



HAL
open science

Optimal proactive monitor placement & scheduling for IoT networks

Basma Mostafa Hassan, Miklos Molnar, Mohamed Saleh, Abderrahim Benslimane, Sally Kassem

► **To cite this version:**

Basma Mostafa Hassan, Miklos Molnar, Mohamed Saleh, Abderrahim Benslimane, Sally Kassem. Optimal proactive monitor placement & scheduling for IoT networks. Journal of the Operational Research Society, 2022, 73 (11), pp.2431-2450. 10.1080/01605682.2021.1992310 . hal-03634316

HAL Id: hal-03634316






<https://hal.science/hal-03634316>

Submitted on 21 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimal Proactive Monitor Placement & Scheduling for IoT Networks

Basma Mostafa^{a, b} , Miklos Molnar^b , Mohamed Saleh^a , Abderrahim Benslimane^c , and Sally Kassem^{a, c}  ^aFaculty of Computers & Artificial Intelligence, Cairo University, Cairo, Egypt; ^bLIRMM, Université de Montpellier, Montpellier, France; ^cLIA, Université d'Avignon, Avignon; France; ^c Smart Engineering Systems Center, Nile University, Cairo, Egypt

ARTICLE HISTORY

Compiled October 7, 2020

ABSTRACT

This work is fulfilled in the context of the optimized monitoring of Internet of Things (IoT) networks. IoT networks are faulty; Things are resource-constrained in terms of energy and computational capabilities; they are also connected via lossy links. For IoT systems performing a critical mission, it is crucial to ensure connectivity, availability, and network reliability, which requires *proactive* network monitoring. The idea is to oversee the network state and functioning of the nodes and links; to ensure the early detection of faults and decrease node unreachability times. It is imperative to minimize the resulting monitoring energy consumption to allow the IoT network to perform its functions. Furthermore, to realize the integration of the monitoring mechanism with IoT services, this latter should work in tandem with the IoT standardized protocols, especially the IPv6 for Low-power Wireless Personal Area Networks (6LoWPAN) and the Routing Protocol for Low-power and lossy networks (RPL). In this paper, an optimized, proactive, passive, centralized monitoring system is proposed for IoT networks. The proposition ensures the optimal placement of monitoring nodes (*monitors*). Leveraging the graph built by RPL for routing (the DODAG), minimal sets of monitors are optimally placed to cover a given domain. The monitoring activity is optimally scheduled between several subsets of nodes to prolong longevity while minimizing the energy consumption for monitoring, communication, and state transitions. Our proposition provides the exact solution to the defined monitoring placement and scheduling problem via a Binary Integer Program. The model serves as a benchmark for the performance evaluation of contemporary models. Experimentation is designed using network instances of different topology. Results demonstrate the proposed model's effectiveness in realizing full monitoring coverage with minimum energy consumption and communication overhead and a balanced distributed monitoring role.

KEYWORDS

IoT; Optimization; Critical missions; Reliability; Proactive monitoring; Monitor assignment; Scheduling; RPL

1. Introduction

The Internet of Things (IoT) is an Internet-based network of networks; which relies on arbitrary smart, low-power devices (Things) capable of performing sensing or actuation, and communication tasks. By connecting billions of Things to the Internet, IoT created a plethora of applications that touch every aspect of human life, including Smart Homes, Wearables, Smart Cities, Smart Grids, Connected Cars, and Connected Health (Dhall & Solanki, 2017; Metcalf, Milliard, Gomez, & Schwartz, 2016).

Contrasting the IoT and the standard Internet, the ultimate difference resides in the fact that IoT networks mainly use Low-power Lossy Networks (LLN). LLNs have stringent resource constraints with respect to energy, processing power, and memory of devices (Montenegro, Kushalnagar, Hui, & Culler, 2007). They are also known for their unreliable, lossy channels, with low-power, low-bit-rate, and unpredictable bandwidth. The links are typically non-transitive, and their local scope is defined by node reachability and radio strength.

The connection of such networks to the IPv6-based Internet required an adaptation layer between the MAC and network layers; hence, the IPv6 over LoWPAN (6LowPAN) (Kushalnagar, Montenegro, & Schumacher, 2007). 6LowPAN enabled the reliance on IPv6 for addressing, which allowed the provision of the large address space required for connecting such a tremendous number of devices to the Internet.

On the other hand, IoT networks are characterized by their vulnerability to security risks from the Internet and the shared wireless medium, self-configuration, lack of infrastructure, and complexity of the design of network protocols (Türkoğulları, Aras, Altinel, & Ersoy, 2010; Yücel & Altın-Kayhan, 2019). They are possibly deployed in unknown, hostile networks with highly dynamic network topologies. Energy constraints impose hard duty cycles to maximize longevity, causing unreliable connectivity and eventual node unreachability (Jara, Ladid, & Gómez-Skarmeta, 2013), leading to incomplete information about the current network state. The situation is regarded as a form of entropy, where a system deteriorates unless effort is invested in developing monitoring and correction mechanisms to maintain a fault-tolerant performance.

Knowing when things break is good. Knowing before they break is even better. Although a significant number of IoT applications are not time-sensitive, there is a whole class of mission-critical applications, especially those that target human safety. For instance, health monitoring, critical control, and fault detection applications (Hassanalieragh et al., 2015; Wu, Wu, & Yuce, 2019). These applications require a high level of network reliability. According to IEEE, reliability is *"the ability of a system or component to perform its required functions under stated conditions and for a specified period of time."* (Stanisavljević, Schmid, & Leblebici, 2010) Unfortunately, given the unreliable nature of LLNs and IoT, faults are common rather than rare events (Kiani, 2018). Maintaining robustness, continuous availability of devices, and communication reliability are critical factors to guarantee a reliable application data flow.

For mission-critical applications, *proactive* monitoring approaches are preferable. As a kind of preventive maintenance, proactive mechanisms enforce continual network monitoring; so that node and link failures are detected early, and alerts are promptly issued. Consequently, disconnectivity and service failures are prevented from occurring in the first place. Nevertheless, all supplementary monitoring mechanisms must have minimal effect on energy consumption and traffic load, leaving the network unconstrained in performing its normal function of sensing, actuation, or transmission. If not applied carefully, proactively verifying network performance will negatively impact nodes' resources. If ignored, this impact may lead to battery depletion due to idle listening to the radio channel and excessive control, increased congestion, or network traffic delays, violating critical applications' requirements. Consequently, the existing (conventional) monitoring mechanisms cannot be applied directly to the IoT. Therefore, optimizing the monitoring energy consumption and traffic overhead is crucial.

To realize the integration with already-existing (and future) IoT services, it is detrimental that monitoring prepositions are entirely interoperable with the standardized IoT protocol suite, especially 6LoWPAN and RPL. Interoperability is challenging mainly because IoT solutions are often tailored to specific scenario requirements without neither focusing on horizontal integration with other IoT services nor re-usability.

In light of the above, the absence of any monitoring mechanism for detecting network faults would dramatically reduce the network's performance, which renders monitoring the IoT network state a vital research area that will only develop in significance. An effective and efficient monitoring mechanism could immensely improve robustness in network connectivity, reliability, and, eventually, Quality of Service, which will significantly increase the uptake of the technology by stakeholders, especially for mission-critical applications.

In this work, we propose a proactive, passive monitoring mechanism of mission-critical 6LoWPAN-based IoT networks. The problem is mathematically formulated, and the optimal solution to the minimum monitor assignment problem and the scheduling of the monitoring roles throughout a predetermined lifetime is given. The objective of the mathematical model is minimizing the amount of energy consumed in (1) monitoring the set of critical nodes, (2) communication of the monitoring data to the central entity (6LoWPAN Border Router), (3) and transition between monitoring states. To the best of our knowledge, the exact solution to monitoring mission-critical 6LoWPAN-based IoT networks has not yet been analyzed. The global optimum will serve as a benchmark for comparisons and performance evaluation of contemporary models.

The rest of the paper is structured as follows. A brief background and overview of the related monitoring techniques for Wireless Sensor Networks (WSN) and IoT are given in Section 2. The monitoring problem requirements, assumptions, and objectives are provided in Section 3. The problem is mathematically formulated in Section 4. The proposed proactive, passive, optimal IoT monitoring mechanism is described in Section 5. The model is then implemented using a domain-specific modeling language for mathematical optimization embedded in Julia language and solved using a Gurobi solver, tested on networks with different topology, and results are illustrated in Section 6. Finally, conclusions and insights into possible future research directions are given in Section 7.

2. Monitoring for IoT Networks: Research Gap

Several network management and routing protocols are designed to deal with the inherent technical challenges of the IoT. However, the Routing Protocol for Low-power and lossy networks (RPL) is the best candidate for critical systems that need fast recovery mechanisms (Geng, 2017). RPL is defined by the Internet Engineering Task Force (IETF), specifically the Routing Over Low-power Lossy networks (ROLL) working group. RPL is a self-healing routing and topology control protocol. It can respond to some node or link failures by applying route repair mechanisms for network recovery.

RPL favors the use of *reactive* repair approaches; (pro)active mechanisms for regularly probing neighbors do not exist in the ContikiRPL implementation, to minimize the cost of monitoring the links that are not being used. Neighbor Unreachability Detection (NUD) is not obligatory in neither 6LoWPAN nor RPL (Gaddour & Koubâa, 2012), not until a node had already failed to reach its default router (parent) (Lin et al., 2017). Only then, RPL triggers a repair mechanism. Thus, fault-tolerance is traded for routing stability and less control traffic (Korte, Sehgal, & Schönwälder, 2012). As a result, the recovery time, which is the time required to establish a new route in the case of node unreachability, could be relatively long.

Monitoring for WSNs has been approached frequently in the literature; a survey is provided by (Suriyachai, Roedig, & Scott, 2011). Several heuristics have been proposed to place monitors to uniquely localize a limited number of link failures (Ma, He, Swami, Towsley, & Leung, 2015; Stanic, Subramaniam, Sahin, Choi, & Choi, 2010). The sniffer technology for WSNs is one of the distinguished passive real-time monitoring tools. Sniffers listen to the packets transmitted over the network and directly capture them from the shared wireless medium. The

information obtained from the sniffed packets gives them real-time access to network operations, the ability to promptly assess network performance, and detect network malfunctions without affecting the network's functioning. Information may include partial topology, routing information, and data content. There have been some related works on sniffing tools for WSNs in academia and industry, such as SNDS (Kuang & Shen, 2010). However, their high costs and lack of integration with LoWPAN protocols such as 6LoWPAN and RPL diminish their application into the IoT domain (Zhao, Huangfu, & Sun, 2012).

Unlike WSNs, most of the research work pertaining to fault-tolerance in the IoT focus on security (Mayzaud, Sehgal, Badonnel, Chrisment, & Schönwälder, 2016), intrusion detection, or anomaly detection (Zarpelão, Miani, Kawakani, & de Alvarenga, 2017). These solutions are security-oriented and do not provide solutions to the monitoring aspects that target the underlying network structure, more specifically, guaranteeing node availability and stable, reliable, and scalable end-to-end connectivity. A survey of the state-of-the-art security methods in IoT is conducted by (Alaba, Othman, Hashem, & Alotaibi, 2017).

The few research work that tackles related IoT network-layer monitoring problems often suffers from being heavy-weight or depend only on highly-powered nodes, such as (Mayzaud et al., 2016). (Sehgal, Perelman, Kuryla, & Schönwälder, 2012) investigated how to adapt existing IP-based network management protocols, namely SNMP and NETCONF, enabling their implementation on resource-constrained devices. Service interfaces were simplified to include a subset of their functions to minimize network overhead. The authors conclude that the time and memory requirements are low, with only trivial security levels. However, enabling authentication and privacy increases message processing times significantly.

The most relevant work is a mechanism for passive monitoring with RPL, proposed by (Mayzaud et al., 2016). The monitoring responsibility is put exclusively on higher-order devices that are not limited in their resources, to reduce the overhead on the constrained IoT nodes. The mechanism imposes a hard constraint since higher-order devices typically do not constitute the majority of nodes in IoT networks. According to their location and the topology, it might be impossible to entirely cover the critical set of nodes and links using only highly powered devices. Moreover, determining the optimal placement of monitoring nodes was beyond the scope of their proposed work.

According to the comprehensive literature review performed in this research, we noticed that the optimal placement of monitors to cover a mission-critical IoT network had not been proposed so far. To the best of our knowledge, no research work has proposed monitoring models with optimized, energy-efficient role scheduling and integration with RPL and 6LoWPAN protocols (Alaba et al., 2017).

3. IoT Monitoring Assumptions, Requirements & Objectives

This section explains the monitoring problem requirements, assumptions, and objectives; stated in a nutshell in Table 1. IoT network topology is often unstable due to node mobility, unreliable connectivity, and the fact that link connections between nodes are transient (Jara et al., 2013) (Table 1: Assumptions 1 & 2). In general, network monitoring mechanisms aim at detecting and localizing network faults. They should provide the appropriate tools for overseeing the network state, availability of, and connectivity between nodes. By mapping symptoms of detected problems to possible root causes, the necessary corrective measures can be taken.

Focusing on *mission-critical* IoT network services, the main problem is to guarantee network reliability, nodes' availability, and robust connectivity. The proposed monitoring mechanism aims to observe network traffic to verify the availability of the critical set of nodes.

Table 1. Monitoring Requirements, Assumptions & Objectives

Assumptions	Reference
The IoT network is <i>unstable</i> , 6LoWPAN-based, uses RPL for routing, and performs a <i>critical-mission</i>	1
Links are lossy and can only be monitored by their extremities	2
In addition to the monitoring role, things perform sensing, transmission, and/or actuation	3
Things have stringent <i>resource constraints</i> with only a fraction of the battery reserved for monitoring	4
The monitoring mechanism is <i>centralized</i> , and the 6LoWPAN Border Router (6BR) is the central entity where the monitoring data is gathered	5
The active/sleep alternation is the turn on/off of the monitoring activity of the active node	6
Requirements	Reference
Monitors should be placed in the correct locations to guarantee full monitoring coverage	1
The monitoring energy and communication costs should be minimal to satisfy the low-cost, low-power and scalable objectives of LLNs	2
The monitoring mechanism should support the presence of sleeping nodes	3
A <i>monitoring</i> duty cycle mechanism is required, where the monitoring function is periodical across the planning horizon	4
Monitoring should be <i>interoperable</i> with the standardized IoT protocol suite, specifically 6LoWPAN and RPL protocols.	5
Objectives	Reference
Minimize the number of placed monitors while respecting the monitoring <i>coverage requirement</i> , regardless of the lack of current network activity	1
Place the monitors such that in the <i>centralized</i> mechanism, the energy consumed in <i>relaying</i> the data to the 6BR is minimized	2
<i>Balance</i> the monitoring role among nodes	3
Provide the <i>exact</i> solution to the optimal scheduling of the monitoring roles throughout a predetermined lifetime, using minimal monitor sets in each period	4
Minimizing the state transitions resulting from <i>duty-cycling the monitoring activity</i>	5
Support <i>passive</i> (and active) monitoring approaches (although passive monitoring is recommended)	6

Node failures can be utilized to model failures of both physical nodes and links (Ma et al., 2015). Performance metrics such as end-to-end delay and link quality level are not that significant in this context as long as successful transmission of mission-critical-related information is always guaranteed.

IoT networks are of enormous scale, consisting of potentially (hundreds of) thousands of nodes. Naturally, it is required that monitors are embedded (*placed*) in the correct locations to guarantee full monitoring *coverage* (Table 1: Requirement 1). Given the constrained resources of LLNs, it is also required to reduce the monitoring cost, this energy consumption is different from the energy dedicated to the thing’s main function (Table 1: Assumptions 3) & 4). Therefore, the number of monitors to be placed must be minimized (Table 1: Objective 1), while satisfying the coverage condition (Table 1: Requirement 2).

The most common IoT architectures are entirely centralized, mainly due to security reasons. The 6LoWPAN Border Router (6BR) is the central entity and is always assumed to be accessible (Sheng et al., 2013). Therefore, 6BRs can perform a potentially crucial role in centralized monitoring (Table 1: Assumption 5). Through multi-hop communication, the gathered data can be forwarded from monitors to the 6BR, then to a Network Operations Center (NOC), where sophisticated data analysis and mining can be performed. However, energy is lost in the communication between monitors and the 6BR, which is why it is necessary to find the *shortest path* in terms of the number of hops to the 6BR (Table 1: Objective 2).

Furthermore, since life span concerns ordinarily constrain the design of LLNs, a prevalent approach toward expanding network longevity is by utilizing duty cycling (Türkoğulları et al., 2010). In duty-cycled networks, nodes enter sleep state frequently to conserve energy, and intermittently wake up to check for action (Sahoo, Thakkar, Hwang, et al., 2017). High redundancy in network deployment is necessary to achieve this goal; only then is it possible to identify small subsets of active nodes at a time and put the major part of nodes into a sleeping state and thus saving energy (Table 1: Requirement 3). Various scheduling algorithms are applied to organize the alternation between active and sleeping node sets to provide continuous network service. For critical applications, monitoring coverage should always be guaranteed throughout the entire network lifetime, where each link is monitored by at least one monitoring node, regardless of the lack of activity in the network. In this context, the active/sleep alternation is the turn on/off of the monitoring activity of the active node (Table 1: Assumption 6). Therefore, effective and energy-efficient monitoring scheduling algorithms are necessary with duty cycling to satisfy the coverage constraints and balance the distribution of the monitoring burden among nodes, thus maximizing longevity (Table 1: Requirement 4 & Objective 3). In LLNs, nodes can only monitor the traffic within their radio transmission range. Consequently, the monitoring coverage and scheduling connectivity problems are significantly challenging in duty-cycled LLNs.

As stated before, to realize the integration with already existing (and future) IoT services, it is required that monitoring prepositions are completely *interoperable* with the standardized IoT protocol suite, especially 6LoWPAN and RPL-based networks (Table 1: Requirement 5). Our research is related to the optimal placement of monitoring devices to cover a given network topology. The *static* problem of assigning the minimum number of monitors to cover a given domain is known in the literature as the *minimum monitor assignment problem* (Kumar & Kaur, 2004), which has been proven to be NP-hard Liu, Gao, Wu, Dong, and Bu (2015); implying that unless $P = NP$, efficient algorithms for solving it do not exist. In a previous proposition towards monitoring mission-critical IoT networks (Mostafa, Benslimane, Saleh, Kassem, & Molnar, 2018), we tackled the problem using a *Divide and Conquer* approach; via decomposing and mapping it into three well-known sub-problems. Separately solving each sub-problem is efficient, given that there exist several heuristics and approximation algorithms for each sub-problem. However, the one major limitation of the proposed

three-phase decomposition is that it is not an exact solution. Therefore, it does not guarantee global optimality.

Here, we target the exact solution to the minimum monitor assignment problem and the optimal scheduling of monitoring roles throughout a predetermined lifetime (Table 1: Objective 4). The formulated mathematical model's objective is minimizing the amount of energy consumed in monitoring the set of critical nodes, communication of the monitoring data to the central entity (6LoWPAN Border Router), and transition between monitoring states (Table 1: Objective 5).

The proposed mechanism is characterized as an optimized, proactive, centralized, and *passive* monitoring for 6LoWPAN-based IoT networks that utilize the standardized RPL protocol for routing. In passive monitoring, the monitors will only *listen* to the channel, and the monitoring energy is the cost energy consumed when the device is in receiving mode. That said, our mechanism also supports active monitoring, where monitors participate in the network traffic; via probing the monitored neighborhood and collecting the response. Active monitoring can be needed if the network traffic is sparse. However, it is not recommended for the more efficient energy consumption of the resource-constrained things, which is why we limit the experimentation to passive monitoring (Table 1: Objective 6). The accurate specification of the model is required for a realistic estimation of energy consumption. In Section 6.1, the exact monitoring energy consumption of the model is computed.

4. Modeling & Mathematical Formulation

The exact mathematical model and the monitoring mechanism are described in this section and Section 5, respectively, while respecting the above requirements and objectives.

4.1. Decision Variables & Parameters

A graph can model the network topology. In tandem with RPL, the directed graph we use is the DODAG, $D = (V, E)$, where $V = \{v_i, i = 1, 2, \dots, n\}$ is the set of vertices representing the entire set of critical nodes, and E is the set of arcs. In a duty-cycled monitoring mechanism where a periodical functioning is assumed, a planning horizon of several periods is defined and represented by $T = \{T_j, j = 1, 2, \dots, m\}$. We formulate a Binary Integer Programming (BIP) model for the exact placement and scheduling of monitors across the planning horizon. The modeling terms are listed in Table 2 and explained as follows.

$xm_{i,j}$ is a binary Decision Variable (DV) that represents whether a node v_i is assigned to monitor in a period T_j (1). In the proposed centralized monitoring approach, monitors forward the monitoring data to their default parents in a path towards the DODAG root. A node routes monitoring packets if it is assigned to monitor, or if it is in the default route of an active-monitoring node in the same period. In the second case, the node is a *relay* node; for which we define another binary DV $xr_{i,j}$. It indicates whether v_i acts as a relay in period T_j (2).

For minimizing the cost of monitoring state transitions which are incurred as a result of the required duty cycle, the transition DV $y_{i,j}^a$ and $y_{i,j}^s$ are introduced to the mathematical model. $y_{i,j}^a$ identifies whether there is a monitoring state transition of v_i from sleep-monitoring in T_j into active in T_{j+1} (3); whereas $y_{i,j}^s$ denotes whether v_i is active-monitoring in T_j and asleep in T_{j+1} (4).

The modeling costs depend on the targeted monitoring mechanism. We opted to target passive monitoring, which implies that to verify their neighbors' availability, monitors need only observe the regular traffic passing through the radio channel. Therefore, the monitoring cost comes in the form of the energy consumed while monitors *listen* to the messages transmit-

Table 2. Glossary of Modeling Terms

Term	Description
V	Set of vertices, $V = \{v_i, i = 1, 2, \dots, n\}$
T	Periodical planning horizon, $T = \{T_j, j = 1, 2, \dots, m\}$
$xm_{i,j}$	Binary decision variable denotes whether v_i is assigned to monitor in T_j
$xr_{i,j}$	Binary decision variable denotes whether v_i acts as a relay in T_j
$y_{i,j}^a$	Binary decision variable denotes whether v_i changed its state from sleep in T_j into active-monitoring in T_{j+1}
$y_{i,j}^s$	Binary decision variable denotes whether v_i changed its state from active in T_j into sleep-monitoring in T_{j+1}
eM_i	Energy consumption of passive monitoring of neighbors
eC_i	Energy consumption of relaying messages to the preferred parent
$eActive$	Energy consumption of transitioning from sleep into active-monitoring state
$eSleep$	Energy consumption of transitioning from active into sleep-monitoring state
$reservedBattery_i$	Fraction of the available battery reserved for monitoring functions
$P(v_i)$	List of parents of v_i in the DODAG

ted by their neighbors. Depending on the total number of packets overheard from its set of neighbors N , a monitor v_i consumes an amount of energy expressed as eM_i during a period T_j . eC_i , on the other hand, stands for the amount of energy consumed by the relays while forwarding the monitoring data. The communication cost eC_i for each node v_i is a function of the number of times v_i forwards monitoring data through its default parent to the root. Finally, the transition costs are defined as $eActive$ and $eSleep$. They respectively denote the transition costs from sleep-monitoring to active and that from active-monitoring to asleep.

$$xm_{i,j} = \begin{cases} 1, & \text{if } v_i \text{ is assigned to monitor in period } T_j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$xr_{i,j} = \begin{cases} 1, & \text{if } v_i \text{ is a relay node in period } T_j \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$y_{i,j}^a = \begin{cases} 1, & \text{if } v_i \text{ is sleep-monitoring in period } T_j \\ & \text{and awake in period } T_{j+1} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

$$y_{i,j}^s = \begin{cases} 1, & \text{if } v_i \text{ is active-monitoring in period } T_j \\ & \text{and asleep in period } T_{j+1} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

It is worth mentioning that the transition decision variables are dependent on $xm_{i,j}$ and

$xm_{i,j+1}$; they are the product of the later two binary variables. Equations (5) and (6) represent the dependence relationship. Consequently, with the addition of $y_{i,j}^a$ and $y_{i,j}^s$, the mathematical model becomes a quadratic BIP model, which implies an expensive computational cost that goes against scalability objectives. However, we use standard techniques like the one mentioned in (Adamatzky, 2016) to convert the quadratic formulation into a linear one (described in Section 4.2).

$$y_{i,j}^a = (1 - xm_{i,j}) xm_{i,j+1} \quad (5)$$

$$y_{i,j}^s = xm_{i,j} (1 - xm_{i,j+1}) \quad (6)$$

It is assumed that each node has a reserved battery ($reservedBattery_i$) for the monitoring activity across the entire planning horizon. This monitoring energy is different from the energy dedicated to the thing's main function (Table 1: Assumption 3). Thus, it is critical to ensure that all sorts of energy consumption never exceed the reserved battery threshold. This threshold is assigned by the network operator, depending on the monitoring role's criticality relative to the primary function. Section 4.2 describes the BIP model for optimal monitor placement and scheduling problems.

4.2. Binary Integer Program

The objective of the proposed model is to find the exact placement of monitors in the DODAG, as well as the optimal scheduling of multiple sets of monitors across the planning horizon; while minimizing the total energy consumed for monitoring, transmitting the monitoring data to the root, and the monitoring state transitions (Table 1: Objective 4). Remember that the overriding goals are to ensure full monitor coverage, and a balanced, energy-efficient duty-cycling of monitoring and relaying roles across the node set (Table 1: Objectives 1), 3, 5), & 2). Eq. (7) represents the model's objective function.

The constraint stated in (8) guarantees that the energy consumed by each monitoring node at the end of the planning horizon must never exceed the ($reservedBattery_i$) for monitoring. Optimal monitor placement entails finding the *minimal* number of monitoring nodes placed on the graph while ensuring full monitor coverage. The coverage is forced in the optimal solution by including constraint (9); which states that each edge in the DODAG ($(v_i, v_k) \in E$) is incident to at least one active-monitoring node in the current period T_j .

There must be a connected path of active nodes from each monitor or relay node towards the root to ensure the successful transmission of the monitoring data from the monitors to the DODAG root. Constraints (10) and (11) are introduced to the model for this purpose. They state that in each period T_j , if v_i is currently monitoring or relaying, then at least one member of its set of parents, denoted as $P(v_i)$, is also active in the same period.

Linear constraints are augmented to the mathematical model to eliminate the computational cost coming from the quadratic product of the binary variables $xm_{i,j}$ and $xm_{i,j+1}$. Constraints (12) and (13) ensure that $y_{i,j}^a$ takes the value of zero if either $(1-xm_{i,j})$ or $xm_{i,j+1}$ are zero, while (14) makes sure that $y_{i,j}^a$ takes the value of one if both binary variables are set to 1.

Similar constraints, (15) - (17), are defined for $y_{i,j}^s$.

$$\begin{aligned} \min \quad & \sum_{j \in T} \sum_{i \in V} \left(eM_i xm_{i,j} + eC_i xr_{i,j} \right) \\ & + \sum_{j=1}^{m-1} \sum_{i \in V} \left(eActive y_{i,j}^a + eSleep y_{i,j}^s \right) \end{aligned} \quad (7)$$

$$\begin{aligned} s.t. \quad & \sum_{j \in T} \left(eM_i xm_{i,j} + eC_i xr_{i,j} \right) \\ & + \sum_{j=1}^{m-1} \left(eActive y_{i,j}^a + eSleep y_{i,j}^s \right) \\ & \leq reservedBattery_i \quad \forall i \in V \end{aligned} \quad (8)$$

$$xm_{v_i,j} + xm_{v_k,j} > 0 \quad \forall T_j \in T \text{ and } \forall \{v_i, v_k\} \in E \quad (9)$$

$$xm_{v_i,j} \leq \sum_{v_k \in P(v_i)} xm_{v_k,j} + xr_{v_k,j} \quad \forall v_i \in V \text{ and } \forall T_j \in T \quad (10)$$

$$xr_{v_i,j} \leq \sum_{v_k \in xr_{i,j}} xm_{v_k,j} + xr_{v_k,j} \quad \forall v_i \in V \text{ and } \forall T_j \in T \quad (11)$$

$$y_{i,j}^a \leq 1 - xm_{i,j} \quad (12)$$

$$y_{i,j}^a \leq xm_{i,j+1} \quad (13)$$

$$y_{i,j}^a \geq xm_{i,j+1} - xm_{i,j} \quad (14)$$

$$y_{i,j}^s \leq xm_{i,j} \quad (15)$$

$$y_{i,j}^s \leq 1 - xm_{i,j} \quad (16)$$

$$y_{i,j}^s \geq xm_{i,j} - xm_{i,j+1} \quad (17)$$

$$xm_{i,j}, xr_{i,j}, y_{i,j}^a, y_{i,j}^s \in \{0, 1\} \forall v_i \in V \text{ and } \forall T_j \in T \quad (18)$$

5. Optimal Proactive Passive Monitoring Mechanism

5.1. Centralized Passive Monitoring

This section presents how to implement and use the proposed mathematical program to perform a schedule of passive monitoring in relatively small IoT domains. Targeting full interoperability with 6LoWPAN-based IoT networks, this section highlights the relations between the mathematical model and the implemented network monitoring mechanism. The BIP model is implemented using Julia, a high-performance dynamic programming language for numerical computing (Bezanson, Edelman, Karpinski, & Shah, 2017) and solved using Gurobi solver (Gurobi Optimization, 2018); which is a powerful mathematical programming solver integrated into JuMP (Dunning, Huchette, & Lubin, 2017). The latter is a modeling language for mathematical programming that extends Julia. Algorithm 1 describes the procedure `SOLVE_EXACT_MODEL`; which takes as arguments the DODAG representing the topology of the mission-critical network, the formulated BIP (*cf.* Section 4.2), and the model's parameters (*cf.* Section 4.1). Gurobi solves the BIP and the optimal solution is returned to the calling procedure; `SOLVE_EXACT_MODEL` (step 5.2). The following matrices represent the solver's output, the dimensions of each are the number of nodes \times number of periods:

- (1) *optimal_monitors_solutions*,
- (2) *optimal_relays_solutions*,
- (3) *optimal_trans_to_sleep*, and
- (4) *optimal_trans_to_active*

`SOLVE_EXACT_MODEL` interprets the output of the solved mathematical model and translates it into the required *exact_monitoring_schedule* and *exact_relay_schedule* (steps 5.5 & 5.9); which are returned to `EXACT_MONITORING` (Algorithm 2) at step 5.13.

Procedure `EXACT_MONITORING` (Algorithm 2) is the interface between the mathematical formulation and the monitoring mechanism. The procedure is functioning during the length of the monitoring timeline, represented by *timeline_timer*. Initially, the optimal duty cycle of monitors is obtained by calling the procedure `SOLVE_EXACT_MODEL` (Algorithm 1); which returns the *exact_monitor_schedule* (step 6.1). Next, following a passive monitoring approach, the neighbors' availability of each *monitor* is verified via calling the procedure the `LISTEN_TO_NEIGHBORS` (Algorithm 3); which returns the set of *unreachable_neighbors* detected during the period it is assigned to (step 6.4). `LISTEN_TO_NEIGHBORS` (Algorithm 3) works as follows: during the period it is assigned to, given by *period_length*, the *monitor* checks whether each of its *neighbors* had participated in the radio transmission (step 7.4). Note that the set of neighbors (parents, children, and siblings) is known from the DODAG, which is passed as a parameter to the calling procedure. If there is a *neighbor* from which the *monitor* has not received any messages throughout the entire period, its address is appended to the list of *unreachable_neighbors* (step 7.5). After the expiry of the *period_timer*, the list of *unreachable_neighbors* is returned to the

Algorithm 1 PROCEDURE SOLVE_EXACT_MODEL

Input: *BIP, DODAG, model_parameters***Output:** *exact_monitor_schedule, exact_relay_schedule*

```
begin
5.1 exact_monitor_schedule  $\leftarrow$  exact_relay_schedule  $\leftarrow$   $\emptyset$ ;
5.2 optimal_monitors_solution, optimal_relays_solution  $\leftarrow$  GUROBI_SOLVER
   (BIP, DODAG, model_parameters);
5.3 forEach period  $\in$  monitoring_timeline do
5.4   forEach node  $\in$  DODAG do
5.5     if optimal_monitors_solution[node, period] == 1 do
5.6       exact_monitor_schedule  $\leftarrow$  exact_monitor_schedule  $\cup$  node;
5.7     end if
5.8     if optimal_relays_solution[node, period] == 1 do
5.9       exact_relays_schedule  $\leftarrow$  exact_relays_schedule  $\cup$  node;
5.10    end if
5.11  end forEach
5.12 end forEach
5.13 return exact_monitor_schedule, exact_relay_schedule;
end
```

procedure EXACT_MONITORING (Algorithm 2).

In the centralized monitoring approach, the list of *unreachable_neighbors* of each *monitor* is forwarded to its default parent (step 6.6); which is responsible for relaying that message to its default parent, and so forth until it reaches a central entity, the 6BR or the DODAG root. The 6BR has unconstrained resources and a global view of the network state, enabling it to perform powerful analysis of the monitoring data and further corrective measures. On the other hand, leveraging RPL's repair mechanism, an attempt at a fast recovery of the DODAG is made by the *monitor* via calling LOCAL_REPAIR (step 6.7) (Gaddour & Koubâa, 2012).

It is noteworthy that the *period_length* should be carefully chosen, such that it is neither too short nor too long. Too short *period_length* may result in false alarms. A false positive alarm occurs when a monitoring mechanism reports as fault a state that is legitimate network activity (Chen et al., 2016), whereas failure to detect a faulty state is termed as a false negative alarm. On the other hand, too long *period_length* will unnecessarily exhaust the energy of monitors as they are awake-monitoring for quite a long time. Fortunately, some studies focus on the optimal period length, such as the one in (Bergmann, Molnár, Gönczy, & Cousin, 2010).

5.2. Separate DODAG for Routing

Even though the centralized approach allows for sophisticated monitoring tasks, which might otherwise exhaust the resources of the things, they are achieved at the expense of high communication overhead. To take the burden of routing the monitoring data to the DODAG root off the constrained nodes, we leverage the multiple instance feature of RPL and create another DODAG. It is considered as an overlay structure within which the monitors communicate separately. The separate DODAG may consist of only monitoring nodes, and the monitoring data is transmitted from monitors to the 6BR through the shortest paths. Those routes may not have been defined by application-specific routing.

However, according to the network topology, the separate instance of monitors might not

Algorithm 2 PROCEDURE EXACT_MONITORING

Input: $BIP, DODAG, model_parameters$ **Output:** $unreachable_neighbors \quad \forall monitor \in exact_monitoring_schedule$

```
begin
6.1  $exact\_monitoring\_schedule \leftarrow$ 
    SOLVE_EXACT_MODEL( $BIP, DODAG, model\_parameters$ );
6.2 while  $timer < timeline\_timer$  do
6.3   forEach  $monitor \in monitoring\_schedule$  do
6.4      $unreachable\_neighbors \leftarrow$ 
        LISTEN_TO_NEIGHBORS( $DODAG, period\_length$ );
6.5     forEach  $neighbor \in unreachable\_neighbors$  do
6.6       FORWARD_TO_PARENT( $unreachable\_neighbors$ );
6.7       LOCAL_REPAIR( $DODAG$ );
6.8     end forEach
6.9   end forEach
6.10 end while
end
```

Algorithm 3 PROCEDURE LISTEN_TO_NEIGHBORS

Input: $DODAG, period_length$ **Output:** $unreachable_neighbors$

```
begin
7.1  $unreachable\_neighbors \leftarrow \emptyset$ ;
7.2 while  $timer < period\_timer$  do
7.3   forEach  $neighbor \in DODAG$  do
7.4     if! RECEIVE_MESSAGE( $neighbor$ ) do
7.5        $unreachable\_neighbors \leftarrow unreachable\_neighbors \cup neighbor$ ;
7.6     end if
7.7   end forEach
7.8 end while
7.9 return  $unreachable\_neighbors$ ;
end
```

guarantee the presence of a connected path between themselves and the DODAG root. Therefore, for the successful transmission of monitoring data, *relay* nodes may be required. The optimal placement of the required relays is obtained from the *exact_relay_schedule* given by procedure SOLVE_EXACT_MODEL (Algorithm 1). Even though it is probable that relay nodes are not currently involved in monitoring, they still share the monitoring burden as there is an amount of energy consumed by forwarding the monitoring data. This cost is represented in the model's objective function by eC (cf. Section 4.2).

6. Experimental Evaluation

Extensive computational experiments are conducted using different network sizes and topologies to verify and validate the BIP model. The experiments are designed to test the model's ability to optimally place monitors and relays on the DODAGs representing the nodes associated with the critical mission, and to find answers to the following questions:

- How long does it take to find the optimal schedule of monitors?
- How much energy is consumed for monitoring the critical set of nodes?
- Can the model operate smoothly under challenging battery conditions?
- Is the overhead of the communication between monitors and the central node significant?
- Does the model scale well?
- Is it beneficial to leverage the separate DODAG feature of RPL?

6.1. Parameter Settings

Before commencing the experiments, the parameters must be carefully chosen to reflect a real-life mission-critical IoT network. This section explains how energy costs are estimated. The parameters' settings are displayed in Table 3. The amount of energy consumption is platform-

Table 3. Default Values of Physical Network Parameters

Parameter	Description	Default Value
e_{listen}	Energy cost of listening to the radio channel	0.58 <i>mJ</i>
e_{CCA}	Energy cost of Clear Channel Assessment	0.08 <i>mJ</i>
e_{send}	Radio energy consumption of sending 1 byte	0.0020 <i>mJ</i>
$e_{receive}$	Radio energy consumption of receiving 1 byte	0.0022 <i>mJ</i>
s_{msg}	Size of the data packet	19 <i>B</i>
s_{ack}	Size of acknowledgment packet	11 <i>B</i>

dependent. TelosB, also known as the TMote Sky motes, is selected as a target platform for regular and monitoring nodes; its computational resources allow it to function as an RPL router node within the Contiki RPL implementation (Mayzaud et al., 2016). Tmote Sky motes use 2 AA batteries with a total available energy of 30780 Joules. Referring to Tmote Sky's datasheet (Datasheet, 2006), its measurements in different modes of operation are shown in Table 4. While monitors passively listen to their neighbors, they consume a significant

Table 4. Tmote Sky Measurements in Typical Operating Conditions (Datasheet, 2006)

Parameter	Default Value
Maximum supply voltage	3.6 <i>V</i>
Radio transmitting current consumption	17.4 <i>mA</i>
Radio receiving current consumption	19.7 <i>mA</i>
Radio on current consumption	365 μA
Power down current consumption	1 μA
Maximum startup time	860 μs
Maximum supply energy	30780 <i>J</i>
Transmit bit rate	250 <i>kbps</i>

amount of energy, which can be expressed as e_{listen} . The total monitoring cost, eM_i (19), is

a function of the listening cost, the cardinality of a monitor's set of neighbors $|N|$, and the number and size of the sniffed packets. Based on the framework of 6LoWPAN (Kushalnagar et al., 2007), the size of the data packet, hereby denoted s_{msg} , is 17 bytes for the header frame in addition to the size of the data payload. Assuming that the payload is 2 bytes, s_{msg} is 19 bytes. Acknowledgment packets are not mandatory in IEEE 802.15.4. Nevertheless, since the target is reliable communication, acknowledgment costs are considered in our computations. 6LoWPAN dedicated 11 bytes to the acknowledgment packet, which is expressed here as s_{ack} .

To compute the communication cost, eC_i (20) for IEEE 802.15.4 networks that employ Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), there is a fixed energy cost associated with the Clear Channel Assessment; symbolized as e_{CCA} . Moreover, it is necessary to estimate the costs related to the radio energy consumption of receiving and relaying messages, expressed as $e_{receive}$ and e_{send} , respectively.

Given Tmote Sky measurements in Table 3, we compute the costs of receiving eM_i and relaying one message of size 19 bytes eC_i using Equations 19 and 20, respectively. Accordingly, the default values of the BIP model are obtained and given in Table 5.

Table 5. Default Values of Model Parameters

Parameter	Default Value
eM_i	0.621 mJ
eC_i	0.486 mJ
$eActive$	0.0011 mJ
$eSleep$	0.02 μ J
$reservedBattery_i$	50 mJ
T	20

$$eM = e_{listen} + e_{receive} \times s_{msg} \quad (19)$$

$$eC = e_{CCA} + e_{send} \times s_{msg} + e_{receive} \times s_{ack} \quad (20)$$

6.2. Results & Discussion

Experiments are performed on a personal computer with 16 Gigabytes of RAM and 2.20 Gigahertz Intel Core i7 processor. The proposed model is tested using 21 instances, some of which are of random network topology with variable densities, while the rest are of famous networks, namely KARATE (Zachary, 1977), DOLPHINS (Lussseau et al., 2003), POL-BOOKS (Rossi & Ahmed, 2015), FOOTBALL (Girvan & Newman, 2002), POWER (Watts & Strogatz, 1998), and NETSCIENCE (Newman, 2006). Results are displayed in Table 6, where the values of the following metrics are recorded:

- number of nodes ($|V|$), links ($|E|$), and graph's density (ρ),
- percentage of monitors (**monitor(%)**),
- average number of neighbors per monitor ($|N|$),
- average energy consumption per node in monitoring, relaying, and state transitions (**Energy cons. (mJ)**),

Table 6. Experimental Results of Exact Monitor Placement & Scheduling

Instance	Topology	V	E	ρ	monitor(%)	N	Energy cons.(mJ)	battery cons.(%)	msgs	time (sec)
1	Random	25	150	0.250	80	13	9.936	19.872	5	109.66
2	Random	25	180	0.300	76	16	9.439	18.878	5	7.20
3	Random	25	210	0.350	84	18	10.432	20.865	6	28.82
4	Random	25	240	0.400	84	21	10.432	20.865	5	8.30
5	Random	25	270	0.450	88	23	10.929	21.859	6	8.75
6	Random	25	300	0.500	96	25	11.923	23.846	5	8.46
7	Random	27	330	0.550	88	26	11.040	22.080	5	56.76
8	Random	30	130	0.150	66	11	8.280	16.560	6	59.73
9	Random	30	174	0.200	77	13	9.522	19.044	5	153.82
10	KARATE	34	78	0.101	41	5	5.607	11.214	5	66.90
11	Random	34	470	0.430	87	29	10.914	21.829	5	49.57
12	DOLPHINS	62	159	0.024	56	5	7.013	14.026	5	175.30
13	DOLPHINS*	62	207	0.054	58	9	7.211	14.850	7	104.95
14	POLBOOKS	105	441	0.040	60	8	7.570	15.430	6	13.61
15	FOOTBALL	115	613	0.047	82	11	10.152	20.304	5	78.55
16	POWER	200	209	0.005	44	2	5.437	10.875	5	51.30
17	POLBOOKS*	399	950	0.005	29	6	3.673	7.346	6	15.00
18	Complete Graph	500	249500	0.990	99	499	12.395	24.790	6	276.60
19	Random	600	179700	0.500	99	599	12.3993	24.798	5	466.90
20	NETSCIENCE	1589	4331	0.002	57	6	45.005	90.011	5	2.82
21	POWER*	4941	11535	0.0005	47	5	5.799	11.598	5	111.24

- percentage of battery consumed for monitoring, relaying and state transitioning from *reservedBattery* per node (**Battery consns. (%)**),
- average number of overheard packets per monitor (**msgs**), and
- model execution time in seconds (**time**).

Initially, the model was tested on a relatively sparse topology, that of the KARATE (Zachary, 1977) network of 34 nodes and 114 links. Fig. 1 pinpoints the optimal placement of three types of monitors; specifically, the ones that: (1) transitioned from active-monitoring in the previous period T_{j-1} to sleep in T_j , (2) transitioned from sleep-monitoring to active in said periods, and (3) the ones that did not transition at all, *i.e.*, they are monitoring in both periods T_{j-1} and T_j . Node labeled 1 represents the 6BR, and the rest of the vertices correspond to the monitored nodes.

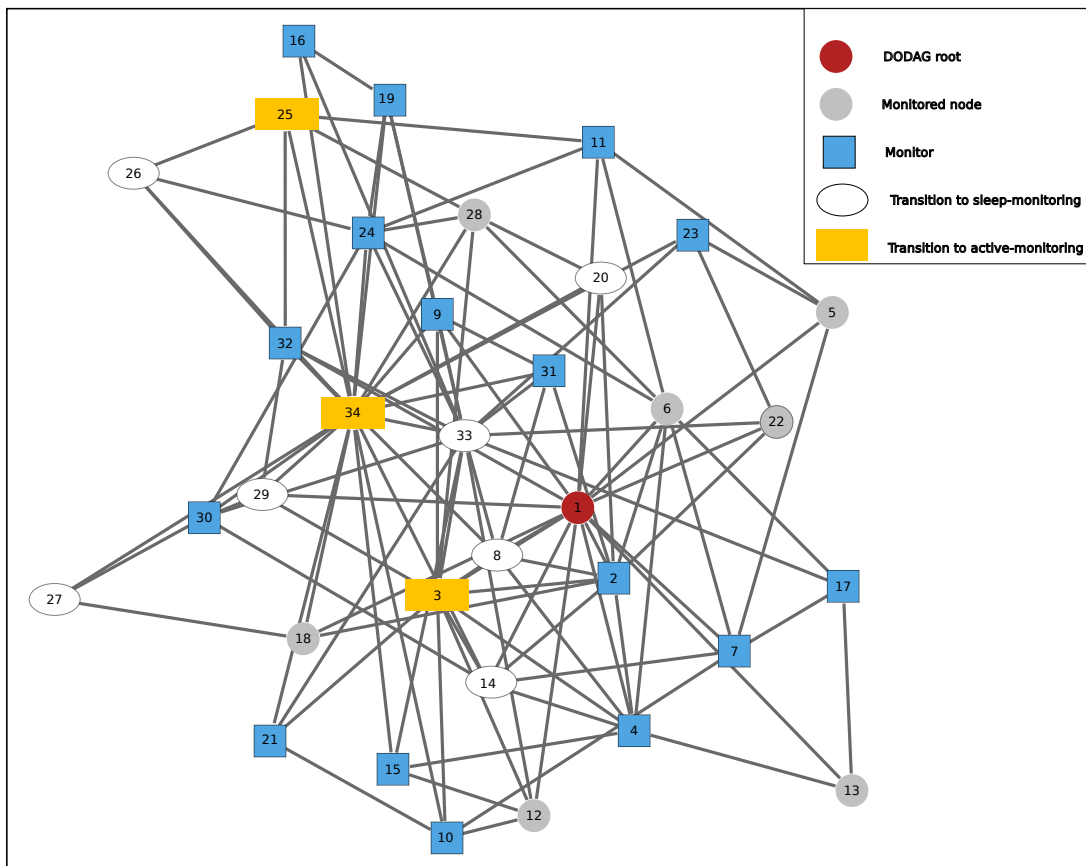
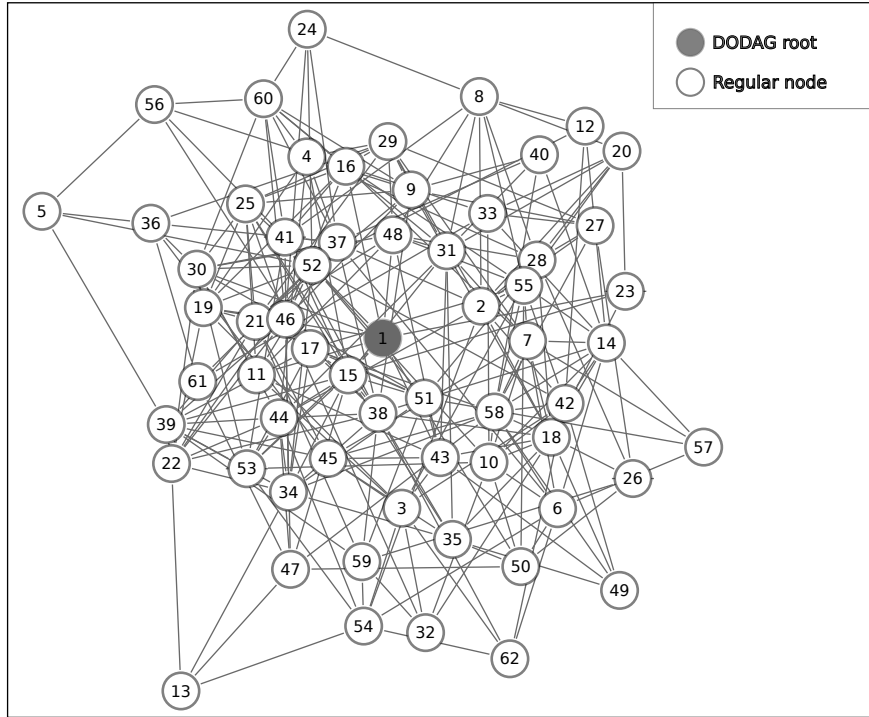


Figure 1. Optimal solution in a given period for the KARATE (Zachary, 1977) network topology with 34 Vertices and 114 edges. *reservedBattery_i* = 1641.6 mJ (0.05 % of total power).

Next, we experimented with a slightly denser network, the DOLPHINS (Lusseau et al., 2003) topology, consisting of 62 nodes and 159 edges; node 1 represents the 6BR. Here, the multiple instance feature of RPL is exploited. A separate DODAG is created to reduce the

communication overhead of routing the monitoring data to the 6BR and verify the model’s placement of the relaying nodes if required. As mentioned in Section 5.2, the optimal placement of the required relays is obtained from the *exact_relay_schedule* given by procedure SOLVE_EXACT_MODEL (Algorithm 1).

Fig. 2a shows the original DOLPHINS topology. Creating a separate instance consisting of monitors only does not provide a connected path between the monitors and the 6BR (except for monitors 14 and 61). Fig. 5 displays the required relay nodes for an arbitrary period, T_j , corresponding to the values of $xr_{i,j}$ in the optimal solution of the proposed mathematical model.



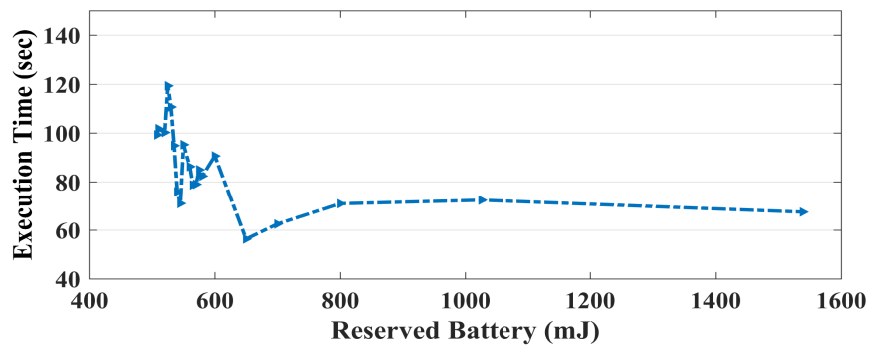
(a)

Figure 2. The DOLPHINS topology (Lusseau et al., 2003) (62 vertices and 159 edges).

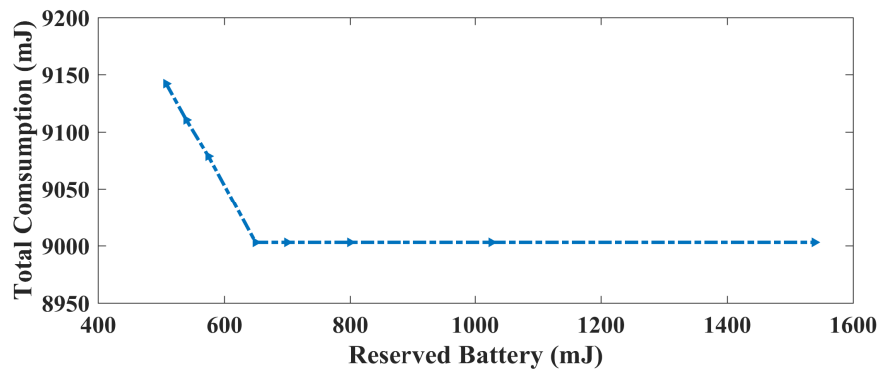
Furthermore, we tested the model’s response towards variations in the Right Hand Side (RHS) vector of the *reservedBattery_i* for the monitoring constraint (Section 4.2 - (8)). Using the same KARATE topology depicted in Fig. 1, we run 14 trials while varying the node’s *reservedBattery_i* in the range between 1539 - 307800 *mJ*; which corresponds to 0.05% - 10% of Tmote Sky total power. This particular range is selected as the nodes are incapable of monitoring the network for the length of the planning horizon on a *reservedBattery_i* below the range’s lower bound, and the model converges to the same solution with the upper bound and beyond. The results of these trials are displayed in figures 3a, 3b, and 4.

Fig. 3a shows a clear trend; the more the *reservedBattery_i* constraint is tightened, the more is the execution time. Fig. 3b, on the other hand, shows the significant effect of the size of the *reservedBattery_i* on the total energy consumption. The interpretation is that the tighter the battery constraint, the more is the number of monitoring state transitions since a node does not have enough power to continue monitoring for several periods, thus goes to sleep.

A remarkable conclusion emerges from the experimental results: tightening the



(a)



(b)

Figure 3. Effect of variation of $reservedBattery_i$ for the KARATE (Zachary, 1977) topology; $T = 20$; (a) $reservedBattery_i$ versus model execution time; (b) $reservedBattery_i$ versus network total energy consumption.

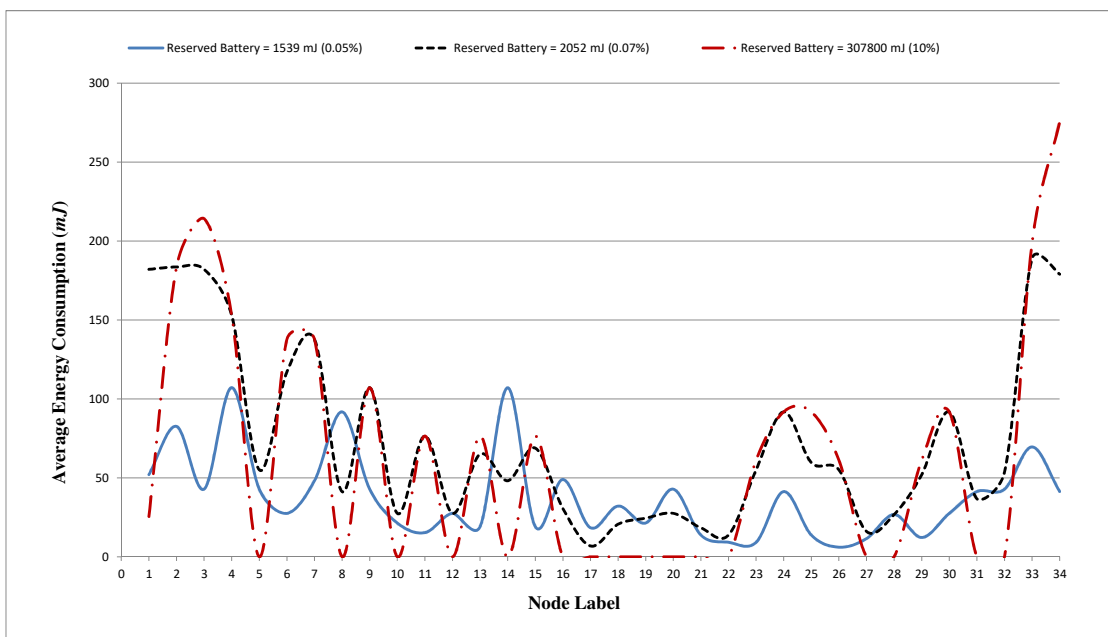


Figure 4. Average energy consumption per node for the KARATE (Zachary, 1977) topology; $reservedBattery_i = 307800$ mJ , 2052 mJ and 1641.6 mJ ; corresponding to 10 %, 0.07% and 0.05% of Tmote Sky total power, respectively; $T = 20$.

$reservedBattery_i$ constraint has the advantage of balancing the monitoring role among nodes. Consequently, the average energy consumption per node decreases. This result is depicted in Fig. 4; which shows the average energy consumption per node after 20 periods as the $reservedBattery_i$ is varied between 307800 mJ , 2052 mJ , and 1641.6 mJ ; corresponding to 10 %, 0.07% and 0.05% of the total power of Tmote Sky, respectively. This finding highlights the fact that the size of $reservedBattery_i$ should be set with considerable care.

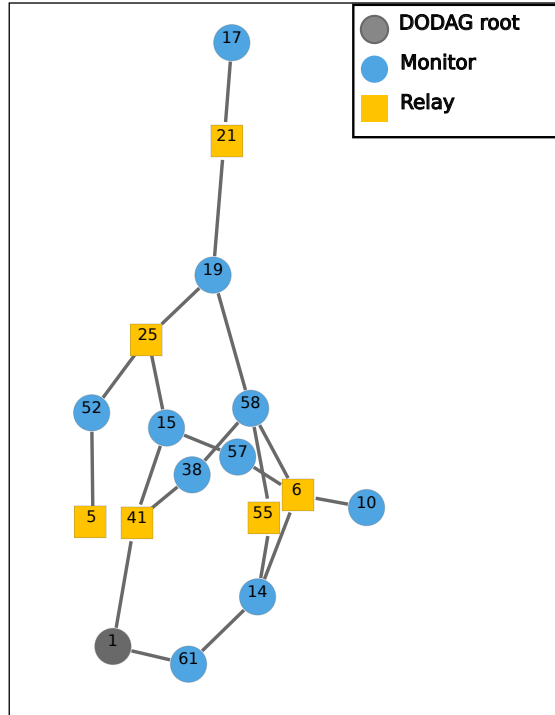


Figure 5. Monitoring DODAG with the necessary relays for the DOLPHINS Lusseau et al. (2003) topology of 62 Vertices and 159 edges. Node labeled 1 is the root.

Fig. 4 also emphasizes the efficiency of the proposed passive monitoring mechanism, since the average energy consumption per node does not exceed 300 mJ (approximately 0.01% of its total power); regardless of the size of $reservedBattery_i$ for monitoring. It can be seen in Table 6 and Fig. 6, that the execution time for a dense network of 600 nodes and 17970 links is approximately 8.5 minutes, which is not exceedingly expensive. However, it is a fact that computing the exact solution to the BIP model for large-sized networks will be computationally expensive, a result of the NP-hardness of the MVC problem, which is represented in constraint 9 (Beame, Impagliazzo, & Sabharwal, 2007). Nevertheless, it is worth mentioning that the model's performance was tested under a tight battery constraint, with a $reservedBattery_i$ 50 mJ from a total available power of 30780 *Joules* of the TMote Sky mote.

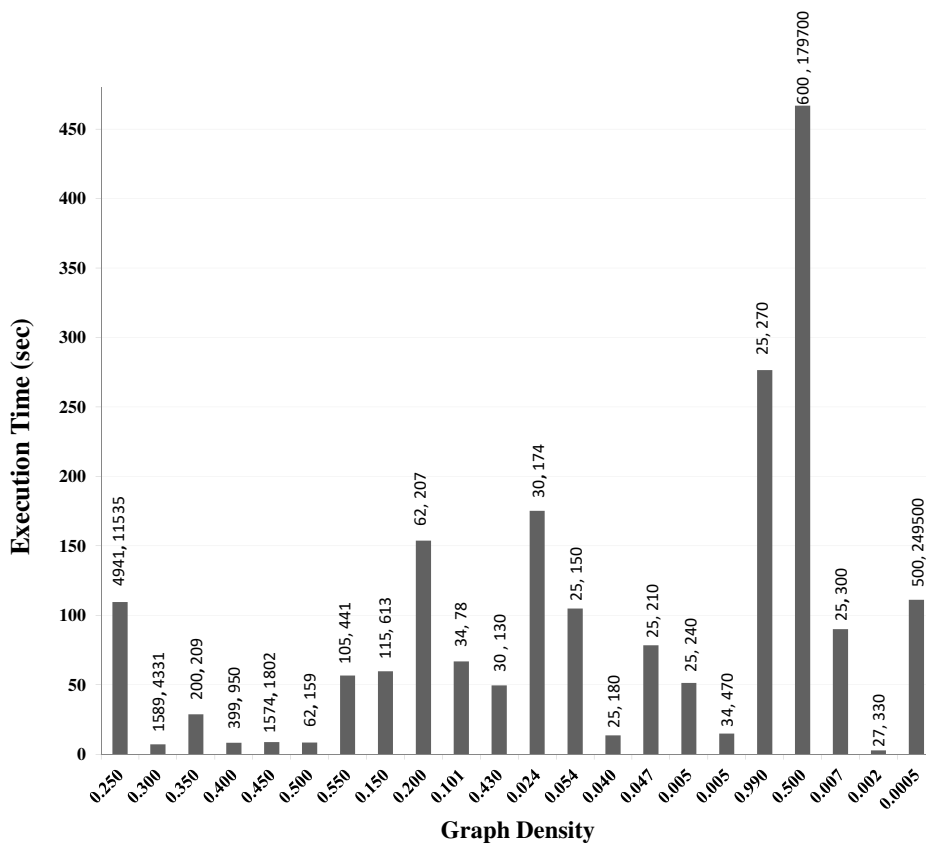


Figure 6. Effect of varying network density on the execution time. Data labels are the number of nodes and links.

7. Conclusions & Perspectives

Our study presents the exact solution to the minimum monitor assignment problem with a duty-cycled monitoring approach. The optimal schedule guarantees monitor coverage with minimum energy consumption. The solution is incorporated into a centralized, passive monitoring mechanism that is interoperable with RPL and 6LoWPAN protocols.

The overall findings confirm the proposed model's effectiveness in achieving full monitor coverage with minimum energy consumption in all tested network topologies. The results also confirm that the execution times are tolerable even for relatively large or dense networks. These conclusions are crucial for the adoption of network monitoring into critical-mission IoT networks.

Even though the optimization is computed off-line, for benchmarking, we tested how scalable the model is, concerning the size and density of networks. We have seen that the execution time for relatively dense networks is not very expensive. However, it is a fact that computing the exact solution to the BIP model for large-sized networks will be computationally expensive, a result of the NP-hardness of the MVC problem. This fact implies an exponential lower bound on the running time of solving the proposed linear BIP model. Despite the computational limitations, the proposed mechanism can serve as a benchmark for comparisons and performance evaluation of contemporary models, which has been missing from the literature.

The proposed models represent a static view of the IoT network, rendering the optimal solution unrepresentative to a new situation if a significant change in the network topology is detected. Dynamic models are required to avoid a complete re-optimization of the problem, which is the current study.

References

- Adamatzky, A. (2016). *Advances in Unconventional Computing: Volume 1: Theory* (Vol. 22). Springer.
- Alaba, F. A., Othman, M., Hashem, I. A. T., & Alotaibi, F. (2017). Internet of Things security: A survey. *Journal of Network and Computer Applications*, 88, 10–28.
- Beame, P., Impagliazzo, R., & Sabharwal, A. (2007). The resolution complexity of independent sets and vertex covers in random graphs. *computational complexity*, 16(3), 245–297.
- Bergmann, G., Molnár, M., Gönczy, L., & Cousin, B. (2010). Optimal period length for the cgs sensor network scheduling algorithm. *2010 Sixth International Conference on Networking and Services*, 192–199.
- Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, 59(1), 65–98.
- Chen, Z., Xu, G., Mahalingam, V., Ge, L., Nguyen, J., Yu, W., & Lu, C. (2016). A cloud computing based network monitoring and threat detection system for critical infrastructures. *Big Data Research*, 3, 10–23.
- Datasheet, T. S. (2006). Moteiv corporation. *Saatavissa (viitattu 1.10. 2017): <http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>*.
- Dhall, R., & Solanki, V. (2017). An iot based predictive connected car maintenance. *International Journal of Interactive Multimedia & Artificial Intelligence*, 4(3).
- Dunning, I., Huchette, J., & Lubin, M. (2017). Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2), 295–320.
- Gaddour, O., & Koubâa, A. (2012). RPL in a nutshell: A survey. *Computer Networks*, 56(14), 3163–3178.
- Geng, H. (2017). *Internet of things and data analytics handbook*. John Wiley & Sons.
- Girvan, M., & Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12), 7821–7826. (dataset: FOOTBALL)
- Gurobi Optimization, L. (2018). *Gurobi Optimizer Reference Manual*. Retrieved from <http://www.gurobi.com>
- Hassanalieragh, M., Page, A., Soyata, T., Sharma, G., Aktas, M., Mateos, G., . . . Andreescu, S. (2015). Health monitoring and management using internet-of-things (iot) sensing with cloud-based processing: Opportunities and challenges. *2015 IEEE International Conference on Services Computing*, 285–292.
- Jara, A. J., Ladid, L., & Gómez-Skarmeta, A. F. (2013). The internet of everything through ipv6: An analysis of challenges, solutions and opportunities. *JoWua*, 4(3), 97–118.
- Kiani, F. (2018). A survey on management frameworks and open challenges in iot. *Wireless Communications and Mobile Computing*, 2018.
- Korte, K. D., Sehgal, A., & Schönwälder, J. (2012). A study of the RPL repair process using ContikiRPL. *IFIP International Conference on Autonomous Infrastructure, Management and Security*, 50–61.
- Kuang, X., & Shen, J. (2010). SDNS: A distributed monitoring and protocol analysis system for wireless sensor network. *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing*, 2, 422–425.
- Kumar, R., & Kaur, J. (2004). Efficient beacon placement for network tomography. In *Proceedings of the 4th acm sigcomm conference on internet measurement* (pp. 181–186).
- Kushalnagar, N., Montenegro, G., & Schumacher, C. (2007). *IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals* (Tech. Rep.).
- Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., & Zhao, W. (2017). A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal*, 4(5), 1125–1142.
- Liu, X., Gao, Y., Wu, W., Dong, W., & Bu, J. (2015). Robust monitor assignment with minimum cost for sensor network tomography. *International Journal of Distributed Sensor Networks*, 11(8), 512463.
- Lussseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E., & Dawson, S. M. (2003). The

- bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4), 396–405. (dataset: DOLPHIN)
- Ma, L., He, T., Swami, A., Towsley, D., & Leung, K. K. (2015). On optimal monitor placement for localizing node failures via network tomography. *Performance Evaluation*, 91, 16–37.
- Mayzaud, A., Sehgal, A., Badonnel, R., Chrisment, I., & Schönwälder, J. (2016). Using the RPL protocol for supporting passive monitoring in the Internet of Things. *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, 366–374.
- Metcalf, D., Milliard, S. T., Gomez, M., & Schwartz, M. (2016). Wearables and the internet of things for health: Wearable, interconnected devices promise more efficient and comprehensive health care. *IEEE pulse*, 7(5), 35–39.
- Montenegro, G., Kushalnagar, N., Hui, J., & Culler, D. (2007). *Transmission of IPv6 packets over IEEE 802.15. 4 networks* (Tech. Rep.).
- Mostafa, B., Benslimane, A., Saleh, M., Kassem, S., & Molnar, M. (2018, June). An Energy-Efficient Multiobjective Scheduling Model for Monitoring in Internet of Things. *IEEE Internet of Things Journal*, 5(3), 1727–1738.
- Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3), 036104. (dataset: NETSCIENCE)
- Rossi, R. A., & Ahmed, N. K. (2015). The Network Data Repository with Interactive Graph Analytics and Visualization. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. Retrieved from <http://networkrepository.com> (dataset: POLBOOKS)
- Sahoo, P., Thakkar, H., Hwang, I., et al. (2017). Pre-scheduled and self organized sleep-scheduling algorithms for efficient k-coverage in wireless sensor networks. *Sensors*, 17(12), 2945.
- Sehgal, A., Perelman, V., Kuryla, S., & Schönwälder, J. (2012). Management of resource constrained devices in the Internet of Things. *IEEE Communications Magazine*, 50.
- Sheng, Z., Yang, S., Yu, Y., Vasilakos, A., Mccann, J., & Leung, K. (2013). A survey on the IETF protocol suite for the Internet of Things: Standards, challenges, and opportunities. *IEEE Wireless Communications*, 20(6), 91–98.
- Stanic, S., Subramaniam, S., Sahin, G., Choi, H., & Choi, H.-A. (2010). Active monitoring and alarm management for fault localization in transparent all-optical networks. *IEEE Transactions on Network and Service Management*, 7(2), 118–131.
- Stanisavljević, M., Schmid, A., & Leblebici, Y. (2010). *Reliability of nanoscale circuits and systems: methodologies and circuit architectures*. Springer Science & Business Media.
- Suriyachai, P., Roedig, U., & Scott, A. (2011). A survey of mac protocols for mission-critical applications in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 14(2), 240–264.
- Türkoğulları, Y., Aras, N., Altınel, İ. K., & Ersoy, C. (2010). Optimal placement, scheduling, and routing to maximize lifetime in sensor networks. *Journal of the Operational Research Society*, 61(6), 1000–1012.
- Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of "small-world" networks. *nature*, 393(6684), 440. (dataset: POWER)
- Wu, F., Wu, T., & Yuce, M. (2019). An internet-of-things (iot) network system for connected safety and health monitoring applications. *Sensors*, 19(1), 21.
- Yücel, T., & Altın-Kayhan, A. (2019). A copy-at-neighbouring-node retransmission strategy for improved wireless sensor network lifetime and reliability. *Journal of the Operational Research Society*, 70(7), 1193–1202.
- Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4), 452–473. (dataset: KARATE)
- Zarpelão, B. B., Miani, R. S., Kawakani, C. T., & de Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications*, 84, 25–37.
- Zhao, Z., Huangfu, W., & Sun, L. (2012). Nssn: A network monitoring and packet sniffing tool for wireless sensor networks. *012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 537–542.



Basma Mostafa received a dual Ph.D. degree in Computer Science in 2019 from the University of Montpellier, Montpellier, France, and the Faculty of Computers and Artificial Intelligence, Cairo University, Cairo, Egypt. She received the B.S. and M.S. degrees in Operations Research from the Faculty of Computers and Artificial Intelligence, Cairo University in 2008 and 2013, respectively, where she is currently a Lecturer in the Department of Operations Research. Dr. Mostafa was awarded the 2017 "L'Oréal-UNESCO For Women In Science Levant and Egypt" for her research work on developing optimized models for monitoring IoT networks. Her research activities focus mainly on combinatorial optimization, network optimization, linear and integer programming, operations research methodologies, modeling, and simulation.



Miklos Molnar received the graduation degree from the Faculty of Electrical Engineering, University BME, Budapest, Hungary, in 1976, the Ph.D. degree in computer science from the University of Rennes 1, Rennes, France, in 1992, and the French HDR degree in 2008. He has been with the University Montpellier, Montpellier, France, since 2010. He is a Full Professor with the Department of Computer Science, IUT. His research activities are in combinatorial optimization, network design, and optimization algorithms and tools in the laboratory LIRMM of Montpellier. He conducted several studies to find dependable routes for sensible communications, efficient multicast routes, energy-aware k-coverage, and routing protocols, and different optimizations in ad hoc wireless networks. Dr. Molnar participates as a PC member in the organization of conferences and on the Editorial Board of several journals.



Mohamed Saleh received the master's degree from Bergen University, Bergen, Norway, the M.B.A. degree from the Maastricht School of Management, Maastricht, The Netherlands, and the Ph.D. degree in system dynamics from the University of Bergen, Bergen. He is a Professor and the former Head with the Faculty of Computers and Information, Department of Operations Research and Decision Support, Cairo University, Cairo, Egypt. He is also an Adjunct Professor with the System Dynamics Group, University of Bergen. He has authored or co-authored numerous papers in several international journals and conferences. He is currently the Manager of the Virtual Center of Excellence for Data Mining and Computer Modeling, Cairo University. His current research interests include system dynamics, simulation, futures studies, revenue management optimization, and management science. Dr. Saleh was a recipient of the IBM Faculty Award.



Abderrahim Benslimane received the B.S. degree in computer science from the University of Nancy, France, in 1987, and the DEA (M.S.) and Ph.D. degrees in computer science from the Franche-Comte University, France, in 1989 and 1993, respectively. He has been a Full Professor of computer science with Avignon University, France, since 2001. He has been recently a Technical International Expert with the French Ministry of Foreign and European Affairs, from 2012 to 2016. He served as a Coordinator with the Faculty of Engineering, French University in Egypt. He has been an Associate Professor with the University of Technology of Belfort-Montbeliard, France, since 1994. He obtained the title to supervise research (HDR 2000) from the University of Cergy-Pontoise, France. He has been involved in many national and international projects, such as ANR, PHC, and H2020. His current research interests include the development of secure communication protocols for vehicular networks and the Internet of Things. He has over 140 refereed international publications, including books, journals, and conferences in the above areas. Dr. Benslimane was a recipient of the French Award of Sci-

entific Excellency from 2011 to 2014. He is an Area Editor of Wiley Security and Privacy Journal and an Editorial Board member of IEEE Wireless Communication Magazine and Elsevier Ad Hoc. He has been serving as the General-Chair of IEEE WiMob since 2008; he was launched and had been serving as the General-Chair of iCOST and MoWNet since 2011. He served as the Symposium Co-Chair/Leader in many IEEE international conferences, such as ICC, Globecom, AINA, and VTC. He was a Guest Editor of many special issues. He participates in the Steering and the Program Committee of many IEEE international conferences. He was a Board Committee member, the Vice-Chair of student activities of the IEEE France Section/Region 8. He was the Publication Vice-Chair, the Conference Vice-Chair, and is currently the Chair of the IEEE Communication Society TC of Communication and Information Security. He participates on the Steering and the Program Committees of many IEEE international conferences. He was a member of the French Council of Universities (CNU Section 27) from 2003 to 2007.

Sally Kassem received the graduation degree in 1998 and an M.Sc. degree in industrial engineering from the Faculty of Engineering, Mechanical Design and Production Department, Cairo University, Cairo, Egypt, and a Ph.D. degree in industrial engineering from Concordia University, Montreal, QC, Canada, in 2011. She has been an Assistant Professor with the Faculty of Computers and Information, Department of Operations Research and Decision Support, Cairo University since 2012. She is also an Assistant Professor with the School of Engineering and Applied Science, Industrial Engineering Program, Nile University, Giza, Egypt. Her current research interests include mathematical modeling and optimization, linear and integer programming, operations research methodologies and tools, supply chain, logistics, and modeling and simulation.