Article

Routing Algorithm Based on Trajectory Prediction in Opportunistic Networks

Peijun Zou, Ming Zhao *, Jia Wu and Leilei Wang

School of Software, Railway College, Central South University, Changsha 410075, China; m13875930510@163.com (P.Z.); jiawu5110@163.com (J.W.); wll_1234@163.com (L.W.)
* Correspondence: meanzhao@csu.edu.cn

Received: 17 January 2019; Accepted: 29 January 2019; Published: 4 February 2019

Abstract: Due to the dynamic change of the opportunistic network topology and the lack of stable information transmission paths between nodes, the traditional topology-based routing algorithm cannot achieve the desired routing performance. To address this problem, this paper proposes a routing algorithm based on trajectory prediction (RATP). The routing protocol based on trajectory prediction can efficiently and quickly adapt to the network link quality instability and the dynamic changes of network topology. RATP algorithm constructs a node mobility model by analyzing the historical mobility characteristics of the nodes. According to the node prediction information, the metric value of the candidate node is calculated, and the node with the smaller metric value is selected as the data forwarding node, which can effectively reduce the packet loss rate and avoids excessive consumption. Simulation results show that compared with other algorithms, the proposed algorithm has higher data delivery ratio, and end-to-end data delay and routing overhead are significantly reduced.

Keywords: opportunistic networks; routing; trajectory prediction; node mobility model

1. Introduction

An opportunistic network is a kind of network that can construct communication services between nodes with the network structure fracture and topology dynamic change. It has extensive research and application value in military and civilian fields [1,2].

At present, the main research hotspots of the opportunistic networks are: routing algorithms, network information security [3], energy research, mobility model research and so on. The topology of the opportunistic networks changes dynamically and no relatively stable data transmission path between nodes. This makes it difficult to achieve more efficient performance based on the change of network topology [4]. Therefore, how to find a routing protocol with good routing performance is one of the most challenging problems in opportunistic networks research. Opportunistic network routing protocols face the following challenges:

a. Unevenness of opportunities to encounter between nodes. In the opportunistic network, the information interaction occurs only when two nodes enter each other’s communication range. The chance of encountering between the two nodes is unpredictable, which leads to the imbalance of the chances of encountering the nodes.

b. Topology structure information is difficult to grasp. The topology of the opportunistic network changes dynamically, and the mobile information of nodes in the network is difficult to obtain accurately and timely.

c. Resource constraints. Most of the opportunistic network nodes are portable devices, so the nodes themselves have limitations such as energy, contact time, and storage space.
d. Unpredictability of node movement trajectory. The mobile behavior of a node is affected by many external factors, and it is impossible to obtain the precise location information of the node in time.

At present, researchers have proposed many opportunistic network routing protocols. The opportunistic network information transmission mechanism is “store-carry-forward”. According to the routing strategy, it can be divided into three categories: code-based, scene-based and switched-based routing [5]. However, they did not consider the location information of the node. The location information of the node plays an important role in quickly discovering the information of the next-hop forwarding node. At present, there are some location-based routing algorithms [6,7]. In these algorithms, the next-hop forwarding node can be found through the location information of the neighbor and destination nodes, which effectively improves the routing performance and has strong adaptability to the dynamic change of the network topology. The DIG (Direction based geographic routing scheme) [8] algorithm is a representative location-based algorithm. However, the algorithm does not fully utilize the location information of the neighbor nodes, which makes the delay performance unsatisfactory and requires improvement. The LAR (Location Aided Routing) [9] algorithm and the LOTAR (Location Trace Aided Routing) algorithm [10] establish a source node to a destination node route by directly utilizing location information, thereby transmitting data packets. GPSR (Greedy Perimeter Stateless Routing) [11] algorithm and GLS (Grid Location Service) [12] algorithm also transmit information by using node location information, but the two algorithms do not establish a data transmission path from the source node to the destination node.

To solve the problems, this paper proposes a routing algorithm based on trajectory prediction. This approach is based on the historical mobility characteristics of the nodes to establish the mobility model of the node. Based on the predicted location information, the metric value of the candidate node is calculated, and the node with the smaller metric value is selected as the data forwarding node. The presented method may be successfully used for vehicular ad hoc networks (VANETs).

The main contributions of this article are in the following areas:

(1) Constructing the node mobility model based on the historical mobility characteristics of the node. By changing the value of the node’s mobile coordination factor, the possible running speed of the node at the next moment is obtained.

(2) Using the Gaussian process to model the speed probability of different mobile coordination factors, and using the maximum likelihood estimation (EM) algorithm to obtain the model parameters, so that the probability of speed based on historical mobility characteristics is best. On this basis, the node movement distance and node location information are calculated.

(3) Calculating the metric value of the candidate node based on the predicted location information. During the information transmission, the node does not need to store too much routing information, and the network resource consumption is significantly reduced. The use of the location information can effectively avoid loop generation during data delivery and also have strong adaptability to the dynamic network topology change.

(4) The simulation platform ONE simulates a large amount of data and evaluates the performance of the RATP algorithm, with improved transmission success rate, reduced data delay and routing overhead.

The rest of this paper mainly includes the following parts: the second part analyzes some existing research works; the third part analyzes and describes the algorithm; simulation results are presented in Section 4. The last section concludes this paper.

2. Related Work

Many previous research results have shown that the acquisition of node location information has a very significant impact on the performance of the opportunistic network routing. At present, the way to obtain location information of nodes is mainly through location system. When a node transmits information, it only needs to obtain the location information of the neighbor and destination
node to select the appropriate next-hop node, thus reducing the network resource overhead. Nodes only need to obtain location information, which can improve the utilization of network bandwidth. The distance between neighbors can be calculated by using location information, which can save energy consumption [13].

In recent years, the routing algorithm based on location information has many advantages over the routing algorithm based on network topology, so the routing protocol based on location information has attracted the attention of many researchers [14–16]. According to the routing strategies, location-based routing can be roughly divided into two categories: proactive routing protocol and reactive routing protocol. Among them, proactive routing protocols represent algorithms such as LOTAR, GPSR, DREAM, and reactive routing protocols represent algorithms such as LAR, GLS, DIG, etc. LAR routing protocol is a location aided routing algorithm. Based on the node location information, the algorithm finds the path of data transmission so as to narrow the routing search area and improve the data delivery ratio. In the LAR [9] algorithm, each node carries a positioning system that can obtain its current location information at any time, and assumes that each node in the network knows the mobility rate of other nodes. GLS [12] is a hierarchical routing protocol consisting of two parts: location information acquisition and data forwarding. The GLS algorithm divides the whole network into several small grids. The nodes in the grid periodically broadcast their own geographical location information. The other nodes update their own location information table according to the broadcast information. The nodes outside the grid act as the location information server through the selected nodes in the grid. By using these nodes to update and query the location information in time, the neighbor nodes closest to the destination node are selected as the next-hop data forwarding node. DREMA (Distance routing effect algorithm for mobility) [17] algorithm shows that each node in the network carries its own location information and periodically broadcasts the information. Other nodes in the network receive the location information, update and store the new location information in their own location information table. This method can effectively reduce the network control information overhead and ensure the accuracy of the link information. Although the algorithm has very good performance, it is not suitable for networks with relatively large network topology, and the performance of the algorithm will gradually decrease when the path between nodes does not exist.

The approach in [18] proposes a Centrality History-Based Prediction Routing Protocol (CHBPR). The algorithm focuses on the use of the concept of centrality to achieve information transfer between nodes. Unlike other nodes in the network, the central node is more likely to act as a communication hub for message forwarding. Dhurandher et al. [19] proposed a centrally-based mechanism (HiBOp) to improve the performance of history-based routing protocols. GEAR [20] is an algorithm of greedy forwarding and energy efficiency. The core idea of the GEAR is to combine the energy information of each neighbor node with the distance between the neighbor node and the destination node when selecting the next-hop data forwarding node. GR (Greedy Routing) [21], the selection of the next-hop forwarding node must satisfy that the node is the closest distance to the target node among all neighbor nodes, and the distance is smaller than the distance from the current node to the target node. By searching the network, the node satisfying the condition is found as the optimal next-hop data forwarding node for data delivery.

Reference [22] proposed a new method to find the nearest neighbor from the current sending node in a given angle range as the next-hop node. In the routing algorithm based on location information, the greedy data forwarding strategy is the most efficient and simple. Based on the definition of the particle fitness function with neighbor node information, a single-hop routing clustering protocol (DPSOCA) based on discrete particle swarm optimization is proposed in [23]. Reference [24] proposed a new routing protocol called greedy peripheral coordinator routing (GPCR) algorithm, which does not require external information of the node, nor the entire network information. It only realizes the data forwarding between nodes by coordinating the relationship between nodes. However, it is easy for this selection mechanism to fail in the case of low node density. The improved greedy traffic
sensing routing protocol [25] is mainly characterized by considering node density before forwarding data packets.

3. RATP

Each node carries its own location information (LM) and neighbor node information table (NT), and broadcasts the information at a certain interval. Where the LM information is expressed as follows: LM (Location, velocity) indicates location information, which refers to a message that carries location information and can be broadcast within a certain range. Location: it is indicated by the two-dimensional coordinates of node \((x_i, y_i)\), while the velocity is specified by the current moving speed value of the node and the direction of node movement.

3.1. Trajectory Prediction

3.1.1. Node Mobility Model

In this paper, through the LM information carried by the node itself, the mobility characteristics of the node are analyzed, and the appropriate node mobility model is constructed. The average node moving speed in a long period of time is calculated by analyzing the node moving speed.

\[
\mathcal{J} = \lim_{t \to \infty} \left( \frac{\sum_{t=0}^{\infty} v_i}{t} \right) 
\]

We assume that the node mobility satisfies the Gaussian distribution. According to the historical movement information of the node, the motion state of the node in the future motion state is analyzed. Use \(\delta_n\) to indicate the state of the node moving in the future.

\[
\delta_n = \frac{1}{\lambda \sqrt{2\pi}} e^{-\frac{\lambda^2}{2\lambda^2}}
\]

In the above formula, \(\lambda\) is the node mobile coordination factor, satisfies \(0 \leq \lambda \leq 1\). The magnitude of \(\lambda\) represents the mutation of node speed in a short time, and it is a specific performance of the node itself mobility performance. Therefore, the node mobility model can be defined as follow:

\[
v_{t+\Delta t} = ((\lambda - 1)v_i + (1 - \lambda)J + \sqrt{1 - \lambda^2} \delta_n)\Delta t
\]

At \(t + \Delta t\) moment, the value of the node’s moving speed \(v_{t+\Delta t}\) is obtained by the influence of the average node moving speed in a long period of time, the random state of the node moving in the future and the moving coordination factor on the speed.

However, a simple next-time velocity value does not produce a relatively high and stable prediction accuracy for node trajectory prediction, and different mobile coordination factor values will have a corresponding impact on the next-time moving speed value.

3.1.2. Probability of Node Velocity

Suppose that each data point \(v_i\) can be obtained by transformation the data point \(\lambda_i\) through \(\Lambda \lambda_i + \mu\) (\(\Lambda\) is a matrix) deformation. Because the historical track GPS data often have noise, we can assume that the noise data also obeys the Gauss distribution, that is \(\chi \sim N(0, \psi)\), so the predicted data is \(\mu + \Lambda \lambda_i + \chi\). The trajectory probability model can be obtained as follows:

\[
p(V|\lambda) = \prod_{i=1}^{k} \sum_{j=1}^{k} \sigma_j \varphi(v_j|\mu_j + \Lambda \lambda_j, \psi_j)
\]
where \( \sigma_j \) is the weight of the \( j \) Gaussian process and satisfy \( \sigma_j \geq 0, \sum_{j=1}^{k} \sigma_j = 1 \), \( \phi(\cdot) \) represents the Gauss process probability density function, and \( k \) represents the number of Gauss processes. Using the Gauss hybrid model to estimate the parameter \( \sigma, \mu, \Lambda, \psi \) accurately. In general, the expectation-maximization algorithm is used to estimate the parameters of the GMM model. A set of parameters was found through iterative training to maximize the value of \( p(V|\lambda) \), that is:

\[
(\sigma, \mu, \Lambda, \psi) = \arg \max_{(\sigma, \mu, \Lambda, \psi)} p(V|\lambda)
\]

(5)

Because each \( x_i \) is transformed by the corresponding \( \lambda_j \), plus the noise, so \( V \) is the posterior distributions of \( \lambda: V|\lambda \sim N(\mu_v + \Lambda \lambda_v, \psi_v) \). Therefore, \( \lambda \) and \( V \) obey the joint Gaussian distribution:

\[
\left[ \begin{array}{c} V \\ \lambda \end{array} \right] \sim N(\mu_v\lambda, \Sigma)
\]

(6)

Due to the fact that \( \lambda \) follows a normal distribution with a mean of zero, therefore \( E(\lambda) = 0 \). All elements of the matrix \( \Lambda \) are constants, and their expected value is still itself, so you can get \( \mu_v\lambda = \left[ \begin{array}{c} 0 \\ \mu \end{array} \right] \).

Solving \( \Sigma \) requires calculation

\[
\Sigma_{\lambda\lambda} = E[(\lambda - E[\lambda])(\lambda - E[\lambda]^T)]
\]

(7)

\[
\Sigma_{v\lambda} = E[(\lambda - E[\lambda])(v - E[v]^T)]
\]

(8)

\[
\Sigma_{vv} = E[(v - E[v])(v - E[v]^T)]
\]

(9)

\[
\Sigma_{\lambda\lambda} = \text{Cov}(\lambda) = 1, \text{E}[(\lambda - E[\lambda])(\lambda - E[\lambda]^T)] = \Lambda^T. \text{And } \text{E}[(v - E[v])(v - E[v]^T)] = \Lambda \Lambda^T + \psi \text{ can be obtained from the } \text{E}[\Lambda \lambda^T] = \text{Cov}(\lambda), \text{E}[\lambda \chi^T] = \text{E}[\lambda]E[\chi^T] = 0. \text{ So the edge distribution of } V \text{ is as follows: } v \sim N(\mu, \Lambda \Lambda^T + \psi) \text{ and the conditional distribution } v \sim N(\mu + (\Lambda \Lambda^T + \psi)\Lambda^{-1}\lambda, \Lambda - (\Lambda \Lambda^T + \psi)\Lambda^{-1}). \text{ Set of training trajectories data sets } \{v_{ij}; i = 1, 2, \ldots, k\} \text{, the logarithmic likelihood function of the parameter } \ell, \lambda, \psi \text{ is established.}
\]

\[
\log p(V) = \log \prod_{j=1}^{k} \frac{1}{(2\pi)^{k/2}|\Lambda \Lambda^T + \psi|^{1/2}} \exp(-\frac{1}{2}(v_j - \mu)^T(\Lambda \Lambda^T + \psi)^{-1}(v_j - \mu))
\]

(10)

The following requires the probability of a further motion pattern \( i \), which requires calculation of the parameter \( \sigma \) and the value obtained when the partial derivative of the parameter \( \mu, \Lambda, \psi \) in Equation (10) is zero. Use EM algorithm solution, divided into two steps E-step (seeking expectations) and M-step (seeking extreme value), the specific method is as follows:

**Step E:** The probability that the node mobile coordination factor is \( \lambda_j \) and the node speed is \( v_j \).

\[
p(j|v_j, \lambda) = \frac{\sigma_j p(v_j|j, \lambda)}{p(v_j|\lambda)} = \frac{\sigma_j \phi(v_j|\mu_j + \Lambda \lambda_j \psi_j)}{\sum_{i=1}^{k} \sigma_j \phi(v_j|\mu_i + \Lambda \lambda_j \psi_i)}
\]

(11)

M-step: using the maximum expectation method to obtain the parameter \( \sigma \) iterative estimation formula:

\[
\sigma = \sum_{j=1}^{k} p(j|v_j, \lambda)
\]

(12)

Maximizing the likelihood function of the parameter \( \mu, \Lambda, \psi \), find the iterative estimation formula of the parameter \( \mu, \Lambda, \psi \):
The coordinate of node location can be obtained by the node coordinate location at \( t \) moment (through LM information) and node moving velocity vector angle.

The location of the node at time \( t \) can be identified by coordinates \( L_t(x_t, y_t) \), so at time \( t + \Delta t \), the geographic location of the node can be predicted based on the speed \( v_t \), node position \( L_t(x_t, y_t) \), direction \( \beta_t \) (speed vector change angle) and speed \( v_{t+\Delta t} \) at time \( t+\Delta t \).

If in \( \Delta t \) time, the node driving direction has changed:

\[
\begin{align*}
(x_{t+\Delta t}, y_{t+\Delta t}) = \begin{cases} 
  x_{t+\Delta t} = x_t + \left( \frac{v_{t+\Delta t} + v_t}{2} \right) \Delta t \cos \beta_t \\
  y_{t+\Delta t} = y_t + \left( \frac{v_{t+\Delta t} + v_t}{2} \right) \Delta t \sin \beta_t 
\end{cases}
\end{align*}
\]

If in \( \Delta t \) time, the node driving direction has not changed:

\[
\begin{align*}
(x_{t+\Delta t}, y_{t+\Delta t}) = \begin{cases} 
  x_{t+\Delta t} = x_t + \left( \frac{v_{t+\Delta t} + v_t}{2} \right) \Delta t \\
  y_{t+\Delta t} = y_t + \left( \frac{v_{t+\Delta t} + v_t}{2} \right) \Delta t 
\end{cases}
\end{align*}
\]

3.1.4. Node Moving Distance

The moving distance of the node means the length of the node moves from moment \( t \) to moment \( t + \Delta t \), which can be denoted by \( \Delta D_s \). Since the movement of the node in a short period of time approximates a uniform motion, so the distance can be expressed by the displacement formula of the physics:

\[
\Delta D_s = D_{t+\Delta t} - D_t = \left( \frac{v_{t+\Delta t} + v_t}{2} \right) \times \Delta t
\]

where \( D_{t+\Delta t} \) represents the total moving distance of node in \( t \) time. The \( D_{s_{t+\Delta t}} \) represents the total moving distance of node in \( t + \Delta t \) time.

\[
D_{s_{t+\Delta t}} = D_s + \frac{1}{4} \times \left[ \theta(v_{t+\Delta t} + v_t) + (1 - \theta)(\eta_{t+\Delta t} + \eta_t) + \sqrt{1 - \theta^2(\xi_{t+\Delta t} + \xi_t)} \right] \Delta t^2
\]

3.2. Data Forwarding Mode

From Section 3.1, the predicted location is obtained. As shown in Figure 1, Node A is the current node and Node D is the destination node. The distance from Node B to line AD is as follows:

\[
d_{B-AD} = \frac{|(D_2 - x_2)B_1 + (x_1 - D_1)B_2 + (d_1 - x_1)(x_2 - x_1)|}{d_0}
\]
The formula of the relative deflection angle is:

\[ \alpha = \arcsin \frac{d_{B-AD}}{d}, \quad \alpha \in \left[0, \frac{\pi}{4}\right] \]  \hspace{1cm} (20)

Calculating the metric value of the candidate node based on the predicted location information. Then, the candidate node with the smallest metric will be selected as the next-hop data forwarding node. The metric value calculation formula is as follows.

\[ \varsigma_j = \tau_a \frac{d}{d_0} + \frac{4\alpha}{\pi} + \tau_c (1 - \frac{n}{N}) \]  \hspace{1cm} (21)

where \( \tau_a, \tau_b, \text{ and } \tau_c \) are distance factor, angle factor and neighbor node factor. The \( d \) is the Euclidean distance between the candidate node and the destination node, \( d_0 \) is the Euclidean distance between the current node and the destination node, \( \alpha \) is the deflection angle of the connecting line between the alternative node and the destination node, \( n \) is the number of candidate nodes, and \( N \) is the total number of network nodes. Select a node that is closer to the destination node to reduce the number of route hops. The neighbor nodes with the smallest deflection angle are selected to reduce the appearance of boundary effect. Select a node with an alternate node to reduce the probability of encountering routing holes. Different decision factors have different effects on the routing effect.

According to the NT table, the metric \( \varsigma_j \) of each node is calculated, and the node with the smallest metric be selected as the next-hop data forwarding node. The data forwarding mode can effectively reduce the packet loss rate of the boundary, avoid excessive consumption of edge nodes, and reduce the probability of encountering routing holes.

3.3. Algorithm Complexity Analysis

According to the above algorithm description, the algorithm RATP (Algorithm 1) pseudo code as follows:
Algorithm 1. Routing Algorithm based on Trajectory Prediction—RATP

Input: Historical trajectory information of nodes
Output: optimal path

1: $\vartheta$: the average speed of vehicle since long time movement;
2: $\delta_n$: vehicles’ random and independent processes when the vehicles move at infinity;
3: $\lambda$: the mobile coordination factor;
4: the node mobility model:
   \[ v_{t+\Delta t} = ((\lambda - 1)v_t + (1 - \lambda)\theta + \sqrt{1 - \lambda^2}\delta_n)\Delta t; \]
5: each data point $v_i$ can be obtained by transformation the data point $\lambda_i$ through $\Lambda \lambda_j + \mu$;
6: the trajectory probability model:
   \[ p(V|\lambda) = \prod_{i=1}^{k} \sum_{j=1}^{k} c_j \psi(v_j|\mu_j + \Lambda \lambda_j, \psi_j); \]
7: use the EM algorithm to optimize the parameter $\sigma, \mu, \Lambda, \psi$;
8: get the speed of the next moment.
9: the vehicle position:
10: if the vehicle driving direction has changed do
    Equation (15);
11: else:
    Equation (16);
12: end if
13: the distance of the vehicle moving in $t + \Delta t$ time: Equation (18);
14: calculate the metric value of the candidate node based on the predicted location information:
   \[ \varsigma_j = \tau_a d_0 + \tau_b \frac{4a}{\pi} + \tau_c (1 - \frac{n}{N}); \]
15: the candidate node with the smallest metric be selected as the next-hop forwarding node;
16: end;

The complexity of the algorithm is the resource required by the algorithm after it is written into an executable program. The resources include time resources and memory resources. The same problem can be solved by different algorithms, and the quality of an algorithm will affect the efficiency of the algorithm and even the program. The purpose of algorithm analysis is to select appropriate algorithm and improve algorithm. The evaluation of an algorithm is mainly considered from the time complexity and space complexity. The RATP algorithm uses the Gaussian process to calculate the probability of the node’s future time velocity, and uses the candidate node metric to calculate the next-hop data forwarding node. The complexity of the algorithm is \( O(n + \log n) \).

4. Performance Evaluation

The simulation experiment platform is ONE, and the RATP is compared with Greedy protocol, CHBPR protocol and GPCR protocol. To accurately evaluate our protocol, we selected smaller areas from the 250 $\times$ 250 km trajectory area because the simulation of 200,000 nodes makes the simulation experiment extremely slow. The simulation experiment scene model uses 5 $\times$ 5 Manhattan model, and the simulation scene is 2000 $\times$ 2000 m. All nodes were equipped with 802.11b wireless radio interfaces. The communication distance between nodes was 10 m and all the broadcasts were set to the same channel frequency. In this paper, through the results of data training and simulation experiments the weight values of each factor were set to $\tau_a = 0.8, \tau_b = 0.1, \tau_c = 0.1$ in order to ensure the association between nodes and data forwarding probability, which can significantly improve the performance of experimental results.
4.1. Performance Measurement Parameters

In order to better verify the performance of the RATP algorithm, the following metrics are used for evaluation:

**RMSE**: It is used to calculate the geometrical space error between the real and the predicted position, that is, the geometric space errors between the predicted trajectory points and the actual trajectory points.

\[
RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^{N} (real_j - pred_j)^2}
\]

(22)

where \(real_j\) is the real location \((x_{real}, y_{real})\), \(pred_j\) is the predicted location \((x_{pred}, y_{pred})\), and \(N\) is the number of predicted trajectory points.

**Delivery ratio**: The number of packets successfully received by all the destination nodes in the network accounts for the ratio of the number of packets sent by the source node within a certain time. The higher the value, the higher the transmission success rate in the network.

**End-to-end delay**: The end-to-end delay is the average time that packets take place from the initial node to the destination node.

**Routing overhead**: The total number of forwarded packets (FoM) minus the number of packets successfully received by the destination node (ReM), and then divided by the number of packets successfully received by the destination node. The formula is as follows:

\[
ROR = \frac{FoM - ReM}{ReM}
\]

(23)

4.2. Results and Discussion

4.2.1. Prediction Error Analysis

Aiming at the simulation experiment, the performance of the algorithm is analyzed. The influence of different mobile coordination factors on error RMSE is introduced.

As shown in Figure 2, as the value of \(\lambda\) increases, the RMSE error gradually decreases. When the value of \(\lambda\) is greater than 0.75, RMSE shows a slow growth trend. The \(\lambda\) represents the node’s mobile performance. In real life, the movement of the node requires a corresponding time to adapt to the mobile coordination performance. As can be seen from Figure 2, the value has a certain influence on predicting the speed probability of the next moment. Therefore, it is not that the greater the value of \(\lambda\), the better the performance of the algorithm.

![Figure 2. RMSE vs value of \(\lambda\).](image-url)
4.2.2. Delivery Ratio

This part mainly analyzes the performance of data delivery ratio of different algorithms. The change law of data delivery ratio with simulation time length is introduced. Additionally, the effect of the number of nodes on the data delivery ratio is studied.

In the same simulation scenario mode, the delivery ratio performance of different algorithms is studied. For the simulation results of each algorithm, we can find that the following Figure 3 mainly has five signs (max, the third quartile, median, the first quartile and min). As shown in Figure 3, in some special cases, two or more signs have very similar values. For example, the algorithm RATP proposed in this paper, the third quartile and median values are roughly around 0.01, which is very close. As can be seen from the Figure 3, although CHBPR has a high delivery ratio, it only considers the density of nodes and does not consider the distribution relationship between nodes, resulting in poor connectivity, making the data unable to be more efficient and rapid transport to the destination node, delivery ratio decline. Compared with other algorithms, the RATP algorithm has a higher delivery ratio because the algorithm not only analyzes the different location probabilities, but also analyzes the positional relationship among the node at the same time, the better the connectivity of selected routes, the delivery ratio is higher.

![Figure 3. Delivery Ratio.](image)

Figure 4 shows the data delivery ratio of the RATP algorithm, CHBPR algorithm, GPCR algorithm and Greedy algorithm under different simulation times. It can be seen from the figure that as the simulation time increases, the data delivery ratio of the four algorithms also increases. The experimental results show that the RATP algorithm has a low data delivery ratio when the simulation time is between 1.5 h and 2.5 h. This is because a large amount of data is dropped, causing congestion in the network. However, RATP still has a higher data delivery ratio than other algorithms.

Figure 5 shows the data delivery ratio of the RATP algorithm, CHBPR algorithm, GPCR algorithm and Greedy algorithm under different node numbers. As can be seen from Figure 5, as the number of nodes increases, the RATP algorithm has a higher data delivery ratio than other algorithms. The simulation results show that when the number of nodes is within a certain range, the data delivery ratio increases with the number of nodes. As the number of nodes increases, the chances of encountering between nodes increases, network connectivity increases, and information transmission efficiency increases. However, when the number of nodes exceeds a certain range, the number of neighbor nodes increases due to the increase of the node density, thereby increasing the probability of the selected next-hop node, and finally the data delivery ratio is lowered.
This is because, as the simulation time increases, the number of packets in the network increases, so the entire network takes more time to process, resulting in an increase in data transmission delay.

In any case, the RATP algorithm average end-to-end delay is significantly slower than the other algorithms. The change law of average end-to-end delay with simulation time length is introduced and the effect of the number of nodes on the average end-to-end delay is studied.

Figure 6 shows the average end-to-end delay of the RATP algorithm, CHBPR algorithm, GPCR algorithm and Greedy algorithm under different simulation times. It can be seen from the figure that as the simulation time increases, the average end-to-end delay of the four algorithms also increases.

4.2.3. End-to-End Delay

This section mainly analyzes the average end-to-end delay performance of four different algorithms. The change law of average end-to-end delay with simulation time length is introduced and the effect of the number of nodes on the average end-to-end delay is studied.

Figure 6 shows the average end-to-end delay of the RATP algorithm, CHBPR algorithm, GPCR algorithm and Greedy algorithm under different simulation times. It can be seen from the figure that as the simulation time increases, the average end-to-end delay of the four algorithms also increases. This is because, as the simulation time increases, the number of packets in the network increases, and the entire network takes more time to process, resulting in an increase in data transmission delay. In any case, the RATP algorithm average end-to-end delay is significantly slower than the other three algorithms.
Additionally, the effect of the number of nodes on the routing overhead is studied.

Figure 6. End-to-end delay vs number of nodes.

Figure 7 shows the average end-to-end delay of the RATP algorithm, CHBPR algorithm, GPCR algorithm and Greedy algorithm under different node numbers. It can be seen from the figure that as the number of nodes increases, the average end-to-end delay of the four algorithms increases first and then decreases. When the number of nodes is small, the selected forwarding node is far away, resulting in an increase in data transmission delay. When the number of nodes reaches a certain level, the number of forwarding nodes in the network increases, the data transmission time is relatively reduced, and the transmission delay is relatively reduced. Compared with the other three algorithms, the RATP algorithm has the lowest average end-to-end delay. This is because the RATP analyzes the distribution of the entire network node and the global information of the network, making the selected routing more connected, the higher the reliability and real-time, the lower the delay.

Figure 7. End-to-end delay vs number of nodes.

4.2.4. Routing Overhead

This section mainly analyzes the performance of the routing overhead of the four different algorithms. The change law of routing overhead with simulation time length is introduced. Additionally, the effect of the number of nodes on the routing overhead is studied.

Figure 8 shows the routing overhead of the RATP algorithm, the CHBPR algorithm, the GPCR algorithm, and the Greedy algorithm at different simulation times. It can be seen from the figure that
as the simulation time increases, the routing overhead of the four algorithms presents an unstable state. The RATP algorithm selects the next-hop data forwarding node by analyzing the node moving position. When the information is forwarded to the next-hop node, it can sense the location information of other nodes in advance and forward the information to the appropriate node. Therefore, path discovery time is reduced, routing overhead is reduced, and route cost changes are more stable than other algorithms.

Figure 8 shows the routing overhead vs number of nodes. It can be seen from the figure that as the number of nodes increases, the routing overhead of the four algorithms increases gradually. This is because when the number of nodes increases, the path propagation fails more frequently. As a result, the number of nodes passing through the message increases relatively and the path is found to be relatively slow. Since the RATP algorithm can further select the information forwarding path by analyzing the location information of the node, the algorithm can obtain other mobile node location information in advance and select an appropriate node to forward the information, so the routing cost is lower than other algorithms.

Figure 9 shows the routing overhead of the RATP algorithm, CHBPR algorithm, GPCR algorithm, and Greedy algorithm under different node numbers. It can be seen from the figure that as the number of nodes increases, the routing overhead of the four algorithms increases gradually. This is because when the number of nodes increases, the path propagation fails more frequently. As a result, the number of nodes passing through the message increases relatively and the path is found to be relatively slow. Since the RATP algorithm can further select the information forwarding path by analyzing the location information of the node, the algorithm can obtain other mobile node location information in advance and select an appropriate node to forward the information, so the routing cost is lower than other algorithms.
5. Conclusions

The opportunistic network has the characteristics of topological dynamics and bandwidth first. The good routing protocol is the primary problem of constructing the opportunistic network, and it is also a hotspot and a difficult point of research. This paper proposes a routing algorithm based on trajectory prediction (RATP). The RATP algorithm uses the Gaussian process to calculate the probability of the node’s future time velocity by analyzing the historical trajectory characteristics of the node. According to the predicted location information, the candidate node metric value is calculated and the node with the smaller metric value is selected as the data forwarding node, which effectively reduces the packet loss rate of the boundary and avoids excessive consumption of the edge node. The simulation results show that compared with other algorithms, the algorithm has higher transmission success rate, and the transmission delay and routing overhead are significantly reduced.

In future work, first of all we hope to realize the position prediction algorithm and routing protocol in practical applications, and verify the effectiveness of the prediction algorithm and routing protocol in this paper by using real application scenarios. Secondly, we hope to consider the constrained factors of more nodes, so as to select a better forwarding node to achieve higher routing performance.

Author Contributions: Conceptualization, Methodology, Analysis, Investigation, Writing—original draft preparation, Writing—review and editing, Visualization, P.Z., L.W. and J.W.; Supervision, Project Administration, Funding Acquisition, M.Z.

Funding: This research was funded by the National Natural Science Foundation of China (No.61572526).

Conflicts of Interest: The authors declare no conflicts of interest.

References

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).