Performance Evaluation of Routing Protocols for Opportunistic Networks: An Energy Efficiency Perspective

Rodney S. Tholanah[1*], Zelalem S. Shibeshi[1], and Khuram Khalid[2]
[1]Rhodes University, Makhanda, South Africa
g18t4035@campus.ru.ac.za, Z.Shibeshi@ru.ac.za
[2]Ryersone University,Toronto, Ontario, Canada
khuram.khalid@ryerson.ca

Abstract

In most computer networks, a path between a sending node and the destination node is required to transfer data. There are instances where such a path for communication may not exist, such as in space communications, and the standard message routing techniques will not work. Opportunistic Networks were developed to solve this problem. Infrastructure-less Opportunistic Networks (OppNets) are a type of Delay Tolerant Networks where nodes in the network are responsible for forwarding messages to the destination nodes under the constraints of intermittent connectivity, dynamic topology changes, and non-guarantee of an end-to-end path. Various routing protocols were developed for OppNets; however, these routing protocols are energy-inefficient, and this is a problem because devices that are used in OppNets are usually battery-powered. In this project, new energy-efficient routing protocols for OppNets, namely, E_FirstContactRouter, E_WaveRouter, E_FloatingContentRouter and E_LifeRouter were developed from their energy-inefficient base routing protocols, namely FirstContactRouter, WaveRouter, FloatingContentRouter and LifeRouter. The energy-efficient routing protocols were compared against their energy-inefficient base protocols and other developed energy-efficient routing protocols, namely E-Prophet, E-Epidemic, E-MaxProp and E-Spray&Wait routing protocols. The energy metrics that we used to do the tests were the number of dead nodes and the average remaining energy. Simulations were carried out using the Opportunistic Network Environment (ONE) simulator and it was discovered that the proposed energy-aware routing protocols outperform their non-energy-aware counterparts in terms of the above-mentioned performance metrics.

Keywords: ONE simulator, opportunistic networks, energy-efficiency, first contact, wave router, floating content, life router

1 Introduction

In computer networks, routing protocols facilitate communication between nodes on the network by determining the path the messages will follow on their way to the destination from the sender. In most computer networks, the routing protocols rely on a complete path between the sending node and the destination node. However, the existence of such a path might not be guaranteed in certain types of networks. Such networks are used in areas such as space communications and mobile sensor networks. According to Qin et al. [12], there is a type of Delay Tolerant Networks called an Opportunistic Network in which a path between the sending node and receiving node is not always available. In such networks, the traditional routing protocols are not suitable because the network's topology is always changing [12]. This is because some nodes may run out of battery. Another reason is that the nodes are mobile, and

Research Briefs on Information & Communication Technology Evolution (ReBICTE), Vol. 7, Article No. 6 (September 30, 2021)
DOI:*10.22667/ReBiCTE.2021.09.30.006*

*Corresponding author: Rhodes University, Makhanda, South Africa

sometimes they might be too far from other nodes to communicate. New routing protocols have been developed which solve this problem. In Opportunistic Network routing protocols, all nodes accept messages, store the messages in their buffers, carry the messages and send the messages to other nodes when they get the opportunity [6]. The routing protocol being used determines whether a message should be forwarded to a particular node. This process can lead to energy inefficiency, [6] and it is a significant problem since the nodes are battery-powered.

In this paper, the first objective was to develop new energy-efficient routing protocols through the modification of energy-inefficient base routing protocols. The base routing protocols that were used were First Contact Router, Wave Router, Floating ContentRouter and Life Router. The energy-efficient versions of these protocols that were developed were E_FirstContactRouter, E_WaveRouter, E_FloatingContentRouter and E_LifeRouter. The modifications were based on two strategies, namely, to avoid forwarding messages to nodes whose energy level is below a certain threshold and to stop unnecessary forwarding of delivered messages in the network. The newly developed energy-efficient routing protocols' energy performance was then evaluated using the Opportunistic Network Environment (ONE) simulator against that of their energy-inefficient counterparts, namely: First Contact Router, Wave Router, Floating Content Router and Life Router.

The second objective was to compare the new routing protocols' energy performance against the energy performance of energy-efficient routing protocols that were already developed by other researchers, namely: E Prophet, E Epidemic, E MaxProp and E SprayWait. The energy performance metrics that were used were number of dead nodes and average remaining energy.

The paper is structured in the following way: Section 2 gives the background and related work, Section 3 provides a description of the design of the newly developed energy-efficient routing protocols, Section 4 gives the performance evaluation and Section 5 concludes the paper.

2 Background and Related Work

This section will provide a description of the base energy-inefficient routing protocols namely, First Contact, Wave Router, Floating Content and Life Router.

In First Contact Router, only one copy of a message exists in a network and nodes forward messages to the first nodes they encounter [4].

The Wave Router is designed to make messages circulate in a network [2]. To achieve this, it uses a tracking list and a value called the immunity time. Each node has a tracking list which stores the messages that have been received and each message will have an immunity time associated with it and it is the duration for which a node will keep a message before it deletes it [2]. If node X receives a message from another node, node Y, and node X's immunity time for that message is not over yet, node X will not accept the message.

Floating Content router uses nodes' geographical location in order to determine if messages messages are to be replicated or not. Each message includes two values namely: r, which is a radius that defines the area in which a message can be replicated and a, which is a radius that defines the area in which the message is kept by a node and it is called the anchor zone [2].

E_LifeRouter was developed through modifying Life Router. Life Router is based on Game of Life automaton. The router's behaviour is controlled by two integer values, m and n [2]. If a node is connected to k nodes, and k is such that $m \leq k \leq n$, the node will get a copy of the message if it did not have the message and it will keep it if it already had the message [2]. However, if k was such that $k > n$ or $k < m$, the node will not be forwarded to the node and it will be deleted from the buffer of the node with the copy [2].

The problem of high energy consumption by routing protocols for OppNets has resulted in new routing protocols being developed to solve the problem. A common strategy in the design and implementation of energy-efficient routing protocols for OppNets is the use of base routing protocols which are energy inefficient. These routing protocols are modified through the inclusion of energy-efficiency logic.

Borah et al. [3] developed the following energy-efficient routing protocols for OppNets, namely: E-Prophet, E-PRoWait and E-EDR.The E-Prophet routing protocol was developed by modifying the Prophet routing protocol, and E-Prophet has better energy performance than Prophet. In the E-Prophet routing protocol, a source node can obtain residual energy information of neighbouring nodes. The source node creates a Hashmap of neighbouring nodes that have higher remaining energy than its own [3]. The source node also then creates another Hashmap of nodes that have a higher probability of delivering the message and this probability is calculated using the usual Prophet rules [3]. The sending node only sends messages to nodes that appear in the second Hashmap.

The E-PRoWait routing protocol extends the ProWait routing protocol by enabling the sharing of residual energy information. Similar to the E-Prophet routing protocol, the source node in an OppNet using E-PRoWait obtains the residual energy information of neighbouring nodes and adds nodes whose residual energy is greater than its own into a Hashmap [3]. This means neighbouring nodes whose residual energy is less than that of the source node will not be considered in selecting nodes that will forward the message. The probability of delivering the message is then calculated for each node in the Hashmap using PRoWait routing protocol rules, and the suitable nodes are added into another Hashmap [3]. The message is then forwarded to all nodes in the second Hashmap [3].

The E-EDR routing protocol is an energy-efficient version of the EDR protocol [3]. The EDR routing protocol (Encounter and Distance Based Routing Protocol) was developed by Dhurandher et al. [8], and it uses a forward parameter to determine the next hop selection . The forward parameter is calculated using the value of encounters and the distance of neighbour nodes to the destination node [8]. The E-EDR routing protocol works similarly to that of E-Prophet and E-PRoWait. After the first Hashmap is created in the E-EDR routing protocol, those nodes with a likelihood of delivering the message equal to or higher than a set threshold are added to the second Hashmap. The source node then forwards the message to nodes in the second Hashmap [3].

Lu et al. [11] proposed an energy-efficient version of the epidemic routing protocol called n-epidemic routing protocol. In the epidemic routing protocol, the source node sends messages indiscriminately to all the node it meets. This strategy results in nodes losing energy quickly. The n-epidemic solves this problem by limiting the number of copies of message a node can send. A node will only forward the message when it has at least n neighbours [11]. However, a challenge faced when using n-epidemic routing protocol is determining the value n; if n is too large, the probability of the message being forwarded becomes very low, but if the value of n is too low, the probability of the message being forwarded becomes too high resulting in excessive use of energy resources [11]. In addition to being more energy-efficient, the n-epidemic routing protocol also had a higher delivery rate than the epidemic routing protocol. Unlike all of the energy-efficient routing protocols mentioned so far, the n-epidemic routing protocol does not consider the residual energy of nodes

Similarly, Khalid et al. [7] also created an energy-efficient version of the HBPR routing protocol called energy-efficient HBPR (AEHBPR). In the AEHBR routing protocol, an acknowledgement message is sent by the destination node to the node which forwarded the message to it. This message is forwarded to other nodes using a mechanism called one-hop acknowledgement, and they delete the message from their buffer storage [7]. A source node will not forward a message to a node whose energy level is less than a threshold set by the network administrator, and a node will not perform the utility metric calculation when it wants to forward the message to the destination node [7]. We decided to use a similar approach because HBPR is similar to the base protocols we used. Another reason was that the the algorithms that were used in developing AEHBR routing protocol are easier to implement compared to the other algorithms discussed in this section.

Kavian et al. [9] proposed energy-aware algorithms for Delay Tolerant Networks (DTN) routing protocols namely: Threshold, RRE and ORRE algorithms. These strategies can be combined with PRoPHET, Epidemic and Spray and Wait routing protocol, making them energy-efficient. The three algorithms are described below:

- Threshold Algorithm: This algorithm is the simplest of the three. The user specifies an energy threshold as a percentage of the node's energy. If a node's energy becomes lower than the user-defined energy threshold, the node stops forwarding messages to other nodes and will only transmit messages to a base station [9]. This enables nodes to function for a longer duration but finding the optimal value for the energy threshold can be a difficult task [9].

- Remaining Required Energy (RRE) Algorithm: The RRE builds on top of the Threshold algorithm. However, instead of a user specifying the energy threshold, the algorithm determines it automatically [9]. The algorithm uses the energy needed by the node to transmit messages obtained from the last encounter with a base station and energy needed to run by the node to run its basic functions [9].

- Optimized Remaining Required Energy (ORRE) Algorithm: According to Kavian et al. [9], the RRE algorithm can overestimate or underestimate the energy resources available if the node movement is not uniform. The ORRE algorithm does not have this limitation. Unlike the RRE algorithm, the ORRE algorithm also depends on the number of messages stored in the buffer [9].

These algorithms solve the energy-inefficiency problem by stopping nodes whose energy level is less than the energy threshold level from forwarding messages which saves energy. We used a similar approach to develop the new routing protocols.

3 Design of the energy-efficient routing protocols

This section describes the design considerations for the minimum energy threshold and one hop acknowledgement mechanisms that were used in the project to make the base routing protocols energy efficient.

The design of the new energy-efficient routing algorithms were based on two techniques to improve energy efficiency, namely minimum energy threshold and one-hop (acknowledgement) mechanism.

3.1 Minimum Energy Threshold

As explained in Khalid et al. [7], the minimum energy threshold is a value that the network administrator sets and potential relay nodes whose energy level is below the minimum energy threshold will not be selected to relay the message. The minimum energy threshold value was determined by simulations, as explained in [7]. The newly developed energy-efficient routing protocols were run with the Opportunistic Network Environment (ONE) simulator [5] using different minimum energy threshold values. The average remaining energy was recorded, and graphs of the minimum energy threshold and the average remaining energy were created. The minimum energy threshold value that corresponded to the highest average remaining energy was then used as the minimum energy threshold. 600J was used as the minimum energy threshold because it corresponded to the highest average remaining energy for all the newly developed routing protocols.
Algorithm 1 shows how the message forwarding takes place, taking into consideration the minimum energy threshold. The algorithm was adapted from [7].

```
Explanation of notation used:

NN: Next Neighbor Node
CN: Current Node
DN: Destination Node
MET: Minimum Energy Threshold
M: Message
Ack Table: Acknowledgment Table

Algorithm 1:  Steps taken when forwarding messages based on
minimum energy threshold:

Step 1: Select the NN
Step 2: If NN is busy then go to Step 1
Step 3: Repeat for all messages in buffer of CN
           4a: if NN has M then goto Step 3
           4b: check Ack Table of NN for M
                   If Ack Table of NN has M then
                        Remove M from the buffer of CN
                        Update Ack Table of CN
                        go to Step 3 for next M
                   end if
           4c: If energy level of NN < MET and NN is not the DN
                 go to Step 3
           4d: If NN is DN then
                    forward M to NN
                  else follow FirstContactRouter to send M to NN
```

As can be seen in Algorithm 1, the “current node” checks the energy level of the “next neighbour node”. If the “neighbour node’s” energy level is below the minimum energy threshold and the “neighbour node” is not the destination node, the “current node” will not forward the message to the “neighbour node” and it will select another node. However, if the “neighbour node” is the “destination node”, the “current node” will forward the message to it regardless of the “neighbour node’s” energy level.

3.2 One-Hop acknowledgement

When a message has been delivered, its copies will still be in the buffers of some nodes, and the messages have to be deleted to save energy and buffer storage. The one-hop acknowledgement mechanism is the strategy that is used for deleting messages from nodes' buffers that took part in the forwarding of the messages. Each node will have an (Ack) table that will contain the information about the delivered messages, namely: Message ID, Source ID and Destination ID [7]. Algorithm 2 shows the steps taken by a destination node after it receives a message. The algorithm was adapted from [7].

```
Algorithm 2. Steps taken by destination node receiving
a message:

Step 1: Receive M from the last sender node (LSN)
Step 2: If DN of M is current CN
then
               send Ack M to LSN
               update Ack Table of CN with Ack M
               remove M from the buffer of CN
```

As can be seen in Algorithm 2, the "destination node" sends an acknowledgement message to the "last sender node", updates its acknowledgement table and removes the message from its buffer. When the "last sending node" receives the acknowledgement message from the "destination node", it updates its acknowledgement table and removes the message from its buffer. The following algorithm details the steps taken by the "last sending node". It is also adapted from [7].

```
Algorithm 3: Steps taken by the last sender after receiving the
acknowledgement:

Step1: Receive message M from the DN
Step2: If M contains Ack M then
                update Ack Table of CN with Ack M
                remove M from the buffer of CN
```

As can be seen from Algorithm 3, the "last sending node" will then contact other nodes and compare its acknowledgement table with their acknowledgement tables. If the particular node still has the delivered message in its buffer, its acknowledgement table is updated, and the message is deleted from its buffer. This process is repeated until all copies of the messages are removed from the nodes in the network [7].

4 Implementation

This section describes how the new energy-efficient routing protocols were developed, namely, E_FirstContactRouter, E_WaveRouter, E_FloatingContentRouter and E_LifeRouter.

The ONE simulator is Java-based; therefore, the Java programming language was used to implement the new energy-efficient routing protocols. The minimum energy threshold mechanism and one-hop acknowledgement mechanisms described in the Design section were incorporated into the new protocols. The *battery_level_threshold* variable was used when adding the minimum energy threshold mechanism, and it stored the minimum energy threshold value. The *delivered* HashMap was used to keep track of delivered messages in the one-hop acknowledgement mechanism. Both mechanisms were added to all the new energy-efficient routing protocols except for E_FirstContactRouter. This is because E_FirstContactRouter does not replicate messages; therefore, the one-hop acknowledgement mechanism could not be added, so E_FirstContactRouter only uses the minimum energy threshold mechanism. Various methods were also modified, including the *update*, *receiveMessage* and *tryOtherMessages* methods. These methods are utilised by the routing protocols when nodes scan for other nodes and forward messages to the nodes they contact.

5 Simulation results and Discussion

This section describes how the simulations were carried out and it gives a summary of the results in the form of graphs. The simulations were performed using ONE simulator using the Shortest Path Map Based Movement Model (SPMBM).

5.1 Simulation settings

In ONE simulator, the values for parameters like number of nodes, message size, mobility model etc are specified in default settings file. Table 1 shows the values that were used in the simulations.

Table 1: Table showing the ONE simulator settings parameter values

Parameters	**Values**
Simulation Area	4500 m * 3400 m
Movement Model	Shortest Path Map Based Movement model and Real-Trace mobility model
Buffer Size	5 M
Communication Interface	Bluetooth
Routing Protocols	E-Prophet, E-S&W, E-MaxProp, E-Epidemic, E-FirstContactRouter, E-WaveRouter,E-FloatingContentRouter and E-FloatingCounterRouter
Simulation Time	43200 seconds
Message Time To Live	300 minutes
Initial Energy of All Nodes	5000 Joules
Scan Energy	0.1 Joules
Scan Response Energy	0.1 Joules
Transmit Energy	0.2 Joules

The energy performance metrics that were used were average remaining energy and number of dead nodes. Average remaining energy is the average remaining energy of the nodes and number of dead nodes is the number of nodes whose remaining energy is below the minimum energy threshold value.

6 Simulation Results Using Shortest Path Map Based Movement Model

When the Shortest Path Map Based Movement Model (SPMBM) is used, the nodes choose a random point on the map and move to that point following the shortest path [4]. The nodes move like this multiple times for the duration of the simulation.

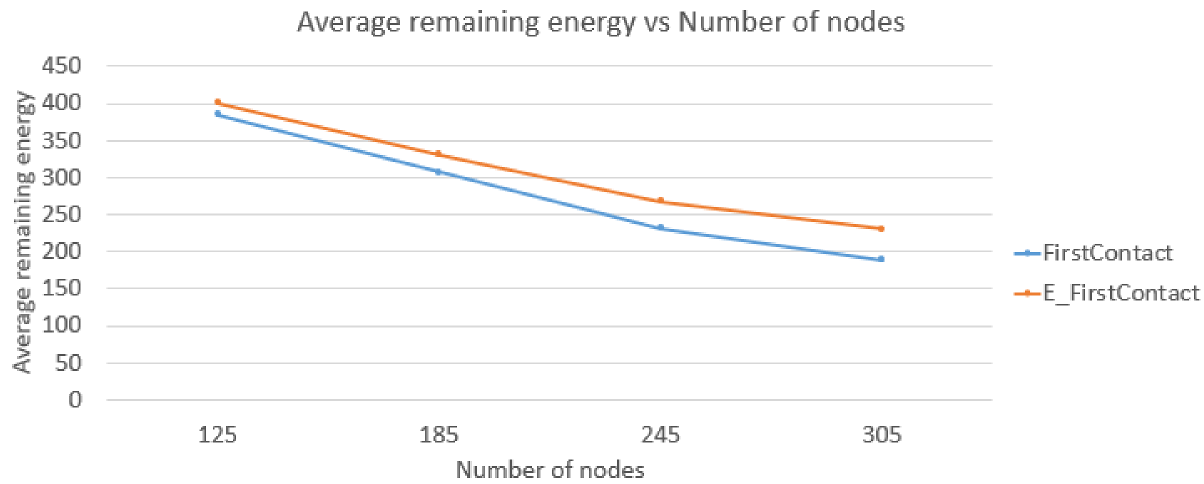


Figure 1: Graph showing the relationship between average remaining energy and number of nodes for FirstContactRouter and E_FirstContactRouter using SPMBM

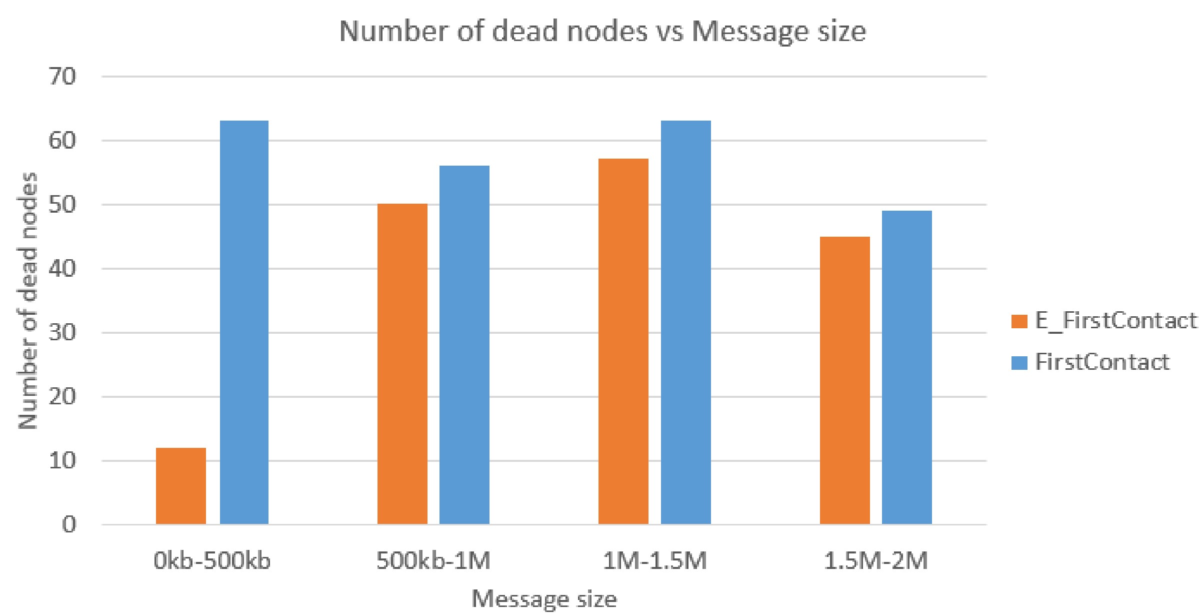


Figure 2: Graph showing the relationship between the number of dead nodes and message sizes for FirstContactRouter and E_FirstContactRouter using SPMBM

The graph for the average remaining energy for E_FirstContactRouter shows higher values compared to that of the FirstContactRouter. It can be observed in Figure 1 that the average remaining energy decreases as the number of nodes is increased. This is because as the number of nodes increase, more messages are sent which results in nodes performing more scans and forwarding more messages which makes the nodes' energy decrease more. Figure 2 shows how the number of dead nodes varies with messages size. It can be seen that for all message ranges, E_FirstContactRouter has fewer number of dead nodes compared to FirstContactRouter which indicates that E_FirstContactRouter is more energy efficient. FirstContactRouter had less average remaining energy and higher number of dead nodes compared to E_FirstContactRouter because as mentioned in [4], when FirstContactRouter is used, nodes forward messages to the first nodes they encounter without taking into consideration the energy levels of the potential forwarding nodes. This results in nodes forwarding messages even if their batteries are about to run out and ultimately leads to high number of dead nodes and low average remaining energy. On the other hand, when E_FirstContactRouter is used, nodes will not take part in the forwarding of messages if their energy levels are less than a certain threshold thereby saving energy which results in high average remaining energy and low number of dead nodes.

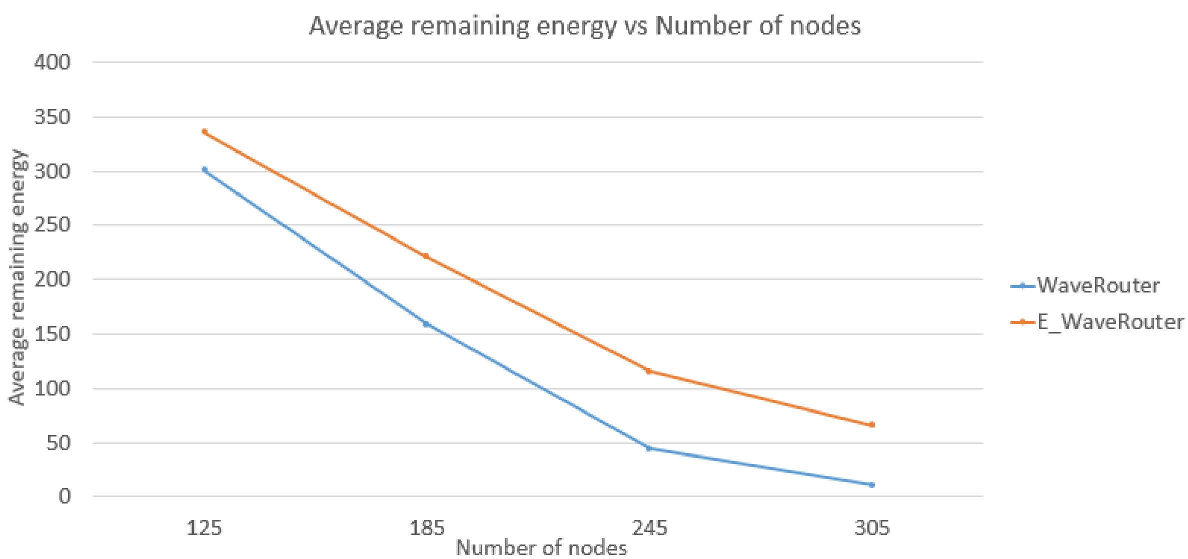


Figure 3: Graph showing the relationship between average remaining energy and number of nodes for WaveRouter and E_WaveRouter using SPMBM

Figure 3 shows that average remaining energy is inversely proportional to the number of nodes in the network. This is because as the number of nodes increase, more messages are sent which results in nodes performing more scans and forwarding more messages which makes the nodes' energy decrease more. However, still E_WaveRouter has higher average remaining energy compared to WaveRouter.

Figure 4 shows how the number of dead nodes varies with messages size. It can be seen that for all message ranges, E_WaveRouter has fewer number of dead nodes compared to WaveRouter which indicates that E_WaveRouter is more energy efficient. WaveRouter had less average remaining energy and a higher number of dead nodes compared to E_WaveRouter because as discussed in [2], when WaveRouter is used, immunity time for a particular message is used in order to determine if a message will be forwarded. On the other hand, E_WaveRouter uses both immunity time and the minimum energy threshold mechanism which results in energy being saved because a potential forwarding node's energy level is taken into consideration before a message is forwarded to it. In addition, unlike WaveRouter,

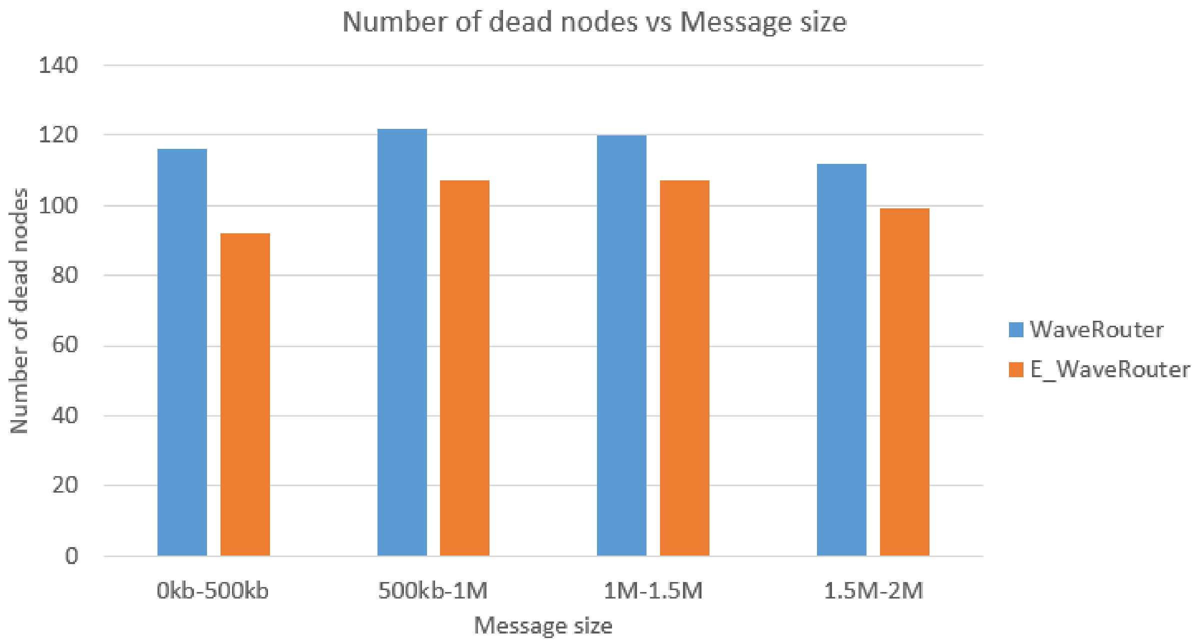


Figure 4: Graph showing the relationship between the number of dead nodes and message sizes for WaveRouter and E_WaveRouter using SPMBM

E_WaveRouter uses the one-hop (acknowledgement) mechanism to remove delivered message copies from the buffers of nodes that took part in forwarding the messages thereby saving energy by preventing unnecessary forwarding of delivered messages.

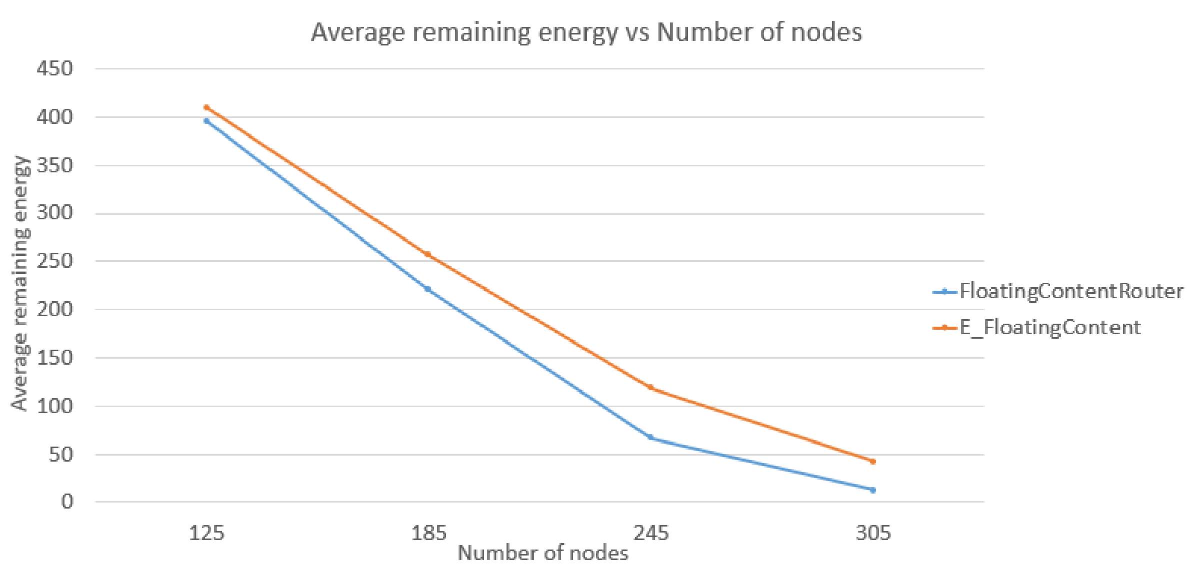


Figure 5: Graph showing the relationship between average remaining energy and number of nodes for FloatingContentRouter and E_FloatingContentRouter using SPMBM

Figure 5 shows how the average remaining energy varies with messages size. It can be seen that for all number of nodes, E_FloatingContentRouter has higher remaining energy compared to FloatingContentRouter which indicates that E_FloatingContentRouter is more energy efficient due to the energy-efficiency logic.

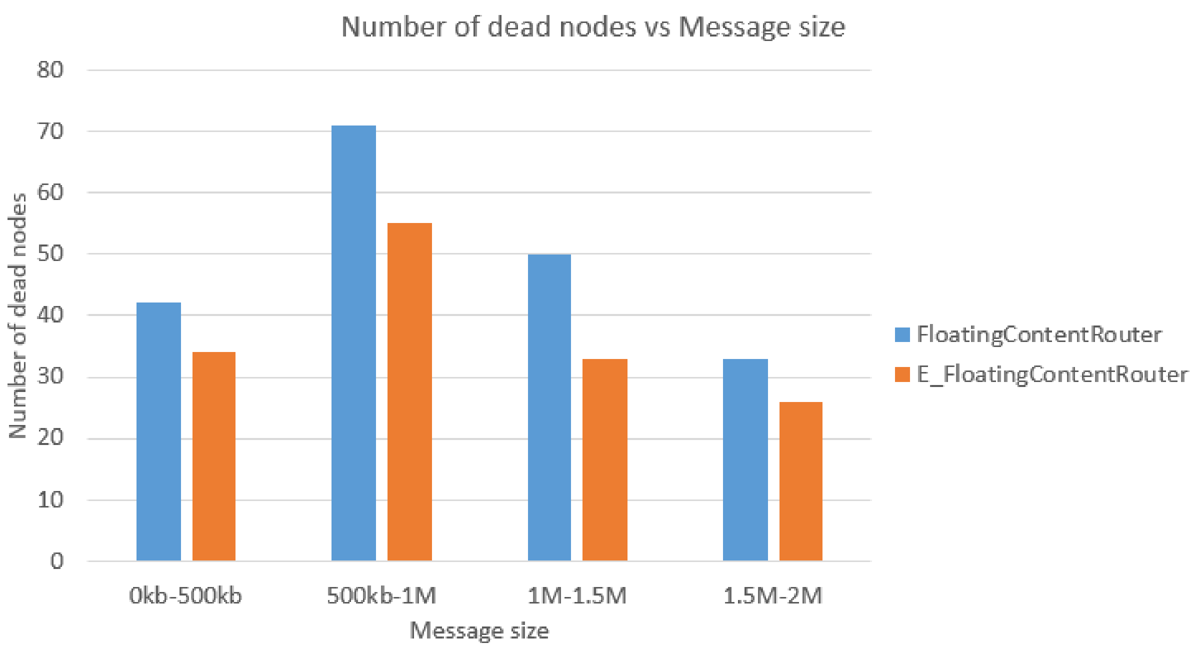


Figure 6: Graph showing the relationship between the number of dead nodes and message sizes for FloatingContentRouter and E_FloatingContentRouter using SPMBM

Figure 6 shows how the number of dead nodes varies with messages size. It can be seen that for all message ranges, E_FloatingContentRouter has fewer number of dead nodes compared to FloatingContentRouter. FloatingContentRouter had less average remaining energy and had a higher number of dead nodes because as mentioned in [2], when FloatingContentRouter is used, messages are replicated according to geographic location. This can result in energy being wasted especially if nodes get into areas where many messages are being replicated. However, when E_FloatingContentRouter is used, the energy level of potential forwarding nodes is considered through the use of minimum energy threshold thereby saving energy.

It can be observed in Figure 7 that both E_LifeRouter and LifeRouter's average remaining energy is inversely proportional to the number of nodes. However, E_LifeRouter's average remaining energy energy is higher compared to that of E_LifeRouter due to the energy efficiency logic that was added.

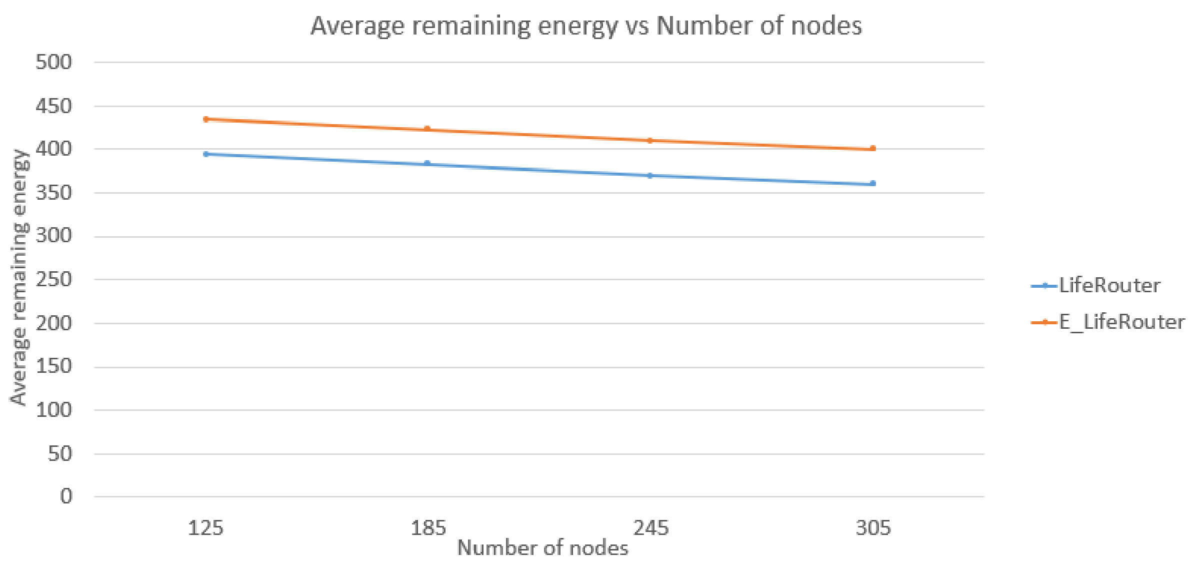


Figure 7: Graph showing the relationship between average remaining energy and number of nodes for LifeRouter and E_LifeRouter using SPMBM

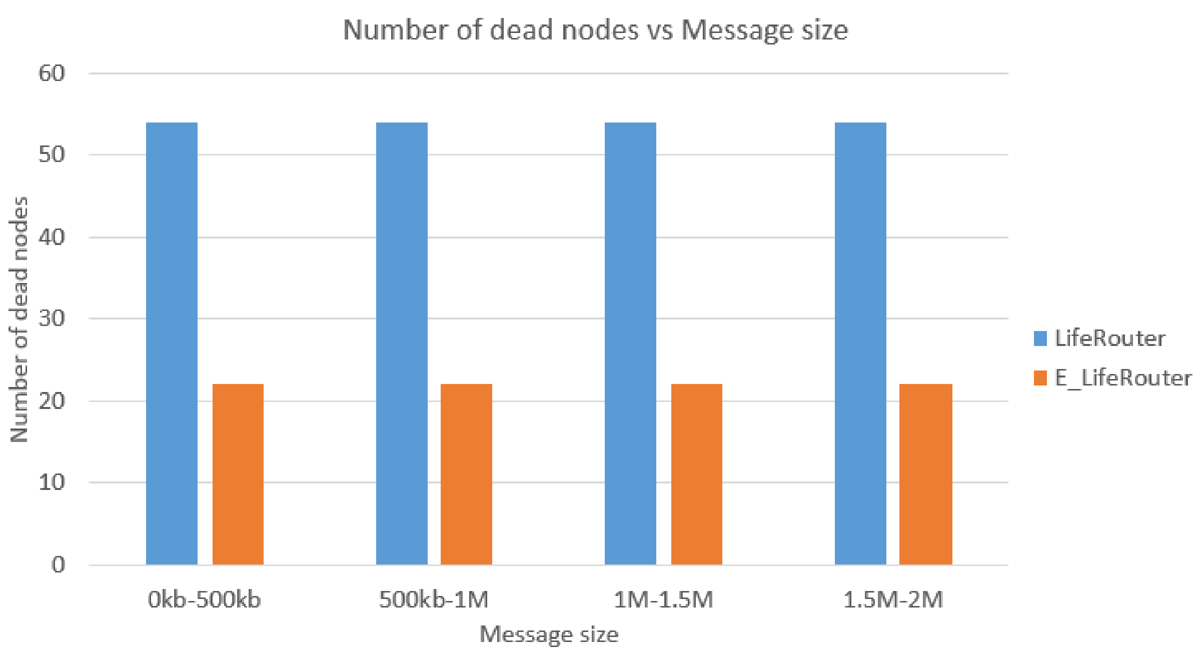


Figure 8: Graph showing the relationship between the number of dead nodes and message sizes for LifeRouter and E_LifeRouter using SPMBM

Figure 8. shows the variation of the number of dead nodes as message sizes are increased for both E_LifeRouter and LifeRouter. The addition of energy efficiency routing logic resulted in significantly less number of dead nodes for E_LifeRouter and the number of dead nodes was constant for both LifeRouter and E_LifeRouter which means the there is no strong relationship between message size and number

of dead nodes when either LifeRouter or E_LifeRouter is used. LifeRouter had less average remaining energy and had a higher number of dead nodes compared to E_LifeRouter because according to [2], LifeRouter uses Game of life atomation rules in determining if a message is to replicated or not and it does not take nodes' energy values into consideration. However, when E_LifeRouter is used, the energy levels of potential forwarding nodes is checked thereby improving energy efficiency.

6.1 Comparative Analysis of the Proposed Routing Protocols and Previously Created Energy-Efficient Routing Protocols

This section describes the analysis that was done when comparing the newly created energy-efficient routing protocols and those that were previously created.

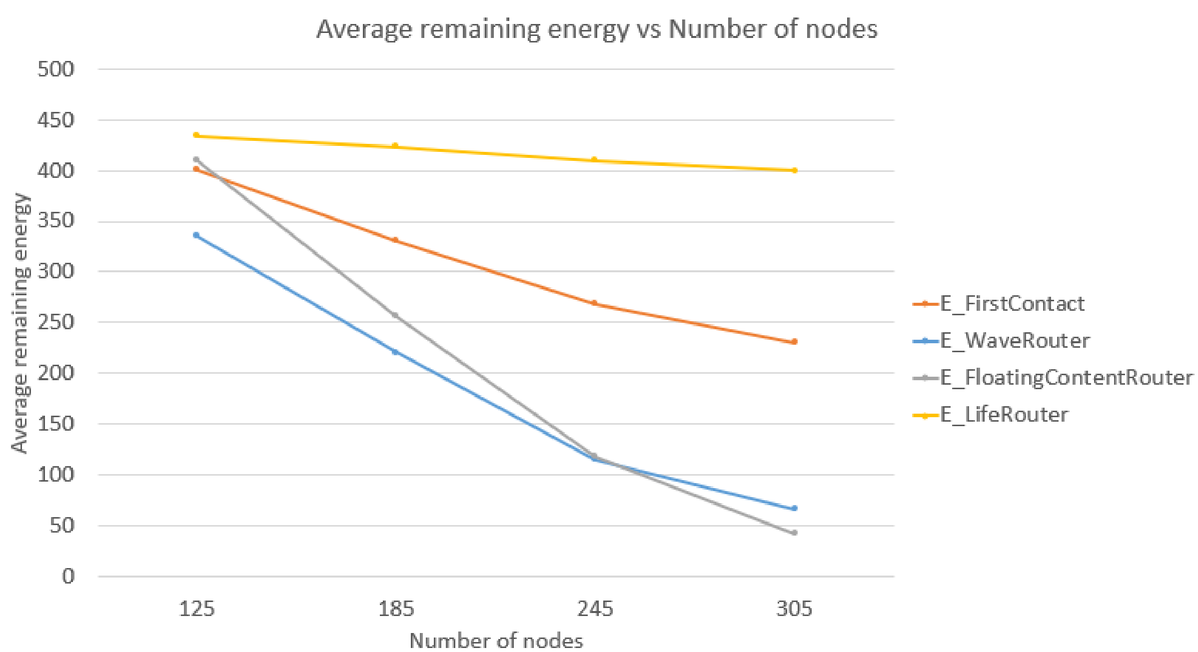


Figure 9: Graph showing the relationship between average remaining energy and number of nodes for the energy-efficient routing protocols using SPMBM

As can be seen in Figure 9, E_LifeRouter had the best energy performance for all number of nodes. This is because LifeRouter uses a parameter called *nmcount* [2] which is composed of 2 values, *n* and *m*. As described in [2], messages are forwarded or deleted depending on the number of nodes that already have the message. This behaviour results in fewer messages being forwarded which saves energy.
E_FloatingContentRouter had the lowest average remaining energy for all number of nodes except 305 nodes. This is because as described in [2], Floating Content Router forwards messages to nodes if they are within a certain geographic area which results in the messages multiple times resulting in energy loss. E_FirstContactRouter performed better than both E_FloatingContentRouter and E_WaveRouter because when E_FirstContactRouter is used, nodes do not continuously share the same messages unlike when E_FloatingContent and E_WaveRouter are used.

Figure 10 shows the energy performance in terms of average remaining energy for the routing protocols that were already developed. It can be observed that E_Epidemic and E_S&W had the highest

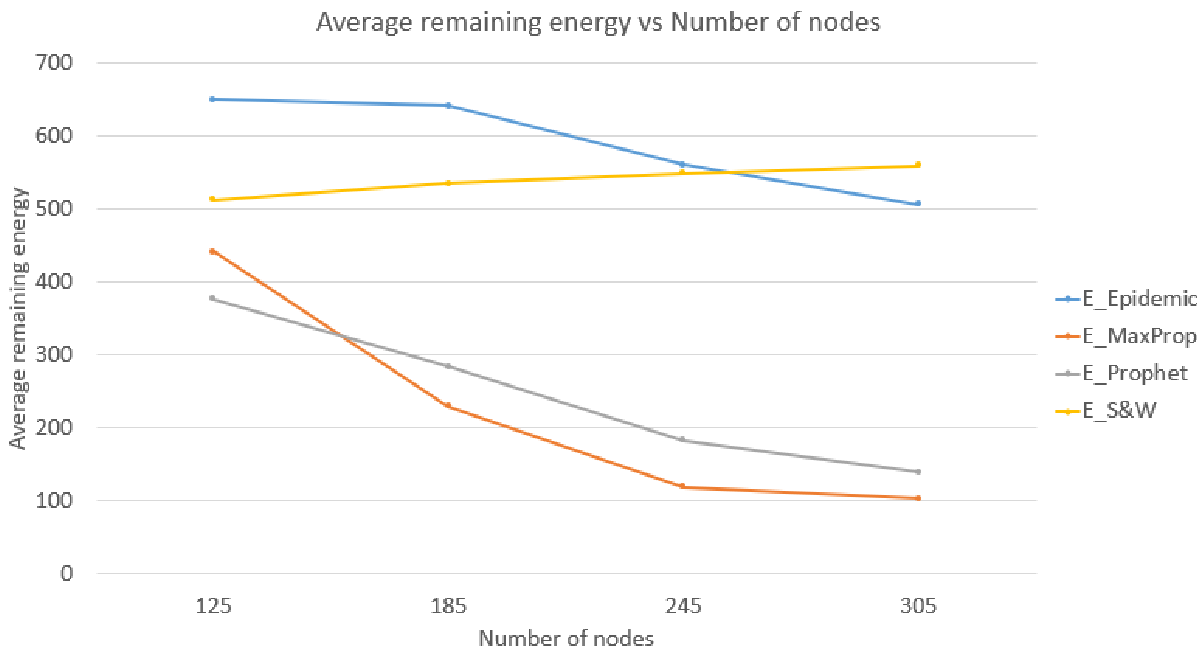


Figure 10: Graph showing the relationship between average remaining energy and number of nodes for the previously created energy-efficient routing protocols using SPMBM

average remaining energy compared to the other energy-efficient routing protocols. This is because the E_Epidemic routing protocol uses hop counts which help to limit the number of times messages can be replicated thereby saving energy [1]. An interesting observation is that the average remaining energy for E_S&W seemed to increase as the number of nodes increased. This might be because E_S&W sends a limited number of message copies to nodes which will be responsible for delivering the messages to the destination nodes [10], which helps in saving energy.

Figure 11 is a graph showing the number of dead nodes and message sizes for the energy efficient routing protocols. It can be observed that E_WaveRouter has the highest number of dead nodes for all message sizes. This is because as described in [2], WaveRouter circulates messages in the network which results in more dead nodes as more energy is used. E_FloatingContentRouter does not have dead nodes due to reasons described in the last section.

Figure 12 is a graph that shows the number of dead nodes and message sizes for the already developed energy-efficient routing protocols. E_S&W performed very well for message sizes below 1M but does not perform as well for message sizes greater than 1M. Among all the energy-efficient routing protocols, E_LifeRouter had the lowest number of dead nodes. This is to be expected as the protocol's average remaining energy was high for all number of nodes and only decreased slightly as the number of nodes increased.

Overall, E_Epidemic, E_S&W and E_LifeRouter had the best performance as the number of nodes was increased. E_LifeRouter, E_FloatingContent and E_MaxProp had better performance compared to the other routing protocols as the message sizes were increased.

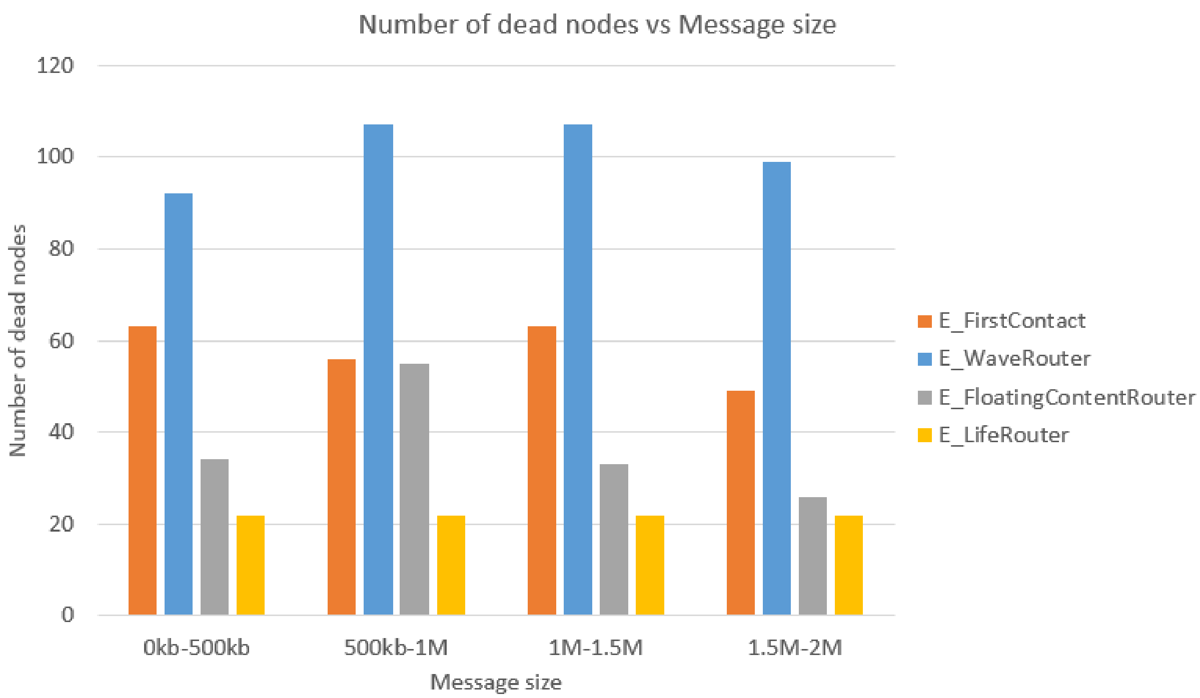


Figure 11: Graph showing the relationship between the number of dead nodes and message sizes for the energy-efficient routing protocols using SPMBM

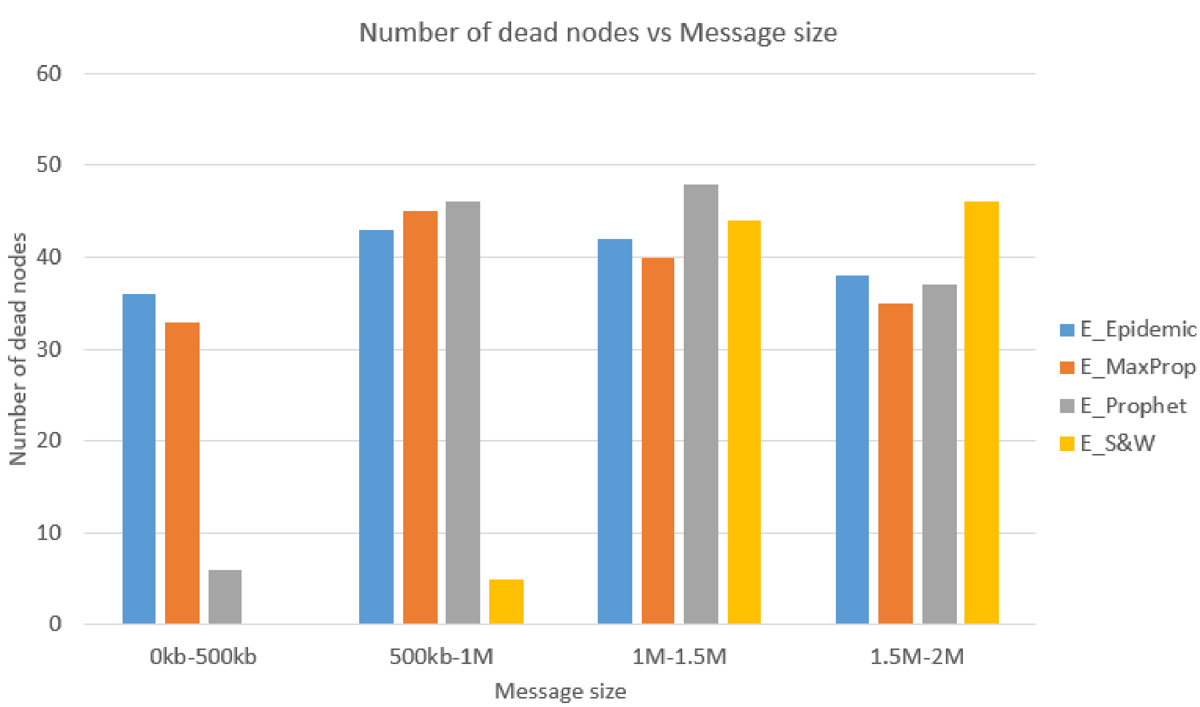


Figure 12: Graph showing the relationship between the number of dead nodes and message sizes for the previously created energy-efficient routing protocols using SPMBM

7 Conclusion

The first objective of this project was to develop new energy-efficient OppNet routing protocols that have better energy performance than their energy-inefficient counterparts. This was accomplished through the addition of the minimum energy threshold and one-hop acknowledgement mechanisms to the base routing protocols. Both mechanisms were added to all the new energy-efficient routing protocols except for E_FirstContactRouter. This is because E_FirstContactRouter does not replicate messages; therefore, the one-hop acknowledgement mechanism could not be added, so E_FirstContactRouter only uses the minimum energy threshold mechanism. The second objective was to compare the new routing protocols' energy performance. The third objective was compare the new routing protocols' energy performance against the energy performance of other energy-efficient routing protocols that were already developed. All the objectives were satisfied. The Simulation Results section shows that the newly developed energy-efficient routing protocols namely, E_FirstContactRouter, E_WaveRouter, E_FloatingContactRouter and E_LifeRouter have better energy performance than their base routing protocols. The new routing protocols have a higher average remaining energy and lower number of dead nodes compared to the energy-inefficient base routing protocols. The Simulation Results section also shows the comparisons between the new energy-efficient routing protocols and also comparisons between the new routing protocols and previously developed energy-efficient routing protocols.

For future work, simulations need to be done using real-trace mobility models in order to determine if the results that would be obtained would the same as those obtained when using the synthetic mobility model. The mechanisms that were used in this project i.e. minimum energy threshold and one-hop acknowledgement can also be applied to other routing protocols such as ProWait routing protocol,Rapid routing protocol etc in order to develop energy-efficient versions of these routing protocols. Another issue that needs to be investigated is the impact the addition of the energy-efficiency logic has on message delivery. The change in the average remaining energy for the routing protocols when more than 245 nodes are being used should be investigated. The decrease in the number of dead nodes for messages of sizes between 1.5M-2M for the routing protocols should also be investigated.

Acknowledgments

This work was undertaken in the Distributed Multimedia CoE at Rhodes University, with financial support from Telkom SA and CORIANT. The authors acknowledge that opinions, findings and conclusions or recommendations expressed here are those of the author(s) and that none of the above mentioned sponsors accept liability whatsoever in this regard.

References

[1] V. Amin and B. David. Epidemic Routing for Partially Connected ad hoc Networks. Technical Report CS-200006, Duke University, 2000.

[2] M. S. Desta, E. Hyytiä, A. Keränen, T. Kärkkäinen, and J. Ott. Evaluating (Geo) Content Sharing With the ONE Simulator. In *Pro. of the 11th ACM international symposium on Mobility management and Wireless access (MobiWac'13), Barcelona, Spain*, pages 37–40. ACM, November 2013.

[3] B. S. J., D. S. K., T. Suryansh, W. Isaac, and O. M. S. Energy-efficient Prophet-prowait-edr Protocols for Opportunistic Networks. In *Proc. of the 2017 IEEE Global Communications Conference (GLOBECOM'17), Singapore*, pages 1–6. IEEE, December 2017.

[4] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *Proc. of the 2nd International Conference on Simulation Tools and Techniques (Simutools'09), Rome, Italy*, pages 1–10. ACM, March 2009.

[5] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE simulator for DTN protocol evaluation. In *Pro. of the the 2nd Inter-national Conference on Simulation Tools and Techniques (Simutools'09), Rome, Italy*, pages 55:1–55:10. ACM, March 2009.

[6] K. Khalid, I. Woungang, S. K. Dhurandher, J. Singh, and J. J. P. C. Rodrigues. Energy-efficient Check-and-spray Geocast Routing Protocol for Opportunistic Networks. *Information*, 11(11):504, October 2020.

[7] K. Khalid, I. Woungang, D. S. K, B. Leonard, C. G. HS, and T. Makoto. An Energy-efficient Routing Protocol for Infrastructure-less Opportunistic Networks. In *Proc. of the 10th international conference on innovative mobile and internet services in ubiquitous computing (IMIS'16), Fukuoka, Japan*, pages 237–244. IEEE, July 2016.

[8] D. S. Kumar, B. Satya, W. Isaac, S. D. Kumar, A. Kunal, and A. Divyansh. Edr: An Encounter and Distance Based Routing Protocol for Opportunistic Networks. In *Proc. of the 30th IEEE International Conference on Advanced Information Networking and Applications (AINA'16), Crans-Montana, Switzerland*, pages 297–302. IEEE, March 2016.

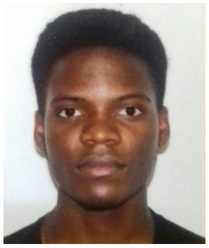
[9] K. Mahzad, K. Branislav, J. Raja, B. Neil, and L. Vicky. Energy-aware Forwarding Strategies for Delay Tolerant Network Routing Protocols. *Journal of Sensor and Actuator Networks*, 5(4):18, 2016.

[10] S. Thrasyvoulos, P. Konstantinos, and R. C. S. Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. In *Proc. of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (SIGCOMM'05), Philadelphia, Pennsylvania, USA*, pages 252–259. ACM, August 2005.

[11] L. Xiaofeng and H. Pan. An Energy-efficient n-Epidemic Routing Protocol for Delay Tolerant Networks. In *Proc. of the 5th IEEE International Conference on Networking, Architecture, and Storage (NAS'10), Macau, China*, pages 341–347. IEEE, July 2010.

[12] Q. Xueyang, W. Xiaoming, W. Liang, L. Yaguang, and W. Xinyan. An Efficient Probabilistic Routing Scheme Based on Game Theory in Opportunistic Networks. *Computer Networks*, 149:144–153, February 2019.

Author Biography



Rodney S. Tholanah is studying for his B.S. honours degree at Rhodes University. His research interests are Networking and Routing Protocols.

Zelalem S. Shibeshi holds a PhD from Rhodes University, a BSc in Physics, Associate Degree in Computer Science, and MSc in Information Science, all from Addis Ababa University, Ethiopia. He is a Senior Lecturer at Rhodes University. His research interests are Multimedia Service Development and Real Time Communication, APIs for Telco Services, Machine Learning (especially NLP), Information Retrieval (IR), and IoT in Agriculture.

Khuram Khalid received MSc and PhD degrees in Computer Science from Ryerson University. He is a Research Assistant at Ryerson University.His research interests are Networking and Routing Protocols