Panda and Supriya *J Wireless Com Network* (2022) 2022:118
https://doi.org/10.1186/s13638-022-02200-9

EURASIP Journal on Wireless Communications and Networking

RESEARCH **Open Access**

Efficient data transmission using trusted third party in smart home environments

Niharika Panda* and M. Supriya

*Correspondence: p_niharika@blr.amrita.edu

Department of Computer Science and Engineering, Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, Bengaluru, India

Abstract

Currently, deployed Internet of Things (IoT) technology acts as a passive observer of the environment that sends data to a remote location. Developing and deploying future IoT applications will need re-tasking this one-way behaviour in a reliable manner. A novel computationally tractable optimization technique that can accept cross-layer resource configurations and focus on network enhancement with longevity should be created for the smart home, one of the heterogeneous IoT applications. This study shows different smart home architectures in static and mobile environments, taking into account some of the challenges like orchestration, mobility, and range in IoT. For network communication, routing protocol over 6LoWPAN (RPL) is used. The goal of the work is to optimize the communication network in both static and mobile environment. To attain the goal, this paper proposes an algorithm that improves the path selection by modifying the existing objective functions of RPL. The proposed smart home architectures are analysed and compared based on different parameters such as packet reception ratio, network overhead, throughput, average latency, and total energy consumption. Even when some of the devices in the smart home are mobile, the modified smart home-optimized path (MSHOP) is found to achieve a packet reception ratio of 99.93%, minimum latency of 0.9 s, and the highest total energy usage in the network of 3373 millijoules. In conclusion, proposed MSHOP outperforms all existing smart home architectures when considering network efficiency, time, and usability.

Keywords: Smart home, WSN, RPL, Mobility, Cooja

1 Introduction

Recent developments and the plethora of research have led to drastic growth in the Internet of Things [1]. Typically, IoT means communication from anywhere at any time between people and things [2]. The things can be static or mobile with power and energy constraints and can communicate using different protocols. The concept is continuously evolving, with an ever-increasing range of applications that lead to the development of new technologies [3]. During the deployment of an IoT network, sensor nodes produce a variety of traffic patterns, ranging from basic measurements like temperature readings to high-volume multimedia communications [2]. The nature of the sensor that gathers and disseminates data between source and sinks nodes which involve multi-hop communications affects the protocol stack design of wireless sensor networks (WSN) [4].

Different layers of WSN use various protocols and standards for wireless medium access and data transmission [5]. Applications of IoT include smart cities, vehicular networks, smart grids, smart agriculture, and cyber-physical systems, among which the smart home is the focus of this paper [6]. Understanding and analysing redundant sensor nodes in present smart homes are critical for designing the future smart city. It serves as the foundation for the growth of the smart housing industry by merging artificial intelligence algorithms into it.

Smart homes use information and communications technology (ICT) to make infrastructure accessible, interactive, and efficient through the integration of multiple technologies. Smart homes can securely manage a home's assets [7]. Data collection, transmission, and processing are the technical focus area in the smart home. In the era of smart homes, low-cost and low-power devices are interconnected in a network to control and monitor a wide range of home appliances remotely and intelligently [8]. The gateway connects the smart nodes to send data from the wireless sensor network to the Internet [9]. Data from nodes and sensors are read from an internal database and displayed graphically in an easy-to-understand manner through a simple, secure web-based system [10].

Simulators and testbeds are the pioneers for experiments before the realistic implementations in IoT [11]. There are three categories of simulators for IoT research [12]. The first category is full stack simulators, aiming to provide end-to-end support for all IoT elements. The second is big data processing simulators that focus on cloud performance [13]. The third category is the network simulators famous for wireless sensor network research. IoT research uses many WSN simulators, namely CupCarbon, Cooja, OMNeT++ [14], NS-3, and QualNet. Cooja is an emulator capable of instruction-level emulation of node firmware execution in a simulated wireless communications environment [15]. It is a companion simulator included with the Contiki Operating System (OS), one of the most widely used OSs for programming IoT sensors. Cooja has WSN motes that access most standards and protocols [3]. As a result, researchers can simulate realistic scenarios using standard application-layer protocols like message queue telemetry transport (MQTT) and constrained application protocol (CoAP) over 802.15.4 and low-power wireless personal area networks using IPv6 (6LoWPAN). With modest modifications, the simulation firmware running on virtual nodes can be deployed to real physical hardware, bridging the gap between the proof-of-concept and prototype phases [16]. For the above reasons, the current work uses the Cooja simulator.

The contribution of this paper is as follows

- Proposed different smart home architectures with the help of IoT technologies.
- Optimal configuration selection of the smart home scenarios is achieved with trusted third party (TTP).
- Modified smart home implemented the concept of event timer to achieve optimized path to the server node.

This work aims to propose different smart home architectures with the help of IoT technologies. This work developed the proposed architecture considering various parameters, namely packet reception ratio (PRR), throughput, average latency, network overhead, convergence time, and total energy consumption. The modified path selection approaches achieve optimal configuration selection of the smart home scenarios.

2 Related work

When heterogeneous IoT comes into picture, several unresolved challenges come along. Figure 1 shows the challenges and solutions for the IoT, which are elaborately covered in [1, 6, 17–20] and guide the future work. While designing IoT, challenges like resource allocation, multiple access, small cell deployment, and cell association need to be investigated further [21]. Especially when exploring smart home applications, security and privacy challenges have to be analysed and addressed. This section presents few of the issues and solutions with the help of the literature.

Examination of the RPL protocol in the context of a diverse traffic pattern described in [5] suggests a new protocol based on queue and workload conditions (QWL-RPL) that caused lighter load, less packet loss, and proper traffic distribution and packet reception ratio (PRR) of 91%. Kamgue et al. [22] show the way to create and implement a routing protocol for low-power lossy networks using the node's remaining energy as the primary routing measure. In various topologies (grid, random), RPL has been examined in terms of two objective functions (OF), i.e. minimum rank with hysteresis objective function (MRHOF) and objective function zero (OF0). RPL performance is found to be optimal for both OFs when the network density is between 30 and 40 nodes for Rx 60% using random or grid topology in a light density network [23]. The results reveal in [24] that changing the sending interval significantly impacts the performance of the RPL's OFs. Packet delivery ratio (PDR) increases, while the power consumption decreases and the sending interval increases. The RPL routing combines node link quality and residual energy ETX-ENR to minimize network latency and grow the sensor node's life. This showed a 10% improvement in network latency with energy still being available [25]. RPL-MRHOF deemed to be a better solution for PDR and latency in networks

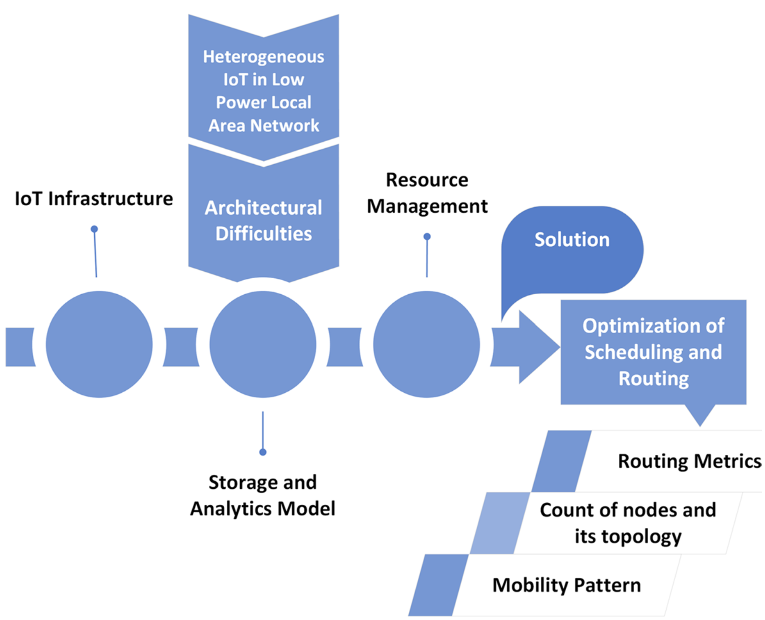

Fig. 1 Literature survey flow

with fewer nodes. OF0 appears more attractive as the node density climbs above 90 [3]. OF0 is appropriate for usage in networks with mobile nodes and power constraints [26]. Zungeru et al. [27] proposed a design and implementation of secure smart house switching system based on wireless communications and self-energy harvesting. An IFTTT-based smart home system model and an anti-tracking mutual authentication technique proposed in [28] deliver higher security and privacy.

Hatem et al. in [29] consider two metrics instead of one to produce more dependable and optimized routes. While employing MRHOF with ETX + Energy, it was discovered that the network's performance improved in terms of PDR without influencing the network's power usage. In addition, Brachman [30] discusses network transmission performance when created with various OFs. A simple but effective queue utilization-based RPL (QU-RPL), which considerably achieves load balancing, increases end-to-end packet delivery performance over normal RPL presented in [31]. The metrics of the minimal number of hops and the ETX are coupled to solve the problem mentioned above by inventing a new routing metric called SIGMA-ETX. It has an 87.18% PRR [32]. Considering network longevity, delay, energy consumption, and PRR, the neo-hybrid composite routing metric (NCRM) exceeds all others. For 30 nodes, PDR is 85% and Latency is 3.5 s [33]. An energy-efficient and path reliability-aware objective function (ERAOF) is designed for IoT applications that achieve a PDR of 93% for network size 20, 40, and 60 [34]. The work proposed in [35] discovered that the fuzzy-based hand-off strategy has a high level of dependability, delivering approximately 100% of data packets with a minimal latency. A mobility-aware energy-efficient parent selection algorithm for RPL proposed by Lamaaz et al. [36] supports random node mobility and a dynamic trickle algorithm. RPL methods have an average packet delivery ratio of 85.6% under 50% mobile nodes and 78.4% under 100% mobile nodes during the simulation period. A summary of the above discussed works is presented in Table 1.

Furthermore, a smart house architecture based on 6LoWPAN is designed that is connected to a root node with sensor nodes. It has been discovered that the sensor node's systematic arrangement is superior [23]. In [37], two SWIPT cooperative spectrum sharing strategies have been developed to enhance the spectrum and energy efficiency for the 6G-enabled cognitive IoT network. Through combined power and sub-carrier optimization, achievable rates of the cognitive IoT system with amplify-and-forward (AF) and decode-and-forward (DF) relaying mode are maximized while maintaining the primary system's target rate. A secure communication method for a NOMA-based UAV-MEC system toward a flying eavesdropper is suggested in the paper [38]. The simulation findings demonstrate that, in terms of system security and computation performance, the suggested approach outperforms the benchmark schemes. According to simulation results presented in [39], the proposed multiagent strategy optimizes the offloading effectively and achieves better system utility when compared to the single-agent method. In addition to advancing human computer interaction for gesture detection, the study presented in [40] seeks to advance the growth of Internet of Things technology. A particle filter optimization employing the maximum variance weight segmentation re-sampling approach has been suggested in [41] to enhance the particle filter performance.

In conclusion, the existing survey mostly focuses on IoT and wireless sensor networks. The majority of the study ignores a smart device's mobility. Few of the existing

Table 1 Objective functions-related assessment research

OF	Node	Topology	Environment	Tx–Rx	Metrics	PRR	Remark
OF	Node	Topology	Environment	Tx–Rx	Metrics	PRR	Remark
OF0, MRHOF [5]	20, 30, 40, 50, 100	Random	Static	Default	ETX	91%	Lighter load Less packet loss
MRHOF [22]	20	Random	Static	Tx = 80%	ETX Energy	98%	Increased transmission accuracy decreased energy consumption
OF0, MRHOF [24]	20, 30, 40, 45	Random	Static	Tx = 100% Rx = 20, 40, 60, 80, 100 m	ETX	98%	More packets delivered
MRHOF [25]	20, 30, 40	Random	Static	Tx = 50 m	ETX Energy ETX–ENR	100%	Increased data transmission decreased energy consumption
OF0, MRHOF [32]	20,40, 50, 100	Random	Static	Tx = 150 m	HC ETX SIGMA-ETX	87.18%	Increased PRR Increased network lifetime
OF0, MRHOF [33]	10, 20, 30, 40, 50	Random	Static	Tx = 100 m	ETX Energy SIGMA-ETX NCRM	95%	Less packet loss, Minimum latency
OF0, MRHOF [34]	20, 40, 60	Random	Static	Default	ETX ERAOF	95%	Reliable data communication
OF0, MRHOF [35]	12	Random	Mobile	Tx = 100 m	ETX 3FBMOF	Nearly 100%	reduced latency increased PRR
OF0, MRHOF [36]	10, 20, 30, 40, 50, 60	Random	Mobile	Tx = 50 m	ETX	85.6%	Reliable network increased sustainability
OF0, MRHOF [23]	15, 25, 30	Random	Static	Tx = 75 Rx = 20, 40, 80 m	HC ETX Energy	100%	Network lifetime increased Less energy consumption
OF, MRHOF [PROPOSED]	11, 21, 31, 34	Random, ellipse linear	Static mobile	Tx = 40, 70, 80 m Rx = 60, 90, 100 m	HC ETX MSHOP	Static-90.16% Mobile-99.93%	Maximum packet delivery reduced latency increased throughput

work which addresses this domain simulates experiments on static environment as presented in Table 1. On an average, PRR is found to be 91% for 30 nodes in a static environment. Mobility's impact on various network topologies and communication is less illustrated. The proposed work demonstrates the modified smart home architecture considering different set of static and mobile nodes and analyses the results for parameters identified.

In general, Internet of Things necessitate a great deal of effective orchestration. When we refer to different IoT applications, massive sensors deliver massive data to the server at the same time. Longevity and power consumption are more significant elements to be considered in an internet-enabled network. The following work has been proposed with all of these factors in mind.

3 Proposed methodology

Establishing viable heterogeneous IoT applications is complex due to the application requirements, heterogeneity of network architectures, and communication technologies. RPL, the routing protocol used for such a network, operates less efficiently with heavy traffic. In this section, the heterogeneous IoT problems are discussed and the solution is explained with the help of the proposed model.

3.1 Problem statement

3.1.1 IoT application orchestration

Choosing the optimal orchestration for an IoT application in a heterogeneous environment is essential. In addition, a computational model that can run and dynamically detect faults across many paths of IoT infrastructure and dynamically manage data available in IoT is to be considered for improvement. As a result, the suggested model will employ the modified smart home optimal path (MSHOP), which handles storage selection, analytics programming, and network fault detection.

3.1.2 Optimal configuration selection

IoT applications necessitate custom resource configurations at several layers, such as the things, edge, and cloud. We must consider data source location, battery, network latency, network type, endurance, and sensor types at the things layer. As a result, creating computationally tractable optimization approaches that can accommodate cross-layer resource configurations with security and privacy concerns is a priority. In addition, the architectural layers of smart home as an IoT application include a trusted third party in the proposed work.

3.1.3 Holistic monitoring

Monitoring IoT application graphs is crucial for automatically predicting and detecting anomalies and their primary causes. As a result, new monitoring strategies that provide detailed data flow and QoS information for IoT applications are needed. Script analysis using Python 3.8 is used in this work to gain a more profound understanding of how data analysis jobs and resources work.

3.2 Solutions to stated problems

This article aims to create a smart home environment where TTP acts as a middleman framework for communication and data transmission in wireless sensor networks, resulting in increased energy efficiency, longer lifetime, and faster data delivery. An

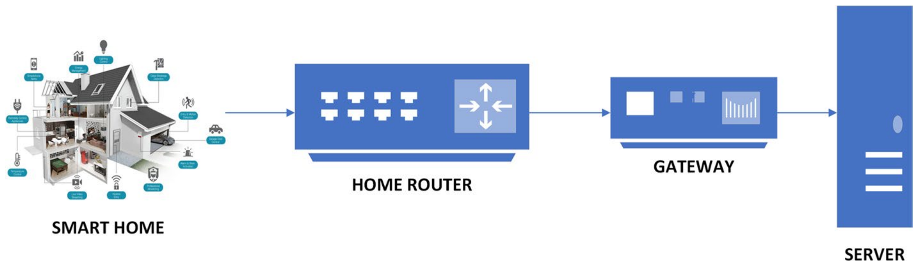

Fig. 2 Smart home with sensor nodes and routers

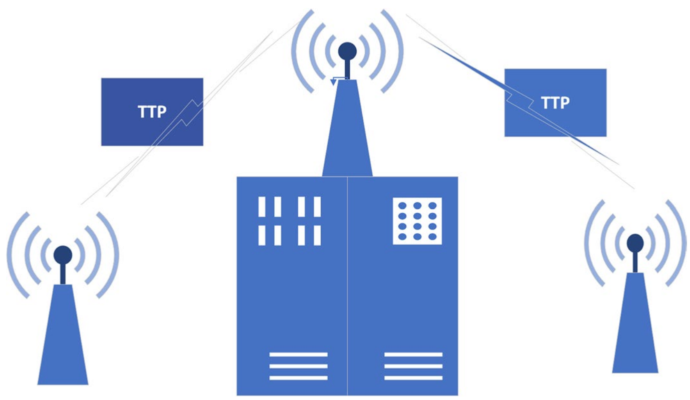

Fig. 3 Architecture of IoT gateway with TTP in internal layer

event timer is also included to help with optimal configuration choices. With the help of Python, data analysis is also implemented to monitor performance indicators.

3.3 Proposed method

3.3.1 System model

Smart home presented in Fig. 2 is an essential IoT application where rapid investigation is going on. It is an urban system that makes infrastructure interactive, accessible, and efficient through the use of ICT. The technical focal area of smart home is data collection, transmission, and processing. As smart homes develop, a number of challenges emerge including security and privacy, heterogeneity, reliability, large scale, legal and social aspects, sensor networks, multiple data fusion, and energy management.

3.3.2 Proposed orchestration

An IoT application is expected to address the performance parameters like energy conservation, orchestration, network connectivity, development techniques, etc. Additionally, it is also expected to consider interoperability, security, and privacy at a large scale. Keeping these challenges of IoT in mind, this paper proposes an efficient orchestration for a smart home application that uses a trusted third party (TTP) presented in Fig. 3. TTP is a platform that tries to facilitate interaction between two parties by establishing trust. It is a domain entity that can be trusted to accomplish a given task. It verifies that data are delivered to the intended destination and examine critical transaction communication between parties.

The TTP presented in Fig. 4 in this proposed has the following functionalities :

- It examines the nodes that are interested in establishing a connection.
- With low overhead, it provides trusted communication verification.
- Each node participating in the connection is given a unique id.
- When numerous nodes are attempting to connect with TTP, the time stamp determines which node will create the connection, reads the contents of an incoming packet, sets the priority for data transmission among the nodes, and increases the energy efficiency of the suggested system.
- The protocol ensures that an honest participant does not lose any data, while guaranteeing a faster data delivery.

The next subsection discusses various protocols and objective functions used in the simulation set-up.

3.3.3 Communication protocols and objective functions

The internet engineering task force (IETF) has established 6LoWPAN as a set of standards [42]. 6LoWPAN standards enable IPv6 to be effectively used over low-power, low-rate wireless networks on basic embedded devices by providing an adaption layer and optimizing related protocols. LoWPANs over IPv6 enable low-power devices to connect to the Internet [4]. 6LoWPAN enables network and link layer routing. The routers, host node, and edge router make up the architecture of its network. Communication between 6LoWPAN devices, the Internet, and other IP networks is handled by the edge router [43]. Figure 5 represents the architecture of 6LoWPAN.

Routing protocol RPL is a lossy and low-power IPv6 routing protocol with a distance vector [44]. RPL separates packet processing and forwarding from the routing optimization goal. It is committed to maintaining the security and integrity of messages [4]. It supports data path validation and loop detection routing optimization and groups nodes into a destination-oriented acyclic graph (DODAG), with each router identifying a set of parents. RPL can handle a variety of network traffic, including point to point, multipoint, and multicast [43, 44].

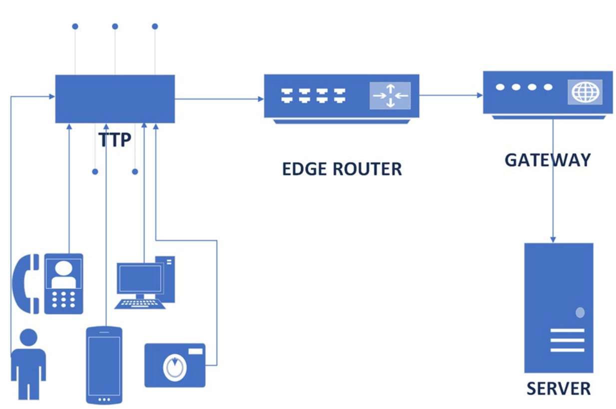

Fig. 4 TTP in smart home

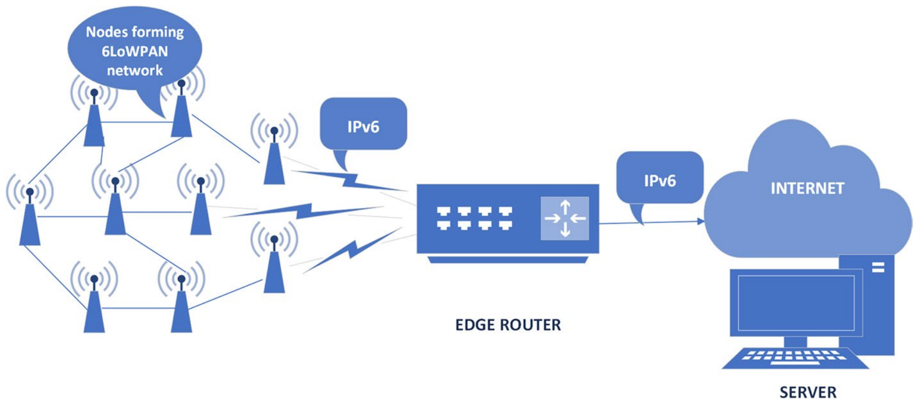

Fig. 5 6LoWPAN architecture

Objective function In order to build the directed acyclic graphs (DAGs), the user specifies an OF in RPL [4]. The OF specifies, directs, and guides RPL nodes in the construction and optimization of routes within an RPL instance to locate the best-optimized path [36]. The ROLL working group of the IETF has adopted the OF0 and the MRHOF as default functions for routing [45].

Objective function zero (OF0) With this objective function, the network is optimized by choosing neighbours with the lowest hop count. It is used to figure out how many hops are present between a source and a sink node. The parent of a node is determined by lowest rank. There is no change if both are equal [46].

Minimum rank with hysteresis objective function (MRHOF) It selects the parent node based on the minimum expected number of transmissions (ETX) [47]. The root ETX presented in Eq. 1 is calculated by adding the probability that a data packet will arrive successfully at its destination plus the probability that it will receive an acknowledgement packet [48].

$$\mathrm{ETX} = 1/(\mathrm{Df} * \mathrm{Dr}) \tag{1}$$

where Df = the probability that a packet will be received from an adjacent node; Dr = the probability that an acknowledgement will be received successfully.

4 Simulation setup

As mentioned in the initial sections, the work simulates the smart home environment using Contiki OS. It provides a variety of modules for various tasks. A sub-folder exists in the contiki/core/net/module that implements the MAC, IPV6, App-Layer, routing and security protocol stack [49]. When it comes to routing, the use of rpl lite or classic depends on whether the necessity is to use non-storing mode or storing mode. Some of the most essential files in the rpl lite subdirectory are rpl.c, rpl.h, rpl-dag.c, mrhof.c, of0.c, icmp6.c, and conf.h where the modifications have been done [50]. RPL also uses a set of control messages, such as the DODAG information object (DIO), DODAG information solicitation (DIS), and the DODAG destination advertisement Object (DAO) and DAO-ACK, to spread node information across the network topology. DIO messages contain data such as the objective function, rank, and node id. This message is broadcast to all other nodes in the network, encouraging them to join [3]. A DAO message is a request for permission to join the network issued by a child to a parent or root [18]. The DODAG information is pro-actively sought from neighbouring nodes via DIS messages. The considered parameters and their respective values are presented in Table 2.

4.1 Proposed architectures

The orchestration, TTP, and event timer are the most significant components of this project. When a network is established, communication occurs, with the best parent being selected in an upward manner from client to server. When the number of sensor nodes or client nodes increases, there is cause for concern. In this case, reachability to the server may be an issue for all nodes. Many people will be concerned about data loss and network performance. TTP proposed and designed in this work overcome all such scenarios in a heterogeneous IoT network, with event timer taking charge of network performance. A detailed approach to the proposed modified smart home optimized path (MSHOP) is presented in Fig. 6.

A detailed explanation of the approach for MSHOP development is provided below with the aid of a use case. A group of nodes form the basis of the network. Different topologies can be selected according to the requirements, after which the network is created. The structure of the network graph can be random, linear, or elliptical. After the network is configured, the environment should be assigned with an objective function.

Table 2 Simulation criterion and merit

Simulation criterion	Merit
Contiki version	Contiki-NG
Mote device	Z1 Zolertia
Mote voltage	3 V
Mote TX current	17.4 mA
Mote Rx current	18.8 mA
Mote CPU active	10 mA
Mote CPU LPM	0.023 mA
Packet size	127 bytes
Simulation time	1800 s
No. of client nodes	10, 20, 30
Transmission range	40, 70, 80
Interference range	60, 90, 100
Server nodes and TTPs	1 and 3
Environment	Static and mobile
Topology	Random, linear, ellipse
No. of mobile nodes	2, 4, 6
Lowest speed	1.0 m/s
Highest speed	80 m/s
Lowest pause time	1.0 m/s
Highest pause time	80 m/s
Max *X*-axis	2 s
Max *Y*-axis	15 min
Script data analysis	Python 3.8
PHY and MAC protocol	802.15.4 with CSMA
RTIMER_ARCH_SECOND	32,768 ticks per second

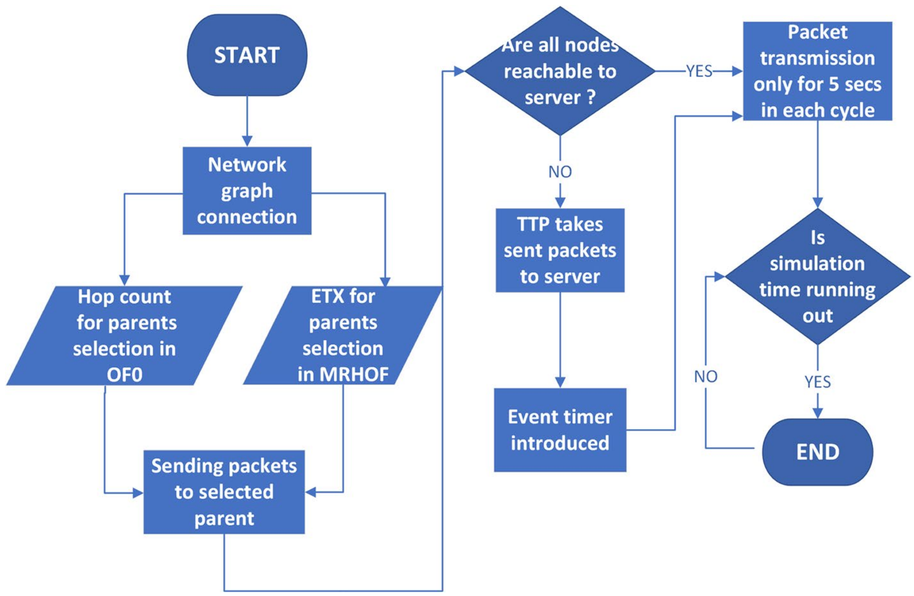

Fig. 6 Approach for MSHOP development

The objective function aids in determining the most efficient path for transferring data from a sink node to a server node. It comes in two types: OF0 and MRHOF. The least amount of hops is used to find the route when OF0 is chosen as the objective function, while in MRHOF, the route is chosen based on minimum ETX or the quantity of transmitted packets that were successfully received by the destination. As soon as the route is decided upon, the network is aware of the parent node and the transmission begins.

Every sink node is attempting to connect to the server in order to communicate. While there are numerous nodes, the server node is not covered by the transmission range. TTP is used as an intermediary structure if the network discovers that many nodes are unable to connect to the server. Now, the data and packets will be first sent to TTP. The received packets from sink nodes in TTP are then forwarded to the server in a group. If sink nodes can be reached, they can go on. To enhance the communication performance, this work introduces an event timer. Each cycle's packet transmission only lasts for duration allotted by the event timer. The entire process will come to an end after the simulation period is up.

The smart home scenario is essentially the focus of our efforts. A typical smart home might be a network of smart gadgets all attempting to communicate data to a server. This study takes into account the above system, with 10, 20, and 30 client nodes and a single sink/server node apiece naming them normal, average, and advanced smart home, respectively. With MRHOF, all nodes begin the data communication process by evaluating three different types of Tx/INT ranges and topologies. When a heterogeneous network has a large number of devices, such as 30 or more as presented in Fig. 7, there is a significant risk of data loss, increased latency, and reduced throughput. Another point of worry is the smart home's size. There will be no obstacles if the house is of single floor as there are less smart devices. The issue arises when the smart house has two or more storeys with a large number of smart gadgets. There could also be concerns with reachability, in light of the fact that a modified smart house is proposed in this work.

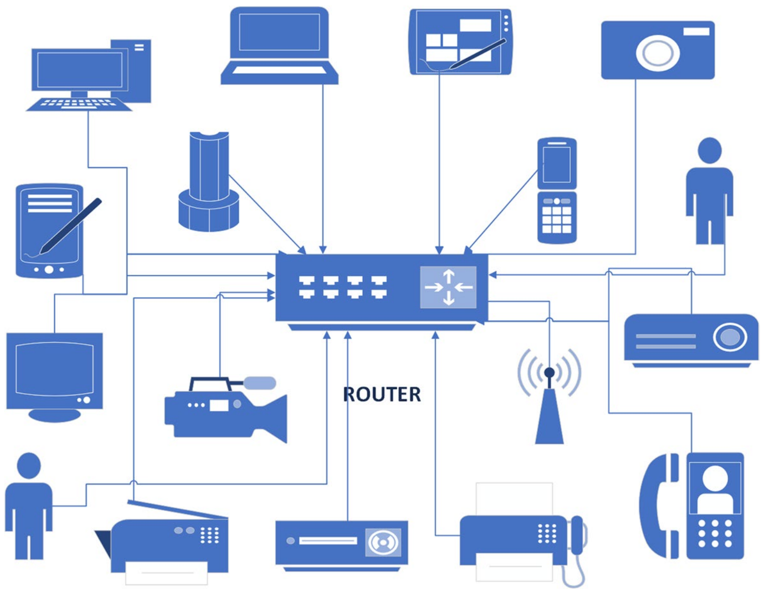

Fig. 7 Advanced smart home sensor and router communication

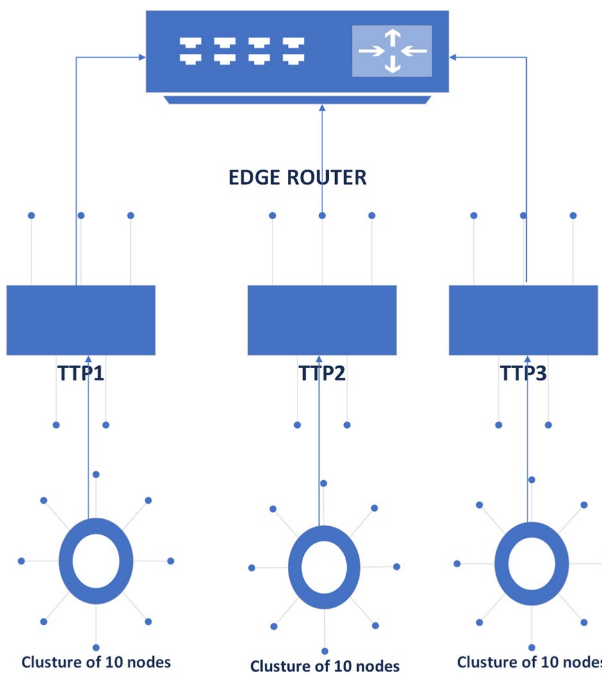

Fig. 8 Modified smart home communication structure

The clustering of nodes is created by the modified smart home (MSH) as depicted in Fig. 8. A smart house with advanced features is divided into three clusters, each with ten smart gadgets assuming each cluster belonging to one of the house's floors. Because a single home router will no longer be able to connect all 30 nodes, they will interact through TTP. As a result, the TTP functions as a go-between for the client nodes and the server, forwarding packets from the devices to the sink. 6LoWPAN protocol is used for communication between the clusters and TTP, TTP, and Edge Router.

Later comes the concept of an event timer in proposed work. An etimer is supplied in the rpl-udp client. The start interval is set at 15 s on the clock. Every 5-s interval, a message is sent. Then, a simple udp connection is formed, followed by the declaration of sender and recipient ids. The process begins with the creation of a timer object. If all of the aforementioned conditions are met, it creates an instance of the etimer. It is set with the function inside of the while loop. The software is essentially relinquishing control of the system because this function accepts arguments and sets the etimer to expire after interval, instructing the client nodes to pause until they receive an event. A client process receives an event notification when an etimer expires. As a result, it instructs the process to resume operating now that the etimer has run out and an event has been posted. It continuously sends data until the event timer expires. At the end, the details of communication and packet information values are saved to the output log file. The aforementioned is achieved by setting UIP CONF STATISTICS to 1 and adding printing statements in both client and server files for the number of packets received, sent, forwarded, and dropped. The algorithm is presented below. With addition of etimer to MSH, the proposed MSHOP is formed.

Algorithm 1 PROPOSED ALGORITHM

```
Include "Contiki.h"
Include "net/routing/routing.h"
Include "net/ipv6/simple_udp.h"
Include "sys/node_id.h"
Include "sys/etimer.h"
Define "LOG_Module APP"
Define UDP_CLIENT_PORT
Define UDP_SERVER_PORT
Define START_INTERVAL as 15 Clock_Seconds
Define SEND_INTERVAL as 5 Clock_Seconds
Set_up Simple_udp_connection
Set_up udp_client_process
Sender_id = sender_addr →u8[sizeof(sender_adde →u8)-1]
Receiver_id = receiver_addr →u8[sizeof(receier_adde →u8)-1]
Create a timer object
Process Begin
while true do
    etimer_set(&periodic_timer, SEND_INTERVAL)
    PROCESS_WAIT_EVENT_UNTIL(etimer_ex-
pired(&periodic_timer))
    Set event timer of 5 clock seconds
end while
if Routing_node_reachable() then
    Print Data send to receiver, count
    count++
else if
     then"Not reachable yet"
end if
Print number of packets sent uip.stat.ip.sent
Print number of packets received uip.stat.ip.recv
Print number of packets forwarded uip.stat.ip.forwarded
Print number of packets dropped uip.stat.ip.drop
Process End
```

Another crucial component of WSN is mobility. When a gadget moves, it disrupts a number of elements. Mobility can stifle data delivery, raise energy usage, and exacerbate road congestion, among other things. Mobile nodes are introduced in each form of smart house in this work. The effect of mobility is also boosted by the redesigned smart home and proposed algorithm. For the scenario described above, both objective functions have been investigated. Devices numbered 2 in the normal home, 4 in average home, and 6 in advanced home are mobile. In Table 2, node locations, mobility speed, and timings are listed.

4.2 Architectural implementation in simulator

The simulation starts with a simple smart home with 10 smart devices and a sink node/home router. The topologies followed for these 10 devices are random, ellipse, and linear. Each topology uses three different combinations of transmission and interference range to communicate between the sender and the sink. 40–60, 70–90, and 80–100 are the TX/INT ranges that have been considered. The simulation then moves on to a typical smart home with 20 devices and an advanced smart home with 30 devices, each with its own server node considering different topologies and ranges.

An advanced smart home with a linear topology represented in Fig. 9 is demonstrated by taking into account a larger number of nodes and attempting to achieve interoperability. The difficulty arises when a large number of devices attempt to connect to a single edge router. Poor connectivity, data loss, increased energy, and time usage are all the possibilities. A problem with scenario of a multi-story house is that it cannot connect all of the smart devices at different levels.

To overcome the aforementioned concerns, revised smart house architecture is presented called as modified smart home (MSH) in Figs. 10 and 11. Each cluster of nodes is linked to the TTPs closest to them. From the devices to the sink node, the trusted third party operates as a data carrier. The advanced smart home's 30 nodes are dispersed among three unique levels. A TTP is assigned to each level, and it connects a cluster of ten devices before establishing interoperability. TTP operates as a middleware entity in this case, reducing the amount of connections and making things less messy, resulting in increased security, reduced susceptibility, and no danger of data leaking. Heterogeneity due to various types of hardware and software utilized by different levels will be taken care by TTP. Task monitoring is challenging due to

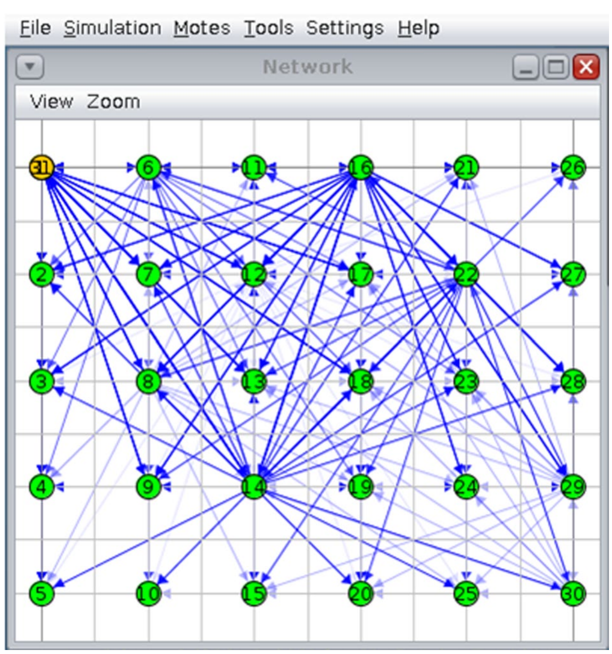

Fig. 9 Advanced smart home with linear topology

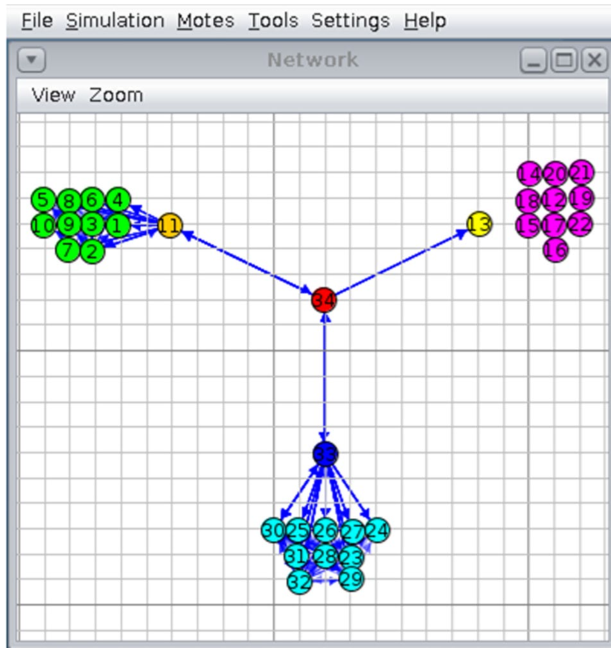

Fig. 10 Modified smart home with linear arrangement

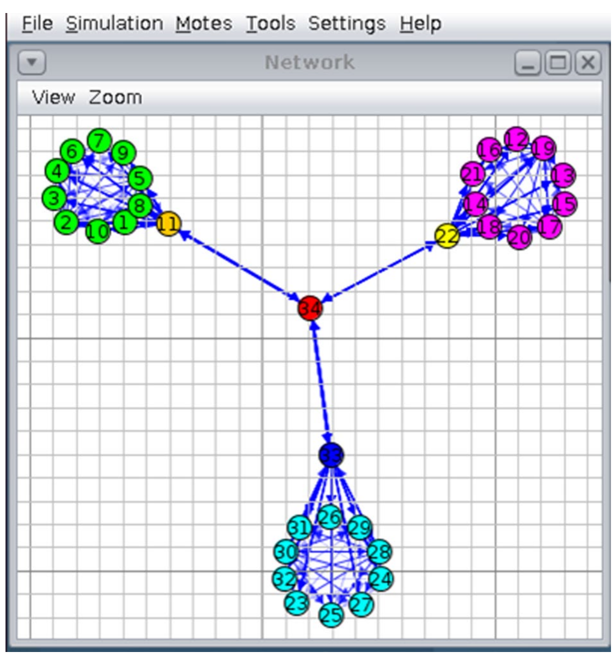

Fig. 11 Modified smart home with elliptical arrangement

the large-scale deployment. It also causes communication delays. This can be avoided since TTP makes the network more efficient in terms of both time and energy.

The above-mentioned proposed smart home architecture is implemented taking into consideration linear, ellipse, and random arrangement of the smart devices around the sink node.

There are a variety of smart devices that are mobile. Humans with smart phones or smart watches, movable doors, vacuum cleaners, smart doorbells, smart key chains are a few examples. Maintaining high interoperability, effective communication, energy-efficient, and long-lasting network while taking into account mobile devices is a difficult task. Due to the mobility of nodes, reliability becomes a major concern. Because TTP uses a queue to store the data received from a node and is fault tolerant, mobility will not be determined to communicate. Figure 12 depicts a network in which level 2 pink-coloured nodes and level 3 blue-coloured nodes are moved to smart home's level 1 and connected to TTP1 for communication.

4.3 Data analysis

In the IoT, objects that are effectively WSNs have been shown to create huge volumes of data. The information included in those files is referred to as big data. Data analysis, which is an important part of IoT, should be carried out in order to extract useful information from such data [51]. All network simulation tasks in the Cooja simulator generate execution log files which have no tools or programmes that can be used to summarize and analyse them. This slows down the research rate in complex network situations [52]. Python 3.8 is used to analyse the script. This processing script looks through the generated log files line by line. Following that, specific individual information on different performance metrics is provided. The process of data analysis is shown in Fig. 13. Figures 14 and 15 showcase a snippet of the simulation output.

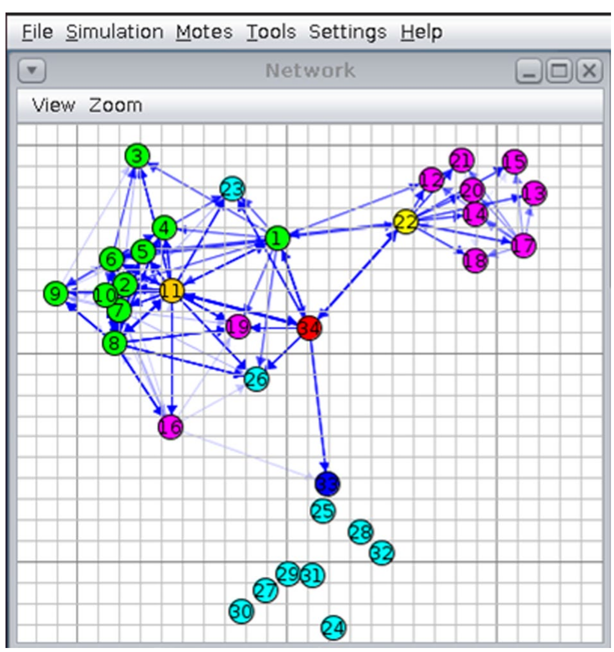

Fig. 12 Modified random smart home with mobility

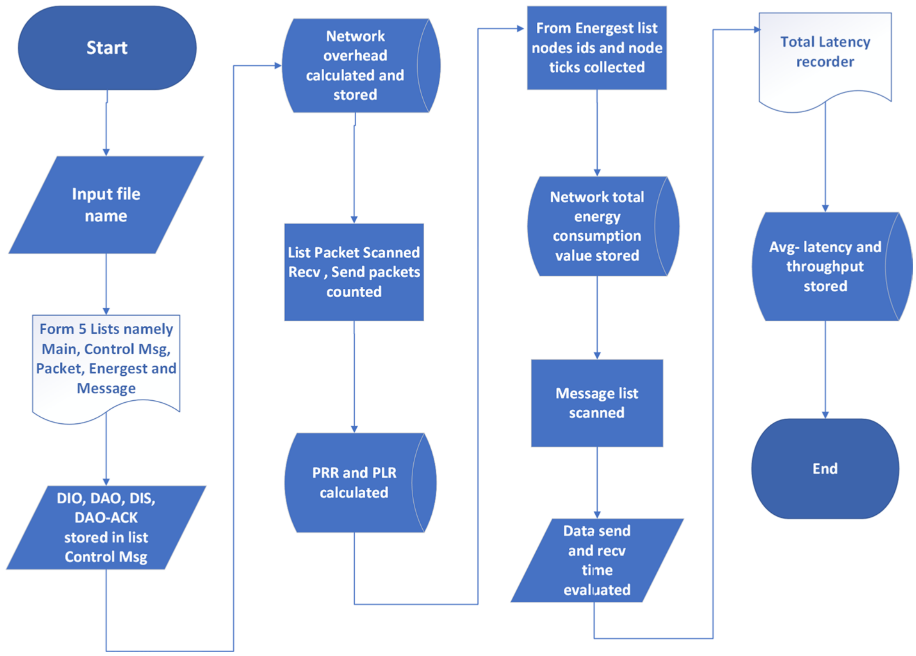

Fig. 13 Flow chart for data analysis using python 3.8

```
6348 02:01.716    ID:17   [INFO: Energest  ] Total time  :    1965906
6349 02:01.723    ID:17   [INFO: Energest  ] CPU         :      18352/   1965906 (9 permil)
6350 02:01.730    ID:17   [INFO: Energest  ] LPM         :    1947554/   1965906 (990 permil)
6351 02:01.736    ID:17   [INFO: Energest  ] Deep LPM    :          0/   1965906 (0 permil)
6352 02:01.742    ID:17   [INFO: Energest  ] Radio Tx    :       1183/   1965906 (0 permil)
6353 02:01.744    ID:22   [INFO: App       ] Not reachable yet
6354 02:01.747    ID:22   [INFO: Packet    ] Packet Send : 17
6355 02:01.749    ID:17   [INFO: Energest  ] Radio Rx    :    1964708/   1965906 (999 permil)
6356 02:01.749    ID:22   [INFO: Packet    ] Packet Recv : 25
6357 02:01.752    ID:22   [INFO: Packet    ] Packet Forwarded : 0
6358 02:01.755    ID:22   [INFO: Packet    ] Packet Drop : 0
6359 02:01.756    ID:17   [INFO: Energest  ] Radio total :    1965891/   1965906 (999 permil)
6360 02:01.772    ID:26   [INFO: App       ] Not reachable yet
6361 02:01.774    ID:26   [INFO: Packet    ] Packet Send : 23
6362 02:01.777    ID:26   [INFO: Packet    ] Packet Recv : 39
6363 02:01.780    ID:26   [INFO: Packet    ] Packet Forwarded : 0
6364 02:01.782    ID:26   [INFO: Packet    ] Packet Drop : 0
6365 02:01.787    ID:13   [INFO: Energest  ] --- Period summary #1 (59 seconds)
6366 02:01.790    ID:9    [INFO: App       ] Not reachable yet
6367 02:01.791    ID:13   [INFO: Energest  ] Total time  :    1966079
```

Fig. 14 Simulation output

The equivalent python script is presented in Figs. 16, 17, and 18. The mote output in the Cooja simulator is derived from the simulation environment and customized to display a node's performance every second. The network communication includes numerous cycles for all the sink nodes, as shown in Fig. 14. Every cycle makes an unique observation regarding the packets delivered, received, control messages sent, and energy usage. A python script is now built to summarize various performance parameters. Figure 16 shows the beginning of the script, where different lists are created w.r.t different performance metrics. Figures 17 and 18 show how latency and control overhead are calculated. Each of the identified performance metrics considered are calculated in the similar manner. There are over 100 distinct simulations carried out in both static and mobile environments for the topologies, objective

```
TOTAL FORWORDED PACKETS = {'ID:24': 0, 'ID:8': 0, 'ID:28': 68, 'ID:11': 3393, 'I
'ID:4': 0, 'ID:33': 2501, 'ID:1': 0, 'ID:27': 0, 'ID:7': 0, 'ID:14': 0, 'ID:15':
12': 0, 'ID:29': 40, 'ID:22': 2403, 'ID:26': 0, 'ID:9': 0, 'ID:5': 0, 'ID:19': 0
0, 'ID:32': 39, 'ID:34': 0}
TOTAL DROPPED PACKETS = {'ID:24': 0, 'ID:8': 0, 'ID:28': 0, 'ID:11': 0, 'ID:31':
0, 'ID:33': 0, 'ID:1': 0, 'ID:27': 0, 'ID:7': 0, 'ID:14': 0, 'ID:15': 0, 'ID:25'
29': 0, 'ID:22': 0, 'ID:26': 0, 'ID:9': 0, 'ID:5': 0, 'ID:19': 0, 'ID:17': 0, 'I
'ID:34': 0}
TOTAL SENT PACKETS = {'ID:24': 306, 'ID:8': 317, 'ID:28': 553, 'ID:11': 3929, 'I
18': 312, 'ID:4': 321, 'ID:33': 2979, 'ID:1': 328, 'ID:27': 308, 'ID:7': 329, 'I
16': 63, 'ID:10': 315, 'ID:12': 320, 'ID:29': 522, 'ID:22': 2919, 'ID:26': 313,
13': 306, 'ID:3': 125, 'ID:21': 308, 'ID:23': 311, 'ID:32': 520}
TOTAL RECEIVED PACKETS BY SERVER = 9901
TOTAL SENT PACKETS BY SERVER = 3929
TOTAL SENT PACKETS BY SERVER = 2919
TOTAL SENT PACKETS BY SERVER = 2979
TOTAL PCKET SENT BY ALL THE SENDERS = 19734
sent by clients packet only = 9907
PACKET RECEPTION RATIO = 99.93943676188553 %
PACKET LOSS RATION = 0.06056323811446873 %
Total energy consumption = 3366666.039425537
TOTAL SEND MESSAGE = 2445
TOTAL RECEIVED MESSAGE = 2444
PACKET DELIVERY RATIO = 99.95910020449898 %
TOTAL LATENCY = 2249886.942000001
AVERAGE LATENCY = 920.5756718494276
TOTAL SIMULATION TIME IN SEC = 1800.041
THROUGHPUT = 5588.437152264865
```

Fig. 15 Analysis output

```
filename = 'MOTEOUTPUTS/MOBILITY/LINEAR/newexperimentlinear-3-34-80-100mobile.txt'

#Creating list
MAIN=[]
RPL_CONTROL_MSG=[]
ENERGEST= []
MESSAGE=[]
PACKET=[]

with open(filename) as wsn:
    for line in wsn:
        #print(line.rstrip())


        if "INFO: Main" in line:
            MAIN.append(line)
        if "INFO: Energest" in line:
            ENERGEST.append(line)
        if "INFO: RPL-CM" in line:
            RPL_CONTROL_MSG.append(line)
        if "INFO: Message" in line:
            MESSAGE.append(line)
        if "INFO: Packet" in line:
            PACKET.append(line)
```

Fig. 16 Script analysis

functions, and Tx–INT ranges considered in this work. The outcome of this analysis is saved as script output, as shown in Fig. 15. This shows the total number of packets sent, forwarded, and received. The other performance parameters are also presented for analysis purpose.

```
TIME_MSG_ACK=Msg_ack_time+Msg_ack_time1
#print(TIME_MSG_ACK)
def Diffval(TIME_MSG_ACK,TIME_MSG_SENT):
    return list(set(TIME_MSG_ACK)-set(TIME_MSG_SENT))

LATENCY = Diffval(TIME_MSG_ACK,TIME_MSG_SENT)
#print(LATENCY)
total_latency = 0
for ele in range(0, len(LATENCY)):
    total_latency = total_latency + LATENCY[ele]

print("TOTAL LATENCY","=", total_latency)
Avg_latency=total_latency/count_recv
print("AVERAGE LATENCY","=",Avg_latency)
```

Fig. 17 Script analysis for latency

```
count_loop=0
for node_id in ID:
    count_loop+=1
    count_dio=0
    count_dis=0
    count_dao=0
    count_daoack=0

    for controlmsg in RPL_CONTROL_MSG:

        if "INFO: RPL-CM-DIO" in controlmsg and node_id in controlmsg:
            count_dio=count_dio+1
        if "INFO: RPL-CM-DIS" in controlmsg and node_id in controlmsg:
            count_dis=count_dis+1
        if "INFO: RPL-CM-DAO" in controlmsg and node_id in controlmsg:
            count_dao=count_dao+1
        if "INFO: RPL-CM-DAO-ACK" in controlmsg and node_id in controlmsg:
            count_daoack=count_daoack+1

    CONTROL_OVERHEAD=CONTROL_OVERHEAD+count_dio+count_dao+count_dis

    node_DIO[node_id]=count_dio
    node_DIS[node_id]=count_dis
    node_DAO[node_id]=count_dao
    node_DAOACK[node_id]=count_daoack
```

Fig. 18 Script analysis for control overhead

4.4 Performance metrics

This work considers the most responsive and necessary parameters, such as packet reception ratio, throughput, average latency, network overhead, convergence time, and total energy consumption, to evaluate network performance.

(1) Packet reception ratio (PRR): The ratio of the total number of packets delivered to the sink by all client nodes to the total number of packets received by the server [5].

$$\mathrm{PRR} = \frac{\mathrm{Pr}}{\mathrm{Ps}} * 100\% \tag{2}$$

where Pr denotes total packets received by the edge router; Ps denotes total number of packets send by all the smart devices.

(2) Throughput: The rate at which a message is successfully delivered via a communication channel is throughput.

$$\text{Throughput} = (\text{Dp} * \text{Sp} * 8)/(S.T) \text{ bits/s} \tag{3}$$

where Dp = total number of delivered packets to the sink; Sp = size of the packets; S.T = total simulation time.

(3) Average latency: The average of total time it takes for a bit of data to go from one communication endpoint to another across the network in each cycle is average latency.

$$\begin{aligned} \text{Total latency} &= \sum_{p=1}^{n} (\text{Tr}(p) - \text{Ts}(p)) \quad \text{ms} \\ \text{Avg latency} &= \frac{\text{T.L}}{\text{T.R}(p)} \quad \text{ms} \end{aligned} \tag{4}$$

where p is packet; Tr denotes time of packet reception; Ts denotes time of packet sent; T.L is total latency; T.R(p) denotes total received packets by the sink.

(4) Network overhead: Control packet overhead refers to the number of DIO, DAO, and DIS messages generated during the network's operation.

$$\text{Network overhead} = T_{\text{DIS}} + T_{\text{DIO}} + T_{\text{DAO}} \tag{5}$$

where T_{DIS} denotes total DODAG information solicitation (DIS) of all nodes; T_{DIO} represents total DODAG information object (DIO) of all nodes; T_{DAO} represents total DODAG destination advertisement Object (DAO) of all nodes.

(5) Convergence time: The amount of time it takes for all of the nodes to join the network is known as convergence time [5]. Basically the network time at which the first and the last node joined the DODAG has to be stored.

$$\text{Convergence time} = L_{\text{DIO}}T - F_{\text{DIO}}T \quad \text{ms} \tag{6}$$

where $L_{\text{DIO}}T$ = the time at which last node joined the DAG; $F_{\text{DIO}}T$ = the time at which first node joined the DAG.

(6) Total energy consumption: It includes power consumption of CPU (idle), LPM (low power mode of CPU), Tx (transmission), and Rx (reception) in each cycle of communication [22]. Statewise energy consumption is calculated using Eq. 7 through 10.

$$\text{CPU} = (\text{CPU} * 10 * 3)/32{,}768 \tag{7}$$

$$\text{LPM} = (\text{LPM} * 0.23 * 3)/32{,}768 \tag{8}$$

$$\text{Tx} = (\text{Tx} * 17.4 * 3)/32{,}768 \tag{9}$$

$$\text{Rx} = (\text{Rx} * 18.8 * 3)/32{,}768 \tag{10}$$

$$\text{T.E}_\text{C} = \text{CPU} + \text{LPM} + \text{Tx} + \text{Rx} \quad \text{mJ} \tag{11}$$

where T.E_C = total energy consumption.

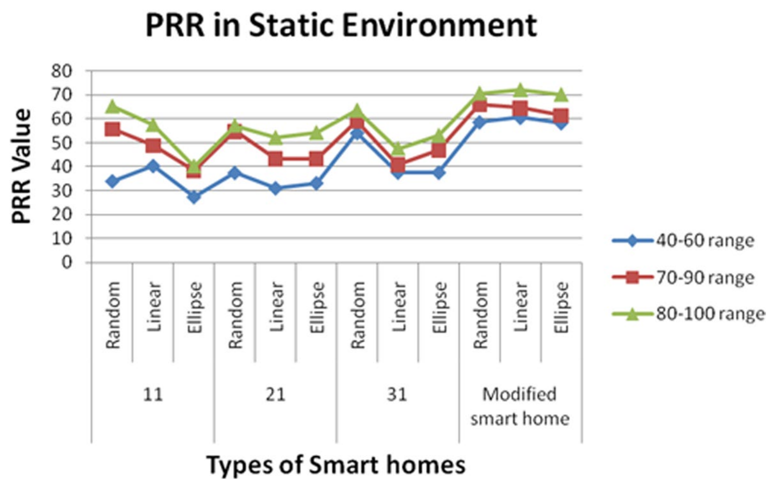

Fig. 19 PRR in static environment

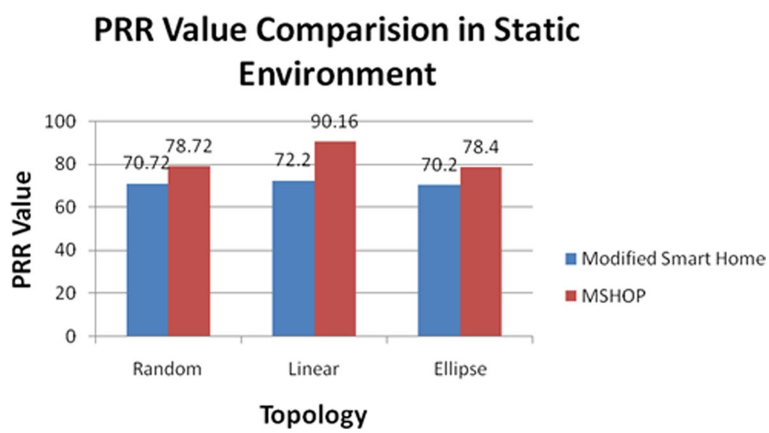

Fig. 20 PRR compared between MSH and MSHOP in static environment

5 Results and discussion

The outcome of the simulations is examined in this section. There are different smart home scenarios examined and compared taking into consideration multiple types of topologies and Tx–Rx ranges. It starts with basic 10 client nodes, average 20 sender nodes, and advanced 30 sender nodes in smart homes. Later, the performance is analysed for MSH and MSHOP and compared to show why the proposed architecture and topology are better than existing. Packet reception ratio, throughput, average latency, network overhead, convergence time, and energy consumption are some of the parameters that have been studied in this work.

Packet reception ratio which is calculated as per Eq. 2 is compared for different smart home scenarios considering the parameters, namely sender nodes count, transmission range, and ability to group in a static environment, the results of which are presented in Fig. 19. MRHOF is chosen as the objective function, and in terms of all ranges and topologies, all modified smart homes outperform. The modified smart home PRR score is then compared to MSHOP, which is the highest performing option. In a linear topology, it received a PRR of 90.16% for the 80–100 ranges as shown in Fig. 20. MSHOP is conscious of congestion and workload. It has the capability of reducing traffic congestion.

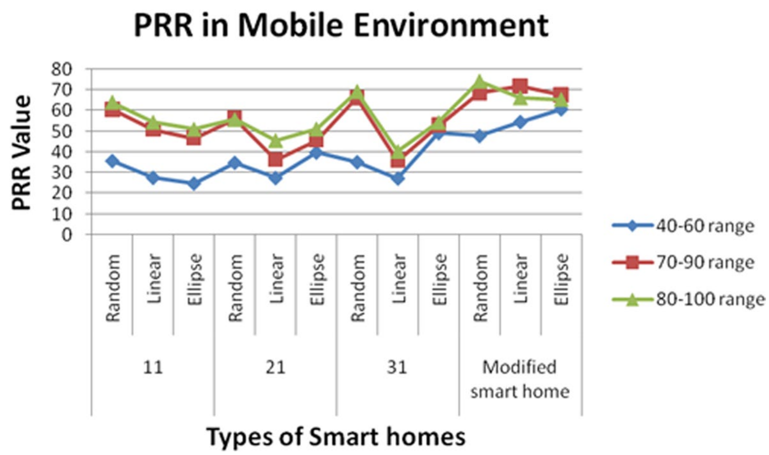

Fig. 21 PRR in mobile environment

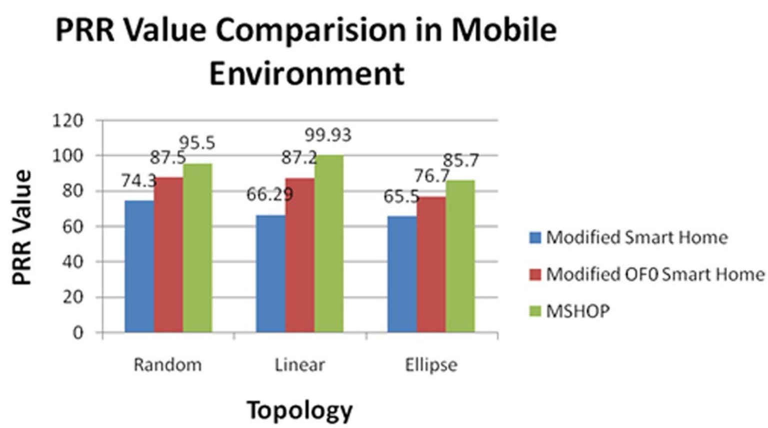

Fig. 22 PRR compared between MSH with MRHOF, MSH with OF0, and MSHOP in mobile environment

As a result, it is able to maximize PRR. The PRR rate in any smart house under the 80–100 Tx/INT range gives better result than other ranges. When it comes to topologies, linear is believed to be the best for MSH and MSHOP.

The influence of mobile nodes is shown using PRR when considering the mobile environment in various smart homes. When a node is mobile, it may or may not be reachable by the respective router, but it may be reachable by another router. As there are three TTPs operating as middlemen in MSHOP, each one can provide connectivity to its nearest node. The PRR values obtained in several smart homes where MRHOF is chosen for the shortest path selection in Cooja are shown in Fig. 21. It is obvious that the random topology, rather than others, determines the better data reception ratio. When it comes to transmission ranges 70 and 80, the turns are taking place. In terms of PRR, the modified one outperforms all other smart homes. According to the survey, MRHOF outperforms OF0 in terms of shortest path selection, packet reception ratio, and data loss. OF0 shows a better performance than the previous one when it comes to mobility. It has been demonstrated that MSH outperforms using MRHOF when modelling the influence of mobile nodes. Comparing it to MSHOP is the next stage. The simulation is now done in a modified smart home utilizing MRHOF and OF0, and it is compared to MSHOP, which uses OF0 to determine the shortest path to the router. In all the three

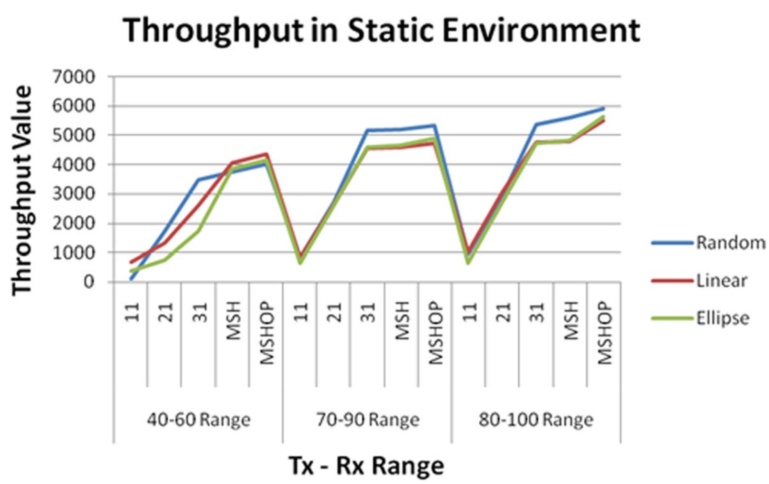

Fig. 23 Throughput in static environment

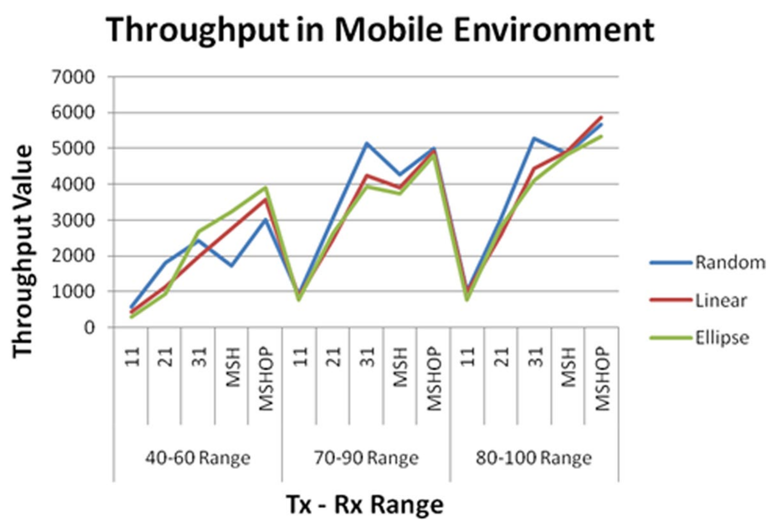

Fig. 24 Throughput in mobile environment

situations presented in Fig. 22, it is evident that the PRR rate utilizing OF0 is higher than that using MRHOF. MSHOP eventually outperforms the formers with a PRR value of 99.93% in mobile environment. When it comes to arrangement of the devices, linear topology works the best. The outcome supports the suggested method's effectiveness because MSHOP outperforms it even when devices are in a mobile state. Out of the 30 min of simulation time, 6 devices in MSHOP, including a vacuum sweeper, a human wearing a smart watch, a human carrying a smart phone, door and window movement sensors, etc., are moveable for a significant amount of time in the network. It nevertheless provides a packet reception ratio of over 99% despite its mobility.

Another experiment is run while taking into account a smart home scenario where the mobile nodes are moving continuously. All six of these nodes move continuously for 30 min during the experiment, with a pause for 30 s between cycles. The indefinite movement scenario was also studied using the aforementioned settings considering the network's linear architecture with an 80–100 Tx–INT range. The results thus obtained

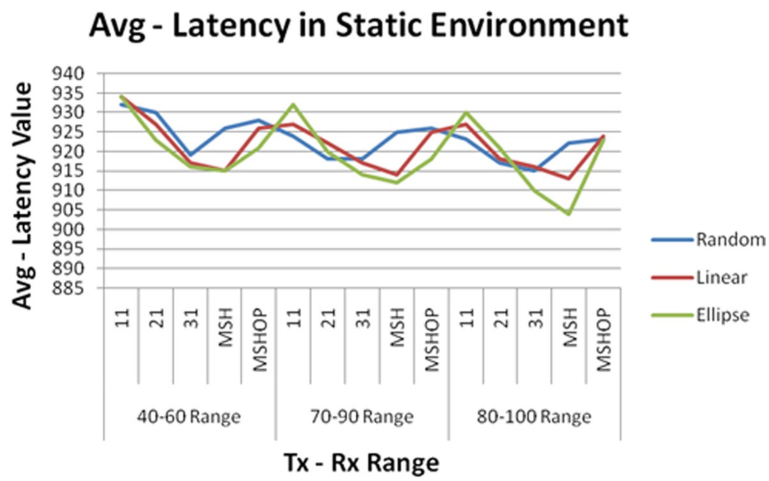

Fig. 25 Average latency in static environment

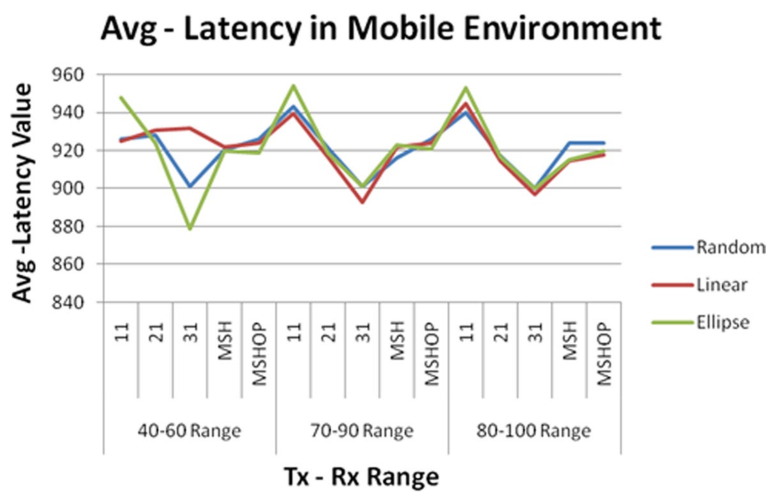

Fig. 26 Average latency in mobile environment

prove that MRHOF as the objective function achieves the highest percentage of PRR. A python script has been used to generate the positions of all 6 nodes. The simulation is carried out using the mobile positions, and the outcome is shown with the aid of the python script as depicted in Fig. 13. Despite being continuously mobile, 85.23% of packets are found to be successfully received on an average.

As mentioned in Eq. 3, the rate at which a message is successfully delivered via a communication channel is throughput. It is measured in both the static and mobile environment and presented in Figs. 23 and 24. Throughput increases with the number of nodes and transmission range in both scenarios. It is obvious from the observations that MSHOP provides greater throughput numbers in the ranges. As shown in Figs. 23 and 24, with transmission range 80, a maximum throughput of 5911 bits/s and 5880 bits/s has been achieved in random and linear topologies, respectively.

Network delay is another name for average latency that is presented in Eq. 4. Even if the number of devices participating in the communication is larger and more

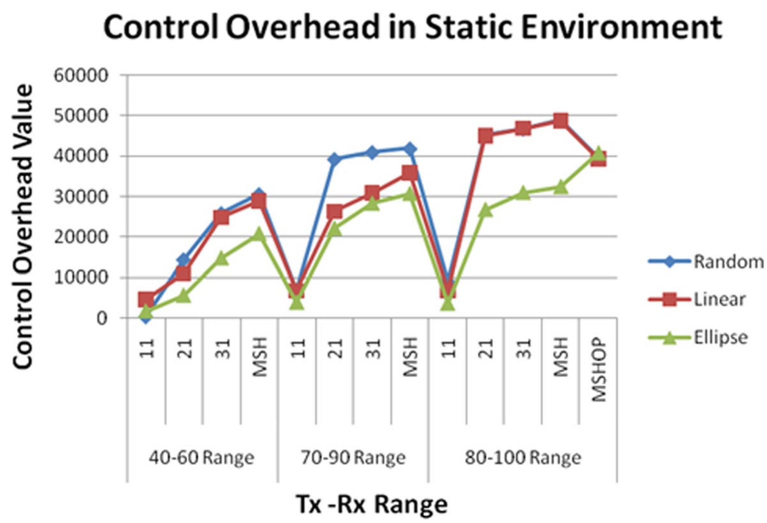

Fig. 27 Network overhead static environment

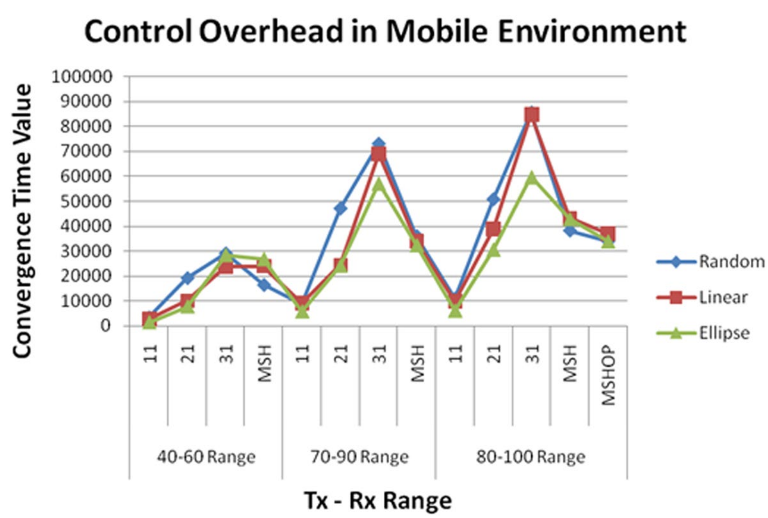

Fig. 28 Network overhead mobile environment

mobile in nature, the goal is to attain a lower average latency value. In both static and mobile environments, the MSHOP has the lowest latency value compared to the MSH. The reason for this is that when MSHOP is used, the messages are routed to the edge router from their appropriate TTP. A minimum of 923 ms or 0.9 s of delay is found for MSHOP with any transmission range and in all the topologies, as shown in Fig. 25. When mobility is taken into account, a minimum average latency of 918 ms is attained as can be seen in Fig. 26.

RPL uses a collection of ICMPv6 control messages, including DIO, DIS, and DAO to spread node information across the network topology [44]. If a DODAG is to be built, the root node sends out a DIO message to all client nodes in the range, and the client nodes react with a DAO message to the root, indicating that they want to join the DODAG [20]. Out-of-range nodes will send DIS messages to neighbouring nodes

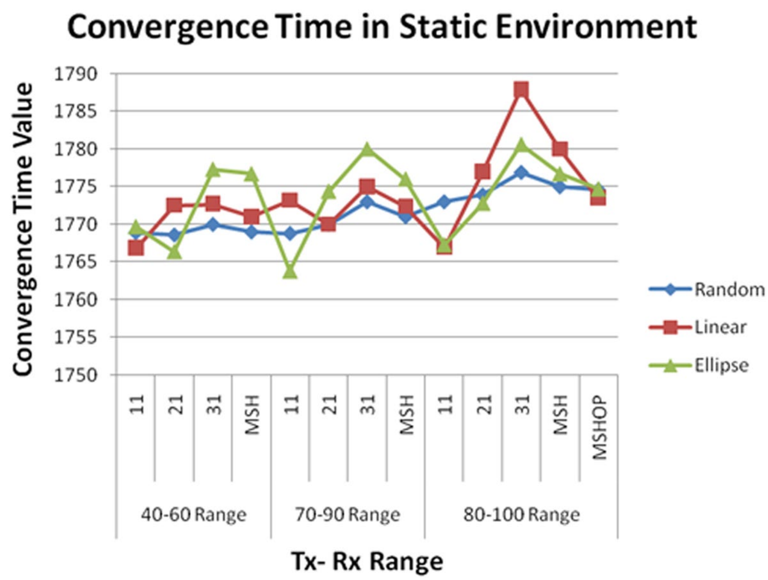

Fig. 29 Convergence time in static environment

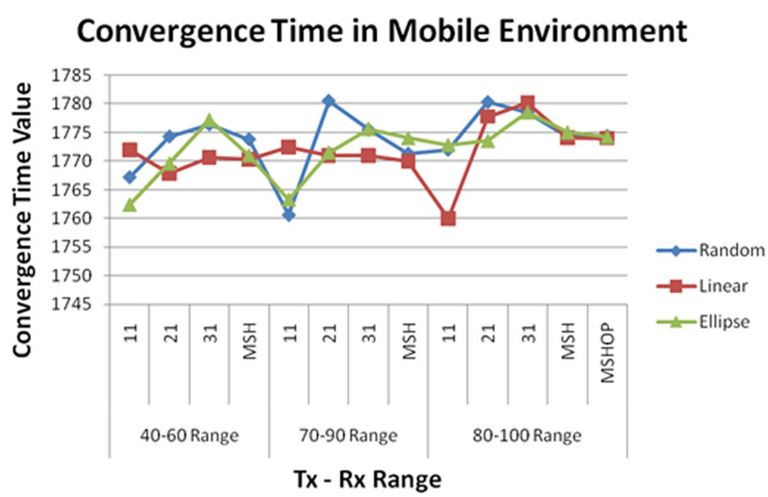

Fig. 30 Convergence time in mobile environment

to request DIO. When a neighbour node receives a DIS message, it sends a DIO message to that node (which is out of range) and that node responds with a DAO message, which is transmitted to the root node [18]. The total number of DIS, DAO, and DIO for each node is used to calculate Network Overhead, also known as Control Overhead as given in Eq. 5. It summarizes the network's traffic load. The cost of overhead rises in direct proportion to the number of nodes. Higher traffic means more collisions and data packet loss. The goal of using MSHOP is to reduce overhead in both the static and mobile context.

The overhead of different smart homes in static environment is explained in Fig. 27. It can be shown that when the network density grows, the network overhead value grows as well. Even in the modified smart house, where TTP serves as a

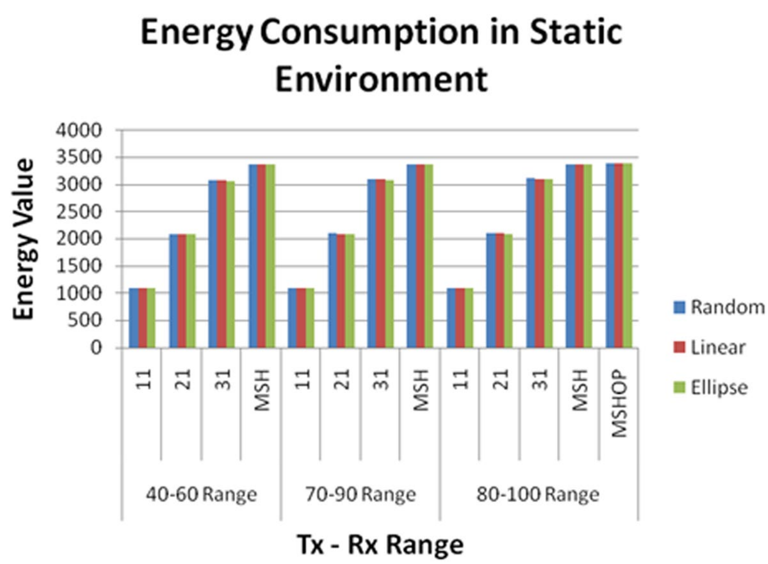

Fig. 31 Total energy in static environment

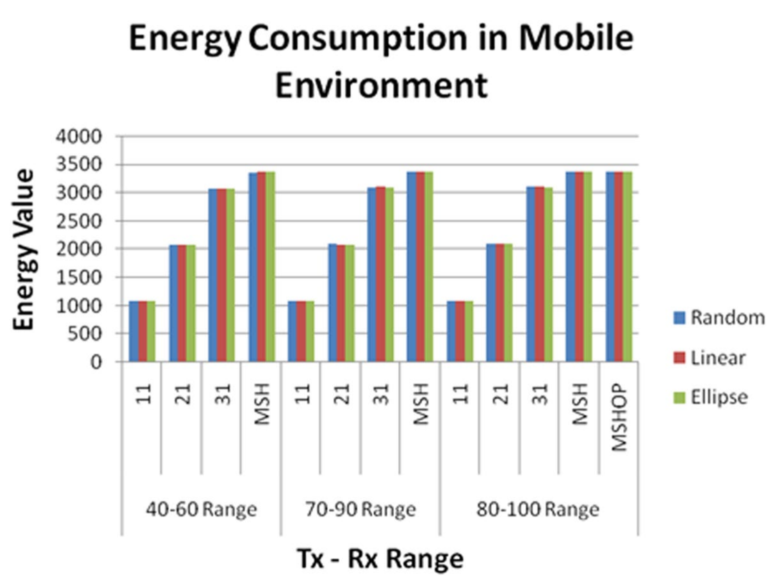

Fig. 32 Total energy in mobile environment

communication intermediary, the control overhead is higher than before. This is due to the fact that MSH has more nodes than an advanced smart home. Finally, MSHOP is tested for the parameter, and it outperforms all of the others. Even if the range is 80–100, when data transmission and delivery rates are higher, MSHOP still exhibits a lower overhead. When working in a mobile environment, network overhead may increase. It will, of course, be proportional to the number of nodes. The overhead of a network is affected by mobile nodes, as shown in Fig. 28. It is obvious that, despite having a higher number of nodes than earlier smart homes, both MSH and MSHOP are successful in lowering network overhead.

The network shows that the convergence time increases as the network density increases. Convergence time can be calculated referring Eq. 6. The more the traffic, the longer it takes to reach a point of convergence. There is no specific range or topology that would result in a faster convergence time. The DIO time is always a factor. Despite the fact that density is directly proportional to convergence time, MSH has a faster one. MSHOP outperforms MSH in terms of convergence time, yielding a lower value. Both MSH and MSHOP have more client nodes doing the operation; however, the convergence time illustrated in Figs. 29 and 30 is comparatively less than other smart homes.

The quantity of energy dissipated by network nodes per unit time is referred to as energy consumption [45]. This research uses the Z1 mote platform in each sensor node to conduct a complete analysis of energy use. Contiki-NG has an energest module that can be used to develop a lightweight, software-based energy estimating strategy for IoT devices with limited resources [53, 54]. The Z1's current consumption is considered in four stages, i.e. CPU, LPM, Tx, and Rx. For each stage, their respective current consumption and Z1's current and voltage consumption values are listed in Table 2. The power consumption of each stage is done with the help of Eq. 7 through Eq. 10. The total energy consumption in a network is estimated using Eq. 11 and compared with other networks.

Figures 31 and 32 show the network's entire energy consumption, including CPU, LPM, Tx, and Rx states. It has been discovered that as a network delivers a greater number of packets, it consumes more energy. Of the four states, transmission and reception consume more energy. The data transmission and energy consumption will both increase as the network density increases. Energy consumption has increased in both static and mobile environments as transmission range, network density, and PRR have all increased. As a result, the MSH consumes more energy than the smart homes with 11, 21, and 31 nodes. The MSHOP is then compared with MSH. It has successfully decreased energy usage in mobility environments compared to static environments. In a static setting, the MRHOF is utilized to calculate ETX value to find the shortest section, which requires more time, memory, and energy. In a mobile context, the OF0 outperformed MRHOF in terms of PRR, minimizing latency, and network overhead. As a result, OF0 is used to determine the shortest path in a mobile environment, and it uses less energy than the former.

Table 3 depicts the result found in this work in comparison with the existing ones. According to simulation study, the default objective functions, OF0 and MRHOF, can easily manage traffic variability, but the suggested technique will exhibit a substantial shift for the better. The proposed technique aims to provide a uniform allocation of

Table 3 Comparison of performance metrics

Paper	Environment	Nodes	PRR (%)	Throughput (bits/s)	Avg. latency (ms)	Network overhead	Convergence time (ms)	Total energy consumption (mJ)
Musaddiq et al. [5]	Static	20	91.8	–	40.25	17,007	–	3,744,494
Solapure et al. [18]	Static	30	99	–	1796.5	16%	3000	1400
Urama et al. [35]	mobile	12	100	–	125	570	–	–
Proposed	Static	30	90.16	5911	923	39366	1773.5	3373.7
	Mobile	30	99.93	5880	918	33,922	1744	3365.2

traffic while minimizing a significant number of control overheads, resulting in higher PRR, lower E2E packet latency, and higher throughput. The goal is achieved in the same way utilizing the MSHOP model with the proposed algorithm. The finding explains why the Tx/INT range of 80–100 works better than 40 and 60. In both static and mobile environments, the linear architecture of PRR produces better results. Similarly, in terms of throughput, it has performed better in linear topology. When it comes to latency, it might be either random or linear. The network overhead grows as the number of devices increases, resulting in a higher traffic load. Despite the fact that MSHOP has the most devices, it has managed to keep the network overhead value low. Energy usage rises as density and PRR rise; however, MSHOP in a mobile environment succeeded to lower it compared to a static setting.

Sensor nodes are limited in terms of battery life, memory capacity, and durability in any IoT system. The smart home application, where the smart devices need to be charged frequently, has been chosen for the current work environment. Communication takes place in a mobile and static setting, which makes it dynamic. Undoubtedly, situations like this use up more energy. The proposed MSHOP focuses on reducing energy usage while optimizing PRR, PLR, latency, and throughput. Even while energy consumption is constant across the vast intended network of smart homes, it is not reduced. This is thought to be the area of this work that needs improvement; thus, attention will be given to it in the future.

6 Conclusion

In this article, an optimization technique is developed with a smart home scenario called modified smart home optimized path. MSHOP has TTP working as a middleman framework and an event timer for efficient delivery of packets. This design has been put to the test in a variety of scenarios and compared to a number of LNN classified as basic, average, and advanced smart homes. Static and mobile environment is also considered for the performance evaluation of the proposed topology. The MSHOP routing protocol RPL is evaluated under heavy load by focusing on packet reception ratio and network efficiency. With the help of TTP and event timer, the PRR is almost 100%, network overhead is reduced, average latency of packets has a significant reduction, and throughput has an improvement. The above scenario is tested through simulation in Cooja. The desire to use IoT solutions in the future is the motivation behind the work. The IoT network is enormous, and there are also increasing dangers of loss. The suggested strategy can be utilized to minimize data loss, time consumption, and energy usage.

As a future scope, the experimental implementation using raspberry-pi, arduino, and different sensor nodes is considered. In addition, a mechanism for slotting could also be thought off to improve the message transmission rate. An intelligent decision could also be taken to transmit the important messages on time to avoid missing predominant messages.

Abbreviations

IoT	Internet of Things
RPL	Routing protocol over 6LoWPAN
MSHOP	Modified smart home-optimized path
WSN	Wireless sensor networks
ICT	Information and communications technology
OS	Operating system
6LoWPAN	Low-power wireless personal area networks using IPv6
TTP	Trusted third party
QWL-RPL	Protocol based on queue and workload conditions
PRR	Packet reception ratio
OF	Objective function
MRHOF	Minimum rank with hysteresis objective function
OF0	Objective function zero
PDR	Packet delivery ratio
QU-RPL	Queue utilization-based RPL
NCRM	Neo-hybrid composite routing metric
ERAOF	Energy-efficient and path reliability-aware objective function
F and F	Filter and fuzz
CABAN	Cloud-assisted body area network
DAG	Directed acyclic graphs
DODAG	Destination-oriented acyclic graph
ETX	Expected number of transmission
IETF	Internet engineering task force
DIO	DODAG information object
DIS	DODAG information solicitation
DAO	DODAG destination advertisement Object
MSH	Modified smart home
Rx	Reception current
Tx	Transmission current
Z1	Zolertia 1

Author contributions

NP and MS conceived the main idea and proposed the architecture and algorithm for smart home network. NP performed the implementation of the proposed methodology in the Contiki Cooja simulator. NP contributed to the structuring of the manuscript. MS reviewed and finalized the content of the manuscript. MS suggested the final changes to be made and gave the final approval of the manuscript. Both authors read and approved the final manuscript.

Authors information

Niharika Panda is a Ph.D. candidate in the Department of Computer Science and Engineering, Amrita School of Engineering, Bengaluru, Karnataka, India. In 2013, she earned a B.Tech in Information Technology from Koustuv Institute of Self Domain, BPUT, and in 2015, she earned an M.Tech in Computer Science from Berhampur University, Odisha, India. She has two years of experience as a teacher. Internet of Things and wireless sensor networks are two of her main research interests.

M. Supriya currently serves as associate professor at the Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, Bengaluru, India. She received her Ph.D. (CSE) from Amrita Vishwa Vidyapeetham in the year 2017. She has 20+ years of teaching experience among which 10 years contribute to research. She has more than 30 Scopus indexed publications in peer-reviewed journals and conferences. She is an active ACM professional member and leads various ACM club activities at the college. Her research interests include parallel/distributed computing, cloud computing, and fog computing.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 6 June 2022 Accepted: 30 November 2022
Published online: 12 December 2022

References

1. R. Ranjan, O. Rana, S. Nepal, M. Yousif, P. James, Z. Wen, S. Barr, P. Watson, P.P. Jayaraman, D. Georgakopoulos, M. Villari, The next grand challenges: integrating the Internet of Things and data science. IEEE Cloud Comput. **5**(3), 12–26 (2018)
2. T. Qiu, N. Chen, K. Li, M. Atiquzzaman, W. Zhao, How can heterogeneous internet of things build our future: a survey. IEEE Commun. Surv. Tutor. **20**(3), 2011–2027 (2018)
3. M. Chernyshev, Z. Baig, O. Bello, S. Zeadally, Internet of Things (IoT): research, simulators, and testbeds. IEEE Internet Things J. **5**(3), 1637–1647 (2017)

4. H. Yoo, M. Shim, D. Kim, K.H. Kim, GLOBAL: a gradient-based routing protocol for load-balancing in large-scale wireless sensor networks with multiple sinks, in *The IEEE symposium on Computers and Communications* (IEEE, 2010), pp. 556–562
5. A. Musaddiq, Y.B. Zikria, S.W. Kim, Routing protocol for low-power and lossy networks for heterogeneous traffic network. EURASIP J. Wirel. Commun. Netw. **2020**(1), 1–23 (2020)
6. F. Al-Turjman, L. Mostarda, E. Ever, A. Darwish, N.S. Khalil, Network experience scheduling and routing approach for big data transmission in the Internet of Things. IEEE Access **7**, 14501–14512 (2019)
7. B. Agarwal, A. Chakravorty, T. Wiktorski, C. Rong, Enrichment of machine learning based activity classification in smart homes using ensemble learning, in *2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC)* (IEEE, 2016), pp. 196–201
8. K. Kotobi, M. Sartipi, Efficient and secure communications in smart cities using edge, caching, and blockchain, in *2018 IEEE International Smart Cities Conference (ISC2)* (IEEE, 2018), pp. 1–6
9. A. Durand, Y. Ngoko, C. Cerin, Distributed and in-situ machine learning for smart-homes and buildings: application to alarm sounds detection, in *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)* (IEEE, 2017), pp. 429–432
10. Y. Chen, J.P. Chanet, K.M. Hou, H. Shi, G. De Sousa, A scalable context-aware objective function (SCAOF) of routing protocol for agricultural low-power and lossy networks (RPAL). Sensors **15**(8), 19507–19540 (2015)
11. H. Lamaazi, A. El Ahmadi, N. Benamar, A.J. Jara, OF-ECF: a new optimization of the objective function for parent selection in RPL, in *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (IEEE, 2019), pp. 27–32
12. N. Pradeska, W. Najib, S.S. Kusumawardani, Performance analysis of objective function MRHOF and OF0 in routing protocol RPL IPV6 over low power wireless personal area networks (6LoWPAN), in *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)* (IEEE, 2016), pp. 1–6
13. S. Kuwelkar, H.G. Virani, Performance analysis of the objective functions in IPv6 routing protocol for LLN's used in IOT applications, in *2019 IEEE Pune Section International Conference (PuneCon)* (IEEE, 2019), pp. 1–5
14. W.A. Hussein, B.M. Ali, M.F.A. Rasid, F. Hashim, Design and performance analysis of high reliability-optimal routing protocol for mobile wireless multimedia sensor networks, in *2017 IEEE 13th Malaysia International Conference on Communications (MICC)* (IEEE, 2017), pp. 136–140
15. A.Y. Barnawi, G.A. Mohsen, E.Q. Shahra, Performance analysis of RPL protocol for data gathering applications in wireless sensor networks. Procedia Comput. Sci. **151**, 185–193 (2019)
16. Z. Royaee, H. Mirvaziri, A. Khatibi Bardsiri, Designing a context-aware model for RPL load balancing of low power and lossy networks in the internet of things. J. Ambient Intell. Humaniz. Comput. **12**(2), 2449–2468 (2021)
17. S. Madakam, V. Lake, V. Lake, V. Lake, Internet of Things (IoT): a literature review. J. Comput. Commun. **3**(05), 164 (2015)
18. S.S. Solapure, H.H. Kenchannavar, Design and analysis of RPL objective functions using variant routing metrics for IoT applications. Wirel. Netw. **26**(6), 4637–4656 (2020)
19. J.V. Sobral, J.J. Rodrigues, R.A. Rabêlo, J. Al-Muhtadi, V. Korotaev, Routing protocols for low power and lossy networks in internet of things applications. Sensors **19**(9), 2144 (2019)
20. H. Lamaazi, N. Benamar, A comprehensive survey on enhancements and limitations of the RPL protocol: a focus on the objective function. Ad Hoc Netw. **96**, 102001 (2020)
21. L.P. Qian, Y. Wu, B. Ji, L. Huang, D.H. Tsang, HybridIoT: integration of hierarchical multiple access and computation offloading for IoT-based smart cities. IEEE Netw. **33**(2), 6–13 (2019)
22. P.O. Kamgueu, E. Nataf, T.D. Ndié, Q. Festor, Energy-based routing metric for RPL, Doctoral dissertation, INRIA (2013)
23. M. Joshi, B. Kaur, Web integrated smart home infrastructure using internet of things. Int. J. Eng. Res. Gen. Sci **3**(6), 153–158 (2015)
24. M. Qasem, H. Altawssi, M.B. Yassien, A. Al-Dubai, Performance evaluation of RPL objective functions, in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing* (IEEE, 2015), pp. 1606–1613
25. A. Touzene, A. Al Kalbani, K. Day, N. Al Zidi, Performance analysis of a new energy-aware RPL routing objective function for internet of things, in *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)* (IEEE, 2020), pp. 1–6
26. B. Kang, S. Kim, M.I. Choi, K. Cho, S. Jang, S. Park, Analysis of types and importance of sensors in smart home services, in *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)* (IEEE, 2016), pp. 1388–1389
27. A.M. Zungeru, J. Gaboitaolelwe, B. Diarra, J.M. Chuma, L.M. Ang, L. Kolobe, M. David, I. Zibani, A secured smart home switching system based on wireless communications and self-energy harvesting. IEEE Access **7**, 25063–25085 (2019)
28. Q. Lyu, N. Zheng, H. Liu, C. Gao, S. Chen, J. Liu, Remotely access"my" smart home in private: An anti-tracking authentication and key agreement scheme. IEEE Access **7**, 41835–41851 (2019)
29. J.A. Hatem, H. Safa, W. El-Hajj, Enhancing routing protocol for low power and lossy networks, in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)* (IEEE, 2017), pp. 753–758
30. A. Brachman, RPL objective function impact on LLNs topology and performance, in *Internet of Things, Smart Spaces, and Next Generation Networking* (Springer, Berlin, Heidelberg, 2013), pp. 340–351
31. H.S. Kim, H. Kim, J. Paek, S. Bahk, Load balancing under heavy traffic in RPL routing protocol for low power and lossy networks. IEEE Trans. Mob. Comput. **16**(4), 964–979 (2016)
32. P. Sanmartin, A. Rojas, L. Fernandez, K. Avila, D. Jabba, S. Valle, Sigma routing metric for RPL protocol. Sensors **18**(4), 1277 (2018)
33. A.J. Charles, K. Palanisamy, Neo-hybrid composite routing metric for RPL. Procedia Comput. Sci. **171**, 1819–1828 (2020)

34. N. Sousa, J.V. Sobral, J.J. Rodrigues, R.A. Rabêlo, P. Solic, ERAOF: a new RPL protocol objective function for Internet of Things applications, in *2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)* (IEEE, 2017), pp. 1–5
35. I.H. Urama, H. Fotouhi, M.M. Abdellatif, Optimizing RPL objective function for mobile low-power wireless networks, in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 2 (IEEE, 2017), pp. 678–683
36. S. Murali, A. Jamalipour, Mobility-aware energy-efficient parent selection algorithm for low power and lossy networks. IEEE Internet Things J. **6**(2), 2593–2601 (2018)
37. W. Lu, P. Si, G. Huang, H. Han, L. Qian, N. Zhao, Y. Gong, SWIPT cooperative spectrum sharing for 6G-enabled cognitive IoT network. IEEE Internet Things J. **8**(20), 15070–80 (2020)
38. W. Lu, Y. Ding, Y. Gao, Y. Chen, N. Zhao, Z. Ding, A. Nallanathan, Secure NOMA-based UAV-MEC network towards a flying eavesdropper. IEEE Trans. Commun. **70**, 3364–3376 (2022)
39. W. Lu, Y. Mo, Y. Feng, Y. Gao, N. Zhao, Y. Wu, A. Nallanathan, Secure transmission for multi-UAV-assisted mobile edge computing based on reinforcement learning. IEEE Trans. Netw. Sci. Eng. (2022). https://doi.org/10.1109/TNSE.2022.3185130
40. G. Li, H. Wu, G. Jiang, S. Xu, H. Liu, Dynamic gesture recognition in the internet of things. IEEE Access **18**(7), 23713–24 (2018)
41. L. Huang, Q. Fu, G. Li, B. Luo, D. Chen, H. Yu, Improvement of maximum variance weight partitioning particle filter in urban computing and intelligence. IEEE Access **31**(7), 106527–35 (2019)
42. Z. Shelby, C. Bormann, *6LoWPAN: The Wireless Embedded Internet* (Wiley, Hoboken, 2011)
43. I.H. Urama, Optimizing routing protocol for low power and lossy network (RPL) objective function for mobile low-power wireless networks (2016)
44. T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.P. Vasseur, R. Alexander, RPL: IPv6 routing protocol for low-power and lossy networks (No. rfc6550) (2012)
45. A. Sanila, B. Mahapatra, A.K. Turuk, Performance evaluation of RPL protocol in a 6LoWPAN based smart home environment, in *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)* (IEEE, 2020), pp. 1–6
46. P. Thubert, *Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL), RFC 6552* (IETF, 2012)
47. D. Schweizer, M. Zehnder, H. Wache, H.F. Witschel, D. Zanatta, M. Rodriguez, Using consumer behavior data to reduce energy consumption in smart homes: applying machine learning to save energy without lowering comfort of inhabitants, in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)* (IEEE, 2015), pp. 1123–1129
48. O. Gnawali, P. Levis, The minimum rank with hysteresis objective function, in *RFC 6719* (2012), p. 13
49. I. Romdhani, M. Qasem, A.Y. Al-Dubai, B. Ghaleb, *Cooja Simulator Manual* (Napier University, Edinburgh, 2016)
50. S. Duquennoy et al., Contiki-ng: The OS for next generation IoT devices (2019)
51. M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I.A.T. Hashem, A. Siddiqa, I. Yaqoob, Big IoT data analytics: architecture, opportunities, and open research challenges. IEEE Access **5**, 5247–5261 (2017)
52. I.N.R. Hendrawan, I.G.N.W. Arsa, Zolertia Z1 energy usage simulation with Cooja simulator, in *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)* (IEEE, 2017), pp. 147–152
53. R. Zolertia, Mote hardware development platform (2015)
54. A. Elsts, Instrumenting contiki ng applications with energy usage estimation (2020). https://github.com/contiki-ng/contiking/wiki/Instrumenting-Contiki-NG-applications-with-energyusage-estimation

Publisher's Note

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