

Robust Optimal Selection of Radio Type and Transmission Power for Internet of Things

DI MU, YUNPENG GE, and MO SHA, State University of New York at Binghamton, USA
STEVE PAUL, NIRANJAN RAVICHANDRA, and SOUMA CHOWDHURY, University at Buffalo, USA

Research efforts over the last few decades produced multiple wireless technologies, which are readily available to support communication between devices in various dynamic Internet of Things (IoT) and robotics applications. However, single radio technology can hardly deliver optimal performance across all critical quality of service (QoS) dimensions under the typically varying environmental conditions or under varying distance between communicating nodes. Using a single wireless technology therefore falls short of meeting the demands of varying workloads or changing environmental conditions. Instead of pursuing a one-radio-fits-all approach, we design *ARTPoS*, an *Adaptive Radio and Transmission Power Selection* system, which makes available at runtime multiple wireless technologies (e.g., WiFi and ZigBee) and selects the radio(s) and transmission power(s) most suitable for the current conditions and requirements. The principal components of ARTPoS include new empirical models of power consumption and packet reception ratio (the latter can also be refined online) and online optimization schemes. We have implemented our system and evaluate it on the physical testbed consisting of our new embedded platforms with heterogeneous radios. Experimental results show that ARTPoS can significantly reduce the power consumption, while maintaining desired link reliability, compared to standard baselines.

CCS Concepts: • **Computer systems organization** → **Sensor networks**; • **Networks** → **Link-layer protocols**; **Network performance modeling**;

Additional Key Words and Phrases: Internet of Things, heterogeneous radios, optimization, power efficiency, quality of service

ACM Reference Format:

Di Mu, Yunpeng Ge, Mo Sha, Steve Paul, Niranjan Ravichandra, and Souma Chowdhury. 2019. Robust Optimal Selection of Radio Type and Transmission Power for Internet of Things. *ACM Trans. Sensor Netw.* 1, 1, Article 1 (February 2019), 25 pages. <https://doi.org/0000001.0000001>

1 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. Single radio technology can hardly deliver optimal performance in all desirable quality of service (QoS) dimensions, especially under varying environmental conditions. For instance, WiFi

Authors' addresses: Di Mu; Yunpeng Ge; Mo Sha, State University of New York at Binghamton, 4400 Vestal Parkway East, Binghamton, NY, 13902, USA, {dmu1,yge6,msha}@binghamton.edu; Steve Paul; Niranjan Ravichandra; Souma Chowdhury, University at Buffalo, 246 Bell Hall, Buffalo, NY, 14260, USA, {stepaup,nravicha,soumacho}@buffalo.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

1550-4859/2019/2-ART1 \$15.00
<https://doi.org/0000001.0000001>

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 [32]. **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 [32]. 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 [1] supports Bluetooth low energy (BLE) and ZigBee and uses a 32 bit low-power microcontroller with the duty cycling capability. TI CC2650 [34] integrates two radios (i.e., ZigBee and BLE) on a single chip. Raspberry Pi 3 model B [28] uses a Broadcom single-chip radio supporting both WiFi and BLE. IOT-Gate-iMX7 [20] is an industrial IoT gateway, which supports 4G/LTE, WiFi, Bluetooth, and Zigbee. The Zifi device [39] support both WiFi and ZigBee. 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 offline 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 problem¹ and develop two practical (lightweight) online solutions; the latter solution uniquely allows online updating of the link reliability models, to enable adapting to runtime environments that deviate from the offline settings that were used to train the models.

¹In this paper, we focus on minimizing the energy consumption on the link level and the sender side (IoT end devices), since the IoT gateways are usually not or much less energy-constrained.

- 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.

We show that our ARTPoS implementations clearly outperform two baselines (Fixed-power and ART-WiFi) in terms of power consumption, while delivering similar link reliability. Expectedly, ART-ZigBee registers the lowest power consumption, but fails to provide any meaningful link reliability for all data rates above 1000 packets/period. Importantly, this advantage of the ARTPoS implementations is shown to hold under various indoor and outdoor settings, and with and without interference. Lastly, we show that the newer ARTPoS-irp version is able to exploit its special (runtime) model adaptation capacity to provide, on average, a 3.7% better packet delivery rate and 13.5mW power savings, over the original ARTPoS implementation.

The remainder of the paper is organized as follows. Section 2 reviews related work and Section 3 introduces our ARTPoS design. Section 4 presents the power consumption and link reliability modeling and Section 5 introduces our problem formulation and solution strategy. Section 6 presents our experimental evaluation. Section 7 concludes the paper.

2 RELATED WORKS

Bandwidth aggregation for a device with multiple network interfaces has been studied extensively in the literature and many techniques are readily available [12]. 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 [17, 24]. There has also been increasing interest in studying the energy-aware bundling or switching between WiFi and 3G/4G radios on smartphones [4, 35]. 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 [7, 9, 23, 37, 38] and WiFi/3G/LTE [2, 14, 31] and developing radio switching or bundling approaches [17, 23–25, 27]. These existing approaches are either limited to mainly WiFi and 3G/4G on smartphone platforms or unaware of transmission power control, thus they are not directly applicable to support energy-efficient data transfer using heterogeneous radios in various IoT embedded platforms. 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) [18, 19] or direct link quality metrics such as packet reception ratio (PRR) and packet error rate (PER) [10, 13] have been used to measure the link quality. Heuristics [5, 11, 13] and control-theoretic approaches [10, 18, 19] have been applied to achieve the desirable link quality by controlling the transmission power at runtime. These existing approaches, designed to select the transmission power of 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

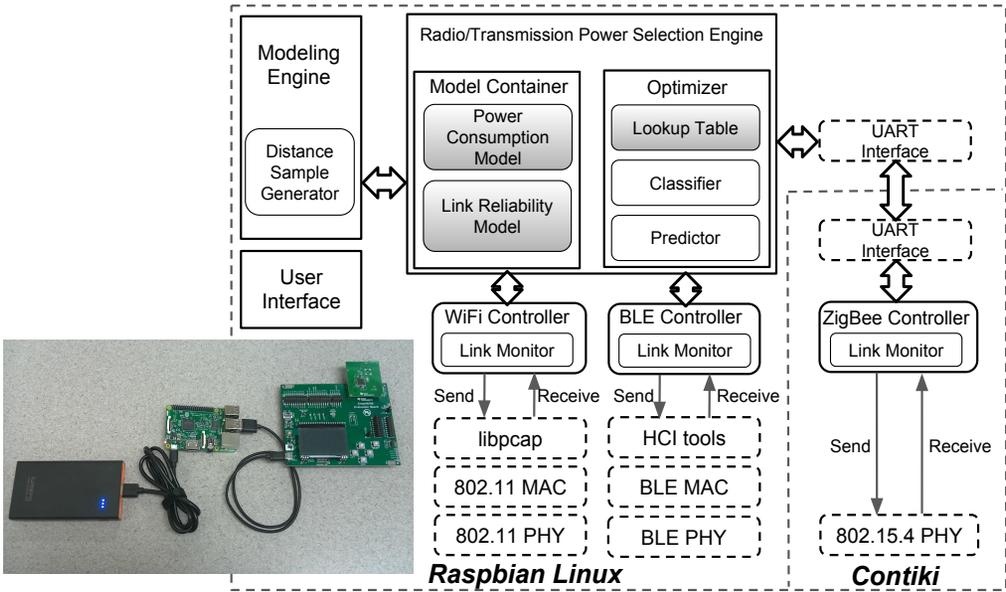


Fig. 1. System architecture.

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.

3 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 3.1). 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 3.2). 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 3.3).

To support the realization of ARTPoS, we have built a new embedded platform (as shown in Fig. 1) with heterogeneous radios consisting of WiFi, ZigBee, and BLE by instrumenting a Raspberry Pi 3 Model B [28] with a TI CC2650 Development Kit [34], which is connected to the Raspberry Pi through a USB port. Raspberry Pi integrates a Broadcom BCM43438 single chip radio processor supporting WiFi and BLE, while CC2650 is the core wireless MCU supporting ZigBee and BLE on CC2650 Development Kit (currently, we use the BLE radio on Raspberry Pi since the Contiki has not yet implemented the BLE stack in its master branch). The integrated emulator (XDS100v3) on the CC2650 Development Kit enables the communication between the Raspberry Pi and the CC2650 MCU through UART. To power the device, we use a USB battery to which a Monsoon power meter [33] is connected to measure the power consumption.

We have realized ARTPoS in Raspbian Linux [29], a Debian based Linux system for Raspberry Pi, and Contiki [8], an operating system for low-power wireless IoT devices. To support WiFi, our

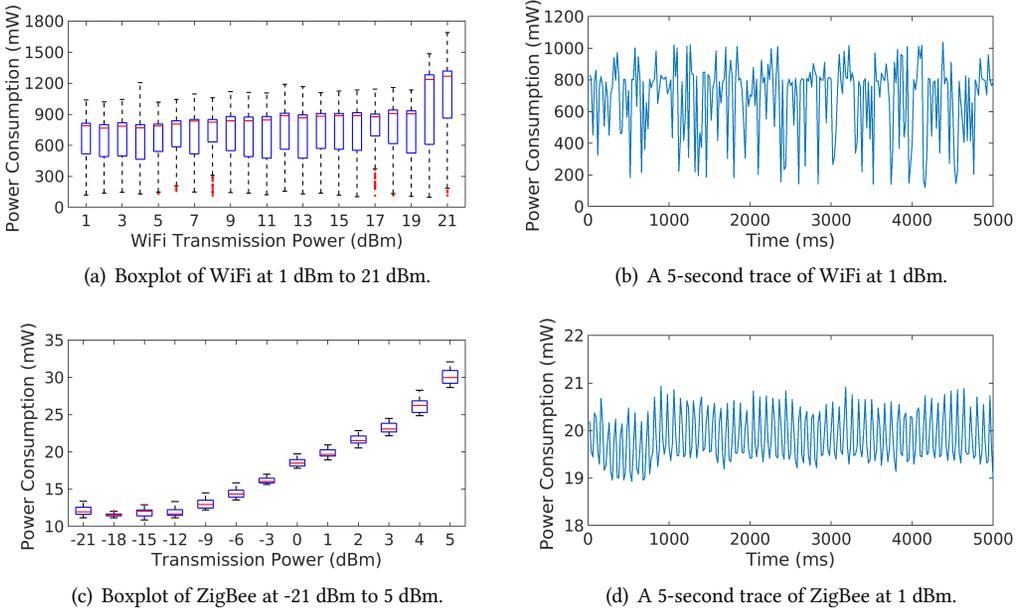


Fig. 2. Radio power consumptions when WiFi and ZigBee turn on respectively and transmit at the maximum speed. The traces are measured by a Monsoon power meter [33]. In boxplot, central red mark in box indicates median; bottom and top of box represent the 25th percentile (q_1) and 75th percentile (q_2); 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.

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.

3.1 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 radio power consumptions when the WiFi and ZigBee radios on our embedded platform turn on respectively and transmit at the maximum speeds at all available transmission power settings. As shown in

Fig. 2(a), the median power consumption increases from $789mW$ to $905mW$ to $1269mW$ when WiFi is on and the transmission power increases from $1dBm$ to $19dBm$ to $21dBm$, while the median power consumption increases from $11.9mW$ to $18.5mW$ to $30mW$ when ZigBee is on and the transmission power increases from $-21dBm$ to $0dBm$ to $5dBm$ as shown in Fig. 2(c). Large variances can be seen in the boxplot in Fig. 2(b) and Fig. 2(d), which show the 5-second power measurements when WiFi and ZigBee transmit at $1dBm$, respectively. The large variance is caused by the power consumption differences when the radio hardware is at different states, making the first statistical moments (e.g., mean or median) unsuitable to estimate the radio power consumption.

The Modeling Engine also generates the link reliability model based on the PRR measurements at different 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 statistical design of experiments approach, commonly used in Engineering optimization, is employed 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 emerging cloud robotic and multi-robot 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 [21]. 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, $(0, 1/n), (1/n, 2/n), \dots, (1 - 1/n, 1)$. We use the optimal LHS implementation, which maximizes the minimum Euclidean distance between the samples [26]. 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 4.2 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.

3.2 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 5 will discuss the problem formulation and optimization in detail.

3.3 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.

4 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 5.2).

4.1 Power Consumption Modeling

The measurements from Section 3.1 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 consumption 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 3.1), 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. Secondly, since battery capacity is currently a critical bottleneck in most wireless IoT and embedded system devices, and radios can be a major contributor to power consumption in such devices, we argue that energy over-expenditure (and the uncertainty associated with it) should be perceived as a *risk* – one that can lead to significantly reduced device uptime and/or frequent switching to low performance modes for the concerned device. Hence, we propose to use the notion of **s-risk** [36], to provide a robust or *uncertainty-aware* scalar measure of the risk associated with this expense under any given radio setting.

The notion of s-risk, also known as “conditional-value-at-risk”, originated in the Finance domain [15, 30]. Among risk metrics, the s-risk model is well established as a more generalizable model [15] (requires minimal assumptions w.r.t. the underlying process), and thus considered to be a suitable choice in this nascent application setting. 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-averse 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:

$$S_\gamma(E_w(V, p_w)) = \frac{1}{(1 - \gamma)N} \sum_{v_k \in \Gamma} [E_w(V, p_w)_k] \quad (1)$$

$\Gamma = \text{set of the highest } (1 - \gamma)100\% \text{ values of } E_w$

It is readily evident from Eq. 1 that higher values of γ leads to greater aversion of (energy expenditure) risk or more conservative decisions, in determining the optimal radio settings (optimization approach is described in the next section). From a practical perspective, this “risk-averse parameter” γ can be designed to be adaptive to the battery state – e.g., the system will use increasingly greater value of γ when the device goes from normal to low and low to critical battery states. Such heuristics could preserve operational feasibility albeit at the cost of reduced data transfer rates. Owing to its ability to consider tails of probability distributions (with the help of higher values of γ) 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: $p_z \in \{-6, -3, 0, 1, 2, 3, 4, 5\}$ dBm;
- s-risk values of the platform power consumption with only WiFi On (S_w) at the following different transmission power settings: $p_w \in \{1, 2, \dots, 20, 21\}$ 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.

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, and a single linear regression is used in the case of ZigBee. 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:

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

The s-risk models were fitted based on actual power data (collected in the offline experiments). Illustration of the training data and resulting curve fits can be found in [], demonstrating the predictability of the s-risk parameter that captures the energy expenditure risk.

4.2 Link Reliability (PRR) Modeling

The PRR measurements from Section 3.1 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 5).

We observe large variations in PRR measurements, especially when the links are in the transitional region. 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.

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 to model the “PRR-p” relationships between PRR values and transmission power settings. An implementation, called L4P [6], 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. 3(a) and 3(b). It is readily evident from Fig. 3 that while 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

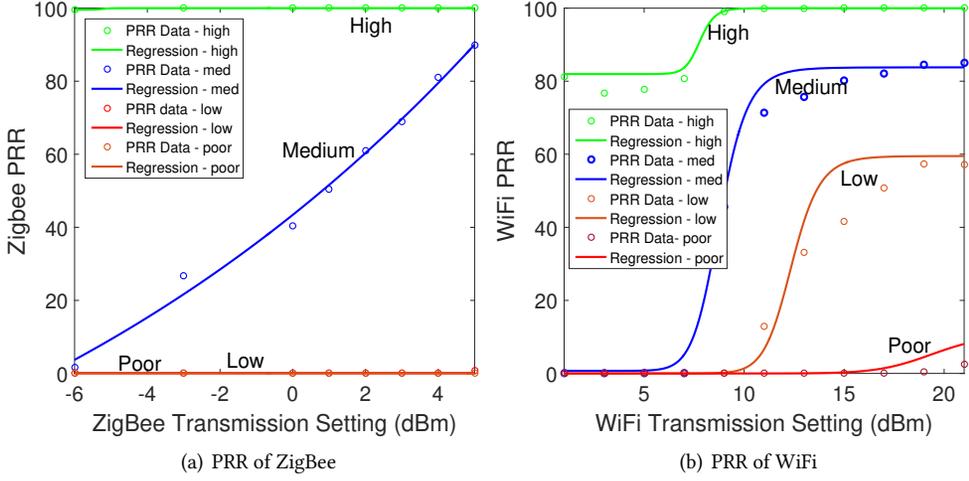


Fig. 3. Regression plots of PRR as functions of radio transmission settings; PRR data segregated into *poor*, *low*, *medium*, and *high* states.

power, in response to the need to increase PRR, over the entire range of available transmission power settings.

5 OPTIMIZATION

5.1 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: **restrict packet loss to within a small/acceptable bound, while platform power consumption attributed to the radios is 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, that given the current state of the radio performance chooses the optimum action. 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, at given transmission settings (p_w and p_z), will be respectively represented by $r_w(p_w)$ and $r_z(p_z)$ or simply as r_w and r_z , where $0 \leq r_w, r_z \leq 1$; 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:

$$G_{w,z}(p_w, p_z) = h_w r_w(p_w) + h_z r_z(p_z) \quad (4)$$

If only one of the radios is on, the aggregated goodput 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 approximated by:

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

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 ($\min(1, D/G_{w,z})$). When the WiFi is off, $E_w(\text{Off}) = E_p$ and $r_w(\text{Off}) = 0$; similarly, when the ZigBee is off, $E_z(\text{Off}) = E_p$ 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 enabled in 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:

$$\begin{aligned} & \min_{p_w, p_z} f_E(p_w, p_z, h_w, h_z) \\ & \text{s.t.} \\ & 1 - \min\left(1, \frac{D}{G_{w,z}(p_w, p_z)}\right) \geq \epsilon \end{aligned} \quad (6)$$

where

$$\begin{aligned} p_w & \in \{\text{Off}, 1, 2, \dots, 20, 21\} \\ p_z & \in \{\text{Off}, -6, -3, 0, 1, 2, 3, 4, 5\} \end{aligned}$$

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. 3). 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 [16], 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. Instead of formulating the optimization under uncertainty as a classical (computationally costly) reliability based optimization problem, uncertainties are addressed apriori using the combination of s-risk measures of power consumption and regression modeling of PRR and s-risk measures (as presented in Section 4). The online execution time of solving this optimization problem is presented in Section 6.1.

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.

5.2 Study on the Impact of Uncertainties

An offline optimization study is set up to investigate how the radio selection is affected by the environmental uncertainties (that cause ill-predictable PRR variations) and systemic uncertainties (that cause power consumption variations). Hence, in this study, we deliberately neither employ any smoothing operation on the empirical data nor use the regression models developed in Section 4.

Optimization is performed for different sample combinations of distance between sender and receiver (X) and data rate (D), where $X \in \{10, 20, 30, \dots, 150\}$ m and $D \in \{25, 50, 75, \dots, 150\}$ 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 3); a risk-averse 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 only a small set of radio settings are available – i.e., 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. 4. For illustration purposes, the results for three data rates (150, 175, and 200 packets/s) are shown. In Fig. 4, 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. 4 indicates whether the optimal radio setting succeeded (= 1) or failed (= 0) to satisfy the data-rate/goodput ratio constraint (in Eq. 6).

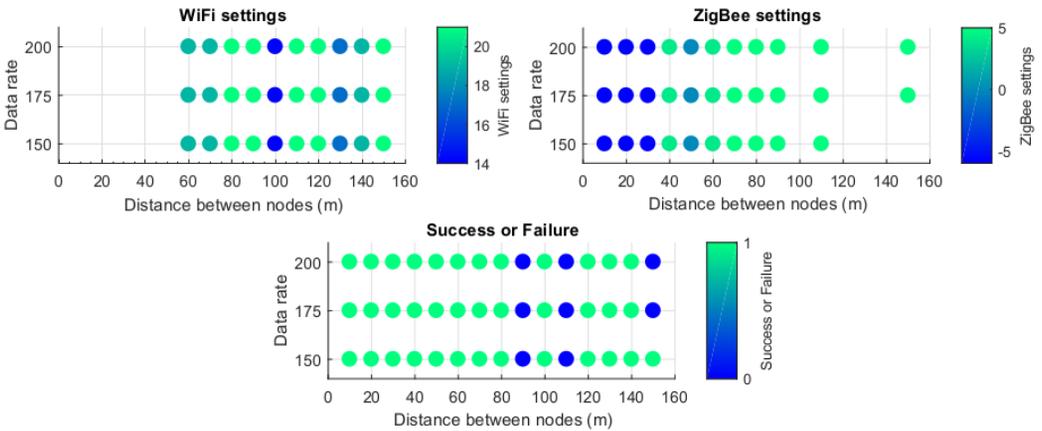


Fig. 4. Offline study (without smoothing measures or regression models): Top: Optimal transmission power settings of WiFi and Zigbee when operating together; Bottom: success (= 1) or failure (= 0) in meeting the “data-rate/goodput” ratio constraint for different distance and data rate combinations.

The impact of noise/uncertainty of the empirical data (driving the nominal decisions) is apparent in the offline optimization results as shown in Fig. 4. 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 4, and 2) the design of the two *online* algorithms that use these models to offer robust solutions, which will be described in the next sub-sections.

5.3 Fast Online Optimization (ARTPoS)

The ARTPoS online optimization artifact 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 and typical reliability-based optimization techniques. These latter 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 4, 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:

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

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 (p_z^j, p_w^i); it is derived from the s-risk measures of power consumption (Section 4.1), where the s-risk value of the platform baseline power consumption with both 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, at the corresponding transmission settings. These state values are given by the PRR regression functions developed in Section 4.2 (Figs. 3(a) and 3(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 algorithm/program, that uses this lookup table, is designed as a four-step process: **sense**→**classify**→**predict**→**search**. A pseudocode of this runtime program is given in Algorithm 1, and the individual steps are described below.

Algorithm 1 *ARTPoS*

```

1: Read: x, y, prrw, prrz, Dt, hw, hz; ▷ Input of current state variables from receivers, Sense stage
2: function FINDNEAREST(prrx, x)                                ▷ Classify stage
3:   if min(|prrArray – prrx|) > prrLimit then
4:     prrState = min(|prrArray – prrx|);
5:     refitArray = REFITDATA(prrArray, x, prrx);                ▷ ARTPoS-new refit
6:     return min(refitArray) satisfying |refitArray – prrx|;
7:   else
8:     return min(prrArray) satisfying |prrArray – prrx|;
9:   end if
10: end function
11: function SEARCHFUNC(x, y, prrw, prrz, Dt, hw, hz);
12:   return Dt/((hw * FINDNEAREST(prrw, x)) + (hz * FINDNEAREST(prrz, y)));
13: end function
14: exclude ← powerTable[x+1, y+1];                             ▷ Power values to omit in search
15: for (a, b) in powerTable do;
16:   if powerTable[a, b] in exclude then
17:     continue;
18:   else
19:     currentSettings = SEARCHFUNC(a, b, prrw, prrz, Dt, hw, hz);    ▷ Search stage
20:     if currentSettings > 0.9 then
21:       continue;
22:     else
23:       append (a, b) to feasibleSettings;
24:       append powerTable[a, b] to feasiblePower;
25:     end if
26:   end if
27: end for
28: return feasibleSettings[index of min(feasiblePower)]           ▷ Predict stage;

```

Algorithm Nomenclature:

prrw, prrz: Packet reception ratio of WiFi and ZigBee radios.

Dt: Data rate.

prrArray: Array of pre-generated PRR values that *prrw* and *prrz* are classified against.

refitArray: Array of PRR values including new *prrw* and *prrz* that violates the *prrLimit* threshold (only in ARTPoS-new).

SEARCHFUNC: function implementing *Search* stage to evaluate the datarate to goodput ratio.

powerTable: pre-computed lookup table from *Predict* stage.

feasibleSettings: array of WiFi and Zigbee dBm values that satisfy search conditions.

feasiblePower: array of power values (from lookup table) that satisfy search conditions.

- **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 averse 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. 3(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:

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

where $i = 1, 2, \dots, 22; j = 1, 2, \dots, 9$

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:

$$r_{w,12}^t = r_{w,12}^{\text{low}} + \frac{r_{w,14}^t - r_{w,14}^{\text{low}}}{r_{w,14}^{\text{medium}} - r_{w,14}^{\text{low}}} \left(r_{w,12}^{\text{medium}} - r_{w,12}^{\text{low}} \right) \quad (9)$$

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:

$$\begin{aligned} & \min_{i,j} f_{E,ij}^t \\ & \text{subject to } 1 - \frac{D^t}{h_w^t r_{w,ij}^t + h_z^t r_{z,ij}^t} \geq \epsilon \end{aligned} \quad (10)$$

where $i = 1, 2, \dots, 22; j = 1, 2, \dots, 9$

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, $(p_z^j, p_w^i) = (5, 21)$ 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 ARTPoS online optimization is 49ms on an ARM processor. Section 6.1 will present our micro-benchmark evaluations in detail.

5.4 Online Optimization with Insitu Refinement of PRR Models (ARTPoS-irp)

ARTPoS-irp is our first step towards a system that is also capable of judging how far the real environment (during operation) deviates from the offline training environment, and adapts its models online in order to provide more reliable decisions. The modified algorithm mainly extends the *Classify* step in the ARTPoS system (see Section 5.3) with the aim of increasing the reliability with which the radio's performance ((p_w^t, r_w^t) and (p_z^t, r_z^t)) is classified into the *high*, *medium*, *low*, *poor* state categories. This is achieved by identifying significant deviations (from the offline trends) and responding to it by dynamically refitting the PRR regression models used in the *Classify* step. As the radio's performance now seeks to be reflective of the environment in which the system operates, the algorithm is expected to become more robust in its adaptation.

The refit is invoked by consistent over/underestimation of the PRR state compared to the classified curves (given in Fig. 3). In *ARTPoS-irp*, the measure of over/underestimation is through the observation of the difference between the measured PRR value and the PRR value at the given transmission setting (dBm) based on the state it is classified under. If the difference exceeds a certain threshold, $r_{TH} = 0.3$ for n consecutive time steps, the refit is performed; for example, perform refit of the WiFi PRR curve if:

$$|r_{w,i}^{\text{classified}} - r_{w,i}^{\text{meas}}| > r_{TH}, \text{ for } n \text{ consecutive time steps} \quad (11)$$

Here, n is set at 5, and $r_{w,i}^{\text{classified}}$ is the PRR given by the curve into which the current state has been classified, and $r_{w,i}^{\text{meas}}$ is the online measured PRR value. If the difference does not exceed the threshold, the refit is not invoked and ARTPoS-irp behaves identically to ARTPoS.

Next, the violating PRR values are added to the existing dataset, and the refit is performed using the logistic regression [6] described in Eq. 3. In order to prevent ever-growing size of the dataset during operation, a forgetting strategy can be used (after a threshold size is exceeded) where every time new data is added for refit the oldest data at the corresponding transmission dBm can be removed from the set.

To illustrate the role played by this online updating strategy, we provide a representative example in Fig. 5. Here, the leftmost plot (Fig. 5(a)) shows the original offline trained PRR models for WiFi. The next plot (Fig. 5(b)) shows the PRR models after one round of updating invoked by new deviating data at 7dBm classified under the "*medium*" category PRR model (note that the blue curve fit, corresponding to "Model - med" has got updated). The rightmost plot (Fig. 5(c)) shows the PRR models after another round of updating, in this case invoked by new deviating data at 1dBm classified under the "*high*" category (note that the green curve fit, corresponding to "Model - high" has got updated). It is important to note from Fig. 5 that, the observed effectiveness of adapting the PRR models to the varying runtime environment is attributed to both the new online updating scheme in ARTPoS-irp and the original choice of the (logistic) regression fitting.

6 EVALUATION

To examine the efficiency of ARTPoS and ARTPoS-irp, we perform a series of experiments on our embedded platform presented in Section 3. We first measure the overhead of the key operations such as the time duration of the optimizer selecting the best radio(s) and needed transmission power(s) and the overhead attributed to turning the radio(s) On and Off. We then evaluate ARTPoS/ARTPoS-irp's impact on power consumption and link reliability, and compare their performance against

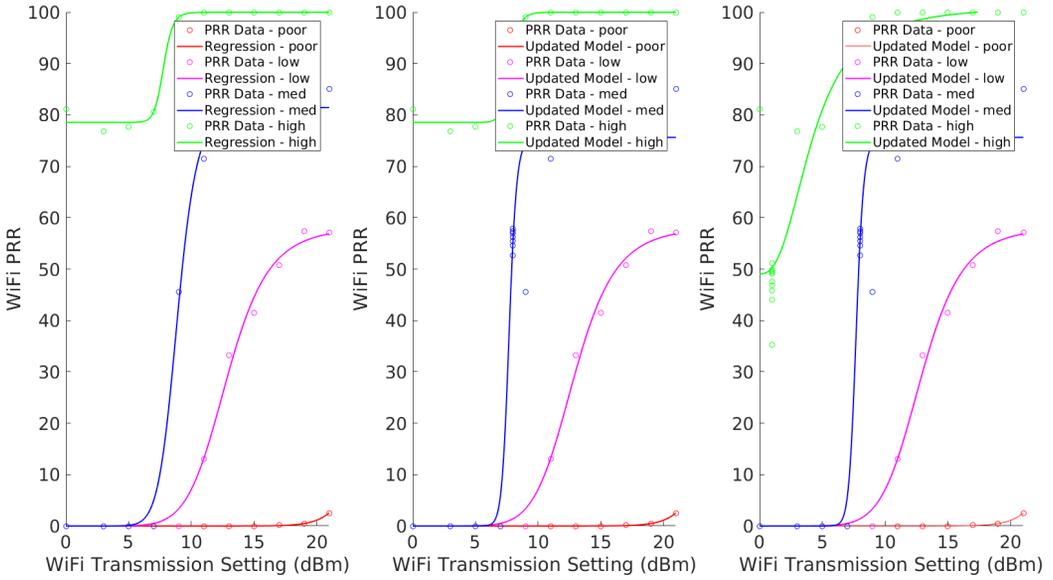


Fig. 5. WiFi PRR Models: showing how the offline trained model gets updated online in response to new PRR data that deviate from the offline fits

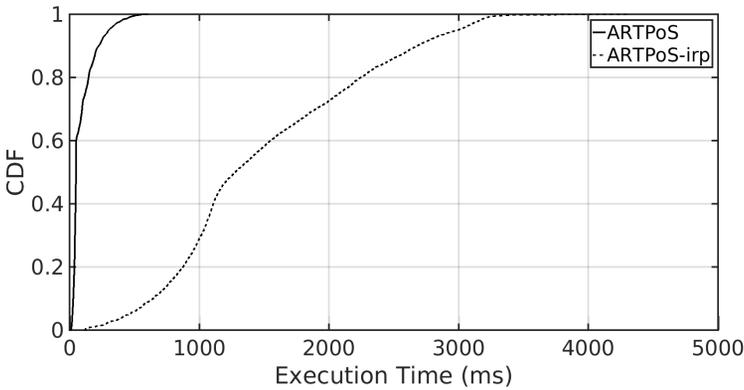


Fig. 6. CDF of the time duration for ARTPoS and ARTPoS-irp to determine the optimal radio and transmission power.

three baselines. A power meter from Monsoon Solutions [33] is connected to the sender to measure the power consumption.

6.1 Micro-Benchmark Experiments

We first evaluate the time duration taken by the two online optimal approaches 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 for both ARTPoS and ARTPoS-irp (with refit), using randomly generated inputs, on our 1.2GHz 64-bit quad-core ARMv8

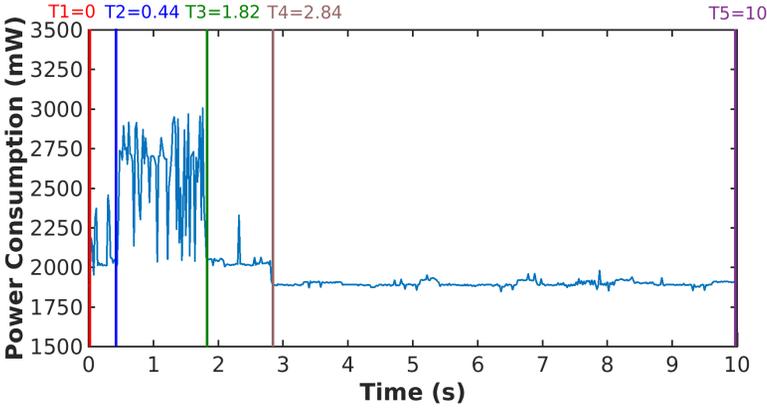


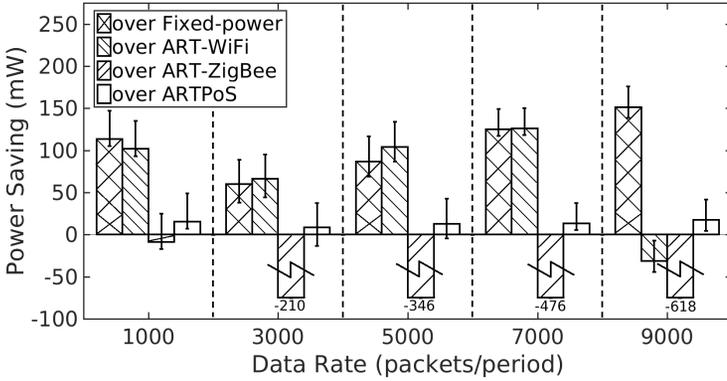
Fig. 7. Radio activities when the WiFi controller manages packet transmission in a 10s period; averaged power consumption over the first three time periods, $T_1 \rightarrow T_2$, $T_2 \rightarrow T_3$ and $T_3 \rightarrow T_4$, respectively are $2.09mW$, $2.61mW$ and $2.03mW$.

CPU platform. In order to show the difference to ARTPoS, we force ARTPoS-irp to invoke its refit every time by feeding in randomized inputs, since ARTPoS-irp's behaviour is identical to ARTPoS's without invoking its refit. The difference of the execution time between the two methods represents the time taken to perform the refit triggered by estimation errors. Figure 6 compares the cumulative probability density (CDF) of the algorithm execution time of ARTPoS and ARTPoS-irp. As shown in Fig. 6, the median execution time of ARTPoS is $49ms$ (consuming $13.5mJ$ more energy than CPU idling), where 90% and 99% of the experimental runs finish within less than $225ms$ and $456ms$, respectively. In comparison, the median execution time of ARTPoS-irp is $1273ms$, where 90% of the experimental runs finish within $2675ms$; this additional computing burden can be directly attributed to the refitting of the PRR function (via logistic regression) performed insitu in ARTPoS-irp. This burden can be alleviated by increasing the deviation threshold and/or the number of consecutive time steps for which deviation is allowed (refer Eq. 11) before invoking the refit; future work would explore how computational efficiency trades-off with energy and link reliability performance in this context.

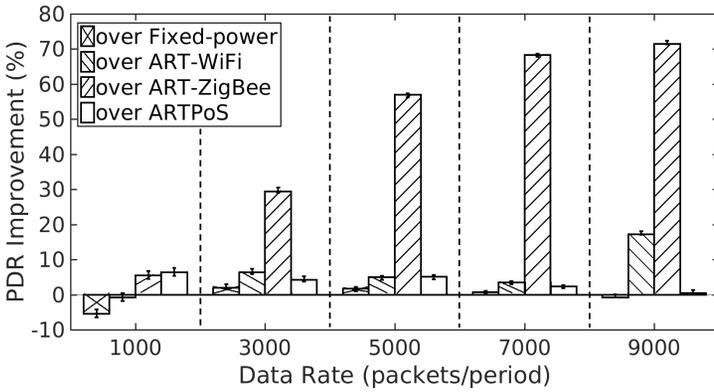
We also measure the time duration and energy consumption of other key operations in ARTPoS and ARTPoS-irp. Figure 7 shows an example power consumption trace where the WiFi controller switches On the WiFi radio, transmits 1000 packets, and then switches Off the radio. The platform takes $T_2 - T_1 = 0.44s$ 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.

6.2 Impact on Power Consumption and Link Reliability

To understand how the proposed methods impact power consumption and link reliability, we performed a set of experiments comparing the performance of ARTPoS and ARTPoS-irp with three baselines. In all experiments, we deploy a benchmark application on top of the ARTPoS and



(a) Power consumption saving provided by ARTPoS-irp over the baselines and ARTPoS



(b) PDR improvement provided by ARTPoS-irp over the baselines and ARTPoS

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

ARTPoS-irp by generating data packets periodically. ARTPoS and ARTPoS-irp are 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/ARTPoS-irp switches to a new best-suited setting. Non-overlapping channels are used for ZigBee and WiFi to avoid interference. Radios are turned Off after the last transmission in each period and the unselected one is kept Off to reduce power consumption for our approaches and the 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 [13], 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 on running ART (*ART-WiFi*), and one with both radios on operating at their default powers, i.e., 21dBm for WiFi and 5dBm for ZigBee (*Fixed-power*).

We performed five experimental runs, respectively with Fixed-power, ART-WiFi, ART-ZigBee, ARTPoS, and ARTPoS-irp, in a round robin fashion to minimize the temporal effects of the dynamic wireless environment (for fair comparison). Figure 8 shows the power consumption and packet delivery rate (PDR) comparisons between our approaches and the baselines. To explore ARTPoS-irp'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.

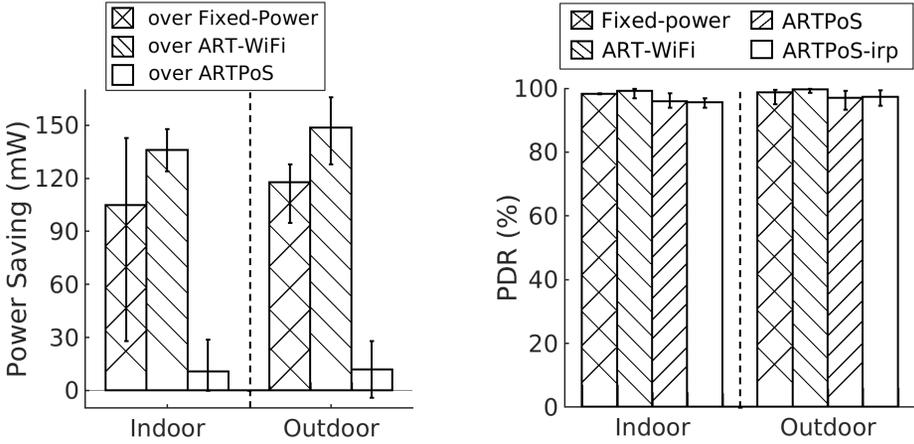
As shown in Fig. 8(a), both of our proposed methods, ARTPoS and ARTPoS-irp, provide significant power savings compared to the Fixed-power and ART-WiFi baselines. For example, our ARTPoS-irp reduces the average power consumption by $114mW$ and $102mW$ over Fixed-power and ART-WiFi, respectively, when the data rate is $1000 \text{ packets/period}$. Similarly, ARTPoS-irp achieves significant power savings over Fixed-power and ART-WiFi at higher data rates ($60.1mW$ and $66.2mW$ at $3000 \text{ packets/period}$, $86.5mW$ and $104mW$ at $5000 \text{ packets/period}$, and $125mW$ and $126mW$ at $7000 \text{ packets/period}$). As a comparison for power saving values, the CC2650 radio consumes $30mW$ power when transmitting at $5dBm$ [34]. The original ARTPoS demonstrates significant improvements over the baselines (Fig. 8). It is however important to note that ARTPoS-irp does outperform ARTPoS, by providing 0.4% to 6.4% greater PDR (3.7% increase on average), while consuming $8.7mW$ to $17.3mW$ less power ($13.5mW$ decrease on average) than ARTPoS for each data rate. These observations provide direct evidence for the conceived benefits of the insitu (PRR model) refinement incorporated in ARTPoS-irp.

Compared to ART-ZigBee, ARTPoS-irp consumes $8.5mW$ more power at the lowest data rate since it initially turns on the WiFi and ZigBee radios to measure their channel conditions. More importantly, although ARTPoS-irp consumes more power than ART-ZigBee, the latter is not able to deliver satisfactory PDRs at high data rates because of the ZigBee's limited bandwidth (i.e., the average 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-irp and ARTPoS (as seen from Fig. 8(b)). Neither WiFi nor ZigBee alone can support the data rate of 9000 *packets/period*, while our ARTPoS-irp and ARTPoS provide satisfactory PDRs by bundling the WiFi and ZigBee radios.

In order to examine ARTPoS-irp's performance under different environments, we set the data rate to $7000 \text{ packets/period}$ and performed a set of experiments comparing the performance of ARTPoS-irp with ARTPoS and two baselines (Fixed-power and ART-WiFi)² at different indoor and outdoor locations. The transmitters and receivers are placed at different rooms in an indoor office environment and in an outdoor open space. Fig. 9 shows the power consumption and PDR comparisons between our approaches and the baselines. At each location, we repeat the experiments with each approach five times and present the confidence intervals in Fig. 9. As shown in Fig. 9(a), both of our proposed methods, ARTPoS and ARTPoS-irp, provide significant power savings compared to Fixed-power and ART-WiFi baselines. For example, ARTPoS-irp reduces the average power consumption by $105mW$ and $136mW$ over Fixed-power and ART-WiFi, respectively, when performed indoors, and saves $118mW$ and $148mW$ when performed outdoors. As shown in Fig. 9(b), ARTPoS and ARTPoS-irp achieve average PDRs over 95% at both indoor and outdoor locations, which are very close to that of Fixed-power and ART-WiFi.

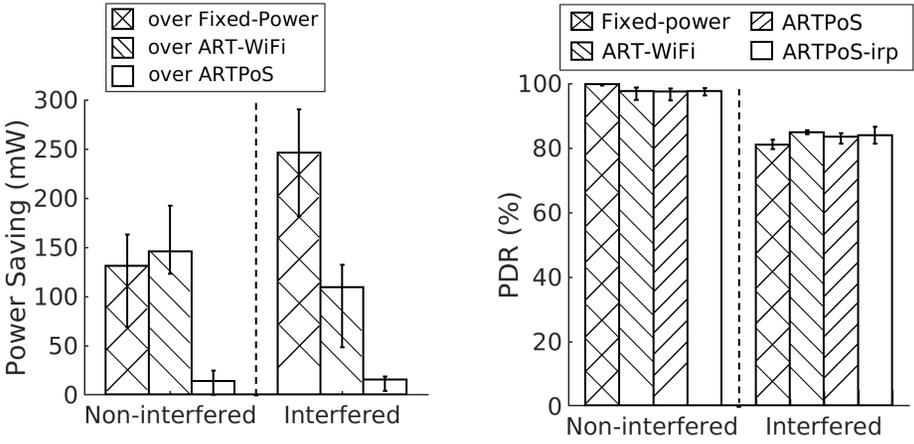
To evaluate ARTPoS-irp's performance under different interference conditions, we set the data rate to $7000 \text{ packets/period}$ and performed a set of experiments comparing the performance of ARTPoS-irp with ARTPoS and two baselines (Fixed-power and ART-WiFi) with and without

²We did not run experiments to evaluate ART-ZigBee because the data rate ($7000 \text{ packets/period}$) is beyond ZigBee's capacity.



(a) Power consumption saving provided by ARTPoS-irp over the baselines and ARTPoS (b) PDR achieved by the baselines and our approaches over the baselines and ARTPoS

Fig. 9. Power consumption and PDR comparison between our approaches (ARTPoS and ARTPoS-irp) and the baselines (Fixed-power, ART-WiFi and ART-ZigBee) at different locations.



(a) Power consumption saving provided by ARTPoS-irp over the baselines and ARTPoS (b) PDR achieved by the baselines and our approaches over the baselines and ARTPoS

Fig. 10. Power consumption and PDR comparison between our approaches (ARTPoS and ARTPoS-irp) and the baselines (Fixed-power and ART-WiFi) with and without interference.

interference. We run JamLab [3] on a TelosB mote [22] to generate controlled interference. The jammer is placed one meter away from the receiver. Fig. 10 shows the power consumption and PDR comparisons between our approaches and the baselines. Under each channel condition, we repeat the experiments with each approach five times and present the confidence intervals in Fig. 10. As shown in Fig. 10(a), ARTPoS and ARTPoS-irp provide significant power savings compared to Fixed-power and ART-WiFi baselines. For example, ARTPoS-irp reduces the average power consumption

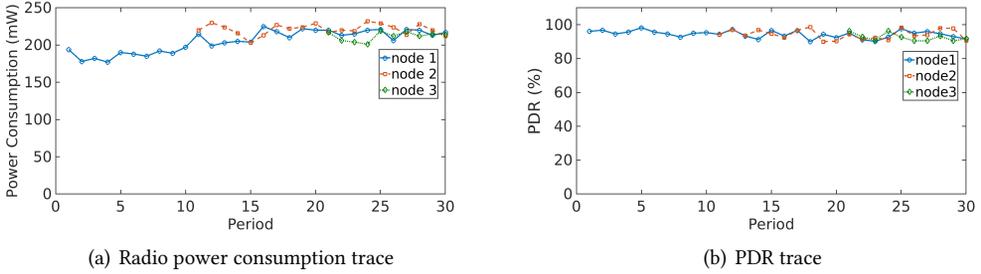


Fig. 11. Radio power consumption and PDR traces of three transmitters for 30 periods.

by $131mW$ and $146mW$ over Fixed-power and ART-WiFi, respectively, when performed without interference, and saves $246mW$ and $109mW$ with interference. It is notable that more energy is consumed by the WiFi radio when the WiFi channel is interfered, especially for ART-WiFi which uses the WiFi radio only. As shown in Fig. 10(b), the average PDRs of all approaches are over 95% without interference and decrease to the range between 81% and 85% with interference.

Finally, we examine the performance of ARTPoS-irp when new nodes join the network. When multiple senders transmit data to a single receiver, the network is configured to run a TDMA-based MAC to avoid packet collisions. In the experiment, we configure three senders to join the network one by one with the data rate of $3000 \text{ packets/period}$. Specifically, only node 1 sends data to the receiver during the first 10 periods (200s). Node 2 begins to transmit at the 11th period, while node 3 joins the network at the 21st period. Fig. 11 plots the power consumption and PDR during the experiment. Each node runs ARTPoS-irp to select its radios and transmission powers. As Fig. 11(a) shows, each node consumes slightly more power when multiple nodes are present because of the idle listening. For example, the median power consumption of node 1 during the first 10 periods is $188mW$. It increases to $212mW$ during the next 20 periods when two senders transmit. It further increases to $218mW$ when three senders are present. As Fig. 11(b) shows, the PDRs remain stable when new nodes join the network, demonstrating the effectiveness of ARTPoS-irp on preserving the link reliability through running a TDMA-based MAC protocol.

The overall experimental results thus show that ARTPoS-irp and ARTPoS can effectively reduce the energy consumption while maintaining satisfactory link reliability, to meet varying network traffic demands under different real/uncertain environments. Moreover, the new ARTPoS-irp consistently delivers superior performance compared to the original ARTPoS, particularly in terms of power consumption, thereby demonstrating the advantage of the novel online adaptation mechanism built into ARTPoS-irp.

7 CONCLUSION AND FUTURE WORK

Given the dynamic nature of communication in IoT (e.g., moving IoT/robotic units in uncertain commercial/residential/industrial environments), a traditional one-radio-fits-all approach cannot meet the challenges under typically varying operating conditions and traffic. This paper presents the new ARTPoS system that makes available multiple wireless technologies at runtime and selects the radio(s) and their transmission power(s) most suitable for the current conditions. The selection process aims to preserve link reliability within acceptable thresholds, while minimizing the power consumption of the node attributed to radio operation. To this end, empirical approaches to modeling power and PRR are presented, which allow the system to proactively adapt to large variations in power consumption and link reliability observed runtime. This is followed by the development of

two computationally light-weight online optimization schemes, based on a unique sense-classify-predict-search process, with the latter scheme also employing an insitu (runtime) refinement of the PRR models for added robustness in meeting the QoS objectives. Experimental evaluations of the thus formulated online optimization schemes, and their 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] Michael P Andersen, Gabe Fierro, and David E Culler. 2016. System Design for a Synergistic, Low Power Mote/BLE Embedded Platform. In *Proceedings of the 5th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '16)*. IEEE, Vienna, Austria, 1–12.
- [2] Niranjan Balasubramanian, Aruna Balasubramanian, and Arun Venkataramani. 2009. Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement (IMC '09)*. ACM, Chicago, Illinois, USA, 280–293.
- [3] Carlo Alberto Boano, Thiemo Voigt, Claro Noda, Kay Römer, and Marco Zúñiga. 2011. Jamlab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation. In *International Conference on Information Processing in Sensor Networks (IPSN '11)*. ACM, Chicago, IL, USA.
- [4] Duc Hoang Bui, Kilho Lee, Sangeun Oh, Insik Shin, Hyojeong Shin, Honguk Woo, and Daehyun Ban. 2013. GreenBag: Energy-efficient Bandwidth Aggregation For Real-time Streaming in Heterogeneous Mobile Wireless Networks. In *Proceedings of the 34th IEEE Real-Time Systems Symposium (RTSS '13)*. IEEE, Vancouver, BC, Canada, 57–67.
- [5] Martin Burkhart, Pascal Von Rickenbach, Rogert Wattenhofer, and Aaron Zollinger. 2004. Does Topology Control Reduce Interference?. In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '04)*. ACM, Roppongi Hills, Tokyo, Japan, 9–19.
- [6] Giuseppe Cardillo. 2013. Four Parameters Logistic Regression - There and Back Again. online.
- [7] Xiaomeng Chen, Ning Ding, Abhilash Jindal, Y Charlie Hu, Maruti Gupta, and Rath Vannithamby. 2015. Smartphone Energy Drain in the Wild: Analysis and Implications. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '15)*. ACM, Portland, OR, USA, 151–164.
- [8] Contiki. 2015. Contiki: The Open Source OS for the Internet of Things. Retrieved August 10, 2018 from <http://www.contiki-os.org/>
- [9] Ning Ding, Daniel Wagner, Xiaomeng Chen, Abhinav Pathak, Y Charlie Hu, and Andrew Rice. 2013. Characterizing and Modeling the Impact of Wireless Signal Strength on Smartphone Battery Drain. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '13)*. ACM, Pittsburgh, PA, USA, 29–40.
- [10] Yong Fu, Mo Sha, Gregory Hackmann, and Chenyang Lu. 2012. Practical Control of Transmission Power for Wireless Sensor Networks. In *Proceedings of the 20th IEEE International Conference on Network Protocols (ICNP '12)*. IEEE, Austin, TX, USA, 1–10.
- [11] Yan Gao, Jennifer C Hou, and Hoang Nguyen. 2008. Topology Control for Maintaining Network Connectivity and Maximizing Network Capacity under the Physical Model. In *Proceedings of the 27th Conference on Computer Communications (INFOCOM '08)*. IEEE, Phoenix, AZ, USA, 1013–1021.
- [12] Karim Habak, Khaled A Harras, and Moustafa Youssef. 2015. Bandwidth Aggregation Techniques in Heterogeneous Multi-homed Devices: A Survey. *Computer Networks* 92, 1 (2015), 168–188.
- [13] Gregory Hackmann, Octav Chipara, and Chenyang Lu. 2008. Robust Topology Control for Indoor Wireless Sensor Networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08)*. ACM, Raleigh, NC, USA, 57–70.

- [14] Junxian Huang, Feng Qian, Alexandre Gerber, Z Morley Mao, Subhabrata Sen, and Oliver Spatscheck. 2012. A Close Examination of Performance and Power Characteristics of 4G LTE Networks. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*. ACM, Low Wood Bay, Lake District, UK, 225–238.
- [15] Jianmin Jia and James S Dyer. 1996. A standard measure of risk and risk-value models. *Management Science* 42, 12 (1996), 1691–1705.
- [16] Michael Junger, Thomas M Liebling, Denis Naddef, George L Nemhauser, William R Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence A Wolsey (Eds.). 2009. *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*. Springer, Heidelberg, Germany.
- [17] Yeon-sup Lim, Yung-Chih Chen, Erich M Nahum, Don Towsley, Richard J Gibbens, and Emmanuel Cecchet. 2015. Design, Implementation, and Evaluation of Energy-Aware Multi-Path TCP. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT '15)*. ACM, Heidelberg, Germany, 30.
- [18] Shan Lin, Fei Miao, Jingbin Zhang, Gang Zhou, Lin Gu, Tian He, John A Stankovic, Sang Son, and George J Pappas. 2016. ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks. *ACM Transactions on Sensor Networks* 12, 1 (2016), 6–19.
- [19] Shan Lin, Gang Zhou, Kamin Whitehouse, Yafeng Wu, John A Stankovic, and Tian He. 2009. Towards Stable Network Performance in Wireless Sensor Networks. In *Proceedings of the 30th IEEE Real-Time Systems Symposium (RTSS '09)*. IEEE, Washington, DC, USA, 227–237.
- [20] CompuLab Ltd. 2018. IOT-GATE-iMX7 - Industrial Internet of Things Gateway. Retrieved January 25, 2019 from <https://www.compulab.com/products/iot-gateways/iot-gate-imx7-nxp-i-mx-7-internet-of-things-gateway/>
- [21] M. D. McKay, R. J. Beckman, and W. J. Conover. 1979. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics* 21, 2 (1979), 239–245. <http://www.jstor.org/stable/1268522>
- [22] MEMSIC. 2014. TELOS B MOTE PLATFORM. Retrieved January 16, 2019 from http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf
- [23] Ana Nika, Yibo Zhu, Ning Ding, Abhilash Jindal, Y Charlie Hu, Xia Zhou, Ben Y Zhao, and Haitao Zheng. 2015. Energy and Performance of Smartphone Radio Bundling in Outdoor Environments. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15)*. ACM, Florence, Italy, 809–819.
- [24] Ashkan Nikravesh, Yihua Guo, Feng Qian, Z Morley Mao, and Subhabrata Sen. 2016. An In-depth Understanding of Multipath TCP on Mobile Devices: Measurement and System Design. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom '16)*. ACM, New York City, NY, USA, 189–201.
- [25] Shahriar Nirjon, Angela Nicoara, Cheng-Hsin Hsu, Jatinder Pal Singh, and John A Stankovic. 2014. MultiNets: A System for Real-Time Switching between Multiple Network Interfaces on Mobile Devices. *ACM Transactions on Embedded Computing Systems* 13, 4s (2014), 121.
- [26] Jeong-Soo Park. 1994. Optimal Latin-hypercube Designs for Computer Experiments. *Journal of Statistical Planning and Inference* 39, 1 (1994), 95 – 111.
- [27] Qiuyu Peng, Minghua Chen, Anwar Walid, and Steven Low. 2014. Energy Efficient Multipath TCP for Mobile Devices. In *Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '14)*. ACM, Philadelphia, PA, USA, 257–266.
- [28] Raspberry Pi. 2016. Raspberry Pi 3 Model B. Retrieved August 10, 2018 from <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [29] Raspbian. 2012. Welcome to Raspbian. Retrieved August 10, 2018 from <https://www.raspbian.org/>
- [30] R Tyrrell Rockafellar and Stanislav Uryasev. 2002. Conditional Value-at-risk for General Loss Distributions. *Journal of Banking & Finance* 26, 7 (2002), 1443–1471.
- [31] Aaron Schulman, Vishnu Navda, Ramachandran Ramjee, Neil Spring, Pralhad Deshpande, Calvin Grunewald, Kamal Jain, and Venkata N Padmanabhan. 2010. Bartendr: A Practical Approach to Energy-aware Cellular Data Scheduling. In *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking (MobiCom '10)*. ACM, Chicago, Illinois, USA, 85–96.
- [32] Mo Sha, Meng Xu, Chenyang Lu, Linh TX Phan, Tae-Suk Kim, and Taerim Park. 2013. Self-Adapting MAC Layer for Wireless Sensor Networks. In *Proceedings of the 34th IEEE Real-Time Systems Symposium (RTSS '13)*. IEEE, Vancouver, BC, Canada, 192–201.
- [33] Monsoon Solutions. 2014. LVPM Product Documentation. Retrieved August 10, 2018 from <https://www.msoon.com/lvpm-product-documentation>
- [34] TI.com. 2016. CC2650 SimpleLink Multi-Standard 2.4 GHz Ultra-Low Power Wireless MCU. Retrieved August 10, 2018 from <http://www.ti.com/product/CC2650>
- [35] Cheng-Lin Tsao and Raghupathy Sivakumar. 2009. On Effectively Exploiting Multiple Wireless Interfaces in Mobile Hosts. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '09)*. ACM, Rome, Italy, 337–348.

- [36] R Tyrrell Rockafellar and Johannes O Royset. 2015. Engineering Decisions under Risk Averseness. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 1, 2 (2015), 04015003.
- [37] Fengyuan Xu, Yunxin Liu, Qun Li, and Yongguang Zhang. 2013. V-edge: Fast Self-Constructive Power Modeling of Smartphones Based on Battery Voltage Dynamics. In *Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation (NSDI '13)*. USENIX, Lombard, IL, USA, 43–56.
- [38] Lide Zhang, Birjodh Tiwana, Zhiyun Qian, Zhaoguang Wang, Robert P Dick, Zhuoqing Morley Mao, and Lei Yang. 2010. Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones. In *Proceedings of the 8th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS '10)*. IEEE, Scottsdale, AZ, USA, 105–114.
- [39] Ruogu Zhou, Yongping Xiong, Guoliang Xing, Limin Sun, and Jian Ma. 2010. ZIFi: Wireless LAN Discovery via ZigBee Interference Signatures. In *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking (MobiCom '10)*. ACM, Chicago, Illinois, USA, 49–60.