

# A Novel Mobility-Based Routing Protocol for Semi-Predictable Disruption Tolerant Networks

Saeid Iranmanesh<sup>1</sup> · Kwan-Wu Chin<sup>1</sup>

Received: 17 July 2014 / Accepted: 4 April 2015 / Published online: 14 April 2015  
© Springer Science+Business Media New York 2015

**Abstract** This letter considers routing in delay tolerant networks whereby nodes have semi-predictable mobility patterns within a time period. We propose a mobility-based routing protocol (MBRP) where nodes construct a space–time graph dynamically. As the space–time graph may be incomplete, MBRP presents a heuristic that evaluates encountered nodes based on their recorded mobility patterns in order to disseminate a finite number of bundle replicas. Simulation results, over a service quality metric comprising of delivery, delay and overhead, show that MBRP achieves up to 105 % improvement as compared to four well-known routing protocols. Finally, MBRP is capable of achieving 50 % of the performance attained by the optimal algorithm, whereby all nodes are preloaded with a space–time graph.

**Keywords** Delay tolerant networks · Vehicular communications · Mobility models · Space–time graph · Time evolving graph

## 1 Introduction

A form of Delay/Disruption Tolerant Networks (DTNs) are those constructed out of vehicles, and thus have scheduled contacts [1] or a space–time graph [2–5]. Thus, their schedule allows other nodes to determine suitable point of

contacts and critically, allow them to compute different paths that meet various criteria. For example, bundles can be delivered through routes with the minimum delay or hop count. Moreover, it is possible to determine the remaining time until a pair of nodes meets each other again. Also, contact duration can be computed, and thereby, allowing nodes to determine the amount of data that can be transferred in advance.

To date, the key assumption of past works that use a space–time graph assume nodes are pre-installed with the said graph. Consequently, their main research question is how to compute a suitable route that meets a given criterion over the space–time graph. However, in practice, nodes will have to gradually learn the mobility pattern of nodes upon each contact and update their space–time graph accordingly. For the space–time graph protocols described in [2, 4, 6–9], every node has a fixed mobility pattern for an unspecified time period, meaning the space–time graph is not dynamic. Hence, past works assume that the space–time graph is available in full at each node. Also, in both [8] and [10], all nodes are preloaded with a space–time graph and have a predictable mobility pattern, one that is repeated periodically or fixed for a given time period. This assumption, however, is unrealistic because nodes may have a different mobility pattern at different time periods, and are not likely to have a space–time graph upon system bootup. Moreover, the mobility patterns of nodes may have an expiration time. Although the nodes in [11] start with zero information and gradually learn the network topology, the employed routing algorithm will flood bundles throughout the network if a route is not present in the current space–time graph. This thus increases signalling overheads. Also, when a space–time graph is not complete, a detected route may not be optimal.

Henceforth, this paper presents a mobility based routing protocol (MBRP) whereby each node's trajectory or

---

✉ Kwan-Wu Chin  
kwanwu@uow.edu.au

Saeid Iranmanesh  
si175@uowmail.edu.au

<sup>1</sup> School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Northfields Ave, Wollongong, NSW 2522, Australia

mobility pattern has an expiration time. MBRP is the first hybrid routing protocol that uses a space–time graph and heuristics to route bundles. Also, when nodes have an incomplete space–time graph, unlike history based routing protocols [12] that rely on previous encounters to estimate future contacts, MBRP evaluates the reachability of encountered nodes based on recorded mobility patterns. Simulation results show that MBRP achieves up to 150 % improvement as compared to other well-known protocols such as EPIDEMIC, PROPHET, EBR, and MAXPROP.

The remainder of the paper is organized as follows. Section 2 reviews space–time routing protocols. Section 3 outlines the system model and key notations. Section 3 describes the problem. Section 4 presents the simulation set-up. This is then followed by obtained experimental results in Sect. 5. Finally, Sect. 6 concludes the paper.

## 2 System Model

Consider a DTN with  $v$  mobile nodes represented by the set  $N = \{n_1, \dots, n_v\}$ . Every node is equipped with a GPS unit and moves independently with a different speed and has a radio range of  $R$ . Nodes are assumed to have unlimited buffer. Also nodes have a semi-deterministic mobility pattern, meaning they visit a sequence of locations in a predictable manner for a given time period. The term “cycle” is used to denote a time period in which a node has a known mobility pattern. Consequently, if nodes repeat their cycle, we say they have a “periodic” mobility pattern. For example, a person may leave his/her home at 7:00 a.m., go to work and return home at 10 p.m. every day. He/she then repeats this routine every day; i.e., they have a cycle of 24 h. A node may also have “dynamic” mobility pattern where once a cycle finishes, a new with a different mobility pattern starts. As an example, the mobility pattern of a taxi changes depending on passengers. In this case, the taxi will set a new trajectory after picking up a new passenger. Nodes move on a grid with  $w \times w$  cells. Each cell size is  $2 \times R$ . This means if two nodes are located in a cell, they are in communication range of one another. Let  $\mathcal{C} = \{c_1, \dots, c_i, \dots, c_m\}$  be the set of all cells, where  $m = |\mathcal{C}| = w \times w$ . As an example, a DTN that is overlaid on a grid of size  $4 \times 4$  has 16 cells  $\mathcal{C} = \{c_1, \dots, c_{16}\}$ . Another key assumption is that time is discrete and it is divided into slots of equal length, denoted as  $t = \{1, \dots, T\}$ . Moreover, nodes are synchronized in time, which can be achieved via GPS. Based on the space and time information, every node  $a$  records its mobility pattern  $MP_a$  as a sequence of ordered pairs  $(c_i, t)$ , where  $(c_i, t)$  denotes cell  $i$  and time  $t$ . For example node  $a$  may have the following mobility pattern within five time slots  $t = 5$ ,  $MP_a = \{(c_5, 1), (c_4, 2), (c_6, 3), (c_2, 4), (c_1, 5)\}$ . Node  $a$  is called the

“owner” of  $MP_a$ . In addition, each mobility pattern of node  $i$  has an expiration time  $EXP_i$ . Let  $RT_i$  be the routing table of node  $n_i$ . The notation  $RT_i \cdot MP_a$  is used to denote the mobility pattern of node  $a$  in node  $i$ 's routing table. Also, let  $L(t)$  be the set of contacts at time slot  $t$ .

To capture node contacts at different points in time as well as represent the routing table maintained by nodes, a space time graph is used, denoted as  $G(t) = (N, L(t))$ , where  $t = \{1, \dots, T\}$ . There are two types of links in a space time graph: *spatial* and *temporal*. A spatial link is a directed link between two nodes if they meet each other at the same time  $t$ . For example,  $n_j$  has a spatial link to  $n_i$  in  $G(3)$  if  $n_j$  is located in the same cell as  $n_i$  at time slot 3. This means a bundle can only be forwarded from one node to the other through a spatial link. Temporal links on the other hand capture the connection of the same node  $n_j$  across the  $(t-1)$ -th and  $t$ -th time slots. Every node is connected to itself in every slot, implying it can carry a bundle over all time slots. Nodes are located in one of the cells over time; i.e.,  $MP_{n_3} = \{(c_3, 1), (c_4, 2), (v)\}$ .

## 3 The Problem

The main problem is how to forward bundles based on an incomplete routing table information while nodes are learning the space–time graph such that the delivery ratio is maximized and delay is minimized. If a source node generates a bundle for a given destination, it is faced with one of the following forwarding problems: (1) there is no route to a given destination. This means a source has to either wait until a route is available, which incurs delays that may exceed a bundle's expiration time, or (2) there is at least one route to the given destination. Here, a source needs to decide whether to use available routes, which may be sub-optimal or wait for a better route in the future that has less delay. Notice that the maximum performance is achieved when every node has a complete space–time graph, which they can then use to compute the optimal route to any destinations.

## 4 Mobility Based Routing Protocol (MBRP)

MBRP considers the trajectory of nodes and the time of last contact between owners in order to minimize delay and maximize delivery ratio concurrently. In addition, MBRP is a quota protocol that limits the number of replicas for each generated bundle. This reduces the number of relay nodes required to deliver bundles. MBRP consists of the following two routing phases: space–time and heuristic. Briefly, in the former phase, each node constructs a space–

time graph based on its recorded mobility pattern and contacts. Then, by applying a modified Dijkstra algorithm on the space–time graph, each node finds the fastest path. In the heuristic phase, nodes use recorded mobility patterns to predict subsequent contacts when their space–time graph is incomplete. Recall that a space–time graph is *incomplete* if a node’s space–time graph does not contain the mobility pattern of all nodes. Also, if at least one recorded mobility pattern expires, the space–time graph becomes incomplete.

Nodes maintain the following data structure. A node’s MP within a time period  $t$  is stored in a one dimensional array of size  $t$ . Every element  $i$  of the array indicates the geographical location of the node at time slot  $i$ . Each geographical location is assigned a unique integer number. Specifically, in a grid of size  $w \times w$  where the grid coordinates  $x$  and  $y$  are between 1 and  $w$ , the unique integer number of each cell is calculated as follows.

$$s(x, y) = (y - 1) \times w + x \quad (1)$$

The space–time graph can be represented by a three-dimensional matrix  $M$ . Each element  $(i, j, k)$  of matrix  $M$  represents the time of the  $k$ -th contact between nodes  $i$  and  $j$ . For example, if nodes  $i$  and  $j$  meet each other two times at  $t = 4$  and  $t = 10$ , matrix  $M$  is updated to  $M(i, j, 1) = 4$  and  $M(i, j, 2) = 10$ . Hence, the number of entries in matrix  $M$  is dependent of the number of contacts.

#### 4.1 Space–Time Phase

In this phase, each node uses the space–time graph constructed using learned mobility patterns from each contact to forward bundles via the fastest path. In order to find the fastest path from a source to a destination node, the source node assigns a cost  $l_i$  to every link  $i$  as follows

$$l_i = T_i - T_{i-1} \quad (2)$$

where  $T_i$  represents the time that the  $i$ -th link occurs in the path. For example, node  $S$  is connected to node  $A$  at  $t = 1$  and then node  $A$  is connected to node  $B$  at  $t = 4$ . In this case, assuming the current time is zero, the delay of the link is one, and the delay of the link between  $A$  and  $B$  is three. As a result, any bundles on the route from node  $S$  to  $B$  will take  $1 + 3 = 4$  time units. Formally, the cost of a route  $\omega$  is calculated as follows,

$$\text{Cost } t_\omega = \sum_{i=1}^{|L_\omega|} l_i \quad (3)$$

where  $|L_\omega|$  represents the number of links on path  $\omega$ . In order to store the cost of links, a three-dimensional matrix, called cost matrix (CM), is established where each element  $(i, j, k)$  represents the cost of the  $k$ -th contact between nodes  $i$  and  $j$ . Each discovered path may have a different cost. In

order to find the fastest path, nodes use a modified Dijkstra algorithm based on the proposed cost function. Algorithm 1 presents the pseudo-code used by nodes to find the fastest path towards a given destination. As mentioned, node  $i$  considers the recorded mobility patterns to find contacts (line 3). If a contact is detected, the time of contact is added to matrix  $M$  (line 4). Based on matrix  $M$  and the proposed cost function (See Eq. 2), a node determines the CM matrix (line 9). Then, CM and a bundle’s destination ID are fed into *Dijkstra()* in order to find the fastest path  $\mathcal{L}$  towards destination  $d$  (lines 12–13). Lastly, a single copy of bundle  $M_b$  is forwarded over route  $\mathcal{L}$  (line 14).

Algorithm 1: The space-time phase

---

**Input:**  $RT_i$

**Output:** the fastest path  $\mathcal{L}$

**Begin**

- 1- **FOR** every order pair  $X$  of recorded mobility patterns in  $RT_i$  **DO**
- 2-     **FOR** every pair of nodes  $j$  and  $k$  where  $MP_j$  and  $MP_k \in RT_i$  **DO**
- 3-         **IF**  $MP_j(X) = MP_k(X)$
- 4-              $M(j, k, q) \leftarrow X$ ;
- 5-         **ENDIF**
- 6-     **ENDFOR**
- 7- **ENDFOR**
- 8- **FOR** every link  $l_e$  that connects nodes  $j$  and  $k$  **DO**
- 9-      $CM(j, k, q) \leftarrow T_e - T_{e-1}$ ;
- 10- **ENDFOR**
- 11- **FOR** every buffered bundle  $M_b$  at node  $n_i$  **DO**
- 12-      $d \leftarrow M_b, \text{destination}$
- 13-      $\mathcal{L} \leftarrow \text{Dijkstra}(CM, d)$
- 14-     send  $(M_b, \mathcal{L})$
- 15- **ENDFOR**

**END**

---

#### 4.2 Heuristic Phase

The aim of this phase is to route bundles when the space time graph is incomplete. The main idea is to take advantage of knowing the number of ordered pairs to estimate the reachability of nodes. Accordingly, the main observation is that when an encountered node has a large number of ordered pairs, it will be a good bundle carrier. Suppose that node  $i$  has recorded  $MP_j$  at time  $t$ . In this case, node  $i$  will mark an ordered pair of a mobility pattern  $MP_j$  as “expired” in  $RT_i$  if the second element of  $MP_j$ , namely time, is less than or equal to  $t$ . Node  $i$  also marks the remaining ordered pairs of  $MP_j$  as “valid”, meaning their second element i.e., time, is greater than  $t$ . For example, in Fig. 1, when node  $A$  meets node  $B$  at  $t = 4$ , node  $A$  is not

aware of any new contacts that nodes  $C$  and  $F$  have had after  $t = 2$  and  $t = 1$  respectively. In this example, node  $C$  meets node  $E$  at  $t = 3$  but at  $t = 4$  node  $A$  will not be aware of this contact given that the said contact occurs after the last contact with node  $C$ . Hence, when nodes  $A$  evaluates node  $B$  based on the number of valid ordered pairs, there are eight valid ordered pairs in node  $B$ 's routing table. Also, there is one valid ordered pair in node  $A$ 's routing table. Suppose that node  $A$  sends a number of replicas to node  $B$ . Based on Scenario 1 (see Sect. 4), the bundle is delivered at  $t = 5$ . Based on the second scenario, when node  $B$  meets node  $C$  at  $t = 6$ , there is one valid ordered pair in node  $B$ 's routing table; i.e.,  $(c_{2,1}, 5)$  in  $MP_A$ . In contrast, node  $C$  finds  $MP_E$  in its routing table with has two valid ordered pairs:  $(c_{5,3}, 4)$  and  $(c_{5,3}, 5)$ .

In order to calculate the number of valid ordered pairs, every node  $i$  establishes a metric called "Contact Time" or  $CT_j$  for each encountered node  $j$ . This metric represents the last contact time between nodes  $i$  and  $j$ . For example, when nodes  $i$  and  $j$  meet each other, they set  $CT_j$  and  $CT_i$  to the contact time. In addition, they will also exchange  $MP_j$  and  $MP_i$ . Upon contact, both connected nodes count the number of valid ordered pairs that belong to nodes with periodic and dynamic mobility patterns. Specifically, in terms of periodic mobility pattern, node  $i$  counts the number of valid ordered pairs as follows,

$$PMP_i = \sum_{MP_k \in RT_i} (|MP_k| - RT_i \cdot CT_k) \tag{4}$$

where  $|MP_k|$  indicates the total number order pairs of node  $k$ 's mobility pattern and  $RT_i \cdot CT_k$  represents the last contact time that node  $i$  recorded for node  $k$ . In words, Eq. 4 counts the number of ordered pair of all periodic mobility patterns in node  $i$ 's routing table since their last Contact Time up to the time that the cycle finishes. Recall that a cycle is a time period in which a node has a known mobility pattern.

As nodes with a dynamic mobility pattern, e.g., taxis, set a new trajectory in each cycle, these nodes will have more valid order pairs as compared to a node with periodic mobility pattern. Hence, the number of valid order pairs in a dynamic mobility pattern is dependent on the summation of all its cycles' length, called  $CL$ . Specifically, the number of valid ordered pairs for the dynamic case at node  $i$  is calculated as follows,

$$DMP_i = \sum_{MP_k \in RT_i} (CL - RT_i \cdot CT_k) \tag{5}$$

In words, Eq. 5 counts the number of order pairs of all learned dynamic mobility patterns since their last Contact Time up to the time that the last cycle finishes. Here,  $CL$  is

assumed to be equal to the time when the last recorded mobility pattern expires. Based on Eqs. 4 and 5, the total number of valid order pairs,  $VOP_i$ , in the routing table of node  $i$  is computed as,

$$VOP_i = DMP_i + PMP_i \tag{6}$$

The next issue is forwarding of bundles. A sender node

specifies the number of replicas to be forwarded to an encountered node based on the ratio of the number of valid order pairs in its routing table and the encountered node's routing table. For two nodes  $a$  and  $b$ , for the  $i$ th bundle  $M_i$  that is headed to destination  $d$ , node  $a$  sends the following number of replicas to node  $b$ ,

$$m_i \times \frac{VOP_b}{VOP_b + VOP_a} \tag{7}$$

where  $m_i$  is the available number of replicas for the  $i$ th

bundle at node  $a$ . In words, using Eq. 7, node  $a$  compares the number of valid ordered pairs in its routing table and node  $b$ 's routing table. If  $VOP_a$  is smaller than  $VOP_b$ , node  $a$  does not need to keep a large number of replicas for itself. As a result, if node  $b$  has a larger number of valid ordered pairs, more replicas are forwarded to node  $b$ .

For example, assume node  $a$  has 10 replicas of a bundle  $M_1$  and meets node  $b$ . Furthermore, assume node  $a$  with  $VOP_a = 10$  and  $VOP_b = 90$ . Node  $a$  sends  $\frac{90}{90+10} = \frac{90}{100}$  of available replicas of  $M_1$  to node  $b$ . Therefore, node  $a$  forwards  $10 \times \frac{90}{100} = 9$  replicas of  $M_1$  to node  $b$ . Now assume  $VOP_a = 60$  and  $VOP_b = 10$ , then  $\frac{10}{10+90} = \frac{10}{100}$  of replicas of  $M_1$  to node  $b$ . In this case, node  $a$  forwards  $10 \times \frac{10}{100} = 1$  replica of  $M_1$  to node  $b$ .

Algorithm 2 presents the steps performed by the heuristic phase. The algorithm is executed by every node  $i$  whenever it encounters another node  $j$  (line 3). Node  $i$  calculates the ratio of  $VOP_i$  and  $VOP_j$  in order to forward a portion of a bundle's replicas to node  $j$  (line 5–6).

Algorithm 2: The heuristic phase

---

```

Input:  $RT_i$ 
1- FOR every encountered  $n_j$  DO
2-   FOR every buffered bundle  $M_b$  at node  $n_i$  DO
3-      $m_b \leftarrow M_b.numOfReplicas$ 
4-      $d \leftarrow M_b.destination$ 
5-      $m_{send} \leftarrow m_b \times \frac{VOP_j}{VOP_j + VOP_i}$ 
6-     send  $m_{send}$  replicas of  $M_b$  to  $n_j$ 
7-   ENDFOR
8- ENDFOR

```

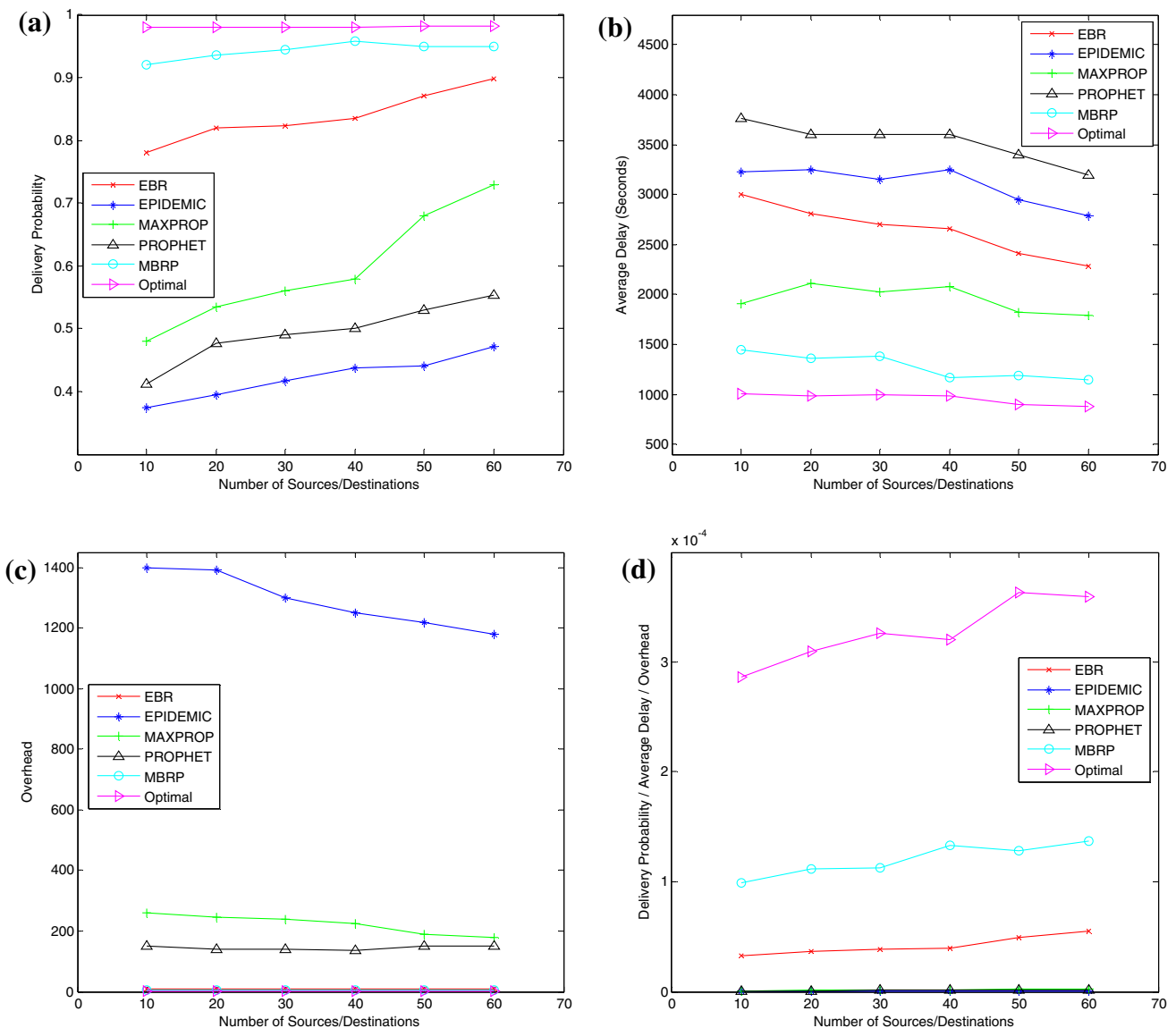
---

### 5 Evaluation

We have evaluated MBRP using the Java based simulator Opportunistic Network Environment (ONE) [13]. This simulator is able to generate vehicle movements using different mobility models [14–16] where nodes can have different cycle lengths. A deterministic network is created where nodes can have a periodic or dynamic mobility pattern in different cycles. MBRP is compared against four well-known protocols. Namely, EBR [17], EPIDEMIC [18], MAXPROP [19], PROPHET [20] and Optimal [2].

Nodes have a periodic mobility pattern and move in an area of approximately  $5 \times 3 \text{ km}^2$  in downtown Helsinki, Finland. Nodes repeat their pattern every 12 h. All experiments adopt ONE’s default settings, whereby 64 % of

nodes are pedestrians that move with a speed between 0.5 and 1.5 m/s. The other 32 % are vehicles that move with a speed ranging from 2.7 and 13.9 m/s. Other nodes are trams that move with a speed between 7 and 10 m/s. Note that all nodes have a fixed transmission range of 20 m and they also have a buffer size with a capacity of 20 bundles, except trams, where they have the capacity to store 500 bundles. In all experiments, the bundle size is 100 KB. All nodes have a transmission speed of 250 kbps except trams, which has a transmission speed of 10 Mbps. Note, we assume that trams have the ability to store more bundles and have a higher bandwidth as compared to other nodes. This is because, in reality, trams carry a large number of passengers and a large amount of data may be transmitted. Also, based on the benchmark setting of the ONE



**Fig. 1** Network performance when the number of sources and destinations is varied between 10 and 60, **a** delivery probability, **b** average delay, **c** overhead, **d** DAO

simulator, trams have a higher transmission speed and larger buffer space than other types of nodes. Each simulation lasts for three simulated cycles, i.e., 36 h, and each data point is an average of 10 runs. In all these experiments, the number of sources/destinations is varied from 10 to 60 in increment of 10.

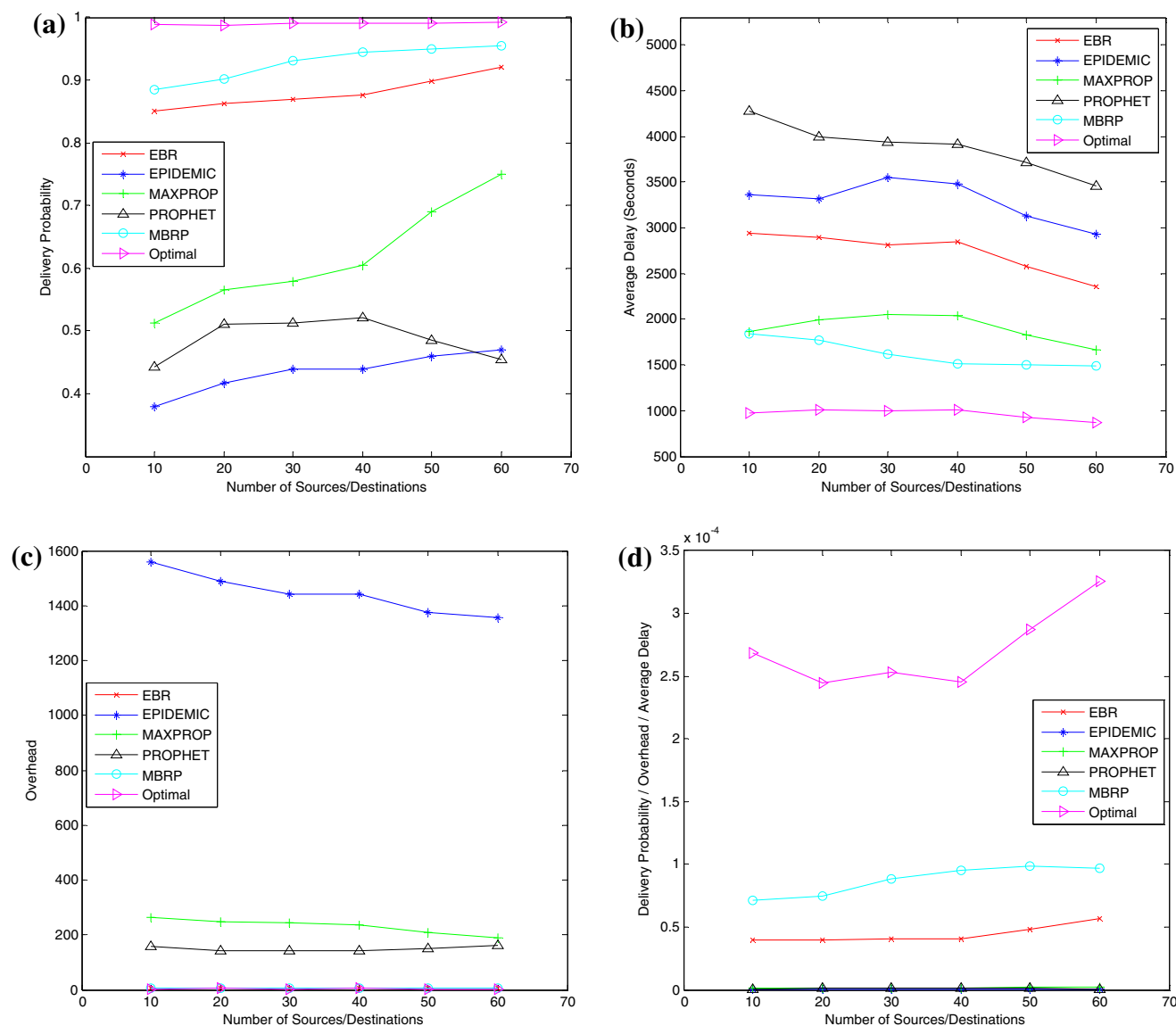
The routing protocols are evaluated using three well-known performance metrics, namely (1) *delivery probability*, (2) *overhead*, and (3) *average delay*. However, as mentioned in [17, 21], many protocols optimize one metric at the expense of another. For this reason, this work also uses a composite metrics namely, *DAO*; all of which are introduced by the authors of [17]. This composite metric provides a ratio between delivery probability and

conventional metrics. Specifically, Eq. (8) defines *DAO* that scales the performance of a protocol based on delivery probability (DP), average delay (AD) and overhead (OR).

$$DAO = DP \times \frac{1}{AD} \times \frac{1}{OR} \tag{8}$$

### 5.1 Periodic Mobility Patterns

Figure 1 shows the performance of a DTN where every node has a fixed mobility pattern for each cycle and contacts occur periodically. Nodes do not change their trajectory. This means the space–time graph will reach a steady state once nodes record all mobility patterns. Figure 1a shows that MBRP delivers up to 16 % more bundles

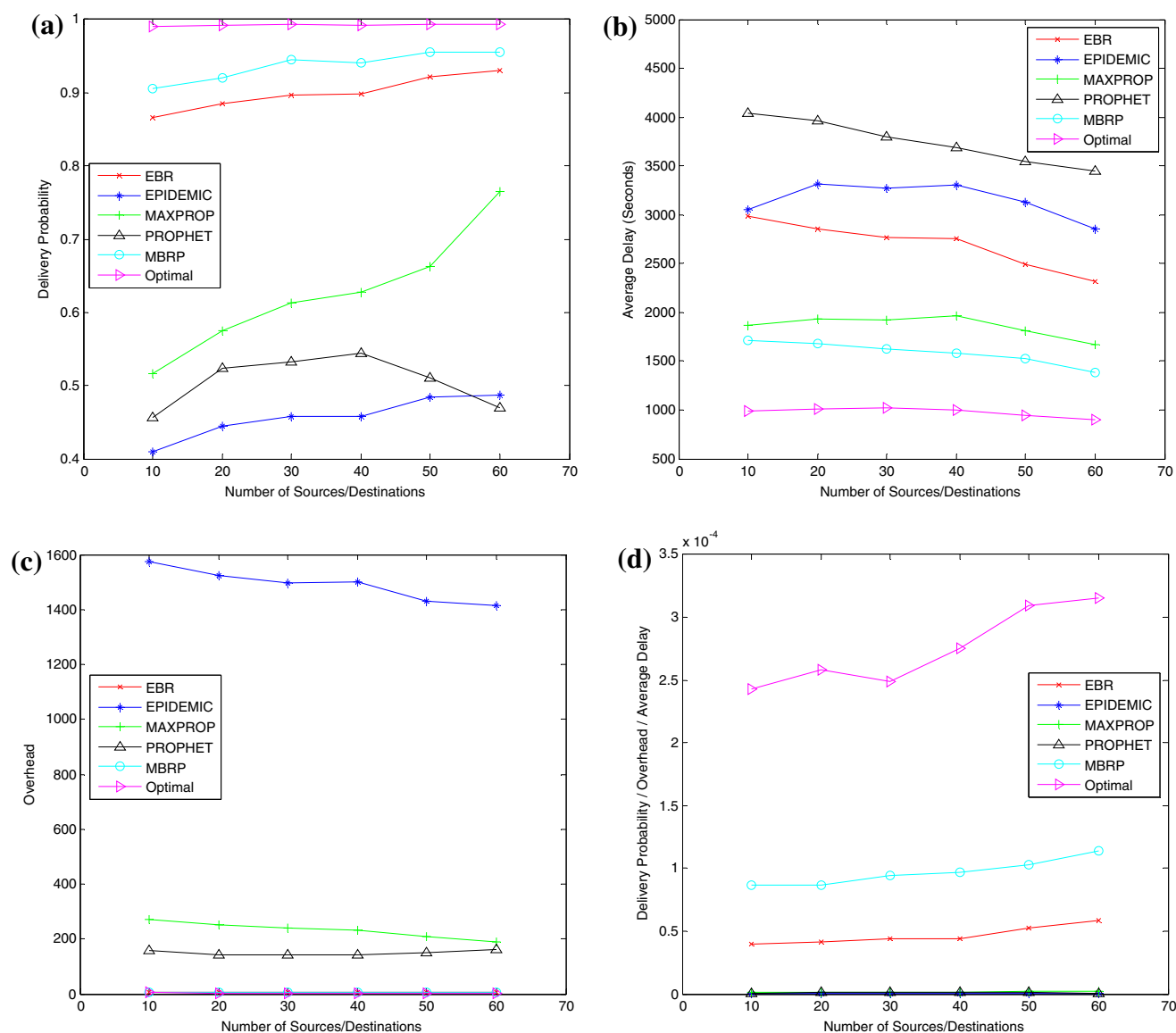


**Fig. 2** Network performance when the number of sources and destinations is varied between 10 and 60, **a** delivery probability, **b** average delay, **c** overhead, **d** DAO



as compared to EBR. This is because MBRP is guaranteed to deliver a bundle if a route is discovered. In addition, when there is no route towards a destination, MBRP estimates the future reachability of nodes to select a bundle's next hop. We see that EBR outperforms MAXPROP, PROPHET and EPIDEMIC. The reason is because EBR limits the number of replicas and hence, there are fewer number of dropped bundles as compared to flooding protocols. However, EBR may fail to deliver a bundle if the destination is located in a low density area. Figure 1a also shows that the Optimal protocol has up to 9 % improvement as compared to MBRP. This is because few nodes using MBRP may have an incomplete space–time graph.

In terms of delay, as shown in Fig. 1b, we see that MBRP delivers bundles up to 35 % quicker than MAXPROP. Recall that MBRP sends bundles via the fastest discovered path. Consequently, bundles are delivered on a path with much smaller delays as compared to MAXPROP, PROPHET, and EBR. In terms of overhead, Fig. 1c shows that MBRP and EBR use a small number of relays due to the finite number of replicas. This is because MBRP uses the space–time phase where only a single copy is forwarded and bundle is guaranteed to be delivered. This results a high delivery ratio and low overhead. Lastly, Fig. 1d shows that MBRP performs 150 % better than EBR in terms of *DAO*.



**Fig. 3** Network performance when the number of sources and destinations is varied between 10 and 60, **a** delivery probability, **b** average delay, **c** overhead, **d** DAO

## 5.2 Dynamic Mobility Patterns

In this set of experiments, a node changes its mobility pattern once it reaches a random POI. Figure 2a shows MBRP is up to 6 % better than EBR in terms of delivery ratio. Although nodes only have a valid mobility pattern for a given time period, the space–time phase may find a route towards a destination before their recorded mobility patterns expire. This causes MBRP to outperform EBR in terms of delivery ratio. As we can see from Fig. 2a, when the number of source/destination nodes increases, MBRP delivers up to 94 % of bundles. This is because when the number of source/destination nodes increases, the probability that a sender node has a destination’s mobility pattern increases. In other words, MBRP enters the space–time phase frequently. Figure 2b shows that MBRP reduces delays by up to 25 % as compared to MAXPROP. As mentioned, the space–time phase reduces delays as bundles are forwarded via the fastest discovered path. As shown in Fig. 2b, when the number of sources and destinations increases, due to the use of the space–time phase, the delivery delay of bundles decreases. In terms of overheads, Fig. 2c shows that MBRP incurs 14 % less resources usage as compared to EBR. This is because nodes using MBRP only forwards a single copy of each bundle. Finally, Fig. 2d depicts that MBRP performs up to 100 % better than EBR.

## 5.3 Mixed Mobility Patterns

Lastly, we consider the scenario where 20 % of nodes have a periodic mobility pattern and the remaining nodes have dynamic mobility patterns. Specifically, 20 % of the nodes have a fixed routing table. Figure 3a shows that compared to EBR, MBRP achieves 7 % improvement in delivery ratios. Also, MBRP’s performance is 5 % less than the optimal protocol. In terms of delay, Fig. 3b shows that MBRP delivers bundles up to 15 % quicker compared to MAXPROP. Figure 3c shows that MBRP consumes less resource as compared to PROPHET. This is because the number of replicas is limited in MBRP. Compared to EBR, Fig. 3c also shows that MBRP has 21 % reduction in overheads. This is due to its use of the space–time phase that forwards a single copy of bundles. Also, in terms of DAO, Fig. 3d shows that MBRP performs up to 105 % better than EBR.

## 6 Conclusion

This paper has proposed MBRP, a protocol that considers an incomplete space–time graph to send a single copy of each bundle over the fastest discovered path. In addition, as

initially the space–time graph may not be complete, they evaluate the reachability of encountered nodes based on the number of valid (remaining) order pairs of encountered nodes that are recorded in their routing table in order to send a proportional number of replicas to them. Simulation results, over a DTN comprising of nodes with dynamic and periodic mobility patterns show that compared to EBR, MBRP achieved up to 105 % improvement in a service quality metric called DAO which comprises of delivery, delay and overhead.

## References

1. M. Z. Araki and K.-W. Chin, TrainNet: A Novel Transport System for Delivering Non Real Time Video, *Elsevier Computer Communications*, Vol. 33, No. 15, pp. 1850–1863, 2010.
2. M. Huang, S. Chen, Y. Zhu, B. Xu and Y. Wang, *Topology Control for Time-Evolving and Predictable Delay-Tolerant Networks*, IEEE MASSValencia, 2011.
3. L. Huai-En, C. Chen, J. Chang, C. Shen, and R. Jan, “Shortest path routing with reliability requirement in delay tolerant networks”, *IEEE First International Conference on Future Information Networks (ICFIN)*, Beijing, China, 2009.
4. M. Huang, S. Chen, Y. Zhu and Y. Wang, *Cost-Efficient Topology Design Problem in Time-Evolving Delay-Tolerant Networks*, IEEE GLOBECOMMiami, FL, USA, 2010.
5. J. Leguay, T. Friedman, and V. Conon, “DTN routing in a mobility pattern space”, *Proceedings of the workshop on Delay-tolerant networking*, Philadelphia, Pennsylvania, USA, 2005.
6. S. Jain, K. Fall and R. Patra, Routing in a delay tolerant network, *ACM SIGCOMM Computer Communication Review*, Vol. 34, No. 4, pp. 145–158, 2004.
7. B. B. Xuan, A. Ferreira and A. Jarry, Computing Shortest, Fastest, and Foremost Journey in Dynamic Networks, *International Journal of Foundations of Computer Science*, Vol. 14, No. 2, pp. 267–285, 2003.
8. S. Merugu, M. Ammar, and E. Zegura, “Routing in Space and Time in Networks with Predictable Mobility”, Georgia Institute of Technology, Technical Report, GIT-CC-04-07, 2004.
9. D. Hay and P. Giaccone, “Optimal routing and scheduling for deterministic delay tolerant networks”, *IEEE Sixth International Conference on Wireless On-Demand Network Systems and Services*, Snowbird, UT, 2009.
10. C. Liu and J. Wu, Practical Routing in a Cyclic MobiSpace, *IEEE, Transactions on Networking*, Vol. 19, No. 2, pp. 369–382, 2011.
11. R. Handorean, C. Gill and G.-C. Roman, Accommodating Transient Connectivity in Ad Hoc and Mobile Settings, *Elsevier Pervasive Computing*, Vol. 3001, pp. 305–322, 2004.
12. Z. Zhang, Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges, *IEEE Communications Surveys & Tutorials*, Vol. 8, No. 1, pp. 24–37, 2006.
13. A. Keränen, J. Ott, and T. Kärkkäinen, “The ONE simulator for DTN protocol evaluation”, *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, Rome, Italy, 2009.
14. A. Keränen, and J. Ott, “Increasing Reality for DTN Protocol Simulations”, Helsinki University of Technology, Technical Report, July, 2007.



15. F. Bai, N. Sadagopan and A. Helmy, *IMPORTANT: a framework to systematically analyze the Impact of Mobility on Performance of Routing Protocols for Adhoc Networks*, IEEE INFOCOM San Francisco, CA, USA, 2003.
16. T. Camp, J. Boleng and V. Davies, A survey of mobility models for ad hoc network research, *Wireless Communications and Mobile Computing*, Vol. 2, No. 5, pp. 483–502, 2002.
17. S. C. Nelson, M. Bakht, R. Kravets and A. Harris, Encounter: based routing in DTNs, *ACM SIGMOBILE Mobile Computing Communications Review*, Vol. 13, No. 1, pp. 56–59, 2009.
18. A. Vahdat, D. Becker, “Epidemic routing for partially connected ad hoc networks”, Technical Report, CS-200006, Department of Computer Science, Duke University, Durham, NC, 2000.
19. J. Burgess, B. Gallaghe, D. Jensen, and B. N. Levine, “MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks”, *IEEE INFOCOM*, Barcelona, Spain, 2006.
20. A. Lindgren, A. Doria and O. Schelen, Probabilistic routing in intermittently connected networks, *ACM SIGMOBILE Mobile Computing Communications Review*, Vol. 7, No. 3, pp. 19–20, 2003.
21. S. Iranmanesh, R. Raad, and K-W Chin, “A Novel Destination-Based Routing Protocol (DBRP) in DTNs”, IEEE, The International Symposium on Communications and Information Technologies (ISCIT), pp. 325–330, QLD, Australia, 2012.



**Saeid Iranmanesh** was born in Kerman, Iran, in September 1984. He received the bachelor degree in Software Engineering and the master degree in Artificial Intelligence from Azad University, Iran, in 2007 and 2010 respectively. He is currently Ph.D. candidate in computer and electrical engineering at University of Wollongong, Australia. His research interests include machine learning, routing in wireless networks, optimization algorithms.



**Kwan-Wu Chin** obtained his Bachelor of Science with first class Honours from the Curtin University of Technology, Australia. He then pursued his Ph.D. at the same university, where he graduated with distinction and the vice-chancellor’s commendation. After obtaining his Ph.D., he joined Motorola Research Lab as a Senior Research Engineer, where he developed zero configuration home networking protocols and designed new medium access control protocols for wireless sensor networks. In 2004 he joined the University of Wollongong as a Senior Lecturer, and was subsequently promoted to Associate Professor in 2011. His current research areas include medium access control protocols for wireless networks, routing protocols for delay tolerant networks, RFID anti-collision protocols, and the application of network coding in wireless networks. To date, he holds four US patents, and has published more than 90 articles in numerous conferences and journals.