



ARPMEC: an adaptive mobile edge computing-based routing protocol for IoT networks

Miguel Landry Foko Sindjoun¹ · Mthulisi Velempini¹ · Vianney Kengne Tchendji^{1,2}

Received: 24 January 2024 / Revised: 9 March 2024 / Accepted: 19 March 2024
© The Author(s) 2024, corrected publication 2024

Abstract

The Internet of Things (IoT) networks comes with many challenges, especially in network architecture designs. IoT is populated by several kinds of devices with different characteristics that are autonomously managed. These devices do not have enough resources and they require to process data in real-time. Hence, there is a need to design suitable architectures for IoT networks that are as efficient as possible. Previously, Cloud Computing (CC) seemed to provide a good solution of processing data from IoT networks. Recently, Mobile Edge Computing (MEC) seems to be offering a better solution than CC by ensuring a better Quality of Services (QoS) provisioning. As a result, many MEC solutions have emerged for QoS improvement in IoT networks. These solutions mainly focus on device resource management without considering data routing from an end-user device to another, especially when the latter are mobile and need to communicate with each other. In this paper, we propose to design an adaptive routing protocol for a MEC-based network to manage efficiently, the end-user devices' energy consumption during data routing. The proposed adaptive Mobile Edge Computing-based protocol consists of two main phases: firstly, we subdivide the network's objects into clusters by exploiting a link quality prediction algorithm. Secondly, we route the data to their destination adaptively by considering the object's movement during the routing process. As presented in the simulation results, our protocol outperforms other existing routing protocols for IoT networks in terms of energy consumption. We then propose the use of our solution for data routing in IoT networks that require huge data processing and forwarding.

Keywords Adaptive routing protocol · Cloud computing · Energy efficiency · Internet of Things · Link quality prediction · Mobile edge computing

1 Introduction

The fifth generation (5G) and Internet of Things (IoT) networks have become part of our everyday life in recent years [1]. They are applied in various domains such as the

intelligent transportation systems (ITS) where vehicles and other devices autonomously communicate to avoid accidents [2, 3]. Smart health is another application where 5G and IoT networks are applied. In Smart health, the idea is to enhance traditional medical systems into a more robust healthcare system able to improve patient follow-up. In hydroponic applications, plants are cultivated without using soil, rather, they are planted in water supplied with nutrient-rich solutions. 5G and IoT networks are used in this case to monitor the oxygen supply and quality of water for optimal growing conditions. The other related applications are aquaponic and robotics. We regard these application domains, as there is a need to develop and deploy solutions that are effective and efficient given the nature of services. Among the solutions that are proposed to improve the use of 5G and IoT networks, architecture designs and data routing protocols are some of the

✉ Miguel Landry Foko Sindjoun
miguel.fokosindjoun@ul.ac.za; miguelfoko@gmail.com

Mthulisi Velempini
mthulisi.velempini@ul.ac.za

Vianney Kengne Tchendji
vianneykengne@gmail.com

¹ Department of Computer Science, University of Limpopo, Mankweng, South Africa

² Department of Mathematics and Computer Science, University of Dschang, Dschang, Cameroon

commonly used techniques. When it is known that the routing problem in these networks is an NP-complete problem [4, 5].

The 5G and IoT networks are part of the mobile ad hoc networks (MANET) family. The networks that do not require pre-existing infrastructure for deployment and to function [6], MANET are able to organise themselves without external interventions [7] and can be made of many heterogeneous devices such as the 5G and IoT networks devices. Moreover, they generate a large amount of traffic and data so that the need for specific infrastructure for data processing and communication becomes a requirement [8–10]. To solve the data processing issue, cloud computing (CC) was initially proposed [1, 11]. Unfortunately, due to the large amount of traffic generated by the network, CC became a challenge for the network, namely due to the network latency that increases when the number of nodes increases. These challenges have resulted in the emergence of Mobile Edge Computing (MEC). It can be observed that, MEC brings computing and storing resources from cloud data centres to the edge closer to the end-user devices with the aim of reducing data traffic within the network and to maximise the quality of service (QoS) [11–13]. In summary, many MEC-based solutions have been proposed to manage and improve 5G and IoT networks QoS [11, 12, 14, 15]. To the best of our knowledge, most of them focus on physical architecture which compounds the effects of the problem. By doing so, they overlook the part of the network consisting of data routing between network end devices, mainly the end-to-end device communication. In this paper, we propose to solve the data routing problem between end-to-end devices in a MEC-based network. Given the characteristics of IoT networks such as dynamic topology, frequent changes in link quality and device mobility, it is important to develop routing protocols that adapt to prevailing situations [16, 17]. To be precise, the solution we propose is an adaptive routing protocol for a MEC-based network. Our main objective is to improve the energy efficiency of end-user devices in routing data in the MEC architecture. Our adaptive routing protocol (ARP-MEC) makes use of link quality prediction as proposed by authors in [6] to ensure the efficient use of available links and better QoS provisioning. The main techniques we use to achieve efficient routing are network clustering and link quality prediction. It should be noted that, the solution that we propose consists of a scheme that has two steps: we firstly use the link quality prediction to subdivide the network into clusters by ensuring that end devices use the best existing link (to limit the energy consumption for data transmission). Secondly, based on the previously obtained clusters, we ensure the data routing by considering the end device nodes' mobility to guarantee the adeptness of our scheme. The proposed ARPMEC protocol is one of the

protocols which address data routing in MEC ecosystems. Furthermore, it is an ideal candidate for applications such as autonomous vehicle networks, smart health and smart city applications (just to name a few) where end-user devices are mobile.

The rest of the paper is organised as follows: In Sect. 2, we present the state of the art concerning our study. Our proposed solution for data routing in a MEC-based network is described in Sect. 3. The simulation results are presented in Sect. 4. Finally, we conclude the paper and present some open issues in Sect. 5.

2 State of the art

Many works have been proposed for 5G and IoT network deployment and functioning in literature. Some of them present solutions that make use of mobile edge computing for a better quality of services, while others do not. In this section, we present a review of related work in the field of 5G, IoT and beyond networks for QoS improvement. We start by the related works on MEC-based 5G and IoT network architectures. Then, we continue with a review of routing protocols for IoT networks, and finally, we present a brief status of link quality prediction for the improvement of QoS in IoT networks.

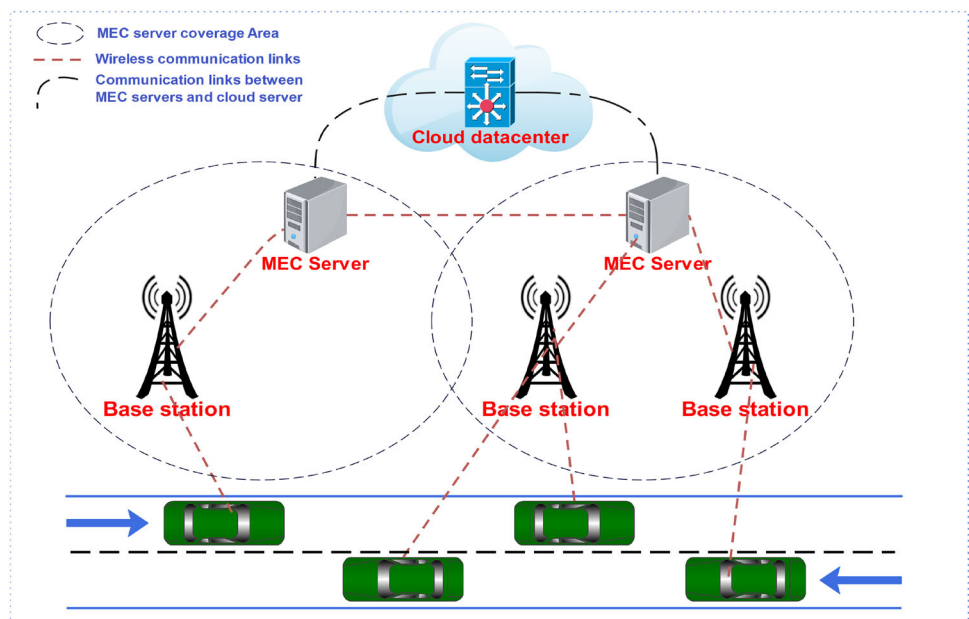
There are a number of authors that focused on developing new architectures for mobile edge computing in present networks. Indeed, after illustrating the integration of MEC into 5G networks, Filali et al. [12] proposed an architectural framework for a MEC environment based on Network Function Virtualization (NFV) [18] technology. Checkired et al. [9] proposed a MEC architectural framework based on Software Defined Networking (SDN) [19], NFV and Network Slicing (NS) [20] for autonomous vehicular networks (AVNET). Their solution sought to improve the transmission efficiency and to satisfy the low latency constraint. Yala et al. [14] proposed a placement scheme applicable to MEC architecture in a NFV environment with the objective to minimize the access latency and to maximize the service availability. Cziva et al. [15] investigated the allocation of Virtual Network Functions (VNF) running on multiple hosts for the minimization of latency between end user equipment (UE) and their respective VNF. Then, they proposed a placement model for VNF in a dynamic way according to the latency fluctuation, user mobility and demand. For the computing, storage and resource management in an AVNET, Peng et al. [21] proposed a MEC approach that supports the increase in data traffic and computing/storage tasks while reducing the response delay for cooperative driving between autonomous vehicles (AVs). Thereafter, by studying the bandwidth, computing and storage within an

MEC-based network, Peng et al. [22] made use of reinforcement learning to improve the vehicular communication QoS. More recently, Foko et al. [11] proposed an MEC architecture based on SDN, NFV, Service Function Chaining (SFC) [23] and NS technologies that sought to improve the QoS in an AVNET. They presented a mathematical model to evaluate resource consumption that helps to prove the effectiveness of QoS improvement in the AVNET. An example of MEC architecture applied to AVNET as presented by Foko et al. [11] is presented in Fig. 1. Khanh et al. [24] proposed a cloud-fog-edge integrated computing framework for Internet of Health Things (IoHT) which shows a queuing model for Edge Computing nodes. Computing nodes such as Fog Computing and Cloud computing, however, end-device nodes were not considered. The previous works focus on the server side when it comes to improving the network QoS and network service. They do not focus on how end-user devices communicate which means that the way these devices manage the routing of data remains a challenge. As a result of the previous observation, we continue this section by presenting state-of-the-art routing protocols for IoT networks. The objective of the study is to see how to incorporate routing into a MEC-based network for QoS improvement.

The Internet of Things (IoT) network is a mobile ad hoc network (MANET) that consists of hundred to thousand of connected devices called objects [1]. Ad hoc networks do not need preexisting infrastructure to work properly. Objects that populate IoT networks collect, process and share data from their surrounding environment for decision purposes. The way the data are collected and shared needs to be clearly defined in order to optimize the functionalities

of the network. Routing protocols are important at this level to define how collected and processed data can be transmitted within the network in a more efficient manner. Following are some routing protocols designed for IoT networks. Bomgni et al. [7] proposed a tree-based permutation routing protocol for wireless sensor networks (WSN) that make use of clustering techniques to facilitate data routing in a multi-hop communication network. Myoupo et al. [25] proposed a fault-tolerant and energy-efficient routing protocol for efficient data routing to a base station. The solution is designed for any type of WSN, unlike most of the previously existing solutions that were only designed to run on dense networks. The two previously mentioned solutions were developed for a network in which all the devices have the same characteristics (in terms of storage, computing, etc.) while in real-life IoT networks, devices in the network can be different. The routing protocol for low power and lossy networks (RPL) [26] is one of the first protocols that was developed for IoT networks and considering the diversity of network devices. Unfortunately, RPL is destined for a network where objects are static, then, suffer from many issues when nodes become mobile [27]. Lakhlef et al. [28] proposed a routing protocol for one-hop IoT networks by using a clustering approach. Unfortunately, in [28] objects remain awake during the routing, which does not help to solve the energy efficiency problem of IoT networks. The energy efficiency problem is solved by Bomgni et al. [1] where authors proposed a routing protocol for single-hop IoT networks, that is more energy efficient than the work presented in [28]. The solution proposed by Bomgni et al. makes use of the wake and sleep technique to guarantee the efficient use of

Fig. 1 An AVNET based MEC architecture



objects' energy. The aforementioned existing solutions do not consider object mobility and link quality prediction for better QoS. This differs from the work proposed by Orchu et al. [16] which is an adaptive routing protocol for ad-hoc networks that considers both the link quality estimation and the nodes' mobility to ensure an efficient data routing inside the deployed network. On the other hand, Ramkumar et al. [17] proposed a multi-adaptive routing protocol for IoT-based ad-hoc networks that aims to use the most efficient path for data communications, by ensuring the connectivity of the network and its operations before sending the data items. Unfortunately, the previous solution does not use a link quality prediction for better use of existing routes. The proposed scheme in [17] is not designed for MEC environments in contrast to the routing protocol proposed by Khanh et al. [5]. The authors enhance the performance and reduce the energy consumption of end-user devices. Nevertheless, they considered the use of a mobile agent to achieve the objectives of their routing protocol. However, if the mobile agent is compromised, the routing protocol is affected. Data routing in IoT networks also depends on the quality of links that are used for data transmission. We, therefore, consider these state-of-the-art works in this section and present some related work on link quality prediction in IoT networks.

A number of schemes designed for link quality estimation (LQE) in IoT networks state that the use of poor link quality leads to bad quality of service and bad quality of experience provided by the network [6, 29, 30]. It is therefore recommended to make use of LQE in order to ensure that objects use a better link for communication. In the literature, two categories of LQE are proposed. The first is based on a single metric: they can either be based on hardware metrics such as the received signal strength indicator (RSSI), link quality indicator (LQI) and signal-to-noise ratio [31–34], or on the software metrics such as the packet delivery ratio (PDR), expected transmission count (ETX) and expected transmission time (ETT) [35–37]. The second category of LQE is based on several metrics: in this case, many metrics are combined to obtain a more accurate link prediction. The combined metrics can be hardware-based, software-based or both. Examples of such link quality predictors are four-bit, F-LQE, etc. [29, 30, 38–42]. Authors in [29] use a hardware-based metric (RSSI) with a software-based metric (PDR) to perform link quality estimation in an IoT network using four machine learning algorithms that are Logistic Regression (LR), Linear Sup-

port Vector Machine (LSVM), Vector Machine (VM) and Random Forest (RF). The results in [30] show that the RF machine learning algorithm is the best in the context of IoT networks applied to the PDR metric. Thereafter, the combination of machine learning algorithms for link quality prediction in [6] with the RSSI and PDR metrics in the same framework to obtain a more accurate link quality estimation for IoT networks was proposed. In this paper, we use the combination of machine learning prediction algorithms to limit the energy consumption of end-user devices during the data routing. Table 1 summarizes the solutions we presented in this section focusing on the protocol type, the use of MEC infrastructure and the use of link quality estimation. The results presented in the table show that none of these solutions use MEC and LQE. They are also not adaptive in data routing in IoT networks.

In regards to the related work, our paper proposes an energy-efficient routing protocol for IoT networks, based on an MEC infrastructure, that makes use of link quality estimation and the clustering technique to improve the efficient use of energy by end-user devices. It also considers the movement of devices within the network. The proposed protocol (ARPMEC) is adaptive and it ensures that even in case of mobility, a device remains connected to the network and can route its data items.

3 Our proposed adaptive routing protocol for MEC-based 5G and IoT networks

In IoT networks, adaptive routing is applicable when end-user equipment (objects) move regularly, thereby changing the network topology. In such a case, objects must adapt to the new topology to route their data. In this section, we design an adaptive routing for IoT networks that uses the mobile edge computing infrastructure. Our methodology is described as follows: by predicting a link quality, we first subdivide the objects into groups called clusters each with a cluster head (CH). The main role of CH is to aggregate the collected data to prevent objects from transmitting the same data in the network. Secondly, we organise the routing of data to their final destination adaptively by considering the movement of objects within the network during the routing process. The Pseudo code of our routing protocol is given in Algorithm 1. The proposed protocol makes use of some annotations described in Table 2.

Algorithm 1 Adaptive routing protocol for MEC-based networks

```

1 Begin
2   Phase 1 : Clustering of objects using link quality prediction (Algorithm 2);
3   Phase 2 : Adaptive Data routing between network's objects (Algorithm 3);

```

Table 1 Existing routing protocols for IoT networks

Protocol	Type	MEC used	LQE used
Winter et al. [26]	Routing protocol for WSN	No	No
Bomgni et al. [7]	Permutation routing protocol for WSN	No	No
Myoupo et al. [25]	Routing protocol for WSN	No	No
Lakhlef et al. [28]	Permutation routing protocol for IoT	No	No
Ramkumar et al. [17]	Adaptive routing for IoT	No	No
Bomgni et al. [1]	Permutation routing protocol for IoT	No	No
Orchu et al. [16]	Adaptive routing for IoT	No	Yes
Khanh et al. [5]	Routing protocol for IoT	Yes	No

Table 2 Some notations

Symbol	Definition
ID	Object identity
C	Number of available channels
N	Number of objects in the network
GPS	Global Positioning System
$a(v)$	Data that belongs to the object with $ID = v$
K	The number of MEC servers
R	Minimum number of messages required to perform an accurate link quality prediction
Δ	The number of items to be routed during the protocol
CM	Cluster member
CH	Cluster HEAd
GW	Gateway
K	The number of clusters generated after the clustering algorithm 2
T	The number of rounds for data routing
ζ	The energy threshold required by a CH to initiate a round with the same role
HUB_{max}	The maximum number of objects in a single cluster after the clustering
$energy(i)$	The residual energy of the object with $ID = i$
ϑ_i	The total number of items that will be shared by the object with $ID = i$ during the current round
λ	The number of objects in the cluster with $CH i$
$\theta = \sum_{b=1}^{\lambda} \vartheta_b$	The total number of items that will be shared in the cluster during the current round
$j, 1 \leq j \leq \lambda$	The rank of object i in the current cluster
\varkappa	The number of items that belong to the current cluster which are possessed by the GW

Algorithm 2 Clustering of objects using link quality prediction

```

Input: Network with  $K$  clusters,  $\zeta$ ,  $HUB_{max}$ .
/* This algorithm is executed by the object with  $ID = i$  */
1 Begin
2   For  $t = 1; t \leq T; t++$  Do
3     If  $i$  is a  $CH$  Then
4       If  $energy(i) > \zeta$  Then
5         Listen to the network during  $HUB_{max}$  slots to receive the number of
          items each object intends to share during the round;
          /* Including the  $GW$  */
6         Broadcast the list of objects with their corresponding  $\vartheta$ ;
7         Listen during  $\theta + \varkappa$  slots;
8         If Some items that arrived do not belong to the current cluster Then
9           Send these items to the  $GW$ ;
10        Else
11          Send a message to all the  $CM$  to abdicate as  $CH$ ;
12      Else
13        Listen to the communications in the group for 1 slot;
14        If The abdicate message received from the current  $CH$  Then
15          Listen to the clusters communications until the  $(j - 1)^{th}$  slot;
16          Broadcast message  $(i, energy(i))$  in the cluster at the  $j^{th}$  slot;
17          Listen to the clusters communications until the  $\lambda^{th}$  slot;
18          If  $energy(i) > \zeta$  and  $energy(i)$  is the higher energy value received
            from the other  $CM$  Then
19            If  $i$  is the higher  $ID$  compared to the  $IDs$  of the cluster with
              respect of the conditions in Line 18 Then
20              Becomes  $CH$  for the cluster and sends notification to the  $GW$ ;
21            Else
22              Remains  $CM$  and joins the one with the higher  $ID$  compared to
                the objects according to the conditions in line 18;
23        Broadcast  $\vartheta_i$ ;
24        Listen to the cluster communications during the  $\theta - \vartheta_i$  slots that  $i$  does not
          broadcast;

```

However, before we go deeply in the Algorithm 1 description, we start by presenting the need of an adaptive routing in the context of IoT networks. Indeed, communications may be intensive due to the higher number of objects that compound the network and the lower number of available communication channels. Furthermore, the nature of IoT objects make them dynamic in the sensing area in real world applications. Thus, during their movement, it may happen that objects lose the connection from the channel from which they were initially connected. Adaptive routing techniques can help to overcome these issues [43]. The idea of the adaptive routing we are proposing is then to make sure that even in the situation of object movement, the latter is still able to route/receive data from/to the other networks' objects by using the most accurate available channel as possible. The proposed scheme should be deployed both at the end device and at the server.

3.1 Clustering algorithm

The clustering algorithm we present uses the link quality prediction to build a more reliable structure. Also, it is executed at the beginning of each round of the protocol to make sure about the adaptability for data routing. To start, we define the following assumptions:

- The network is Time Slotted Channel Hopping (TSCH) [44] with C available channels;
- There are N objects in the network and each object is identified by an ID ($1 \leq ID \leq N$) which is an integer;
- Before deployment, each object knows the ID of the other objects;
- Each network object is equipped with a global positioning system (GPS);
- An item is the couple $(a(v), v)$ where $a(v)$ is the data that belongs to the object with $ID = v$;

- Each item is delivered in a single slot for one-hop communications (that is, each broadcast is done in one slot);
- Objects' energy capacity is high when they are deployed;
- The number of MEC servers (MS) K is such that the whole network is covered by these MS;
- We assume that R is the integer number representing the minimum of messages that are required to perform an accurate link quality prediction;
- The total number of items Δ that are supposed to be routed during the protocol is far greater than that of objects N that are presented in the network and denoted by $N \ll \Delta$;
- $C \ll N \ll \Delta$.

This clustering scheme aims to create a similar structure as the one presented in Fig. 2. In this figure, the Gateways (GWs) are MEC servers for the deployed MEC infrastructure.

To ensure that objects use the best link for their communications, we make use of link quality prediction. The clustering procedure is described as follows (the pseudo-code is presented in Algorithm 2):

1. After deployment, each object broadcasts R *HELLO* messages in each of the C available channels (Line 4), where R is a number that will allow the network's objects to have enough data to perform an accurate link quality prediction. The broadcast is done starting from the object with the smallest ID to the object with the largest ID . The order in which messages are sent in this step is known by all the objects. Since each object performs a broadcast on each channel, an object with $ID = I$ knows that it starts its R th broadcast at time slot $(I - 1) \times R \times C + (R - 1) \times C + 1$ slots (Object with ID I starts its R th broadcast when the $I - 1$ objects preceding it have made their R broadcasts on all the channels. In addition, I must have already made its $R - 1$ first broadcast in all the C channels).
2. Upon receipt of a *HELLO* message, each object records the signal strength (RSSI) with which it received the item (Line 8), then keeps the number nb of received items from each of its potential neighbours (Line 9). nb will allow the receiver object to deduce the average packet delivery ratio (PDR) from the link that exists

between itself and the items' sender. This PDR is computed on each of the channels. Indeed, knowing that each object is supposed to send R *HELLO* messages on each of the channels, $nb \leq R$, and therefore the average PDR of the associated channel will be $\frac{nb}{R}$ for a given link (Line 10).

3. After $R \times N \times C$ slots, all the objects know their potential neighbours, as well as the average values of PDR and RSSI associated with each of their potential neighbours. Let us call *score*, the pair (PDR, RSSI) that describes the link between two potential neighbours. By using the prediction method presented by authors in [6], object I makes a prediction on all the existing links between itself and the potentially connected objects using the corresponding *score*, then, decides to join the one with the best prediction efficiency (Lines 11 to 12). This decision is marked by the broadcasting of a message *JOIN*(M) in the network (Line 13) (where M is the ID of the object with the best link quality prediction). In this step, the diffusion is once more time done from the object with the smallest ID to the object with the largest ID . N slots are then required for this broadcast to be accomplished by all the objects, while each object remains awake during all the N slots.
4. Once all messages from the previous step have been received by all the network's objects, if I has received several *JOIN*(I) messages, then, it becomes *CH* for the transmitting object (19). On the other hand, if I did not receive any *JOIN*(I), it becomes a cluster member (*CM*) of the cluster for which it obtained the best link quality prediction (Line 17). The *CH* obtained here broadcast the list of their *CM* to the MEC servers they are connected with (Line 20). At receiving the messages from *CH* in this phase, all the *MS* forward the received messages to the only cloud server in the network. That cloud server is responsible for cluster cleaning, by making sure that each object belongs to a single cluster at any given moment. After clusters are cleaned, the cloud server sends the cleaned clusters to *MS* which broadcasts the said list in the network to inform all the objects about the clusters they belong to. The list also contains the geographical coordinates of the clusters (in terms of geographical position). From this point, each object uses its *CH* when it needs to communicate with an object that does not belong to its cluster or if it wants to communicate directly with the object if they are in the same cluster. The completion of this phase takes $2 \times K$ slots. K slots for *CH* to send their *CM* to the *MS*, and K slots for *MS* to send the definitive *CM* to the whole network.

At the end of this clustering, a network with a physical structure similar to that of Fig. 2 is generated. It

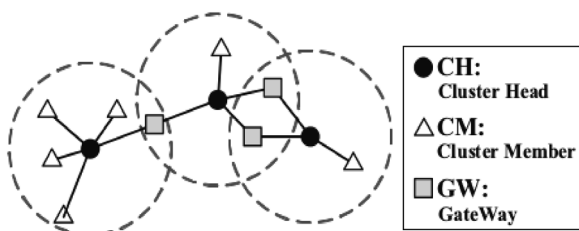


Fig. 2 Cluster topology in one-hop communication [45]

corresponds to the network with all the clusters already formed. Each cluster has a *CH* and each *CH* is connected to one or more MEC servers that will be used for multi-hop communications thereafter.

Lemma 1 *The time complexity required by our clustering algorithm is in $O(N)$.*

Proof The sending of *HELLO* messages takes $R \times C \times N$ slots. The N objects of the network take N slots to send the *JOIN(M)* messages to their potential *CH*. All the *CH* take K slots to send their *CM* to the MS for cleaning, and finally, MS take K slots to send back the cleaned *CM* networks to *CH*. Then by summing all the required slots, we obtain the result presented in Eq. 1, which corresponds to the complexity in $O(N)$ given that R , C and K are negligible compared to the number of objects within the network. \square

$$\begin{aligned} \text{ClusteringTime} &= R \times N \times C + N + 2 \times K \\ &= N(R \times C + 1) + 2 \times K \end{aligned} \quad (1)$$

3.2 Data routing protocol

The data routing protocol is executed after the clustering protocol. It takes T rounds to execute. Each round corresponds to a cycle within which each object routes items which it has to transmit to the destination node before receiving new items. An item is the couple $(a(v), v)$ where $a(v)$ is the data that belongs to the object with $ID = v$. As an assumption for the data routing process, it is considered that during a round $t \in T$, an object cannot move outside its cluster, but at the end of each round, it can decide to move or not (our proposed routing protocol then adapts itself according to object movements at the beginning of each round). The pseudo-code of the protocol is described in Algorithm 3. That protocol is executed by an object with $ID = i$, $1 \leq i \leq N$. If i is a *CH* for its cluster, then it executes Lines 3 to Line 11, otherwise, it executes Line 12 to Line 24.

Algorithm 3 Adaptive data routing between network's objects

Input: Network with N objects not partitioned.
Output : Network with K clusters.

```

1 Begin
2   For  $object = 1; object \leq N; object++$  Do
3     /* While an object is broadcasting, all others are listening */
4     If  $object == i$  Then
5       /* Where  $i$  is the ID of the object executing the algorithm */
6       Broadcast the  $R$  HELLO messages on each of the  $C$  channels;
7     Else
8       /* Listen to the broadcasts from another object */
9       Listen to the network for  $R * C$  slots;
10      For Each channel  $C$  Do
11        Record the RSSI of each HELLO message that arrives from this object ;
12        Count the number  $nb$  of HELLO received from this object ;
13        Compute the PDR for this object:  $PDR = \frac{nb}{R}$ ;
14
15  For Each of the neighbors whose HELLO messages were received Do
16    Make a link prediction using the learner presented in [6];
17  Broadcast the message JOIN(M);
18  /*  $M$  designates the object with which the prediction was the best */
19  Listen to the network for  $N - 1$  slots;
20  Begin
21    If No message arrives Then
22      Become CM for  $M$ ;
23    ElseIf One or more messages received Then
24      Become CH for all objects from which JOIN messages were received ;
25      Send the CM list to the connected MEC servers;
26  Receive the cleaned CM list from the connected MEC server;

```

When i is a CH , it first checks if its residual energy can allow it to complete the whole round which starts from (Line 4). If it is not possible, it sends a message to the group, to give up its role of CH (Line 11). In the other case, if it has enough energy to execute the round, it starts by listening to the network for HUB_{max} slots (Line 5) to receive the items from the GW/MEC Server (MS), and other objects of the cluster have to share within the group during the round. After that, it shares the list of objects and the corresponding items within the cluster (Line 6). The idea is to help each object of the cluster to know when it should start its data routing during the round. The broadcast of messages is preceded by the listening of the cluster's communications during the communication of the cluster's objects and during the duration required by the GW to share the items that belong to the current cluster (Line 7). At the end of the cluster's listening phase, if there are items that arrived at the CH and do not belong to the current cluster, the items are sent to the GW (Line 9).

When i is not CH , it starts by listening the cluster's communications for one slot (Line 13). If it receives the abdicated message from the CH during that slot, it starts the new CH election process. The process includes the need to listen during $j - 1$ slots (Line 15, where j is the rank of object i in the current cluster), the broadcast of the message (i , $energy(i)$) within the cluster at the j th slot (Line 16), and the listening of the cluster's communications until the λ th slot (Line 17, where λ is the number of objects in the current cluster). After the listening phase, if the residual energy of i is the higher energy value compared to those it has received and that enables it to be CH (Line 18), then, if i is the higher ID value compared to objects that have the higher energy values, i becomes CH for the current cluster and sends the notification to the GW (During this phase, all the CM of the cluster know it is the new CH since they have received all the messages that help i to decide to become CH). However, if i is not the higher value of CM that respects the energy condition to become CH , then i remains CM and considers the one with the higher ID as CH (Line 22). Finally, after the new CH election, i broadcasts ϑ_i messages in the network (Line 23) then, listens to the cluster communications during the $\theta - \vartheta_i$ slots that it does not broadcast, where θ is the total number of items that will be shared in the cluster during the current round while ϑ_i is the total number of items that i will broadcast during a given round. Lastly, concerning the adaptability of the routing algorithm, it is important to note that each object which moves out of a cluster to another in a new round, knows the new cluster it belongs to, since that the list of clusters is shared with all the objects at the end of the clustering process by the MEC server as presented in Line 21 of Algorithm 2.

Lemma 2 *The time complexity required for our adaptive data routing protocol in IoT by network's object to complete is in $O(T\Delta)$, where T represents the number of rounds of the data routing protocol and Δ is the number of items to be routed.*

Proof Let

$$A = \theta + \kappa + HUB_{max} + 1 \quad (2)$$

$$B = \theta - \vartheta_i + \lambda + (1 \vee 0) \quad (3)$$

$$\begin{aligned} RTime &= \text{Max}\left(T \cdot \left(\begin{cases} A \vee 1 & \text{if } i \text{ is } CH \\ 1 + (B \vee 0) & \text{if } i \text{ is not } CH \end{cases}\right)\right) \\ &= \text{Max}\left(\begin{cases} (T.A) \vee (T.1) & \text{if } i \text{ is } CH \\ T.1 + (T.B) \vee T.(\theta - \vartheta_i) & \text{if } i \text{ is not } CH \end{cases}\right) \\ &= \text{Max}\left(\begin{cases} (T.A) \vee T & \text{if } i \text{ is } CH \\ T + (T.B) \vee T.(\theta) & \text{if } i \text{ is not } CH \end{cases}\right) \end{aligned} \quad (4)$$

where \vee represents the logical **or** operation, A and B respectively the values computed in Eqs. 2 and 3.

In the worse case, $\theta = \vartheta_i = \kappa$, and $\lambda = HUB_{max}$. Meaning that Eqs. 2 and 3 become Eqs. 5 and 6 respectively.

$$\begin{aligned} A &= \theta + \kappa + HUB_{max} + 1 \\ &= \theta + \theta + HUB_{max} + 1 \end{aligned} \quad (5)$$

$$\begin{aligned} B &= \theta - \vartheta_i + \lambda + (1 \vee 0) \\ &= \theta - \theta + HUB_{max} + (1 \vee 0) \\ &= HUB_{max} + (1 \vee 0) \end{aligned} \quad (6)$$

When we substitute Eqs. 5 and 6 into Eq. 4, we obtain the Eq. 7

$$\begin{aligned} RTime &= \text{Max}\left(\begin{cases} (T.(2.\theta + HUB_{max} + 1)) \vee T & \text{if } i \text{ is } CH \\ T + (T.(HUB_{max} + (1 \vee 0))) \vee T.(\theta) & \text{if } i \text{ is not } CH \end{cases}\right) \\ &= \text{Max}\left(\begin{cases} (2.T.\theta + T.HUB_{max} + T) \vee T & \text{if } i \text{ is } CH \\ T + T.HUB_{max} + T.(1 \vee 0) + T.(\theta) & \text{if } i \text{ is not } CH \end{cases}\right) \\ &= \text{Max}\left(\begin{cases} (2.T.\theta + T.HUB_{max} + T) \vee T & \text{if } i \text{ is } CH \\ T + T.HUB_{max} + T.(1 \vee 0) & \text{if } i \text{ is not } CH \end{cases}\right) \\ &= \text{Max}[(2.T.\theta + T.HUB_{max} + T) \vee T, \\ &\quad (T + T.HUB_{max} + T.(1 \vee 0))] \\ &= \text{Max}[T.((2\theta + HUB_{max} + 1) \vee 1), \\ &\quad T.(HUB_{max} + 1 + (1 \vee 0))] \\ &= T.\text{Max}[2\theta + HUB_{max}, HUB_{max} + 1 + 1] \\ &= T.\text{Max}[2\theta + HUB_{max}, HUB_{max} + 2] \\ &= T.(2\theta + HUB_{max}) \\ &= 2\theta T + HUB_{max}T \end{aligned} \quad (7)$$

Given that $N \ll \Delta$, if we consider the worse case where all the items of the network are to be routed in the same cluster, then, we can deduce that $HUB_{max} \ll \theta$ because $HUB_{max} \leq N$, thus, we conclude that the complexity of the routing protocol presented in Algorithm 3 is in $O(T\Delta)$. \square

4 Simulation results and interpretation

To evaluate the effectiveness of our proposed scheme, we performed some simulations. The results of these simulations are presented in this section. These simulations were performed on a Core i5 1.6 GHz * 8, 8 Go of RAM laptop running on Ubuntu 22.04.2 LTS Operating System. Matlab was used as a simulation environment. As stated by authors in [46], some of the metrics that can be used for evaluating routing protocols in wireless networks are network delay, throughput, success rate, latency, energy consumption and network lifetime. Hence, the main objective for our simulations was to evaluate the energy consumption of nodes involved in the data routing. For this purpose, we used the energy model presented in Eq. 8 where:

$$E = Q * n(e_t + e_{amp} * d^2) + e_r * n \quad (8)$$

- E is the total energy consumed by an object;
- e_t , e_{amp} and e_r are the transmission, the amplification and the reception energy respectively. It is important to note that a node spends more energy when it transmits items compared to the one required for reception and amplification.
- d is the average Euclidean distance between two objects;
- Q is the energy parameter and n is the number of items that should be routed.

In Table 3, we present the values that have been used for parameters during the simulations. Among these

parameters, Δ is the number of items to be routed within the network. We set it to 10,000 because, in an IoT network, the number of items to be routed can be enormous depending on the application scenario. The number of channels C is another important parameter that we described in Table 3. It is the number of available channels that can be used by objects to route items. This parameter is essential because each item to be routed should follow a given path that is defined in a channel. Generally, the number of channels is far less than the number of objects in the network, which is also far less than the number of items to be routed [7] in a MANET. In our simulations, C varies from 1 to 16 which is the number of channels used in [6] for LQE. The number N of nodes in the network can increase from 1 to 500 in the considered perimeter, while the Euclidean distance (d) between two nodes should be less than 1 km before these nodes can be able to communicate directly. Finally, the number of messages required for an accurate LQE (R) varies from 25 to 100 because we want to reach the most realistic conditions possible.

4.1 Clustering algorithm analysis

We evaluated the clustering algorithm we proposed (Algorithm 2). The objective was to evaluate its energy consumption and compare it to existing clustering algorithms in the same category. The comparison was made with the Instantaneous Clustering Protocol (ICP) [47] and the Instantaneous and Secure Clustering Protocol (ISCP) [45].

In Fig. 3 we ran the clustering protocols with 500 nodes within a network with a single communication channel. In the simulation, we vary the number R of *HELLO* messages required by nodes to perform the link quality estimation (LQE) from 25 to 100. In Fig. 3a with $R = 25$, Fig. 3b with $R = 50$, Fig. 3c with $R = 75$ and Fig. 3d with $R = 100$ our clustering algorithm (in red colour) consumes more energy than ICP (in blue colour) but less than ISCP (in black

Table 3 Experimental parameters

Parameter	Value
Energy mitigation (n)	$2 \leq n \leq 4$
Number of items (Δ)	$1 \leq \Delta \leq 10,000$
Number of channels (C)	1, 4, 8, 16
Number of nodes (N)	$1 \leq N \leq 500$
Number of rounds (T)	$1 \leq T \leq 200$
Number of messages required for an accurate LQE (R)	25, 50, 75, 100
Euclidean distance (d)	1 km
Transmission energy (e_t)	0.03J
Reception energy (e_r)	0.02J
Amplification energy (e_{amp})	0.01J

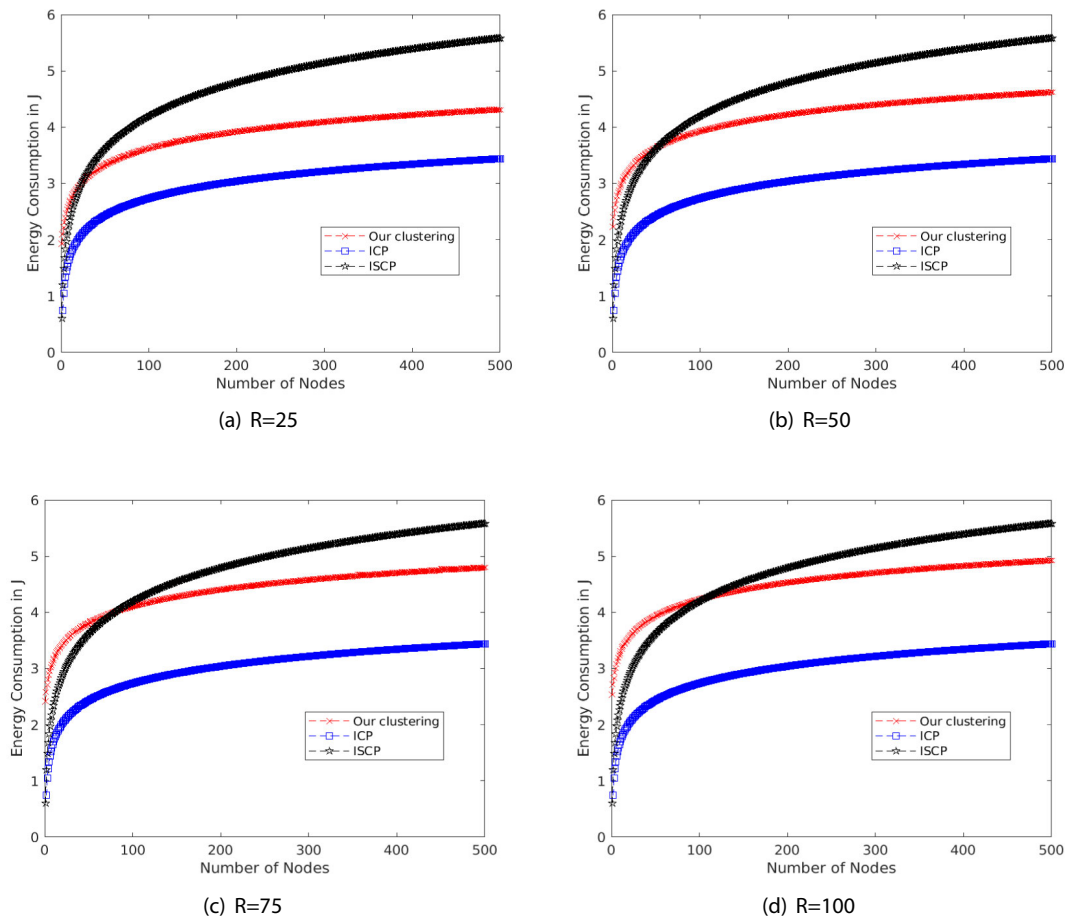


Fig. 3 Clustering algorithm in a single channel network ($C=1$) (Color figure online)

colour). Our observation is that the ICP and ISCP energy consumption does not vary as the value of R changes (which is expected since there is no LQE in these two algorithms). Secondly, our clustering algorithm consumes more energy when R goes up. Meaning that the more the *HELLO* messages will be sent, the more energy is consumed by our clustering algorithm. Finally, in the single channel configuration presented in Fig. 3a–d, we observed that ISCP consumes most energy in comparison to our clustering algorithm.

To fulfil the requirements of authors in [6] regarding the number of *HELLO* messages that will be sent to achieve an accurate LQE, we fixed R to 100, then we varied the number of channels from 1 to 16 (16 is the number of channels that was also considered by authors in [6]). The obtained results are presented in Fig. 4. These results show that the energy consumption of ICP (in blue color) and ISCP (in black color) does not change. However, our clustering algorithm (in red color) consumes more energy than ISCP when the number of channels increases (from Fig. 4a with $C = 1$ to Fig. 4d with $C = 16$). Particularly, when the number of channels becomes greater or equal to 8

(Fig. 4c, d), both ICP and ISCP are more energy efficient than our scheme. This observation is clearly understandable because neither ICP nor ISCP depends on the *HELLO* messages that are used by our proposed clustering scheme for LQE. In the rest of our simulations, we consider the number of channels to be equal to 16 and the number of *HELLO* messages to be equal to 100, meaning that the generated energy consumption in the rest of the simulations corresponds to the one shown in Fig. 4d. The justification of this choice is based on the configuration used for link quality estimation in [6] where R was set to 100 and C to 16.

4.2 The analysis of the routing algorithm

In Sect. 4.1, we observed that the clustering algorithm we employed in our solution is less energy efficient than ICP [47] and ISCP [45]. However, considering the same clustering algorithms and using our proposed routing algorithm (ARPMEC), we obtained the results presented in Fig. 5. In the figure, we ran our routing protocol and considered the ICP protocol (in blue colour), the ISCP protocol (in black

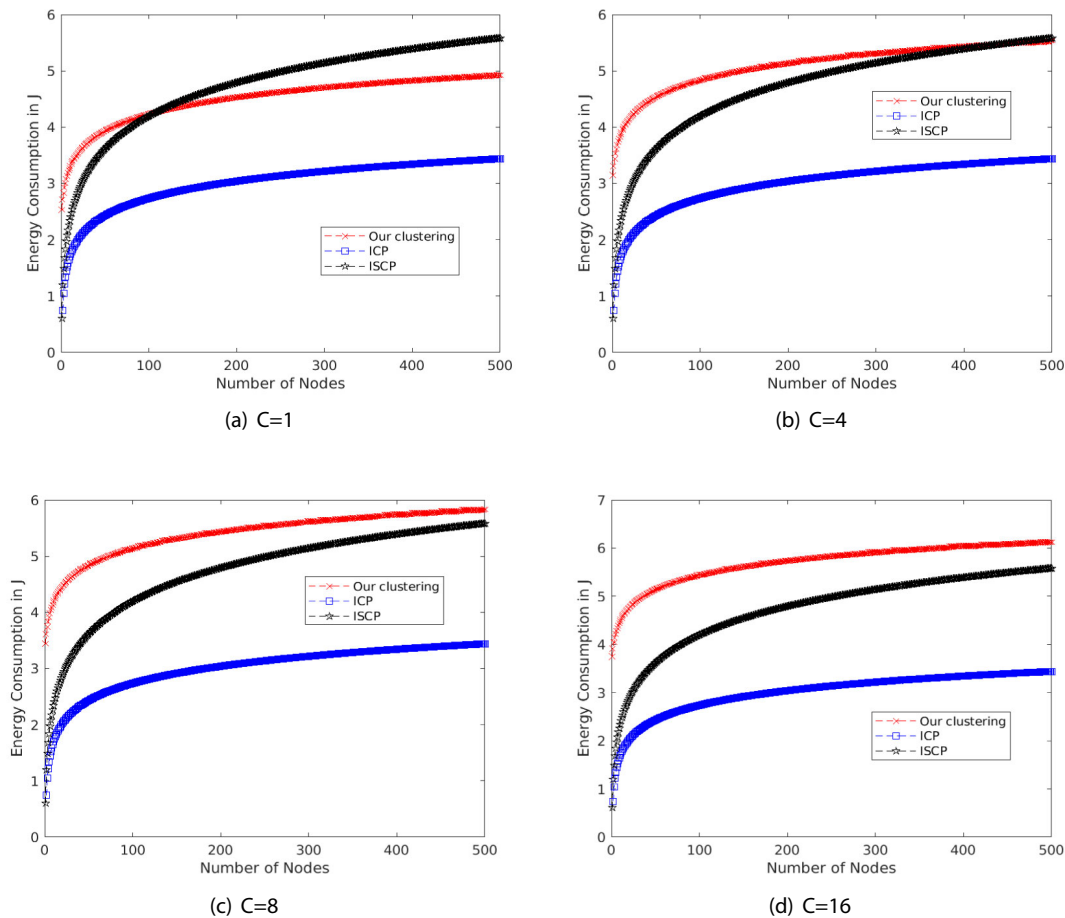


Fig. 4 Clustering algorithm in a network with 100 Hello messages ($R = 100$) for LQE (Color figure online)

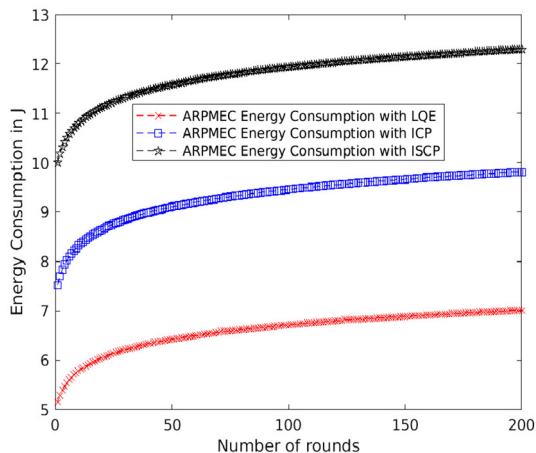


Fig. 5 Energy consumption of the routing algorithm with clustering (Color figure online)

colour) and the clustering algorithm that we proposed in Algorithm 2 (in red colour) as the clustering algorithm before the execution of our adaptive data routing algorithm proposed in Algorithm 3. The results confirm that

ARPMEC consumes less energy when using our clustering algorithm compared to ISC or ISCP when used as clustering algorithms. This is understandable since our clustering process with the desired link quality prediction, objects use available better links for data transmission, while it is not the case with ICP and ISCP.

4.3 Comparative performance analysis of ARPMEC

In this section, we compare ARPMEC with existing protocols in terms of node energy consumption. The existing protocols are the new scheme for energy-efficient permutation routing in IoT networks (NESEPRIN) [1] and the Agent-based broadcast protocols for wireless heterogeneous node networks (ABBPWHN) [28]. For comparison purposes, we maintained the number of rounds to evaluate each protocol to 200 ($T = 200$), and then we varied the number of nodes from 125 to 500. We observed that the total energy consumption of a node when the number of items to route increases in the network also increases. From the obtained results presented in Fig. 6, we observe that

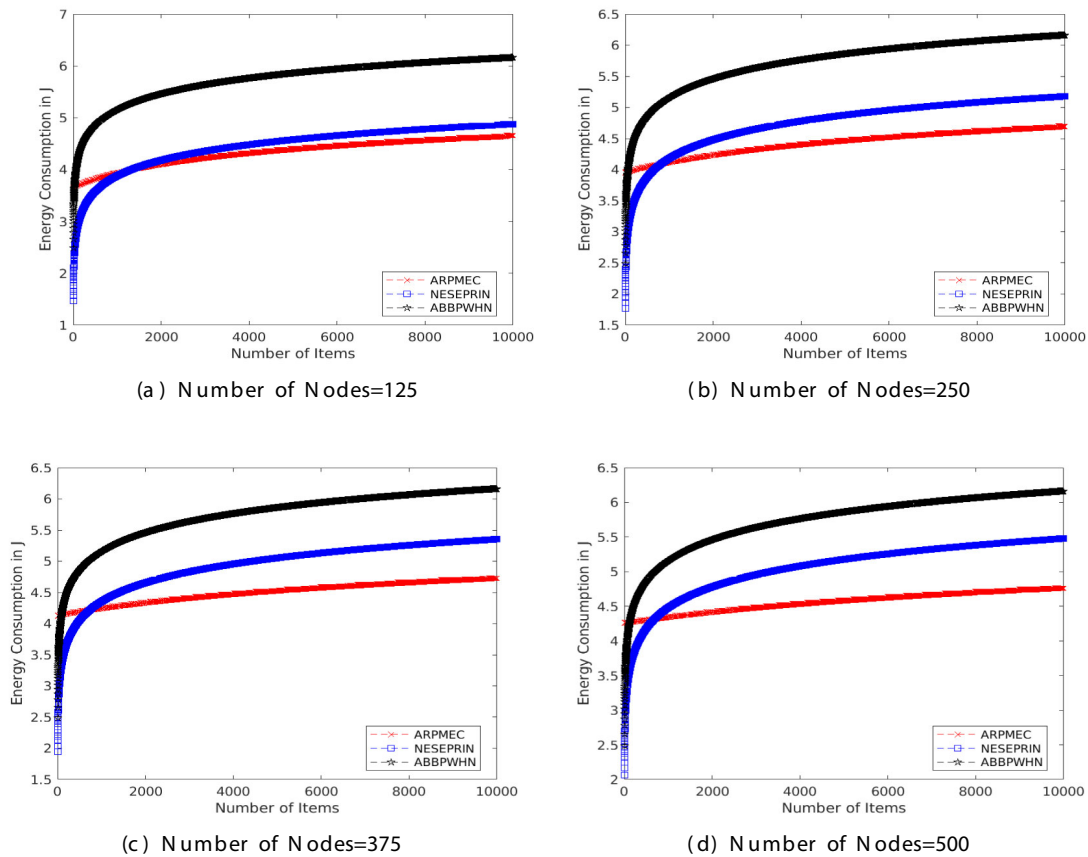


Fig. 6 Energy comparative study between ARPMEC and other existing protocols in IoT networks (Color figure online)

when the number of items to route is less than 700, ARPMEC (in red colour), consumes more energy than NESEPRIN (in blue colour) in all the scenarios (Fig. 6a–d) but is more energy-efficient than ABBPWHN (in black colour). However, when the number of items significantly increases, ARPMEC outperforms both the NESEPRIN and the ABBPWHN protocols. Given this observation, we can conclude that our proposed protocol is superior in energy conservation in comparison to NESEPRIN and ABBPWHN regarding data routing in IoT networks.

5 Conclusion

In this work, we have proposed an adaptive routing protocol for IoT network based on Mobile Edge Computing architecture (ARPMEC). Our proposed scheme is designed in two main phases. In the first phase, we partition the objects into clusters with elected cluster heads (CH). The clustering process uses the link quality estimation technique to make sure that nodes use the most accurate links that exist for data transmission. The generated clusters can communicate with each other through MEC servers which ensures that nodes' movements do not disconnect them

from other networks' objects. In the second phase, data routing is processed using the previously generated clusters. The complexity analysis of ARPMEC shows that it requires $O(T\Delta)$ time slots to execute, where T is the number of rounds and Δ is the number of items to be routed. Moreover, the simulation results show that ARPMEC is more energy efficient than the existing routing protocols for IoT networks, namely ABBPWHN [28] and NESEPRIN [1]. Based on the previously highlighted results, even if the clustering phase of our protocol consumes more energy than existing clustering protocols, we observed that overall, our routing protocol including the clustering protocol is energy efficient (the energy consumption during the data routing phase is low due to the use of bests links for data routing). These results also show that our proposed protocol when it is simultaneously executed with the clustering protocol we proposed in this paper, helps to efficiently manage the end-user devices during the data routing in an MEC ecosystem which is our main objective of our study. As a result, we recommend the use of ARPMEC when it comes to deploying IoT networks based on MEC architectures. Since the protocol is executed in rounds, a node can join the network at the beginning of any round without interrupting the existing

communications, and this makes our protocol scalable in the context of IoT networks.

Although the generated results are encouraging, more work can be done to improve our proposed scheme. The fault tolerance and security aspects have not been considered in our proposal. In future works, we plan to consider fault tolerance and security in the design of ARPMEC as we know that IoT networks are ad hoc networks which are more vulnerable to security attacks than classic networks. Furthermore, we plan to propose a federated learning-based algorithm to improve the efficiency of ARPMEC.

Author contributions M.L.F.S.: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing—original draft. M.V.: Conception and design of study, Writing—review & editing. V.K.T.: Conception and design of study, Writing—review & editing.

Funding Open access funding provided by University of Limpopo. The paper was funded by the National Research Foundation of South Africa (Grant Number: 141918).

Data availability The data used to support the findings of this study are available from the corresponding author upon request.

Declarations

Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Bomgni, A.B., Foko Sindjoun, M.L., Tchibonsou, D.K., Velepini, M., Myoupo, J.F.: NESEPRIN: a new scheme for energy-efficient permutation routing in IoT networks. *Comput. Netw.* **214**, 109162 (2022). <https://doi.org/10.1016/j.comnet.2022.109162>
- Zhao, L., Zhao, Z., Zhang, E., Hawbani, A., Al-Dubai, A.: A digital twin-assisted intelligent partial offloading approach for vehicular edge computing. *IEEE J. Sel. Areas Commun.* (2023). <https://doi.org/10.1109/JSAC.2023.3310062>
- Zhao, L., Zhang, E., Wan, S., Hawbani, A., Al-Dubai, A.Y., Min, G., Zomaya, A.Y.: MESON: a mobility-aware dependent task offloading scheme for urban vehicular edge computing. *IEEE Trans. Mob. Comput.* (2023). <https://doi.org/10.1109/TMC.2023.3289611>
- Zeitz, T.: NP-hardness of shortest path problems in networks with non-FIFO time-dependent travel times. *Inf. Process. Lett.* **179**, 106287 (2023). <https://doi.org/10.1016/j.ipl.2022.106287>
- Khanh, Q.V., Chehri, A., Minh, Q.N., Nguyen, V.-H., Ban, N.T.: An efficient routing algorithm for self-organizing networks in 5G-based intelligent transportation systems. *IEEE Trans. Consum. Electron.* (2023). <https://doi.org/10.1109/TCE.2023.3329390>
- Foko Sindjoun, M.L., Velepini, M., Minet, P.: Combining learners to predict link quality in wireless IoT networks. In: MELECON2022: 21st IEEE Mediterranean Electrotechnical Conference (2022)
- Bomgni, A.B., Futé, E.T., Foko Sindjoun, M.L., Djamegni, C.T.: A tree-based distributed permutation routing protocol in multi-hop wireless sensors network. *Wirel. Sens. Netw.* **8**, 93–105 (2016). <https://doi.org/10.4236/wsn.2016.86010>
- Foko Sindjoun, M.L., Alain Bertrand Bomgni, E.T.F., Chendjou, J.: An improved version of lambda architecture. In: CARI 2018: African Conference on Research in Computer Science and Applied Mathematics, Stellenbosch, South Africa, pp. 247–255 (2018)
- Chekired, D.A., Togou, M.A., Khoukhi, L., Ksentini, A.: 5G-slicing-enabled scalable SDN core network: toward an ultra-low latency of autonomous driving service. *IEEE J. Sel. Areas Commun.* **37**(8), 1769–1782 (2019). <https://doi.org/10.1109/JSAC.2019.2927065>
- Miller, R.: Autonomous Cars Could Drive a Deluge of Data Center Demand. *Data Center Frontier*, Atlanta (2017)
- Foko Sindjoun, M.L., Velepini, M., Bomgni, A.B.: A MEC architecture for a better quality of service in an autonomous vehicular network. *Comput. Netw.* **219**, 109454 (2022). <https://doi.org/10.1016/j.comnet.2022.109454>
- Filali, A., Abouamar, A., Cherkaoui, S., Kobbane, A., Guizani, M.: Multi-access edge computing: a survey. *IEEE Access* **8**, 197017–197046 (2020). <https://doi.org/10.1109/ACCESS.2020.3034136>
- Ali, B., Gregory, M.A., Li, S.: Multi-access edge computing architecture, data security and privacy: a review. *IEEE Access* **9**, 18706–18721 (2021). <https://doi.org/10.1109/ACCESS.2021.3053233>
- Yala, L., Frangoudis, P.A., Ksentini, A.: Latency and availability driven VNF placement in a MEC-NFV environment. In: 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1–7 (2018). <https://doi.org/10.1109/GLOBECOM.2018.8647858>
- Cziva, R., Anagnostopoulos, C., Pezaros, D.P.: Dynamic, latency-optimal VNF placement at the network edge. In: IEEE INFOCOM 2018—IEEE Conference on Computer Communications, pp. 693–701 (2018). <https://doi.org/10.1109/INFOCOM.2018.8486021>
- Aruna, O., Sharma, A.: An adaptive routing protocol in flying ad hoc networks. *J. Discret. Math. Sci. Cryptogr.* **25**(3), 757–770 (2022). <https://doi.org/10.1080/09720529.2021.2016223>
- Ramkumar, J., Vadivel, R.: Multi-adaptive routing protocol for internet of things based ad-hoc networks. *Wirel. Pers. Commun.* **120**, 887–909 (2021). <https://doi.org/10.1007/s11277-021-08495-z>
- Yi, B., Wang, X., Li, K., Das, S., Huang, M.: A comprehensive survey of network function virtualization. *Comput. Netw.* **133**, 212–262 (2018). <https://doi.org/10.1016/j.comnet.2018.01.021>
- Kreutz, D., Ramos, F.M.V., Verissimo, P., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: a comprehensive survey. <http://arxiv.org/abs/1406.0440> (2014)
- Barakabitze, A.A., Ahmad, A., Mijumbi, R., Hines, A.: 5G network slicing using SDN and NFV: a survey of taxonomy, architectures and future challenges. *Comput. Netw.* **167**, 106984 (2020). <https://doi.org/10.1016/j.comnet.2019.106984>

21. Peng, H., Ye, Q., Shen, X.S.: SDN-based resource management for autonomous vehicular networks: a multi-access edge computing approach. *IEEE Wirel. Commun.* **26**(4), 156–162 (2019). <https://doi.org/10.1109/MWC.2019.1800371>
22. Peng, H., Shen, X.: Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks. *IEEE Trans. Netw. Sci. Eng.* **7**(4), 2416–2428 (2020). <https://doi.org/10.1109/TNSE.2020.2978856>
23. Bhamare, D., Jain, R., Samaka, M., Erbad, A.: A survey on service function chaining. *J. Netw. Comput. Appl.* **75**, 138–155 (2016). <https://doi.org/10.1016/j.jnca.2016.09.001>
24. Vu Khanh, Q., Vi Hoai, N., Van Dang, A., Nguyen Minh, Q.: An integrating computing framework based on edge-fog-cloud for internet of healthcare things applications. *Internet Things* **23**, 100907 (2023). <https://doi.org/10.1016/j.iot.2023.100907>
25. Myoupo, J.F., Nana, B.P., Tchendji, V.K.: Fault-tolerant and energy-efficient routing protocols for a virtual three-dimensional wireless sensor network. *Comput. Electr. Eng.* **72**, 949–964 (2018). <https://doi.org/10.1016/j.compeleceng.2018.02.012>
26. Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, J., Alexander, R.: RPL: IPv6 routing protocol for low-power and lossy networks. IETF, RFC 6550 (2012)
27. Shah, Z., Levula, A., Khurshid, K., Ahmed, J., Ullah, I., Singh, S.: Routing protocols for mobile Internet of Things (IoT): a survey on challenges and solutions. *Electronics* (2021). <https://doi.org/10.3390/electronics10192320>
28. Lakhlef, H., Bouabdallah, A., Raynal, M., Bourgeois, J.: Agent-based broadcast protocols for wireless heterogeneous node networks. *Comput. Commun.* **115**, 51–63 (2018). <https://doi.org/10.1016/j.comcom.2017.10.020>
29. Foko Sindjoug, M.L., Minet, P.: Wireless link quality prediction in IoT networks. In: The 8th IFIP/IEEE International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks—(PEMWN 2019) (2019). <https://doi.org/10.23919/PEMWN47208.2019.8986920>
30. Foko Sindjoug, M.L., Minet, P.: Estimating and predicting link quality in wireless IoT networks. *Ann. Télécommun.* **77**(5–6), 253–265 (2022). <https://doi.org/10.1007/s12243-021-00835-1>
31. Senel, M., Chintalapudi, K., Lal, D., Keshavarzian, A., Coyle, E.J.: A Kalman filter based link quality estimation scheme for wireless sensor networks. In: IEEE GLOBECOM 2007—IEEE Global Telecommunications Conference (2007). <https://doi.org/10.1109/GLOCOM.2007.169>
32. Millan, P., Molina, C., Medina, E., Vega, D., Braem, B., Blondia, C.: Tracking and predicting link quality in wireless community networks. In: IEEE WiMob 2014, Larnaca, Cyprus (2014)
33. Marinca, D., Minet, P.: On-line learning and prediction of link quality in wireless sensor networks. In: IEEE GLOBECOM 2014, Austin, United States (2014). <https://hal.inria.fr/hal-01094446>
34. Baccour, N.: Radio link quality estimation in wireless sensor networks: a survey. *ACM Trans. Sens. Netw.* **8**(4), 1–33 (2012). <https://doi.org/10.1145/2240116.2240123>
35. Draves, R., Padhye, J., Zill, B.: Routing in multi-radio, multi-hop wireless mesh networks, pp. 114–128 (2004). <https://doi.org/10.1145/1023720.1023732>
36. Couto, D., James, D.: High-throughput routing for multi-hop wireless networks. PhD Thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science (2004)
37. Brun-Laguna, K., Henrique Gomes, P., Minet, P., Watteyne, T.: Moving beyond testbeds? Lessons (we) learned about connectivity. *IEEE Pervasive Computing, Special Issue on Beyond Testbeds: Real-World IoT Deployments* **17**(4) (2018)
38. Baccour, N., Koubâa, A., Youssef, H., Ben Jamâa, M., Rosário, D., Alves, M., Becker, L.B.: F-LQE: a fuzzy link quality estimator for wireless sensor networks. In: Silva, J.S., Krishnamachari, B., Boavida, F. (eds.) *Wireless Sensor Networks*, pp. 240–255. Springer, Berlin, Heidelberg (2010)
39. Fonseca, R., Gnawali, O., Jamieson, K., Levis, P.: Four bit wireless link estimation. In: *ACM HotNets-VI*, Atlanta, GA (2007)
40. Rekik, S., Baccour, N., Jmaiel, M., Drira, K.: Low-power link quality estimation in smart grid environments. In: *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1211–1216 (2015). <https://doi.org/10.1109/IWCMC.2015.7289255>
41. Xu, Y., Lee, W.C.: Exploring spatial correlation for link quality estimation in wireless sensor networks. In: *Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06)*, pp. 10–211 (2006)
42. Sun, W., Lu, W., Li, Q., Chen, L., Mu, D., Yuan, X.: WNN-LQE: wavelet-neural-network-based link quality estimation for smart grid WSNs. *IEEE Access* **5**, 12788–12797 (2017). <https://doi.org/10.1109/ACCESS.2017.2723360>
43. Chapter 3—dynamic virtual channel routers with congestion awareness. In: Wang, Z., Ma, S., Huang, L., Lai, M., Shi, W. (eds.) *Networks-On-Chip*, pp. 77–105. Morgan Kaufmann, Oxford (2015). <https://doi.org/10.1016/B978-0-12-800979-6.00003-2>
44. Alexandros, M., Georgios, Z.P., Xenofon, F., Atis, E., George, O., Theo, T.: Impact of guard time length on IEEE 802.15.4e TSCH energy consumption. In: *IEEE International Conference on Sensing, Communication and Networking* (2016). <http://georgiospapadopoulos.com/docs/poster/ieeesecon2016-poster.pdf>
45. Foko Sindjoug, M.L., Bomgni, A.B., Futé, E.T., Djamegni, C.T., Chalhoub, G.: ISCP: an instantaneous and secure clustering protocol for wireless sensor networks. *Netw. Protoc. Algorithms* **10**(1), 65–82 (2018). <https://doi.org/10.5296/npa.v10i1.12574>
46. Alazzawi, L., Elkateeb, A.: Performance evaluation of the WSN routing protocols scalability. *J. Comput. Netw. Commun.* (2008). <https://doi.org/10.1155/2008/481046>
47. Kong, L., Xiang, Q., Liu, X., Liu, X.-Y., Gao, X., Chen, G., Wu, M.-Y.: ICP: instantaneous clustering protocol for wireless sensor networks. *Comput. Netw.* (2016). <https://doi.org/10.1016/j.comnet.2015.12.021>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Miguel Landry Foko Sindjoun is currently pursuing a postdoctoral research fellowship at the University of Limpopo (South Africa), Department of Computer Science. He is Lecturer in the Department of Computer Engineering, Fotso Victor University Institute of Technology, University of Dschang (Cameroon). He obtained the Ph.D. in 2020, the M.S. degree in 2015, and the B.S. degree in 2012, all in computer science at the University of Dschang. His

current research interests include parallel algorithm, distributed systems, wireless sensor networks, internet of things, big data, cognitive radio networks and edge computing.



Vianney Kengne Tchendji is currently an Associate Professor of computer science at the University of Dschang, Dschang, Cameroon. He received the Ph.D. degree in computer science from the University of Picardie Jules Verne, Amiens, France, in June 2014. He has served as member of program committee of international conferences, and reviewer for many international journals in computer science.

His current research interests include parallel algorithms and architectures, network virtualization, internet of things, ad hoc networking and cyber security.



Mthulisi Velempini is an IEEE member and active researcher in Medium Access Control protocols, Routing protocols and Security in Computers. He is an emerging researcher in Wireless Access Network Technologies.