International Journal of Future Generation Communication and Networking
Vol. 6, No. 4, August, 2013

An Improved Buffer Scheme in Delay Tolerant Networks

Jian Shen*[1,2,], Wenying Zheng[3], Jin Wang[1,2], Yuhui Zheng[1,2] and Yuan Mei[1,2]

[1]*Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science &Technology, Nanjing, 210044, China*
[2]*School of Computer & Software, Nanjing University of Information Science &Technology, Nanjing, 210044, China*
[3]*School of Applied Meteorology, Nanjing University of Information Science &Technology, Nanjing, 210044, China*

Abstract

Delay Tolerant Networks (DTNs) are a class of emerging networks where disconnections may occur frequently due to propagation phenomena, node mobility, and power outages. 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, which is a well-known routing protocol in DTNs. Epidemic Routing utilizes the FIFO buffer management scheme to manage the queuing. However, Epidemic Routing is still very sensitive to the buffer size. Epidemic Routing, furthermore, can only achieve good performance with respect to infinite buffer size due to that a node may have enough space to store lots of messages in its buffer and carry them along for long periods of times until appropriate forwarding opportunities arise. Once there is not enough buffer size, then the performance of Epidemic Routing will fall down hastily. We propose a new buffer scheme in Epidemic Routing, named Location and Direction Aware Drop Scheme (LDDS). LDDS 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) for the beacon. In particular, LDDS is able to guarantee the high packet delivery ratio under small buffer size.

Keywords: *Delay tolerant networks; buffer scheme; received signal strength indicator*

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 [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. Due to wireless propagation phenomena, node mobility, low power nodes periodically shutting down, *etc.*, connectivity in many wireless networks is, more often than not, intermittent [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. To enable some services to operate even under these challenging conditions, store-carry-and-forward protocols are proposed, where a

---

[*] The corresponding author

node may store a message in its buffer and carry it along for long periods of time, until it can forward it further. This routing may happen randomly, be based on statistical information [3], or even other relevant information about the destination (*e.g.*, social links, affiliation, *etc.*,). Furthermore, due to the inherent uncertainty caused by the lack of complete (or any) information about other nodes in the network, many replicas of the same message may be propagated to increase probability of delivery.

Lots of routing protocols in DTNs have been studied in [4-10]. Among them, the first and most popular routing protocol in DTNs is Epidemic Routing [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 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.

In this paper, we improve the buffer scheme of Epidemic Routing, which utilizes the FIFO buffer management scheme to manage the queuing. However, Epidemic Routing can only achieve good performance with respect to infinite buffer size due to that a node may have enough space to store lots of messages in its buffer and carry them along for long periods of times until appropriate forwarding opportunities arise. Once there is not enough buffer size, then the performance of Epidemic Routing will fall down hastily. We proposed a new buffer scheme in Epidemic Routing, named Location and Direction Aware Drop Scheme (LDDS). Compared with other proposed buffer management schemes in DTNs, the most distinguished difference in LDDS is that LDDS 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, LDDS 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. Location and Direction Aware Drop Scheme (LDDS) is described in detail in Section 3. Performance evaluation is presented in Section 4. Finally, the conclusions of this paper are covered in Section 5.

2. Related Works

Epidemic Routing [4], as the name suggests, likes the pattern of pandemic virus transmitting. 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. Of course Epidemic Routing relies upon carriers coming into contact with another node in the network by node mobility. We assume that: (i) the sender does not know where the receiver is currently located or the best "route" to follow, (ii) the receiver may also be a roaming wireless host, and (iii) pairs of hosts (not necessarily the sender and receiver) periodically and randomly come into communication range of one another through 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. For solving this problem as much as possible, the objective of Epidemic Routing is to maximize the delivery rate, while minimize the transmit latency and the consumption of the resources.

For explicitly explaining the process of Epidemic Routing, we give an example as depicted in Figure 1.When host A comes into transmission range of host B, an anti-entropy session is initiated. In the first step, A transmits it summary vector (called SVA) to B, SVA is a compact representation of all the messages which are buffered at A. Next, B performs a logical AND operation between the negation of its summary vector (given a symbol like ┐ SVB) and SVA. We can easily get the conclusion the negation of B's summary vector representing the messages that it has never been seen. It implies, B finds the different vector, which B wants to need, compared with A's summary vector. And then transmits a vector requesting these messages from A. In the third step, A transmits the messages to B which are requested by B. This process is repeated transitively when B comes into contact with a new neighbor. Given sufficient buffer space and time, these anti-entropy sessions guarantee the message can be eventually delivered to the destination.

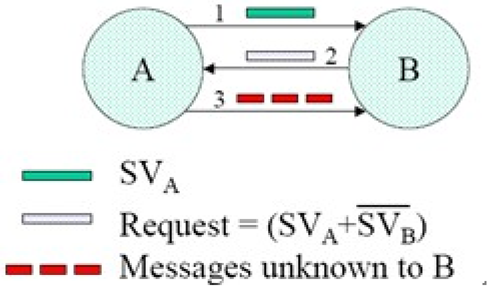


Figure 1. The Process of Epidemic Routing Protocol

The critical resource in Epidemic Routing is the buffer. 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 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. An intelligent buffer management scheme can improve the delivery ratio of Epidemic Routing over LDDS buffer management scheme, which 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. Location and Direction Aware Drop Scheme

3.1. Key Idea in LDDS

Before describing Location and Direction Aware Drop Scheme (LDDS) in detail, we briefly present the key idea about it. 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.

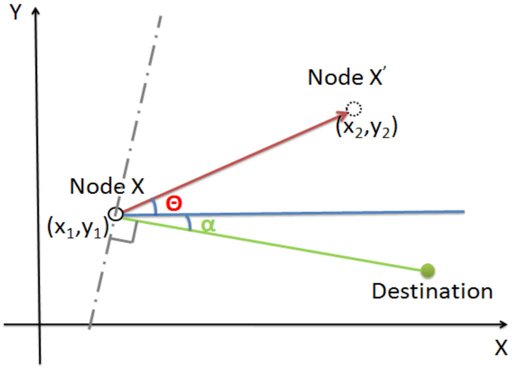


Figure 2. An Example of Calculating Node's Moving Direction by using Location Information

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 Figure 2, 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: $\theta = \arctan \frac{y_2 - y_1}{x_2 - x_1}$. We also can calculate the moving speed of node X: $S_X = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{T_2 - T_1}$. Observed from Figure 2, we note that the message transmission direction is α. If $|\theta - \alpha| \leq 45°$, 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.

3.2. LDDS in Detail

For explaining LDDS in detail, an explicit example is given in Figure 3. 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.

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.

Figure 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_{old}s, 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 Figure 4). However, if p_{new} is higher than some of p_{old}s, 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 Figure 3, 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 in LDDS as described above.

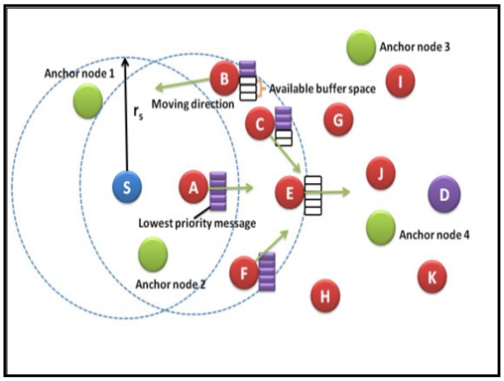
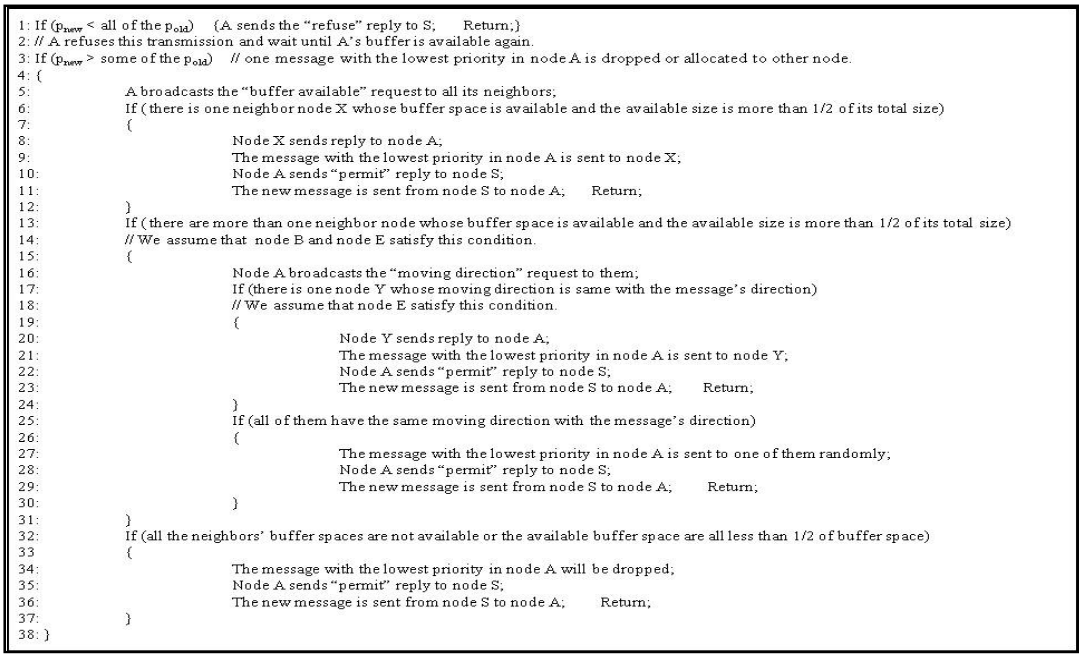


Figure 3. An Example in Drop Scheme



```
If ($p_{new}$ < all of the $p_{old}$)    {A sends the "refuse" reply to S;    Return;}
// A refuses this transmission and wait until A's buffer is available again.
If ($p_{new}$ > some of the $p_{old}$)    // one message with the lowest priority in node A is dropped or allocated to other node.
{
        A broadcasts the "buffer available" request to all its neighbors;
        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)
        {
                Node X sends reply to node A;
                The message with the lowest priority in node A is sent to node X;
                Node A sends "permit" reply to node S;
                The new message is sent from node S to node A;    Return;
        }
        If ( there are more than one neighbor node whose buffer space is available and the available size is more than 1/2 of its total size)
        // We assume that node B and node E satisfy this condition.
        {
                Node A broadcasts the "moving direction" request to them;
                If (there is one node Y whose moving direction is same with the message's direction)
                // We assume that node E satisfy this condition.
                {
                        Node Y sends reply to node A;
                        The message with the lowest priority in node A is sent to node Y;
                        Node A sends "permit" reply to node S;
                        The new message is sent from node S to node A;    Return;
                }
                If (all of them have the same moving direction with the message's direction)
                {
                        The message with the lowest priority in node A is sent to one of them randomly;
                        Node A sends "permit" reply to node S;
                        The new message is sent from node S to node A;    Return;
                }
        }
        If (all the neighbors' buffer spaces are not available or the available buffer space are all less than 1/2 of buffer space)
        {
                The message with the lowest priority in node A will be dropped;
                Node A sends "permit" reply to node S;
                The new message is sent from node S to node A;    Return;
        }
}
```

Figure 4. Priority Comparing Algorithm in Drop Scheme

4. Performance Evaluation

4.1. Simulation Environment

We implemented LDDS by using the ns-2 simulator. Unless otherwise noted, our simulations are run with the following parameters. We model 20, 50, 100 and 150 mobile nodes moving in a square area 1000 m x 1000 m in dimension during the simulation time 1000s. Each node picks a random spot in the square and moves there with a speed uniformly distributed between 0~5 m/s. The radio transmission range is assumed to be 250 meters and a two-ray ground propagation channel is assumed. Most other parameters use ns-2 defaults.

Due to space restrictions, we have focused on comparing the performance of the scheme with regards to three metrics: packet delivery ratio, packet delivery end-to-end delay and normalized routing overhead. We ran simulations for each scenario.

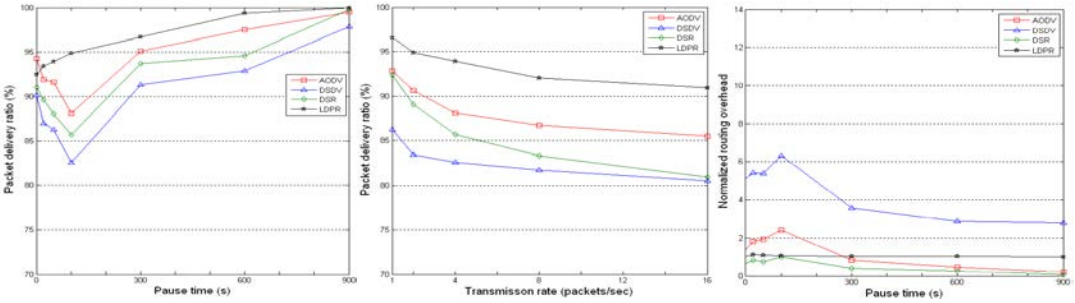
4.1. Results and Discussion

We present a comparative simulation analysis of LDDS with AODV, DSDV and DSR. The number of nodes is set to be 50 and the buffer size of each node is assumed to be 50.

The first interesting aspect that we analyze is the packet delivery ratio. As shown in Figure 5(a), the packet delivery ratio gradually increases as the pause time increasing. In particular, the packet delivery ratio of DSDV, DSR and AODV has a transient decrease when the pause time is 100s because of the unstable path. However, the curve of LDDS shows that the packet delivery ratio persistently increases as the pause time rising up. Figure 5(b) describes the change of packet delivery ratio as the packet outgoing rate increasing. Intuitively, the packet delivery ratio, under a certain buffer size, will reduce as the packet outgoing rate raising. If the packet outgoing rate exceeds the maximum accommodation capability of buffer, it must lead to some packets drop during the transmission. Here, LDDS performs the highest packet delivery ratio under various packet outgoing rates.

Another critical aspect we investigated is the normalized routing overhead. As we know, normalized routing overhead indicates the system resource utilization and consumption. It is an important criterion to estimate the performance of routing protocols. In these simulations, it is easy to tell that LDDS brings out lower routing overhead compared with other routing protocols and the curve of LDDS looks like more stable than other routing protocols'. Figure 5(c) shows that the routing overhead decreases with the pause time increasing. In particular, the change of LDDS's curve is small. Meanwhile, the routing overhead of LDDS is lower than AODV and DSDV when the pause time is less than 300s. However, when the pause time is more than 300s, the routing overhead of LDDS is a little higher than AODV and DSR. That's because the simulation environment trends to be static and the mobility of node declines so as to improve the performance of AODV and DSR and decrease the routing overhead of AODV and DSR. Figure 5(d) represents that the routing overhead increases as the packet outgoing rate increasing. Additionally, seen from the shape of the LDDS's curve, the routing overhead of LDDS always maintains a low level.

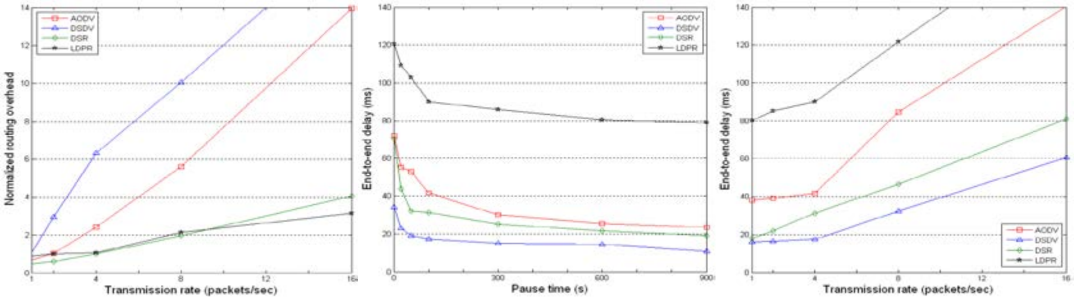
Delay is inevitable in DTNs, it is still of interest to consider the end-to-end delay to find out how much time it makes a message to be delivered. Let us observe the results reported in Figures 5(e) and 5(f). We note that the packet delivery delay of LDDS is longer than other three routing protocols in the condition of different pause time and various transmission rates. In LDDS, nodes are required enough time to obtain the information of location and moving direction. It dooms that the nodes need to usually gather and update the information in a certain time interval. The feature of packet delivery delay in LDDS determines that LDDS can be only implemented in the environment which focuses on the validity and integrity of the message rather than delivery delay.



(a) Packet delivery ratio versus pause time

(b) Packet delivery ratio versus transmission rate

(c) Normalized routing overhead versus pause time



(d) Normalized routing overhead versus transmission rate

(e) End-to-End delay versus pause time

(f) End-to-End delay versus transmission rate

5. Conclusions

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 have proposed a new buffer management scheme named Location and Direction Aware Drop Scheme (LDDS) in order to improve 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, LDDS takes advantages of the nodes' information of the location and moving direction to store the message into buffer space.

We hope that LDDS is able to ensure good performance with a higher packet delivery rate and lower routing overhead. More specifically, LDDS should be able to outperform FIFO in Epidemic Routing and in other routing protocols which is characterized by smaller buffer size.

Acknowledgements

This work is supported in part by the research fund from Nanjing University of Information Science and Technology (Grant No. S8113003001), National Natural Science Foundation of China (Grant No. 61232016), National Basic Research Program 973 (Grant No. 2011CB311808), and the Priority Academic Program Development of Jiangsu Higer Education Institutions fund. It was also supported by the Industrial Strategic Technology Development Program (10041740) funded by the Ministry of Knowledge Economy (MKE) Korea, and by the Natural Science Foundation of Jiangsu Province (No. BK2012461).

References

[1] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets", Annual Conf. of the Special Interest Group on Data Communication (ACM SIGCOMM'03), **(2003)**, pp. 27-34.

[2] A. Krifa, C. Barakat and T. Spyropoulos, "Optimal Buffer Management Policies for Delay Tolerant Networks", 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops, **(2008)**.

[3] A. Lindgren, A. Doria and O. Schelen, "Probabilistic Routing in Intermittently Connected Networks", ACM SIGMOBILE Mobile Computing and Communications Review, vol. 7, no. 3, **(2003)**, pp. 19-20.

[4] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks", Technical Report, CS-200006, Duke University, **(2000)**.

[5] R. Ramanathan, R. Hansen, P. Basu, R. R. Hain and R. Krishnan, "Prioritized Epidemic Routing for Opportunistic Networks", ACM MobiSys workshop on Mobile Opportunistic Networks (MobiOpp 2007), **(2007)**.

[6] S. Jain, K. Fall and R. Patra, "Routing in a delay tolerant network", ACM SIGCOMM, **(2004)**.

[7] E. P. C. Jones and P. A. S. Ward, "Routing Strategies for Delay-Tolerant Networks", Computer Communication Review, **(2008)**.

[8] E. P. C. Jones, L. Li and P. A. S. Ward, "Practical Routing in Delay-tolerant Networks", ACM SIGCOMM workshop on Delay-tolerant networking, pp. 237-243, **(2005)**.

[9] C. Liu and J. Wu, "Scalable Routing in Delay Tolerant Networks", MobiHoc'07, **(2007)**, pp. 51-60.

[10] J. Shen, S. Moh and I. Chung, "Routing Protocols in Delay Tolerant Networks: A Comparative Survey", 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), **(2008)**, pp. 1577-1580.

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

[12] K. Massri, A. Vernata and A. Vitaletti, "Routing Protocols for Delay Tolerant Networks: a Quantitative Evaluation", ACM MSWiM'12, **(2012)**, pp. 1-8.

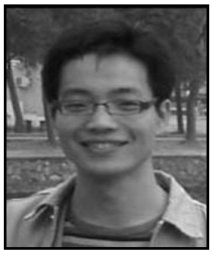
[13] Y. Zhu, B. Xu, X. Shi and Y. Wang, "A Survey of Social-Based Routing in Delay Tolerant Networks: Positive and Negative Social Effects", IEEE Communications Surveys & Tutorials, vol. 15, no. 1, **(2013)**, pp. 387-401.

Authors

Jian Shen Jian Shen received the B.E. degree from Nanjing University of Information Science and Technology, Nanjing, China, in 2007 and the M.E. and Ph.D. degrees in Computer Science from Chosun University, Gwangju, Korea, in 2009 and 2012, respectively. Since late 2012, he has been a faculty member in the School of Computer and Software at Nanjing University of Information Science and Technology, Nanjing, China. His research interests include computer networking, security systems, mobile computing and networking, ad hoc networks and systems, and ubiquitous sensor networks.

Wenying Zheng received the B.E. degree from Yanbian University, Yanji, China, in 2007 and the M.E. degrees in Electronic Engineering from Chosun University, Gwangju, Korea, in 2009, respectively. Since late 2012, She has been a faculty member in the School of Applied Meteorology at Nanjing University of Information Science and Technology, Nanjing, China. Her research interests include image security, image recognition, and security systems.

Jin Wang received the Ph.D. degree in the Ubiquitous Computing laboratory from the Computer Engineering Department of Kyung Hee University Korea in 2010. Now, he is a professor in the Computer and Software Institute, Nanjing University of Information Science and technology. His research interests mainly include routing protocol and algorithm design, performance evaluation and optimization for wireless ad hoc and sensor networks. He is a member of the IEEE and ACM.



Yuhui Zheng, Assosiate Processor at the School of Computer and Software, Nanjing University of Information Science and Technology. His research interest covers image processing, pattern recognition, and remote sensing image restoration.

Yuan Mei received the Ph.D. degree in pattern recognition and intelligence system from Nanjing University of Science & Technology in 2009, China. He is currently working at the school of computer and software, NanJing University of Information Science & Technology. His research interests include fingerprint recognition, image processing and machine learning.