Adaptive Radio and Transmission Power Selection for Internet of Things

Di Mu, Yunpeng Ge, Mo Sha
Department of Computer Science
State University of New York at Binghamton
{dmu1, yge6, msha}@binghamton.edu

Steve Paul, Niranjan Ravichandra, Souma Chowdhury
Department of Mechanical and Aerospace Engineering
University at Buffalo
{stevepau, nravicha, soumacho}@buffalo.edu

***Abstract*—Research efforts over the last few decades produced multiple wireless technologies, which are readily available to support communication between devices in various Internet of Things (IoT) applications. However, none of the existing technologies delivers optimal performance across all critical quality of service (QoS) dimensions under varying environmental conditions. Using a single wireless technology therefore cannot meet the demands of varying workloads or changing environmental conditions. This problem is exacerbated with the increasing interest in placing embedded devices on the user's body or other mobile objects in mobile IoT applications. Instead of pursuing a one-radio-fits-all approach, we design ARTPoS, an *adaptive radio and transmission power selection* system, which makes available multiple wireless technologies at runtime and selects the radio(s) and transmission power(s) most suitable for the current conditions and requirements. Experimental results show that ARTPoS can significantly reduce the power consumption, while maintaining desired link reliability.**

I. Introduction

Diverse wireless technologies, produced by research over the years, are available to support communication between devices in various Internet of Things (IoT) applications. However, each of these technologies were originally designed with different goals, such as high throughput, low power consumption, low latency, and robustness to interference, and thus offer very different characteristics. None of the existing technologies delivers optimal performance in all desirable quality of service (QoS) dimensions, especially under varying environmental conditions. For instance, WiFi can provide high throughput, but suffers from high power consumption. A considerable amount of energy can be wasted if a WiFi radio experiences irregular data transmission at low data rate such that it stays longer in a power-hungry active mode, rather than in the power save mode. On the other hand, ZigBee is power-efficient, but cannot support high data rate applications.

Using a single wireless technology therefore cannot meet the demands of varying workloads or changing environmental conditions. This issue becomes further pronounced with emerging mobile IoT applications that involve placing embedded devices on the user's body or other mobile objects. Monitoring and controlling mobile objects open up opportunities for novel and exciting IoT applications (e.g., assisted living, health monitoring, and multi-agent autonomous vehicular and robotic systems), while also introducing the fundamental challenge of maintaining optimal wireless communication between devices under the following uncertainties: **Network Traffic Uncertainties**: The network traffic is subject to spontaneous changes. For instance, in a health monitoring application, a wearable device may produce low amount of data during some hours of the day, but sporadically require rapid transmission of large volume of data in response to a critical medical condition. Moreover, devices may have multiple sensors, with diverse traffic patterns, and the system may turn ON or OFF any of the sensors at any given time. **Wireless Environment Uncertainties**: The wireless environment changes when the device moves around. At times, a mobile device will need to be able to deal with a highly noisy environment; at other times it may enjoy a clean environment [1]. A stationary device may also experience environment changes due to changing ambient interference. Given the dynamic nature of communication in IoT applications, a traditional one-radio-fits-all approach cannot meet the challenges associated with the dynamics and uncertainties in network traffic and operating conditions.

Fortunately, embedded system hardware and radio technologies have been seeing appreciable advancement. Heterogeneous radios, e.g., WiFi, LTE, Bluetooth, and ZigBee are becoming increasingly available in modern embedded or mobile devices. Most smartphones nowadays support WiFi, LTE, and Bluetooth. A majority of modern devices designed for IoT applications also support heterogeneous radios. For instance, Firestorm platform [2] supports Bluetooth low energy (BLE) and ZigBee and uses a 32 bit low-power microcontroller with the duty cycling capability. TI CC2650 [3] integrates two radios (i.e., ZigBee and BLE) on a single chip. Raspberry Pi 3 model B [4] uses a Broadcom single-chip radio supporting both WiFi and BLE. Recent hardware advancement offers new opportunities to use multiple wireless technologies efficiently.

This paper aims to address the previously stated networking challenges, while leveraging the above-stated hardware advancements; specifically, it makes the following contributions:

- We design the *Adaptive Radio and Transmission Power Selection (ARTPoS)* system that makes available multiple wireless technologies at runtime and selects the radio(s) and their transmission power(s) that are best suited for the current network traffic and operating conditions.
- We develop new modeling approaches that allow the selection system to **adapt** to large variance in power consumption and link reliability measurements.

- We formulate the problem of radio and transmission power selection as an optimization problem and develop a practical (lightweight) online solution.
- We implement the ARTPoS in Raspbian Linux and Contiki and evaluate it on a new embedded platform supporting WiFi, ZigBee, and BLE; these efforts demonstrate the unique benefits of **adaptive** runtime selection of radios and their transmission powers.

The remainder of the paper is organized as follows. Section II reviews related work and Section III introduces our ARTPoS design. Section IV presents the power consumption and link reliability modeling and Section V introduces our problem formulation and solution strategy. Section VI presents our experimental evaluation. Section VII concludes the paper.

II. Related Works

Bandwidth aggregation for a device with multiple network interfaces has been studied extensively in the literature and many techniques are readily available [5]. Those early efforts are not directly applicable to embedded wireless devices with power constraints since they were not designed to provide *energy-efficient* wireless radio interfaces [6], [7]. There has also been increasing interest in studying the energy-aware bundling or switching between WiFi and 3G/4G radios on smartphones [8], [9]. There exists software, e.g., VideoBee, Super Download Lite-Booster, MPTCP in iOS, KT's GiGA LTE, that support concurrent use of WiFi and cellular radios. More recently, research efforts have begun to pay more attention to energy efficiency in the context of smartphones. Examples include generating energy models for smartphones [10]–[14] and WiFi/3G/LTE [15]–[17] and developing radio switching or bundling approaches [6], [7], [13], [18], [19].

These existing approaches are not directly applicable to support various embedded and IoT applications in an energy-efficient manner using heterogeneous radios since they are tailored for smartphones to support high data rate applications, limited to mainly WiFi and 3G/4G, and not integrated with transmission power control. Generally speaking, it is largely unknown how to energy-efficiently use radios with very different characteristics through runtime radio and transmission power adaptation. To address this critical gap in the current state of the art, this paper investigates the joint impact of radio and transmission power selection on energy efficiency and link reliability, and proposes a practical approach that intelligently uses a high throughput radio (i.e., WiFi) and an energy-efficient radio (i.e., ZigBee). To our knowledge, the ARTPoS system presented in this paper is the first to support not only runtime bundling and switching between WiFi and ZigBee but also adaptive transmission power control, that proactively minimizes power consumption subject to given network traffic and operating conditions.

Transmission power control for a single radio has been extensively investigated in the literature of wireless sensor networks and wireless mesh networks. Indirect link quality metrics such as received signal strength indication (RSSI) and link quality indicator (LQI) [20], [21] or direct link quality metrics such as packet reception ratio (PRR) and packet error rate (PER) [22], [23] have been used to measure the link quality. Heuristics [22], [24], [25] and control-theoretic approaches [20], [21], [23] have been applied to achieve the desirable link quality by controlling the transmission power at runtime. These existing approaches, designed to control a single radio, are not directly applicable here, since the power consumptions have to be compared between different radios and the link quality and power consumption of multiple radios have to be jointly considered. In contrast, this paper employs a pragmatic integrated systems approach to optimize the transmission power selection **together** with the radio selection. The performance of our ARTPoS has been demonstrated via implementation and experiments on real hardware.

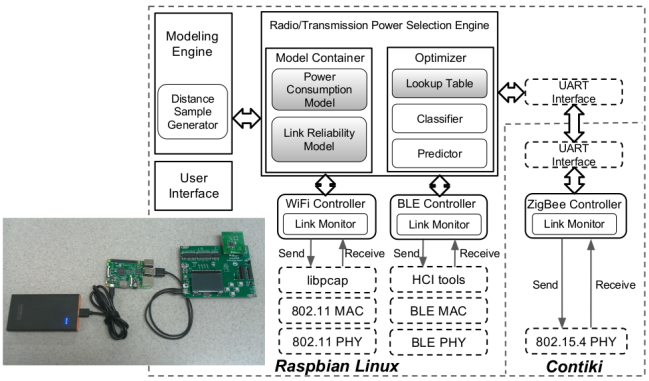
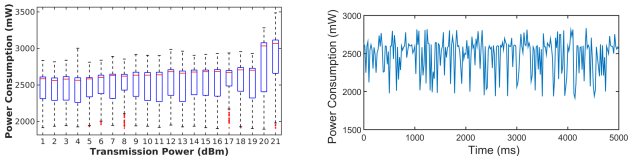


Fig. 1. System architecture.

III. ARTPoS System Architecture

This section presents the design of ARTPoS. Fig. 1 shows the system architecture. The **Modeling Engine** generates the power consumption and link reliability models needed for the radio and transmission power selection (Section III-A). The **Radio/Transmission Power Selection Engine** selects the best-suited radio(s) and transmission power(s) based on the application specified data rate and the throughput of each available link measured at runtime (Section III-B). Multiple **Radio Controller** modules (e.g., WiFi, BLE, and ZigBee controllers) exist in ARTPoS. Each radio controller controls the state (i.e., on or off) of a radio and sets its transmission power based on the decision made by the Radio/Transmission Power Selection Engine, while the **User Interface** supports the interactions with system users (Section III-C).

To support the realization of ARTPoS, we have built a new embedded platform with heterogeneous radios consisting of WiFi, ZigBee, and BLE by instrumenting a Raspberry Pi 3 Model B [4] with a TI CC2650 development kit [3] as shown in Fig. 1. CC2650 is connected to the Raspberry Pi through a USB port. The integrated emulator (XDS100v3) on CC2650 enables the communication between Raspberry Pi and CC2650 through a UART. Raspberry Pi integrates a Broadcom BCM43438 single chip radio processor supporting WiFi and BLE, while CC2650 has a multi-standard wireless MCU supporting ZigBee and BLE. (Currently, we use the BLE radio on Raspberry Pi since the Contiki has not yet implemented the BLE stack in its master branch). To power



(a) Boxplot at transmission powers of 1 dBm to 21 dBm.

(b) A 5-second trace at minimum transmission power (1 dBm).

Fig. 2. Power consumptions of our embedded platform when only WiFi radio is on and transmits at maximum speed. The traces are measured by a Monsoon power meter [26]. In boxplot, central red mark in box indicates median; bottom and top of box represent the 25th percentile (q_1) and 75th percentile (q_2); crosses indicate outliers ($x > q_2 + 1.5 \cdot (q_2 - q_1)$ or $x < q_1 - 1.5 \cdot (q_2 - q_1)$); whiskers indicate range excluding outliers.

the device, we use a USB battery to which a Monsoon power meter [26] is connected to measure the power consumption.

We have realized ARTPoS in Raspbian Linux [27], a Debian based Linux system for Raspberry Pi, and Contiki [28], an operating system for low-power wireless IoT devices. To support WiFi, our ARTPoS implementation adopts the 802.11 MAC and physical layer implementations provided by the Linux kernel and employs the libpcap library for sending and receiving packets to/from the MAC layer. Similarly, our implementation adopts the Linux's BLE implementations and HCI tools to support BLE and uses the 802.15.4 physical layer implementations in Contiki to support ZigBee. Our implementation also adopts the existing UART implementations in Raspbian and Contiki to support the communication between Raspberry Pi and CC2650. In Fig. 1, the existing implementations in Raspbian Linux and Contiki adopted by ARTPoS are marked with dash lines, while our new designs are marked with solid lines. WiFi controller, BLE controller, and ZigBee controller are three radio controllers that control WiFi, BLE, and ZigBee radios, respectively. We intentionally implement all modules except the ZigBee Controller in Raspbian Linux, since Raspberry Pi has richer hardware resources. The design of the major modules in ARTPoS are discussed next.

A. Modeling Engine

The Modeling Engine generates the power consumption model and link reliability model to support runtime radio and transmission power selection. Most existing solutions for transmission power control for a single radio use a simple power model assuming that using a lower transmission power level leads to lower power consumption. However, this simple model no longer works for a device with multiple radios since the power consumptions have to be compared between different radios. Hence, our Modeling Engine is designed to take real power consumption traces as input and generate power models accordingly. As an example, Fig. 2 shows the power consumptions of our embedded platform when its WiFi radio transmits at maximum speed. As shown in Fig. 2(a), the median power consumption increases from $2585mW$ to $3065mW$ when the WiFi transmission power increases from $1dBm$ to $21dBm$. Large variances can be seen in the boxplot in Fig. 2(a) as well as Fig. 2(b), which shows a 5-second power measurement when the WiFi is transmitting at $1dBm$. The large variance is caused by the power consumption differences when the WiFi radio hardware is at different states. We also measure the power consumption of the platform when its radios are under various modes (e.g., only ZigBee radio on and transmits, only BLE radio on and transmits, and all radios off) and observe large variance and non-normal distribution of the measurements. Using the first statistical moments (e.g., mean or median of the data) is deemed not suitable. Instead, we use a more robust scalar measure of the power consumption in ARTPoS, which will be discussed in Section IV-A in detail.

The Modeling Engine also generates the link reliability model based on the PRR measurements at difference distances between the sender and the receiver, and when the sender transmits at different transmission power. PRR can be defined as the fraction of transmitted packets successfully received by the receiver. Our Modeling Engine provides a feature that controls each radio to transmit packets using a single transmission power, then proceeds to the next power in a round robin fashion. With this feature, the PRR measurements for all radios and transmission powers can be done automatically at each distance. However, changing the distance between the sender and receiver has to rely on human operators, introducing labor-intensive measurement overheads. Therefore, it is important to use a frugal set of distance samples that will produce a training data set suitable for effective (subsequent) model development.

Therefore, the **Distance Sample Generator** is designed to generate suitable distance samples based on a feasible communication range and the desired number of distance samples. The desired number of distance samples is decided by the total time allowed for PRR measurements divided by the measurement execution time at each distance. A design of experiments approach adopted from the Engineering Design paradigm is used to generate the distance samples. For instance, the communication range considered, $0 - 200m$ (based on our observed maximum communication range of WiFi/ZigBee/BLE), is divided into three zones. Zone 1, $0 < x \leq 30m$, corresponds to the spatial range in typical home or office-space IoT applications, where a low-power radio like ZigBee is seeing increasing popularity; Zone 2, $30 < x \leq 100m$, corresponds to the spatial range in typical commercial/residential buildings as well as factories and warehouses (i.e., industrial IoT or IIoT applications) where ZigBee becomes progressively less effective, and WiFi is expected to become more dominant; and Zone 3, $x > 100m$, corresponds to the spatial range (typical of small autonomous ground/aerial vehicle applications) where WiFi with greater range capacity will typically dominate. In each of these ranges, we use the *Latin hypercube sampling (LHS)* method to generate 10 distance samples. LHS is a popular approach to generate near-random samples that can provide a relatively uniform coverage of an input space or a probability space [29]. Unlike factorial design or simple Monte Carlo simulations, the size of the sample set yielded by LHS does not scale exponentially with the number of input parameters, thereby making LHS more suitable to design frugal set of experiments (as needed here). A LHS containing n sample points (between 0 and 1)

over m dimensions is a matrix of n rows and m columns. Each row corresponds to a sample point. The values of n points in each column are randomly selected, one from each of the intervals, $(0, 1/n), (1/n, 2/n), \ldots, (1 - 1/n, 1)$. We use the optimal LHS implementation, which maximizes the minimum Euclidean distance between the samples [30]. To demonstrate the PRR measurement process, we collect a series of PRR traces by varying the distance between the sender and receiver following the 30 distance samples generated by LHS. Section IV-B will discuss the method that is used to train models of PRR as functions of the respective radio transmission power settings based on our collected PRR traces.

B. Radio/Transmission Power Selection Engine

The Radio/Transmission Power Selection Engine implements ARTPoS core logic. It is designed to facilitate the identification of the best-suited radio(s) and transmission power(s) at runtime. The **Model Container** stores the power consumption model and link reliability model generated by the Modeling Engine. With these two models, the **Optimizer** selects the best radio (or a set of radios) and their optimal transmission power(s) based on the application specified data rate and the throughput of all available links measured by the radio controllers. Section V will discuss the problem formulation and optimization in detail.

C. Radio Controllers and User Interface

The Radio Controllers are important design constructs of ARTPoS. Their main purpose is to forward data packets between the application and the radio stacks. The Radio Controllers are responsible for switching on the radio(s) selected by the *Radio/Transmission Power Selection Engine*, keeping the unselected radio(s) off, applying the selected transmission power(s), and routing data packets between the application and the radio stack(s) of the selected radio(s). The **Link Monitor** gathers the runtime link statistics (i.e., throughput and PRR) and feeds them to the Optimizer. To support WiFi, BLE, and ZigBee on our embedded platform, we have implemented three Radio Controllers (i.e., WiFi Controller, BLE Controller, and ZigBee Controller as shown in Fig. 1).

The User Interface supports the interactions between our ARTPoS and its user. First, it allows the system user to reveal the debugging and operation logs through a SSH connection. Second, it notifies the user to move the device to the next distance when the Modeling Engine finishes the PRR measurements at the current distance. Third, it allows the application to set its desired data rate at runtime.

IV. Modeling

This section presents the development of tailored regression models with specialized smoothing characteristics, to represent the (uncertain) nodal power consumption and PRR variations as functions of the radio transmission power settings. This modeling approach is aimed to facilitate robust radio and transmission power selection decisions (failure to address these uncertainties undermines radio selection processes, as demonstrated later in Section V-B).

A. Power Consumption Modeling

The measurements from Section III-A are used to develop quantitative models of power consumption, as functions of the transmission power setting (p) of the concerned radio. As evident from Fig. 2, significant variations, which cannot be solely attributed to change in radio transmission power, are inherent in the measurements. We therefore represent the platform base power consumption with all radios off ($E_p(V)$), and the respective platform power consumptions with only Bluetooth on ($E_b(V, p_b)$), only Zigbee on ($E_z(V, p_z)$), and only WiFi on ($E_w(V, p_w)$) as functions of uncertain parameters V and the respective transmission power of the Bluetooth, ZigBee, and WiFi radios (p_b, p_z, and p_w, respectively).

Here the quantity of interest (QoI), i.e., total power consumption, is a function of the design variable (radio transmission power setting) and a vector of uncertain parameters V, where the latter can be assumed to be outside the control of the designer and not practically measurable in the current context (e.g., radio backoffs caused by failed clear channel assessment and inaccurate power meter reading). Considering the availability of dedicated QoI data (Section III-A), it can be assumed that the uncertainty therein is quantifiable. However, given the observed large variance and non-normal distribution of the platform power consumption data (Fig. 2), using the first statistical moments (e.g., mean or median) is deemed not suitable. Instead, perceiving platform power consumption as an "expense", the notion of **s-risk** [31] is used here – to provide a robust or *uncertainty-aware* scalar measure of this expense under any given radio setting.

The notion of s-risk, also known as "conditional-value-at-risk", originated in the Finance domain [32]. We use the example of the platform power consumption with only WiFi on (E_w), to further describe the s-risk concept. Assuming that E_w follows a continuous probability distribution, for a given risk-aversive parameter γ ($0 \leq \gamma \leq 1$), the s-risk of E_w can be defined as the average value of E_w over its worst $1 - \gamma$ outcomes. Therefore, assuming N samples of E_w are available, s-risk can be expressed as:

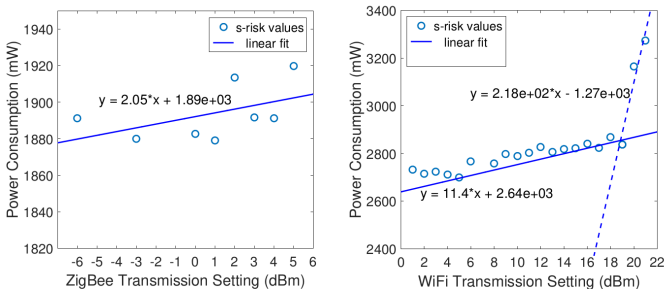
$$S_\gamma(E_w(V, p_w)) = \frac{1}{(1-\gamma)N} \sum_{\forall k \in \Gamma} [E_w(V, p_w)_k] \qquad (1)$$
$$\Gamma = \text{set of the highest } (1-\gamma)100\% \text{ values of } E_w$$

It is readily evident from Eq. 1 that higher values of γ leads to greater risk aversion or more conservative decisions. Owing to its ability to consider tails of probability distribution and ease of interpretation and computation, s-risk provides a tractable stochastic measure of the worst-case scenarios. Based on the definition in Eq. 1, we compute the following:

- s-risk value of the platform baseline power consumption (S_p) when all radios are off;
- s-risk value of the platform power consumption with only BLE on (S_b). (Raspberry Pi only supports single transmission power for BLE.);
- s-risk values of the platform power consumption with only ZigBee on (S_z) at the following different transmission power settings: $p_z \in \{-6, -3, 0, 1, 2, 3, 4, 5\}\,\mathrm{dBm}$;

- s-risk values of the platform power consumption with only WiFi on (S_w) at the following different transmission power settings: $p_w \in \{1, 2, \ldots, 20, 21\}$ dBm.

All s-risk values are computed at a prescribed $\gamma = 0.8$, which here calls for averaging over the worst 50 values in each case.



(a) With ZigBee on (b) With WiFi on

Fig. 3. Regression plots of s-risk values of platform power consumption as functions of radio transmission settings.

The s-risk values of the platform power consumption with only WiFi on and only ZigBee on are then separately modeled as linear regressions of their respective transmission settings. A piecewise linear regression is used in the case of WiFi (Fig. 3(a)), and a single linear regression is used in the case of ZigBee (Fig. 3(a)). The linear regressions provide a smoothing of the large variations in the power traces, while also yielding a monotonically increasing (instead of oscillatory) trend w.r.t. transmission power – which promotes a more robust template for selecting transmission settings (guided by power savings). The trained regression functions can be expressed as:

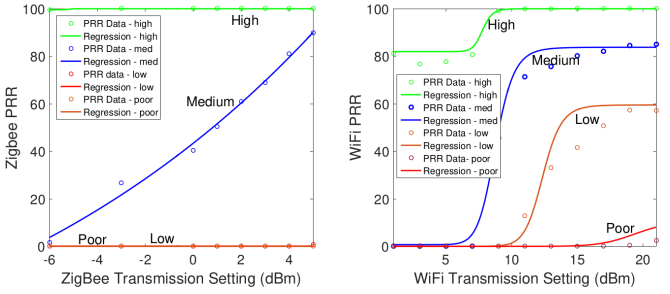
$$S_{z,0.8} = 2.05p_z + 1.89e03, \quad -6 \leq p_z \leq 5$$
$$S_{w,0.8} = \begin{cases} 1.14e01p_w + 2.64e03, & 1 \leq p_w \leq 19 \\ 2.18e02p_w - 1.27e03, & 20 \leq p_w \leq 21 \end{cases} \quad (2)$$

B. Link Reliability (PRR) Modeling

The PRR measurements from Section III-A are used to train models of PRR as functions of the respective radio transmission power settings. Here, we particularly develop the PRR models for ZigBee and WiFi, since multiple transmission power settings are available for these two radios on our platform, and they are the ones also considered in the optimal radio and transmission selection process (Section V).

We observe large variations in PRR measurements, especially when the links are in the transitional region. Specifically, at many sample transmission power settings and sender-receiver distances, significant variations in PRR are observed. The radio control scheme in practice will usually be unaware of the exact distance between the sender and receiver, as well as of the other uncertain environmental factors affecting the PRR. Instead, what is measurable at runtime are the PRR values being experienced by the individual radios. With this perspective, we propose the state of the system associated with the PRR recordings to be segregated into different performance categories. In this context, the PRR and throughput of an individual radio can also be simultaneously considered, where the categories will then represent the state of the **goodput** (i.e., PRR $\times$ throughput) in that case.

In the current implementation, four categories, namely "*poor*", "*low*", "*medium*", and "*high*" performing states, are defined w.r.t. PRR. For every transmission power setting of a radio (WiFi/ZigBee), the top 25% PRR measurements are assigned to the "*high*" state, the next 25% are assigned to "*medium*" state, the subsequent 25% are assigned to "*low*" state, and the bottom 25% are assigned to the "*poor*" state. Although the recorded (sample) distance between the sender and receiver is not explicitly considered when making this state-category assignments (i.e., all PRR measurements under a given radio setting are pooled together), the assignments are implicitly sensitive to the distance – this is because sender-receiver distance has a strong adverse impact on PRR.The mean of the PRR values categorized under each state for a given transmission setting is then computed to serve as the representative bounding value of the PRR for that state (to be referred to as the *PRR state* or *state-representative PRR* values in the remainder of this paper). Regression functions are subsequently used to fit the high, medium, low, and poor state PRR values of a radio as four separate functions of its transmission settings.



(a) PRR of ZigBee (b) PRR of WiFi

Fig. 4. Regression plots of PRR as functions of radio transmission settings; PRR data segregated into *poor*, *low*, *medium*, and *high* states.

The PRR state values were observed to present S-shaped trends w.r.t. the corresponding radio transmission power settings. This observation led to the choice of logistic regression (and sigmoid fit, a special case of logistic) to model the "PRR-p" relationships between PRR values and transmission power settings. An implementation called L4P [33] of the four parameter logistic function is used, with the PRR expressed as a function of the radio transmission power, p, as given by

$$PRR(p) = d + (a - d)/(1 + (p/c)^b) \quad (3)$$

Here, the four parameters a, b, c, and d respectively represent the minimum asymptote, the stiffness of the curve, the inflection point, and the maximum asymptote. The estimated values of the 8 sets of these four parameters are not listed here, since they are subjective to our recorded PRR measurements, and do not add significant generalized value. Instead, the four logistic functions, that are trained on the high/ medium/ low/ poor state PRR values of ZigBee and WiFi, are respectively shown in Figs. 4(a) and 4(b). It is readily evident from Fig. 4 that while adequately capturing the nonlinear S-shaped "PRR-p" relationship, the logistic regression also provides monotonically increasing "PRR-p" functions. Such a positive "PRR-

p" correlation is imperative to promoting robust transmission setting modulation − where an optimal scheme should seek to increase the radio transmission power, in response to the need to increase PRR, over the entire range of available transmission power settings. Given the noisy PRR measurements, such a trend would have been difficult to achieve using other models such as polynomial regression or interpolating functions.

V. Optimization

A. Problem Formulation

As stated before, the generalized objective of the radio and transmission selection is to adapt to the current needs of the application (under the current environment) in a way that: ***both packet loss and platform power consumption attributed to the radios are minimized***. These two criteria, packet loss and power consumption, can be perceived as the *state parameters*; and the choice of the radio type (ZigBee, WiFi, BLE, or any of their combinations) and their transmission power setting can be perceived as *action variables*. This perspective lends to formulating the radio and transmission selection process as an optimization problem. The Raspberry Pi only supports single transmission power for BLE; we therefore only consider ZigBee and WiFi in our problem formulation. (We plan to implement our own CC2650 BLE driver under Contiki and include BLE into our optimization as our future work.)

In the remainder of the paper, the PRR of WiFi and ZigBee will be respectively represented by r_w and r_z (where $0 \leq r_w, r_z \leq 1$), and the throughput of WiFi and ZigBee will be expressed in terms of the number of packets transmitted, and represented by h_w and h_z, respectively. The packet size for WiFi and ZigBee is considered to be 64 bytes. The aggregated goodput ($G_{w,z}$) of the radios is then given by:

$$G_{w,z} = h_w r_w + h_z r_z \tag{4}$$

If only one of the radios is on, the aggregated gootput reduces to the individual goodput of that radio. The power consumption of the transmitting platform can then be expressed as a function of the data rate (D), the aggregated goodput $G_{w,z}$, the platform baseline power consumption (E_p), and the estimated platform power consumption when radios operate at the given transmission settings (E_w and E_z). The time averaged power consumption of the platform is thus given by:

$$f_E = \min\left(1, D/G_{w,z}\right)\left(E_w + E_z - 2E_p\right) + E_p \tag{5}$$

where $(E_w + E_z - 2E_p)$ gives a measure of the power consumption attributable to the active radios. This measure is multiplied by the fraction of the time when the radios need to be active in a given interval; the latter is given by the "*data rate/goodput*" ratio ($\min\left(1, D/G_{w,z}\right)$). When the WiFi is off, $E_w(\text{Off}) = \text{E}_\text{p}$ and $r_w(\text{Off}) = 0$; similarly, when the ZigBee is off, $E_z(\text{Off}) = \text{E}_\text{p}$ and $r_z(\text{Off}) = 0$. It is also important to note that Eq. 5 assumes that the data is split between the two radios based on the ratio of their individual goodputs, and retransmission of lost packets is implicit to the system.

The generalized optimization problem, with the WiFi and ZigBee transmission settings (p_w and p_z, respectively) serving as the decision variables, can therefore be defined as follows:

$$\begin{aligned} &\min_{p_w, p_z} \quad f_E\left(p_w, p_z, h_w, h_z\right) \\ &\text{subject to} \qquad 1 - \frac{D}{h_w r_w\left(p_w\right) + h_z r_z\left(p_z\right)} \geq \epsilon \\ &\text{where} \\ &\qquad p_w \in \{\text{Off}, 1, 2, \ldots, 20, 21\}; \\ &\qquad p_z \in \{\text{Off}, -6, -3, 0, 1, 2, 3, 4, 5\} \end{aligned} \tag{6}$$

where the tolerance parameter ϵ represents a safety margin in the "*data rate/goodput*" ratio; e.g., $\epsilon = 0.1$ indicates a safety margin of 10% in the "*data rate/goodput*" ratio. It is important to note that both the objective function, f_E (Eq. 5), and the "*data rate/goodput*" (Eq. 6) constraint are nonlinear, since the PRR is a nonlinear function of the radio transmission power (as seen from Fig. 4). In addition, owing to the uncertainties in the PRR and throughput of the radios, and uncertainties in the power consumption of the platform, both the objective and constraint functions are also uncertain. As a result, we have an integer non-linear programming (INLP) problem with uncertainties. Although the INLP problem is NP-hard [34], the relatively limited number of transmission power settings that the two radios can assume (WiFi: 22 and ZigBee: 9) alleviates the computational burden of solving this optimization at runtime. On the other hand, uncertainties are addressed using the combination of s-risk measures and specific regression modeling of PRR and power consumption (as presented in Section IV). The execution time of solving this optimization problem is presented in Section VI-A.

An offline optimization study, illustrating the impact of the PRR and power consumption uncertainties (when left untreated) on the radio selection decisions, and the design of our online optimization scheme for runtime radio and transmission selection are discussed next.

B. Study on the Impact of Uncertainties

An offline optimization study is set up to investigate the impact of environmental uncertainties (that cause PRR variations) and systemic uncertainties (that cause power consumption variations) on the radio selections. Therefore, in this study, we neither employ any smoothing operation on the empirical data nor use the regression models developed in Section IV.

Optimization is performed for different sample combinations of distance between sender and receiver (X) and data rate (D), where $X \in \{10, 20, 30, \ldots, 150\}\,\text{m}$ and $D \in \{25, 50, 75 \ldots, 150\}\,\text{packets/s}$. A conservative safety margin of 20% ($\epsilon = 0.2$) is imposed on the data rate/goodput ratio. For a given distance, data rate, and radio transmission settings (p_w, p_z), the objective function is evaluated by directly computing the s-risk value of f_E (Eq. 5) from the platform power measurements data pertaining to the stated radio transmission settings and the PRR measurements data pertaining to given distance and radio transmission settings (Section III); a risk-aversive parameter of $\beta = 0.8$ is used here. Considering

the comparatively smaller variance in the throughput measurements and the focus of the paper on dynamic systems (where distance variation mainly affects PRR), the throughput of ZigBee and WiFi is fixed at their respective measured median values ($h_w = 800$ packets/s and $h_z = 225$ packets/s).

Since a small number of radio settings are available – 22×9 possible combinations of (p_w, p_z) – those violating the data rate/goodput ratio constraint are first filtered out; then a simple $\min$-search is employed to identify the optimal feasible setting, p_w^*, p_z^*, that yields the minimum power consumption. This process is performed for all the sample combinations of sender-receiver distance and data rate. The radio transmission setting decisions yielded by this uncertainty-sensitive optimization is shown in Fig. 5. For illustration purposes, the results for three data rates (150, 175, and 200 packets/s) are shown. In Fig. 5, the X-axis and Y-axis respectively represent the sender-receiver distance and the data rate; in the top two plots, the color of the circles represent the optimal WiFi and ZigBee transmission settings in dBm; and a missing circle indicates that particular radio was set to "OFF" (for the given data rate/distance sample). The last plot in Fig. 5 indicates whether the optimal radio setting succeeded ($= 1$) or failed ($= 0$) to satisfy the data rate/goodput ratio constraint.

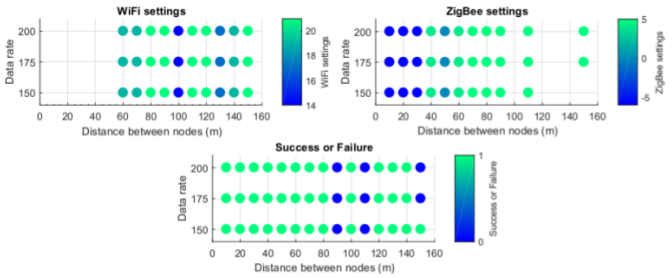


Fig. 5. Offline study: Optimal transmission power settings of ZigBee/WiFi (operating together) and success ($= 1$) or failure ($= 0$) in meeting the "data rate/goodput" ratio constraint for different distance and data rate combinations

The impact of noise/uncertainty of the empirical data (driving the nominal decisions) is apparent in the offline optimization results as shown in Fig. 5. For example, it can be seen that when increasing the sender-receiver distance, the radios often switch back and forth between higher and lower settings (instead of a more robust monotonic variation); secondly, no feasible/successful radio setting combination is found for distances of $90m$ and $110m$, although feasible/successful settings were found for higher distances of $120-140m$. **These observations highlight the detrimental impact that directly using recorded data (with their associated uncertainties) can have on any empirical decision-making strategy.** This directly motivates 1) the uncertainty-aware power consumption and PRR models developed in Section IV, and 2) the new *online* algorithm that uses these models to offer robust solutions, which will be described in the next section.

C. Online Optimization

The online optimization approach is developed to serve as a first foray into training a light-weight solution for runtime selection of radio and transmission power under an energy-scarce and uncertain/dynamic environment – typical of application domains such as home/commercial area networks or highly mobile networks. The online scheme should be able to process, interpret, and optimally respond to the uncertainties, without resorting to expensive uncertainty quantification/resolution and typical robust optimization techniques – these techniques are generally not suited to be executed at runtime on embedded systems with humble computing capacities.

Our approach aims to construct a novel runtime scheme with the following desirable characteristics: (i) **lightweight execution**, (ii) **uncertainty-awareness**, and (iii) promotion of a **power-saving radio/transmission selection policy**. It is important to reiterate that the unique models of power consumption (s-risk models) and PRR (logistic regressions), presented in Section IV, are particularly aimed at enabling this light-weight runtime scheme. Drawing parallels to robust control and Markov Decision Processes, the overall objective of the online scheme can be stated as: to maintain/accomplish desirable values of the state parameters (e.g., goodput and platform power consumption) under a dynamic and uncertain environment, by optimally modulating the action variables (i.e., selection of radio(s) and transmission setting(s)).

A look-up table system (*radio-settings-table*) is first generated. Each row (i) and each column (j) of this table respectively corresponds to a WiFi and a ZigBee transmission setting (p_z^j, p_w^i); the table thus comprises a total of 22×9 cells (See Eq. 6), where each cell C_{ij} contains one scalar value and two 4-tuples, as shown below:

$$
\begin{aligned}
&C_{ij} = \{E(p_{z,j}, p_{w,i}),\ R(p_{z,j}), R(p_{w,i})\} \\
&E(p_z^j, p_w^i) = S_{z,0.8}\left(p_z^j\right) + S_{w,0.8}\left(p_w^i\right) - 2S_{p,0.8} \\
&R(p_{z,j}) = \left(r_{z,j}^{\text{high}}, r_{z,j}^{\text{medium}}, r_{z,j}^{\text{low}}, r_{z,j}^{\text{poor}}\right) \\
&R(p_{w,i}) = \left(r_{w,i}^{\text{high}}, r_{w,i}^{\text{medium}}, r_{w,i}^{\text{low}}, r_{w,i}^{\text{poor}}\right) \\
&\text{where} \quad i = 1, 2, \ldots, 22; \quad j = 1, 2, \ldots, 9
\end{aligned}
\tag{7}
$$

In Eq. 7, the scalar $E(p_{z,j}, p_{w,i})$ represents the power consumption attributed to the active radios, when operating at the associated transmission setting combination (i, j); it is derived from the s-risk measures of power consumption (Section IV-A), where the s-risk value of the platform baseline power consumption with radios off ($S_{p,0.8}$) is estimated to be $1831mW$; the s-risk values of the platform power consumption with ZigBee on ($S_{z,0.8}$) and that with WiFi on ($S_{w,0.8}$) are estimated from the linear regressions in Eq. 2.

The two 4-tuples in Eq. 7, $R(p_{z,j})$ and $R(p_{w,i})$, represent the four PRR values corresponding to the *high*, *medium*, *low*, and *poor* operational (or performance) states of ZigBee and WiFi, respectively. These state values are given by the logistic PRR functions developed in Section IV-B (Figs. 4(a) and 4(b)).

It is important to note that in practice, the look-up table is stored/loaded in a more compact form, instead of the 22×9 table (described here for ease of illustration). Since the WiFi and ZigBee settings (i, j) are essentially independent of each other, the look-up table can be stored in the actual test bed in a form that yields a frugal set of "$1+(5\times(22+9))$" floating point

values, making it highly effective for fast runtime decision-making on embedded devices.

The runtime radio and transmission selection, that uses this lookup table, is designed as a ***sense→classify→predict→search*** process.

• ***Sense***: The online process measures PRR (reported by the receiver) and throughput of each radio at a desired sampling frequency; it computes the data rate/goodput ratio (D^t/G^t) based on the time averaged values of PRR and throughput over the last time window t. If the constraint, $1 - D^t/G^t \geq \epsilon$, is violated, it invokes the succeeding steps; otherwise, no change is made. In addition, the process computes and checks if the relative change in the D/G ratio is greater than 10%, i.e., $|D^t/G^t - D^{t-1}/G^{t-1}| > 0.1$. If this criteria is met, the succeeding steps are again invoked; otherwise no changes are made. The frequency of the constraint computation and the D/G change computation depends on the designer's preferences. More risk aversive strategies will call for higher frequency of the former, and more energy-conscious strategies will demand higher frequency of the latter. Too frequent changes however may not be recommended, as it might entail unnecessary computing overhead on the system.

• ***Classify***: If the sense process invokes the succeeding steps, first, the current state of each radio's performance, (p_w^t, r_w^t) and (p_z^t, r_z^t), is classified into the *high*, *medium*, *low*, and *poor* (or in-between) state categories. This is accomplished by the following rule: *Classify the current state of the WiFi into lying at one or between the two categories, whose associated PRR values immediately bound the measured PRR*. For example (using Fig. 4(b)), if the PRR of WiFi transmitting at $14dBm$ is 70%, then its performance/operation is classified to currently lie between the "*medium*" and "*low*" states; or if the PRR of WiFi transmitting at $4dBm$ is 90%, then its operation is classified into purely "*high*" state. A similar rule applies to ZigBee as well. More sophisticated classification schemes, such as using Bayes rule, can also be readily implemented within this process. This being the first implementation of this novel online scheme, the simpler interval based classification is instead employed here.

• ***Predict***: After the classification step, the D/G constraint (where $G = h_w^t r_{w,ij}^t + h_z^t r_{z,ij}^t$) and the energy objective function (f_E) are evaluated for each cell of the *radio-settings table*, where the latter is given by:

$$f_{E,ij}^t = \min\left(1, \frac{D^t}{h_w^t r_{w,i}^t + h_z^t r_{z,j}^t}\right) E(p_z^j, p_w^i) + S_{p,0.8}$$
$$\text{where} \quad i = 1, 2, \ldots, 22; \quad j = 1, 2, \ldots, 9 \tag{8}$$

where the PRR values of ZigBee and WiFi for each cell of the lookup table $(r_{w,ij}^t, r_{z,ij}^t)$ correspond to the classified category. More specifically, a linear interpolation is used. Taking the previous example of PRR of WiFi transmitting at $14dBm$ to be 70% – where its operational state is estimated to lie between the "*medium*" and "*low*" categories, the expected PRR of WiFi (at that time point) for say $12dBm$ will be given by:

$$r_{w,12}^t = r_{w,12}^{\text{low}} + \frac{r_{w,14}^t - r_{w,14}^{\text{low}}}{r_{w,14}^{\text{medium}} - r_{w,14}^{\text{low}}} \left(r_{w,12}^{\text{medium}} - r_{w,12}^{\text{low}}\right).$$

For purely *high* or purely *poor* states, 100 and 0 are used as the respective upper and lower bounds for the interpolation.

• ***Search***: Once the expected power consumption ($f_{E,ij}$) and the D/G constraint has been computed for all 22×9 ZigBee/WiFi settings, those violating the D/G constraint are first filtered out. A min-search is then executed to identify the optimal ZigBee/WiFi setting, $(i, j)^*$, as the one that yields the smallest value of $f_{E,ij}$. The system immediately switches to this new setting. This step can be expressed as:

$$\begin{aligned} &\min_{i,j} \quad f_{E,ij}^t \\ &\text{subject to} \qquad 1 - \frac{D^t}{h_w^t r_{w,ij}^t + h_z^t r_{z,ij}^t} \geq \epsilon \\ &\text{where} \qquad i = 1, 2, \ldots, 22; \quad j = 1, 2, \ldots, 9 \end{aligned} \tag{9}$$

In practice, the filtering of feasible solutions and searching for the optimal solution are both performed in computational efficient ways – e.g., the filtering is initiated by searching from the highest setting, $(p_z^j, p_w^i) = (5, 21)\text{dBm}$, and moving somewhat diagonally, until a setting (k, l) is reached where the constraint is violated; all other lower settings (i.e., $\forall(i \leq k, j \leq l)$) are filtered out without computing the constraint.

The median execution time of our online optimization is $49ms$ on an ARM processor. Section VI-A will present our micro-benchmark evaluations in detail.

VI. Evaluation

To examine the efficiency of ARTPoS, we perform a series of experiments on our embedded platform presented in Section III. We first measure the overhead of the key ARTPoS operations such as the time duration of the optimizer selecting the best radio(s) and needed transmission power(s) and the radio turning on and off overhead. We then evaluate ARTPoS's impact on power consumption and link reliability, and compare its performance against three baselines.

In all experiments, we deploy a benchmark application on top of the ARTPoS by generating data packets periodically. The ARTPoS is configured to perform the radio and transmission power selection in each period (i.e., 10s) based on the measured PRR and throughput of the ZigBee and WiFi links. If the then-active radio and transmission power setting is found to be the best-suited, it is retained; else the ARTPoS switches to a new best-suited setting. Non-overlapping channels are used for ZigBee and WiFi to avoid interference. A power meter from Monsoon Solutions [26] is connected to the sender to measure the power consumption. Radios are turned off after the last transmission in each period and the unselected one is kept off to reduce power consumption for both our approach and baselines. If both radios are selected for use, packets are partitioned based on their throughput ratio, allowing the platform to sleep earlier and save energy. Due to the lack of a baseline that jointly optimizes the selection of both radio and transmission power, we extend the ART [22], a practical state-of-the-art transmission power control approach designed for ZigBee, and create three baselines: one with only ZigBee radio on running ART (*ART-ZigBee*), one with only WiFi radio

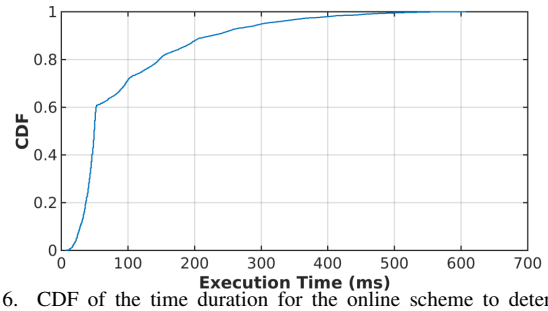


Fig. 6. CDF of the time duration for the online scheme to determine the optimal radio and transmission power.

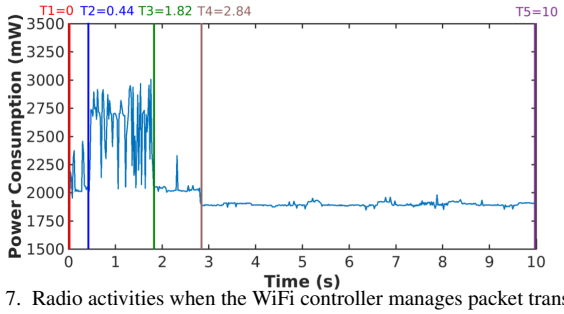


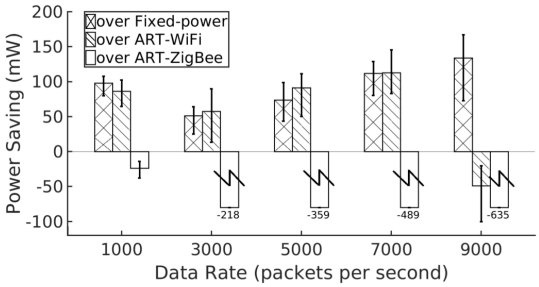
Fig. 7. Radio activities when the WiFi controller manages packet transmission in a 10s period.

on running ART (*ART-WiFi*), and one with both radios on operating at their default powers (*Fixed-power*).

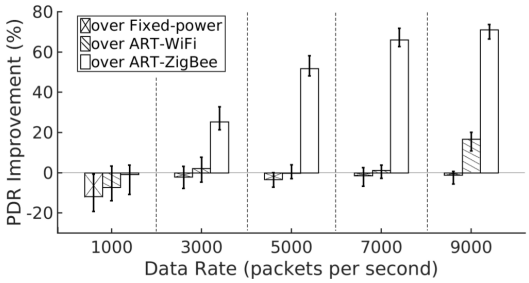
A. Micro-Benchmark Experiments

We first evaluate the time duration taken by the designed optimal online scheme to select the best radio(s) and minimum needed transmission power(s). We record the time of the events when the input is fed into the optimizer and the output (i.e., radio and transmission power selection) is generated. For this experiment, we repeat the measurement 10,000 times, using randomly generated inputs, on our 1.2GHz 64-bit quad-core ARMv8 CPU platform. Fig. 6 shows the cumulative probability density (CDF) of the time duration of the 10,000 runs. As shown in Fig. 6, the median execution time is $49ms$ (consuming $13.5mJ$ more energy than CPU idling), while 90% and 99% of the experimental runs finish within less than $225ms$ and $456ms$, respectively.

We also measure the time duration and energy consumption of other key operations in ARTPoS. Fig. 7 shows an example power consumption trace where the WiFi controller switches on the WiFi radio, transmits 1000 packets, and then switches off the radio. The platform takes $T_2 - T_1 = 0.44s$ and consumes $0.92J$ of energy to turn on the radio and set its transmission power. Transmitting 1000 packets takes $T_3 - T_2 = 1.38s$, while turning off the radios takes $T_4 - T_3 = 1.02s$. The platform consumes $3.60J$ and $2.07J$ of energy to transmit the data and turn off the radio, respectively. The radios are kept off for the rest of the period $T_5 - T_4 = 7.16s$. These results demonstrate the efficiency of the optimizer and the radio controllers, as well as the advantage of turning the radios off after transmissions in each period, and also illustrate the significant need of developing new low-power platforms for IoT applications to achieve lower baseline power consumption.



(a) Power consumption improvement over baselines



(b) PDR improvement over baselines

Fig. 8. Power consumption and PDR differences between ARTPoS and three baselines (Fixed-power, ART-WiFi, and ART-ZigBee) at different data rates.

B. Impact on Power Consumption and Link Reliability

To understand the effectiveness of our ARTPoS and its impact on power consumption and link reliability, we performed a set of experiments comparing ARTPoS's performance with three baselines. We performed the four experimental runs, respectively with ARTPoS, Fixed-power, ART-WiFi, and ART-ZigBee, in a round robin fashion to minimize the temporal effects of the dynamic wireless environment (for fair comparison). Fig. 8 shows the power consumption and packet delivery rate (PDR) comparisons between our ARTPoS and the three baselines. To explore ARTPoS's performance under different traffic demands, we repeated the experiments by controlling the application to generate data at different rates. Under each data rate and approach, we repeat the experiments five times and present the confidence intervals in Fig. 8.

As shown in Fig. 8(a), ARTPoS reduces the average power consumption by $98.2mW$ and $86.3mW$ over Fixed-power and ART-WiFi, respectively, when the data rate is 1000 $packets/period$. Similarly, ARTPoS achieves significant power savings over Fixed-power and ART-WiFi at higher data rates ($51.4mW$ and $57.5mW$ at 3000 $packets/period$, $73.7mW$ and $91.2mW$ at 5000 $packets/period$, $111.9mW$ and $112.7mW$ at 7000 $packets/period$). As a comparison for power saving values, the CC2650 radio consumes $27.3mW$ power when transmitting at $5dBm$ [3]. ARTPoS consumes $23.9mW$ more power than ART-ZigBee at the lowest data rate since it needs to occasionally turn on the WiFi radio to measure its channel condition. It is to be noted that ARTPoS consumes more power than ART-ZigBee but ART-ZigBee is not able to to deliver satisfactory PDRs at high data rates because of the ZigBee's limited bandwidth (i.e., the

median PDRs under ART-ZigBee are 68.7%, 44.6%, 31.0%, and 25.0% when the data rate is 3000, 5000, 7000, and 9000 $packets/period$, respectively, i.e., significantly inferior to ARTPoS (as seen from Fig. 8(b)). Neither WiFi nor ZigBee alone can support the data rate of 9000 $packets/period$, while our ARTPoS provides satisfactory PDRs by bundling the WiFi and ZigBee radios. The experimental results show that ARTPoS can effectively reduce the energy consumption, while maintaining satisfactory link reliability.

VII. Conclusion and Future Work

Given the dynamic nature of communication in IoT (e.g., mobile/vehicular and industrial wireless sensor networks), a traditional one-radio-fits-all approach cannot meet the challenges under typically varying operating conditions and traffic. This paper presents the ARTPoS system that makes available multiple wireless technologies at runtime and selects the radio(s) and transmission power(s) most suitable for the current conditions. New power and PRR modeling approaches are presented, which allow the system to proactively adapt to large variations in power consumption and link reliability measurements. This is followed by the development of a light-weight online optimization scheme, based on a sense-classify-predict-search process. Experimental evaluations of the thus formulated online optimization scheme, and its comparison with different baselines, show that ARTPoS can remarkably reduce the power consumption, while maintaining satisfactory link reliability. We plan to integrate ARTPoS with the low power listening technique to support efficient duty cycling and enable model updating at runtime as our future works. In addition, we are also currently investigating approaches to extend this fundamental radio/transmission selection technique from a one-to-one communication to a many-to-many/network-scale communication framework involving gateways. Decomposed problem formulations and decentralized decision-making are expected to serve as two other core elements in facilitating this important next step in this research.

Acknowledgment

This work was supported by the NSF through grant CRII-1657275 (NeTS).

References

[1] M. Sha, R. Dor, G. Hackmann, C. Lu, T.-S. Kim, and T. Park, "Self-Adapting MAC Layer for Wireless Sensor Networks," in *RTSS*, 2013.

[2] M. P. Andersen, G. Fierro, and D. E. Culler, "System Design for a Synergistic, Low Power Mote/BLE Embedded Platform," in *IPSN*, 2016.

[3] TI CC2650 SimpleLink Multi-Standard 2.4 GHz Ultra-Low Power Wireless MCU. [Online]. Available: http://www.ti.com/product/CC2650

[4] Raspberry Pi 3 Model B. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[5] K. Habak, K. A. Harras, and M. Youssef, "Bandwidth Aggregation Techniques in Heterogeneous Multi-homed Devices," *Computer Networks*, vol. 92, no. P1, 2015.

[6] Y. Lim, Y. Chen, E. M. Nahum, D. Towsley, R. J. Gibbens, and E. Cecchet, "Design, Implementation, and Evaluation of Energy-Aware Multi-Path TCP," in *CoNEXT*, 2015.

[7] A. Nikravesh, Y. Guo, F. Qian, Z. M. Mao, and S. Sen, "An In-depth Understanding of Multipath TCP on Mobile Devices: Measurement and System Design," in *MobiCom*, 2016.

[8] C. Tsao and R. Sivakumar, "On Effectively Exploiting Multiple Wireless Interfaces in Mobile Hosts," in *CoNEXT*, 2009.

[9] D. Bui, K. Lee, S. Oh, I. Shin, H. Shin, H. Woo, and D. Ban, "GreenBag: Energy-efficient Bandwidth Aggregation For Real-time Streaming in Heterogeneous Mobile Wireless Networks," in *RTSS*, 2013.

[10] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang, "Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones," in *CODES+ISSS*, 2010.

[11] F. Xu, Y. Liu, Q. Li, and Y. Zhang, "V-edge: Fast Self-Constructive Power Modeling of Smartphones Based on Battery Voltage Dynamics," in *NSDI*, 2013.

[12] N. Ding, D. Wagner, X. Chen, A. Pathak, Y. C. Hu, and A. Rice, "Characterizing and Modeling the Impact of Wireless Signal Strength on Smartphone Battery Drain," in *SIGMETRICS*, 2013.

[13] A. Nika, Y. Zhu, N. Ding, A. Jindal, Y. C. Hu, X. Zhou, B. Y. Zhao, and H. Zheng, "Energy and performance of smartphone radio bundling in outdoor environments," in *WWW*, 2015.

[14] X. Chen, N. Ding, A. Jindal, Y. C. Hu, M. Gupta, and R. Vannithamby, "Smartphone Energy Drain in the Wild: Analysis and Implications," in *SIGMETRICS*, 2015.

[15] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications," in *IMC*, 2009.

[16] A. Schulman, V. Navda, R. Ramjee, N. Spring, P. Deshpande, C. Grunewald, V. Padmanabhan, and K. Jain, "Bartendr: A Practical Approach to Energy-aware Cellular Data Scheduling," in *MobiCom*, 2010.

[17] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A Close Examination of Performance and Power Characteristics of 4G LTE Networks," in *MobiSys*, 2012.

[18] Q. Peng, M. Chen, A. Walid, and S. Low, "Energy Efficient Multipath TCP for Mobile Devices," in *MobiHoc*, 2014.

[19] S. Nirjon, A. Nicoara, C.-H. Hsu, J. P. Singh, and J. A. Stankovic, "MultiNets: A System for Real-Time Switching between Multiple Network Interfaces on Mobile Devices," *ACM Transactions on Embedded Computing Systems*, vol. 13, no. 4S, 2014.

[20] S. Lin, F. Miao, J. Zhang, G. Zhou, L. Gu, T. He, J. A. Stankovic, and G. P. S. Son, "ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks," in *ACM Transactions on Sensor Networks*, vol. 12, no. 1, 2016.

[21] S. Lin, G. Zhou, K. Whitehouse, Y. Wu, J. Stankovic, and T. He, "Towards Stable Network Performance in Wireless Sensor Networks," in *RTSS*, 2009.

[22] G. Hackmann, O. Chipara, and C. Lu, "Robust Topology Control for Indoor Wireless Sensor Networks," in *SenSys*, 2008.

[23] Y. Fu, M. Sha, G. Hackmann, and C. Lu, "Practical Control of Transmission Power for Wireless Sensor Networks," in *ICNP*, 2012.

[24] M. Burkhart and P. V. Rickenbach, "Does Topology Control Reduce Interference?" in *MobiHoc*, 2004.

[25] Y. Gao, J. Hou, and H. Nguyen, "Topology Control for Maintaining Network Connectivity and Maximizing Network Capacity under the Physical Model," in *INFOCOM*, 2008.

[26] Power Monitor by Monsoon Solutions Inc. [Online]. Available: https://www.msoon.com/LabEquipment/PowerMonitor/

[27] Raspbian Linux. [Online]. Available: https://www.raspbian.org/

[28] Contiki. [Online]. Available: http://www.contiki-os.org/

[29] M. D. McKay, R. J. Beckman, and W. J. Conover, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979. [Online]. Available: http://www.jstor.org/stable/1268522

[30] J.-S. Park, "Optimal Latin-hypercube Designs for Computer Experiments," *Journal of Statistical Planning and Inference*, vol. 39, no. 1, pp. 95 – 111, 1994.

[31] R. Tyrrell Rockafellar and J. O. Royset, "Engineering Decisions under Risk Averseness," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, vol. 1, no. 2, p. 04015003, 2015.

[32] R. T. Rockafellar and S. Uryasev, "Conditional Value-at-risk for General Loss Distributions," *Journal of banking & finance*, vol. 26, no. 7, pp. 1443–1471, 2002.

[33] G. Cardillo, "Four Parameters Logistic Regression - There and Back Again," online, April 2013.

[34] R. Hemmecke, M. Koppe, J. Lee, and R. Weismantel, *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*. Springer, 2008, ch. Nonlinear Integer Programming.