# Performance-optimized Opportunistic Routing with Temporal-spatial Awareness and Collaborative Caching

Guangxu Wang, Peng Li*, Fei Hao, Shu Zhao, Xuanwen Hao, and Huan Jia

*Abstract*—**Opportunistic networks enable communication in infrastructure-deficient scenarios such as IoT deployment and disaster rescue through node mobility. However, existing routing protocols relying solely on contact frequency neglect link duration, leading to unstable transmission and inefficient caching. This paper proposes PORTAC, a historical contact–enhanced collaborative cache routing mechanism integrating contact stability–aware routing and multi-level cache scheduling. A spatio-temporal Contact Stability Index (CSI) evaluates link quality via duration-based logarithmic gain and historical context, while a core–radiation–edge cache hierarchy manages messages by value, forwarding count, and lifetime. Simulations on the ONE platform show that PORTAC increases delivery ratio by over 10.6% compared with PropheRouter, enhancing efficiency in dynamic networks.**

*Index Terms*—**opportunistic network, contact stability index, collaborative cache management, spatio-temporal coupling model.**

## I. Introduction

Communication networks have undergone significant and profound changes in today's information age. The currently widely used telecommunications networks rely on many base stations to achieve communication. This network requires each user's terminal to establish an end-to-end connection with nearby base stations. The current communication principle of telecommunications networks requires establishing a large amount of infrastructure, thus increasing the cost of communication. The emergence of opportunistic networks has overcome this situation to a certain extent. Each user who uses opportunistic networks to communicate has two identities: terminal and base station. From the perspective of a terminal, each user can send and receive information; from the perspective of a base station, each user is a communication intermediary for other users, helping neighboring users forward messages. Opportunistic networks save the cost of establishing infrastructure, such as base stations, during communication, significantly reducing communication costs [1].

As an ad hoc network [2] that relies on node movement to achieve intermittent communication, the working principle of the opportunistic network determines its main application scenarios. With unique advantages in vehicle interconnected networks, drone emergency communications, and disaster rescue scenarios, Jianming Cui and others applied opportunistic networks to vehicle interconnected networks to improve communication during rapid vehicle movement [3]. Niebla-Montero and others applied opportunity networks to controlled and industrial scenarios and found that they performed well in environments with minimal signal blocking and short distances. [4]. Puneet Garg and others applied opportunistic networks to construct smart cities to provide seamless communication in areas with poor network coverage [5].

At the beginning of the 21st century, opportunistic networks have developed into a relatively perfect research system, and many routing algorithms have developed rapidly. However, there is still an imbalance between limited resources and complex service requirements in highly dynamic environments. Therefore, this paper puts forward the following innovative solutions in the field of opportunistic networks:

### 1) *Quantitative Contact Stability Model (CSI)*

A CSI model based on spatiotemporal coupling is proposed to solve the problem of existing routing strategies ignoring the link stability caused by node contact duration. The model considers node contact duration and environmental historical state index in the time dimension, comprehensively evaluates the current node state, historical node state, and neighbor node state, and quantifies the consideration indicators in the time dimension. Calculate the activity factor and congestion risk control coefficient of nodes in the spatial dimension to complete the quantification task of the spatial dimension. Finally, a multi-dimensional evaluation of link status is realized.

### 2) *Tiered Cache Management Framework (TCA)*

This paper proposes a multi-level cache dynamic scheduling mechanism. Aiming at the problem that existing cache management strategies lack fine-grained message priority mechanisms, it stores hot messages with high access frequency

Manuscript received November 10, 2025; revised March 6, 2026. Date of publication August 26, 2026. Date of current version August 26, 2026.

G. Wang, P. Li, F. Hao and H. Jia are with the School of Artificial Intelligence and Computer Science, Shaanxi Normal University, Xi’an 710119, China (e-mails: {wgx2827, lipeng, fhao, jiahuan}@snnu.edu.cn).

S. Zhao is with the Faculty of Education, Shaanxi Normal University, Xi’an 710119, China (e-mail: zhaoshu@snnu.edu.cn).

X. Hao is with the Shaanxi Normal University Library, Xi’an 710119, China (e-mail: haoxuan@snnu.edu.cn).

Digital Object Identifier (DOI): 10.24138/jcomss-2025-0219

* Corresponding author

in the core buffer, pending messages with small forwarding volume in the radiation buffer, and messages with large volume, value, and low density in the edge buffer.

This paper makes the following contributions to the development of opportunistic network theory:

- This paper proposes a PORTAC routing strategy, which combines temporal dimension stability perception and spatial dimension risk control to quantify node contact stability. It also establishes a spatiotemporal coupling model, breaking the single consideration of time or space in existing routing strategies.
- Using a multi-level cache management strategy, the value of messages is dynamically calculated and sorted internally at all levels, and policies are formulated between all levels for message flow, which refines the priority level of messages in the cache and provides theoretical exploration for multi-level and fine-grained management of cache strategies.

The remaining portion of the paper is structured as follows: an introduction to the problem and proposed PORTAC framework, a related work section reviews prior research. The methodology is then detailed in two parts: the contact stability model for routing and a tiered cache architecture. The performance evaluation section compares PORTAC against benchmarks via simulation, and the paper concludes with a summary of the results.

## II. Related Work

In recent years, research on routing strategies in opportunistic networks has gradually developed from one dimension to multiple dimensions. With the popularization of deep learning technology, research has begun integrating spatio-temporal characteristics such as node movement trajectory and link stability. At the same time, cache management strategies have started to consider multiple optimization goals such as message value and node resources, and hierarchical architecture and collaborative replacement caching mechanisms have also begun to be studied.

### *A. Study on Contact Stability based on Spatio-temporal Characteristics*

In recent years, opportunistic network routing strategies have gradually shifted from single contact frequency prediction to multi-dimensional stability assessment. Huang, F et al. [6] proposed an improved spray and wait (STCESW) routing algorithm based on node stable transmission capacity evaluation. Integrate network models that consider node connection stability and transmission capacity. Fu, YH et al. [7] proposed an opportunistic network routing algorithm based on connection stability and energy recovery (ST-ProPhet), which developed a node encounter probability prediction model that integrates the average connection duration. Jun Ren et al. [8] proposed a method for detecting link transmission stability in opportunistic networks based on deep learning. Use deep learning algorithms to extract network-level features, construct classification prediction models, and complete link transmission stability detection.

Recently, progress has been made in the research of mixed spatio-temporal characteristics. Xiao Han et al. [9] proposed a multi-attribute routing method based on the Pareto optimal solution (MR-Pareto) to balance node energy consumption and resource constraints in the transmission protocol. This paper analyzes attribute relationships between nodes, and a non-dominant ranking method is implemented to achieve a set of joint optimization candidate nodes. Jian Shu et al. [10] proposed an opportunistic network link prediction algorithm based on three-dimensional convolutional neural networks (3D-CNN). This paper combines network attributes in different dimensions, such as time, space, and correlation between nodes, to represent an opportunistic network. Anna Maria Vegni et al. [11] proposed a MemOy-based vehicle social forwarding method, which builds its packet forwarding logic by considering the nodes' past and current "social" patterns. This mechanism integrates social components and physical characteristics into its forwarding mechanism.

### *B. Research on Multi-dimensional Collaborative Cache Management*

Multi-dimensional cache optimization has become key in breaking through a single elimination strategy. Yumei Cao et al. [12] defined a message's utility value to describe the message's spread degree and the energy consumption of the current node. Then they managed the cache according to the utility value. By creating a concurrent hash map of transmitted messages, the system informs the remaining nodes to delete the transmitted messages. It frees up cache space in time after the message is transmitted to its target. Sheng Yin et al. [13] proposed a cache management strategy that dynamically adjusts based on the cache value of messages, which comprehensively considers the urgency and content importance of messages. When a node's cache space is insufficient, a greedy algorithm similar to a 0-1 knapsack problem selects the messages to be retained. The cache management strategy proposed by Shupei Chen et al. [14] dynamically calculates the remaining cache ratio of nodes, combines a fuzzy inference system to evaluate cache and energy states, prioritizes high-resource nodes as relays, and uses a two-hop feedback mechanism to expand the forwarding range. At the same time, the minimum cache threshold eliminates inefficient nodes and achieves adaptive congestion control and efficient data transmission.

Layered caching architectures has begun to attract attention. Dapeng Wu et al. [15] evaluated users 'content sharing capabilities based on their social relationships, divided interest communities, and dynamically predicted content popularity. They divided user caches into four targeted areas (local-local, local-other, etc.) and dynamically combined them with popularity. They replaced low-value content to achieve efficient collaborative caching.

Reviewing and analyzing the literature found that research on routing strategies and collaborative cache management in opportunistic networks has made significant progress in spatio-temporal feature modeling and multi-dimensional resource optimization. Existing routing algorithms break through the limitations of traditional single indicators and achieve joint optimization of node stability and energy consumption; existing cache management strategies significantly improve cache utilization and system energy efficiency through methods such as dynamic utility evaluation, fuzzy inference resource allocation, and hierarchical interest community architecture.

However, there are still key shortcomings in terms of dynamic environment adaptability and cross-layer collaboration mechanisms: (1) existing routes are separated in time and space during the decision-making process and are difficult to follow the dynamic network topology;(2) The message priority calculation method in node cache management strategies is single and lacks multiple levels of density value perception. To this end, this paper proposes a spatio-temporal coupled CSI model to solve the former. It breaks through the latter through a multi-level buffer dynamic scheduling mechanism and a collaborative elimination strategy.

## III. Building PORTAC Routing

PORTAC establishes a model for the temporal dimension and spatial dimension information of nodes, proposes a formula to consider both temporal and spatial aspects comprehensively, and realizes routing decisions based on this theoretical basis during the operation of the opportunistic network.

### *A. Overall Architecture Diagram*

PORTAC routing involves two parts: routing decision and buffer management. In the routing decision stage, node contact stability is calculated, and specific routing decisions are realized based on the calculation results, as shown in Fig 1; in the buffer management stage, inspired by the three-level cache management of the computer CPU, a three-stage cache management strategy is innovatively proposed. Fig. 2 shows the three-level cache architecture.

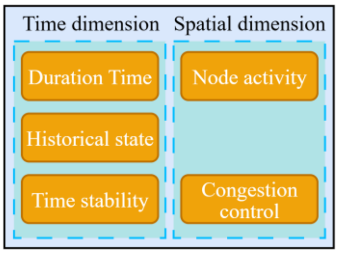


Fig. 1. Route Management-CSI Calculation

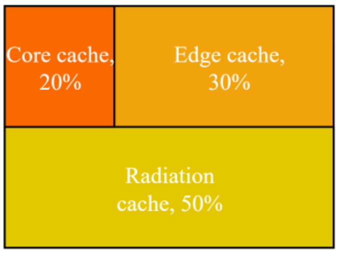


Fig. 2. Cache Management-Level 3 Cache

### *B. Contact Stability Model (CSI)*

The opportunistic network features prominent features such as node mobility and indirectness of links. The contact stability model is established based on indicators such as node contact frequency, duration, and recent duration.

All calculation indicators involved in the formula in this chapter are node pair specific indicators between observation node $i$ and encountered neighbor node $j$, rather than the global single value of observation node $i$; all indicators are based on historical contact data of nodes $i$ and $j$, and locally maintained neighbor status information, and updates are triggered only when nodes $i$ and $j$ encounter each other.

B.1 Time Dimension: Stability Perception Modeling

1) Duration gain function [16]. In actual social network contact, there are two extremes: instantaneous and long-term connections. Transient links transmit less data and contribute less to data transmission in the entire opportunistic network. Long-term links have relatively weak flexibility. To avoid frequent interference from transient links and suppress the tailing effect of long-term links, equation (1) is proposed when calculating the duration gain:

$$G_{\mathrm{d}}(i,j) = \sum \ln\left(1+D_{ij}\right) \tag{1}$$

where $G_{\mathrm{d}}(i,j)$ represents the sum of the duration benefits of node $i$ and node $j$ within the time window and $D_{ij}$ represents the single duration within the time window. Compared to linear or power-law functions, the logarithmic function demonstrates superior adaptability to the connectivity characteristics of opportunistic networks as a duration gain function. A linear gain function, $G_{\mathrm{d}}(i,j) = D_{ij}$, assigns disproportionately higher weight to overly long connections, which may cause the algorithm to over-rely on a few persistent links while neglecting numerous short-lived ones. A power-law function, $G_{\mathrm{d}}(i,j) = {D_{ij}}^{k}$ (where $0 < k < 1$), can compress the contribution of long connections; however, it exhibits lower discriminative power in the short-duration range. In contrast, the logarithmic function, as formulated in Equation (1), possesses a steeper slope when the duration Dur is small. This property makes it highly sensitive in distinguishing between transient contacts and valuable short-duration connections. As Dur increases, the growth of its gain decelerates rapidly, thereby effectively suppressing the excessive influence of overly long connections. Consequently, it achieves a balance between "sensitivity to short-lived contacts" and "suppression of overly long connections."

2) Historical environmental status index. As nodes contact each other, the state information of neighboring nodes changes rapidly. When calculating stability, the environmental conditions of the current node in a historical unit time are evaluated. The historical ecological state index is calculated by formula (2):

$$D_r(i,j) = \beta \cdot \frac{\Delta N}{\Delta t} \cdot e^{-R/\Delta t} \tag{2}$$

where $D_r(i,j)$ represents the historical environmental state index of node $j$, $\beta$ represents the network type coefficient, $\Delta t$ represents the length of the time window, $\Delta N$ represents the change of neighbor nodes in a time window, and $R$(recency) represents the most recent contact interval. An increase in the node change rate $\frac{\Delta N}{\Delta t}$ per unit time means more neighbor nodes are meeting the node. The smaller the closest contact time $R$, the larger $e^{-R}$, which means that the node was active for a short period.

3) Comprehensive indicator of time stability. Formula (3) is the comprehensive index of time stability, which is obtained by weighting the duration function and the historical environmental state index:

$$T_{\text{stable}}(i,j) = \gamma \cdot \frac{G_{\text{d}}(i,j)}{\sqrt{F_{ij}}} + (1-\gamma) \cdot D_{\text{r}}(i,j) \quad (3)$$

where $T_{\text{stable}}(i,j)$ denotes temporal comprehensiveness, $\gamma$ denotes weight parameter, $G_{\text{d}}(i,j)$ denotes duration gain, $F_{ij}$ represents the number of contacts within the time window. A complete process of "link establishment-data interaction-link disconnection" is completed between nodes, which is counted as 1 effective encounter. $D_r(i,j)$ denotes historical environmental state index. This formula calculates $F_{ij}$ to normalize the duration function to prevent overfitting of high-frequency nodes and enhance the model's universality to various nodes. When $\gamma$ increases, the weight of $G_d(i,j)$ increases, which means that the system pays more attention to the short-term state changes of nodes; when $\gamma$ decreases, the situation is just the opposite, and the system relies more on historical indicators to smooth short-term changes.

$$\gamma = \frac{\sigma_{\text{recent}}(j)}{\sigma_{\text{recent}}(j) + \sigma_{\text{history}}(j)} \quad (4)$$

Shown in formula (4). To cope with sudden fluctuations, a dynamic adaptive strategy is used to change the weight parameter $\gamma$, $\sigma_{\text{recent}}(j)$ is the standard deviation of the recent connection duration $D_{ij}$ in the time window, reflecting the current fluctuation of the contact duration, and $\sigma_{\text{history}}(j)$ is the standard deviation of the node's historical connection duration $D_{ij}$, reflecting the baseline stability of the contact duration. When $\sigma_{\text{recent}}$ increases, $\gamma$ increases, and when calculating $T_{\text{stable}}(i,j)$, the proportion of the current change increases.

B.2 Spatial Dimension: Risk Awareness Modeling

1) Node activity factor. Calculating the node activity factor considers the need to use subnets of different sizes and reserves the relative magnitude for routing decisions. For this reason, the node activity factor calculation formula (5) is proposed.

$$A_{\text{f}}(i,j) = \frac{F_j}{max(F_{\text{network}})} \quad (5)$$

where $A_{\text{f}}(i,j)$ represents the activity factor, $F_j$ represents the current node's contact frequency, and $F_{\text{network}}$ represents the frequency's maximum value in the global network. Due to isolated information islands in the opportunistic network, nodes can only obtain neighboring data through indirect contact, and nodes cannot know the maximum contact frequency in the whole area. For this reason, each node is allowed to maintain an $\widehat{M}$ value estimate $max(F_{\text{network}})$. Initially, the $\widehat{M}$ of each node is $F_{ij}$. When node a and node b meet, $max(\widehat{M}_a, \widehat{M}_b)$ is updated to the current $\widehat{M}$ value of both parties.

2) Congestion risk control factor. Due to physical constraints, communication resources in opportunistic networks are limited. When measuring a network's stability, the congestion level under the current physical constraints must be considered—proposed congestion risk control formula (6).

$$R_{\text{c}}(i,j) = 1 - e^{-\alpha(A_{\text{f}}(i,j)-\theta)} \quad (6)$$

where $R_{\text{c}}(i,j)$ represents the congestion risk control coefficient, $\alpha$ represents the sensitivity coefficient, the activity factor of node $A_{\text{f}}(i,j)$, and $\theta$ represents the threshold for starting the congestion control calculation. When the node activity level is lower than $\theta$, the congestion risk control coefficient approaches 0, and no interference is caused to the original link stability evaluation result; when the node activity level far exceeds $\theta$, the control coefficient quickly converges to 1, and the link is strongly suppressed. The scoring is strongly suppressed to prevent highly congested nodes from being selected as relays.

B.3 Global Synthesis Model

Stability integrates temporal and spatial dimension models. It draws on joint PSI (Population Stability Index) and CSI monitoring to achieve "high stability × low risk" joint optimization. Formula (7) gives the CSI calculation process.

$$CSI_{ij} = T_{\text{stable}}(i,j) \cdot (1 - R_{\text{c}}(j)) \quad (7)$$

*C. Implementation of Adaptive Routing Algorithm*

In studying the dynamic changes of the network environment, this study emphasizes that each network node must synchronize the relevant data of the contact stability model described in Section III.B in real time and accurately calculate their respective CSI values to ensure that accurate and reliable information can be provided when responding to query requests from other nodes.

Algorithm 1: Find next-hop routing process

```
when node a encounters node B{b1, b2,..., bn}:
  bestb = b1
  for each bi in B:
    if bi is the message destination:
      transfer_messages(a, bi)
      send_ack(bi)
    csi = compute_CSI(a, bi)
    if csi > current_best:
      bestb = bi
      current_best = csi

  establish_connection(bestb)
  exchange_message_indexes(a, bestb)
  send_messages_for(bestb)
  for each buffer in [core, radiation, edge]:
    while messages in the buffer:
      transfer _messages(a, bestb)

  remove_transmitted_nodes(bestb)
  recalculate_CSI_neighbors(a)
  return
```

Algorithm 1 is the next-hop routing process. When implementing the next-hop node selection mechanism, the node must first traversal and identify neighboring nodes. The node should determine whether there is a target node that matches the destination address of the message it carries. If a matching target node is found, a connection is established with it, and after the message transmission is completed, an acknowledgement message (ACK) is generated and stored in the core buffer; if no matching node is found, CSI contact stability calculation needs to be performed on neighboring nodes. On this basis, nodes should select the node with the highest CSI value as the next hop node to establish a connection.

After establishing the connection, both nodes should exchange the message index and Ack list in their respective buffers. Based on the received counterpart index information, a node should send a message to the counterpart it does not yet own. Messages whose destination address points to the other node should be sent first when sending a message. Subsequently, the node should perform message sending operations according to the priority of the core buffer, the radiation buffer, and the edge buffer.

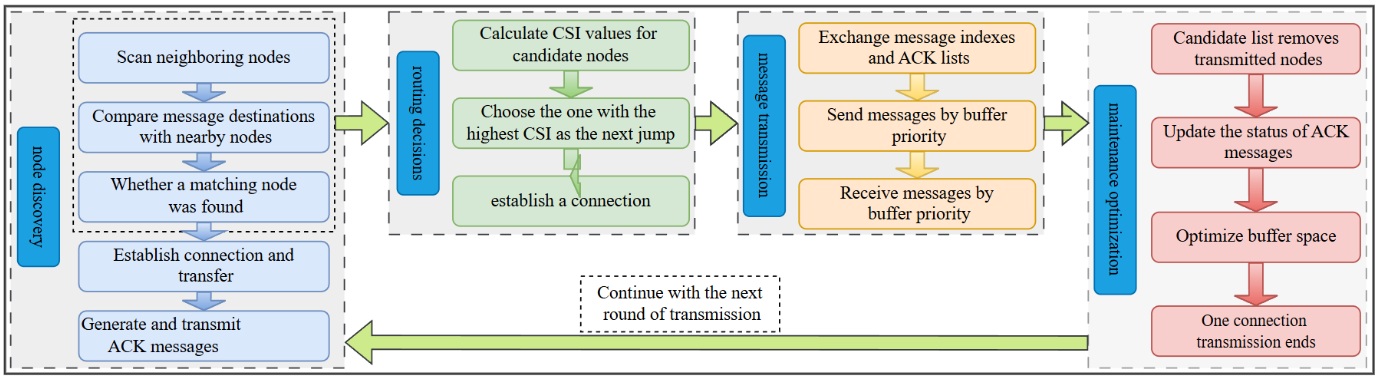


Fig. 3. PORTAC Routing Operation Process.

During the process of receiving a message, once the current connection completes message transmission, the node needs to remove the transmitted node from the candidate list. Afterward, the node should re-perform CSI calculations on the neighboring nodes to determine the node with the highest CSI value in the next round of links. Fig. 3. shows the running process of PORTAC routing.

Routing algorithm complexity analysis: When a single node meets, $CSI_{ij}$ is calculated as a fixed number of arithmetic operations, with a time complexity of O(1); when a node encounters n neighbors, the time complexity of traversing and selecting the route is O(n); Nodes only need to maintain historical data of neighbor nodes, and the spatial complexity is O(n)(n is the number of neighbors, which is much less than the total number of nodes in the entire network N).

## IV. Building Tiered Cache Architecture

Based on the three-level cache architecture in computer systems [17, 18], this paper proposes a multi-level cache dynamic scheduling mechanism suitable for opportunistic networks. The cache within a node is divided into three parts: core cache, radiation cache, and edge cache.

### *A. Dynamic Scheduling Mechanism for Multi-level Cache*

The three-tier cache architecture allocates 20%, 50%, and 30% of a node's total cache capacity to the Core, Radiation, and Edge caches, respectively. This heuristic ratio is established based on observations and analysis of typical message workloads in opportunistic networks:

1) The Core cache is designed to store a small number of critical messages with extremely high access frequency or urgent time-to-live (TTL). Its relatively small capacity (20%) facilitates the implementation of strict eviction policies, ensuring the rapid forwarding of the highest-priority messages.

2) The Radiation cache serves as the primary working area of the system, accommodating the majority of messages that are in an active forwarding state with medium priority. Its largest share (50%) provides ample buffer space for message sorting and inter-tier flow.

3) The Edge cache is intended for messages that are larger in size, have a relatively low value density, or exist in numerous copies. Its moderate capacity (30%) helps to conserve resources while extending the potential transmission window for these messages.

This ratio serves as the default system configuration and can be uniformly adjusted based on the overall resource profile of the network in practical deployments.

### A.1 Edge Cache

The edge buffer calculates the residual value of the message based on the number of times it has been forwarded and its size. It then sorts the messages from small to large and discards the messages at the end. When the edge buffer is complete, the residual value of the message is calculated using formula (8):

$$V_{\text{remain}} = Forward_{\text{remain}} \times MsgSize \tag{8}$$

where $Forward_{\text{remain}}$ is the number of remaining forwarded messages, and $MsgSize$ is the size of the message.

This strategy specifically addresses the problems of short link times and less data transmission in a single connection in the opportunistic network. Prioritizing the retention of messages with few remain forwarded times increases their survival time in the network. It extends transmission opportunities to a certain extent, thereby helping to improve the overall delivery success rate. The feature of preferential retention of small messages reduces the time required for single connection transmission and adapts to the characteristics of short-term connections in opportunistic networks.

A.2 Radiation Cache

Messages entering the radiation buffer are prioritized based on the number of remaining forwarding times and the current remaining lifetime, and are sorted from smallest to largest. When the radiation buffer is full, messages at the end are eliminated first. Calculate priority using formula (9):

$$Priority = TTL_{\text{remain}} \times Forward_{\text{remain}} \quad (9)$$

where $Priority$ represents the priority of the message, $TTL_{\text{remain}}$ represents the remaining life cycle of the message, and $Forward_{\text{remain}}$ is the number of remaining forwarded messages.

The above formula (9) uses weighted calculation to preferentially retain messages with short TTL or few remaining remaining forwarding times, extending the survival time of high-value messages. Messages with short remaining lifetimes are retained first, increasing the probability of messages being passed again before they are dropped on a deadline.

A.3 Core Cache

After each message enters the core buffer, the most recent forwarding timestamp (the initial value is the entry time) and the number of forwarding times are recorded. The message's elimination score is calculated based on these two indicators and sorted from smallest to largest. When optimizing the buffer space, messages at the end of the elimination score are discarded first, and the elimination score of the message is calculated using formula (10):

$$Evict_{\text{Score}} = \alpha \cdot LAT + (1-\alpha) \cdot TTL_{\text{remain}}{}^{2} \quad (10)$$

In the above formula, $Evict_{\text{Score}}$ is the elimination score, $LAT$ (LastAccessTime) is the interval from the last access, $TTL_{\text{remain}}$ is the message remaining life cycle, and $\alpha$ is the weight coefficient.

Based on the recent access time and the remaining lifetime, priority should be given to retaining messages with high activity or high transmission value to avoid accidentally deleting messages accessed at low frequencies but not expired. Setting $\alpha$ balances message activity and lifetime, preventing extreme elimination caused by a single indicator. $TTL_{\text{remain}}$ is a reflection of the message's remaining transmission window. It uses a quadratic term formula to amplify the priority of messages with a high lifetime automatically, and unexpired messages are more likely to be retained.

*B. Cache Management Implementation*

In the multi-level cache dynamic scheduling mechanism, when a new message is received, it is judged whether the core cache has enough space. If not, the message in the core cache is moved to the edge cache to make room according to the management policy.

Algorithm 2 is the cache management algorithm process. After a communication between two terminals is completed, optimizing the buffer space in time is necessary. First, check the ACK message, delete the messages delivered from the buffer, and then check the remaining space of your buffers at all levels. Suppose the inspection result shows that the remaining space of a specific buffer level is less than 20%. In that case, the files in the current buffer will be moved to the next buffer level according to the cache management policy at all levels. Excess messages in the edge buffer will be directly discarded according to the policy. At this point, the buffer optimization process is completed and waiting for the next message delivery. Fig 4 shows the operating method of a multi-layered cache architecture.

Algorithm 2: Tiered cache management algorithm

```
def maintain_cache():
    for ack in ackList:
        if ack.msg in buffer: drop(msg)
    if (reSizeOf (core) < 0.2*N*0.2):
        sortBuffer(core)
        while(sizeOf(core) < 0.2*N*0.2) {move(core[-1],
                                            radiation)}
    if (reSizeOf (radiation) < 0.5*N*0.2):
        sortBuffer(radiation)
        while(sizeOf(radiation) < 0.5*N*0.2) {
                move(radiation[-1], edge)}
    if (reSizeOf (edge) < 0.3*N*0.2):
        sortBuffer(edge)
        while(sizeOf(edge) < 0.5*N*0.2) { drop(edge[-1])}

msgs = receive msg from other
for msg in msgs:
    if reSizeOf(core) > sizeOf(msg): core.add(msg)
    else:
        sortBuffer(core)
        while(sizeOf(core) < sizeOf(msg)){ move(core[-1],
                                            radiation)}
        core.add(msg)
maintain_cache()
```

Complexity analysis of cache management algorithm: The core overhead of a single cache maintenance is message sorting. The total number of messages is K, and the time complexity is $O(Klog_K)$, which is in the same order as mainstream benchmark algorithms such as LRU and TTL-based, and has no additional computing burden; the time complexity of message insertion and elimination is $O(log_K)$. The orderly list structure can achieve fast operations.

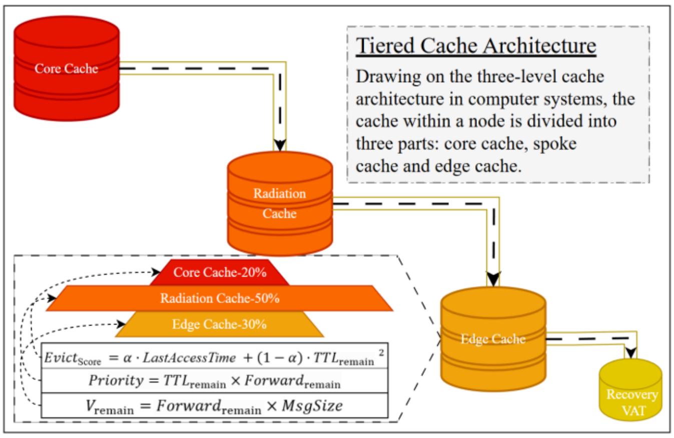


Fig. 4. Tiered Cache architecture operating process.

### C. Definition of Experimental Evaluation Index

This experiment first selects fundamental evaluation indicators such as Delivery ratio, Latency, and Overhead [19].

1) Delivery Ratio. Proportion of the number of successfully delivered messages to the total number of messages. Delivery ratio is the most direct measure of opportunistic network performance and is solved using equation below:

$$Delivery\ Ratio = \frac{Successfully\ delivered\ messages}{All\ messages} \times 100\% \tag{11}$$

2) Latency. The average time it takes for a message to be delivered from the source node to the destination node. Latency is the most direct indicator for measuring the timeliness of the opportunistic network, and it is calculated using the following formula:

$$Latency = \frac{\sum_{i=1}^{n} T_i}{n} \tag{12}$$

where $Latency$ represents the average time taken to transmit messages, $n$ represents the number of successful messages transmitted, and $T_i$ represents the time taken to transmit the ith message.

3) Overhead. Formula (13) shows the calculation process for the proportion of redundancies controlled messages (ACK messages, replica forwarding, routing requests) in the network.

$$Overhead = \frac{Data\ message}{Data\ message + Redundant\ messages} \times 100\% \tag{13}$$

## V. Performance Evaluation

The ONE simulator [20] was selected to conduct experimental simulations on PORTAC and compared it with other classic algorithms in three aspects: delivery ratio, average latency, and communication overhead. Control irrelevant variables in the experiment. Check the performance of PORTAC when buffer-size, TTL, and number of hosts are changed, respectively. In this paper, some hyperparameters are set up to macro-adjust the routing decision-making process. To this end, experiments are also conducted to verify the performance under different hyper-parameters.

### A. Experimental Environment Configuration

This experiment was simulated on the school's public experimental platform. The following is the specific experimental environment configuration in Table I.

TABLE I
SIMULATION ENVIRONMENT CONFIGURATION

| Name | Configured |
|---|---|
| CPU | Intel(R) Xeon(R) Gold 6254 CPU @ 3.10 GHz |
| operating system | CentOS Linux release 7.8.2003 |
| simulator | The ONE v1.5.1 |

The simulation selected The ONE (The Opportunistic network Environment) [20]. The ONE is an open-source simulator dedicated to opportunistic network simulation. It mainly simulates intermittent node connections and highly dynamic topology changes in network environments. It supports generating various movement trajectory models, built-in mainstream routing protocols, and dynamic display of network performance indicators. Table II shows the simulator simulation parameter settings.

TABLE II
THE ONE SIMULATOR PARAMETER CONFIGURATION

| Parameters | Value |
|---|---|
| network size | 4500*3400 |
| simulation duration | 43200s |
| preheating time | 1000s |
| communication range | 10m |
| number of nodes | [50, 70, 100, 120, 150, 170, 200] |
| mobility model | ShortestPathMapBasedMovement, RandomWaypoint |
| movement speed | 0.5m, 1.5m |
| message interval | 10s,35s |
| message size | 500k,1M |
| Message Live Time | [10, 20, 30, 50, 70, 100, 130, 150, 170, 200] min |
| buffer size | ["10M", "20M", "30M", "40M", "50M", "60M", "70M", "100M", "120M", "150M"] |

### A.1 Comparison Benchmark Selection

In this experiment, routes such as Prophet [21], Epidemic [22] were selected as the benchmark routes. These routes cover multiple routing decision methods such as probability prediction, flooding, and hybrid optimization. At the same time, the scenarios applicable to each are also different. We hope to select these routing strategies as benchmark strategies to improve the persuasiveness of the experiment. Table III shows the analysis of several comparison benchmark algorithms.

This experiment selects TTL-based and LRU as benchmark cache management strategies. TTL-based focuses on message life cycle control, and LRU focuses on dynamic message updates. Combining the two can effectively evaluate the effect of caching. Table IV provides an analysis of benchmark cache management policies.

TABLE III
COMPARATIVE BENCHMARK ALGORITHM ANALYSIS

| Route name | Core Mechanism | Comparative Significance |
|---|---|---|
| Epidemic | Messages are delivered through full-network broadcasting to maximize delivery efficiency, but consume many resources. | A benchmark for performance online that measures the balance between message delivery ratios and resource consumption of other algorithms. |
| SprayAndWait[23] | First, unthinkingly spray and forward, and wait for passive transmission after leaving one copy. | Strike a balance between Epidemic and Direct Delivery to verify the effectiveness of copy number control. |
| Prophet | Predict future probability based on historical encounter records and priority forwarding to high-probability nodes. | The performance of reflective predictive models is stable in sparse networks, but mismeasurements may cause transmission redundancy. |

TABLE IV
COMPARATIVE BENCHMARK CACHE MANAGEMENT STRATEGY ANALYSIS

| Policy Name | Core Mechanism | Comparative Significance |
|---|---|---|
| TTL-based | The buffer is sorted from largest to smallest based on the TTL of messages. When the buffer is complete, the messages at the end are discarded. | A buffering strategy based on temporal locality, but possibly ignoring the value of the message itself. |
| LRU | Sort according to the number of times the small ones in the buffer are used, prioritizing the messages that have not been used for the longest time. | Reflects time-sensitive caching strategies and reduces redundant transmissions in dynamic networks. |

## *B. Result Visualization*

To evaluate the comprehensive performance of PORTAC, this study will compare it with baseline algorithms from the following three dimensions: (1) Delivery Efficiency, measured by the Delivery Ratio; (2) Timeliness, measured by the Average latency; and (3) Resource Consumption, represented by the Overhead Ratio. An ideal routing algorithm should ensure delivery efficiency while also maintaining low latency and resource consumption. During the experiments, the default values for unspecified parameters are: TTL=100 min, Buffer-size=100M, Number of hosts=100.

### B.1 Performance under Different Buffer Sizes

When the message Time-to-Live (TTL) is fixed at 100 min and the number of hosts in the network environment is fixed at 100, this section presents a comprehensive performance evaluation of PORTAC against four stream routing algorithms. The key metrics include delivery ratio, average latency, and overhead ratio. Experimental data indicate that the PORTAC algorithm demonstrates outstanding performance in terms of both delivery ratio and average latency, achieving an effective balance between critical performance indicators and resource consumption.

As illustrated in Fig 5 (Delivery Ratio), the delivery ratio of PORTAC increases rapidly initially as the buffer size grows, and then begins to saturate after the buffer reaches approximately 70 MB, stabilizing at over 90%. This saturation point occurs significantly earlier than that of algorithms such as Epidemic, which require a buffer size exceeding 140 MB to achieve a comparable delivery ratio. Both the Prophet and SprayAndWait algorithms ultimately attain lower delivery ratios than PORTAC. These experimental observations indicate that PORTAC can utilize buffer space more efficiently, reaching its optimal performance level quickly even under limited buffer conditions.

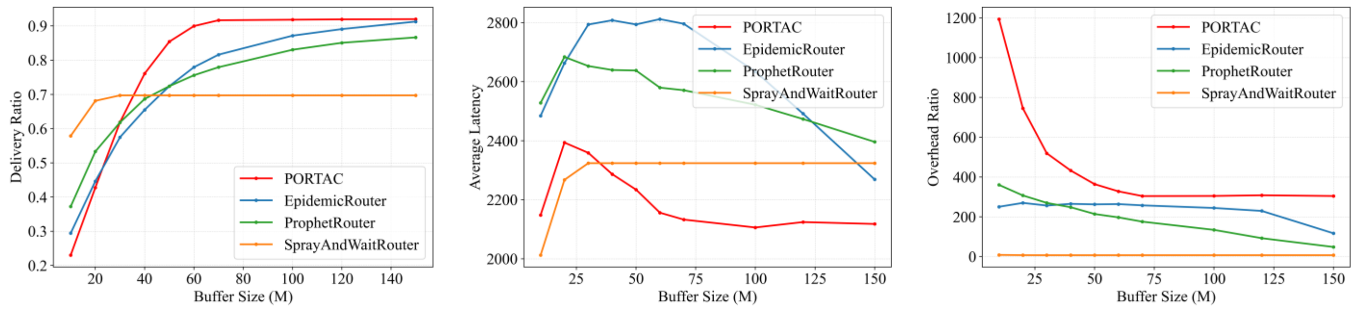


Fig. 5. When TTL=100 min and number of hosts=100, the delivery ratio, average latency and overhead ratio change with Buffer-size.

As shown in Fig 5 (Average latency), in the average latency comparison, POTACC maintains the lowest latency among all buffer sizes than other algorithms when the buffer is greater than 30M. And the average delay will quickly stabilize to around 2100 seconds. In contrast, the average latency of other algorithms is higher than PORTAC, and the average latency of some algorithms fluctuates significantly as the buffer size changes. These experimental observations show that PORTAC's routing mechanism can help shorten message transmission time between endpoints to a certain extent.

Fig 5 (Overhead Ratio) illustrates the unique behavioral pattern of PORTAC in terms of overhead ratio. Under severely constrained buffer conditions (<30M), PORTAC exhibits the highest overhead ratio, indicating that the algorithm likely incurs significant control and management overhead to safeguard core performance metrics (such as delivery probability and average latency) at this stage. When the buffer size increases beyond 20M, its overhead ratio decreases sharply and stabilizes at a moderate level. This variation in the overhead ratio suggests that PORTAC makes a reasonable trade-off, prioritizing the most critical performance indicators at the expense of higher overhead.

The experimental comparisons demonstrate that under the conditions of TTL=100 and Number of hosts=100, as the buffer capacity increases, PORTAC outperforms other algorithms in both delivery efficiency and average latency. Furthermore, PORTAC exhibits a higher overhead ratio compared to Prophet and SprayAndWait. This is a consequence of an inherent design trade-off made to achieve superior delivery ratio and lower latency. In strictly resource-constrained scenarios, this fixed overhead constitutes a more significant proportion of the total network traffic, making the ratio more pronounced. However, as the buffer size increases, PORTAC's overhead ratio decreases sharply and stabilizes. This trend indicates that when resources become ample, the relative cost of its coordination diminishes, allowing its performance advantages to be fully realized.

B.2 Performance under Different TTL

Under the condition of a fixed buffer size of 100M and the number of hosts in the network environment is fixed at 100, this section evaluates the sensitivity of PORTAC and comparative algorithms to changes in the TTL parameter, with respect to delivery ratio, average latency, and overhead ratio. A comprehensive analysis reveals that variations in TTL impose a key constraint on algorithmic performance. The PORTAC algorithm, however, demonstrates robust adaptability to the message lifetime, achieving an effective synergy of high delivery ratio, low latency, and controllable overhead across most TTL settings.

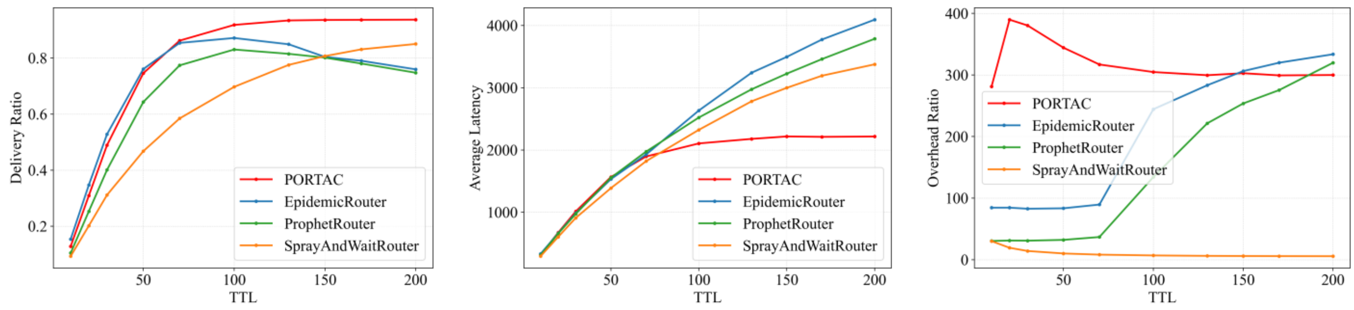


Fig. 6. When Buffer size=100MB and number of hosts=100, the delivery ratio, average latency and overhead ratio change with TTL.

Fig 6 (Delivery Ratio) illustrates the relationship between Time-to-Live (TTL) and delivery ratio. As TTL increases, all algorithms benefit from a higher probability of successful delivery, albeit with distinct patterns and efficiencies. PORTAC demonstrates a rapid rise in delivery ratio, reaching and maintaining a plateau exceeding 90% once TTL approaches approximately 100 minutes. Although algorithms like Epidemic also show an upward trend, their delivery ratios consistently remain lower than PORTAC's across the evaluated TTL range. These results indicate that PORTAC's routing mechanism can efficiently complete message delivery within a relatively generous time window, achieving the highest delivery ratio under equivalent time constraints.

Fig 6 (Average Latency) reveals the transmission timeliness of each algorithm within the reachable range. For PORTAC, the average latency stabilizes at approximately 2100 seconds once the Time-to-Live (TTL) exceeds 100, with no sustained upward trend observed. This performance is primarily driven by two core design innovations: first, the spatio-temporal coupled Contact Stability Index (CSI) model eliminates the invalid forwarding behavior that is pervasive in conventional algorithms, which drastically reduces the residence time of messages within the network; second, the three-tier hierarchical caching mechanism effectively mitigates the forwarding waiting latency caused by buffer congestion. EpidemicRouter consistently yields the highest latency across the entire TTL range. Its flooding-based network-wide broadcast forwarding mechanism generates an enormous volume of redundant message replicas as TTL extends, which in turn exacerbates network congestion. ProphetRouter and SprayAndWaitRouter achieve slightly lower latency than EpidemicRouter, yet their latency remains substantially higher than that of PORTAC, and both also increase linearly with rising TTL.

Fig 6 (Overhead Ratio) reflects the additional cost incurred by the network for message transmission as TTL increases. PORTAC exhibits a trend of rapid initial growth, followed by a decline and steady-state convergence, with no overhead explosion observed in long-TTL scenarios. EpidemicRouter and ProphetRouter maintain a low and stable overhead ratio at short TTL values, but show an exponential surge with no convergence trend as TTL extends. SprayAndWaitRouter sustains an extremely low overhead level near zero throughout the entire TTL range, with a slow downward trend as TTL increases, representing the scheme with the lowest resource consumption among all compared algorithms. Conventional routing protocols for opportunistic networks face the inherent contradiction that high delivery ratio is inevitably accompanied by high overhead. In contrast, PORTAC achieves an optimal trade-off between performance and overhead: it maintains stable and controllable overhead in long-TTL scenarios, while simultaneously achieving the highest delivery ratio and the lowest transmission latency among all evaluated algorithms.

When facing variations in the key parameter TTL, PORTAC demonstrates excellent and stable performance across all three dimensions: delivery ratio, delay, and overhead. Unlike the compared algorithms, which exhibit significant weaknesses in specific metrics (e.g., Epidemic's high overhead), PORTAC achieves a superior overall balance.

B.3 Performance under Different Host Counts

Under the conditions of a fixed buffer size (100M) and message Time-to-Live (TTL=100), this section evaluates the scalability of PORTAC and comparison algorithms by varying the network scale. The analysis results indicate that increasing network density significantly impacts algorithm performance. PORTAC demonstrates a balanced performance, maintaining a high delivery ratio, low delay, and controllable overhead even in high-node-density environments.

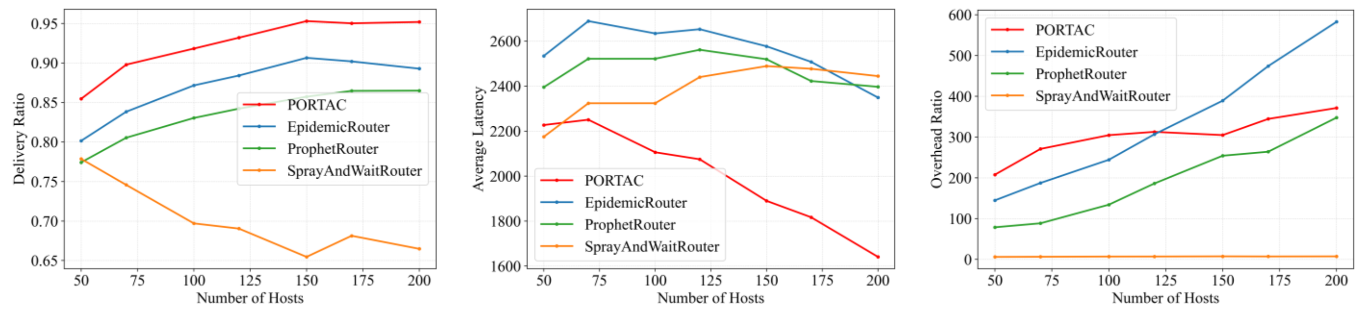


Fig. 7. When TTL=100 min and Buffer size=100MB, the delivery ratio, average latency and overhead ratio change with the number of hosts.

As shown in Fig 7 (Delivery Ratio), The four algorithms exhibit distinct adaptability to changes in network density. PORTAC consistently ranks first in delivery ratio across the entire range of network scales: its delivery ratio rises continuously with the increase in the number of hosts, reaches a peak value of 0.95 at 150 hosts, and exhibits no performance degradation in high-density scenarios thereafter. EpidemicRouter attains a peak delivery ratio of 0.9 at 150 hosts, followed by a significant decline as the number of hosts further increases. The delivery ratio of ProphetRouter rises slowly with the growing number of hosts, yet remains substantially lower than that of PORTAC throughout the entire test range. SprayAndWaitRouter shows a continuous linear decrease in delivery ratio as the number of hosts increases. The benchmark algorithms suffer from insufficient delivery ratio in sparse networks and notable performance degradation in high-density networks. In contrast, PORTAC maintains a prominent relative superiority across all scenarios, ranging from sparse networks (50 hosts) to high-density networks (200 hosts).

Fig 7 (Average Latency) reveals the transmission delays of various algorithms under different network densities. The average latency of PORTAC shows a continuous downward trend as the number of hosts increases. This indicates that the increase in network density provides more opportunities for path optimization in PORTAC, enabling it to discover more efficient transmission paths more quickly. Although the delay of Prophet also decreases with network scale, it consistently remains higher than that of PORTAC. The delay reduction of Epidemic is limited, maintaining a relatively high latency even in dense networks. The delay curve of SprayAndWait exhibits noticeable fluctuations. These observations highlight PORTAC's advantage in leveraging network density to improve transmission timeliness.

Fig 7 (Overhead Ratio) intuitively illustrates the resource consumption changes of the algorithms as the network scale expands. The overhead ratio of PORTAC exhibits a gentle and controllable linear growth with the increase in the number of hosts, with the overall growth amplitude fully constrained throughout the entire test range. EpidemicRouter shows a steep exponential rise in overhead ratio as the number of hosts increases. The overhead ratio of ProphetRouter rises continuously with the growing number of hosts. SprayAndWaitRouter maintains an extremely low overhead level near zero throughout the entire test range, with no noticeable fluctuation as the number of hosts increases. The proposed PORTAC in this paper consistently maintains a high delivery ratio across all network scale scenarios, from sparse networks to high-density networks, while its overhead ratio only presents a gentle linear growth. This achieves an optimal trade-off between high delivery ratio and controllable overhead.

Through this experimental analysis, PORTAC demonstrates notable stability across all dimensions, confirming that its algorithm design effectively adapts to networks ranging from sparse to dense. It further translates increased network connectivity opportunities into improvements in transmission efficiency. The controllable growth pattern observed in the overhead ratio constitutes one of PORTAC's most critical advantages. Compared to other algorithms, PORTAC effectively avoids the overhead explosion problem that arises with network scale expansion, highlighting its potential for large-scale deployment.

### B.4 Performance under Different Mobility Models

To evaluate the robustness of PORTAC under different mobility assumptions, we employ two distinct mobility models: 1) the ShortestPathMapBasedMovement model, which simulates map-constrained and destination-oriented movement (e.g., pedestrian or vehicle traffic), and 2) the RandomWaypoint model, which represents unconstrained, random movement patterns. This combination allows us to test the algorithm’s performance in both structured and unstructured dynamic environments. Table V shows the performance under different mobility models.

TABLE V
PERFORMANCE OF ROUTES UNDER DIFFERENT MOVEMENTMODELS

| Route Name | MovementModel | | | | | |
|---|---|---|---|---|---|---|
| | ShortestPathMapBasedMovement | | | RandomWaypoint | | |
| | Delivery Ratio | Average Latency | Overhead Ratio | Delivery Ratio | Average Latency | Overhead Ratio |
| SprayAndWaitRouter | 0.697 | 6.929 | 2324.038 | 0.106 | 3184.2809 | 26.886 |
| ProphetRouter | 0.830 | 134.126 | 2521.846 | 0.104 | 3447.735 | 22.363 |
| EpidemicRouter | 0.871 | 244.282 | 2634.219 | 0.100 | 3475.066 | 40.868 |
| PORTAC | 0.918 | 304.676 | 2106.046 | 0.110 | 3488.080 | 118.587 |

As shown in Table V. Performance Analysis under the ShortestPathMapBased-Movement Scenario： In terms of delivery ratio, PORTAC achieves a message delivery ratio of 0.918, ranking first among all compared algorithms and demonstrating significant performance superiority over the benchmark algorithms. Specifically, it realizes a 5.3 percentage points improvement in delivery ratio compared with the flooding-based EpidemicRouter (0.871), a 10.6 percentage points improvement over ProphetRouter (0.830) that relies on contact frequency prediction, and a 22.1 percentage points improvement compared with SprayAndWaitRouter (0.697) adopting a minimalist forwarding mechanism. This result validates that the spatio-temporal coupled Contact Stability Index (CSI) model can accurately capture the link state of nodes, and select highly reliable relay links with higher precision relative to the benchmark algorithms.

In terms of average latency, SprayAndWaitRouter achieves the lowest average latency of 6.929 by virtue of its minimalist mechanism that forwards messages only when the node directly encounters the destination node, yet this performance comes at the cost of an extremely low delivery ratio. PORTAC yields an average latency of 304.676, which is slightly higher than that of EpidemicRouter (244.282) and ProphetRouter (134.126). This, however, represents a reasonable trade-off to guarantee a high delivery ratio. To ensure the success rate of end-to-end transmission, PORTAC selects multi-hop relay paths with longer link duration and more stable transmission performance, ultimately constraining the latency within a fully acceptable range while achieving the highest delivery ratio.

In terms of overhead ratio, PORTAC attains an overhead ratio of 2106.046, the lowest among all algorithms, demonstrating a prominent advantage in resource utilization efficiency over the benchmark algorithms. Through the precise routing decision-making of the CSI model and the three-tier hierarchical caching mechanism, PORTAC eliminates low-value messages via a fine-grained priority policy, thus achieving a globally optimal trade-off between performance and overhead.

Performance Analysis under the RandomWaypoint Scenario: This scenario serves as an extreme stress test for the robustness of the routing algorithms, where nodes follow completely unregulated mobility patterns with extremely scarce valid link contacts. While the absolute performance of all algorithms exhibits significant degradation in this scenario, the relative performance differences fully validate the superior environmental adaptability of PORTAC.

In terms of delivery ratio, constrained by the completely random mobility pattern, the effective encounter probability between nodes and their destination nodes is extremely low, resulting in a cliff-like drop in the delivery ratio of all algorithms. Despite this, PORTAC still ranks first among all evaluated algorithms with a delivery ratio of 0.110.

In terms of average latency, the average latency of all algorithms surges to above 3000. This is mainly attributed to the negligible encounter probability between nodes and destination nodes in the random mobility scenario, where messages must undergo long-term retention in node buffers while waiting for the scarce transmission opportunities.

In terms of overhead ratio, PORTAC achieves an overhead ratio of 118.587, which is higher than that of the other benchmark algorithms. This phenomenon requires an objective analysis in conjunction with the scenario characteristics and delivery performance. In the extremely dynamic scenario with completely random mobility, valid links between nodes are extremely scarce. To capture the limited transmission opportunities, PORTAC requires more frequent interaction of CSI state metrics and message index exchange, leading to an increase in the relative proportion of control signaling in the total transmission volume.

The controlled comparative experiments across the two mobility models fully validate that the proposed PORTAC algorithm outperforms all benchmark algorithms in terms of comprehensive performance in the map-constrained structured scenario. In the extreme scenario with unconstrained random mobility, PORTAC still maintains the highest delivery ratio relative to the benchmark algorithms, with no additional degradation in transmission latency. This result demonstrates that the proposed algorithm possesses a certain degree of self-adaptability to dynamic network environments.

## *C. Robustness Analysis*

In order to verify the universality of the 20-50-30 cache ratio, this section carries out robustness verification from two dimensions: heterogeneous node caching capacity and asymmetric traffic distribution. The experimental scenario is still based on the ShortestPathMapBased Movement model. The default core parameters are node size 100, message TTL=100 min, and total cache capacity 100MB (if there are specific parameters, the actual situation shall prevail).

### C.1 Performance of nodes under heterogeneous caching capabilities

Experimental setup 3 sets of heterogeneous caching scenarios: The total number of hosts is 450, which is light and heterogeneous (30% nodes 20MB, 40% nodes 50MB, 30% nodes 80MB), medium and heterogeneous (20% nodes 10MB, 50% nodes 50MB, 30% nodes 100MB), heavy heterogeneity (20% nodes 5MB, 30% nodes 30MB, 30% nodes 70MB, 20% nodes 100MB), all node caches maintain an internal ratio of 20-50-30. Table VI is the performance of nodes under heterogeneous caching capabilities.

TABLE VI
PERFORMANCE OF NODES UNDER HETEROGENEOUS CACHING CAPABILITIES

| Level | Delivery Ratio | Overhead Ratio | Average Latency |
|---|---|---|---|
| light | 0.809 | 1600.020 | 1427.402 |
| medium | 0.852 | 1322.280 | 1404.674 |
| heavy | 0.760 | 1787.689 | 1467.207 |

Delivery Ratio: The optimal performance of medium and heterogeneous scenarios reaches 0.852, and significant attenuation occurs in heavy heterogeneous scenarios. In medium and heterogeneous scenarios, a 50MB node can serve as a stable relay cache node, a 10MB node ensures high-priority message storage through a ratio of 20-50-30, and a 100MB node caches low-value-density messages. The three-level caching mechanism maximizes the use of cache resources at this level of heterogeneity. Due to the overall balanced cache capacity in light and heterogeneous scenarios, there is a lack of large-capacity nodes to handle large-volume messages, and the cache resources of small-capacity nodes are not fully utilized. The delivery rate of heavily heterogeneous scenarios has been significantly attenuated. The core reason is that 20% of nodes only have 5MB cache. Even if the internal ratio of 20-50-30 is maintained, the core cache is only 1MB and the radiating cache is only 2.5MB, resulting in frequent overflows and losses of critical messages. The final delivery rate has significantly deteriorated.

Overhead Ratio: The medium heterogeneous cost ratio is 1322.280, which is the lowest value among all scenarios. The heavy heterogeneous cost ratio is 1787.689, which is the most serious waste of resources. In the middle and heterogeneous scenario, a 50MB node can accurately schedule messages through a three-level caching mechanism to reduce invalid forwarding and redundant copy generation; a 10MB node uses a priority elimination policy to promptly clean up low-value messages to avoid repeated forwarding caused by cache overflows; A 100MB large-capacity node reduces the overhead of frequent migration of messages due to insufficient caching, and the three work together to achieve the lowest resource consumption; The high overhead of heavy heterogeneous scenarios stems from the frequent overflows of the cache of nodes with a minimum capacity of 5MB. Messages are repeatedly forwarded, creating a large number of redundant copies. At the same time, the frequency of cache state interactions between nodes is greatly increased, and the proportion of control message overhead increases. Eventually, the overhead ratio increases significantly.

Average Latency: Each scene remains stable, with little impact from heterogeneity. The core reason is that the core routing decision of the algorithm relies on the CSI contact stability model rather than the node buffer capacity, which can accurately screen relay nodes with stable links and reduce the retention time of messages in the network.

The PORTAC algorithm achieves comprehensive optimal performance in delivery rate, overhead ratio, and latency in medium and heterogeneous scenarios. It proves that the internal three-level cache ratio design of 20-50-30 can effectively adapt to moderately heterogeneous caching environments and maximize the utilization of differentiated cache resources.

### C.2 Performance under asymmetric flow distribution

The experiments were divided into three groups for comparison. The benchmark mixed traffic scenario (BaseMix) was used as a typical load scenario for comparison to simulate the daily stable load of the opportunistic network. The high-priority message surge scenario (HPScan) simulates the sudden extreme load of critical instructions (such as rescue dispatch) in emergency communications, disaster rescue and other scenarios. The Large-Volume Low-Value Message Surge Scenario (LLSurge) simulates the load of large-volume low-value messages such as log data in daily scenarios. Table VII is the performance under asymmetric flow distribution.

TABLE VII
PERFORMANCE UNDER ASYMMETRIC FLOW DISTRIBUTION

| Scene | Delivery Ratio | Overhead Ratio | Average Latency |
|---|---|---|---|
| BaseMix | 0.918 | 243.372 | 2071.112 |
| HPSurge | 0.924 | 264.748 | 2090.391 |
| LLSurge | 0.901 | 231.420 | 2269.509 |

Delivery Ratio: The delivery rate for each scenario remains above 0.9, indicating that PORTAC's three-level cache architecture can effectively adapt to the sudden change of twice the traffic proportion, and the elastic design of core/edge cache meets the reliable transmission needs of extreme traffic scenarios.

Overhead Ratio: The overhead is controllable under sudden changes in traffic and there is no risk of explosion. The changes in the overhead ratio are all less than 10%, verifying that PORTAC has no overhead explosion caused by redundant forwarding and cache congestion under sudden traffic changes.

Average Latency: The proliferation of high-priority messages has no significant impact on latency, while the proliferation of large-volume messages leads to a moderate increase in latency. The priority scheduling mechanism of the core cache in the HPScan scenario ensures that high-priority messages are forwarded first, and the stable links screened by the CSI model reduce invalid jumps of messages. Even if the traffic doubles, the transmission delay of key instructions remains basically unchanged. The transmission of large-volume messages in the LLSurge scenario requires more link bandwidth, resulting in an increase in latency of about 9.57%. However, this type of message latency is not sensitive, and this increase is in line with the scenario requirements.

In the extreme scenario where high-priority/large-volume message traffic doubles, the PORTAC algorithm still maintains a delivery rate above 0.9, and the change in overhead and latency is little, verifying the elastic design of the three-level cache architecture (20-50-30). It can fully adapt to twice the mutation of asymmetric traffic.

## VI. CONCLUSION

This paper proposes PORTAC, which innovatively combines time and space dimensions to build a node stability model. At the same time, it also applies a three-stage caching mechanism to the opportunistic network to systematically optimize the conflict between transfer efficiency and limited resources in the opportunistic network. Experimental verification of the PORTAC routing policy was completed using the opportunistic network simulator, The Other NE. The verification results show that the delivery ratio of the PORTAC routing policy reaches up to 91.8% in the urban public network scenario, which is increased by more than 10.6% compared with the traditional PropheRouter routing (TTL=100min, hosts=150, Butter_size=100M).

Future work will design an online adaptively adjusted three-level cache ratio mechanism based on the real-time traffic characteristics of the network and the heterogeneous capabilities of nodes to further improve performance under asymmetric traffic and heterogeneous node scenarios; at the same time, for IoT nodes with limited computing power, design a lightweight CSI indicator calculation method to reduce control signaling interaction and calculation overhead. We hope to have more important significance for resource optimization and transmission efficiency improvement in a highly dynamic network environment.

## ACKNOWLEDGEMENT

This work is partly supported by the key laboratory project of the Ministry of Culture and Tourism under grant No 2022-13, 2023-02, the university-industry collaborative education program of Ministry of Education under grant No. 231101867011642, the fundamental research funds for the central universities China under grant No. GK202407007, the national natural science foundation of China under grant No. 62377034, the Shaanxi key science and technology innovation team project under grant No. 2022TD-26, the national social science fund of China under grant No. 24CTQ016, Xi'an social sciences planning fund project under grant No. 24LW31.

## REFERENCES

[1] A. Dutta, S. J. Borah, and J. Singh, "Location-Based Clustering Approach for Next-Hop Selection in Opportunistic Networks," in The 6th International Conference on Wireless, Intelligent and Distributed Environment for Communication, Cham, I. Woungang and S. K. Dhurandher, Eds. Cham, Switzerland: Springer Nature Switzerland, 2024, pp. 91-109.

[2] A. Malik, B. Bhushan, and A. Kumar, "Association Rule-Based Routing Protocol for Opportunistic Network," in Innovations in Electronics and Communication Engineering, H. S. Saini, R. K. Singh, M. Tariq Beg, R. Mulaveesala, and M. R. Mahmood, Eds. Singapore: Springer, 2022, pp. 391-399.

[3] J. Cui, Y. Zhang, S. Gao, and R. Wang, "Analysis of Routing Performance Based on Opportunistic Networks in Vehicular Ad-Hoc Networks," in CICTP 2020, Proceedings, 2020, pp. 234–245.

[4] Á. Niebla-Montero, I. Froiz-Míguez, P. Fraga-Lamas, and T. M. Fernández-Caramés, "Design, Implementation and Practical Evaluation of an Opportunistic Communications Protocol Based on Bluetooth Mesh and libp2p," Sensors, vol. 25, no. 4, doi: 10.3390/s25041190.

[5] P. Garg, A. Dixit, P. Sethi, and J. Pruthi, "Strengthening Smart City with Opportunistic Networks: An Insight," in 2023 International Conference on Advanced Computing & Communication Technologies (ICACCTech), 2023, pp. 700-707, doi: 10.1109/ICACCTech61146.2023.00117.

[6] F. Huang, J. Q. Cui, Y. A. Chang, T. Wang, M. Wang, and Y. Yang, "An Improved Spray and Wait Routing Algorithm Based on Stable Transmission Capacity Evaluation of Nodes in Opportunistic Network," International Journal of Distributed Sensor Networks, vol. 2024, Art no. 7471329, doi: 10.1155/2024/7471329.

[7] Y. H. Fu, B. Zhou, G. Xu, Y. Q. Wang, B. Q. Huang, and F. Zhang, "An energy recovery routing algorithm for opportunistic networks based on connection stability and node encounter type," Ad Hoc Networks, vol. 157, 2024, Art no. 103455, doi: 10.1016/j.adhoc.2024.103455.

[8] J. Ren, R. Wang, H. Jia, Y. Li, and P. Pei, "Link Transmission Stability Detection Based on Deep Learning in Opportunistic Networks," in Multimedia Technology and Enhanced Learning, Cham, B. Wang, Z. Hu, X. Jiang, and Y.-D. Zhang, Eds. Cham, Switzerland: Springer Nature Switzerland, 2024, pp. 110–126.

[9] X. Han, H. Wang, J. Tan, H. Lv, and C. Wang, "MR-Pareto: A Multiattribute Opportunistic Routing Method Based on Pareto Optimal Solution for Mobile Crowdsensing," Wireless Communications and Mobile Computing, vol. 2022, Art. no. 3123615, doi: 10.1155/2022/3123615.

[10] J. Shu, J. Li, and X. Zhang, "Link Prediction Based on 3D Convolutional Neural Network," in *2022 IEEE/CIC International Conference on Communications in China (ICCC)*, 2022, pp. 156–161, doi: 10.1109/ICCC55456.2022.9880708.

[11] A. M. Vegni, C. B. Iglesias, and V. Loscrí, "MOVES: A MemOry-based VEhicular Social forwarding technique," *Computer Networks*, vol. 197, Art. no. 108324, 2021, doi: 10.1016/j.comnet.2021.108324.

[12] Y. Cao, P. Li, T. Liang, X. Wu, X. Wang, and Y. Cui, "A Novel Opportunistic Network Routing Method on Campus Based on the Improved Markov Model," Applied Sciences, vol. 13, no. 8, doi: 10.3390/app13085217.

[13] S. Yin, J. Wu, and G. Yu, "Low energy consumption routing algorithm based on message importance in opportunistic social networks," Peer-to-Peer Networking and Applications, vol. 14, no. 2, pp. 948–961, 2021/03/01 2021, doi: 10.1007/s12083-021-01072-y.

[14] S. Chen, Z. Chen, J. Wu, and K. Liu, "An Adaptive Delay-Tolerant Routing Algorithm for Data Transmission in Opportunistic Social Networks," Electronics, vol. 9, no. 11, doi: 10.3390/electronics9111915.

[15] D. Wu, B. Liu, Q. Yang, and R. Wang, "Social-aware cooperative caching mechanism in mobile social networks," *Journal of Network and Computer Applications*, vol. 149, Art. no. 102457, 2020, doi: 10.1016/j.jnca.2019.102457.

[16] P. Li, H. Liu, L. Guo, L. Zhang, X. Wang, and X. Wu, "High-Quality Learning Resource Dissemination Based on Opportunistic Networks in Campus Collaborative Learning Context," in *Wireless Sensor Networks*, S. Guo, K. Liu, C. Chen, and H. Huang, Eds. Singapore: Springer Singapore, 2019, pp. 236–248, doi: 10.1007/978-981-15-1785-3_18.

[17] Z. Zeng et al., "CMCache: An Adaptive Cross-Level Data Placement Method for Multi-Level Cache," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, pp. 1–1, 2025, doi: 10.1109/TCAD.2025.3534116.

[18] M. Jalili, and M. Erez, "Reducing Load Latency with Cache Level Prediction," in 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA), 2022, pp. 648–661, doi: 10.1109/HPCA53966.2022.00054.

[19] H. Ma, Y. Gu, H. Wu, L. Xing, and X. Zhang, "A dual incentive mechanism based on graph attention neural network and contract in mobile opportunistic networks," *Physical Communication*, vol. 67, Art. no. 102485, 2024, doi: 10.1016/j.phycom.2024.102485.

[20] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *Proc. 2nd International Conference on Simulation Tools and Techniques (SimuTools)*, 2009, doi: 10.4108/ICST.SIMUTOOLS2009.5674.

[21] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," SIGMOBILE Mob. Comput. Commun. Rev., vol. 7, no. 3, pp. 19–20, 2003, doi: 10.1145/961268.961272.

[22] A. Vahdat, and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Handbook of Systemic Autoimmune Diseases, 2000.

[23] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," USC, 2005.

**Guangxu Wang** is pursuing the M.S. degree with the School of Artificial Intelligence and Computer Science, Shaanxi Normal University, Xi'an, China. His main research interests are opportunistic networks, learner social networks.

**Peng Li** received his Ph.D. degree in Educational Technology from Beijing Normal University in 2010. Since July 2010, he has been engaged in teaching and research in the School of Computer Science, Shaanxi Normal University. He is an associate professor, vice dean, member of Chinese Computer Society, executive director of Shaanxi Computer Education Society, member of Shaanxi Computer Society, and member of IEEE CS. His main research interests are internet of things, media computing, opportunistic networks, and educational information science and technology.

**Fei Hao** received the Ph.D. degree in Computer Science and Engineering from Soonchunhyang University, South Korea, in 2016. From 2020 to 2022, he was a Marie Curie Fellow with the University of Exeter, Exeter, United Kingdom. He is currently a Professor with the School of Artificial Intelligence and Computer Science, Shaanxi Normal University, Xi'an, China. His research interests include social computing, soft computing, big data analytics, pervasive computing, and data mining. He has published more than 200 papers in the leading international journals and conference proceedings, such as IEEE Transactions on Parallel and Distributed Systems (TPDS), IEEE Transactions on Services Computing (TSC), IEEE Transactions on Network Science and Engineering (TNSE), IEEE Communications Magazine, IEEE Internet Computing, ACM Transactions on Multimedia Computing, Communications and Applications as well as ACM SIGIR, IEEE GLOBECOM. In addition, he was the recipient of the Best Paper Award from IEEE GreenCom 2013. He was also the recipient of the Outstanding Service Award at IEEE DSS 2018, and IEEE SmartData 2017, the IEEE Outstanding Leadership Award at IEEE CPSCom 2013, and the 2015 Chinese Government Award for Outstanding Self-Financed Students Abroad. He is also a member of ACM, CCF, and KIPS.

**Shu Zhao** received her Ph.D. in Educational Technology from Beijing Normal University in 2013. She is currently an associate professor in the School of Education at Shaanxi Normal University. Her research interests are primarily focused on AI-supported teaching.

**Xuanwen Hao** received his Ph.D. degree in cryptography from Xidian University in 2013. Since July 2001, he has been working in computer education, network security, intellectual property and other related fields at Shaanxi Normal University. His research interests include cybersecurity, computer education, intellectual property, and more. Since 2016, served as the publication chair of NaNA (International Conference on Networking and Network Applications, IEEE Publication).

**Huan Jia** is pursuing the M.S. degree with the School of Computer Science, Shaanxi Normal university, Xi'an, China. His main research interests are opportunistic networks, social network and mobile computing.