An Algorithmic Approach for Core Election in Mobile Ad-hoc Network

Faheem Khan[1], Abdul Wahid Khan[2], Khalid Shah[2], Iqbal Qasim[2], Asad Habib[3]

[1] Department of Computer Science, Bacha khan University Charsadda, KPK, Pakistan
[2] Department of Computer Science, University of Science and Technology Bannu, Pakistan
[3] Department of Computer Science, Kohat University of Science & Technology, Pakistan
kfaheem81@gmail.com, wahidkn@gmail.com, skhalids2000@yahoo.com, iq_ktk@hotmail.com, asadsan@gmail.com

Abstract

Multicasting has an important contribution in a wireless sensor network. Through multicasting, multiple copies of data at the same time in a single transmission can be transmitted to a group of interested users. This reduces the multiple unicasting and hence increases efficiency in term of overhead and reliability in term of delay in the presence of dynamic topology. In this paper, an efficient core is elected within the receiver group on some predefined parameters. Likewise, to increase the reliability in term of delay, a mirror core is introduced. In case of the failure of primary core, the mirror core takes the responsibility as a primary core and communication of the group is continued without any delay. To achieve the above goals, an algorithm is proposed for core-based protocol and mathematical model is used to analyze the overhead performance. Finally, the performance of efficiency and reliability is measured in Network Simulator 2 (NS-2).

Keywords: Multicasting, Core, Mirror core, MANET

1 Introduction

The rapid improvement in wireless network has enabled the researcher/industry to develop a mobile ad-hoc network (MANET) [1-2]. In MANET, routing is simply divided into broadcasting, unicasting and multicasting. In broadcasting one-to-all communication occurs, in unicasting one-to-one communication occurs and in multicasting one-to-many communication is used. The transmission of multiple copies in a single communication increases an efficiency, reliability, routing processing etc. and decreases channel capacity consumption, energy consumption, end-to-end delay [3-6] etc.

Multicasting is very effective in one-to-one situation and the packet is copied when required. Therefore, minimum number of copies per packet are required to distribute the packet to all receivers. Multicasting is very economical in bandwidth and attains higher efficiency as compared to unicasting. In multicasting, MANET is a very well-studied area and myriad of researches and RFCs have been available in this research area. The technology of mobile nodes in MANET interests attracts the attention of the researcher and industry.

The advancement of technology in MANET has gained popularity dramatically from 2000 to 2018. MANET deals more efficiently in term of delay, overhead and bandwidth of communicating data within decentralized environment. The advancement in hardware design, the development in the protocols and the high user requirements in case of mobility and geographical dispersion, continue to produce an excellent need in dynamic environment.

The interest in MANET is increasing by the researcher because of its simple deployment in places where no established infrastructure exists and the applications that need MANET becoming more widespread such as nuclear disaster, earthquake situation, special operation of soldier with certain number of groups in battleground [7-12] etc.

The random changes in topology and mobility are important factors by limiting reliability and efficiency of MANET. The random topology and mobility increases the link failure, which thus increases the core failure in core assisted protocol, where core is the leader of the group by serving all the nodes in the group. It is the responsibility of the core node to spread the Hello messages all over the nodes. The core node is the first node that starts the group however another node can also be selected as a core node when the core node fails. The core failure increases the flooding, needed for the maintenance and update of topology. This frequent flooding increases the overhead and therefore battery lifetime will decrease. Similarly, the link failure increases the packet drop, which causes an increase in delay between the communication and therefore decreases the reliability. This frequent flooding, overhead and delay interested the researcher for the development of various routing protocols to achieve the efficiency and reliability.

*Corresponding Author: Abdul Wahid Khan; E-mail: wahidkn@gmail.com
DOI: 10.3966/160792642019072004010

To reduce the overhead and delay cause by the frequent flooding and core failure, An Efficient and Reliable Core Assisted Multicast Routing in Mobile Ad-hoc Network (ERASCA) [7] is used through core election and mirror core selection, which is discussed below. The paper is arranged as follows. Section 2 describes the related work of multicasting, receiver-initiated mesh based and core-based algorithms. Section 3 shows the complete description of ERASCA, in which connectivity list, mesh formation, core election and an algorithm of ERASCA are explained in detail. Section 4 describes the mirror core responsibility in situation, when a primary core failure occurs. Section 5 describes the overhead analysis created due to mesh formation, maintenance and packet forwarding. In section 6 simulation is performed in Network Simulator (NS2) by taking different scenarios under different parameters between ERASCA, ERASCA-MC, PUMA [6] and MAODV [7]. ERASCA-MC shows the presence of mirror core. Finally, Section 7 concludes the paper.

2 Related Work

The primary goal of ad-hoc multicast protocols should be to construct and maintain a robust and efficient topology even during high network dynamics and limited bandwidth. In MANETs, exchange of information among nodes falling outside of their respective radio ranges within a network is accomplished with the help of intermediate nodes and hence can be termed as multi-hop packet radio network [13]. Multicasting in MANET faces many challenges which deal with problems like scalability, decentralized infrastructure and random mobility.

Multicasting is divided into receiver initiated and sender-initiated mesh-based protocols. In this paper, only receiver-initiated protocols are discussed. In receiver-initiated protocols, receivers of a group elect a core node that is a point of contact among the mesh member nodes. With the help of core node, the group is maintained and updated through periodic hello messages.

Since core election or selection process may affect the overall performance of the protocols. It is necessary to realize the effect of a core selection/election on the performance of routing schemes. Poor core selection/election processes would deteriorate network performance and consume network resources inefficiently. Therefore, a suitable core election process is required to ensure good performance of the nodes and the group. We begin by evaluating the reasons in which the core election is crucial for improving the routing in different scenarios because when the variation in performance is substantial then more appropriate methods for core election are selected. In the existing literature, the core is selected by three ways, i.e. random selection [14], connectivity based selection [15] and first come first serve method [6].

In random selection approach [14], the core in a group is selected on a randomly generated number. Hence, a core is selected without concerning its battery capacity or position. In this approach, the pre-information is not required for the core selection. Therefore, a node in a bad position having minimum number of neighbors and battery capacity can be selected as a core in a group. Because of this inappropriate core selection, frequent core failures appear and therefore another core in a group will be again selected through the data collection process. This frequent data collection process creates substantial overhead and consumes network resources quickly.

In connectivity-based approach [15], the core is selected based on number of connections. The node having maximum number of connections is selected as a core node in a group. In this approach, a node with a good location or with maximum numbers of neighbor can be selected as a core. However, the problem arises when a core node with maximum number of neighbors but having low battery capacity. This core is not maintained for a long time and a core failure will occur soon. This approach is suitable in small networks but in large network its performance decreases drastically because with maximum number of connections the nodes deplete its energy quickly.

In first come first serve method [6], the receivers which start the communication first will become the core of a group. If this core is not suitable in term of battery capacity and position in the group, then core failure will occur soon because the chosen core is selected without knowing the battery capacity and numbers of connected neighbors. This frequent core failure increases the data collection process for another core and quickly consumes network resources. Hence, all those approaches are not favorable in term of lifetime of the network, as in all the schemes the core dies faster. Furthermore, the time and network resources required for the new core selection may cause the protocols to become inappropriate for a Quality of Service (QoS) based applications, especially the delay and overhead caused in the process. In [16-18], the performance deteriorates in case of high mobility and do not deliver the required results. After carefully studying the related literature, the subsequent issues are mentioned.

First, in receiver-initiated mesh-based multicasting the core selection is very important and to the best of our knowledge the algorithms suggested for the core selection are not effective because these algorithms selected several cores within the small region. When multiple cores are chosen then the flooding and overhead escalate, thus the network lifetime will drop. This high flooding increases the collision and packet drop, hence throughput and efficiency decreases.

Second, a situation arises when a single core is used for the operation and maintenance of a group, but when the core node fails or disappears abnormally due to some reasons, for example, flat battery, out of range, hardware fault etc. then the group members again make the reconfiguration/ data collection for another core. This reconfiguration/ data collection process increases the overhead in the form of regular flooding of control messages and an ongoing communication is delayed. Hence, the system is considered as unreliable.

3 Protocol Description

ERASCA practices the IP multicast service model for communication. It is a receiver-initiated protocol and the core node is used for the maintenance and update of the receiver group through Status Declaration (SD) message. In ERASCA, the core node starts the formation of the receiver group through SD message. The SD message contains the core ID, group ID, distance to the core, sequence number and parent node. All receivers in the group are connected to the core through SD message, which is flooded by the core node and establishes the connectivity list.

With the help of connectivity list, all intermediate nodes connecting the receiver with the core and form the mesh (explain in Section 3.2). The connectivity list enables the sender to send its data to any receiver in the group through best possible path.

3.1 Connectivity List

A periodic transmission of SD message from the core node to the concerned group establish a connectivity list in the group. Having connectivity list, every node estimates a best possible route from a sender to the desired group through parent node. Parent node shows shortest path from source node to core node. The source may or may not be the member of the group. Every node stores the information they receive from their neighboring nodes through SD messages along with the receive time. SD message with a higher ID within the neighbors is preferred over the node with lower ID for the same group in the connectivity list. Likewise, for the similar core ID, SD message with a higher sequence number is preferred. Also, for the similar core ID and higher sequence number, SD with a minimum distance to the core is preferred. When all the above fields are similar for the similar core ID then SD message that came earlier is given a preference.

Figure 1 illustrates the distribution of SD message and the structure of connectivity list at node A all through the network. The solid arrow illustrates the reception of best SD message from the neighboring node. Node A having four neighbors like D, X, B and E. Node E cannot be chosen as best possible entry because its hop distance is larger with less battery capacity and larger delay. Node D is also not preferred over node X and B as a valid entry because of its larger delay and less battery capacity. For node A, the best entry is node B because it receives the data earlier than node X. Therefore, node A prefers node B as a best entry to announce its own SD message as given in Table 1. Here node C is acting as a core node.

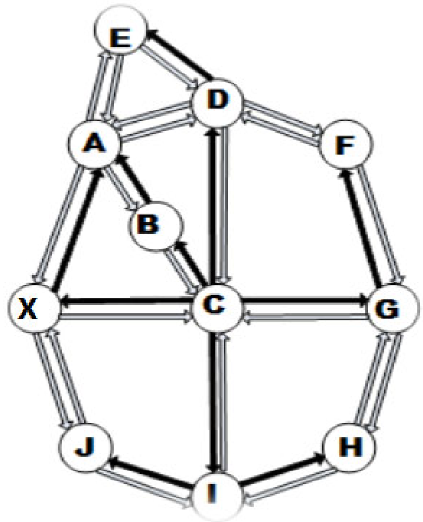

Figure 1. Dissemination of SD Message

Table 1. Connectivity list at node A

N.br	Core ID	Gp. ID	Seq. no	Parent	Dist. to core	Time
B	C	224.1.2.3	76	C	1	22245
X	C	224.1.2.3	75	C	1	22251
D	C	224.1.2.3	75	C	1	22253
E	C	224.1.2.3	75	D	2	22254

N.bor represents Neighbor
Seq. no represents Sequence Number
Gp represents group
Dist represents distance

3.2 Mesh Formation

The nodes are divided into group and non-group member (NM). NM nodes are not part of the mesh and are illustrated in black dots. Likewise, the group member nodes are categorized into Group Relay (GR) nodes, End Receiver (ER) and Intermediate Receiver (IR). The white dot shows the ER and are known as terminal receivers. ER do not take part in packet relay process and the mesh terminated on them. While GR nodes are not receivers and shown in blue dots and acts as middle/intermediate nodes between the core and receivers. It should be noted that it cannot take part in core election process. Similarly, IR nodes are shown in red dots and it acts both as a receiver node and intermediate node simultaneously and can participate in core election process. In Figure 2, R42 is an intermediate node and lies between the core and R47. In this situation R42 acts as a receiver as well as an intermediate node and termed as IR. At first, only receiver acts a mesh member but then GR also performs as a mesh member because GR exists between ER and core and communicates data packets between core and receivers and considers it a part of the mesh. Thus, a mesh group is a combination of GR, IR and ER. IR and GR nodes only forward the transmitted SD message of the core, instead as every member of the mesh. Thus, the flooding is only limited to IR and GR nodes and this decrease the overhead considerably. It should be noted that only IR and ER nodes are selected as a core node, while GR node is not considered as a core node because it only works as an intermediate node and not as a receiver node.

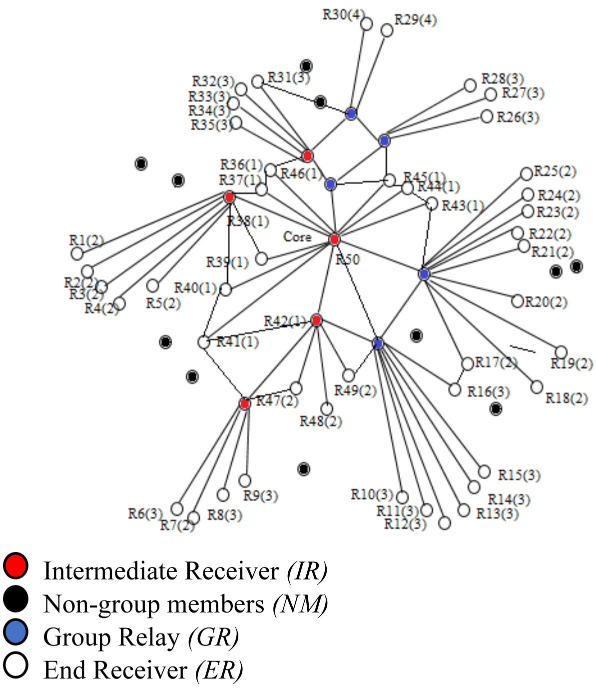

Figure 2. Mesh Formation

3.3 Core Election Algorithm

In previous methods, the core node was selected irrespective of its battery capacity and position within the group. With inappropriate core position, i.e. in less populated area of receivers, the core communicates with remote receiver on the cost of larger overhead and hence decreases the efficiency of the group as well as the network. Similarly, the core with less battery capacity and frequent core failure decreases the lifetime of the network. In ERASCA, an election is performed, to elect the most resourceful receiver as a core based on battery capacity and position (maximum connectivity).

To propose the core election algorithm, the following conditions are necessary: (1) Only a receiver with maximum battery capacity and connectivity in the group should be elected as a core node. (2) All the nodes in the mesh should be maintained and updated by the core node. This algorithm is accomplished on every node in the mesh taking into the respect of the following suppositions about the receivers, nodes and the mesh architecture:

First, every node in the mesh should aware about its 2-hop neighborhood through connectivity list.

Second, every receiver should know every other receiver in the mesh through Receiver Table List (RTL).

Finally, all the receiver in the mesh should aware about the entering of a new member or leaving of an existing member.

3.4 Core Election

To begin a core election in the group, four types of messages are used i.e., SD message, used by every receiver in the group to start the election process; Begin Election Request message, used to announce election by requesting the cost (battery capacity and position) of each receiver in the group; Acknowledge (r), reply by all receivers through Election Reply message; Send CEM, used to flood the cost of every receiver in the group:

receiver-table(r): the list of all receivers in the group voted for the election of the core node *k*.

cost-table(r): the cost of every receiver in which each receiver keeps the cost of all receiver in the group.

neighbors(r): the set of receivers *k*'s neighbors.

corenode(r): The ID of receiver *k*'s core.

core(r): A Boolean variable that sets to TRUE if receiver *k* is a core and FALSE otherwise.

At the start of the communication, a node (Idn) is searching for the existence of any receiver group (group g). If it receives SD message from any receiver group, then it joins that receiver group and become a member of receiver group as Idr. On the other hand, if there is no receiver group then it announces itself as a core node and makes its own receiver group.

Algorithm 1. Before formation of receiver group (start of communication)

```
/* on receiving Status Declaration (SD) message, all
   the nodes will reply along with their cost */
If Idn (received SD message from receiver group g)
then
        Include receiver-group(g);
else if (Idn = Φ) then
        Send SD message
end if
```

However, when the core failure happens, the members of the group are informed through Core Failure Announcement (CFA). The CFA is communicated by IR and GR nodes in the group, if the IR and GR nodes do not receive 3 periodic SD messages from the core. The core node informs the group after every 3 seconds.

As soon as the CFA is announced, an election is performed within a receiver group. A receiver n floods the Election Request message in a receiver group. In reply all the receivers show their availability through Election Reply message, if all the group receivers also verify the core failure. All the receivers in the group are connected to each other in a group through Election Reply message.

To find the remaining batter capacity and number of connected neighbors, a Cost i.e., Core Election Message (CEM) is flooded within receiver group to vote the topmost receiver as a core. In reply all the receivers also flood the CEM to elect the best receiver in a group. Thus, all the receivers having a list of receivers.

Algorithm 2. After formation of receiver group

```
/* After included in the receiver group, all receiver
   replies with their costs via SD message*/
if Idr (receives CFA) then
    Start Begin-election Request message
    (Idr; Costr);
        Send Acknowledge (r);
        Send vote (Idr; Costr);
        Corenode (r) = i;
else if (neighbors(r) = Φ) then
        Send SD message
end if
```

After the information is shared between all the receivers through CEM, a core is elected. The core node floods the news of its selection through SD message in a group and updates the receiver group. The core node selects the mirror core immediately based on cost (explain in Section 4). All receivers will acknowledge core node by receiving it through SD message.

Algorithm 3. Execution by the Elected core node

```
/* Send an Ack to the receiver in a group */
if Core (i) = TRUE; then
        Update receiver-group(g);
        Update mesh-group(g);
        Update receiver-table(r);
        Update cost-table(r);
        Select mirror-core (Idr; Costr);
        Acknowledge (r);
        Send SD message (i);
end if
```

3.5 Adding a New Receiver

When a new receiver enters the mesh then it continually receives the SD message from the core node. To include a new receiver in the mesh, three messages are required. Join Request, Join Reply and SD message. A new receiver n sends Join Request along with Id and Cost to show it willingness for joining the group. Upon receiving the Join Request message, the mesh members reply through Join Reply message. This message allows a new receiver to join the mesh. If the mesh member is a core node, then the Join Reply message contains its cost. Alternatively, if it is a mesh member then the Join Reply message contains the ID of the core node.

Algorithm 4. Execution by the Elected core node or mesh member

```
/* The mesh members send 'Join Reply' to new
   receiver */
if (Core (r) = TRUE then
        Join Request = (Idr; Costr);
        Send Join Reply (Idr; Costr);
        Update receiver-group(g);
        Update mesh-group(g);
        Update receiver-table(r);
        Update cost-table(r);
else
        Join Reply = leadermode(r)
end if
        Send SD message (i);
```

3.6 Removing a New Receiver

When a receiver leaves a mesh due mobility, hardware fault or battery failure, then the core node updates the receiver group as well as mesh group through SD message about the departure of the specific receiver from the mesh.

Algorithm 5. Execution by the Elected core node or mesh member

```
/* The receiver send 'leave' to mesh member */
if (receiver (r) = leave) then
        Update receiver-group(g);
        Update mesh-group(g);
        Update receiver-table(r);
            Update cost-table(r);
end if
Send SD message (i);
```

The core election process having the following aspects. A stable core is elected on parameters like battery capacity and numbers of connected neighbors. Likewise, a core failure does not occur frequently and hence decreases the data collection process. Thus, decreases the overhead.

4 Mirror Core Selection

To solve the problems related to core failure, an idea of mirror core is introduced. The core node selects the mirror core according to battery capacity and distance to the core instead of election because election uses more battery capacity and overhead as compared to selection. Hence, when the core node fails because of mobility, hardware fault and battery capacity then the mirror core will act as a core node and maintains and updates the mesh without any delay. Mirror core has several advantages like less delay, less data collection process and increase in reliability.

After core election process, it is necessary for the core node to select the second topmost receiver as a mirror core (shown in yellow) with high battery capacity and with less distance to the core for the mesh as shown in Figure 3. For the selection of mirror core the following steps are followed.

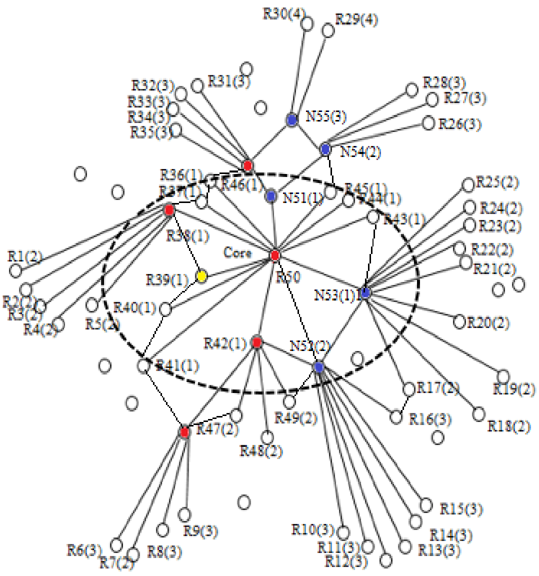

Figure 3. Mirror Selection within neighborhood

Step 1: The core node floods the *Mirror Core Selection Request (MCSReq)* through *ER* and *IR*. Through this message, *ER* and *IR* are informed that the mirror core is not yet selected. In reply, *ER* and *IR* unicast the *MCSRep* towards the core node to confirm the absence of mirror core. Through *MCSRep, ER* and *IR* are showing their willingness in mirror core selection procedure. Also, all *ER* and *IR* will create routes to the core within the group.

Step 2: To find battery capacity and distance to core of each *ER* and *IR*, a *Mirror Core Selection Message (MCSM)* is flooded by the core in a group. In response *ER* and *IR* also unicast *MCSM* to the core.

Step 3: Finally, a core node is having a list of all receivers with their *BC* and *distance to core* and now capable to select a mirror core from the list. Hence, when the main core fails the mirror core acts as a main core and maintains and updates the group.

It should be noted that when the mirror core will take the responsibility as a core node. First, when the core node moves out of the radio range of mesh members because of hardware fault or mobility. Second, when the core node having the shortage of its resources, e.g. battery capacity and announcement of “resource exhaustion” message. It should be noted that only IR and ER are selected as a mirror core and not a ground relay receiver.

4.1 Connectivity List of Mirror Core

The mirror core should be selected with high battery capacity and within one-hop distance from the main core. However, if a mirror core is not found within one-hop distance then the mirror core could be selected up to 4-hop distance. To select the mirror core, Mirror Core Selection Message (MCSM) is flooded by the core node in the group. The purpose of this message is to receive the status of every receiver in the group. In respond, all the receivers unicast their battery capacity and distance from the core to the core node, which gives an authority to the core for the selection of mirror core.

4.2 Multicast Data Forwarding

The performance of parent node is very important in ERASCA. It simply forwards a data packet of neighbors to the mesh group. When connectivity lists are formed for the mesh group, the sources which are not the member of the mesh group are also aware about the presence of group and become capable to send the data packets towards the destined receiver of a mesh group. As soon as, the data packet is received to any member of the mesh group, it is flooded inside the mesh group for the purpose to be received by the destined receiver. The data packet received twice by any mesh member is rejected because the identities of the flooded packets in the mesh are saved in every mesh member ID cache. In Figure 4, R50 is parent of R38, R38 is parent of D and so on. The relationship of parent to child continues until it reaches to source. The SD message flows from R50 to sender in a parent to child fashion to form a connectivity list. On the other hand, the data flows from sender to core in a child to parent fashion to reach the mesh.

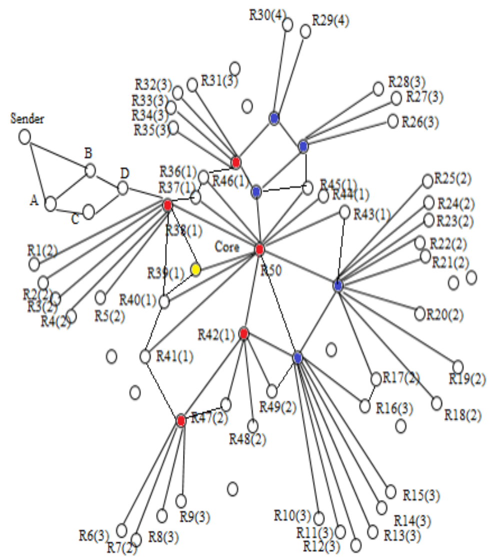

Figure 4. Data Forwarding

5 Mesh Formation Analysis of ERASCA

At the start of communication, the first core in ERASCA is selected on traditional approaches like PUMA and MAODV. However, after the failure of the first core an election is conducted within the receiver group to elect the core as explained in Section 3.3. The messages created by all receivers during election are represented as R= EReq + ERep + CEMReq + CEMRep and the messaging overhead created by all the receivers during election within the group is (EReq + ERep + CEMReq + CEMRep) × d (k-1)2. The total numbers of receiver transmission are calculated with the help of (R - IRGR). When the core is elected, core floods the SD message Core × d (k-1)2 to inform every receiver within the group. In ERASCA, it is the responsibility of IR and GR to further flood the SD message within the group. Collectively the overhead of IR and GR is calculated in term of IRGR and this IRGR overhead is considered mostly as the messaging overhead for the formation of mesh. The flooding of SD message that creates due to the transmission of messages from the IR and GR is IRGR × d (k-1)2.

It is assumed that *IRGR* are uniformly distributed in a group with random topology. The messaging complexity between two mesh members is 2k + IRGR [2 + $(4d_{IRGR} - 1) \in$], where $\in$ is the percentage of *IRGR* in the mesh under the core node. When the core is elected, core informs all *IRGR* through a total of IRGR (1-$\in$) messages. After mesh formation, each *IRGR* announces itself to receivers within a 3-hop neighborhood. For a uniform distribution, the average number of *IRGR* up to k-hop neighborhood is dk^2 [19], where *k* is the number of hops and *d* is the average degree of the *IRGR* in the group. *IRGR* announcement consumes d $(3-1)^2$ = 4d messages and the total numbers of messages are IRGR + IRGR (1-$\in$) + 4 d_pIRGR = IRGR [2 + $(4d_{IRGR} - 1) \in$]. Also, the path length between two *IR* and *GR* is *2k*. Thus, produce 2k + IRGR [2 + $\in$].

The analytical performance calculation of ERASCA is shown in Figure 5, depends on the following assumptions. First, *IRGR* can be approximated by a broadcast range *r* with radius = kr. Second, the core must be selected at the center of the group. Depending on the above assumption, it is concluded in [19] that *IRGR* would have an average number of receivers within its k-hop is R×IRGR = dk^2. Therefore, the average number of *IRGR* within the group are IRGR = R /dk^2. However, the number of receivers up to *k-1* hops are d $(k-1)^2$. By combining all the steps, the total number of messages for the formation of mesh in ERASCA are:

$$
\begin{aligned}
Message_{IRGR} &= (R - IRGR) + Core.d(k-1)^2 + \\
&\quad IRGR.d(k-1)^2 + 2k.IRGR[2 + (4d_{IRGR} - 1) \in] \\
&= Core.d(k-1)^2 + (R - IRGR) + IRGR.d(k-1)^2 \\
&\quad + 2k.IRGR[2 + (4d_{IRGR} - 1) \in] \\
&= Core.d(k-1)^2 + (R - IRGR) + IRGR.d(k-1)^2 + \\
&\quad 2k.IRGR[2 + (4d_{IRGR} - 1) \in]
\end{aligned} \tag{1}
$$

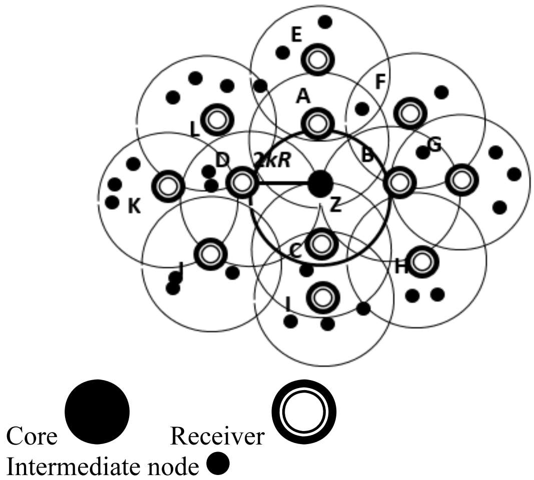

Figure 5. Uniform cluster density with a distance of 2kR between two intermediate nodes in a row

Putting the value of IRGR in equation 1

$$
\begin{aligned}
&= Core.d(k-1)^2 + (R - \frac{R}{dk^2}) + d(k-1)^2.IRGR + \\
&\quad 2k.IRGR[2 + (4d_{IRGR} - 1) \in] \\
&= Core.d(k-1)^2 + (\frac{Rdk^2 - R}{dk^2}) + d(k-1)^2.IRGR + \\
&\quad 2k.IRGR[2 + (4d_{IRGR} - 1) \in] \\
&= Core.d(k-1)^2 + (\frac{R(dk^2 - 1)}{dk^2}) + d(k-1)^2.IRGR + \\
&\quad 2k.IRGR[2 + (4d_{IRGR} - 1) \in]
\end{aligned}
$$

As we know R/ dk^2 is equal to IRGR,

$$
\begin{aligned}
&= Core.d(k-1)^2 + (dk^2 - 1).IRGR + d(k-1)^2.IRGR + \\
&2k.IRGR[2 + (4d_{IRGR} - 1) \in] \\
&= Core.d(k-1)^2 + (dk^2 - 1).IRGR + d(k-1)^2.IRGR + \\
&2k.IRGR[2 + (4d_{IRGR} - 1) \in]
\end{aligned}
$$

Taking IRGR common

$$
\begin{aligned}
&= Core.d(k-1)^2 + IRGR([(dk^2 - 1) + d(k^2 + 1 - 2k)] \\
&\quad + 2k[2 + (4d_{IRGR} - 1) \in] \\
&= Core.d(k-1)^2 + IRGR([(dk^2 - 1) + dk^2 + d - 2kd] \\
&\quad + 2kd[2 + (4d_{IRGR} - 1) \in] \\
&= Core.d(k-1)^2 + IRGR([(dk^2 - 1 + dk^2 + d - 2kd)] \\
&\quad + 2k[2 + (4d_{IRGR} - 1) \in]
\end{aligned}
$$

$$
\begin{aligned}
Message_{IRGR} &= Core.d(k-1)^2 + IRGR([2dk^2 \\
&\quad - 2kd + d - 1 + 2k[2 + (4d_{IRGR} - 1) \in])
\end{aligned} \tag{2}
$$

5.1 Maintenance Overhead Analysis

Consider a link failure rate "μ" because of the mobility. The effect of overhead is calculated for the maintenance of the route among the mesh in the presence of link failure, which are as follows:

A. Receiver to move out of the range of its (a) existing *IR* or *GR* (b) move to a neighboring *IR* or *GR* within a range.

B. Mirror core to move out of the range of its (a) existing *IR* or *GR* (b) move to a neighboring *IR* or *GR* within a range

C. A core to become unreachable to mesh members.

For ERASCA, the probability is calculated of the above-mentioned scenarios and evaluate the messaging overhead for the maintenance of mesh. The analysis is given below to estimate k-hop with a radio range of radius r.

In scenario A (a), the messaging overhead is 2k × 2, because before leaving it informs the existing IRGR from which the receiver is related and the neighbor IRGR. Likewise, after joining a new IRGR, it informs two neighboring nodes on the mesh. The probability that a receiver leaves the existing IR or GR is $\frac{(2k-1)}{k^2}\times\infty$ based on the following derivation:

The average number of IRGR up to k-hop neighborhood is calculated as dk2 and the number of periphery receivers on the edge are $dk^2-d(k-1)^2 = d(2k-1)$. The probability that one of the receiver leaves the IRGR is $\frac{d(2k-1)}{dk^2}$ and the probability for the edge receivers to depart from the IRGR out of total receiver's R is $\frac{d(2k-1)}{R}=\frac{d(2k-1)}{dk^2}$.

The term ¥ shows the probability that a receiver moves outside the range of the mesh i.e., in the uncovered area. Here the uncovered area is calculated as:

$\frac{size\ of\ all\ IRGR}{size\ of\ covered\ area}=\frac{\pi(kr)^2\times IRGR}{area}$, where r is the radio range of the receiver. Thus,

$$\infty=\{(1-\frac{\pi(kr)^2\times IRGR}{area}), if\ \frac{\pi(kr)^2\times IRGR}{area}<10, otherwise\}$$

In scenario A (b), the messaging overhead is 2k to inform the old and new neighbor i.e., IR or GR. It means that the receiver is moving from one neighbor to another neighbor within the covered area. The probability of this movement is $\frac{(2k-1)}{k^2}\times 1-\infty$.

In scenario B (a), the messaging overhead is 2k × 2 and the probability that a mirror core leaves the existing IR or GR is $\frac{(2k-1)}{k^2}\times\infty$ as explained above. In scenario B (b) the messaging overhead is 2k because mirror informs its own IR or GR as well as the core node about its movement. The probability of this movement within a neighborhood is $\frac{(2k-1)}{k^2}\times 1-\infty$.

In scenario (C) when the core node moves because of mobility from the center of the group to the edges of the group, then the group members are in the orphanage phase, this movement of the core node from the center to the edges is handled by situation (A) and (B). Generally, the core node leads the orphanage receivers and the probability for this is $\frac{2}{dk^2}$ [19] based on the radio range placed at the center of the group. The number of messages for the new core to lead the receiver group is $\frac{1}{2}dk^2-1$. Also 2k× dIRGR messages are necessary to update the receiver group. The average number of messages to handle changes caused by the departure of the receiver from the group, i.e., in (A) and (B) is:

$$Message_{Receiver}=2[4k\times\frac{(2k-1)}{k^2}]\times +2k\frac{(2k-1)}{k^2}\cdot 1-\infty]$$

Taking common 2k$\frac{(2k-1)}{k^2}$

$$Message_{Receiver}=2[2k\times\frac{(2k-1)}{k^2}\times(2\infty+1-\infty)]$$

$$Message_{Receiver}=2[2k\times\frac{(2k-1)}{k^2}(2\infty+1-\infty)]$$

$$Message_{Receiver}=\frac{4k-1}{k}(\infty+1)$$

Likewise, the number of messages required, when the core leaves the group is:

$$Message_{Core}=\frac{2}{dk^2}[(\frac{1}{2}dk^2-1)+(2k\times d_{IRGR})+ +4k\times\infty+2\text{k}\times(1-\infty)]$$

$$Message_{Core}=\frac{2}{dk^2}[(\frac{1}{2}dk^2-1)+(2k\times d_{IRGR})+ +2k(2\infty+1-\infty)]$$

$$Message_{Core}=\frac{2}{dk^2}[(\frac{1}{2}dk^2-1)+(2k\times d_{IRGR})+ +2k(\infty+1)]$$

The average number of messages by combining MessagesReceiver and MessagesCore for maintaining the mesh is given in equation 3:

$$Message_{ERASCA}=\frac{2}{dk^2}[(\frac{1}{2}dk^2-1)+(2k\times d_{IRGR})+ +2k(\infty+1)]+\frac{4(2k-1)}{k}(\infty+1) \quad (3)$$

5.2 nalysis of Overhead

In ERASCA three steps are followed:

Go to the mesh node: To transfer the data from source to group, the source sends the data towards IRGR. In favorable situation, hop between the source and the mesh member is 1k, however, in other situations it is 2 or 3 i.e., nk.

Flood within the IRGR: As soon as any mesh member (IRGR) receives the data from the source, it is flooded within IRGR.

Deliver to destined receiver: After flooding within *IRGR*, the destined receiver receives the data. Assume that there are *n* members in the multicast group and the number of *IRGR* in the mesh is a fraction 5. Therefore, several *IRGR* in the core is 55×5555. Assume that *n* receivers are equally dispersed within the *IRGR*, and (1-5) is on the edge of the *IRGR*. Therefore, the numbers of transmissions in ERASCA for packet forwarding are:

$$= 5+25\times(5555-1) + 5\times(1-5)\,55 + 5\times5n$$
$$= 5+25\times(555555-1) + 55-555 + 55\times5n$$
$$= 5+25\times(55555-1) + 55 \quad \textbf{(4)}$$

The analytical performance estimates for ERASCA in mesh formation is shown in equation 2. Likewise, the mesh maintenance and packet delivery messaging complexity are shown in equation 3 and 4 subsequently. Consider the path length between two intermediate nodes in a circle of radius k are 2k hops. Similarly, consider a uniform distribution, where two intermediate nodes are adjacent if their overhead is 2k hop neighbors between each other as shown in the Figure 5. Table 2 shows the summery of analytical performance of ERASCA.

Table 2. Analytical performance estimates for ERASCA and MCEDAR

Messaging Complexity	CCCC ERASCA
Formation	Core.d(k-1) + IRGR ([$2dk^2$ –2kd + d-1+ 2k [2 + (4d_{IRGR} -1) ε])
Maintenance	2/dk2 2/dk^2[(1/2 dk^2-1) + (2k× 4d_{IRGR}) 2k(∞+1) + 4(2 k-1)/k (∞+1)
Packet Delivery	5+25×(5-1)+55

6 Performance Evaluations

6.1 Simulation Setup

ERASCA and ERASCA-MC are compared with the benchmark protocols like PUMA and MAODV in a Network Simulator-2 (NS-2). NS-2.35 version is used with Tcl/otcl and C++ on Ubuntu platform. An AWK script is developed to gather data from trace files and analyze the performance of ERASCA-MC, ERASCA, PUMA and MAODV. The simulation parameters are shown in Table 3.

6.2 Protocol Evaluation

Four scenarios are performed to evaluate number of receivers, mobility, ifqlen and simulation area for protocols like ERASCA-MC, ERASCA, PUMA and MAODV. Likewise, with the help of these scenarios conclusions are drawn on the attained results.

Table 3. Simulation parameters

Simulator	Network simulator (NS2)
Observed Protocols	ERASCA-MC, ERASCA, PUMA a
Simulation time	450 Sec
Number of nodes	45
Simulation area	1000m x 1000m
Data packet size	512 bytes
IfqLen	60
MAC type	802_11g

6.2.1 Variation in Mobility

In this scenario, mobility is changed form 0 m/s – 40 m/s to determine its effect on ERASCA-MC, ERASCA, PUMA and MAODV. In high mobility, the core failure increases. This core failure increases the packet drop between the source and the destination because it is the core node which maintains and updates the group. Thus, the source will resend the data until another core is elected. The time until another core is elected creates a delay and destination receives the data with longer delay as shown in Figure 6. This core failure triggers the flooding for another core election and hence produces the overhead. This frequent flooding and overhead consumes network resources quickly and hence decreases the energy of individual node as well as of the group and hence the system is not considered efficient in term of energy and not reliable in term of delay. The performance of PUMA and MAODV is not satisfactory because of the inappropriate core election. On the other hand, the performance of ERASCA is better than both protocols because of the stable core election based on battery capacity and position in the group and hence core failure, delay and energy consumption is minimum. Likewise, ERASCA-MC further improves the performance in term of delay, core failure and energy consumption because in core failure situation the mirror core becomes a primary core and decrease the flooding for another core election.

6.2.2 Variation in Receivers

In this scenario, receivers are varied from 1-5 within a simulation time of 450 seconds. As shown in a Figure 7, when receivers are increasing then delay is decreasing because a distance between receivers is decreasing. The less distance between receivers decreases the packet drop and therefore decreases resending of data between a source and a destination and hence decreases the core failure. This decrease in core failure decreases the reconfiguration for another core and thus decreases the energy consumption. The core failure occurs soon, when the number of receivers

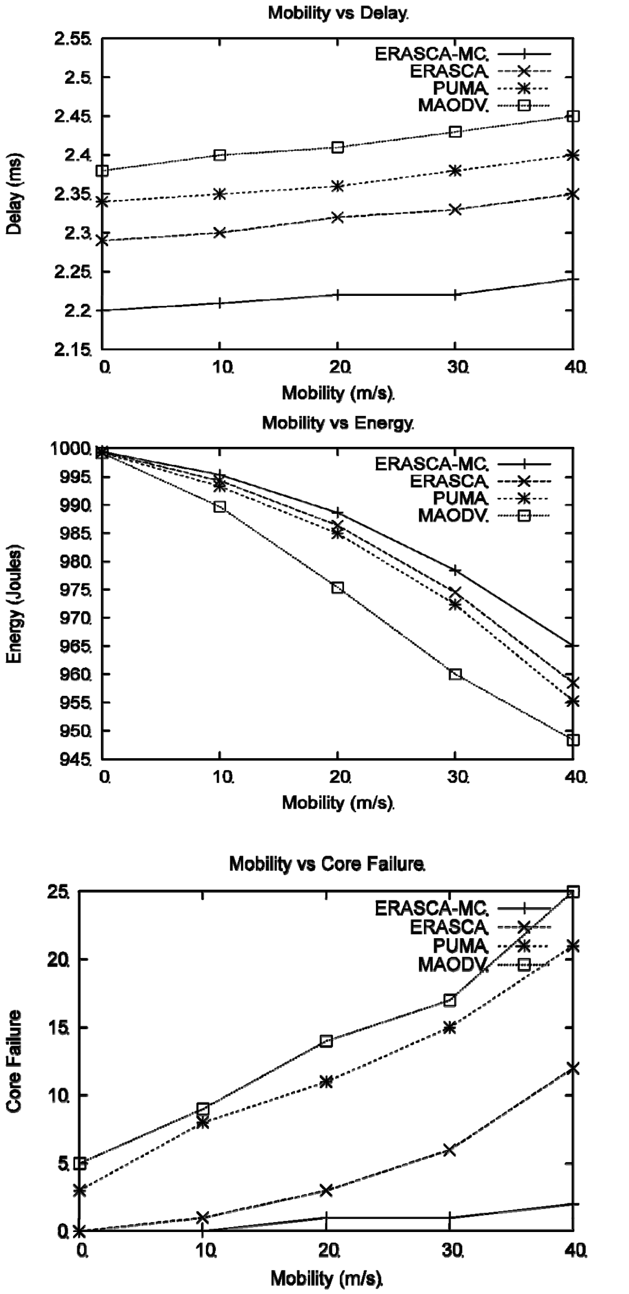

Figure 6. Comparison of mobility with delay, energy and core failure

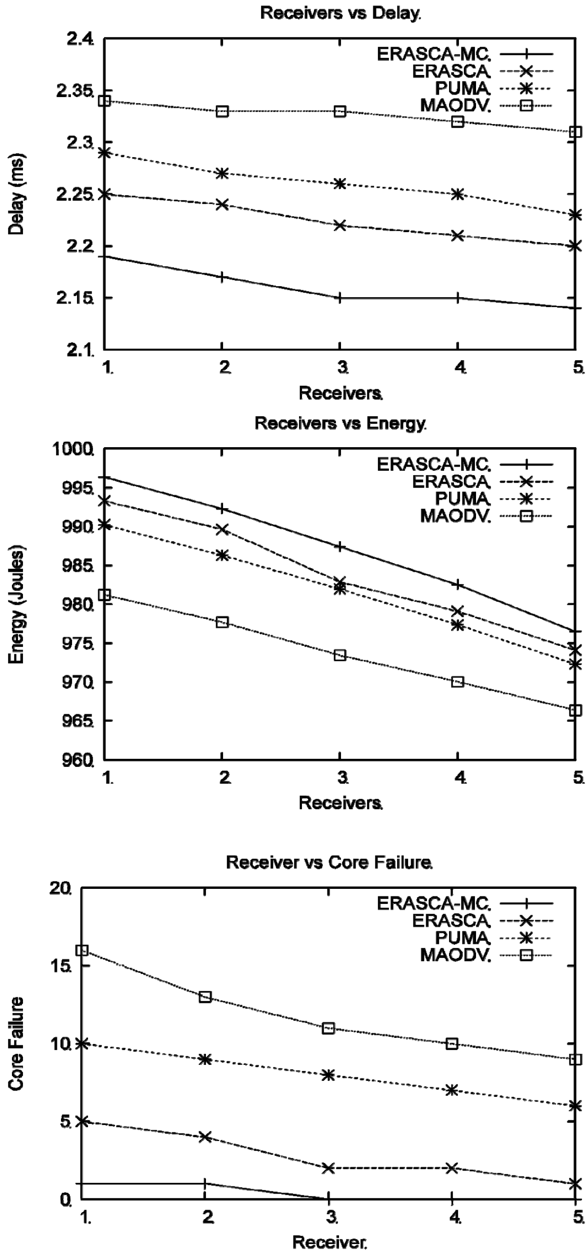

Figure 7. Comparison of receivers with delay, energy and core failure

increases in PUMA and MAODV because of the inappropriate (core with less battery capacity and less connected neighbors) core election. Hence, the increase in receivers deteriorate the performance of PUMA and MAODV as compared to ERASCA and ERASCA-MC, where a stable core is elected with maximum battery capacity. Also, an increase in receivers do not make the core failure soon. Thus, the group with less delay, core failure and energy consumption will have an increase in lifetime of the network in ERASCA and ERASCA-MC and the users entertain from the resources without any interruption. ERASCA-MC performs better in term of delay, core failure and energy consumption than ERASCA, PUMA and MAODV because of the mirror core.

6.2.3 Variation in ifqLen

In this scenario, ifqLen is changed from 10-80 as shown in a Figure 8. Figure 8 represents that an increase in the ifqLen decreases the packet drop because in large ifqLen maximum number of packets reaches from source to destination with less packet drop and core failure and therefore, less delay occurs. However, the decrease in ifqLen narrowing the buffer size and hence core failure and packet drop is increasing with maximum delay as shown in the Figure 8, however, the performance of ERASCA and EASCA-MC are better than PUMA and MAODV because of the presence of stable core and mirror core.

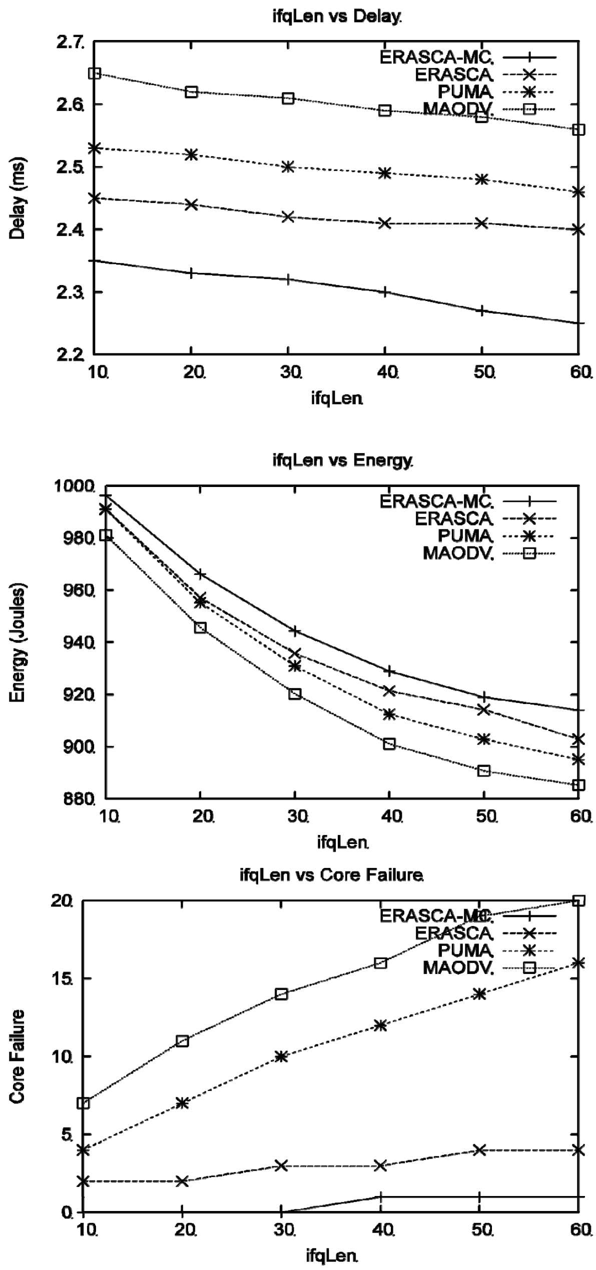

Figure 8. Comparison of ifqLen with delay, energy and core failure

6.2.4 Variation in Simulation Area

In this scenario, simulation area is changed from 500-2000 as shown in a Figure 9. Generally, all routing protocols in MANET show better results within a small simulation area with maximum number of nodes in comparison to large simulation area, where the performance goes down drastically. The same situation is shown in a Figure 9, where the performance of PUMA and MAODV is decreasing when the simulation area is larger. In large simulation area, the core failure increases, which trigger the flooding and therefore overhead of the group increases. This frequent flooding and overhead increases the energy consumption of the group and hence decreases the lifetime of the network. In contrast, the performance of ERASCA and ERASCA-MC is better than PUMA and MAODV because in the former approaches the stable core is elected on high battery capacity and best position (probably in the center) in the group. Also, in large simulation area when the core failure increases the mirror core takes the responsibility as a main core and the group minimizes the data collection/ reconfiguration process for another core. Hence decreases the flooding, overhead, core failure and delay with improve performance.

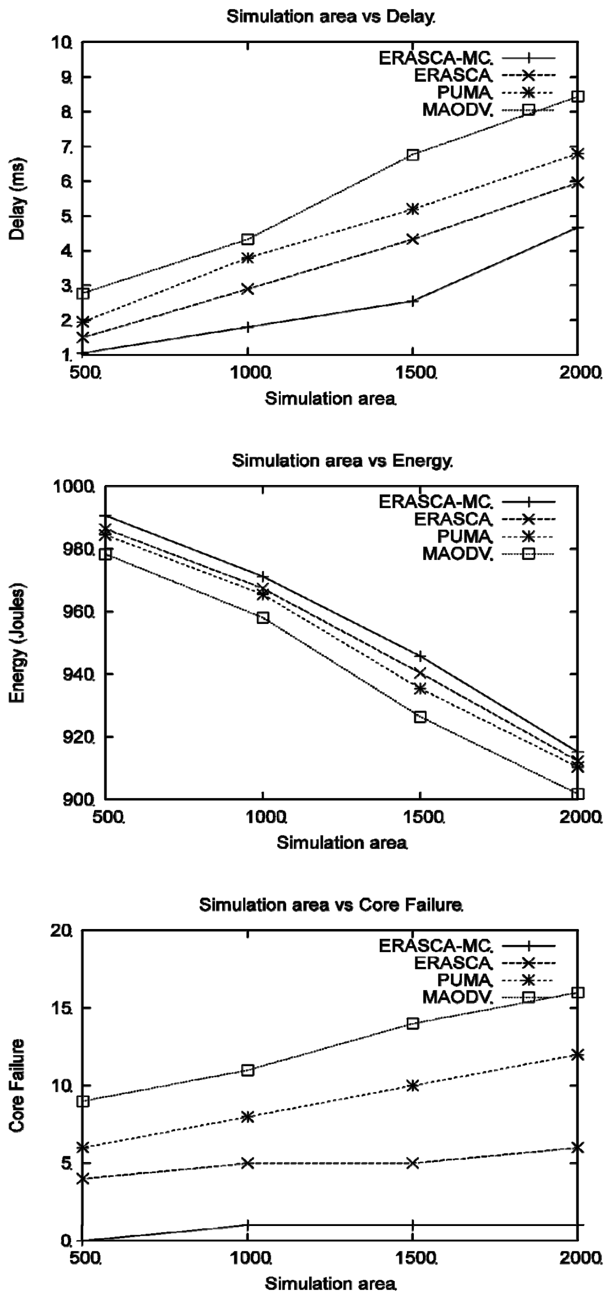

Figure 9. Comparison of simulation area with delay, energy and core failure

7 Conclusion

This paper gives the concept of algorithmic approach of ERASCA and ERASCA-MC in multicasting protocols. In ERASCA, an efficient core based on battery capacity and position is elected and in ERASC-MC the mirror core is introduced in case of core failure situation. A comparison of ERASCA is made with PUMA and MAODV in term of mobility, core failure, delay and energy consumption, which shows that ERASCA performance is better than PUMA and MAODV. To further improve the performance of ERASCA by reducing the data collection process during core failure, a mirror core is introduced and hence termed as ERASCA-MC. ERASCA-MC further improves the performance in term of core failure, delay and energy consumption and hence shows the dominance of ERASCA-MC over ERASCA, PUMA and MAODV.

There is some future work in this paper and if these problems are addressed properly then it will further improve the performance of our approach. In overlapping area, a receiver may be included in more than one group and included in more than one overlapping area as shown in Figure 5. Overlapping area increase the resilience of mesh-based protocols in MANET by providing redundant paths between two groups, increasing the group robustness against boundary receiver's failures and sharing the packet forwarding loads. However, receiving data from two groups is not useful for the concern receiver battery, which falls in the overlapping area. Therefore, in future we should evaluate the overlapping area and avoid the receivers not to receive the data from two intermediate nodes. As a result, a receiver battery lifetime will increase.

The most important factor in MANET is energy consumption. If the energy consumption of the network is high, then the lifetime of the network will decrease, and partition of the network will occur soon. For network stabilization, the energy consumption of the battery should be minimized because if the battery depletes the energy quickly, then the network performance and stability will degrade considerably. Therefore, we should use energy efficient techniques in designing new protocols. First, energy buffers should be used, which will take small input current and produce a high output current. Second, nodes should be entering the power saving mode for a short period of time to minimize the energy consumption. Third, energy consumption can be reduced by powering off the nodes that are not between the communications nodes with the help of specific time slotting. Fourth, if appropriate (in a remote place) use solar cells.

References

[1] M. Bouhorma, H. Bentaouit, A. Boudhir, Performance Comparison of Ad-hoc Routing Protocols AODV and DSR, *International Conference on Multimedia Computing and Systems 2009*, Ouarzazate, Morocco, 2009.

[2] J. Yao, X. Yen, C. Qian, H. Li, Wireless Network Localization Algorithm Based on Tikhonov Regularization for Anisotropic Networks, *Journal of Internet Technology*, Vol. 19, No. 3, pp. 927-938, May, 2018.

[3] L. Junhai, Y. Danzia, X. Liu, F. Mingyu, A Survey of Multicast Routing Protocols for Mobile Ad-Hoc Networks, *IEEE Communications Surveys & Tutorials*, Vol. 11, No. 1, pp. 78-91, March, 2009.

[4] C. E Perkins, *Ad Hoc Networking*, Addison-Wesley Professional, 2008.

[5] P. Mohapatra, S. Krishnamurthy, *Ad Hoc Networks: Technologies and Protocols*, Springer Science & Business Media, 2004.

[6] Y. Padmanaban, M. Muthukumarasamy, Energy-efficient Clustering Algorithm for Structured Wireless Sensor Networks, *IET Networks*, Vol. 7, No. 4, pp. 265-272, July, 2018.

[7] F. Khan, S. Abbas, 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, September, 2016.

[8] M. Vijayalakshmi, D. S. Rao, Quality of Service Optimization of Multicast Routing Protocols in Mobile Ad-Hoc Networks, AM Route, PUMA and ODMRP, *International Journal of Computer Applications*, Vol. 130, No, 1, pp. 10-14, November, 2015.

[9] S. Baolin, L. Layuan, On the Reliability Of MAODV in Ad Hoc Networks, IEEE International Symposium on Microwave, Antenna, *Propagation and EMC Technologies for Wireless Communications*, Beijing, China, 2005.

[10] W.-H. Lee, C.-M. Chou, Yu.-H. Kuo, W.-Lun. Hung, Da-Wei. Chang, Development of a Roadside Unit Middleware for Vehicular Ad-Hoc Network Based Intelligent Transportation Systems, *Journal of Internet Technology*, Vol. 18, No. 6, pp. 1431-1439, November, 2017.

[11] K. Kumar, G. P. Mishra, A. P. Prakash, R. Tripathi, A Proactive Spectrum Handoff Scheme with Efficient Spectrum Utilisation for Cognitive Radio Ad Hoc Networks, *International Journal of Internet Protocol Technology*, Vol. 10, No. 3, pp. 160-176, October, 2017.

[12] G.-S. Li, X.-H. Chen, Ke.-S. Wu, R. Chen, Direction-based Urban Broadcast Protocol for Vehicular Ad Hoc Networks, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 28, No. 3, pp. 180-192, June, 2018.

[13] B. Soujanya, T. Sitamahalakshmi, C. Divakar, Study of Routing Protocols in Mobile Ad-Hoc Networks, *International Journal of Engineering Science and Technology*, Vol. 3, No. 4, pp. 22-27, 2011.

[14] A. Kodole, P. Agarkar, A Survey of Routing Protocols in Mobile Ad Hoc Networks, *Multidisciplinary Journal of Research in Engineering and Technology*, Vol. 2, No. 1, pp. 336-341, April, 2015.

[15] R. Malhotra, R. Aggarwal, S. Monga, Analyzing Core Migration Protocol Wireless Ad Hoc Networks by Adopting Multiple Nodes Diverse Test-bed, *International Journal of Engineering Science and Technology*, Vol. 2, No. 10, pp.

5326-5331, November, 2010.

[16] Y. Liu, Y. Han, Z. Yang, H. Wu, Efficient Data Query in Intermittently-Connected Mobile Ad Hoc Social Networks, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 26, No. 5, pp. 1301-1312, May, 2015.

[17] Y. Liu, H. Wu, Y. Xia, Y. Wang, F. Li, P. Yang, Optimal Online Data Dissemination for Resource Constrained Mobile Opportunistic Networks, *IEEE Transactions on Vehicular Technology*, Vol. 66, No. 6, pp. 5301-5315, June, 2017.

[18] Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, H. Zhang, Incentive Mechanism for Computation Offloading Using Edge Computing: A Stackelberg Game Approach, *Computer Networks*, Vol. 129, No. 2, pp. 399-409, December, 2017.

[19] M. Youssef, A. Youssef, M. Younis, Overlapping Multihop Clustering for Wireless Sensor Networks, *IEEE transactions on parallel and distributed systems*, Vol. 20, No. 12, pp. 1844-1856, February, 2009.

Biographies

Faheem Khan is currently working CS as an Assistant Professor at the Department of Computer Science, Bacha Khan University Charsadda, Pakistan.

Abdul Wahid Khan did his PhD from University of Malakand, KPK, Pakistan. He is working as Assistant Professor at Deptt: of CS, University of Science & Technology Bannu, KPK, Pakistan.

Khalid Shah did his PhD in Biomedical Imaging at University of East Anglia (UEA). He is working as Assistant Professor at Deptt: of CS, University of Science & Technology Bannu, KPK, Pakistan.

Iqbal Qasim did Computer Eng: at Hanyang University, South Korea. He is working as Assistant Professor at Deptt: of CS, University of Science & Tech: Bannu, KPK, Pakistan.

Asad Habib earned his Doctor of Engineering degree from the Graduate School of Information Science, NAIST (Nara Institute of Science and Technology) Japan. At present, he is the Director of the IIT, Kohat University of Science and Technology, Pakistan.