Improving Message Deliverability of Opportunistic Network Protocols

Bhed Bahadur Bista
Faculty of Software and Information Science, Iwate Prefectural University, 020-0693, Japan
Email: bbb@iwate-pu.ac.jp

***Abstract*—Opportunistic Networks (OppNets) are designed to transfer messages between intermittently connected mobile nodes which are unaware of network topology. As the traditional routing protocols, which assume continuous connection between nodes and need to be aware of network topology, cannot be implemented in OppNets, various OppNet routing protocols, such as Epidemic, PRoPHET etc., have been proposed for message delivery. An OppNet's message delivery mechanism is store-and-forward in which a node stores incoming messages and forwards the copies of the messages to nodes it encounters causing a large overhead in the network. Furthermore, as nodes are mobile and are powered by batteries, it is essential to consume less energy and extend the network life so that a large number of messages can be delivered. In order to improve the message deliverability of a node, network overhead and network life expectancy in existing OppNet protocols, we consider the node's resources, whose values change continuously, while transferring messages in the networks. In our proposal, a node first calculates its Message Deliverability (MD) which depends upon the protocol used and its resources. The node then uses its MD and the encountered node's MD to make message forwarding decision. Extensive simulation results have shown that the message delivery probability, network overhead and network life of existing OppNet protocols can be improved significantly using our proposal.**

***Index Terms*—OppNets, message deliverability, routing protocols, overhead, network life**

I. INTRODUCTION

The number of mobile wireless communication devices, equipped with Bluetooth [1] and WiFi [2], in the world is estimated to be more than world population [3]. Such ubiquitous devices have introduced the concept of OppNets which are extensions of MANETs [4]. Unlike MANETs, OppNets, also referred as Delay Tolerant Networks (DTNs), don't have any specific network topology and nodes are always in motion causing intermittent connections. As the existing MANET protocols cannot be used in OppNets, various OppNet routing protocols have been designed to deliver messages from source to destination nodes. Some OppNet protocols use flooding approach to spread the message in the network expecting it to reach the destination node such as Epidemic [5], [6] and Spread-and-Wait [7] protocols. There are some OppNet protocols that use nodes' probabilistic encountering to other nodes and/or human mobility approach such as PRoPHET [8], [9] and PRoWait [10] to deliver messages to destination nodes. Some OppNet protocols consider mobility of nodes only to find the best possible approach to deliver messages to destination nodes [11].

Other different approaches are also used to design routing protocols for OppNets. Ants' social-aware approach is used to propose routing protocol for OppNets based on Cultural Algorithm and Ant Colony Optimization to identify the most promising social-aware forwarder in the network [12]. A genetic algorithm based routing protocol for OppNets is proposed in [13]. It uses genetic search algorithm to predict the path for a message by dynamically updating the context information stored in each node. Game theory based approach is taken to optimize routing protocol strategy in the networks when nodes energy is limited [14]. Authors in [15] proposes protocol for Delay tolerant networks when the privacy of nodes or sensitive information such as contact history is required to be preserved. Further details and different types of routing protocols can be found in OppNets routing protocols surveys [16]-[18].

Nodes in OppNets are mobile, i.e., are powered by battery, and employ store-and-forward mechanism to deliver messages. As a result, a large number of messages are copied and forwarded in the network causing overhead and reducing the network life as nodes will consume energy faster. In order to improve the message deliverability, overhead and network life of OppNet routing protocols we consider resources of nodes, whose value change continuously, while transferring messages from one node to another in the network. First, we define the message deliverability of a node which considers the current level of the most important resources, energy and available free buffer, of the node and the OppNet routing protocol that has been used. When a node encounters another node, it compares its message deliverability with that of the encountered node's message deliverability. If its message deliverability is less than that of the encountered node's message deliverability, it will transfer copies of messages it has but the encountered node does not have to the encountered node. On the other hand, if its message deliverability is higher than the encountered node's message deliverability, it will receive copies of messages it does not have but the encountered node has from the encountered node.

Manuscript received April 1, 2018; revised May 15, 2018; May 21, 2018.
Corresponding author: Bhed Bahadur Bista (email: bbb@iwate-pu.ac.jp)

©2019 Int. J. Elec. & Elecn. Eng. & Telcomm.
doi: 10.18178/ijeetc.8.1.51-56

We have applied our proposal to well-known Epidemic and PRoPHET protocols and the extensive simulations have shown that the message delivery probability of the protocols have been improved significantly. Moreover, we also checked the effect of our proposal to the network life and the overhead of the network and found that the proposal extends the network life and reduces the overhead of message transfer in the network.

The remaining of the paper is organized as follows. In Sect. II, we explain the proposed message delivery improvement of OppNet routing protocols. In Sect. III, we present the simulation environment and in Sect. IV, we present the performance results and analysis. Finally we conclude and give the future directions of our work in Sect. V.

II. Message Delivery Improvement Proposal

As nodes are not aware of network topology in OppNets, a source node is unable to search routes to a destination node before sending messages. Store-and-forward mechanism is the best approach to deliver messages in OppNets. A node stores incoming messages and forwards copies of the messages to nodes it encounters in OppNets. In order to improve the message delivery probability of OppNets, a node with higher message deliverability should receive the forwarded messages and store and forward them to encountered nodes with higher message deliverability than itself. When a node encounters another node, they exchange their message bundles, explained below, to find out which messages it has but the encountered node does not have and vice versa.

A. *Message Bundle*

Each node in OppNets holds messages it has generated and messages forwarded to it by other nodes. Messages are stored and indexed in a hash table for efficiency purpose. Furthermore, each node prepares a bit vector called the "summary vector (*SV*)" indicating which entries in the local table are set. When two nodes come to the transmission range of each other, they exchange their summary vectors. After receiving the summary vector, each node determines which messages it does not have but another node has by negating its *SV* and logically ANDing it with the received *SV*. By doing so, a node knows which messages it does not have but the encountered node has. Nodes then forward messages according to the forwarding algorithm.

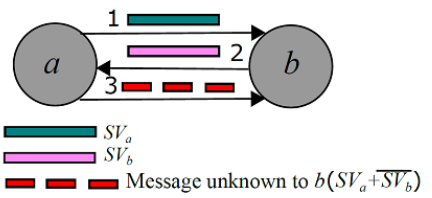

Fig. 1. Example of finding unknown message bundle

As an example, in Fig. 1, node *b* receives summary vector SV_a from node *a*. Node *b* then performs negation to its summary vector SV_b and then logical AND operation with SV_a indicating which messages it does not have but node *a* has. Node *a* also can find out which messages it does not have but node *b* has by performing the similar operation.

For example, for SV_a and SV_b given below, node *b* finds that it does not have messages m1 and m3 but node *a* has. A node forwards copies of messages to the encountered node which does not have the messages it has using the implemented algorithm.

SV_a

m1	m2	m3	m4	m5
1	0	1	0	0

SV_b

m1	m2	m3	m4	m5
0	1	0	1	1

$SV_a \wedge (\neg SV_b)$

m1	m2	m3	m4	m5
1	0	1	0	0

B. *Message Deliverability of a Node*

Deliverability of a message stored at a node depends upon the property of the routing protocol used in OppNets, remaining energy of the node and the available free buffer of the node to store incoming messages. We define the message deliverability as shown below.

Definition: Message Deliverability of node *a*, MD_a, for messages stored in it for some other destination nodes is defined as shown in Eq. (1).

$$MD_a = PV + E_a + FB_a \tag{1}$$

where *PV* is the protocol value of the routing protocol used in OppNets, E_a is the remaining energy of node *a*, and FB_a is the available free buffer of node *a*.

PV of routing protocols in OppNets depends upon a node's relationship with other nodes to make message forwarding decision [21]. It may be social relationship such as friends, relatives, co-coworkers, SNS connections, etc. It may be number of times a node encounters other nodes. It may be social communities one belongs to and so on.

Since mobile nodes in OppNets are powered by battery, it is important to consider remaining energy of a node. If the node has very little energy, it will die before it can forward/deliver messages. Here we assume that a node's battery is not recharged or replaced until the end of the simulation. So higher the energy a node has, higher the probability that the node can deliver or forward messages to other nodes.

Available free buffer of a node is also very important to be considered in OppNets since store-and-forward mechanism is used to deliver messages. In this mechanism, when the buffer is full, a node drops older messages in the buffer to make space for new incoming messages. If a node has higher available free buffer, it can store incoming messages without dropping the older messages in the buffer improving the probability of messages being delivered to the destination nodes.

C. *Message Delivery Algorithm*

The higher the message deliverability a node has, the higher the chances that the node will deliver the messages it has to destination nodes or be able to forward the

©2019 Int. J. Elec. & Elecn. Eng. & Telcomm.

messages to other nodes. When a node encounters another node, they exchange their message deliverability (*MDs*), Summary Vectors (*SVs*) and how much free available buffer (*FB*) they have. Each node performs routing protocol calculations if there is any. The value of the calculation depends upon the routing protocol that has been used. From the S*Vs*, each node finds the message bundle, i.e., messages it has but the encountered node does not have. It then compares its *MD* with that of the encountered node's *MD*. If it is less than that of the encountered node's *MD* and the encountered node has enough buffer to store the messages, it will put the messages in message bundle to the message send list and send it. Otherwise, it will wait to receive messages from the encountered node. The message delivery algorithm we have proposed is outlined in Algorithm 1 which is self-explanatory due to the comment in each line.

We define the following notations to use in the algorithm.

SV_a: summary vector of node *a*.
SV_b: summary vector of node *b*.
FB_a: free available buffer of node *a*.
FB_b: free available buffer of node *b*.
MD_a: message deliverability of node *a*.
MD_b: message deliverability of node *b*.

Algorithm 1 When node *a* encounters node *b*

1: sendInfo(MD_a, SV_a, FB_a) ▹ *a* sends its info. to *b*.
2: receiveInfo(MD_b, SV_b, FB_b) ▹ *a* receives *b*'s info.
3: perform routing calculation if any
4: MsgBundle = $SV_a \wedge \overline{SV_b}$ ▹ *a* checks which messages
5: ▹ it has but *b* does not have.
6: **if** $MD_a < MD_b$ **then**
7:     **for all** $m \in$ MsgBundle **do**
8:         **if** $FB_b >$ MsgSize(m) **then** ▹ check if *b* has enough
9: ▹ buffer for the message
10:             addMsgSendLst(m) ▹ *a* puts message *m*
11: ▹ to the send list.
12:             $FB_b = FB_b -$ MsgSize(m) ▹ reduce *b*'s buffer
13: ▹ by message size *m*.
14:         **end if**
15:     **end for**
16:     sendMsg() ▹ *a* sends the message list to *b*.
17: **else**
18:     recMsg() ▹ *a* waits to receive messages from *b*.
19: **end if**

III. Simulation Environment

We applied our message delivery algorithm to two of the well-known OppNet routing protocols, PRoPHET [8] and Epidemic [5]. In Epidemic, a node floods messages to all nodes it encounters and does not require any routing protocol calculation (step 3 of Algorithm 1). Whereas when applying our algorithm to PRoPHET we need to perform the following Delivery Predictability (DP) calculation of PRoPHET which is assigned to *PV*. DP of a node to another node depends how often it encounters the node. DP of node *a* to node *b*, $P_{(a,b)}$, in direct encounter is calculated as shown in Eq. (2). $P_{init} \in [0, 1]$ is an initial randomly chosen constant.

$$P_{(a,b)} = P_{(a,b)old} + \left(1 - P_{(a,b)old}\right) P_{init} \tag{2}$$

DP decays over time and is calculated as

$$P_{(a,b)} = P_{(a,b)old} \cdot \gamma^k \tag{3}$$

where $\gamma^k \in [0, 1]$ is an aging constant and *k* is the number of time units that has elapsed.

If node *a* meets node *b* and node *b* meets node *c*, then DP node *a* has for node *c* is calculated as

$$P_{(a,c)} = P_{(a,c)old} + \left(1 - P_{(a,c)old}\right) P_{(a,b)} P_{(a,c)} \beta \tag{4}$$

where $\beta \in [0, 1]$ is an impact scaling factor which is constant. The values of $P_{(a,b)}$ and $P_{(a,c)}$ is assigned to *PV* in (1).

We simulated our proposed message delivery algorithm applied to PRoPHET and Epidemic. In order to show the improvement in message delivery, the network life expectancy and the overhead of OppNets, we implemented the original PRoPHET and Epidemic also. We used the well-known OppNet protocol simulator called "Opportunistic Network Environment (ONE)" [19], [20]. Since ONE is written in Java, we need to implement the algorithm in Java. Simulations were performed for 100~500 nodes and buffer size of 25MB~125MB. The movement speed of a node was set to 0.5~1.5 m/s to simulate human walking speed. We used Shortest Path Movement model for node movement. A node selects its destination in a map and chooses the shortest path to reach the destination. The map used was Helsinki City map which is included in ONE. The rest of the other parameters are shown in Table I and should be self-explanatory.

Energy parameters of nodes were set as shown in Table II. All nodes have the same initial energy (in units). Scan energy represents the energy for scanning or discovering devices/neighbors. Scan response energy represents the energy consumed while responding the neighbors on discovery. Transmit energy is energy used when transmitting messages and is higher than other values. Base energy is the energy consumed while a node is idle. We assume that when a node's energy is zero it does not execute any functions, i.e., a dead node. When all nodes die, the network also dies and shows its life expectancy.

Table I. Simulation Parameters

Parameters	**Values**
Simulation Area	4500m × 3400m
Number of Nodes	100 ~ 500
Interface	Wifi
Interface Data Rate	2Mbps
Radio Range	100m
Movement Speed	0.5 ~ 1.5m/s
Buffer Size	25MB ~ 125MB
Message Size	500KB ~ 1MB
Message Generation Interval	25s ~ 35s
Message TTL	300 minutes (5 hours)
Simulation Time	43200s (12 hours)

©2019 Int. J. Elec. & Elecn. Eng. & Telcomm.

TABLE II. ENERGY SETTINGS

Parameters	Values (units)
Initial Energy	4800
Scan Energy	0.12
Scan Response Energy	0.14
Transmit Energy	0.15
Base Energy	0.13

IV. PERFORMANCE RESULTS

Here, MD_PRoPHET and MD_Epidemic are PRoPHET and Epidemic that implement our proposed algorithm respectively. We compare MD_PRoPHET and MD_ Epidemic with that of PRoPHET and Epidemic for message delivery probability, network overhead and network life.

A. Message Deliverability

The message delivery probability is defined as shown in (5), i.e., total messages delivered over total messages generated in the network.

$$\text{delivery Probability}=\frac{Total_{msgDeliv}}{Total_{msgGen}} \quad (5)$$

where $Total_{msgDeliv}$ is the total number of messages delivered in the network and $Total_{msgGen}$ is the total number of messages generated in the network by source nodes for delivering to destination nodes. It is important to note that $Total_{msgGen}$ does not include messages that are forwarded by nodes which are the copies of the messages generated by the source nodes. If all messages that are generated are delivered to the destination nodes, the delivery probability becomes one which is the best scenario of the network. However, due to the resource constraints of nodes or the nature of the routing algorithm, some messages are dropped before they are delivered to the destination nodes. It is essential to deliver as many messages as possible and maximize the message delivery probability.

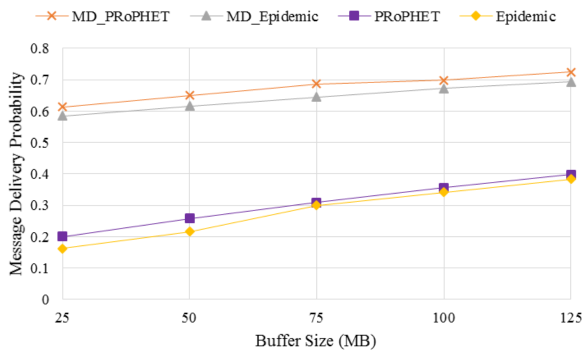

Fig. 2. Message delivery probability for varying buffer size for 300 nodes in the network.

Fig. 2 shows the message delivery probability for different buffer size of nodes when there are 300 nodes in the network. Though we presented the result of 300 nodes simulation only, the similar pattern is found in other number of nodes in the network also. As we can see, the delivery probability of messages increases as the buffer size increases which is obvious because a node with higher buffer size will be able to hold messages longer, i.e., messages will be dropped less often due to full buffer, and thus the chances of delivering the messages to destination nodes increases. From the figure, we can see that the message delivery probability of MD_PRoPHET and MD_Epidemic is higher than that of PRoPHET and Epidemic. Fig. 3 shows the message delivery probability for different number of nodes in the network with 50MB buffer size of each node. In this case also MD_PRoPHET and MD_Epidemic perform better than PRoPHET and Epidemic. From the results we confirm that our proposed algorithm increases the delivery probability of messages in the network even if the number of nodes and buffer size of each node are changed.

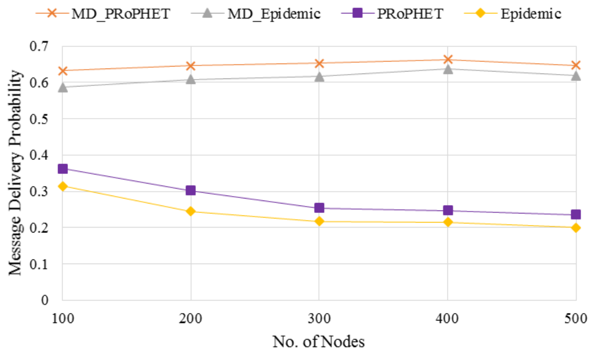

Fig. 3. Message delivery probability for varying no. of nodes for 50MB buffer size of each node.

B. Overhead Ratio

The overhead ratio is defined as shown in Eq. (6), i.e., the difference of total messages forwarded and total messages delivered over total messages delivered in the network.

$$\text{Overhead Radio}=\frac{Total_{msgFrd}-Total_{msgDeliv}}{Total_{msgDeliv}} \quad (6)$$

where $Total_{msgFrd}$ is the total number of messages forwarded/relayed in the network and $Total_{msgDeliv}$ is as defined in Section IV. A. The forwarded messages are the copies of messages generated by source nodes for destination nodes. There are many such messages in the network compared to the messages generated by source nodes. Though the increase in forwarded messages in the network increases the chances of delivering messages to destination nodes earlier, they consume a lot of resources of nodes also. The overhead ratio is essentially the number of copies of messages that are created per delivered message in the network. It can be considered as the assessment of bandwidth efficiency also because if more messages are copied then there will be more transmissions thus consuming more bandwidth, buffer and energy of nodes.

In both Fig. 4 and Fig. 5 the overhead ratio of MD_PRoPHET and MD_Epidemic is significantly less than that of PRoPHET and Epidemic because in comparison in MD_PRoPHET and MD_Epidemic, forwarding of messages as well as dropping of older messages from the buffer due to buffer overflow occurs less frequently than PRoPHET and Epidemic. Though this may result in delay of messages delivery, which is not an important issue in OppNets as they are designed for delay tolerant of message delivery, the consumption of nodes' resources is less. In effect, many more messages are delivered to the destination nodes as we can see in improved delivery probability of messages in Sect. IV.A. For PRoPHET and Epidemic, the overhead ratio decreases as the buffer size increases as shown in Fig. 4, because messages are retained in the buffer longer. However, it increases as the number of nodes in the network increases as shown in Fig. 5, because messages are copied more often as the encountering of nodes happens more frequently. This also results older messages being dropped more frequently due to buffer being filled frequently. This causes more messages to be forwarded again as nodes will not have the messages that are dropped.

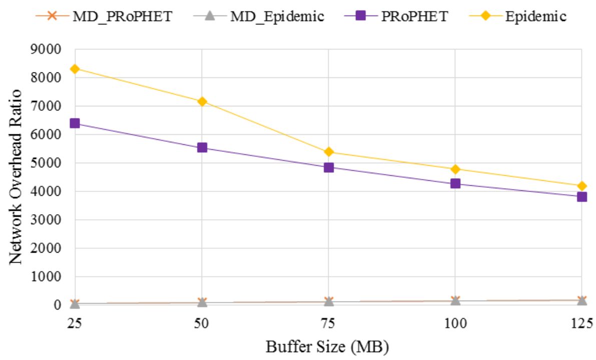

Fig. 4. Network overhead ratio for varying buffer size for 300 nodes in the network.

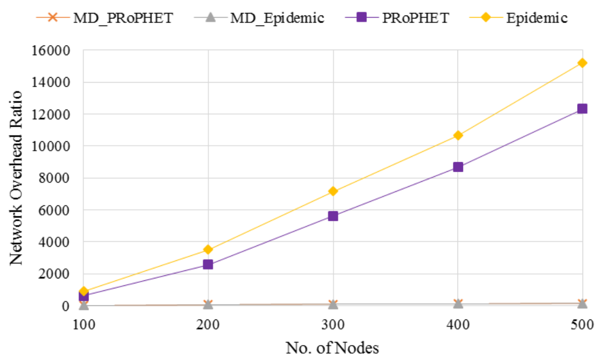

Fig. 5. Network overhead ratio for varying no. of nodes for 50MB buffer size of each node in the network.

C. Network Life

We also performed simulation until all nodes consume their all energy to find the network life. Though in reality, nodes' batteries may be recharged or replaced, we assume that the batteries will not be recharged or replaced for the simulation. When all nodes consumed their all energy, they cannot perform any functions and the network is considered to be dead. In 300 nodes simulation (Fig. 6), the network life of MD_PRoPHET and MD_Epidemic is extended by 60 minutes and 105 minutes compared to PRoPHET and Epidemic respectively. Whereas, in 500 nodes simulation (Fig. 7), it is extended by 90 minutes and 120 minutes respectively. From the figures, we also see that for PRoPHET and Epidemic, the network life decreases as the number of nodes in the network increases because nodes consume the resources for forwarding messages very frequently as we have explained above. The network life in MD_PRoPHET and MD_Epidemic does not change. All the nodes die around the same time. As we can see from figures, in PRoPHET and Epidemic some nodes consume energy faster than others and they die earlier. Dead nodes may be destination nodes for some messages in which cases they will never be delivered. As a result, the message delivery probability will decrease also.

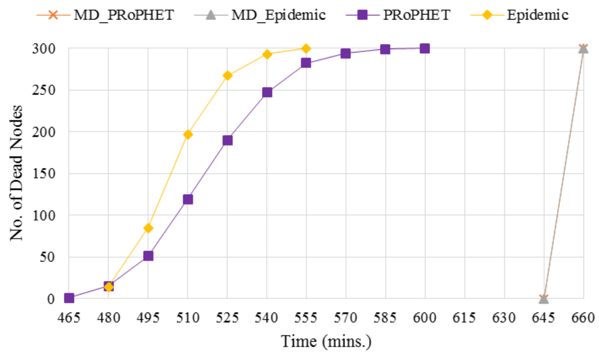

Fig. 6. No. of dead nodes for 300 nodes simulation

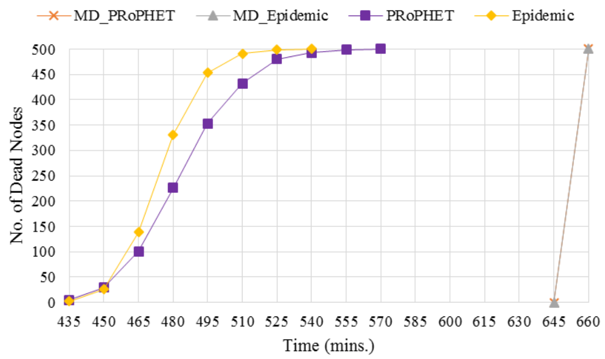

Fig. 7. No. of dead nodes for 500 nodes simulation

V. Conclusions

In this paper, we defined message deliverability of a node and proposed an algorithm that uses it to improve the message delivery probability of OppNet routing protocols. We applied the proposed algorithm to well-known OppNet routing protocols, PRoPHET and Epidemic, to show its effectiveness. The extensive simulation results show that the efficiency of PRoPHET and Epidemic in term of the message delivery probability, the network overhead and the network life has improved significantly. The improvement is due to the consideration of nodes' resources while defining the message deliverability of nodes. Since, in OppNets, nodes continuously move and exchanges messages as they encounter with each other consuming their resources, the nodes' resources changes dynamically and it is essential to consider the nodes' resources every time they exchange the messages. Nodes with higher resources should receive copies of messages from the nodes with lower resources as they will more likely to retain messages in the buffer and live longer also. Though, we left it for future work, we believe that the proposed algorithm can be applied to other OppNet routing protocols to improve their message delivery probabilities, overhead and network life.

REFERENCES

[1] The Bluetooth Specification. [Online]. Available: http://www.bluetooth.com

[2] WiFi. IEEE 802.11 Standard. [Online]. Available: http://www.ieee802.org/11/

[3] GSMA. GSMA real-time tracker. [Online]. Available: https://www.gsmaintelligence.com/

[4] L. Lilien, A. Gupta, and Z. Yang, "Opportunistic networks for emergency applications and their standard implementation framework," in *Proc. IEEE Int. Performance, Computing, and Communications Conf.*, 2007, pp. 588–593.

[5] A. Vahdat and D. Becker. "Epidemic routing for partially-connected ad hoc networks," Tech. rep., Duke University, Apr. 2000.

[6] X. Lu and P. Hui, "An energy-efficient n-epidemic routing protocol for delay tolerant networks," in *Proc. IEEE Fifth Int. Conf. on Networking, Architecture, and Storage*, July 2010, pp. 341–347.

[7] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM Workshop on Delay-tolerant Networking*, New York, NY, USA, 2005, pp. 252–259.

[8] A. Lindgren, A. Doria, E. Davies, and S. Grasic. (August 2012). Probabilistic routing protocol for intermittently connected networks. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6693.txt

[9] C. K. Lee, L. J. Chen, and T. K. Huang, "Prophet+: An adaptive prophet-based routing protocol for opportunistic network," in *Proc. 24th IEEE Int. Conf. on Advanced Information Networking and Applications*, 2010, pp. 112–119.

[10] S. K. Dhurandher, S. J. Borah, M. S. Obaidat, D. K. Sharma, S. Gupta, and B. Baruah, "Probability-based controlled flooding in opportunistic networks," in *Proc. 12th Int. Joint Conf. on e-Business and Telecommunications*, 2015, vol. 6, pp. 3–8.

[11] M. Kawecki and R. O. Schoeneich, "Mobility-based routing algorithm in delay tolerant networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 1, pp. 1–9, 2016.

[12] A. C. K. Vendramin, A. Munaretto, M. R. Delgado, A. C. Viana, and M. Fonseca, "A social-aware routing protocol for opportunistic networks," *Expert Systems with Applications, Elsevier*, vol. 54, pp. 351-363, July 2016.

[13] D. K. Sharma, S. K. Dhurandher, I. Woungang, A. Bansal, and A. Gupta, "GD-CAR: A genetic algorithm based dynamic context aware routing protocol for opportunistic networks," in *Proc. Int. Conf. on Network-Based Information Systems*, 2017, pp. 611-622.

[14] S. L. F. Maia, R. Silva, and P. R. Guardieiro, "A Bayesian game based optimization strategy proposal for routing in energy constrained DTNs," in *Proc. 33th Brazilian Symposium on Computer Networks and Distributed Systems*, 2015, pp. 1–9.

[15] N. Magia, C. B. Iglesias, P. Pereira, and M. Correia, "PRIVO: A privacy-preserving opportunistic routing protocol for delay tolerant networks," in *Proc. IFIP Networking Conf. (IFIP Networking) and Workshops*, 2017, pp. 1-9.

[16] V. G. Menon and P. M. J. Prathap, "Comparative analysis of opportunistic routing protocols for underwater acoustic sensor networks," in *Proc. of 2016 Int. Conf. on Emerging Technological Trends*, 2016, pp. 1-5.

[17] M. Alajeely, R. Doss, and A. Ahmad, "Routing protocols in opportunistic networks - A survey," *IETE Technical Review*, pp. 1–19, 2017.

[18] L. Wan, F. Liu, Y. Chen, and H. Zhang, "Routing protocols for delay tolerant networks: survey and performance evaluation," *Int. Journal of Wireless & Mobile Networks*, vol. 7, no. 3, pp. 55-69, June 2017.

[19] A. Ker ̈anen, J. Ott, and T. K ̈arkk ̈ainen, "The ONE simulator for DTN protocol evaluation," in *Proc. 2nd Int. Conf. on Simulation Tools and Techniques*, 2009, pp. 55:1–55:10.

[20] The ONE. [Online]. Available: https://akeranen.github.io/the-one/

[21] C. Prabha, S. Kumar, and R. Khanna, "Analysis of routing and forwarding protocols in opportunistic networks," in *Proc. Int. Conf. on Computational Modeling and Security*, 2016, pp. 892–898.

Bhed Bahadur Bista received the B.Eng. degree in Electronics from the University of York, England and the M.S. and Ph.D. degrees in Information Science from Tohoku University, Japan. After his Ph.D., he worked at the Miyagi University, Japan, for one year as a Research Associate and moved to the Iwate Prefectural University, also in Japan as an Assistant Professor. Currently, he is an Associate Professor with the Faculty of Software and Information Science at the same university. His research interests include energy efficient networks, mobile networks, sensor networks, ad hoc networks, cognitive radio networks, and cellular networks. He has organized International Workshops and has actively taken part as a Program Chair, a Track Chair and a Program Committee Member in various international conferences including flagship IEEE AINA, NBiS, BWCCA and IMIS. He is a member of IPSJ, IEICE and IEEE.