

An Efficient and Reliable Multicasting for Smart Cities

Faheem Khan¹, Muhammad Zahid², Hüseyin Gürüler³, İlhan Tarimer³ and Taegkeun Whangbo^{1,*}

¹Department of Computer Engineering,, Artificial intelligence Lab, Gachon University, Seongnam, 13120, Korea

²Department of Computer Science, Bacha Khan University Charsadda, Charsadda, 24450, Pakistan

³Department of Information Systems Engineering, Mugla Sitki Kocman University, Mugla, 48000, Turkey

*Corresponding Author: Taegkeun Whangbo. Email: tkwhangbo@gachon.ac.kr

Received: 23 August 2021; Accepted: 06 October 2021

Abstract: The Internet of thing (IoT) is a growing concept for smart cities, and it is compulsory to communicate data between different networks and devices. In the IoT, communication should be rapid with less delay and overhead. For this purpose, flooding is used for reliable data communication in a smart cities concept but at the cost of higher overhead, energy consumption and packet drop etc. This paper aims to increase the efficiency in term of overhead and reliability in term of delay by using multicasting and unicasting instead of flooding during packet forwarding in a smart city using the IoT concept. In this paper, multicasting and unicasting is used for IoT in smart cities within a receiver-initiated mesh-based topology to disseminate the data to the cluster head. Smart cities networks are divided into cluster head, and each cluster head or core node will be responsible for transferring data to the desired receiver. This protocol is a novel approach according to the best of our knowledge, and it proves to be very useful due to its efficiency and reliability in smart cities concept because IoT is a collection of devices and having a similar interest for transmission of data. The results are implemented in Network simulator 2 (NS-2). The result shows that the proposed protocol shows performance in overhead, throughput, packet drop, delay and energy consumption as compared to benchmark schemes.

Keywords: Multicasting; unicasting; flooding; network simulator-2; internet of thing; smart cities

1 Introduction

Advanced countries are facing many social issues related to safety, security, healthcare, disaster, aged population etc. The idea of smart cities utilizes the technologies of IoT to improve social infrastructure and to design an innovative solution to the issues mentioned above. Significant improvement in the wireless network, mobile ad-hoc network, 5G, and telecommunication through fiber optic encourages the researcher to use pervasive intelligence. IoT is used in almost every field of life especially



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

in e-health, smart learning, smart homes, and professional level, i.e., smart transportation, remote monitoring, automation, smart cities etc. [1–3].

An increasing number of objects are now connecting to the internet through IoT. Some general examples that enable smart homes are monitoring system, security system, control system, thermostats, heating, air-conditioning, ventilation and much more. IoT [4] is also playing an important role to improve the quality of lives, including industrial automation, healthcare, emergency response to artificial and natural disasters, transportation etc. IoT devices connect and share their coordinates, information, and decisions through embedded devices, ubiquitous and pervasive computing, internet protocols, and applications. The objects/devices perform their task either in a domain-specific application or an independent domain application, whereas in each domain, actuators and sensors communicate with each other through some predefined protocols, as shown in Fig. 1 [5].



Figure 1: Domain specific and independent application in IoT

Flooding [6] is the most appropriate method of transferring sensing data within the local network because these devices/objects work at a distance of 100-meter minimum and 400-meter maximum. A source node transmits data through every outgoing object/device in flooding because it uses all the available paths in the network. This mechanism is very favorable in applications like IoT, but it suffers duplicate packet transmission and collision in the wireless medium, which leads to network congestion. Therefore, this paper proposes effective multicasting and unicasting method to transfer local data in community-based IoT networks.

Multicasting and unicasting are performed through tree and mesh-based routing protocols [5–9]. In the IoT, mesh-based multicasting is used, where multiple paths exist between group members to enrich connectivity for better reliability and less delay. Furthermore, the receiver-initiated protocol is used for data dissemination, where the receiver will be responsible for data transfer within the group members [10]. After going through the literature, the researcher pointed out the following problems. In receiver-initiated mesh-based approaches like Protocol for unified multicast through announcement (PUMA) [11], Efficient and reliable core assisted multicast routing in mobile ad-hoc network (ERASCA) [12] and On demand multicast routing protocol [13], the information is flooded inside the mesh, and such information is quickly received by all mesh members but at the cost of extra overhead. In flooding, as the destination node is moving away due to mobility, then the destined receiver receives the data with less at the cost of maximum overhead and energy consumption in a wireless network [14–16]. All the protocols use the cluster-based approaches where a core node/leader node acts as a central point of an entity. The core node is considered a group leader of mesh. The mesh consists of receivers, intermediate nodes, mesh relay nodes, non-member nodes and terminals nodes, as shown in Fig. 2.

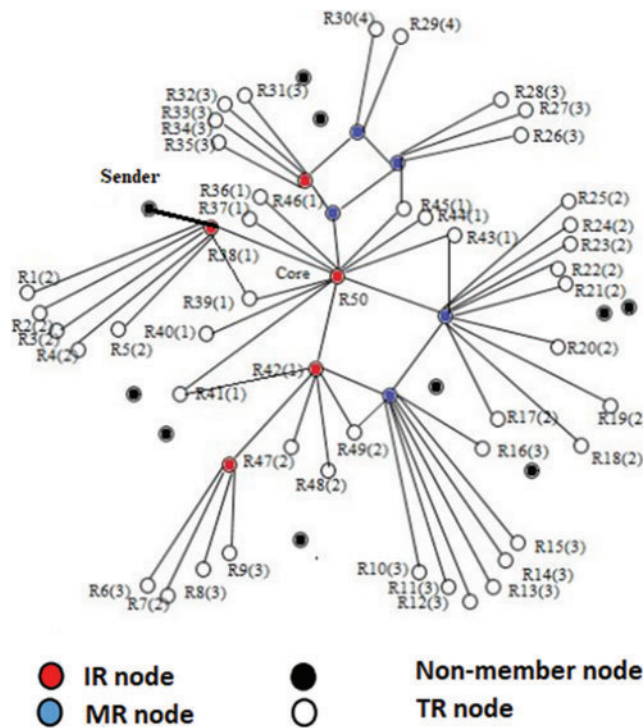


Figure 2: Mesh formation

In this paper, flooding is avoided for packet forwarding to the destination node. In the proposed protocol, any mesh member receives the packet forwarded to the core node despite the flooding. Now it is the responsibility of the core node to send it to the desired destined node through unicasting or multicasting. This transferring of data depends on the distance from the core to the destination node. If the destined receiver is within the 3-hop distance from the core node, it will be transferred through unicasting, but if the distance is more than 3-hops, then multicasting will be preferred. Due to the limited resources in MANET, 3-hop distance is selected because by increasing the hop distancing, a resource consumption of the overall system will also increase in term of overhead, delay, energy

consumption and packet drop. Hence, the objective is achieved with less overhead and delay. The evaluation results show that the proposed method improves the data delivery, reduces the delay, energy consumption and throughput in smart cities using IoT.

The remaining paper is organized in the following fashion. Section 2 explains the related work in detail with mesh and tree-based multicasting approaches. Then, mesh-based approaches are further refined into receiver-initiated and sender-initiated approaches. Finally, the existing problem of flooding is explained in the literature. Section 3 explains the multicasting and flooding in the term of the core election process. The core is elected on multiple factors to decrease the overhead. In this section, the methodology of the proposed protocol is explained with a packet forwarding approach and justifies the use of multicasting and unicasting instead of flooding. In section 4, the simulation is performed in NS-2 Simulator between Modified ERASCA, ERASCA and PUMA. Through simulation, it is concluded that the Modified ERASCA is superior to ERASCA and PUMA. The superiority is because of the use of multicasting and unicasting instead of flooding. Finally, Section 5 is the conclusion of the paper.

2 Related Work

A simple flooding mechanism is introduced in [6], where the node receives the data immediately within the group. This creates a flooding storm due to the increase of node density, increasing packet drop, packet collision, network trafficking etc. The Neighbor supporting multicast protocol (NSMP) [17,18] is a sender-initiated approach and is less vulnerable to link failure because of its path discovery through flooding of route discovery and local route discovery. The flood route discovery is used to establish the route, and the local route discovery is used for network partition in a small multicast group. In this approach, the path having already forwarding nodes is preferred compared to the path where the new forwarding nodes are first requested and then included in the path. This technique minimizes the trafficking, congestion, overhead, end to end delay and, as a result, increases the efficiency of the routes. However, refreshing is very necessary for NSMP for group neighbors and forwarding nodes to operate correctly.

Dynamic core-based multicast routing protocol (DCMP) [19] uses a cluster-based routing protocol. It decreases the flooding from the source node by selecting three types of senders, i.e., passive sender, active sender and core active sender. The active sender and core active sender are flooding the Join Request. The packet begins from passive sender and is communicated to the core active sender transmitted to mesh. The number of hops between passive sender and core active sender is limited and due to which the overhead is limited. The MaxPassSize and MaxHop are the two factors in DCMP, and it allows the mesh to have a maximum forwarding node to assure robustness. Due to passive nodes, high throughput with less overhead provides more scalability to the network. The main drawback of DCMP is its dependency on the core node, which is a very alarming situation because, in a core failure situation, the total number of passive nodes also fails.

In receiver-initiated protocols, it is the responsibility of the receiver to maintain and update the mesh. However, when the core node is selected as a leader, the receiver starts forming the mesh, and the core node maintains and updates the mesh. In PUMA [19], sources are not required to join the multicast group before any communication to the group recipient, and less delay and overhead is utilized [20]. In this approach, the core is selected within the group receivers. The initiator sends the data packet to the receivers through the shortest possible path. Flooding is used within the multicast group as soon as the data reaches any mesh member. A Multicast announcement (MA) is flooded within the mesh for maintenance and connectivity of the mesh. The MA consists of distance to the

core, group ID, core ID, parent node, sequence number and M_flag. Because of such information, the initiator outside the receiver group can send the data to the desired receiver. With the help of such information, the group is informed about the entry and departure of a node. There are many drawbacks to PUMA, like data duplication and core failure. In PUMA, flooding is used for data forwarding inside the mesh instead of multicasting.

ERASCA is a receiver-initiated protocol, which selects a stable core on certain parameters like location and battery capacity. The EERASCA overcomes the disadvantage of PUMA, where frequent core failure produces extra overhead and an extra election process. In this protocol, the concept of mirror core is also introduced to decrease the core failure and core re-election process. Hence, the ERASCA protocol is much more efficient than the approaches mentioned above in term of overhead because of the stable core election and more reliable than the above-mentioned approaches in term of delay because of the introduction of the mirror core.

On the other hand, flooding is used in data forwarding to the desired receiver within the mesh. This compromises the efficiency in terms of overhead. ERASCA further increases the efficiency through a selection of the proper core node. However, flooding is used mainly in PUMA and ERASCA for communication, but this increases the overhead and decreases the group's efficiency.

All the approaches mentioned above use the core node/leader node/cluster head concept to maintain and update the group. However, in the proposed approach, the core node will be responsible for the maintenance and update and data forwarding to the desired destination, and this is the novelty of the proposed protocol. To increase the efficiency of a protocol, a proposed protocol avoids flooding and uses multicasting and unicasting for data forwarding inside the mesh, which is the main theme of this paper. Hence, in the above-mentioned approaches the proposed protocol shows improvement in overhead, packet delay, packet drop, throughput and energy.

3 Proposed Methodology

3.1 Overview

In this paper, a modified version of ERASCA is proposed for IoT with a core selection method. The proposed method is explained in [Tab. 1](#) with their communication, forwarding scheme and mesh formation. In this section, an efficient and reliable protocol is proposed using multicasting and unicasting instead of flooding within the receiver group. This receiver group is a combination of objects and devices which forms the IoT network for smart cities.

Table 1: Comparison of ERASCA and proposed protocol

Hierarchy							
Protocols	Broad-casting message	Connect-ivity List	Election	Responsibility of the core node	Group formation	Topology	Communi-cation process
Modified ERASCA	SD	Parent-child	Core election	Maintenance, update and data forwarding	Receiver initiated	Mesh	Multicast-ing and unicasting
ERASCA protocol	SD	Parent-child	Core election	Maintenance and update	Receiver initiated	Mesh	Flooding

Modified ERASCA protocol implemented the Internet protocol (IP) service model, where a node makes a multicast group through status declaration (SD) message. The Status Declaration periodically flooded inside a group at a regular interval through the core node and formed the network's connectivity list. The connectivity list forms the best possible route in a network as well as in the mesh. After reaching the data inside the mesh, it transfers the packet of the desired receiver to the core node. Now the core node decides through SD message whether to forward the packet through unicasting and multicasting.

Like PUMA and ERASCA, a first receiver that starts communication in Modified ERASCA announces itself as a first receiver of the receiver group. A receiver group is a member of the receiver that having a similar interest in data. All the receivers are connected with the core node through intermediate nodes, and such nodes are also called a Mesh relay node (MR), and all the receivers containing MR collectively make a mesh. The core node is selected on some pre-defined parameters, and the core node must continuously update the group member.

3.2 Core Selection in Modified ERASCA

Flooding is an essential factor that declined the performance of data communication in the IoT network. Core election/selection further deteriorates the performance if it is not managed correctly. The core node is a central point of contact that maintains and updates all the group members and is also responsible for data transmission [11]. The core node will change if the threshold value drops from specific parameters like battery capacity and many connected neighbors, which are necessary for selecting the core node. If the core node fails due to any reason, then another node will be elected as a core node, and the group will continuously receive the information of all the group members. In a wired network, when the core node fails, then the whole network collapses and stops all the communication, but here the situation is different because when the core node fails, another node will take the responsibility of core through some defined procedure. Therefore, a stable core election/selection is required for efficient performance.

Previously, the core is selected through random selection [21,22] and the first come first serve method [22]. The core is randomly selected through a randomly generated number in the former approach without concerning the position and battery capacity. As a result, the core in a bad position and have less battery capacity fails soon, and again the re-election procedure starts, which decreases the lifetime of the network. In the latter approach, when the core node fails, then the first most receiver in a group that starts the transmission is selected as a core node without knowing its position and battery capacity, and hence this core node also dries faster re-election procedure is started again. This frequent re-election consumes network resources quickly and decreases the network's lifetime with the decrease of efficiency. In the above-mentioned approaches, the core is not selected within the proper position and having a resourceful battery capacity and hence may not be considered an efficient approach because of the frequent core failure and frequent re-election for the new core node.

In Modified ERASCA, the core is elected based on battery energy and location in the group. The node in the center of the group with high battery capacity is chosen as a leader or core for the group. The stable core will have a high lifetime, and hence the lifetime of the group increases. Due to stable core, re-election and reconfiguration also decrease, which ultimately increases the throughput, decreases the delay and energy consumption, and increases the efficiency of smart cities in IoT.

3.3 Status Declaration Message

The core node uses a packet format of SD Message to maintain and update the mesh of a group. The core in the network periodically broadcasts these SD messages to form a connectivity list at each node. Connectivity lists enable the sources to send data towards the group. Each SD message specifies a core ID, group ID, parent node, battery capacity, sequence number, distance to the core and mesh member code.

The SD message starts the code from 0 for a non-member node. On the other hand, the mesh membership code starts from 1 to 3. 1 shows that the node is a receiver node but cannot represent a mesh member. 2 shows that a node is a mesh member but cannot be represented as a receiver and 3 shows that a node is a mesh member as well as a receiver. With such information in the SD message, the sender outside the mesh group can communicate the data within the group. It is the SD message which updates and maintains the mesh by notifying other members in the group. The parent node in the packet format shows the nearest neighbor from which it establishes the best SD message. The core node allots a unique number to SD message known as the sequence number. A fresher/higher sequence number is preferred over the lower/older sequence number. A node that receives the SD message with a fresher sequence number waits for a specified duration and then produces its SD message to form a connectivity list. Connectivity lists are formed through the periodic broadcast of the SD messages. With the help of connectivity lists, sources outside the group and receivers inside the group can estimate the best possible route to a group through the parent node and capable the mesh to send their data to the desired destination.

3.4 Formation of the Mesh

The formation of mesh is necessary after the formation of the receiver group. In IoT, nodes can move from one location to another location due to frequent topology changes. However, all the receiver nodes receive the data packet, although they are not within the radio range of each other. This transferring of data between the receiver group is possible due to the presence of mesh. The nodes in the mesh are characterized as Non-group member (NM) and group members. The former is not part of the mesh and represented in a black dot, as shown in Fig. 2. The latter is divided Mesh relay (MR), Intermediate receiver (IR) and Terminal receiver (TR). The MR is the intermediate node between the core and a receiver and represented as a blue color. The TR is known as a terminal receiver, and the mesh finish on it, TR does not contribute in the packet forwarding and is represented in white color. Likewise, the IR acts as an intermediate node and a receiver and are represented as a red dot. The intermediate node between R48 and the core node is the R42 node. At the start of the communication, only receivers are mesh members, but later on, the MR will also be a part of the mesh member because they occur between the TR and core node and communicate the data.

3.5 Data Forwarding in Mesh

The parent node performs an essential role in the Modified ERASCA and forwards data to its surrounding nodes within the mesh. The communication process in Modified ERASCA is form connectivity with a parent-child relationship. The connectivity list formation informs all the sources about the presence of the mesh group, and allow the sources to transmit the data to the desired receiver by the parent node, as shown in Fig. 3. Fig. 3 shows that the core node is the parent node of R38, and R38 is the parent node of R1, R2, R3, R4, R5, R39 and D. Hence the connectivity list continues in a Parent-child fashion.

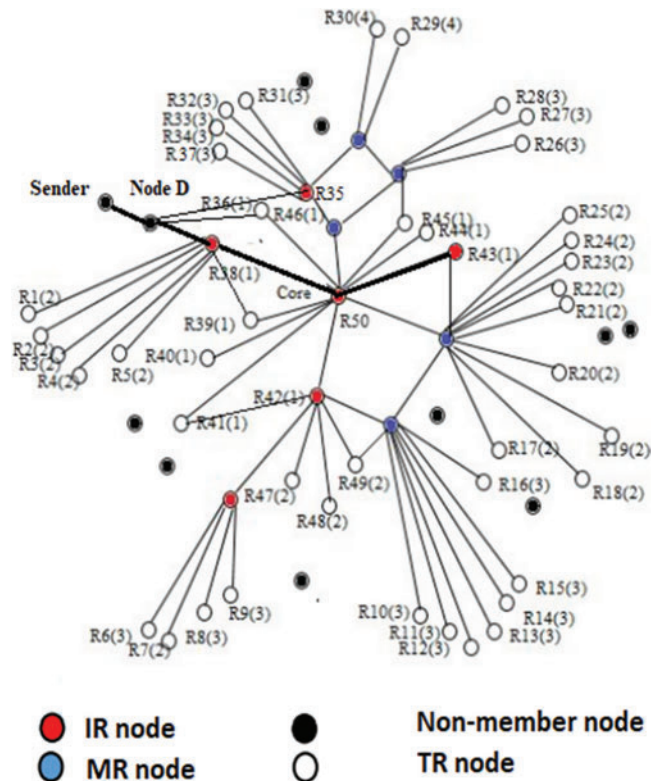


Figure 3: Modified ERASCA

For example, in previous protocols, the data transfer is from sender to node D, and now it is the responsibility of node D to transfer the data from sender/source to any mesh member. As discussed previously, the flooding produces many duplicate packet transmissions and leads towards packet drop, congestion and wastage of resources, and it is shown in Fig. 2. Let's suppose that a destined receiver is R43. The sender transmits a data packet and its sequence number s i.e., D_s to node D. Node D Broadcasts D_s to nodes R35, R36, core node, and R38. R35 broadcast D_s to neighbor R46; R36 broadcast D_s to core node; R38 broadcast D_s to R1, R2, R3, R4, R39 and core node and the core node broadcast D_s to R36, R38, R39, R40, R41, R42, R43, R44, R46 and R50, and finally the data is received to R43. This amount of needless transmission consumes the resources of wireless medium very quickly.

Now in Modified ERASCA, two heuristics are proposed: (1) as soon as the node D transmits the data packet to any mesh member, it is the mesh member responsibility to transmit the packet to the node with a higher priority. As the core node still hasn't received the packet, it has the maximum connected neighboring nodes within the group center and should prioritize multicast/Unicast the D_s . (2) Also, other nodes will cancel their broadcasting as the other node already receives the D_s .

Now it is the responsibility of the core node to send the packet to the destined receiver either through multicasting or unicasting. Now, if the destined receiver is at a 3-hops distance to the core node, it sends the packet through unicasting, as in this case, R43 is at a one hop distance from the core node. However, if the destined receiver is more than 3-hop away, then multicasting will be used. Multicasting increases the reliability in the term link failure or packet drop of the desired packet during

transmission. For example, that D broadcast D_s ; as a result, its neighbor also receives the D_s at the same time. However, it is clear that the neighbors of R35, R36, core node and R38 still didn't receive the D_s . In that situation, the core node will transmit D_s , and R43 will receive it because the core node has a higher priority for D_s with the maximum number of neighbors. Also, the core node transmits the D_s earlier than R35, R36 and R38 and the D_s are overheard by R35, R36 and R38. After overhearing, R35, R36 and R38 do not broadcast their D_s again since R35, R36, and R38 have already received them.

3.6 Algorithm of Modified ERASCA

Depending on the two heuristics as discussed, a mechanism to avoid flooding is used in this paper. The proposed algorithm is consisting of 5 steps which are as follow:

Algorithm: The Proposed Algorithm of Modified ERASCA

```

/*****the initialization phase*****/
1 Broadcast "SD message" about core location
2 Receiving of "SD message" about core node from neighbors
/***** The running phase*****/
3 while (receive a packet  $D_s$ )
4   Calculate the coverage area of  $D_s$ 
5   Update the neighbors that still not receive  $D_s$ 
   and assume that neighbors are represented by  $G$ 
6   if ( $G==0$ ) then
7     Cancel the broadcast of  $D_s$ ;
8   else
9     update the timer to broadcast  $D_s$ ;
10  if (the timer of  $D_s$  expires);
11  Broadcast  $D_s$ ;
12  Multicast/Unicast  $D_s$ ;
13  Break from while loop;

```

- (1) Initially, every node generates its SD message about the status of its position within its neighbors. These phases frequently change in the mobile ad-hoc network and wireless network, but in IoT, the topology slowly changes.
- (2) When node D broadcast D_s within its neighbors, i.e., R35, R36 core node and R38, then the neighboring nodes estimate the coverage area of transmission range R on the position of D.
- (3) By joining the positions of all connected neighbors as shown in step 1 and the transmission range of D_s is calculated in step 2. R35, R36, core node and R38 count the number of neighbors that still have not received the D_s . R35 has one neighbor of R46. Similarly, R36 has one neighbor of the core node, and R38 has neighbors of R1, R2, R3, R4, R5, R39 and the core node. It should be noted that a node with the maximum number of neighbors will set a short timer to receive D_s quickly and then broadcast it immediately before the timing of the other nodes is completed. In this situation, the core node has the maximum number of nodes, i.e., R36, R38, R39, R40, R41, R42, R43, R44, R46 and R50.
- (4) R35, R36, core node and R38 set a timer, which is inversely proportional to the number of connected neighbors that didn't receive D_s , i.e., core node set a shorter timer than R35, R36

and R38. When the number of neighbors drops to zero, it cancels the broadcast Ds , i.e., R35, R36 and R38 cancel the broadcast to avoid duplicate transmission.

- (5) Before the timer goes to zero, all the nodes overhear the broadcast Ds message from its neighbors, before the timer expires. After that, it completes steps 3 and 4 recursively to update the broadcasting Ds timer, i.e., when R35, R36 and R38 overhears Ds broadcasted by the core node, it cancels the broadcast of Ds , as every neighbor has received Ds now.

4 Results Comparison

4.1 Network Simulation

The Modified ERASCA is implemented and compared with the benchmark schemes like ERASCA and PUMA in a Network Simulator 2. The NS-2.35 Ubuntu platform uses C++ and Tcl/Otcl as a back-end language and front-end language for implementing Modified ERASCA. With the help of BASH script, 30 random scenarios are executed through multiple AWK scripts for the collection of data from trace files. The simulations parameters are shown in [Tab. 2](#).

Table 2: Simulation parameters

Simulator	Network simulator (NS-2)
Benchmark protocols	ERASCA and PUMA
Number of ISI	100
Number of sensors	50
Transmission range	250 meters
Simulation area	1000 m × 1000 m
Size of routing queue	50 packets
Total number of generated packets	10000
Packet transmission time	10 ms

However, to do it practically, a social infrastructure should be equipped with short-range wireless communication units, which should be installed in a local area network. The communication is bi-directional, and the IoT-enabled social infrastructures (ISIs) transmission range is defined as R . In smart cities, one ISI should be in the range R of the other transmitter to successfully receive the data packet. Due to the presence of heterogeneous sensors, the status of the resident and community can be monitored and communicated, i.e., children crowd, movement of elderly and disable people, road traffic etc. At least 50 sensors are used to communicate such data to the nearest ISI, and then the data is broadcasted in the IoT network to execute their functions successfully. Here, we assume a vending machine as a reference point in the IoT network. About 100 vending machines can be used within a transmission range of 250 meters in a simulation area 1000 m × 1000 m from each other. Also, 10000 packets should be generated by the sensors with the transmission time of 10 ms of a data packet and the routing queue capacity of 50 packets. Smart cities without public and private transport are not possible, and the speed of the node is taken as 40 m/s.

4.2 Matrices

The following matrices like throughput, overhead, delay, energy and packet drop are used to calculate the performance of protocols using the following parameters as shown in [Tab. 2](#).

Two scenarios are simulated for Modified ERASCA, ERASCA and PUMA protocol in different environment to calculate overhead, throughput, delay, packet drop and energy consumption through Packet transmission interval and mobility.

Case 1: Changing the Packet transmission interval across X-axis

Case 2: Changing the mobility across X-axis

4.2.1 Changing the Packet Generation Interval Across X-Axis

As shown in [Fig. 4a](#), the packet drops decrease in Modified ERASCA compared to ERASCA and PUMA with different packet generation intervals. Within a short interval of packet generation, many data packets were broadcasted in IoT networks and caused to increase the trafficking. This trafficking causes congestion in the network and, as a result, increases the packet drop significantly. Likewise, the flooding in ERASCA and PUMA further increases the congestion and increases the packet drop up to 52% as compared to modified ERASCA, where the packet drop is 26%. Hence the packet drop is almost decreased to half. Nevertheless, the proposed protocol packet drop significantly decreases by avoiding flooding, and hence the packet drops decrease.

In [Fig. 4b](#), the delay decreases in Modified ERASCA compared to ERASCA and PUMA with different packet generation intervals. In short, the interval of packet generation, broadcasting increases in IoT networks and increases congestion significantly. This causes the packet to set in a large queue and wait for the packet delivery. The increase in the queue then increases the delay, and hence the performance of the ERASCA and PUMA decreases. On the other hand, the congestion is negligible, and hence the queue in Modified ERASCA is small. As a result, as shown in [Fig. 4b](#), the delay drops to 1, almost 33% smaller than ERASCA and PUMA. As a result, the delay decreases, and the ISI will receive the packet earlier than the other approaches.

In [Fig. 4c](#), the throughput is increasing in Modified ERASCA compared to ERASCA and PUMA with different packet generation intervals. [Figs. 4a](#) and [4b](#) show that the Modified ERASCA shows more improvement in packet drop and delay than ERASCA and PUMA due to the decrease in congestion. As a result, a decrease in packet drops and delay in Modified ERASCA increases the throughput by 70%. The values are almost double that of ERASCA and PUMA, where throughput drops to 35% and 30%. It should be noted that the improvement of packet drops and delay in [4a](#) and [4b](#) is directly proportional to the increase in the throughput, as shown in [Fig. 4c](#). [Figs. 4d](#) and [4e](#), with different packet generation intervals, increase overhead and energy consumption very slowly in Modified ERASCA compared to ERASCA and PUMA, where the overhead and energy consumption increase at a fast rate. This increase in the rate of overhead and energy is due to the use of flooding. Due to flooding, the congestion increases and increases the packet drop. This causes the source to send the packet to the desired ISI again. This retransmission of data packets in a congested network increases the overhead. As overhead and energy consumption are directly related to each other, the energy consumption increases as shown in ERASCA and PUMA. However, in Modified ERASCA, both overhead and energy consumption are increasing but at a very slow rate due to improved methodology. It is shown in the results that the improvement of Modified ERASCA in overhead and energy is twice greater than ERASCA and PUMA. Also, duplicate packet transmission in flooding affects the performance in terms of overhead and energy.

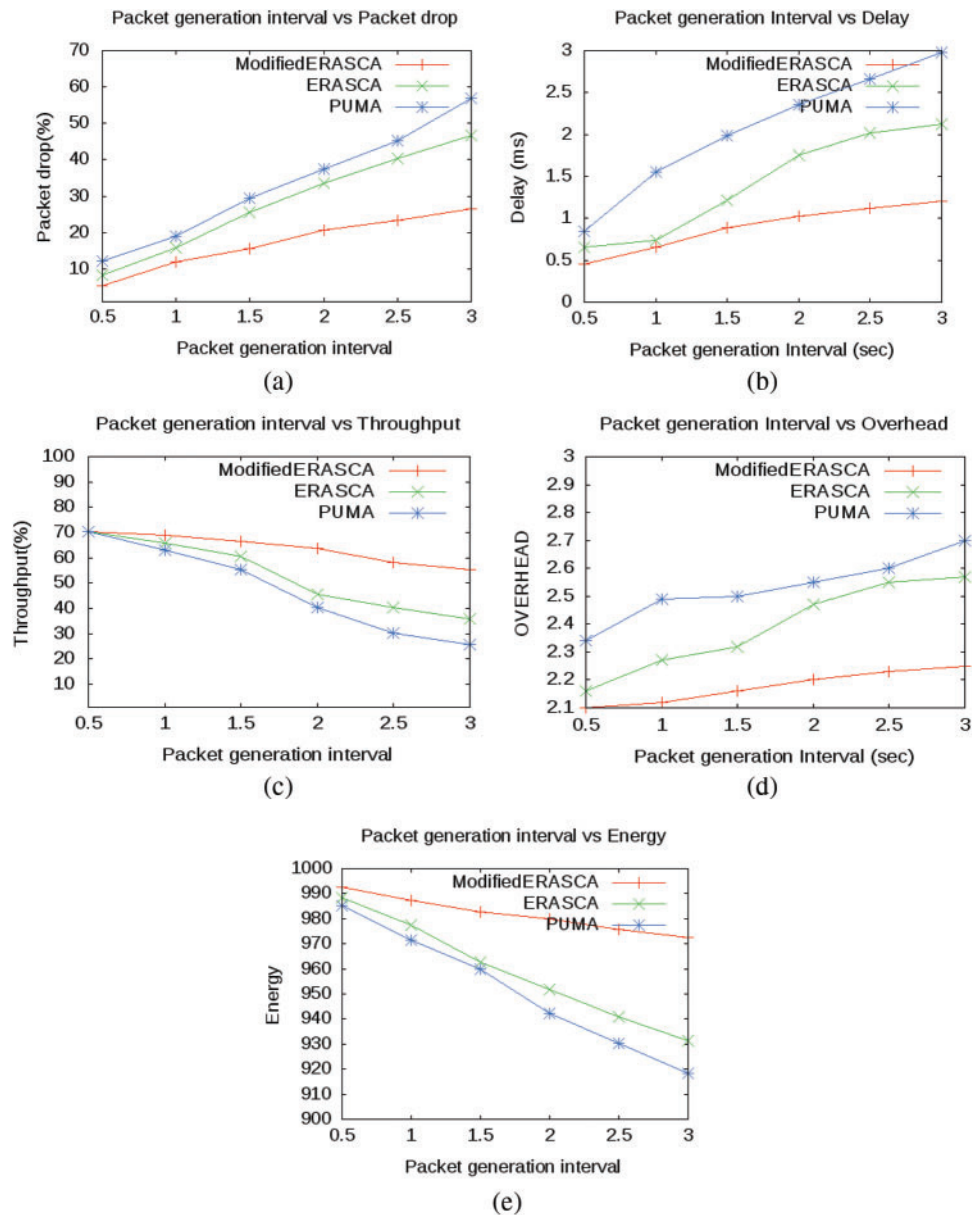


Figure 4: Comparison of packet generation interval with packet drop, delay, throughput, overhead and energy

4.2.2 Changing the Mobility Across X-Axis

In Fig. 5a, mobility is on the X-axis and packet drop is on the Y-axis to compare Modified ERASCA with ERASCA and PUMA. Due to the use of flooding in ERASCA and PUMA, congestion and packet duplication increases. This causes a long queue, and the packet drop increases. The packet drops further increases when the mobility increases. This packet drop is increasing quickly in ERASCA and PUMA due to the use of flooding and decreasing at a considerable low rate in Modified ERASCA due to the use of multicasting and unicasting. Fig. 5b shows that mobility is

shown on the X-axis against the delay to compare Modified ERASCA with ERASCA and PUMA. As shown in Fig. 5a, packet drop increases due to the increase in mobility. As a result, the packet will be retransmitted again; this frequent retransmission creates a flooding storm in the network. This creates congestion and hence delay increases. Also, in high mobility packet drop increases further increases the reconfiguration and hence further increases the delay. Therefore, the delay is more in ERASCA and PUMA due to frequent retransmission, but the delay is less in Modified ERASCA due to its improved technique of avoiding excessive flooding.

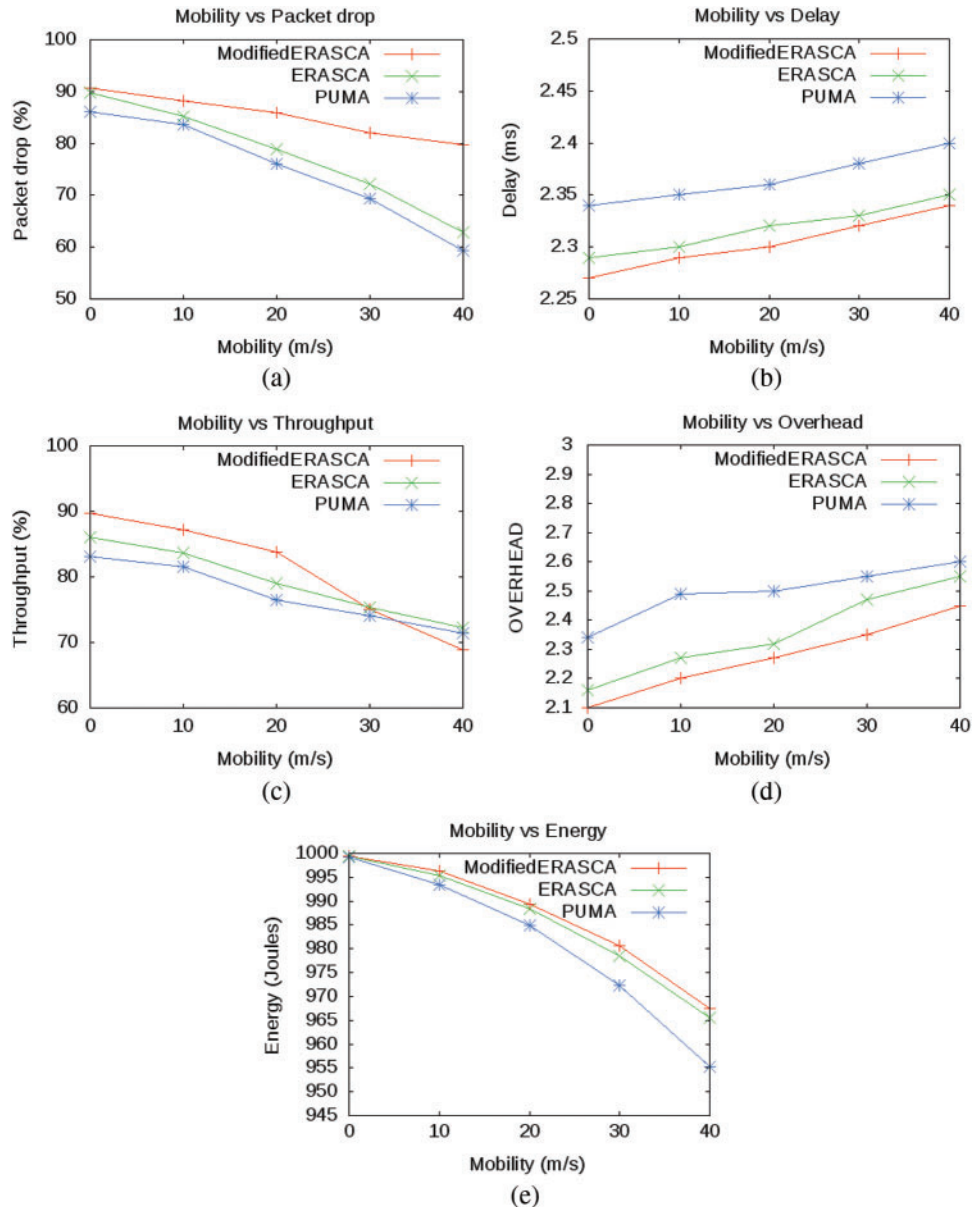


Figure 5: Comparison of mobility with packet drop, delay, throughput, overhead and energy

Fig. 5c shows that mobility is shown on the X-axis against the throughput to compare Modified ERASCA with ERASCA and PUMA. Throughput is affected directly by packet drop, delay, and

mobility, as shown in Figs. 5a and 5b. The increase in packet drop and long delay cause frequent link failure, and the mobile vehicle does not receive the data within a predefined time and decreases the throughput. The throughput is an important factor for mobile nodes in smart cities. Hence, increasing the throughput will improve the activities of the IoT smart cities concept. Fig. 5c shows that the throughput of Modified ERASCA is better than the previous two approaches, but its performance goes down during high mobility. Hence, in high mobility, flooding is the appropriate option, but it creates high overhead, delay and energy consumption. The packet drop is considered from ISI to ISI, and throughput is considered from source to destination.

In Fig. 5d, mobility is shown on the X-axis against overhead to compare Modified ERASCA with ERASCA and PUMA. The overhead of ERASCA and PUMA is already higher due to flooding, but it further increases due to mobility. The mobility further increases the packet drop, packet retransmission, and duplicate packet transmission due to frequent entry and exit of mobile nodes, increasing the overhead. This increase in overhead also increases the energy consumption of the mobile nodes and the IoT network, as shown in Fig. 5e. The performance of Modified ERASCA is stable in terms of overhead and energy as compared to ERASCA and PUMA.

5 Conclusion

This paper introduced a data dissemination mechanism in a IoT network for smart cities through multicasting and unicasting. The proposed technique discourages flooding and using multicasting and unicasting for data transmission broadcasting in the group. The results have validated the improvement in performance in term of packet drop, delay, throughput, overhead and energy utilization by avoiding flooding in Modified ERASCA. On the other hand, PUMA and ERASCA use flooding for data dissemination at the cost of extra overhead and energy and further decrease the performance in terms of throughput, packet drop and delay by introducing mobility. Hence, the Modified ERASCA outperform the PUMA and ERASCA in term of overhead, energy, delay and throughput and packet drop.

Regarding future work, the results showed that the throughput of the proposed mechanism is not giving promising results in high mobility and should be improved. In future, the computational complexity of the proposed scheme can also be performed for better results.

Acknowledgement: The authors thank their families and colleagues for their continued support.

Funding Statement: No funding was received to support any stage of this research study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] K. Shafique, B. A., Khawaja, F. Sabir, S. Qazi and M. Mustaqim, "Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios," *IEEE Access*, vol. 8, pp. 23022–23040, 2020.
- [2] S. K. Tayyaba, A. Akhunzada, N. U. Amin, M. A. Shah, F. Khan *et al.*, "NPRA: Novel policy framework for resource allocation in 5G software defined networks," *EAI Endorsed Transactions on Mobile Communications and Applications*, vol. 4, no. 13, pp. 1–7, 2018.

- [3] S. Ullah, R. Ullah, A. Khan, H. A. Khalid, Q. Zhang *et al.*, “Optical multi-wavelength source for single feeder fiber using suppressed carrier in high capacity LR-WDM-PON,” *IEEE Access*, vol. 6, pp. 70674–70684, 2018.
- [4] S. Ahmad, S. Malik, I. Ullah, M. Fayaz, D. H. Park *et al.*, “An adaptive approach based on resource-awareness towards power-efficient real-time periodic task modeling on embedded IoT devices,” *Processes*, vol. 6, no. 7, pp. 1–29, 2018.
- [5] A. Fuqaha, M. A. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [6] P. V. Patel and B. Kadhiwala, “Broadcasting techniques for route discovery in mobile ad-hoc network-a survey,” in *3rd Int. Conf. on Computing for Sustainable Global Development*, India, pp. 671–674, 2016.
- [7] A. Manimozhi and J. G. Jayanthi, “A technical insight of manet,” *Advances and Applications of Mathematical Sciences*, vol. 18, no. 9, pp. 835–842, 2019.
- [8] A. O. Bang and P. L. Ramteke, “MANET: History, challenges and applications,” *International Journal of Application or Innovation in Engineering & Management*, vol. 2, no. 9, pp. 249–251, 2013.
- [9] S. Mirza and S. Z. Bakshi, “Introduction to manet,” *International Research Journal of Engineering and Technology*, vol. 5, no. 1, pp. 17–20, 2018.
- [10] J. Choi, C. HanGyu and W. Lee, “Method and apparatus for allocating wireless resource in order to prevent interference in wireless LAN,” U.S. Patent No. 10,085,262, 2018.
- [11] H. S. Bedi, K. K. Sharma and R. Gupta, “A review paper on performance analysis of IEEE 802.11e,” in *Proc. of First Int. Conf. on Computing, Communications, and Cyber Security*, Singapore, pp. 47–56, 2020.
- [12] F. Khan, A. W. Khan, S. Khan, I. Qasim and A. Habib, “An algorithmic approach for core election in mobile ad-hoc network,” *Journal of Internet Technology*, vol. 20, no. 4, pp. 1099–111, 2019.
- [13] A. Ghafouri, A. Ghasemi and M. R. H. Ahangar, “A power-based method for improving the ODMRP protocol performance in mobile ad-hoc networks,” *International Journal of Wireless and Microwave Technologies*, vol. 2, pp. 27–36, 2018.
- [14] A. Rashid, F. Khan, T. Gul, S. Khan and F. K. Khalil, “Improving energy conservation in wireless sensor networks using energy harvesting system,” *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 1, pp. 354–361, 2018.
- [15] F. Javed, S. Khan, A. Khan, A. Javed, R. Tariq *et al.*, “On precise path planning algorithm in wireless sensor network,” *International Journal of Distributed Sensor Networks*, vol. 14, no. 7, pp. 1–12, 2018.
- [16] S. M. Hussain, A. Wahid, M. A. Shah, A. Akhunzada, F. Khan *et al.*, “Seven pillars to achieve energy efficiency in high-performance computing data centers,” *Recent Trends and Advances in Wireless and IoT-Enabled Networks*, vol. 2, pp. 93–105, 2019.
- [17] F. Khan, S. Abbas and S. Khan, “An efficient and reliable core-assisted multicast routing protocol in mobile ad-hoc network,” *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 5, pp. 231–242, 2016.
- [18] F. Khan, A. W. Khan, S. Khan, I. Qasim and A. Habib, “A secure core-assisted multicast routing protocol in mobile ad-hoc network,” *Journal of Internet Technology*, vol. 21, no. 2, pp. 375–83, 2020.
- [19] T. S. Sobh, A. Elgohary and M. Zaki, “Evaluation of ASHFUK as core-based routing protocol for critical manets,” *Wireless Personal Communications*, vol. 87, no. 4, pp. 1191–1208, 2016.

- [20] P. Vashist and K. Hema, "New multicast routing protocol in ad-hoc network," *International Journal of Innovations in Engineering and Technology*, vol. 2, no. 2, pp. 2319–1058, 2013.
- [21] A. M. Pushpa and K. Kathiravan, "Resilient PUMA (protocol for unified multicasting through announcement) against internal attacks in mobile ad hoc networks," in *Int. Conf. on Advances in Computing, Communications and Informatics*, Mysore, India, pp. 1906–1912, 2013.
- [22] M. Kumar and R. Mishra, "An overview of MANET: History, challenges and applications," *Indian Journal of Computer Science and Engineering*, vol. 3, no. 1, pp. 121–125, 2012.