

Adjusting Group Communication in Dense Internet of Things Networks with Heterogeneous Energy Sources

Renato Mota¹, André Riker¹, and Denis Rosário¹

¹Federal University of Pará (UFPA) – Belém, PA – Brasil

renato.mota@icen.ufpa.br, ariker@dei.uc.pt, denis@ufpa.br

Abstract. *Internet-of-Things (IoT) environments will have a large number of nodes organized into groups to collect and to disseminate data. In this sense, one of the main challenges in IoT environments is to dynamically manage communication characteristics of IoT devices to decrease congestion, traffic collisions, and excessive data collection, as well as to balance the use of energy resources. In this paper, we introduce an energy-efficient and reliable Self Adjusting group communication of dense IoT Network, called SADIN. It configures the communication settings to ensure a dynamic control of IoT devices considering a comprehensive set of aspects, i.e., traffic loss, event relevance, amount of nodes with renewable batteries, and the number of observers. Specifically, SADIN changes the communication interval, the number of data producers, the reliability level of the network. Extensive evaluation results show that SADIN improves system performance in terms of message loss, energy consumption, and reliability compared to state-of-the-art protocol.*

1. Introduction

Internet-of-Things (IoT) objects are commonly playing monitoring roles, due to their low cost and size. These tiny machines are often deployed in large quantities and lead to the formation of dense IoT environments. Smart cities and precision agriculture will be common scenarios with dense disposal of IoT objects.

IoT is expanding at a fast rate to provide internet connections for IoT devices, which are expected to reach 4.1 billion by 2024 [Cerwall et al. 2015]. In this sense, the highly dense IoT networks will suffer from excessive energy consumption and network congestion [Sekhar 2014]. In these environments, it is common to experience the peak of traffic burst caused by detected event occurrences. Usually, many objects transmit information about the same event, leading to network congestion and high energy consumption [Orsino et al. 2016].

Moreover, the congestion obstacle in IoT comes with associated issues, such as the network’s reliability level. Message loss is linked to network congestion because the wireless channels become less accessible and the collision rate tends to increase. Hence, missing relevant data is a risk to be taken. Therefore, during periods of congestion, adjusting the message confirmation mechanism is crucial. In general, setting messages as Confirmable or Non-Confirmable solves this issue.

Group communication is becoming a promising solution for some of the problems in dense IoT networks [Islam et al. 2019]. That is because it enables the nodes to reduce

the amount of traffic by avoiding unnecessary messages. Also, it decreases the complexity of dealing with a large number of devices individually. Thus, group communication can be applied in dense IoT environments to approach energy consumption and congestion problems. Additionally, nodes with a renewable energy source can perform more energy consuming tasks in a group. Consequently, allowing the nodes with non-replenishable to act as a compliment for the network.

IoT devices might collect data with different sensing relevance levels, where the relevance can be defined based on node location or collected data [Costa and Guedes 2013]. Hence, this is very important for group communication mechanism, as these should provide a consistent control on how much data the network produces and which messages must be more protected against traffic loss.

An important player in this scenario is the Constrained Application Protocol (CoAP), that provides energy efficient communication for devices with energy, memory, and processing constraints, where CoAP clients define multiple groups of nodes, called monitoring groups, to observe points of interest, specifying the client's preferred communication settings (*e.g.*, communication periodicity). [Riker et al. 2018]. This article presents a solution that capitalizes on these core features of CoAP.

There are some solutions available in the current state-of-the-art for IoT group communication. However, these solutions fail to integrate renewable energy sources and to tackle congestion, lack of reliability and excessive energy consumption. Also, they fail to coordinate the IoT groups in a way that enables them to differentiate their communication reliability according to the relevance of the data traffic. Finally, they do not provide adaptive control for IoT groups in terms of traffic reliability, frequency data reporting, and the number of nodes that send sensing information.

In this paper, we propose a Self Adjusting group communication for dense IoT Network, called SADIN. The main idea of SADIN is to adjust the notification interval, the rate of confirmable messages and the number of observers. These adjustments are decisions made upon some core factors. SADIN consider the following parameters for the decision-making: (i) the critical level of an event; (ii) the energy source (*i.e.*, renewable and traditional batteries); (iii) the percentage of active observers; (iv) the current loss rate of each node. In this way, SADIN considers these set of parameters as input for a Fuzzy Decision System for adjusting the group communication.

Simulation results show that SADIN improves the performance of IoT applications in terms of notification interval, message loss rate, and energy consumption compared to CoAP. SADIN protected relevant messages, that might contain crucial data, causing the network to be more reliable and efficient.

The paper is structured as follows. Section 2 discusses the related works. Section 3 details SADIN components, namely Settings Adjust and Decision Fuzzy System, as well as the implementation details. Section 4 presents the evaluation methodology and results. Section 5 presents the conclusions.

2. Related Work

There is a wide variety of green solutions for IoT, but only a few of them consider traffic quality indicators, network resources, and heterogeneous energy sources. [Rani et al. 2015]

proposed a clustering technique that allows an IoT network to become more energy efficient. However, they do not consider, for instance, loss rate or reacting to changes in the sensor values.

[Betzler et al. 2016] introduced a congestion control mechanism on top of CoAP which provided a significant decrease in re-transmissions compared to other solutions. However, this approach just acknowledged the internal factors of congestion *e.g.*, message type. This work does not explore event criticalness. Moreover, it does not consider different energy sources, which benefits the network's lifetime. [Jarvinen et al. 2018] later improved this work, they took a similar path, but regarded heavy congestion scenarios.

[Jaramillo et al. 2015] tackles some of the problems in wireless sensor networks, such as traffic loss and delay. The authors focused on providing improvements in the MAC protocol, while considering critical factors of the network's service quality, namely latency and throughput. The positive points are the fact that the proposed solution considers different energy sources and it improves the duty cycle technique. As of drawbacks, it may fail to manage the network traffic, due to lack of a more general view of the network, its resources, and its nodes.

[Correia et al. 2016] addresses the energy consumption issue, alongside with the traffic congestion issue in IoT. They propose a solution that uses proxies and caches, aiming to reduce energy consumption and to use the bandwidth more effectively. Though, this work does not consider different energy sources, either event classification.

Based on the state-of-the-art analysis, it is possible to observe that current works are not designed to aim for dynamic adjustment of IoT groups. The problems approached often seek to gather data provided by a group of nodes or aim on how to communicate with a group using as few messages as possible. However, important issues are not addressed, such as excessive and redundant data output. Also, the reliability of the network is left aside. Finally, the state-of-the-art does not discuss how to balance the energy used by renewable and not renewable sources.

3. Self Adjusting Group Communication for Dense IoT Networks

This section introduces SADIN, an IoT solution that aims to improve group communications. It contemplates the providence of more efficient use of the network's resources. SADIN allows adjustments in the notification interval of the observation, the nature of the notification (confirmable or non-confirmable), and the number of observers monitoring an event. SADIN takes into consideration the event relevance level, the type of energy source, the percentage of active observers, the current loss rate.

A Fuzzy Decision System makes the decisions and sends them back as instructions to the nodes. We chose the fuzzy logic for the decision making because we consider time-variant conditions and the knowledge of all factors is difficult to obtain precisely. In this way, fuzzy logic offers automated decisions about communication settings. The fuzzy sets and membership functions were defined based on an exploratory analysis.

SADIN was designed as a software layer to provide a self-adjustment capability for a network of IoT devices. The IoT nodes keep communication with SADIN, by sending information about the events via CoAP. They act as the server side of the application, sending data about selected resources to the clients. When SADIN receives a

message, it computes and stores it. Then, the supposed adjustment settings of the network are updated, achieving more efficient use of its resources. Hence, SADIN uses a group of monitoring parameters as input for the fuzzy decision system. The output is a set of adjustment parameters, as shown in Figure 1.

To present the details of SADIN, this section is divided as follows: Section 3.1 presents the monitoring parameters and Section 3.2 shows the Adjustment Parameters. Section 3.3 and 3.4 describes the fuzzy rules and the implementation, respectively.

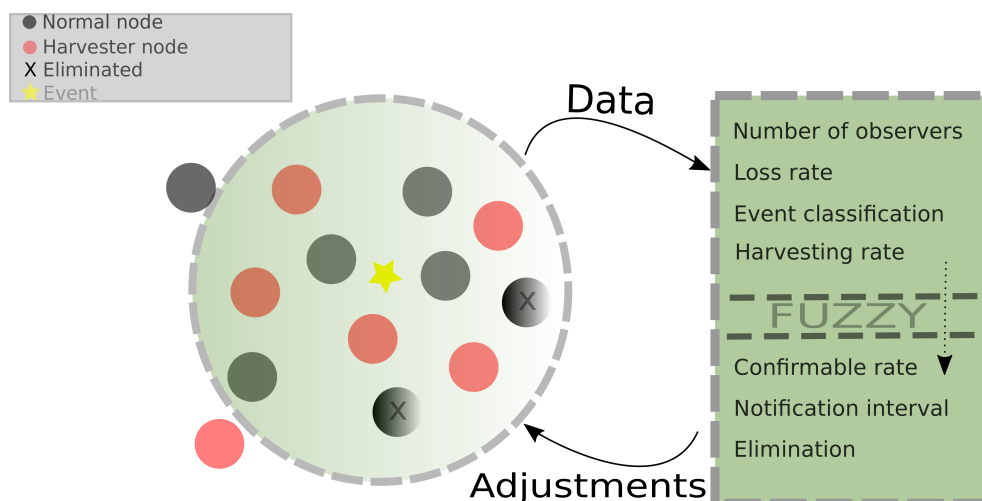


Figure 1. SADIN Overview

3.1. Monitoring Parameters

SADIN defines a time threshold to indicate whether or not a given IoT device is still observing a particular event. Therefore, SADIN compares the registered node's timestamp with the current system's timestamp as soon as a notification arrives. It checks the predefined time threshold to determine which nodes are observing the event. Additionally, it determines which nodes interrupted the observation or never detected it at all. The number of observers can directly influence the amount of generated data. Hence, SADIN acts upon that through its instructions.

The representation of the number of observers divided the parameter into four groups: low, low-mid, high-mid, and high. This value represents the ratio of nodes observing a single event. SADIN calculates this value by dividing the total number of observers by the total number of nodes in the environment.

The rate of traffic loss is one of the core parameters that influence the decision to increase or decrease the confirmable message rate. In this sense, SADIN periodically computes the packet loss ratio. In order to obtain this value, SADIN checks the MID of each incoming message, updating the loss rate of each node accordingly. The fuzzy system classifies the normalized loss rate into the following groups: low, medium, and high, as presented in Figure 2(b).

SADIN considers the type of energy source of each IoT device, *i.e.*, nodes with renewable or renewable batteries. Harvester nodes have renewable batteries, so they are assigned to take more workload for the group, allowing non-harvester nodes to use less of

their limited energy reserve. Figure 2(c) shows the harvesting rate membership function, classified into three fuzzy sets, namely low, medium, and high. SADIN normalized this value in the interval $[0, 1]$ by calculating the total number of nodes observing the event divided by the number of nodes executing energy harvesting.

The critical level of the event is the core parameter of SADIN. That is because the event level directly affects all the system's decisions, *e.g.* controls how much data the network is producing and which messages must be more protected against loss based on the sensing relevance. So, each IoT device that is observing an event assigns a numeric value to represent the relevance level for said event. This classification of the event level ranges from 1 to 3, indicating low, medium, and high event relevance, respectively. Figure 2(d) illustrates the membership function for the event relevance.

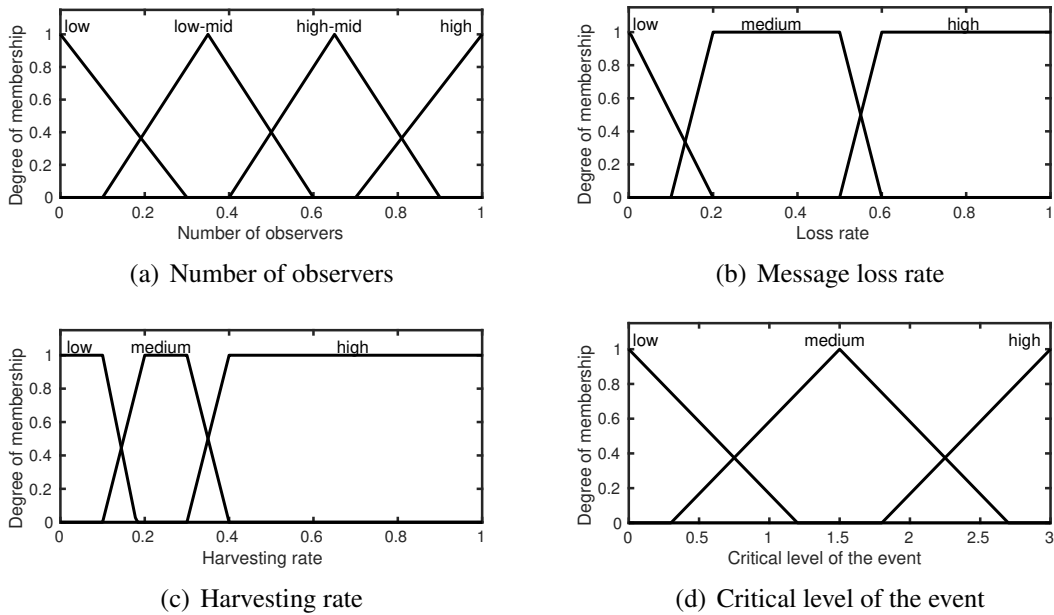


Figure 2. Fuzzy Inputs Membership Function

3.2. Adjustment Parameters

SADIN considers the four monitoring parameters defined above to adjust the group communication, namely the number of devices observing an event, transmission frequency, and type of notifications. SADIN computes the adjustments parameters based on fuzzy logic decisions. These adjustments derivate from the fuzzy membership functions.

SADIN adapts the network's confirmable rate, which is the ratio between the total number of transmitted message and the number of confirmable messages. In this sense, we assume a lowly relevant situation should tolerate higher traffic loss. On the other hand, in case of a critical event, the confirmable rate should be high. Figure 3(a) shows the membership function for the confirmable rate of the notification messages. This function splits itself into four groups, low, low-mid, high-mid and high. These will determine at which rate confirmable messages the traffic will be sent from the servers.

SADIN adjusts the notification interval, which, namely, is defined as the time interval between two consecutive notification messages sent by the IoT device. It has an

important impact on energy consumption, as well as in avoiding traffic congestion. We defined the notification interval into four groups, short, short-mid, long-mid, and long. This membership function is shown in Figure 3(b).

Finally, SADIN determines if it should eliminate a node from the observation group. This decision is motivated to save energy and to avoid congestion. Introducing the membership function for this parameter is not necessary, since this output is classified as Do Eliminate if the fuzzy system gives an output greater than 0.5 and the possible interval for this output goes from 0 to 1. This output classifies as Do not Eliminate for all other output values.

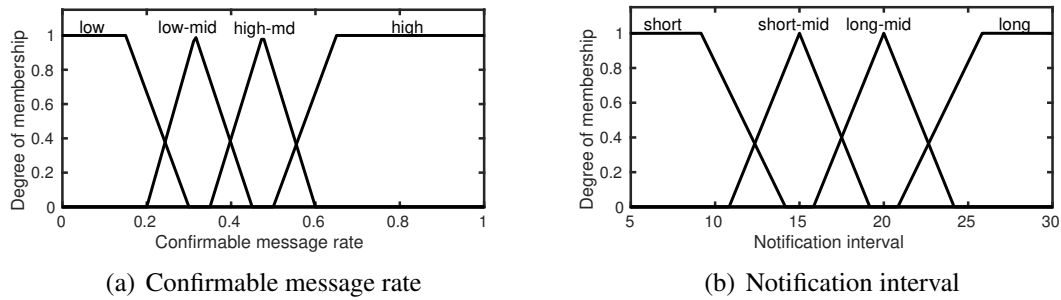


Figure 3. Fuzzy Outputs Membership Function

3.3. Fuzzy Rules

After the definition the membership functions, we had to define a set of fuzzy rules, which are present in Table 1. The elimination of a node from the observation is a good example, specifically, in the case which the number of observers is high and most of them are powered by a traditional battery, as seen in the Rule (R14). SADIN receives the decision to eliminate a node and chooses which node should exit the observation. This selection considers the individual loss rate of that node and its energy source. The removed node is the normal battery one with the highest loss rate.

Table 1. Fuzzy Rules.

RULE	IF	THEN
R1	Number of observers is high and rate of harvesters is low	Eliminate node
R2	Number of observers is high and loss is high	Interval is long
R3	Number of observers is low-mid and loss is medium	Interval is short-mid
R4	Number of observers is high-mid and loss is medium	Interval is short-mid
R5	Event level is low	Interval is long and confirmability is low
R6	Event level is medium	Interval is short-mid confirmable is low-mid
R7	Event level is high	Interval is short and confirmability is high
R8	Number of nodes is not high and loss is high	Confirmability is high
R9	Number of nodes is not low-mid and loss is medium	Confirmability is low-mid
R10	Number of nodes is not high-mid and loss is medium	Confirmability is high-mid
R11	Number of nodes is not high and rate of harvesters is not low	Do not eliminate node and confirmability is high-mid
R12	Number of nodes is high-mid and loss is low	Interval is short-mid
R13	Number of nodes is low	Do not eliminate node
R14	Number of nodes is high and loss is high and event level is not high	Eliminate node

Once we defined the fuzzy rules, we needed to insert them into the Fuzzy Logic Controller as part of the Decision Fuzzy System. After the configuration of the rule set, SADIN became able to calculate the degree of membership of each input, resulting in a dynamic and on-the-fly control of the output.

3.4. Implementation

We considered Contiki, CoAP, and Californium to implement SADIN, as shown in Figure 4. Specifically, Contiki is the operating system for low power and tiny IoT devices. We had to modify Contiki for adequate interaction with SADIN, *i.e.*, how each node is susceptible to adjustments in the communication setting through instruction messages. CoAP is a lightweight application protocol for IoT and has important mechanisms to reduce network redundancy and energy. In this way, CoAP users register interest in specific data or measurements produced by CoAP servers. Afterward, the CoAP server sends a notification message to the interested entities whenever an event occurs.

We implemented the CoAP client using the Californium platform, which is an open-source CoAP implementation. Californium allowed the management of all the nodes, collecting information about events, at the same time, as well as sending the instruction messages to adjust the network communication parameters. Contiki permitted configuring each node to be able to change some system parameters in real time. The core functionalities of SADIN reside in Californium, namely event and fuzzy modules.

An event module was necessary, so we implemented it at Californium, which computes and stores all relevant monitoring parameters information about each IoT devices, such as current loss rate, number of active observers, harvesting rate, and event classification. At each incoming message, SADIN updates its the database. Accordingly, the Fuzzy module computes new decisions to refresh the Adjustment parameters for the group communication considering the most recent information from the event module. SADIN piggybacks the adjustment parameters in an acknowledgment message sent by the CoAP client.

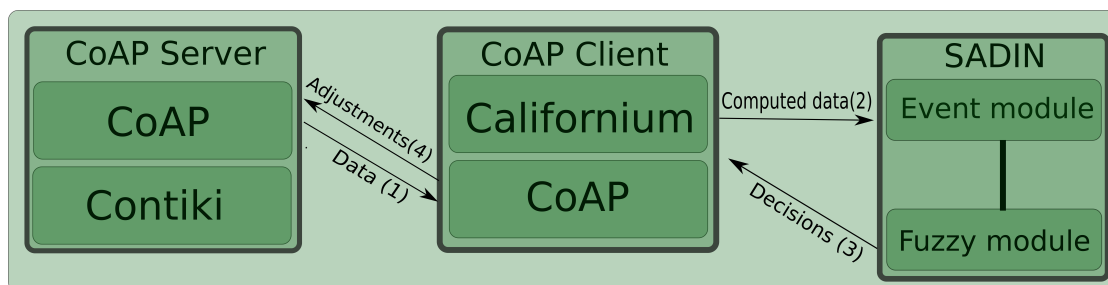


Figure 4. SADIN components

4. Evaluation

SADIN ran in a simulation environment, concerning the comparison between it and the standard CoAP. These simulations resulted in data about energy consumption, traffic loss rate, notification interval and number of eliminations. The remaining of this section is structured as follows: Section 4.1 presents the scenario description, while Section 4.2 shows the obtained results.

4.1. Scenario Description

We implemented SADIN on Cooja simulator, which is a tool that enables the emulation of nodes and simulates a wireless network. Cooja supports many architectures for IoT nodes, where we considered a group of Wismote nodes deployed in an area of $100m^2$ for

our simulation. The duration of the simulation was 30 minutes, and we repeated the tests 30 times to provide results with a reliable confidence level. The tests ran on two density levels: 20 nodes and 40 nodes in 100 square meters, which is denoted by as $p=0.2$ or $p=0.4$, respectively.

The event occurrence follows a Poisson distribution with $\lambda = 9$, where each event has a fixed duration of 5 minutes. We tested SADIN in three different scenarios, where each considered only one of the relevance levels. That is because we wanted to isolate the network's capability to disseminate data upon each situation.

Our evaluation regarded a dataset with indoor light energy measurements collected by Columbia University, which considers a small solar panel with $100cm^2$ and efficiency of 20%. In our simulation scenario, 50% of the deployed nodes are execute energy-harvesting by collecting solar energy. We integrated The Kinetic Battery Model (KiBaM) into Contiki OS to consider the non-linear behavior of the batteries [Riker et al. 2017]. The Powertrace tool provided the energy consumption data. Therefore, we measure the energy consumption with satisfactory accuracy in the following states: Transmit, Receive, Idle Listen, Active CPU, and Low Power CPU.

4.2. Results

Figure 5(a) and 5(b) show the average energy consumption by each node in different event relevance levels for densities of $p=0.2$ and $p=0.4$, respectively. These results show that SADIN reduces energy consumption in both network densities. Regarding the events with low relevance, the nodes coordinated by SADIN consumed an average of 17% less energy than standard CoAP. When the events become more critical, SADIN makes more intense use of its resources to send information to the client, causing energy consumption to increase. Although, even in the case of high relevant events, SADIN continues to consume less energy than the standard CoAP. One of the factors that contribute to the energy consumption reduction is the adjustment introduced to regulate the notification interval, as well as the dynamic control of confirmable messages.

The loss rate is another relevant network performance metric, which is present in Figure 5(c). There is a noticeable difference in loss rate for the two evaluated network scenarios, due to the impact that the density of the network has on itself. In both situations, When the events get more critical, SADIN adjusts the network's behavior. These changes in behavior encompass node elimination, the confirmable message rate, and the notification interval. The results, as expected, showed that the losses remain the same or decrease accordingly with the event's critical level. Therefore, SADIN had a positive impact the group management in dense IoT environments.

Figure 5(d) presents the average notification interval between consecutive messages for different event relevance level. It is possible to see a meaningful difference between the notification interval of the lowly and highly relevant events. When an event has a low importance classification, each node takes an average of 20 seconds to send a notification, compared to 9.5 seconds in the higher event level. It is a positive result because SADIN delivers more messages related to critical events with less delay. Standard CoAP offers no adaptability in terms of communication interval, which means that during all the network operation, all the messages are communicated with the same frequency, regardless of the relevance level of the data.

The elimination of nodes from the groups aims to save the limited energy of non-harvester nodes. During the simulations, when an event is classified as not relevant, about 13 and 8 nodes were eliminated, considering $p=0.4$ and 0.2 , respectively (see Figure 5(e)). Although, when classified as critical, SADIN did not remove any nodes from observation.

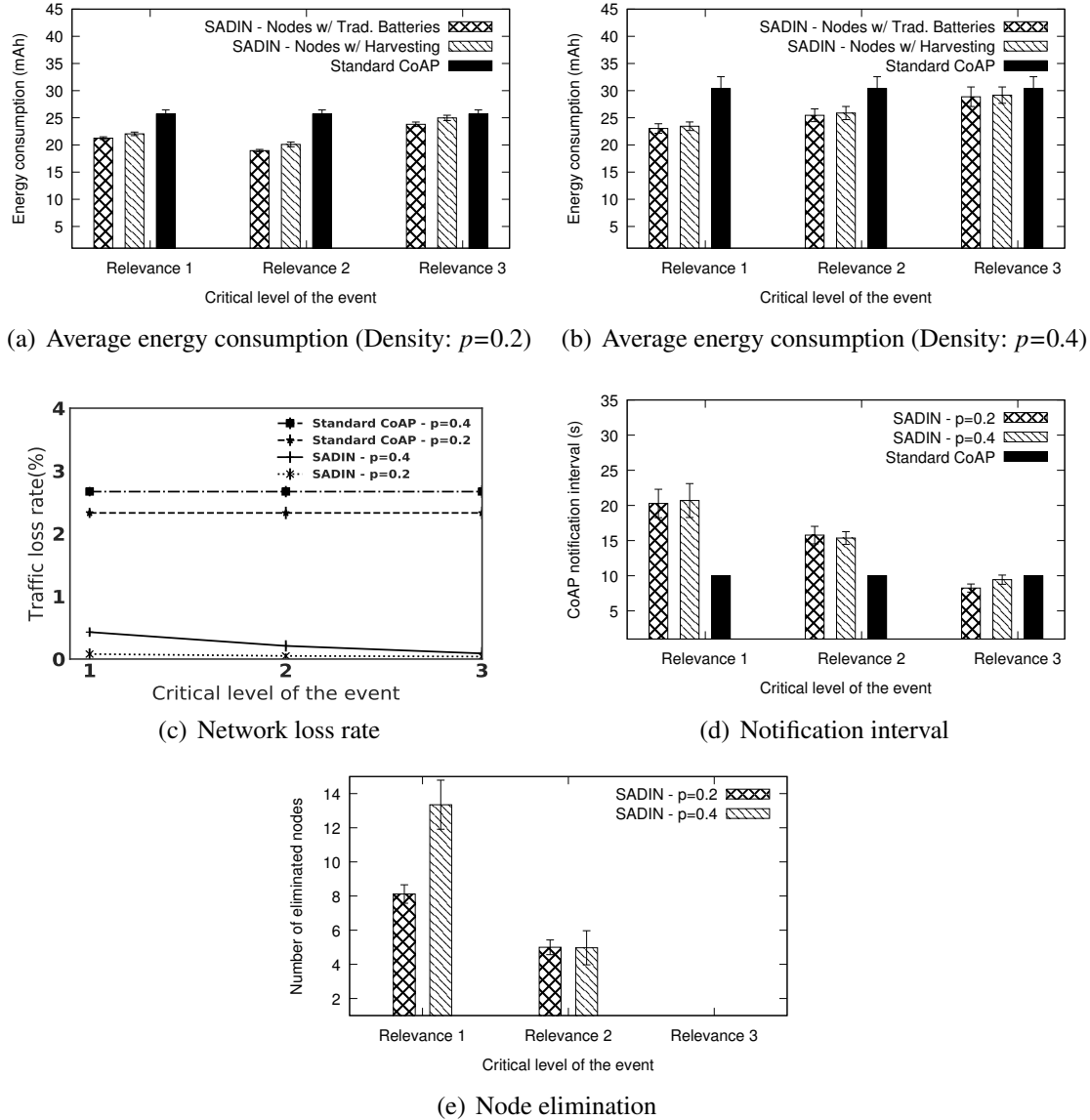


Figure 5. Simulation Results

5. Conclusion

Dense IoT scenarios are an inevitable future, such as smart cities and sensor networks. In this case, IoT has problems in terms of energy consumption, traffic congestion, and redundant data transmission. The recent IoT researches have already addressed some of these obstacles. However, traditional solutions do not provide dynamic control of the communication settings while the users receive notifications from a group of IoT nodes. Also, they do not consider the heterogeneity of energy sources.

SADIN aims to monitor a set of parameters that indicate the current state of the network. Based on these monitored parameters, SADIN makes decisions on how to adjust the group's communication settings. Simulation results showed that SADIN reduces message loss when important events happen. Besides, SADIN decreases energy consumption and eliminates nodes when events are classified as not relevant. By this means, the network is capable of adjusting according to diverse situations and become more efficient.

References

- Betzler, A., Gomez, C., Demirkol, I., and Paradells, J. (2016). Coap congestion control for the internet of things. *IEEE Communications Magazine*, 54(7):154–160.
- Cerwall, P., Jonsson, P., Möller, R., Bävertoft, S., Carson, S., and Godor, I. (2015). Ericsson mobility report. *On the Pulse of the Networked Society. Hg. v. Ericsson*.
- Correia, N., Sacramento, D., and Schütz, G. (2016). Dynamic aggregation and scheduling in coap/observe-based wireless sensor networks. *IEEE Internet of Things Journal*, 3(6):923–936.
- Costa, D. G. and Guedes, L. A. (2013). Exploiting the sensing relevancies of source nodes for optimizations in visual sensor networks. *Multimedia tools and applications*, 64(3):549–579.
- Islam, H. M. A., Lagutin, D., Ylä-Jääski, A., Fotiou, N., and Gurtov, A. (2019). Transparent coap services to iot endpoints through icn operator networks. *Sensors*, 19(6).
- Jaramillo, R., Quintero, A., and Chamberland, S. (2015). Energy-efficient mac protocol for wireless body area networks. In *2015 International Conference and Workshop on Computing and Communication (IEMCON)*, pages 1–5.
- Jarvinen, I., Raitahila, I., Cao, Z., and Kojo, M. (2018). Faser retransmission timeout and congestion control mechanism for coap. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7.
- Orsino, A., Araniti, G., Militano, L., Alonso-Zarate, J., Molinaro, A., and Iera, A. (2016). Energy efficient iot data collection in smart cities exploiting d2d communications.
- Rani, S., Talwar, R., Malhotra, J., Ahmed, S. H., Sarkar, M., and Song, H. (2015). A novel scheme for an energy efficient internet of things based on wireless sensor networks.
- Riker, A., Curado, M., and Monteiro, E. (2017). Neutral operation of the minimum energy node in energy-harvesting environments. In *2017 IEEE Symposium on Computers and Communications (ISCC)*, pages 477–482.
- Riker, A., Fonseca, C. M., Curado, M., and Monteiro, E. (2018). Energy-efficient multi-group communication. *Transactions on Emerging Telecommunications Technologies*, 29(3):e3232.
- Sekhar, S. (2014). Performance evaluation of advanced congestion control mechanisms for coap. *Sensors*, 2015, 15.