Wireless Networks
https://doi.org/10.1007/s11276-019-02146-9

Dynamic modulation scaling enabled multi-hop topology control for time critical wireless sensor networks

Arda Gumusalan[1] · Robert Simon[1] · Hakan Aydin[1]

Abstract
The previous work on connection driven topology control has shown that it has significant potential to reduce energy consumption of Wireless Sensor Networks (WSNs). Dynamic Modulation Scaling (DMS) which is a technique that manages transmission power levels in order to change the number of bits encoded per symbol has a direct impact on connection driven topology control. In this paper we investigate the transmission scheduling of multi-hop real-time WSNs equipped with DMS enabled radio chips while taking the effect of DMS on topology control into account. To our best knowledge, this is the first paper that addresses this issue. The current work on DMS enabled WSN tend to rely on theoretical DMS models to predict network performance metrics. However, there is little, if any, work that is based upon empirically verified network performance outcomes using DMS especially on its effect on connection driven topology control. This paper fills this gap by using GNU Radio and Software Defined Radio hardware to show how to emulate DMS in low power wireless systems and measure the impact of varying Signal-to-Noise levels, distance and elevation on throughput and delivery rates for different DMS control strategies. Next, we present the Mixed Integer Nonlinear Optimization Problem of minimizing energy consumption of DMS enabled connection driven topology control on real-time WSNs. Lastly, we present two polynomial time heuristics and compare their performance against the optimal solution.

1 Introduction

Wireless Sensor Networks' (WSN) usage in industrial applications such as industrial control networks (ICN), critical infrastructure monitoring has been rising in recent years. This is a paradigm shift since these systems traditionally have been installed with wired channels between communicating devices. Wireless system cuts costs due to the elimination of expensive cables and related maintenance fees [18]. Protocols such as IEEE 802.15.4, WIA-PA, WirelessHART, and ISA100.11a are currently commonly deployed wireless ICN protocols in fields such as manufacturing, electrical generation, and chemical refining [15].

✉ Arda Gumusalan
agumusal@gmu.edu

Robert Simon
simon@gmu.edu

Hakan Aydin
aydin@gmu.edu

[1] Department of Computer Science, George Mason University, Fairfax, VA 22030, USA

DMS, also known as Adaptive Modulation, is a technique which manipulates the constellation size of a modulation schema in order to change the transmission time and energy consumption. Higher (lower) number of bits transmitted per symbol reduces (increases) the transmission time. However, encoding more bits per symbol with constant bit error rate requires an increase in the transmission output power. DMS is commonly used in many telecommunications systems and offers great potential in low-power communication in terms of reducing energy consumption and/or increasing reliability especially for applications requiring high data rates over long ranges [34]. For instance, IEEE 802.15.4k, also known as LECIM [35], aims to establish connection up to 20 km. The choice of modulation for LECIM is the Gaussian Frequency Shift Keying (GFSK). The authors of [34] emphasize the need for carefully selecting the modulation technique; an example is deciding between GFSK, binary phase shift keying (BPSK) and offset quadrature phase shift keying (O-QPSK). DMS is also a core component of IEEE 802.11ah which supports BPSK, QPSK, 64QAM (Quadrature Amplitude Modulation), and 256QAM [6]. It must be noted that higher modulation sizes used in high

Published online: 24 September 2019

data rate applications bring higher Signal-to-Noise Ratio (SNR) requirements.

DMS has also been studied in wireless ICN domain. Wireless ICN protocols are typically organized as cluster, star topology, or multi-hop topologies. To ensure real-time and predictable performance, variations of time division multiple access (TDMA) based scheduling and link access mechanisms are widely adopted [9]. In these deployments, a coordinator is in charge of distributing the time-slots where only the owner(s) of that time-slot is(are) allowed to talk. TDMA also provides additional energy savings since the nodes can sleep except when they are scheduled to transmit or receive [30].

A detailed description of existing work on DMS is presented in Sect. 2. In particular, DMS has been the subject of several research articles in the Wireless Sensor Network (WSN) domain [4, 10, 13, 16, 17, 36, 37]. However, due to the lack of commercially available hardware, these evaluations were typically conducted via simulation and/or using the pre-supposed close-form energy scaling formulas. This paper has two main goals; (1) This paper aims to fill the gap in the empirical analysis of DMS when applied in the WSN domain and (2) Provide optimal interference-aware time-slot allocation formulation for DMS-enabled, time-critical, multi-hop WSNs. We are particularly interested in investigating the following questions: Given a set of environmental conditions, modulation techniques and power levels, *how do the packet delivery rates change with DMS* and *how does it affect parent node selection process for time critical multi-hop WSNs*?

To answer the first question, we have used Ettus B210 Software Defined Radios [31] (SDRs) and configured them according to IEEE 802.15.4 [21] base band, symbol rate, data rate and samples per symbol settings. Using this configuration we make the following contributions. First, we show how to emulate DMS in low-power networks using existing GNURadio blocks and SDR hardware. We also provide a detailed look into the signal recovery process which maybe used for future researchers to expand our layout. The next three contributions are meant to model the various types of environmental transmission conditions to which wireless systems may be subject. We vary the Signal-to-Noise-Ratios (SNRs) of {2, 4, 8, 16} PSK and DPSK modulations' of the channel to observe how packet delivery rates (PDRs) change. This approach gives a detailed insight on how the output power can be managed to achieve the desired PDR. These tests required a controlled noise environment with a noise generator, and therefore was conducted using Faraday Cages. We compare our findings with what existing general scaling function and conclude that the necessary energy increase is greater than what the models suggest when we increase the modulation levels. Furthermore, we repeat the same experiments with Differential-PSK and report a very significant increase in performance. We then conduct a set of distance tests for {2, 4, 8} DPSK and measured PDRs for distances up to 100 m. Distance tests are of utmost importance for multi-hop networks and provide great insight into how it can be used to set up more energy and/or latency efficient topology control. Finally, we test the impact of elevation difference between transmitter and receiver which is a common scenario for applications such as residential sensor monitoring, where some nodes are placed on objects such as lampposts.

To answer the second question, we first present the Mixed Integer Nonlinear Programming problem formulation of the problem (and later in Sect. 7 explain how to convert it to linear programming problem); *given a set of connections between nodes per modulation level, choose a modulation level and a parent for each node to talk to while minimizing the total network energy consumption given a maximum delay bound of when the base station has to collect all the information from the network.* This problem description assumes the links between the nodes *per modulation level* are known, in other words, it assumes given a node, let's say $node_a$, the set of nodes that $node_a$ can talk to using modulation levels 1, or 2, or 3,… is known. These set of nodes per modulation level are not necessarily identical. Generally speaking, mixed integer programming problems are known to be intractable. Interference aware multi-hop scheduling has been studied in literature and proven that a throughput optimal TDMA schedule is an NP-complete problem even for linear topologies [1] and [14] proves that interference and energy aware speed assignment problem is NP-hard. This led us to provide two polynomial time heuristics and compare them against the optimal solution using the simulation we built by incorporating the data we obtained from our real world tests.

In summary this paper has the following contributions; (1) we describe a new perspective on how DMS can be applied to WSNs. The papers in literature typically assume discrete and single power levels for each modulation level. While this is valid, it is very limited and not necessarily the only valid model. It is possible to adjust power output levels independently from the modulation levels that is used. To our best knowledge, this is the first paper to use this extended model in DMS enabled WSN domain. (2) We run extensive SDR based DMS tests on various conditions and obtain the necessary data to form the basis of our new extended model. (3) We show the optimal MINLP formulation of DMS enabled parent node selection problem for time critical WSNs (later show how to convert it into a MILP formulation) and (4) we propose two polynomial time heuristics and compare them against the optimal solution.

2 Related work

The pioneering work presented in [32] led many researchers to consider DMS as an important leverage tool for low-power wireless communication. The authors presented the underlying principals of DMS and demonstrated how it can be used as an energy management technique. Some of these techniques are presented in Sect. 3. In the same paper, the authors also propose packet scheduling algorithms for real-time and non-real-time communication using DMS.

The work in [10] proposes modulation optimization algorithms for MQAM and MFSK schemas. The authors first break down the overall energy consumption of DMS into each hardware component's energy consumption and then analyze energy consumption per bit for different modulation levels. This paper was also one of the first to point out that the effectiveness of DMS depends on transmission distances. It has been shown that for sufficiently large distances, the lowest modulation level is the most energy efficient. However, for shorter distances this *may not* be the case, considering the increased weight of electronic circuitry consumption compared to that of the wireless radio.

DMS has also been studied in Wireless Sensor Network (WSN) domain, and this is the main focus of our paper. The authors of [36] studied integration of DMS into real-time data gathering for tree-topology based WSNs. They proposed an offline optimization algorithm as well as a distributed online algorithm for efficient modulation level assignments and their simulation results have shown up to 90% energy savings. However, the proposed solutions were not applicable to TDMA-based protocols and assumed very minimal collision detection delay with light-weight traffic scenarios.

The work in [37] also incorporated DMS into tree-topology based WSNs. The novelty of the paper came from combining energy harvesting with optimal modulation level selection. The authors proposed a centralized optimal as well as a distributed near-optimal solution.

The authors of [13] proposed a slack generation and reclamation mechanism for real-time WSNs. They generated slack by analyzing data redundancies and by eliminating redundant data transmissions which in return created slack time. They then used this slack time to lower modulation levels to reduce energy consumption while meeting deadlines. The same research group later formulated joint scheduling of computation and communication tasks with DVS and DMS respectively as Mixed Integer Linear Programming Problem in [14]. They also proposed a polynomial-time heuristic and compared it against the optimal solution.

In [4], the authors applied DMS into cluster based WSNs with probabilistic workloads. For this scenarios, they formulated an optimal modulation level assignment to minimize overall energy consumption while meeting the deadlines. The same group later formulated optimal modulation level assignment for cluster based topologies while taking energy harvesting information into account. They have also proposed several polynomial-time algorithms and compared against the optimal solution.

The work presented in [16, 17] also studied optimal modulation level assignment for cluster based WSNs with probabilistic workload information. The authors proposed an efficient low-power-listening mechanism to dynamically reallocate available slack times which the nodes can use to lower their modulation levels.

The above papers, while important, exhibit a number limitations. Most crucially, they all assume a discrete output power level for a given modulation level. Although a higher power level is necessary to increase the modulation levels, the distance and current noise in the channel will also affect the exact output level which needs to be used for each pair of transceivers. Next, the modulation schema that have been used in the performance evaluations is primarily and mainly Quadrature Amplitude Modulation (QAM). However, only 16QAM was recently added to IEEE 802.15.4-2015 standard [22]. It is possible that the selection of QAM in the vast majority of the DMS papers followed from the use of theoretical energy scaling equations in QAM in pioneering research articles [10, 32] papers, coupled with the absence of DMS capability in off-the-shelf commercially available low-power radios. In Sect. 5, we show our real world test results and how they differ from what the generalized mathematical formulas suggest.

IEEE 802.15.4 standard specifies many modulations for its physical layer, including DSSS-OQPSK, MSK, FSK, ASK, GFSK, SUN-OFDM, and MPSK [22]. More recently, at the sub-GHZ level there are a number of commercially available radios starting to be offered with support to different modulation methods. These include GFSK, FSK, OOK, and MSK modulations [25].

Other work has been performed using SDR to analyze the impact of DMS. This includes using SDRs in *Inter-Vehicle Communication* domain [6]. This work has analyzed BPSK, QPSK, 16QAM, and 64QAM in 802.11p stack. The authors reported the signal-to-noise ratios and corresponding packet-delivery-ratios for each modulation schema used. These tests were set up according to IEEE 802.11p standard where the underlying modulation is OFDM and the aforementioned modulation schemas are used as modulation mapping.

3 A DMS primer

In this section we explain the fundamental underlying principals of DMS. In essence, DMS exploits the tradeoff between transmission time and energy consumption by controlling the number of bits encoded per symbol. Increasing (decreasing) the number of bits encoded per symbol decreases (increases) the transmission time but requires higher (lower) transmission energy consumption.

The energy consumption part is divided into two parts; energy consumption from transmission denoted as p_s, and electronic circuitry consumption denoted as p_e. These are related by Eq. (1) [32]:

$$\begin{aligned} p_s &= C_s \cdot \phi(b) \cdot R_s \\ p_e &= C_e \cdot R_s \end{aligned} \tag{1}$$

Above, R_s is the number of symbols transmitted per second, b is the modulation size, C_e and C_s are constants that depend on the hardware, as well as the current environmental conditions such as atmospheric conditions, transmitter–receiver distance and temperature. $\phi(b)$ is the scaling function given in Table 1, as suggested in [32]. As can be seen, the specific increase in the energy consumption depends on the modulation technique in use.

Transmission time is independent of the modulation technique; it only depends on the symbol rate and modulation size. The time to transmit a single bit is formulated as $t_{bit} = \frac{1}{b.R_s}$ where R_s is constant and equals to 62,500 symbols/s for 802.15.4 [22]. Hence, we can find that the energy consumption to send a single packet can be expressed as in Eq. (2), where L_{packet} is the length of a packet in bits.

$$\begin{aligned} e_{packet} &= L_{packet} \cdot (p_s + p_e) \cdot t_{bit} \\ &= \frac{L_{packet} \cdot (C_s \cdot \phi(b) + C_e)}{b} \end{aligned} \tag{2}$$

4 System model

In a typical system for environmental data monitoring, there are 3 types of nodes; sensing-nodes, sense-and-relay-nodes and a base-station. Sensing nodes are located on the edge of the network whose only duty is to sense and report. Sense-and-relay-nodes, on the other hand, not only sense the environment but also receive packets from sensing-nodes and forward it to other sense-and-relay-nodes or to the base-station. Here sensing duty of sense-and-relay nodes is optional. We also assume each node generates a constant amount of data after each sensing operation. The base-station is a node which is the final destination for all the information collected from the network. Sensing and sense-and-relay-nodes are assumed to be battery powered whereas base-station has unlimited power supply. Most importantly, we assume each node is equipped with a radio that supports DMS.

Our system model also assumes the application using this network is time-critical. More specifically, there is an upper bound on the delay. All the sensed data from the nodes needs to arrive to base-station within a specific delay bound depicted as D in Fig. 1. This effectively implies that each sensing-node's packet generation rate, λ, is $1/D$. Also, the nodes are time-synchronized. This is a common requirement for ICN systems such as WirelessHART. More importantly, optimization formulation presented in Sect. 6 assumes we have a network such that the information of a set of parent nodes to forward the packets given a node and a modulation level (potentially different for each {2, 4, 8} PSK modulations) is known. Also, we assume the interference range given a modulation level (once again different for each {2, 4, 8} PSK modulations) and the set of nodes within that interference range is known. ICNs typically have a centralized network manager with full topological knowledge. WirelessHART for example requires a centralized network manager who has the global information about each link quality and network topology. Each node submits *health reports* every 15 min. These health reports contain neighbor table with observed RSSI values as well as traffic rates, priorities and deadlines [26]. It is network manager's duty to compose a routing graph (offline) according to these collected reports. In case of a change in topology or significant changes in network link qualities, network manager is expected to recompute this routing graph.

Figure 2 shows an example network setup. Here $\{n_1, n_5\}$ are sensing-nodes. $\{n_2, n_3, n_4, n_6, n_7\}$ are sense-and-relay-nodes, n_8 is the base-station. The arrows show the directional links with a given modulation level. Please note that these interference ranges are drawn as unit-disk models for

Table 1 Scaling functions

Modulation scheme	2^b-QAM	2^b-PSK	2^b-PAM
$\phi(b)$	$2^b - 1$	$(\sin\frac{\Pi}{2^b})^{-2}$	$\frac{2^{2b}-1}{3}$

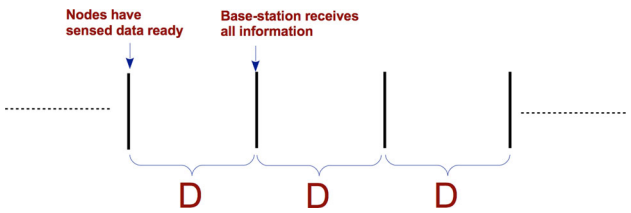

Fig. 1 Delay bound

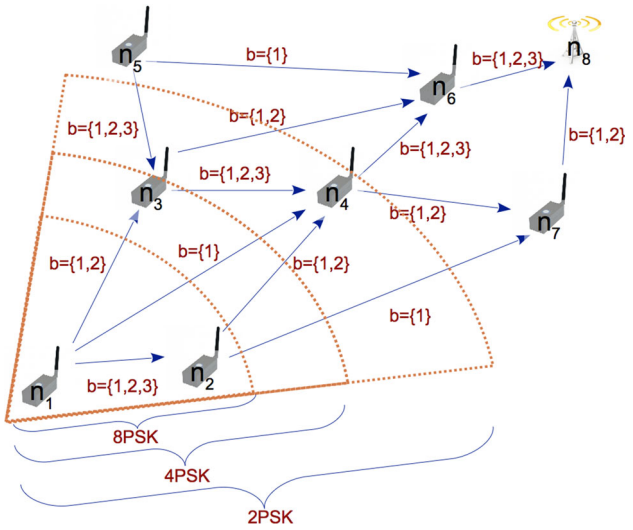

Fig. 2 An example network

ease of demonstration but this may not be the case for real-world deployments. According to this setup, n_1 can talk to n_3 using modulation levels 1 and 2. The Figure also shows the communication range of n_1 for each modulation level. For 8PSK n_1 has the interference range set of $\{n2\}$, for 4PSK $\{n_2, n_3\}$, for 2PSK $\{n_2, n_3, n_4\}$.

Note that it is possible to apply the same system model to multi-cluster networks such that each sensing-node and sense-and-relay nodes can be replaced with a cluster where a cluster head can report the aggregated sensing information from its own cluster and forward it to other cluster heads.

5 Software defined radio tests

The system model we have adapted (see Sect. 4) assumes the performance of different modulation levels in terms of the distance they can reach while preserving a healthy communication is known. However, due to the lack of commercially available hardware such information is not available. The mathematical model which is generally adapted by DMS enabled WSN researches (see Sect. 3) is not sufficient for us to move forward with the solution for the problem we are addressing in this paper. Therefore, we have conducted various experiments and obtained empirical analysis of the relationship between energy, distance, elevation difference, and packet delivery rate for DMS. We present our results in this Section. We kept the scope of our empirical analysis of DMS larger than what is needed for this paper to make sure the results presented here can be widely adapted for future researches in this domain.

We base our Software Defined Radio (SDR) experiments on the Ettus USRP B210 radio [31]. SDRs differ from traditional radios by implementing various components such as amplifiers, modulators/demodulator, filters on a software layer rather than hardware. This enables higher configurability and flexibility for experiments in the DMS realm.

The software we used is built using the GNURadio, an open source platform that comes with a number of reusable *blocks*. Figure 3 shows our transmitter and receiver implementation. PSK modulation and demodulation blocks are native GNURadio blocks. Our data source is a text file consisting of randomly generated ASCII characters. The subsequent blocks shown in Fig. 3 are needed to packetize the data from the source file. We have chosen 128 bytes as the packet payload length in order to comply with the IEEE 802.15.4 standard. Packetizing and tagging the input flow allow the receiver to distinguish between data from noise and other communications. We have also appended a 32-bit CRC to make sure we consider only successfully received packets after the demodulation phase.

In general, in order to successfully demodulate a signal with PSK, there has to be a signal recovery process preceding demodulation as demonstrated in Fig. 4. The PSK demodulation block already has signal recovery embedded in it with adjustable parameters. However, it only outputs the demodulated data, not the recovered signal. For the demonstration purposes, we have created a signal recovery process as described in [3].

Figure 5 shows the implementation of a signal recovery process with the existing GNURadio blocks and only one of the many possible techniques. Figures 6 and 7 show the constellation display of QPSK-encoded signal before and after the recovery process. Here *Polyphase Clock Sync* is used for timing recovery, inter-symbol-interference, and downsampling of sampling rate from 4 to 1 sample/s. The next *CMA Equalizer* block is used to react to multipath fading and as the final step the *Costas Loop* block is used to correct phase and frequency offset. Even though this signal recovery process could successfully lock the constellation points, it may have locked them with a 0, 90, 180 or 270 degrees difference, a problem going forward with demodulation. There are two ways to fix this. The first is to have a predetermined *access code* and try each of the possible phase difference to see which one decodes successfully. The second is to use *Differential PSK* (DPSK) instead. DPSK only transmits the phase shift difference as reference to the previously submitted signal. Hence, no matter what the phase difference between recovered and transmitted signal is, the relative phase difference between consecutive phase shifts still holds.

Before we present the results, we would like to elaborate more on our experience with the aforementioned PSK demodulation block's performance, as potential hints for future testbed implementations. First, in order to have

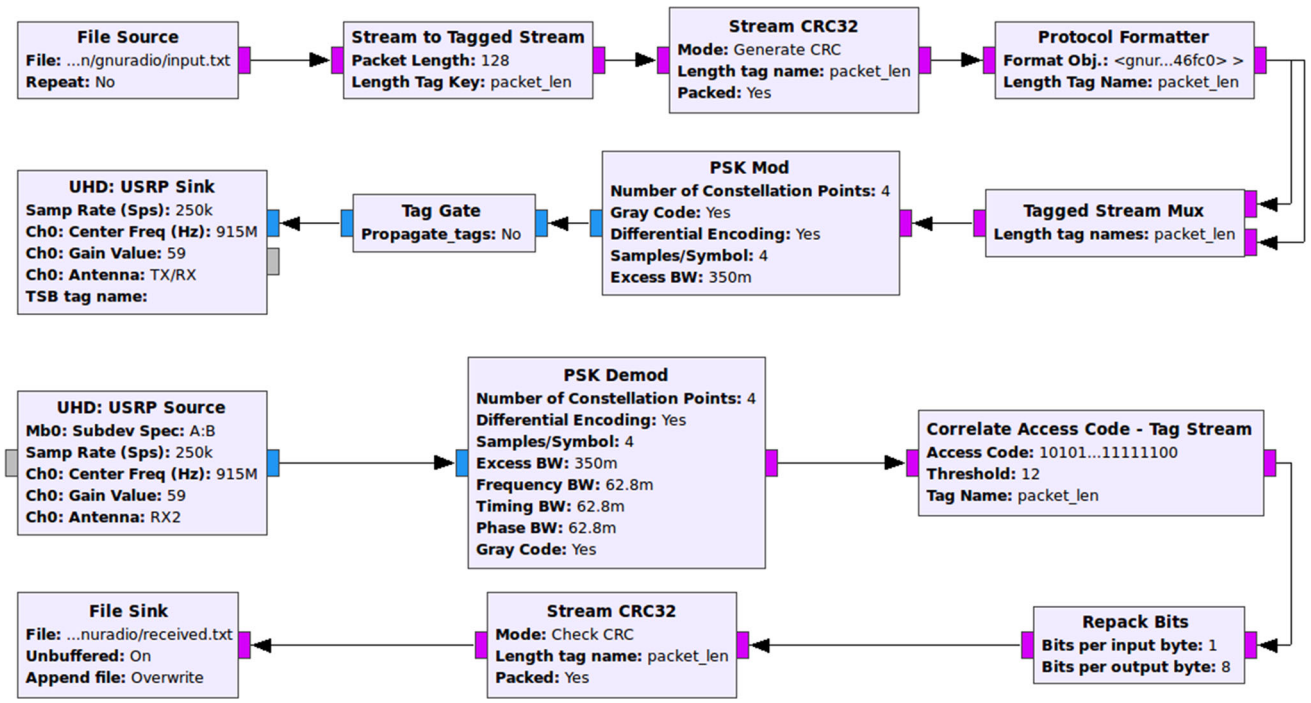

Fig. 3 Implementation of transmitter and receiver in GNURADIO

Fig. 4 Steps of signal decoding

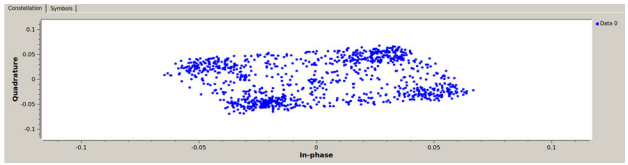

Fig. 6 Received signal

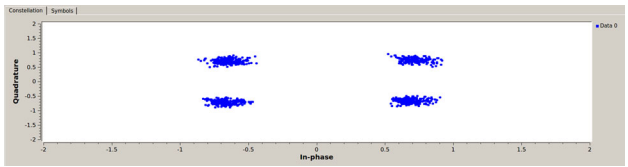

Fig. 7 Recovered signal

successful communication, we always had to start the sender first and then the receiver; doing the the other way around with significant delays prevented the receiver from recovering the signal. We have also observed that it is not always advisable to set the transmitter and receiver channel gain values equally. In some cases, especially for higher modulation levels, we have observed a lower receiver gain compared to the transmitter gain yielded higher performance especially for noisy enviroments. Last but not least, it is very helpful to set the PSK Demod block parameters as real-time configurable variables which can be tested on the fly and adjusted accordingly to each modulation level and environment conditions.

For our testing, we have configured our Ettus B210 Radios to use 915 MHz center frequency and 62,500 symbols/s symbol rate. We have also used 4 samples/s which gave a total of 250 kb/s data rate (for BPSK and QPSK 2 samples/s also worked well). The 915 MHz ISM band was chosen due to its higher output power levels

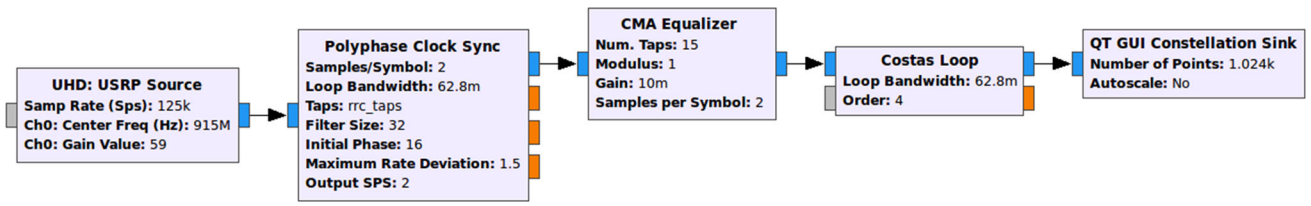

Fig. 5 An example PSK signal recovery process

compared to 2.4 GHz center frequency with the same output amplifier gain.

Our primary objective was to obtain empirical measurement of energy consumption for different modulation levels. To achieve this, we have measured the necessary transmission power output with different noise levels to establish reliable communication. In order to have a noise-controlled environment and eliminate the effect of moving objects, we used Faraday Cages shown in Figs. 8 and 9.

We have used a total of three Ettus B210 SDRs, one sender, one receiver, one noise generator. We have placed the sender and noise generator at equal distances from the receiver to make sure signal and noise powers are equally affected by the distance (Fig. 8).

5.1 Experimental results

In this section we explain the experiment setup and present the results. Using these results, we also provide some suggestions for future work in reliability and power management.

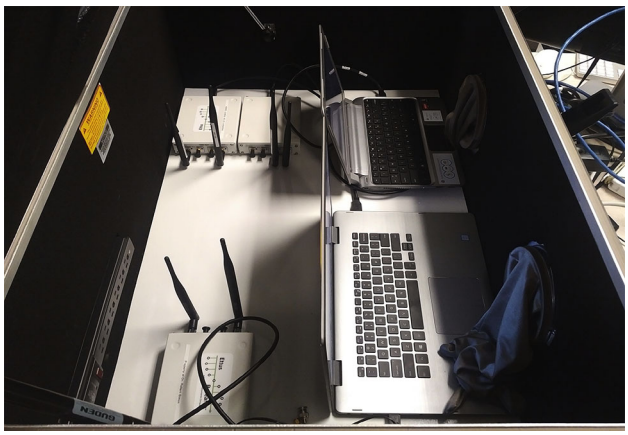

Fig. 8 Inside look

Fig. 9 Outside look

5.1.1 Signal-to-noise-ratio testing

Most of the previous work involving DMS with low-power wireless communication tends to assume constant energy output levels per each modulation level and does not take into account the fact that output energy consumption can be controlled independent of the modulation level adjustments, in particular by varying the output amplifier gain. To better understand this issue we have measured signal-to-noise-ratios (SNRs) of MPSK modulation and the corresponding packet-delivery-rate (PDR), for a given modulation level but by varying the amplifier gain. Here higher SNR values correspond to higher transmission power consumption. Our primary goal in this set of experiments is to analyze the energy increase required to have a reliable communication with different modulation levels and various output power levels using the MPSK modulation.

Each test involved sending 300 KB of payload. In order to make sure the results were minimally affected by our design, we created a *control case* by averaging how much data we can decode without any noise using the second modulation level. Hence, we have compared the amount of the data received to this control case value to compute the reported PDR values. The control case was never able to receive all of the 300 KB of data. The highest we could decode was 99.9%. This is because the signal recovery process takes a while and loses some of the initial data. This can be fixed by sending a data link layer preamble before the data packets or creating a more robust and optimized demodulation process. We believe the latter will be the case when DMS-compliant low-power radios are commercially available. Each PDR point is averaged over 100 experiments. Lastly, we have adjusted AWGN noise level to at 10% of the maximum analog gain of Ettus B210 radios (max gain is 89.8 dB) and adjusted transmission gains accordingly to create the desired SNR ratio in decibels. The formula used to compute SNR in dB is $10\log_{10}(P_{\text{signal}}/P_{\text{noise}})$. For the $P_{\text{signal}}/P_{\text{noise}}$ ratios greater than 10 times, we have reduced the noise level. The receiver gain values varied across the different settings. Hence, the reported increases in transmission power are only for the transmitter but a close approximation for the receiver.

Figure 10 shows the results obtained for {2, 4, 8, 16} PSK. In addition, we have also plotted the ideally scaled values calculated for {4, 8, 16} PSK using the scaling function reported in Table 1 by using empirical 2PSK measurements as the base case. We can see a marked difference between the ideally scaled values and empirical measurements. In order to have at least an average PDR of 99% 2PSK required an SNR of 4. For 4PSK, this value increased by a factor of 6 (to 12). However, the scaling

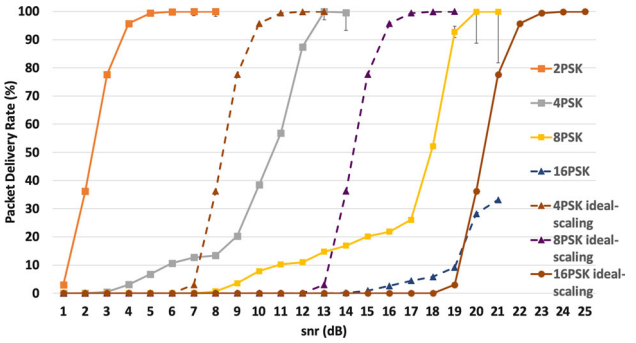

Fig. 10 SNR–PDR comparison of empirical measurements and ideally scaled calculations for MPSK

function suggests only an increase of 4 times. 8PSK required an SNR of 16, an increase of 2.5 times over 8PSK, which is much lower than what the scaling function suggests. Unfortunately for 16PSK case, the best we could received was an average of 33% PDR (we were unable to pass 35% even in the absence of noise). This may be due to not having enough output power level and as well as low performance of signal recovery process of built in PSK demodulation for higher constellation sizes also the noise generated by the transmitting SDR and the internal noise of the receiving SDR's hardware. However, if we compare this 33% PDR value to 8PSK, a 2.5 times increase is observed, once again a value lower than what the scaling function suggests.

Next, we repeated the same experiments with Differential PSK (DPSK) modulation. This time no ideal scaling is plotted due to unknown scaling function of DSPK to us. It has been shown that in communications over fading channels DPSK is preferable over PSK. DPSK uses the previously transmitted symbol interval's phase as a reference to decode the current symbol interval. In channels with slow condition variations compared to the symboling rate, DPSK gives good performance [12, 19].

Figure 11 validates the better performance of DPSK over PSK. The results of 2DPSK and 2PSK were very close

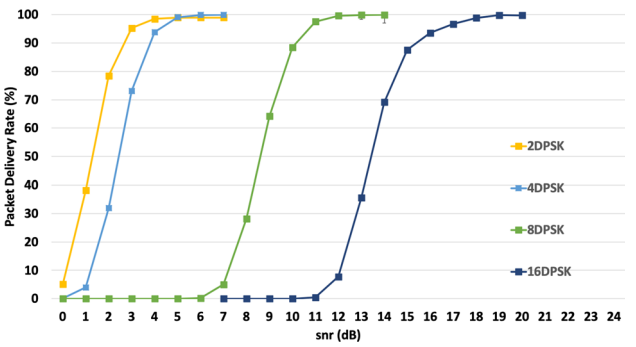

Fig. 11 SNR–PDR comparison of empirical measurements for MDPSK

and both required an SNR value of 4 to have at least a 99% delivery rate. The major difference is observed for {4, 8, 16} DSPK. 4DPSK achieved 99% PDR for SNR = 5 and required roughly 1.25 times more output gain compared to 2DPSK. This value was 6 for 4PSK. 8DPSK achieved the same PDR for of 12 which is 5 times the output gain compared to 4DPSK. However, it only required 6 times the output gain of 2DPSK. Lastly, 16DPSK also required 5 times more output gain compared to 8DPSK. {4, 8, 16} DSPK all performed better than the ideally scaled PSK results as expected; 31%, 39%, 50% of the ideally scaled results, respectively.

A final conclusion can be drawn from both Figs. 10 and 11; we can see that the scaling ratios between consecutive modulation levels vary and are not a constant value.

Translation from the results reported in Figs. 10 and 11 to the exact energy consumption of an 802.15.4 compliant radio will depend on the hardware design (the constants mentioned in Sect. 3), efficiency of the demodulation technique, as well as the channel access algorithm. As with many wireless standards, the 802.15.4 data link layer has two operating modes: The *beacon* mode and *non-beacon* mode. *Beacon* mode consists of two phases; *contention allowance period* (CAP) and *contention free period* (CFP). During CAP the nodes use slotted-CSMA to get an access to channel and in CFP the nodes have guaranteed time slots where the nodes take turn for channel access in time division multiple access fashion. *Non-beacon* mode is a simple CSMA access mechanism. For both cases the nodes assess the current noise level in the channel before transmitting. Since our reported results are based on SNR values, lower noise levels will require lower output power. However, the increase in power consumption between different modulation levels will remain valid for a given noise level.

It is now time to give a real-world use case where reported SNR based energy consumption is quite useful. Protocols for Wireless Industrial Control Networks (ICNs) such as WirelessHART or ISA 100.11a require real-time packet delivery with application specific packet delivery rate requirements. Wireless ICN protocols typically uses time-division-multiple-access based channel access mechanism with primary and backup time slots for reliability [28, 33]. In an application where a 98% PDR is the goal, SNR-based output power control provides a very fine tuned energy control mechanism. In this scenario rather than aiming for 98% or above delivery rate with a single transmission, aiming for a lower PDR rate and retransmitting in case of a packet loss may give a lower expected energy consumption. For example, if we are aiming for at least a 98% PDR using 16DPSK, we can either use an SNR value of 18 and transmit only once or use 15 (88% PDR) with a retransmission if fails. In the latter, we will have to

retransmit with a probability of 12% which will give a total of 98.56% PDR. Using an SNR of 18 with a single transmission will consume 1.78 times more energy as oppose to the expected energy consumption of using 15 with retransmission. Integrating DMS into this scenario gives the application the ability to not only fine tune its output power level but also take advantage of available transmission delay. An application can use a lower modulation level as long as the time slot length is large enough. Let us assume the primary time slot requires modulation level 16 whereas the backup slot can compensate for 8 (twice as long). Once again aiming for 88% PDR in both time slots for an expected PDR of 98% will consume 52% of a single transmission energy consumption.

We also suggest a slightly different perspective to the application of DMS in the low-power wireless domain. The primary focus of DMS-enabled low-power protocols has been energy management, and DMS is utilized as a tradeoff between time and energy while keeping PDR constant. In other words the goal is to manage output power while achieving the same PDR. However, it is also possible to view DMS as a *reliability* mechanism that provides a tradeoff between time and reliability. In other words, DMS can be used to keep the output power constant while reducing the modulation level in order to achieve the same or higher PDR. For example, let us assume we have a sensor mote which is currently using 8DPSK and achieving 97% PDR with a SNR value of 11. Now assume that there is a sudden change in channel condition which reduced our SNR value to 7. Instead of increasing the output power to restore SNR value back to 11, the mote can decrease its modulation level to 4 and still achieve 97% and above PDR while using the same output power level as long as the latency requirements are not violated. Even in non-changing channel conditions where SNR ratio is fairly constant, if the current quality-of-service requirements of the application-layer protocol changes and requires a higher PDR, it is still achievable with the same transmission power by decreasing the modulation level rather than increasing output gain. We believe this type of DMS application is just as important and a viable direction for the future research of DMS in low-power wireless networks.

5.1.2 Distance testing

In the next set of experiments, we tested the PDR of M-DPSK in terms of distance. Distance testing is important due to its impact on topology control. In wireless domains, topology control is mainly used to find an optimal subset of the nodes in the network to ensure connectivity. Notice that connectivity may have slightly different meaning in the low-power domain, so here we are referring to the ability that each sensing node can successfully communicate with the coordinator with or without using relay nodes. One important advantage of this approach is that nodes can better manage their sleep schedules [2]. DMS gives the ability to control the transmission distance without increasing the output power consumption but with a penalty of increased transmission delay.

Figure 12 shows the picture of the test area. We chose this location because it was very isolated and we measured very weak WiFi activity. Also, it allowed us to test up to 100 m. The location of the receiver was chosen in such a way that the multi-path fading was minimal and it was located in the exact same spot for each test. The location of the sender was adjusted accordingly for each desired distance on a straight line. The readers should be aware that the results can change significantly for different environments especially indoor or urban areas.

Our first experiment was conducted in the 2.484-GHz center frequency with the same parameters used for the SNR tests. Low-power radios such as CC2420 has 0 dBm maximum transmission power. Hence, our goal was to set our SDRs to use 0 dBm output power as well. However, SDR only allows us to configure its output amplifier gain (in dB) which produces different absolute transmission power levels depending on the center frequency being used. In order to get very accurate mapping of output amplifier gain to the absolute transmission output power, spectrum analyzer and highly sensitive power monitors are needed. We identified a forum post on the official National Instruments website by a SDR Product manager of the company, which provided their internal test results of mapping amplifier gain to specific output power for different center frequencies using the Ettus B200 series SDRs [20]. According to their internal test results an amplifier gain between 61 and 75 dB corresponds to 0 dBm output

Fig. 12 The test area that was used for distance measurements. Sender and receiver are 100 m apart

power. In order to pinpoint a specific gain value within this range, we have measured RSSI values with Zolertia Z1 motes set as sender and receiver equipped with CC2420 radios—manufactured by Texas Instruments [24]—transmitting at 0 dBm. Next, we have repeated the same test with the same Z1 receiver instead of Ettus B210 transmitter. An output gain of 70 dB gave the closest RSSI values to match 0 dBm absolute transmission output power.

Figure 13 shows the results we obtained for {2, 4, 8} DPSK. We were unable to decode any signal using 16DSPK with 70 dB output amplifier gain. The error bars seen on the graph are 95% confidence levels. From these tests, we can conclude that increasing modulation levels decreases PDR at a specific distance (Figs. 14, 15).

We then repeated the same test with the 915-MHz center frequency. National Instruments internal tests indicate 70 dB amplifier gain corresponds to a little less than 10 dBm output power. Similarly, Texas Instruments sub-1-Gig low power radios have around 10 dBm maximum output power as oppose to 0 dBm of CC2420 2.4 GHz radio. As expected PDR has increased for each modulation level for each distance value significantly. So much that both 2 and 4 DPSK has maintained above 90% delivery rate for up to 100 m. This increase is due to aforementioned increase in the output power level. For a good measure of comparison we ran tests with 915 MHz but this time with 60 dB amplifier gain. This gain value corresponded approximately to 0 dBm according to National Instruments internal tests [20]. However, this time we did not have a sub-1gig radio to compare RSSI values. We have observed similar but slightly better results compared to 2.4 GHz. This was likely due to a combination of slightly higher output power level created by 60 dB gain

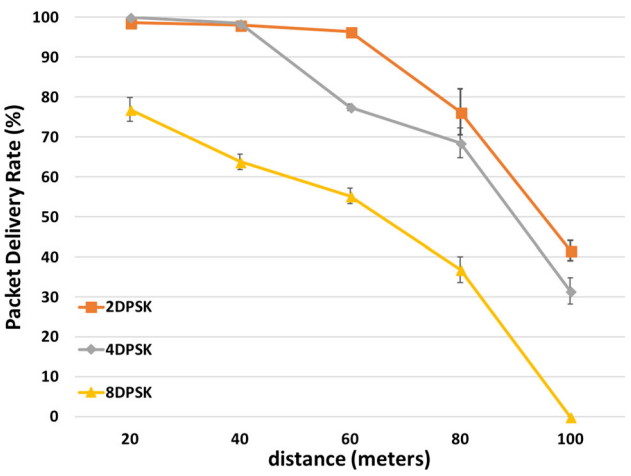

Fig. 13 Packet delivery rates of different modulation levels using 2.4 GHz center frequency with 70 dB output amplifier gain in terms of distance

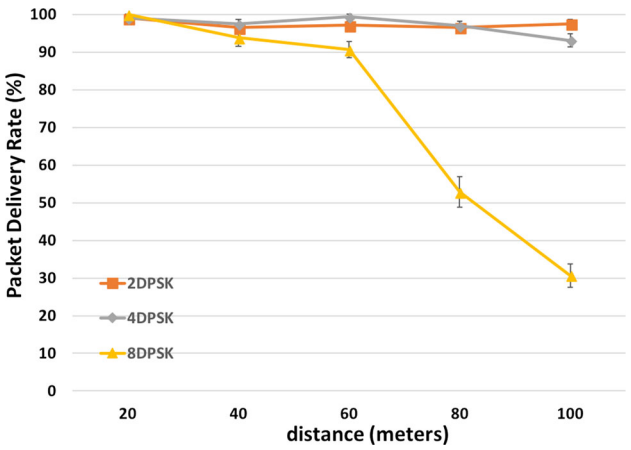

Fig. 14 Packet delivery rates of different modulation levels using 915 MHz center frequency with 70 dB output amplifier gain in terms of distance

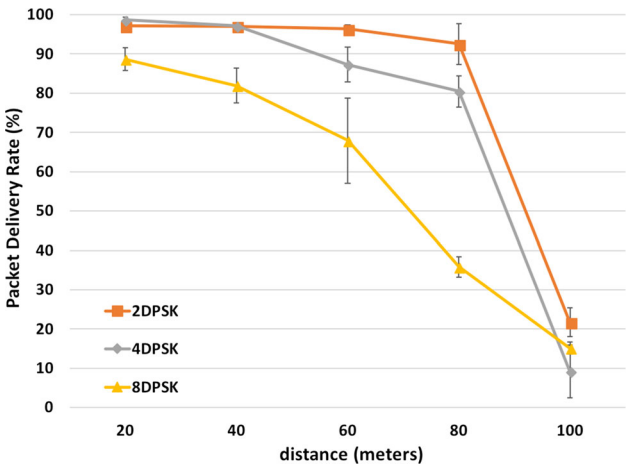

Fig. 15 Packet delivery rates of different modulation levels using 915 MHz center frequency with 60 dB output amplifier gain in terms of distance

915 MHz center frequency as oppose to 70 dB 2.4 GHz and lower free space propagation loss of 915 MHz band as oppose to 2.4 GHz band [29].

Lastly, we have expanded our distance testing to include wide variety of output amplifier gain values. Our optimization problem (see Sect. 6) assumes the energy consumption per modulation level, distance and output power level is known. In order to set realistic parameters for our simulation setup, we have run these additional SDR tests. These tests aim to find the PDR given a modulation level, sender–receiver distance and output power level. To accomplish this, we have used various output power levels, specifically $-10, -7, -5, -3, -1, 0$ dBm levels which CC2420 supports [24]. More precisely, we have set the output and receiver amplifier gains to 60, 63, 65, 67, 69, 70 dB respectively to achieve the desired power output levels using 2.485 GHz center frequency. Figures 16, 17,

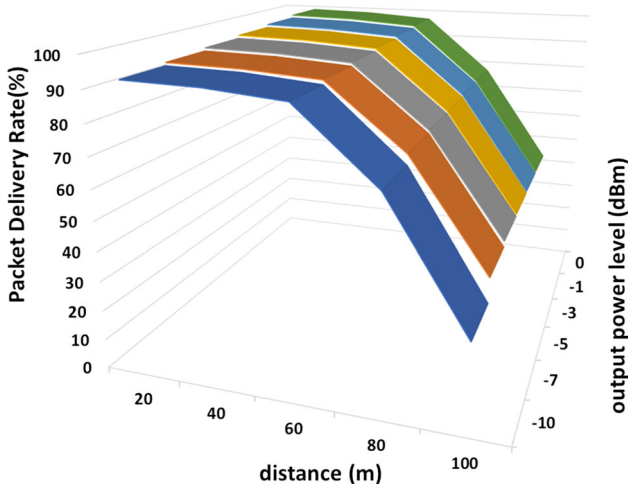

Fig. 16 2DPSK packet delivery rate given a transmitter–receiver distance and an output power level

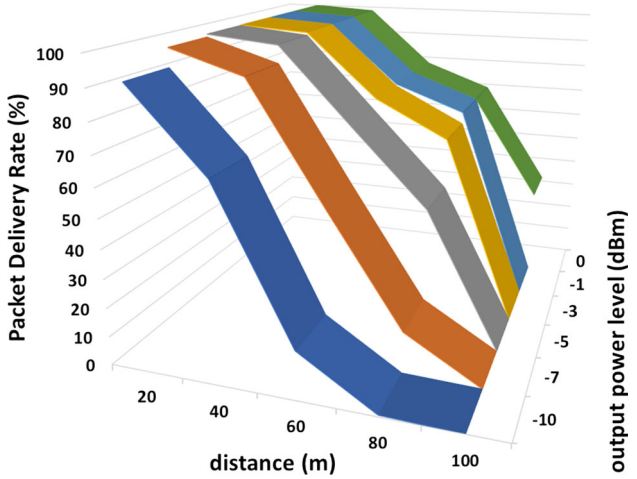

Fig. 17 4DPSK packet delivery rate given a transmitter–receiver distance and an output power level

and 18 shows the obtained results for 2DPSK, 4DPSK, 8DPSK respectively. In these Figures, you can see distance matrix on the x-axis, output power levels on z-axis and corresponding PDR values on y-axis.

These results have important implications for topology control. Connection driven topology control protocols such as Span [8], ASCENT [7] aim to find the minimum number of relay nodes necessary to forward packets to the coordinator. The goal is to reduce the number of active nodes and hence the number of hops. Lowering modulation level allows approaches that work by enabling nodes to broaden their communication range while reducing their energy consumption. As a result, these similar topology control protocols can further reduce the number of active relay nodes. Assume an example where a sensing node is 120 m away from the coordinator and needs 2 relay nodes positioned 40 m apart to forward its packets using modulation

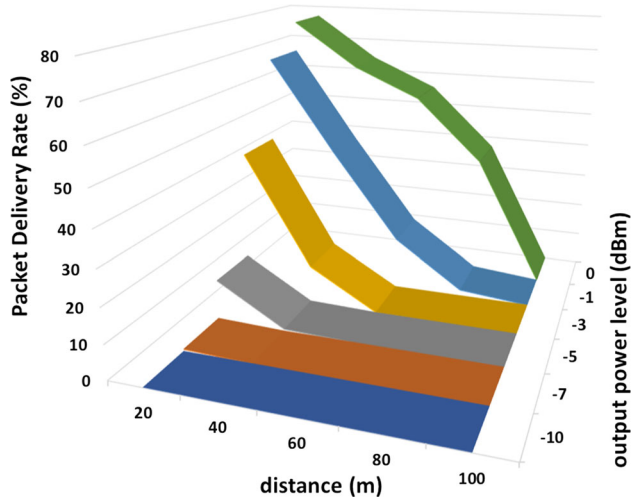

Fig. 18 8DPSK packet delivery rate given a transmitter–receiver distance and an output power level

level 4. According to our tests, reducing modulation level to 2 will increase the reliable communication range up to 60 m and only a single relay node will be sufficient to forward the packets. Here, the total energy savings will be the sum of *energy saving from the sensing node, relay node and coordinator by reducing their modulation levels* and *energy savings by putting a relay node into deep sleep*, a figure which is quite significant.

5.1.3 Elevation testing

In our last set of experiments, we have measured PDR of {2, 4, 8} DPSK in terms of elevation difference between sender and receiver. We have placed the sender outside of a building at 1.08 m, 8.9 m, and 13.6 m high in reference to the receiver. The receiver is placed on the ground level (slightly elevated). Figure 19 shows the tests area and Fig. 20 shows the height of the sender and the distance between sender and the receiver. In applications such as residential automatic meter reading, it is possible for sender and receiver to have elevation difference such as a transmitter can be placed on a current-meter higher of the ground or an application where an elevated pipeline equipped with sensors broadcasting the current pressure inside the pipe.

Figure 21 shows the PDR of {2, 4, 8} DPSK with 915 MHz center frequency and 60 dB output amplifier gain as we have used in Sect. 5.1.2. Some conclusions we draw can be listed as follows: (1) Higher elevations cause higher packet loss for both 915 MHz and 2.484 GHz center frequency although 2.484 GHz center frequency with 70 dB output amplifier gain had higher PDR for both 4 and 8 DPSK as shown in Fig. 22. We should mention here that we were using different antennas for 915 MHz and

Fig. 19 Test area that was used for elevation tests. Receiver is placed on the ground and the sender is elevated

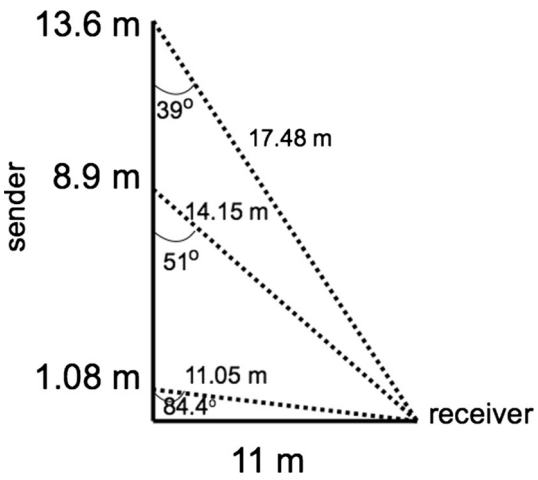

Fig. 20 The height of the sender and the distance of the sender from the receiver

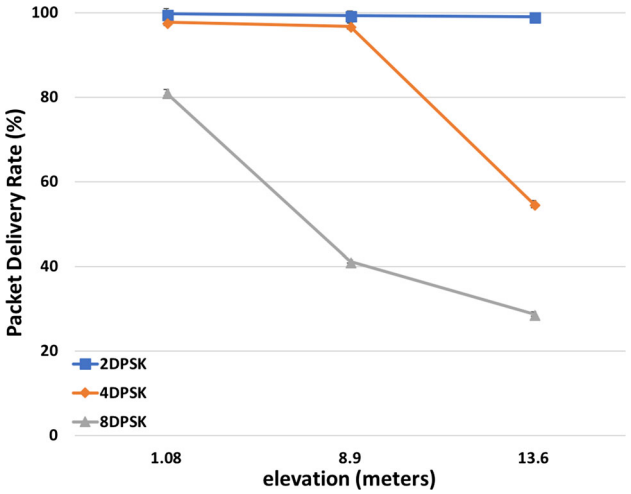

Fig. 21 Packet delivery rates of difference modulation levels using 915 MHz center frequency with 60 dB output amplifier gain in terms of elevation difference

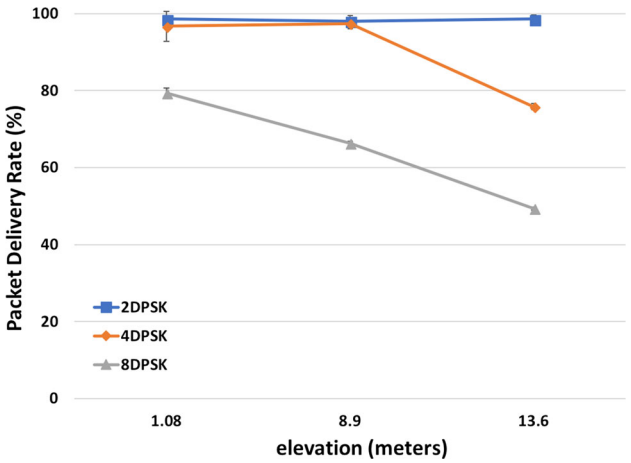

Fig. 22 Packet delivery rates of difference modulation levels using 2.484 GHz center frequency with 70 dB output amplifier gain in terms of elevation difference

2.484 GHz tests that are both omni-directional vertical antennas (Ettus Research VERT900 and VERT2450 respectively). Different antennas may have different polarization characteristics typically either vertical, horizontal or circular in x–y axis. Elevation difference requires the signals to be transmitted in the z axis where VERT900 and VERT2420 antennas likely to have different characteristics in this direction. (2) 2DPSK seems to be not effected by the elevation difference with the tested output gain values making it the best candidate for the aforementioned types of applications. (3) If we compare the results from Figs. 21 and 22 with Figs. 13 and 15, it shows us that elevation difference has a higher impact on PDR compared to distance difference with the same elevation. Figures 13 and 15 indicates that all the modulation levels have 80% or above PDR for distances up to 20 m (with no elevation difference) whereas when the transmitter is elevated 13.6 m and placed 17.48 m apart from the receiver, the PDR reduces dramatically for 4 and 8 DPSK as low as 54% and 28% respectively .

6 Optimization problem formulation

Previous work in literature has shown that connection driven topology control has tremendous potential to decrease energy consumption and/or latency [2, 7, 8, 11, 27]. DMS changes transmission energy levels

and has a direct impact on packet loss rate and propagation distance. However, current work does not provide any multi-cluster communication solutions which incorporate DMS into already managed transmission energy level control. In this section, we address this gap by formulating Mixed Integer Nonlinear Programming formulation of DMS enabled transmission scheduling for deadline driven networks. Later in Sect. 7, we explain how to convert it to a linear programming problem.

Here, we present the optimization problem formulation of *a given a network topology with known* potential *parent nodes per modulation level, choose a parent and a modulation level for each node while minimizing the total network energy consumption given a maximum delay bound of when the base station has to collect all the information from the network.*

Figure 23 gives an example case from a single node's perspective, node$_4$ in this case. The arrows show the connection between nodes with possible modulation levels and corresponding delay and energy consumption values. The goal is to find a path from node$_4$ to the base-station, denoted with b, bounded by 3t. There are three possible routes in this case; $r_1 : n_4 \rightarrow n_2 \rightarrow n_1 \rightarrow b$, $r_2 : n_4 \rightarrow n_3 \rightarrow b$, and $r_3 : n_4 \rightarrow n_1 \rightarrow b$. The lowest delay r_1 can provide is 3t with 12e of corresponding total energy consumption. For r_2 and r_3, these values are 4t with 2e and 3t with 5e respectively. Only r_1 and r_3 satisfies the given delay bound where r_3 has the lower corresponding energy consumption. Hence, in this example n_4 should use r_3.

Our optimization formulation finds these paths for each node and their transmission start times such that the total energy consumption of the whole network is minimized while the time it takes to collect data from each node is bound by the given deadline. Table 2 is the list of the

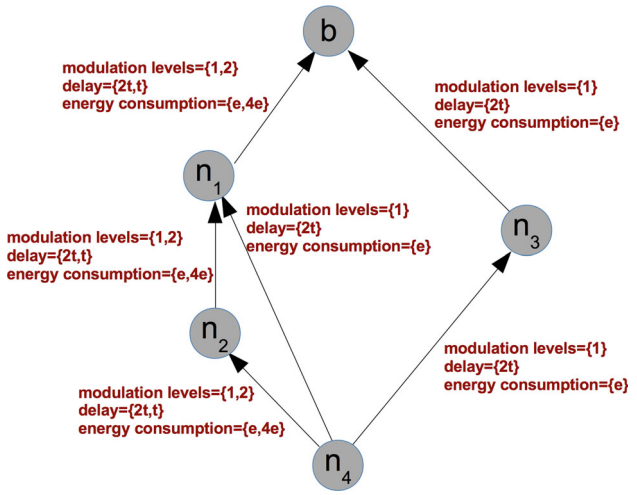

Fig. 23 Optimization problem demonstration from a single node's perspective

variables and their definitions used in our problem formulation. The integer variables in our formulation are the $\beta_l^{i,k}$ and $I^{i,k}$ binary integer variables. The remaining variables are real numbers (except for IR_l^i which is a set). When $\beta_l^{i,k}$ equal to 1, that means node i will transmit to node k using modulation level l. The same notation is also used for the remaining variables. The value of $e_l^{i,k}$ might differ for each link even for the ones which are using the same modulation level depending on the current noise level of the channel as well as the distance between transmitter and the receiver. Energy consumption between node i and all non-reachable nodes are set to ∞. Similarly, delay variable $d_l^{i,k}$ is also set to ∞ if node i cannot reach node k via modulation level k.

$$\min. \sum_{i=1}^{n} \sum_{k=1}^{n} \sum_{l=b_{min}}^{b_{max}} \beta_l^{i,k} e_l^{i,k} \tag{3a}$$

$$\text{s.t.} \sum_{k=1}^{n} \sum_{l=b_{min}}^{b_{max}} \beta_l^{i,k} = 1 \tag{3b}$$

$$\beta_l^{i,k}(S_i + d_l^{i,k}) = \beta_l^{i,k} E_i \tag{3c}$$

$$\beta_l^{i,k} E_i \leq S_k \tag{3d}$$

$$\beta_l^{i,k} E_i \leq \beta_l^{i,k}(S_j + I^{i,j} Q) \tag{3e}$$

$$\beta_l^{i,k} E_j \leq \beta_l^{i,k}(S_i + (1 - I^{i,j})Q) \tag{3f}$$

$$\beta_{i_1}^{i,k} \beta_{l_2}^{m,t} E_i \leq \beta_{i_1}^{i,k} \beta_{l_2}^{m,t}(S_m + I^{i,m} Q) \tag{3g}$$

$$\beta_{i_1}^{i,k} \beta_{l_2}^{m,t} E_m \leq \beta_{i_1}^{i,k} \beta_{l_2}^{m,t}(S_i + (1 - I^{i,m})Q) \tag{3h}$$

$$S_i \geq 0 \tag{3i}$$

$$S_{basestation} \leq D \tag{3j}$$

$$\beta_l^{i,k} \in \{0, 1\} \tag{3k}$$

$$I^{i,k} \in \{0, 1\} \tag{3l}$$

Equation 3a is the objective function which is to minimize the total sum of each node's energy consumption. When $\beta_l^{i,k}$ is 1, we sum the corresponding energy consumption of node i transmitting to node k using modulation level l.

Constraint 3b forces each node to select a single modulation level and a parent node to transmit its data.

Constraint 3c sets the transmission end time of each node, E_i, to its transmission start time plus transmission delay of the chosen parent and chosen modulation level. E_i is an auxiliary variable to ease the readability of the formulation. When $\beta_l^{i,k}$ is 1, the transmission start time of node i (S_i) plus the transmission delay of node

Table 2 The variables used in the problem formulation and their description

$e_l^{i,k}$	Transmission plus computation energy consumption when node i sends a packet to node k via modulation level l
$d_l^{i,k}$	Transmission and computation delay when node i sends a packet to node k via modulation level l
$\beta_l^{i,k}$	Binary integer variable to ensure node i only picks a single parent node k to send its packets
S_i	The time when node i starts transmitting
E_i	The time when node i stops transmitting
$I^{i,k}$	Binary integer variable to ensure only node i or node k is scheduled to transmit at any given time, never both
n	Total number of nodes in the tree
IR_l^i	Modulation l interference range of node i
D	Deadline constraint
Q	A very large number

i transmitting to node k using modulation level l ($d_l^{i,k}$) equals to the transmission end time of node i.

Constraint 3d makes sure if node i chooses node j as its parent then node j can start transmitting only after node i stops transmitting. If there are are multiple nodes who transmits to node k, S_k has to be greater than the largest E_i value which implies that a node can start transmitting only after *all* of its children stops transmitting. When $\beta_l^{i,k}$ is 0, S_k has to be greater than or equal to 0 which is trivial.

Constraints 3e and 3f ensure the nodes which are interfering with each other do not transmit simultaneously. More specifically, either the start time of node i is after the end time of node j or the start time of node j is after the end time of node i. When $\beta_l^{i,k}$ is 1, for all the nodes which are in node i's interference range of modulation level l the following holds; $E_i \leq S_j + I^{i,j}Q$ and $E_j \leq S_i + (1 - I^{i,j})Q$. If $I^{i,j}$ equals to 1 then it becomes $E_i \leq S_j + Q$ and $E_j \leq S_i$. Since Q is a very large number $E_i \leq \infty$ is trivially satisfied and node i is allowed to transmit only after node j stops transmitting. If $I^{i,j}$ equals to 0 then the constraints become $E_i \leq S_j$ and $E_j \leq S_i + Q$. The second constraint is trivially satisfied again and node j is allowed to start transmitting only after node i stops transmitting.

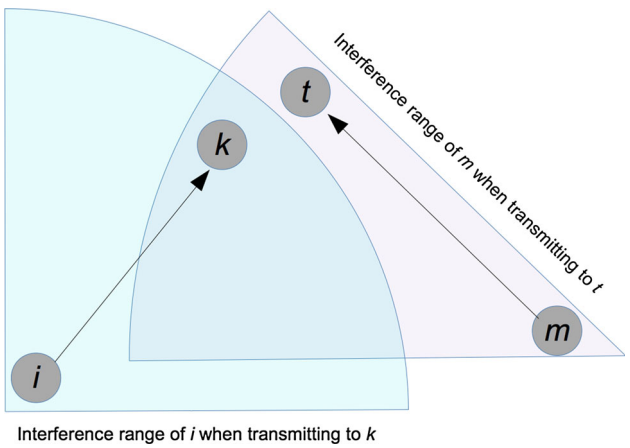

Fig. 24 Hidden node scheduling problem

Constraints 3g and 3h are to prevent collision due to hidden node problem. Even though we have prevented interfering nodes from co-scheduling (Constraints 3e and 3f), it is still possible to have colliding nodes. In Fig. 24 nodes i and m are not within each other's interference range. However, if they were to transmit to nodes k and t respectively, node k may not have been able to recover i's signal since k is within interference range of m. Hence, node k shouldn't be scheduled to receive any packets while node m is transmitting. When node i transmitting to node k and any node which node k is in its interference range is also transmitting, the product $\beta_{l_1}^{i,k}\beta_{l_2}^{m,t}$ becomes 1. Here we used two different variables for the modulation levels l_1 and l_2 to emphasize that the nodes i and m might be using different modulation levels and node k has to be in the interference range of node m's currently chosen modulation level l_2. When this holds, the constraints becomes $E_i \leq S_m + I^{i,m}Q$ and $E_m \leq S_i + (1 - I^{i,m}Q)$. Using the same logic we have demonstrated for the Constraints 3e and 3f, these conditions enforces either the transmission start time of node m to be after node i's transmission end time or vice versa.

The remaining constraints are straight forward. Constraint 3i is to make sure nodes do not start transmitting prematurely whereas Constraint 3j is to enforce the deadline which is to make sure base-station receives all the packets in at most D seconds. Lastly, Constraints 3k and 3l defines the variable β and I as binary integers.

6.1 Polynomial time heuristics

Generally, integer programming problems are known to be intractable. We therefore designed polynomial time heuristics. To evaluate the performance of the optimization problem described in Sect. 6, we have designed two polynomial time heuristics (explained in Sect. 6.1) and

compared them against the optimal solution. There are some components which are shared by both, and are now explained next.

6.1.1 Finding the minimum delay path

First shared step is finding the minimum delay path. This step involves running Dijkstra's shortest path algorithm from each node until the shortest path to the base-station is found. We assigned the delay of each edge as its weight. Let's call the total number of nodes in the network including the base-station n, and the total number of modulation levels b. Complexity of Dijkstra's algorithm implemented with a binary heap is $O(|E| + |V|log|V|)$ [5]. In our set up, the number of edges in the worst case is $b \cdot n^2$. Hence each Dijkstra's algorithm run is $O(b \cdot n^2 + n \log n)$. We run Dijkstra's algorithm from each node so the total is $O(b \cdot n^3 + n^2 \log n)$. Here b is constant so the total complexity is $O(n^3)$.

After the end of this process, each node now knows their shortest path parent and which modulation level to use when sending to the base-station.

6.1.2 Interference aware ordering

The next shared step is interference aware ordering. This step aims to find a schedule among all the nodes such that no interfering nodes will be co-scheduled as well as no parent will be scheduled before its children. In order to accomplish this, we run a *topological sort* on the directed-acyclic-graph (dag) generated after the first step described above. Topological sort of a dag is the ordering of its vertices such that if there is an edge (u, v), u appears before v in the ordering. After we know the topological sort of the dag, we start assigning orders to the nodes as the following. We initialize each node's order to -1. Following the topological sort, each node is assigned the first available order in the range [*maximum order of its children* + 1, *maximum order among all the nodes in its interference range of every modulation level and their minimum delay parents* + 1]. Moreover, when a node is assigned an order, its parent node is also temporarily assigned the same order to prevent interference due to hidden node. Topological sort makes sure no parent is scheduled before its children and our ordering mechanism makes sure no interfering nodes are co-scheduled. Time complexity of topological sort is $O(|V| + |E|)$, the worst case number of edges in our setup is $b \cdot n^2$ where b is number of modulation levels and n is the number of nodes in the network including the base-station. The time complexity of assigning orders is $O(n^2)$. All together, the total time complexity of this step is $O((b + 1) \cdot n^2 + n)$ which is $O(n^2)$.

6.1.3 Computing minimum delay

The third and the last shared step is computing minimum delay. Since now we know each node's minimum delay parent, minimum delay modulation level and its order, in this step we compute each node's minimum delay such that the minimum time it takes for that node to send its packet to the base-station. This is calculated by first computing *time-slot lengths*. There are same number of time-slots as the maximum order value (the order of the base-station). Each time slot length is equal to the maximum delay of the nodes which has the same order as that time slot number. Minimum delay of a node is then computed by summing up the time slot lengths from its order till (and including) the order of the node in its shortest path which is directly connected to the base-station. The calculated minimum delay of a node means this is the shortest time it would take for this node to send its packet to the base-station according to the path and ordering chosen by the previous steps. The time complexity of computing time slot lengths is $O(n)$. Computing minimum delay of each node can be done recursively such that *mindelay*(i) = *sum of time slot lengths from i to its minimum delay parent p* + *mindelay*(p). And if we *memoize* the previously computed minimum delay values, each node is only visited once and hence the time complexity is $O(n)$. The total time complexity of this step is $O(2n)$ which is $O(n)$.

The total time complexity of all shared steps is $O(b \cdot n^3 + n^2 \log n + (b + 1) \cdot n^2 + 3n)$ where b is a constant and equals to the number of modulation levels and n is the number of nodes in the network including the base-station which is $O(n^3)$.

Next, we describe our heuristics.

6.1.4 Aggresive

Our first heuristic starts with computing minimum delay of each node by following the steps described in the beginning of this section. When given a maximum delay bound, a.k.a. deadline, *Aggressive* first checks to make sure the minimum delay of each node is at least as long as the deadline. If all the nodes can meet the deadline, we calculate the *slack* which is the difference between the delay of node with the longest minimum delay and the deadline. As long as the slack is large enough to compensate for the additional delay by a reduction of a modulation level, the time-slot which has the most number of nodes with minimum delay modulation levels greater than the lowest modulation level is chosen. The modulation levels of these nodes are reduced by one level. The slack is then recomputed and the process is repeated till we no longer have a large enough slack. Algorithm 1 is the pseudocode of *Aggressive*.

```
for each node in the network do
    Check if minimum delay ≤ deadline;
    return no answer if the check is false;
end
slack = deadline - max of the min delays of each
  node;
while slack ≥ increase in delay by reducing
  modulation levels by one do
    slot := most occupied time-slot by nodes with
      modulation levels greater than b_min;
    updated := false;
    for node := nodes in slot do
        if modulation level of node > b_min then
            reduce modulation level of node by one;
            updated = true;
        end
    end
    if updated is false then
        break the while loop;
    end
    recompute min delays of the nodes;
    slack = deadline - max of the min delays of each
      node;
end
```

Algorithm 1: The algorithm of Aggressive Heuristic

Comparing minimum delay of each node against the deadline can be done in $O(n)$ time complexity. Finding the time-slot with the with the highest number of nodes with modulation levels greater than b_{min} can be done in $O(2n)$; first by going over the each node and updating the corresponding time-slot information and then by iterating over the time-slots to return the maximum. In the worst case each node will occupy a different time-slot and hence it will take $O(n)$ time. Lowering each node's modulation level can also be done in $O(n)$ in the worst case which in when all the nodes share a single time-slot and all are eligible for a modulation level reduction. The time complexity of computing minimum delays of the nodes are explained in the third bullet of the shared steps explained previously which is $O(2n)$. The *while loop* can at most loop for $(b-1)n$ times since each node can at most reduce its modulation level $b-1$ times. Hence the total complexity of *Aggressive* without the initial shared steps is $O(5(b-1)n^2+n)$ if we include these steps the time complexity becomes $O(bn^3 + n^2 \log n + (6b-4)n^2 + 4n)$ which is $O(n^3)$.

6.1.5 Explorer

Our second heuristic is an improved version of Aggressive where it also starts with computing the minimum delay of each node by following the shared steps with one addition: while we are running Dijkstra's algorithm to compute the minimum delay paths, we also store the **cumulative energy consumption** of each node. Suppose a node's, $node_a$'s, cumulative energy consumption equals to the sum of each node's individual energy consumption which are on that $node_a$'s shortest path including $node_a$.

Using the collective information of both *cumulative energy consumption* and *minimum delay*, the nodes now *explores* more options to forward their packets other than their minimum delay parents. Instead of *aggressively* forwarding packets to their minimum delay parents, now the nodes look among all the nodes within its reach with higher order number and picks the one with the (minimum cumulative energy consumption + energy to send to it) among the nodes which have short enough (minimum delay + delay to send to it) value to meet the deadline. Algorithm 2 shows the pseudocode of the Explorer Heuristic.

```
Input: deadline
initialize each node's individual deadline to deadline;
initialize each order's minimum requirement to
  deadline;
sorted := sort the nodes according to their order;
for i in sorted do
    update i's individual deadline requirement to the
      updated minimum delay requirement from
      previous orders;
    min := ∞;
    for j within i's reach per modulation level do
        if j's order > i's order AND j's minimum
          delay ≤ i's individual deadline AND j's
          cumulative energy consumption + energy
          consumption from i to j < min then
            choose j;
            choose modulation level;
            update min;
        end
    end
    update j's individual deadline to MINIMUM of
      j's individual deadline OR i's individual
      deadline - delay from i transmitting to j;
    update the minimum requirement of the order
      (i's order + 1) to MINIMUM of minimum
      requirement of (i's order + 1) OR i's individual
      deadline - delay from i transmitting to j;
end
recompute interference aware ordering;
recompute minimum delay;
if all the nodes meet deadline then
    run Aggressive Heuristic with the new edges;
end
else
    run Aggressive Heuristic with Dijkstra's shortest
      path edges;
end
```

Algorithm 2: The algorithm of Explorer Heuristic

Explorer keeps track of each node's individual deadline requirement separately. This is because each node's

individual deadline will change according to the nodes' which pick it as their parent. Algorithm 2 starts with initializing each node's individual deadline requirement to the given application's deadline requirement. Then, it proceeds node by node from the one with the smallest order to the largest. Let's assume Explorer currently picked node$_i$ and will try to find a parent for it. The other nodes need to satisfy two conditions to be qualified as a candidate; (1) its order has to be greater than the node$_i$'s, (2) its minimum delay has to be less than or equal to node$_i$'s individual delay. By iterating over the nodes according to their orders and picking the nodes with greater order ensures that the minimum delay computed initially still holds because the nodes with greater order still has the potential to pick their minimum delay parents if necessary. Node$_i$ picks the node with the lowest cumulative energy consumption among the set of candidates, let's call this node$_j$. Next the individual delay of node$_j$ has to be updated. Node$_j$ now has to be make sure it can delivery the packet in the worst case of (individual delay of node$_i$—transmission delay from node$_i$ sending to node$_j$ by using the chosen modulation level). Only updating the individual delay of node$_j$ is not enough. Not only node$_j$ but also all the nodes with greater order than node$_i$'s has to have at least the individual delay of node$_i$. This is because if any of the nodes with greater order than node$_i$'s, delivers its packet with a longer delay than of node$_i$'s individual delay then there is a chance for node$_i$ to miss the deadline.

After Explorer picks a parent for each node, it recomputes the interference aware ordering with the new set of edges. Please recall that interference aware ordering was topological sort based, now with the new set of edges, it is possible for a node to be a leaf node whereas it was not according to initially computed shortest paths based on Dijkstra's algorithm, this has the potential to lower its minimum delay because now it does not have have to wait for its children and grandchildren to finish before start transmitting. On the down side, the nodes which has the same order with different minimum delay parents might now be interfering if Explorer was to choose them the same parent. Because of the potential changes in ordering, Explorer recomputes the orders and checks if the deadline is still met. If so it runs the Aggressive Heuristic with the new edges chosen by Explorer. Otherwise, the Aggressive Heuristic is run with the edges chosen by Dijkstra's algorithm as if Explorer did not even execute to begin with.

Sorting the nodes according to their orders take $O(n \log n)$ time. There can be at most $b(n-1)$ nodes with the reach to be a parent candidate per node. Hence the for loop of Algorithm 2 has the time complexity of $O(n \log n + bn(n-1))$. We have already discussed the time complexity of interference aware ordering and computing minimum delay which are $O((b+1) \cdot n^2 + n)$ and $O(2n)$ respectively. The time complexity of the last step which is to run Aggressive Heuristic without the shared steps is $O(5(b-1)n^2 + n)$ and the time complexity of the shared steps were $O(b \cdot n^3 + n^2 \log n + (b+1) \cdot n^2 + 3n)$. altogether the time complexity of Explorer becomes $O(bn^3 + n^2 \log n + 8bn^2 - 3n^2 + n \log n + 7n)$ which is $O(n^3)$.

6.1.6 noDMS

Our last algorithm is designed to compare the performance our heuristics against no dynamic modulation scaling based energy management. Here, we assume the nodes are equipped with radios fixed to a modulation level which will guarantee to meet a given deadline. However, noDMS is capable of selecting different output power levels based on the distances between pair of nodes for a given modulation level.

noDMS follows a greedy algorithm where each nodes picks a parent which is closest to itself and hence uses the lowest possible output power levels to communicate.

7 Performance evaluation

In order to evaluate the optimal solution and the heuristics, we have created a custom Java simulator to simulate various network setups. The implementation of optimization problem is done by using IBM ILOG CPLEX Optimization Studio libraries [23].

In order to use CPLEX we needed to take the following into account. In the Mixed Integer Nonlinear Programming formulation we have presented in Sect. 6, the Constraints 3c, 3e, 3f, 3g and 3h are not necessarily convex functions which is a problem for the CPLEX software because CPLEX will throw an exception and stop execution when faced with non-convex constraints. To overcome this issue and convert the formulation into a linear programming problem, we needed to create an additional auxiliary variable, z, for each integer variable indicator (β) and a continuous variable (S and E) product such that:

$$z \leq deadline \cdot \beta$$
$$z \geq 0$$
$$z \leq S$$
$$z \geq S - deadline \cdot (1 - \beta)$$

Here is β is 0 then $z \leq 0$ and $z \geq 0$ forces z to be equal to 0. If β is 1 then $z \leq S$ and $z \geq S$ forces z to be equal to S.

Lastly, for the deadline factor of 1 all the algorithms, except for Optimal, choose the closest parent to itself and

only uses 2DPSK since meeting the deadline is trial and does not require DMS.

7.1 Network topology setup

Our simulation starts with the network topology setup process with the following parameters:

- x–y plane size: The size of x axis and the y axis of the setup area.
- number of nodes: The total number of nodes in the network in addition to the base station.
- energy consumption: The energy consumption of each modulation level per transmitter and receiver distance obtained from Figs. 16, 17, and 18.
- delay: The time it takes to send a packet per modulation level.
- deadline factor: the percentage of the longest delay which is when each node is using the lowest modulation level. For example when *deadlinefactor* is 0.6 that means the delay bound D of the Constraint 3j is (number of nodes x delay of a transmitting a packet with modulation level 1×0.6).

We preset the location of the base-station and x, y positions for the remaining nodes are randomly generated while making sure no two nodes are assigned the same coordinates. Next step is to setup the connections. For each pair of nodes (including the base-station), their straight line distance to each other is computed and the energy consumption for this pair of nodes per each modulation level is assigned as the following; for each curve that is plotted in Figs. 16, 17, and 18, we have fitted the curves with 5th degree polynomial trend-lines with the coefficient of determination values of 0.99 and above. We have then assigned the corresponding smallest possible output power levels while making sure there is at least 75% PDR. This corresponds to the $e_l^{i,k}$ parameter of our MINLP where the pair of nodes are i and k and the modulation level is l. If this pair of nodes are unreachable for a modulation level or has a PDR of less than 75%, the energy consumption and the delay of that modulation level is assigned to infinity. Moreover, we have assumed the interference range is 1.2 times of the communications range [38]. If after this process, there exists a node such that it is not within the reach of any other node, then we reassign a new location to that node, once again randomly. This process is repeated till every node is connected.

7.2 Results

In our first set of tests, we have placed the coordinator at the coordinate (0,0) and positioned the remaining 10 nodes by following the procedure described in Sect. 7.1. The optimal algorithm is solution obtained from the CPLEX software. Each data point on Figures are the averaged results over 100 repetitions.

In all experiments, noDMS algorithm has used a fixed modulation of 8DSPK for deadline factors of [0.25, …, 0.45], 4DPSK for [0.5, …, 0.95] and 2DPSK for 1. Because only these modulation levels can give real-time guarantees for their respective deadline factors. As a result you will see flat performance lines from noDMS for these intervals.

We have experimented with 250×250, 500×500, and 750×750 m^2 areas with deadline factor from 0.25 to 1. Figures 25, 26, 27 show the total sum of each node's power consumption (except for the base-station) for a given deadline factor and the maximum size of the area which the nodes are spread. It is seen that for each density level and deadline factor, there is a strict ordering of power consumption as Optimal < Explorer < Aggressive < noDMS except for when the deadline factor equals 1. At this point, the deadline is large enough for all the nodes to use the lowest modulation level to meet the deadline by simply choosing the closest node as their parents. Note that Optimal in Fig. 26 is missing data points for deadline factor 0.25 and in Fig. 27 for 0.25 and 0.30. For these data points, no solution has been found. This shows that when the density of the network decreases, it is harder to meet the shorter deadlines. If you recall our distance test results based on various power output levels (Figs. 13, 14, 15, 16, 17, 18), higher modulation levels have lower successful communication ranges which forces sparse networks to use lower modulation levels to stay connected which in turn increases transmission delay. For the same reason, noDMS was also not be able to find a parent for almost every time when it was only allowed to use 8DPSK as a result no data points are shown in Figs. 26 and 27.

In terms of the impact of network density, a denser network decreases the Optimal's as well as the Heuristics' total power consumption for tighter deadlines. As networks get denser, they will have more potential paths to choose from. In fact a network will preserve all of its potential paths if it was to get *squeezed*. This is because any connection provided by any modulation level will still remain valid when the distances between sender and receiver pairs are decreased. It is possible there will even appear new potential paths. However, the co-schedulability of the nodes will be effected due to the increased interference. Another observation is the performance difference between our algorithms shrinks as the network gets more sparse. This also due to decrease in the number of potential paths. Higher number of potential paths allows Optimal solution to perform exhaustive search over more potential solutions. However, due to the polynomial time nature of our

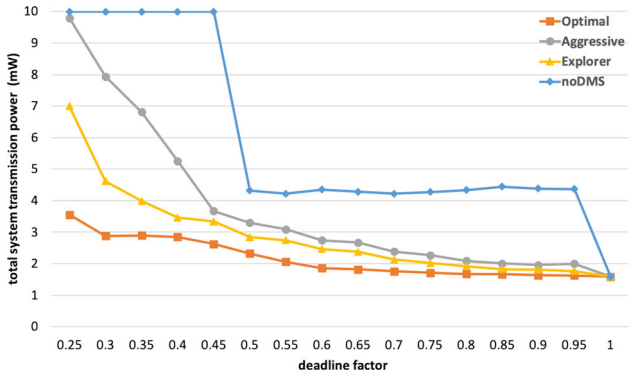

Fig. 25 Comparison of optimal solution against heuristics when the base-station is placed at (0,0) with 10 nodes on a 250 × 250 m^2 area

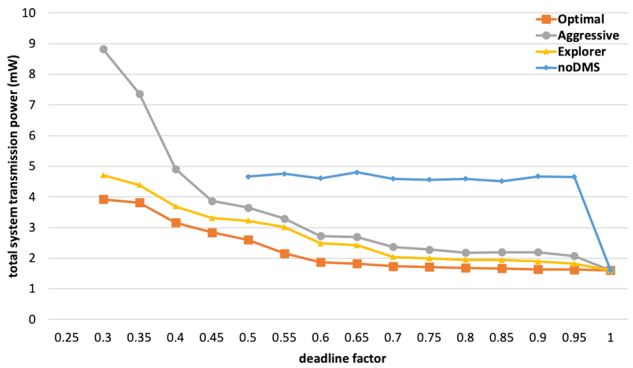

Fig. 26 Comparison of optimal solution against heuristics when the base-station is placed at (0,0) with 10 nodes on a 500 × 500 m^2 area

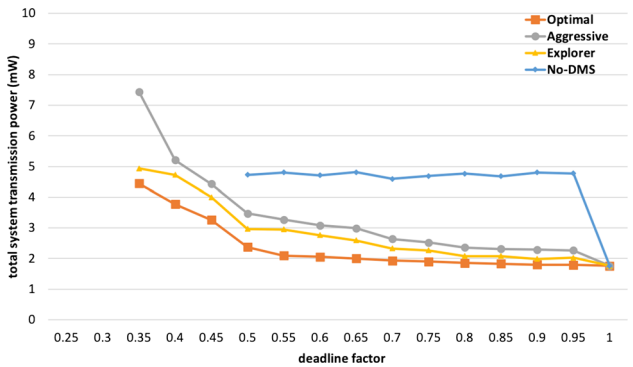

Fig. 27 Comparison of optimal solution against heuristics when the base-station is placed at (0,0) with 10 nodes on a 750 × 750 m^2 area

Heuristics, Aggressive and Explorer does not benefit from higher number of potential paths as much as Optimal does.

The rate of which the total power consumption decreases as the deadline factor increases is not constant. We can see that there are slight reductions in total power consumption when the deadline factor goes from 0.6 to 1 for all density levels. Although from the Figures the line looks relatively flat, there are roughly 14%, 35%, and 42% reduction in power consumption between deadline factors of 0.6 and 1 for Optimal, Explorer, and Aggressive respectively for each case. Our SDR based experimental results indicated that the power consumption difference between 2DPSK and 4DPSK is significantly lower than the difference between 4DPSK and 8DPSK. Hence, the reduction in total power consumption when more and more nodes are allowed to use 2DPSK rather than 4DPSK as the deadline factor approaches 1 is less than that of when more and mode nodes are allowed to use 4DPSK as opposed to 8DPSK when the deadline factor approaches 0.5. This further emphasizes the benefit of DMS on wide ranges of topologies for multi-hop networks and its relatively low scarifies of additional power consumption for a significant reduction in total transmission delay.

Similar traits are also observed when we repeat the tests but this time the base-station is placed in the center of the grid as oppose to corner. Placing the base-station in the center will reduce the average height of the tree and the nodes will have less hops to reach to it. This in return will improve co-schedulability due to less number of nodes affected by parent-children constraint (Sect. 6, Optimization Formulation 3, Constraint 3d). Since the size of the area is kept the same, the co-schedulability due to interference constraints will be unaffected (Sect. 6, Optimization Formulation 3, Constraints 3e, 3f, 3g, and 3h). The results are shown in Figs. 28, 29, 30. If we compare the results shown in Figs. 25 to 28, 26 to 29, and 27 to 30, we can see that total power consumption has been reduced in each case. This is due to the improved co-schedulability as explained. Moreover, we can see that as the network area increase, so does the total power consumption just as we have seen in Figs. 25, 26, 27.

Next, in order to analyze the effect of interference we have conducted a test by significantly reducing the test area by forcing the nodes to be placed on a line with 250, 500, and 750 m length. Reducing the test area size to $\frac{1}{250}$, $\frac{1}{500}$, and $\frac{1}{750}$ of the previous test areas substantially increased the interference and reduced co-schedulability. However, due to the significantly reduced space the average distance between nodes are also reduced which was reflected as the overall reduction in total system power. Figures 31, 32, 33 show the results. Once again we have observed that as the network area gets larger, the total power consumption increases. The reason is the reduced number of potential paths as we discussed previously. If we compare these results with Figs. 28, 29, 30, we can see that the power consumption in each case has descreased, that is for each area size and deadline factor. This comparison also gives us a good insight into the co-schedulability performance of our heuristics. We observe that the performance difference between our algorithms has decreased as we reduced the test area size. This is due to the suboptimal ordering

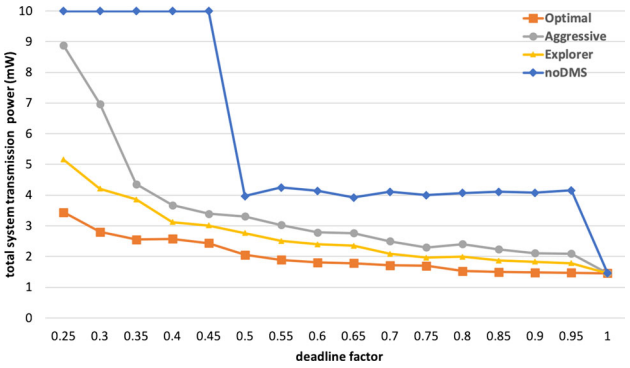

Fig. 28 Comparison of optimal solution against heuristics when the base-station is placed at the *center* with 10 nodes on a 250 × 250 m^2 area

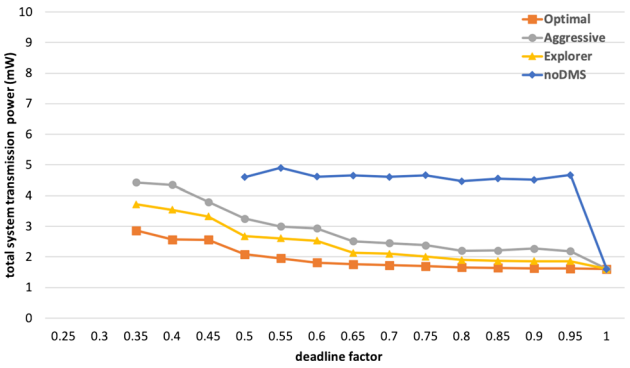

Fig. 29 Comparison of optimal solution against heuristics when the base-station is placed at the *center* with 10 nodes on a 500 × 500 m^2 area

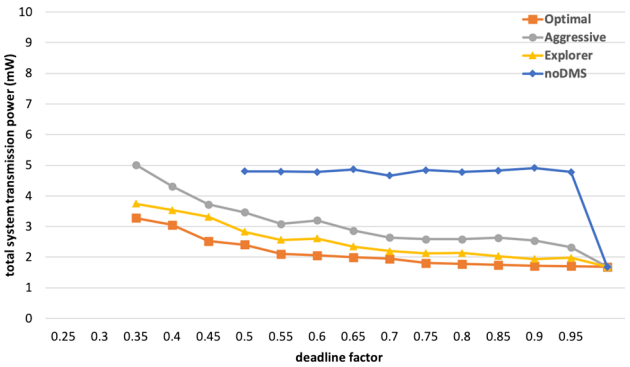

Fig. 30 Comparison of optimal solution against heuristics when the base-station is placed at the *center* with 10 nodes on a 750 × 750 m^2 area

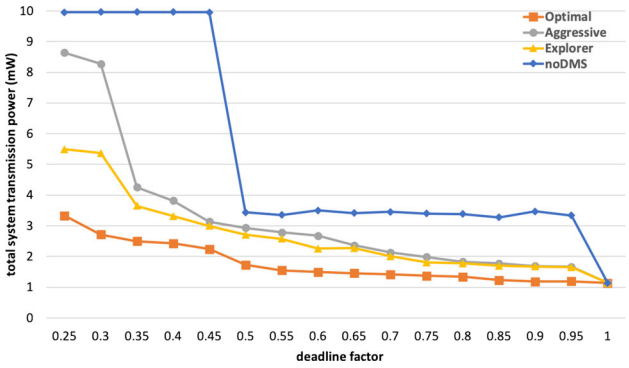

Fig. 31 Comparison of optimal solution against heuristics when the base-station is placed at (0,0) with 10 nodes on a 250 × 1 m^2 area

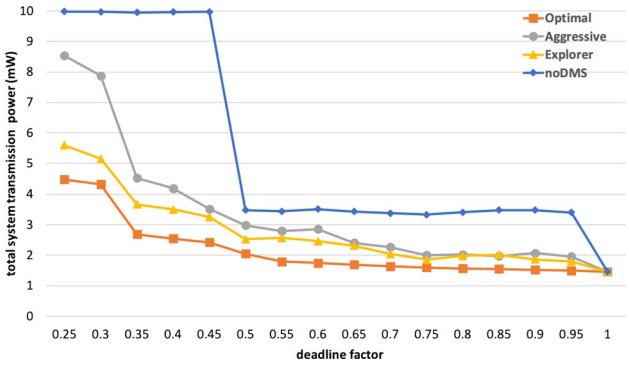

Fig. 32 Comparison of optimal solution against heuristics when the base-station is placed at (0,0) with 10 nodes on a 500 × 1 m^2 area

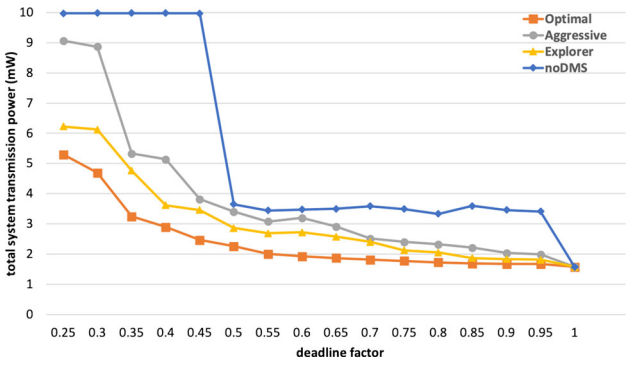

Fig. 33 Comparison of optimal solution against heuristics when the base-station is placed at (0,0) with 10 nodes on a 750 × 1 m^2 area

mechanism used by our heuristics (See Sect. 6.1) which has a negative impact on co-schedulability.

The interference model we have adapted in this paper is first and big step forward towards a more realistic analysis of DMS on WSNs. In our interference model, each modulation level has its own interference range which is 1.2 times of their real work communication range measurements obtained from our SDR tests. The work in literature typically assumes equal interference ranges for different modulation levels. In order to see the effect of our new interference model, we conducted the following test; 10 nodes plus the coordinator placed at (0,0) on a 500 × 500 m^2 area. The same layout used for generating Fig. 26. However, this time the interference range of all the

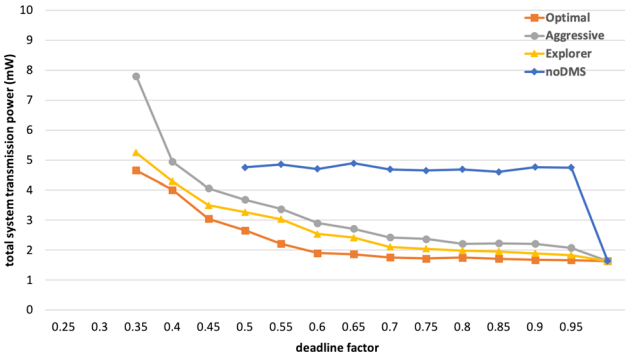

Fig. 34 Comparison of optimal solution against heuristics when every modulation level has the same interference range of modulation level 1. Area size is 500 × 500 m^2 and network size is 10 nodes plus the coordinator placed at (0,0)

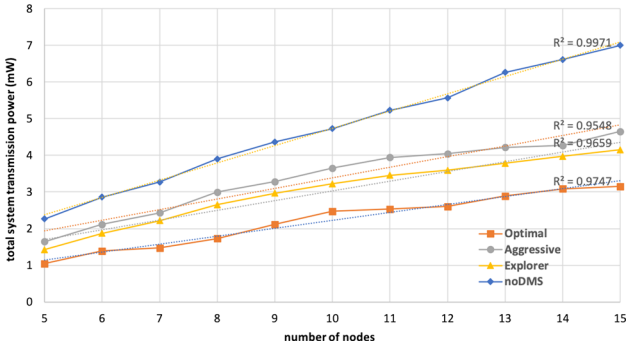

Fig. 35 Scalability analysis in terms of number of nodes in a 500 × 500 m^2 area and the coordinator is placed at the coordinates (0,0) with a deadline factor of 0.5

modulation levels equal to that of modulation level 1. In other words, 4DPSK and 8DPSK also has the same interference range of 2DPSK. This is a hypothetical assumption since the communication ranges of all the modulation levels were kept the same. noDMS is not affected by interference because it always uses a fixed modulation level which already guarantees to meet the deadline. Figure 34 shows the results. On average there were 5.7%, 2.46%, and 4.11% increase on power consumption for Optimal, Aggressive, and Explorer respectively when compared to the results shown in Fig. 26. The increase in power consumption was more pronounced for smaller deadline factors. For the deadline factor of 0.35, the performance difference was 21.95%, 5.92%, and 19.52% for Optimal, Aggressive, and Explorer respectively. This shows the significance of co-schedulability especially for shorter deadlines.

We want to mention another observation from comparing the results from various topologies presented so far. The performance difference between these various tests becomes relatively smaller when the deadline factor is 0.5 or more. For Optimal at deadline factor of 0.5, the difference between the maximum and the minimum observed power consumption is 22%. This value is 33% and 37% for Explorer and Aggressive respectively. For the deadline factor of 0.6, these differences shrink to 14%, 25%, and 35% for Optimal, Explorer, and Aggressive. The reason behind this is the relatively close performance of 2DPSK and 4DPSK. At these deadline factors, the nodes are dominantly using either 2DPSK or 4DPSK. The percentage of each changes from case to case but the differences in these percentages create a relatively small difference in performance.

To analyze the scalability of our algorithms, we have run a set of experiments with 5–15 nodes in a 500 × 500 m^2 area and a coordinator is placed at the coordinates (0,0) with a deadline factor of 0.5. We chose 500 × 500 m^2 and a deadline factor of 0.5 to have relatively low computational load (smaller area sizes significantly increases the time optimal solution is computed due to increased interference among nodes) and still have a connected network (as the deadline factor decreases so does the connectivity of the network). We could only go up to 15 nodes because after that point, we were unable to find an optimal solution even after 5 days of running. Figure 35 shows the results fitted with linear regression lines to each algorithm. Coefficient of determination (R^2) values of linear regression analysis were above 95% for all of the algorithms. This is a strong indication that the energy consumption of the particular setup we had grows linearly in terms of number of nodes. Moreover, the aims of the linear regression lines were also similar for the algorithms, 0.29, 0.26, and 0.22 for Aggressive, Explorer and Optimal respectively. That means although the performance difference between the algorithms diverges as the number of nodes increases, it is with a slow rate. So one can expect similar performances from all the algorithms as the number of nodes increases.

8 Conclusion

Applicability of Dynamic Modulation Scaling (DMS) in low-power wireless systems, including IEEE 802.15.4 networks, as an energy management technique has been widely studied in literature and shown that DMS can achieve significant energy savings. In this paper we have addressed the problem of applying DMS to multi-hop/multi-cluster real-time WSNs. We first presented the Mixed Integer Nonlinear Programming (MINLP) formulation of minimizing total network energy consumption while satisfying the deadline requirements. This formulation picks the most energy efficient parent node and a modulation level for each node in the network. We later

show how to linearize this MINLP formulation to convert it into Mixed Integer Linear Programming (MILP) problem which is typically a requirement for problem solving engines such as IBM's CPLEX.

However, much the work that has been done so far lacks substantial empirical analysis of DMS on Wireless Sensor Networks (WSNs). As a result, only a generalized mathematical evaluation of DMS were available to the researchers. These evaluations were not enough to for us to move forward with the solution to the problem we have formulated. Hence, this paper addresses that gap and provides empirical results that can be used by not just us but future researchers as a complement to currently available mathematical techniques. To accomplished this, we used Ettus B210 Software Defined Radios (SDRs) and configured them according to the 2015 IEEE 802.15.4 standard. We experiments with various Signal-to-Noise-Ratios and corresponding Packet-Delivery-Rates for {2, 4, 8, 16}-PSK as well as DPSK modulations. Our results show that increasing PSK modulation levels requires more transmission power increase compared to scaling functions (such as those shown in Table 1). On the other hand, DPSK requires significantly *less* power increase compared to the same scaling function. We have also shown that the scaling ratios between consecutive modulation levels are not equal. Next, we measured the PDRs for {2, 4, 8}-DPSK for distances up to 100 m. The results have shown that lower modulation levels can achieve higher communication distances. We have also shown the performance difference between 915 MHz and 2.484 GHz and observed 915 MHz has higher packet delivery rate with the same output amplifier gain. Lastly, we have shown that elevation difference between transmitter and receiver affects these results tremendously. 4-DPSK and 8-DPSK had as low as 54% and 28% PDR for elevation difference of 11.6 m and 17.48 m apart. The same modulations had 99% and 76% PDR respectively for distances up to 20 m with no elevation difference.

We later designed two polynomial time heuristics and compared them against the optimal solution obtained by CPLEX software. In order to have an accurate network model, we have incorporated the empirical measurements obtained from our SDR tests into our custom simulator. We ran various tests with different network area sizes and base-station locations and showed our polynomial time heuristics perform very closely to the optimal solution especially for larger deadlines. We have also shown that there is relatively less energy consumption penalty if the deadline of the system were to be halved that is if more and more nodes needed to use 4DPSK rather than 2DPSK. However, as the deadline further reduced and more nodes are required to use 8DPSK to meet the deadline, there is a significant increase in total energy consumption.

Acknowledgements The authors would like to thank Dr. Duminda Wijesekera from Computer Science Department, George Mason University for giving us access to Radar and Radio Engineering Lab's Faraday cages. Experiments presented in Sect. 7 were run on ARGO, a research computing cluster provided by the Office of Research Computing at George Mason University, VA. (URL:http://orc.gmu.edu).

References

1. Adler, M., Sitaraman, R. K., Rosenberg, A. L., & Unger, W. (1998). Scheduling time-constrained communication in linear networks. In: Proceedings of the tenth annual ACM symposium on parallel algorithms and architectures (pp. 269–278). ACM.
2. Anastasi, G., Conti, M., Di Francesco, M., & Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, *7*(3), 537–568.
3. Bandara, K. D. S., Melaragno, A., Wijesekera, D., & Costa, P. (2017). A case study of cognitive radio networks: Secure spectrum management for positive train control operations. In: Spectrum access and management for cognitive radio networks (pp. 121–152). Springer.
4. Bandari, M., Simon, R., & Aydin, H. (2016). On minimizing expected energy usage of embedded wireless systems with probabilistic workloads. *Sustainable Computing: Informatics and Systems*, *11*, 50–62.
5. Barbehenn, M. (1998). A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices. *IEEE Transactions on Computers*, *47*(2), 263.
6. Bloessl, B., Segata, M., Sommer, C., & Dressler, F. (2013). Towards an open source IEEE 802.11 p stack: A full SDR-based transceiver in GNU radio. In: 2013 IEEE vehicular networking conference (VNC), (pp. 143–149). IEEE.
7. Cerpa, A., & Estrin, D. (2002). Ascent: Adaptive self-configuring sensor network topologies. *ACM SIGCOMM Computer Communication Review*, *32*(1), 62–62.
8. Chen, B., Jamieson, K., Balakrishnan, H., & Morris, R. (2002). Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, *8*(5), 481–494.
9. Cionca, V., Newe, T., & Dadarlat, V. (2008). TDMA protocol requirements for wireless sensor networks. In: Proceedings of second IEEE international conference on sensor technologies and applications. https://doi.org/10.1109/SENSORCOMM.2008.69.
10. Cui, S., Goldsmith, A. J., & Bahai, A. (2005). Energy-constrained modulation optimization. *IEEE Transactions on Wireless Communications*, *4*(5), 2349–2360.
11. Dousse, O., Mannersalo, P., & Thiran, P. (2004). Latency of wireless sensor networks with uncoordinated power saving mechanisms. In: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (pp. 109–120). ACM.
12. Edbauer, F. (1992). Bit error rate of binary and quaternary DPSK signals with multiple differential feedback detection. *IEEE Transactions on Communications*, *40*(3), 457–460.
13. Fateh, B., & Govindarasu, M. (2013). Energy minimization by exploiting data redundancy in real-time wireless sensor networks. *Ad Hoc Networks*, *11*(6), 1715–1731.
14. Fateh, B., & Govindarasu, M. (2015). Joint scheduling of tasks and messages for energy minimization in interference-aware real-time sensor networks. *IEEE Transactions on Mobile Computing*, *14*(1), 86–98.
15. Galloway, B., & Hancke, G. (2013). Introduction to industrial control networks. *IEEE Communications Surveys Tutorials*,

15(2), 860–880. https://doi.org/10.1109/SURV.2012.071812.00124.
16. Gumusalan, A., Simon, R., & Aydin, H. (2016). Adaptive transmission scheduling for energy-aware real-time wireless communication. In: Proceedings of the 2016 international conference on embedded wireless systems and networks, EWSN'16 (pp. 1–12). USA: Junction Publishing. http://dl.acm.org/citation.cfm?id=2893711.2893713.
17. Gumusalan, A., Simon, R., & Aydin, H. (2018). Flexible real-time transmission scheduling for wireless networks with non-deterministic workloads. *Ad Hoc Networks*, *73*, 65–79.
18. Gungor, V. C., & Hancke, G. P. (2009). Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE Transactions on Industrial Electronics*, *56*(10), 4258–4265.
19. Ho, P., & Fung, D. (1991). Error performance of multiple symbol differential detection of PSK signals transmitted over correlated Rayleigh fading channels. In: IEEE international conference on communications, 1991. ICC'91, conference record (pp. 568–574). IEEE.
20. https://forums.ni.com/t5/USRP-Software-Radio/How-to-set-the-trasmit-power-of-USRP/td-p/2202190
21. IEEE. https://standards.ieee.org/findstds/standard/802.15.4e-2012.html. Retrieved May 22, 2017.
22. IEEE. https://ieeexplore.ieee.org/document/7460875.
23. ILOG, I. (2018). CPLEX optimizer. http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer.
24. Instruments, T. Cc2420: 2.4 GHz IEEE 802.15.4/zigbee-ready RF transceiver. http://www.ti.com/product/CC2420.
25. Instruments, T. Sub 1 GHz radios. http://www.ti.com/wireless-connectivity/simplelink-solutions/sub-1-ghz/products.html. Retrieved March 5, 2018.
26. Khader, O., & Willig, A. (2013). An energy consumption analysis of the wireless HART TDMA protocol. *Computer Communications*, *36*(7), 804–816.
27. Kong, Z., & Yeh, E. M. (2007). Distributed energy management algorithm for large-scale wireless sensor networks. In: Proceedings of the 8th ACM international symposium on mobile ad hoc networking and computing (pp. 209–218). ACM.
28. Lennvall, T., Svensson, S., & Hekland, F. (2008). A comparison of wirelesshart and zigbee for industrial applications. In: Proceedings of IEEE international workshop on factory communication systems.
29. Park, M. (2015). IEEE 802.11 ah: sub-1-GHz license-exempt operation for the internet of things. *IEEE Communications Magazine*, *53*(9), 145–151.
30. Raghunathan, V., Schurgers, C., Park, S., & Srivastava, M. B. (2002). Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, *19*(2), 40–50.
31. Research, E. Ettus usrp b210. https://www.ettus.com/product/details/UB210-KIT. Retrieved February 2, 2018.
32. Schurgers, C., Raghunathan, V., & Srivastava, M. B. (2003). Power management for energy-aware communication systems. *ACM Transactions on Embedded Computing Systems*, *2*(3), 431–447.
33. Song, J., Han, S., Mok, A. K., Chen, D., Lucas, M., & Nixon, M. (2008). Wirelesshart: Applying wireless technology in real-time industrial process control. In: Proceedings of IEEE real-time and embedded technology and applications symposium.
34. Wang, H., & Fapojuwo, A. O. (2017). A survey of enabling technologies of low power and long range machine-to-machine communications. *IEEE Communications Surveys & Tutorials*, *19*(4), 2621–2639.
35. Xiong, X., Zheng, K., Xu, R., Xiang, W., & Chatzimisios, P. (2015). Low power wide area machine-to-machine networks: Key techniques and prototype. *IEEE Communications Magazine*, *53*(9), 64–71.
36. Yu, Y., Prasanna, V. K., & Krishnamachari, B. (2006). Energy minimization for real-time data gathering in wireless sensor networks. *IEEE Transactions on Wireless Communications*, *5*(11), 3087–3096.
37. Zhang, B., Simon, R., & Aydin, H. (2013). Harvesting-aware energy management for time-critical wireless sensor networks with joint voltage and modulation scaling. *IEEE Transactions on Industrial Informatics*, *9*(1), 514–526.
38. Zhou, G., He, T., Stankovic, J., Abdelzaher, T., et al. (2005). RID: Radio interference detection in wireless sensor networks. In: Proceedings of 24th IEEE annual joint conference of the IEEE computer and communications societies (Vol. 2, pp. 891–901).

Arda Gumusalan received the B.S. from State University of New York, M.S., and Ph.D. degrees from George Mason University, all are in Computer Science. He is currently working for IBM Cloud Infrastructure—Network Services. His research interests include cloud-computing/networking, internet of things, and industrial control networks. Received best unclassified paper award in 2017 IEEE Military Communications Conference (MILCOM).

Robert Simon received the B.S. degree in history and political science from the University of Rochester, Rochester, NY, and the Ph.D. degree in computer science from the University of Pittsburgh, Pittsburgh, PA. He is a Professor of Computer Science at George Mason University, Fairfax, VA. His research has been supported by a number of agencies, including NSF, DARPA, the U.S. Department of Defense and private industry. His research interests include embedded systems, wireless and mobile computing, distributed systems and performance modeling and analysis, and distributed computing. He has published over 115 peer-reviewed journal and conference papers on these topics, and has received 8 best paper awards.

Hakan Aydin received the B.S. and M.S. degrees in Control and Computer Engineering from Istanbul Technical University, Istanbul, Turkey, and the Ph.D. degree in Computer Science from the University of Pittsburgh, PA. He is currently a Professor of Computer Science at George Mason University, Fairfax, VA. He has served on the program committees of several real-time and embedded systems related conferences and workshops. His research interests include real-time embedded systems, low-power computing, and fault tolerance. Dr. Aydin received the U.S. National Science Foundation (NSF) Faculty Early Career Development (CAREER) Award in 2006. He was the Technical Program Committee Chair of the 2011 IEEE Real-Time Technology and the Applications Symposium (RTAS'11).