CAIS: A Copy Adjustable Incentive Scheme in Community-Based Socially Aware Networking

Zhaolong Ning, *Member, IEEE*, Li Liu, Feng Xia, *Senior Member, IEEE*, Behrouz Jedari, Ivan Lee, *Senior Member, IEEE*, and Weishan Zhang

***Abstract*—Socially aware networking (SAN) is a new communication paradigm, in which the social characteristics of mobile nodes are exploited to improve the performance of data distribution. In SAN, mobile carriers may exhibit selfish behaviors and refuse to relay messages for others for various reasons, such as limited resources (e.g., buffer, energy, and bandwidth) or social relationships. Several incentive schemes have recently been investigated to stimulate selfish users for cooperation in data forwarding. However, a majority of the existing methods have not fully studied nodes' social relationships in their selfish behaviors. In this paper, we propose a copy adjustable incentive scheme (CAIS), which adopts the virtual credit concept to stimulate selfish nodes to cooperate in data forwarding. In CAIS, we consider a network in which the nodes are divided into certain communities based on their social relationships. Then, we apply two types of credits, i.e., social credit and nonsocial credit, to reward the nodes when they relay data for other nodes inside their community or outsiders, respectively. Based on our mechanism, the number of messages a node can replicate to other nodes is adjusted according to its cooperation level and earned credits. To further improve the performance of CAIS, a single-copy data replication policy is employed, which manages the credit distribution of each node according to its available resources. The results of our extensive experiments using both synthetic and trace-driven simulations illustrate that CAIS copes well with node selfishness in community-based networks and outperforms other benchmark protocols with high data delivery ratio, low communication overhead, and short data delivery latency.**

***Index Terms*—Cooperative data routing, incentive scheme, socially aware networking (SAN), user selfishness.**

Manuscript received August 9, 2015; revised January 25, 2016 and May 23, 2016; accepted July 13, 2016. Date of publication July 19, 2016; date of current version April 14, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61572106, Grant 61502075, and Grant 61401082; by the Fundamental Research Funds for the Central Universities; by the China Postdoctoral Science Foundation under Project 2015M580224; and by the Liaoning Province Doctor Startup Fund under Grant 201501166. The review of this paper was coordinated by Prof. J. Sun. *(Corresponding author: Feng Xia.)*

Z. Ning, F. Xia, and B. Jedari are with the School of Software, Dalian University of Technology and the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian 116620, China (e-mail: f.xia@ieee.org).

L. Liu is with the School of Software, Dalian University of Technology, Dalian 116620, China, and with the School of Information Science and Electronic Engineering, Shandong Jiaotong University, Jinan 250357, China.

I. Lee is with the School of Information Technology and Mathematical Sciences, University of South Australia, Adelaide, S.A. 5001, Australia.

W. Zhang is with the Department of Software Engineering, China University of Petroleum, Qingdao 266580, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2016.2593051

I. Introduction

The increasing popularity of low-cost mobile devices, such as smartphones and on-board units, has enabled users to communicate with each other via short-range Bluetooth or Wi-Fi interfaces [1]. In this setting, an end-to-end connectivity between mobile nodes (i.e., devices and their carriers) might not exist due to the lack of infrastructure, and they contact each other in an opportunistic manner. Delay-tolerant network (DTN)-based technologies are promising in dealing with data sharing and distribution in such disruptive scenarios. In DTNs, mobile nodes employ a store-carry-and-forward scheme where they use a specific buffering mechanism to store their messages until a new route becomes available to forward (or replicate) the messages [2]. For example, vehicular DTNs (VDTNs) have emerged as modern DTN technologies that allow forwarding messages to reach the destinations through traveling vehicles in roads or highways [3]. In this setting, mobile nodes are encouraged to share their resources (such as energy and bandwidth) to promote the overall routing performance. However, some nodes may exhibit selfish behavior and refuse to carry messages on behalf of others, unless their cost can be compensated [4].

In recent years, the socially aware networking (SAN) paradigm has emerged as a modern type of communication in the DTN context [5]. For example, vehicular social networks [6] have gained much attention as a new application of the SAN paradigm that combines the characteristics of VDTNs and social networks to streamline networking protocols in the context of transportation systems and smart cities. Specifically, social network analysis concepts (such as node centrality, similarity, and community structure) are utilized in the SAN paradigm to exploit the nodes' social relationships. The motivation is that the social features and behavior of mobile users stand for a long time, thus providing stable connectivity among them [7], [8].

Several social-based data routing and dissemination protocols have been investigated to improve the performance of opportunistic routing in the SAN paradigm (see [9] and [10]). By exploring the existing data forwarding protocols, it can be inferred that the performance of routing protocols is mainly affected by three factors: node contact probability (contact time and intercontact time), data replication policy (e.g., single-copy or multicopy), and node cooperation model (cooperative or selfish). In some methods (see, for example, [11]), future contacts among nodes are predicted to select the appropriate relaying nodes. Furthermore, some routing algorithms (such as Spray and Wait [12]) aim to control the number of message copies to leverage a tradeoff between resource consumption and data delivery probability. However, the existing works are far from enough, and the main challenges come from the following four aspects.

- A fundamental assumption in most of the existing routing protocols is that mobile nodes are fully cooperative in data forwarding. However, in the real DTN applications, mobile nodes may not always carry and forward messages for others due to various reasons, such as resource limitation, privacy concern, or lack of interest in helping nodes from other communities. Therefore, the selfish behavior of mobile users is a nontrivial issue, which can affect the performance of data routing in the SAN paradigm dramatically [13].
- Generally speaking, two types of selfish nodes can be identified: *individually selfish* node and *socially selfish* node [14]. The former exhibits the same degree of selfishness toward other nodes, whereas the latter mitigates the degree of selfishness based on the strength of her social relationship. In other words, individually selfish nodes do not relay messages for other nodes, regardless of their social relationship, whereas socially selfish nodes relay messages for nodes with strong social ties (i.e., close friends) while refusing to carry messages for strangers. However, most of the existing research studies mainly focused on one type of selfish nodes.
- In spite of the fact that many incentive schemes have been studied to stimulate selfish nodes in mobile networks [15], the social relations and characteristics of the nodes have not been fully considered in their stimulation mechanisms. The reason behind is that the charge and rewarding mechanisms applied in the existing mechanisms generally aim to maximize the individual utility of nodes. Nevertheless, we observe in the true sense of the world that users mitigate their selfish behaviors based on their social relationships and interests to maximize their social benefits.
- In addition, most of the existing incentive schemes require a centralized trusted authority to verify and manage the payment services timely and fairly, which is difficult to realize for a network with intermittent connections among users.

To copy with the aforementioned challenges, we propose a copy adjustable incentive scheme (CAIS) for the community-based SAN, which employs the concept of virtual credit to stimulate selfish nodes in message forwarding. Community is divided based on individual interests, and the node with similar interests is divided into the same community. Specifically, CAIS applies a credit-based mechanism with two types of credits, i.e., social credit and nonsocial credit, to, respectively, reward nodes for relaying messages inside their community (intercommunity) and outsiders (intracommunity). In CAIS, we apply a single-copy message replication strategy, where the source of a message can replicate the message to other intermediate nodes. Therefore, the credit distribution is simplified without the requirement of a third party. Furthermore, CAIS considers the nodes' forwarding criteria and requirements to keep a balance among their cooperation level, resource consumption, and routing performance. Thus, nodes with both individually selfish and socially selfish characters are encouraged to carry messages for other nodes in their own community, as well as other communities.

Our major contributions can be summarized as follows.

- We design an incentive scheme, namely, CAIS, for the community-based SAN that stimulates selfish nodes to cooperate with nodes inside their own community and outsiders in data delivery.
- We establish relationships among the nodes' requirements based on their available network resources. Thus, CAIS facilitates selfish nodes to save their resources with respect to the history of their cooperation and guarantees their delivery performance.
- Our extensive experiments using both synthetic and trace-driven simulations give directions to the researcher community for efficient networking protocol design by considering the users' real selfish behaviors.
- The comparison with some benchmark routing protocols illustrates that CAIS can effectively stimulate selfish nodes to cooperate and outperform the other algorithms in terms of message delivery ratio, delay, and communication cost.

The rest of this paper is organized as follows. In Section II, we review some recent studies about the impact of node selfishness on the performance of DTN routing protocols and explore some well-known incentive schemes. In Section III, we elaborate on the framework of CAIS, including its system model, data replication method, architecture, and data exchange process. In Section IV, the implementation details of CAIS are outlined. We report the evaluation results in Section V. Finally, we conclude this paper in Section VI.

II. Related Work

Several social-based data routing and dissemination protocols such as Bubble Rap [16], dLife [17], and PIS [18] have been proposed in DTNs (see [9] and [15] for surveys). In [16], Hui *et al.* exploited the community structure and node centrality to make efficient forwarding decisions by utilizing the contact history of nodes to form social communities. Then, node centrality is identified from the local and global perspective. Each message is replicated based on global centrality until it reaches the destination community. After that, local centrality is used to deliver the message to its destination. In dLife [17], a social graph with weighted ties among nodes was designed to model the dynamic behaviors of mobile carriers. Afterward, a message is replicated to an encountered node if it has a stronger social tie with the destination node than the current message carrier. However, it is potentially assumed in the majority of existing algorithms that mobile nodes relay messages for other nodes and do not show selfish behavior in data relaying. However, in reality, some nodes attempt to maximize their own utility, and they are not interested in relaying data for other nodes, i.e., they have behave selfishly.

The impact of individual selfishness on the performance of DTN routing protocols has been extensively investigated (e.g., [19]), whereas the socially selfish behaviors are not sufficiently explored. Li *et al.* [20] studied how social selfishness influences the performance of Epidemic routing. They assumed that selfish nodes cooperate with other nodes inside their own community, but they refuse to relay messages for nodes in other communities. Give2Get [21] aimed to detect selfish nodes in Epidemic routing under the assumption that nodes are selfish with outsiders. In other words, they deviate from the protocol for their personal interest only when this does not damage individuals from the same community.

Recently, different incentive schemes have been proposed to promote the cooperation of mobile nodes in DTNs. In general, existing methods can be categorized into three types [9]: Tit-For-Tat (TFT), credit-based, and reputation-based schemes. The basic idea of the TFT scheme is to force encountered nodes to exchange the same amount of network traffic. For example, Buttyn *et al.* in [22] applied a two-person game approach in which messages are traded between users, and a user can download a message from another user if she can give a message in return. ConSub [23] aims to maximize the nodes' personal utility when their storage space is limited. To this aim, it determines the utility of each message based on the contact probability and cooperation level of nodes.

Reputation-based incentive mechanisms have been investigated to avoid the unfairness problem of TFT schemes. The reputation of nodes indicates their cooperation degree, which helps them make appropriate forwarding decisions when they contact each other. Generally speaking, reputation-based schemes increase the reputation of cooperative nodes while decreasing the reputation of noncooperative nodes. For example, IRONMAN [24] utilized preexisting social network information of nodes to detect their selfish behaviors. In this method, the sender of a message keeps the forwarding records of her encountered nodes, which contain the identity of each message, the destination of the message, and the forwarding time. FITS [25] was a hybrid incentive scheme (game-theoretic and reputation-based) that had rigorous analysis and guaranteed incentive compatibility in a realistic model; two incentive techniques (simple and sophisticated) were employed, both of which had a subgame perfect Nash equilibrium. However, the performance of a reputation-based incentive scheme is severely degraded when messages are lost. In addition, the distribution of nodes' reputation in DTNs is very challenging because there is no end-to-end connection among nodes most of the time.

In credit-based mechanisms, the concept of virtual credit or currency is employed to stimulate selfish nodes for cooperation in data forwarding. In this approach, a node receives credit as a reward when she relays messages received from others. Using the same mechanism, the node pays credit to other nodes when they relay messages for it. For example, Ning *et al.* [26] proposed a credit-based scheme in which data fall into a range of interest types, and each node has multiple interests. The content exchange between two nodes is formulated as a two-person cooperative game, and a utility function is designed for each node to maximize her expected credit reward. Similarly, Wang *et al.* [27] proposed a multireceiver incentive-based dissemination method, in which nodes use their local historical paths and interest information to forward packets of interest to as many subscribers as possible by the fewest transmissions. Wu *et al.* [28] studied a game-theoretic-based scheme, which is motivated by the observation that message exchange in probabilistic routing is analogous to commodity exchange in markets. However, the social characteristics of mobile nodes are not considered in this work.

Quite recently, Seregina *et al.* [29] explored credit-based mechanisms in two-hop DTNs under the circumstances that a source node promises each relay it meets a reward, if its message is delivered to the destination as fast as possible. In particular, they showed that under fairly weak assumptions, the expected reward the source pays remains the same irrespective of the information it conveys, provided that the type of information does not dynamically vary over time. However, this study only considered the case that one source–destination pair generates packets, while no message discarding policy has been employed in case node buffer overflows. Similarly, Cai *et al.* [30] proposed an incentive-compatible routing protocol for two-hop and multicopy DTNs based on game theory. In this method, both the encounter probability and transmission cost were considered in dealing with the misbehavior of selfish nodes. In addition, optimal relay nodes were selected based on the optimal sequential stopping rule and Vickrey–Clarke–Groves auction to ensure that nodes can honestly report their encounter probability and transmission cost to maximize their rewards.

Unlike the schemes previously mentioned, in this paper, we devise a credit-based mechanism in which mobile nodes can adjust their selfish behavior according to their individual and social requirements. Notably, we utilize the social relationships among nodes to form communities. In addition, we use social and nonsocial credits to reward selfish nodes when they relay messages for nodes inside their community and for outsiders, respectively.

III. Framework of Copy Adjustable Incentive Scheme

Here, we present our designed incentive scheme, namely, CAIS, in four subsections: system model, data replication method, architecture, and data exchange process.

A. System Model

In CAIS, there exist K nodes in a network, and a portion of nodes are selfish. We employ the concept of virtual credit to reward cooperative nodes when they forward messages for other nodes. We establish a relationship between the nodes' cooperative degree and the number of message copies they can replicate. A node with a high number of credits is able to generate and replicate a large number of messages to others. In contrast, a node with a low number of received credits (i.e., noncooperative node) cannot forward or replicate as many messages as cooperative nodes. As a result, the individual utility of a noncooperative node is decreased, and it cannot forward messages in its buffer to other encountered nodes.

We assume that nodes with similar interests or social relationships belong to the same community, and they adopt different

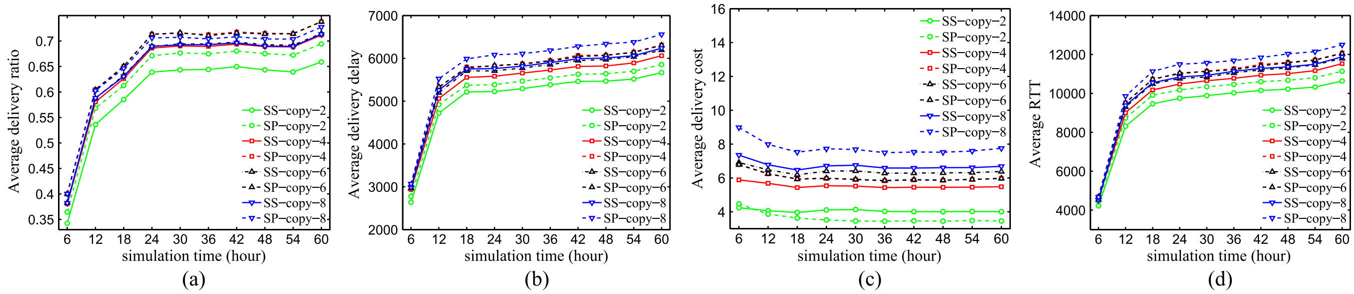


Fig. 1. Performance comparison of single-copy and multicopy data replication policies. (a) Delivery ratio. (b) Delivery delay. (c) Delivery cost. (d) RTT.

cooperative (or selfish) behaviors to nodes inside their own community and to outsiders. For instance, they are willing to relay messages for nodes with strong social ties, while refusing to relay messages for strangers. To deal with this problem, CAIS utilizes social and nonsocial credits to manage the nodes' rewarding mechanism when they respectively apply intracommunity and intercommunity data forwarding schemes. In our method, nodes adjust their selfish behaviors according to their current credit values. Variables $SC_{\min}$ and $NSC_{\min}$ represent the social and nonsocial threshold values. When a node exhibits selfish behavior in data forwarding for others and its current social (or nonsocial) credit is less than $SC_{\min}$ (or $NSC_{\min}$), it cannot replicate messages to other encountered nodes. The values of social and nonsocial credits determine the number of copies of messages a node can generate and forward to other nodes inside her community and to outsiders, respectively. We note that the credit for message relaying is attached to the message, and it is awarded to an intermediate node when it delivers the message to its destination successfully. If a node obtains more social credit, it implies that it contacts nodes in its own community more frequently. Therefore, it can be deduced that cooperation with nodes inside her community would improve network performance. This policy can also be implied as a punishment mechanism for it. Consequently, nodes can keep a balance among cooperation and resource consumption.

CAIS adopts the *single-copy data replication* policy to distribute messages and credits among nodes. Using this strategy, the source node of a message can only forward the copy of her messages to other encountered nodes, but the intermediate nodes cannot replicate messages received from other nodes to each other, except for their destinations nodes. However, intermediate nodes can also drop messages in their buffer when they expire or their buffers are full.

B. Single-Copy Versus Multicopy Data Replication

Generally speaking, two data replication policies are adopted for data forwarding in DTNs: *single-copy routing* [31] and *multicopy routing* [32]. In the former, one copy of a message is spread through the network. During the transmission process, if a sender node transfers a message to a receiver node successfully, it will delete the message from its buffer. In contrast, multiple copies of a message can be replicated in multicopy routing mechanisms. When a sender node forwards a copy of a message to an intermediate node, she keeps the message until it is expired or drops due to the limited buffer space or delivers to its destination node. Although the multicopy routing policy can improve the performance of data forwarding, a large number of message copies are generated, which dramatically increases network overhead.

Recently, extensive efforts have been carried out to devise effective data replication policies in disruptive scenarios and leverage a tradeoff between resource usage and data delivery. For instance, the binary Spray and Wait [12] routing protocol is investigated, which consists of two phases. In the Spray phase, a specific number of copies of a message, e.g., L, is initially generated and replicated to other intermediate nodes by the source node. Through the forwarding process, each node (either the source or the intermediate nodes) with more than one copy of the message delivers half the number of the message copies to another encountered node and keeps the remaining half for itself. In this replication method, a message can be delivered by either its source or an intermediate node to its destination that is called *multicopy data replication*. Since several intermediate nodes take part in data delivery, the credit distribution is very challenging.

In *single-copy data replication*, the source node can replicate her messages to other intermediate nodes. Then, the intermediate nodes can deliver each message to their destinations. In other words, the intermediate nodes carry the messages until they contact the destination of each buffered message. Using this method, the number of message copies in the network can be controlled. In addition, through the message forwarding process, only one intermediate node will exist in the delivery path that simplifies the credit distribution mechanism.

To evaluate and compare the performance of single-copy and multicopy replication methods, we perform a set of experiments using the Prophet routing protocol [33]. We run the two replication methods with different numbers of message copies (2, 4, 6, and 8). Herein, SS stands for the single-copy scheme, and SP represents the Spray and Wait protocol [12]. As demonstrated in Fig. 1, we note that although the average delivery ratio obtained by SP is 3.43% higher than that by SS in total, the average delivery delay, delivery cost, and round-trip time (RTT) gained by SS are 3.25%, 3.58%, and 4.27% lower than those by SP, respectively. The main reason is that in the single-copy replication method, message copies are replicated only between source and intermediate nodes. Moreover, message copies slowly spread in the network, and the spread range is also limited. Above all, the credit distribution is simplified without the requirement of a third party in SS; therefore, the single-copy replication method is considered as the data replication policy in CAIS.

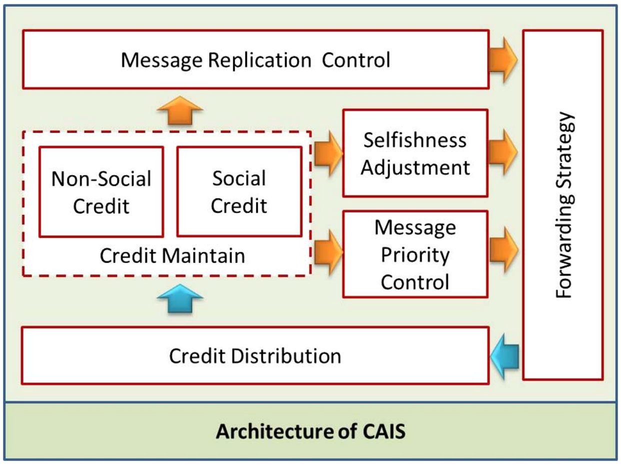


Fig. 2. Architecture of CAIS.

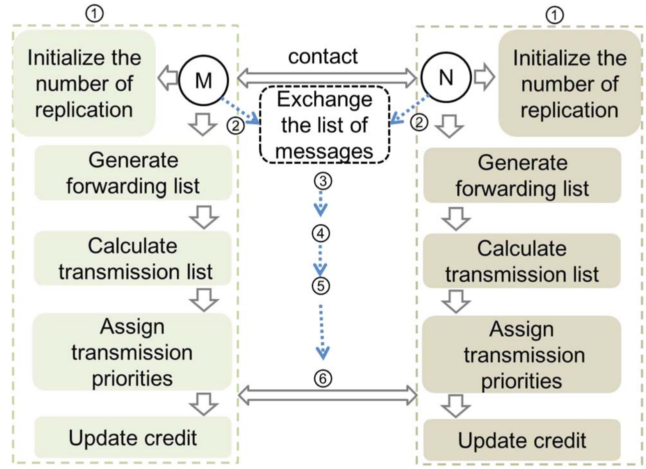


Fig. 3. Data exchange process in CAIS.

C. Architecture of CAIS

Fig. 2 shows the architecture of CAIS that consists of six components: *credit maintain*, *message replication control*, *selfishness adjustment*, *message priority control*, *forwarding strategy*, and *credit distribution*. In the following, we introduce each component and explain the corresponding functionality.

Credit maintain is the most important component in CAIS, which manages both social and nonsocial credits for each node. As previously mentioned, we group the network nodes into different communities based on the strength of their social relationships. When two nodes from different communities exchange messages (i.e., intercommunity forwarding), they trade nonsocial credit with each other. Then, the acquired credit can be employed by each node to stimulate other selfish nodes to forward messages to other communities. In contrast, the social credit is awarded to intermediate nodes when they relay messages inside their communities (i.e., intracommunity forwarding). Similarly, the social credit can be used by each node to pay for forwarding messages inside the community.

Message replication control takes the values of social and nonsocial credits into account to identify the number of copies of a message that a node can replicate inside its community and other communities. If a node forwards its buffered messages and does not relay messages for others, its social and nonsocial credits will be gradually decreased. In the worst case, the selfish node will not be able to replicate its messages to other intermediate nodes in the network, which will degrade its individual benefits as well as the overall routing performance.

Selfishness adjustment identifies the cooperation level of each node based on its credit value. Node selfishness inside the community belongs to social selfishness. Node selfishness between communities is viewed as individual selfishness. This is becaus when nodes have the requirement for data delivery, they may help other nodes forward data. However, when their credit is enough (i.e., the amount of credit she possesses is more than the threshold value), they would also refuse to forward data to save their limited network resources. Therefore, we separate social selfishness and individual selfishness and evaluate them, respectively.

Message priority control manages the forwarding priority of buffered messages for each node. Considering the fact that CAIS manages nonsocial and social credits independently, a node may have different amounts of social and nonsocial credits. To deal with credit availability, this component adjusts the priority of forwarding messages for each node based on its requirements and available resources. For example, if a node intends to increase its social credit, the *message priority control* will give higher forwarding priority to nodes destined to nodes inside the community.

Forwarding strategy selects appropriate intermediate nodes to forward messages for each node. This component relays a message to another encountered node if she has higher contact probability to the destination of the message.

Credit distribution manages credit assignment when a node relay messages for other encountered nodes. If a node relays a message for another node in another community, this component increases the nonsocial credit of the node, whereas the social credit of the node is increased by relaying messages for nodes in the same community.

D. Data Exchange Process in CAIS

Fig. 3 shows the data exchange process in CAIS. For the sake of simplicity, we assume that two nodes M and N contact each other to exchange their messages. Thus, data forwarding among these nodes is carried out in six steps as follows.

1) Nodes M and N generate the list of their forwarding messages based on their current social and nonsocial credits.
2) Nodes M and N exchange the list of their buffered messages stored in their buffers.
3) Based on the message list, the nodes exchange their candidate messages by removing messages that they have previously received.
4) Nodes M and N filter their forwarding list according to the *selfishness adjustment* module.
5) Nodes M and N assign the priority of their forwarding messages using the *message priority control* module to obtain the required credits effectively.
6) Finally, nodes M and N pay credit for each successfully delivered message using the *credit distribution* module,

TABLE I
EXPLANATION OF NOTATIONS

Notation	Explanation
NSC_{cur}	Value of current non-social credit
SC_{cur}	Value of current social credit
NSC_{init}	Initiate value of non-social credit
SC_{init}	Initiate value of social credit
NSC_{min}	Minimum value of non-social credit
SC_{min}	Minimum value of social credit
CPN_{max}	Maximum limited copy number
CPN_{nso}	Copy number for non-social node
CPN_{so}	Copy number for social node
CPN	Summation of message copy
α	Unit credit required to pay by each copy
β	Unit credit gained by message forwarding
msg	One message in message list
D_{msg}	Destination of the message
$P(M, D_{msg})$	Probability to meet between M and D_{msg}
RLN_{nso}	Sum number of relayed messages for non-social nodes
RLN_{so}	Sum number of relayed messages for social nodes

where the values of the social and nonsocial credits are updated accordingly.

IV. IMPLEMENTATION OF COPY ADJUSTABLE INCENTIVE SCHEME

This section provides the details of the implementation of CAIS.

A. Message Replication Control

This section describes the details of the message replication strategy in CAIS. The number of copies of a message has a direct influence on the routing performance. Meanwhile, the delivery ratio can be increased as the number of message copies increases. Furthermore, it can be induced that the distribution of a large amount of messages through the network can cause a data congestion problem, which seriously degrades the delivery performance. To control data congestion, we define a maximum value $CPN_{\max}$ for the number of copies of each message according to the nodes' current social and nonsocial credits (SC_{cur} and NSC_{cur}). For instance, if the nonsocial credit of a node is higher than its social credit, the node will forward messages in its buffer that are destined to nodes in other communities. As shown in Fig. 1, better network performances are accompanied with a larger value of $CPN_{\max}$. We observe that network efficiency is high when $CPN_{\max} = 6$ and $CPN_{\max} = 8$. We note that the gap of network performance achieved by $CPN_{\max} = 6$ and $CPN_{\max} = 8$ is not obvious. Since more load will be brought when $CPN_{\max} = 8$, we set $CPN_{\max} = 6$. For each message replication, a node needs to pay α credit (social or nonsocial credit) to another node, while she obtains β credit by forwarding each message. Table I summarizes the main notations used in the CAIS scheme.

To determine the number of message copies for each node, we consider three situations as follows.

1) $NSC_{\text{cur}} > NSC_{\text{min}}$ and $SC_{\text{cur}} > SC_{\text{min}}$: If the current values of nonsocial credit (NSC_{cur}) and social credit (SC_{cur}) are, respectively, greater than NSC_{min} and SC_{min}, the number of message copies a nonsocial and a social node can generate is calculated by (1) and (2) as follows:

$$CPN_{\text{nso}} = \frac{NSC_{\text{cur}}}{NSC_{\text{cur}} + SC_{\text{cur}}} \times CPN_{\max} \quad (1)$$

$$CPN_{\text{so}} = \frac{SC_{\text{cur}}}{NSC_{\text{cur}} + SC_{\text{cur}}} \times CPN_{\max}. \quad (2)$$

2) $NSC_{\text{cur}} \leq NSC_{\text{min}}$ and $SC_{\text{cur}} \leq SC_{\text{min}}$: If the values of NSC_{cur} and SC_{cur} are not greater than CPN_{nso} and CPN_{so}, then NSC_{cur} and SC_{cur} are set to 0. This implies that if the credit value is less than the threshold value, the node cannot replicate its messages, and the source node can only deliver its buffered messages to the destination node directly.

3) If the value of NSC_{cur} or SC_{cur} is greater than the minimum value, the maximum value between NSC_{cur} and SC_{cur} is assigned to $CPN_{\max}$ as the number of copies, whereas the smaller value is set to 0. If the nodes need to replicate their messages, they are encouraged to forward messages for others to gain credit for their own messages.

When the number of message copies is identified, the corresponding credit is decreased by $CPN \times \alpha$, where CPN denotes the summation of the number of messages copies consisting of CPN_{nso} and CPN_{so}. When the value of one credit is less than the threshold value, the number of message copies is set to 0. It implies that the other nodes cannot relay messages forwarded for this node since no credit can be provided by this node. This mechanism stimulates selfish nodes to cooperate with other nodes in data forwarding to obtain sufficient credit for their own message forwarding.

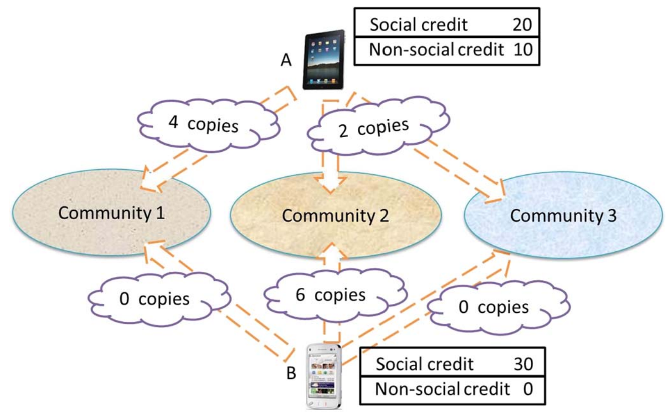


Fig. 4. Example of message replication control in CAIS.

We provide a scenario to describe the implementation of the *message replication control* component, as shown in Fig. 4, where two nodes A and B with different values of social credit and nonsocial credit contact each other. Without loss of generality, we assume that node A belongs to Community 1 and that node B belongs to Community 2. The threshold value of credits for both nodes is set to 0, and the maximum number of message copies is set to 6. The social credit of node A is 20, whereas its nonsocial credit is 10. When node A generates a message, four copies of messages are forwarded by nodes in Community 1, whereas two copies of messages are forwarded by nodes to nodes in other communities (for example, Community 2 or Community 3). As shown in this figure, the values of social and nonsocial credits for node B are 30 and 0, respectively. Thus, node B can generate six copies of messages to be forwarded to Community 2. However, node B cannot forward its messages to other nodes outside its community. Based on this scenario, it can be seen that node B has to participate in relaying messages for nodes in other communities to obtain nonsocial credit. Thus, the selfish behavior of node B can be tolerated, which dramatically affects the performance of data routing. Algorithm 1 presents the message replication control in CAIS.

Algorithm 1 Pseudocode of Message Replication Control

// When node M creates a message
if $NSC_{\text{cur}} > SC_{\min}$ and $SC_{\text{cur}} > SC_{\min}$ **then**
  $CPN_{\text{nso}} = (NSC_{\text{cur}}/(NSC_{\text{cur}} + SC_{\text{cur}})) \times CPN_{\max}$
  $CPN_{\text{so}} = (SC_{\text{cur}}/(NSC_{\text{cur}} + SC_{\text{cur}})) \times CPN_{\max}$
  $NSC_{\text{cur}} -= CPN_{\text{nso}} \times \alpha$
  $SC_{\text{cur}} -= CPN_{\text{so}} \times \alpha$
else if $NSC_{\text{cur}} > NSC_{\min}$ **then**
  $CPN_{\text{nso}} = CPN_{\max}$
  $CPN_{\text{so}} = 0$
  $NSC_{\text{cur}} -= CPN_{\max} \times \alpha$
else if $SC_{\text{cur}} > SC_{\min}$ **then**
  $CPN_{\text{so}} = CPN_{\max}$
  $CPN_{\text{nso}} = 0$
  $SC_{\text{cur}} -= CPN_{\max} \times \alpha$
else if $NSC_{\text{cur}} \leq NSC_{\min}$ and $SC_{\text{cur}} \leq SC_{\min}$ **then**
  $CPN_{\text{so}} = 0$
  $CPN_{\text{nso}} = 0$
end if

B. Forwarding Strategy

In CAIS, a source node replicates its messages to nodes that have higher contact probability with destination nodes. In addition, the intermediate nodes cannot replicate the relaying messages to other intermediate nodes due to the single-copy data replication policy in CAIS. Consequently, they have to either deliver the messages to their destination or drop them. In case an intermediate node does not contact the destination of its buffered message node, the messages in its buffer are expired and deleted, or they are replaced with other incoming messages.

The list of candidate messages in each node consists of two parts. The first part includes messages that should be delivered to their destinations directly and cannot be replicated to intermediate nodes (one-copy message list). The second part contains the list of messages that can be replicated to other encountered nodes (multicopy message list).

The *forwarding strategy* module identifies which messages should be added to the forwarding list. For example, when two nodes M and N contact each other, they exchange the list of their candidate messages to determine their final forwarding list. For the one-copy list of messages in node N, node M only selects messages destined to itself. The multicopy messages are selected based on the following conditions.

1) For each message in the message list of node N, if node M is the destination node (i.e. $D_{\text{msg}} == M$), the message is added to the forwarding list.
2) If $CPN_{\text{so}} == 0$, node N cannot forward her messages to other nodes inside her community. If $CPN_{\text{nso}} == 0$, node N cannot forward her messages to nodes in other communities. Note that CPN_{so} and CPN_{nso} cannot be 0 at the same time.
3) In case nodes M and N are in the same community and $CPN_{\text{so}} \neq 0$ for node N, if nodes M and N belong to different communities and $CPN_{\text{nso}} \neq 0$ for node N, the messages in node N can be forwarded to node M based on a contact prediction algorithm. In this paper, we utilize node contact history to predict the future contacts among nodes. For each message, if the probability of contacting the destination of the message by node M is higher than that of node N [i.e., $P(M, D_{\text{msg}}) > P(N, D_{\text{msg}})$], the message is forwarded to node M.

In summary, a message is added to the forwarding list of node N to be forwarded to node M based on the following conditions.

1) $D_{\text{msg}} == M$.
2) $D_{\text{msg}} \neq M$ and $CPN_{\text{nso}} \neq 0$, under the circumstances that nodes M and N are not in the same community and $P(M, D_{\text{msg}}) > P(N, D_{\text{msg}})$.
3) $D_{\text{msg}} \neq M$ and $CPN_{\text{so}} \neq 0$, under the circumstances that nodes M and N are in the same community and $P(M, D_{\text{msg}}) > P(N, D_{\text{msg}})$.

The forwarding strategy is described in Algorithm 2.

Algorithm 2 Pseudocode of Forwarding Strategy

//Node M contacts node N
//M and N exchange the list of their messages
for each msg in the message list of N **do**
  if $(CPN_{\text{nso}} == 0)$ *and*
    $(CPN_{\text{so}} == 0)$ *and*
    $(D_{\text{msg}} \neq M)$ **then**
    continue
  end if
  if $(CPN_{\text{nso}} == 0)$ *and*
    $(D_{\text{msg}} \neq M)$ *and*
    (*M and N belong to the same community*) **then**
    continue
  end if
  if $(CPN_{\text{so}} == 0)$ *and*
    $(D_{\text{msg}} \neq M)$ *and*
    (*M and N belong to the same community*) **then**
    continue
  end if
  if $P(M, D_{\text{msg}}) > P(N, D_{\text{msg}})$ **then**
    Add msg to the forwarding list
  end if
end for

C. *Selfishness Adjustment*

We aim to retain a balance among node cooperation, resource consumption, and the message delivery requirements in CAIS. The node pays a particular amount of its credit to another encountered node to relay its messages. Meanwhile, she can gain credit by relaying messages for other nodes. Nevertheless, a selfish node in CAIS can exhibit different behaviors according to its requirements. If a node is selfish and has enough credit to forward its messages, it can behave selfishly to save resources. It should be noted that a node with selfish behavior in data relaying can transfer messages partly. In this case, messages destined to this node are accepted. Otherwise, if the node does not have enough credit for its own forwarding requirement, it has to participate in message forwarding with other nodes to obtain enough credit and satisfy its message forwarding demand.

The *selfishness adjustment* module identifies the selfish behavior of a node according to its current credit. For example, the consumption of nonsocial credit to carry n messages can be calculated by: $\sum_{i=1}^{n} CPN_{\mathrm{nso}_i} \times \alpha$, where CPN_{nso_i} represents the number of copies of msg_i for nonsocial nodes. Moreover, α stands for the amount of credit each node should pay for replicating message copy.

The gaining of credit can be calculated as $RLN_{\mathrm{nso}} \times \beta$, where RLN_{nso} denotes the sum of relaying messages for nonsocial nodes, and β indicates the gaining of credit for relaying a message copy.

The current value of nonsocial credit NSC_{cur} can be calculated by

$$NSC_{\mathrm{init}} - \sum_{i=1}^{n} CPN_{\mathrm{nso}_i} \times \alpha + RLN_{\mathrm{nso}} \times \beta. \quad (3)$$

We note that the current value of nonsocial credit should be greater than NSC_{min}. Thus, RLN_{nso} should satisfy

$$RLN_{\mathrm{nso}} > \frac{NSC_{\mathrm{min}} - NSC_{\mathrm{init}} + \sum_{i=1}^{n} (CPN_{\mathrm{nso}_i} \times \alpha)}{\beta}. \quad (4)$$

If (4) is satisfied, multiple copies of a message can be generated by this node. Otherwise, if the node has not relayed enough messages, the number of messages she can generate is 0. In other words, other nodes will not help relay messages forwarded by this node. To deal with this situation, the node with selfish behavior is stimulated to relay messages for other nodes to gain a sufficient amount of credit.

Some nodes may willingly relay messages for other nodes with strong social relationships belonging to their own community. We adjust the willingness degree of a node to relay messages for other nodes inside and outside her community by assigning values NSC_{min} and SC_{min}. Thus, a node can have selfish behavior with nodes in her own community. The stimulation strategy designed for the socially selfish nodes in CAIS is similar with the mechanism designed for the individually selfish nodes that is based on (5) and (6), shown below. Thus

$$SC_{\mathrm{init}} - \sum_{i=1}^{n} CPN_{\mathrm{so}_i} \times \alpha + RLN_{\mathrm{so}} \times \beta > SC_{\mathrm{min}} \quad (5)$$

$$RLN_{\mathrm{so}} > \frac{SC_{\mathrm{min}} - SC_{\mathrm{init}} + \sum_{i=1}^{n} (CPN_{\mathrm{so}_i} \times \alpha)}{\beta}. \quad (6)$$

In summary, a node can perform selfish behavior to nodes in her community if the following conditions are simultaneously satisfied.

1) A node is selfish.
2) $$RLN_{\mathrm{nso}} > \frac{NSC_{\mathrm{min}} - NSC_{\mathrm{init}} + \sum_{i=1}^{n} CPN_{\mathrm{nso}_i} \times \alpha}{\beta}.$$
3) $NSC_{\mathrm{cur}} > NSC_{\mathrm{min}}$.

Accordingly, a node can show selfish behavior to nodes in other communities if the following conditions are satisfied at the same time.

1) A node is selfish.
2) $$RLN_{\mathrm{so}} > \frac{SC_{\mathrm{min}} - SC_{\mathrm{init}} + \sum_{i=1}^{n} (CPN_{\mathrm{so}_i} \times \alpha)}{\beta}.$$
3) $SC_{\mathrm{cur}} > SC_{\mathrm{min}}$.

D. *Message Priority Control*

Due to the short contact time among nodes in opportunistic environments, the transmission priority of messages can affect the routing performance [34]. Generally, an encountered node with a high contact probability with the destination of a message is selected as the next intermediate node. Hence, CAIS utilizes the *message priority control* module to select nodes with higher contact probability. In this module, selfish nodes need to consider their current credits, while a lower amount of credit may be compensated by changing the transmission priority of messages. If both social and nonsocial credits possess lower values than their thresholds, CAIS takes the compensation for social credit first. Thus, node M determines the priority of its forwarding messages as follows.

1) For node M, if $SC_{\mathrm{cur}} < SC_{\mathrm{min}}$, social credit requires to be compensated. Thus, messages from nodes that have a social relationship with node M are encouraged to be transmitted by assigning high forwarding priorities.
2) For node M, if $NSC_{\mathrm{cur}} < NSC_{\mathrm{min}}$, nonsocial credit needs to be compensated. Therefore, messages from nodes with a nonsocial relationship with node M should be conveyed by allocating high priorities.
3) In other conditions, messages with high contact probabilities are accompanied with high priority.

E. *Credit Distribution*

The *credit distribution* rewards the cooperative nodes and updates the social and nonsocial credits. By this module, an intermediate node obtains β credit under the following conditions.

1) A message is delivered to its destination.
2) A message is expired.
3) A message is deleted and replaced with a new incoming new message (buffer replacement policy).

V. Performance Evaluation

We evaluate the performance of CAIS using the Opportunistic Network Environment (ONE) simulator [35], which is a trace-driven and event-based simulator specifically developed in Java for evaluation and analyzing DTN protocols. ONE supports both synthetic movement models and real-world traces to implement routing protocols. It includes some ready-to-use and parameterized mobility models and several configurable well-known DTN routing. In addition, it includes an interactive visualization and postprocessing module support evaluating experiments, and an emulation mode allows the ONE simulator to become part of a real-world DTN testbed.

We have two goals in the experiments. First, we intend to evaluate the efficiency of CAIS with respect to different evaluation metrics. Second, we plan to compare its performance with two benchmark mechanisms using both synthetic and trace-based simulations. Here, we first give a brief introduction of the experiment setup, followed by the performance comparison and discussion.

A. Experiment Setup

Here, we explain the synthetic and trace-based simulations' setup. Each simulation is run ten times with different random seeds to provide results with high confidence. Then, we describe the performance metrics.

Two types of mobile nodes coexist in the network:

- **cooperative node (CN)**, which stores and relays messages on behalf of other nodes to increase network utility;
- **selfish node (SN)**, which does not relay messages transmitted from other nodes without appropriate incentives.

1) Synthetic Simulation Setup: We set up a wireless network with a fixed number of mobile nodes, which are randomly distributed in a terrain area 2 km $\times$ 2 km. The nodes are divided into four communities that communicate with each other via Bluetooth technology. The transmission range of each node is 250 m. To handle node mobility, the location coordinations of nodes are generated using a community-based movement model every time interval. The buffer size of each node is set to 10 MB, and $CPN_{\max} = 6$.

Nodes move inside their communities most of the time and travel to other communities for a specific and short time period. The movement speed of each node changes between 2 and 5 m/s. In the network, a node randomly stays in a location between 100 and 200 s and moves to a new location afterward. There are some areas called *sensitive regions*, which are visited by community nodes more frequently. The mobility model generates more location coordinates for community members inside a sensitive region by adjusting a location proportion parameter. For example, if the location proportion parameter is set to 60%, it means that 60% of the locations are generated in sensitive regions. We apply Prophet routing to route messages in the network, where its parameters are set as $P\text{init} = 0.75$, $\beta = 0.25, \gamma = 0.98$. $P(M, D_{\text{msg}})$ is set the same with the Prophet routing algorithm. Furthermore, NSC_{init} and SC_{init} are set to 300, where $NSC_{\min}$ and $SC_{\min}$ are set to 0. Similar parameter settings can be found in [28].

TABLE II
Properties of the Data Sets

Data set	MIT Reality	Social Evolution
Device	Nokia 6600	Smartphone
Year	2004	2009
No. nodes	106	80
Duration (day)	246	240
Granularity (sec)	300	360
No. Bluetooth contacts	1259148	2124565
Avg. Blutooch contacts	5118.45	8852.35

Considering the social properties of nodes, we assign the parameters of the network model as follows. Messages are generated every 4 min based on a message proportion parameter to establish a message transmission pattern among nodes. The message proportion parameter controls the proportion of messages destined inside and outside a community.

2) Trace-Based Simulation Setup: In addition to the synthetic simulations, we use two real-world data sets, i.e., MIT Reality [36] and Social Evolution [37], to further evaluate and compare the performance of our proposed scheme. The corresponding properties of the data sets can be seen on Table II. We use Bluetooth data in each data set to assign the contact time and duration of the nodes. The properties of the data sets can be described as follows.

MIT Reality (MIT): This data set includes the contact information of 106 participants that carry Nokia 6600 smartphones at MIT for nine months. Through processing the data, we filter the contact information of 88 participants with sufficient Bluetooth encounters between January and April 2005. In this data set, we utilize the affiliation information of nodes to identify the social relationships among nodes and form the network communities.

Social Evolution (SocEvo): This data set consists of the contact information of 80 undergraduate students who carry their cell phones at MIT for eight months. We filter the information of 74 participants with sufficient Bluetooth encounters between January and April 2009 to apply in the experiments. In this data set, we employ users' interest to establish social relationships among nodes and form the network communities.

In the set of real-trace-based experiments, we consider a wireless DTN where the transmission speed of the Bluetooth interface is set to 5 Mb/s. Messages are generated with a random size between 500 kB and 1 MB, with a uniform interval of 15$\sim$20. In addition, the TTL of messages varies between 1 day and 30 days, and the buffer capacity of each node varies between 2 and 30 MB. Messages are generated at a fixed time interval. The buffer size of nodes is set to 10 MB. By processing the real-trace-based data sets, three main features are selected to construct individual interest: 1) Bluetooth contact frequency; 2) visited locations (captured from wireless local area networks); and 3) call and SMS logs. Since community detection is not the main concentration of this paper, we use a distributed community detection algorithm based on K-means to form communities.

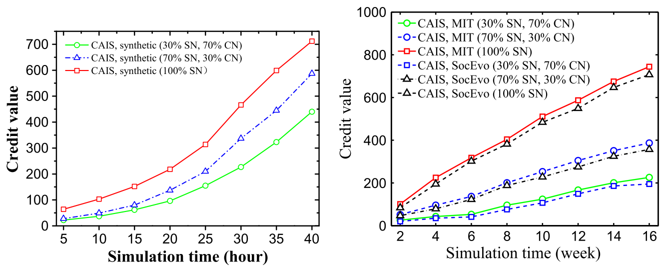


Fig. 5. Variation of the credit value of a sample node (node 4) during the simulation time. (a) Synthetic simulations. (b) Real trace simulations.

3) Performance Metrics and Benchmark Mechanisms: We consider four metrics in our simulations:

- **Node credit**: the cumulative amount of credit received by a node for relaying incoming messages, which includes both social and nonsocial credits, which metric reflects the effectiveness of a credit-based system;
- **Average delivery ratio**: the number of delivered messages to the total number of generated messages;
- **Average delivery delay**: the average time that it takes a message to be delivered to its destination;
- **Average delivery cost**: the total number of replicated messages (including control messages, social information, and dropped messages) to the number of delivered messages.

We compare CAIS against the following protocols:

Prophet-Selfish: a revised version of Prophet routing [33], in which CN and SN interact with each other in data delivery without an incentive mechanism.

Social Selfishness Aware Routing (SSAR) [14]: a socially selfish aware routing, in which a selfish node forwards messages for those with strong social ties. In SSAR, the selfish node gives priority to messages received from those with stronger social ties when the buffer space is not sufficient.

B. Simulation Results

We then make a comparison among different schemes according to various kinds of metrics in both synthetic and real-trace-based simulations.

Node Credit Analysis: Here, we analyze the fluctuation of the node credit through the simulation time. The message TTL is set to 5 h and 15 days for synthetic and real-trace-based simulations, respectively. Note that we only consider the gained credit in these experiments.

The variation of the credit value is shown in Fig. 5. The credit value of a selfish node, e.g., node 4, is compared under the situation that the initial values of both social credit and nonsocial credit are set to 300. We can observe that for both synthetic and real-trace-based simulations, the credit value increases as the value of TTL enhances. This is because as the value of TTL increases, messages can stay longer in the network; thus, more cooperation can be conducted. We also note that a higher credit value can be achieved by a larger number of selfish nodes; this is because selfish nodes would mitigate their selfishness level based on their social information and interests to maximize their social benefits. This phenomenon is particularly obvious in the real-trace-based simulation, since the time settings for both TTL and simulation are longer.

Impact of Message TTL: Then, we study the routing performances (i.e., average delivery ratio, delay, and cost) when the value of message TTL varies. According to the results shown in Fig. 6(a), it can be seen that the CAIS scheme achieves the highest delivery ratio. The reasons are that, on one hand, our method stimulates selfish nodes to transmit data in a cooperative manner to nodes inside their own community and to outsiders. On the other hand, it considers node requirements by constructing the interactions between their demands for high delivery efficiency and cooperation degrees. However, the other two methods do not study the effects brought by the selfish node comprehensively.

Fig. 6(b) and (c) shows the obtained message delivery delay and cost by different methods. As the message TTL increases, the data delivery delay and delivery cost continuously grow because network nodes drop the message that is beneficially low. Comparatively, it can be seen that CAIS has minimum delay and cost. For example, when the TTL is 30 h, the average delay of CAIS is 9% and 28% lower than SSAR and Prophet-Selfish, respectively. In addition, the corresponding cost of CAIS is approximately 17% and 40% lower than SSAR and Prophet-Selfish, respectively. It should be noted that although the average delivery ratio increases as the message TTL enhances, the average delay and cost still add.

In the trace-based simulations, as shown in Fig. 6(d), it demonstrates that as the number of TTL increases, the delivery ratios of CAIS and SSAR gradually grow, whereas the delivery ratio of Prophet-Selfish manifests a downward tendency. When the TTL is four weeks, CAIS outperforms SSAR and Prophet-Selfish by around 15% and 33%, respectively. The main reason is that CAIS can effectively stimulate the selfish nodes to forward messages for others. In contrast, nodes in SSAR make relaying decisions according to their willingness level that is determined by their contact history. The trends in Fig. 6(e) and (f) are almost the same as those in the synthetic simulations in Fig. 6(b) and (c).

Impact of Different Percentage of Selfish Nodes: The obtained network properties of different algorithms are compared in Fig. 7, in case 25%, 50%, 75%, and 100% of nodes are selfish. For delivery ratio, the gap of the network performances gained by CAIS and Prophet-Selfish is not too obvious, and these two methods largely outperform the SSAR mechanism, as shown in Fig. 7(a) and (d). The reason behind is that nodes in the SSAR-based mechanism decide on the relaying operation according to the contact history and cannot adjust node selfishness dynamically. When the percentage of selfish nodes is 25%, the data delivery ratios of CAIS and Prophet-Selfish are 0.85 and 0.78, respectively, which are approximately 40% and 30% higher than SSAR for the synthetic simulations, respectively. A similar conclusion can also be drawn in the trace-based simulations.

Fig. 7(b) and (e) shows the average delivery delay of different protocols. It demonstrates that the latency of SSAR is much higher than that of CAIS and Prophet-Selfish schemes.

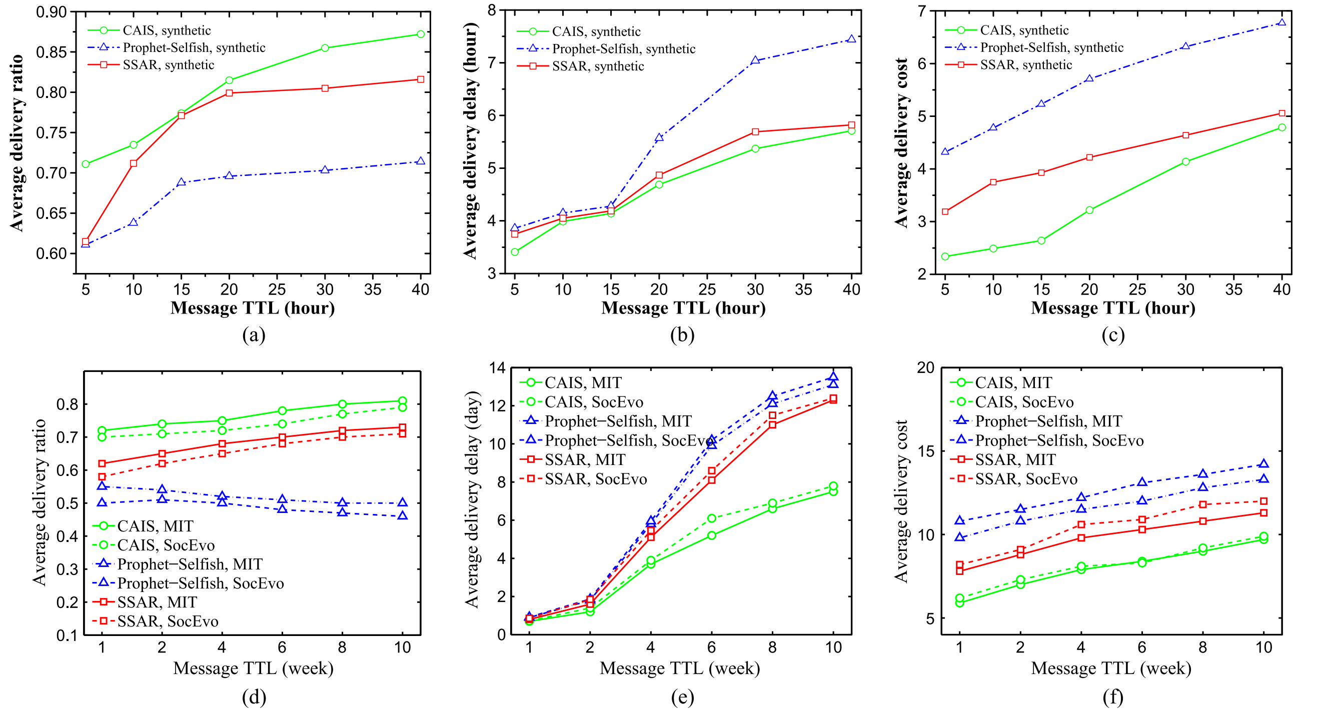


Fig. 6. Performance comparison of the algorithms with changing the message TTL when 70% of nodes are selfish. (a) Delivery ratio—synthetic simulations. (b) Delivery delay—synthetic simulations. (c) Delivery cost—synthetic simulations. (d) Delivery ratio—trace-based simulations. (e) Delivery delay—trace-based simulations. (f) Delivery cost—trace-based simulations.

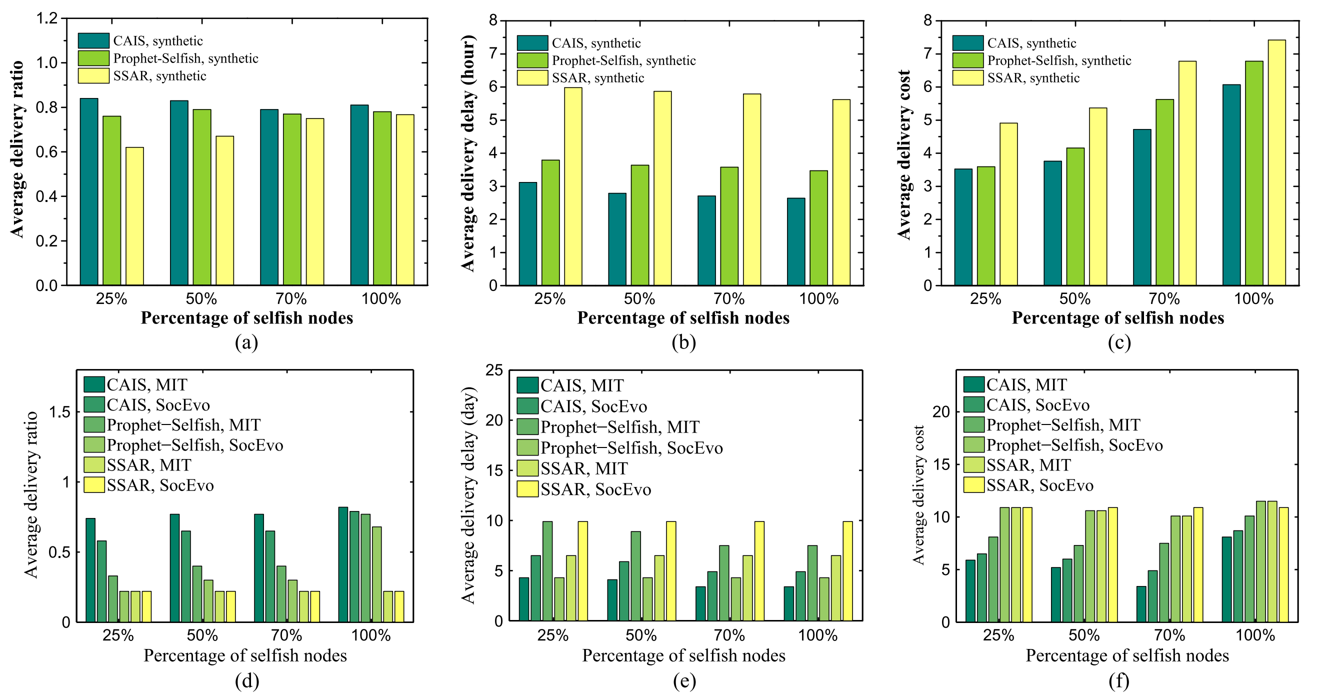


Fig. 7. Performance comparison of the algorithms with different ratios of selfish nodes. (a) Delivery ratio—synthetic simulations. (b) Delivery delay—synthetic simulations. (c) Delivery cost—synthetic simulations. (d) Delivery ratio—trace-based simulations. (e) Delivery delay—trace-based simulations. (f) Delivery cost—trace-based simulations.

This is because the nodes in SSAR rather deny forwarding messages for others if their social relationship is not strong enough. In SSAR, it is generally assumed that nodes in the same community are willing to cooperate with each other. Therefore, the proportion of selfish nodes in this protocol is invariably small. However, in CAIS, it is feasible for nodes to act selfishly when they have enough credit for data forwarding; thus, better network performances can be gained through the

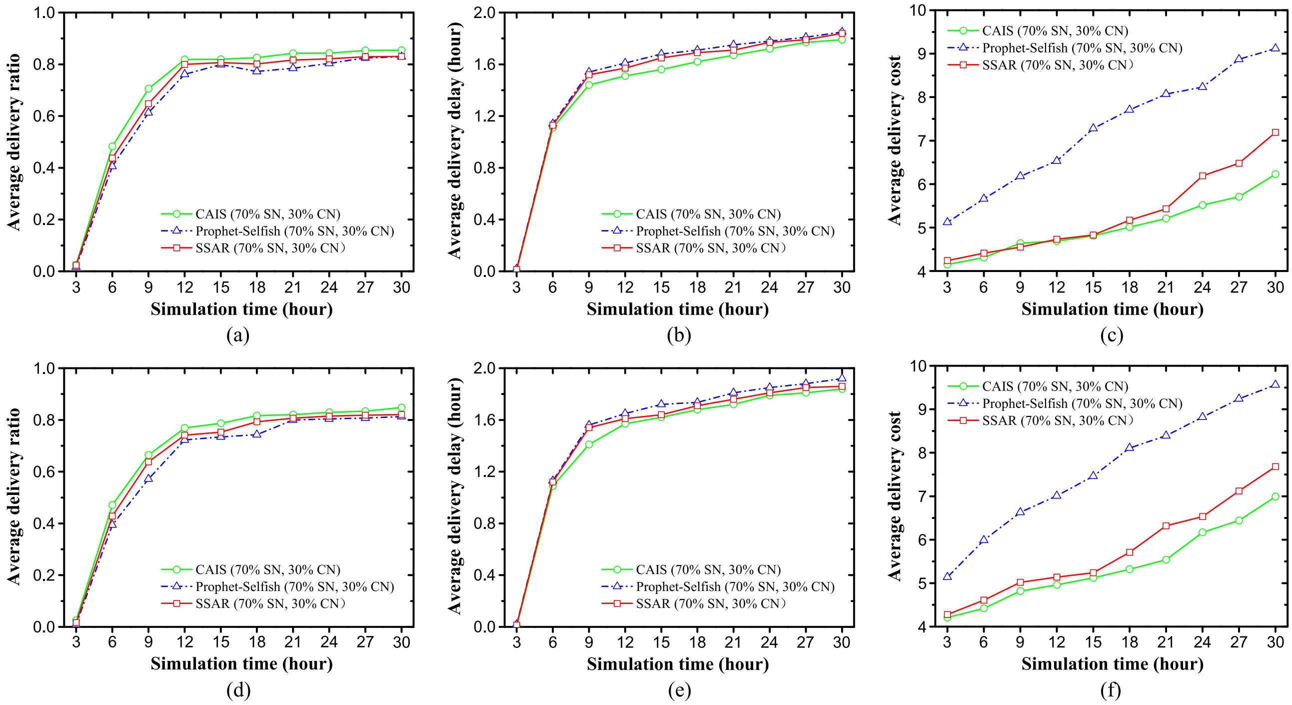


Fig. 8. Performance comparisons of the algorithms with different social properties based on the synthetic simulations. (a) Delivery ratio—L30-M30. (b) Delivery delay—L30-M30. (c) Delivery cost—L30-M30. (d) Delivery ratio—L30-M50. (e) Delivery delay—L30-M50. (f) Delivery cost—L30-M50.

dynamic adjustment. As shown in Fig. 7, we can conclude that CAIS tolerates a large amount of selfish nodes by encouraging them to cooperate with other nodes in message delivery.

Influence of Social Properties: Here, we analyze the influence of social properties on CAIS, as shown in Fig. 8. The social properties consist of two aspects: location proportion and message proportion. Each node belongs to a community with a sensitive region. The location proportion indicates how often a node visits the sensitive region. Thus, the higher the location proportion is, the more frequently nodes inside a community contact each other. Similarly, the message proportion indicates how often a node transfers messages to other nodes inside the community.

We consider two scenarios, i.e., 30% of locations with 30% messages (L30-M30) and 30% of locations with 50% messages (L30-M50), to evaluate the impact of social properties on CAIS. It is shown in Fig. 8(a) and (d) that CAIS obtains the highest delivery ratio, which shows that CAIS alleviates the effect brought by social selfishness efficiently. Fig. 8(b) and (e) shows that Prophet-Selfish has the longest delivery delay, whereas CAIS and SSAR perform neck and neck as the simulation time gets longer. In Fig. 8(c) and (f), it is also shown that CAIS has the lowest cost in comparison with the other algorithms. For example, when the simulation hour is 24 h, the cost in CAIS is 10% and 30% lower than that in SSAR and Prophet-Selfish, respectively. In summary, we conclude that the performance of CAIS is highly affected by the social properties.

VI. Conclusion

In this paper, we have proposed a credit-based incentive mechanism to stimulate selfish nodes in message forwarding in community-based SANs. Considering the social relationships among nodes, we used social and nonsocial credits to reward nodes when they respectively relay messages received from nodes in their own community and other communities. Furthermore, we employed a single-copy-based message replication mechanism in CAIS to distribute credit among mobile nodes without a third-party authority center. Extensive simulations were conducted to evaluate the performance of CAIS with Prophet-Selfish and SSAR protocols with respect to data delivery ratio, delay, and network cost based on both synthetic and real-trace-based simulations. Our evaluation results demonstrated that CAIS tolerates quite a large amount of selfish nodes by efficiently stimulating them to cooperate with other nodes in message delivery. Our work has several extensions. For example, nodes are assumed to perform in a trust manner, and we intend to make a tradeoff between network performance improvement and system cost by adding third-party supervision. Furthermore, a community detection strategy would be studied to identify the community more accurately.

References

[1] Z. Wang, J. Liao, Q. Cao, H. Qi, and Z. Wang, "Achieving k-barrier coverage in hybrid directional sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 7, pp. 1443–1455, Jul. 2014.

[2] S. Tornell, C. Calafate, J. Cano, and P. Manzoni, "DTN protocols for vehicular networks: An application oriented overview," *IEEE Commun. Surveys Tut.*, vol. 17, no. 2, pp. 868–887, 2nd Quart. 2014.

[3] J. Isento, J. Rodrigues, J. Dias, M. Paula, and A. Vinel, "Vehicular delay-tolerant networks—A novel solution for vehicular communications," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 4, pp. 10–19, Apr. 2013.

[4] J. Dias, J. Rodrigues, N. Kumar, and K. Saleem, "Cooperation strategies for vehicular delay-tolerant networks," *IEEE Commun. Mag.*, vol. 53, no. 12, pp. 88–94, Dec. 2015.

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

[6] F. Mezghani, R. Dhaou, M. Nogueira, and A.-L. Beylot, “Content dissemination in vehicular social networks: Taxonomy and user satisfaction,” *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 34–40, Dec. 2014.

[7] H. Zhou, J. Chen, H. Zhao, W. Gao, and P. Cheng, “On exploiting contact patterns for data forwarding in duty-cycle opportunistic mobile networks,” *IEEE Trans. Veh. Technol.*, vol. 62, no. 9, pp. 4629–4642, Nov. 2013.

[8] T. Qiu, D. Luo, F. Xia, N. Deonauth, W. Si, and A. Tolba, “A greedy model with small world for improving the robustness of heterogeneous Internet of things,” *Comput. Netw.*, vol. 101, pp. 127–143, 2016.

[9] Y. Zhu, B. Xu, X. Shi, and Y. Wang, “A survey of social-based routing in delay tolerant networks: Positive and negative social effects,” *IEEE Commun. Surveys Tut.*, vol. 15, no. 1, pp. 387–401, 1st Quart. 2013.

[10] Z. Ning, L. Guo, Y. Peng, and X. Wang, “Joint scheduling and routing algorithm with load balancing in wireless mesh network,” *Comput. Electr. Eng.*, vol. 38, no. 3, pp. 533–550, 2012.

[11] Z. Ning, Q. Song, L. Guo, and K. Okamura, “Social-oriented adaptive transmission in wireless ad hoc networks,” in *Proc. IEEE ICC*, 2014, pp. 490–495.

[12] R. Diana and E. Lochin, “Modelling the delay distribution of binary spray and wait routing protocol,” in *Proc. IEEE WoWMoM*, 2012, pp. 1–6.

[13] Z. Wang, J. Liao, Q. Cao, H. Qi, and Z. Wang, “Friendbook: A semantic-based friend recommendation system for social networks,” *IEEE Trans. Mobile Comput.*, vol. 14, no. 3, pp. 538–551, Mar. 2015.

[14] Q. Li, W. Gao, S. Zhu, and G. Cao, “A routing protocol for socially selfish delay tolerant networks,” *Ad Hoc Netw.*, vol. 10, no. 8, pp. 1619–1632, 2012.

[15] K. Wei, X. Liang, and K. Xu, “A survey of social-aware routing protocols in delay tolerant networks: Applications, taxonomy and design-related issues,” *IEEE Commun. Surveys Tut.*, vol. 16, no. 1, pp. 556–578, 1st Quart. 2014.

[16] 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.

[17] W. Moreira, P. Mendes, and S. Sargento, “Opportunistic routing based on daily routines,” in *Proc. IEEE Int. Symp. WoWMoM*, 2012, pp. 1–6.

[18] F. Xia, L. Liu, B. Jedari, and S. Das, “A multi-dimensional routing protocol for socially-aware networking,” *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2825–2836, Nov. 2016.

[19] Y. Li, G. Su, D. Wu, D. Jin, L. Su, and L. Zeng, “The impact of node selfishness on multicasting in delay tolerant networks,” *IEEE Trans. Veh. Technol.*, vol. 60, no. 5, pp. 2224–2238, Jun. 2011.

[20] Y. Li, P. Hui, D. Jin, L. Su, and L. Zeng, “Evaluating the impact of social selfishness on the epidemic routing in delay tolerant networks,” *IEEE Commun. Lett.*, vol. 14, no. 11, pp. 1026–1028, Nov. 2010.

[21] A. Mei and J. Stefa, “Give2Get: Forwarding in social mobile wireless networks of selfish individuals,” *IEEE Trans. Depend. Sec. Comput.*, vol. 9, no. 4, pp. 569–582, Jul./Aug. 2012.

[22] L. Buttyn, L. Dora, M. Felegyhazi, and I. Vajda, “Barter trade improves message delivery in opportunistic networks,” *Ad Hoc Netw.*, vol. 8, no. 1, pp. 1–14, 2010.

[23] H. Zhou, J. Chen, J. Fan, Y. Du, and S. Das, “ConSub: Incentive-based content subscribing in selfish opportunistic mobile networks,” *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 669–679, Sep. 2013.

[24] G. Bigwood and T. Henderson, “IRONMAN: Using social networks to add incentives and reputation to opportunistic networks,” in *Proc. IEEE Socialcom*, 2011, pp. 65–72.

[25] T. Chen, F. Wu, and S. Zhong, “FITS: A finite-time reputation system for cooperation in wireless ad hoc networks,” *IEEE Trans. Comput.*, vol. 60, no. 7, pp. 1045–1056, Jul. 2011.

[26] T. Ning, Z. Yang, X. Xie, and H. Wu, “Incentive-aware data dissemination in delay-tolerant mobile networks,” in *Proc. IEEE 8th Annu. SECON*, 2011, pp. 539–547.

[27] Y. Wang, M. Chuah, and Y. Chen, “Incentive based data sharing in delay tolerant mobile networks,” *IEEE Trans. Wireless Commun.*, vol. 13, no. 1, pp. 370–381, Jan. 2014.

[28] F. Wu, T. Chen, S. Zhong, C. Qiao, and G. Chen, “A bargaining-based approach for incentive-compatible message forwarding in opportunistic networks,” in *Proc. IEEE ICC*, 2012, pp. 789–793.

[29] T. Seregina, O. Brun, R. Azouzi, and B. Prabhu, “On the design of a reward-based incentive mechanism for delay tolerant networks,” *IEEE Trans. Mobile Comput.*, vol. 16, no. 2, pp. 453–465, Feb. 2017.

[30] Y. Cai, Y. Fan, and D. Wen, “An incentive-compatible routing protocol for two-hop delay-tolerant networks,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 1, pp. 266–277, Jan. 2016.

[31] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Efficient routing in intermittently connected mobile networks: The single-copy case,” *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 63–76, Feb. 2008.

[32] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Efficient routing in intermittently connected mobile networks: The multi-copy case,” *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 77–90, Feb. 2008.

[33] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic routing in intermittently connected networks,” *ACM SIGMOBILE Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, 2003.

[34] Z. Ning, Q. Song, Y. Huang, and L. Guo, “A channel estimation based opportunistic scheduling scheme in wireless bidirectional networks,” *J. Netw. Comput. Appl.*, vol. 39, no. 6, pp. 61–69, 2014.

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

[36] N. Eagle and A. S. Pentland, “Reality mining: Sensing complex social systems,” *Pers. Ubiquitous Comput.*, vol. 10, no. 4, pp. 255–268, 2006.

[37] A. Madan, M. Cebrian, S. Moturu, K. Farrahi, and A. Pentland, “Sensing the health state of a community,” *IEEE Pervasive Comput.*, vol. 11, no. 4, pp. 36–45, Oct.–Dec. 2012.

Zhaolong Ning (M’14) received the M.S. and Ph.D. degrees from Northeastern University, Shenyang, China, in 2011 and 2014, respectively.

He was a Research Fellow with Kyushu University, Fukuoka, Japan. He is an Assistant Professor with the School of Software, Dalian University of Technology, Dalian, China. His research interests include social computing, network optimization, and big scholar data.

Li Liu received the B.S. and M.S. degrees in computer science and technology from Shandong University of Science and Technology, Qingdao, China, in 2001 and 2004, respectively, and the Ph.D. degree from Dalian University of Technology, Dalian, China, in 2015.

Since 2004, she has been with Shandong Jiaotong University, Jinan, China. Her research interests include opportunistic networks, socially aware networking, and mobile social networks.

Feng Xia (M’07–SM’12) received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2001 and 2006, respectively.

He was a Research Fellow with Queensland University of Technology, Brisbane, Australia. He is currently a Full Professor with the School of Software, Dalian University of Technology, Dalian, China. He has published two books and over 200 scientific papers in international journals and conferences. His research interests include computational social science, big data, and mobile social networks.

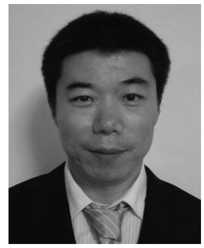
Dr. Xia has been the (Guest) Editor of several international journals. He has served as a General Chair, a Program Committee Chair, a Workshop Chair, and a Publicity Chair for a number of conferences. He is a Senior Member of the Association for Computing Machinery and a member of the American Association for the Advancement of Science.

Behrouz Jedari received the B.Sc. and M.Sc. degrees from the Islamic Azad University, Qazvin, Iran, in 2006 and 2009, respectively. He is currently working toward the Ph.D. degree with the School of Software, Dalian University of Technology, Dalian, China.

His current research interests include delay-tolerant networks, opportunistic communications, and mobile social networks.

Ivan Lee (SM'07) received the B.Eng., MCom, MER, and Ph.D. degrees from The University of Sydney, Sydney, Australia, in 1995, 1997, 1998, and 2005, respectively.

He was a Software Development Engineer with Cisco Systems, a Software Engineer with Remotek Corporation, and an Assistant Professor with Ryerson University, Toronto, ON, Canada. Since 2008, he has been a Senior Lecturer with the University of South Australia, Adelaide, Australia. His research interests include multimedia systems, medical imaging, data analytics, and computational economics.



Weishan Zhang received the Ph.D. degree in mechanical manufacturing and automation from Northwestern Polytechnical University, Xi'an, China, in 2001.

He is a Full Professor and the Deputy Head for Research with the Department of Software Engineering, School of Computer and Communication Engineering, China University of Petroleum, Qingdao, China. His research interests include big data platforms, pervasive cloud computing, and service-oriented computing.