An Optimized Probabilistic Routing Protocol Based on Scheduling Mechanism for Delay Tolerant Network

Yuxin Mao[1](✉), Chenqian Zhou[1], and Jaime Lloret[2]

[1] School of Computer and Information Engineering,
Zhejiang Gongshang University, Hangzhou 310018, Zhejiang, China
maoyuxin@zjgsu.edu.cn

[2] Department of Communications, Universidad Politecnica de Valencia,
Valencia, Spain

Abstract. How to achieve efficient data transmission in delay tolerant network (DTN) is an important issue to many wireless or mobile systems. In this study, we propose a new routing protocol to improve delivery rate and optimize delivery delay with low overhead in DTN. We extend the classic PROPHET routing protocol by a scheduling policy to increase delivery rate and reduce delivery delay in DTN. We try to improve the performance of traditional PROPHET from both storage and transmission. Moreover, we evaluate the performance of the scheduling protocol by ONE Simulator. The simulation result illustrates that the proposed protocol is able to achieve better performance compared with the classic protocols like PROPHET and Epidemic in several key aspects.

Keywords: Scheduling mechanism · Routing · Delay Tolerant Network · ONE simulator

1 Introduction

The development of communication technologies and the wide spread of hand-held devices such as smart-phones and tablets have made unstructured wireless networks more and more common. These unstructured networks can be considered as a type of delay/disruption tolerant networks (DTN) [1, 2]. In DTN, when a node (source) intends to deliver messages to another node (destination) but there doesn't exist a fully connected path between them, messages are then forwarded to intermediate nodes. However, the intermediate nodes may not encounter other nodes during a given period of time. Due to the store-carry-forward mechanism, the uncertainty delivery paths, and the dynamic network topology, traditional routing protocols can not work well in this case. Recent years, numerous protocols were proposed to maximize delivery success, such as First Contact [3], Epidemic [4], Spray and Wait [5], PROPHET [6] etc. PROPHET (Probabilistic Routing Protocol using History of Encounters & Transitivity) is a typical routing protocol based on historical data. However, we notice that there exist some drawbacks with PROPHET [6]:

S. Guo et al. (Eds.): TridentCom 2016, LNICST 177, pp. 148–157, 2017.
DOI: 10.1007/978-3-319-49580-4_14

1. Some delivered messages form source may be damaged by some malicious intermediate nodes.
2. As there are too many intermediate nodes and hops on connected paths, messages are more likely to be aborted.
3. A number of messages may be dropped because of TTL and buffer congestion.

In order to overcome the above drawbacks, we propose a new routing protocol called Scheduling-PROPHET, which can be directly applied to DTN. The protocol is designed to increase the delivery rate, reduce the loss of messages, and improve the efficiency of information transmission in this work. The remainder of the manuscript is organized as follows: Sect. 2 presents the related work. In Sect. 3, we illustrate a new protocol to improve PROPHET routing. Section 4 presents the experimental setting and simulation results. Finally, we conclude the manuscript in Sect. 5.

2 Related Work

There are a series of routing protocols for DTN, each with its own characteristic and optimal goal, and they are suitable for different scenarios. For example, Epidemic [4] is a classic flooding-based protocol that exchanges messages in a simple way that seems like epidemic. This protocol utilizes the movement of nodes to make more connected paths, and increases the delivery rate by replicating messages. To improve routing performance, Lindgren A. et al. considered probabilistic routing and thus proposed PROPHET [6], which is a probabilistic routing protocol computing the probability of success by historical transmission information, and selecting decent intermediate nodes which are more likely (based on past history statistics) to form a reliable path to destination nodes.

In recent years, prediction encountering opportunity and node mobility are two key factors in many new routing protocol design. Many researchers focus on the node movement environment. They consider that the area of nodes' movement and encountering opportunity are closely related to the geographic position at particular time point. For this reason, Anna S. et al. in [7] and W.Z. Li et al. in [8] show the importance of nodes' mobility and distribution in city environment, and try to improve the routing performance. Similarly, Lebrun J. in [9] and Leguay J. in [10] consider that the physical location and mobility behavior of nodes are also common parameters. In [6, 11, 12], protocols used the history of encounters to measure transmission reliability. E. Bulut et al. introduce a new metric called CIT, which computes the average intermeeting time between two nodes to improve its performance [13]. Some protocols exploit two parameters to improve performance, such as the works in [14–16]. Moreover, some protocols make their decision of transmission by using game theory, such as [17].

3 Proposed Routing Protocol

We present an optimized routing protocol called Scheduling-PROPHET, which improves over PROPHET in buffering and transmission. In this section, we illustrate our assumptions and describe the protocol in detail.

3.1 Network Model

In order to formalize the operation of the Scheduling-PROPHET routing protocol, we first give a basic network model for illustrating the protocol as follows.

Nodes Each node is assumed to have a unique nonzero identifier *i*, which is bound to a public identity. We assume that each node has a finite buffer space for carrying messages originated from others, but an unlimited buffer for messages generated by itself. The core of our protocol is delivery predictability. When two nodes encounter each other within communication range, the delivery predictability will be calculated.

Transfer As nodes are not always stationary, we assume that the transfer opportunity is limited in both duration and bandwidth. When two nodes encounter, the number of messages they can transfer is confined. Under communication range, nodes are able to discover each other, and transmit messages to each other.

Storage As nodes receive messages from others, each node must manage its limited buffer space by assigning priority to messages and selecting messages to delete according to the protocol. There are only three cases that nodes will delete messages. Case 1: The messages have been transmitted to the destination, which sends ACK to all senders. Then, nodes will remove the messages from their buffer. Case 2: The messages do not have enough time to stay in buffer space. In our evaluations, the TTL of messages is set to 300 s. Case 3: The messages are with the lowest priority when the buffer space is full.

3.2 Protocol Definition

In order to reduce the impact of network failures on delivery predictability, we propose a new routing protocol called Scheduling-PROPHET. The core of Scheduling-PROPHET routing protocol is delivery predictability according to the encountering frequency among nodes.

We use the incremental average value of delivery predictabilities to make the change of delivery predictability more smooth. The calculation model for delivery predictability is given as follows. We establish a probabilistic value called delivery predictability, noted as $P(A,B) \in [0,1]$. It refers to the probability that node A encounters node B in the network. For each node, it is required to preserve a table to store the information about delivery predictability. When node A encounters node B, they will exchange their table information to determine whether to transfer messages or not. We set an initial value called $P_{init} \in (0,1]$, which is an initialization constant for each node.

When node A encounters node B, the delivery predictability is calculated by Eq. (1), where $\beta \in (0.5, 1)$.

$$P(A,B)_{new} = P(A,B)_{old} + [1 - P(A,B)_{old}] * \beta \tag{1}$$

After that, the delivery predictabilities of all nodes will be recalculated according to Eq. (2), and N denotes the set of all nodes.

$$P(i,j) = \frac{P(i,j)_{old}}{2}, (i,j \in N) \tag{2}$$

When node A encounters note B for the first time, the delivery predictability is calculated by Eq. (3).

$$P(A,B)_{new} = P_{init} \tag{3}$$

Therefore, P is directly related to the frequency by which they meet each other. According to Eq. (2), if two nodes can't encounter each other, the value of P will be much lower.

In the worst case, forwarding messages to intermediate nodes that rarely encounter the destination node will cause delivery failure. Therefore, we try to calculate the complete delivery predictability from current node to destination node. The complete delivery predictability is calculated by Eq. (4), where $S(i, i+1, \cdots, D)$ denotes the total predictability from current node i to destination node D.

$$S(i, i+1, \cdots, D) = \prod_{x=i}^{D-1} P(x, x+1) \tag{4}$$

From source node A to destination node B, we can get a series of values for $S(i, i+1, \cdots, D)$ by Eq. (4), and then we can choose an appropriate path with the highest value for $S(i, i+1, \cdots, D)$.

3.3 Transmitting Messages

To reduce the number of invalid messages in network and increase successful delivery rate of significant messages, we propose two scheduling mechanisms: Message Management Mechanism and Message-Transfer Mechanism.

(1) Message Management Mechanism: In order to make bandwidth more efficient and save buffer space, we set some criteria to coordinate the priority of in-transit messages. The core of the management involves dropping messages and assigning priority. All nodes in a network are notified by ACK and then remove delivered messages from their buffer. If the messages have lower delivery predictability, we assign lower priority to them in order to restrain their transmission. If the buffer space is full, messages with the lowest priority will be removed to make more room for forth-coming messages.

In this way, the routing protocol avoids generating copies for messages with low priority, and improves the utilization of network resources.

(2) Message-Transfer Mechanism: In Scheduling-PROPHET, when node I encounters node J, node I will transmit messages to node J, and the Messages-Transfer Mechanism is illustrated as follows: Firstly, transmit messages to node J which is the destination of those messages. Secondly, exchange the table which holds the delivery predictabilities of all nodes, and decide the transmission path. The determinate values are complete predictability ($S(i, i+1, \cdots, D)$), the amount of hops of path ($h(i, i+1, \cdots, D)$) and traffic load (tl_i).

The algorithm for the proposed routing protocol to explain the procedure is given below (Algorithm: Scheduling-PROPHET). When two nodes (A and B) encounter, they will exchange their tables with each other, and then Scheduling-PROPHET transmits messages by following the rules shown in the algorithm. Moreover, the notations we have mentioned in this algorithm are shown in Table 1 for convenience.

Table 1. Notations in algorithm.

Notation	Description
A	a node carrying messages(M)
B	a node encountered by A
M	the message node A intends to send
D	the destination of M
K	any other node that is not encountered by A
N	the set of all nodes
L_i	the connected path from source to destination
$S(L_i)$	the complete predictability of L_i

(1) The value of $P(A, B)$ is calculated by Eq. (1), and then all values of P are re-normalized by Eq. (2).

(2) Messages are transmitted to the encountered node based on its priority. Specifically, if node A discovers the destination (node D) of the messages (M), A will transmit M to D firstly. In addition to this, transmission is based on a ranked list of messages stored in nodes.

(3) The message receiver (node D) confirms that it has already received the correct message, and notifies acknowledgement character (ACK) to all senders.

(4) Messages that have not been transferred in the buffer space will be transmitted one by one according to the priority. Scheduling-PROPHET calculates the possible paths (L) from current sender to destination according to the historical delivery predictability. In our protocol, we denote the maximum amount of paths to compare as $\triangle\lambda$. In the meanwhile, each possible path is normalized by Eq. (4). For all paths, we rank them from high predictability to low based on the value of $S(i, i+1, \cdots, D)$. Scheduling-PROPHET selects the path with the highest predictability for transmission.

```
Algorithm: Scheduling-PROPHET
for each node $A$:
  when node $A$ encounters node $B$, they exchange their tables;
  $P(A,B)_{new} = P(A,B)_{old} + [1 - P(A,B)_{old}] * \beta$
end for;
for each node pair $i$ and $j$:
    $P(i,j) = \frac{P(i,j)_{old}}{2}$, $(i, j \in N)$
end for;
for each message $M$ of $A$:
 if($D == B$){
   $A$ forwards $M$ to $B$;
   $D$ notifies that $M$ has arrived and it can be deleted from all node buffers;
 }
 else{
   $A$ exchanges table with $B$ and gains the set of connected paths (denoted as $L$);
     if($L \neq NULL$){
       Arrange the paths in $L$ from high to low according to the value of $S$;
       Get the top three paths $L_1, L_2, L_3$ from $L$
         if($S(L_1) = S(L_2)$ □ $S(L_1) = S(L_2) = S(L_3)$){
           $A$ calculates the hops of $L_1, L_2, L_3$ respectively;
             if($h(L_1) = h(L_2)$ □ $h(L_1) = h(L_2) = h(L_3)$){
               $A$ compares the traffic load of next hop on $L_1, L_2, L_3$ respectively;
               $A$ forwards $M$ to the next hop with the least traffic load;
             }
         }
         else {
           $A$ forwards $M$ according to $L_1$;
         }
     }
     else {
       do nothing;
     }
 }
end for;
```

(5) We found that there will be more than one path with the highest predictability in real-life applications, but Scheduling-PROPHET forbids choosing two or more paths concurrently. Therefore, we attempt to give an additional factor, the hop count of path ($h(i, i+1, \cdots, D)$). For the paths with the same predictability, the hop count of path is calculated by Scheduling-PROPHET, and those paths are arranged in order according to their hop counts. Scheduling-PROPHET will choose the path with the fewest hop counts to transmit. In our protocol, we set the maximum amount of paths to compare to 3.

(6) Finally, in the real applications, traffic load is a significant factor for transmission efficiency. Thus, in our protocol, we take traffic load (tl_i) into account. For the

paths with the same hop count, traffic load which is kept by the nodes will be compared. Scheduling-PROPHET will chooses next node with the least traffic load to transmit.

Scheduling-PROPHET keeps choosing paths with the highest predictability of successful delivery, the fewest hop count and the least traffic load. In this way, we are able to choose the most efficient transmission path, to improve transmission efficiency and reduce the message drop rate.

4 Simulations and Results

4.1 Simulation Setting

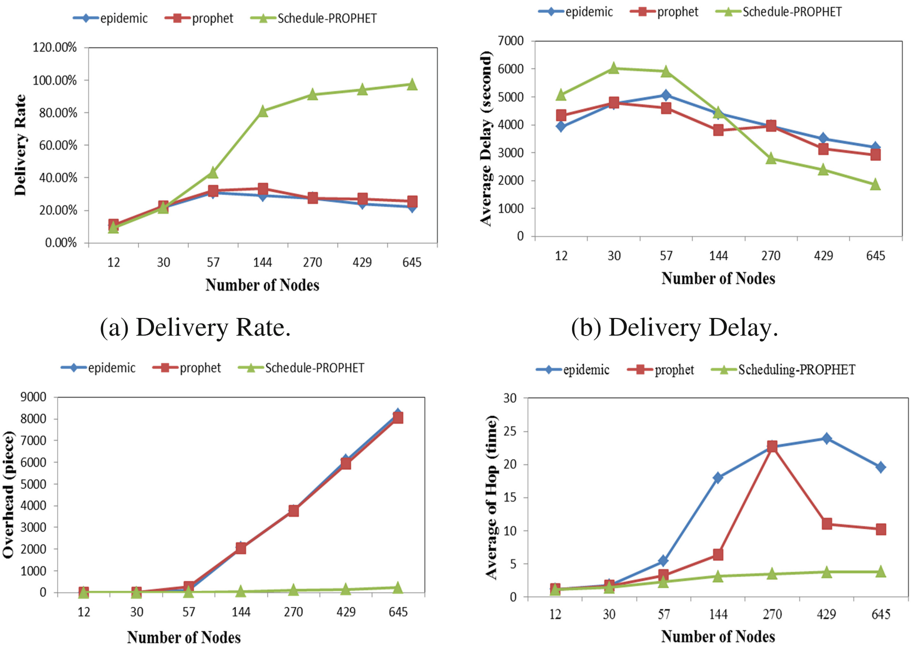
We implemented a test-bed based on ONE (Opportunistic Network Environment) simulator [18] to simulate the proposed routing protocol and evaluate its performance through a variety of simulations. We also compare the performance of Scheduling-PROPHET routing protocol with two classic protocols, Epidemic [4] and PROPHET [6] for DTN.

We have carried out a series of simulations in order to evaluate how the Scheduling-PROPHET routing protocol performs. The experimental scenario is confined to a 4500*3400 m^2 area of Helsinki city, which is included in the ONE by default. There are six groups of nodes in total. Group1 and Group3 are pedestrians groups, and their moving speed is about 0.5 to 1.5 m/s. Pedestrians follow the movement model called ShortestPathMapBasedMovement. Group2 is an automobile group and the nodes in this group can drive only on road, their moving speed is about 2.7 to 13.9 m/s. Group4, Group5 and Group6 are tram groups, and their moving speed is about 7 to 10 m/s, and they must drive on given roads by their movement model called MapRouteMovement. In our simulations, nodes transmit messages to others by using Bluetooth in 250 k/s. When Scheduling-PROPHET compares the possible paths, we have $\triangle\lambda \leq 10$.

4.2 Nodes Distribution

In order to evaluate the adaptability of our protocol in different network scales, we ran the simulation program for several times, with the density of the nodes varying from 10 to 700. The buffer size at Group1, 2, 3 remains as a constant of 5 M, and that of Group 4, 5, 6 is 50 M. The transmission range is a circle with the radius of 10 m. In our simulation, we mainly focus on evaluating the performance with four metrics: Delivery Ratio, Delivery Delay, Overhead and Hop Count.

Figure 1 shows the performance of the three protocols with different nodes distribution. Figure 1(a) illustrates the results of message delivery rate for the three protocols with different node density. There are no significant differences in performance for these three protocols with low density. However, with the density increasing, the proposed protocol has higher delivery ratio than others. The average delivery delay of the three protocols is given in Fig. 1(b). Delivery delay refers to the time of messages be transformed from source node to destination node, and we take the average delivery delay to evaluate the performance. It is easy to see that the proposed protocol performs much better than the other protocols at high density. Moreover, we find that two classic routing protocols perform better than ours at low density. That is mainly because they use simplex and direct mechanisms to transfer messages, and such direct mechanisms may perform efficient at low density, but the drawback will be exposed at high density. To make our protocol more efficient, we propose two improved mechanisms in Sect. 3.3. The overhead for these protocols is shown in Fig. 1(c). Overhead refers to the amount of messages have been transformed by each node in time unit. Figure 1(c) shows that our protocol gets the best performance in this simulation. This is mainly because that Scheduling-PROPHET takes the hop count into consideration, and chooses the shortest path to transmit messages while keeping a high delivery predictability. The result of hop count in Fig. 1(d) has depicts the similar truth with Fig. 1(c).



(a) Delivery Rate.

(b) Delivery Delay.

(c) Overhead.

(d) Hop Count.

Fig. 1. Performance comparison in different nodes distribution with four metrics: (a) Delivery Rate; (b) Delivery Delay; (c) Overhead; (d) Hop Counts.

5 Conclusions

In this paper, we proposed a new routing protocol called Scheduling-PROPHET for DTN. Scheduling-PROPHET is able to achieve high delivery rate and reduce delivery delay with low overhead in transmission. Moreover, the protocol can improve the management of messages in buffer. We also implement simulations to evaluate the performance of the proposed protocol and compare the protocol with classical routing protocols for DTN. There is also room for improvement, e.g. to evaluate Scheduling-PROPHET in different environments. We only simulated this protocol in one specific scenario, and therefore we can't guarantee that the protocol we used will be suitable for other situations in applications. In the future, we will focus on the performance of the proposed protocol and update it to satisfy different application environments.

Acknowledgments. This work is partially supported by Grants from the NSFC Program (NSFC61379121 and NSFC 61305148), the Science and Technology Department of Zhejiang Province Program (no. 2014C33097, 2014C33079, 2014C33099 and 2015C33062), the ZJNSF Program (no. LY14F020003).

References

1. Burleigh, S., Hooke, A., Torgerson, L., et al.: Delay-tolerant networking: an approach to interplanetary internet. IEEE Commun. Mag. **41**(6), 128–136 (2003)
2. Fall, K.: A delay-tolerant network architecture for challenged internets. In: SIGCOMM Proceedings of the Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, vol. 33(4), pp. 27–34. ACM (2003)
3. Jain, S., Fall, K., Patra, R.: Routing in a delay tolerant network. In: SIGCOMM Proceedings of the Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, vol. 34(4), pp. 145–158. ACM (2004)
4. Vahdat, A., Becker, D.: Epidemic Routing for Partially Connected Ad Hoc Networks. Technical report (2000)
5. Spyropoulos, T., Psounis, K., Raghavendra, C.S.: Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In: Proceeding of ACM SIGCOMM Workshop on Delay-Tolerant Networking, pp. 252–259 (2005)
6. Lindgren, A., Doria, A.: Probabilistic routing in intermittently connected networks. ACM SIGMOBILE Mob. Comput. Commun. Rev. **7**(3), 19–20 (2003)
7. Anna, S., Stavros, T.: Wireless mobile DTN routing with the extended minimum estimated expected delay protocol. Ad Hoc Netw. **42**(15), 47–60 (2016)
8. Li, W.Z., Lin, F., Zhou, J.L., Wang, Y.: DTN routing with fixed stations based on the geographic grid approach in an urban environment. Wirel. Pers. Commun. **82**(4), 2033–2049 (2015)
9. Lebrun, J., Chuah, C.N., Ghosal, D., Zhang, M.: Knowledge based opportunistic forwarding in vehicular wireless ad hoc networks. IEEE Veh. Technol. Conf. **4**(4), 2289–2293 (2005)
10. Leguay, J., Friedman, T., Conan, V.: Evaluating mobility pattern space routing for DTNs. In: IEEE 25th International Conference on Computer Communications, pp. 1–10 (2006)
11. Musolesi, M., Hailes, S., Mascolo, C.: Adaptive routing for intermittently connected mobile ad hoc networks. In: Proceeding of the Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks, pp. 183–189. IEEE Computer Society (2005)
12. Nelson, S.C., Bakht, M., Kravets, R.: Encounter-Based Routing in DTNs. Infocom IEEE **13** (1), 846–854 (2009)
13. Bulut, E., Geyik, S.C., Szymanski, B.K.: Utilizing correlated node mobility for efficient DTN routing. Pervasive Mob. Comput. **13**(4), 150–163 (2014)

14. Satya, B.S., Nanaji, U.: Efficient implementation of conditional shortest path routing in delay tolerant networks. Int. J. Comput. Sci. Commun. Netw. **1**(3), 252–257 (2011)
15. Burns, B., Brock, O., Levine, B.N.: MV routing and capacity building in disruption tolerant networks. In: 24th Annual Joint Conference of the IEEE Computer and Communications Socierties, vol. 1(4), pp. 398–408. Infocom IEEE (2005)
16. Spyropolous, T., Psounis, K., Raghavendra, C.S.: Single-copy routing in intermittently connected mobile networks. In: First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE Secon 2004, pp. 235–244 (2004)
17. Lenormant, F.: GTDM: a DTN routing on noncooperative game theory in a city environment. J. Sens. **2015**, 1–9 (2015)
18. Keränen, A., Ott, J., Kärkkäinen, T.: The ONE simulator for DTN protocol evaluation. In: Proceedings of the 2nd International Conference on Simulation Tools and Techniques (2009)