


Article

# A Novel Opportunistic Network Routing Method on Campus Based on the Improved Markov Model

Yumei Cao<sup>1,2,3,4</sup>, Peng Li<sup>1,2,3,4,5,\*</sup> , Tianmian Liang<sup>1,2,3,4</sup>, Xiaojun Wu<sup>1,2,3,4,5</sup>, Xiaoming Wang<sup>1,2,3,4,5</sup> and Yuanru Cui<sup>1,2,3,4</sup>

<sup>1</sup> Key Laboratory of Intelligent Computing and Service Technology for Folk Song, Ministry of Culture and Tourism, Xi'an 710119, China; cym41712121@snnu.edu.cn

<sup>2</sup> School of Computer Science, Shaanxi Normal University, Xi'an 710119, China

<sup>3</sup> Engineering Laboratory of Teaching Information Technology of Shaanxi Province, Xi'an 710119, China

<sup>4</sup> Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an 710062, China

<sup>5</sup> Xi'an Key Laboratory of Cultural Tourism Resources Development and Utilization, Xi'an 710062, China

\* Correspondence: lipeng@snnu.edu.cn

**Abstract:** Opportunities networks' message transmission is significantly impacted by routing prediction, which has been a focus of opportunity network research. The network of student nodes with smart devices is a particular type of opportunity network in the campus setting, and the predictability of campus node movement trajectories is also influenced by the regularity of students' social mobility. In this research, a novel Markov route prediction method is proposed under the campus background. When two nodes meet, they share the movement track data of other nodes stored in each other's cache in order to predict the probability of two nodes meeting in the future. The impact of the node within the group is indicated by the node centrality. The utility value of the message is defined to describe the spread degree of the message and the energy consumption of the current node, then the cache is managed according to the utility value. By creating a concurrent hash mapping table of delivered messages, the remaining nodes are notified to delete the delivered messages and release the cache space in time after the messages are delivered to their destinations. The method suggested in this research can successfully lower the packet loss rate, minimize transmission latency and network overhead, and further increase the success rate of message delivery, according to experimental analysis and algorithm comparison.

**Keywords:** routing prediction; opportunistic network; utility value; cache management; centrality



**Citation:** Cao, Y.; Li, P.; Liang, T.; Wu, X.; Wang, X.; Cui, Y. A Novel Opportunistic Network Routing Method on Campus Based on the Improved Markov Model. *Appl. Sci.* **2023**, *13*, 5217. <https://doi.org/10.3390/app13085217>

Academic Editors: Yang Yue, Runzhou Zhang, Hao Feng, Zheda Li, Lin Zhang and Dawei Ying

Received: 8 March 2023

Revised: 16 April 2023

Accepted: 17 April 2023

Published: 21 April 2023



**Copyright:** © 2023 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 (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

### 1.1. Background

An opportunistic network is a type of self-organizing network [1], which employs node movement to create opportunities for encounter and communication rather than requiring an end-to-end complete connection. The increased usage of portable electronics, including smartphones and tablets, opens up numerous possibilities for the growth of opportunistic networks. Opportunistic networks use the store-carry-forward routing mode and send messages hop by hop between nodes [2]. In order to increase network communication efficiency and achieve timely message transmission, it is now essential to accurately predict node destinations, choose the best next-hop nodes, and reduce the number of message copies in the network, owing to the limited memory and power of portable devices. The transmission path can be planned based on the nodes' geographical locations to address the issues of limited energy and storage overhead in the network [3]. The issue of excessive network resource consumption can be resolved by managing data congestion [4,5] and restricting the transmission range of nodes [6].

### 1.2. Motivation

A wireless self-organizing network made up of learner nodes using smart devices on campus, the campus opportunistic network allows each learner node to communicate with other nodes. Although many academics have made progress in this area, few have concentrated on the campus context and have failed to suggest efficient routing algorithms or manage the cache in a way that is suitable for the specific movement of nodes in the campus. The predictability of node movement trajectories is also determined by how regularly learner nodes travel around campus.

The blind nature of message delivery is one of the major issues when performing message transmission. If the node delivers the message directly to its neighbors, this results in a significant message transmission delay, as messages must pass through numerous relay nodes in order to reach their destinations and they cannot deliver to the recipient before the message survival time ends sometimes. On the other hand, network resources are wasted because some nodes carrying copies of messages do not eventually connect with the destination node, resulting in a large number of duplicate messages in the network that are truly unnecessary. Additionally, since copies of individual messages are forwarded numerous times, they can take up a lot of memory, preventing the reception of recently arrived messages. Since nodes typically have limited memory, it is important to manage the cache because a multi-copy-based routing strategy leaves a lot of copies in the network. This can even have a significant impact on the success rate of message delivery, making it a vital study in this paper. A certain amount of message copies exists in the network to ensure that messages get to their destination quickly, the network needs a certain number of message copies, so when the node cache is too small to accommodate new messages, it should choose which messages to delete initially to make room for new arrivals.

### 1.3. Contribution

We suggest improved Markov path prediction and cache management algorithms on the basis of some of the aforementioned difficult issues. When designing the algorithm, we make an effort to deliver messages as quickly as feasible. Considering that device memory is typically small, then we suggest a proven cache management technique. The following are the major contributions of this paper.

- In this study, we distinguish between intra-group forwarding and extra-group forwarding when it comes to messaging. When a message needs to be sent between groups, we use a novel Markov model to determine the probability that the sender and the recipient will be in the same place. We then send the message to the nodes with a higher probability of doing so. The message only needs to be delivered within the group when the recipient and the source node are both members of the same group. This not only gets the message to its target quickly but also saves a significant amount of cache space by sending the message to those nodes that have high centrality within the group.
- The utility value of a message is defined in terms of both the message's degree of diffusion and the present node's energy usage. According to our theory, if a message has a high degree of diffusion, there are likely already some copies of it in the network. As a result, priority should be given to receiving messages with a lower degree of diffusion. Moreover, if a message requires a lot of energy from the current node that node might not be the best choice to serve as its relay.
- The node also keeps track of both the message list and the delivered message list, prompting the node to remove any messages that have already been delivered.
- Our suggested strategy enhances network performance in terms of packet delivery rate, average delivery delay, average cost and overhead when compared to current methods.

The remaining portions of the paper are structured as follows: related work is discussed in Section 2. Section 3 improves Markov's method and mentions the way messages are forwarded in the network. Section 4 defines message utility values and proposes cache

management methods. Section 5 evaluates the experimental results. Section 6 concludes the paper.

## 2. Related Works

In opportunity networks, predicting node paths, managing caches reasonably, and reducing energy consumption have become research hotspots. Researchers at home and abroad have made many research results in related directions and proposed some solutions to the problems of path prediction and cache management in opportunity networks.

Singh et al. [7] proposed a social-based opportunistic routing algorithm, which only spreads the content when the social relationship between the next relay node and the destination node is closer than all previously encountered nodes and uses the social relationship to determine the most suitable node to forward the message, which significantly reduces the overhead in the message routing process. The routing method given in [8] used a secure routing protocol based on blockchain to design an integrated protocol, which can effectively protect data security. This routing method can effectively prevent eavesdropping, camouflage, wormholes, black holes, and fabrication attacks. In [9], Sharma et al. fully automated the routing process of opportunistic networks by using an iterative strategy algorithm, modeling the network environment as Markov decision processes, and using strategy iteration to solve the optimal strategy obtained by Markov decision processes to optimize the routing process and maximize the possibility of message delivery. Kumar et al. [10] put forward an innovative routing strategy based on node activity, which considers the previous operation of the node on the message, calculates the confidence level of the node through the past behavior and activity of the node to determine whether it is a good candidate to forward a specific message and reduces the message dropping rate. Gou et al. [11] presented a social network evolution analysis method based on triple, including a prediction algorithm and a quantization algorithm. The algorithm reduces the blindness of message forwarding and unnecessary waste of resources by predicting the connection probability between nodes in the network. Chunyue et al. [12] put forward an algorithm combined with the sleep mechanism, which mainly solves the problem of judging the conditions of sleep state and wake-up time, forces the nodes in a low-energy state to sleep and avoids the nodes from consuming energy quickly. Derakhshanfard et al. [13] proposed a method based on a bitmap, which uses a routing tree based on the bitmap to find the path, and when the tree receives a request to send a message to a designated node, it directly sends a packet, which effectively improves the message delivery rate. Chithaluru et al. [14] studied an energy-efficient opportunistic routing protocol based on adaptive ranking. The residual energy and geographical location of nodes are used to calculate the level, and an efficient forwarding mode based on node level is determined, which improves the effective use of energy in the process of data transmission. Hernández-Orallo et al. [15] proposed a method to minimize the consumption of network resources, using an epidemic diffusion model to evaluate the impact of message expiration time on message transmission, calculating the optimal expiration time and dynamically setting the expiration time, which significantly reduced the buffer usage and energy consumption. Raverta et al. [16] proposed the routing under an uncertain contact plan, extended the single copy routing in the Markov decision-making process to multiple copies, and used multiple copies to model the network state, which effectively improved the message delivery rate. Das et al. [17] used special monitoring nodes to check the behavior of other nodes and routed messages to nodes with sufficient residual energy levels, so that most of the forwarded messages are proportional to the energy level of the receiver, effectively solving the problem that nodes in the network are paralyzed due to energy consumption. Kang et al. [18] proposed an improved hybrid routing protocol combining mobile ad hoc networks and latency-tolerant networks. When the routing path to the destination node is not successfully established by using the ad hoc network protocol, the virtual source node is selected according to the predictability of the delivery of the destination node by the Prophet protocol, and then the delivery rate is effectively increased at the cost

of overhead again. Pirzadi et al. [19] explored a reduced-delivery delay routing (RDR) strategy in disaster relief operations, using a simulated annealing algorithm to optimize the message delivery process in the network and achieve an optimal routing method for efficient message distribution. Mao et al. [20] proposed a fair credit-based routing incentive mechanism (FCIM) that uses incentives to the selfishness problem of nodes, uses some trust mechanisms to avoid nodes from cheating the network and ensures fairness among nodes.

A few of the algorithms discussed in this paper include the following:

- Epidemic [21], which is a flooding-based routing method where a node passes a message copy to every node it encounters. By creating numerous message duplicates, it increases the probability that the message will be delivered when it comes across the destination node. However, a lot of copies use up network resources, such as cache space and node energy.
- Prophet [22] is a method that is frequently used to send messages based on predictions. Two nodes exchange vectors of transmission probabilities for recognized destinations when they come into contact. Messages can be sent to nodes that meet regularly by updating the transmission probability between nodes based on how long it has been since their last encounter. Nevertheless, it ignores the location information of the nodes and the number of encounters between them.
- RDR, which chooses the next-hop node based on the node's estimated latency, estimated speed variation, the direction of motion, available space in the buffer, and previously sent messages. It provides a constrained amount of replicas, reducing the network resource footprint. With this approach, the amount of network resources used can be drastically decreased, and the size of the cache area has less of an impact. However, messages may not be delivered for a long time, and it requires a longer message survival time.
- FCIM, where each relay node is rewarded with some points when the source node sends a message to its target, increases the message delivery rate by motivating selfish nodes to actively participate in message forwarding. Nodes are permitted to engage in some acceptable selfish behaviors under this strategy, such as rejecting messages when the cache is full. However, no more properties are considered, such as the energy consumption of nodes to forward messages.

Three types of node misconduct are discussed by Rehman et al. [23], and they look into how these types of misconduct may affect nine VDTN routing algorithms. The third category of misconduct of nodes is specifically presented. The node reduces the message TTL by storing the message for a long time in its own memory after it has been received. An incentive and punishment strategy is suggested by Rehman et al. [24–26] to incentivize the selfish cluster nodes to forward messages. An active node can raise its reputation by passing messages and engaging in conversations. Nodes that exhibit selfish behavior repeatedly are punished. The reward and punishment mechanism can effectively enhance the degree of cooperation among nodes and improve the probability of packet delivery.

In this paper, some novel studies are made in relation to some of the methods previously mentioned. The innovation points of this paper are specified in Table 1.

The literature listed above has looked at social relationship analysis, route prediction, and energy conservation, but it has not developed a workable route prediction method based on the regularity of student node mobility from the context of campus opportunity networks. These schemes are not applicable to the case of regular group movement of nodes in campus opportunity networks. This paper proposes a novel method for routing campus opportunity networks based on improved Markov which predicts node paths by collecting historical movement trajectories of nodes and models message utility values reasonably based on message diffusion and consumption of node energy to improve cache utilization by deleting delivered messages in time. The efficiency of the suggested method is confirmed in the experimental section of this study by comparison with the RDR [19] and FCIM [20] algorithms, as well as the classical methods Epidemic and Prophet.

**Table 1.** Novelities of this paper.

Limitations of Existing Works	Novelties of This Paper
The previous section describes how nodes can reduce network resource usage by providing a restricted number of copies, but messages with a short survival time may not be delivered.	In this paper, we transmit messages based on the probability that the nodes will meet at the next location which can guarantee the successful transmission of messages in a short time.
The prediction-based routing presented above takes into account the encounter interval of the nodes.	We consider the probability that nodes will meet one another at various places and the number of contacts between nodes.
FCIM considers the caching of networks.	Description of the message’s energy consumption and the network’s degree of message spread was added to the node.
They encourage selfish nodes to engage in collaboration.	Skip selfish nodes to avoid being impacted by them.

### 3. Materials and Methods

#### 3.1. Markov-Based Next Destination Prediction

The storage-forward model of messages in opportunity networks relies on human contact, and the potential for message delivery arises as people move around. There are more options for message delivery when nodes are on their way to the same location as the destination node of a message. This is so that these nodes can send the message to the intended node faster. Some of the symbols used in the text and what they represent are listed in Table 2.

**Table 2.** Lists of the notations used in this paper to represent variables.

Notation	Description
$N_a, N_b$	Node $N_a$ and node $N_b$
$fc(N_a, N_b)$	Probability $N_a$ and $N_b$ meet
$m$	Message $m$
$DC$	Centrality degree of node
$D_m$	Destination node of message $m$

According to the way people move in the campus opportunity network, we make the following provisions:

The number of times a node chooses a location as its next destination in time period, 0 to  $t$  is denoted as  $X(t)$ , which is a stochastic process with  $X(0) = 0$ . Assume:

- Within mutually exclusive time intervals, the number of times that nodes choose the place as a destination point is independent of one another;
- The probability distribution of the number of times  $X(s + t) - X(s)$  that a node chooses this location in period  $(s, s + t]$  is independent of  $s$ , where  $s \geq 0$ ;
- $o(\Delta t)$  is the likelihood that a node will choose the same place more than once in a sufficiently little period of time.

According to the above rules, the number of times  $X(t)$  for a node to select the location before time  $t$  is a stochastic process, and the number of times to select the location before  $t_v (t_v > t)$  in the future only depends on the number of times to select the location at time  $t$ . The overall number the location is selected in period  $[0, t]$  and  $(t, t_v]$  is equal to the number of times the location is selected in  $[0, t_v]$ . From the assumption (1) that the number of times the location is selected in period  $[0, t]$  and period  $(t, t_v]$  are independent of each other, it is known that  $X(t)$  has no posteriority and belongs to the Markov process. The parameter set for the number of choices  $X(t)$  in the above Markov process is  $T = [0, \infty)$  and the state space is  $E = \{0, 1, 2, \dots\}$ . Therefore,  $X(t)$  is a Markov process with continuous time and discrete states.

In addition,  $X(t)$  satisfies the following conditions:

- For a sufficiently small  $\Delta t$ ;

$$P_1(t, t + \Delta t) = P\{X(t, t + \Delta t) = 1\} = \lambda\Delta t + o(\Delta t)$$

where the constant  $\lambda$  is called the intensity of process  $X(t)$ , and  $o(\Delta t)$  is the high-order infinitesimal about  $\Delta t$  when  $\Delta t \rightarrow 0$ .

- Furthermore;

$$\sum P_j(t, t + \Delta t) = \sum P\{X(t, t + \Delta t) = j\} = o(\Delta t)$$

That is, for a sufficiently small  $\Delta t$ , the probability of meeting twice or more in  $(t, t + \Delta t]$  period can be ignored compared with the probability of meeting once.

- $X(0) = 0$ .

For this process, it can be seen that  $\sum_{j=0}^{\infty} P_j(t, t + \Delta t) = 1$ , and combined with (2) and (3) we have:

$$P_0(t, t + \Delta t) = 1 - P_1(t, t + \Delta t) - \sum P_j(t, t + \Delta t) = 1 - \lambda\Delta t + o(\Delta t)$$

Conclusion 1:  $p_{ij}^{nk}(t, t + s)$  denotes the transfer probability function of the above Markov process, i.e., the probability can also be written as  $p_{ij}^{nk}(s)$  that the number of times the  $n$ th node chooses location  $k$  as its destination from 0 to  $t$  period is  $i$ , and the number of times it chooses location  $k$  after  $s$  time is  $j$ , where  $s > 0$ .

From conclusion 1, it follows that:

$$\sum_j p_{ij}^{nk}(s) = 1, i = 1, 2, \dots \tag{1}$$

And stipulates that:

$$p_{ij}(0) = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{2}$$

Conclusion 2:  $q_{ij}^{nk}$  denotes the rate function of the above Markov process and describes the rate of change of the transfer probability function  $p_{ij}^{nk}(s)$  at the zero moments.

From conclusion 2 it follows that:

$$\lim_{t \rightarrow 0^+} \frac{p_{ij}(t) - \delta_{ij}}{t} = q_{ij}, i, j = 0, 1, 2 \dots N \tag{3}$$

According to the definition of the derivative, we can get:

$$\frac{dp_{0j}(t)}{dt} = \lambda p_{0j-1}(t) - \lambda p_{0j}(t), j = 1, 2, \dots \tag{4}$$

$$\begin{aligned} P\{X(s + t) = j | X(s) = i\} &= \frac{P\{X(s)=i, X(s+t)=j\}}{P\{X(s)=i\}} \\ &= \frac{P\{X(s)=i, X(s+t) - X(s) = j - i\}}{P\{X(s)=i\}} \\ &= \frac{P\{X(s)=i\} P\{X(s+t) - X(s) = j - i\}}{P\{X(s)=i\}} \\ &= P\{X(s + t) - X(s) = j - i\} \end{aligned} \tag{5}$$

By assumption (1), the transfer probability function is independent of  $s$ . Therefore,  $X(t)$  is a time-Ziemarkov process.



Calculate  $q_{ij}$  based on Conclusion 1 and Conclusion 2:

$$\begin{aligned}
 p_{ij}(\Delta t) &= P\{X(t + \Delta t) = j | X(t) = i\} \\
 &= P\{X(t + \Delta t) = j, X(t) = i | X(t) = i\} \\
 &= P\{\text{the number of times the location is selected as} \\
 &\quad \text{the next destination in } (t, t + \Delta t] \text{ period is } j - i | X(t) = i\} \\
 &= P\{\text{the number of times the location is selected as} \\
 &\quad \text{the next destination in } (t, t + \Delta t] \text{ period is } j - i\} \\
 &= \begin{cases} \lambda \Delta t + o(\Delta t), j = i + 1 \\ 1 - \lambda \Delta t + o(\Delta t), j = i \\ o(\Delta t), j > i + 1 \\ 0, j < i \end{cases} \tag{6}
 \end{aligned}$$

From conclusion 2, it follows that

$$q_{ij} = \lim_{\Delta t \rightarrow 0^+} \frac{p_{ij}(\Delta t) - \delta_{ij}}{\Delta t} = \begin{cases} \lambda, j = i + 1 \\ -\lambda, j = i \\ 0, j < i \text{ or } j > i + 1 \end{cases} \tag{7}$$

Substituting into conclusion 1 and taking  $i = 0$ , we get

$$\begin{cases} \frac{dp_{0j}(t)}{dt} = \lambda p_{0j-1}(t) - \lambda p_{0j}(t), j = 1, 2, \dots \\ \frac{dp_{00}(t)}{dt} = -\lambda p_{00}(t) \end{cases} \tag{8}$$

The solution of this system of equations satisfying the initial condition  $p_{0j}(0) = \delta_{0j}$  is:

$$p_{0j}(t) = \frac{(\lambda t)^j}{j!} e^{-\lambda t} \tag{9}$$

The final solution of this equation satisfying the initial condition  $p_{ij}(0) = \delta_{ij}$  can be found as:

$$p_{ij}(t) = \frac{(\lambda t)^{j-i}}{(j-i)!} e^{-\lambda t}, j = i, i + 1, i + 2, \dots \tag{10}$$

Find  $\lambda$  based on  $p_{ij}(t)$ :

$$\begin{aligned}
 E(X(t)) &= \sum_{n=0}^{\infty} (j - i) \frac{(\lambda t)^{j-i}}{(j-i)!} e^{-\lambda t} \\
 &= (\lambda t) e^{-\lambda t} \sum_{n=0}^{\infty} \frac{(\lambda t)^{j-i-1}}{(j-i-1)!} \\
 &= (\lambda t) e^{-\lambda t} e^{-\lambda t} \\
 &= \lambda t
 \end{aligned} \tag{11}$$

Then  $\lambda$  is the number of times per unit time interval that a node selects location  $k$  as its destination.

A time-continuous Markov chain can be used to describe the entire transfer procedure when considering two nodes in a network. We can symbolize the state  $(u_0, v_0)$  in the Markov chain when node  $N_a$  is at position  $u_0$  and node  $N_b$  is at position  $v_0$ . After a period of time, nodes  $N_a$  and  $N_b$  are shifted to arbitrary positions. If  $N_a$  moves to  $u_k$ ,  $N_b$  moves to  $v_l$ , the current state is  $(u_k, v_l)$  and the transfer rate of this process is  $q_{ij}(N_a)$  and  $q_{ij}(N_b)$ . In specific, the Markov chain enters the absorbing state  $A(u_k, v_k)$  if  $N_a$  and  $N_b$  are moved to the same location. The state transfer process of the node is shown in Figure 1.

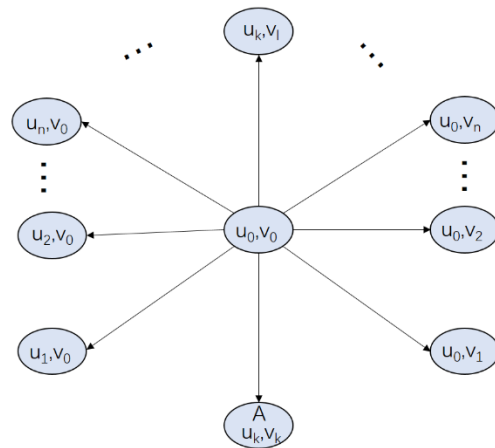


Figure 1. State Transfer Diagram.

From Equation (10), the probability that a node chooses to move to any location in time  $t$  is  $p_{ij}^{nk}(t)$ , where the probability of going to location  $k$  once is expressed as  $f_1^k$ .

$$f_1^k = \{p_{ij}^{nk} = \frac{(\lambda t)^{j-i}}{(j-i)!} e^{-\lambda t} | j = i + 1\} \tag{12}$$

The probability of traveling  $m$  times to a particular place  $k$  is written as Equation (13).

$$f_m^k = \{p_{ij}^{nk} = \frac{(\lambda t)^{j-i}}{(j-i)!} e^{-\lambda t} | j = i + m\} \tag{13}$$

Therefore, the probability of moving to location  $k$  is  $f^k = f_1^k + f_2^k + \dots + f_m^k = 1 - f_0^k$ .

It can be considered that two nodes will meet if node  $N_b$  arrives before node  $N_a$  leaves location  $k$ . To remove the exponential restriction of the traditional Markov process, matrix  $P$  can be used to record the node's choice of the next location when it is at a different location, and the elements  $p_{uk}$  in matrix  $P$  denote the probability that the next location is  $k$  when the node is at location  $u$ . The linked table  $w$  records the dwell time at each location, and  $w_k(t)$  denotes the probability that the node's dwell time at location  $k$  is greater than or equal to  $t$ .

In conclusion, the probability that a node will be active within the  $k$ th location in the upcoming period  $t$  is indicated by  $p^k$ .

$$p^k = (p_{uk} + f^k) \cdot w_k(t) / 2 \tag{14}$$

Therefore, the probability that nodes  $N_a$  and  $N_b$  meet is:

$$fc(N_a, N_b) = p_{N_a}^1 \cdot p_{N_b}^1 + p_{N_a}^2 \cdot p_{N_b}^2 + \dots + p_{N_a}^m \cdot p_{N_b}^m = \sum_{i=1}^m p_{N_a}^i \cdot p_{N_b}^i \tag{15}$$

### 3.2. Node Centering Degree

Nodes move continuously and make contact with other nodes in the network as a result. The destination node is thought to be more likely to be encountered by those who have more contact with other nodes, whereas those who have more contact with the destination node are thought to be closer to it because they meet more frequently. When the current node and the destination node of the message are in the same group, the message only needs to be forwarded within the group because the nodes in the same group are in contact more frequently. This not only gets the message to the destination quickly but also saves a lot of space. In summary, we express the centrality degree  $DC$  of a node as the



following Equation (16), which indicates the size of the node’s ability to deliver messages in this community.

$$DC = \partial \left( \sum_{i=0}^n CN_{current}^i / CN_{total} \right) + \beta \left( \sum_{j=0}^n CT_{current}^j / \sum_{k=0}^{groupsize} \sum_{j=0}^n CT_k^j \right) \quad (16)$$

where,  $\sum_{i=0}^n CN_{current}^i$  is the total number of nodes in this group that the current node has contacted, if the current node has contacted with the  $i$ th node  $CN_{current}^i = 1$ , and  $CN_{total}$  is the total number of nodes in the network.  $\sum_{j=0}^n CT_{current}^j$  is the total number of contacts between the current node and other nodes in in this group,  $CT_{current}^j$  is the number of contacts between the current node and the  $j$ th node,  $\sum_{k=0}^{groupsize} \sum_{j=0}^n CT_k^j$  is the total number of contacts between the nodes and other nodes in this group and  $groupsize$  is the number of nodes contained in this group.  $\partial$  and  $\beta$  are the control coefficients, which are  $1/2$ , respectively.

### 3.3. Historical Information Exchange

Each node must be aware of the historical movement trajectory of other nodes in order to predict its own movement path. The historical movement trajectory of a node is defined as a quintet  $(nodeID, P, W, T_{cur}, Location_{cur})$ , where  $P$  is the transfer probability matrix of the node,  $W$  is a chain of dwell times at each location,  $T_{cur}$  is the time to update the quintet and  $Location_{cur}$  is the current location of the node. The historical movement information spreads epidemically throughout the network. When two nodes meet, the nodes first store each other’s movement trajectory locally and update the information of the other node in the local cache if it already exists. When exchanging the history traces of other nodes stored by a node, the quintet with the latest update time replaces the old quintet based on the comparison of the  $T_{cur}$  stored by both sides. As a result of the remarkable regularity of student movement on campus, the  $P$  matrix will be sparse, demonstrating the strong predictability of student node trajectories.

### 3.4. Forwarding Strategy

In-group forwarding and out-group forwarding are the two ways that nodes forward messages. In-group forwarding is used when the source and destination nodes of the message are in the same group, and out-group forwarding is used otherwise. Two stages make up the message delivery process: first, the message is delivered to the group node, and then it is transmitted from the group node to the message destination.

#### 1. Out-group forwarding

When the sender and the recipient of the communication are not in the same group, forwarding is determined by the probability that the sender and the recipient are going to meet at the same place. If node  $N_a$  carries message  $m$  and encounters node  $N_b$ , and  $fc(N_b, D_m)$  is not less than  $fc(N_a, D_m)$ , where  $D_m$  is the destination node of message  $m$ . It shows that if the encounter probability between  $N_b$  and  $D_m$  is greater than  $N_a$ , then  $N_a$  will forward the message  $m$  to  $N_b$ . Otherwise, it will not.

#### 2. In-group forwarding

When a message’s source and destination nodes are both members of the same group, the message is only transmitted within that group and is forwarded in accordance with the node’s centrality. If  $DC(N_b)$  is not less than  $DC(N_a)$ , then  $N_a$  forwards the message  $m$  to  $N_b$ , otherwise it is not forwarded.

## 4. Routing Algorithms Based on Node Path Prediction and Cache Management

### 4.1. Utility Value of the Message

Nodes carry a large number of message copies in the cache as they transmit messages. When there is not enough room in the cache for new messages to be received, older messages with lower utility values can be discarded to make room. Messages with high diffusion and high energy consumption are classified as having low utility value based on the global dissemination of messages and the energy consumption of messages to the current node.

The utility value of the message can be calculated by the following equation.

$$U^m = 1 / \left( \sum_{i=1}^n node_i^m / node_{all} + \sum tran^m / \sum_{j=1}^{buffsize} tran^j \right) \quad (17)$$

where  $\sum_{i=1}^n node_i^m$  is the number of nodes that the message  $m$  passes through during transmission, the larger the value of  $\sum_{i=1}^n node_i^m$  means the higher the diffusion of the message,  $node_{all}$  is the total number of nodes in the network,  $tran^m$  is the number of times the current node forwards the message  $m$ , the larger the value means the greater the energy consumption of the current node,  $\sum_{j=1}^{buffsize} tran^j$  is the number of times the current node forwards all messages in the cache and  $buffsize$  is the size of the current node cache, according to the above formula, the utility value of any message at any point can be calculated.

To increase the node transmission success rate and reduce energy consumption, when a new message arrives, it is sorted according to the cache space utility value, and the messages with lower utility values are removed from the cache in priority to release the cache, and when the message  $m$  is moved out of the cache due to its lower utility value, no more messages  $m$  forwarded by other nodes are received.

### 4.2. Scheduled Cache Management Mechanism

When more time passes throughout the network's message dispersion process, there will be a lot more copies available. Nodes should swiftly tell other nodes to delete the copies of the message they receive in order to decrease cache occupation and erase unnecessary message copies.

Each node maintains a two-column collection of delivered messages with each element stored in the collection as a key-value pair. Concurrent hash mapping table storage is utilized to enable the simultaneous updating of delivered messages in many connections since a node may generate connections to multiple nodes at once. The delivered collection is constructed as {message ID1: delivery time; message ID2: delivery time; ...}. When two nodes connect and send a message, if the receiving node is the message's destination node and the information contained in the message does not already exist in the current delivered collection, the message's ID and delivery time are added to the collection, and the message that was previously stored in the cache is deleted. Figure 2 displays the set of nodes' delivered messages that are stored in a structure consisting of an array, a chain table, and a red-black tree. Each element of the array is a joint, and each writes operation locks the joint for this operation, and the data is stored by hashing the elements into the chain table or red-black tree of which node. A red-black tree develops from a message chain table when there are more elements in the chain table, while a chain table develops from a red-black tree when there are fewer elements.

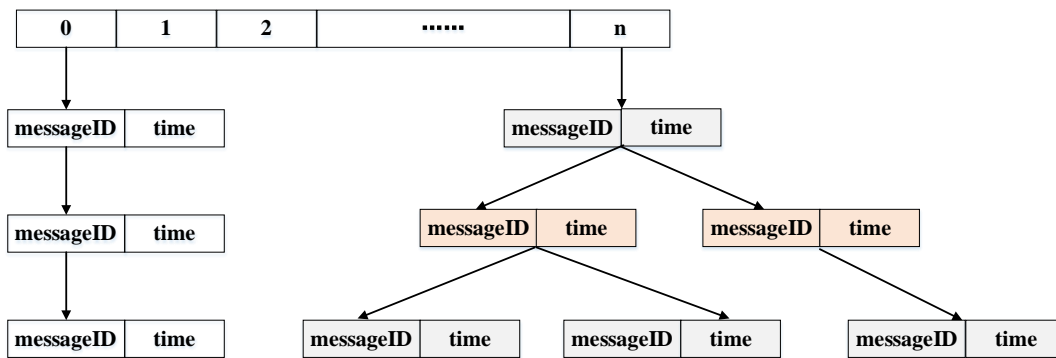


Figure 2. Delivered Message Collection.

When two nodes come together, they check their own caches to see if any messages from the other node’s delivered collection are present, and if they do, they delete this message. Then, they update the delivered messages that are not stored in the current node according to the other node’s collection, and alert other nodes in the network to delete any messages that have already reached their destinations in order to free up memory and maximize cache space.

After the experimental test, the survival time of each message is set to one and a half cycles with a better effect. This can ensure the prompt removal of useless message copies and maintenance of a certain number of useful copies to ensure the timely delivery of messages. One cycle is the average delivery time of messages, and the average delivery time is determined using Equation (18) below. Create a timer for each node’s collection, check for expired delivered messages every minute and delete them in time to release the cache. Just before the simulation is through, remove all messages from the double-column collection of delivered messages and recycle the memory in time.

$$T = \sum (AT_m - GT_m) / N_m \tag{18}$$

where  $AT_m$  is the delivery time of the message,  $GT_m$  is the message generation time and  $N_m$  is the number of messages.

#### 4.3. Markov Path Prediction and Cache Management

The probability that any node  $N_a$  will arrive at any place  $v$  at any time  $t$  can be calculated using the approach described above. A destination set  $V = \{v_1, v_2, v_3, \dots, v_n\}$  is created based on the places that students frequently visit.

To accurately model the movement characteristics of learner nodes, several representative locations on campus are selected, including dormitory, classroom, canteen, playground, supermarket and library. The node arrives at a region and stays there for a random period and then chooses the next region, assuming that transferring between regions is not time consuming. Two main parameters affect the nodes’ mobile behavior, mobile trajectory offset probability  $p_{offset}$  and dwell time  $w_{time}$ , indicating that the nodes move to the pre-set location with a probability of  $p_{offset}$  and to any location with a probability of  $1 - p_{offset}$ . To reflect the predictability of students’ movement behavior,  $p_{offset}$  is set to 0.8, where  $w_{time} \in [1h, 2h]$ .

This study suggests a Modified Markov Path Prediction and Cache Management Routing (MPCM) based on the classification of destinations mentioned above. The learner nodes are grouped to reflect the collaboration between the nodes and to minimize resource waste by forecasting the learner nodes’ trajectories and forwarding the messages. The nodes in this experiment are split into five groups, and each group chooses one of five possible destinations. Once at the destination, the nodes travel arbitrarily within the range of the destination. The pseudo code for the node forwarding process is shown in Algorithm 1.

**Algorithm 1.** MPCM strategy**INPUT:** node  $N_a$ , node  $N_b$ **OUTPUT:** Messages**START:**

1. **WHILE** (node  $N_a$  carries message  $m$  & node  $N_b$  don't carries message  $m$ )
  2. **IF** (the size of free cache space of node  $N_b \geq$  the size of  $m$ )
  3. **IF** (node  $N_b$  is  $D_m$ )
  4. call  $N_b$  receive  $m$ ;
  5. **END IF**;
  6. **IF** (node  $N_a$  has already transmitted the message  $m$  to node  $N_b$ )
  7.  $m$  cannot be transmitted
  8. **END IF**;
  9. **IF** ( $N_a$  and  $D_m$  are in the same group)
  10. **IF** (node  $N_b$  and message  $m$  are in the same group &  $DC(N_b) \geq DC(N_a)$ )
  11. call  $N_b$  receive  $m$ ;
  12. **END IF**;
  13. **ELSE**
  14. **IF** ( $fc(N_b, D_m) \geq fc(N_a, D_m)$ )
  15. call  $N_b$  receive  $m$ ;
  16. **IF** ( $N_b$  and  $D_m$  are in the same group)
  17. call  $N_b$  receive  $m$ ;
  18. **ELSE**
  19.  $m$  cannot be transmitted;
  20. **END IF**;
  21. **END IF**;
  22. **END IF**;
  23. **IF** (the free cache space of node  $N_b <$  the size of  $m$ )
  24. Calculate the utility value  $U^m$  of the message in the node  $N_b$  cache;
  25. Delete the message with the lowest utility value until the message  $m$  can be put down;
  26. **END IF**;
- END**

If the free cache space of a node is insufficient to keep the message received this time when two nodes meet in the opportunity network, the utility value  $U^m$  of the message in the cache of node  $N_b$  is calculated and the message with the lowest utility value is deleted until the message  $m$  can be put. If the free cache space of node  $N_b$  is large enough to accommodate this received message: If node  $N_a$  carries message  $m$  and node  $N_b$  does not, the message is forwarded to node  $N_b$  if it is the intended recipient. Otherwise, node  $N_a$  checks to see if it has already transmitted this message to node  $N_b$ , if it has, the message cannot be transmitted.

This message will be forwarded only within the group if nodes  $N_a$  and  $D_m$  are members of the same group. The message is forwarded if nodes  $N_b$  and  $D_m$  are in the same group and centrality of  $N_b$  is greater than  $N_a$ , otherwise, it is not forwarded. Conversely, if  $N_a$ ,  $N_b$  and  $D_m$  are not in the same group, and the probability of meeting between  $N_b$  and  $D_m$  is not less than  $N_a$ , the message is forwarded, otherwise it is not forwarded. Additionally, the message is forwarded if nodes  $N_a$  and  $D_m$  are not in the same group but nodes  $N_b$  and  $D_m$  are.

## 5. Results

### 5.1. Experimental Scheme Design

ONE (the Opportunity Network Environment) is used as the experimental environment to test the algorithm presented in this paper, and the real data set hagggle6-infocom6 [27] is used for performance verification. A total of 3 days of data transmission between 78 mobile Bluetooth devices and 20 fixed devices were collected in the Infocom06 dataset. The experimental parameters are established in Table 3 below, and the Prophet,

Epidemic, RDR [19] and FCIM [20] algorithms are evaluated alongside the algorithm suggested in this research.

**Table 3.** Simulation parameters.

Parameter	Value
dataset	haggle6-infocom6
simulation time/h	72
simulation area/m <sup>2</sup>	4500 × 3400
number of nodes	98
message generation interval/s	100
message size/kb	50 k~5000 k
message TTL/h	5

The performance of each of the five methods is evaluated under identical circumstances but with varying cache sizes, message generation intervals and message survival times. In this paper, we take into account the following four fundamental metrics: message delivery success rate, average latency, routing overhead and the number of packets dropped. A higher message delivery success rate, a lower average latency, a lower routing overhead and a lower number of packet drops signify a routing algorithm's superior performance. We consider the effect of three parameters: cache spaces, message generation intervals and time to live of messages.

Data normalization is used in the processing of the results of the experiment. The first processing is Equation (19) because a greater message delivery success rate is preferable, and second processing is Equation (20) as smaller average delay, overhead, and packet drops are preferable.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (19)$$

$$x'' = 1 - \frac{x - \min(x)}{\max(x) - \min(x)} \quad (20)$$

The algorithm's overall score is added by the normalized values under various indexes, and among them,  $\min(x)$  and  $\max(x)$  indicate the minimum and maximum values in a collection of data.

## 5.2. Experimental Results Analysis

### 5.2.1. Different Cache Spaces

This section will examine the effects of buffer size from 10 M to 100 M on delivery success rate, average latency, routing overhead and a number of packet losses because cache space size has a significant impact on network performance.

The following Figure 3 demonstrates that in contrast to other protocols, cache space size has a small impact on the delivery success rate of MPCM. Rather than sending messages blindly, MPCM makes every effort to deliver messages to nodes that may come together by predicting those nodes' locations based on their movement patterns. Hence, MPCM performs better in networks with constrained cache space. The one routing protocol most impacted by cache size is Epidemic, which employs a flooding technique to message delivery with massive message copies throughout the network. Under various cache spaces, the MPCM message latency will vary significantly, but this effect can be disregarded. Significantly fewer messages are discarded in Epidemic as cache capacity increases, the messages will be stored there for a long period and latency will rise noticeably. MPCM intra-group message forwarding and forwarding based on transmission probability may effectively regulate the number of message copies in the network, and the influence of cache size on MPCM network overhead is likewise minimal. In conclusion, the routing overhead falls dramatically, the number of messages successfully delivered by each routing algorithm rises as the cache size grows, and MPCM always maintains a low overhead level.

The amount of packet drops in MPCM is already close to zero when the cache size exceeds 60 M, and the number of drops continuously declines as more messages may be put in the Epidemic cache.

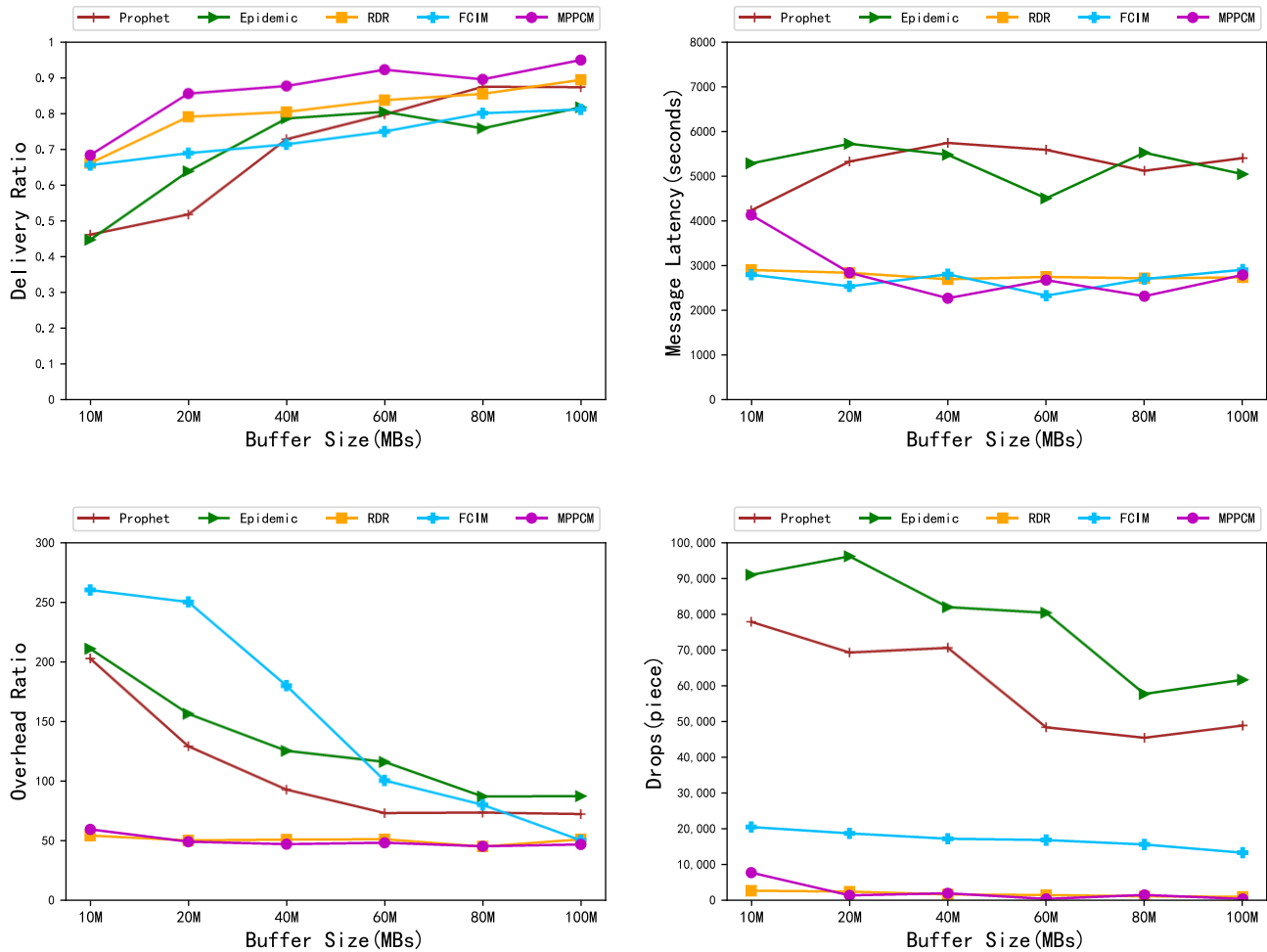


Figure 3. Comparison of success rate, latency, overhead and packet loss with different buffer size.

### 5.2.2. Different Message Generation Intervals

The impact on each metric is observed by varying the message generation interval at a cache size of 50 M. Each routing technique performs noticeably better as the generation interval gets longer.

As shown in Figure 4, when there are too many messages, MPCM moves out messages with lower utility values in accordance with the message utility values, leaving enough space for messages with high utility values, and on this basis, a certain success rate will be guaranteed. MPCM maintains a better state at intervals greater than the 40s, and the delivery success rate is no longer affected. Several other algorithms perform poorly because they are unable to handle excessively packed communications. With an increase in the message generation interval, MPCM performs fairish in terms of latency, and its overall latency is lower than that of several other three algorithms. Since the messages stay in the cache for a long time as the generation interval rises, the network overhead also rises gradually. When the generation interval exceeds the 20s, MPCM and RDR are largely unaffected, but the overhead of the other three algorithms continues to rise. The Epidemic algorithm has the highest number of dropped packets, and a large number of message copies in the network are discarded due to cache limitations and long survival times.

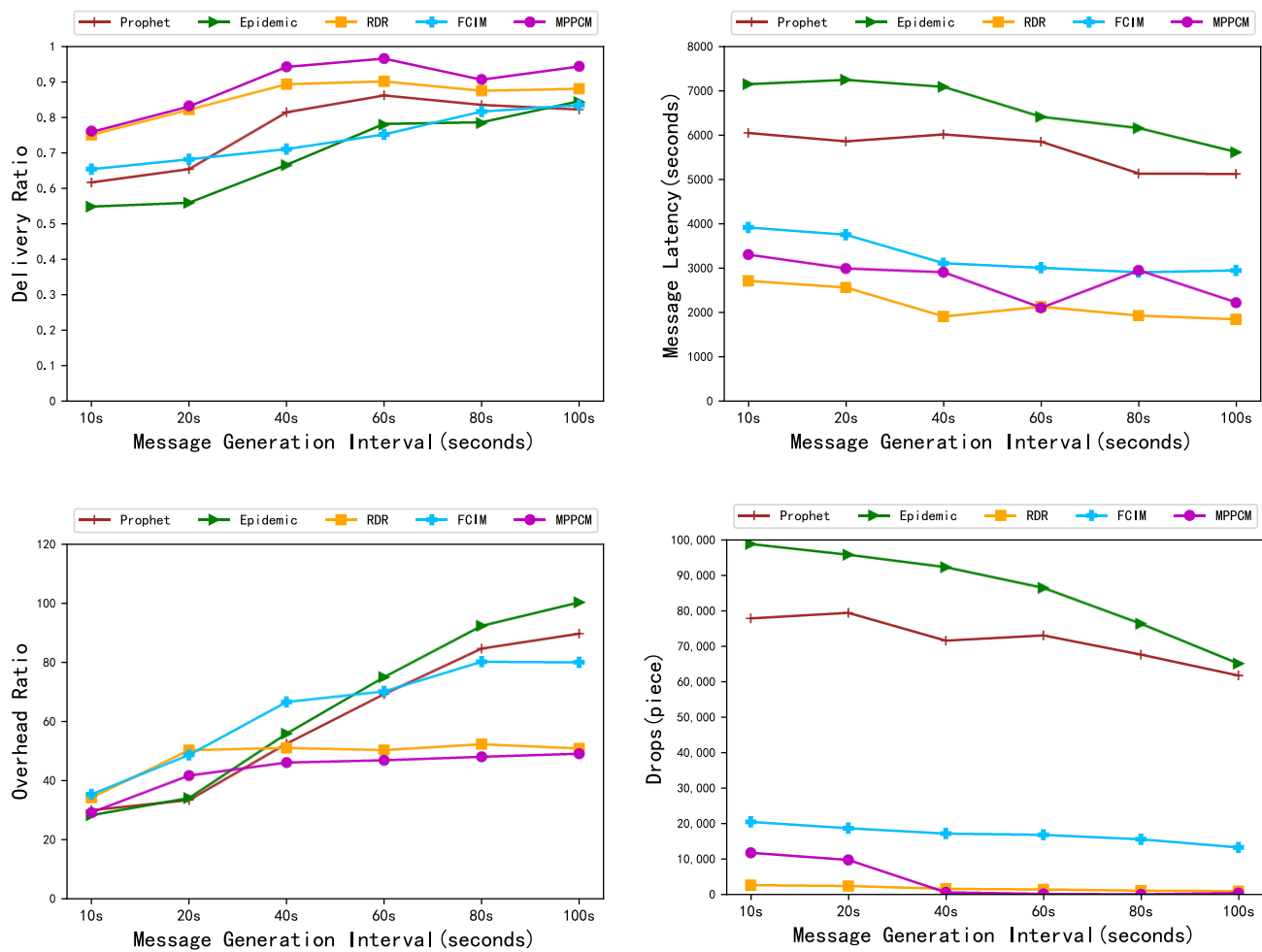


Figure 4. Comparison of success rate, latency, overhead and packet loss with different message generation intervals.

### 5.2.3. Different Time to Live of Messages (TTL)

By altering the time to live (TTL) of various messages at a cache size of 50 M and a message generation interval of 100s, the impact on the metrics is shown in Figure 5. The delivery success rate of each method has a trend of increasing and then decreasing as the message TTL increases. The message initially has adequate time to reach the destination node as the TTL increases. However, when the TTL rises further, a huge number of message copies will exist in the network, lowering the success rate. MPCM shows better performance when the TTL is greater than 3 h. The success rate of message delivery gradually declines as the TTL increases because the Epidemic algorithm causes a significant increase in copies and no more room to receive new messages. Each algorithm’s latency will rise as the TTL rises, the MPCM will not rise any further until it reaches a more stable value, and the overall latency is very low. The network overhead rises as a result of the inability to clear message copies in time as the TTL gets longer. The TTL has the biggest impact on Epidemic’s overhead, while MPCM’s overhead is consistently kept at a minimal level. When the TTL is greater than 3 h, the number of packet losses in MPCM is close to 0, and most of the messages can be delivered to the destination node before the expiration date.



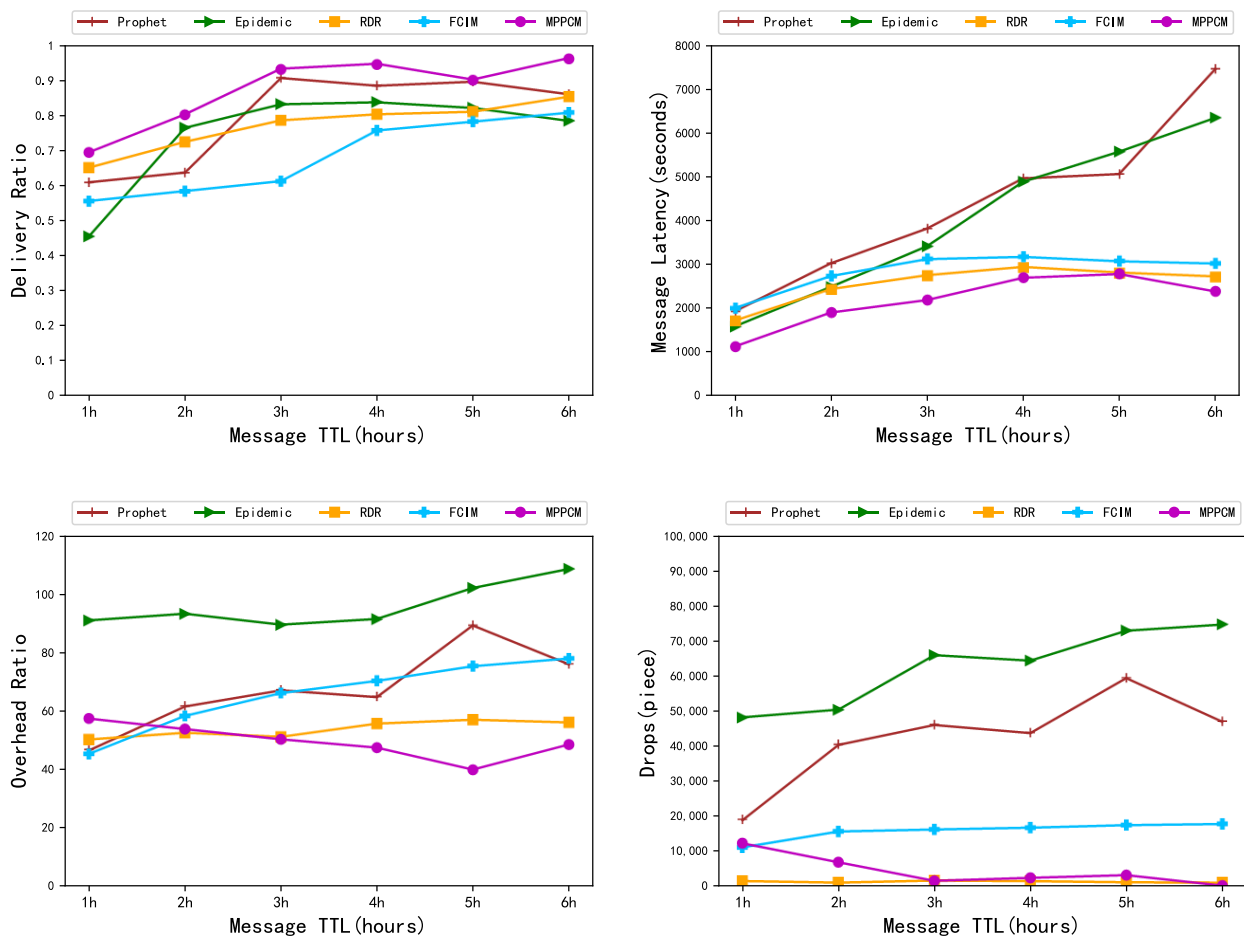


Figure 5. Comparison of success rate, latency, overhead, and packet loss with different message TTL.

The MPCM algorithm presented in this paper performs reasonably well when compared to numerous other algorithms in various cache spaces, message generation intervals and varied message TTLs. Under all circumstances, the delivery success rate of MPCM exhibits the best performance. It also performs exceptionally well in terms of latency, overhead and packet drop which is mostly unaffected by the amount of cache capacity. When messages are excessively dense, the success rate decreases, but it still performs better than many other algorithms in general. Overall, MPCM can adapt to the campus context with a limited node cache, dense message generation and short message TTL. Experiments show that the proposed method in the paper achieves better results in terms of delivery rate, average delivery latency, overhead and several packets drop. The average values of each measure are processed using data normalization, and the experimental results under the influence of three parameters, different cache spaces, different message generation intervals and different message survival times, are tabulated in Tables 4–6 below.

Table 4. Normalized scores under different caches.

Algorithm /Score	Success Rate	Overhead	Latency	Packet Drops	Total Score
Epidemic	0	0.2687	0	0	0.2687
Prophet	0.0034	0.4460	0.0092	0.2374	0.696
RDR	0.6362	0.9897	0.9639	1	3.5898
FCIM	0.1840	0	1	0.7987	1.9827
MPCM	1	1	0.9388	0.9935	3.9323

**Table 5.** Normalized scores for different message generation intervals.

Algorithm /Score	Success Rate	Overhead	Latency	Packet Drops	Total Score
Epidemic	0.3631	0	0	0	0.3631
Prophet	0	0.2107	0.2123	0.1656	0.5886
RDR	0.8048	0.776	1	1	3.5808
FCIM	0.2226	0.0354	0.7517	0.8169	1.8266
MPCM	1	1	0.8713	0.9753	3.8466

**Table 6.** Normalized scores with different TTL.

Algorithm /Score	Success Rate	Overhead	Latency	Packet Drops	Total Score
Epidemic	0.6064	0	0.1506	0	0.757
Prophet	0.34914	0.6095	0	0.3269	1.28554
RDR	0.4633	0.9084	0.8289	1	3.2006
FCIM	0	0.6547	0.6942	0.765	2.1139
MPCM	1	1	1	0.9492	3.9492

The suggested algorithm in this paper has the highest score under the influence of the three parameters, yielding the best outcome.

Our proposed approach works well in the campus context because student nodes are more regular in their movement and the message-forwarding process is not blind, which effectively limits the message copies in the network and maximizes cache space utilization. Moreover, messages are forwarded within groups based on centrality, which means that messages can be transmitted to their destinations through those relays that are more influential. However, because of their potential similarity in sparse networks, it may be impossible to identify which nodes have high centrality.

## 6. Conclusions

This research suggests a modified Markov path prediction algorithm for nodes in campus opportunity networks with specific movement patterns. Students are typically thought to travel in small groups and repeat themselves. The proposed routing strategy is more effective according to these two forwarding methods we present in the paper for nodes: in-group forwarding and out-group forwarding. We first allow the message to reach its group as quickly as possible, then it is further forwarded based on the influence of the node within the group, as nodes with higher influence have more access to the destination node of the message. Moreover, we discovered that storage capacity of nodes is constrained and typically small. As a result, we propose a cache management strategy in this paper. When the node's own cache space is insufficient, the message utility value is calculated based on the message diffusion and the energy consumption to the current node, and the messages with high utility value will be reserved first, achieving a reasonable cache allocation.

The communication between nodes in the same group will be closer as the suggested method is for nodes on campus, and we hope that the following work will result in greater cooperation between nodes in the same group.

**Author Contributions:** Conceptualization, Y.C. (Yumei Cao) and P.L.; methodology, Y.C. (Yumei Cao); software, Y.C. (Yumei Cao); validation, Y.C. (Yumei Cao); formal analysis, P.L.; investigation, T.L.; resources X.W. (Xiaojun Wu); data curation, X.W. (Xiaoming Wang); writing—original draft preparation, Y.C. (Yumei Cao); writing—review and editing, Y.C. (Yumei Cao); visualization, Y.C. (Yumei Cao); supervision, Y.C. (Yuanru Cui); project administration, P.L.; funding acquisition, P.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is partly supported by the National Key R and D Program of China under grant No. 2020YFC1523305, Key Laboratory Funds of the Ministry of Culture and Tourism under grant No 2022-13, the National Natural Science Foundation of China under Grant No. 61877037, 61872228, 61977044, the Shaanxi Key Science and Technology Innovation Team Project under Grant No. 2022TD-26.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** We appreciate the anonymous reviewers' and editorial team members' suggestions and comments. Thanks for the support of the fund projects mentioned above.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Sachdeva, R.; Dev, A. Review of opportunistic network: Assessing past, present, and future. *Int. J. Commun. Syst.* **2021**, *34*, e4860. [[CrossRef](#)]
2. LI, P.; Wang, X.M.; Zhang, L.C.; LU, J.L.; Zhu, T.J.; Zhang, D. A Novel Method of Video Data Fragmentary and Progressive Transmission in Opportunistic Network. *Acta Electronica Sin.* **2018**, *46*, 2165.
3. Bagirathan, K.; Palanisamy, A. Opportunistic routing protocol based EPO-BES in MANET for optimal path selection. *Wirel. Pers. Commun.* **2022**, *123*, 473–494. [[CrossRef](#)]
4. Gautam, T.; Dev, A. Improving Packet Queues Using Selective Epidemic Routing Protocol in Opportunistic Networks (SERPO) BT—Advances in Computing and Data Sciences. In *Advances in Computing and Data Sciences, 4th International Conference, ICACDS 2020, Valletta, Malta, 24–25 April 2020; Revised Selected Papers 4*; Springer: Singapore, 2020; pp. 382–394.
5. Bansal, A.; Gupta, A.; Sharma, D.K.; Gambhir, V. Icar-inheritance inspired context aware routing protocol for opportunistic networks. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 2235–2253. [[CrossRef](#)]
6. Sharma, D.K.; Kukreja, D.; Chugh, S.; Kumaram, S. Supernode routing: A grid-based message passing scheme for sparse opportunistic networks. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 1307–1324. [[CrossRef](#)]
7. Singh, J.; Obaidat, M.S.; Dhurandher, S.K. Location based Routing in Opportunistic Networks using Cascade Learning. In *Proceedings of the 2021 International Conference on Computer, Information and Telecommunication Systems (CITS), Istanbul, Turkey, 29–31 July 2021*; pp. 1–5.
8. Dhurandher, S.K.; Singh, J.; Nicopolitidis, P.; Kumar, R.; Gupta, G. A blockchain-based secure routing protocol for opportunistic networks. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 2191–2203. [[CrossRef](#)]
9. Sharma, D.K.; Rodrigues, J.J.P.C.; Vashishth, V.; Khanna, A.; Chhabra, A. RLProph: A dynamic programming based reinforcement learning approach for optimal routing in opportunistic IoT networks. *Wirel. Netw.* **2020**, *26*, 4319–4338. [[CrossRef](#)]
10. Kumar, P.; Chauhan, N.; Chand, N. Node activity based routing in opportunistic networks. In *Proceedings of the International Conference on Futuristic Trends in Network and Communication Technologies, Taganrog, Russia, 14–16 October 2019*; pp. 265–277.
11. Gou, F.; Wu, J. Triad link prediction method based on the evolutionary analysis with IoT in opportunistic social networks. *Comput. Commun.* **2022**, *181*, 143–155. [[CrossRef](#)]
12. Chunyue, Z.; Hui, T.; Yaocong, D. An Energy-Saving Routing Algorithm for Opportunity Networks Based on Sleeping Mode. In *Proceedings of the 2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Gold Coast, Australia, 5–7 December 2019*; pp. 13–18.
13. Derakhshanfard, N.; Soltani, R. Opportunistic routing in wireless networks using bitmap-based weighted tree. *Comput. Netw.* **2021**, *188*, 107892. [[CrossRef](#)]
14. Chithaluru, P.; Tiwari, R.; Kumar, K. AREOR—Adaptive ranking based energy efficient opportunistic routing scheme in Wireless Sensor Network. *Comput. Netw.* **2019**, *162*, 106863. [[CrossRef](#)]
15. Hernández-Orallo, E.; Borrego, C.; Manzoni, P.; Marquez-Barja, J.M.; Cano, J.C.; Calafate, C.T. Optimising data diffusion while reducing local resources consumption in Opportunistic Mobile Crowdsensing. *Pervasive Mob. Comput.* **2020**, *67*, 101201. [[CrossRef](#)]
16. Raverta, F.D.; Fraire, J.A.; Madoery, P.G.; Demasi, R.A.; Finochietto, J.M.; D'argenio, P.R. Routing in Delay-Tolerant Networks under uncertain contact plans. *Ad. Hoc. Netw.* **2021**, *123*, 102663. [[CrossRef](#)]
17. Das, P.; Nishantkar, P.; De, T. SECA on MIA-DTN: Tackling the Energy Issue in Monitor Incorporated Adaptive Delay Tolerant Network Using a Simplistic Energy Conscious Approach. *J. Netw. Syst. Manag.* **2019**, *27*, 121–148. [[CrossRef](#)]
18. Kang, M.W.; Chung, Y.W. An improved hybrid routing protocol combining MANET and DTN. *Electronics* **2020**, *9*, 439. [[CrossRef](#)]
19. Pirzadi, S.; Pourmina, M.A.; Safavi-Hemami, S.M. A novel routing method in hybrid DTN-MANET networks in the critical situations. *Computing* **2022**, *104*, 2137–2156. [[CrossRef](#)]
20. Mao, Y.; Zhou, C.; Qi, J.; Zhu, X. A fair credit-based incentive mechanism for routing in DTN-based sensor network with nodes' selfishness. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 1–18. [[CrossRef](#)]

21. Vahdat, A.; Becker, D. Epidemic Routing for Partially-Connected Ad Hoc Networks. In *Handbook of Systemic Autoimmune Diseases*; Elsevier: Amsterdam, The Netherlands, 2000.
22. Lindgren, A.; Doria, A.; Schelén, O. Probabilistic Routing in Intermittently Connected Networks. *ACM Sigmobile Mob. Comput. Commun. Rev.* **2003**, *7*, 19–20. [[CrossRef](#)]
23. Rehman, G.U.; Haq, M.I.U.; Zubair, M.; Mahmood, Z.; Singh, M.; Singh, D. Misbehavior of nodes in IoT based vehicular delay tolerant networks VDTNs. *Multimed. Tools Appl.* **2023**, *82*, 7841–7859. [[CrossRef](#)]
24. Rehman, G.U.; Ghani, A.; Zubair, M.; Naqvi, S.H.A.; Singh, D.; Muhammad, S. IPS: Incentive and Punishment Scheme for Omitting Selfishness in the Internet of Vehicles (Iov). *IEEE Access* **2019**, *7*, 109026–109037. [[CrossRef](#)]
25. Rehman, G.U.; Zubair, M.; Qasim, I.; Badshah, A.; Mahmood, Z.; Aslam, M.; Jilani, S.F. EMS: Efficient Monitoring System to Detect Non-Cooperative Nodes in IoT-Based Vehicular Delay Tolerant Networks (VDTNs). *Sensors* **2023**, *23*, 99. [[CrossRef](#)] [[PubMed](#)]
26. Rehman, G.U.; Ghani, A.; Zubair, M.; Saeed, M.I.; Singh, D. SOS: Socially omitting selfishness in IoT for smart and connected communities. *Int. J. Commun. Syst.* **2023**, *36*, e4455. [[CrossRef](#)]
27. Scott, J.; Hui, P.; Crowcroft, J.; Diot, C. Huggle: A networking architecture designed around mobile users. In Proceedings of the Third IFIP Wireless on Demand Network Systems Conference, Les Menuires, France, 18–20 January 2006.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.