Improvement of Buffer Scheme for Delay Tolerant Networks

Jian Shen\textsuperscript{1,2}, Jin Wang\textsuperscript{1,2}, Li Ma\textsuperscript{1,2}, Ilyong Chung\textsuperscript{3}

\textsuperscript{1}Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science \& Technology, Nanjing, 210044, China
\textsuperscript{2}School of Computer \& Software, Nanjing University of Information Science \& Technology, Nanjing, 210044, China
\textsuperscript{3}Department of Computer Engineering, Chosun University, Gwangju, 501759, South Korea

E-mail: s_shenjian@126.com

Abstract. Delay Tolerant Networks (DTNs) are a class of emerging networks where disconnections may occur frequently. Delay is inevitable in DTNs, thus, making better use of buffer space to maximize the packet delivery rate is more important than concentrating on how to decrease the delay. In this paper, we improve the buffer scheme of Epidemic Routing for DTNs, which is a well-known routing protocol in DTNs. We improve the buffer scheme by re-arranging the drop sequence on the basis of the location and direction information of nodes. We hope that the improved buffer scheme is able to guarantee the high packet delivery ratio under small buffer size.

Keywords: Delay Tolerant Networks (DTNs), buffer scheme, packet delivery.

1 Introduction

DTNs are a practical class of emerging networks, which are an occasionally connected network comprised of one or more protocol families and experience frequent and long-duration partitions as well as long delay \cite{1}. The traditional view of a network as a connected graph over which end-to-end paths need to be established might not be appropriate for modeling existing and emerging wireless networks \cite{2}. Because there is no guarantee of end-to-end connectivity in DTNs, the routing protocols which have good performance in the conventional networks are not suitable for DTNs \cite{3}.

Lots of routing protocols in DTNs have been studied in \cite{4, 5, 6, 7, 8, 9, 10, 12, 13}. Among them, the first and most popular routing protocol in DTNs is Epidemic Routing \cite{4}, which disseminates a message replica to every node in the network. Epidemic Routing uses the simplest policy called first-in-first-out (FIFO) as its buffer scheme. This policy is simple to implement and bounds the amount of time that a

\textsuperscript{*} The corresponding author
particular message is likely to remain “live”. As long as the buffer size on all hosts is larger than the expected number of messages in transit at any given time, FIFO is a very reasonable policy.

In this paper, we improve the buffer scheme of Epidemic Routing by re-arranging the drop sequence. Compared with other proposed buffer management schemes in DTNs, the most distinguished difference in our paper is the improvement utilizes the location and moving direction information of nodes to determine the drop sequence. A node can get the location and moving direction information of other nodes by receiving beacon packets periodically from anchor nodes and referring to received signal strength indicator (RSSI) [11] for the beacon. In particular, the improved buffer scheme is able to guarantee the high packet delivery ratio under small buffer size.

The rest of this paper is organized as follows. In the following section, the related work on the buffer management scheme in DTNs is briefly discussed. The improved buffer scheme is described in detail in Section 3. Finally, the conclusions and future works are covered in Section 4.

2 Related Works

In DTNs, all of the node can become the carrier, which can take the message from one node to another. In this way, messages are quickly distributed through the networks due to the random mobility. Epidemic Routing relies upon carriers coming into contact with another node in the network by node mobility [4]. Using Epidemic Routing messages can be ensured that they have a high probability of the transmitting. Meanwhile the resource of network is consumed heavily. Thus, the objective of Epidemic Routing is to maximize the delivery rate, while minimize the transmit latency and the consumption of the resources.

The critical resource in Epidemic Routing is the buffer. Epidemic Routing uses FIFO as its buffer scheme. This policy is simple to implement and bounds the amount of time that a particular message is likely to remain “live”. Once enough new messages have been introduced into the system, older messages are likely to be flushed from most buffers. As long as the buffer size on all hosts is larger than the expected number of messages in transit at any given time, FIFO is a very reasonable policy. Hence, epidemic routing is very sensitive to the buffer size and can only achieve good performance with respect to infinite buffer size. In this paper, we assume that a node should not discard its own valid messages to create places in its buffer for accommodating new messages forwarded by other nodes. If all buffered messages are source ones, and the arriving message is also a source message, then we choose to delete the oldest one. This intuitive idea of giving priority to source messages has been proposed in [13] and was shown to improve the average delivery rate. The improved buffer scheme utilizes the location and moving direction information of nodes to determine the drop sequence. By the way, a node can get the location and moving direction information of other nodes by receiving beacon packets periodically from anchor nodes and referring to received signal strength indicator (RSSI) for the beacon.
3 The Improved Buffer Scheme

3.1 Key idea

We make use of anchor nodes to estimate the location and moving direction information of the nodes. Depending on this information, we choose the drop sequence. During this process, some priority information (e.g. the validity of the message, the security of the message, transmission speed request, the value of information, the cost of the message, the distance to the destination and the direction to the destination) is employed to decide which message should be dropped or be transferred to other nodes having available buffer space.

Fig. 1. An example of calculating node’s moving direction by using location information.

General nodes obtain the location information by making use of RSSI technique. As we known, in RSSI, one general node wanted to estimate its location should at least connect with three anchor nodes so as to calculate the location by trilateration. Moreover, by utilizing the location information, the general node can easily calculate its moving direction information. For example, seen from Fig. 1, the location of node X in time $T_1$ and $T_2$ are $(x_1, y_1)$ and $(x_2, y_2)$, respectively, then we define the moving direction of node X is:

$$\theta = \arctan \left( \frac{y_2 - y_1}{x_2 - x_1} \right).$$

We also can calculate the moving speed of node X:

$$S = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{T_2 - T_1}.$$ 

Observed from Fig. 1, we note that the message transmission direction is $\alpha$. If $|\theta - \alpha| \leq 45^\circ$, then we think the node’s moving direction is same with the message’s transmission direction.

Every node stores its own location and moving direction information. In order to minimize the communication overhead, however, all the information will not be exchanged with each other unless they are required between the nodes. Moreover, when an anchor node is situated in the transmission range of a certain general node, the information of this general node can be stored in this anchor node. For simply description, we can also say that this anchor node lists this general node. After some time interval, yet, the anchor node should update its list so as to re-obtain the latest
location and moving direction information of the general nodes, whose radio transmission range covers this anchor node.

3.1 Improved buffer scheme in detail

For explaining the improved buffer scheme in detail, an explicit example is given in Fig. 2. We assume that node S is the source and node D is the destination. Node A is supposed to be the best next hop determined by the transmission scheme described above. Nodes B, C, E, and F are in the transmission range of node A but not in the range of node S. There is a message which intends to be transmitted from node S to node A.

![Fig. 2. An example in drop scheme.](image)

```c
1. If (p_max = all of the p_max) (A node for “infant” reply to E. Return.)
2. If A receives the transmission and sees that A’s buffer is available again.
3. If (p_min = front of the p_min) If one message with the lowest priority in node A is dropped or allocated to other node
4. {
   A broadcasts the “buffer available” request to all its neighbors.
5. If (there is one neighbor node X whose buffer space is available and the available size is more than 1/2 of its total size)
6.   {
      Node X sends reply to node A;
7.      The message with the lowest priority in node A is sent to node X;
8.      Node A sends “permit” reply to node X;
9.      The new message is sent from node S to node A. Return.
10. }
11. }
12. 
13. (If there is more than one neighbor node whose buffer space is available and the available size is more than 1/2 of its total size)
14. If we assume that node B and node E satisfy this condition
15. {
   Node A broadcasts the “moving direction” request to them;
16. If (there is one node V whose moving direction is same with the message’s direction)
17. If (we assume that node V satisfy this condition)
18. {
      Node V sends reply to node A;
19.      The message with the lowest priority in node A is sent to node V;
20.      Node A sends “permit” reply to node V;
21.      The new message is sent from node S to node A. Return;
22. }
23. }
24. 
25. (If all of them have the same moving direction with the message’s direction)
26. {
      The message with the lowest priority in node A is sent to one of them randomly,
27.      Node A sends “permit” reply to node E;
28.      The new message is sent from node S to node A. Return;
29. }
30. }  
31. 
32. (If all the neighbors’ buffer spaces are not available or the available buffer space are all less than 1/2 of buffer space)
33. {
      The message with the lowest priority in node A will be dropped,
34.      Node A sends “permit” reply to node E;
35.      The new message is sent from node S to node A. Return;
36. }
```

![Fig. 3. Priority comparing algorithm in drop scheme.](image)
First, node S sends a “transmission” request to node A. After receiving the request, node A checks its buffer space to determine whether or not the buffer space is available. Node A replies to node S and permits the transmission only if node A’s buffer space is not full. Then node S sends the new message to node A. However, if node A’s buffer space is full, node A still replies to node S and only permits to accept the priority information of this message sent from node S. Then node A compares this priority information with others which have already stored in node A’s buffer space.

Fig. 4 explains the detailed steps of the priority comparing algorithm in drop scheme. Here, p_{new} represents the priority of the new message needed to be delivered to the destination, on the contrary, p_{old} delegates the priority of the old message stored in node A. If p_{new} is lower than all the p_{olds}, then node A will refuse this new message to be transmitted from node S to node A or permit it until node A’s buffer is available again. In this condition, hence, node A sends the "refuse" reply to node S (line 1-2 in Fig. 4).

However, if p_{new} is higher than some of p_{olds}, then one message with the lowest priority in node A will be dropped or be allocated to other available buffer space in other nodes in order to make space for storing this new message (line 3). First, node A broadcasts the "buffer available" request to all its neighbors to do the judgment which node’s buffer space is available. If there is one neighbor node whose buffer space is available and the available size is more than 1/2 of its total size. Then this node replies to node A and allows accepting the lowest priority message sent from node A. In this case, the new message can be transferred from node S to node A successfully (line 5-12). In our example according to Fig. 2, we suppose that node B’s buffer is available and the available size is more than 1/2 of its total size, while node C’s buffer is also available, however C’s available size is less than 1/2 of its total size. In addition, node E has an empty buffer space that can be available completely. To the contrary, node F’s buffer is not available completely. Hence, nodes B and E reply to node A, while nodes F and C keep silent. This is easy to be explained because only node B or E having enough available buffer space can accept this lowest priority message. Secondly, node A broadcasts the "moving direction" request to nodes B and E. We assume that node B’s moving direction is opposite with the message’s direction while node E’s is same. Therefore, indubitability, node E replies to node A and permits to accept the lowest priority message sent from node A. In this case, node A can make room for the new message from node S. Finally, node A replies to S and permits the new message to be delivered (line 13-24). Of course, there exists another case. When node B and node E have the same moving direction with the message’s direction, the message with the lowest priority in node A will be sent to one of them randomly (line 25-31).

In the worst case, additionally, if all the neighbors’ buffer spaces are not available or their available buffer spaces are all less than 1/2 of their buffer spaces respectively, then the message with the lowest priority in node A will be dropped so that node A can accept the new message (Line 32-37).

After the message successfully being transmitted from node S to node A, according to all the messages’ priority, node A continues deciding the next hop of the highest priority message until this message arriving at the destination by using the same drop scheme.
4 Conclusions and Future works

In this paper, we have investigated buffer management scheme in Epidemic Routing, where a simple policy FIFO is employed to increase the packet delivery rate. However, this buffer scheme exhausts the buffer size hastily. Therefore, we improve the buffer management scheme in order to enhance the performance of Epidemic Routing, which utilizes the location and moving direction of nodes to determine the drop sequence. We have shown that a node can get the location and moving direction of other nodes by receiving beacon packets periodically from anchor nodes and referring to received signal strength indicator (RSSI) for the beacon. On the other hand, the improved buffer scheme takes advantages of the nodes’ information of the location and moving direction to store the message into buffer space.

We hope that the improved buffer scheme is able to ensure good performance with a higher packet delivery rate and lower routing overhead. In the future, we will use the simulation tool such as NS-2 to test the performance of the improved buffer scheme.

Acknowledgement. This work was supported by the research fund from Nanjing University of Information Science and Technology under Grant No. S8113003001, National Science Foundation of China under Grant No. 61272421, and Natural science fund for colleges and universities in Jiangsu Province under Grant No. 12KJB520006, and by Natural Science Foundation of Jiangsu Province (No. BK2012461).

References

Improvement of Buffer Scheme for Delay Tolerant Networks

11. Received signal strength indication http://en.wikipedia.org/wiki/Received_Signal_Strength_Indication.