Software-Defined Networking-Assisted Content Delivery at Edge of Mobile Social Networks

Fuliang Li, *Member, IEEE*, Yaoguang Lu, Xingwei Wang, Yuanguo Bi, *Member, IEEE*, Tian Pan, Yuchao Zhang, *Member, IEEE*, Weichao Li, and Yi Wang, *Member, IEEE*

***Abstract*—With the explosive growth of mobile devices at the edge of mobile social networks (MSNs), the amount of the content that needs to be transmitted is exploded. Traditional content delivery mechanisms leverage only local information to make routing decisions, which results in both high latency and low delivery rate. Software-defined networking (SDN) is a novel network paradigm, the design philosophy of which could be applied to MSN for improving the content delivery performance. In this article, the centralized control thought of SDN is introduced into MSN to efficiently process social information. The classical routing algorithm of BubbleRap is improved from the perspective of network density, which is the basis of designing the sparse and dense routing mechanisms for MSNs. In addition, flexibly switching between these two routing mechanisms is implemented by a discriminating scheme, achieving efficient yet adaptive routing. The experimental results show that the delivery ratio of sparse routing is up to 83%, and the dense routing could reach up to 93%.**

***Index Terms*—Content delivery, dense routing, mobile social networks (MSNs), software-defined networking (SDN), sparse routing.**

Manuscript received May 11, 2020; revised June 24, 2020; accepted July 11, 2020. Date of publication July 16, 2020; date of current version September 15, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1702000; in part by the National Natural Science Foundation of China under Grant 61702049 and Grant 61872420; in part by the Project of PCL Future Greater-Bay Area Network Facilities for Large-Scale Experiments and Applications under Grant LZC0019; in part by the Guangdong Key Research and Development Program under Grant 2019B121204009; in part by the Fundamental Research Funds for the Central Universities under Grant N2016005; and in part by the CERNET Innovation Project under Grant NGII20180101. *(Corresponding authors: Fuliang Li; Yaoguang Lu.)*

Fuliang Li is with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China, and also with the Research Center for Network Communication, Peng Cheng Laboratory, Shenzhen 518052, China (e-mail: lifuliang@cse.neu.edu.cn).

Yaoguang Lu, Xingwei Wang, and Yuanguo Bi are with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: 1910620@stu.neu.edu.cn; wangxw@mail.neu.edu.cn; biyuanguo@mail.neu.edu.cn).

Tian Pan is with the Department of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: platinum127@gmail.com).

Yuchao Zhang is with the School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: yczhang@bupt.edu.cn).

Weichao Li and Yi Wang are with the Institute of Future Networks, Southern University of Science and Technology, Shenzhen 518055, China, and also with the Research Center for Network Communication, Peng Cheng Laboratory, Shenzhen 518052, China (e-mail: liwc@sustech.edu.cn; wy@ieee.org).

Digital Object Identifier 10.1109/JIOT.2020.3009854

I. Introduction

With the development of wireless access technologies, as well as the wide popularity of mobile devices, mobile users increase sharply. As a result, the traditional PC end users are replaced by mobile-edge users gradually in online social networks (OSNs, e.g., Ins, Facebook, and Twitter). To provide better services for mobile-edge users, mobile social networks (MSNs) have emerged, which leverage the characteristics of social networks in combination with mobile communication networks [1], [2]. In MSNs, users with similar interests communicate with one another through their mobile devices [3]. Due to the explosive growth of mobile devices, the amount of content that needs to be transmitted among mobile users is exploded. Therefore, how to efficiently deliver the massive amounts of mobile content has become one crucial issue of MSNs. However, content delivery in MSNs presents its unique attributes compared to other network paradigms. The following two challenges still need to be addressed.

1) The content in MSNs is delivered based on social relations, including the community and social characteristics. However, fewer existing solutions take full advantage of social characteristics. To improve the content delivery efficiency of MSNs, high-level and global social behavior patterns should be considered to make the routing decisions.
2) The content has grown by orders of magnitude in MSNs. In addition, the inherent mobility of mobile users is obvious due to the portability. Content delivery in MSNs is more difficult to control compared to OSNs, which are running mainly based on traditional networks. A controller is needed to be introduced into MSNs to make the routing mechanism more efficient, as well as to reduce the workload of mobile devices.

Many routing mechanisms have been proposed for MSNs, such as Spring [4], BubbleRap [14], etc. They design the unique social metrics based on the social relationships prevalent in human society [5]–[9]. A node delivers a message to the designated node by passing the message to a relay node with a larger social metric. The social relationship can be a positive social relationship, e.g., friendship. It may also be a negative social relationship, e.g., selfishness. In addition, a network is usually divided into multiple communities. However, these algorithms deliver a large number of messages to the nodes with high social metrics, causing these nodes to be overwhelmed and affecting the overall performance.

With the management requirements of the systems of the Internet of Things [10], [11], software-defined networking (SDN) is applied to enhance system control and improve transmission performance [12], [13]. Given the above, this article proposes the software-defined MSNs (SDMSNs) to improve content delivery performance. We could get two major benefits with the SDN controller: 1) conveniently capturing social information of the mobile-edge devices and 2) effectively making routing decisions through the centralized control. Referring to separating the data plane from the control plane, the social information is processed by the controller, and the processing result is sent to mobile-edge devices, which are only responsible for delivering the content. This could relieve the bottleneck of mobile-edge devices with limited processing capability. The main contributions of this article can be summarized as follows.

1) We apply the design philosophy of SDN to MSNs for improving the content delivery performance. The centralized controller could not only efficiently process social information but also offload the burden from mobile-edge devices with limited resources.
2) Different from the previous studies, the network density is considered as a crucial metric to improve the classical routing algorithm of BubbleRap, which provides the basis for designing the sparse and dense routing mechanisms for MSNs. In addition, we design a discriminating scheme to achieve flexible switching between these two routing mechanisms.
3) We implement and evaluate the proposed routing mechanisms from the aspects of delivery rate, delay, overhead, and buffertime. The experiment results show that the proposed routing mechanisms are feasible and effective. The overhead of the sparse routing mechanism is 18% lower than BubbleRap, and the overhead of the dense routing mechanism is 54% lower than BubbleRap.

The remainder of this article is organized as follows. Section II presents the related work. Section III describes the architecture of SDMSNs and the proposed routing algorithms. Section IV shows the simulation results. Section V concludes the whole article.

II. Related Work

MSN is a delay-tolerant network that contains many mobile-edge nodes with social attributes. It combines the characteristics of mobile communication networks with social networks. However, the unsustainable connection makes the content delivery in MSNs become a challenging issue. Therefore, many routing algorithms have been proposed for MSNs.

BubbleRap is an information forwarding algorithm based on community and intermediary centrality [14]. Since the buffer and the power of mobile devices are limited, high-rank devices are likely to be overwhelmed. Mao *et al.* [15] designed a community-based forwarding (CBF) policy to improve BubbleRap. SimBet explores the self-mediation centrality and similarity to deliver information [16]. SocialCast [17] is a social-aware MSN routing protocol, which adopts a publish/subscribe message model. PrefCast [18] is a preference-aware content delivery protocol for MSNs. In a traditional MSN, most traffic passes through a small subset of the devices with high centrality. This unfairness would exhaust the limited resources of these devices and decrease the robustness of MSNs as a result. Therefore, FairRoute [19] is proposed to guarantee the fairness of the devices. SANE [20] is a routing mechanism for MSNs based on stateless and social cognition. Gupta *et al.* [21] utilized mobile regularity and community to calculate the most reliable content delivery route. Zhang *et al.* [22] applied edge computing to MSNs and proposed a fuzzy reasoning routing and forwarding algorithm. However, the distributed routing protocols and algorithms have some limitations. For example, it is not easy to find a multihop route from the source to the destination because it is complex to coordinate the mobile nodes and to maintain the network topology in a distributed manner.

SDN separates the control plane from the data plane, which makes it easier to add and deploy new functions to the network. Many studies have applied SDN to some kinds of *ad hoc* networks for constructing efficient content delivery routes. Zhu *et al.* [23] proposed an SDN-based routing framework for efficient message propagation in the vehicular *ad hoc* network (VANET) and designed an algorithm for discovering global optimal route using dynamic network density. Dong *et al.* [24] proposed an SDN-based on-demand routing protocol that separates the data forwarding layer from the network control layer to achieve efficient data forwarding in VANET. FSDN [25] is a new VANET architecture combining SDN with fog computing. SRPA [26] is a SDN-based routing protocol for *ad hoc* networks. It migrates routing decisions from nodes to controllers, discovering the shortest path for each pair of nodes with low latency and less control information exchange. Correia *et al.* [27] introduced an SDN controller to vehicular networks in order to improve the content delivery performance. Ghafoor and Koo [28] proposed a cognitive routing protocol for SDN-based vehicular networks to identify stable routes. However, these SDN-based MSN routing mechanisms do not make full use of the holistic view of the controller and ignore the important feature of network density. In this article, we also introduce the centralized thought of SDN to MSNs and consider the network density as a crucial metric to design the routing mechanisms for the SDN-based MSN, making it provide better content delivery services for mobile-edge users.

III. Routing Mechanism

A. Network Model

In MSNs, due to the mobility of mobile devices, stable connections cannot be easily established between each pair of devices, leading to a poor content delivery performance. Therefore, we propose the SDN-based MSN architecture as shown in Fig. 1. With the holistic view of the network, it is convenient to capture and process social information. Then, routing decisions can be wisely made by the centralized controller instead of the mobile-edge devices, which could also reduce the burden on the devices with limited processing abilities. Note that each mobile device needs to maintain a connection to the controller at any time.

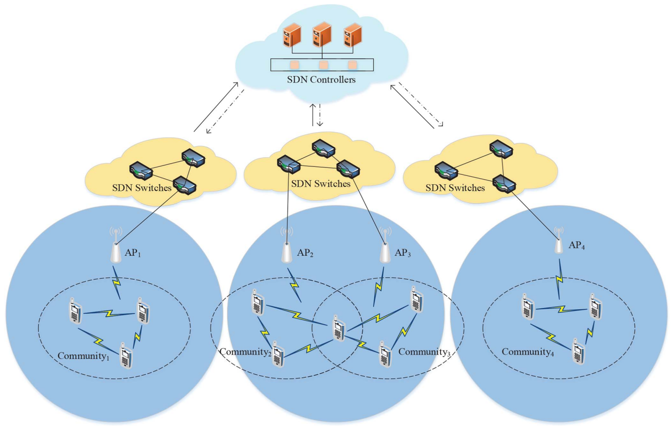


Fig. 1. Architecture of the software-defined MSNs.

The additional round-trip delay from the mobile node to the SDN controller is introduced in the proposed architecture. However, the processing delay of the proposed architecture is roughly equivalent to that of the traditional architecture of MSN, because the processing capacity of the SDN controller is much higher than that of mobile-edge nodes. Benefiting from the powerful processing capability of the SDN controller, the centralized control will not become the bottleneck of content delivery in MSNs. In addition, with the MSN scale and message traffic increasing, an SDN controller cluster could be deployed to handle a large number of messages.

On the basis of the SDN-based MSN architecture, we choose the metric of network density to improve the classical algorithm of BubbleRap. A sparse-mode routing mechanism and a dense-mode routing mechanism are designed for the sparse and dense networks, respectively. In addition, a discriminating scheme is proposed to achieve flexible switching between sparse routing and dense routing. Details about how to improve BubbleRap are described in the following sections.

B. Sparse-Mode Routing Mechanism

The sparse-mode routing mechanism is mainly composed of three parts, including the cluster-based community discovery, the social metric definitions, and the sparse routing algorithm descriptions.

1) Cluster-Based Community Discovery: A mobile node will often appear in the coverage area of several APs. These APs are called common APs for this mobile node. Then, the resident area of the node is determined according to these common APs. Finally, several nodes whose overlapping degree of the resident areas is greater than a certain threshold are divided into the same community. During the period of T, the controller is responsible for counting the APs that the node is connected to. The duration of the connection is also recorded. We denote δ_1 the threshold of connection duration. If the total connection duration between the node and an AP exceeds δ_1, the AP is a common AP of the node. Let B_i be the set of common APs of node i. We use the distance-based clustering algorithm to cluster the points in B_i to generate the resident area of node i. Note that the resident area of the node is a circular area.

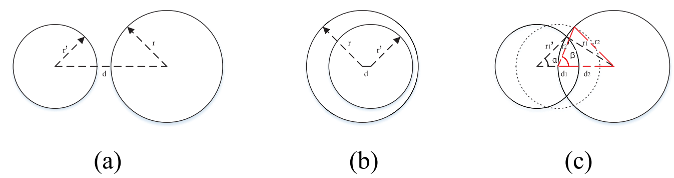


Fig. 2. Relationship of resident area. (a) Disjoint. (b) Inclusiveness. (c) Intersection.

The relationship between the resident areas of two nodes mainly includes disjoint, inclusiveness, and intersection. As shown in Fig. 2(a) and (b), the nodes without intersection do not have the community relationship while the nodes with the inclusion relation are in the same community. As depicted in Fig. 2(c), if the angle is greater than the threshold of δ_2, the nodes are considered to be in the same community.

2) Social Metric Definitions:

a) Circle similarity:

Stable Connection: If the connection time t of nodes a and b is greater than or equal to the threshold of δ_3, the connection between node a and node b is considered to be a stable connection. *Friend Circle:* A collection of nodes that have a stable connection with node u is called a friend circle of node u and it is labeled as $C(u)$.

The ratio of the intersection of $C(u)$ and $C(v)$ to the maximum value of $C(u)$ and $C(v)$ is defined as the circle similarity (CS), which can be expressed as follows, where $C(u)$ and $C(v)$ represent the friend circle of nodes u and v, respectively

$$\mathrm{CS}(u, v) = \frac{|C(u) \cap C(v)|}{\max(|C(u)|, |C(v)|)}. \tag{1}$$

b) Global popularity and in-community popularity:

Encounter Frequent: The encounter frequent is divided into global encounter frequency Fre_g and in-community encounter frequency Fre_i. We denote $\mathrm{Fre}_g(u, v)$ the global encounter frequency of node u to node v, which can be expressed as follows:

$$\mathrm{Fre}_g(u, v) = \frac{f(u, v)}{\max(f(u, k))} \quad \big(k \in NS_g(u)\big). \tag{2}$$

If nodes u' and v' belong to the same community, $\mathrm{Fre}_i(u', v')$ is defined as the in-community encountering frequency of node u' to node v', which can be expressed as follows:

$$\mathrm{Fre}_i\big(u', v'\big) = \frac{f\big(u', v'\big)}{\max(f(u', k'))} \quad \big(k' \in NS_i\big(u'\big)\big) \tag{3}$$

where the function f represents the encounter times of two nodes in a time slice, $NS_g(u)$ is the set of the nodes that node u meets in the whole network, and $NS_i(u')$ is the set of the nodes that node u' meets in the community.

Encounter Stability: The encounter stability between each pair of nodes includes the encounter duration stability Sta_1 and the encounter interval stability Sta_2. In a time slice, supposing nodes u and v encounter ω times in total ($\omega \geq 2$), and the duration of the ith encounter is set as tl_i, and the interval between the ith and the ith + 1 encounter is set as tg_i. The encounter duration stability and the encounter interval stability between node u and node v are defined as (4) and (5), respectively, where S_{t1} and S_{tg} are the standard deviations of the encounter duration and the encounter interval between node u and node v, respectively

$$\mathrm{Sta}_1(u,v) = \frac{1}{\ln(e+S_{tl})} \tag{4}$$

$$\mathrm{Sta}_2(u,v) = \frac{1}{\ln\big(e+S_{tg}\big)}. \tag{5}$$

Thus, the encounter stability between node u and node v is obtained as follows:

$$\mathrm{Sta}(u,v) = \alpha_1 * \mathrm{Sta}_1(u,v) + \alpha_2 * \mathrm{Sta}_2(u,v) \tag{6}$$

where α_1 and α_2 are constants satisfying $0 < \alpha_1, \alpha_2 < 1$ and $\alpha_1 + \alpha_2 = 1$. First, considering the encounter frequency and the encounter stability, we define the global relationship closeness between node u and node v as follows:

$$\mathrm{RC}_g(u,v) = \beta_1 * \mathrm{Fre}_g(u,v) + \beta_2 * \mathrm{Sta}(u,v). \tag{7}$$

Then, the global popularity (GP) of node u is defined as the average global relationship closeness between node u and the other nodes in the network, which can be expressed as follows:

$$\mathrm{GP}(u) = \frac{1}{N_1 - 1}\sum_{i \neq u} \mathrm{RC}_g(u,i) \tag{8}$$

where β_1 and β_2 are constants satisfying $0 < \beta_1, \beta_2 < 1$ and $\beta_1 + \beta_2 = 1$. N_1 is the number of nodes of the entire network. We define the in-community relationship closeness between node u' and node v' as follows:

$$\mathrm{RC}_i\big(u',v'\big) = \gamma_1 * \mathrm{Fre}_i\big(u',v'\big) + \gamma_2 * \mathrm{Sta}\big(u',v'\big). \tag{9}$$

Then, the in-community popularity (IP) of node u' is defined as the average in-community relationship closeness between node u' and the other nodes in the community, which can be expressed as follows, where γ_1, γ_2 are constants satisfying $0 < \gamma_1, \gamma_2 < 1$ and $\gamma_1 + \gamma_2 = 1$. N_2 is the number of nodes of the community

$$\mathrm{IP}\big(u'\big) = \frac{1}{N_2 - 1}\sum_{i \neq u'} \mathrm{RC}_i\big(u',i\big). \tag{10}$$

3) Sparse Routing Algorithm Descriptions: Using BubbleRap, the message will be delivered only when the centrality of the encounter node is higher than that of the source node, which results in a low delivery rate due to fewer mobile nodes in the sparse network. To improve the content delivery rate, the CS is utilized to improve BubbleRap for the sparse network. When the CS of the encounter node and the destination node is higher than that of the source node and the destination node, the encounter node will also be chosen as the relay node to deliver the message. In addition, the global centrality and the local centrality in BubbleRap are replaced by the GP and the IP to further improve the content delivery efficiency.

As shown in Algorithm 1, n_{source}, n_{meet}, and n_{aim} are the source node, the encounter node, and the destination node, respectively. IP(n), CS(n), and GP(n) are the IP, the CS, and the GP of node n, respectively. If the encounter node is the

Algorithm 1 Content Delivery With Sparse-Mode Routing

Require: n_{source}, n_{meet}, n_{aim}.
Ensure: Delivering message or not
1: **if** $n_{meet} = n_{aim}$ **then**
2: n_{source} delivers m to n_{aim};
3: **else if** n_{aim}, n_{meet} and n_{source} are in the same community & $(IP(n_{meet}) > IP(n_{source}) \parallel CS(n_{meet}) > CSn_{source})$ **then**
4: n_{source} delivers m to n_{meet};
5: **else if** n_{aim} and n_{meet} are in the same community **then**
6: n_{source} delivers to m to n_{meet};
7: **else if** $GP(n_{meet}) > GP(n_{source}) \parallel CS(n_{meet}) > CS(n_{source})$ **then**
8: n_{source} delivers m to n_{meet};
9: **end if**

destination node, the source node will deliver the message to the destination node directly (lines 1 and 2). If the destination node, encounter node, and source node are in the same community, and the IP or the CS of the encounter node is greater than that of the source node, the message is delivered to the encounter node (lines 3 and 4). The source node deliveries the message to the encounter node when the destination node and the encounter node are in the same community (lines 5 and 6). If the GP or the CS of the encounter node is greater than that of the source node, the message is delivered to the encounter node (lines 7 and 8).

C. Dense-Mode Routing Mechanism

The dense-mode routing mechanism also consists of three parts, including the SO-based community discovery, the social metric definitions, and the dense routing algorithm descriptions.

1) SO-Based Community Discovery: The cluster-based community discovery algorithm makes full use of the centralized control thought, which is simple and efficient for the sparse MSN. However, there are many nodes in the dense MSN, the calculation will be large if the cluster-based community discovery algorithm is adopted. Therefore, community discovery in the dense network is transformed into an optimization problem, which is solved by sheep optimization (SO), i.e., a swarm intelligence algorithm by simulating sheep behaviors [34], including a sheep leading process, a sheep interaction process, and a shepherd supervision process. We describe the SO-based community discovery from the following three aspects.

a) Optimization objective: In this article, we use the expanded modularity to evaluate the community division as shown in the following [35]:

$$\mathrm{EQ} = \frac{1}{2m}\sum_{i,j}\frac{1}{\alpha_i \alpha_j}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(i,j) \tag{11}$$

where $\delta(i,j)$ is the *Kroneck* function. If $\delta(i,j) = 1$, nodes i and j are in the same community. Otherwise, they are in different communities. We denote A_{ij} the value of the adjacency matrix. $A_{ij} = 1$ means there is an edge between node i and node j. Variable of α_i is the number of communities that node i

Algorithm 2 Content Delivery With Dense-Mode Routing

Require: n_{source}, n_{meet}, n_{aim}.
Ensure: Delivering message or not
1: **if** $n_{meet} = n_{aim}$ **then**
2: $\quad n_{source}$ delivers m to n_{aim};
3: **else if** n_{aim}, n_{meet} and n_{source} are in the same community & $(IP(n_{meet}) > IP(n_{source}))$ **then**
4: $\quad n_{source}$ delivers m to n_{meet};
5: **else if** n_{aim} and n_{meet} are in the same community & $(OP(n_{meet}) > OP(n_{source}))$ **then**
6: $\quad n_{source}$ delivers m to n_{meet};
7: **else if** $CC(n_{meet}) > CC(n_{source}) \parallel CS(n_{meet}) > CSn_{source}$ **then**
8: $\quad n_{source}$ delivers m to n_{meet};
9: **end if**

belongs to, and k_i is the degree of node i. The degree of all nodes is 2 m. The larger the value of EQ, the better the community division. Therefore, the objective of SO is to make EQ as large as possible.

b) Coding method: Assuming that each node belongs to M communities at the most, we need to assign M coding space for each node. In SO, each sheep represents a feasible community division, so the coding space assigned to each sheep is M multiplying by N, where N is the number of nodes. The range of values for each coding space locates in a continuous interval, but the community number is discrete. This requires mapping the continuous data to the discrete data. We map the interval of $[u - 0.5, u + 0.5)$ to community u.

c) Algorithm description: We then perform the SO algorithm to achieve the objective of community division. First, we need to initialize the community division, i.e., randomly assigning a value to each sheep. Then, the sheep leading process, the sheep interaction process, and the shepherd supervision process are executed iteratively until the iteration reaches the upper limit (recommending an experimental value of 1000) or the modularity meets the requirement (recommending an empirical value of 0.7 [35]). In the sheep leading process, all sheep move toward the lead sheep, which is utilized to search the global optimal solution. We apply the sheep interaction process to characterize and enhance sheep mobility, which makes the algorithm more sufficient to search the global optimal solution. The shepherd supervision process is used to avoid falling into the local optimal solution. Finally, we get the community division result.

2) Social Metric Definitions:

a) Out-of-community popularity: First, the out-of-community encounter frequency Fre_o between node u and node v is described in (12), where, $NS_o(u)$ is a collection of nodes outside the community that node u meets

$$\mathrm{Fre}_o(u, v) = \frac{f(u, v)}{\max(f(u, k))} \quad (k \in NS_o(u)). \tag{12}$$

Then, the out-of-community relationship closeness between node u and node v is defined as the weighted sum of the encounter stability (i.e., the encounter stability in the sparse-mode routing mechanism) and the out-of-community encounter frequency

$$\mathrm{RC}_o(u, v) = \gamma_1 * \mathrm{Fre}_o(u, v) + \gamma_2 * \mathrm{Sta}(u, v). \tag{13}$$

The *out-of-community popularity* of node u is defined as the average out-of-community relationship closeness between node u and other nodes outside the community. N is the total number of nodes in the network, and $N(co_u)$ is the number of nodes in the community where node u is located

$$\mathrm{OP}(u) = \frac{1}{N - N(co_u)} \sum_{i \neq co_u} \mathrm{RC}_o(u, i). \tag{14}$$

b) Community closeness: First, the between-community encounter frequency between node u' and node v' is defined in (15). $NS_i(v')$ is a collection of nodes that node u' meets in the community where node v, is located

$$\mathrm{Fre}_b(u', v') = \frac{f(u', v')}{\max(f(u', k), f(v', l))} \quad (k \in NS_i(v'), l \in NS_i(u'), u' \neq k, v' \neq l). \tag{15}$$

Then, the between-community relationship closeness between node u' and node v' is defined as the weighted sum of the encounter stability (i.e., the encounter stability in the sparse-mode routing mechanism) and the between-community encounter frequency

$$\mathrm{RC}_b(u', v') = \delta_1 * \mathrm{Fre}_b(u', v') + \delta_2 * \mathrm{Sta}(u', v'). \tag{16}$$

The *community closeness* between community 1 co_1 and community 2 co_2 is defined as follows:

$$\mathrm{CC}(co_1, co_2) = \frac{\sum_{u' \in co_1} \sum_{v' \in co_2} \mathrm{RC}_b(u', v')(u' \neq v')}{N(co_1) \times N(co_2) - N(co_1 \cap co_2)} \tag{17}$$

where $N(co_1)$ and $N(co_2)$ are the number of nodes of community 1 and community 2, respectively. $N(co_1 \cap co_2)$ is the intersection between community 1 and community 2. Particularly, we set the community closeness between the same communities to 1.

3) Dense Routing Algorithm Descriptions: The message traffic is large in the dense network, but the processing capacity of each node is limited. As a result, content delivery through the high-centrality nodes causes the performance bottleneck of BubbleRap. Therefore, we introduce the community closeness to make the community as a whole participate in content delivery.

As shown in Algorithm 2, n_{source}, n_{meet}, and n_{aim} are the source node, the encounter node, and the destination node, respectively. $\mathrm{IP}(n)$ and $\mathrm{OP}(n)$ are the IP and the out-of-community popularity of node n, respectively. $CC(n)$ is the community closeness between the community of node n and the community of the destination node. If the encounter node is the destination node, the source node will deliver the message to the destination node directly (lines 1 and 2). If the destination node, encounter node, and source node are in the same community, and the IP of the encounter node is greater than that of the source node, the message is delivered to the encounter node (lines 3 and 4). The source node deliveries the message to the encounter node when the source node and the

Algorithm 3 Scheme for Discriminating the Sparse Network From Dense Network

Require: $N(\text{AP})$, $N(\text{node})$.
Ensure: Sparse network or dense network
1: **if** $N(\text{node}) = O(N(\text{AP}))$ **then**
2: The network is sparse;
3: **else if** $N(\text{node}) = \Theta(N(\text{AP}) \times N(\text{AP}))$ **then**
4: The network is dense;
5: **end if**

encounter node are in the same community, and the out-of-community popularity of the encounter node is greater than that of the source node (lines 5 and 6). If the community closeness or the CS of the encounter node is greater than that of the source node, the message is delivered to the encounter node (lines 7 and 8).

D. Discriminating Scheme

The number of nodes in MSN changes over time, so a discriminating mechanism needs to be implemented on the SDN controller. When the number of nodes in MSN is small, the sparse-mode routing mechanism is adopted. Otherwise, the dense-mode routing mechanism is applied.

Learning from the concept of sparse and dense graphs in the graph theory, we design a discriminating scheme to conduct switching between these two routing mechanisms. The discriminating scheme mainly include the following three graph theory concepts.

1) *O Representation:* Supposing there is a nonnegative function $f(n)$ for all nonnegative integer n. If there is an integer n_0 and a normal number c, and for any $n \geq n_0$, there is $f(n) \leq cg(n)$, then $f(n) = O(g(n))$.
2) *Ω Representation:* Supposing there is a nonnegative function $f(n)$ for all nonnegative integer n. If there is an integer n_0 and a normal number c, and for any $n \geq n_0$, there is $f(n) \geq cg(n)$, then $f(n) = \Omega(g(n))$.
3) *Θ Representation:* Supposing there is a nonnegative function $f(n)$ for all nonnegative integer n. If and only if $f(n)$ is both $O(g(n))$ and $\Omega(g(n))$, then $f(n) = \Theta(g(n))$.

We denote $N(\text{AP})$ and $N(\text{node})$ the number of *APs* and the number of mobile nodes in the network, respectively. As shown in Algorithm 3, when $N(\textit{node})$ could be expressed by $O(N(\text{AP}))$, the network is considered as a sparse network and the sparse-mode routing mechanism is utilized. When $N(\text{node})$ could be expressed by $\Theta(N(\text{AP}) \times N(\text{AP}))$, the network is regarded as a dense network and the dense-mode routing mechanism is adopted.

IV. Evaluation

The experimental platform is the opportunistic network environment (ONE) [29]. The benchmark mechanisms include the maximum probability routing [30], BubbleRap [14], the first contact routing [31], the epidemic routing [32], and the direct delivery routing [33]. The sparse-mode routing mechanism and the dense-mode routing mechanism are evaluated, respectively.

TABLE I
Main Parameter Settings

Parameter	Value
cache size	50 MB
message life cycle	300 minutes
number of event generators	1
threshold of common AP	100
threshold of stable connection	10
Number of APs	34
Number of mobile nodes (Sparse Network)	45
Number of mobile nodes (Dense Network)	126

A. Parameter Settings

The parameter values affect the evaluation results greatly, so they should be set reasonably. In this article, we set the parameter values with multiple methods, including logical analysis, conducting comparative experiments, and referring to the recommended values. Table I shows the settings of the main parameters utilized in the simulation.

B. Evaluation Metrics

Delivery Ratio: The delivery rate is used to evaluate the message delivery capability of the routing mechanism in MSNs. The higher the delivery rate, the stronger the delivery capability of the routing mechanism. As depicted in (18), *delivered* is the number of messages successfully arriving at the destination node and *created* is the total number of messages generated in the network

$$\text{delivery\_prob} = \frac{\text{delivered}}{\text{created}}. \tag{18}$$

Average Overhead: The average overhead is used to measure the cost of the message reaching the destination node, which is determined by the forwarding times in the routing process and the total number of messages that successfully arrive at the destination node. To some extent, the overhead reflects the consumption of network resources, e.g., node cache and network bandwidth. As depicted in (19), *delivered* is the number of messages that successfully arrive at the destination node and *relayed* is the forwarding times during the experiment

$$\text{overhead} = \frac{\text{relayed} - \text{delivered}}{\text{delivered}}. \tag{19}$$

Average Delay: Average delay is one of the most important metrics to evaluate the performance of content delivery in MSNs. As depicted in (20), *num* is the number of messages that successfully arrive at the destination node, and *delayofMessage$_i$* is the delay of the ith message that successfully reaches the destination node

$$\text{Average\_delay} = \frac{\sum_{i=1}^{\text{num}} \text{delayofMessage}_i}{\text{num}}. \tag{20}$$

Average Buffertime: The average buffertime is the average caching time of all the discarded and forwarded messages cached in the node. The larger the buffertime, the better the routing mechanism. As depicted in (21), *deleted* is the number

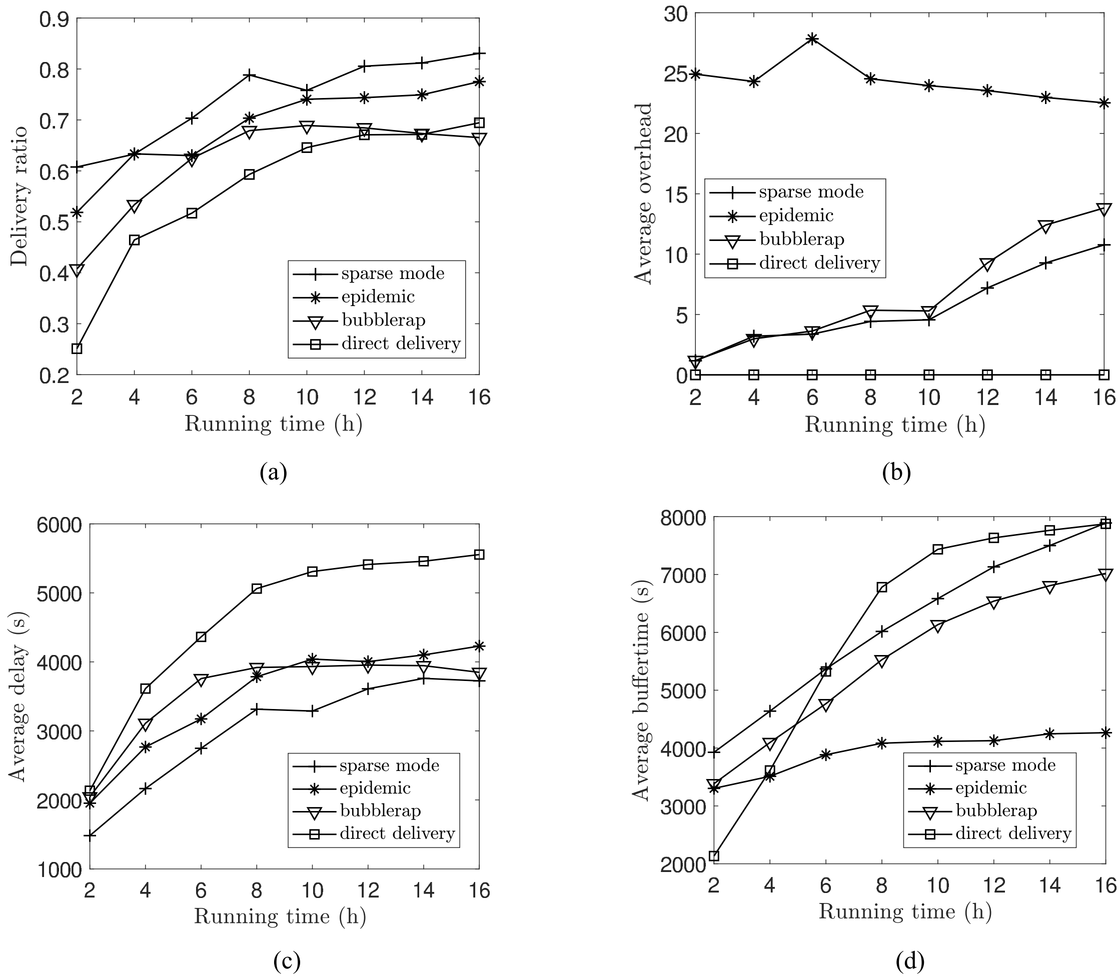


Fig. 3. Performance evaluation of the sparse-mode routing mechanism. (a) Delivery ratio. (b) Average overhead. (c) Average delay. (d) Average buffertime.

of the messages that have been deleted and *Buffertime* is the sum of the cache time of all the deleted messages

$$\text{average\_buffertime} = \frac{\text{Buffertime}}{\text{deleted}}. \tag{21}$$

C. Performance of the Sparse-Mode Routing Mechanism

We first evaluate the content delivery performance of the sparse-mode routing mechanism from the aspects of delivery ratio, overhead, delay, and buffertime.

Delivery Ratio: As shown in Fig. 3(a), the delivery ratio of the sparse-mode routing increases with the growth of the running time. The sparse-mode routing outperforms the other three routing mechanisms in the delivery ratio. This is because the community members of the sparse-mode routing tend to be stable over time. In addition, a variety of social attributes are considered when selecting a relay node. Thus, a node that is more likely to encounter the destination node could be screened effectively according to the social attributes. In a sparse network, the delivery ratio of the sparse-mode routing is up to 83%, which is nearly 20% higher than that of BubbleRap.

Average Overhead: As shown in Fig. 3(b), since the node only forwards the message when encountering the destination node, the overhead of the direct delivery routing is 0. The sparse-mode routing is just a little better than BubbleRap. The sparse-mode routing utilizes dual social attributes to forward the messages, which will increase the forwarding times. However, more messages are successfully delivered to the destination node at the same time, so the overhead of the message arriving at the destination node is slightly reduced. We also find that the epidemic routing is the most expensive because it forwards the messages in a flooding manner, although it performs well in the metric of delivery ratio.

Average Delay: As shown in Fig. 3(c), the average delay of all the routing mechanisms increases at the beginning and tends to be stable over time. The direct delivery routing is the most time consuming because only when the source node encounters the destination node, the message is forwarded. However, these two nodes are less likely to meet in a sparse network. Although the design concepts of BubbleRap and the epidemic routing are different, the delay of them does not present a significant difference. The sparse-mode routing applies dual social attributes to send messages to more efficient relay nodes, so the messages could be delivered to the destination node as soon as possible, resulting in a lower delay of 3011.4 s on average.

Average Buffertime: As shown in Fig. 3(d), the direct delivery routing is a single-copy routing mechanism, so the

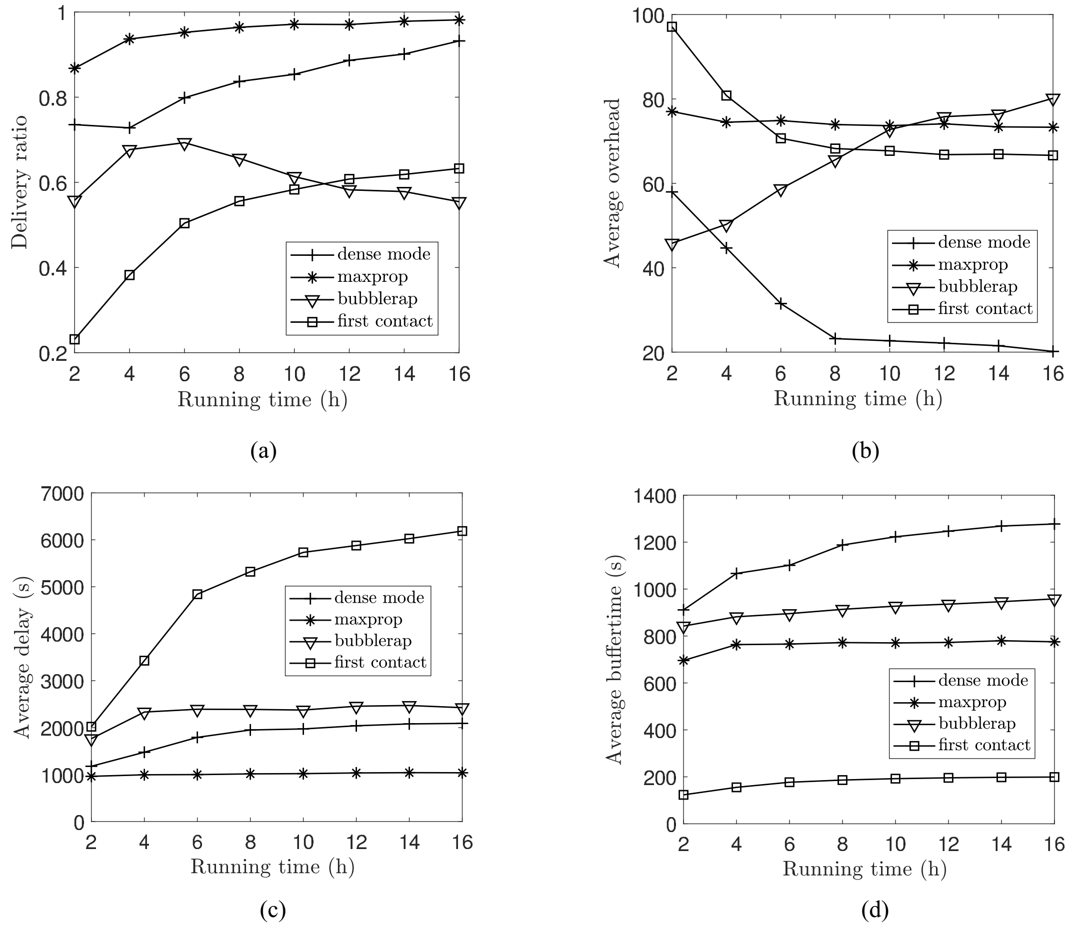


Fig. 4. Performance evaluation of the dense-mode routing mechanism. (a) Delivery ratio. (b) Average overhead. (c) Average delay. (d) Average buffertime.

buffertime is relatively large. The other three mechanisms are multicopy routing, i.e., a node forwards the message and caches a copy of the message locally. The message will be evicted only when the TTL expires or the cache of the node is filled up. If only considering the deleted messages caused by TTL expiration, the buffertime of the multicopy routing will tend to be 18 000 s. Note that the TTL of a message is set to 18 000 s in the experiment. However, the deleted messages caused by the filled cache will lower the buffertime of the multicopy routing in actual. Given this, the average buffertime could reflect the flooding levels of message copies. When adopting the sparse-mode routing, the source node does not send messages out unless it encounters a satisfying relay node, which could avoid the flooding of message copies. Therefore, the message copies are controlled effectively and the number of the deleted messages caused by the filled cache is very small, resulting in a higher buffertime up to 6131.2 s.

D. Performance of the Dense-Mode Routing Mechanism

We then evaluate the content delivery performance of the dense-mode routing mechanism from the aspects of delivery ratio, overhead, delay, and buffertime.

Delivery Ratio: As depicted in Fig. 4(a), the maximum probability routing presents the best performance in delivery rate. The maximum probability routing is a flooding-like routing mechanism, which adopts a more efficient message caching management mechanism than the epidemic routing. It increases the probability of the message delivered to the destination node and thus improves the content delivery rate. Although the delivery rate of the dense-mode routing is a little lower than the maximum probability routing, it outperforms BubbleRap and the first contact routing. The delivery rate of the proposed routing mechanism can reach up to 93%, which approaches to the maximum delivery rate achieved by the maximum probability routing. In the dense-mode routing, the community closeness is introduced to make the community as a whole participate in content delivery. This is an improvement of BubbleRap that only utilizes the high-centrality nodes to forward messages.

Average Overhead: As depicted in Fig. 4(b), the maximum probability routing has a better delivery rate due to its flooding policy, but it brings a large routing overhead at the same time. The first contact routing forwards the message to the relay node encountered for the first time without any basis. This causes a large number of unnecessary forwarding operations, so it also presents a large routing overhead. BubbleRap deliberately forwards messages according to the social relationships between each pair of nodes, which makes each forwarding useful for content delivery. However, packet loss caused by the filled cache of high-centrality nodes is getting worse over time. As a result, its overhead increases gradually. Compared to BubbleRap, the dense-mode routing has a lower overhead of 30.5 on average, because it adopts a more intelligent community division algorithm and takes advantage of some more powerful social attributes. In addition, it also solves the packet loss problem.

Average Delay: As depicted in Fig. 4(c), since the maximum probability routing takes a flooding policy, its average delay is the lowest. BubbleRap only forwards the message to the node with higher centrality, which consumes a certain amount of time to select the relay node. The first contact routing is the most time consuming because it only forwards the message to the first met node. As a result, the message is very likely to be delivered to the destination node through multiple hops. Compared to BubbleRap, the delay is reduced 503.3 s on average by the dense-mode routing. This is because the social attributes are considered more comprehensively in the dense-mode routing, which results in a high-transmission speed in content delivery.

Average Buffertime: As depicted in Fig. 4(d), the buffertime of the dense network is much lower than that of the sparse network. This is because the nodes keep frequent contacts in a dense network, which could promote the message forwarding and arriving at the destination node successfully. Once encountering a node, the first contact routing will forward the message to this node, which causes plenty of message copies in the network. Although the maximum probability routing adopts a routing-loop avoidance policy, i.e., a message will not be delivered to the node which has received this message, the flooding thought inevitably causes massive message copies. BubbleRap only forwards the message to the node with higher centrality, which consumes a certain amount of time to select the relay node. Bubblerap only forwards the message to the node with high centrality, which more or less controls the amount of message copies. The proposed dense-mode routing has the highest buffertime of 1160.2 s on average because it set more stringent conditions to prevent the flooding of message copies.

V. Conclusion

This article designed the SDN-assisted content delivery architecture and mechanisms for MSNs. Applying the centralized control thought of SDN, we proposed the sparse-mode and dense-mode routing mechanisms that could be flexibly be switched according to network density. The network density was first considered as a crucial metric to design the routing mechanisms for MSNs. We evaluated the proposed routing mechanisms from the aspects of delivery ratio, overhead, delay, and buffertime. Compared to BubbleRap, the delay of the sparse-mode routing and the dense-mode routing was reduced by 16% and by 22%, respectively. In addition, buffertime is increased by 13% and 27%, respectively, at the same time. The experiment results revealed that the proposed routing mechanisms were feasible and effective.

As part of future work, we will consider the selfishness of mobile nodes and design an incentive mechanism to encourage mobile nodes to participate in content delivery at the edge of MSNs.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments.

References

[1] X. Hu, T. H. S. Chu, V. C. M. Leung, E. C. Ngai, P. Kruchten, and H. C. B. Chan, "A survey on mobile social networks: Applications, platforms, system architectures, and future research directions," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1557–1581, 3rd Quart., 2015.

[2] F. Li, "Data-driven mobile social networks," in *Encyclopedia of Wireless Networks*, X. Shen, X. Lin, and K. Zhang, Eds. Cham, Switzerland: Springer, 2019.

[3] F. Li *et al.*, "A local communication system over Wi-Fi direct: Implementation and performance evaluation," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5140–5158, Jun. 2020.

[4] R. Lu, X. Lin, and X. S. Shen, "Spring: A social-based privacy preserving packet forwarding protocol for vehicular delay tolerant networks," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[5] J. Tao, H. Wu, S. Shi, J. Hu, and Y. Gao, "Contacts-aware opportunistic forwarding in mobile social networks: A community perspective," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2018, pp. 1–6.

[6] H. Zhou, M. Ruan, C. Zhu, V. C. M. Leung, S. Xu, and C. Huang, "A time-ordered aggregation model-based centrality metric for mobile social networks," *IEEE Access*, vol. 6, pp. 25588–25599, 2018.

[7] F. Li, H. Jiang, H. Li, Y. Cheng, and Y. Wang, "SEBAR: Social-energy-based routing for mobile social delay-tolerant networks," *IEEE Trans. Veh. Technol.* vol. 66, no. 8, pp. 7195–7206, Aug. 2017.

[8] B. Jedari, F. Xia, and Z. Ning, "A survey on human-centric communications in non-cooperative wireless relay networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 914–944, 2nd Quart., 2018.

[9] B. B. Chen and M. C. Chan, "MobiCent: A credit-based incentive system for disruption tolerant network," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[10] S. Bera, S. Misra, and A. V. Vasilakos, "Software-defined networking for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1994–2008, Dec. 2017.

[11] F. Li, J. Cao, X. Wang, and Y. Sun, "A QoS guaranteed technique for cloud applications based on software defined networking," *IEEE Access*, vol. 5, pp. 21229–21241, 2017.

[12] 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, May 2011.

[13] D. Yu *et al.*, "Stable local broadcast in multihop wireless networks under SINR," *IEEE/ACM Trans. Netw.*, vol. 26, no. 3, pp. 1278–1291, Jun. 2018.

[14] X. Liu *et al.*, "Fast and accurate detection of unknown tags for RFID systems—Hash collisions are desirable," *IEEE/ACM Trans. Netw.*, vol. 28, no. 1, pp. 126–139, Feb. 2020.

[15] Z. Mao, Y. Jiang, G. Min, S. Leng, X. Jin, and K. Yang, "Mobile social networks: Design requirements, architecture, and state-of-the-art technology," *Comput. Commun.*, vol. 100, pp. 1–19, Oct. 2018.

[16] F. Xia, L. Liu, J. Li, J. H. Ma, and A. V. Vasilakos, "Socially aware networking: A survey," *IEEE Syst. J.*, vol. 9, no. 3, pp. 904–921, Sep. 2015.

[17] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin, and F. R. Yu, "A survey of mobile information-centric networking: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2353–2371, 3rd Quart., 2018.

[18] D. Xu *et al.*, "A survey of opportunistic offloading," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2198–2236, 3rd Quart., 2018.

[19] A. Roy, T. Acharya, and S. DasBit, "Quality of service in delay tolerant networks: A survey," *Comput. Netw.*, vol. 130, pp. 121–133, Jan. 2018.

[20] 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, pp. 1–13, Jan. 2017.

[21] R. Gupta, N. Krishnamurthi, U. T. Wang, and T. Tamminedi, "Routing in mobile ad-hoc networks using social tie strengths and mobility plans," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2017, pp. 1–6.

[22] H. Zhang, Z. Chen, J. Wu, and K. Liu, "FRRF: A fuzzy reasoning routing-forwarding algorithm using mobile device similarity in mobile edge computing-based opportunistic mobile social networks," *IEEE Access*, vol. 7, pp. 35874–35889, 2019.

[23] M. Zhu, J. Cao, D. Pang, Z. He, and M. Xu, “SDN-based routing for efficient message propagation in VANET,” in *Proc. Int. Conf. Wireless Algorithms Syst. Appl.*, 2015, pp. 788–797.

[24] B. Dong, W. Wu, Z. Yang, and J. Li, “Software defined networking based on-demand routing protocol in vehicle ad hoc networks,” in *Proc. 12th Int. Conf. Mobile Ad Hoc Sensor Netw.*, 2016, pp. 207–213.

[25] N. B. Truong, G. M. Lee, and Y. G. Doudane, “Software defined networking-based vehicular adhoc network with fog computing,” in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, 2015, pp. 1202–1207.

[26] D. Wei, Z. Liu, and H. Cao, “SRPA: SDN-based routing protocol for ad hoc networks,” in *Proc. 9th Int. Conf. Inf. Technol. Med. Educ.*, 2018, pp. 1012–1017.

[27] S. Correia, A. Boukerche, and R. I. Meneguette, “An architecture for hierarchical software-defined vehicular networks,” *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 80–86, Jul. 2017.

[28] H. Ghafoor and I. Koo, “CR-SDVN: A cognitive routing protocol for software-defined vehicular networks,” *IEEE Sensors J.*, vol. 18, no. 4, pp. 1761–1772, Feb. 2018.

[29] A. Keränen, J. Ott, and T. Kärkkäinen, “The ONE simulator for DTN protocol evaluation,” in *Proc. 2nd Int. Conf. Simulat. Tools Techn.*, vol. 55, 2009, pp. 1–10.

[30] L. Zhang, X. Wang, J. Lu, M. Ren, Z. Duan, and Z. Cai, “A novel contact prediction-based routing scheme for DTNs,” *Trans. Emerg. Telecommun. Technol.*, vol. 28, no. 1, pp. 1–12, 2017.

[31] S. Jain, K. Fall, and R. Patra, “Routing in a delay tolerant network,” in *Proc. ACM SIGCOMM*, 2004, pp. 145–158.

[32] Q. Xu, Z. Su, K. Zhang, and P. Ren, “Epidemic information dissemination in mobile social networks with opportunistic links,” *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 3, pp. 399–409, Sep. 2015.

[33] A. Majeed, D. Robin, and A. Ahmad, “Routing protocols in opportunistic networks—A survey,” *IETE Tech. Rev.*, vol. 35, no. 4, pp. 1–19, 2018.

[34] D. Qu, L. Xu, Y. Lu, X. Yuan, M. Huang, and X. Wang, “A new swarm intelligence algorithm by simulating sheep behaviors,” *Acta Electronica Sinica*, vol. 46, no. 6, pp. 1300–1305, 2018.

[35] H. Shen, X. Cheng, K. Cai, and M. Hu, “Detect overlapping and hierarchical community structure in networks,” *Physica A Stat. Mech. Appl.*, vol. 388, no. 8, pp. 1706–1712, 2009.

Fuliang Li (Member, IEEE) received the B.S. degree in computer science from Northeastern University, Shenyang, China, in 2009, and the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2015.

He is currently an Associate Professor with the School of Computer Science and Engineering, Northeastern University. He was a Postdoctoral Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, from 2016 to 2017. He published 50 journal/conference papers, including journal papers, such as IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON CLOUD COMPUTING, *Computer Networks*, COMPUTER COMMUNICATIONS, *Journal of Network and Computer Applications*, and mainstream conferences, such as IEEE INFOCOM, IEEE ICDCS, IEEE/ACM IWQoS, IEEE GLOBECOM, IEEE LCN, IEEE CLOUD, and IFIP/IEEE IM. His research interests include network management and measurement, mobile computing, software-defined networking, and network security.

Yaoguang Lu received the M.S. degree from the School of Computer Science and Engineering, Northeastern University, Shenyang, China, in 2019, where he is currently pursuing the Ph.D. degree.

His research interests include vehicle networks, software-defined networking, and mobile social networks.

Xingwei Wang received the B.S., M.S., and Ph.D. degrees in computer science from the Northeastern University, Shenyang, China in 1989, 1992, and 1998, respectively.

He is currently a Professor with the School of Computer Science and Engineering and the Head of College of Software, Northeastern University. His research interests include future Internet, cloud computing, mobile computing, and mobile social network.

Yuanguo Bi (Member, IEEE) received the B.Sc. degree in computer science from Liaoning University, Shenyang, China, in 2003, and the M.Sc. and Ph.D. degrees from Northeastern University, Shenyang, in 2006 and 2010, respectively.

He is an Associate Professor with the School of Computer Science and Engineering, Northeastern University. He published 30 journal/conference papers, including high-quality journal papers, such as the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *Mobile Networks & Applications* (ACM/Springer) and mainstream conferences, such as IEEE Globecom and IEEE ICC. His current research interests focus on medium access control, QoS routing, multihop broadcast, mobility management in vehicular networks, as well as SDN-enabled vehicular networks.

Dr. Bi has served as an editor/guest editor for *IEEE Communications Magazine*, IEEE ACCESS, SpringerPlus, and a TPC member for many IEEE conferences, e.g., IEEE ICC, IEEE Globecom, and IEEE VTC.

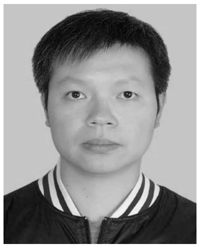
Tian Pan received the B.S. degree from Northwestern Polytechnical University, Xi’an, China, in 2009, and the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2015.

He is currently a Postdoctoral Fellow Researcher with the Department of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing. His research interests include router architecture, network processor architecture, network power efficiency, and software-defined networking.

Yuchao Zhang (Member, IEEE) received the B.S. degree in computer science and technology from Jilin University, Changchun, China, in 2012, and the Ph.D. degree from Computer Science Department, Tsinghua University, Beijing, China, in 2017.

She is currently an Associate Professor with the Beijing University of Posts and Telecommunications, Beijing. Her research interests include large-scale datacenter network, content delivery network, data-driven network, and edge computing.

Dr. Zhang is a member of ACM and CCF.



Weichao Li received the B.E. and M.E. degrees from the South China University of Technology, Guangzhou, China, in 2002 and 2009, respectively, and the Ph.D. degree from the Department of Computing, Hong Kong Polytechnic University, Hong Kong, in 2017.

He currently serves as a Research Assistant Professor with the Institute of Future Networks, Southern University of Science and Technology, Shenzhen, China. His research focuses on network measurement, especially on Internet measurement technologies, large-scale network performance monitoring and diagnosis, mobile network performance monitoring. He is also interested in areas including SDN, industrial Internet, cloud computing, quality of experience measurement, and machine learning.

Yi Wang (Member, IEEE) received the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in July, 2013.

He is currently a Research Associate Professor with the SUSTech Institute of Future Networks, Southern University of Science and Technology, Shenzhen, China. His research interests include router architecture design and implementation, software-defined networks, fast packet forwarding, information-centric networking, and time-sensitive networks.