

MARS: Multi-radio Architecture with Radio Selection using Decision Trees for emerging mesoscale CPS/IoT applications

JOTHI PRASANNA SHANMUGA SUNDARAM, University of California, Merced, USA

ARMAN ZHARMAGAMBETOV, FAIR, Meta AI, USA

MAGZHAN GABIDOLLA, University of California, Merced, USA

MIGUEL Á. CARREIRA-PERPIÑÁN, University of California, Merced, USA

ALBERTO CERPA, University of California, Merced, USA

Internet-of-Things is rapidly growing from traditional small-scale applications like smart homes (10-100m range) to large-scale applications like Microsoft Farmbeats (1-5Km range). Small-scale applications are catered by short-range radios like Zigbee and Bluetooth LE while large-scale applications are catered by long-range radios like LoRa and NB-IoT. The other upcoming category of IoT applications like P2P energy-trade in smart homes (0.1Km-1Km range) are termed mesoscale IoT applications. There are no specialized radios fully developed for mesoscale IoT applications. They either use short-range or long-range radios. To close this gap, we explored mesoscale IoT applications using commercial-off-the-shelf (COTS) IoT radios available. Our qualitative analysis identifies Zigbee and LoRa as potential candidates. Further quantitative analysis of radio candidates on both single-hop and multi-hop topologies showed that Zigbee and LoRa achieve competitive throughput at the distance of 500-1200m from the gateway, termed the *gray-region*. A fundamental finding of these analyses is that a multi-radio system that can efficiently switch between Zigbee and LoRa performs better than the single-radio systems. However, instantaneously selecting and switching to a high-throughput radio at the time of transmission is not trivial because of erratic link quality dynamics in mesoscale IoT environments.

To address this issue, we developed MARS: a Multi-radio Architecture with Radio Selection using Decision Trees that use instantaneous end-to-end path quality metrics to instantaneously select the high-throughput radio at the time of transmission. However, realizing MARS on resource-constrained end-devices entails the challenge of obtaining instantaneous path quality information. Traditional path quality estimation is not instantaneous due to propagation and queuing delays. We overcome this challenge by showing that collecting local path metrics as input to our decision trees provides sufficient information to instantaneously identify the high-throughput radio. The radio selector of MARS is powered by TAO-CART trees. These trees are converted into IF...ELSE statements for efficient deployment on end devices. The evaluation of MARS on a large-scale mesh topology at two different locations show that MARS can efficiently identify and switch to the high-throughput radio at the time of transmission, leading to an average throughput gain of 48.2% and 49.79% than the competing schemes¹.

CCS Concepts: • **Networks** → **Cyber-physical networks; Network design principles; Network protocol design; Data path algorithms; Network experimentation; Network range; Network management.**

¹This article has been submitted to ACM Transactions on Cyber-Physical Systems for consideration. Its contents may not be further disseminated until a final decision regarding publication has been made and further permission has been granted

Authors' Contact Information: Jothi Prasanna Shanmuga Sundaram, jshanmugasundaram@ucmerced.edu, University of California, Merced, California, USA; Arman Zharmagambetov, armanz@meta.com, FAIR, Meta AI, Menlo Park, California, USA; Magzhan Gabidolla, mgabidolla@ucmerced.edu, University of California, Merced, California, USA; Miguel Á. Carreira-Perpiñán, mcarreira-perpinan@ucmerced.edu, University of California, Merced, California, USA; Alberto Cerpa, acerpa@ucmerced.edu, University of California, Merced, California, USA.

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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2024 Copyright held by the owner/author(s).

Manuscript submitted to ACM

Manuscript submitted to ACM

1 Introduction

Traditional IoT networking comprises both small-scale applications, like smart homes (10-100m range), and large-scale applications, like Microsoft FarmBeats [45] (1-5Km range). On the other hand, emerging applications serve in the range of 0.1-1.5Km, like smart-grid Neighborhood Area Networking (NAN) [22], target tracking [46], Industrial Automation [39, 49] and P2P energy-trade in smart neighborhood [36, 41, 44]. These applications are hereafter referred to as *mesoscale IoT applications*. In P2P energy trade [15], multiple nodes bid for available energy like stock market transactions. Industrial automation [14, 16, 51, 56] provides time-critical closed-loop control for M2M communications. Here, the radios should optimize latency/throughput for better application performance [42]. The latency/throughput are calculated on successfully delivered packets, indirectly including reliability. The IoT radio utilized here will be a small component of a larger device. Unlike the typical IoT nodes powered by a smaller battery, these devices are either grid-powered [57, 65] or have a large battery reserve to support the entire device [14–16]. Hence, we chose latency/throughput as the performance indicator with a secondary emphasis on energy consumption.

Unlike the comparatively stable deployed environments of smart homes and Microsoft Farmbeats [45], mesoscale applications serve in urban/semi-urban environments that are relatively dynamic. This dynamic nature is because of environmental obstacles, like buildings made of wood, glass, and concrete, heavy human influx, and vehicle movements. These factors erratically alter the wireless link quality. Some applications may need direct single-hop communication with the gateway while other applications may need multi-hop communication to communicate with other nodes in the network. Since LoRa covers the entire mesoscale range, an intuitive idea would be to use LoRa in a multi-hop fashion, for applications that need inter-node communications. However, using a low-throughput LoRa radio in a multi-hop fashion will further reduce the end-to-end throughput with the increase in hops [10]. Furthermore, mesoscale IoT applications do not have a dedicated, well-developed radio technology. To close this gap, we explored the performance of the available Commercially-Off-The-Shelf (COTS) IoT radios for the mesoscale IoT applications.

First, we characterized the mesoscale IoT environment and conducted a qualitative analysis on the available IoT radios (§5). This analysis identifies Zigbee and LoRa as potential candidates for mesoscale IoT applications. Further quantitative analysis (refer §5.1) of the radio candidates on both single-hop and multi-hop topologies show that Zigbee and LoRa achieve competitive throughput at the distance of 500-1200m from the gateway, hereafter referred as the *gray-region*. In this *gray-region*, LoRa and Zigbee radios achieve high end-to-end throughput at different time instants due to erratic channel conditions (refer §5.2). Unlike transitional region[48] that shows the dynamicity of PDR between connected and disconnected regions of a single hop, the gray region shows that two radios with an order of magnitude difference in theoretical throughput, achieve competitive throughput, over multiple hops. The fundamental finding is that Zigbee and LoRa can work together as a multi-radio system and efficiently switch between the radios to maximize throughput (refer §6). However, instantaneously selecting a high-throughput radio is not a trivial problem because of erratic link quality dynamics in the mesoscale IoT application environments.

Second, we developed MARS to instantaneously select high throughput radio during transmission, using instantaneous end-to-end path/link quality metrics. MARS's node comprises both Zigbee 2.4 GHz and LoRa 915 MHz radios. While it is intuitive to employ a Machine Learning (ML) model to perform radio selection, it entails the challenge of obtaining instantaneous path/link quality estimations. Multi-hop Zigbee radios propagate the link quality information of all the links along the path to compute path quality [24]. However, the propagated link quality info expires before it reaches the destined node because of temporal link quality variations, propagation, and queuing delays along the path (refer §8.1). While it is ideal to compute the path quality of the entire path, we observed that a part of the entire

path length is sufficient to identify the high throughput radio during transmission. We performed extensive analysis and evaluation on multiple topologies, to find the fewer hops required to estimate the end-to-end path quality as input to our ML model and still obtain good radio selection accuracy (refer §8.2). This way we balance the trade-off between perfectly accurate global metrics that cannot be gathered on time and acceptably accurate local metrics that can be collected on time while still providing good input for our ML model, trained by CART[60], to achieve acceptable radio selection accuracy.

Finally, we optimize the CART with Tree-Alternating Optimization (TAO) [21]. The TAO algorithm optimizes the traditional trees to achieve higher accuracy with lower training data requirements. The TAO-Optimized CART (TAO-CART) can be converted into IF...ELSE statements for efficient deployment on IoT end devices. An evaluation of MARS on a complex mesh topology at two different locations showed that MARS can instantaneously select the high-throughput radio during transmission leading to an average throughput gain of 48.2% and 49.79% than the competing schemes.

In summary, the contributions of our work are:

- (1) Identifying the absence of a dedicated radio for emerging mesoscale IoT apps, we developed an intelligent multi-radio architecture with COTS IoT radios, namely Zigbee and LoRa, that has an order of magnitude difference in theoretical throughput.
- (2) We identified the existence of a *gray-region* between 0.5-1.2 Km from the gateway, through analytic and experimental analysis, showing that due to the temporal variability in the channel, it is uncertain which radio provides the best throughput at any given time.
- (3) We developed an implementation of an ML Decision Tree model for radio selection using the TAO [21] algorithm. To the best of our knowledge, this is the first real-world use case of TAO.
- (4) We showed that partial path quality can provide sufficient information for our TAO-optimized tree to accurately select the high-throughput radio during transmission.

2 Related work

Multi-radio wireless networks have been heavily worked on for a decade using different radio combinations. WiFi + LTE [9, 12], WiFi + Bluetooth [4, 7, 58], 60GHz+WiFi [63] to ameliorate energy efficiency [4, 32, 34, 53], traffic management [50], mobility management [59] and routing management [5, 13]. Bahl et al. [8] identified that multi-radio systems are beneficial for wireless networks. They showed that using different radio technologies on the same platform improves performance. They recommended abstracting multiple radios as a single logical pipeline yet providing access to the networking protocols. While most of the above multi-radio systems were developed for high-power radios like WiFi, the systems designed for IoT networks are discussed below.

Backpacking [17] was developed for high data rate sensor networks like HP's CeNSE [54]. Initially, they experiment with the network-level energy efficiency of the low-power (802.15.4) and high-power (802.11b) radios on different data rates, node densities, and distances. Using these results, a cross-layer empirical model is developed to calculate the optimal density of high-power radios to be augmented with the low-power radios, to strike a balance between the usage of the two radio types. They use 802.15.4 and 802.11b radios in the accumulator node and only 802.15.4 radio in the originator node. Originator nodes send smaller packets of sensed data to the accumulator node using 802.15.4 radio. The accumulator acts as a relay between multiple originator nodes and the base station. An accumulator node gathers all the data to be sent to the base station from multiple originator nodes and uses 802.11b radio to transmit the accumulated data to the base station. Their indoor experiments inside a small room show that Backpacking can

Table 1. A comparison of related work

Multi-radio systems	Radios Used	Optimized Metric				Deployment Range >500m?	Radio Selector	Mobility?
		Energy	Reliability	Throughput	Latency			
Backpacking [17]	802.11 + 802.15.4	✓	✗	✗	✗	✗	✗	✗
Kusy et al. [32]	802.15.4 (915MHz) + 802.15.4 (2.4GHz)	✗	✓	✗	✗	✗	✗	✗
Gummeson et al. [27]	802.15.4 (915MHz) + 802.15.4 (2.4GHz)	✓	✗	✗	✗	✗	Reinforcement Learning	✓
Lymberopoulos et al. [34]	802.11b + 802.15.4	✓	✗	✗	✗	✗	Threshold-based	✗
MARS	802.15.4 (2.4GHz) + LoRa LPWAN	✗	✗	✓	✓	✓	Decision Trees	✗

gain 44% improvement in average energy per bit. This will work perfectly for high-data-rate, non-deadline-oriented applications but won't work for low-data-rate, time-sensitive applications because an accumulator node has to gather a lot of data before transmitting via 802.11 radio. Unlike this work, we develop a multi-radio architecture for low data rate, time-sensitive mesoscale IoT applications.

Kusy et al. [32] develop a multi-radio architecture for wireless sensor networks. They show that employing two multi-hop radios in the same node improves communication reliability but pays a toll of 3-33% higher energy consumption. They exploit radio diversity with two 802.15.4 radios working in mutually exclusive bands of 2.4 GHz and 900 MHz to enable simultaneous transmission and reception capabilities. They abstract the radios behind a software driver that enables simultaneous, independent control and data plane operations. Their experiment on a 400m×450m area achieves communication reliability but pays a 33% overhead in energy consumption as multi-hop networks will incur more delay and energy when more hops are added [17]. Unlike this work, we focus on optimizing throughput and latency for mesoscale IoT applications, ranging from 100-1000m from the gateway. Hence, it is clear that using two independent, simultaneously operating multi-hop networks for a mesoscale IoT application is not energy-inefficient and they incur more delay at each hop leading to a reduced throughput. These results suggest to use one multi-hop radio and one single-hop radio with diverse frequency ranges to improve the performance of mesoscale IoT applications.

Gummeson et al. [27] found that range diversity and channel dynamics are the major impediments to energy efficiency in mobile sensor networks. They developed a Reinforcement Learning (RL) based adaptive link layer to predict channel dynamics. The reward of this RL algorithm is the expected energy efficiency when choosing a particular radio for transmission. On learning the channel dynamics, they employ a formula to translate the learned channel dynamics to estimate the energy required by each radio for transmitting a packet. Finally, they choose a radio that consumes less energy. They also propose a protocol to coordinate the data plane and control plane operations of the two radios. The RL algorithm is the appropriate choice for learning channel dynamics of complex mobile sensor networks, but their complexity is very high (refer §9 (ii)) for static IoT network employed for mesoscale IoT applications.

Lymberopoulos et al. [34] provided guidelines to design a multi-radio architecture for energy-efficient Wireless Sensor Networks (WSN). They use 802.11b and 802.15.4 radios in the same platform and study the effect of packet size, and transmission time on energy consumption. Through experiments, they identified the bottlenecks, trade-offs, and break-even points of multi-radio systems. Based on these break-even points, they developed a threshold-based radio-switching algorithm to optimize energy efficiency. This threshold-based algorithm utilizes the break-even point regions, where one radio performs better than the other, to switch to an energy-efficient radio.

A comparison of related work with MARS is tabulated in Table 1. While all the previous works employed multi-radio systems for energy efficiency and improving reliability, MARS is optimizing throughput and latency. To the best of our knowledge, MARS is the first to explore multi-radio architecture for mesoscale applications ranging between 100-1000m from the gateway. Two closely related works are Lymberopoulos et al. [34] and Gummeson et al. [27]. We made our best effort to adopt the threshold-based algorithm of Lymberopoulos et al. [34] for comparison with MARS. The sophisticated RL-based radio switching protocol of Gummeson et al. [27] suffers from the below-described problems:

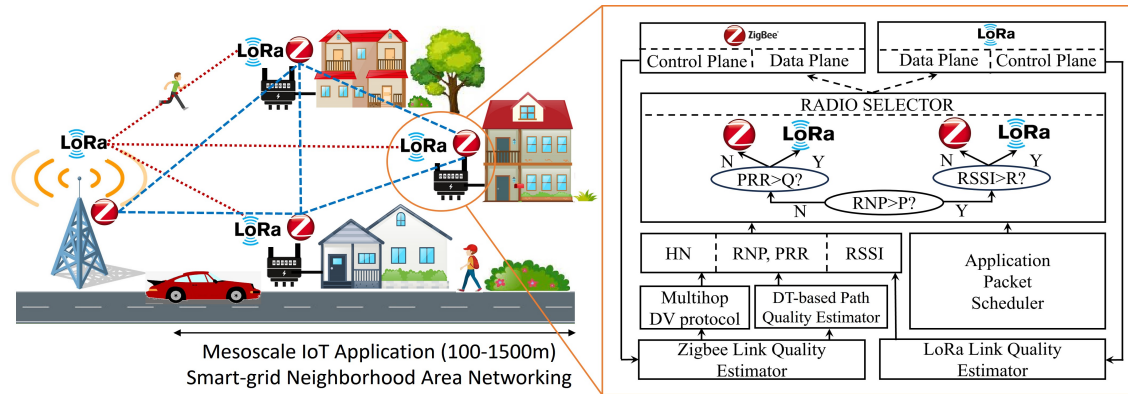


Fig. 1. MARS System Overview

(i) During data transfer between two radios in the communication range, the radio switching protocol, designed for energy efficiency, does a three-way handshake to switch radios. This incurs additional latency which will heavily degrade the throughput. (ii) A well-known issue of model-free RL is that it requires heavy training data to converge to an acceptable performance [23] and the amount of data used for training their model-free RL model is obscure. We compared the training data requirements of Q-learning and Decision Tree models in §7.2. Since Gummeson et al. [27] developed custom hardware for energy efficiency, we made our best effort to adapt their work to compare with MARS.

3 MARS System Overview

In this section, we provide an overview of MARS. Figure 1 shows the system overview with all the components. MARS consists of a network of nodes using multiple radios, chosen using the methodology explained in § 5. Multi-hop Zigbee network uses a distance-vector routing [19] protocol for multi-hop routing. The control plane of both radios constantly feeds the link quality estimators. LoRa link quality estimator directly feeds the Radio selector with RSSI. The DT-based path-quality estimator takes input from the Zigbee LQ estimator to estimate path qualities RNP and PRR. The specific path quality metrics used are not completely end-to-end. The analysis and determination of those metrics are explained in § 8.2, where we show that a partial path quality is sufficient to provide inputs to the ML model to achieve acceptable radio-selection accuracy, while at the same time being able to gather this data in a fast enough manner as to preserve the temporal correlations in the RF channel. The ML model uses an input feature vector composed of multiple path and link quality metrics explained in § 7.1. The reason for using an ML model is provided in § 5.2, and it is related to the temporal dynamics in the gray-region explained in §5.2. Taking these inputs, the radio selector powered by an ML model, explained in § 7, chooses the high throughput radio at the time of transmission.

4 Factors affecting mesoscale IoT applications

A mesoscale IoT environment ranges anywhere between 100-1500m [22, 46, 62]. The factors affecting the performance of mesoscale IoT applications are three-fold: (i) Environmental obstacles, (ii) Vagarious IoT radios, and (iii) Applications demanding different communication paradigms and Quality of Service (QoS).

Firstly, a mesoscale IoT environment will be very dynamic in nature. Mesoscale IoT environment comprises buildings, heavy human influx, and vehicle movements. The buildings are made of materials like steel, glass, concrete, physical firewalls, etc. Wireless signals passing through these objects get attenuated to a fixed level that can be approximated

Table 2. A comparison of different IoT radios in the context of city-scale smart environments showing that Zigbee and LoRa are the better-suited radios.

	SigFox	WiFi HaLow	BLE	Zigbee	LoRa
Open-source?	No	Yes	Yes	Yes	Yes
Link Budget (dB)	158 [2]	24.5 [37]	108 [35]	103 [1]	150 [40]
Topology Type	LPWAN	LPWAN	PAN	PAN/LAN	LPWAN
Communication Range (m)	5000	1000	100	125 [38]	5000
Max bitrate (bps)	600	upto 4M	up to 1M	250K	upto 27K

with a mathematical model. Whereas, humans and moving objects, like vehicles, attenuate the wireless signals in an erratic manner. Capturing these erratic variations through a mathematical model is complicated and overwhelming.

Secondly, radios developed for IoT applications are designed to be very cheap with low energy consumption so that they can be deployed in thousands of IoT devices. These low-cost IoT radios are vagarious. For example, a LoRa radio slightly deviates from its central frequency during transmission [43]. These vagarious IoT radios, deployed in a mesoscale IoT environment that causes various levels of signal attenuation, will fluctuate the performance of IoT networks in an unpredictable manner.

Thirdly, with the rise of multi-tenancy [11] in IoT networks, where each end-node serve multiple applications, each application may need different communication paradigms. For example, In an asset monitoring application, a node reports the status of an asset to the base station periodically. A single-hop link to the base station is more suitable for this application than the multi-hop paradigm. Applications like target-tracking need multi-hop inter-node communications to coordinate with other devices for efficiently tracking the target and actuating necessary functions. LoRa works better for applications like asset monitoring and Zigbee works better for applications like target tracking.

Furthermore, for applications requiring high data rate bi-directional communications, WiFi-HaLow will achieve higher performance than LoRa as LoRa is precisely designed for up-link oriented applications [61]. Since each radio has its own merits and demerits, using single-radio networks to cater to all the applications in the above-described dynamic mesoscale IoT environment is debatable. Hence, it is crucial to analyze the characteristics and performance of different IoT radios for mesoscale IoT applications.

5 LP-Radios for mesoscale applications

In pursuit of finding a radio for mesoscale IoT applications, we first qualitatively compare the suitability of different IoT radios for mesoscale IoT applications. On identifying suitable radio candidates, we further conduct quantitative performance analysis to understand their strengths and pitfalls.

A qualitative comparison. A comparison of different IoT radios is tabulated in Table 2. SigFox is not an open-source technology. Hence, it cannot be used for deploying private networks. WiFi HaLow’s (802.11ah) low link budget won’t allow the signals to penetrate through the environmental obstacles. Hence, SigFox and WiFi-HaLow do not suit mesoscale IoT applications. Zigbee is a multi-hop, high-bit rate radio with a moderate link budget to penetrate environmental obstacles. Single-Hop LoRa is also a better fit as it has a strong link budget satisfying all the requirements of mesoscale IoT applications. Zigbee and LoRa are chosen for further quantitative analysis.

Table 3. Throughput and packet loss rate of Zigbee 2.4GHz, Zigbee 915 MHz and LoRa 915 MHz showing that Zigbee 2.4GHz is better for short-range communications and LoRa 915MHz is better for long-range communications in smart-neighborhood environment.

Radio candidate	Throughput (bps)	PLR (%)
Zigbee 2.4 GHz in free space	77,634	0
Zigbee 2.4 GHz in smart environment	56,530	33.40
Zigbee 915 MHz in free space	9,530	0
Zigbee 915 MHz in smart environment	6,777	24.60
LoRa 915 MHz in free space	4,579	0
LoRa 915 MHz in smart environment	4,579	0

5.1 A quantitative comparison of IoT radios

Zigbee and LoRa are identified as potential radio candidates for further quantitative analysis to choose a suitable radio for mesoscale IoT applications. In this analysis, the performance of both radio candidates is compared by their throughput and Packet Loss Ratio (PLR) in free space and an urban-like mesoscale environment. This analysis is done for both single and multi-hop topologies for a Zigbee radio.

5.1.1 Single-hop experiments. A sender and receiver of both the radio candidates are placed in both free space and an urban-like mesoscale environment at a distance of 20m from each other. The latter spans multiple wooden walls, a glass door, and three humans in between the nodes. One thousand packets are transmitted from the sender to the receiver to average the achieved throughput and PLR.

Throughput of Zigbee 2.4GHz. One thousand 29-byte-sized packets are sent back-to-back without any delay. Another receiving mote logs the received packets. CSMA and link-layer acknowledgments are disabled in TinyOS. There is no reliable mechanism for the retransmission of packets. Zigbee achieves a 100% packet reception ratio with an average throughput of 77,634 bps in free space. The theoretical maximum of 250 Kbps is not achieved as throughput heavily depends on packet size [26]. In an urban-like environment, 56,530 bps throughput was achieved with 33.40 % PLR. Our experiments closely matched the results by Opal [31], Hamdy et al. [30] and Yousuf et al. [47]. The difference between free-space and urban-like environments is the increase in packet loss ratio and a reduction in throughput. Heavy packet loss is due to the fact that the signals are either lost or corrupted because of the urban-like environment. The reduction in throughput is because the signals are penetrating through the obstacles in an urban-like environment. Any retransmission mechanism incorporated will reduce the overall network throughput. This packet loss is due to the fact that signals transmitted in 2.4GHz have moderate penetration capacity and were not received/decoded by the receiver. An intuitive idea would be to utilize Zigbee radio at a lower frequency of 915 MHz for an increased signal penetration capacity.

Throughput of Zigbee 915MHz and LoRa 915MHz. Zigbee915 achieves a packet reception ratio of 100% in free space with an average throughput of 9530 bps. An average throughput of 6777 bps is achieved with 24.60% PLR in a mesoscale environment. As the loss is higher in Zigbee 915 MHz, we wanted to compare a relatively newer LoRa radio working in 915 MHz with Zigbee 915 MHz. Two LoStik [55] LoRa USB nodes using SF7 in 125KHz bandwidth are placed in free-space and urban-like environments at a distance of 20m in between them. One thousand 29-byte-sized packets were sent from one node to another. LoRa was able to achieve 4579 bps average throughput with a 100% reception ratio in both free-space and smart-neighborhood environments. The penetration capacity of Zigbee 915MHz cannot match LoRa's penetration capacity as LoRa generates higher link-budget signals. From the above single-hop experiments, it is evident that Zigbee 2.4 GHz achieves higher throughput than Zigbee 915MHz. This high throughput characteristic is

highly desirable for mesoscale applications despite the higher loss rate since the throughput will degrade further in multi-hop communications owing to queuing and channel sensing delays [10]. If we choose Zigbee 915 MHz because of a lower loss rate, its lower throughput will further degrade when employed in a multi-hop fashion. So, we choose Zigbee 2.4GHz over Zigbee 915MHz. Comparing LoRa 915MHz and Zigbee 915MHz, LoRa 915 MHz achieves lower PLR than Zigbee 915 MHz as LoRa signals are robust owing to its CSS modulation scheme [43]. Hence, it is identified that Zigbee 2.4GHz is the better radio for mesoscale IoT apps demanding short-range, multi-hop communications while LoRa is better for mesoscale IoT applications demanding long range, single-hop communications.

5.1.2 Multi-hop experiments. The next step is to test the performance of Zigbee and LoRa in a simple multi-hop line topology. Multi-hop line topology might affect the throughput of Zigbee due to channel sensing (CSMA) and queuing (forwarding) delays. The multi-hop experiments are conducted in two folds. First, an analytic performance analysis is conducted using the available models. Second, the result of this analytic performance analysis is compared with the real-world experimental performance analysis.

Topology setup. A simple line topology is used for this analysis. One gateway and fifteen nodes are placed in line topology in free space such that each hop spans approximately 100m. Packets of size 29 bytes are generated according to an Independent Poisson Process at the rate of 1 packet every 3 seconds. Each node has both LoRa and Zigbee radios. All generated packets are destined for the gateway. LoRa can reach the gateway in a single hop whereas Zigbee takes multiple hops. LoRa nodes transmit in a 125KHz channel with an SF7 spreading factor. LoRa gateway is capable of receiving eight packets concurrently [43]. This highly reduces collision.

Analytic throughput analysis. LoRa uses ALOHA where nodes can transmit at their will. Packets are transmitted at the Poisson rate λ packets/second assuming that each packet occupies the channel for τ seconds. Then the normalized traffic $G = \lambda \times \tau$. A packet will be corrupted due to channel noise or when two packets are transmitted at the same time. The former case is ignored in this analysis. The occurrence of the correctly received packets can be defined as $\lambda' < \lambda$. Then the normalized channel throughput is given by $S = \lambda' \times \tau$. Packet reception is successful only if there is no other transmission in the interval $[-\tau, \tau]$. Since the Poisson process defines all the transmission times, the probability that two packets won't collide is $e^{-2\lambda\tau} = e^{-2G}$. The throughput is given by multiplying the normalized channel throughput and the probability of successful packets [3]. Therefore the throughput of LoRa is given by $S = Ge^{-2G}$.

Zigbee network employs CSMA. Each node schedules packets to its neighbors according to an independent Poisson point process. By assuming an independent Poisson process, relaying and queuing delays can be ignored. This network can be modeled using a Continuous Time Markov Chain with a set of nodes transmitting at any time instant [10]. Let i be a node. N_i be the set of all neighbors of i . Let g_{ij} and s_{ij} be the scheduled and desired packet rates when a packet is transmitted from node i to j . Let $P(A)$ be the probability that all nodes in set A are silent at any given instant of time. According to CSMA, a node can transmit only when its neighbors are not actively receiving or transmitting. This gives us

$$\frac{s_{ij}}{g_{ij}} = P(N_i \cup N_j) \quad (1)$$

The network state for given g_{ij} is defined by the set of nodes D that are in the transmitting state. Using steady-state probabilities and global balance equations $P(A)$ can be given as

$$P(A) = \frac{SP(A^c)}{SP(V)} \quad (2)$$

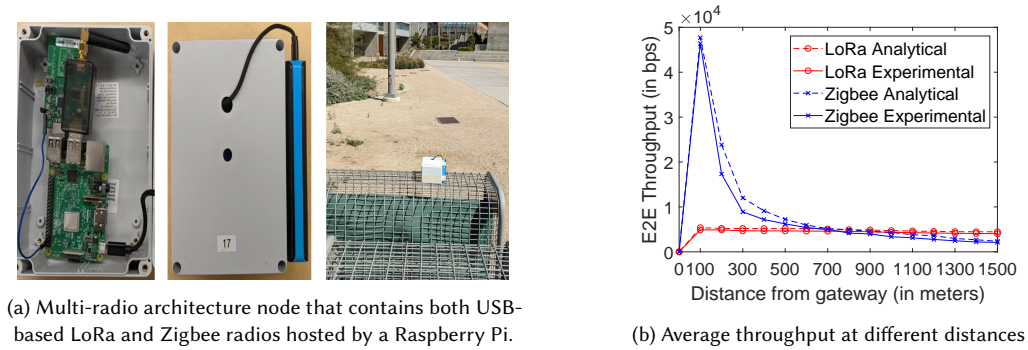


Fig. 2. Hardware used and multi-hop line topology results.

where SP is the sum-of-products, A^c is the set of nodes that are not in set A, and $SP(V)$ is the sum-of-products of all the nodes in the network. At any given time, there will be D independent sets containing i nodes that can transmit concurrently in a CSMA network. SP can be calculated as follows: First, calculate the product of all the scheduled rates of all the nodes in set i . Then, the product of all these independent sets is summed. SP can be expressed as:

$$SP(A) = \sum_{D \in A} \prod_{i \in D} g_{ij} \quad (3)$$

Now, substituting 2 in equation 1 gives,

$$\frac{s_{ij}}{g_{ij}} = \frac{SP(\left[(N_i \cup N_j)\right]^c)}{SP(V)}, j \in N_i \quad (4)$$

The above equation 4 can be written for all the transmit/receive pairs in the network to form a system of linear equations that can be solved iteratively for s_{ij} with given g_{ij} which gives the end-to-end throughput of a multi-hop CSMA network [10].

Experimental throughput analysis is conducted with a multi-radio hardware node shown in Figure 2a. This is a Raspberry Pi 3B hosting a USB-based TelosB [64] Zigbee mote and a USB-based LoStik LoRa [55] nodes. The Raspberry Pi unit is powered by a portable external Power Bank and protected by a PVC casing. TelosB [64] and LoStik nodes [55] are programmed to transmit a 29-byte packet on receiving a command from the Raspberry Pi. During this experiment, link-level acknowledgments are disabled and CSMA is enabled in the TelosB motes. ALOHA MAC is enabled in LoRa radios. The nodes are placed in a line topology to have *connected links* [18] to the neighboring nodes. While it is tedious to find the spots with a high packet reception ratio at longer distances, multiple iterative efforts helped to identify them.

5.1.3 Multi-hop result analysis. Figure 2b depicts the End-to-End (E2E) throughput as the function of distance. This end-to-end throughput at a specific distance is the average of one thousand packets. The reason for a drastic difference in E2E throughput between LoRa and Zigbee at 100m is the fact that only one node is transmitting without any contention. A considerable drop in Zigbee throughput is seen from 100m-300m because it uses a single channel with three contenders. CSMA mechanism blocks two other links from transmitting to avoid collisions, allowing only 1 of 3 links to transmit at any given time until 300m. One could argue to use multi-channel Zigbee, but this will be detrimental for MARS (refer §10 for more details). After 300m, Zigbee's throughput steadily decreases with distance because of CSMA blocking delay and queuing delay. LoRa's throughput does not show any drastic decrease but slightly decreases with distance. This is because the LoRa signals are robust enough to pass through the mesoscale IoT application environment

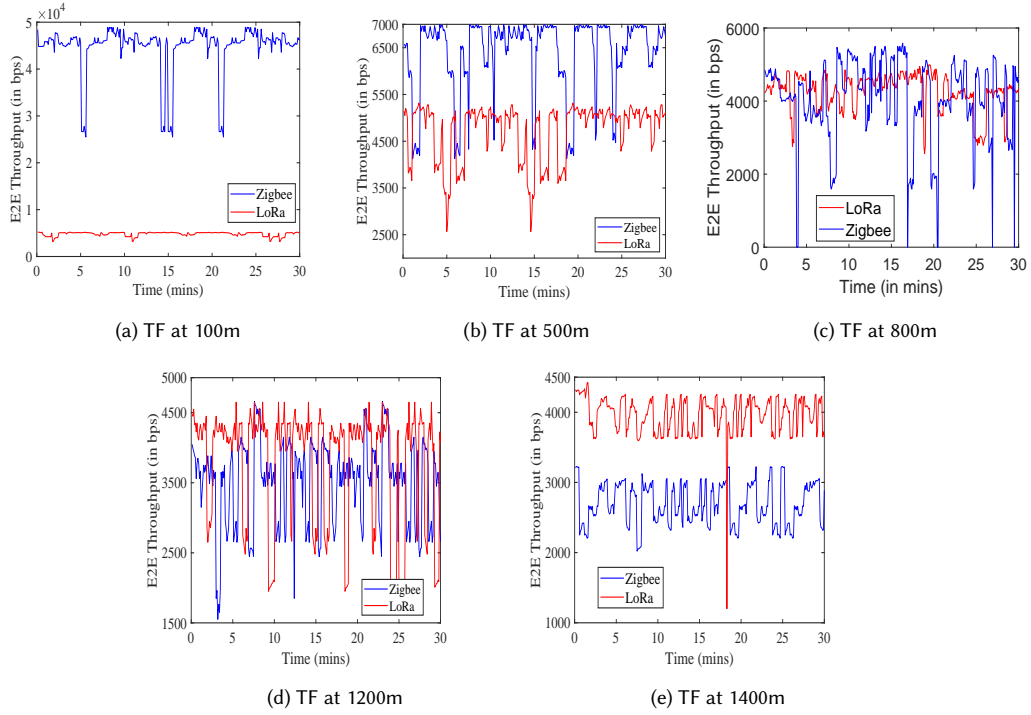


Fig. 3. Throughput Fluctuations (TF) at different distances from the gateway

as the Chirp Spread Spectrum used by LoRa for modulation makes the signal highly resistant to attenuation. Also, the LoRa gateway can receive and decode eight packets concurrently. Figure 2b shows that Zigbee wins until 500m and LoRa wins after 1200m. LoRa and Zigbee achieve competitive throughput between 500m-1200m from the gateway, henceforth referred as the *gray region*. If a LoRa gateway is placed at the center of the deployed environment, the *gray-region* will cover an area of $3,738,495m^2$.

5.2 E2E Throughput fluctuations in gray region

Fig. 2b shows that LoRa and Zigbee provide competitive throughput between 500m-1200m from the gateway. The end-to-end throughput at different distances from the gateway depicted in Fig. 2b is the average E2E throughput of a thousand packets. This does not depict the end-to-end throughput fluctuations of different packets over time. So, we take a deeper look into the end-to-end throughput fluctuations over time at different distances from the gateway.

Figs.3a - 3e show the throughput fluctuations for 30 minutes between 100m to 1400m from the gateway. Zigbee wins at 100m (Fig.3a) and LoRa wins at 1400m (Fig.3e). At 500m from the gateway (Fig. 3b), Zigbee achieves higher throughput most of the time. Whenever Zigbee's throughput is falling, LoRa is able to back up Zigbee to provide better throughput. The difference in throughput of Zigbee and LoRa is considerably higher at 500m.

Fig. 3c shows the throughput fluctuations at 800m. Zigbee mostly wins but the throughput of Zigbee is highly fluctuating as packets are experiencing multiple hops. The average throughput of Zigbee and LoRa is very close to each other. Fig. 3d shows that Zigbee experiences low E2E throughput. So, it mostly underperforms at this longer distance of 1200m from the gateway.

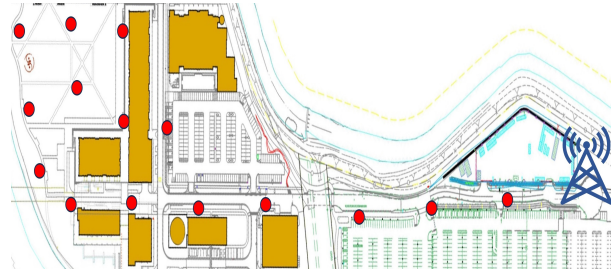


Fig. 4. Location A Mesh topology - nodes populated at the gray region.

The fundamentally surprising information here is that two radios with an order of magnitude difference in theoretical throughput, are achieving competitive throughput performance in the gray region. From the above throughput fluctuations, it is evident that using a single radio IoT network leads to throughput loss even on a simple line topology. While most of the real-world IoT applications will employ the more complex mesh topology, this throughput loss will get further amplified in real-world mesh topology applications.

Key observations:

- (1) There are no dedicated radios for mesoscale IoT applications.
- (2) Qualitative analysis identified Zigbee 2.4GHz and LoRa 915MHz as radio candidates. Quantitative multi-hop experiments showed that Zigbee and LoRa radio achieve competitive throughput from 500-1200m from the gateway, called the *gray region*.
- (3) Erratic end-to-end throughput fluctuations were observed in the gray region. Predicting these throughput fluctuations will help to select a higher throughput radio at the time of transmission.
- (4) Employing radios in a multi-hop fashion, heavily reduces end-to-end throughput.

6 Why multi-radio for mesoscale IoT networks?

The above key observations made on a simple line topology suggest that Zigbee and LoRa together will provide higher throughput for mesoscale IoT applications. Line topology is seldom used in real-world applications. Hence, it will be interesting to explore the throughput fluctuations of LoRa and Zigbee in a mesh topology.

Experimental setup. In this preliminary motivation experiment, nodes are populated in the gray region to form a mesh topology as shown in Figure 4. The hardware setup explained in section 5.1.2 is utilized. A total of one thousand 29-byte packets are transmitted by each node in the network destined for the gateway. The R Pi host commands both radios every 3 seconds to transmit a packet concurrently. The LoRa radios use ALOHA MAC. The Zigbee radios use CSMA MAC and link-level acknowledgments are enabled in TinyOS. After multiple iterative efforts, every Zigbee node is placed in a position that provides a connected link [18] to its immediate neighbors.

MAC protocol. Although concurrent communications [25] and TDMA [29, 51] approaches may outperform CSMA, we considered CSMA MAC since the mesoscale applications will generate different payloads at different times, that needs to be transmitted immediately. This is better suited for CSMA than TDMA/Concurrent communications. Concurrent communications requires time-synchronized nodes to leverage constructive interference. In mesoscale IoT applications, like P2P energy trade, two IoT nodes may not be tightly time-synchronized. If an IoT end-node wants to send an inquiry for energy trade, it has to send the message immediately without waiting for its next time slot for transmission. Hence,

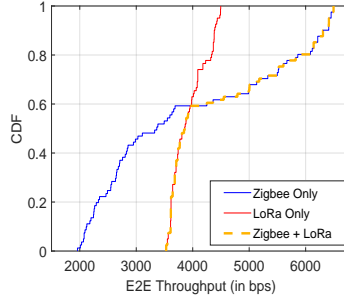


Fig. 5. Multi-radio system with Zigbee+LoRa radios achieves higher throughput than single-radio systems.

Table 4. Latency of different radio systems

Average Required Latency	Zigbee+LoRa	Zigbee Only	LoRa Only
55ms	50.62ms	62.32ms	66.55ms

CSMA MAC is the right MAC choice for mesoscale IoT applications. The Zigbee radios employ a Distance-Vector protocol [19] for multi-hop routing.

Results. The throughput achieved by both the radios of the network is plotted as a CDF in Fig. 5. This figure shows that LoRa achieves higher throughput for 59% of the transmissions and Zigbee achieves higher throughput for 41% of the transmissions. The throughput is calculated as the fraction of total bits sent over the latency incurred. Throughput has an inverse relationship with latency. According to 5G America’s report [6], the average required latency for mesoscale IoT applications is 55ms. The average latency achieved by Zigbee-only and LoRa-only radios is 62.32 ms and 66.55 ms respectively. This is higher than the average required latency, 55ms.

Based on the above observations, a trace-driven simulation is conducted on the multi-radio network containing both Zigbee and LoRa radios. This trace-driven simulation mimics the performance of multi-radio networks that can choose higher throughput radio for every transmission. This is plotted as the dashed golden line in figure 5. This dashed golden line perfectly traces LoRa radio until the first 59% of the transmissions and follows Zigbee radio for the next 41% of the transmissions. The average latency achieved by this simulated multi-radio network maximizing for throughput, 50.62ms, falls within the bounds of the average required latency. This shows the necessity of an intelligent multi-radio system for mesoscale IoT applications that can choose a higher throughput radio at the time of transmission.

While it is convenient to choose a high-throughput radio based on traces, it is tedious to predict a high-throughput radio in real-world deployments. The end-to-end throughput fluctuates over time as shown in Figures 3a - 3e. If an end node is able to predict these throughput fluctuations, a high-throughput radio can be predicted during transmission.

7 Building a machine learning model

This section explains the process of building a Machine-Learning (ML) model for instantaneously selecting a high-throughput radio at the time of transmission. This radio selection problem is formulated as a classification problem in §7.1. Data is collected on three different topologies. "Location A - Line" is the Line topology described in § 5.1.2 and "Location A - Mesh" is the mesh topology described in § 6. "Location B - Mesh" is the mesh topology mentioned in § 9.

We develop separate ML models for each topology because our efforts for Model Retraining did not yield fruitful results since the signal attenuation, fixed, and moving obstacles are different at each location (refer §10 for more details).

Each topology consists of 15 end nodes and one gateway. Since our region of interest is in the *gray region*, nodes are populated in this region to form a mesh topology. Data packets transmitted by the nodes in the *gray region* are considered for model training and testing. During this data collection experiment, the host Raspberry Pi will command both radios to transmit a 29-byte data packet concurrently. A total of 25,500 data packets were recorded. This comprehensive data set covers all the different dynamics of the deployed environment. The throughput of each transmitted data packet is recorded. This data set is manually labeled by a human to identify the high-throughput radio for each transmitted data packet. LoRa radios recorded the RSSI of the ACK sent by the gateway to acknowledge the previous data packet (refer §7.1 for more details). Zigbee radios sent beacon packets every 30ms to estimate and record local link qualities at each end node. These local link qualities were utilized to manually calculate the path quality metrics. This manually calculated path quality metric is then fed as an input to the ML models.

Three widely used classification models, Support Vector Machine (SVM), Logistic Regression (LR), and CART Decision Trees (DT) are trained and tested using the SKLearn library. In addition to these three models, the CART is further optimized with the Tree Alternating Optimization (TAO) algorithm [21]. The TAO-optimized CART model outperforms all the other models. The TAO-optimized CART model is deployed as IF...ELSE statements in IoT end devices. This model gets the input features from path quality estimations (see §7.1 and §8.2).

7.1 Problem formulation

The classification model takes the input features E2E path quality of LoRa radios ($E2E-PQ_{LoRa}$), and the E2E path quality of Zigbee radios ($E2E-PQ_{Zigbee}$) to output a high throughput radio. The input feature vector can be expressed as:

$$Input_i = [E2E - PQ_{LoRa}, E2E - PQ_{Zigbee}] \quad (5)$$

The output of the machine learning model is the radio predicted to have higher instantaneous throughput. This can be expressed as:

$$Output_i = [Zigbee|LoRa] \quad (6)$$

Feature selection and engineering. The classification problem formulated above needs $E2E-PQ_{LoRa}$, $E2E-PQ_{Zigbee}$ to predict the instantaneous high throughput radio. Traditional path-quality estimation in multi-hop Zigbee networks utilizes the link quality estimations of all the links along a path. The most common E2E path quality metrics used for multi-hop Zigbee networks are *Hop_Number* (HN), Packet Reception Ratio (PRR), Expected Transmission Count (ETX) [24], and Required Number of Packets (RNP) [20]. HN is the node's distance from the gateway in terms of hops. It is a discrete value ranging from [5,12] inclusively. It is obtained via the distance vector routing protocol run by Zigbee radios. The high data rate Zigbee Radio frequently transmits short beacon packets to estimate the path quality metrics. PRR is a well-known metric calculated as the ratio of the total number of packets received over the total number of packets sent by an end device. We calculate PRR, RNP and ETX over a window of size α . $\alpha=10$ gave us better results in our experiments, although this may vary for different deployments. ETX considers both forward and backward link qualities to calculate the metric. In our case, only the forward link quality is required to estimate E2E path quality estimation from an end device toward the gateway. So, the ETX becomes $1/PRR$, making this a redundant metric in the presence of PRR. RNP has the unique characteristic of capturing the underlying distribution of packet losses [20]. So

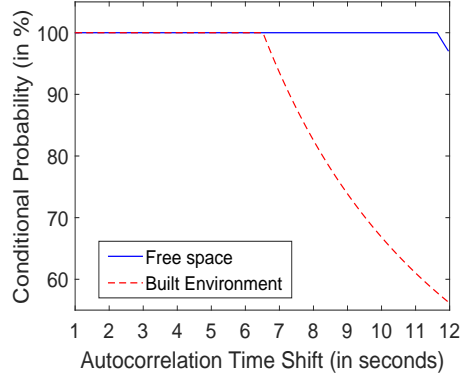


Fig. 6. RSSI auto-correlation of LoRa links

the HN, E2E PRR, and E2E RNP path quality metrics are considered for $E2E-PQ_{Zigbee}$. These metrics are calculated with frequent beacon packets since Zigbee is a high-data-rate radio.

On the other hand, the communication channels of low data-rate LoRa radio will be clogged if frequent beacon packets are sent. Hence the RSSI of the ACK sent by the gateway for the previous data packet is considered for estimating the end-to-end path quality of the LoRa radios ($E2E-PQ_{LoRa}$). This might be counter-intuitive because of two reasons:

- (1) How is it possible to predict the uplink channel quality based on a downlink packet? Quail [28] identifies and utilizes *Channel Reciprocity* between LoRa's uplink and downlink. This characteristic is leveraged to solve this issue.
- (2) How long is the uplink quality correlated?

How long is the uplink quality correlated? The inter-packet interval of data packets is 3 seconds. The time interval between the ACK corresponding to a previous data packet and the next data packet will be at least 2.5 seconds. If the metric related to a previous packet is utilized to predict the channel for the next data packet, it is not clear whether the uplink quality will be the same during this time interval. This question is answered through experiments.

An end node transmits a 29-byte packet to the gateway every second. The RSSI value of these packets is recorded at the gateway. The autocorrelation between the RSSI values of two packets is analyzed as the Conditional Probability of two events: These two packets have (i) the same RSSI values, (ii) different RSSI values. This experiment is conducted in both free space and built environments.

Figure 6 shows that the channel quality is correlated for 11.6 seconds in free space and 6.5 seconds in built environments during our experiments. So, if the packet generation interval between the data packets is less than the above-mentioned duration for our deployed environment, this feature works well as the end-to-end path quality indicator for LoRa. If no packet is sent for 6.5 seconds, we transmit a dummy packet to preserve this correlation.

After finalizing all the input features, the input feature vector eq. 5 becomes:

$$Input_i = [HN, E2E_RSSI_{LoRa}, E2E_PRR_{Zigbee}, E2E_RNP_{Zigbee}] \quad (7)$$

7.2 Prediction methods and results

The three widely used classification models, namely Logistic Regression (LR), Support Vector Machine (SVM), and CART Decision Trees (CART) were trained using the trace-driven data set obtained from large-scale real-world experiments from different topologies. An ML model is built offline for each location. MARS will deploy the chosen ML model in

Table 5. Training and Testing accuracy of different ML models and optimizations for all the topologies

ML Models and Optimizations	Location A				Location B	
	Line		Mesh		Mesh	
	Training Accuracy (in %)	Testing Accuracy (in %)	Training Accuracy (in %)	Testing Accuracy (in %)	Training Accuracy (in %)	Testing Accuracy (in %)
SVM	83.59±1.60	83.37±2.51	82.15±0.89	80.87±3.50	78.31±1.90	76.62±4.65
LR	83.65±1.01	83.87±3.566	82.59±0.83	81.25±3.46	81.21±0.57	80.50±2.21
CART	93.00±0.49	88.00±3.40	92.53±0.52	83.75±1.97	90.56±0.25	82.37±2.94
TAO-CART	93.00±0.56	88.00±2.33	89.375±0.26	85.625±4.12	94.71±0.38	93.87±1.69

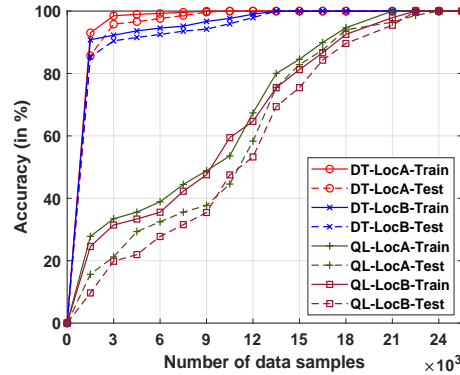


Fig. 7. Training data requirements

all the nodes in that specific topology. Their training and testing accuracy are averaged over 5-fold cross-validation on a data set based on real-world experiments. The testing and training accuracy tabulated in Table 5 shows that the TAO-optimized CART (TAO-CART) is better for deployment than the other widely used models.

CART Decision Tree Classifier [60] is a classical and one of the most popular algorithms to train a DT. The TAO algorithm [21], takes an initial tree, either generated randomly or induced by traditional algorithms (e.g. CART), and optimizes it jointly over the parameters of all the nodes in the tree. TAO works in alternating optimization fashion by cycling over different depths of a tree. At a given depth, TAO optimizes all nodes in that specific depth in parallel while guaranteeing a monotonic decrease of the desired objective function, such as misclassification errors. This fact backed with empirical evaluations [66] makes TAO an attractive algorithm.

The training and testing accuracy of the widely used classification models, namely SVM, LR, and CART, is tabulated in Table 5. From these results, it is clear that CART can achieve higher accuracy than SVM and LR. Compared with CART, TAO-CART achieves similar accuracy for the simple "Location A - Line topology" which are seldom used in real-world applications, and higher accuracy than CART for the "Location B-Mesh" topology. Also, the difference between the Training and Testing accuracy of TAO-CART is comparably smaller than that of CART for all the topologies. This means that TAO-CART optimization highly generalizes the model for unseen data.

Training data requirements. A major concern of a data-driven approach is the data required to train a model. Fig. 7 plots the accuracy of CART Decision Tree (DT) model and Q-Learning [27] (QL) as the function of required training data samples. In Fig. 7, we can see that the DT can quickly reach over 93% (Location A) and 90% (Location B) training accuracy with 1500 samples and show a gradual increase to 100% with 9000 (Location A) and 13500 (Location B) samples. Since there is only a gradual increase in accuracy after 1500 samples, we trade off accuracy to reduce the data collection effort. This trade-off reduces the deployment effort by 88.99%. So, all the models listed in Table 5 were trained with 1200 samples and tested on 300 samples. Parametric details of this QL are given in §9. A well-known issue of QL (model-free reinforcement learning) is the humongous training data requirement [23]. From Fig. 7, it is evident that

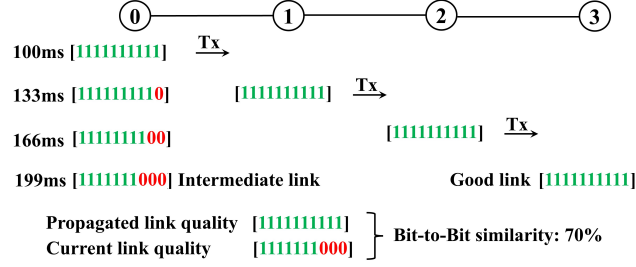


Fig. 8. Traditional path quality propagation is not instantaneous at all distances from the gateway.

using a DT model achieves significantly higher accuracy than QL with 1500 data samples. It should also be noted that QL needs an order of magnitude higher data samples to achieve the accuracy of DT. We optimized DT with TAO [21].

8 Realizing TAO-CART on IoT devices

The TAO-CART radio selector is found to achieve higher accuracy. However, realizing and deploying TAO-CART on an IoT end device entails the following challenges:

(i) *Instantaneous path quality estimations.* The TAO-CART radio selector needs instantaneous end-to-end path quality as an input to accurately predict the high-throughput radio. The models trained and tested in the previous section used a trace-driven data set, whose multi-hop end-to-end path qualities of Zigbee radios were manually calculated. In real-world deployments, the local link qualities should be propagated throughout the network for path quality estimation. However, this network-wide multi-hop link quality propagation may not be instantaneous at all distances from the gateway due to the chaotic link quality variations, queuing and propagation delays. We solve this challenge by developing a DT-based instantaneous path quality estimator (refer §8.2).

(ii) *Model size and Inference Latency.* Deploying TAO-CART with a minimal memory footprint on resource-constrained IoT devices is crucial. TAO-CART outputs a tree-like structure for predicting the high-throughput radio. This tree-like model, converted as IF..ELSE statements, take 36KB of memory on disk. In Raspberry Pi, this model takes 0.008ms for selecting the high-throughput radio. This deployment is highly feasible on resource-constrained IoT end devices because of the lower time and space complexity, although we are certain that this has a larger margin for optimization.

8.1 Traditional path quality estimations are not instantaneous

An intuitive idea to identify end-to-end throughput fluctuations is to utilize the end-to-end Path Quality (PQ) metrics. Since LoRa is a *single-hop* network, E2E PQ can be easily estimated. However, traditional PQ estimation for a *multi-hop* ZigBee network may not be instantaneous at all distances from the gateway.

Traditional PQ estimations for a *multi-hop* ZigBee network need the link quality estimation of all the links along the path. In section 7.2, TAO-CART was trained with a trace-driven data set, whose path qualities were manually calculated based on the recorded local link qualities. In practice, the qualities of all the links along the path should be propagated throughout the network so that they can be used to estimate the path quality. The packet in which these link qualities are propagated will experience delays at different hops due to (i) wireless link quality variations and (ii) packet processing delay. Packet processing delay includes read/write delays to append link quality sequences of the intermediate nodes, link-layer Acknowledgment delay, CSMA delay, propagation and queuing delays along a multi-hop network. So, the propagated local link quality may expire when it reaches a node estimating the metric.

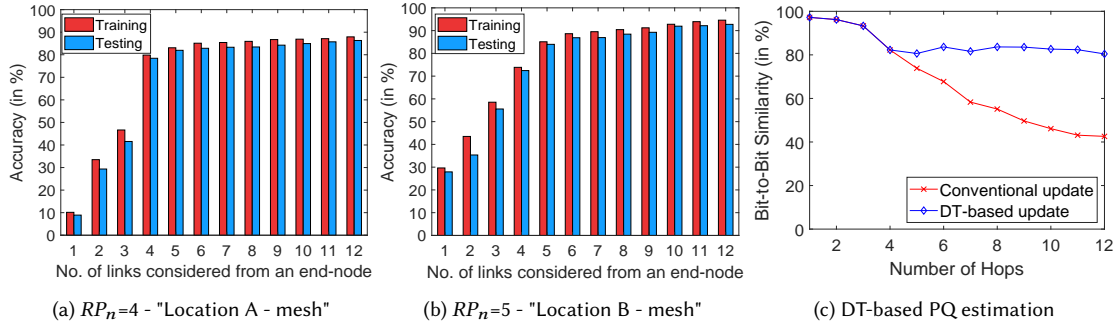


Fig. 9. DT-based PQ estimation. In figures 9a, 9b, the x-axis denotes the number of links considered to calculate PQ metrics.

The problem of traditional PQ estimation is depicted in Fig. 8. In this figure, Node 1 sends a beacon packet to Node 0 every 30ms. Node 0 stores the reception and loss of the beacons as a (1/0) binary bit-sequence respectively. It is identified through experiments that propagating a packet containing this link quality sequence takes 33ms on average for a single hop transmission accounting for all the above-mentioned delays while the network has fully functional control and data planes. Figure 8 depicts that the link quality sequence of link 1-0 propagated by Node 0 becomes invalid when it reaches Node 3. The bit-to-bit similarity of the link sequence of Node 0 at 199ms is not the same as propagated by Node 0. The bit-to-bit similarity reduces to 70% at the third hop. This problem will further be amplified if all the nodes along the path append their link quality sequence to this packet for network-wide advertisement. Since the gray region is 500m-1200m from the gateway, estimating path quality with expired link quality sequences will not be instantaneous, and this cannot act as an indicator of E2E throughput fluctuations.

8.2 DT-based path quality estimation

DT-based path quality estimation is developed to mitigate the problem of conventional path quality estimation for multi-hop networks. While the conventional path quality is calculated with Link Quality(LQ) information from all the links encompassed by the end-to-end path, DT-based path quality estimation requires LQ information only from a portion of the entire end-to-end path to compute the PQ metrics. This will highly reduce the delay incurred to propagate LQ information in a multi-hop Zigbee network. DT-based path quality estimation surfaces from the two important observations listed below: (i) The link quality of each link in the path may independently change based on the deployed environment and (ii) Each end node runs MARS to select a radio for transmitting the packet. A path from an end node to the gateway consists of multiple links. While it is intuitive to understand that LQ information from a portion of the entire end-to-end path is enough to compute PQ metrics, the challenge here is to identify the required path length, RP_n , so that the TAO-CART radio selector can accurately predict the high-throughput radio. For example, say a path from the end node to the gateway consists of 10 links. Traditional PQ estimation will use LQ information from all these 10 links to compute PQ metrics, whereas DT-based PQ estimation requires LQ information only from RP_n (<10) links to compute PQ metrics. The problem here is to define RP_n . We address this problem by training and testing decision trees with PQ metrics computed from different partial path lengths (RP_n) to understand the prediction accuracy of Decision Trees (DT).

A decision tree takes the end-to-end path quality metric of both the radios as input and chooses one radio for transmitting the packet. The following test is conducted to identify this Required Path length RP_n : The path quality



Fig. 10. Location B Mesh topology - nodes populated at the gray region.

Table 6. Thresholds identified to achieve higher throughput in the gray-region based on Lymberopoulos et al. [34]

	500-700m	700-1000m	1000-1200m
Zigbee RR	$\geq 77\%$	$\geq 80\%$	$\geq 83\%$
LoRa RSSI	≥ -72 dBm	≥ -71 dBm	≥ -71 dBm
Fallback radio	Zigbee	LoRa	LoRa

metrics are calculated with LQ information from the different number of links, from an end node towards the gateway. The training and testing accuracy of the models using LQ information from the different number of links along the path for mesh topologies at Locations A and B are depicted in Figures. 9a and 9b respectively. From these figures, it is inferred that there is a very gradual increase in accuracy after the fourth and fifth hops for the "LocationA-mesh" and "LocationB-mesh" topologies respectively. These decision trees are further optimized by the TAO algorithm. The above values are set to RP_n for our large-scale experiments. Figure 9c shows that the bit-to-bit similarity of conventional PQ estimation decreases with increase in hops while the bit-to-bit similarity of DT-based PQ estimation averages to 81%.

9 Large-scale experimental results

MARS is evaluated through large-scale real-world, mesh topology experiments conducted on our campus. Two mesh topologies are set up in two different locations as shown in Figs. 4 and 10. These topologies are deployed at locations with complex environments with different building materials and heavy human influx. Each node uses the hardware setup explained in §5.1.2. A total of 10,400 data packets were transmitted at both locations for this evaluation.

Benchmarks. The performance of MARS is compared with the single radio systems formed by (i) Zigbee-only and (ii) LoRa-only radios. MARS is also compared with two Multi-radio systems: (i) Q-learning-based radio selector [27] and (ii) Threshold-based radio selector [34], optimizing for energy efficiency. We made our best effort to adopt these works for throughput optimization. (i) *The threshold-based algorithm* [34] is based on the break-even points identified through experiments similar to Fig. 2b. The thresholds are set based on a node's distance from the gateway and the instantaneous end-to-end path quality of the radios. End-to-end reception ratio and RSSI are used as the instantaneous path-quality indicator for the multi-hop Zigbee and the single-hop LoRa radios respectively. These thresholds are identified from the traces obtained from real-world experiments. This threshold-based algorithm divides the gray region into three sub-regions 500-700m, 700-1000m, and 1000-1200m based on grouping similar indicators achieving higher throughput. Thresholds are set based on these sub-regions for a fair comparison. Thresholds tabulated in Table 6 are identified via experiments to achieve higher throughput in each sub-region. A radio having end-to-end reception ratio greater than 77%, 80%, and 83% achieves higher throughput in the sub-regions 500-700m, 700-1000m, and 1000-1200m

respectively while LoRa's RSSI ≥ -72 dBm, ≥ -71 dBm and ≥ -71 dBm achieve higher throughput in the sub-regions 500-700m, 700-1000m, and 1000-1200m respectively. So, in the case of one radio performing better than the other, it will be indicated by the thresholds, eventually selected for transmitting the packet. From the experimental traces, the fall-back radio is identified to achieve better throughput if either both radios fall inside or outside the defined threshold region.

(ii) Q-learning-based radio selector [27] is trained with 1500 samples since the TAO-CART model of MARS is also trained with 1500 samples. Comparing with the TAO-CART model of MARS, Q-Learning needs an order of magnitude higher data to converge (refer Fig. 7). We still compare the performance of MARS and Q-learning-based radio selector [27] to understand the overall throughput performance of the models trained with the same amount of training data. We follow the same strategy devised by Gummesson et al. [27] to set Q-learning model parameters except the changes described below. They optimize for lower energy consumption. Since energy is a cost, not a reward, they consider their reward as " $-energy$ ". In our case, we optimize for higher throughput, so our reward is *throughput*. Gummesson et al. [27] *initially* has two states, one for each radio. Since the transmit power level is *one of the factors* directly affecting energy consumption, this two-state model can be expanded to an n-state model where each state represents a radio at a particular transmit power level. For example, four states are required for two radios, each with two transmit power levels. They also identify that an increase in the number of states increases exploration overhead or decreases exploration frequency. They solve this issue by considering only three states at a time.

In our case, throughput/latency is directly affected by the link/path quality. We represent the link/path quality using the input feature vector described in §7.1. Unlike the countable/controllable transmit power levels of Gummesson et al. [27], the link/path quality states are numerous and uncontrollable. This significantly increases the number of states leading to an increase in exploration overhead. We optimized the model's hyper-parameters through grid search. Also, the sophisticated RL-based radio switching protocol of Gummesson et al. [27] suffers from the below-described problems: (i) During data transfer between two radios in the communication range, the radio switching protocol, designed for energy efficiency, does a three-way handshake to switch radios. This incurs additional latency which will heavily degrade the throughput. (ii) A well-known issue of model-free RL is that it requires heavy training data to converge to an acceptable performance [23] and the amount of data samples used by Gummesson et al. [27] for training is obscure. We compared the training data requirements of Q-learning and TAO-CART Decision Tree models in §7.2.

One can argue that our design of Q-Learning for throughput maximization can still be optimized, although it will be incremental. Our Q-Learning model provides the best results after multiple iterative efforts. Moreover, our ability to optimize is also restricted in order to respect the originality of their work. It is to be noted that the above-mentioned works optimize for energy consumption. In addition, Gummesson et al. [27] use custom hardware. The performance of Gummesson et al. [27] without their custom hardware is obscure. However, we made our best effort to adopt Gummesson et al. [27] for throughput optimization without custom hardware.

9.1 Performance evaluation

The throughput gain of MARS is shown in Figure 11. The Q-learning-based radio selector [27] achieves the least throughput gain of all the Multi-radio systems because of the three-way handshake and limited training data.

The threshold-based radio selector [34] fails to identify LoRa as a high throughput radio for 19% of the transmissions. Also, the threshold-based radio selector converges towards the high-throughput Zigbee only after 60% of the transmissions. This is because the threshold-based radio selector is not able to identify the high-throughput radio when

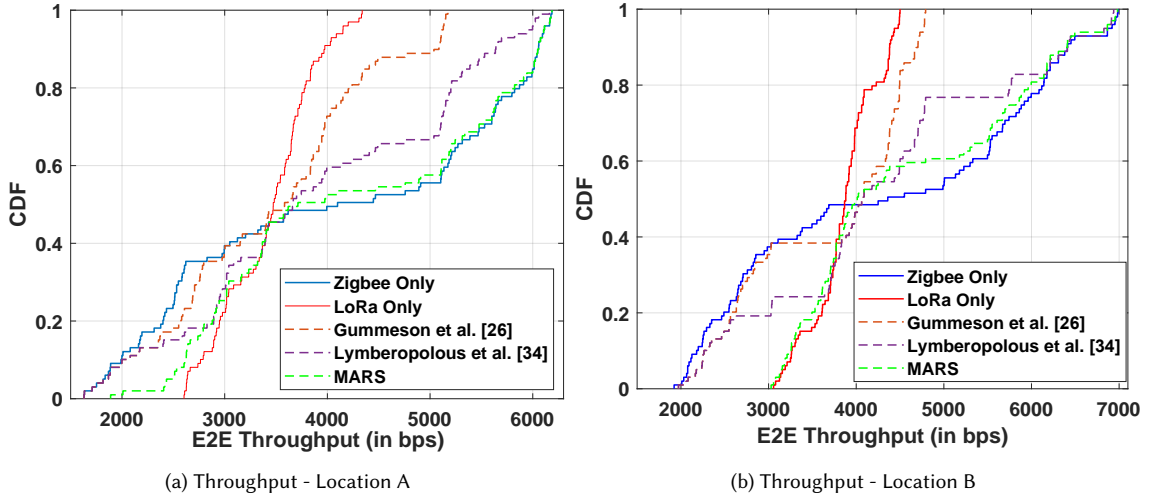


Fig. 11. Throughput gain of MARS

both radios fall inside the threshold region. The identified fallback radios do not achieve higher throughput all the time because of erratic link quality variations.

MARS closely follows the high-throughput radio as it instantaneously identifies the throughput fluctuations. In this location, the threshold-based radio selector algorithm achieves an average throughput gain of 18.79% and 15.31% than Zigbee-only and LoRa-only networks respectively. Whereas, MARS achieves an average throughput gain of 55.93%, 57.22%, and 36.32% than Zigbee-only, LoRa-only, and Threshold-based radio selector networks respectively.

Fig. 11b shows similar trends in Location B. In this location, all three multi-radio systems tend to cross the solid red line unlike Fig. 5, where the optimal performance was obtained offline through a trace-based evaluation. Hence, the simulated optimal multi-radio performance has to be chosen from any one of the available throughputs making the golden dashed lines of Fig. 5 to stay within the solid lines. The performance evaluations were conducted in real-world deployments. The channel conditions may not be identical when each system is evaluated. This change in channel conditions led to a small difference in throughput making the multi-radio systems cross the solid line. In this location, the threshold-based radio selector algorithm achieves an average throughput gain of 16.06%, and 19.57% than Zigbee-only and LoRa-only networks respectively. MARS achieves an average throughput gain of 53.77%, 58.34%, and 32.49% than Zigbee-only, LoRa-only, and Threshold-based multi-radio networks respectively.

MARS on different packet generation intervals. The performance ratio is calculated as the ratio of the average per-packet throughput of MARS over the average per-packet throughput of the best radio. The Performance ratio for different packet generation intervals is plotted in Fig. 12. We observed a slight decrease in the performance ratio from packet generation interval 5s - 1.5s and a significant decrease in performance ratio for intervals 1.4s and 1.3s. This directly reflects on the average latency. MARS is able to achieve the required latency for intervals 5-1.4s, while MARS fails to achieve the required latency for interval 1.3s. This degradation in MARS's performance is due to the increase in packet generation rate leading to queuing delays in the network. This increase in queuing delay affects the beacon packets used for local link quality and path quality estimations. This directly affects the required fine-grained link/path quality estimations.

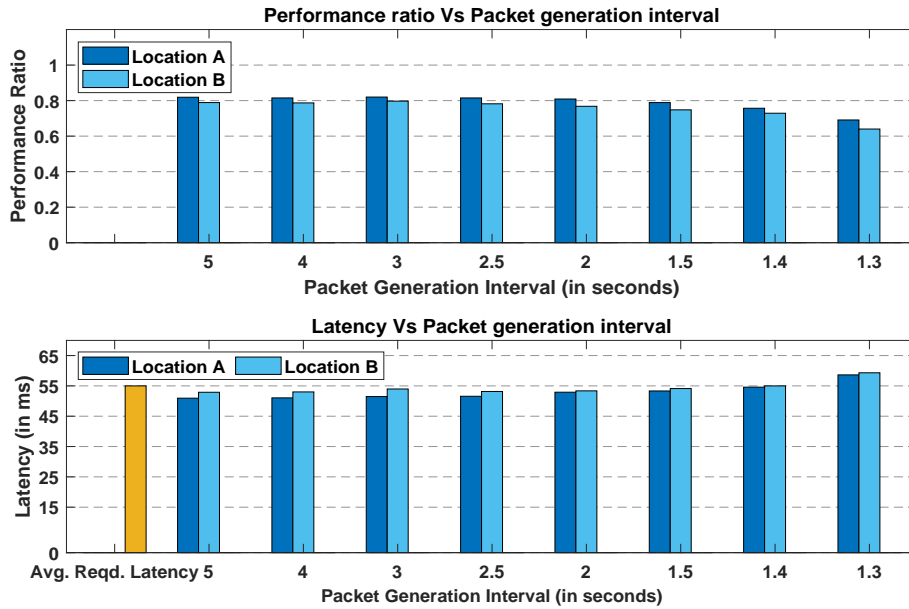


Fig. 12. MARS on different packet generation intervals.

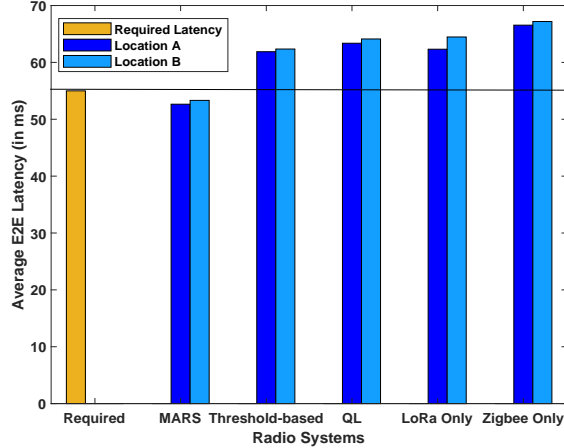


Fig. 13. MARS achieves required average latency

Average required latency for mesoscale IoT. According to 5G America’s report [6], the average required latency for mesoscale IoT applications is 55ms. Fig. 13 shows the average latency achieved by different radio systems at two different locations. LoRa-only and Zigbee-only networks are not able to achieve the required latency in the gray region. MARS is able to achieve an average latency of 51.48ms and 53.98ms in Locations A and B respectively. MARS achieves the goal while the other radio systems fail.

10 Discussion

Energy Overhead: MARS relies on the neighbor discovery beacon packets sent every 30ms for local link quality estimation. The reception/loss of a neighbor discovery beacon, stored as a bit array, is appended to the routing table

when the distance vector protocol periodically broadcasts the routing table. We piggyback on neighbor discovery beacons for local link quality estimation. This optimizes both energy consumption and queuing delays. However, appending this bit array to routing table broadcast packets leads to transmission overhead. On average, the energy toll of this additional transmission overhead per data packet is 13.60% of the energy consumed for a data packet transmission. If the neighbor discovery beacons use Trickle timer [33], additional beacon packets would be needed for channel estimation, significantly increasing energy consumption. In real-world mesoscale IoT applications like P2P energy trading of smart meters in the smart neighborhood and Industrial/Factory automation, the IoT radio utilized here will be a small component of a larger device. Unlike the typical IoT nodes powered by a smaller battery, these devices are either grid-powered [57, 65] or have a large battery reserve to support the entire device [14–16] making this energy overhead tolerable for these applications.

Model Retraining: We evaluated MARS at different seasons of the year. The above results were obtained by deploying the ML model that was trained with data collected in the Summer and it was evaluated during the same season. Several factors affect the dynamicity of link and path qualities that makes the input feature vector of the ML model. For example, the node placement, the impact of temperature and pressure variations, and the effects of fleeting reflectors. Through experiments, we were able to find that a model developed during one season was yielding poor performance for a different season that has completely different temperatures and fleeting reflectors. A user should be able to collect data required for MARS in 4 hours including topology setup and data collection. The TAO-trees used in MARS can be trained with CPUs within a few minutes. Since the effort for deploying MARS is reasonable, it should be easier to develop a model for each season, that can be used for the upcoming years. In future work, we plan to add a more detailed analysis of the retraining gap and the quality of estimation.

If MARS is deployed in deserts where the temperature is very high during the day and very low during the night with predictable temperature increases and decreases, MARS can be equipped with two models, one for day time and another for night time. MARS can then adapt between these models based on the readings from a temperature sensor. If MARS is deployed in a region where the temperature changes drastically, like the islands of Bora Bora and Maldives, the data collection, training and deployment effort for any ML model will be significant, including MARS. It is hard to design a robust ML-based system for such environments irrespective of the effort made for data collection, training and deployment.

Model Adaptation: Model adaptation is a well-known technique for ML systems to utilize the ML model developed for one location to build the ML model for the other location with reduced data collection and training effort. This is inevitable for training large ML models like LLM’s or autonomous driving that require significant time and resources for training a model. However, the effort and resources required for MARS are minimal. So, MARS does not require model adaptation. Each location can train MARS’s model to accurately reflect the characteristics of the deployed environment.

Multi-channel Zigbee is detrimental for MARS. The drop in E2E throughput performance from 100-300m from the gateway shown in Figure 2b happens because 3 nodes are contending in a single channel. One could argue that using multi-channel Zigbee will improve the throughput. In general, using multi-channel Zigbee might reduce contention delays and improve throughput for applications that does not focus on throughput and latency. The reason we did not choose multi-channel Zigbee are two-folds: (i) Channel switching needs the transmitting radio to switch to another channel and the receiving radio should also switch to the same channel to receive and process packets. This is could lead to delays. In worst case, it will be similar to the three-way handshake during the radio switching of Gummesson et al. [27] (refer §9). (ii) MARS needs fine-grained estimation for the multi-hop Zigbee radios and our deployments uses the $\alpha=10$. This means a node switching channels should send 10 beacon packets to obtain the fine-grained information

of the channels quality, each and every time during a channel switch. The above-mentioned delays combined together will cause a significant delay that will diminish the throughput performance significantly. For mesoscale applications like P2P energy trade in smart-homes where multiple nodes has to bid and respond quickly to a bid, similar to a stock market transaction, the above-mentioned delays will diminish the application performance. So, we decided not to use multi-channel Zigbee.

11 Future Work and Conclusion

This work represents the initial foray into multi-radio architectures for mesoscale IoT applications. There are many aspects we would like to continue developing. On the modeling aspect, we are experimenting with TAO-optimized oblique trees for radio selection. While the accuracy could be further improved, the benefits are more related to tree stability when adding additional training data since with axis-aligned trees the tree structure can change dramatically. This can help gain an understanding of the fundamental properties of the system. In addition, we would also like to explore different input features for the decision trees. In particular, we would like to try to feed raw data packet sequences instead of path-quality estimation metrics, and let the model find patterns not easily detected by a human designer on the raw data. We would also like to incorporate the latest LMAC [52] for LoRa to enhance the throughput.

To conclude, we presented MARS a Multi-radio Architecture with Radio Selection using Decision Trees. The system deploys MARS' nodes in a multi-hop network. Each of these nodes has two radios, a Zigbee and a LoRa RF transceiver with different throughput/latency features that are optimized for different ranges. The final goal of MARS is to select the best radio to be used at any point in the network, using different network paths and link-layer metrics gathered from the radios, to maximize the end-to-end throughput of the data packets being transmitted. The radio selection is done using novel TAO-optimized decision trees, which are easy to train with limited training data, and easy to deploy in an IoT end device with limited computational power. In addition, we show that collecting local path metrics as input to our decision trees provides sufficient information to identify the high-throughput radio over the entire path. MARS is evaluated on a large-scale complex mesh topology at two different locations. The results show that MARS can identify the high-throughput radio at the time of transmission. This leads to an average throughput gain of 48.2% and 49.79% than the competing schemes, in both locations, while achieving the required latency.

References

- [1] [n. d.]. CC2520 datasheet. <http://tinyurl.com/pf6ecmr6>.
- [2] [n. d.]. SIGFOX System Description. <http://tinyurl.com/45tbwraf>.
- [3] Norman Abramson. 1977. The throughput of packet broadcasting channels. *IEEE Transactions on Communications* (1977).
- [4] Yuvraj Agarwal, Trevor Pering, Roy Want, and Rajesh Gupta. 2008. Switchr: Reducing system power consumption in a multi-client, multi-radio environment. In *2008 12th IEEE International Symposium on Wearable Computers*. IEEE, 99–102.
- [5] Mansoor Alicherry, Randeep Bhatia, and Li Li. [n. d.]. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *ACM MobiCom, 2005*.
- [6] 5G Americas. 2019. *5G - The future of IoT*. <http://tinyurl.com/26y5epsp>
- [7] Ganesh Ananthanarayanan and Ion Stoica. [n. d.]. Blue-Fi: enhancing Wi-Fi performance using bluetooth signals. In *ACM MobiSys '09*.
- [8] Paramvir Bahl, Atul Adya, Jitendra Padhye, and Alec Wolman. 2004. Reconsidering wireless systems with multiple radios. *ACM SIGCOMM Computer Communication Review* 34, 5 (2004), 39–46.
- [9] Mehdi Bennis, Meryem Simsek, Andreas Czylik, Walid Saad, Stefan Valentin, and Merouane Debbah. 2013. When cellular meets WiFi in wireless small cell networks. *IEEE communications magazine* 51, 6 (2013), 44–50.
- [10] R. Boorstyn and et al. 1987. Throughput analysis in multihop CSMA packet radio networks. *IEEE Transactions on Communications* (1987).
- [11] Sylvain Cherié, Zahra Movahedi, and Yacine M Ghamri-Doudane. 2015. Multi-tenancy in decentralised IoT. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 256–261.

- [12] Shuo Deng, Ravi Netravali, Anirudh Sivaraman, and Hari Balakrishnan. 2014. WiFi, LTE, or both? Measuring multi-homed wireless Internet performance. In *Proceedings of the 2014 Conference on Internet Measurement Conference*. 181–194.
- [13] Richard Draves, Jitendra Padhye, and Brian Zill. [n. d.]. Routing in multi-radio, multi-hop wireless mesh networks. In *ACM MobiCom, 2004*.
- [14] Mitsubishi Electric. 2024. Large-capacity battery control system. <http://tinyurl.com/2ryvbcvx>.
- [15] Mitsubishi Electric. 2024. Smart meter system. <http://tinyurl.com/4z2dy6nc>.
- [16] Mitsubishi Electric. 2024. "BLEnDer®". <http://tinyurl.com/3k435ymu>.
- [17] Al Islam et al. 2011. Backpacking: Deployment of heterogeneous radios in high data rate sensor networks. In *IEEE ICCCN '11*.
- [18] Baccour et al. 2012. Radio link quality estimation in wireless sensor networks: A survey. *ACM TOSN* 8, 4 (2012), 1–33.
- [19] Cheng et al. 1989. A loop-free extended Bellman-Ford routing protocol without bouncing effect. *ACM SIGCOMM Computer Communication Review* (1989).
- [20] Cerpa et al. 2005. Temporal properties of low power wireless links: modeling and implications on multi-hop routing. In *In MobiHoc '05*.
- [21] Carreira-Perpinán et al. 2018. Alternating optimization of decision trees, with application to learning sparse oblique trees. In *NeurIPS '18* (2018).
- [22] Ding et al. 2019. Constrained broadcast with minimized latency in neighborhood area networks of smart grid. *IEEE Transactions on Industrial Informatics* (2019).
- [23] Ding et al. 2020. Mb2c: Model-based deep reinforcement learning for multi-zone building control. In *ACM BuildSys'20*.
- [24] De Couto et al. 2003. A high-throughput path metric for multi-hop wireless routing. In *MobiCom '03*.
- [25] Ferrari et al. 2011. Efficient network flooding and time synchronization with glossy. In *ACM/IEEE IPSN'11*.
- [26] Ganti et al. 2006. Datalink streaming in wireless sensor networks. In *ACM SenSys'06*.
- [27] Gummeson et al. 2009. An adaptive link layer for range diversity in multi-radio mobile sensor networks. In *IEEE INFOCOM '09*.
- [28] Gadre et al. 2020. Quick (and dirty) aggregate queries on low-power wans. In *ACM/IEEE IPSN'20*.
- [29] Huang et al. 2012. The evolution of MAC protocols in wireless sensor networks: A survey. *IEEE communications surveys & tutorials* 15, 1 (2012), 101–120.
- [30] Hamdy et al. 2020. Evaluation of ZigBee topology effect on throughput and end to end delay due to different transmission bands for IoT applications. *Journal of Communications Software and Systems* 16, 3 (2020), 254–259.
- [31] Jurdak et al. 2011. Opal: A multiradio platform for high throughput wireless sensor networks. *IEEE Embedded Systems Letters* 3, 4 (2011), 121–124.
- [32] Kusy et al. 2014. Radio diversity for reliable communication in sensor networks. *ACM TOSN* 10, 2 (2014), 1–29.
- [33] Levis et al. [n. d.]. Trickle: A self-regulating algorithm for code maintenance and propagation in wireless sensor networks. In *USENIX/ACM NSDI'04*.
- [34] Lymberopoulos et al. 2008. Towards energy efficient design of multi-radio platforms for wireless sensor networks. In *ACM/IEEE IPSN '08*.
- [35] Le et al. 2017. Bluetooth® low energy and the automotive transformation. *TI* (2017).
- [36] Liu et al. 2019. Intraday residential demand response scheme based on peer-to-peer energy trading. *IEEE Transactions on Industrial Informatics* (2019).
- [37] Lee et al. 2021. WiFi HaLow for Long-Range and Low-Power IoT: System on Chip Development and Performance Evaluation. *IEEE Communications Magazine* (2021).
- [38] Milenkovic et al. 2005. An environment for runtime power monitoring of wireless sensor network platforms. In *In IEEE SSSST'05*.
- [39] Ojo et al. 2018. Throughput maximizing and fair scheduling algorithms in industrial Internet of Things networks. *IEEE Transactions on Industrial Informatics* (2018).
- [40] Petäjajarvi et al. 2017. Evaluation of LoRa technology for indoor remote health and wellbeing monitoring. *Intl. Journal of Wireless Information Networks* (2017).
- [41] Pilz et al. 2017. Recent advances in local energy trading in the smart grid based on game-theoretic approaches. *IEEE Transactions on Smart Grid* (2017).
- [42] Sisinni et al. 2018. Industrial internet of things: Challenges, opportunities, and directions. *IEEE transactions on industrial informatics* (2018).
- [43] Sundaram et al. 2019. A survey on LoRa networking: Research problems, current solutions, and open issues. *IEEE Communications Surveys & Tutorials* (2019).
- [44] Saif et al. 2022. Impact of distributed energy resources in smart homes and community-based electricity market. *IEEE Transactions on Industry Applications* (2022).
- [45] Vasisht et al. 2017. {FarmBeats}: an {IoT} platform for {Data-Driven} agriculture. In *USENIX NSDI'17*.
- [46] Yeow et al. 2007. Energy efficient multiple target tracking in wireless sensor networks. *IEEE Transactions on Vehicular Technology* 56, 2 (2007), 918–928.
- [47] Yousuf et al. 2018. Throughput, coverage and scalability of LoRa LPWAN for internet of things. In *IEEE/ACM IWQoS'18*.
- [48] Zuniga et al. 2004. Analyzing the transitional region in low power wireless links. In *IEEE SECON '04*.
- [49] Zhang et al. 2019. Throughput optimization with delay guarantee for massive random access of M2M communications in industrial Internet of Things. *IEEE IoT Journal* (2019).
- [50] Simone Ferlin, Thomas Dreiholz, and Özgü Alay. [n. d.]. Multi-path transport over heterogeneous wireless networks: Does it really pay off?. In *IEEE GLOBECOM, 2014*.
- [51] Ruijie Fu, Lancong Guo, An Zou, Cailian Chen, Xiping Guan, and Yehan Ma. 2024. Comprehensive Optimal Network Scheduling Strategies for Wireless Control Systems. *ACM Transactions on Cyber-Physical Systems* (2024).

- [52] Amalinda Gamage, Jansen C. Liando, Chaojie Gu, Rui Tan, and Mo Li. 2020. LMAC: Efficient Carrier-Sense Multiple Access for LoRa. In *Proceedings of the the 26th Annual International Conference on Mobile Computing and Networking (MobiCom '20)*. ACM.
- [53] Tao Jin, Guevara Noubir, and Bo Sheng. [n. d.]. Wizi-cloud: Application-transparent dual zigbee-wifi radios for low power internet access. In *IEEE INFOCOM, 2011*.
- [54] HP Limited. 2009. CeNSE. <https://www.hpl.hp.com/news/2009/oct-dec/cense.html>.
- [55] Ronoth LLC. [n. d.]. LoStik. <https://github.com/ronoth/lostik>.
- [56] Yehan Ma, Yebin Wang, Stefano Di Cairano, Toshiaki Koike-Akino, Jianlin Guo, Philip Orlik, Xinpeng Guan, and Chenyang Lu. 2022. Smart actuation for end-edge industrial control systems. *IEEE Transactions on Automation Science and Engineering* 21, 1 (2022), 269–283.
- [57] Yukimasa Nagai, Jianlin Guo, Takenori Sumi, Kieran Parsons, Philip Orlik, Benjamin A Rolfe, and Pu Wang. 2024. Improve IEEE 802.15. 4 Network Reliability by Suspendable CSMA/CA. In *2024 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1–6.
- [58] Trevor Pering, Yuvraj Agarwal, Rajesh Gupta, and Roy Want. [n. d.]. Coolsports: reducing the power consumption of wireless mobile devices with multiple radio interfaces. In *ACM MobiSys '06*.
- [59] Christopher Pluntke, Lars Eggert, and Niko Kiukkonen. [n. d.]. Saving mobile device energy with multipath TCP. In *In ACM MobiArch, 2011*.
- [60] Praagman. 1985. Classification and regression trees.
- [61] Mattia Rizzi, Paolo Ferrari, Alessandra Flammini, Emiliano Sisinni, and Mikael Gidlund. 2017. Using LoRa for industrial wireless networks. In *2017 IEEE 13th international workshop on factory communication systems (WFCS)*. IEEE, 1–4.
- [62] S Sastry, D Culler, M Howard, T Roosta, B Zhu, J Taneja, Sukun Kim, S Schaffert, J Hui, P Dutta, et al. 2006. Instrumenting wireless sensor networks for real-time surveillance. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. IEEE, 3128–3133.
- [63] Sanjib Sur, Ioannis Pefkianakis, Xinyu Zhang, and Kyu-Han Kim. [n. d.]. WiFi-assisted 60 GHz wireless networks. In *ACM MobiCom, 2017*.
- [64] Crossbow Technology. [n. d.]. TelosB. <http://tinyurl.com/2svnbhjs>.
- [65] Kang Yang, Miaomiao Liu, and Wan Du. 2024. : Rateless-Enabled Link Adaptation for LoRa Networking. *IEEE/ACM Transactions on Networking* (2024).
- [66] Arman Zharmagambetov and Miguel Á. Carreira-Perpiñán. [n. d.]. Smaller, More Accurate Regression Forests Using Tree Alternating Optimization. In *ICML'20* ([n. d.]).