things"), functional diversity (sensors, actuators, and relays in a single deployment), and cost-driven choices from the competitive vendor landscape (each employing its own choice of communication interface). Thus, incompatible wireless interfaces are inevitable, for example, when a new "thing" with a different wireless interface from a specific vendor is to be deployed into an existing network. Traditionally, internetworking in such heterogeneous deployments is achieved through special gateways, potentially supporting a subset of wireless interfaces in the network, connected through a cloud assisted backbone. This approach can be both costly, as in the example above a new thing would also require a new gateway, and inefficient, as it potentially incurs cloud induced latencies.

We propose a novel approach to solve this looming heterogeneity problem: *dynamic reconfiguration of a single radio to support multiple link layers simultaneously.* This is enabled by the increasing availability of multi-standard supporting 2.4-GHz radios in modern COTS IoT platforms such as SensorTag, Zolertia, Arduino Primo, and Redbear. However, the current system support limits these SoC radios to be statically configured with a particular standard while employing the relevant driver at runtime. Our main contribution is that we bring this radio (re)configuration to runtime and allow dynamic switching between corresponding radio drivers. We then combine this ability with the duty-cycling nature of IoT link layers to time-multiplex multiple link layers over a single radio. This facilitates IoT deployments in many ways: multihop RPL topology over multiple link layers (imagine Zigbee and BLE subtrees emerging from a DODAG root), extensible deployments (rapid integration of new types of nodes in existing topologies without installing gateways), and seamlessly integrating IoT with the traditional Internet (bridging between a BLE-enabled smartphone and a Zigbee network). As a proof of concept, below we describe how to dynamically reconfigure a radio to support both BLE and Zigbee standards.

2 INTERLEAVING LINK LAYERS

Our proposed solution exploits the duty-cycling nature of IoT link layers, which are designed to keep power consumption at a minimum by keeping the radio in sleep mode most of the time. Thus, as shown in Figure 1, we use this sleep time of one link layer standard at node B to temporarily switch the radio to support the other link layer(s), allowing node B to communicate with different subsets of neighbors equipped with incompatible link layers. To illustrate, node B first communicates with node A over Zigbee operating at

SenSys '17, Delft, Netherlands

DOI: 10.1145/3131672.3136966

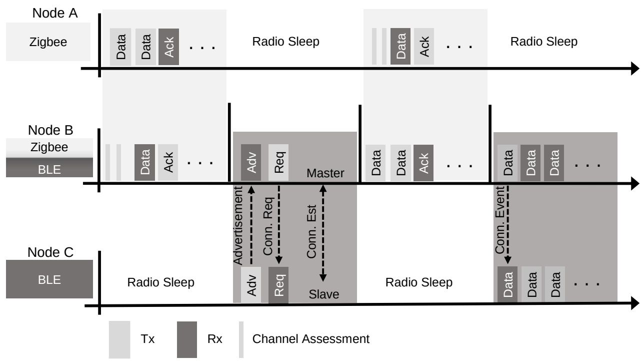


Figure 1: Interleaving link layers: A multi-standard radio at node B alternates between Zigbee and BLE to internetwork both nodes A and C, each equipped with a single but incompatible link layer standard.

its given duty cycle. When the communication finishes and the radio at node B is about to enter its sleep mode, we reconfigure the multi-standard radio at node B with BLE. Hence, our approach trades this additional energy consumption of the radio at node B with the ability to orchestrate multiple link layers for a better internetworking. Switching the radio between link layers requires the following three sequential operations: (i) check if the channel is free and the switching operation can be initiated, (ii) reboot the radio, and (iii) reconfigure the radio with new parameters. This is followed by the normal operation of the newly reconfigured link layer until the next switching operation.

For this approach to work, the timescales and energy consumption of this switching operation must be significantly smaller to achieve successful operation of multiple link layers over a single radio hardware. For instance, the sleep time of one link layer serves as a theoretical upper bound on switching time if only two link layers are being multiplexed. We next evaluate these factors as primary benchmarks, defining the future performance of our system.

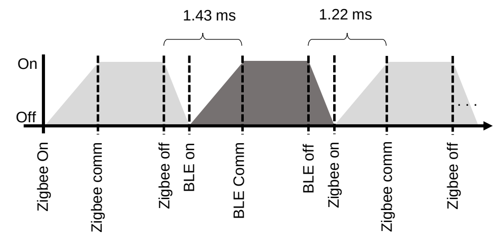
3 PRIMARY BENCHMARKS

Our evaluation (cf. Figure 2) is performed on TI's SensorTag CC2650, with a multi-standard radio, using the Contiki operating system.

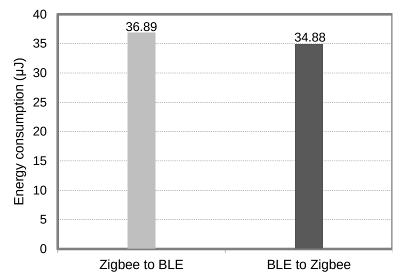
Figure 2(a) depicts the **switching time** for Zigbee to BLE and vice versa. These time scales are indeed significantly smaller than the typical duty-cycles of respective link layers reported in the literature, indicating at the successful operation of the respective link layers. To contextualize, we can also compare these switching times (1.43 ms and 1.22 ms) with a typical packet transmission time: Zigbee needs around 5-8 ms [1], depending upon the size of payload, to successfully transmit a single packet. The switch from Zigbee to BLE takes a little longer because it is a statefull protocol, thus requiring more parameters to be reconfigured. Figure 2(b) shows the corresponding **energy consumption**, which is proportional to the time, of the two switching operations.

4 CHALLENGES AND FUTURE WORK

After establishing the principle feasibility of our approach, we now elaborate on challenges that delineate our future work.



(a) Switching time



(b) Energy Consumption

Figure 2: Primary benchmarks: switching time and energy consumption of interleaving BLE and Zigbee.

Switching between multiple standards requires to *maintain state* across interleaves. This includes, for example, remembering the time slots in TDMA based schemes, and the active and passive neighbors in connection oriented standards such as BLE. Precise identification of these relevant states is fundamental for the correctness of our scheme, as well as of the entire networking stack. Similarly, these interleaves must also meet protocol specific *deadlines*, such as *connection intervals* between a master and its slaves in BLE. Furthermore, interleaving could be challenging under conflicting requirements such as network congestion on one link layer vs meeting a deadline on the other.

Finally, we will have to develop a suitable software abstraction to define the interface of such reconfigurable link layers with the higher layers of the protocol stack. Some important decisions are to be made with regard to, for example, whether to expose each link layer separately or hiding unnecessary complexities by providing an aggregated functionality to the networking layer. In the latter case, the software abstraction must also handle generic network operations, such as routing broadcasts (should be sent on all link layers), and protocol specific details including address translations and fragmentation on each supported link layer standard.

ACKNOWLEDGMENTS

This work is partly funded by the Higher Education Commission (HEC) Pakistan through grant NRPU2306, Swedish Research Council (VR) through the project ChaosNet, and by the Swedish Foundation for Strategic Research (SSF) through the project LoWi.

REFERENCES

[1] Muhammad Hamad Alizai, Olaf Landsiedel, Jó Ágila Bitsch Link, Stefan Götz, and Klaus Wehrle. 2009. Bursty Traffic over Bursty Links. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*.

[2] C. Perera, C. H. Liu, and S. Jayawardena. 2015. The Emerging Internet of Things Marketplace From an Industrial Perspective: A Survey. *IEEE Transactions on Emerging Topics in Computing* 3, 4 (2015), 585–598.