IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 1, 2019, accepted February 20, 2019, date of publication March 13, 2019, date of current version April 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2904750

An Adaptive Spray and Wait Routing Algorithm Based on Quality of Node in Delay Tolerant Network

JIANQUN CUI[1], SHUQIN CAO[1], YANAN CHANG[1], LIBING WU[2], DAN LIU[3], AND YI YANG[4]
[1]Computer School, Central China Normal University, Wuhan 430079, China
[2]Computer School, Wuhan University, Wuhan 430072, China
[3]School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China
[4]Department of Math and Computer Science, Fontbonne University, St. Louis, MO 63105, USA

Corresponding authors: Yanan Chang (ynchang@mail.ccnu.edu.cn) and Libing Wu (wu@whu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672257, Grant 61702210, and Grant 61772377, and in part by the Science and Technology Planning Project of Shenzhen under Grant JCYJ20170818112550194.

ABSTRACT The Internet of Things is one of the new emerging application domains that require delay tolerant network (DTN) support, where an end-to-end path between the source and the destination may not always exist. Due to the intermittent connectivity of DTN, the design of an efficient routing algorithm is the main challenge. In this paper, we first define a metric called message handling capacity to determine the ability of a node to forward messages. Then, we introduce a concept called connection strength to reflect the connection time between nodes and then integrate the concept into delivery predictability used by Prophet to determine the chance of a node completely delivering a message to the destination. Subsequently, we present a metric called quality of node (QoN), which is calculated by combining the relative weights of the message handling capacity and the improved delivery predictability. Finally, we present an adaptive spray and wait routing algorithm based on QoN (QoN-ASW). The QoN-ASW adaptively allocates the number of message copies between the encountered nodes according to the proportion of quality of node in the spray phase, which avoids the blindness of replica distribution. In addition, a forwarding scheme is implemented in the wait phase, which takes full advantage of encounter opportunities. In the simulation, we demonstrate the efficiency of integrating the connection strength into delivery predictability and compare the QoN-ASW with four existing DTN routing algorithms from four aspects. The simulation results show that the QoN-ASW can significantly improve the delivery rate and reduce the average delay while achieving a relatively low overhead.

INDEX TERMS Connection strength, delay tolerant network, message handling capacity, quality of node, spray and wait.

I. INTRODUCTION

With the arise of the Internet of Things (IoT), several specific protocols are being widely used by application developers. Delay Tolerant Networks (DTN) are designed to support interoperable communications between challenging networks, where an end-to-end path between the source and the destination may not always available [1]. DTN has been widely applied to various fields, such as social learning network [2], vehicular Ad-Hoc network [3]–[7], sensor network [8], interplanetary network [9], which is considered as a key technology to achieve ubiquitous networks. It is hard to predict encounter opportunity, nodes, and place in DTN. Therefore, DTN communication adopts a ''Store-Carry-Forward'' scheme [10]. Due to the intermittent connectivity, long delay, and high bit error rate of DTN, the design of an efficient routing algorithm is the main challenge.

Many routing algorithms have been proposed to enable message delivery in such challenging environment. For example, Epidemic [11] is the most famous DTN routing algorithm, in which each node spreads its messages to all encountered nodes. Epidemic gets a high delivery rate at the

The associate editor coordinating the review of this manuscript and approving it for publication was Huan Zhou.

expense of network resource utilization. In order to limit the useless replication of messages, Lindgren *et al.* [12] proposed a probabilistic routing algorithm called Prophet, which adopts delivery predictability to achieve effective selection of the next relay nodes. However, Prophet still has a high overhead, because it does not limit the maximum number of replications. Spyropoulos *et al.* [13] proposed a new routing scheme called Spray and Wait, the scheme makes an effort to reduce network consumption further, the source of a message initially starts with L copies, each node sprays half of its copies to encountered nodes, when a node is left with only one copy, it switches to wait phase: it waits for a direct transmission until it encounters the destination.

Since the network environment is heterogeneous, then many variants of Spray and Wait have been proposed [14]–[17], all of these studies essentially introduced various utility metrics, such as delivery predictability, encounter frequency, and node centrality, to select relay nodes or smartly distribute message copies. Previous studies perform satisfactorily, but they still have some disadvantages. First, nodes with a high metric value may have a poor ability to forward messages to other nodes. If a message is forwarded to a node with poor ability, the message may be discarded and the delivery rate may be reduced. Second, these schemes may not perform well in the following scenario. If the connection time between nodes is very short, then it is unreasonable to distribute message copies only according to some utility metrics without consideration of connection time. Since nodes are relatively mobile and their speed of message transmission is limited. A successful transmission requires the node carrying the message to meet the destination, but more than accomplishing that, the connection time should be long enough to completely deliver the message.

In this paper, we present an adaptive spray and wait routing algorithm based on quality of node (QoN-ASW). Our main contributions are summarized as follows:

(i) We define a new metric called message handling capacity to reflect the ability of a node to forward messages to other nodes, and then apply the metric to avoid messages flow to some nodes with poor ability to forward messages. Moreover, a time slicing technique is adopted to calculate the metric more accurately.

(ii) We define a concept called connection strength to reflect the connection time between nodes, and then improve the delivery predictability metric by adding connection strength to avoid a situation where the message cannot be delivered completely owing to the short connection time.

(iii) we present a metric called quality of node, which is calculated by combining the relative weights of the message handling capacity and the improved delivery predictability. Then we present an adaptive asymmetric spray scheme based on quality of node to avoid the blindness of replica distribution in spray phase. The motivation is to adaptively distribute the number of message copies between encountered nodes to adapt to the changing network conditions.

(iv) We change passive wait scheme into forwarding scheme to improve the wait phase. Our scheme takes advantage of encounter opportunities (unlike direct transmission). When a node is left with only one copy, it forwards the copy to an appropriate candidate node with higher improved delivery predictability instead of natively waiting for the destination to be encountered.

The rest of the paper is organized as follows. Related work is discussed in Section II. In Section III, we present the system model. The details of the proposed scheme are described in section IV. Section V demonstrates the efficiency of connection strength and compares the performance of the proposed scheme with existing routing schemes. Finally, we give the conclusions and the directions for future work in section VI.

II. RELATED WORK

In view of the fact that the design of an efficient routing algorithm is one of the most challenging problems in DTN, a growing class of researches have been done on the DTN routing problems in recent years. For example, Direct Delivery (DD) [18] is the simplest method, in which a source node holds its messages until it encounters the destination, thus directly delivering the message to the destination without any additional relay operation. Therefore, DD has the smallest overhead but a long delivery delay. On contrary, Epidemic is a typical flooding-based routing algorithm. Messages are simply flooded to all encountered nodes (without any optimization strategy) to maximize the chances of reaching the destination. Epidemic creates a highest delivery rate with minimum delay when network resources are unlimited, which is never the case in reality [19], [20]. However, redundant message copies in the network will consume a lot of network resources and suffer from contention. Therefore, Epidemic is not suitable for resource-constrained scenarios.

Then several routing algorithms have taken the selection of the next hop into consideration to improve the routing performance of Epidemic. For example, Prophet [12] adopts the delivery predictability metric to reflect the probability of encountering a certain node. The algorithm decides whether to forward a certain message to the encountered nodes based on the delivery predictability, which reduces the useless replication. Balasubramanian *et al.* [21] proposed a rapid (Resource Allocation Protocol for International DTN) routing, which treats the routing as a resource allocation problem. At each transfer opportunity, a rapid node replicates messages in the order of their marginal utility of replication.

Spray and Wait [13] is another significant work, which combines the fast message diffusion speed of Epidemic and the simplicity of DD. It dedicates to achieving balance between the delivery rate and the overhead. In the algorithm, the source of a message initially starts with L copies; any node v_i that has $n > 1$ copies of a message (source or relay), and encounters another node v_j (with no copies), hands over to v_j $\lfloor n/2 \rfloor$ copies and keeps $\lceil n/2 \rceil$ copies for itself;

when it is left with only one copy, it switches to direct transmission. Since the wait phase does not perform well, relay nodes passively wait for the opportunity to meet with the destination. For that, Spray and Focus (SNF) [22] has been proposed to change the passive wait scheme of Spray and Wait. Messages are forwarded to the nodes with higher utility value until the destination is reached, thereby improving the routing performance.

Moreover, considering the node movement state, delivery predictability, encounter counter information, distance, and other factors have effects on the performance of routing algorithms [23]–[26], many variants have been proposed to optimize Spray and Wait. Such as R-ASW (Relay-probability-based Adaptive Spray and Wait) [27], which uses the Relay-probability of receiver nodes to decide whether to forward a certain message to the encountered nodes. When a node is left with only one copy, it will still forward the copy to the node with higher Relay-probability until the destination is reached. Kim *et al.* [15] proposed a probability-based spray and wait (PBSW). The main contribution of PBSW is to allocate the number of message copies between encountered nodes according to the proportion of the delivery predictability. The study of [16] uses the node movement state information and encounter counter information to select the next hop, and an active search phase is implemented in wait phase. Wang *et al.* [28] proposed the SW-CCN routing algorithm, which uses node connection intensity (NCI) and node connection stability (NCS) to evaluate the connection-capability of node (CCN). The SW-CCN routing calculates the NCI based on the average separation time and calculates the NCS based on the root mean square value of the average separation time. However, if messages are forwarded to nodes with a high CCN but a poor ability to forward messages to other nodes, it may cause messages to be discarded and reduce the delivery rate. Therefore, the ability of a node to forward messages to other nodes is also need to be considered. Wang *et al.* [29] proposed the CoN-ASW routing algorithm, which uses capability of node to calculate the number of message copies to be forwarded and decide whether to forward a message to another node. The capability of node is calculated according to the encounter times and the link time between nodes. However, the network environment is dynamically changing, the capability of node needs to be updated dynamically. Sadat and Mohiuddin [30] proposed a spraying heuristic which is based on neighborhood contact history. They hold the intuition that frequent neighbor nodes present a high neighborhood index and then can expect that these nodes will successfully deliver the message to the destination. For that, the algorithm calculates the number of message copies to be forwarded based on node's neighborhood index.

To further improve the routing performance, several routing algorithms integrate with other schemes to better guide forwarding decisions. The aim is to make full use of their advantages, such as SFMS (Spray and Forwarding scheme with Message Scheduling) [31], which integrates the flooding-based and forwarding-based approaches. The algorithm lets a more popular node keep more message copies to spray in popularity spray phase and forwards messages to the node with a higher delivery utility in utility-based forwarding phase. Matis *et al.* [32] proposed an enhanced hybrid social based routing algorithm, which is fully decentralized exploiting main techniques of both Dynamic Source Routing and Social Based Opportunistic Routing algorithms. Pan *et al.* [33] proposed the Spray and Wait with Probability Choice (SWPC) routing, which is a combination of Prophet and Spray and Wait. The algorithm sets up a delivery predictability function to adjust the number of message copies in spray phase. And messages are forwarded to the nodes with higher delivery predictability in wait phase. Mei *et al.* [34] implemented a simple routing rule based on interest profile similarity to overcome the storage capacity problem, which combines both advantages of social-aware and stateless algorithms. The algorithm makes use of the observation that the individuals with similar interest profile tend to meet more often.

Notice that many mobile devices are used by people, individuals always try to maximize their utilizes through rationally selecting specific behaviors, which directly influence decision making in the network [35], [36]. Therefore, Faye *et al.* [37] explored different methodologies to characterize human mobility features computed from wearable sensor data. Hui *et al.* [38] discover that both of community structure and centrality can be effectively used to achieve further performance improvements in forwarding decisions. When a node has a message destined for another node, the node will first bubble the message up based on the global centrality to a node with the same community as the destination, and then bubble the message up based on local centrality until it encounters the destination. A recent research [17] selects the next hop based on the social relationship to increase the chances of reaching the destination and avoid dead end problem. Wang *et al.* [39] proposed an efficient routing algorithm based on social link awareness called SLABR. The algorithm calculates social links of nodes pairs according to their encounter history, and then uses the social links to construct the friendship communities of nodes. Furthermore, the single-copy based forwarding scheme is implemented in the inter-community, and the multi-copy based replication scheme is adopted in the intra-community. Bulut and Szymanski [40] proposed Friendship-Based Routing (FBR), which utilizes the metric called friendship relations to detect social relations between nodes accurately. Given that the main activities of individuals are periodic, FBR uses periodic friendship communities to capture the periodic changes of node relations. Zhang and Zhao [41] study data diffusion schemes based on the ''homophily'' phenomenon in social network. They suggested that a node should diffuse the data most similar between friends, and diffuse the data most different between strangers.

However, all above works focus on social and mobility features without the consideration of the message handling capacity. If messages are forwarded to nodes with a high metric value but a poor ability to forward messages to other nodes, it may cause messages to be discarded and reduce the delivery rate. In addition, the delivery predictability metric that most of the existing routing algorithms adopt only denotes the encounter frequency between nodes. Actually, a successful transmission depends not only on whether the node carrying the message can meet with the destination, but also on whether the connection time is long enough for delivering the message completely. In our work, we define a new metric called message handling capacity to reflect the ability of a node to forward messages to other nodes, and we also improve the delivery predictability metric by adding connection strength between nodes to optimize the routing performance. Finally, we present an adaptive spray and wait routing algorithm based on quality of node (QoN-ASW). In spray phase, QoN-ASW adaptively allocates the number of message copies between encountered nodes according to the proportion of quality of node, which is given by combining the normalized relative weights of the message handling capacity and the improved delivery predictability. In wait phase, a forwarding scheme is implemented to take full advantage of encounter opportunities.

III. SYSTEM MODEL

A. NETWORK MODEL

We assume that the network contains n nodes, $G = (V, E)$ is a network topology graph, $V = \{v_i | 1 < i < n\}$ represents the set of nodes in the network, and E is the set of edges defined on G. Each message m_k in the network has a unique identifier (messageID). When a node generates a new message m_k, its corresponding destination node v_k, Time to Live (TTL), and maximum copy number L will be pre-assigned. If the TTL of the message expires, it will be discarded.

B. CALCULATION OF MESSAGE HANDLING CAPACITY

In real network environment, the ability to forward messages between nodes is significantly different. The number of message copies that a node has successfully forwarded can intuitively reflect the message handling capacity of nodes. However, the network environment is dynamically changing, some nodes successfully forwarded many message copies at initially, and the later ability of them to forward messages is poor. Therefore, we divide the time into several time slots with the same length to estimate the message handling capacity more accurately.

Suppose node v_i is currently in *kth* time slot, since the number of message copies that node v_i will successfully forward in *kth* time slot is unknown, we estimate the message handling capacity of v_i in *kth* time slot according to the number of message copies that v_i has successfully forwarded since the start time of the *kth* time slot.

Definition 1: Estimated Message Handling Capacity (EMHC). The EMHC of node v_i in kth time slot, say $EMHC_i^k$, is the estimated number of message copies that node v_i will successfully forward in kth time slot. $EMHC_i^k$ can be calculated by:

$$EMHC_i^k = \begin{cases} \dfrac{m_{cur}}{\Delta t} \cdot TS, & \text{if } k = 1; \\ \dfrac{m_{cur} - \sum_{j=1}^{k-1} m\_j}{\Delta t} \cdot TS, & \text{if } k > 1. \end{cases} \tag{1}$$

where m_{cur} denotes the number of message copies that v_i has successfully forwarded so far, m_j indicates the number of message copies that v_i successfully forwarded in *jth* time slot, Δt represents the time that has elapsed since the end time of previous time slot, and TS is the length of time represented by a time slot which is set to 1 hour in this paper.

Based on Definition 1, we introduce a metric called message handling capacity for evaluating the ability of a node to forward messages.

Definition 2: Message Handling Capacity (MHC). The MHC of node v_i in kth time slot, say MHC_i^k, is the normalized number of message copies that v_i has successfully forwarded. MHC_i^k can be computed as:

$$MHC_i^k = \begin{cases} EMHC_i^k, & \text{if } k = 1; \\ \alpha \cdot MHC_i^{k-1} + (1-\alpha) \cdot EMHC_i^k, & \text{if } k > 1. \end{cases} \tag{2}$$

where $\alpha \in [0, 1]$ is a smoothing parameter that decides how much impact the MHC_i^{k-1} should have on the MHC_i^k.

When two nodes encounter, both of them should update their MHC_i^k firstly. The value of MHC_i^k can be used to measure the ability of v_i to forward messages. Node v_i with a larger MHC_i^k has better message handling capacity, which means we can distribute more number of message copies to v_i.

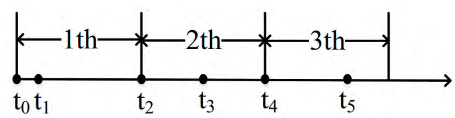


FIGURE 1. Time division diagram.

We take Fig. 1 as an example to illustrate the calculation and updating of MHC_i^k. In this figure, we only show the calculation and updating of MHC_i^k at some time points of the first three time slots. Some parameters are given in advance. The m_1 that indicates the number of message copies that v_i has successfully forwarded in 1*th* time slot is 100, m_2 is 156, and other specific parameters given in advance are shown in Table 1.

Then we use the values given in advance to calculate the value of $EMHC_i^k$ and MHC_i^k in Fig. 1. We take the concrete calculation and updating of MHC_i^k at time t_1 as an example. At t_1, node v_i is currently in 1*th* time slot and has successfully forwarded 16 message copies, the Δt represented by $[t_0,t_1]$ interval is 10 minutes. Refer to (1), $EMHC_i^1 = \frac{16}{10} \cdot 60 = 96$. Because $k = 1$, $MHC_i^1 = EMHC_i^1 = 96$. The $EMHC_i^k$ and MHC_i^k of v_i at other time points are shown in Table 2.

TABLE 1. Specific parameters.

	time (min)	m_{cur}
t_0	0	0
t_1	10	16
t_2	60	100
t_3	90	189
t_4	120	256
t_5	160	266

TABLE 2. The value of $EMHC_i^k$ and MHC_i^k in Fig. 1.

	$EMHC_i^k$	MHC_i^k
t_1	96	96
t_2	100	100
t_3	178	166.3
t_4	156	147.6
t_5	15	34.89

C. CALCULATION OF THE IMPROVED DELIVERY PREDICTABILITY

The network environment is relative unpredictability, but human mobility is not completely randomly, and if node v_i frequently encounters node v_j in the past, it is likely that they will be encounter again in the future. Lindgren *et al.* [12] proposed Prophet based on this observation. The routing algorithm uses encounter frequency to predict the probability of encountering a certain node. The algorithm defines a probabilistic metric called delivery predictability to reflect the chance of a node delivering a message to the destination node. When two nodes encounter, a decision is then made on whether to forward a message to another node based on the probabilistic metric. However, messages have some certain size and the speed of message transmission is limited, connection time is another critical factor for achieving a successful message delivery. For example, in case where two nodes meet frequently and the connection time between the two nodes is very short, messages cannot be delivered completely in such a short period of time. Therefore, we define a metric called connection strength.

Definition 3: Connection Strength (CS). The connection strength between node v_i and node v_j, say $CS_{(i,j)}$, is the ratio of the total connection time between v_i and v_j to the total time. $CS_{(i,j)}$ is determined by:

$$CS_{(i,j)} = \frac{\sum_{l=1}^{n}(S_{ij}^l - E_{ij}^l)}{T} \tag{3}$$

where S_{ij}^l indicates the start time of the lth connection between v_i and v_j, E_{ij}^l represents the end time of the lth connection between v_i and v_j, n is the number of connections between v_i and v_j, and T denotes the total time.

If the total connection time between v_i and v_j is longer, then the connection strength between v_i and v_j is stronger, and the probability of v_i completely delivering a message to v_j is higher. The encounter frequency is a decisive factor that affects the encounter probability between different nodes. Connection strength between nodes is a decisive factor that affects whether a message can be delivered completely after an encounter. Combining these two factors together, the delivery predictability is more representative. Therefore, we integrate connection strength between nodes into delivery predictability to improve the delivery predictability. The calculation of the improved delivery predictability contains the following three parts.

- Direct update: When node v_i encounters node v_j, improved delivery predictability should be updated according to (4). Where P_{init} refers to an initialization constant, and λ is the influence factor of the connection strength on delivery predictability,
$$P_{(i,j)} = P_{(i,j)_{old}} + (1 - P_{(i,j)_{old}}) \cdot P_{init} \cdot \lambda^{CS_{(i,j)}} \tag{4}$$
- Aging: If a pair of nodes does not encounter each other in a while, then the metric must age according to (5), where $\xi \in [0, 1)$ refers to an aging constant, and the number of time units that have elapsed since the last time the improved delivery predictability was aged is denoted as τ.
$$P_{(i,j)} = P_{(i,j)_{old}} \cdot \xi^{\tau} \tag{5}$$
- Transitive update: The metric also has a transitive property, which means if v_i frequently encounters v_j, and v_j frequently encounters v_r, then it can be inferred that v_r is probably an appropriate node to forward messages to v_i. Just as (6), where $\beta \in [0, 1]$ refers to a scaling constant that determines the impact of transitivity on the metric.
$$P_{(i,r)} = P_{(i,r)_{old}} + (1 - P_{(i,r)_{old}}) \cdot P_{(i,j)} \cdot P_{(j,r)} \cdot \beta \tag{6}$$

D. QUANTIFYING METHOD OF THE QUALITY OF NODE

To make the replica distribution scheme more efficient, QoN-ASW calculates the number of message copies to be forwarded based on the difference of the quality of node, which mainly includes two positive factors, that is, the message handling capacity and the improved delivery predictability.

Definition 4: Quality of Node (QoN). The QoN of node v_i, say QoN_i, is the weighted sum of the message handling capacity utility U_{MHC} and the improved delivery predictability utility U_P. QoN_i is obtained by:

$$QoN_i = \eta \cdot U_{MHC} + (1 - \eta) \cdot U_P \tag{7}$$

where U_{MHC} and U_P denotes the relative utility of node v_i for delivering a message to destination node v_d compared to node v_j, given by (8) and (9), and η is a tunable parameter, which can be used to adjust the relative importance of the two utility values.

$$U_{MHC} = \frac{MHC_i^k}{MHC_i^k + MHC_j^k} \tag{8}$$

$$U_P = \frac{P_{(i,d)}}{P_{(i,d)} + P_{(j,d)}} \tag{9}$$

The value of QoN_i can be used to quality the ability of node v_i to successfully forward messages. We can distribute more number of message copies to the node with higher QoN.

IV. AN ADAPTIVE SPRAY AND WAIT ROUTING ALGORITHM BASED ON QUALITY OF NODE

In this section, we present a routing algorithm called adaptive spray and wait routing algorithm based on quality of node (QoN-ASW).

A. SPRAY SCHEME

In traditional spray scheme, each node simply sprays half of its message copies to the encountered node. In the real world, the network is usually heterogeneous, and the quality between nodes may also differ widely. Therefore, symmetrical spray scheme is inflexible, which may result in long delay and low delivery rate. For example, the quality of the encountered node is poor, which means it is less likely to successfully forward message copies to the destination. If we allocate half the number of message copies to the encountered node, half of the opportunities will be wasted.

Given above analysis, we propose an asymmetric spray scheme, named QoN-ASW-Spray, to adaptively allocate the number of message copies between the encountered nodes according to the proportion of the QoN. Whenever node v_i meets node v_j, they should update their QoN firstly, and then exchange QoN with each other. QoN-ASW calculates the number of message copies to be forwarded based on the quality of the receiver node and the sender node. The number of message copies between v_i and v_j is allocated by:

$$\begin{aligned} L_{i_{new}}(m_d) &= \lfloor QoN_i \cdot L_{i_{old}}(m_d) \rfloor \\ L_{j_{new}}(m_d) &= L_{i_{old}}(m_d) - L_{i_{new}}(m_d) \end{aligned} \tag{10}$$

where m_d denotes the message with destination node v_d. $L_{i_{old}}(m_d)$ is the number of copies of m_d that v_i keeps before forwarding, $L_{i_{new}}(m_d)$ is the number of copies of m_d that v_i left after forwarding, and $L_{j_{new}}(m_d)$ is the number of copies of m_d that need to allocate to node v_j.

B. FORWARDING SCHEME

In Spray and Wait routing algorithm, when a node is left with only one copy of a message, it performs direct transmission, in which a relay node moves and waits for the destination to be encountered. However, if the movement range of the node is restricted to a small local area, then the node may never meet the destination node. Besides, direct transmission may lead to long delay, because the node will not hand over the only copy to any other nodes that may have more chances to meet the destination, which will waste some encounter opportunities.

Based on the above analysis, we implemented a forwarding Scheme based on the improved delivery predictability, named QoN-ASW-Wait. When a node is left with only one copy of a message, it converts passive wait scheme into forwarding scheme. The node will forward the only copy to other nodes with higher improved delivery predictability. For example, node v_i is left with only one copy of message m_d, when it encounters node v_j, v_i will forward the only copy of m_d to v_j if $P_{(j,d)} > P_{(i,d)}$ is satisfied. This scheme can decrease the delay time of messages reaching the destination node, and also release the cache of nodes to improve the utilization of resources.

Algorithm 1 Pseudo Code for QoN-ASW

1: **if** node v_i encounters node v_j **then**
2: update MHC_i^k and MHC_j^k as (2)
3: update $P_{(i,j)}$ as (4)
4: **for** $\forall m_l \in v_i$ and $m_l \notin v_j$ **do**
5: $L_{i_{old}}(m_l) = m_l's$ number of copies
6: $v_d = m_l.destination$
7: **if** $v_d == v_j$ **then**
8: v_i forwards message m_l to v_j
9: v_i removes message m_l from $v_i's$ buffer
10: **else if** $L_{i_{old}}(m_l) > 1$ **then**
11: calculate QoN_i and QoN_j as (7)
12: $L_{i_{new}}(m_l) = \lfloor QoN_i \cdot L_{i_{old}}(m_l) \rfloor$
13: $L_{j_{new}}(m_l) = L_{i_{old}}(m_l) - L_{i_{new}}(m_l)$
14: **if** $L_{j_{new}}(m_l) > 0$ **then**
15: v_i forwards $L_{j_{new}}(m_l)$ copies of m_l to v_j
16: v_i changes $m_l's$ number of copies to $L_{i_{new}}(m_l)$
17: **end if**
18: **else if** $L_{i_{old}}(m_l) == 1$ **then**
19: **if** $P_{(j,l)} > P_{(i,l)}$ **then**
20: v_i forwards message m_l to v_j
21: v_i removes message m_l from $v_i's$ buffer
22: **end if**
23: **end if**
24: **end for**
25: **end if**

C. ALGORITHM DESCRIPTION

The details of QoN-ASW algorithm are outlined in Algorithm 1. When node v_i encounters node v_j, the MHC_i^k, MHC_j^k, and $P_{(i,j)}$ should be updated firstly. Then we traverse the set of messages that belong to v_i but not belong to v_j. For each message m_l in the set, if v_j is the destination of m_l, v_i directly forwards m_l to v_j and removes m_l from its buffer. Otherwise, if the number of copies of m_l that v_i keeps is greater than 1, we should allocate the number of copies of m_l between v_i and v_j according to (10), if $L_{j_{new}}(m_l) > 0$, v_i should forward $L_{j_{new}}(m_l)$ copies of m_l to v_j and change the number of copies of m_l that itself carries to $L_{i_{new}}(m_l)$; if the number of copies of m_l that v_i keeps is equal to 1, by comparing $P_{(j,l)}$ and $P_{(i,l)}$, a decision is then made on whether to forward m_l to v_j, if the improved delivery predictability of v_j is larger than that of v_i, m_l will be forwarded from v_i to v_j.

The main time consumption of the algorithm is on the step 5 to step 23, the time complexity of the algorithm is O(M). M represents the number of messages that node v_i holds. Suppose that the total number of nodes in the network is N. Since the algorithm needs to collect the quality of node information, the space complexity of the algorithm is O(N).

V. SIMULATION

A. SIMULATION SETUP

The ONE (Opportunistic Network Environment) [42] simulation of 1.4.1 version is used to create the scenario in this paper. In our simulation scenario, the real Infocom06 datasets [43] gathered by the Haggle Project is used, since it is one of the most recognized datasets at present. In Infocom06, except for 20 stationary devices, the mobile devices were distributed to 78 students attending the Infocom student workshop. Participants belong to different communities (depending on their country of origin, research topic, etc.). With reference to [14], the value of α is set as 0.15, and the main parameters are listed in Table 3.

TABLE 3. Parameters of simulation.

Parameters	Values
Simulation time (hours)	40
Message transmission speed (Mbps)	250
Transmission range (meters)	10
Buffer size (Mbytes)	5
Number of copies of a message	6
TTL (minutes)	300
Message size (Kbytes)	300-500

B. EVALUATION METRICS

The three evaluation metrics used in this work are [20]: delivery rate, average delay, and overhead.

The delivery rate indicates the percentage of messages delivered to the destination, which is obtained by:

$$delivery\ rate = \frac{N_{delivered}}{N_{created}} \tag{11}$$

where $N_{delivered}$ denotes the number of messages delivered to the destination, and $N_{created}$ denotes the number of messages created by the source in the network.

The average delay indicates the average time it takes for messages to be delivered from the source to the destination, which is obtained by:

$$average\ delay = \frac{\sum_{i=1}^{N_{delivered}} delay_i}{N_{delivered}} \tag{12}$$

where $delay_i$ denotes the total time it takes for message m_i to be delivered from the source to the destination.

The overhead denotes the ratio of redundant relay forwarding times to the number of messages delivered to the destination, which is obtained by:

$$overhead = \frac{N_{relayed} - N_{delivered}}{N_{delivered}} \tag{13}$$

where $N_{relayed}$ denotes the total number of forwarding times of all messages in the network.

C. PARAMETERS OPTIMIZATION

The adjustable parameter η impacts the performance of the QoN-ASW-Spray. We have done a lot of experiments by using different value of η. From Fig. 2, we can observe that when η is 0.7, the algorithm achieves better performance than the other. Therefore, we use it for the following simulations.

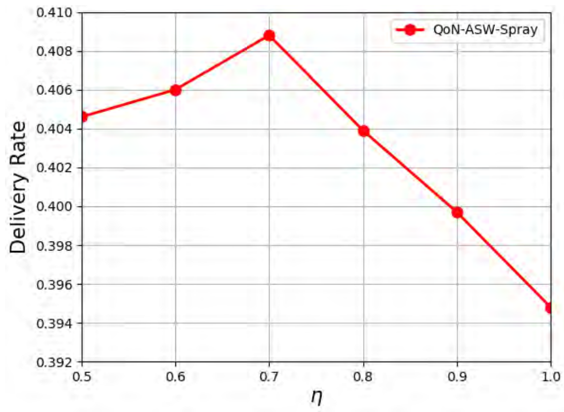


FIGURE 2. Comparison among different η.

β, P_{init}, and ξ of Prophet are adjustable parameters, which can be set according to the specific scene. In the original document of Prophet [12], $\beta = 0.25$, $P_{init} = 0.75$, $\xi = 0.98$. We have done a lot of experiments by using different value of β, P_{init}, and ξ. We find out the optimization parameters as shown in Table 4. Just as Table 4, when $\beta = 0.1$, $P_{init} = 0.7$, $\xi = 0.8$, the overall performance of Prophet is better than that of $\beta = 0.25$, $P_{init} = 0.75$, $\xi = 0.98$.

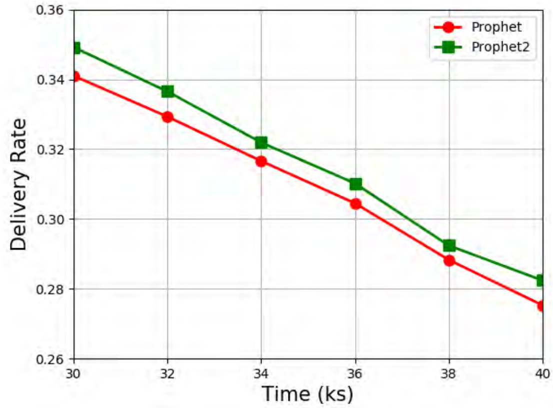
TABLE 4. Optimization parameters setting.

Metrics	$\beta = 0.1, P_{init} = 0.7, \xi = 0.8$	$\beta = 0.25, P_{init} = 0.75, \xi = 0.98$
Delivery rate	0.4116	0.3464
Average delay (s)	5720.2175	5821.7162
Overhead	1352.5196	1578.0891

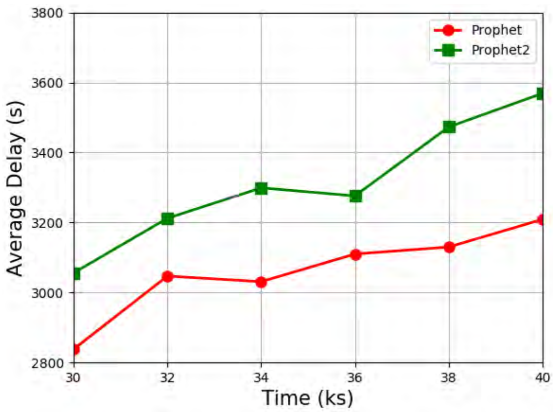
D. DEMONSTRATING THE EFFICIENCY OF CONNECTION STRENGTH

In order to demonstrate the efficiency of integrating the connection strength into delivery predictability for improving the routing performance, we compare the performance of Prophet and Prophet2 which adopts the improved delivery predictability to achieve effective selection of the next relay nodes under different simulation time. The simulation results are shown in Fig. 3.

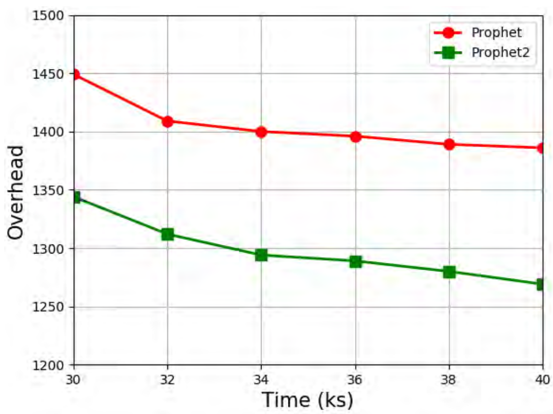
Regarding the results in Fig. 3(a), the delivery rate of Prophet2 is about 2.03% larger on average than that of Prophet. This is because Prophet2 integrates connection strength into delivery predictability, thus making the message copies be forwarded to a more appropriate node. The result in Fig. 3(b) shows that Prophet2 has a longer average delay. The reason is that, compared with Prophet, the improved delivery predictability of Prophet2 does not increase rapidly with the encounter frequency, thus limiting the forwarding of message copies. From Fig. 3(c), we observe that the overhead ratio of Prophet2 is lower than Prophet by 7.61%. The main reason is that Prophet2 uses encounter frequency and connection strength to predict the probability of delivering a message



(a)

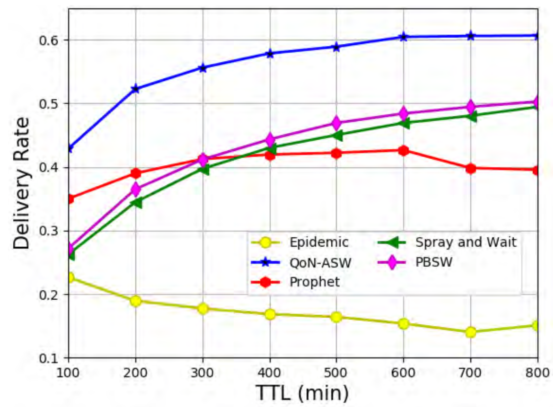


(b)

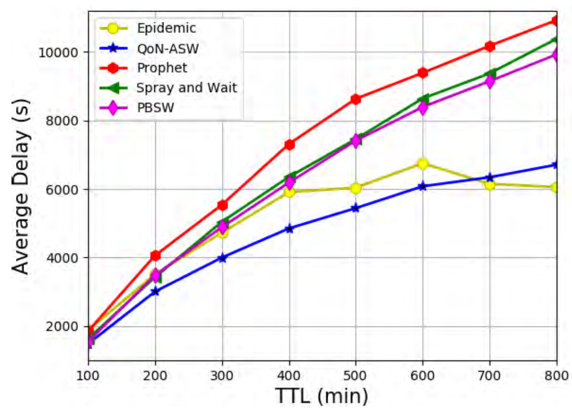


(c)

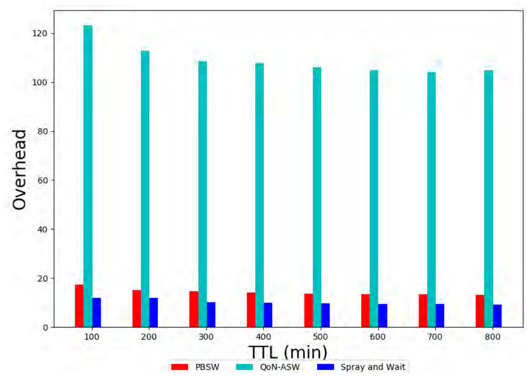
FIGURE 3. Comparison of Prophet and Prophet2.



(a)



(b)



(c)

FIGURE 4. Comparison of various algorithms under different TTL.

to the destination, thereby reducing the waste of resources due to the interruption of transmission. The simulation results show that Prophet2 performs well in both delivery rate and overhead. Thus we can conclude that connection strength is a key factor for successful delivery.

E. COMPARISON WITH OTHER METHODS

In this subsection, we compare the performance of QoN-ASW with the other four routing algorithms, including Prophet, Epidemic, Spray and Wait, and PBSW [15] by varying TTL, buffer size, simulation time, and message generation interval. Moreover, the overhead of Epidemic and Prophet have a magnitude of 10^3 according to experiment result. Therefore, we do not draw the overhead of Epidemic and Prophet in figure to highlight the difference between other algorithms. The extremely high level of the overhead of

Epidemic and Prophet reflect the notion that when they are used to perform simulation in a more realistic scenario, they will have a high overhead [17] .

1) PERFORMANCE EVALUATION BY VARYING TTL

Fig. 4(a) shows the delivery rate. The delivery rate of QoN-ASW, PBSW, and Spray and Wait tends to increase as TTL increases. Because message copies stay longer in the buffer when TTL increase, then nodes carrying messages are more likely to meet the destination. The delivery rate of Epidemic tends to decrease as the TTL increases, since flooding strategy causes the network to be more congested as TTL increases, and then the messages will be discarded. When TTL is greater than 600 minutes, the delivery rate of Prophet shows a decreasing trend, because the resource that Prophet demands for exceed the available capacity of the resource, which may cause network congestion. Therefore, we can infer that a routing algorithm that strictly controls the number of copies is needed when the TTL is large. Simulation results show that the delivery rate of QoN-ASW is improved by 37.29% and 32.56% compared with PBSW and Spray and Wait.

Fig. 4(b) shows the average delay. In general, the average delay increases as the TTL increases, because many messages that cannot be successfully delivered due to short TTL can be successfully delivered when TTL increases. When the TTL is greater than 600 minutes, the average delay of Epidemic shows a downward trend. Because flooding strategy algorithm leads to overflow when TTL is too large, messages that take a long time to be successfully delivered to the destination will be discarded. Besides, the advantage of QoN-ASW algorithm increases with the increase of TTL, which proves the validity of our scheme. Simulation results show that the average delay of QoN-ASW is reduced by 21.85% and 23.72% compared with PBSW and Spray and Wait.

Fig. 4(c) presents the overhead. It is obvious that QoN-ASW has a slightly higher overhead than Spray and Wait, which can be attributed to the forwarding scheme implemented in wait phase to achieve a higher delivery rate. Simulations results show that the overhead of QoN-ASW decreased by 92.08% compared with Prophet.

2) PERFORMANCE EVALUATION BY VARYING BUFFER SIZE

Fig. 5(a) shows the impact of increasing the buffer size on the delivery rate. Just as the advantages of QoN-ASW mentioned above, the delivery rate of QoN-ASW is the highest among all these algorithms. Moreover, we observe that the delivery rate of Epidemic and Prophet increase rapidly with the increase of buffer size, which indicates that they are adapt to the scenario with sufficient resources. And it also can be seen that the delivery rate of PBSW is slightly higher than Spray and Wait, which can be attributed to its replica distribution scheme. Simulation results show that the delivery rate of QoN-ASW is increased by 45.36% and 42.21% compared with PBSW and Spray and Wait.

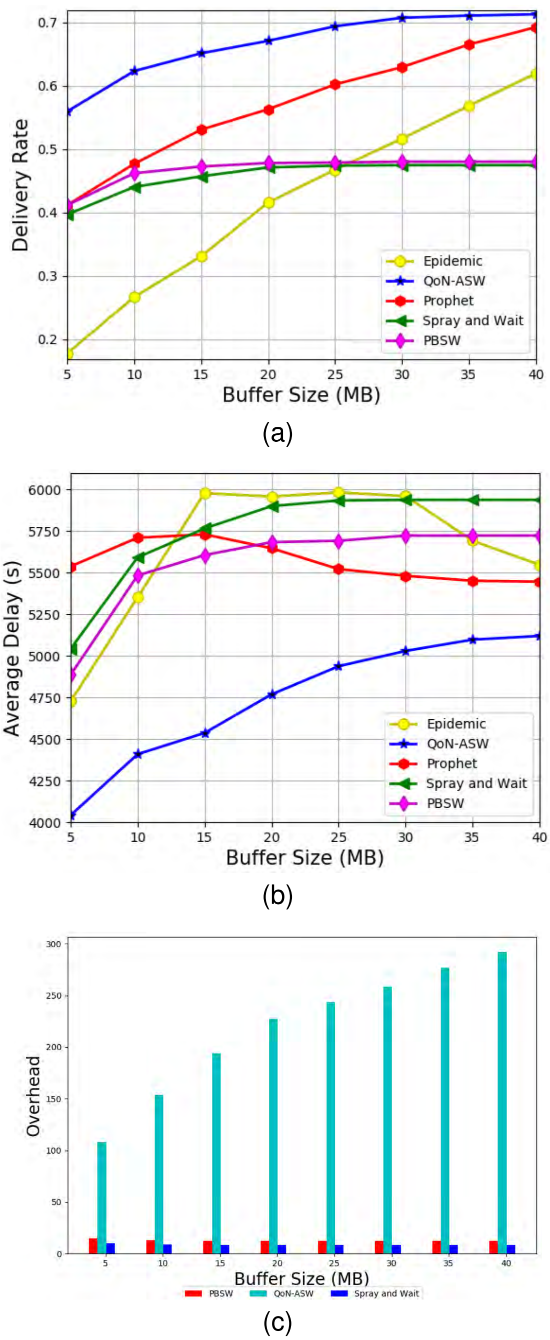


FIGURE 5. Comparison of various algorithms under different buffer size.

Fig. 5(b) presents the average delay. We can observe that QoN-ASW has the lowest average delay. Therefore, an adaptive replica distribution scheme should be considered and a reasonable forwarding scheme is also needed. When the buffer size is small, the average delay of each routing algorithms has an obvious growth trend with the increase of buffer size. Because nodes can carry more number of message copies, the characteristic of intermittent connection leads to messages cannot be forwarded to the destination in time. However, when the buffer size reaches a certain level, the average delay of Prophet and Epidemic present a downward trend, which suggests that we need to control the number of message copies in cache-constrained scenarios. Simulation results show that the average delay of QoN-ASW is reduced by 14.85% and 17.68% compared with PBSW and Spray and Wait.

Fig. 5(c) shows a graph of overhead. We note that Spray and Wait and PBSW have an advantage than other algorithms. Because Spray and Wait and PBSW do not do anything else, just passively wait for the destination to be encountered. Simulations results show that the overhead of QoN-ASW is able to reduce 73.41% on average compared with Prophet.

3) PERFORMANCE EVALUATION BY VARYING SIMULATION TIME

From Fig. 6(a) for delivery rate, we observe that QoN-ASW outperforms all the other routing algorithms. The result can be attributed to two reasons. Firstly, the message copies are sprayed based on the difference of the quality of node to reduce useless replication. Secondly, a forwarding scheme is implemented to make full use of encounter opportunities. It is also clear to see that Prophet achieves better performance than Epidemic, because Prophet creates much fewer copies for each message than Epidemic, which makes more efficient utilization of the network resources. PBSW achieves better performance compared with Spray and Wait because of its reasonable spray scheme. And we observe that the delivery rate of Epidemic is the lowest among all these algorithms, this is because the flooding-based routing algorithm will lead to network congestion when the simulation time is relatively long. Simulation results show that the delivery rate of QoN-ASW is improved by 32.35% and 27.75% compared with PBSW and Spray and Wait.

Fig. 6(b) shows the average delay. We observe that QoN-ASW exhibits the lowest average delay. The result also can be attributed to two factors. Firstly, more number of message copies are distributed to the nodes with higher QoN value, thereby making message copies be forwarded efficiently. Secondly, the forwarding scheme makes message copies be forwarded to the nodes with higher improved delivery predictability, which enables message copies to be successfully delivered to the destination node as soon as possible. Note that PBSW outperforms Spray and Wait in terms of average delay, because PBSW allocates more number of message copies to the nodes with higher delivery predictability. Simulation results show that the average delay of QoN-ASW is reduced by 24.77% and 26.95% compared with PBSW and Spray and Wait.

As shown in Fig. 6(c), the overhead of QoN-ASW is slightly higher than Spray and Wait and PBSW. The result can be explained by the wait scheme, in which allows nodes to forward messages is necessary to achieve effective delivery rate. However, it is worthwhile improving the delivery rate and reducing the average delay with the cost of an acceptable network overhead. Moreover, QoN-ASW outperforms Prophet in terms of overhead, the main reason for this significant improvement is that QoN-ASW limits the maximum number of copies of a message. Simulations results show that the overhead of QoN-ASW decreased by 92.23% compared with Prophet.

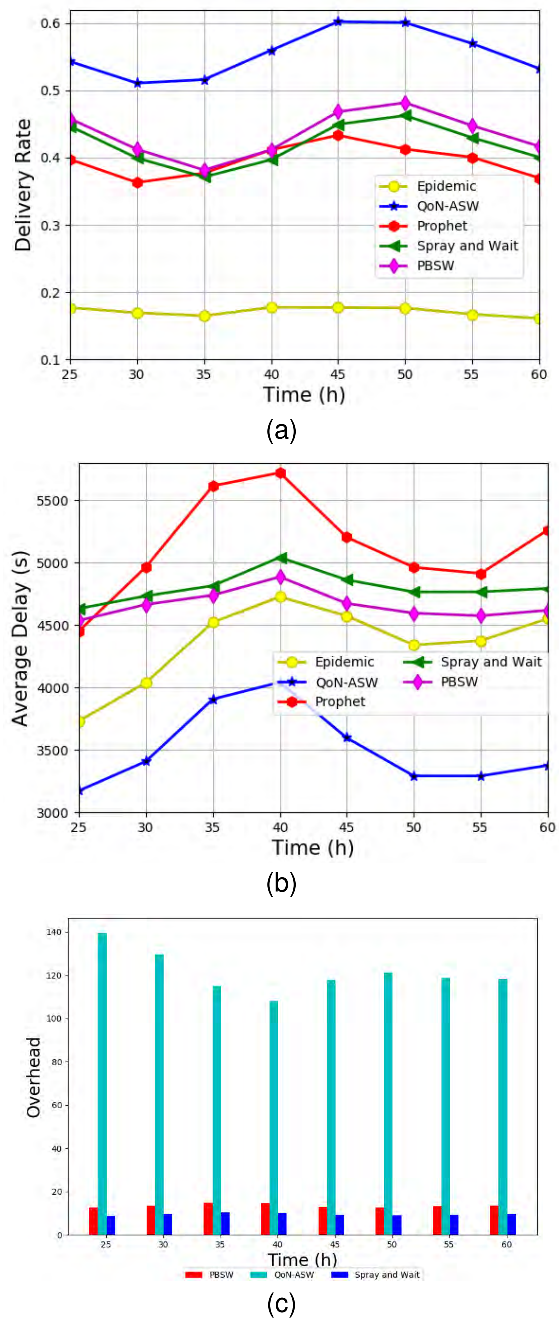


FIGURE 6. **Comparison of various algorithms under different simulation time.**

4) PERFORMANCE EVALUATION BY VARYING MESSAGE GENERATION INTERVAL

Fig. 7(a) shows the delivery rate under different message generation interval. Increasing the message generation interval reduces the number of message copies stored in the

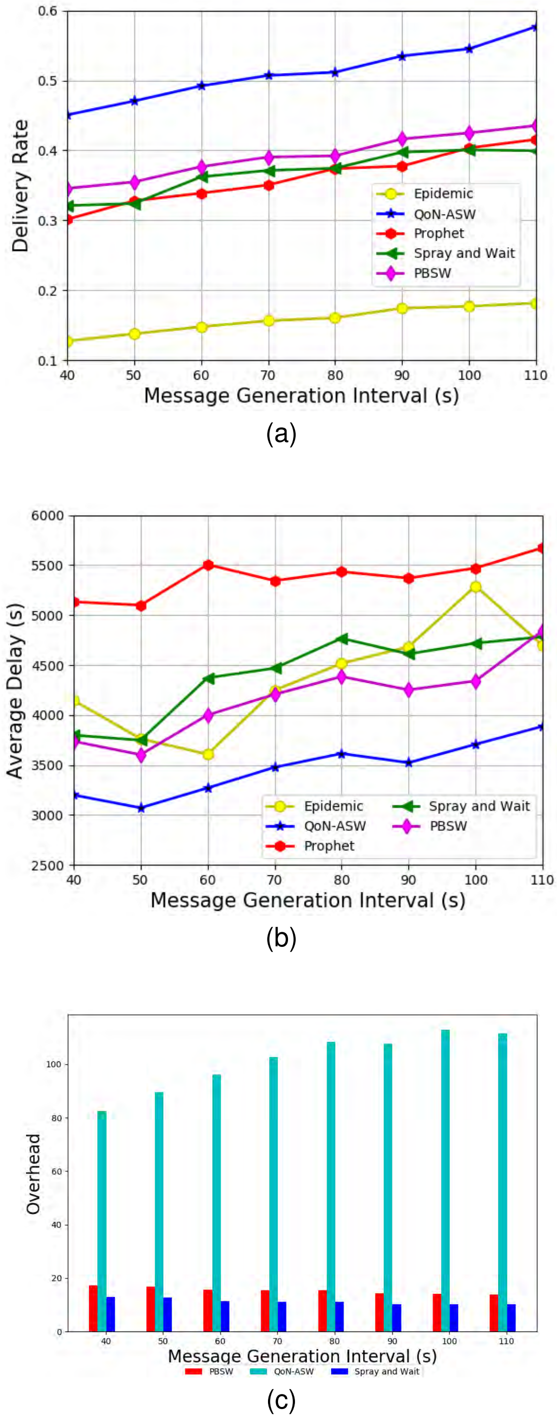


FIGURE 7. Comparison of various algorithms under different message generation interval.

buffer, and then the more available resources can be achieved, so the delivery rate increases as message generation interval increases. Simulation results show that the delivery rate of QoN-ASW is about 38.70% and 30.40% larger on average than that of PBSW and Spray and Wait, which demonstrates the effectiveness of our scheme once again.

Fig. 7(b) shows the average delay. In general, the average delay shows a growth trend with the increase of message generation interval. Because the available cache resources increases with the increase of the message generation interval, then many message copies that take a long time to be successfully delivered will not be discarded due to network congestion. And we observe that QoN-ASW significantly outperforms other algorithms, since it takes the quality of node into account and the forwarding scheme can optimize the average delay. Moreover, the average delay of PBSW is much lower than Spray and Wait due to the smartly replica distribution scheme. Simulation results show that the average delay of QoN-ASW is decreased by 16.73% and 21.15% compared with PBSW and Spray and Wait.

In Fig. 7(c), it is obvious that QoN-ASW has a slightly higher overhead than Spray and Wait, this can be attributed to the forwarding scheme implemented in wait phase. Moreover, simulation results show that the overhead of QoN-ASW is able to reduce 91.03% compared with Prophet. The main reason for this significant improvement is that QoN-ASW controls the maximum number of copies of a message, and the asymmetric spray scheme restricts useless replication.

Based on the above, we observe that QoN-ASW can significantly improve the delivery rate and reduce the average delay while achieving a relatively low overhead. We made algorithm improvement both in spray and wait period, improvement in these two periods can improve the overall performance of the algorithm. From the simulation results, we observe that the improvement in wait period plays more important in the performance improvement, which can be explained by the forwarding scheme implemented in wait phase. The forwarding scheme takes advantage of encounter opportunities arising from node movement, which not only increases the chances of successful transmission, but also releases the buffer space and improves resource utilization.

VI. CONCLUSION

In this paper, we introduce two concepts, that is, message handling capacity (MHC) and connection strength (CS). The MHC is used to reflect the ability of a node to forward messages. The CS is used to determine the connection strength between nodes, and then we integrated CS into delivery predictability adopted by Prophet to achieve better routing performance. Based on these two concepts, the metric called quality of node (QoN) is defined. Finally, we present an adaptive spray and wait routing algorithm based on quality of node (QoN-ASW): in spray phase, we propose an asymmetric replica distribution strategy QoN-ASW-Spray to avoid the blindness of replica distribution; in wait phase, we implement a forwarding scheme QoN-ASW-Forward to make the algorithm more flexible, so that it will take full advantage of encounter opportunities. Applying these techniques, simulation results show that QoN-ASW can significantly improve the delivery rate and reduce the average delay while achieving a relatively low overhead, which demonstrates the effectiveness of our scheme. The overhead of QoN-ASW

is slightly higher than that of Spray and Wait, however, it is worthwhile improving the delivery rate and reducing the average delay with the cost of an acceptable network overhead.

There are some ways in which this work should be continued. Firstly, a natural avenue for future research should consider proposing an efficient buffer management scheme to deeply improve the delivery rate. Secondly, we will dynamically change the maximum number of message copies to automatically adapt to the changing network conditions.

REFERENCES

[1] A. V. Vasilakos, Y. Zhang, and T. Spyropoulos, *Delay Tolerant Networks: Protocols and Applications*. Boca Raton, FL, USA: CRC Press, 2016.

[2] J. J. S. Huang, S. J. H. Yang, Y. M.-Huang, and I. Y. T. Hsiao, "Social learning networks: Build mobile learning networks based on collaborative services," *Educ. Technol. Soc.*, vol. 13, no. 3, pp. 78–92, 2010.

[3] M. Chen, Y. Hao, K. Lin, Z. Yuan, and L. Hu, "Label-less learning for traffic control in an edge network," *IEEE Netw.*, vol. 32, no. 6, pp. 8–14, Nov. 2018.

[4] B. Liu *et al.*, "Infrastructure-assisted message dissemination for supporting heterogeneous driving patterns," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2865–2876, Oct. 2017.

[5] L. Wu, Y. Xia, Z. Wang, and H. Wang, "Be stable and fair: Robust data scheduling for vehicular networks," *IEEE Access*, vol. 6, pp. 32839–32849, 2018.

[6] H. Zhou, H. Wang, X. Chen, X. Li, and S. Xu, "Data offloading techniques through vehicular ad hoc networks: A survey," *IEEE Access*, vol. 6, pp. 65250–65259, 2018.

[7] C.-M. Huang, Y.-F. Chen, S. Xu, and H. Zhou, "The vehicular social network (VSN)-based sharing of downloaded geo data using the credit-based clustering scheme," *IEEE Access*, vol. 6, pp. 58254–58271, 2018.

[8] C. Zhu, V. C. M. Leung, J. J. P. C. Rodrigues, L. Shu, L. Wang, and H. Zhou, "Social sensor cloud: Framework, greenness, issues, and outlook," *IEEE Netw.*, vol. 32, no. 5, pp. 100–105, Sep. 2018.

[9] L. Zhang, X. Zhou, and J. Guo, "Noncooperative dynamic routing with bandwidth constraint in intermittently connected deep space information networks under scheduled contacts," *Wireless Pers. Commun.*, vol. 68, no. 4, pp. 1255–1285, 2013.

[10] Y. Kuronuma, H. Suzuki, and A. Koyama, "An adaptive DTN routing protocol considering replication state," in *Proc. Int. Conf. Adv. Inf. Netw. Appl. Workshops (AINA)*, Taipei, Taiwan, Mar. 2017, pp. 421–426.

[11] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Duke Univ., Durham, NC, USA, Tech. Rep. CS-200006, 2000.

[12] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," in *Proc. Int. Workshop Service Assurance Partial Intermittent Resour. (SAPIR)*, Fortaleza, Brazil, 2004, pp. 239–254.

[13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM Workshop Delay-Tolerant Netw.*, Philadelphia, PA, USA, 2005, pp. 252–259.

[14] G. Wang, B. Wang, and Y. Gao, "Dynamic spray and wait routing algorithm with quality of node in delay tolerant network," in *Proc. Int. Conf. Commun. Mobile Comput.*, Apr. 2010, pp. 452–456.

[15] E. H. Kim, J. C. Nam, J. I. Choi, and Y. Z. Cho, "Probability-based spray and wait protocol in delay tolerant networks," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Phuket, Thailand, Feb. 2014, pp. 412–416.

[16] X. Du, W. Lai, J. Zhang, and H. Wang, "A motion and counting based-routing algorithm for delay tolerant network," *Int. J. Commun., Netw. Syst. Sci.*, vol. 10, no. 5, pp. 14–23, 2017.

[17] J. Guan, Q. Chu, and I. You, "The social relationship based adaptive multi-spray-and-wait routing algorithm for disruption tolerant network," *Mobile Inf. Syst.*, vol. 2017, Jan. 2017, Art. no. 1819495.

[18] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 477–486, Aug. 2002.

[19] Agussalim and M. Tsuru, "Spray router with node location dependent remaining-TTL message scheduling in DTNs," *J. Inf. Process.*, vol. 24, no. 4, pp. 647–659, 2016.

[20] J. G. Filho, A. Patel, B. L. A. Batista, and J. Celestino, Jr., "A systematic technical survey of DTN and VDTN routing protocols," *Comput. Standards Interfaces*, vol. 48, pp. 139–159, Nov. 2016.

[21] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "Replication routing in DTNs: A resource allocation approach," *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 596–609, Apr. 2010.

[22] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, White Plains, NY, USA, Mar. 2007, pp. 79–85.

[23] J. Cui, L. Ma, K. Gao, L. Wu, and Y. Yang, "An adaptive load balancing strategy of application layer multicasts based on score of customer satisfaction," *J. Internet Technol.*, vol. 19, no. 5, pp. 1425–1434, 2018.

[24] J. Cui, N. Xiong, J. H. Park, K. Jia, and L. Wu, "A novel and efficient source-path discovery and maintenance method for application layer multicast," *Comput. Elect. Eng.*, vol. 39, no. 1, pp. 67–75, 2013.

[25] Y. Chang, X. Jia, and J. Cui, "Throughput improvement for delay-sensitive video broadcast in wireless mesh networks," *Comput. Elect. Eng.*, vol. 48, pp. 214–225, Nov. 2015.

[26] Y. Chang, Q. Liu, and X. Jia, "A data rate and concurrency balanced approach for broadcast in wireless mesh networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3556–3566, Jul. 2014.

[27] J. Liu, M. Tang, and G. Yu, "Adaptive spray and wait routing based on relay-probability of node in DTN," in *Proc. Int. Conf. Comput. Sci. Service Syst. (CSSS)*, Shenyang, China, Aug. 2012, pp. 1138–1141.

[28] G. Wang, H. Zhang, D. Pan, and J. Tao, "A routing scheme based on spray-and-wait and connecting-capability of node," in *Proc. Int. Conf. Prog. Inform. Comput.*, Nanjing, China, Dec. 2016, pp. 577–580.

[29] G. Wang, J. Wang, and B. Wang, "An adaptive spray and wait routing algorithm based on capability of node in DTN," *J. Inf. Comput. Sci.*, vol. 11, no. 6, pp. 1975–1982, 2014.

[30] M. N. Sadat and M. T. Mohiuddin, "A neighborhood contact history based spraying heuristic for delay tolerant networks," in *Proc. Int. Conf. Inform., Electron. Vis. (ICIEV)*, Dhaka, Bangladesh, May 2014, pp. 1–5.

[31] T. C. Tsai and Y.-M. Chen, "Popularity spray and utility-based forwarding scheme with message priority scheduling in delay tolerant networks," in *Proc. ACM Int. Symp. Mobility Manage. Wireless Access*, Barcelona, Spain, 2013, pp. 25–28.

[32] M. Matis, L. Doboš, and J. Papaj, "An enhanced hybrid social based routing algorithm for MANET-DTN," *Mobile Inf. Syst.*, vol. 2016, Mar. 2016, Art. no. 4803242.

[33] D. Pan, M. Lin, L. Chen, and J. Sun, "An improved spray and wait with probability choice routing for opportunistic networks," *J. Netw.*, vol. 7, no. 9, pp. 1486–1492, 2012.

[34] A. Mei, G. Morabito, P. Santi, and J. Stefa, "Social-aware stateless routingin pocket switched networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 1, pp. 252–261, Jan. 2015.

[35] M. Ruan, X. Chen, and H. Zhou, "Centrality prediction based on k-order Markov chain in mobile social networks," *Peer-Peer Netw. Appl.*, to be published.

[36] L. Wu, J. Wang, K.-K. R. Choo, and D. He, "Secure key agreement and key protection for mobile device user authentication," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 319–330, Feb. 2019.

[37] S. Faye, N. Louveton, G. Gheorghe, and T. Engel, "A two-level approach to characterizing human activities from wearable sensor data," *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl.*, vol. 7, no. 3, pp. 1–21, 2016.

[38] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE rap: Social-based forwarding in delay-tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 11, pp. 1576–1589, Nov. 2011.

[39] K. Wang, H. Guo, M. Wu, and Z. Yang, "An efficient routing algorithm based on social awareness in DTNs," in *Proc. Veh. Technol. Conf. (VTC)*, Dresden, Germany, Jun. 2013, pp. 1–5.

[40] E. Bulut and B. K. Szymanski, "Exploiting friendship relations for efficient routing in mobile social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 12, pp. 2254–2265, Dec. 2012.

[41] Y. Zhang and J. Zhao, "Social network analysis on data diffusion in delay tolerant networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, New Orleans, LA, USA, 2009, pp. 345–346.

[42] A. Keränen, J. Ott, and T. Kärkkäinen, "The one simulator for DTN protocol evaluation," in *Proc. Int. Conf. Simulation TOOLS Techn.*, Rome, Italy, 2009, p. 55.

[43] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 606–620, Jun. 2007.

JIANQUN CUI received the Ph.D. degree in computer science from Wuhan University, China. She is currently a Professor with the Computer School, Central China Normal University, and a Visiting Scholar with the Department of Computer Science and Engineering, SUNY at Buffalo, USA. Her research interests include opportunistic networks, the Internet of Things, mobile networks, and application layer multicast.

SHUQIN CAO is currently pursuing the M.S. degree in computer science with Central China Normal University. Her current research interests include delay tolerant networks, social networks, and wireless communication.

YANAN CHANG received the Ph.D. degree in computer science by the Joint Ph.D. Program from Wuhan University and the City University of Hong Kong. She is currently an Assistant Professor with the Computer School, Central China Normal University, and a Visiting Scholar with the Tandon School of Engineering, New York University. Her research interests include wireless mesh networks, delay tolerant networks, and social networks.

LIBING WU received the Ph.D. degree in computer science from Wuhan University, where he is currently a Professor with the Computer School. His current research interests include network communication, grid computing, and the Internet of Things.

DAN LIU is currently pursuing the M.S. degree in computer science and technology with the Wuhan University of Technology. His current research interests include delay tolerant networks, wireless sensor networks, and social networks.

YI YANG received the Ph.D. degree in computer science and engineering from Pennsylvania State University, in 2010. She is currently an Associate Professor with Fontbonne University, St Louis, MO, USA, where she is also the Director of BS in cyber security. Her research interests include computer/network security and network management.