Multiple-agent opportunistic data gathering in IoT networks using group selection

Edvar Afonso and Miguel Elias M. Campista
Universidade Federal do Rio de Janeiro - GTA/COPPE/UFRJ), Rio de Janeiro, Brazil
{edvar,miguel}@gta.ufrj.br

***Abstract*—The Internet of Things (IoT) is based on data collection for future processing and decision-making. In multihop Low-Power and Lossy Network (LLN) scenarios, it is important to collect data from sensors and IoT devices with efficient energy consumption. This paper explores the concept of multi-mobile agents (MMA) combined with proactive caching to collect data in an IoT network. In the proposed architecture, the group of itineraries is computed by a mechanism that uses the Hamming distance technique to partition a sensor network into a number of data-centric clusters. In addition, the data collection mechanism is modeled as a knapsack problem that improves the energy efficiency with MA payload variation. The simulations show lower energy consumption from network devices as the payload and the number of MMA increase.**

***Index Terms*—IoT, data gathering, multi-mobile agents**

I. Introduction

The Internet of Things (IoT) is currently transforming the world with a remarkable growth in the number of connected devices. This technology uses data, turning it into information for decision-making, both by humans and automated systems from a variety of areas, such as in Industry 4.0, Smart Cities, Smart Homes, and Smart Agriculture [1]–[4].

IoT devices have hardware constraints with known battery limitations, making energy consumption a considerable concern [5], [6]. Communications, however, are at the top position for energy consumption, even considering the various processing tasks that IoT devices need to perform. If we consider multi-hop communications, this becomes even more critical, as the traffic generated by any node requires multiple wireless medium accesses until it finally reaches the destination. Thus, methods to increase the efficiency of energy consumption in sensors and IoT devices have been the subject of a great deal of work in recent years [7]–[10]. In multihop scenarios, the considerable effort for data gathering and forwarding can lead to a significant increase in latency, energy consumption, and congestion over wireless links. To overcome these issues, the Mobile Agent (MA) paradigm has been revisited as an efficient alternative for data collection in IoT [11]–[14].

A MA is a computer code that is sent to the network to perform tasks on network devices such as data fusion and data aggregation. In a network with many devices it is a good practice to use multi-mobile agents (MMA) for data gathering tasks. This is a strategy to make proposals more efficient in terms of scalability. In order to send MMA to the network, it is important to first divide the network into groups. Thus, each of the MAs can perform the data collection process in a subset of the network devices. This implies fewer data collected by each MA and smaller payloads for MAs. This saves energy from devices in data transmission through the wireless network [15], [16]. To complete the assigned tasks, the MA must visit different network devices following a predetermined itinerary calculated using an optimized algorithm (Itinerary Planning) [15], [17]. The MA itinerary consists of source and intermediate nodes. Source nodes are defined as those that contain the information required for the MA to gather and process. In contrast, the intermediate nodes serve the purpose of forwarding the MA across the multihop network.

This paper proposes Agent-MultiCluster, a network architecture that combines MMA with opportunistic data collection and cache deployment at the network gateway to reduce energy consumption at network devices. Agent-MultiCluster is an evolution of Agent-Knap [18], [19], our first proposal, which relies on a single MA to collect data from a group of source nodes (SNs). An improvement identified for Agent-Knap was when the MA had to cross a long itinerary and collect data from many devices. This situation results in a significant increase in the MA payload size. The larger the MA payload, the more transmission delays, energy consumption and loss of reliability occurs. To address this issue, Agent-MultiCluster sends multiple agents to the network simultaneously. This strategy results in short itineraries for each agent. Consequently, the individual size of each MA payload and the overall energy consumption is reduced. Our proposal computes multiple static itineraries composed of source and intermediate nodes, and employs a selection mechanism that prioritizes data acquisition based on a precomputed data utility. In Traditional Multi-Mobile Agent (TMMA) proposals, each agent migrates from one source node to another, following the sequence of nodes recorded in the itinerary. In this process, the MMA ignore the data available on intermediate nodes. In Agent-MultiCluster, we use groups of two or four simultaneous MAs that collect data from source nodes and opportunistically collect data from intermediate nodes. In order to process the itineraries, the network is divided into groups of devices and the K-means clustering algorithm is used for this purpose. The Agent-MultiCluster uses Weighted Set Cover for source node selection, Christofides and Dijkstra for itineraries computation and 0-1 knapsack algorithm to select the data type to be

978-1-7281-8086-1/20/$31.00 ©2020 IEEE

opportunistically collected at intermediate nodes.

The objective of our research was to increase the lifespan of IoT networks and to reduce the length of the data collection task. In order to achieve our objective, we proposed a new approach in LLNs, which took into consideration the criterion of an appropriate number of MAs, the criterion of the appropriate grouping of SNs, and finally the criterion of the optimal itinerary followed by each MA to visit all its SNs

We compare the Agent-MultiCluster with the TMMA, i.e., without using opportunistic data collection at intermediate nodes. Simulation results reveal a substantial reduction in energy consumption and network traffic. We summarize our main contributions as follows:

- We propose Agent-MultiCluster using K-means to partition the IoT network into a number of data-centric clusters before the itinerary computation. This approach guarantees itineraries that traverse groups formed by devices that are geographically close to each other;
- We extended the opportunistic data gathering used in Agent-Knap [18], [19] to MMA scenarios and to improve the total energy efficiency in data gathering process;
- We propose an approach to increase the efficiency in managing the MA payload size. We add a control mechanism to the gateway for this purpose.

This paper is organized as follows. Section II overviews the related works. Section III presents Agent-MultiCluster. In the following, Section IV describes the simulation environment and the results achieved. Finally, Section V concludes this work and presents future directions.

II. Related Work

Optimized data collection in LLN networks has been the focus of many recent works. The use of mobile agents for data gathering in LLN has been widely used, and a substantial body of research has demonstrated that employing multi-mobile agents (MMA) for itinerary planning outperforms single-mobile agents in large-scale networks [17], [20]–[22].

Bendjima et al. [23] address the challenges of load balancing between MAs, including how to determine the number of agents needed and how to group source nodes. Their main contribution is a clustering algorithm that automatically determines the number of clusters. The proposed method is robust and considers parameters such as the density of nodes, communication and energy costs, and the data size in each sensor. The goal of their proposal is to extend the lifespan of sensor networks and to reduce the duration of the data collection task. Similarly to Agent-MultiCluster, their solution also relies on the gateway as a central element responsible for executing computationally intensive tasks.

Karthik et al. [17] propose a structured sequence of algorithms to calculate multiple itineraries and gather data using MAs. Their approach uses Fitness-based Fuzzy C-Means (FFCM) for cluster creation and the Crossover Mutation based Firefly Algorithm (CM-FFA) for Cluster Head (CH) selection. The main goal is to reduce energy consumption in WSN. For MA migration and Data Gathering (DG), they employ an optimal itinerary planning-based algorithm based on Brownian Motion-Based Flower Pollination Algorithm (BMFPA). The proposal is evaluated through simulations using metrics such as time delay, packet drop ratio, and energy consumption. Their results indicate superior performance compared with existing DG algorithms for WSN with multiple MAs.

In contrast to Agent-MultiCluster, neither of these studies addresses opportunistic data collection on intermediate nodes while the MMAs traverse their itineraries. To the best of our knowledge, the use of the clustering algorithm for calculating the itinerary planning and the use of MMAs for opportunistic data gathering is an original contribution of our work and represents its main innovation.

III. Data Gathering Using Agent-MultiCluster

This paper proposes Agent-MultiCluster, a network architecture that combines MMAs with opportunistic data collection and cache deployment at the network gateway to reduce energy consumption at network devices. The proposal is developed for networks with fixed devices or sensors. We consider a network composed of one gateway and multiple IoT devices randomly deployed in an Area of Interest (AoI). Each device has multiple sensors to collect data from different types (for example: temperature, pressure and humidity). Each device stores only the last sample of each data collected by each sensor. All devices can be considered as a source node that provides the requested data. The proposed architecture is centralized at the gateway, our central entity, which runs itinerary planning and cache management functions. In addition, the gateway is in charge of processing all requests and responses to external clients.

The network operation relies on data requests from external Internet clients sent to the gateway. Each request has information about the desired data, the desired AoI, and the desired maximum data expiration time. Depending on the availability and freshness of the data in the cache, the gateway decides whether a new data gathering round is needed, that is, if it needs to dispatch a new group of MMA. In the proposed architecture, data can be collected from any network device. Thus, at system startup, each device must register within the gateway and inform the types of data it can provide, as well as its coordinates, its neighboring devices, and the size (in bytes) of each data it can provide. We assume that, after system start-up, the gateway has information about the network topology, including the geographic position of all devices in order to compute the MAs itineraries.

A. Data gathering model

The main goal of our proposal is to save energy from IoT devices when multiple mobile agents are used to collect data. The data collected from each device is stored in the gateway cache. Each stored data has a timer and, while valid, helps to avoid a new data collection round if that data is requested. This mechanism reduces the number of requests sent to the network, allowing a reduction in energy consumption. The opportunistic data gathering has a key role in the process, as it brings more data back to the gateway, increasing the freshness of data stored in the cache.

The MA packet format has four fields [24]: the MA ID, for unique identification of the mobile agent; the itinerary; the processing code, for data manipulation; and the payload, reserved for the data collected and processed by the MA. The payload has a maximum size (P), in bytes, and has two fields, G and C. G is reserved for the data collected from source nodes. C is used to carry the data opportunistically collected from intermediate nodes. Even though C is variable, it is upper-bounded by a maximum value determined by the gateway. G carries only the type of data requested from the client in the current data gathering round. Meanwhile, C carries all sorts of data from intermediate nodes visited in the itinerary. Figure 1 details the format proposed for the payload. Considering the example, G is determined by the data type k_1 collected at four different devices with IDs 2, 3, 8, and 12. Hence, G contains k_1^2, k_1^3, k_1^8, and k_1^{12}. C, on the other hand, is determined by four different data types, k_3, k_5, k_2, and k_4 opportunistically collected at four different intermediate devices with IDs 1, 7, 8, and 12, i.e., k_3^1, k_5^7, k_2^8, and k_4^{12}.

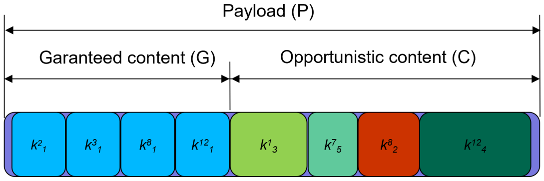

Fig. 1. MA payload P with guaranteed (G) and opportunistic data (C).

B. Clustering process

In the Agent-MultiCluster proposal, K-means is used to partition the network. As mentioned in Section III, the Agent-MultiCluster proposal works in network topologies with fixed devices. After the network initialization, the coordinates of all devices are stored in the gateway. With this information, the gateway uses the K-means algorithm to partition the network into clusters based on the spatial location of all devices.

Spatial coordinates are chosen as the parameter for configuring K-means clustering because it is crucial to establish itineraries between spatially close devices. This approach prevents long itineraries from being calculated, avoiding unwanted growth in the MAs' payload size and preventing these agents from crossing many devices in the network and thus consuming energy inefficiently.

C. Source nodes group selection

When the gateway receives a client request, it first performs a cache lookup for updated data to provide a fast and complete response to the client. A complete response comprises unexpired data of a specific type from multiple devices covering the entire AoI. If the cache does not provide a complete response, the gateway starts a data collection by sending multiple MAs to the network devices. The gateway must collect fresh data from devices to complete the non-expired information in the cache. These particular devices are called source nodes. Thus, when a set of MAs is sent to the network by the gateway, a group of source nodes must be selected beforehand. The selection of source nodes determines the group of devices the MAs must visit to complete the data in the cache. The group of source nodes is selected, taking into account the best possible AoI coverage, which guarantees a complete response regarding the data requested by the client. Hence, the selection of source nodes guarantees that the data provided by those devices is enough to provide a complete set of data for the entire AoI.

We model the source node selection mechanism as a classic coverage problem, the *Weighted Set Cover*. In our model, each sensor $n_i \in \mathcal{N}$ is associated with a coverage subarea of the AoI. A weight w_{n_i} is also associated with each sensor to assign priority in the selection process. The freshness related to each data stored at the gateway cache determines w_{n_i}. The older the data of a given sensor stored in the cache, the higher the priority in the *Weighted Set Cover* problem. Regarding complexity, the *Weighted Set Cover* is NP-hard. The heuristic used in our proposal has time complexity $O(\log n)$.

D. Itinerary planning

Before sending the set of MAs to the network, the gateway must calculate the itineraries. This process is carried out for each cluster of devices, and the itineraries form a closed loop starting and finishing at the gateway. Each loop must visit the subset of source nodes, selected by the gateway, that belongs to its respective cluster. Figure 2 illustrates an example with three defined itineraries. The source nodes appear in red in the figure and were previously selected by the gateway using the selection method discussed in Section III-C. Considering each itinerary, the opportunistic data is obtained from possible intermediate nodes connecting two consecutive source nodes. The proposal manages the MA payload.

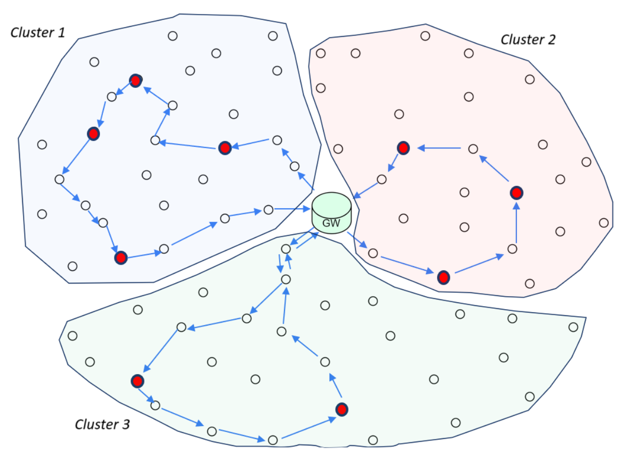

Fig. 2. Three defined itineraries, one for each cluster.

The subset of source nodes per cluster is the main input to calculate the itinerary. In this step, the algorithms of *Dijkstra* and *Christofides* compute each optimized itinerary. The Christofides algorithm is a heuristic used to solve the

Traveling Salesman Problem (TSP), which is the kind of problem to be solved for the MA itinerary computation. The output of the algorithm is a sequence of nodes that must be visited. Since these nodes may not be directly connected, the final itinerary is determined after computing the paths between consecutive source nodes and their respective costs using the *Dijkstra* algorithm. The Christofides algorithm is a TSP heuristic that, in the worst case, guarantees a solution at most 3/2 worse than the optimal one [25]. The heuristic used in our proposal has time complexity $O(n^3)$. This process to calculate the itinerary is repeated for each cluster.

E. Payload computation

The next step is to define the data that must be collected in the intermediate nodes of each itinerary, $\mathcal{I}$, i.e., the data that will be selected for the opportunistic data gathering. Each data available in an intermediate node has a size registered in the gateway at system startup. Thus, to efficiently execute the process, the most valuable data must be collected from the intermediate nodes, considering the payload size restriction of each MA and the size of each data type, in bytes.

This problem can be modeled as a combinatorial optimization problem *Knapsack* which is defined as follows: given a backpack with a defined weight limit capacity and a set of items, each of which has a weight and value. The solution to the *knapsack* problem is determined when a group of items is selected to be included in the backpack, so that the total weight is less than or equal to the capacity of the backpack and the value is as high as possible. In the Agent-MultiCluster, the capacity of the backpack is represented by the MA payload (P) size, in bytes, the weight of each item is defined by the data size, in bytes, and the value of each item is represented by a calculated utility that is related to the data priority.

Let $A_{cache}^{k_j}$ be the corresponding area covered with non-expired data stored in the cache for a type of data k_j (e.g., temperature). For the knapsack problem, the priority computed for a specific type of data is numerically proportional to the contribution it provides to complete $A_{cache}^{k_j}$ in the gateway´s cache. Thus, the smaller the $A_{cache}^{k_j}$, the higher the priority for collecting k_j in the current round. Equation 1 presents the priority p_{k_j} computation for a given type of data k_j.

$$p_{k_j} = AoI - (AoI \cap A_{cache}^{k_j}). \tag{1}$$

The payload computation determines the opportunistic types of data collected along the itinerary. These selected data types are collected from all intermediate nodes that have data of that type available and aggregated in the MA. Only data of the same type is aggregated, and data of different types are concatenated in the C space of the payload. The knapsack ensures that the MA payload is optimally populated by the most valuable type of data provided by the devices at the itinerary $\mathcal{I}$ without violating C.

F. Mobile Agent initial size management

Another feature added as an improvement on our previous proposal [19] is the management of the initial size of the MA. In Agent-MultiCluster, after the gateway calculates the itinerary, a check is made to verify if all the devices on that itinerary have already been visited by one of the MAs in previous rounds. If all the devices on the itinerary have already been visited, the MA is sent with less processing code in its payload - because the processing code was already distributed in a previous round - and, consequently, a smaller initial size. This process helps to reduce energy consumption in the network devices.

IV. SIMULATIONS

In this section, we evaluate the performance of the Agent-MultiCluster throught simulations. We have implemented a simulator using Python and the *NetworkX* package.

A. Simulation setup

The topology consists of fixed devices randomly deployed in an area. The density of the devices is always enough to guarantee complete coverage of our AoI. Each device has a fixed number of sensors and, consequently, can collect different types of data.

We simplify the analysis by considering a few assumptions. We consider that all devices have the same initial battery level and that the different data types have the same size. In addition, the sensing range of each sensor is the same for the entire network, and all devices have four different sensors, one for each type of data. Our application considers Internet clients sending messages to the gateway. Such requests are received with an interarrival rate following a Poisson distribution with $\lambda = 5$.

Each link cost is proportional to the Euclidean distance between devices. Each device can communicate with neighbors within its communication range, and the network gateway has a fixed position in the center of the AoI. Table I shows the parameters used in all simulations. The energy consumption values are the same as those adopted in [26] to evaluate *Tmote Sky* sensors. All the MMA are either executing complete data aggregation of the same content type along the itinerary (factor one). When the MMA performs complete data aggregation, there is no increase in the payload size occupied by collected data of the same type. We considered that the same data type can always be aggregated.

TABLE I
SIMULATION PARAMETERS.

Parameter	**Value**
Number of network devices	300
Number of data types per device	4
Number of Multi-mobile Agents	2 and 4
Device initial energy	20 Joules
Energy Consumption (Transmission)	9.72 μJ/byte
Energy Consumption (Reception)	8.22 μJ/byte
Area of Interest	100 m x 100 m
Gateway coordinates	(50 m, 50 m)
Mobile Agent initial size	128 or 1,024 bytes [27], [28]
Sensed data size	20 bytes
Link communication loss	0 (lossless) or 15% (lossy)
Expiration time for cached data (s)	30, 60, 90, 120, and 180

We compare our proposal with the Traditional Multi-mobile Agent (TMMA) approach. TMMA collects data only from the subset of source nodes needed to completely cover the AoI, ignoring the data available at intermediate nodes. Simulations computed two different metrics: Remaining Energy and Cache Hits. All simulations are performed for a total period of 2,000 seconds each round and a total of 10 rounds per variable parameter.

B. Remaining Energy

In this simulation, we compute the average remaining energy of all nodes after each round. Figure 3 and Figure 4 show the relationship between the average remaining energy of the devices and the expiration time (s) of the data stored in the cache, considering simulations in a network without loss and with loss, respectively. The remaining energy always increases with the persistence of the data in the cache. Also, the remaining energy is always greater when the number of MMA increases from 2 to 4 agents, including in the TMMA simulations. Despite the larger number of agents, clusters built with the k-means algorithm allow shorter itineraries to be calculated for each collection round, resulting in lower energy consumption in the network by each MA.

Concerning the C parameter, the remaining energy fluctuates slightly in the lossless network. Otherwise, in the lossy scenario, it is clear that more space in the MMA payloads to collect data opportunistically always results in more remaining energy in the network at the end of simulations. In all simulations, Agent-MultiCluster presents better results than TMMA. In all scenarios (lossy and lossless network), the most significant Agent-MultiCluster gain appears in the 30-second cache expiration time.

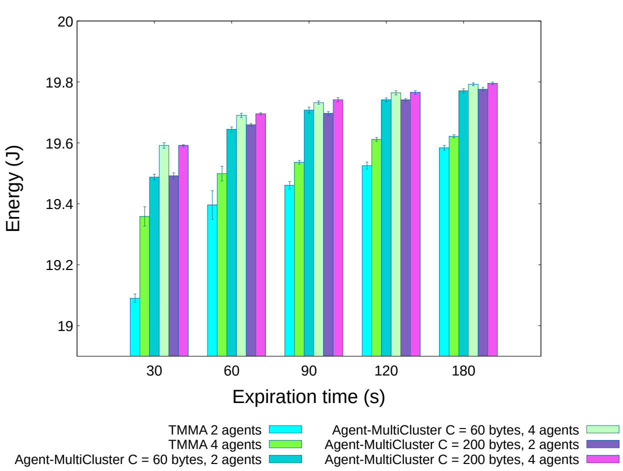

Fig. 3. Average remaining energy for each device in a 300-device lossless network using 60 and 200-byte payload (C) sizes.

C. Cache Hits

The Cache Hit evaluation aims to verify the performance of the cache infrastructure fixed in the gateway, including networks with and without loss. A cache hit occurs when

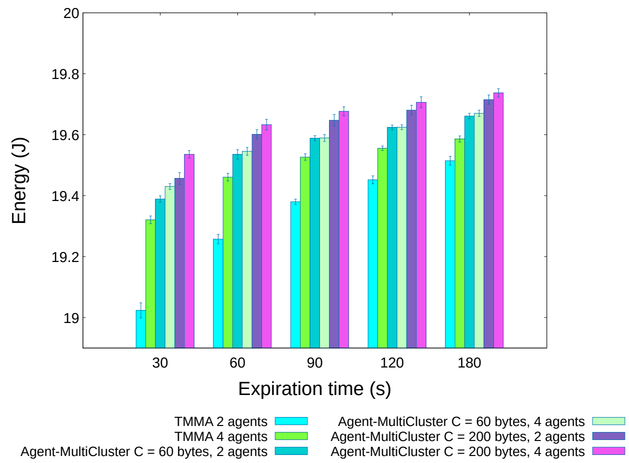

Fig. 4. Average remaining energy for each device in a 300-device lossy network using 60 and 200-byte payload (C) sizes.

the gateway responds to a data request with the information available in its cache, i.e., without dispatching a group of MMA. Similarly to the previous section, we perform an analysis using 60 and 200 byte C sizes. Figure 5 and Figure 6 present the results for a 300-node network. In the Cache Hit simulations, we can see that the number of cache hits with two agents is higher than with four agents, in most cases. However, in the energy results, the remaining energy is lower in the case of two agents. The conclusion is that the itinerary size and the payload growth play an important role in our proposal for opportunistic collection with multiple agents. With two agents, on average, the itineraries are longer in order to collect data that cover the entire AoI.

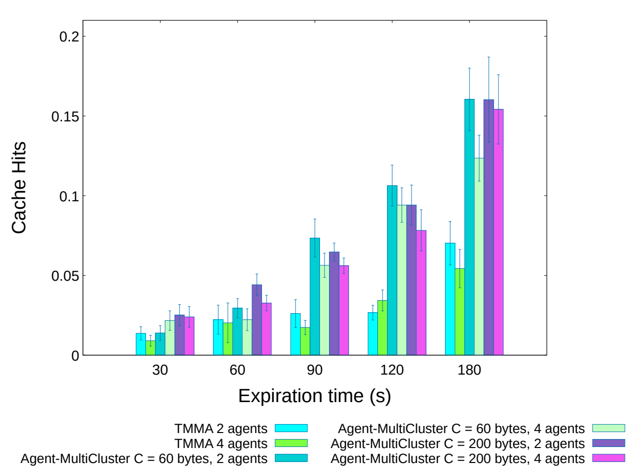

Fig. 5. Proportion of cache hits in lossless network. Experiments consider a 300-device network and 60 and 200-byte payload (C) sizes.

V. Conclusion and Future Work

We proposed the Agent-MultiCluster, a new mechanism for data collection in LLN networks using multi-mobile agents with static itineraries. Our simulation results show the potential

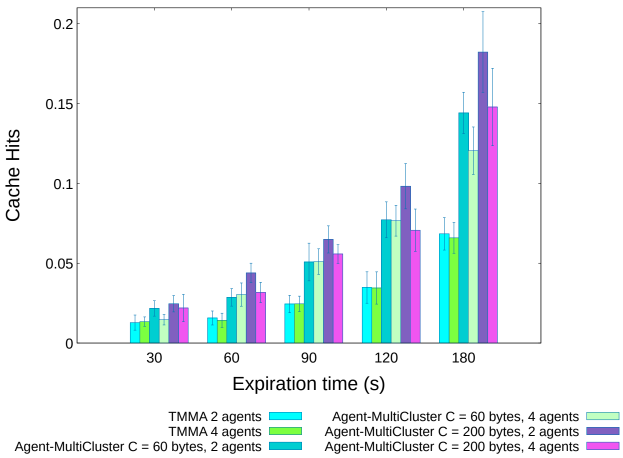

Fig. 6. Proportion of cache hits in lossy network. Experiments consider a 300-device network and 60 and 200-byte payload (C) sizes.

of the opportunistic data gathering mechanism, associated with K-means clustering to reduce the average energy consumption in the network. To our knowledge, this is the first work to use k-means clustering to perform opportunistic data collection with MMA and proactive cache update. As future work, we intend to dynamically change the payload and extend the use of clustering beyond the K-means algorithm.

Acknowledgement

This work was supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) – Funding Code 001 and 88887.954253/2024-00, CNPq (310234/2025-5, 408255/2023-4, and 405940/2022-0), FAPERJ (E-26/200.380/2023), and FAPESP (2023/00673-7 and 2023/00811-0).

References

[1] A. Alotaibi, H. Aldawghan, and A. Aljughaiman, “A review of the authentication techniques for internet of things devices in smart cities: opportunities, challenges, and future directions,” *Sensors*, vol. 25, no. 6, p. 1649, 2025.

[2] K. C. Rath, A. Khang, and D. Roy, “The role of internet of things (iot) technology in industry 4.0 economy,” in *Advanced IoT technologies and applications in the industry 4.0 digital economy*. CRC Press, 2024, pp. 1–28.

[3] L. Liu, A. Wang, G. Sun, J. Li, H. Pan, and T. Q. Quek, “Multi-objective optimization for data collection in uav-assisted agricultural iot,” *IEEE Transactions on Vehicular Technology*, 2024.

[4] R. Bensaid, N. Labraoui, H. Saidi, and H. Bany Salameh, “Securing fog-assisted iot smart homes: A federated learning-based intrusion detection approach,” *Cluster Computing*, vol. 28, no. 1, p. 50, 2025.

[5] Z. Cao, W. Gao, Y. Fu, and C. Mi, “Wireless battery management systems: innovations, challenges, and future perspectives,” *Energies*, vol. 17, no. 13, p. 3277, 2024.

[6] M. H. Alsharif, A. Jahid, A. H. Kelechi, and R. Kannadasan, “Green iot: A review and future research directions,” *Symmetry*, vol. 15, no. 3, p. 757, 2023.

[7] K. Aravind *et al.*, “Optimized fuzzy logic based energy-efficient geographical data routing in internet of things,” *IEEE Access*, vol. 12, pp. 18 913–18 930, 2024.

[8] J. E. Z. Macias and S. Trilles, “Machine learning-based prediction model for battery levels in iot devices using meteorological variables,” *Internet of Things*, vol. 25, p. 101109, 2024.

[9] A. M. Rahmani, A. Haider, K. Moghaddasi, F. S. Gharehchopogh, K. Aurangzeb, Z. Liu, and M. Hosseinzadeh, “Self-learning adaptive power management scheme for energy-efficient iot-mec systems using soft actor-critic algorithm,” *Internet of Things*, vol. 31, p. 101587, 2025.

[10] P. He, Y. Zhou, and X. Qin, “A survey on energy-aware security mechanisms for the internet of things,” *Future Internet*, vol. 16, no. 4, p. 128, 2024.

[11] F. Alzyoud, M. Tarawneh, F. Al-Shrouf, A. Almaghthawi, M. Alrehaili, and H. Kanaker, “Path selection optimization algorithms for mobile agent based on push-all-data strategy.” *International Journal of Interactive Mobile Technologies*, vol. 19, no. 10, 2025.

[12] S. Wang, S. Zhu, C. Chen, F. Shen, W. Zhang, and X. Guan, “Communication and control co-design for heterogeneous industrial iot: A logic-based stochastic switched system approach,” *IEEE Journal on Selected Areas in Communications*, 2025.

[13] Z. Zhou, L. Qian, and H. Xu, “Decentralized multi-agent reinforcement learning for large-scale mobile wireless sensor network control using mean field games,” in *2024 33rd International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2024, pp. 1–6.

[14] A. Kaur, M. Bansal, and D. Kumar, “Mobile agent and aco based data aggregation for wireless sensor network,” *SN Computer Science*, vol. 6, no. 1, p. 77, 2025.

[15] N. Kashyap, S. Upadhyaya, M. Poriye, S. Lalar, and S. Aggarwal, “Assessing the efficacy of a novel adaptive fuzzy c-means (afcm) based clustering algorithm for mobile agent itinerary planning in wireless sensor networks using validity indices,” *Peer-to-Peer Networking and Applications*, vol. 17, no. 4, pp. 2371–2383, 2024.

[16] S. CB *et al.*, “An efficient multi-mobile agent based data aggregation in wireless sensor networks based on hsso route planning.” *Adhoc & Sensor Wireless Networks*, vol. 57, 2023.

[17] S. Karthik, M. Karthick, N. Karthikeyan, and S. Kannan, “A multi-mobile agent and optimal itinerary planning-based data aggregation in wireless sensor networks,” *Computer Communications*, vol. 184, pp. 24–35, 2022.

[18] E. Afonso and M. E. M. Campista, “Opportunistic data gathering in iot networks using discrete optimization,” in *2020 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2020, pp. 1–6.

[19] ——, “Opportunistic data gathering in iot networks using an energy-efficient data aggregation mechanism,” *Annals of Telecommunications*, pp. 1–13, 2024.

[20] T. Alsboui, R. Hill, H. Al-Aqrabi, H. M. A. Farid, M. Riaz, S. Iram, H. M. Shakeel, and M. Hussain, “A dynamic multi-mobile agent itinerary planning approach in wireless sensor networks via intuitionistic fuzzy set,” *Sensors*, vol. 22, no. 20, p. 8037, 2022.

[21] Nidhi and S. Upadhyaya, “Fuzzy c-means clustering of network for multi mobile agent itinerary planning,” in *Smart Trends in Computing and Communications: Proceedings of SmartCom 2022*. Springer, 2022, pp. 589–598.

[22] P. Kaur, S. Batra, and P. S. Rana, “Self configuring mobile agent-based intrusion detection using hybrid optimized with deep lstm,” *Knowledge-Based Systems*, vol. 304, p. 112316, 2024.

[23] M. Bendjima, M. Feham, and M. Lehsaini, “Directional itinerary planning for multiple mobile agents in wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, p. 5584581, 2021.

[24] V. A. Pham and A. Karmouch, “Mobile software agents: An overview,” *IEEE Communications magazine*, vol. 36, no. 7, pp. 26–37, 1998.

[25] N. Christofides, “Worst-case analysis of a new heuristic for the travelling salesman problem,” Carnegie-Mellon University, Pittsburgh, Tech. Rep., Feb. 1976.

[26] Y. Jin, S. Gormus, P. Kulkarni, and M. Sooriyabandara, “Content centric routing in IoT networks and its integration in RPL,” *Computer Communications*, vol. 89, pp. 87–104, 2016.

[27] D. Gavalas, I. E. Venetis, C. Konstantopoulos, and G. Pantziou, “Mobile agent itinerary planning for WSN data fusion: considering multiple sinks and heterogeneous networks,” *International Journal of Communication Systems*, vol. 30, no. 8, p. e3184, 2017.

[28] J. Lu, W. Xiao, E. Song, M. M. Hassan, A. Almogren, and A. Altameem, “iagent: When ai meets mobile agent,” *IEEE Access*, vol. 7, pp. 97 032–97 040, 2019.