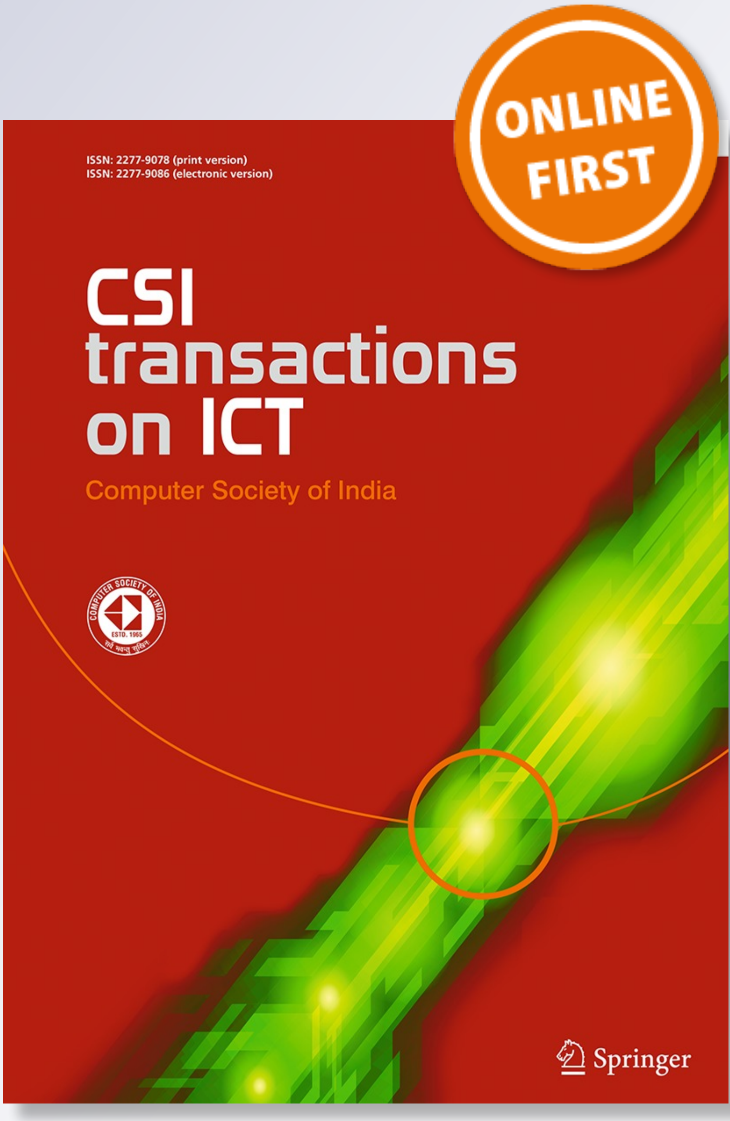
Analyzing and designing energy efficient routing protocol in delay tolerant networks

Nikita Choudhary, Kunal Mittal, Krishna Dwypayan Kota, Preeti Nagrath & Sandhya Aneja

CSI Transactions on ICT

ISSN 2277-9078

CSIT
DOI 10.1007/s40012-016-0094-z



CSIT
DOI 10.1007/s40012-016-0094-z

SPECIAL ISSUE REDSET 2016 OF CSIT

Analyzing and designing energy efficient routing protocol in delay tolerant networks

Nikita Choudhary[1] · Kunal Mittal[1] · Krishna Dwypayan Kota[1] · Preeti Nagrath[1] · Sandhya Aneja[2]

Abstract Delay tolerant networks (Barun Saha, http://delay-tolerant-networks.blogspot.in) are the networks that address the technical issues in heterogeneous network that may lack continuous network connectivity. A message is forwarded to a node if it has higher delivery predictability than the current node for that particular destination. But there obviously exists a tradeoff between delivery delay and energy consumption (Yoon et al. in Tradeoff between energy consumption and lifetime in delay-tolerant mobile network, 2008), as a message (or a file) has to be delivered to each of several destinations by epidemic relaying. With every relay due to the various nodes, energy is consumed by the node. This adds to the total energy spent in relaying and thus increases the network overhead. As the energy available to a particular node is limited, there is a need to conserve this energy. This paper investigates the Energy Aware Prophet protocol and the methods to optimize the delivery probability. Threshold Algorithm and Remaining Required Energy Algorithm (Kaviani et al. in Delay tolerant routing protocols for energy-neutral animal tracking, 2015) leads to an increase of about 42% in delivery probability and energy is saved by 47%. Also, we have found that the Threshold Algorithm leads to the optimal energy value (i.e. threshold value) as 49 J as a constraint to the current energy. Further, a buffer size higher than 8 M is not recommended, as the delivery probability remains constant after 8 M buffer size.

Keywords Delay tolerant networks · Opportunistic network environment · Energy efficient routing · Routing protocols · Prophet protocol · Threshold algorithm · Remaining required energy · Delivery probability · Overhead ratio

✉ Nikita Choudhary
menikitachoudhary@gmail.com

Kunal Mittal
kunalmittal23@gmail.com

Krishna Dwypayan Kota
krishnadwypayan12@gmail.com

Preeti Nagrath
preeti.nagrath@bharatividyapeeth.edu

Sandhya Aneja
sandhya.aneja@ubd.edu.bn

1 Bharati Vidyapeeth College of Engineering, New Delhi, India

2 Faculty of Integrated Technologies, Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei

1 Introduction

The existing TCP/IP based Internet service model requires end-to-end path connection before transferring data. However, maintaining end-to-end connection is not possible for applications like "Interplanetary Networks" [19] and disaster scenarios [20]. Delay Tolerant Network (DTN) has been proposed as a solution for these applications. DTN is implemented over the bundle layer wherein the messages are stored as "bundles" and then forwarded whenever the nodes come in contact with a receiver node.

DTN [10] is a store and forward network in which a node stores the message until it finds a potential forwarder. A number of routing techniques have been proposed for forwarding the message. All of these routing techniques evaluate nodes in contact to find the potential forwarder to the destined node for messages stored in their buffer space. Many strategies proposed for routing rely on either flooding-based techniques or knowledge-based techniques. Epidemic protocol [2] and Prophet routing protocol [3]

Published online: 26 December 2016

take into consideration, the limited amount of energy available for a particular node during the transfer of messages.

Our objective here is to investigate the impact of different buffer sizes on existing DTN routing protocols and DTN architectures [8] and subsequently, come up with a suitable energy efficient DTN routing protocol [5, 16]. A single DTN node cannot continuously forward data to a subsequent node unless the node comes in contact or in range. So the data packets are to be stored locally. For this, we have to provide the system with some initial energy which can be used for the packet transfer, which is a costly task. Also, we cannot risk the disruption in the connection, which would make it prone to attacks [21]. So energy conservation is an important aspect when it comes to data transfer using DTN.

In the following paper, we have showed that energy considerations are critically important where there is an energy constraint deployment and can substantially improve data delivery rates along with the fall in the overhead ratio. We have performed our analysis based on the algorithms proposed in the paper “Delay Tolerant Routing Protocols for Energy Neutral Animal Tracking” [4]. The researchers of The University of Queensland [4] proposed two new energy aware strategies, Threshold Algorithm and Remaining Required Energy Algorithm which show substantial improvement in data yields.

First we implemented the Threshold Algorithm as mentioned in Sect. 5. For different values of the current energy, we find the optimum energy value as a constraint to the current energy as 49 J by using the Threshold Algorithm. This value was compared with the results generated by RRE Algorithm and we find that there is an increase of about 42% in the delivery probability in RRE over the Threshold Algorithm. On the whole, these algorithms conserve the energy by up to 47%.

This paper is structured as follows, Sect. 2 comprises of the related work in the domain of working protocols and energy consumption analysis on delay tolerant networks. Subsequently, we present our problem statement formally in Sect. 3, which is followed by the pre-requisites and the preliminaries to be able to present a solution to the problem statement in Sect. 4 of this paper. Then we present our idea of implementation to conduct extensive analysis and move towards the results in Sect. 5, and finally we conclude our work with Sect. 6.

2 Related work

Many of the explorers of the vast applications of this field of technology aim at achieving higher delivery rates [15] of transmitting the messages to the destination, at the cost of a limited amount of energy. This creates a huge impact over the transmission by not allowing the nodes to send messages when the available energy to a node gets exhausted.

Discussing the issues and the role of energy in the message delivery in delay tolerant networks, researches are on-going that relate to energy consumption analysis [16] and to find various new protocols [9] so that, a solution for the same problem can be discovered [5]. Another important discussion to be made here is that the existing packet delivery protocols make an assumption that there is unconstrained energy available to them and that the mainstream DTN protocols do not adjust their behavior based on the available energy. This is a very wrong assumption and is violated by the very fact that the simple hardware devices participating in the transfer of information are powered by lithium-ion batteries and thus store limited energy.

A cross-layer based energy-efficient routing algorithm [17] was worked upon which broadcasts data packets partially in order to enhance the delivery ratio. But, we wish to stick to the protocols in the bundle layer so that the implementation becomes less complex.

The existing packet delivery protocols include Epidemic protocol [2], which consumes network resources and is therefore not scalable. Epidemic protocol is the simplest form of routing approach where in the replicated message is forwarded to every node coming in contact. The Epidemic approach has been improved by introducing controlled flooding approaches [18].

Another important protocol is the Prophet protocol [3], which uses the history of encounters and transitivity to calculate the probability of a node delivering a message to a particular destination. The Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) protocol [22] delivers data packets by maintaining a set of probabilities for successful delivery to known destinations in the DTN (*delivery predictabilities*) and replicating messages during opportunistic encounters only if the node that does not have the message appears to have a better chance of delivering it. The delivery predictabilities used by each Mule are recalculated at each opportunistic encounter according to the rule:

When the Mule M encounters another Mule E, the predictability for E is increased:

$$P(M,E)_{new} = P(M,E)_{old} + \left(1 - P(M,E)_{old}\right) * L_{encounter}$$

where $L_{encounter}$ is an initialisation constant.

Spray and Wait [6] is a routing protocol that attempts to gain the delivery ratio benefits of replication-based routing as well as the low resource utilization benefits of forwarding-based routing and the MaxProp protocol [7] aims to improve the delivery probability by determining which messages should be transmitted first at next encounter and which messages should be dropped first.

3 Problem statement

The Delay Tolerant Network (DTN) routing protocols have been designed for the environment where there is discontinuous path between source and destination at all points of time. But while transferring messages from source to destination the message is transferred to intermediate nodes which result in the loss of energy of each node as the number of relays increases. In this paper we have focused on the fact that energy conservation is an important factor in the transmission of data as it can increase the data delivery rates along with decrease in overhead ratio.

Here, we have tried to work on two newly proposed algorithms that typically focus on conserving the energy of the nodes.

One is the Threshold Algorithm, which aims at finding an optimum value for energy of a node, which would ensure maximum delivery probability and requires minimum overhead. But then, this is hard to find by the mere use of hit-and-trial approach.

To avoid this, we study the Remaining Required Energy Algorithm that calculates this optimum threshold value as a heuristic function and reduces our efforts, and at the same time, provides us with better performance metrics.

4 Preliminaries

Firstly, we studied the basic DTN routing protocols namely Epidemic, Spray and Wait, Prophet and MaxProp protocols. The protocols were implemented on Opportunistic Networking Environment (ONE) [12] simulator. ONE allows for easy extensibility using Java and has been widely used for DTN and mobility research [13]. The performance of routing protocols is evaluated on the basis of three metrics namely delivery probability [11], number of relays [11] and overhead ratio [11].

We have studied the various packet delivery protocols under the above mentioned performance metrics and came at a conclusion that Prophet router is the most suitable router for transferring data. It proved to be better than the other protocols with a good delivery probability (0.12) also, adding very less to the overhead ratio (ratio came to be 11). Results of the same have been given in Sect. 5.

4.1 Energy module

The energy module [4] introduces the concept of energy consumption in the nodes. In this work, it is assumed that each of the nodes has five states:

Off this state is reached when there is no energy available (when the battery gets discharged). In this state, the node can't establish connections with other nodes.

Inactive The node's interface goes into a sleep state when it is inactive. Although, a node in this state can be detected by other nodes for subsequent communication, the amount of energy spent per unit of time is very much reduced.

Scan Scan state is the state where a node spends energy to find its neighbors and once the contact is established, the node can transmit and receive messages. Each node, switches periodically to this state, according to the *scanInterval* parameter.

- *Transmission* A node in this state is sending messages.
- *Reception* A node in this state is receiving messages.

According to the state, the mechanism uses the method *checkEnergy* to perform the operations that govern the control of energy.

4.2 Threshold algorithm

In Threshold Algorithm [4], there is a user-defined threshold energy below which the node will conserve its energy, and will stop the forwarding of the messages.

The energy threshold E_{th} is a fixed percentage of the harvested energy in the system. A threshold of 0% energy corresponds to no energy awareness and a threshold of 100% energy is equivalent to the Direct Delivery algorithm. Once the stored energy falls below the threshold energy, the protocol will not forward packets to other nodes or receive packets from other nodes for forwarding (it means that, effectively the node changes to the Direct Delivery protocol).

4.3 Remaining Required Energy algorithm (RRE)

In the Threshold Algorithm, user can vary the performance of the routing protocol by controlling the threshold parameter. However, it is not an easy task to find the optimal value of the threshold.

The RRE Algorithm [4] derives the threshold automatically by using a heuristic approach.

The main idea is based on conserving sufficient energy to allow nodes to transmit the existing packets from their buffer to a base station at the time of their next encounter with other nodes.

Energy reservation of a node depends on two factors:

1. The energy required to transmit all the packets collected since the last encounter with a data sink to the next sink or node (E_{tx_packet}) and

2. The energy required for the node to run its basic functionalities until the energy of a node is next recharged (E_{scan}), which is dominated by scanning the radio channel for the presence of a base station. The threshold energy for entering the energy conservation state is therefore:

$$E_{RRE} = E_{tx\_packet} + E_{scan} \tag{1}$$

$$E_{tx\_packet} = N_{packet} * E_{tx\_per\_packet} \tag{2}$$

Here, E_{tx_packet} estimates the energy needed to deliver the packets to a base station which is based on the node's current packet buffer size and E_{scan} can be estimated given the remaining time until the next energy harvesting event. Consider for example, the number of nodes to be 10, generating about packets at 250 kbps for 4320 s. This gives the total data size close to 10.8 MB. Taking the buffer size as 2 MB, number of packets generated at the current time (N_{packet}) is approximately 5. Let the Energy required to transmit a single packet ($E_{tx\_per\_packet}$) be 0.1. The energy required to transmit all the packets (E_{tx_packet}) comes out as 0.1 ($E_{tx\_packet} = 5 \times 0.1$).

Now we can estimate the scan energy to be roughly 0.1 and calculate the E_{RRE} ($E_{RRE} = 0.5 + 0.1$) which sums up as 0.6.

The EnergyAware router class extends the Active router class. It functions in the following manner. First, it checks the energy of a node at a given instant using the *checkEnergy* method and then it analyse the state of that node. When the node is inactive, the energy required is zero. When the node is transmitting or receiving packets, the total time spent is multiplied by the predefined amount of energy spent per unit time in the corresponding operation. The energy spent per unit of time for each operation is supplied as parameter for each class of nodes in the configuration of the simulator. This would also happen when the nodes are in scan state. The second method used in this is the *scanEnergy* parameter, which is the total amount of energy expended in each of the scan operation. And finally, the method used to discount the spent energy is *reduceEnergy*.

5 Implementation and performance evaluation

We first modify the existing Prophet router by implementing it over the EnergyAware Router. We now have the Energy-Aware Prophet router.

We next implement the above stated heuristic based approach to see for ourselves how the energy limitations are taken into consideration and how we can achieve a better overhead ratio at the cost of any value of delivery probability of the message (Table 1).

5.1 Section A

5.1.1 Impact of buffer size on delivery probability, number of relays and overhead ratio

Increasing buffer size increases delivery ratio for all the protocols because as buffer size increases more packets will be stored in the buffer leading to reduction in packets drop by buffer overflow. It can be seen that Epidemic and MaxProp routing gives the better results as compared to Prophet and Spray and Wait routing.

It is clear from Fig. 1 that deliver probability increases as we are increasing the buffer size in all the four routing protocols. We observe that after 8 M buffer size, the delivery probability remains constant.

The lowest energy consumed in terms of relays is achieved by Prophet router. This is due to the reason that nodes in the Prophet router sends packets only to reliable nodes (Fig. 2). In Epidemic router, the overhead ratio is maximum because it sends packets to all possible nodes.

In Epidemic and Spray and Wait routing, as we increased the buffer size, number of relays increased.

The lowest transmission cost for delivering a packet in terms of network overhead and energy consumption, is achieved by Prophet router because Prophet router sends packet only to reliable nodes (Fig. 3). Epidemic has greatest overhead ratio because Epidemic sends packet to all possible nodes. In the case of MaxProp the overhead

Table 1 Simulation parameters

Parameter	Pedestrians (group 1)	Cyclists	Pedestrians (group 2)
Number of hosts	1	17	2
Speed (m/s)	0.5–1.5	2.7–13.9	7–10
Wait time (s)	0–120	0–120	10–30
Buffer size	5 M	5 M	5 M
Packet TTL (min)	300	300	300
Simulation time (s)	4320.0	4320.0	4320.0
Transmission speed	2 Mbps	2 Mbps	2 Mbps
Movement	SPMBM	SPMBM	SPMBM
Interface	Bluetooth	Bluetooth	Bluetooth and high speed

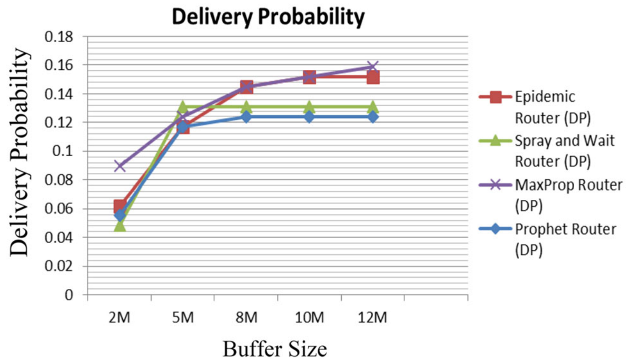


Fig. 1 Impact of buffer size on delivery probability

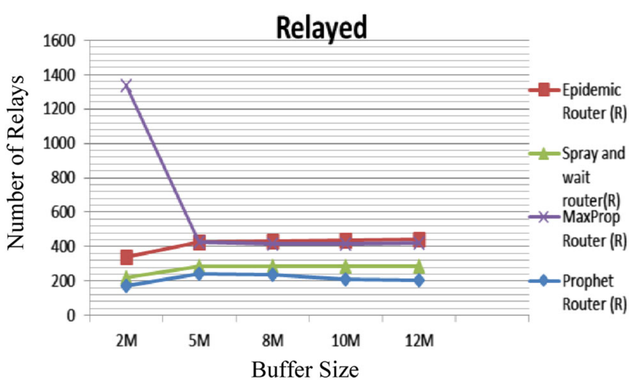


Fig. 2 Impact of Buffer size on number of relays

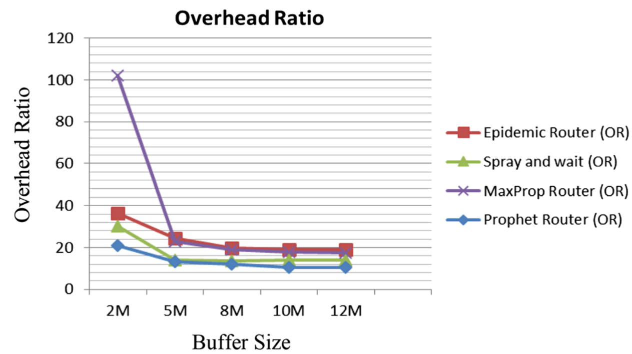


Fig. 3 Impact of buffer size on Overhead ratio

ratio decreases as increase in buffer size because it stores a list of previous intermediaries to prevent data from propagating twice to the same node and prevent unnecessarily relay of packets.

For all the routing protocols, as we increased the buffer size, the overhead ratio decreased. At 2 M buffer size, we observed maximum overhead ratio for Max Prop router.

From the above analysis, we concluded that Prophet router is the most suitable router for transferring data as for a given buffer size it has a good delivery probability, at the same time, less overhead ratio and less number of relays. So, for our further research we have implemented Prophet as the base router.

Further, we have studied two newly proposed algorithms [4] that typically focus on conserving the energy of the nodes. (1) The Threshold strategy enters the conservation state if the total energy falls below a fixed threshold provided by the user; and (2) the Remaining Required Energy (RRE) strategy determines the optimal threshold based on an estimate of the energy that the node needs to transmit existing data to a sink. The efficiency of these simple strategies clearly motivates more research into sophisticated energy-aware strategies for DTN protocols. We evaluate both Threshold and RRE strategies by implementing them on the Energy-Aware Prophet protocol and simulating the results using the ONE simulator [12].

5.2 Section (B)

5.2.1 Algorithm of implementation

1) Implement the ProphetRouter as EnergyAwareRouter.
2) Initialize E_{scan} and $E_{tx\_per\_packet}$

   $E_{scan} = \text{intialEnergy}$

   $E_{tx\_per\_packet} = 0.1$

   $E_{RRE} = E_{tx\_packet} + E_{scan}$ [From reference 4]

   $E_{tx\_packet} = N_{packet} * E_{tx\_per\_packet}$ [From reference 4]
3) Check the values of the energy consumed during the transmission of each packet.
4) Perform the simulations and record the results generated in the messages report file.
5) Using a hit-and-trial approach, try to find the optimum value of current energy.
6) Compare the values corresponding to delivery probability and overhead ratio by plotting suitable graphs.
7) Initialize $E_{scan} = E_{th}$ and compare the results.

5.2.2 Analysis of algorithm

As we have already discussed, the threshold for energy is checked on hit-and-trial basis and this threshold is then used as a constraint on the *currentEnergy*. If the *currentEnergy* is greater than this threshold energy, then only will this energy be accepted and would appear in the output corresponding to the respective node's energy. Else, the node will be prohibited from sending any message.

The number of packets delivered to the destination is more significant performance metric rather than the delivery probability. The reason is that the number of generated packets can vary for different nodes, as nodes do not generate packets when they run out of energy.

In our simulations, we take a total of 20 nodes and the data transfer rate as 250 kbps. The simulation time is limited to 4320 s (72 min). So, the number of potential packets generated per day (if number of nodes run out of Energy) is approximately 20 nodes × 250 kbps × 4320 s.

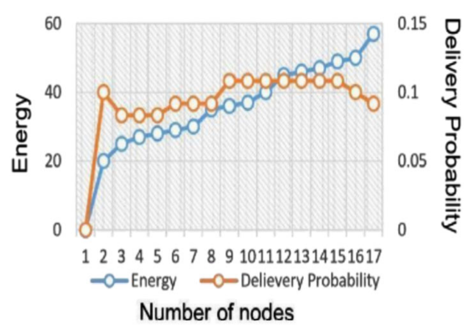


Fig. 4 Delivery probability in Energy Router

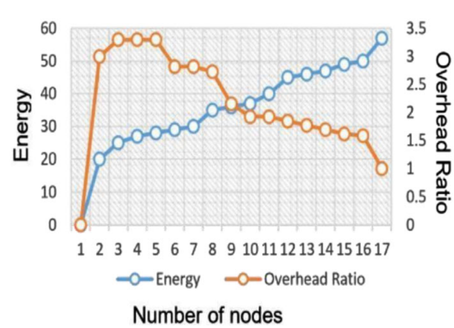


Fig. 5 Overhead Ratio in Energy Router

The exact number for these simulations is 120. Dividing the packets delivered by 120 gives the potential packet delivery percentage (delivery probability).

We use another performance metric in our analysis, overhead ratio, which is defined as the ratio of total number of packets transmitted in the network and the number of uniquely delivered packets. We evaluate performance of both Threshold and RRE algorithms with the Energy Aware Prophet router, and calculate the experiment results corresponding to the various performance metrics.

Figure 4 shows the delivery probability is maximum when the value of the energy level is between 36 and 49. So, the Threshold value came between this energy value (i.e. energy value 36–49). Now, we considered the overhead ratio and found the minimum overhead ratio between the energy values 36–49.

Figure 5 shows that the overhead ratio is minimum at the energy level 49 and also, at this point the delivery probability is maximum.

Then, we cross checked the results with the Threshold Algorithm and compared the results. We simulated this algorithm and then compared the results with the Threshold Algorithm and deduced that the delivery probability is better than the Threshold Algorithm.

The following table (Table 2) gives the results of the implementation of Remaining Required Energy Algorithm wherein the corresponding parameters of Threshold Algorithm are compared.

As it is evident from the results that the delivery probability when RRE Algorithm is used is greater than the Threshold Algorithm by about 42%, which is the primary concern when transmitting the packets in DTN. Further, the number of packets relayed and the number of packets delivered also increase to a marginal extent.

6 Conclusion

So far, we have studied a few of the routing algorithms that delay tolerant networks implement, and of the lot, we base our experiments, to come up with energy efficient DTN framework, on Prophet protocol. Further, we have successfully implemented the Threshold Algorithm and the Remaining Required Energy Algorithm. We have taken

Table 2 Delivery probability, relayed and delivered results

Algorithm	Threshold algorithm (Energy—100 J)			Remaining Required Energy algorithm		
Energy Aware Prophet Routing Protocol	Delivery Probability	Relayed	Delivered	Delivery Probability	Relayed	Delivered
	0.1083	34	13	0.1833	124	22

different values for energies as constraints to the current energy of a particular node, and have arrived at the optimal energy value (i.e. threshold value) as 49 J.

Then, we have implemented the Remaining Required Energy Algorithm and performed a cross comparison of the results generated by the Threshold Algorithm. We conclude with our observations of the compilation, that a routing algorithm can be made energy efficient by implementing it over the EnergyAwareRouter and applying the proposed DTN framework. Our results show that the Remaining Required Energy Algorithm is a better method to find the optimum value of threshold than the Threshold Algorithm. Also, the generated by RRE Algorithm prove to be better than Threshold Algorithm in respects, i.e. the delivery probability, relays and delivered messages.

References

1. S-K Yoon, ZJ Haas, School of Electrical and Computer Engineering Cornell University, JH Kim (2008) Boeing Phantom Works "Tradeoff between energy consumption and lifetime in delay-tolerant mobile network"
2. Vahdat A, Becker D (2000) Epidemic routing for partially connected ad hoc networks, in Technical Report CS200006, Duke University
3. Lindgren A, Doria A, Schelen O (2003) Probabilistic routing in intermittently connected networks. SIGMOBILE Mob Comput Commun Rev 7(3):19–20
4. Kaviani M, Kusy B, Jurdak R, Bergmann N, Zhao K, Liu V (2015) Queensland University of Technology, CSIRO, The University of Queensland, Australia. Delay tolerant routing protocols for energy-neutral animal tracking
5. Cabacas RA, Nakamura H, Ra I-H (2014) Energy consumption analysis of delay tolerant network routing protocols. Int J Softw Eng Appl 8(2):1–10
6. Spyropoulos T, Psounis K, Raghavendra CS (2005) Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In: Proceedings of ACM SIGCOMM workshop on delay-tolerant networking, (Philadelphia, PA, USA, August 2005), pp 252–259
7. Burgess J, Gallagher B, Jensen D, Levine BN (2006) Max-prop: Routing for vehicle-based disruption tolerant networks. In: Proceedings of 25th IEEE international conference on computer communications, pp 1–11
8. Fall K (2003) A delay-tolerant network architecture for challenged internets. In: SIGCOMM '03 Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, pp 27–34
9. Choi BJ (2011) Energy efficient protocols for delay tolerant networks. A thesis presented to the University of Waterloo. https://uwspace.uwaterloo.ca/bitstream/handle/10012/5903/Choi_BongJun.pdf?sequence=1
10. Delay-Tolerant Network—a seminar by Volodymyr Goncharov 07 February 2010
11. Mehto A, Chawla M (2013) Comparing delay tolerant network routing protocols for optimizing L-copies in Prophet routing for minimum delay. In: The conference on advances in communication and control systems
12. Keranen (2008) Opportunistic network environment simulator. Special Assignment, report, Helsinki University of Technology, Department of Communications and Networking
13. Socievole A, Marano S (2012) Evaluating the impact of energy consumption on routing performance in delay tolerant networks. In: Wireless communications and mobile computing conference (IWCMC)
14. Barun Saha. http://delay-tolerant-networks.blogspot.in
15. Han SD, Chung YW (2015) An improved PRoPHET routing protocol in delay tolerant network. Sci World J 2015:623090. doi:10.1155/2015/623090
16. Liu Y, Bild DR, Adrian D, Singh G, Dick RP, Wallach DS, Mao ZM (2015) Performance and energy consumption analysis of a delay-tolerant network for censorship-resistant communication, Proc. Int. Symp. on Mobile Ad Hoc Networking and Computing (accepted)
17. Yao YK, Zheng WX, Ren Z (2012) An energy-efficient routing algorithm for disruption tolerant networks. In: Proceedings of the 2012 2nd international conference on computer and information application (ICCIA 2012)
18. Yao H, Huang H, Zeng D, Li B, Guo S (2014) An energy-aware deadline-constrained message delivery in delay-tolerant networks. Springer, New York
19. Burleigh S, Hooke A, Torgerson L et al (2003) Delay tolerant networking: an approach to interplanetary internet. IEEE Commun Mag 41(6):128–136
20. Campillo AM, Crowcroft J, Yoneki E, Mart R (2013) Evaluating opportunistic networks in disaster scenarios. J Netw Comput Appl 36(2):870–880
21. Ansa G, Cruickshank H, Sun Z (2012) An energy-efficient technique to combat DOS attacks in delay tolerant networks. Centre for Communication Systems Research, University of Surrey, Guildford, United Kingdom
22. Oria A, Scheln O (2003) Probabilistic routing in intermittently connected networks. In: Proceedings of the fourth ACM international symposium on mobile ad hoc networking and computing (MobiHoc 2003)