A Routing Algorithm for Delay Tolerant Networks

LIU Bin*, WANG Zhi-lin2, ZHU Ren-jie3
1R&D department, Zhongke Jiudu (Beijing) Space Information Technology Co., Ltd, Beijing, China
2R&D department, Zhongke edge intelligent information technology (Suzhou) Co., Ltd, Suzhou, China
3School of Computer science and Engineering, Nanjing University of Technology, Nanjing, China

*liubin@datopia.cn

Abstract—Based on the limitations of traditional submission ratio of Prophet caused by the performance difference and state difference of nodes in the new service environment, and the idea of utility quantization, a Prophet-BSAS algorithm based on node state was designed. The utility parameters and utility functions of the improved algorithm were analyzed in detail, and the performance of Prophet-BSAS and other classical routing algorithms was simulated and analyzed under the typical mobility model.

1. INTRODUCTION
Along with the continuous development of computer technology, network technology, wireless communication technology, and the enrichment of application requirements, some emerging networks get extensive attention. Scientists including K.Fall put forward Delay Tolerant network (DTN) [1] in ICIR conference held in 2002. DTN, an emerging field of study, is a kind of overlay network over regional network (including the Internet), which is used to deal with all kinds of network communication under various extreme environments. With application scenes of the DTN network from space to earth, sea and even under the sea, great changes have taken place in the network environment of DTN. For example, in the new application scenes like the communications between the spacecrafts, and the ground command center, as node movements become more random, the performance of the nodes and the status difference becomes more significant, the increase in communication demand and the increase in data communication have brought many new challenges to the routing of nodes, and therefore it is necessary to design more direct and optimal routing algorithms that are more in line with the network status quo.

Prophet, as a typical routing algorithm based on utility quantization, has the freedom of parameter selection, which can effectively control the amount of copies and Network overhead in the network, and ensure a certain quality of transmission and it has the applicability to the opportunistic connection network and strong sociality that does not depend on the node movement, which is of wide application range. The paper is based on routing algorithm to be improved, uncertainty and chance brought about by the randomness the DTN nodes movement, data volume increases and congestion problems inherent caused by the Prophet for some quality message forwarding node priority and optimization of the parking lot the Prophet - BSAS algorithm based on utility of quantitative is designed, and through
simulation experiments, comparison and analysis of some classical routing algorithm are made under random moving model of the nodes.

2. DESIGN CONCEPT
At the beginning of the design of Prophet, the routing algorithm of Prophet only focuses on the possibility of establishing connections between nodes, but different from the former application environment, under the new application environment, we would hope when the routings are being chosen, the nodes selected can transmit more data in a limited time, are able to accept and keep more data, and reduce the generation of useless connection and transmission failure. This paper will consider to design from the following several aspects:

Considering the difference in the transmission capacity of each node, the node with faster transmission ratio can be given priority by adding the peak transmission ratio of the node into the consideration category, so as to improve the efficiency of transmission.

Considering the buffer size of the node, the node with large remaining buffer size can be given priority.

The algorithm of average forwarding success rate of node data. When Prophet routing computing is used for delay tolerant network, the researchers found that the problem of “parking lot” in network, namely frequent short connection was set up between nodes sometimes, which caused inflated illusion of deliver between the nodes because of such a connection, but often difficult to complete a full message transmission process. Focused on this problem, this paper took the node data forwarding average success rate into consideration. In this way, when a node often establishes connections with other nodes, but always fails to complete the message transmission, the data forwarding success rate of the node is low, which will limit the increase of the probability of submission when the node meets with other nodes.

After we have the factors to be considered, we need to do utility quantization of these factors. By borrowing from Edgeworth, British statistician and pioneer in mathematical statistics, who explained a male in the shopping process, using the formula for quantifying the utility of consumption to the satisfaction it produces (as shown in the following formula (1)), he introduced the concept of utility, and utility of quantization to routing algorithm design.

\[ U = F(x, y) \]  

(1)

Where, \( U \) is the degree of satisfying the demand of data transmission. In the Internet, \( U \) is the quality of data transmission. And the influence factors of satisfying this need is turned into factors such as network bandwidth, signal-to-noise ratio, internet rate and so on. In delay tolerant network, \( x \) can represent internet rate of nodes, \( y \) can represent the node storage space. The design of utility function should be able to reflect the influence of the factors selected for routing on the final choice.

In fact, in the process of utility quantization, the more factors are considered, the more intelligent and accurate the routing algorithm will be and the more efficient of the selected nodes in the message passing will be. However, more complex algorithm inevitably will be accompanied by greater energy consumption of resources. Therefore, under the premise of ensuring certain superiority of the algorithm, the paper focuses on the utility quantization of node buffer size, node transmission rate and average transmission success rate to improve Prophet's submission probability formula, combined with Prophet's time decay formula and transmission probability formula, the optimized routing algorithm Of Prophet-BSAS is designed.

3. PROPHET OPTIMIZATION ALGORITHM BASED ON UTILITY QUANTIZATION

3.1. Node transmission rate
The difference of the transmission rate of the delay tolerance network is caused by the various difference of nodes, including the bandwidth resource of the node, the performance of the node CPU, internet rate and the number of tasks to be processed by the node, and so on. In order to reflect the transmission capability of the node, the paper adds the peak transmission rate recorded by the node in
all transmission processes to the calculation of submission probability, but it needs to be normalized before the peak rate is used.

\[ V_{\text{norm}} = \frac{v - v_{\text{min}}}{v_{\text{max}} - v_{\text{min}}} \]  

(2)

We use linear function to normalize peak rate \( v \). Here \( v \) refers to theoretical minimum value of transmission speed. The speed is the lowest when sending is stopped, so \( v_{\text{min}} = 0 \). \( v_{\text{max}} \) is the transmission peak which can be reached for all nodes theoretically, which follows the barrel effect and is decided by the CPU operation rate, ROM reading rate and short board in Network card performance. In the real machine test, usually CPU operation rate is the bottleneck, and in the simulation experiments, \( v_{\text{max}} \) shall be reached when the nodes is performing initialization.

3.2. Node buffer
According to Prophet's original submission probability calculation method, it could be predicted that the more active the nodes, namely the more frequently they moves, and the more nodes that contact with other nodes, the easier it will be selected as the message relay managed object, and the more likely it is to get congested. However, this kind of nodes is the provider of high quality service, which may fall into the dilemma that quality service providers are always in a congested state. Therefore, load balancing must be taken into account in routing calculation to reasonably reduce the probability of transmitting messages for less remaining nodes in buffer resources. To this end, each node normalizes the capacity of the node’s remaining buffer to obtain the variable \( M_{\text{bufferNorm}} \), which should be positively correlated with the capacity of the node's remaining buffer. The larger \( M_{\text{bufferNorm}} \) of the nodes are, the more likely it is to be selected as the next hop node. Unlike the process of normalization of node transfer rate, a basic logic is considered in the normalization: when the node's remaining buffer is smaller, routing selection should be sensitive to the size of the remaining buffer. However, when the node has a large amount of remaining buffer, it is not easy to have insufficient buffer at this time, so the size of remaining buffer is insensitive to the routing selection. According to this logic, the paper designs the normalization function as follows:

\[ M_{\text{bufferNorm}} = \frac{2}{\pi} \tan^{-1} \left( \frac{M_{\text{rest}}}{100} \right) \]  

(3)

\( M_{\text{rest}} \) represents the size of the remaining buffer, and the unit is MB. The \( M_{\text{bufferNorm}} \in [0,1) \). Function image is shown in figure 1. When the remaining buffer is scarce, \( M_{\text{bufferNorm}} \) is sensitive to the size change of buffer; when the remaining buffer is very sufficient, \( M_{\text{bufferNorm}} \) is insensitive to the buffer size change.

![Figure 1. \( M_{\text{bufferNorm}} \) function image](image)

3.3. Node transmission success rate
When Prophet routing algorithm is used in DTN, “parking lot problems” often occur. Especially in the scene of frequent movement, the probability of short connection is much higher than that of stable connection. For the “parking lot problem”, this paper adds the average forwarding success rate of node
data into the calculation of submission probability, which can effectively avoid “fake quality” nodes. In the DTN, when a message fails to be fully transmitted, the node that initiated the transmission will regard the transmission as a failure and wait for the opportunity to re-transmit. Therefore, when there are too many short connections that are difficult to transmit the complete message, the average forwarding success rate of the node is bound to be affected. In order to calculate the average forwarding success rate of data, the nodes need to carry out statistics of the total amount of data processed and the total amount of data successfully forwarded, that is, the sum of the total amount of data of messages entering the node and the total amount of data of messages generated by the node itself, as well as the total amount of data of messages successfully forwarded.

\[ M_{\text{handle}} = \sum_{i=1}^{n} \text{Receive}[m_i \cdot \text{getsize}()] \]
\[ + \sum_{j=1}^{m} \text{Generate}[m_j \cdot \text{getsize}()] \]  
\[ M_{\text{transmit}} = \sum_{k=1}^{p} \text{Transmit}[m_k \cdot \text{getsize}()] \]  
(4)
(5)

\[ m \cdot \text{getsize}() \) represents the size of the acquired message, \( n, m \) and \( p \) represents the total number of messages received by the node, the total number of messages generated by the node and the total number of messages successfully forwarded by the node. After obtaining these two data, the average forwarding success rate of the node is calculated.

\[ P_{\text{transmit}} = \frac{M_{\text{transmit}}}{M_{\text{handle}}} \]  
(6)

### 3.4. Submission probability

On the basis of the above data, the transmission rate, remaining buffer of the nodes, and the calculation of historical success rate of data forwarding in the node can be added to the calculation of the submission probability \( P(a,b) \) to optimize the Prophet's submission probability formula.

Formula (7) is the submission probability formula of Prophet routing algorithm, and gives the increasing rule of Prophet's original submission probability. Every time two nodes meet, they will update the submission probability between themselves and each other, and increase on the original basis:

\[ P(a,b) = P(a,b)_{\text{old}} \times (1 - P(a,b)_{\text{old}}) \times P_{\text{init}} \]  
(7)

Where, \( P(a,b)_{\text{old}} \) is the original probability of meeting between node \( a \) and node \( b \), and \( P_{\text{init}} \in [0,1] \) is the initialization constant, which usually takes the value of 0.75. In the implementation of Prophet by ONE simulator, \( P_{\text{init}} \) also takes the value of 0.75.

According to the logic of Prophet submission probability formula(7), in a new encounter, submission probability of new successful message between nodes is based on the current successful submission probability, and because of this encounter, the ratio of potential in the current undeliverable probability to the ratio of successful submission, is the \( P_{\text{init}} \).

At this time, \( P_{\text{init}} \) reflects the increase of future submission probability between nodes due to the encounter between nodes. According to the above analysis, it is not only the encounter between nodes that affects the submission probability, but also the node's transmission rate, remaining buffer and historical success rate of data forwarding in the node. Therefore, this paper first spreads the proportion of \( P_{\text{init}} \) into the transmission rate \( V_{\text{norm}} \) and the remaining buffer \( M_{\text{bufferNorm}} \). The higher the transmission rate and the more sufficient the buffer are, the more likely the message will be submitted successfully. At the same time, the paper also adds the historical transmission success rate which is directly linked to the submission rate to the formula, which is closest to the transmission probability, and also the one that directly affects the transmit probability. As a result, we have the following new submission probability formula:

\[ P(a,b) = P(a,b)_{\text{old}} \times (1 - P(a,b)_{\text{old}}) \times (P_{\text{init}} + \mu \times V_{\text{norm}} + M_{\text{bufferNorm}}) \times P_{\text{transmit}} \]  
(8)
The coefficient $\mu \in (0, 0.75]$ and could meet $P_{init} + \mu = 0.75$.

4. SIMULATION RESULT AND ANALYSIS

The simulation tool used in the experiment is ONE-1.5.1. When the random Waypoint (RWP) node model is adopted, the moving path, rate and standing time of the nodes are random. In the initial state, nodes obey uniform distribution in the whole simulation area. When the simulation starts, first all nodes choose a position as a destination in the two-dimensional simulation area randomly, and then randomly selected a rate from the $[V_{\text{min}}, V_{\text{max}}]$, where the rate choice also obeys uniform distribution. The node then moves uniformly toward the target at the selected speed. After reaching the target location, node chooses a standing time in $[0, T_{\text{wait}}]$. At this point, the node completes a complete motion. After the standing time, the node starts a new movement, and the selection process is repeated.

In the scenario of random movement, although the parameters of the nodes put are different, they are all randomly generated from the same interval. The specific parameters are shown in Table 1.

| Parameter                        | Value            |
|----------------------------------|------------------|
| Simulation area                  | 2500m×3000m      |
| Node number                      | 150              |
| Message generation interval      | 10s~30s          |
| Message size                     | 500KB~1024KB     |
| Maximum transfer rate            | 250Kbps          |
| Transmission range               | 10m              |
| Node buffer size                 | 100MB            |
| Node movement rate               | 1m/s~10m/s       |
| Simulation time                  | 12h              |
| Message age                      | 4h               |
| Standing time                    | 30s~10min        |

Before comparing the routing algorithm, first we need to determine the value of $P_{init}$ and $\mu$ of submission probability formula in this paper. As the primary purpose of DTN is to complete message transmission, therefore, this paper determines $P_{init}$ and $\mu$ in accordance with the priority principle of message arrival rate. In this scenario, through the simulation experiment of 12 hours, we make the parameter $P_{init}$ increased from 0 each time by 0.01 to 0.75, due to $P_{init} + \mu = 0.75$, at this time $\mu$ reduces from 0.75 to 0, then compare changes in the rate of message arrival, the optimal value of $P_{init}$ and $\mu$ are gotten. The experimental results are shown in figure 2.

![Figure 2. Comparison of message arrival rates with different parameters under RWP model](image.png)

When $P_{init}$ is around 0.3, the message arrival rate is the highest after 12 hours of simulation experiment, so $P_{init}=0.3$ in this scenario, and then $\mu=0.45$. 
In order to test the performance difference of this routing algorithm in the network presenting different node mobility characteristics, this paper presents the comparative simulation results of five different routing protocols in terms of data arrival rate, network overhead, transmission average delay, average hop, etc., as shown in Figure 3 to Figure 6.

1) **Data arrival rate** $P_{\text{suc}}$: $P_{\text{suc}}$ represents the ratio of the total amount of data successfully arrived to the total amount of data generated by all nodes.

![Figure 3. Comparison graph of simulation results of data arrival rate](image)

As can be seen from the above simulation results, with the advance of simulation, flooding-based nodes in the two algorithms including Spray and Wait and Epidemic network start to appear congestion, and the algorithm performance is greatly limited. The socially-based routing Bubble Rap algorithm performs poorly in terms of arrival rates. Prophet-BSAS takes into account the performance differences of nodes and gives priority to nodes with higher data transmission rate and higher data transmission success rate. When the transmission opportunity is obtained, the packets can be forwarded better. It selects nodes with sufficient buffer and wait for the next forwarding when there is no transmission opportunity or transmission interruption, which is not easy to lose data. After a period of node moving and contact process and obtaining relatively accurate submission probability, the performance is improved significantly.

2) **Network overhead ratio** $P_{\text{overheadRatio}}$: $P_{\text{overheadRatio}}$ is the difference between the total amount of message data forwarded on the network and the total amount of message data successfully delivered to the target node and the ratio of the total amount of message data successfully submitted to the target node, to reflect the overhead of additional messages that need to be delivered in order to successfully submitting messages.

![Figure 4. Comparison graph of network overhead ratio simulation results](image)

From the above simulation results, it can be seen that both flooding-based routing algorithms have a huge routing overhead, while the social-based Bubble Rap algorithm performs poorly when the node mobility is less social and needs more overhead to complete the final message passing. Prophet based on submission probability and Prophet-BSAS have brought a good practicability in random
environment, especially the Prophet-BSAS restricts the generation of a message copy to any nodes and at the same time, it could be more efficient to transfer the message to the target node, shorten the process of information copy in the network, so as to effectively control the total amount of the network message copy, which lowers around 20 than Prophet on the network overhead.

3) The average transmission delay $T_{avg}$: $T_{avg}$ represents the average time from generation to successful submission of a successfully submitted message.

![Figure 5. Comparison of simulation results of average transmission delay](image)

As can be seen from the above simulation results, there is no doubt that the flooding-based routing algorithm has an advantage in transmission delay, while the social-based Bubble Rap routing algorithm cannot accurately find the appropriate next hop node, with largest transmission delay and the delay of Prophet-BSAS and Prophet is in the middle position.

4) Average hop $C_{hop\_avg}$: $C_{hop\_avg}$ represents the average number of hops experienced by a message from the source node to the target node.

![Figure 6. Comparison of simulation results of average hops](image)

From the above simulation results, it can be seen that a flooding-based routing has experienced more hops to reach the target node, while the Spray and Wait strategy does not make it more advantageous than the Epidemic in terms of average hops, and Bubble Rap also performs less well in a random scene. Due to the high transmission rate of the selected nodes and sufficient capacity of the buffer, it is not easy for packet transmission to fail halfway and need to be re-transmitted. At the same time, the filtering of fake-quality nodes with low transmission success rate also makes the transmission more efficient. In a word, Prophet-BSAS can more accurately select the next hop of high quality, complete the message forwarding process more efficiently, and thus reduce the average number of hops of transmission.

5. CONCLUSION
On the basis of Prophet routing algorithm, this paper analyzes the algorithm improvement idea based on the new usage scenario, and designs the improved routing algorithm Prophet-BSAS algorithm
based on the difference in node performance and state. The optimization algorithm adopts the idea of utility quantization and conducts extraction and utility quantization for several important conditions that will affect the message transmission arrival rate, efficiency, security and network load stress, and eventually integrates into submission formula of the Prophet-BSAS routing algorithm, which makes each node can obtain uniform parameters through the conditions of the current node finally to choose routing through the parameter. Through the simulation experiment, comparison and analysis of several classical routing algorithms are made under the model of random movement of nodes, and the correctness and superiority of the proposed Prophet-BSAS routing algorithm are proved.

REFERENCES
[1] Fall K. A delay-tolerant network architecture for challenged Internets[C]//Acm Sigcomm’03, 2003: 27-34.
[2] Grasic S. Davies E, Lindgren A, et al. The evolution of a DTN routing protocol-PRoPHETv2[C]//Proceedings of the 6th ACM workshop on Challenged networks. ACM, 2011: 27-30.
[3] Bista B B, Rawat D B. EA-PRoPHET: An energy aware Prophet-based routing protocol for delay tolerant networks[C]//2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA). IEEE, 2017: 670-677.
[4] Shuibing LIU. Routing in a Delay Tolerant Network Based on Location Prediction[D]. Huazhong University of Science and Technology.2015.
[5] Hui MA, Tao LI. Application of Prophet Routing Based on Throughput in DTN[J]. Computer Technology and Science, 2018, v.28 ; No.255(07):193-197.
[6] Khan S, Kim K I. A New DTN Routing Protocol for Sparse Airborne Networks[C]//2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems. IEEE, 2013: 417-418.
[7] Hongcheng YAN, Qingjun ZHANG, Yong SUN. Research on Control Strategy of Spatial Delay/Interruption of Tolerant Network Congestion[J].Journal on Communications, 2016, v.37; No.340(01):146-154.
[8] Dapeng WU, Xiangjiu FU, Hongpei ZHANG, etc. On the Control Strategy of Node State Awareness in Delay Tolerant Network Congestion[J].Electronics, 2016, 44(1):186-192.