# SMART: Secured and Mobility Aware Routing Technique for Opportunistic IoT Network in Smart Cities

S. P. Ajith Kumar[1], Hardeo Kumar Thakur[2], Koyel Datta Gupta[3] and Deepak Kumar Sharma[4,*]

[1]*Research Scholar, Manav Rachna University & Department of Computer Applications, Bhai Parmanand DSEU Shakarpur Campus II, India*
[2] *SCSET, Bennett University, Greater Noida, India*
[3]*Department of Computer Science & Engineering, Maharaja Surajmal Institute of Technology, New Delhi, India*
[4]*Department of Information Technology, Indira Gandhi Delhi Technical University for Women, Delhi, India*
*E-mail: s.p.kumar@dseu.ac.in; Hardeo.Thakur@bennett.edu.in; koyel.dg@msit.in; dk.sharma1982@yahoo.com*
[*]*Corresponding Author*

Received 09 February 2023; Accepted 31 March 2023;
Publication 29 March 2024

## Abstract

Transferring data between nodes in the Opportunistic Internet of Things (OppIoT) network may lead to the transmission of multiple copies of each message, which can increase communication costs and jeopardise network security. This necessitates a routing method that is effective and can address both problems. To protect transmitted data and reduce communication overhead, this study suggests a Secured and Mobility Aware Routing Method (SMART) routing algorithm for OppIoT networks in smart cities. With a buffer size of 30 MB and an overhead ratio of 27.9, the delivery probability

*Journal of Mobile Multimedia, Vol. 20_2,* 335–358.
doi: 10.13052/jmm1550-4646.2024

can be increased by more than 50%. The simulation's findings demonstrate that, in terms of delivery probability, overhead ratio, and reports, the proposed SMART protocol outperforms more traditional routing methods.

**Keywords:** Internet of Things, opportunistic network, smart cities, routing protocols, delivery probability, time to live, scheduled energy, ONE simulator.

## 1 Introduction

Growth of smart cities has resulted in immense use of IoT devices (smart vehicle, smart grid, smart appliances, smart health monitoring systems etc.). These IoT enabled devices are spread over the city to effectively monitor and control public resources. Thus, rapid growth is observed in the IoT [1] network that comprises sensors, IoT-enabled devices and other objects, that can communicate over the wireless network using their IP address. OppIoT [2] network is an amalgamation of Opportunistic Network (OppNet) and Internet of Things (IoT). OppNets aims to deliver a message even in absence of a route from source to destination. In these networks, there is no explicit connection between the nodes and messages are circulated in a "store-carry-forward" manner [3–5]. These networks establish routes opportunistically by selecting nodes that have the potential to deliver the message nearer to the destination node, as the succeeding hop. OppIoT [6–8] (Figure 1) empowers smart devices to communicate with one another by selecting moving nodes opportunistically to deliver the message to the target device.

OppNets suffers from several problems including uncertainty in message delivery, substantial delay in message delivery and energy consumption for sensors [9]. The diversified components that constitute the OppIoT further intensify the risk of compromise in data security and privacy. Miscreant nodes can drop the received packets deteriorating the performance of the network. Thus, providing security and privacy for OppIoT network is very crucial. The parameters like message delivery/drop rate, hop count, average latency, and message overhead determine the efficiency of the routing protocol. Cluster-based routing protocols [10] are popular in wireless sensor networks and are used in OppIoT networks. In protocols like the Spray and Wait [11], the concepts of security and mobility awareness lack due consideration while determining the forwarding strategy. This accounts for over or underestimation of the number of copies to be forwarded or even failed/incomplete transfers due to lack of time and energy of the sending

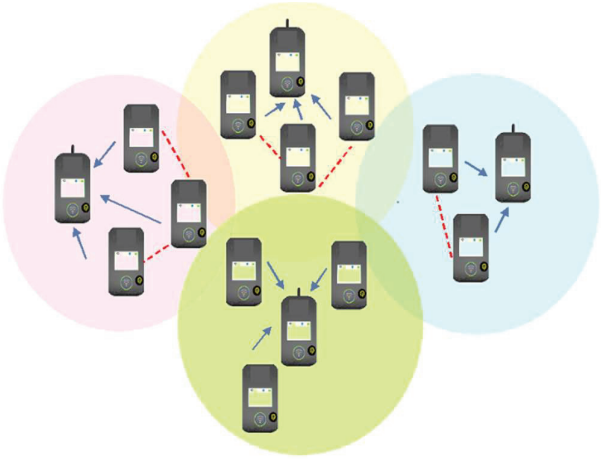

**Figure 1** Opportunistic IoT network.

or receiving node. SMART proposes a novel technique to overcome these obstacles and promises to ensure better security and delivery ratios backed by lower message losses. Researchers in cloud computing are increasingly shifting their attention to edge computing and fog computing research due to growing need for latency minimization in IoT applications [12]. So, in the future SMART protocol can be beneficial in smart city applications.

The key functions of the proposed SMART model include:

- A dynamic trust-based and mobility-aware routing approach that provides lower message drop probability and higher security.
- Scheduled energy is also used as a factor to determine the number of copies a node should forward to an encountered node from the number of messages it has left with itself.
- Further, SMART analyses the amount of time that is required to initiate and successfully transfer a certain number of copies of a message from a sender node to a receiver node before the actual transfer takes place so that any failures due to a shortage of remaining time of node or lack of time left of transferring node are drastically reduced. This is highly beneficial for all smart city applications.

The rest of the paper is organised as follows. Section 2 includes a list of relevant related works. Section 3 describes the proposed SMART model. Section 4 presents the results of the experiments. Finally, the work is concluded in Section 5.

## 2 Related Work

In OppIoT the fixed and moving IoT devices [13] collect data from the surroundings and transmit the information for further processing and analysis. Efficient IoT systems emerge as opportunistic routes are built on the opportunistic nature of human and device contact. OppIoT can prove to be extremely effective in IoT applications [14] like industrial control systems, smart homes, fire detection, environmental monitoring, disaster management, healthcare, wildlife monitoring and much more. However, there are several challenges in the practical implementation of OppIoT networks providing secured transmission of information like identifying a "good" neighbour and operating under different protocols followed by the heterogeneous IoT devices. The authors in [15] proposed a software design to aid IoT interoperability using a smartphone-based application. In [16], the presented routing technique selects routing paths between the sender and receiver depending on the node's speed, direction, and remaining energy. A trust-based opportunistic routing is presented in [17] which enhances the forwarding proficiency and transmission security of nodes in the OppNets. Zhu et al. [18] presented a miscreant node detection scheme named iTrust that inspects the cooperation among the nodes and a central Authority.

Of late many researchers are extensively inspecting the prospects of OppIoT in smart cities. The work in [29] evaluates the IoT and smart city trends of today and tomorrow. The authors explore the nature of interaction between smart cities and IoT and some of the factors that have influenced IoT development and smart city progress. The authors [30] divide a smart city divided into zones, and consider a group of a few terminals that are never of the same type The group consists of fixed terminals, two types of mobile nodes with differing equipment, pedestrians, and vehicles. In addition, the work interested gathers data from many sources, including the environment as well as perhaps other sources, using sensors, reading tags, or communications.

## 3 Proposed Work

The mobility factor of an OppIoT scenario makes the delivery of data unreliable because of poor connectivity and security threats posed by several attacks like blackhole [31], sybil [32]. The workflow of the proposed model (Figure 2) depicts the steps used in identifying a safe neighbouring node and then computing the suitability of the node for the transfer of data.

**Table 1** Comparative study of some OppIoT routing protocols

| S. No. | Name of Protocol | Aim | Methodology | User Parameters | Merits | Demerits |
|---|---|---|---|---|---|---|
| 1 | Epidemic [19] | The increasing delivery ratio in OppNets. | This protocol keeps a directory called a summary vector. When any two nodes that come together share summary vector and rearrange excluded messages. | Latency, HOP count, buffer space and overhead ratio. | Great Delivery ratio. | Creates a huge quantity of message redundancy in the network. |
| 2 | Spray and wait [11] | Limit the volume of message flooding. | The spray phase transmits data to a static number of neighbour nodes and waits to reach the message to the destination, in the waiting phase intermediate node will act as the source node suppose the information has not reached the end node. | Average Residual Energy, Average Buffer Time, Average HOP Count and Overhead ratio. | Less delivery delay and fewer transmission. | A larger value of L proceeds it like Epidemic protocol and a smaller value of L proceeds it like Direct delivery protocol. |
| 3 | PRoPHET [20] | Restrict the flooding of messages. | Delivery predictability is prepared according to the history of visits or history of encounters between nodes and a node will select a neighbour node to forward a message which is having higher delivery predictability. | Buffer Space, Overhead Ratio, Predictable Delivery and Latency. | Easy to implement and better performance. | Acts only on historical data, ignoring present node attributes such as battery life, reduced performance, and so on. |

(*Continued*)

**Table 1** Continued

| S. No. | Name of Protocol | Aim | Methodology | User Parameters | Merits | Demerits |
|---|---|---|---|---|---|---|
| 4 | Firefly PRoPHET [21] | Enhance the functionality of conventional PRoPHET. | The approach used a Firefly algorithm to configure intrinsic and extrinsic settings for the node that were inspired by natural phenomena. Using chaos maps to improve speed allowed for additional protocol optimization. | Buffer, Energy, Desirability, Transmission and Bandwidth Indices | This protocol outperforms PRoPHET in respect of delivery ratio when using fewer resources. | The delivery rates of machine learning-based algorithms are higher than this algorithm. |
| 5 | MLProph [22] | Enhance PRoPHET's performance. | It employs a combination of machine learning and neural network techniques to device the delivery predictability of a likely system to push a message. | Speed, buffer occupancy, message live time, and node energy. | Compared with PRoPHET shows high delivery rates. | The absence of generalization in their approach leads to not giving an optimal solution. |
| 6 | KNNR [23] | Using the KNN clustering technique to create node groups and then enabling routing in an opportunistic environment. | The KNN classifier is prepared using the node encounter history dataset in the training phase; in the application phase, the message is transferred to a relay node that is similar to the dataset and the routing process is maintained to shortlist the relay nodes on the way to destination. | Node speeds, hop counts, time-out ratio, buffer availability, and distance. | Uses Machine Learning techniques to get the best nodes and current node information helps it adapt to changing network scenarios quickly. | KNNR uses hard clustering to divide nodes into non-overlapping network topology partitions. |
| 7 | GMMR [24] | Employs machine learning-based clustering technique. | GMMR uses a soft clustering approach that allows clusters to overlap, allowing each node to have multiple salient features and be depicted by multiple groups at the same time. | Displacement from the goal node, Frequency-destination encounter, Capacity of the buffer, Success Rate. | A soft clustering technique is used, this is a step up from KNNR. | Due to the computational complexity of the training phase, the approach is less effective for dynamic networks. |

| | | | | | | |
|---|---|---|---|---|---|---|
| 8 | CAML [25] | Used Machine Learning oriented method to improve PRoPHET's performance. | CAML is a machine learning-based scheme for OppIoT networks that uses classifier cascades. It extracts relevant data from network nodes and then to calculate the likelihood of delivery, a pre-trained classifier is used. | Average hop count, network overhead ratio, message delivery probability and packets dropped | As compared with PRoPHET, CAML protocol improves its performance by adding an MLP Neural Network classifier. | Computationally heavy and impractical in cases where training must be more frequent. |
| 9 | RLProph [26] | Using the Policy Iteration method to enhance message delivery possibilities. | The OppIoT routing context is represented as a discrete stochastic method that defines the RL model elements of agent, status, reward, action, and environment. The optimum routing process is achieved through policy iteration in the resolution of the Markov [MDP] decision process. | The number of lost packets, message delivery probability, and network overhead ratio. | The routing process employs the optimal policy to make smart and effective forwarding decisions. | Any time the system inputs increase leads to handling a large number of inputs causing the system to slow down. |
| 10 | kROp [27] | To develop a context-aware routing protocol for dynamically making routing decisions. | Utilizes K-Means Clustering to form the best node cluster for message forwarding, evaluated by proximity to the ideal scenario. Uses Spray & Wait like message delivering. | The number of successful deliveries, the encounter with the destination, and the available buffer space. | Providing higher delivery ratios with small drops and average hop count. | High latency. Does not address energy consumption and security issues. |
| 11 | HiLSeR [28] | Routing Resource efficiency in opportunistic IoT network. | Hierarchical learning-based sectionalized routing paradigm is a method to group nodes for intelligent transmission and network topology sectionalization based on node properties. | Energy Unit per packet, Overhead Ratio, Average Latency, Dead node Percentage, and Success Ratio. | It combines a context-aware hierarchical learning paradigm to solve the critical problem of routing in Opp-IoT networks. | Does not address energy consumption and security issues. |

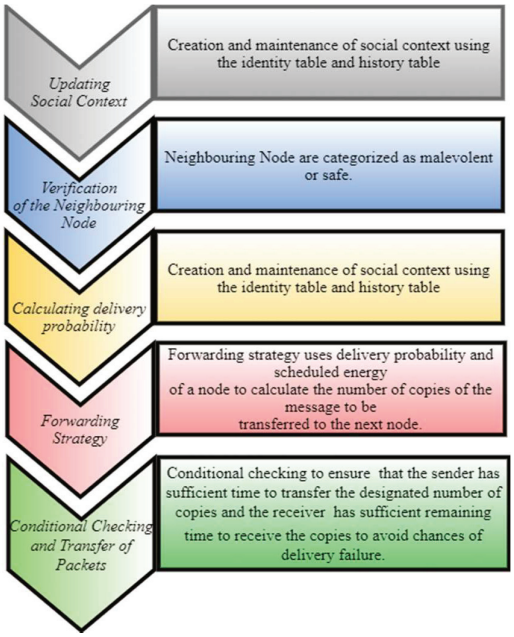


**Figure 2** Workflow of the proposed model.

### 3.1 Updating Social Context

The algorithm proposes the creation and maintenance of social context using the identity table and history table similar to the History Based Routing Protocol for Opportunistic Networks (HiBOp) [33]. These two main components can be briefly described as follows:

#### 3.1.1 Identity table

The Identity Table (Table 2) is used to provide both the current context of the user itself which consists of its information (e.g. System Information, Node ID, Personal Information) as well as the information about its current neighbours obtained during a set time interval called the neighbour discovery and exchange phase which takes place regularly in the system to keep updating information about recent encounters in both tables. This gives an overview of the node's local context at that time. Similar to HiBOp, this current context is used to evaluate the degree to which the node encountered is fit to onward the message to the end point, explained thoroughly in the steps stated further in the paper.

**Table 2** Sample of identity table

| **Personal Info** | |
|---|---|
| Name | ABC |
| Surname | K |
| Email | ABC@gmail.com |
| Phone | 9864324598 |
| NodeID | abc0001 |
| **Residence** | |
| Street | Ganesh Nagar |
| City | New Delhi |
| **Work** | |
| Street | Madhuban Road |
| City | New Delhi |
| Organization | XYZ |
| **Hobbies & Fun** | |
| Address | 438, GRD Nagar |
| City | New Delhi |
| Association | White Club |
| **System** | |
| MAC-Bluetooth | 00:11:22:33:FF:EE |
| MAC address | **00:00:5e:00:53:af** |
| IP-Address | 172.5.5.45 |

**Table 3** Sample of history table

| Aggregate | Class | $P_c$ | H | R |
|---|---|---|---|---|
| Pisa | City | ... | ... | ... |

### 3.1.2 History table

The History Table (Table 3), also known as the legacy of the node is used to record attributes and values of features that were seen in the past by the node in the identity table of its neighbours. It then uses these records to find similarities and trends in the movements of nodes to come to a probable metric that may describe its possibility of reaching the destination. All attributes add to the legacy of the node and are taken into consideration when delivering a probability value.

## 3.2 Verification of the Neighbouring Node

Once a neighbouring node is identified, the source node starts a timer. The transmitting node checks whether the selected node is present in the malevolent list. After that, the transmitter transmits a packet to the selected

node. If the source receives a reply from the selected node, the timer is reset and the selected node is marked as safe. Otherwise, the selected node's information is broadcasted by the source node. The nodes in the vicinity of the alleged node try to send and receive a reply from the selected node. If a reply is received by anyone, the node is then considered safe; otherwise, it is added to the list of malevolent nodes.

### 3.3 Calculating the Delivery Probability

The protocol evaluates the probabilities based on both identity and history tables. In the case of the Identity Table, the node matches the values of attributes in its own IT to attributes of the destination node and calculates $P_{IT}$ (using Equation (1)) assuming weighted values as follows:

$$P_{IT} = \frac{\sum_{j \in \{match\}} w_j}{\sum_{j \in \{dst\_info\}} w_j} \tag{1}$$

It matches the same for all its current neighbours and finds the maximum of all such values and sets it to $P_{cc}$.

Similarly, for the history table, using the HiBOp Formula, $P_H$ is calculated using Equation (2).

$$P_H = \max\{1, P'_H + \Delta_{max}[1 - e^{-(\hbar - 1)}]\} \tag{2}$$

where $P'_H$ is calculated using Equations (3a) and (3b).

$$P'_H = \frac{\sum_{j \in \{match\}} P_{op}^{(j)} \cdot w_j}{\sum_{j \in \{dst\_info\}} w_j} \tag{3a}$$

$$P_{op}^{(j)} = P_{cc}^{(j)} \cdot \frac{R^{(j)}}{r} \tag{3b}$$

Now, $P_{CC}$ and $P_H$ are used to come to a final delivery probability value in Equation (4).

$$P = \alpha \cdot P_H + (1 - \alpha) \cdot max\{\eta \cdot P_{cc}, P_{IT}\} \tag{4}$$

Where $\alpha$ is a smoothing factor that lies between 0 and 1 and the scaling parameter $\eta$ has a value greater than 0.

### 3.4 Forwarding Strategy

The algorithm consists of a forwarding strategy which uses delivery probability and scheduled energy of a node as parameters to calculate the number

of message copies to be transferred to the subsequent node. This number is a fraction of the number of copies left with the transferring node.

Delivery Probability: As discussed above, the delivery probability of the encountered node is calculated and compared with the probability of the transferring node.

Scheduled Energy: Scheduled Energy refers to the energy left of the node before it expires its Time To Live (TTL), in other words, denotes the energy left of the node that is usable and can effectively transfer the message from itself to the other.

Both delivery probability and scheduled energy are calculated for transferring node (P1, E1) as well as encountered node (P2, E2) and a metric of the comparison of the two is used to calculate the number of copies to be transferred. The value of the number of copies to be made can be calculated as follows:

$$N2 = \left( \frac{P2}{(P1 + P2)} + \frac{E2}{(E1 + E2)} \right) \times \frac{N1}{2} \tag{5}$$

### 3.5 Final Checking Using Mobility Awareness Before Spray

The time required for successful beaconing (delta) for a single message is equal to the sum of the average time required to set up or initiate the transfer process and the average time required to send one copy of the message. Then the status related to the time of the receiver and average pause time of the sender are evaluated to ensure that the sending node has sufficient time to transfer the proposed N2 number of copies and the receiver node has sufficient remaining time to receive the same number of copies to avoid chances of delivery failure.

The time taken to send N2 number of copies along with the initialisation and setup time is calculated as:

$$\Delta = (t_{setup} + t_{transfer}) \times N2 \tag{6}$$

where $t_{\mathrm{setup}}$ is the average time to set up or initiate the transfer process and $t_{\mathrm{transfer}}$ is the time required to successfully transfer one copy.

### 3.6 Algorithm: Secured and Mobility Aware Routing Technique (SMART)

**Step 1:** ***Begin***

**Step 2:** Every node initializes or update their Identity and History tables at the start of the Neighbour Discovery Phase.

**Step 3:** Until messages exist, the source node either creates a new message or selects the next message in its message queue. Make this message M.
**Step 4:** Repeat for each Source node's neighbouring node or any Intermediate node till the message gets to the destination node.
**Step 5:** When a node encounters a neighbouring node, a verification request message is sent. If the reply is not received within the threshold time, then the receiver is added to the suspect list and its IP is broadcasted.
**Step 6:** All the nodes receiving the message try to verify the suspect node and send a verification message to the suspect node.
**Step 7:** If none of the nodes receives a reply within the threshold time, the suspect is marked a malevolent node and added to the malevolent list.
**Step 8:** Else the node is removed from the suspect list.
**Step 9:** Now the node calculates delivery probability using Equation (4). Using the delivery probability and scheduled energies of both the sender (P1, E1) and receiver node (P2, E2), the no of copies, N2, to be sent to the receiver is evaluated using Equation (5).
**Step 10:** The remaining time of receiver (Rt) and average pause time of sender (Tp) is requested. The time taken to send the N2 number is calculated using Equation (6).
**Step 11:** Both times are compared with the amount of time needed to successfully transfer the copies (as calculated in Step 10).
**Step 12:** If (Rt$>\Delta$) and (Tp$>\Delta$)
**Step 13:** Then transfer takes place and N2 copies of message M are passed to the encountered node. Now sender node has (N1–N2) copies of the message left.
**Step 14:** Else
**Step 15:** The message is not passed to the encountered node.
**Step 16:** ***End***

## 4 Simulation Results

The projected model is implemented using ONE simulator [34]. The SMART model is compared to other existing models like Epidemic routing [19], Spray and Wait [11] and PRoPHET [20]. The proposed model's performance is evaluated with respect to three metrics namely delivery probability, overhead ratio and messages dropped. The metrics are evaluated under three

conditions, varying the simulation time, changing the TTL and increasing the buffer size.

The simulation parameters that are set during the experiment are as follows:

i. Group.movementModel is set to the 'ShortestPathMapBasedMovement'.
ii. Group.nrofHosts (No. of hosts) is set to '30'.
iii. Group.router is set to the proposed SMART model.
iv. Group.msgTtl denotes the time to live parameter and is varied during the simulation.
v. Group.bufferSize represents the buffer size is set to different values to evaluate its impact on the model.
vi. Scenario.endTime signifies the simulation time and is varied from 10800 sec to 43200 sec.

## 4.1 Simulation Time

The simulation time of the experimental setup varied between 10800 seconds to 43200 seconds. The delivery probability (Figure 3) demonstrates that the proposed model is capable of achieving a delivery probability of more than 0.28 and higher than the existing models. Figure 4 depicts the overhead

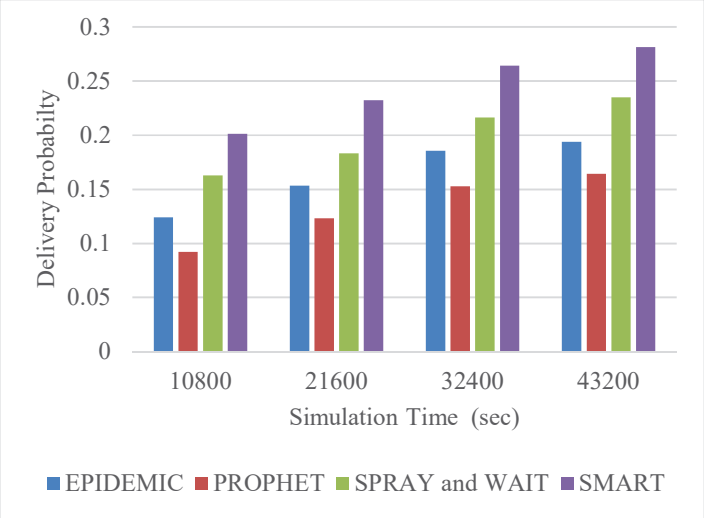


**Figure 3** Delivery probability with varying simulation time.

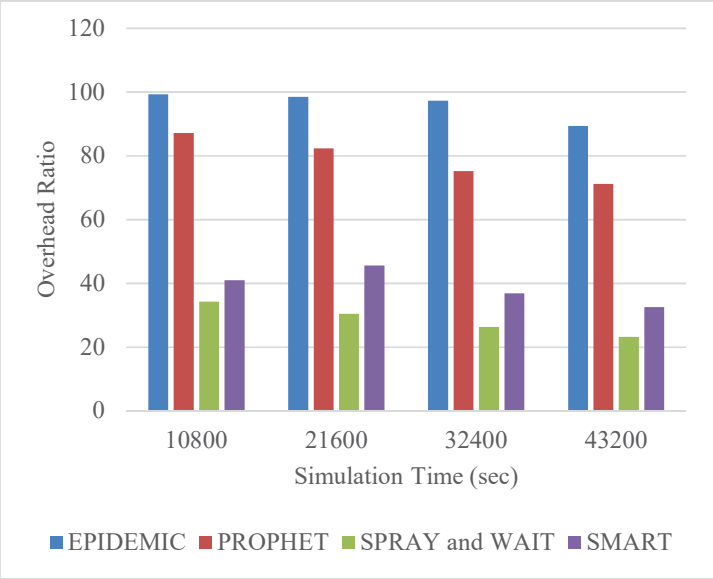


**Figure 4** Overhead ratio with varying simulation time.

ratio with varying simulation times. The overhead of the models is more than the SMART model. However, the overhead ratio of the Spray and Wait model is lesser than the proposed model. One primary reason is the proposed model identifies a malicious node to ensure better delivery probability. It is noticeable in Figure 5, that the number of messages dropped is insignificant (less than 5000) in the SMART model.

### 4.2 TTL

The TTL is varied in the following simulation phase to evaluate the effect of TTL on the performance metric. TTL varies between 100 and 300 minutes to compute the system performance. Figures 6 and 7 show that increasing TTL increases protocol delivery probability while also increasing overhead. However, as the TTL increases, the number of dropped messages decreases (Figure 8).

### 4.3 Buffer Size

In this segment, the parameters are analysed by varying the buffer size (10–30 MB). Figure 9 confirms the positive impact of an increase in buffer

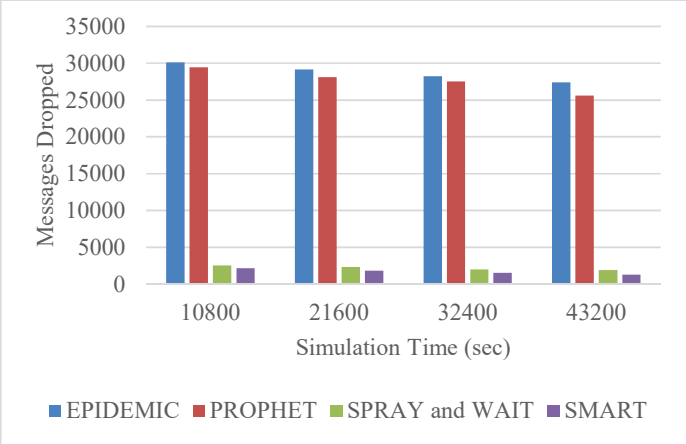


**Figure 5** Messages dropped with varying simulation time.

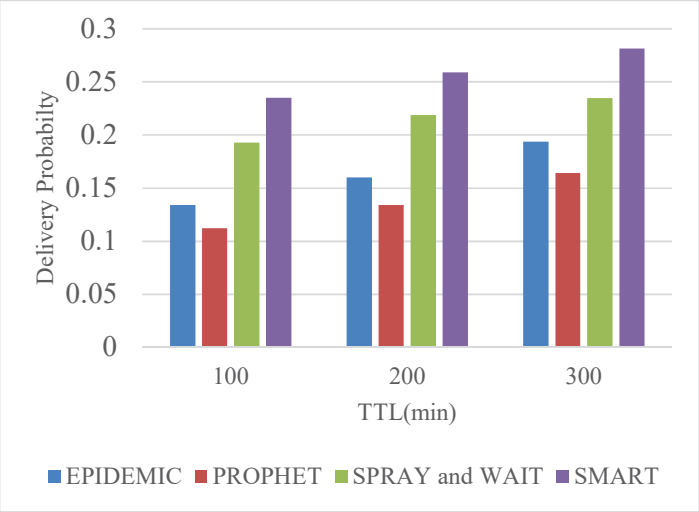


**Figure 6** Delivery probability with varying TTL.

size over the delivery probability. However, as the size of buffer is about 30 MB, the models behave persistently. Figure 10 indicates that enhancing the buffer size reduces the overhead ratio reported in Epidemic and Prophet. However, it almost has very little effect on the overhead ratio incurred by the SMART model. It is evident from Figure 11, that the increase in buffer size lowers the messages dropped.

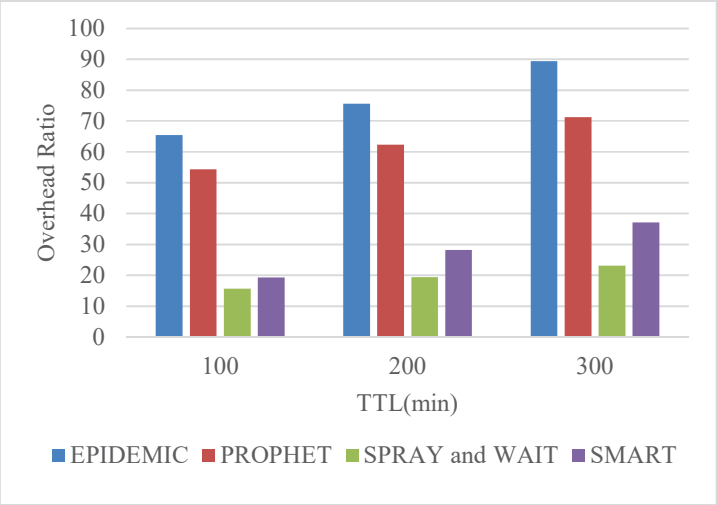


**Figure 7** Overhead ratio with varying TTL.

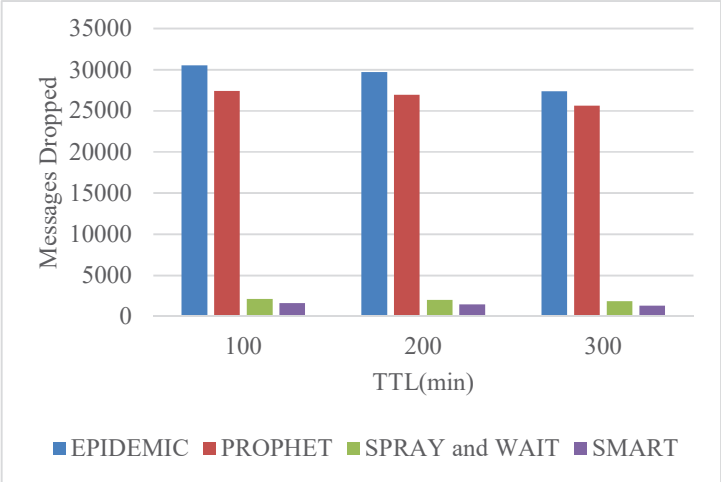


**Figure 8** Messages dropped with varying TTL.

## 4.4 Discussion

One crucial feature of opportunistic IoT is that it may be applied in situations where there is no network design, no understanding of the network topology, and haphazard device mobility models. Also, these characteristics of the network make routing a particularly difficult process. Some OppNets benchmark

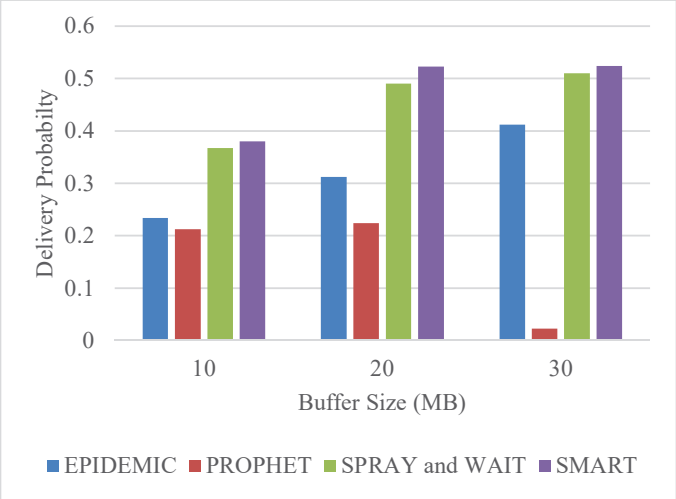


**Figure 9** Delivery probability with varying buffer size.

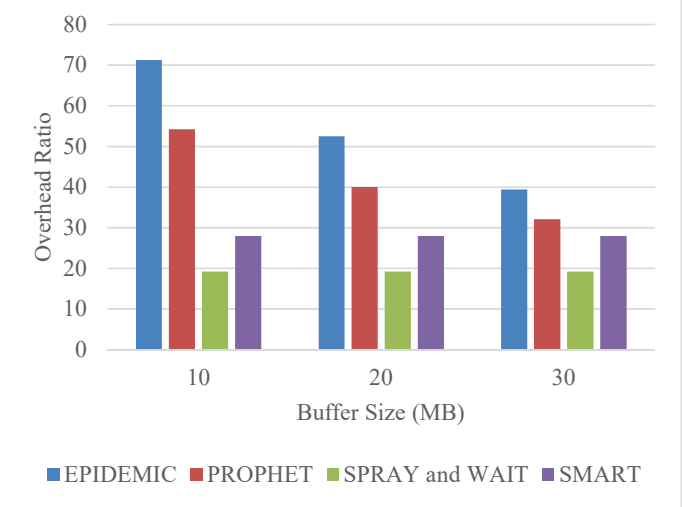


**Figure 10** Overhead ratio with varying buffer size.

routing methods are also relevant to OppIoT. Although the suggested SMART model has security characteristics and has improved network security, the model has a few drawbacks. As the overhead ratio is higher than the Spray and Wait approach, it should be reduced even more. When the simulation time is extended, this observation becomes more obvious.

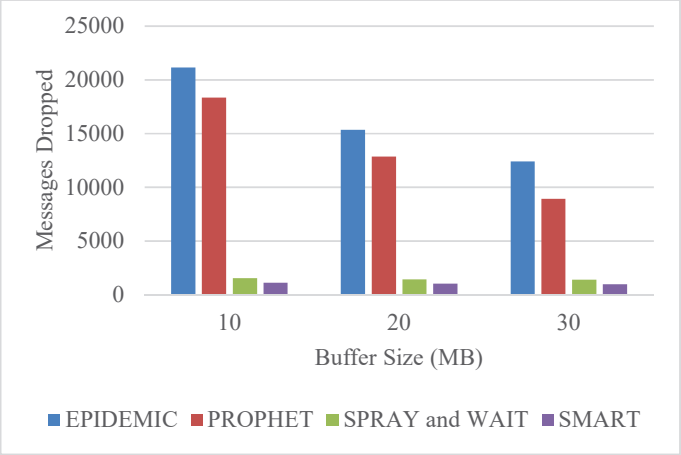


**Figure 11** Messages dropped with varying buffer size.

## 5 Conclusion and Future Work

A secured mobility-aware model has been proposed in this study for use in smart city applications. The simulation is run using the ONE simulator. The delivery probability, overhead ratio, and message lost characteristics all showed good performance for the suggested SMART model. Additionally, it made sure that the nodes chosen weren't malicious. The simulation time of 43200 seconds, the TTL of 300 minutes, and the buffer size of 30 MB provide the model with its best performance. The likelihood of delivery is greater than 0.52.

Although the suggested approach offers some security, it might be improved to offer better security and raise network quality. Message acknowledgements may be added to the SMART protocol in the future, which will allow it to be compared to other routing protocols presently in use in infrastructure-less OppIoT networks. The potential of the SMART protocol can be more extensively appraised by using other metrics like buffer availability, average delay, success ratio, and so on by changing other parameters such as TTL and node speed.

## References

[1] Atzori, L., Iera, A., Morabito, G., The Internet of Things: a survey, *Comput. Netw*., 2010, 54, (15), pp. 2787—2805.

[2] Pelusi, L., Passarella, A., Conti, M., Opportunistic Networking: Data forwarding in disconnected mobile ad hoc networks, IEEE *Commun. Mag*., 2006, 44, (11), pp. 134–141.

[3] H. C. Gao, X. J. Chen, D. Xu, Y. Peng, Z. Y. Tang, and D. Y. Fang, Balance of energy and delay opportunistic routing protocol for passive sensing network, *Journal of Software*, vol. 30, no. 8, pp. 2528–2544, 2019.

[4] H. D. Ma, P. Y. Yuan, and D. Zhao, Research progress on routing problem in mobile opportunistic networks, *Journal of Software*, vol. 26, no. 3, pp. 600–616, 2015.

[5] Y. Lu, W. Wang, L. Chen, Z. Zhang, and A. Huang, Opportunistic forwarding in energy harvesting mobile delay tolerant networks, *in Proceedings of the 2014 IEEE International Conference on Communications (ICC)*, pp. 526–531, Sydney, NSW, Australia, June 2014.

[6] A. Lohachab and A. Jangra, Opportunistic Internet of Things (IoT): Demystifying t he Effective Possibilities of Opportunistic Networks towards IoT, *in Proceedings of the 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 1100–1105, Noida, India, March 2019.

[7] V. Petrov, A. Samuylov, V. Begishev et al., Vehicle-based relay assistance for opportunistic crowdsensing over narrowband IoT (NB-IoT), *IEEE Internet of things Journal*, vol. 5, no. 5, pp. 3710–3723, 2018.

[8] M. Gharbieh, H. ElSawy, M. Emara, H.-C. Yang, and M.-S. Alouini, Grant-free opportunistic uplink transmission in wireless-powered IoT: a spatio-temporal model, *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 991–1006, 2021.

[9] Yugank, H.K., Sharma, R. and Gupta, S.H. An approach to analyse energy consumption of an IoT system. *Int. J. Inf. Tecnol.* 14, 2549–2558 (2022). https://doi.org/10.1007/s41870-022-00954-5.

[10] Bongale, A.M., Nirmala, C.R. and Bongale, A.M. Energy efficient intra-cluster data aggregation technique for wireless sensor network. *Int. J. Inf. Tecnol.* 14, 827–835 (2022). https://doi.org/10.1007/s41870-020-0 0419-7.

[11] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, *in Proceedings of the 2005 ACM Security and Communication Networks 11 SIGCOMM Workshop on Delay-Tolerant Networking*, pp. 252–259, New York, NY, USA, August 2005.

[12] Arora, U., Singh, N. IoT application modules placement in heterogeneous fog–cloud infrastructure. Int. j. inf. tecnol. 13, 1975–1982 (2021). https://doi.org/10.1007/s41870-021-00672-4.

[13] Guo B., Zhang D., Wang Z., Yu Z., Zhou X, Opportunistic IoT: Exploring the harmonious interaction between human and the Internet of Things,*J. Netw. Comput. Appl.* 2013, 36, 1531–1539. doi: 10.1016/j.jnca.2012.12.028.

[14] Rishiwal, V., Singh, O. Energy efficient emergency rescue scheme in wireless sensor networks. *Int. J. Inf. Tecnol.* 13, 1951–1958 (2021). https://doi.org/10.1007/s41870-020-00584-9.

[15] Aloi G.,Caliciuri G.,Fortino G.,Gravina R., Pace P., Russo W.,Savaglio C., Enabling IoT interoperability through opportunistic smartphone-based mobile gateways,*J. Netw. Comput. Appl.*, 2017, 81, 74–84. doi:10.1016/j.jnca.2016.10.013.

[16] Chalew Zeynu Sirmollo, Mekuanint Agegnehu Bitew, Mobility-Aware Routing Algorithm for Mobile Ad Hoc Networks, *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6672297, 12 pages, 2021.

[17] F. Li and Y. Si, Trust-based security routing decision method for opportunistic networks, *Journal of Software*, vol. 29, no. 9, pp. 2829–2843, 2018.

[18] Zhu, H., Du, S., Gao, Z., et al., A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks, *IEEE Trans. Parallel Distrib. Syst*., 2013, 25, (1), pp. 22–32.

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

[20] P. Sok, S. Tan and K. Kim, PRoPHET Routing Protocol Based on Neighbor Node Distance Using a Community Mobility Model in Delay Tolerant Networks, *2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, 2013.

[21] S. Banyal, K. Bhardwaj, and D. Sharma, Probabilistic routing protocol with firefly particle swarm optimisation for delay tolerant networks enhanced with chaos theory. *Int. J. Innovative Computing and Applications*, 12, 2, pp. 25–37, 2021.

[22] D. Sharma, S. Dhurandher, I. Woungang, R. Srivastava, A. Mohananey and J. Rodrigues, A Machine Learning-Based Protocol for Efficient

Routing in Opportunistic Networks, *IEEE Systems Journal*, vol. 12, no. 3, pp. 2207–2213, 2018.

[23] D. Sharma, Aayush, A. Sharma and J. Kumar, KNNR: K-nearest neighbour classification-based routing protocol for opportunistic networks, *Tenth International Conference on Contemporary Computing (IC3)*, 2017.

[24] V. Vashishth, A. Chhabra, D.K. Sharma, GMMR: A Gaussian mixture model based unsupervised machine learning approach for optimal routing in opportunistic IoT networks, *Comput. Commun.* 134 (2019) 138–148.

[25] V. Vashishth, A. Chhabra and D. Sharma, A Machine Learning Approach Using Classifier Cascades for Optimal Routing in Opportunistic Internet of Things Networks, *16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, Boston, MA, USA, 2019, pp. 1–9, 2019.

[26] D. Sharma, J. Rodrigues, V. Vashishth, A. Khanna and A. Chhabra, RLProph: A dynamic programming based reinforcement learning approach for optimal routing in opportunistic IoT networks, *Wireless Networks*, vol. 26, no. 6, pp. 4319–4338, 2020.

[27] Sharma DK, Dhurandher SK, Agarwal D, Arora K. kROp: k-means clustering based routing protocol for opportunistic networks. *J Ambient Intel Human Comput*. 2019;10(4):1289–1306.

[28] Banyal, Siddhant, Bharadwaj, Kartik, Sharma, Deepak, Khanna, Ashish and Rodrigues, Joel. HiLSeR: Hierarchical Learning-based Sectionalised Routing Paradigm for Pervasive Communication and Resource Efficiency in Opportunistic IoT Network. *Sustainable Computing: Informatics and Systems*. 30. 100508. doi:10.1016/j.suscom. 2021.100508.

[29] Badis, H., Rida, K., Sherali, Z., Achraf, F., Lyes, K,. Internet of Things (IoT) Technologies for Smart Cities, *IET Networks*, 2018, 7(1), pp. 1–13.

[30] Corrente, G., Random motion nodes empowering opportunistic networks for smart cities, *Internet of Things*, 2020, 11, pp. 100258.

[31] Pham, T.N.D., Yeo, C.K., Detecting colluding blackhole and greyhole attacks in delay tolerant networks, *IEEE Trans. Mob. Comput.*, 2016, 15(5), pp. 1116–1129.

[32] Douceur, J.R., The sybil attack. *Int. Workshop on Peer-to-Peer Systems, Cambridge, MA*, *USA*, 2002, pp. 251–260.

[33] C. Boldrini, M. Conti, J. Jacopini and A. Passarella, "HiBOp: a History Based Routing Protocol for Opportunistic Networks," *2007 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 2007, pp. 1–12, doi: 10.1109/WOWMOM.2007.4351716.

[34] Ari Keränen, Jörg Ott and TeemuKärkkäinen, The ONE simulator for DTN protocol evaluation, *Proceedings of the 2nd International Conference on simulation tools and techniques*, pp. 55, 2009.

## Biographies

**S. P. Ajith Kumar** received Master of Computer Application from University of Madras, Master of Philosophy in Computer Science from Alagappa University and Master degree in Computer Technology from Delhi Technological University, India respectively. He is currently working as an Associate Professor in the Computer Application Department, Bhai Parmanand DSEU Shakarpur Campus – II, Delhi, India. Also, he is a research scholar of Manav Rachna University, Faridabad, India. His research area includes Opportunistic Network, Sensor Network and Machine Learning.

**Hardeo Kumar Thakur** is working as an Associate Professor SCSET, Bennett University, Greater Noida, He has more than 15 years of teaching

and research experience in leading institutions of India. He has earned his Ph.D (Computer Engineering) from University of Delhi in 2017 in the field of data Analytics. Dr. Thakur has published 25 research papers in international journal of repute, 15 papers in international conferences and 2 Edited books. His current research interest are Data Mining, Dynamic Graph Mining, Machine Learning and Big Data analytics. He is an active referee for many international Journals and Conferences.

**Koyel Datta Gupta** received the bachelor's degree in Computer Engineering from University of Kalyani in 2003, the master's degree in Computer Technology from Jadavpur University, India in 2007, and the Philosophy of Doctorate degree from Jamia Milia Islamia University in 2015, respectively. She is currently working as an Associate Professor at the Department of Computer Engineering, in Maharaja Surajmal Institute of Technology (MSIT) (under the GGSIP University), India. Her research areas include Network Security, Digital Signal Processing, Pattern Recognition and Machine Learning.

**Deepak Kumar Sharma** is working as an Associate Professor in the Department of Information Technology, Indira Gandhi Delhi Technical University for Women (IGDTUW), Kashmere Gate, Delhi, India. He obtained his

Ph.D in Computer Engineering from University of Delhi, India in 2016. His research interests include opportunistic networks, wireless ad hoc and sensor networks, Software Defined Networks and IoT Networks. He has over 17 years of experience in Academics. He has published various research papers in reputed international journals like ETT Wiley, IEEE Systems Journal, IEEE IoT Journal, Computer Communication Elsevier, IJCS Wiley etc. and conferences of repute like IEEE AINA, GLOBECOM etc. He has also authored various book chapters in edited books of IET, Wiley, Springer, Elsevier etc. He has served as session chair in many conferences and is also a reviewer of various reputed journals like ETT Wiley, AIHC Springer, IJCS Wiley etc.