

An Efficient Multiple-Copy Routing in Intermittently Connected Mobile Networks

Fuquan Zhang^{1*}, Inwhee Joe², Demin Gao¹ and Yunfei Liu¹

¹College of Information and Sciences

NanJing Forestry University, NanJing, China

²Department of Electronic and Computer Engineering

Hanyang University, Seoul, Korea

zfq@njfu.edu.cn; iwjoe@hanyang.ac.kr; dmgao@njfu.edu.cn; lyf@njfu.edu.cn

Abstract

PRoPHET (probability routing protocol using history of encounters and transitivity) reduces the invalid copy of message, so that effectively reduce the network overhead and the traffic load in the network. However, the calculation of probability in Prophet does not consider the network conditions. In this paper, we improve the performance of PRoPHET protocol by taking into account connection status to calculate nodes' encountering probability. The performance of the proposed protocol was analyzed in terms of delivery probability, average delay, and overhead ratio. Numerical results show that the proposed protocol can improve the performance to some extent.

Keywords: probability, simulation, overhead, DTN

1. Introduction

The source to destination path may not be connected at any given time instant in delay tolerant networks (DTN). Thus, data is delivered using a store-carry-forward model. Node in the network stores data locally, and upon contact with other nodes, forwards the data. In other words, a link is established between the pair of nodes, whenever they encounter. This link is time-sensitive in that it is only valid for the duration when the nodes are in range of one another [1].

An end-to-end path is not guaranteed. The packets are delivered from a source node to a destination node with store-carry-forward strategy In delay tolerant network. A source node or an intermediate node stores packets in buffer and carries them while it moves around.

Due to its characteristics such as long communication delay, high dynamic topology, sparse distribution of nodes and frequent link break, routing are the key research issues of delay tolerant networks.

The typical DTN protocols can be classified into single-copy protocol and multi-copy protocols. Only single-copy of each message exists in the network in single-copy protocols. On the contrary, multiple copy of each message exists in multi-copy protocols.

In Direct Delivery [2], the node carries messages until it meets their final destination. In First Contact [3] routing the nodes forward messages to the first node they encounter, which results in a "random walk" search for the destination node and only a single copy of every message in the network. To prevent two nodes who stay in contact for a long time exchanging the same messages back and forth, the receiving node accepts a message only if the message has not passed through it before[8].

Epidemic routing [4] [5], also known as Flooding, in contrast, forwards all non-duplicated messages, including its own, to every node it encounters – eventually delivering its messages to the appropriate destinations [7].

Epidemic router is assumed that each node has infinity memory. In this case, all the nodes can store the entire messages during contact with other nodes. When the contact is available node can exchange all the messages existing in its buffer [12].

This simple routing protocol is very powerful even when the buffer size of nodes is sufficient. However, if the buffer size is not sufficient, especially as in mobile nodes, epidemic protocol generates significant message overhead and the performance degrades [11].

Most practical algorithms seek a balance between these extremes. Some approaches limits replications within the network. Several schemes have been proposed on the purpose of solving the message overhead problem in epidemic protocol, such as Spray and Wait protocol [13] and P_{RO}PHET protocol [6]. In these protocols, the total number of message copies inject in a network is limited by a certain number or message is forwarding only when a certain condition is met.

Spray and wait router has spray phase and wait phase. In spray phase source node forward the packet to a limited number of L different node. If destination is found then the message or packet transfer is successfully transmitted. If not than wait phase is started. When the destination is encounter the message will be directly transmitted [14, 15].

In Spray phase, message copies are disseminated to other nodes until there is no node which has more than one message copy. Then, the protocol works in the Wait phase and the message copy is delivered to a destination node only. In order to enhance the dissemination speed of message copies[11]. An binary Spray and Wait protocol was proposed, where half of stored message copies are distributed to another node without considering the characteristics of receiving nodes[15].

Suppose, for the purpose of illustration, that there are M nodes in the network, that messages are N bytes long, and that they can be delivered in their entirety during the contact times between nodes. Direct delivery would use only N bytes of bandwidth and minimum buffer space since only its own messages are stored and the message is transferred only once[10].

However, since delivery depends on the source and the destination eventually meeting, packets may never be delivered or may experience long latencies. Epidemic routing, on the other hand, forwards all its messages to intermediaries conceivably using $N(M-1)$ bytes of bandwidth to transfer data in addition to the auxiliary information exchanged during encounters to determine which packets are non-duplicated and ready for transfer. In addition to bandwidth, large buffers may be required to store in-transit packets to achieve minimum latency [7]. Single-copy and multi-copy routing protocols represent a natural trade-off in DTN networks [8].

The Prophet (Probabilistic Routing Protocol using History of Encounters and Transitivity) [6] attempts to exploit the likelihood of real-world encounters by maintaining a set of probabilities for successful delivery to known destinations in the DTN, and replicating messages during opportunistic encounters only if the mobile node in contact that does not possess the message appears to have a better chance of delivering it to the destination [8].

That is, the delivery predictability between two nodes is calculated based on contact history between them. Here, a higher delivery predictability implies a higher probability of future contacts between them. A message is copied to a contact node only when the delivery predictability to a destination node of the contact node is larger than that of the transmitting node. This makes P_{RO}PHET protocol improve the delivery probability with low overhead [11].

Prophet reduces the invalid copy of message, so that effectively reduce the network overhead and the traffic load in the network. However, the calculation of probability in Prophet does not consider the network conditions [10].

In this paper, we proposed an Iprophet (Improved Prophet) algorithm, which take into account connection status to calculate nodes' encountering probability.

Simulations show that the proposed algorithm improved the performance in terms of delivery rate, end-to-end delay, routing overhead.

The remainder of this paper is organized as follows. Section 2 presents the proposed algorithm. Simulation results are presented in Section 3. Conclusion is in Section 4.

2. Proposed Algorithm

2.1. Literature Review of Prophet Algorithm

PRoPHET protocol uses nonrandom mobility and contact patterns in real application scenarios to copy messages to other nodes in order to improve the routing performance [13]. The PRoPHET protocol assumed that if a node has visited a location or contacted with a node frequently, the node has a higher probability of visiting the location and contacting the node again.

In Prophet, the calculation of the delivery predictability is to update the metric whenever a node is encountered. The nodes that are often encountered have high delivery predictability, as calculated in Eq. 1, where $p_{init} \in [0,1]$ is an initialization constant [6].

$$P_{(a,b)} = P_{(a,b)old} + (1 - P_{(a,b)old}) * P_{init} \quad (1)$$

Prophet introduced a metric called "delivery predictability", $0 \leq P_{(a,b)} \leq 1$, is established at every node A for each known destination B. A node which has a higher delivery predictability value for a particular destination is assumed to be a better router for delivering a message to that destination. For example, if $P_{(a,b)} > P_{(c,b)}$, message for destination B are preferable to forward to node A rather than node C [16].

The aging equation is shown in Eq. 2, where $\gamma \in [0,1)$ and is the aging constant, and k is the number of time units that have elapsed since the last time the metric was aged.

$$P_{(a,b)} = P_{(a,b)old} * \gamma^k \quad (2)$$

2.2. Proposed IProphet

In traditional routing protocols, the message is usually sent to the neighbor that has the shortest path to the destination. However, network conditions are completely different in delay tolerant networks. Since the source to destination path may not be connected at any given time instant. A link is established between the pair of nodes, whenever they encounter. This link is time-sensitive in that it is only valid for the duration when the nodes are in range of one another.

In order to let the delivery predictability calculation reflect the actual link situation, we established a link status table of nodes.

As shown in Table 1, if two nodes, m and n, met each other for the first time, in this two-node communication cycle. T_{con} is communication connection time. T_{brk} is node disconnect time in the communication. T_{sum} is communication cycle. Based on

the communication connection time value within the cycle, the link connection time ratio can be calculated.

Table 1. Link Table of Node

Current node	Next node	T_{sum}	T_{con}	T_{brk}
m	n	T_{sum1}	T_{con1}	T_{brk1}
m	n	T_{sum2}	T_{con2}	T_{brk2}
m	n	T_{sum3}	T_{con3}	T_{brk3}

By considering link connection time ratio in the link status table, wireless link reliability is incorporated with neighbor one-hop count to discover more reliable pair paths. The calculation of the delivery predictability is shown in Eq. 3, where $p_{init} \in [0,1]$ is an initialization constant.

$$P_{(a,b)} = P_{(a,b)old} + (1 - P_{(a,b)old}) * P_{init} * (T_{con} / T_{sum}) \quad (3)$$

If a pair of nodes does not encounter each other in a while, the delivery predictability values should age, being reduced in the process. The aging equation considering the link status is shown in Eq. 4, where $\gamma \in [0,1)$ and is the aging constant, and T_{brk} is node disconnect time in the communication.

$$P_{(a,b)} = P_{(a,b)old} * \gamma^{T_{brk}} \quad (4)$$

If node A frequently encounters node B, and node B frequently encounters node C, then node C probably is a good candidate node to forward messages destined for node A to. That is, the delivery predictability also has a transitive property [6]. Same as in Prophet, Eq. 5 shows how this transitivity affects the delivery predictability, where $\beta \in [0,1]$ is a scaling constant that decides how large impact the transitivity should have on the delivery predictability.

$$P_{(a,c)} = P_{(a,c)old} + (1 - P_{(a,c)old}) * P_{(a,b)} * P_{(b,c)} * \beta \quad (5)$$

3. Simulation

The performance of proposed protocol is compared with several related works. The simulations were performed in ONE 1.41 simulator [1].

3.1. Performance Metrics

Delivery probability, overhead ratio and latency are used to evaluate the performance of protocols. Delivery probability is the ratio of data packets that has successfully received by the destination nodes versus data packets that has been sent by the source nodes.

Latency is the average message delay from creation to delivery.

Overhead Ratio is the number of delivered messages versus the difference between the number of relayed messages and number of delivered messages. It's some measure of how much "overhead" there was in relation to delivered messages.

3.2. Simulation Configuration

Table 2 shows the simulation configuration.

Table 2. Simulation Configuration

<i>Parameters</i>	<i>Values</i>
Simulation time	43200s
Buffer size	5M
Number of Nodes	400
Max. Mobile Speed	4m/s
Message TTL	18000s
Transmit speed	2 Mbps, 10Mbps
Event generators	25s-35s

3.3. Varying Mobile Speed

As shown in Figure 1.A, the delivery rate is sensitive to mobile speed. When the mobile speed of node is slower than 3m/s, the deliver rate of two algorithms is increasing. Proposed algorithm is relatively good.

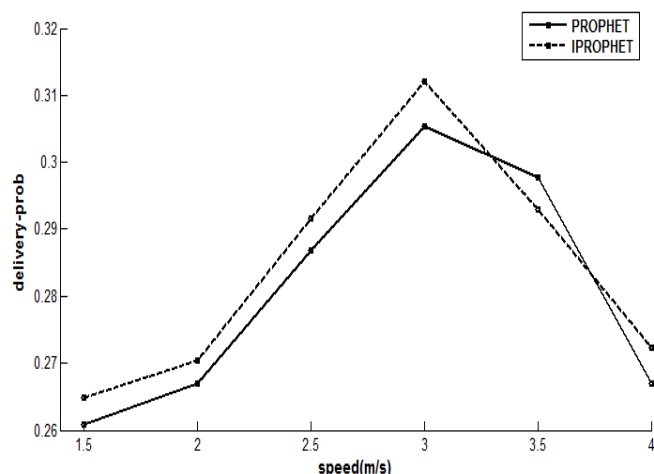
When the node movement is getting faster, the chance of encountering destination node is increasing. This can improve message delivery rate to some extent. However, when speed is faster than 3m/s, due to the high mobility, connection time between the nodes is relatively smaller, which may cause delivery rate decreasing in both algorithms [9].

As shown in Figure 1.2.C, with the increasing in mobile speed, the delay is sharply decreasing and reaching to a stable value. The increasing of mobile speed can improve the delivery rate.

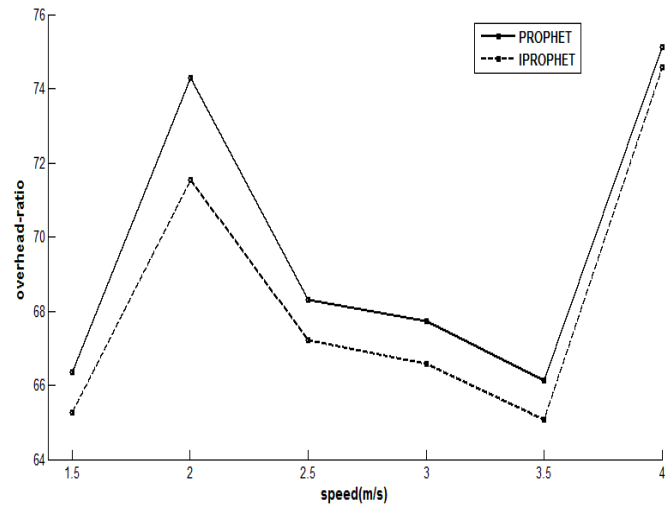
Thus, the caching delay, waiting time and propagation delay are reducing. There's little difference on the performance in delay of both algorithms. After the 3M/s, delay changes very little. Prophet performs slightly better than Propheth.

As shown in Figure 1.3.C, when the speed is less than 3.5 m/s, there is no obvious trend between mobile speed and overhead.

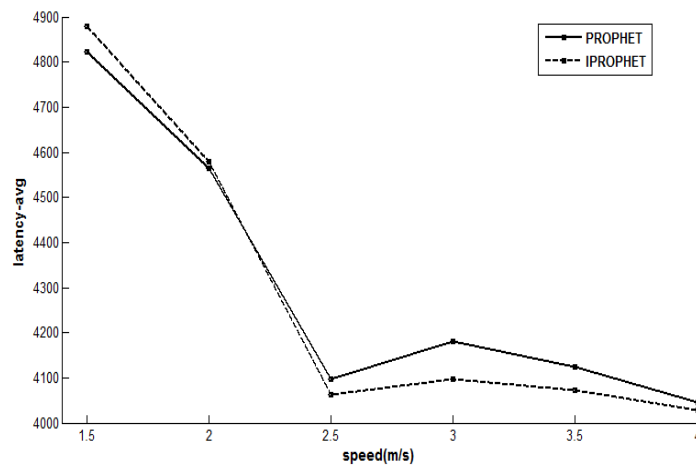
IPropheth performs better than Propheth. Forwarding of messages results in sharply changing in overhead when the speed is greater than 3.5M/s. There's little difference on the performance in both algorithms.



(A, Delivery Probability)



(B, Overhead Ratio)



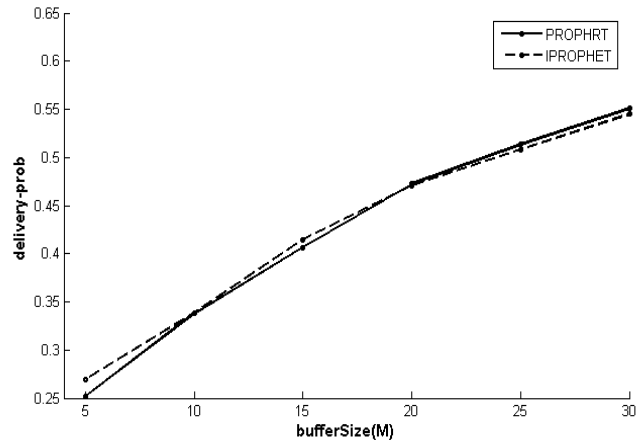
(C, Latency)

Figure 1. Varying Mobile Speed (A, Delivery Probability B, Overhead Ratio C, Latency)

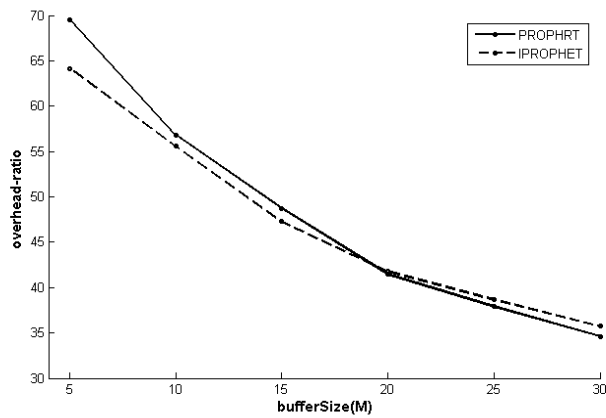
3.4. Varying Memory Size

Figure 2 shows that memory constraints can severely affect the performance of DTN routing schemes. Performance increases in both algorithms with increasing memory size in terms of delivery rate and overhead.

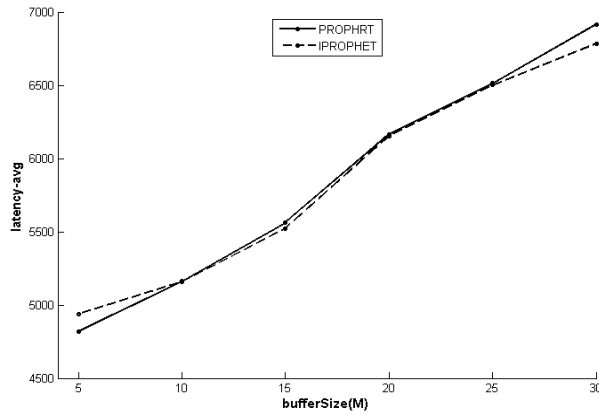
Because of more messages can be stored in a node compared to small memory size, the more messages stored in the memory, the higher probability to be delivered.



(A, Delivery Probability)



(B, Overhead Ratio)



(C, Latency)

Figure 2. Varying Memory Size (A, Delivery Probability B, Overhead Ratio C, Latency)

With very small memory size, the stress of memory requirement is relative heavy. When the queue size of a station is increasing, the performance change of both protocols is not obvious because there is not much requirement on the computational cost and memory space in choosing of relay node.

As shown in Figure 2.C, with the cache size increasing, more and more messages will carry out the store-and-forward scheme, and will not be dropped until their time-to-live expiration. This leads to the increasing in delay of both algorithms.

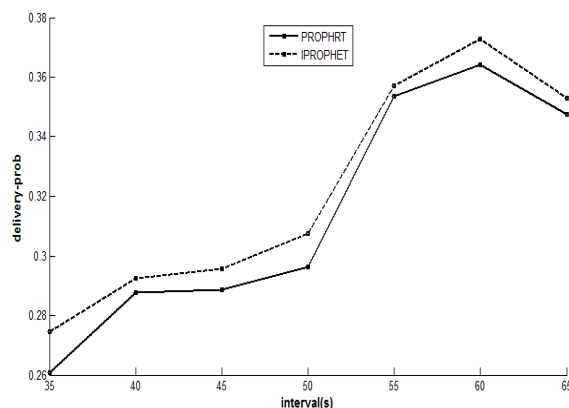
3.5. Varying Time Interval of Message Generation

Figure 3.A shows performance with varying interval of message generation. Delivery rate improves in most cases with increasing interval of message generation (decreasing offered load). Iprophet is better than Prophet in delivery rate. After 60s, the delivery rate of both algorithms decreases. This shows that long time interval of message generation will lead to a lower delivery rate because the number of messages injected in the network will reduce.

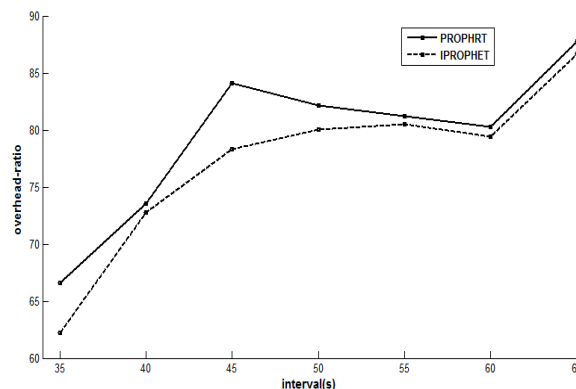
With increasing in message generation time interval, overhead is increasing corresponding, as shown in Figure 3.B. Proposed algorithm has lower overhead than Prophet. The overhead is increasing gradually at 35s-60s.

We observed that an appropriate message generation interval can improve the delivery rate, reduce the delay and keep a relatively stable overhead. After the 60s, on the contrary, the lower generation rate leads to an increase in overhead.

As shown in Figure 3.C, with the increasing of message generation interval, delay is decreasing after 35s. A short interval means a high generation rate that requires more memory to store message. This requires that protocols can handle the stress of data delivery of nodes before the memory is full.



(A, Delivery probability)



(B, Overhead Ratio)

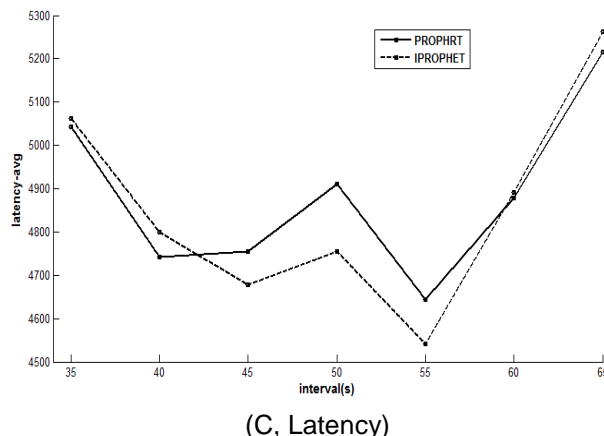


Figure. 3. Varying Time Interval of Message Generation (A, Delivery Probability B, Overhead ratio C, Latency)

If messages in the memory cache have not been delivered before the memory is full, the new generated message will cause the dropping of old messages. However, there is not obvious tradeoff between interval of message generation and delay. Iprophet performs better than Prophet. Delay is increasing after 55s. This showed that a moderate message generation interval can not only improve the delivery rate but also reduce the delay to some extent.

4. Conclusion

The Prophet attempts to exploit the likelihood of real-world encounters by maintaining a set of probabilities for successful delivery to known destinations. Prophet reduces the invalid copy of message, so that effectively reduce the network overhead and the traffic load in the network.

The delivery predictability between two nodes is calculated based on contact history. The nodes with higher delivery predictability imply a higher probability of future contacts.

In P_RO_PHET protocol, a message is copied to a contact node only when the delivery predictability to a destination node of the contact node is higher than that of the transmitting node.

However, the calculation of probability in Prophet does not consider the network conditions. We proposed an IProphet algorithm, which take into account link status to calculate nodes' encountering probability.

In this paper, we improve the performance of P_RO_PHET protocol by taking into account connection status to calculate nodes' encountering probability.

The performance of the proposed protocol was analyzed in terms of delivery probability, average delay, and overhead ratio. Numerical results show that the proposed protocol can improve the performance to some extend.

Acknowledgments

The project sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry. This work was supported by the Foundation of NanJing Forestry University, G2014105. This work was supported by Natural Science Foundation of Jiangsu Province, BK20150880. The corresponding author is fuquan zhang.

References

- [1] A. Keränen, J. Ott and T. Kärkkäinen, “The ONE Simulator for DTN Protocol Evaluation”, SIMUTools, (2009).
- [2] T. Spyropoulos, K. Psounis and C. S. Raghavendra, “Single-copy routing in intermittently connected mobile networks” In Proceedings on Sensor and Ad Hoc Communications and Networks SECON, (2004), pp. 235–244.
- [3] A. Vahdat and D. Becker, “Epidemic routing for partially connected ad hoc networks”, Technical Report CS, Duke University, (2006).
- [4] S. Jain, K. Fall and R. Patra, Technical Report CS-200006, Duke University, (2006).
- [5] Q. Wang, Z. J. Haas, “Performance Analysis of Epidemic Routing for Delay-Tolerant Networks, Resource Management in Mobile Computing Environments Modeling and Optimization in Science and Technologies”, vol. 3, (2014).
- [6] A. Lindgren, A. Doria, E. Davies and S. Grasic, “Probabilistic routing protocol for intermittently connected networks”, IETF RFC 6683, (2012).
- [7] Y. Wu, S. Deng and H. Huang, “Performance analysis of hop-limited epidemic routing in DTN with limited forwarding times”, International Journal of Communication Systems, (2014).
- [8] R. J. D'Souza and J. Jose, “Routing Approaches in Delay Tolerant Networks: A Survey”, International Journal of Computer Applications, (2010).
- [9] F. Zhang, “Using local speed information as routing metric for delay tolerant Networks”, WSEAS Transaction on computers, (2015).
- [10] F. Zhang, I. Joe, D. Gao and Yunfei Liu, “A Connection Information Aided Routing Protocol in Delay Tolerant Network”, The 5th International Conference on Computer Science and Technology, (2016).
- [11] S. D. Han and Y. W. Chung, “An Improved PProPHET Routing Protocol in Delay Tolerant Network,” The Scientific World Journal, vol. 2015, Article ID 623090, 7 pages, (2015).
- [12] B. Patel, K. Dave and V. Pandya, “Spray and Wait Routing Protocol in Delay Tolerant Networks”, International Journal of Emerging Technology and Advanced Engineering, (2014).
- [13] T. Spyropoulos, K. Psounis and C. S. Raghavendra, “Efficient routing in intermittently connected mobile networks: the multiple-copy case”, IEEE/ACM Transactions on Networking, vol. 16, no. 1, (2008), pp. 77–90.
- [14] T. Spyropoulos, K. Psounis and C. S. Raghavendra, “Spray and wait: an efficient routing scheme for intermittently connected mobile networks”, in Proceedings of the ACM SIGCOMM 2005 Workshop on Delay-Tolerant Networking (WDTN '05), (2005), pp. 252–259.
- [15] T. Spyropoulos, K. Psounis and C. S. Raghavendra, “Efficient routing in intermittently connected mobile networks: the multiple-copy case”, IEEE/ACM Transactions on Networking, vol. 16, no. 1, (2008), pp. 77–90.
- [16] J. Borah, D. Devi and Y. J. Singh, “Analysis and evaluation of probabilistic routing protocol for intermittently connected network”, IOSR Journal of Computer Engineering, (2013).

Authors



Zhang Fuquan, he received M.S degree in Computer Science from Shen Yang Li Gong Univ. in 2005, and Ph.D. degree in Hanyang University, Seoul, Korea. His research fields include 3G/4G cellular systems and wireless mesh network.



Inwhae Joe, he received B.S. and M.S. degrees in Electronics Engineering from Hanyang University, Seoul, Korea, and Ph.D. degree in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta, GA in 1998. Since 2002, he has been a faculty member in the Division of Computer Science & Engineering at Hanyang University, Seoul, Korea. His current research interests include wireless sensor networks, 3G/4G cellular systems, mobility management, multimedia networking.



Demin Gao, he received a Ph.D. degree in 2012 from Nanjing University of Science and Technology Department of Computer Science and Engineering. In 2012, he joined in the Nanjing Forestry University as a lecturer. His research interests include routing protocols for delay tolerant, data aggregation and multi-constrained routing algorithms in wireless sensor networks and wireless sensor networks.



YunFei Liu, he received a Ph.D. degree from Nanjing University of Aeronautics & Astronautic of China in 2005. He is a professor of Electronic and Communication Engineering at the College of Information Science and Technology, Nanjing Forestry University. His current research interests is signal processing technology.

