DOI 10.12694/scpe.v20i1.1479
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. 27–40. http://www.scpe.org

ESTABLISHING RELIABILITY FOR EFFICIENT ROUTING IN OPPORTUNISTIC NETWORKS

DEEPAK KUMAR SHARMA* AND DEEPIKA KUKREJA†

Abstract. Opportunistic network (Oppnet) is a class of networks where connections between the nodes are not permanent. The nodes are continuously moving and some nodes even switch off their batteries to conserve energy. Reliable delivery of messages in Opportunistic network is one major inherent issue. It is unreliable in the sense that once the source node has forwarded its message, then it will never get to know about its status in the network like whether the message has got discarded at an intermediate node or at the destination node (due to buffer overflow) or the successful delivery of the message has taken place. This work tries to make Oppnet as much reliable as possible. It proposes a reliability protocol named as Reliability in Oppnet (RIO). RIO improves the routing in Oppnet and works in parallel with the existing routing protocols. It makes the source node aware about the status of message so that if an error occurs then the source node can take suitable action to resend the message. It considers the redirection error, buffer overflow error, Time Limit Exceeded (TLE), parameter problem and destination unreachable errors that may occur inside the network. RIO has been tested using ONE simulator and implemented with Spray and Wait routing protocol. Results show that the RIO with Spray and Wait protocol outperforms normal Spray and Wait protocol in terms of average message delivery probability.

Key words: Routing Protocol, Reliability, Opportunistic networks, Average message delivery, Network errors, ONE simulator

AMS subject classifications. 68M12

1. Introduction.

Opportunistic network [1] as the name signifies, is the type of network that is based on the opportunities of contacts that exist in the network. The connections between the nodes are created and terminated on demand i.e. based on the availability of suitable nearby node that can take the message in the vicinity of destination node. The nodes try to utilize the best opportunity to connect to a node that will take the message in close proximity of intended destination node. Opportunistic networks inherit its characteristic features from two super classes that are Delay tolerant networks (DTN) [2, 3] and Mobile Ad hoc Networks (MANETs) [4, 5, 6, 7]. Oppnet inherits the property of discontinuous network connections from DTN. The connections between the nodes in Oppnets are not permanent like TCP/IP model, so these networks are prone to long unpredictable delay in packet transmission [8]. Oppnet acquires the property of nodes from MANETs as the nodes are mobile in these types of networks. Hence it can be said that Oppnet is a kind of network that is somewhat like a mixture of MANET and DTN i.e. is a network with no continuous node connections with nodes being mobile.

Routing in Opportunistic network is a challenging task as nodes connections are not permanent. A message can be delivered instantly (if the connections that leads to successful delivery of message are available at that time) or the message can take hours or even days before it gets successfully delivered to the intended destination. Traditional routing protocol does not solve the problem of routing of message in Oppnet. Routing protocols in Oppnet are broadly classified in two categories; these are infrastructure-based routing protocols and infrastructure less routing protocols. Infrastructure based routing protocols use infrastructure in some form like message ferries, info stations and cloud computing. It is obvious that these types of routing protocols require expensive technologies and equipment. HVF Scheme [9], Message ferry scheme [10], Cloud Computing based routing protocol (CCBRP) [11] and Infostation Model [12], are some examples of infrastructure base routing protocols. Infrastructure less routing protocols are those in which no infrastructure is required to support routing in Oppnet. In these type of routing protocols congestion control, node buffer storage, node battery [13, 14], and traffic in the network are main issues of concern in routing. Many protocols have been developed to tackle these problems for example epidemic routing protocol [15], PROPHET [16] routing, Spray and Wait routing protocol [17] to name a few. Protocols that handles security issues [18, 19], machine learning techniques [20, 21], and other miscellaneous routing related works [22, 23, 24] has also been designed for Oppnets recently.

*Division of Information Technology, Netaji Subhas University of Technology (Formerly Netaji Subhas Institute of Technology), New Delhi, India (dk.sharma1982@yahoo.com), corresponding author.

†Division of Information Technology, Netaji Subhas University of Technology (Formerly Netaji Subhas Institute of Technology), New Delhi, India (deepikakukreja18@gmail.com)

All the previously designed protocols are unreliable in the sense that once the source node sends message in the Oppnet, it assumes that the successful transfer will take place to the destination. But this is not always true in many cases. Consider a scenario in which the forwarded message gets dropped at the intermediate node as it was unable to store message in its buffer due to full buffer capacity (buffer overflow). Consider one another scenario where the forwarded message reaches some intermediate node and this intermediate node finds that Time to Live (TTL) has been expired, and then this node simply drops the message. Message can even be discarded at destination node. For example, consider a situation where a destination node cannot accept the message due to full buffer capacity at that time. In order to resolve unreliabilities like these, a reliability protocol named as RIO has been proposed. The proposed protocol works in parallel with the existing Oppnet routing protocols and hence enhance routing in Oppnet.

The errors that may occur in the Oppnet are classified into five categories: redirection error, buffer overflow error, parameter problem, Time Limit Exceeded i.e. TTL expiration error and destination unreachable. In this work, solutions are provided for the above-mentioned errors, and through simulation using Opportunistic Network Environment simulator i.e. ONE simulator [25] it has been found that RIO enhances routing of existing protocols while working in parallel with them. For testing purpose, RIO is combined with Spray and Wait protocol. Through simulations it has been proved that Spray and Wait routing gets enhanced when reliability protocol RIO works in parallel with it.

The paper is organized as follows. Section 2 discusses the related work done in past. Section 3 explains the proposed work and demonstrates the working of the RIO protocol in detail. Simulation results and observations are given in section 4. Finally, Section 5 concludes this work and talk about future work.

2. Background and Related Work. This section is dedicated to explain briefly some existing infrastructure less routing protocols with their merits and demerits.

PROPHET PROPHET [16] protocol is a probability-based routing where nodes forward messages to other nodes based on a probabilistic metric. This metric is estimated based on history of encounters. Whenever nodes interact with each other to exchange messages, they also exchange node encounter history with each other. In this way, this protocol is transitive in nature i.e. if N1 and N2 nodes share information with each other and N2 node and N3 node share information with each other, then it is equivalent to sharing information between N1 node and N3 node. In this protocol, a node forwards the message to that neighboring node that has highest probability metric.

EPIDEMIC Epidemic routing [8] is based on the flooding technique. Every node in the network floods the message to its neighboring nodes which then flood this message to their neighbors. Each node has two buffers, one for its own messages and second for storing the messages received from other nodes. Every message in the system is flagged with a unique ID. On meeting, two nodes swap their summary vectors and exchange those messages which are not present in their buffers. This procedure is followed at each pair of nodes and eventually every node in the network gets the message copy. This Protocol creates lot of message copies which make it robust against network failure and result in lesser delay, but at the same time creates network congestion.

HBPR HBPR [26] stands for History Based Routing Protocol and is based on behavior and stability of nodes. This protocol tries to deliver message using a path which is most visited by the destination node. In this way the intermediate node selects those nodes as next hop that tries to deliver message to the path that is most visited by the destination and thus harness the behavior of nodes in the network. HBPR has high average message delivery probability as compared to tradition Oppnet protocol like epidemic. The average latency and overhead ratio are also less in HBPR as compared to epidemic routing protocol.

EDR EDR [27] is an Encounter and Distance based protocol that forwards the message based on a metric known as forward parameter. Forward parameter is calculated based on encounter history of nodes with destination node and on the distance of nodes from destination node. This metric is calculated for each neighboring node and the message is given to that node which has the metric value above a threshold.

Spray and Wait Spray and Wait protocol [17] is a controlled flooding-based routing protocol. It limits the number of messages forwarded in the network unlike epidemic routing protocol where the nodes forward message to every node it encounters. This routing protocol works in two phases the first phase is the

Spray phase and the other is Wait phase. In the Spray phase, the source node forwards the message to L nodes. In the Wait phase, if the message is not yet delivered to destination node then it performs a direct delivery of message. Spray and Wait reduces the traffic and the number of messages dropped in the network as compared to epidemic routing protocol and the delivery probability using Spray and Wait protocol is also high due to controlled flooding.

GAER GAER [28] is Genetic Algorithm based Efficient Routing protocol. This is an energy efficient protocol as the energy spent in routing of the messages is lowest as compared to other available routing protocols. This protocol uses personal information of nodes and applies genetic operations and algorithm which are used to decide the next hop. It calculates a fitness function based on some parameters and then forwards the message to a node whose fitness value is above the cut off threshold value.

Spray and Focus Spray and Focus [29] is mobility assisted routing protocol. It has two phases; one is the Spray phase and the other is Focus phase. A node forwards message to L relay nodes in Spray phase, and in Focus phase the message can be forwarded to different relay nodes based on some forwarding criterion. In Focus phase, the transmission is not direct unlike Spray and Wait protocol. Spray and Focus is much more efficient than available mobility assisted routing protocol and it outperforms them in terms of average message delivery probability and average latency.

3. Proposed Work. In Oppnets, nodes contact with each other opportunistically that is no stable or permanent path exist between the sender and receiver node. Hence, Oppnets are highly unreliable in terms of message delivery. Once the source node forwards message, it will never be able to know whether its message successfully reaches to the destination or not. The message might get lost in the network or it might be discarded by some intermediate node due to the following reasons.

- Intermediate nodes buffer overflow that is, there might be a case where the buffer queue of the intermediate node might be full.
- Ambiguity in the addresses that are incorporated in the header of the message.
- Their might be a case where the Time to Live (TTL) of the message might become zero when it reaches an intermediate node.

There can be numerous cases of message not getting successfully delivered to the intended destination. So, now the question arises that how to cope up with this unreliability that exist in the network? How to devise a way so that the sender of the message must be able to track the status of its forwarded messages and if there occurs some error in delivery of messages, then the sender node can act accordingly.

The solution to these problems is to ensure reliability in routing of messages in Opportunistic networks. Reliability can be ensured through error reporting method. Whenever an error occurs during routing of messages, an observing node (which has noticed the error) sends an error reporting message to the source node. After receiving these error reporting messages, the source node can enquire for the cause of failure and can resends the respective message or takes some other suitable action.

3.1. Routing Errors. Errors in routing can be broadly specified into following five types.

3.1.1. Redirection Error. There might be a case where the message forwarded by the source node travels in wrong direction and it gets continuously forwarded in the same wrong direction by the intermediate nodes like in case of MoVe(Motion Vector)[30] routing protocol which forwards the message to its nearby node that is closest to the node. In MoVe protocol, there might occur a case where the nodes continuously forward the message in the wrong direction due to their characteristic property of forwarding message to the nodes that have shortest distance to it. In First Contact routing protocol nodes forward messages randomly based on the available contacts at a particular instant of time. In this protocol their might occur a case where the nodes continuously forward the message in the wrong direction because the node that is forwarding the message always finds a connection to a node that is in the wrong direction. To handle such situations in message routing and to make routing reliable, the intermediate node that notices such wrong redirection of messages sends a redirection error message to the source node and simultaneously tries to forward the message in the right direction which is towards the intended destination. Figure 3.1 shows a scenario of redirection error. In the figure, the source node forwards the message in wrong direction and the intermediate nodes also forward the message in the same wrong direction. After passing through several intermediate nodes, one of the intermediate nodes notices this

redirection error and tries to forward the message in the direction of destination node and reports the same to the source node.

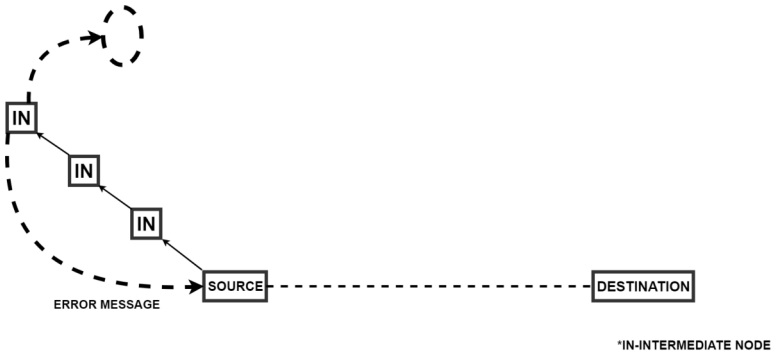

FIG. 3.1. *Redirection Error Reporting*

3.1.2. Buffer overflow. While routing the messages from source node to destination node, there is very high probability of encountering an intermediate node whose buffer is full. In this case, the intermediate node drops or discards the message without bothering about it much. Epidemic routing protocol follows flooding-based approach for message passing which means that a node forwards multiple copies of messages in the network. In this, the buffer occupancy time of messages is very high which leads to high chances of getting a relay node in routing path which has full buffer or overflowed buffer. This same reasoning follows for Spray and Wait, PROPHET and MaxProp [31] routing protocols. In order to deal with such situations, using RIO, the node that notices such a buffer overflow problem sends a buffer overflow error message to the source node which has generated the message. This error message is sent to the source node before the source node discards the message. Figure 3.2 depicts a scenario of buffer overflow problem. In this, the source nodes forward a message which passes through numerous intermediate nodes and finally arrives at destination node, after arriving at destination node, the message gets discarded because the destination node notices that its buffer is full and there is no room to store the incoming message, now the destination node sends an error message to the source node.

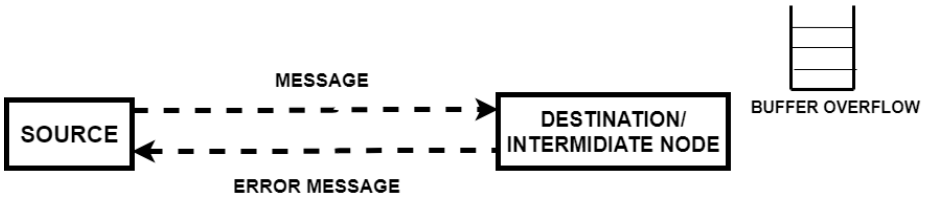

FIG. 3.2. *Buffer Overflow Error Reporting*

3.1.3. Time Limit Exceeded. There might occur a case while routing of the message from source to destination, that the Time to Live (TTL) of the message has got expired that is the value of TTL becomes zero. At that time, the receiving node of the message simply discards the message and the source node of the message never be able to know that whether the message gets successfully delivered or not. RIO protocol is implemented to deal with such situations, using RIO, the node that notices such phenomenon sends a TLE message to the source node. The TLE message is sent to the source node before the source node discards that message, so that it can resend the message in order to get its message successfully delivered to its intended destination. Figure 3.3 shows a scenario of Time Limit Exceeded. In the figure, the source node forwards a message and the message passes through numerous nodes and finally its TTL expires. Now the node at which TTL expires sends an error message to the source node.

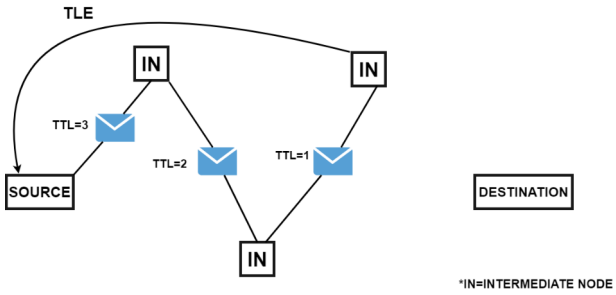

FIG. 3.3. *TLE Error Reporting*

3.1.4. Parameter Problem. Their might occur a case where the receiving node of a particular message notices that the fields in the header of the message are ambiguous. For example, the receiving node might notice that the source address or the destination address is ambiguous. It might happen that the destination address is set to all zeros or all ones. So, to handle situations like these, using RIO protocol, the receiver of the message sends a parameter problem error message to the source node before the source node discards it. The source node resends the message after rectifying. Figure 3.4 depicts a scenario of parameter problem. In the figure, the source node forwards the message and after passing through several nodes, one of the node notices that the destination field in the header of the message is ambiguous. It notifies the source node by sending a parameter problem error message to it.

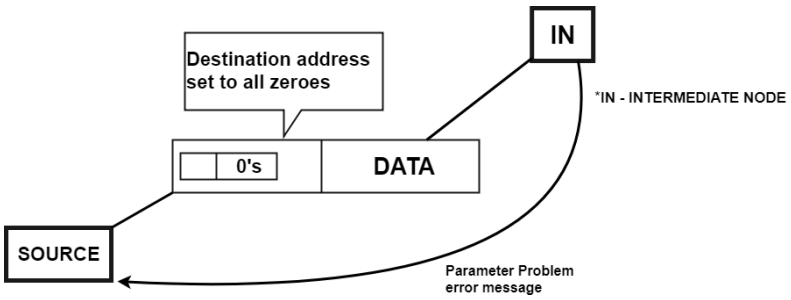

FIG. 3.4. *Parameter Problem Error Reporting*

3.1.5. Destination Unreachable. In Oppnets, nodes sometimes switch off or turn off their batteries in order to conserve the energy. So, in Oppnet there might occur a case where the destination node power is not on and the message that was intended for it might get lost in the network because the node for which it was destined was invisible to the network. To cope up with situations like this, using RIO protocol, the node that is currently holding the message defers the routing of the message until the node for which it is destined turns on its power. If the destination node doesnt turn on its power until a certain threshold value of time duration then the node (i.e. currently holding the message) sends a destination unreachable error message to the source node. Figure 3.5 depicts a destination unreachable scenario where the source node forwards the message and this message passes through nodes IN-1, IN-2 and finally IN-3. IN-3 node notices that the destination is not in the network (i.e. its battery is off) so it defers sending the message until some threshold time (assuming twice of remaining TTL). If until then, the destination node does not switch on its power then a destination unreachable error message is sent to the source node.

3.2. Working of RIO. RIO is a dependent reliability protocol. It improves the performance of an existing routing protocol, if reliability features that are established in RIO are allowed to work in parallel with other existing infrastructure less routing protocol. In RIO, every node first checks if any incoming message is an error message. If an incoming message is an error message then it matches the destination address (i.e. embedded in the error message) of error message with its own address. If match occurs then it suspects whether the error

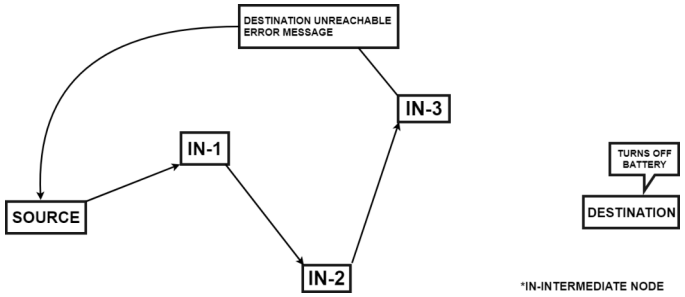

FIG. 3.5. *Destination Unreachable*

message is redirection error message or not. If it is not redirection error message then it resends the message or take some suitable action. If the address does not match then it routes this error message in the Oppnet because this error message is not intended for this node. If the incoming message is not an error message then it checks for the errors that were specified in the previous section i.e. the node checks for redirection error, buffer overflow error, Time Limit Exceeded, Parameter problem and destination unreachable error.

In the following paragraphs, strategies are discussed to deal with these errors. All these errors are checked at a node when the node receives any incoming message. For example, a node when receives a message checks whether its buffer is full or not, whether TTL of message is greater than zero or not, likewise it checks for other errors.

The redirection error is checked by calculating the angle between the two imaginary lines that are formed by joining the coordinates of the sender node to the coordinates of the current node and the coordinates of the sender node to the coordinates of the destination node. The first line is between the coordinates of the sender of the message and the current node. The second line is between the coordinates of destination node of the message and the sender node of the message (i.e. the node from which the current node has received the message). If these two lines are making an angle greater than ninety degrees then there is a redirection error and the current node tries all connections which result in angle less than ninety degrees between the above two specified lines and simultaneously sends a redirection error message to the source of this message. The two possible scenarios have been shown in figure 3.6 and figure 3.7. In figure 3.6 i.e. scenario 1, the intermediate node notices that the angle formed between the two lines that are specified above is greater than ninety degrees, so it sends a redirection error message to the source node. In figure 3.7 i.e. scenario 2 the angle formed between the lines specified above is less than ninety degrees, so it continues with normal routing. For finding the coordinates of the nodes, Global positioning system (GPS) is used, which is now-a-days pre-installed on every mobile and laptops. GPS does not require any other technologies like internet to operate, it operates independently. Hence this technology is handy to find out the coordinates of the nodes.

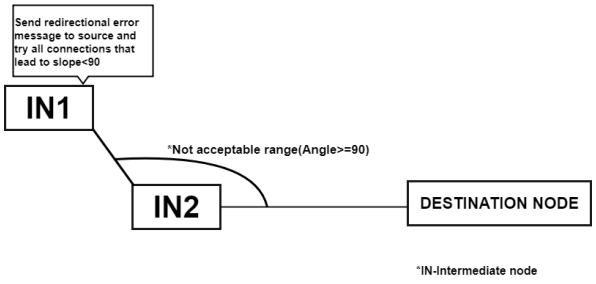

FIG. 3.6. *Redirection error (scenario 1)*

The buffer overflow problem at a node is detected by checking the current free buffer space. If the free buffer space is less than or equal to zero then the buffer overflow error reporting message is sent to the source node of incoming message before this node discards the message.

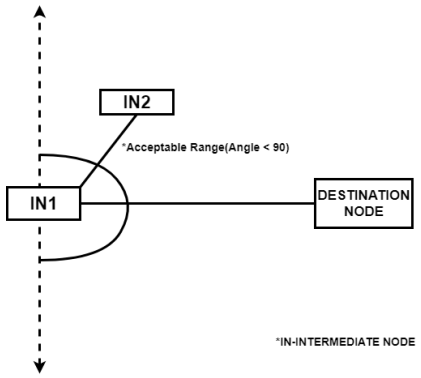

FIG. 3.7. *Redirection error (scenario 2)*

The TLE error at a node is detected by checking the values of TTL for all the incoming messages, if the TTL is found to be zero then the TLE error reporting message is sent to the source node.

The parameter problem error is detected by checking the validity of destination address specified in the incoming message. If the destination address that is embedded in the incoming message is found to be invalid then a parameter problem error message is sent to source of incoming message before discarding it.

The Destination unreachable error is detected by checking the visibility of destination node address through GPS. The node holds the message until the destination node gets visible and keeps the message in its buffer until some threshold is reached. If the destination node is not visible until a certain threshold (twice of remaining TTL) then a destination unreachable error message is sent to the source node.

4. Simulation Results. Opportunistic Network Environment (ONE) [25] simulator has been used to evaluate the performance of RIO protocol. ONE is an open source Oppnet simulator. It is built with java programming language. It comes with several built-in modules that facilitate routing in Oppnet. ONE generates reports based on the settings that have been made to run the simulation. There are infinite number of different scenarios that can be run in ONE simulator. This characteristic makes it one of the most popular simulators for simulating Opportunistic network environment protocols.

4.1. Simulation Setup. Table 4.1 shows the simulator parameters that have been chosen for the set up to simulate RIO protocol. The pseudo code that has been presented in the above section has been implemented in update method of the routing class of ONE simulator. To test the performance of RIO with existing routing protocol, Spray and Wait routing protocol is used with RIO. RIO has been simulated by varying following metrics while keeping the other parameters constant.

1. Variation in the nodes: The nodes in the network are changed from 40 to 240 by increasing 40 nodes at each step and keeping all others parameters constant.
2. Variation in TTL of nodes: The message TTL has been changed to different values from 100 to 300 with an increment of 50 while keeping all others parameters constant.
3. Variation in message generation interval: This interval has been varied (while keeping other parameters constant) that changes the messages generated in the network, for example if this interval is 0 to 5 then the message will be generated at each node at a timestamp that ranges from 0 to 5.

The following metrics are used to evaluate the performance of RIO:

1. Average Message Delivery probability: It is the probability of successfully delivered messages.
2. Average Hop Count: It denotes the average number of hops or nodes that a message travels to reach its intended destination.
3. Average Message Delay: It denotes the average time taken from message creation t its delivery.

4.2. Results and Observations. In this section, the results and observations after performing the simulation are presented.

TABLE 4.1
Simulation Parameters

Parameter	Simulation Value
Area of simulation	4500 x 3400 sq. m
Transmission Rate	250 Kbps
Groups of nodes	6
Transmission range	10 m
Movement model	Shortest Path Map Based Movement
Each nodes Storage Space	5 Mb
Simulation time	43200 s
Speed Range	1-14m/s
Range of Wait Time	0-120 s
Message size	500 Kb to 1Mb
Generation time of Messages	25-35 s

4.2.1. Average Message Delivery Probability. This section compares the results of average message delivery probability for Spray and Wait (SAW) protocol and SAW with RIO protocol by varying the parameters like number of network nodes, TTL and message generation interval. Figure 4.1 depicts the average message

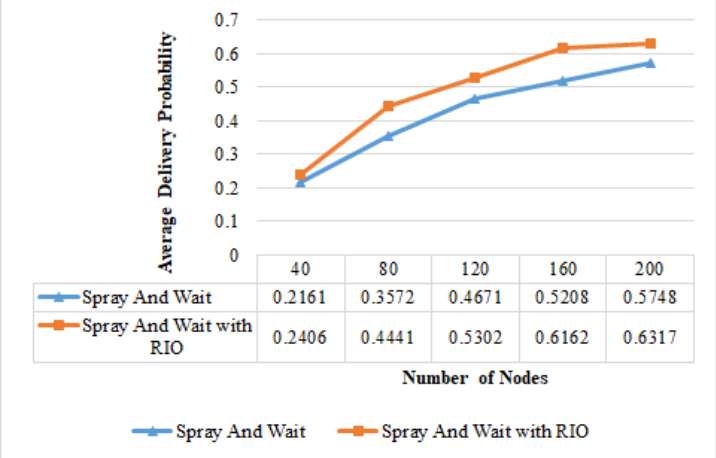

FIG. 4.1. *Average Message Delivery Probability v/s Number of Nodes*

delivery probability v/s the number of nodes. As the number of nodes increases the average message delivery probability increases in Spray and Wait because the network becomes dense and messages easily find nodes that can deliver them to destination node. The probability is further increased in Spray and Wait with RIO because now some of the messages which were not delivered to destination in Spray and Wait are retransmitted by the source node after receiving the error message. The cause of failure is specified within the error message. In figure 4.1, the average message delivery probability increases by 15.29% in Spray and Wait with RIO as compared to Spray and Wait. Hence, on an average the delivery probability of messages increases in RIO with Spray and Wait as compared to Spray and Wait. It can be justified by the fact that the messages which previously get discarded by the intermediate nodes (due to various reasons like buffer overflow, TTL expire, parameter problem etc.) now get successfully delivered to their intended destination because of the reliability imposed. If the message gets discarded, an error reporting message is sent to source node which leads to retransmission of message and thus resulting in more number of successfully delivered messages.

Figure 4.2 depicts average message delivery probability v/s TTL of the messages. It has been observed that RIO with Spray and Wait has better message delivery probability as compared to Spray and Wait. As the TTL

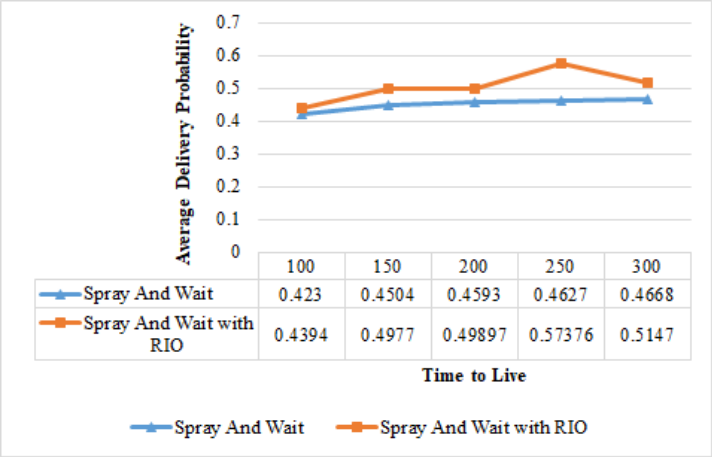

FIG. 4.2. *Average Message Delivery Probability v/s TTL of the messages*

increases, the time that a message can search for their respective destination increases. If some messages get discarded or by some means do not reach to their destination, now have more probability for successful delivery because of the error reporting mechanism of RIO. Using RIO, the average message delivery probability of Spray and Wait protocol has increased by 11.59%.

Figure 4.3 depicts average message delivery probability v/s message generation interval. As this interval

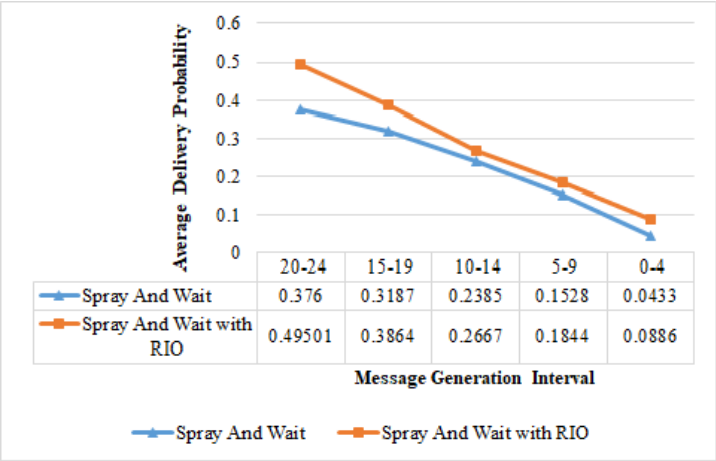

FIG. 4.3. *Average Message Delivery Probability v/s Message Generation Interval*

decreases, more number of messages get generated in the network, leading to increase in network traffic. Spray and Wait protocol is a controlled flooding-based technique which in turns add to the already increased traffic. Hence the number of messages dropped at intermediate nodes increases which results in decrease in average message delivery probability of Spray and Wait routing protocol. The average message delivery probability increases when RIO works alongside Spray and Wait because now some of the messages which were previously gets dropped are delivered to their respective destination because of the added reliability features of RIO. The average message delivery probability of SAW with RIO has increased by 25.83%.

4.2.2. Observing Average Hop Count. Implementing RIO with Spray and Wait, the average hop count increases as now the hop counts of the messages which were previously get discarded by the intermediate nodes now travel again from source node to destination node because of the error reporting mechanism used in RIO.

This increases the number of hops through which the message has been passed in order to reach its destination node. Figure 4.4 shows the average hop count v/s number of network nodes. The average hop count of Spray and Wait using RIO increases by 15.83% because now the discarded messages travel through more nodes as the network is dense, thus this will add to hop counts.

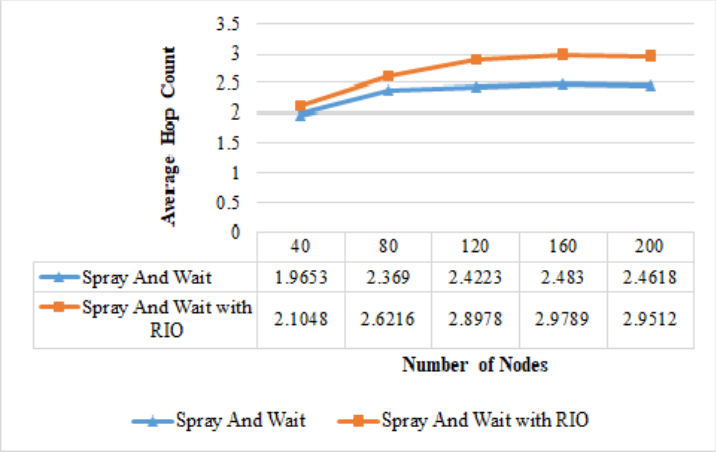

FIG. 4.4. *Average Hop Count v/s Number of Nodes*

Figure 4.5 depicts average hop count v/s Time to Live. In the figure the average hop count increases by 15.71%, which is justified by the fact that as the TTL increases the messages have more time to live in the network and if some messages get discarded by some intermediate nodes then an error message is sent to the source node, which leads to retransmission of the messages and thus increases the average hop count.

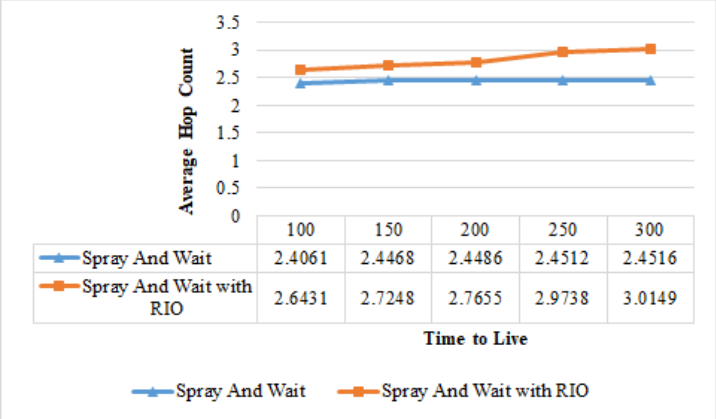

FIG. 4.5. *Average Hop Count v/s Time to Live*

Figure 4.6 shows the average hop count v/s message generation interval. The average hop count of Spray and Wait has increased by 19.45% using RIO which is justified by the fact that as the message generation interval decreases the number of messages generated increases, this increases the number of undelivered messages and some of these undelivered messages are retransmitted because of the reliability features of RIO and thus this further adds to the hop count which is evident from the figure.

4.2.3. Observing Average Latency. In Spray and Wait using RIO, on an average the latency increases as compared to Spray and Wait. This is justified by the fact that now the transmission time of sending the

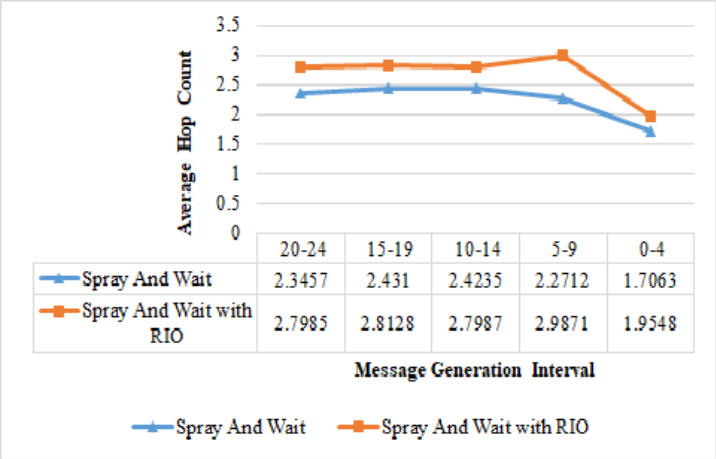

FIG. 4.6. *Average Hop Count v/s Message Generation Interval*

error reporting message back to source node is added to the total latency. And also, the time to retransmit the message (which was previously discarded) from the source to destination node has also been added to latency leading to higher latency as compared to Spray and Wait routing protocol. This is evident from figures 4.7, 4.8 and 4.9. Figure 4.7 depicts that the average latency of Spray and Wait increases by 8.02% with varying number of network nodes. Figure 4.8 shows that the average latency of Spray and Wait with RIO increases by 3% by

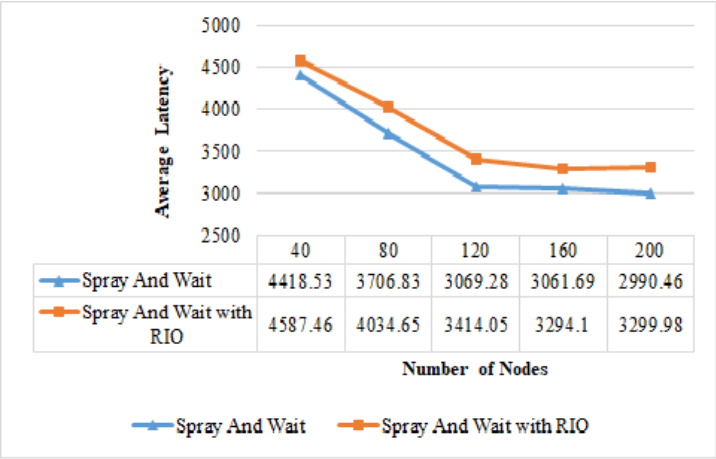

FIG. 4.7. *Average Latency v/s Number of Nodes*

increasing TTL. Figure 4.9 depicts that the average latency of Spray and Wait increases by 11.94% while using RIO v/s message generation interval.

5. Conclusion and Future Work. The work establishes the reliability in Opportunistic Network through the application of error reporting. In this, a novel reliability protocol, RIO has been proposed, implemented and tested. RIO works in parallel with the existing routing protocols and it enhances routing performance because of the added reliability features. The proposed work classifies errors that may happen in the Oppnet into five categories, Redirection error, Time Limit Exceeded (TLE) error, Buffer overflow error, Parameter Problem error and Destination unreachable error. RIO protocol has been implemented along with Spray and Wait routing protocol. It has been observed that RIO causes increase in average message delivery probability, average hop count and average delay of Spray and Wait protocol.

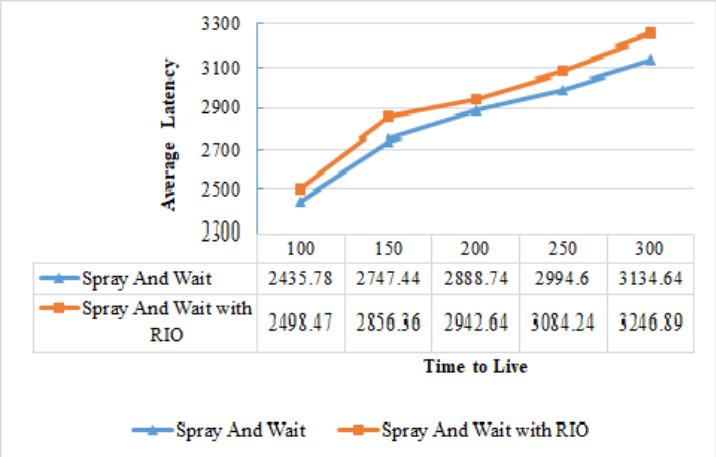

FIG. 4.8. *Average Latency v/s Time to Live*

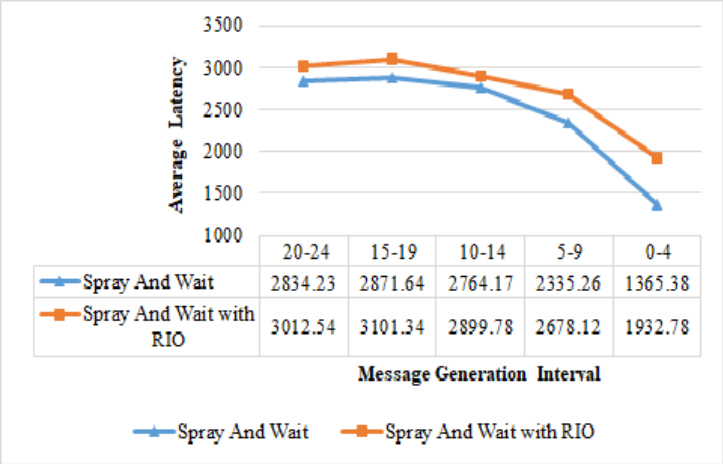

FIG. 4.9. *Average Latency v/s Message Generation Interval*

In future, we will try to establish reliability in Cloud Computing based routing protocol and hence can improve the efficiency of Oppnet so that it can be used for practical purposes.

REFERENCES

[1] L. LILIEN, Z. H. KAMAL, V. BHUSE, A. GUPTA A, *Opportunistic networks: the concept and research challenges in privacy and security*, In: Proceedings Of NSF intl. workshop on research challenges insecurity and privacy for mobile and wireless networks (WSPWN 2006), Miami, March 2006, pp. 134–147.

[2] K. FALL, *A delay-tolerant network architecture for challenged internets*, In: Proceedings of ACM SIGCOMM 2003, Karlsruhe, Germany, 2529 Aug 2003, pp. 27–34.

[3] S. JAIN, K. FALL, R. PATRA, *Routing in a delay tolerant network*, In: Proceedings of ACMSIGCOMM 2004, pp. 145–158.

[4] C. K. TOH, *Ad hoc mobile wireless networks: protocols and systems*, Prentice Hall PTR, Englewood Cliffs.

[5] D. KUKREJA, S. K. DHURANDHER AND B. V. R. REDDY, *Power aware malicious nodes detection for securing MANETs against packet forwarding misbehavior attack*, Journal of Ambient Intelligence and Humanized Computing, Springer, April 2017, doi:10.1007/s12652-017-0496-2, pp. 1–16.

[6] D. KUKREJA, S. K. DHURANDHER AND B. V. R. REDDY, *Enhancing the security of dynamic source routing protocol using energy aware and distributed trust mechanism in MANETs*, Intelligent Distributed Computing (Springer) Series, Volume 321, 2015, pp. 83–94.

[7] M. MIGLANI, D. KUKREJA, S. K. DHURANDHER, B. V. R. REDDY, *Power aware and secure dynamic source routing protocol*

in mobile ad hoc networks, Security in Computing and Communications, Communications in Computer and Information Science, Volume 467, 2014, pp. 45–56.

[8] S. K. Dhurandher, D. K. Sharma, I. Woungang, and H.C. Chao, *Performance evaluation of various routing protocols in opportunistic networks*, in Proceedings of IEEE GLOBECOM Workshop 2011, Houston, Texas, USA , 5-9 December, 2011, pp. 1067–1071.

[9] S. K. Dhurandher, D. K. Sharma, S. Gupta, I. Woungang, and M. S. Obaidat, *Integration of fixed and mobile infrastructure for message passing in opportunistic networks*, in proceedings of JOURNAL OF NETWORKS, vol. 10, No. 12, Acadmey Publisher, December 2015, pp. 642–657.

[10] W. Zhao, M. Ammar, and E. Zegura, *A message ferrying approach for data delivery in sparse mobile ad hoc networks*, in proceedings of 5th ACM Intl. Symp. Mobile Ad Hoc Networking and Computing 2004 (MobiHoc 04), ACM Press, Tokyo, Japan, 24-26 May 2004, pp. 187–198.

[11] D. K. Sharma, S. K. Dhurandher, A. Kumar, A. Kumar, and A. K. Jha, *Cloud computing based routing protocol for infrastructure-based opportunistic networks*, in proceedings of IEEE India International Conference on Information Processing (IICIP), D.T.U., Delhi, India, 12-14 August, 2016.

[12] D. J. Goodman, J. B., N. B. Mandayam and R. D. Yates, *INFOSTATIONS: A new system model for data and messaging services*, IEEE Vehicular Technology Conference 1997(VTC97), vol. 2, May 1997, pp. 969–973.

[13] A. Chhabra, V. Vashishth, and D. K. Sharma, *SEIR: A stackelberg game based approach for energy-aware and incentivized routing in selfish opportunistic networks*, in proceedings of 51st Annual Conference on Information Sciences and Systems (CISS), 2017, 22-24 March 2017, Baltimore, MD, USA, pp. 1–6.

[14] S. K. Dhurandher, D. K. Sharma, I. Woungang, and A. Saini, *An energy-efficient history based routing scheme for opportunistic networks*, International Journal of Communication Systems, Special Issue - Energy Efficient Networking, Volume 30, Issue 7, May 2017, Wiley, DOI: 10.1002/dac.2989.

[15] A. Vahdat, and D. Becker, *Epidemic routing for partially connected ad hoc networks*, Technical Report CS-2000-06, Dept. of Computer Science, Duke University, Durham, NC, 2000.

[16] A. Lindgren, A. Doria, and O. Schelen, *Probabilistic routing in intermittently connected networks*, ACM SIGMOBILE, Mobile Computing and Communications Review, vol. 7, Issue 3, July 2003, pp. 19–20.

[17] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, *Spray and wait: An efficient routing scheme for intermittently connected mobile networks*, in proceedings of ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN 05), Philadelphia, PA, USA, 22-26 Aug. 2005, pp. 252–259.

[18] D. K. Sharma, S. K. Dhurandher, I. Woungang, J. Arora, and H. Gupta, *History-based secure routing protocol to detect blackhole and greyhole attacks in opportunistic networks*, Journal of Recent Advances in Communications and Networking Technology, Vol. 5, No. 2, Bentham Science, November 2016, ISSN (Print): 2215-0811, ISSN (Online): 2215-082X, DOI: 10.2174/2215081106666161206124014, pp. 73–89.

[19] A. Chhabra, V. Vashishth, and D. K. Sharma, *A fuzzy logic and game theory based adaptive approach for securing opportunistic networks against black hole attacks*, International Journal of Communication Systems, Wiley, 2017, DOI: 10.1002/dac.3487.

[20] D.K. Sharma, S.K. Dhurandher, I. Woungang, R. K. Srivastava, A. Mohananey, and J. J. P. C. Rodrigues, *A machine learning-based Protocol for efficient routing in opportunistic networks*, IEEE SYSTEMS JOURNAL, December 2016, ISSN (Print): 1932-8184, ISSN (Online): 1937-9234, DOI: 10.1109/JSYST.2016.2630923, , pp. 1-7.

[21] D. K. Sharma, A. Yadav, A. Sharma, and J. Kumar,*KNNR:K-nearest neighbour classification based routing protocol for opportunistic networks*, in proceedings of IEEE Tenth International Conference on Contemporary Computing (IC3 2017), 10-12 August 2017, Noida, India.

[22] D. K. Sharma, S. K. Dhurandher, I. Woungang, A. Bansal, A. Gupta, *GD-CAR: A genetic algorithm based dynamic context aware routing protocol for opportunistic networks*, in proceedings of 20th Intl.Conference on Network-Based Information Systems (NBiS-2017), Aug. 24–26, 2017, Toronto.

[23] D. K. Sharma, D. Kukreja, P. Aggarwal, M. Kaur, A. Sachan, *Poisson's probability-based Q-routing techniques for message forwarding in opportunistic networks*, International Journal of Communication Systems, Wiley, 2018; e3593. https://doi.org/10.1002/dac.3593.

[24] D. K. Sharma, S. K. Dhurandher, D. Agarwal, K. Arora, *kROp: k-Means clustering based routing protocol for opportunistic networks*, Journal of Ambient Intelligence and Humanized Computing, Springer, ISSN: 1868-5137 (print version), ISSN: 1868-5145 (electronic version),pp. 1-18, https://doi.org/10.1007/s12652-018-0697-3.

[25] A. Keranen, *Opportunistic network environment simulator*, Special assignment report. Helsinki University of Technology, Dept. of Communications and Networking, May 2008.

[26] S. K. Dhurandher, D. K. Sharma, I. Woungang, S. Bhati, *HBPR: history based prediction for routing in infrastructureless opportunistic networks*, In: Proceedings of IEEE 27th international conference on advanced information networking and applications (AINA 2013), Barcelona, Spain, 2528 March 2013, pp. 931–936.

[27] S. K. Dhurandher, S. Borah, I. Woungang, D. K. Sharma, K. Arora, D. Agarwal, *EDR: An encounter and distance based routing protocol for opportunistic networks*, In: Proceedings of IEEE 30th international conference on advanced information networking and applications (AINA 2016), Crans-Montana, Switzerland, 23-25 March 2016.

[28] S. K. Dhurandher, D. K. Sharma, I. Woungang, R. Gupta, S. Garg, *GAER: genetic algorithm-based energy-efficient routing protocol for infrastructure-less opportunistic networks*, The Journal of Supercomputing, September 2014, Volume 69, Issue 3, pp. 1183–1214.

[29] T. Spyropoulos, K. Psounis, C. S. Raghavendra, *Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility*, In: Proceedings of the fifth IEEE international conference on pervasive computing and communications workshops (PerComW 07), White Plains, NY, 1923 March 2007, pp. 79–85.

[30] Y. Zhao, *Motion vector routing protocol: A position based routing protocol for mobile ad hoc networks*, In the Graduate College, The University of Arizona.
[31] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, *Maxprop: Routing for vehicle-based disruption-tolerant networks*, Proc. of IEEE INFOCOM 2006, 2006, pp. 1–11.

Edited by: Khaleel Ahmad
Received: Nov 25, 2018
Accepted: Feb 11, 2019