Dynamic Distributed Monitoring for 6LoWPAN-based IoT Networks

Basma Mostafa Hassan, Miklós Molnár, Mohamed Saleh, Abderrahim Benslimane, Sally Kassem

Dynamic Distributed Monitoring for 6LoWPAN-based IoT Networks

Basma Mostafa*† , Miklos Molnar† Mohamed Saleh* Abderrahim Benslimane‡ and Sally Kassem*§
*Faculty of Computers & Artificial Intelligence, Cairo University, Cairo, Egypt
†LIRMM, Université de Montpellier, Montpellier, France
‡LIA, Université d'Avignon, Avignon, France
§Smart Engineering Systems Center, Nile University, Cairo, Egypt

***Abstract*—Mission-critical Internet of Things (IoT)-based networks are increasingly employed in daily and industrial infrastructures. The resilience of such networks is crucial. Given IoT networks' constantly changing nature, it is necessary to provide dependability and sustainability. A robust network monitoring can reinforce reliability, such that the monitoring mechanism adapts itself to real-time network instabilities. This work proposes a *proactive, dynamic, and distributed* network monitoring mechanism with monitor placement and scheduling for 6LoWPAN-based IoT networks intended for mission-critical applications. The proposed mechanism aims to ensure real-time monitoring coverage while respecting the limited and changing power resources of devices to prolong the network lifetime.**

***Keywords*—IoT networks; Reliability; Monitor Scheduling; Dynamic; Proactive Monitoring; Critical Missions**

I. INTRODUCTION

The Internet of Things (IoT) is a global network and service infrastructure composed of heterogeneous things with identities and physical and virtual attributes and seamlessly integrated into the Internet [29]. The IoT aims to offer various services by enabling things to be connected anytime, anyplace, with anything and anyone, and ideally using any network infrastructure. By connecting billions of things to the Internet, IoT created a plethora of applications that touch every aspect of human life, to name but a few: wearables, smart homes, smart cities, smart grids, and connected cars. IoT is present mainly in manufacturing, production, system monitoring, automation, and also in the Industrial IoT (IIoT) (often referred to as Industry 4.0) [7].

An essential category of IoT networks in industrial applications is the Low-Power and Lossy Network (LLN). Following the indications in RFC 7102 [43]: an LLN is a network of embedded devices with limited power, memory, and processing resources. LLNs are typically optimized for energy efficiency. They may use IEEE 802.15.4, which can be applied in IIoT, building automation, connected homes, healthcare, environmental monitoring, urban sensor networks, asset tracking, and more. However, the use of LLN in critical systems is challenging.

IoT networks have self-configuring capabilities and should be based on standard and interoperable protocols to foster smart, sustainable, and inclusive IoT services and products [11]. Unfortunately, they are often characterized by several challenges. The tight energy, memory, and processing constraints of the things and unreliable radio communication are naturally added to the difficulties of node failures, long-term network instability, security, and resource-exhaustion attacks [19, 32].

For resource-constrained IoT entities, minimizing the energy consumed for communication and computing is a primary constraint [42]. Moreover, there is an aggravating need to devise solutions that optimize energy and enhance IoT sustainability, which recently became a hot research area [30, 11]. Hence, the motivation for investigating the development of detailed protocol (re)design and usage to reduce energy consumption during normal operation and under Denial of Service (DoS) attacks, especially for domains where network robustness and safety requirements are crucial [19, 18].

The following section, Section II, describes the motivations behind our research in response to the requirements of mission-critical IoT solutions and those of a consequent monitoring mechanism. Section III reviews the state of the art and mentions the gap that our research fills. Section IV describes the essential elements of the original CGS scheduling algorithm for area coverage, and then it presents our adaptation for realizing IoT network monitoring. The experimentation of the proposed mechanism is shown in Section V. Finally, Section VI summarizes the conclusions of our and the possible future research directions.

II. CHALLENGES & REQUIREMENTS

Several challenges are faced when deploying IoT solutions, (summarized in Table I).

Our research focuses on applications where the results are essential, and the mission has to be successful at any cost. Such applications are well-known as *mission-critical* since they deal with serious situations with high priorities for increased *reliability* and *network coverage*.

Examples of mission-critical applications are safety-oriented ones such as surveillance for safety and security applications [13]. IoT is ideal here since it can be successfully integrated within mission-critical systems deployed at locations where human presence is impossible due to human

Table I
IoT solutions' Deployment Challenges

Instability. Unreliable, lossy channels with unpredictable bandwidth between things [33], and eventual node unreachability [37].
Limited network lifetime. Lifetime should be maximized by incorporating duty-cycling mechanisms (*i.e.*, Active/sleep alternation by the turn on/off of the nodes' activity)[17].
Resource constraints. Things have stringent resource constraints for energy, processing power, and memory of devices [33].
Mobility. Mobile devices and highly dynamic network topology [14].
Vast number of heterogeneous devices. The increasing number of connected devices produces scalability issues in data communication, networking, service provisioning, and management.
vulnerability to security risks. The shared wireless medium and access to the Internet alleviate the effect of security risks [21].
Denial of Service (DoS). Also known as resource-exhaustion attack is one of the significant threats to availability, depriving users of services by consuming IoT nodes [41].

life's dangers. In such cases, gathering information can be done through IoT sensors and sent directly to the processing hubs [35] to detect failures and assess dangerous events. Thus, corrective, preventive, and rescue actions can be taken promptly.

Other examples of mission-critical applications are:

- military applications such as intrusion in remote or hostile environments,
- environmental monitoring such as detecting the presence of methane and carbon monoxide gases in mines and triggering rescue protocols,
- disaster management, for instance, detecting radioactive and toxic gases in hostile environments,
- rescue operations, for instance, detecting fires, jostling in large smart stores, and triggering evacuation protocols,
- health monitoring to monitor chronic disease patients [8] and heart, panic, and epileptic-related attacks of drivers through the Internet of Vehicles (IoV) to prevent accidents [15].

For critical applications, the recovery time in case of network failure could be intolerable. Application robustness, fault avoidance, and recoverability of communicant objects in the network in uncertain information require effective defense mechanisms; weaknesses must be controlled and corrected *before* disrupting service provision. Thus, to prevent the deterioration of IoT systems and maintain a fault-tolerant solution, effort should be invested in developing *proactive*, efficient monitoring of the network, and fast correction mechanisms [37, 24, 31].

III. Background, Related Work, & Research Gap

A. *Related IoT Enabling Protocols & Monitoring Techniques*

Several solutions to solve the connection and cooperation of things under different exigences exist in IoT. Some permit long-range communication using low energy (*cf.* [34] for a survey on Low Power Wide Area Networks). Let us note as an example LORA [6], the counterpart of the solution is sporadic communication resulting in very low bandwidth (*cf.* [1] for the limitations of LORA). For these limitations, we exclude long-range networks from our study.

A significant step to creating a serviceable IoT domain is the adaptation of the functioning to IP protocols. IEEE 802.15.4 is a well-known and widely used standard launched in 2003. It defines how Low-Rate Wireless Personal Area Networks (LR-WPANs) operate and the specifications of the Medium Access Control (MAC) and Physical (PHY) layers for LR-WPANs [16]. Moreover, IEEE 802.15.4 can be used for many higher-layer standards, such as Zigbee, Wireless HART, and radio frequency for consumer electronics.

Intending to allow low-end devices with limited power to connect to the Internet, the IETF created 6LoWPAN in 2004 [26]. The goal of 6LoWPAN was to include an adaptation layer between the IPv6 and the IEEE networks. This layer has encapsulation and compression techniques to enable adequate IPv6 packets' transmission over IEEE 802.15.4 communication channels. For recent reviews and studies of 6LoWPAN *cf.* [9, 44, 12].

As mentioned above, one of the pertinent challenges in LLNs is to use an efficient routing protocol that meets the applications' requirements, such as considering low-power IoT devices and short transmission ranges. In response to these challenges, standardization groups, specifically the Internet Engineering Task Force (IETF) and the Institute of Electrical and Electronics Engineers (IEEE), standardized a broadly applied Routing Protocol for Low-Power and Lossy Networks (RPL) that was proposed in RFC 6550 [4].

In this cost-based routing protocol, a Destination-Oriented Directed Acyclic Graph (DODAG) is built, directed from the things to a central node corresponding to a border router (BR) toward the other parts of the Internet. In the DODAG, nodes are organized into a "layered" tree, starting from a single root (the BR). The construction of each tree-like DODAG is based on the attribution of" ranks" (the rank defines the position of the node in the DODAG to the neighbors and the BR descending from the root), which is computed via applying an Objective Function (OF) which can eventually be defined based on QoS metrics (e.g., delay). Usually, a node forwards the data to a parent (with a lower cost, *i.e.*, lower rank) toward the sink. The DODAG can be used to send messages to things and actuators [9].

Network monitoring tools generally aim at detecting and localizing network faults and taking corrective actions. A monitoring technique ensures earlier detection of failures and

Table II
CENTRALIZED MONITORING ARCHITECTURES' PROS & CONS

Centralized monitoring	
Advantages:	• allows for simpler network management • The base station is always assumed to be accessible and can be equipped with unlimited resources, • can perform complex management tasks, thus, reducing the processing burden on resource-constrained nodes, and • the base station has a global knowledge of the network, and therefore, it can provide accurate management decisions.
Disadvantages:	• it incurs a high message overhead (bandwidth and energy) from data polling, and this limits scalability, • the base station is a single point of failure, • they limit the possibility of creating ad-hoc domains without dedicated infrastructures, • they represent a more static worldview, where device roles are fixed, rather than a dynamic worldview that recognizes that networks and devices, and their roles, may change over time, and • if a network is partitioned, nodes that cannot reach the base are left without any management functionality.

accelerates the repair. RPL proposes local repair and global *repair* mechanisms to reestablish the routing structure if it is failed [9]. These repair mechanisms are *reactive* since they are activated when a failure is detected. Consequently, nodes might be unreachable. As reported in [37], the average unreachable time of the node during DODAG reconstruction is almost three and a half minutes, which is the reason why numerous researchers have expressed concern over the routing issues in the IoT environment (*cf.* [3, 2]). For a recent review of the security of RPL-based 6LoWPAN networks in the Internet of Things [44].

A relatively similar problem is the monitoring of Wireless Sensor Networks, for which several monitor placement algorithms have been proposed. An extensive survey of MAC protocols can be found regarding mission-critical applications in [17]. There are a few propositions to supervise IoT network nodes [22, 23]. They propose passive monitoring techniques that use RPL in which the monitors are special, higher-order devices not limited in their resources. Unfortunately, this constraint is challenging since higher-order devices typically only constitute the minority of nodes in IoT networks. Consequently, it is only possible to cover the nodes and links partially.

The management of the monitoring system (and its scheduling) can be *centralized or distributed* (*cf.* [38, 25]). In centralized network management, a central entity, generally known as the base station, acts as the management station that collects information from all nodes and controls the entire network. Table II summarizes the advantages and disadvantages of centralized monitoring.

In previous work, energy-efficient monitor placement and scheduling in 6LowPAN were formulated in [27] as a multi-objective scheduling problem. The paper proposes a centralized computation where the objectives cover the minimization of energy consumption and the communication cost of monitoring.

The proposition is split into three phases. At first, the potential monitor sets (minimal vertex cover sets) are generated. The energy-efficient alternation of monitor sets needs the assignment of monitor sets on time periods. This subproblem is modeled as a Multi-Objective Generalized Assignment Problem. In the third phase, nodes' state transitions are optimized by solving a Traveling Salesman Path Problem (each node corresponding to a monitor set in a period). As a result of the decomposition, the method is *not exact* but gives optimums in each individual phase.

In subsequent work (*cf.* [28]), the exact formulation of the corresponding NP-hard optimization problem is described. The proposed model is based on a Binary Integer Program. The computed solution of centralized scheduling is optimal. However, as the problem size gets larger, the networks get denser. As a result, computing the assigned monitors' optimal schedule results requires a significantly long time. Hence, a distributed and simple mechanism is preferable. sporadic communication resulting in very low bandwidth (*cf.* [1] for the limitations of LORA). For these limitations, we exclude long-range networks from our study.

B. *Research Gap & Contributions*

Compared to existing solutions, the roadblocks to overcome include integration and interoperability to standardized protocols and advanced technologies across the value chain (devices, networks, middleware, service platforms, and application functions) to foster smart, sustainable coverage of user needs for IoT services and products in the specific real-life scenarios of the pilot.

Concerning the potential application of 6LowPAN-based IoT networks and applying RPL in systems with a critical mission, our work focuses on developing a proactive monitoring solution that monitors *cover* the entire IoT network. From the point of view of efficiency, the *placement of monitors* is crucial. Consequently, the following research question is posed to ensure full monitoring coverage: How many monitors are required, and where should they be placed?".

Considering the limited computational capacity of simple IoT devices, it is imperative to *reduce and distribute* the added energy and communication cost of monitoring. To preserve the power of batteries and prolong the network's lifetime in battery-powered WSNs and IoT, *duty cycles* (alternations of awake and sleep states of nodes performing primary functions) are usual, required techniques [17].

A crucial requirement is achieving *real-time adaptability* to network changes by providing a dynamic worldview that recognizes that network connectivity and devices' health and roles may change over time. Consequently, monitoring coverage should be ensured while respecting the devices' *limited and changing resources*.

Moreover, given the fragility of the centralized solutions, and the controls, we propose a distributed and simple scheduling mechanism to compute and alternate monitor sets.

In a nutshell, this work proposes a *proactive, dynamic, and distributed* network monitoring mechanism with monitor placement and scheduling for 6LoWPAN-based IoT networks intended for mission-critical applications. The proposed mechanism aims to ensure real-time, efficient monitoring coverage while respecting devices' limited and changing power resources to prolong the network lifetime.

According to the comprehensive literature review performed in this research, to the best of our knowledge, no research work has proposed monitoring models with dynamic, energy-efficient role scheduling and integration with the standardized RPL and 6LoWPAN protocols.

IV. PROPOSED PROACTIVE DYNAMIC IoT NETWORK MONITORING TECHNIQUE

A. *IoT monitoring specifications, requirements, & objectives*

This section explains the main requirements, assumptions, and objectives for network monitoring for resource-constrained IoT.

One of the critical requirements is designing a monitoring mechanism that is entirely interoperable with the standardized IoT protocol suite, especially the IPv6 for Low-power Wireless Personal Area Networks (6LoWPAN) and the Routing Protocol for Low-power and lossy networks (RPL).

As mentioned in the Related Work section, the authors in [23] proposed a monitoring technique in which the monitors are special, higher-order devices with unlimited resources. Since higher-order devices typically only constitute the minority of nodes in IoT networks, the nodes, and links could only be partially covered, risking the possibility of an undetected node failure, which is unacceptable in mission-critical applications. For this reason, in the proposed model it is required to perform monitoring by ordinary, resource-constrained nodes in the application.

Moreover, we propose a passive monitoring technique to supervise the network's state and the availability of nodes and links. This requirement ensures observing the network's functioning and traffic without causing additional monitoring traffic and overhead.

Since monitoring is only one of the activities performed by the things, the power of batteries is consumed by monitoring and other activities like sensing, transmitting, and receiving. Such activities are the primary activities defined by the IoT network's original mission; the monitoring mechanism does not control them. However, the eventual changes in the power resources as a result of the primary function of the things should be dynamically observed since they affect the scheduling of the monitors. Table III summarizes the proposed model's requirements.

Given the above-stated requirements, **the objectives of the propositions of this work are as follows:**

Table III
MONITORING SPECIFICATIONS & REQUIREMENTS

Interoperability. Monitoring should be interoperable with the standardized IoT protocol suite, specifically 6LoWPAN and RPL protocols.
Passive Monitoring using resource-constrained nodes.
Efficient Monitoring. Things have stringent *resource constraints* with only a fraction of the battery reserved for monitoring. **Pervasive Monitoring.** In addition to the monitoring role, Things perform sensing, transmission, and/or actuation. **Dynamic Monitoring.** Real-time adaptability that recognizes the change in network connectivity and devices' health.

- improving the resilience of critical-mission IoT domains via scalable, real-time monitoring that covers all network elements belonging to the concerned instance,
- balancing energy usage between monitors and following the eventual changes in the topology. The monitor set can (should) be changed dynamically, and
- computing (electing) the monitor set should be distributed.

be changed dynamically.

B. *Overview of Controlled Greedy Sleep(CGS) Algorithm*

The Controlled Greedy Sleep (CGS) algorithm proposed in [40] targets Wireless Sensor Networks used to monitor an area. Leveraging the high redundancy feature usually present in sensor networks, the mechanism's objective is ensuring that a required number k of sensors can provide measurements from each point in the area.

The mechanism is periodical; the lifetime of the WSN is prolonged by using different sensor sets in the periods. The selection and scheduling (duty-cycling) mechanism should ensure the k-coverage by the active sensors at each period. The distributed CGS provides a quasi-optimal sensor scheduling solution while respecting sensor node deployment and energy constraints. The same idea (duty-cycling) is used to organize network monitoring. Precisely, it is required that the additional monitoring load is distributed on the nodes by alternating between the monitor sets.)

The sensing assignment in the sensor network is represented by a bipartite graph $G_a = (S \cup R, E)$, where two disjoints sets of vertices represent the nodes S and geographical regions R (*cf.* Fig. 1), respectively. A region is the set of points in the area that a given sensor set can cover. In G_a, there is an edge e between sensor $s \in S$ and region $r \in R$ if and only if s covers region r. Sensors covering the same region can communicate directly since the communication range is at least twice the sensing range. The algorithm applies a *drowsiness factor*, which models the state of the sensors and their "desire" to sleep. The factor is computed at the beginning of each period for each node in a distributed manner. Supposing that a sensor

Table IV
GLOSSARY OF MODELING TERMS

Term	Description
k	Required number of sensors to provide measurements from each point in the monitored area.
S	Set of vertices representing the sensing nodes, $s \in S$.
R	Set of vertices representing the geographical regions, $r \in R$.
G_a	Bipartite graph representing the sensing assignment in the sensor network, consisting of the two disjoint sets S and R and the set of edges E.
E_s	Remaining energy of sensor node s
D_s	*Drowsiness factor* of sensor s, which represents the state of the sensor and its desire to sleep.
Φ_r	Coverage ratio of region r.
C_r	The number of sensors covering region r (the degree of r in G).
DTD_s	Decision Time Delay of node s, the time elapsed until each node s decides whether to stay awake or go to sleep.
AM	An Awake Message broadcast by s to inform the other nodes of its decision to stay awake.
DL_s	Delay List of node s
LAN_s	List of Awake Neighbors of node s.

node s has E_s remaining energy and can cover a set $R_s \subseteq R$, its drowsiness factor D_s is defined as follows:

$$D_s = \left\{ \begin{array}{cc} \frac{1}{E_s^\alpha} \sum_{r \in R_s} \Phi_r & \text{if } \Phi_r > 0, \forall r \\ \text{-1} & \text{otherwise.} \end{array} \right\} \quad (1)$$

where α is a positive constant (e.g. $\alpha = 2$), and Φ_r is the coverage ratio of region r, defined as follows:

$$\Phi_r = \left\{ \begin{array}{cc} \frac{1}{C_r - k} & \text{if } C_r > k \\ \text{-1} & \text{otherwise.} \end{array} \right\} \quad (2)$$

Here, C_r is the degree of the region r in G, *i.e.*, the number of sensors covering r. k is the desired level of redundancies in the coverage. This so-called "coverage ratio" Φ_r is positive if the region r is over-covered, *i.e.*, more than k sensors can cover it, and negative otherwise.

The drowsiness factor expresses a certain degree of the critical situation of the sensor. A sensor covering regions with low over-coverage could and should participate in more possible solutions than those covering regions also covered by many other sensors. The drowsiness factor is computed as the sum of the coverage ratios of the regions the sensor can observe. Consequently, sensors in critical positions could go to sleep whenever possible. Moreover, the drowsiness factor considers the energy of the sensor s; the smaller the sensor's energy, the larger its drowsiness. This factor permits a trade-off between energy usage and critical situations.

Depending on D_s, each node s computes a Decision Time Delay DTD_s inversely proportional to D_s and broadcasts it to its neighbor. When a sensor decides to be awake, it informs the neighbor nodes with an awake message (AM). From the received DTD and AM messages, each node builds a Delay List (DL_s) and a List of Awake Neighbors (LAN_s). After DTD_s time elapsed, each node s decides based upon its lists: If all $r \in R_s$ can be covered using only nodes present in LAN_s and nodes present in DL_s (these latter nodes are nodes not yet decided), then node s goes to sleep. Otherwise, s decides to be active and broadcasts an AM to inform the other nodes of its decision. Briefly, the CGS algorithm works as follows:

1) Run the network for a period of T
2) Wake up all sensors
3) Nodes with energy enough for at least one more period broadcast local Hello messages containing node geographical location
4) Each node s calculates its own drowsiness factor D_s
5) Based on D_s each node selects a Decision Time Delay (DTD_s)
6) Each node s broadcasts its DTD_s and collects other nodes' DTD and AM
7) From the received messages and after DTD_s, each node s decides its state for the next period.

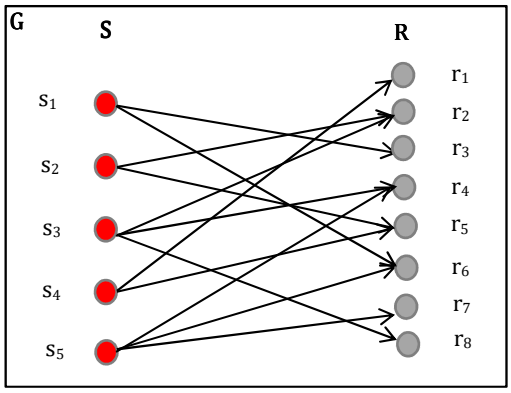


Figure 1. Bipartite graph showing the coverage of the sensors $(s_1, ..., s_5)$ and the sensing regions $(r_1, ..., r_8)$

This mechanism is a valid starting point for IoT network monitoring, but necessary adaptations are needed.

The following Section describes the proposed model and algorithmic solution.

Suppose a critical mission is realized using a DODAG topology $G_t = (V_t, E_t)$ for routing.

C. *Organization and Concepts*

The starting point of the developed proposition is based on the mechanism of the Controlled Greedy Sleep (CGS) [40] algorithm. Necessary adaptations are needed to satisfy the IoT network monitoring requirement.

The proposition contains two major elements:

- a cooperation protocol between nodes to assure the distributed scheduling, and
- an efficient computation algorithm to prepare the monitors' awake/sleep decisions.

The monitoring activity is organized in a timeline that is decomposed into a sequence of periods, $T = \{t_1, t_2, ..., t_m\}$ (*cf.* Fig. 2). Each period is characterized by the set of active monitors and the duration of the period: $t_m = (S_j^a, t_j)$, where the set S_j^a is the active monitor subset during t_j that

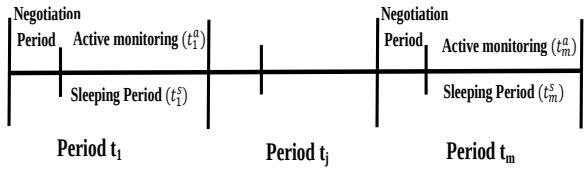


Figure 2. Monitoring Timeline, $T = \{t_1, t_2, ..., t_m\}$.

solves the coverage of the graph representing the *current* network topology. All periods assume the same period length. Moreover, a significant topology change (for instance new duty cycle) involves a new computation timeline T for the monitoring. The following definitions for node sets are used to describe the monitor selection and scheduling algorithm:

Table V
NODE SETS IN THE MONITOR SELECTION AND SCHEDULING ALGORITHM

	Definition
- *Candidate neighbor set.*	Each node $n \in V_t$ has its candidate neighbor set N_n, which is the set of nodes in the reception range of n (these nodes can be safely observed by n). It is the set of a potential target for n to monitor (set R_n in the bipartite graph G defined by the CGS mechanism).
- *Concurrent monitor set.*	This set S_n is composed of other potential concurrent monitors that can monitor at most one of the potential targets of n. Nodes in N_n but not only them are in S_n. A node x, which is outside the reception range of n but can monitor a node y inside of N_n, should be considered a concurrent monitor node for n. x is a neighbor node of y, a neighbor of n.

Consequently, it is essential to increase the knowledge of every node such that it knows the neighbors of its neighbors. One neighbor of one of its neighbors is usually called Neighbor-of-Neighbor (NoN) [20].

- *Awake Neighbors.* Similarly to CGS, each node n and at each moment of negotiations should know the set of concurrent monitors that are still awake *for monitoring*. This subset of S_n can be represented by the List of Awake Neighbors of n LAN_n.

Symmetric links are assumed; n can communicate with the neighbor nodes in N_n and observe them. During monitoring, a node selected as a monitor covers a subset of its candidate neighbor set.

Similar to CGS, at the beginning of each monitoring period, there is a short negotiation period (*cf.* Fig. 2), where the communication between neighbors is established. This periodical communication between neighboring nodes must be accomplished for the following purposes:

- updating the candidate neighbor set S_n for all $n \in V_t$, so that the current List of Awake Neighbors LAN_n is known, which constantly changes due to the lossy nature of 6LoWPANs or simply because of the applied duty cycling mechanism,
- informing neighboring nodes of the updated coverage ratio of each node, and
- informing neighboring nodes of each node's monitoring awake/sleep decision for the next period.

The problem presented in this work is mapped to the one dealt with by the CGS algorithm [40], albeit with significant differences.

- In CGS, the set S of concurrent sensors and observed regions R are disjoint (they are geographical regions, *cf.* Fig. 1). The two sets are composed of the same nodes for the problem in hand, where a *monitored* node can also be a monitor for another. The organization of the node sets is not a simple duplication of nodes. The edges should reflect the real possibilities of monitoring (the NoN set of nodes should be considered).
- In CGS, sensors are identified by their geographical locations. On the other hand, in 6LoWPAN-based IoT networks, the nodes' radio communication ranges are defined by link-local reachability, where nodes are discovered by the 6LoWPAN *Neighbor Discovery Protocol (NDP)* [1], and identified by unique RIME/IPv6 addresses.
- There is an edge between monitor s and element r in the graph G_a, if and only if r is within the radio environment of s. For monitor placement, the direction of edges is irrelevant, which implies that if there is a directed edge from s to r, s will be able to monitor r, and r can monitor s. Undirected graphs are used in several routing protocols, as in the models of [36], and [10].
- In the area coverage by WSNs, sensors that can observe the same region are neighbors for communications, a fact that does not apply to monitoring the wireless network itself. In the proposed dynamic monitor scheduling algorithm, it is essential to increase the knowledge of every node, such that it knows the state of all nodes that can observe at least one of its neighbors. Moreover, each node must know the neighbors of its neighbors, known as the Neighbor-of-Neighbor (NoN) set [20]. A node is awake/sleep schedule in the next period t_j is affected by the state of its NoN. The concurrent monitor set of a node n is the union of its neighbor and NoN sets, as given in (3).

$$S_n = N_n \bigcup NoN_n \tag{3}$$

The requirement of the knowledge of NoNs is illustrated in Fig. 3.

For v_1 and v_3, $N_1 = \{2, 3, 5\}$, and $N_3 = \{1, 6\}$ respectively. Suppose that it is decided that v_6 is "sleep-monitoring" in the next period t_j. If v_1 goes to sleep mode in the same period, v_3 will not be covered. Therefore, v_1 should know its neighbors of neighbors, which includes $NoN_1 = \{v_4, v_5, v_6, v_7\}$. For v_1, knowing that a member

[1]NDP is a messaging protocol that facilitates the discovery of neighboring devices over a network [39].

of its NoN_1, namely, v_6, is sleeping, it *should* decide to stay "active-monitoring". Otherwise, the neighbor of v_1, v_3, will not be covered.

The bipartite graph giving the relations between this illustrating network's potential monitoring and monitored nodes and a possible monitoring set of nodes are depicted in Fig. 4.

In our proposition, the monitoring system is relatively simple (in this case, the minimal coverage ratio k is equal to 1). It is a cheap and straightforward mechanism. The inconvenience of this solution is that some nodes (eventually monitor-actives) can be in a critical situation in the monitoring. It is the case when an active node is the only one monitoring another node. This case is illustrated in Fig. 4. In this example, nodes 2, 3, and 6 are monitored by only one monitor. Suppose that node 3 (which is also a monitor) fails. Then, until the repair of this failure, node 6 is not monitored. A k-coverage of nodes with $k > 1$ constraint can be applied to improve the fault tolerance of the monitoring system. For instance, by applying a 2-coverage of nodes, the monitoring system will tolerate a first failure of a monitor and can continue the monitoring. However, a node failure impacts the network's primary critical mission and the communication between the nodes and the BR. Consequently, after detecting a failure, it is necessary to repair the DODAG used by the application and immediately recompute its monitoring system.

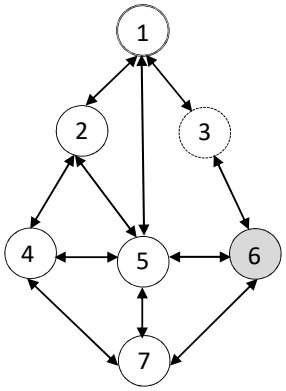


Figure 3. For v_1, knowing that a member of its NoN_1, namely, v_6, is sleeping decides to stay active-monitoring to ensure that its neighbor, v_3, is covered.

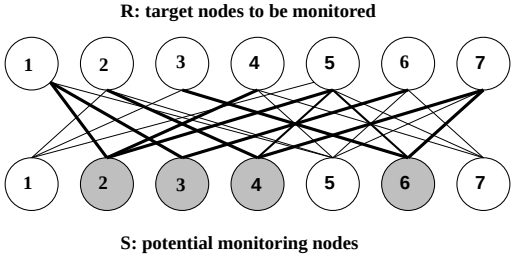


Figure 4. A possible result of the election of monitors in the example of Fig. 3.

D. Scheduling Mechanism

The proposed monitoring and the corresponding scheduling are described in Algorithms 1, 2, 3, and 4. The monitoring must function during the timeline's length, represented by $Timeline\_Length$. At the beginning of a new monitoring period, t_m, all nodes wake up (Algorithm 2 Step 2.1), estimate their remaining power (E_s), and initialize their parameters. Nodes with a remaining energy level high enough for monitoring (more than a given $Energy\_Threshold$) for at least one more period locally broadcast an Awake Message (AM) (Step 2.8). Otherwise, to conserve the remaining power for its primary function (sensing, actuation, and transmission), it broadcasts a Sleep Message SM and chooses the "monitoring-sleep" state (Steps 2.9 & 2.10). Naturally, those nodes are considered "sleeping" nodes.

It is noteworthy that the monitoring mechanism does not influence the node's primary duty cycle, *i.e.*, the radio is turned off only if it is idle for its primary function. Our computation concerns only the monitoring task, and the decisions are to select either a state of "monitoring-active" or a state of "monitoring-sleep".

When a node n receives a message from a neighbor, there are several tasks to perform: (1) update its List of Awake Neighbors (LAN_n), either by adding or removing this neighbor's address according to the neighbor's received state (monitoring-active or monitoring-sleep) (Algorithm 3 Step 3.2). Then, (2) update its list of Neighbors-of-Neighbors (NoN_n) from the received list of neighbors, $LAN_{neighbor}$ (Step 3.3). (3) Compute its own coverage ratio (*cf.* Equation 4), and (4) update its Delay List DL_n (Steps 3.4 & 3.5). Finally, (5) broadcast the updated parameters to its neighbors (Step 3.8).

At the end of the negotiation period, each node has to decide whether it will be "monitoring-active" or "monitoring-sleep" for the rest of the monitoring period. Algorithm 4 describes how these decisions are made, which mainly depend on the $drowsiness\_factor_n$ (*cf.* Equation 1). Each monitor's drowsiness factor includes the sum of coverage ratios of the objects it can monitor. Negative drowsiness indicates that a monitor cannot choose the "monitor-sleep" state. The smaller the energy E_n of a monitor candidate, the larger its drowsiness factor. On the contrary, minor drowsiness means a long Decision Time Delay (DTD) (*cf.* Equation 5). These delays provide priorities when nodes announce their Awake Messages (AM).

A monitor participating in several critical coverages is more likely to engage in several possible solutions than other potential monitors simultaneously covered by alternative nodes. Therefore, they have more significant drowsiness factors (*cf.* Equation 4). This property forces the nodes in critical situations to deactivate monitoring whenever possible and permits the loading of monitors which are in less critical situations.

Each node $n \in V_t$ has received the $coverage\_ratio_{neighbor}$

Algorithm 1 PROCEDURE DYNAMIC_DISTRIBUTED_MONITORING

Input: $Energy\_Threshold, Timeline\_Length, Period\_Length, Negotiation\_Period$
Output: Real-time monitoring schedule of 6LoWPAN-based IoT network

```
begin
1.1   while timeline_timer < Timeline_Length do
1.2     while period_timer < Period_Length do
1.3       while negotiation_timer < Negotiation_Period do
1.4         forEach n ∈ G do
1.5           START_UP();
1.6           RECEIVE_MESSAGE();
1.7           DECIDE_STATE();
1.8         end forEach
1.9       end while
1.10    end while
1.11  end while
end
```

Algorithm 2 PROCEDURE START_UP

Input: $Energy\_Threshold$
Output: Initialize node state

```
begin
2.1   RADIO_ON();
2.2   if E_n > Energy_Threshold do
2.3     state_n ⟵ 1;
2.4     drowsiness_factor_n ⟵ -1;
2.5     coverage_ratio_n ⟵ -1;
2.6     DTD_n ⟵ 0;
2.7     LOCAL_BROADCAST(AM, state_n, drowsiness_factor_n, coverage_ratio_n, DTD_n);
2.8   else
2.9     LOCAL_BROADCAST(SM, state_n ⟵ 0);
2.10     RADIO_OFF();
2.11  end if
end
```

Algorithm 3 PROCEDURE RECEIVE_MESSAGE

Input: Address of $neighbor$, $state\_neighbor$, $LAN\_neighbor$, $coverage\_ratio\_neighbor$, $DTD\_neighbor$
Output: Update LAN_n, NoN_n, $coverage\_ratio_n$, DL_n; LOCAL_BROADCAST updated parameters

```
begin
3.1   if state_neighbor > 1 do
3.2     LAN_n ⟵ LAN_n ∪ neighbor;
3.3     NoN_n ← NoN_n ∪ LAN_neighbor;
3.4     UPDATE_COVERAGE_RATIO(coverage_ratio_neighbor);
3.5     UPDATE_DL(DTD_neighbor);
3.6   else LAN_n ⟵ LAN_n/neighbor;
3.7   end if
3.8   LOCAL_BROADCAST(state_n, coverage_ratio_n, LAN_n, NoN_n);
end
```

Algorithm 4 PROCEDURE DECIDE_STATE

Input: LAN_n, NoN_n, DL_n
Output: node s decides whether to stay active monitoring or sleep in t_j and accordingly broadcast AM or SM

begin
4.1 **if** $coverage\_ratio_{neighbor}$ & $coverage\_ratio_{NoN} >= 0$ $\forall neighbor \in LAN_n, \forall NoN \in NoN_n$ **do**
4.2 COMPUTE_DROWSINESS_FACTOR();
4.3 COMPUTE_DTD();
4.4 **if** $DTD_n < DTD_{neighbor}$ $\forall neighbor \in LAN_n$, $\forall DTD_{neighbor} \in DL_n$
4.5 LOCAL_BROADCAST($SM, state_n \longleftarrow -1$);
4.6 WAIT(DTD_n);
4.7 RADIO_OFF();
4.8 **end if**
4.9 $drowsiness\_factor_n \longleftarrow -1$;
4.10 LOCAL_BROADCAST(AM);
4.11 **end if**
end

of the members of its LAN_n and NoN_n. If at least one of its neighbors or Neighbors-of-Neighbors is under-covered, (*i.e.*, has a negative coverage ratio), this indicates that at most one node can monitor it. Therefore n decides to stay awake for monitoring to maintain successful coverage (Step 4.9). Accordingly, it broadcasts an AM (Step 4.10). Otherwise, it *can* choose the "monitoring-sleep" decision depending on the comparison between its own DTD_n (*cf.* Equation 5) with its neighbors' $DTD_{neighbor}$. The different $DTD_{neighbor}$ values were previously received and saved in the Delay List DL (Step 4.4). In the case where n has the smallest DTD_n, it broadcasts an SM and turns off the monitoring activity (Steps 4.5 - 4.7).

$$\Phi_r = \left\{ \begin{array}{cc} \frac{1}{C_r - 1} & \text{if } C_r > 1 \\ \text{-1} & \text{otherwise.} \end{array} \right\} \quad (4)$$

$$DTD = \left\{ \begin{array}{cc} \frac{1}{D_s} & \text{if } D_s > 1 \\ 0 & \text{otherwise.} \end{array} \right\} \quad (5)$$

The decision delays DTD of nodes are less than the length of the negotiation period. In this manner, all available nodes in the network decide to be monitor-active or monitor-sleep before the next period. As it was shown, if the concurrent and awake monitoring nodes of a node n can ensure the coverage of the potential target nodes of n, then this latter chooses the monitor-sleep state.

Property 1: In each period, the set of monitor-active nodes of a connected DODAG is a minimal covering set[2].

V. Experimentation of the Proposition

The following experimentation illustrates the functioning and the performance of distributed scheduling. We analyze the effect of the reserved battery level for monitoring, the period length on the distribution of energy usage, and the network size. There is no existing, similar heuristic to compare with our distributed solution. The proposition for IoT network monitoring in [23] is different because it is based on particular monitor nodes.

[2]Remember, a minimal covering set is not obligatory a minimum set, but it can not be reduced without the loss of coverage.

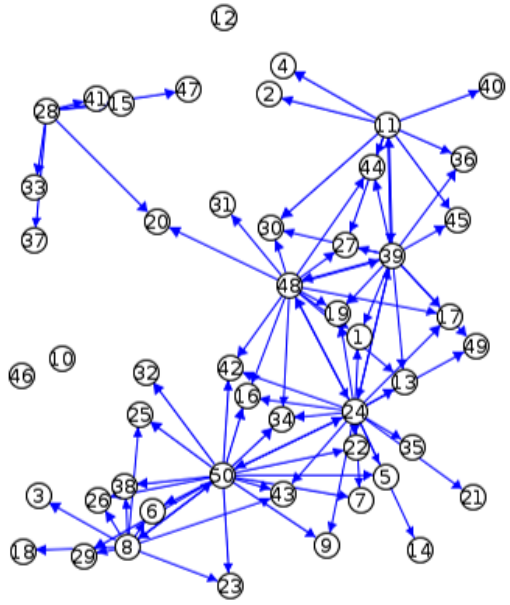


Figure 5. Radio communication within a network of 50 devices of type WisMote; node 1 is the Border Router.

A. *Implementation & Experimental Setup*

The proposed monitoring system for the resilience of critical IoT domains is implemented on the Contiki Operating System. For dynamic monitoring placement and scheduling, nodes'

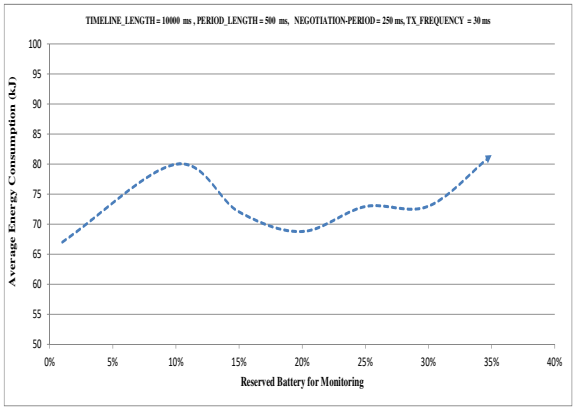


Figure 6. Effect of varying the size of the reserved battery for monitoring on the average energy consumption. Timeline Length = 10000 ms, Negotiation Period = 50 ms, Tx Frequency = 30 ms.

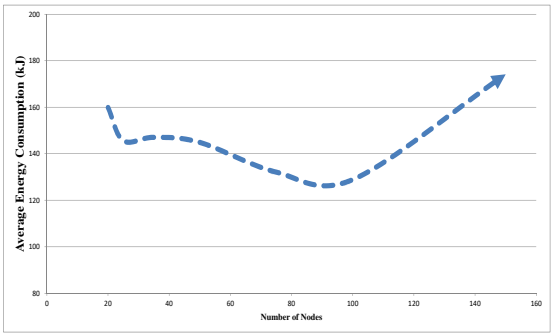
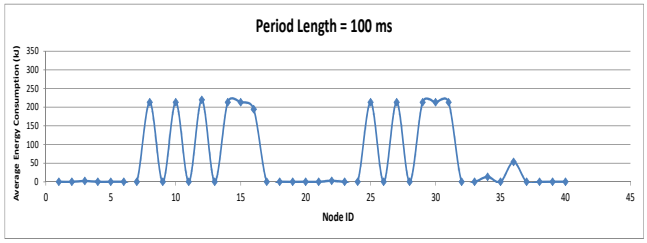


Figure 7. The average energy consumption of different-sized networks, Timeline Length = 10000 ms, Period Length = 2000 ms, Negotiation Period = 50 ms, Tx Frequency = 30 ms, Reserved Battery = 10%.

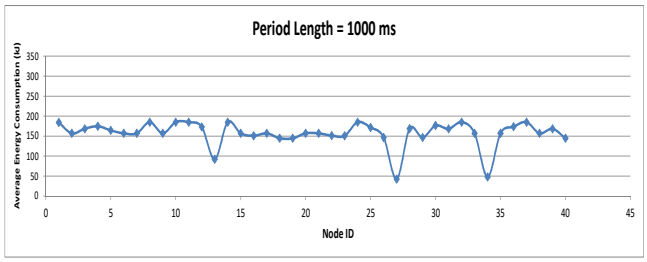
current power levels should be estimated as accurately as possible. The device's power is allocated to monitoring and, more importantly, to the primary activities, including sensing, actuation, processing, and transmission. The monitoring schedule should be efficient enough not to influence the energy required to perform the primary functions. The WisMote [5] is taken as a candidate platform for the monitoring mechanism. It features a 16-bit MSP430 with 20-bit support, 16 kB RAM, a nominal 128 kB, 192 kB or 256 kB ROM, and CC2520 radio transceiver, with light, battery, and radio sensors. It is powered by a pair of AAA batteries with 3 volts. The total energy available by the WisMote is calculated as follows:

$$2 \times (1.15\ Ah) \times (1.5\ V) \times (3600\ s) = 11421\ J = 11421000\ mJ \quad (6)$$

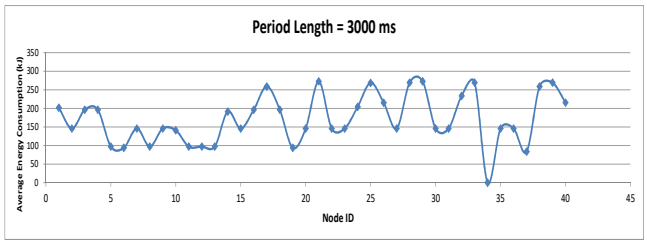
The `POWER_TRACE` procedure is embedded in Contiki to estimate the current energy level of nodes. `POWER_TRACE` procedure embedded in Contiki is used. Its output is printed



(a)



(b)



(c)

Figure 8. Effect of varying the period length on the average energy consumption, (a) period length = 100 ms; (b) period length = 1000 ms; (c) period length = 3000 ms. Timeline Length = 10000 ms, Negotiation Period = 50 ms, Tx Frequency = 30 ms, Reserved Battery = 10%

in timer ticks as follows:

- tx - the number of ticks the radio has been in transmit mode *(energest-type-transmit)*
- rx - the number of ticks the radio has been in receive mode *(energest-type-listen)*
- cpu - the number of ticks the CPU has been in active mode *(energest-type-cpu)*
- $cpu - idle$ - the number of ticks the CPU has been in idle mode *(energest-type-lpm)*

With each call of the `START_UP` procedure, `POWER_TRACE` is called, and the current energy level E_s is estimated by executing the following computations (Algorithm 2 Step 2.2).

$$\textit{ticks-in-tx-mode} = \textit{energest-type-time} \times \textit{energest-type-transmit} \quad (7)$$

$$\textit{seconds-in-tx-mode} = \frac{\textit{ticks-in-tx-mode}}{\textit{rtimer-arch-second}} \quad (8)$$

To compute the average current consumption (in milliamperes, mA), multiply each of tx, rx, cpu, cpu-idle with the respective

current consumption in that mode in mA (the values are obtained from the datasheet of the node), sum them up, and divide by *rtimer-arch-second*,

$$current = tx \times \textit{current-tx-mode} + rx \times \textit{current-rx-mode} + cpu \times \textit{current-cpu} + \frac{\textit{cpu-idle} \times \textit{current-idle}}{\textit{rtimer-arch-second}} \quad (9)$$

$$charge = \frac{current \times (cpu + \textit{cpu-idle})}{\textit{rtimer-arch-second}} \quad (10)$$

To compute the power (in milliwatts, mW), multiply the average current consumption by the voltage of the device:

$$power = current \times voltage \quad (11)$$

Finally, to compute the energy consumption (in millijoules, mJ), multiply the power with the duration in seconds or multiply the charge with the voltage of the system:

$$energy = charge \times voltage \quad (12)$$

B. *Experimental Results*

Experimentation is performed within the Contiki OS using the COOJA network simulator, the *de facto* simulator for constrained-IoT applications. The dynamic distributed monitoring mechanism is tested using network instances of random sizes and topologies (network size ranges from 20 to 200 nodes). Fig. 5 illustrates the radio communication and states in a network of 50 nodes of type WisMote. At the moment of the snapshot, only a subset of nodes is active for the communications (nodes 8, 11, 13, 24, 28, 39, 49, and 50). One can see the neighbors of these nodes, which receive (and can detect) the messages. Other nodes (nodes 10, 12, and 46) do not receive any messages and can not monitor the mentioned active node set; however, they can eventually monitor other nodes.

It is assumed that each node has a reserved battery for the monitoring activity across the entire timeline length, apart from the energy dedicated to the main functions. We tested the model's sensitivity towards variations in the reserved battery for monitoring during experimentation. Eight trials were run for which the reserved battery ranged from 1% to 35% of the total available battery of the WisMote, which corresponded to 112.10 - 3997.35 kJ. The $Timeline\_Length$, $Period\_Length$, $Negotiation\_Period$, and the frequency of transmission ($Tx-frequency$) were set in these trials to 10000 ms, 500 ms, 250 ms, and 30 ms, respectively. The results are displayed in Table VI.

Comparing the two extreme thresholds, one where the reserved battery is tightened the most (1%) and another where it is stretched to 35%, produced an interesting result: the average energy consumption in the case of the 1% reserved battery is reduced by 21.55%. This result highlights the model's adaptability towards tight energy constraints, as it strives to preserve scarce resources by effectively distributing the monitoring role. Some nodes decided not to participate in the monitoring activity, thus rendering a zero energy consumption level. Those nodes decided after ensuring that other monitors covered the entire set of neighbors.

Another set of experiments was designed to test the correlation between the period length and the average energy consumption. The $period\_length$ should be carefully chosen such that it is neither too short nor too long. A too-short $period\_length$ may result in false alarms. On the other hand, a shorter $period\_length$ may unnecessarily exhaust the energy of monitors as they are awake-monitoring for quite a long time. Each negotiation also corresponds to an additional cost, communications, frequent transitions, and an overhead for monitoring. A too-short $period\_length$ can lead to an unnecessary increase in overheads. A too-long $period\_length$ may drain some monitors' power as they are awake-monitoring for quite a long time, giving an unbalanced energy usage across the set of nodes.

Table VII displays the average energy consumption and the standard deviation in response to varying the $Period\_Length$. The $Timeline\_Length$, $Negotiation\_Period$, and $Tx-frequency$ were fixed in all trials to 10000 ms, 250 ms, and 30 ms. A subset of those trials is displayed in Fig. 8. There is a trade-off between energy consumption and the balance of the monitoring load among the nodes. It can be seen in Fig. 8 (also *cf.* the Standard Deviation column in VII) that a very short $Period\_Length$, 100 ms, results in an unfair distribution of the monitoring load, where some nodes exhaust comparatively high amounts of energy for monitoring while others are at a zero level consumption; which is illustrated by the high standard deviation value in Table VII. On the other hand, a too-long $Period\_Length$ of 3000 ms revealed a significant rise in the average energy consumption, which is justified by the long monitoring duty cycles. It is detected that the best combination of relatively low average energy consumption and a good balance between the monitoring loads is achieved when the $Period\_Length$ is set to 1000 ms.

The final set of experiments was performed to test the effect of the network size on energy consumption and the model's scalability. The network size was increased to 150 nodes and 3200 links. It can be seen from VIII that the proposition developed in this research is robust towards the network size. Results show that the percentage of energy consumption from the total available battery of WisMote never exceeds 1.36%, regardless of the network size. Fig. 7 depicts the increase in the average energy consumption against the network size, which is almost negligible.

VI. Conclusion

The proposed model targets the dynamic distributed monitoring placement and scheduling of mission-critical IoT networks, with complete interoperability with the IoT standardized protocols. The model's dynamic feature ensures the real-time adaptation of the monitoring schedule to the frequent network instabilities without requiring to re-solve the monitoring placement and scheduling problems with each abrupt change in the network topology or the nodes' availability. The distributed feature aims to reduce the communication overhead

Table VI
ENERGY CONSUMPTION OF A NETWORK OF 20 NODES WITH DIFFERENT LEVELS OF RESERVED BATTERY FOR MONITORING. $Timeline\_Length$ = 10000 ms, $Period\_Length$ = 500 ms, $Negotiation\_Period$ = 250 ms, $Tx - frequency$ = 30 ms.

Reserved battery(%)	**Reserved battery(kJ)**	**Avg. consump.(kJ)**
1	114.21	67.01
10	1142.10	79.97
15	1713.15	72.03
20	2284.20	68.78
25	2855.25	72.94
30	3426.30	72.99
35	3997.35	81.45

Table VII
ENERGY CONSUMPTION OF A NETWORK OF 40 NODES WITH DIFFERENT PERIOD LENGTHS. $Timeline\_Length$ = 10000 ms, $Negotiation\_Period$ = 50 ms, $Tx - frequency$ = 30 ms, RESERVED BATTERY = 10% (1142.1 KJ).

Period Length (ms)	**Avg. consumption (kJ)**	**Standard Dev. (kJ)**
100	60.17	95.23
500	142.27	144.20
1000	156.82	31.34
2000	94.64	105.65
2500	125.79	45.37
3000	215.43	66.41

Table VIII
AVERAGE AND PERCENTAGE OF ENERGY CONSUMPTION OF DIFFERENT-SIZED NETWORKS. $Timeline\_Length$ = 10000 ms , $Period\_Length$ = 500 ms, $Negotiation\_Period$ = 250 ms, $Tx - frequency$ = 30 ms.

Number of nodes	**Avg. consumption (kJ)**	**% of consumption**
20	159.89	1.36
25	145.61	1.27
50	144.89	1.26
75	132.09	1.27
100	128.89	1.14
150	174.21	1.13

between monitors and the Border Router, often resulting from centralized monitoring mechanisms.

The dynamic monitoring mechanism follows the basic idea of the Controlled Greedy Sleeping (CGS) algorithm proposed in [40]. Necessary adaptations on CGS for the scheduling of monitoring activities have been proposed. The monitoring awake/sleep schedule of nodes is computed using the notions of coverage ratio and drowsiness factor, which ensure the coverage of the entire set of critical nodes while prioritizing the awake/sleep decision based on coverage and energy levels. Successful neighbor discovery and knowledge about the neighbors' state are achieved by inter-communication between nodes. This communication is scheduled at the beginning of each period, namely, the negotiation period. Nodes with critical monitoring coverage, *i.e.*, monitoring neighbors not covered by other monitors) are not allowed to sleep.

Performance evaluations and accurate energy levels estimation are achieved using Contiki/COOJA, the *de facto* network simulator for constrained IoT. Simulations were performed to evaluate the model's adaptability to tight energy constraints. The results show that the tighter the energy constraint, the lower the average energy consumption while ensuring full monitor-network coverage. The monitoring schedule guarantees a smooth operation of the things' main functions, as it strives to preserve scarce resources by effectively distributing the monitoring role. A sensitivity analysis was conducted within the experiments to obtain the "best" combination between the parameters and minimize the trade-off between them.

Compared to the previously proposed three-phase decomposition [40], the dynamic distributed heuristic achieves better computational complexity and scalability results. The only limitation is that the schedule is not *exact*. However, with the benefit of achieving robust, real-time adaptability to network changes and the distributed mechanism's reduced computational and communication overhead, the dynamic model's performance is superior. Further experimentation and comparisons between the two models are required regarding energy consumption, the size of the monitoring sets, and the time required to obtain the monitoring schedule, depending on the network's size. The evaluation of the dynamic heuristic's approximation factor is left for future work.

REFERENCES

[1] F. Adelantado et al. "Understanding the Limits of LoRaWAN". In: *IEEE Communications Magazine* 55.9 (2017), pp. 34–40. DOI: 10.1109/MCOM.2017.1600613.

[2] David Airehrour, Jairo Gutierrez, and Sayan Kumar Ray. "Secure Routing for Internet of Things: A Survey". In: *Journal of Network and Computer Applications* 66 (2016), pp. 198–213. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2016.03.006. URL: https://www.sciencedirect.com/science/article/pii/S1084804516300133.

[3] David Airehrour, Jairo A. Gutierrez, and Sayan Kumar Ray. "SecTrust-RPL: A Secure Trust-Aware RPL Routing Protocol for Internet of Things". In: *Future Generation Computer Systems* 93 (2019), pp. 860–876. ISSN: 0167-739X. DOI: 10.1016/j.future.2018.03.021. URL: https://www.sciencedirect.com/science/article/pii/S0167739X17306581.

[4] Roger Alexander et al. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. RFC 6550. Mar. 2012. DOI: 10.17487/RFC6550. URL: https://rfc-editor.org/rfc/rfc6550.txt.

[5] ARAGO. *WisMote*. URL: http://www.aragosystems.fr/produits/wisnet/wismote/.

[6] Martin Bor, John Vidler, and Utz Roedig. "LoRa for the Internet of Things". In: *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*. EWSN '16. Graz, Austria: Junction Publishing, 2016, pp. 361–366. ISBN: 9780994988607.

[7] Brian Buntz. *The Top 20 Industrial IoT Applications*. https://www.iotworldtoday.com/2017/09/20/top-20-industrial-iot-applications/. Accessed: 2020-11-08. 2017.

[8] Madhavi Latha Challa, K. L. S. Soujanya, and C. D. Amulya. "Remote Monitoring and Maintenance of Patients via IoT Healthcare Security and Interoperability Approach". In: *Cybernetics, Cognition and Machine Learning Applications*. Ed. by Vinit Kumar Gunjan et al. Singapore: Springer Singapore, 2020, pp. 235–245. ISBN: 978-981-15-1632-0. DOI: 10.1007/978-981-15-1632-0_22.

[9] Khalid A. Darabkh et al. "RPL Routing Protocol over IoT: A Comprehensive Survey, Recent Advances, Insights, Bibliometric Analysis, Recommendations, and Future Directions". In: *Journal of Network and Computer Applications* 207 (2022), p. 103476. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2022.103476. URL: https://www.sciencedirect.com/science/article/pii/S1084804522001242.

[10] Pierre Fraigniaud and George Giakkoupis. "Greedy Routing in Small-World Networks with Power-Law Degrees". In: *Distributed computing* 27.4 (2014), pp. 231–253. DOI: 10.1007/s00446-014-0210-y.

[11] Peter Friess and Rolf Riemenschneider. "IoT Ecosystems Implementing Smart Technologies to Drive Innovation for Future Growth and Development". In: *Digitising the Industry Internet of Things Connecting the Physical, Digital and VirtualWorlds*. River Publishers, 2022, pp. 5–13. DOI: org/10.1201/9781003337966.

[12] Matheus Araujo Gava et al. "Optimizing Resources and Increasing the Coverage of Internet-of-Things (IoT) Networks: An Approach Based on LoRaWAN". In: *Sensors* 23.3 (2023). ISSN: 1424-8220. DOI: 10.3390/s23031239.

[13] Gopal Ghosh, Monica Sood, Sahil Verma, et al. "Internet of Things based video Surveillance Systems for Security Applications". In: *Journal of Computational and Theoretical Nanoscience* 17.6 (2020), pp. 2582–2588. DOI: 10.1166/jctn.2020.8933.

[14] Brij B Gupta and Megha Quamara. "An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols". In: *Concurrency and Computation: Practice and Experience* 32.21 (2020), e4946.

[15] Baofeng Ji et al. "Survey on the Internet of Vehicles: Network Architectures and Applications". In: *IEEE Communications Standards Magazine* 4.1 (2020), pp. 34–41. DOI: 10.1109/MCOMSTD.001.1900053.

[16] Wafa'a Kassab and Khalid A Darabkh. "A–Z survey of Internet of Things: Architectures, protocols, applications, recent advances, future directions and recommendations". In: *Journal of Network and Computer Applications* 163 (2020), p. 102663. DOI: 10.1109/ICCCNT.2017.8203943.

[17] Adam Kozłowski and Janusz Sosnowski. "Energy efficiency trade-off between duty-cycling and wake-up radio techniques in IoT networks". In: *Wireless Personal Communications* 107.4 (2019), pp. 1951–1971. DOI: 10.1007/s11277-019-06368-0.

[18] In Lee. "The Internet of Things for enterprises: An ecosystem, architecture, and IoT service business model". In: *Internet of Things* 7 (2019), p. 100078. ISSN: 2542-6605. DOI: 10.1016/j.iot.2019.100078.

[19] Ankur Lohachab and Bidhan Karambir. "Critical analysis of DDoS—An emerging security threat over IoT networks". In: *Journal of Communications and Information Networks* 3 (2018), pp. 57–78. DOI: 10.1007/s41650-018-0022-5.

[20] Gurmeet Singh Manku, Moni Naor, and Udi Wieder. "Know Thy Neighbor's Neighbor: The Power of Lookahead in Randomized P2P Networks". In: *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*. ACM. 2004, pp. 54–63. DOI: doi.org/10.1145/1007352.1007368.

[21] George Matta et al. "Risk management and standard compliance for cyber-physical systems of systems". In: *Infocommunications Journal* 13.2 (2021), pp. 32–39. DOI: 10.36244/ICJ.2021.2.5.

[22] Anthéa Mayzaud, Rémi Badonnel, and Isabelle Chrisment. "A distributed monitoring strategy for detecting version number attacks in RPL-based networks". In: *IEEE Transactions on Network and Service Management* 14.2 (2017), pp. 472–486. DOI: 10.1109/TNSM.2017.2705290.

[23] Anthéa Mayzaud et al. "Using the RPL Protocol for Supporting Passive Monitoring in the Internet of Things". In: *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*. IEEE. 2016, pp. 366–374. DOI: 10.1109/NOMS.2016.7502833.

[24] Bacem Mbarek, Mouzhi Ge, and Tomáš Pitner. "Proactive trust classification for detection of replication attacks in 6LoWPAN-based IoT". In: *Internet of Things* 16 (2021), p. 100442. DOI: 10.1016/j.iot.2021.100442.

[25] Jozef Mocnej et al. "Decentralised IoT Architecture for Efficient Resources Utilisation". In: *IFAC-PapersOnLine* 51.6 (2018). 15th IFAC Conference on Programmable Devices and Embedded Systems PDeS 2018, pp. 168–173. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2018.07.148. URL: http://www.sciencedirect.com/science/article/pii/S2405896318308942.

[26] Guido Moritz et al. "Beyond 6LoWPAN: Web services in wireless sensor networks". In: *IEEE Transactions on Industrial Informatics* vol.9 (Nov. 2013), pp.1795, 1805. DOI: 10.1109/TII.2012.2198660.

[27] B. Mostafa et al. "An Energy-Efficient Multiobjective Scheduling Model for Monitoring in Internet of Things". In: *IEEE Internet of Things Journal* 5.3 (2018), pp. 1727–1738. ISSN: 2327-4662. DOI: 10.1109/JIOT.2018.2792326.

[28] Basma Mostafa et al. "Optimal proactive monitor placement & scheduling for IoT networks". In: *Journal of the*

Operational Research Society 73.11 (2022), pp. 2431–2450. DOI: 10.1080/01605682.2021.1992310.

[29] Radouan Ait Mouha. "Internet of Things (IoT)". In: *Journal of Data Analysis and Information Processing* 9.2 (2021), pp. 77–101. DOI: 10.4236/jdaip.2021.92006.

[30] Seyyed Esmaeil Najafi, Hamed Nozari, and Seyyed Ahmad Edalatpanah. "Investigating the Key Parameters Affecting Sustainable IoT-Based Marketing". In: *Computational Intelligence Methodologies Applied to Sustainable Development Goals*. Ed. by José Luis Verdegay, Julio Brito, and Carlos Cruz. Cham: Springer International Publishing, 2022, pp. 51–61. ISBN: 978-3-030-97344-5. DOI: 10.1007/978-3-030-97344-5_4.

[31] Nasser Al-Qadami and Andrey Koucheryavy. "Fault-Tolerance Algorithm in Wireless Sensor Networks." In: *Infocommunications Journal, Hungary, ISSN* (2015), pp. 2061–2079.

[32] Rixuan Qiu et al. "A Fine-grained Dynamic Access Control Method for Power IoT Based on Kformer". In: *INFOCOMMUNICATIONS JOURNAL* 14.4 (2022), pp. 79–85. DOI: 10.36244/ICJ.2022.4.11.

[33] Pethuru Raj et al. *The Internet of Things and Big Data Analytics: Integrated Platforms and Industry Use Cases*. CRC Press, 2020. DOI: 10.1201/9781003036739.

[34] U. Raza, P. Kulkarni, and M. Sooriyabandara. "Low Power Wide Area Networks: An Overview". In: *IEEE Communications Surveys Tutorials* 19.2 (2017), pp. 855–873. DOI: 10.1109/COMST.2017.2652320.

[35] Sudhir K. Routray et al. "Satellite Based IoT for Mission Critical Applications". In: *2019 International Conference on Data Science and Communication (IconDSC)*. 2019, pp. 1–6. DOI: 10.1109/IconDSC.2019.8817030.

[36] Olivier Ruas. "Neighbor-of-Neighbor Routing In Small-World Networks With Power-Law Degree". PhD thesis. INRIA-IRISA Rennes Bretagne Atlantique, équipe ASAP, 2013.

[37] Rashmi Sahay, G. Geethakumari, and Barsha Mitra. "A novel Network Partitioning Attack against Routing Protocol in Internet of Things". In: *Ad Hoc Networks* 121 (2021), p. 102583. ISSN: 1570-8705. DOI: 10.1016/j.adhoc.2021.102583.

[38] O. Salman et al. "An Architecture for the Internet of Things with Decentralized Data and Centralized Control". In: *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*. 2015, pp. 1–8. DOI: 10.1109/AICCSA.2015.7507265.

[39] Zach Shelby and Carsten Bormann. *6LoWPAN: The Wireless Embedded Internet*. Vol. 43. John Wiley & Sons, 2011. DOI: 10.1002/9780470686218.

[40] G. Simon et al. "Dependable k-coverage Algorithms for Sensor Networks". In: *2007 IEEE Instrumentation Measurement Technology Conference IMTC 2007*. May 2007, pp. 1–6. DOI: 10.1109/IMTC.2007.379153.

[41] Rahim Taheri et al. "Similarity-based Android malware detection using Hamming distance of static binary features". In: *Future Generation Computer Systems* 105 (2020), pp. 230–247. DOI: 10.1016/j.future.2019.11.034.

[42] G. Tangari et al. "Self-Adaptive Decentralized Monitoring in Software-Defined Networks". In: *IEEE Transactions on Network and Service Management* 15.4 (2018), pp. 1277–1291. DOI: 10.1109/TNSM.2018.2874813.

[43] Jean-Philippe Vasseur. "Terms Used in Routing for Low-Power and Lossy Networks". In: *RFC* 7102 (2014), pp. 1–8. DOI: 10.17487/RFC7102. URL: https://doi.org/10.17487/RFC7102.

[44] Abhishek Verma and Virender Ranga. "Security of RPL Based 6LoWPAN Networks in the Internet of Things: A Review". In: *IEEE Sensors Journal* 20.11 (2020), pp. 5666–5690. DOI: 10.1109/JSEN.2020.2973677.

Basma Mostafa received a dual Ph.D. degree in Computer Science and Operations Research in 2019 from the University of Montpellier, France, and the Faculty of Computers and Artificial Intelligence, Cairo University, 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 an Assistant professor 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 on developing optimized models for monitoring IoT networks. Her research activities focus mainly on combinatorial optimization, network optimization, linear and integer programming, modeling, and simulation.

Miklos Molnar received the graduation degree from the Faculty of Electrical Engineering, University BME, Hungary, in 1976, the Ph.D. degree in computer science from the University of Rennes 1, France, in 1992, and the French HDR degree in 2008. He has been with the University of Montpellier, France, since 2010. He is a Professor Emeritus with the Department of Computer Science in the laboratory LIRMM of Montpellier. His research activities are in combinatorial optimization, network design, and optimization algorithms and tools. He conducted several studies to find dependable routes for sensible communications, efficient multicast routes, energy-aware k-coverage, 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 a master's degree from Bergen University, Norway, the M.B.A. degree from the Maastricht School of Management, 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, Egypt. He is also an Adjunct Professor with the System Dynamics Group at the 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, 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 recently been a Technical International Expert with the French Ministry of Foreign and European Affairs, from 2012 to 2016. Dr. Benslimane received the French Award of Scientific 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. His current research interests include the development of secure communication protocols for vehicular networks and the Internet of Things.

Sally Kassem received her graduate 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, 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 at the School of Engineering and Applied Science, Industrial Engineering Program, Nile University, Egypt. Her current research interests include mathematical modeling and optimization, linear and integer programming, operations research methodologies, supply chain, logistics, and modeling and simulation.