



# An Equivalent Exchange Based Data Forwarding Incentive Scheme for Socially Aware Networks

Zenggang Xiong<sup>1,2</sup> · Nan Xiao<sup>1,2</sup> · Fang Xu<sup>1,2</sup> · Xuemin Zhang<sup>1</sup> · Qiong Xu<sup>1,2</sup> · Kaibin Zhang<sup>3</sup> · Conghuan Ye<sup>1</sup>

Received: 12 September 2020 / Revised: 7 October 2020 / Accepted: 26 October 2020 / Published online: 2 November 2020  
© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

As nodes have limited resources in the socially aware networks, they will have strong selfish behaviors, such as not forwarding messages and losing packets, which will lead to poor network performance. Thus, an equivalent-exchange-based data forwarding incentive scheme (EEIS) will be proposed in this paper. It is main that messages forwarding will be abstracted into a transaction in EEIS. The buyer and seller respectively make a price about the message according to its own resource state and negotiate twice the pricing both side until they agree, then the buyer will send the message and pay a certain virtual currency to the seller. Otherwise, the next message will continue to be traded. Meanwhile, both parties' resource status, wealth status and the price of messages must be open and transparent to prevent the nodes from making false pricing during the transaction. Ultimately, the experimental results show the delivery ratio about messages is improved significantly and verify the effectiveness of EEIS.

**Keywords** Socially aware networking · Selfish nodes · Data forwarding · Virtual currency · Negotiation twice

## 1 Introduction

In recent years, wireless communication and network technologies are developing rapidly, and intelligent mobile devices are quickly connected to the Internet. The explosion of data [1] and the emergence of real-time multiprocessor [2] brings a good beginning to the Socially Aware Networks (SAN). SAN is a new type of network integrating the Delay Tolerant Network (DTN) [3], Cyber-Physical System (CPS) and Social Network Analysis Theory [3–5], which mainly takes the social connections or characteristics between mobile nodes as the basis of network communication.

SAN combines computer science, humanities and social science and cognitive science to promote human-centered intelligent communication by revealing memory regulation [6] of human activities. It can perceive and utilize the context

information of nodes, and provide services for human beings more intelligently through context awareness computing [7–10]. SAN is one of the important development directions of mobile ad hoc network, and it brings new opportunities for the development of computer network [11]. Nowadays, the extensive use of social network applications makes SAN enter the eyes of many researchers in academia and industry [12, 13]. Therefore, in-depth study of SAN is helpful to further promote the development of universal network.

There are still many problems to be solved in SAN, especially designing effective message forwarding algorithms [14–17]. Although nodes' social relations are in a relatively stable and social activities have a certain regularity in the SAN, but due to nodes' resource-constrained and dynamic network topology changes, nodes will not be willing to sacrifice their own resources to forward messages to others, and will prevent routing to work properly. As a result, the transmission success rate will be lower, and the network performance will be degraded. So, the selfishness of the nodes in the network [18–20] is a problem that needs to be solved.

At present, the solution to the selfishness of nodes is mainly to design incentive mechanism to encourage selfish nodes to participate in cooperation. The existing classic incentive mechanism can be divided into three categories: reputation-based incentive mechanism, virtual currency-based incentive mechanism and game-based incentive mechanism. Reputation-based incentive mechanism [21–24] is mainly to

✉ Fang Xu  
xf2012@whu.edu.cn

<sup>1</sup> School of Computer and Information Science, Hubei Engineering University, Xiaogan 432000, Hubei, China

<sup>2</sup> School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, Hubei, China

<sup>3</sup> School of Electronics and Information, Xian Polytechnic University, Xian Shanxi 710048, China

check the behavior of the node according to the reputation table in each node, and judge whether the node can be trusted, then select the node with large reputation value as the forwarding node. The main purpose of incentive mechanism based on virtual currency [25–27] is to virtualize the process of message forwarding into a form of currency transaction. First pricing the message, and then obtaining the corresponding currency after the message forwarding. However, the process of message forwarding in game-based incentive mechanism [28–32] is to turn it into a bargaining process between both parties. Therefore, both parties will maximize their own interests and the nodes can help to forward the message after they both agree the pricing. All the three incentive mechanisms only pursue the cooperation between nodes and do not consider the influence of the nodes' states on the behavior of nodes. Such mechanism not only fails to improve network performance, but also speeds up the consumption of resources. This can make it impossible for nodes to communicate normally, thus further degrading network performance. And traditional currency-based incentive mechanism requires the participation of the third party, which is difficult to achieve without infrastructure [33] and fixed equipment.

Literature [34] has proposed Bargaining based Incentive Protocol (BIP). It is a kind of incentive mechanism based on business model. It mainly considers the message properties, nodes' residual resource and environmental properties such as wealth, then respectively designs the bidding function for the buyer (source node) and the asking price function for the seller (the forwarding nodes). If the seller's asking price is less than or equal to the buyer's bidding, this will suggest that the forwarding node are happy to receive messages and obtain the buyer's offering virtual currency. But if it is larger than the bidding, the forwarding deal will be abandoned and then they will start to trade the next message. However, energy consumption also affects whether a node is willing to forward messages. While the designed bidding function and asking price function both ignore the influence of energy factors on pricing and the opacity of information between the two parties may lead to the problem of node's false offer.

In literature [35], a game-based incentive strategy (GIS) has been proposed, which mainly adopts the three-time bargaining game mechanism without the supervision of a third-party organization, so that both parties can forward messages at a reasonable price and successfully complete the transaction. Although GIS can restrain the problem of false pricing to a certain extent, there are still some problems such as the information of both sides is not open and the currency is insufficient. Moreover, the mechanism of three times bargaining will lead to the high delay of message delivery and excessive energy consumption.

All of the above incentive strategies encourage the nodes to cooperate with each other and optimize the network, but they all have their own problems, so Equivalent Exchange Based

Data Forwarding Incentive Scheme (EEIS) will be proposed in this paper. In this incentive scheme, it comprehensively considers the nodes' resource states such as cache, energy, wealth and message attributes, and the messages pricing function is designed for the nodes, which enables the both parties to negotiate twice. At the same time, the relevant resources information and wealth status of nodes are also disclosed, so as to suppress the false pricing of nodes.

## 2 Network Model and Hypothesis

In this paper, SAN is abstracted as a disconnected directed graph  $G = (v, e)$ , where vertex  $v$  represents a node in the socially aware networking, and edge  $e$  represents the encounter connection between two nodes.

In socially aware networking, the nodes have limited energy and cache. If the energy of nodes is exhausted, the nodes will no longer be able to communicate with each other. Similarly, if the nodes' cache space is used up, the nodes will be unable to receive messages. Therefore, in order to ensure the operation of network, nodes will try their best to save their own resources, resulting in serious packets loss behavior. This fully reflects the selfishness of nodes. As can be seen from Fig. 1, the delivery ratio of messages decreases significantly due to the selfish nodes' joining the network.

Because the "store-carry-forward" method is applied to transmit messages in the socially aware networking, all source nodes forward messages to the destination nodes by mainly moving and meeting with relay nodes according to the specific routing algorithm. In socially aware networking, the routing algorithms can be divided into single-copy routing and multi-copy routing on the basis of the number of messages copies in the network. Multi-copy routing needs a lot of cache and consumes a lot of energy, so Prophet routing algorithm [36] is mainly applied in this paper to explain the proposed incentive scheme.

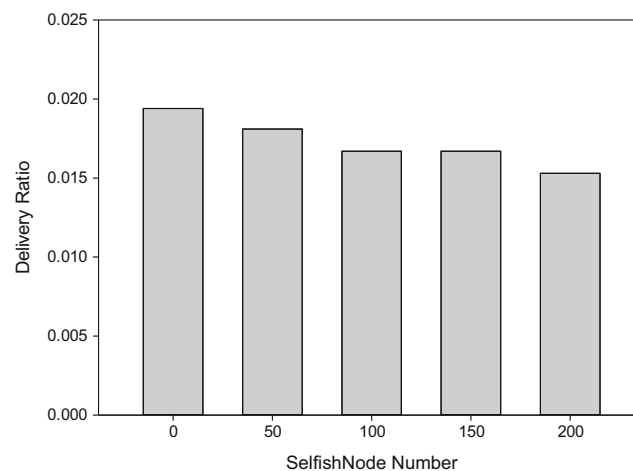


Figure 1 The effect of selfish nodes on message delivery ratio.

The following assumptions are put forward to analyze in this paper:

- 1) If the relay node is a selfish node and not the destination node of messages, it will not receive any messages from any nodes.
- 2) Nodes' selfishness is rational. Nodes will pursue their own maximum interests under the conditions of their own circumstances. At this time, nodes will only consider whether it is beneficial to receive messages at present, but not consider whether there will be problems such as insufficient currency after forwarding messages.
- 3) Selfish nodes are not malicious nodes. They do not intentionally delete messages in their own cache, but treat equally all messages in the cache space. At the same time, selfish nodes will actually bid on the basis of their own situation according to the pricing method in this paper.
- 4) Each node has defined initial currency and initial energy. The node mainly conducts messages forwarding transactions through payment of currency and energy consumption.

### 3 Equivalent Exchange Based Data Forwarding Incentive Scheme

An equivalent-exchange-based data forwarding incentive scheme (EEIS) will be proposed in this paper. On the basis of the virtual currency incentive mechanism, it will mainly add the bargaining model based on equivalent exchange. In EEIS, both sides of the transaction will negotiate the message price according to the node's own resource status and the size of messages. After the price negotiation is completed, the message forwarding process and the transaction process will be carried out. This algorithm is mainly to manage their own virtual currency by the nodes themselves. After each message is forwarded, the sending node will immediately pay the certain currency to the relay node. And it does not require a third-party institution to monitor the messages transaction process.

EEIS is mainly composed of three parts: nodes' resource status, equivalent-exchange model and message forwarding scheme. The resource state is a key factor for the node's message pricing. The equivalent-exchange model is the process of two nodes' negotiation and transaction about messages to be forwarded, and the message forwarding scheme is the process of the node's forwarding the message to the meeting node.

#### 3.1 Resource State

In the network, the resource status is directly related to whether the node is willing to forward the message, which will also affect the respective pricing of the message by both nodes.

Therefore, the data forwarding incentive algorithm mainly considers some relatively important attributes such as the remaining cache space of the node, the remaining energy of the node, the wealth status of the node, and the size of the message.

#### 3.1.1 Node's Remaining Cache Space

The remaining cache space of nodes can be expressed as percentage, which mainly reflects the current storage capacity of nodes. Therefore, the node's remaining cache percentage  $B_{p_i}$  can be seen in Formula (1), where  $i$  denotes the node  $i$ ,  $B_{p_i}$  represents the remaining cache space percentage of node  $i$ ,  $B_{left_i}$  denotes the remaining cache of node  $i$  at the current moment, and  $B_{init_i}$  represents the initial cache space of node  $i$ .

$$B_{p_i} = \frac{B_{left_i}}{B_{init_i}} \tag{1}$$

It can be seen from Formula (1) that the larger the node's remaining buffer space, the larger the node's the storage capacity. So that it is easier for the node to receive messages sent from other nodes.

#### 3.1.2 Node's Remaining Energy

Similarly, the remaining energy of a node can also be expressed as a percentage, which mainly embodies whether the node can forward messages, so the remaining energy percentage  $E_{p_i}$  of node  $i$  is visible in Formula (2). In Formula (2),  $i$  denotes the node  $i$ ,  $E_{p_i}$  represents the remaining energy percentage of node  $i$ ,  $E_{left_i}$  denotes the remaining energy of node  $i$  at the current moment, and  $E_{init_i}$  represents the initial energy of node  $i$ .

$$E_{p_i} = \frac{E_{left_i}}{E_{init_i}} \tag{2}$$

It can be known form Formula (2) that the more remaining energy a node has, the more messages it sends, and the easier the node cooperates with other nodes.

#### 3.1.3 Node's Wealth

The node's wealth denotes the amount of virtual currency held by the current node. Nodes will decide whether they are willing to forward messages to other nodes according to the amount of currency they currently own. It can be known from the Formula (3) that the wealth  $R_i$  of node  $i$  can be divided into three states: rich, general and poor. When  $V_i(t) < V_{poor\_min}$ , the wealth status of node  $i$  is poor; when, the wealth status of node  $i$  is rich; otherwise, the wealth status of node  $i$  is general state.

$$R_i = \begin{cases} \frac{V_i(t)}{V_{poor_{min}}}, & \text{Poor Status} \\ 1, & \text{General Status} \\ \frac{V_i(t)}{V_{rich_{max}}}, & \text{Rich Status} \end{cases} \quad (3)$$

In Formula (3),  $V_i(t)$  is the amount of virtual currency held by node  $i$  at time  $t$ ,  $V_{poor_{min}}$  denotes the threshold of the minimum amount of virtual currency owned by node  $i$  when the wealth status of node  $i$  is poor.  $V_{rich_{max}}$  represents the threshold of maximum amount of virtual currency owned by node  $i$  when the wealth status of node  $i$  is rich. And the condition  $V_{rich_{max}} > V_{poor_{min}}$  must be satisfied.

### 3.1.4 Message Size

At all nodes in the network, the message size is proportional to the energy consumption and the use of cache space. The larger the message size, the more the consumed energy, and the larger the occupied cache space, so the message size is also an essential factor for the message pricing.

## 3.2 Equivalent Exchange Model

In Socially Aware Networking, the message forwarding process is abstracted as a currency transaction process in EEIS. When two nodes meet, the buyer is the party that needs to send messages, and the seller is the other party that receives messages. The buyer uses the virtual currency it owns to purchase the seller’s forwarding service, and the seller obtains the certain virtual currency by selling its service. However, in the process of both parties’ transactions, both sides must understand the other side’s resource state, wealth status and their respective pricing for the message, which inhibits nodes from making false pricing.

### 3.2.1 Source Node’s Bidding Function

Here the source node represents the node that wants to send messages. Before the source node forwards message, it needs to consider its remaining resource state and wealth status, so as to make its reasonable price for messages, and pay the forwarding node the corresponding amount of virtual currency. If the remaining buffer space of the source node is insufficient, it will send messages at a relatively high price in order to reduce the pressure. However, considering that messages are not suitable for long-term storage in the cache, the source node will utilize the price as a factor so as to promote it to send messages as soon as possible. Therefore, while the source node’s wealth status is rich, it will price messages based on the node’s remaining resource state and wealth status. But while its wealth status is poor, it will only consider the node’s

resource state. Hence the bidding function of the source node is expressed as

$$P = M \times [\alpha \times (1 - B_s) + \beta \times (1 - E_s)] \times R'_s \quad (4)$$

In which,  $M$  is the size of the message to be sent,  $P$  is the source node’s pricing for the message,  $B_s$  represents the remaining cache percentage of the source node  $s$ , and  $E_s$  represents the remaining energy percentage of the source node  $s$ .  $R'_s$  denotes the wealth status of the source node  $s$ , and  $R'_s = \{R_s, \text{Rich Status } 1, \text{Others}\}$ . Moreover,  $\alpha$  and  $\beta$  are weights, which represent the influence of the remaining cache and remaining energy of the source node  $s$  on the pricing for the message, and  $\alpha + \beta = 1$ . At this time, the experiment in Fig. 2 proves that the values of  $\alpha$  and  $\beta$  are both 0.5 respectively, which are the most appropriate.

It can be seen from Formula (4) that when  $R_s > 1$ , that is, the wealth status of node  $s$  is rich, it is willing to realize the message forwarding at a relatively high price. On the contrary it will pay a lower price. When the message takes up the node’s cache for a long time, it will cause the node’s remaining cache  $B_s$  to become smaller and the node needs to pay a higher reward while forwarding messages. Therefore, if the node sends the message as soon as possible, it will relieve the pressure on the cache and pay a relatively low price.

### 3.2.2 Forwarding Node’s Pricing Function

The forwarding node means the node that receives messages. Similarly, it must also consider its own remaining resource state and wealth status before the forwarding node receives messages, so as to set a reasonable price for messages to be received and obtain a certain amount of virtual currency from the source node. If the forwarding node’s wealth status is poor, it will receive messages at a low price. Thereby it needs to

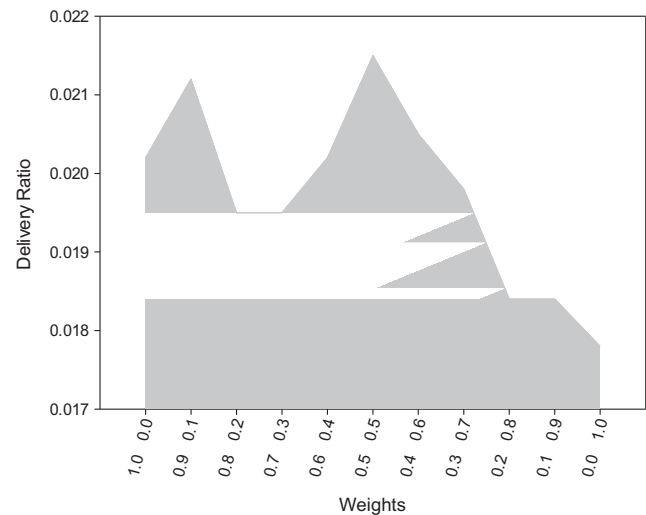


Figure 2 The effect of weights on message delivery rate.

forwarding more messages to acquire more virtual currency and alleviate its own wealth crisis. Of course, if the forwarding node’s remaining resource is inadequate, then it will set a higher price to receive messages. Therefore, when its wealth status is poor, it will consider its own remaining resource state and wealth status, otherwise, it will only consider the node’s resource state. Hence the pricing function of the forwarding node is expressed as

$$S = M \times [\lambda \times (1 - B_f) + \omega \times (1 - E_f)] \times R'_f \tag{5}$$

In which,  $M$  is the size of the message to be sent,  $S$  is the forwarding node’s pricing for the message,  $B_f$  represents the remaining cache percentage of the forwarding node  $f$ , and  $E_f$  represents the remaining energy percentage of the forwarding node  $f$ .  $R'_f$  denotes the wealth status of the forwarding node  $f$ , and  $R'_f = \{ R_f, \text{ Poor Status } 1, \text{ Others} \}$ . Moreover,  $\lambda$  and  $\omega$  are weights, which represent the influence of the remaining cache and remaining energy of the forwarding node  $f$  on the pricing for the message, and  $\lambda + \omega = 1$ . At this time, the experiment in Fig. 2 proves that the values of  $\lambda$  and  $\omega$  are both 0.5 respectively, which are the most appropriate.

It can be seen from Formula (5) that when  $R_f < 1$ , that is, the wealth status of node  $f$  is poor, it will receive more messages at a relatively low price to obtain more virtual currency. When the node continues to forward messages, its remaining buffer space will be larger and the price for messages will also become lower, which promotes the forwarding of messages. When the node’s resource is insufficient, it is unwilling to receive messages, and the price for message will become higher, which may cause both parties to abandon the transaction. Therefore, the node will receive messages from other nodes as much as possible if its own conditions permit.

### 3.2.3 Trade Rules

When the source node meets the forwarding node, both sides start negotiation and transaction for the message that needs to be forwarded. The trading rules of EEIS are shown in Fig. 3, and the specific description is as follows (both nodes must understand each other’s message prices, resource state and wealth status):

- (1) For the message to be sent, the price given by the source node  $s$  is denoted as  $P$ ; for the message to be received, the price given by the forwarding node  $f$  is denoted as  $S$ . If  $S \leq P$ , the transaction between the two parties will be successful and the message forwarding will be completed. Then continue the next message’s forwarding transaction.
- (2) If  $S > P$ , the source node  $s$  and the forwarding node  $f$  cannot negotiate successfully, then both sides re-price respectively the message according to both parties’ the

remaining resource state and wealth status and make second price negotiation.

- (3) The rules for second bargaining are divided into the following three situations:

a. If the wealth status of the forwarding node  $f$  is rich, that is  $R_f \geq 1$ , the price given by the source node  $s$  will be still  $P$ , and the forwarding node  $f$  discounts the price  $S$  according to the encounter probability to try the next bargaining to complete the message forwarding trade. The price given by the forwarding node after the discount is

$$S_{discount} = S \times p \tag{6}$$

Where  $S_{discount}$  is the discounted price of the forwarding node  $f$ ,  $S$  is the initial message pricing of the forwarding node  $f$ , and  $p$  is the encounter probability between the forwarding node and the destination node.

b. If the wealth status of the forwarding node  $f$  is poor and that of the source node  $s$  is rich, that is  $R_f < 1$  and  $R_s \geq 1$ , then the price given by the forwarding node  $f$  will be still  $S$ . When the remaining resource of the source node  $s$  is sufficient, there is no pressure at this time, so the price it gives can be slightly increased to try the next negotiation and complete the message forwarding transaction; When the remaining resource of the source node  $s$  is inadequate, it wants to forward the message as soon as possible. At this time, the source node  $s$  needs to increase the price of the message according to its own resource state and wealth status to try the next negotiation and complete the forwarding transaction. Therefore, the price raised by the source node  $s$  is

$$P_{raise} = P \times R_s \times \frac{B_s + E_s}{B_s^2 + E_s^2} \tag{7}$$

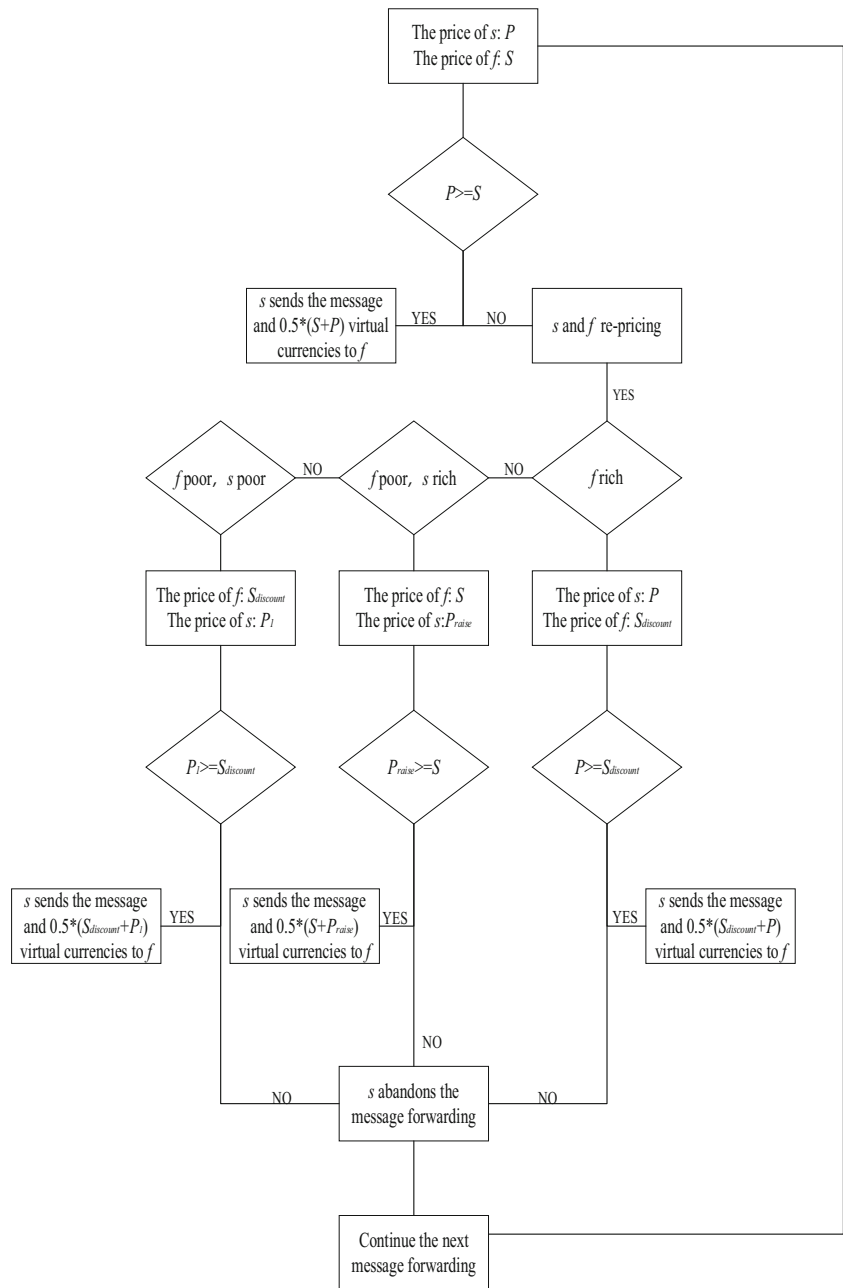
Where  $P_{raise}$  is the increased price of source node  $s$ ,  $P$  is the original message price of source node  $s$ ,  $R_s$  is the current wealth status of source node  $s$ ,  $B_s$  is the remaining cache percentage of source node  $s$ , and  $E_s$  is the remaining energy percentage of source node  $s$ .

c. If the wealth status of the source node  $s$  and the forwarding node  $f$  is both poor, that is  $R_f < 1$  and  $R_s < 1$ , then both sides need to change their message pricing. The changed price of forwarding node  $f$  is calculated according to Formula (6), and that of source node  $s$  is

$$P_I = P \times R_s \tag{8}$$

Where  $P_I$  is the changed price of source node  $s$ ,  $P$  is the original price of source node  $s$ , and  $R_s$  is the current wealth status of source node  $s$ .

**Figure. 3** Message forwarding transaction process



- (4) If the source node and the forwarding node negotiate successfully for the second time, the message forwarding transaction will be completed, and then the forwarding trade for the next message will be continued. If the both sides' negotiation fails, the message forwarding trade will be abandoned and the next message forwarding transaction will be started.

The pseudo-code of the message forwarding transaction process is shown in Algorithm 1. Algorithm 1 describes the two-time negotiation process of messages forwarding between the nodes. Algorithm input:  $P$ , the price of source node  $s$  for message  $m$ ;  $S$ , the price of forwarding node  $f$  for message

$m$ ;  $R_s$ , the wealth status of node  $s$ ;  $R_f$ , the wealth status of node  $f$ ;  $probability$ , the encounter probability between node  $f$  and the destination node of message  $m$ ;  $B_s$ , the remaining buffer space percentage of source node  $s$ ;  $E_s$ , the remaining energy percentage of node  $s$ . First, compare price  $P$  and  $S$  made respectively by node  $s$  and node  $f$  for message  $m$ . If  $P$  is greater than or equal to  $S$ , node  $s$  forwards message  $m$  and pay half virtual currency of the sum of  $P$  and  $S$  as the trade price to node  $f$ . Otherwise, according to the respective wealth status of source node  $s$  and forwarding node  $f$ , two nodes respectively make re-pricing  $P_l$  and  $S_l$ , then compare them again. If  $P_l$  is greater than or equal to  $S_l$ , node  $s$  forwards message  $m$  and pay half virtual currency of the sum of  $P_l$

and  $S_f$  as the trade price to node  $f$ . If two nodes' bargaining twice still fails, the forwarding transaction of message  $m$  will be given up and continue the forwarding trade of the next message.

### 3.3 Game Analysis

The message forwarding process can be seen as a two-stage game, that is whether the source node  $s$  chooses to send the

---

#### Algorithm 1 EEIS Algorithm

---

**Input:**  $S, P, R_s, R_f, m, probability, B_s, E_s$

**Output:** *List messages, cost*

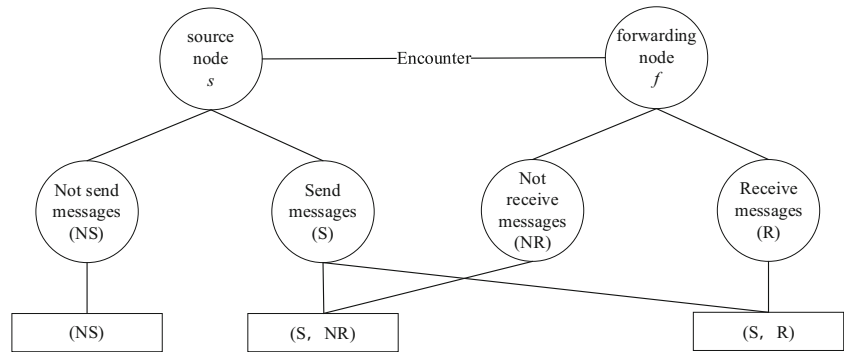
```

1: if  $S \geq P$  then
2:    $m \rightarrow messages$ 
3:    $cost = \frac{(S+P)}{2}$ 
4: else
5:    $P_1 = GetPriceTwice_S(S, P, R_s, R_f)$ 
6:    $S_1 = GetPriceTwice_F(S, P, R_s, R_f)$ 
7:   if  $S_1 \geq P_1$  then
8:      $m \rightarrow messages$ 
9:      $cost = \frac{(S_1+P_1)}{2}$ 
10:  else
11:    continue next message
12:  end if
13: end if
14: function  $GetPriceTwice_S(S, P, R_s, R_f)$ 
15:   if  $R_f \geq 1$  then
16:      $P_1 = P$ 
17:   end if
18:   if  $R_f < 1$  and  $R_s \geq 1$  then
19:      $P_1 = P * R_s * \frac{B_s + E_s}{B_s^2 + E_s^2}$ 
20:   end if
21:   if  $R_f < 1$  and  $R_s < 1$  then
22:      $P_1 = P * R_s$ 
23:   end if
24:   return  $P_1$ 
25: end function
26: function  $GetPriceTwice_F(S, P, R_s, R_f)$ 
27:   if  $R_f \geq 1$  then
28:      $S_1 = S * probability$ 
29:   end if
30:   if  $R_f < 1$  and  $R_s \geq 1$  then
31:      $S_1 = S$ 
32:   end if
33:   if  $R_f < 1$  and  $R_s < 1$  then
34:      $S_1 = S * probability$ 
35:   end if
36:   return  $S_1$ 
37: end function

```

---

**Figure 4** Game process of message forwarding



message and whether the forwarding node  $f$  chooses to receive the message, as shown in Fig. 4.

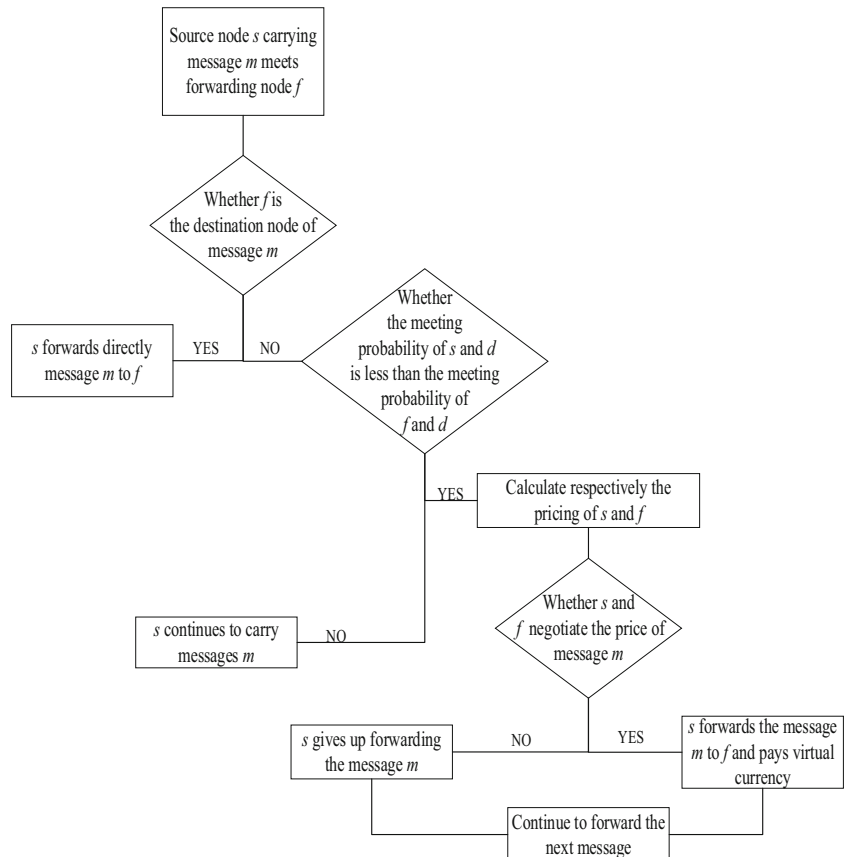
Assuming that at time  $t$ , for message  $m$ , the price given by source node  $s$  is  $P$ , the price given by forwarding node  $f$  is  $S$ , and  $P$  is less than or equal to  $S$ . At this time, if source node  $s$  does not choose to send the message, then at time  $t + 1$ , when node  $s$  wants to send messages, it must consider the cost of it in term of cache and energy.

$$l_1 = \frac{M}{E_s} \times L_1 \times \gamma \times R_s \tag{9}$$

In which,  $L_1$  represents the length of message to be forwarded by source node  $s$  at time  $t + 1$ ,  $E_s$  is the remaining energy percentage of source node  $s$  at time  $t + 1$ ,  $R_s$  is the wealth status of node  $s$  at the current time, and  $M$  is the length of message  $m$ .

- (1) Since the energy of the source node  $s$  is strictly decreasing, at time  $t + 1$ , when node  $s$  wants to forward messages, the price calculated according to Formula (4) will be increased. Therefore, the part of the additional expenditure is expressed as
- (2) Since the source node  $s$  does not send the message  $m$  at time  $t$ , resulting that a part of the buffer space is occupied. At time  $t + 1$ , when node  $s$  wants to forward messages, the price calculated according to Formula (4) will be also increased. Hence, the part to spend more is expressed as

**Figure 5** Messages forwarding process





$$l_2 = \frac{M}{B_s} \times L_2 \times \eta \times R_s \tag{10}$$

In which,  $B_s$  is the remaining cache of source node  $s$  at time  $t + 1$ ,  $R_s$  is the wealth status of node  $s$  at the current moment,  $M$  is the length of message  $m$ , and  $L_2$  is the length of the message to be forwarded by source node  $s$  at time  $t + 1$ .

In Summary, the cost of source node  $s$  choosing not to forward message  $m$  at time  $t$  is at least

$$l = \min\{l_1, l_2\} \tag{11}$$

Similarly, if the forwarding node  $f$  does not choose to receive the message  $m$  at time  $t$ , then it will not receive any currency at this time, and its pricing may become higher at the next moment, which will eventually lead to lower cost savings or even trade failure. But if the node  $f$  chooses to receive message  $m$  at time  $t$ , then the cost it saves is expressed as

$$C = P - S, \quad C > 0 > -l \tag{12}$$

Therefore, if the source node  $s$  and the forwarding node  $f$  choose to send and receive messages at time  $t$ , their obtained gains will be much greater than the gains when they choose not to send and receive, that is, (S, R) is the optimal choice. Hence, the source node  $s$  is willing to forward messages, and the forwarding node  $f$  is also willing to receive messages. And after node  $f$  receives messages, it will also forward messages as soon as possible.

### 4 Data Forwarding Scheme

In Socially Aware Networking, all nodes will use Prophet routing algorithm for messages forwarding, and the encounter probability is used as the measurement unit. Figure 5 describes the general messages forwarding process. Assuming that message  $m$  is a message in the buffer space of source node  $s$ , and node  $d$  is the destination node of message  $m$ . The specific forwarding process is as follow:

When the source node  $s$  and the forwarding node  $f$  encounter, if the meeting probability of the forwarding node  $f$  and the destination node  $d$  is greater than that of the source node  $s$  and the destination node  $d$ , then the source node  $s$  will send a request information about purchasing the forwarding service to the forwarding node  $f$ . After the forwarding node  $f$  (it needs to ensure that it is not the destination node of the message  $m$ ) receives the request information, the source node  $s$  and the forwarding node  $f$  will enter the message forwarding transaction stage. According to formula (4) and formula (5), they first calculate their own prices, and then the two parties begin to negotiate the message price. After the negotiation is successful, the forwarding node  $f$  will return the information about

**Table 1** Basic parameters of simulation experiment

Parameter	Value	Unit
Simulation time	24	$h$
Simulation area	4500*3400	$m^2$
Warmup time	1000	$s$
Transmit range	10	$m$
Transmit speed	250	$kB/s$
Node number	200	$pcs$
Movement speed	0.5–1.5	$m/s$
Message size	500–1024	$kB$
Message TTL	600	$minutes$
Buffer size	2	$M$
TTL	10	$h$
Initial energy	150	$J$
Initial currency	500	$pcs$

confirming the sale of forwarding services to the source node  $s$ , indicating that the forwarding node  $f$  can receive the message  $m$ . The source node  $s$  starts to send the message  $m$  to the forwarding node  $f$  after receiving the confirmation message. After the message  $M$  is forwarded, the source node  $s$  will pay the negotiated certain virtual currency to the forwarding node  $f$  and then the message  $m$  will be deleted in the cache of the source node  $s$ .

In the messages forwarding process, each node maintains an information table, which stores the remaining resources state, wealth status, message price of the source node and the forwarding node. Both parties can check this table to determine whether the other party has made false pricing for the message.

## 5 Experimental Results and Analysis

### 5.1 Experimental Environment

On PC Lenovo, CPU 3.60GHz, Memory 16.0GB, OS Windows 10 and Software IntelliJ IDEA, this paper applies the ONE (Opportunistic Network Environment) [37] simulation platform for experiments. In the experiments, the performance analysis of the EEIS algorithm is mainly divided into two situations.

Situation 1: Compare NotWith\_EEIS, With\_EEIS these two algorithms for performance analysis under different numbers of selfish nodes.

NotWith\_EEIS algorithm, there are probably selfish nodes in the network, but the selfish nodes do not forward any messages for other nodes. Only when it encounters the destination node, it can forward the carried messages to the destination node.

With `_EEIS` algorithm, there are selfish nodes in the network, but selfish nodes will forward messages to other nodes according to the `EEIS` algorithm.

Situation 2: Compare `EEIS`, `BIP`, `GIS` these three algorithms for performance analysis on Prophet Routing under different simulation time and messages TTL.

- (1) `EEIS` algorithm, all nodes in the network are selfish nodes, but selfish nodes will forward messages to other nodes according to the `EEIS` algorithm.
- (2) `BIP` algorithm, all nodes in the network are selfish nodes, but selfish nodes will forward messages to other nodes according to the `BIP` algorithm. In the `BIP` algorithm, the two nodes make price for the message according to the message's remaining TTL and the node's remaining cache, but the two parties of transaction only negotiate the price once. If it succeeds, then trading and forwarding. Otherwise, the transaction is abandoned and the next message is to be traded and forwarded.
- (3) `GIS` algorithm, all nodes in the network are selfish nodes, but selfish nodes will forward messages to other nodes according to the `GIS` algorithm. In `GIS` algorithm, the relay node first bids on the message, and if the sending node accepts this price, it will forward and trade. Otherwise, the sending node re-bids. At this time, the relay node will forward and trade if it accepts price. Or else the relay node will bid again, and then the sending node must accept this price and carry out the forwarding transaction.

The detailed simulation parameters are shown below Table 1.

The simulation experiment parameter settings of the two cases are basically the same. The only difference is that in situation 1, 60 messages per hour will be generated, while a message every 25 to 35 s will be generated in situation 2.

## 5.2 Network Performance Parameters

This paper mainly uses the following three parameters to evaluate and compare the network performance of the above mechanisms to verify the availability of the data forwarding incentive algorithm.

### 5.2.1 Delivery Ratio

The delivery ratio of messages refers to the ratio of the number of successfully delivered messages to the total number of created messages, it can be expressed as.

$$delivery\_prob = \frac{nrofDelivered}{nrofCreated} \quad (13)$$

Where *delivery\_prob* is the delivery ratio, *nrofDelivered* is the number of successfully delivered messages, *nrofCreated* is the total number of created messages.

### 5.2.2 Average Latency

The average latency refers to the average time required for messages from creation to delivery successfully, it can be expressed as

$$latency\_avg = \frac{\sum Count^1 (DeliveryTime - CreatTime)}{Count} \quad (14)$$

In which, *latency\_avg* is the average latency for the successful delivery of the messages, *DeliveryTime* is the time when the messages is delivered to the destination node, *CreatTime* is the time the messages is created, *Count* is the total number of messages successfully delivered.

### 5.2.3 Network Overhead

The network overhead refers to the ratio of the difference between the number of relay times of messages sent from the source node to the destination node and the number of successfully delivered messages to the total number of successfully delivered message, it can be expressed as

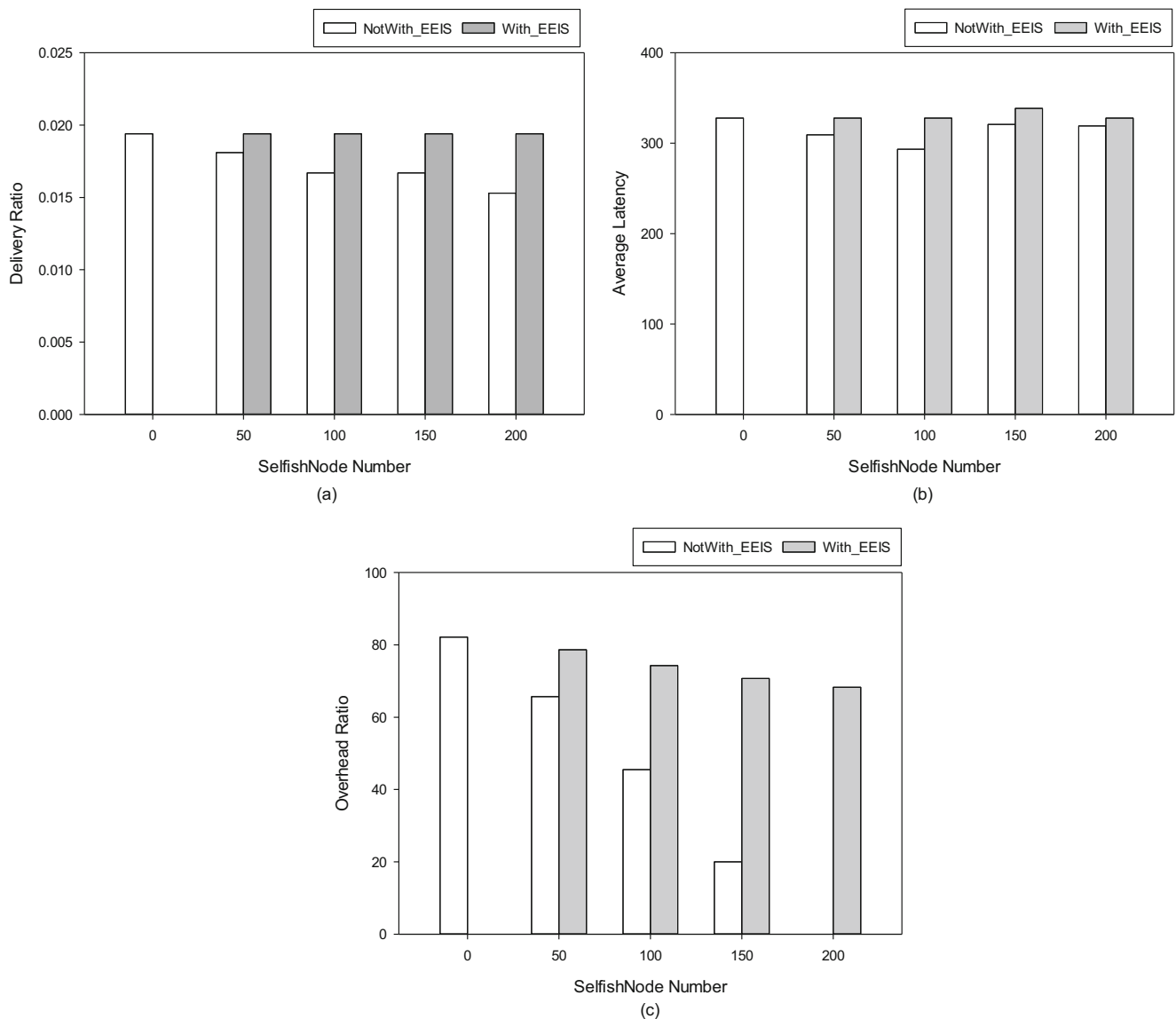
$$overhead\_ratio = \frac{(nrofRelayed - nrofDelivered)}{nrofDelivered} \quad (15)$$

Where *overhead\_ratio* is network overhead for successful messages delivery, *nrofRelayed* is the number of hops passed during messages sending, and *nrofDelivered* is the total number of messages successfully delivered.

## 5.3 Simulation Results Analysis

(1) The influence of the number of selfish nodes on network performance.

Figure 6 describes the performance comparison of two algorithms of `NotWith_EEIS` and `With_EEIS` in term of delivery ratio, average latency and network overhead as the number of selfish nodes changes. It can be seen from the figure that `With_EEIS` algorithm has achieved better network performance. With the continuous joining of selfish nodes, the network overhead can be reduced while maintaining the delivery ratio and average latency. Figure 6(a) describes the comparison of the delivery ratio of two algorithms. With the change in the number of selfish nodes, the delivery ratio of `NotWith_EEIS` algorithm continues to decrease, and the messages delivery ratio of `With_EEIS` algorithm is almost the same as that of `NotWith_EEIS` algorithm when the number of selfish nodes is zero. Therefore, there is almost no impact



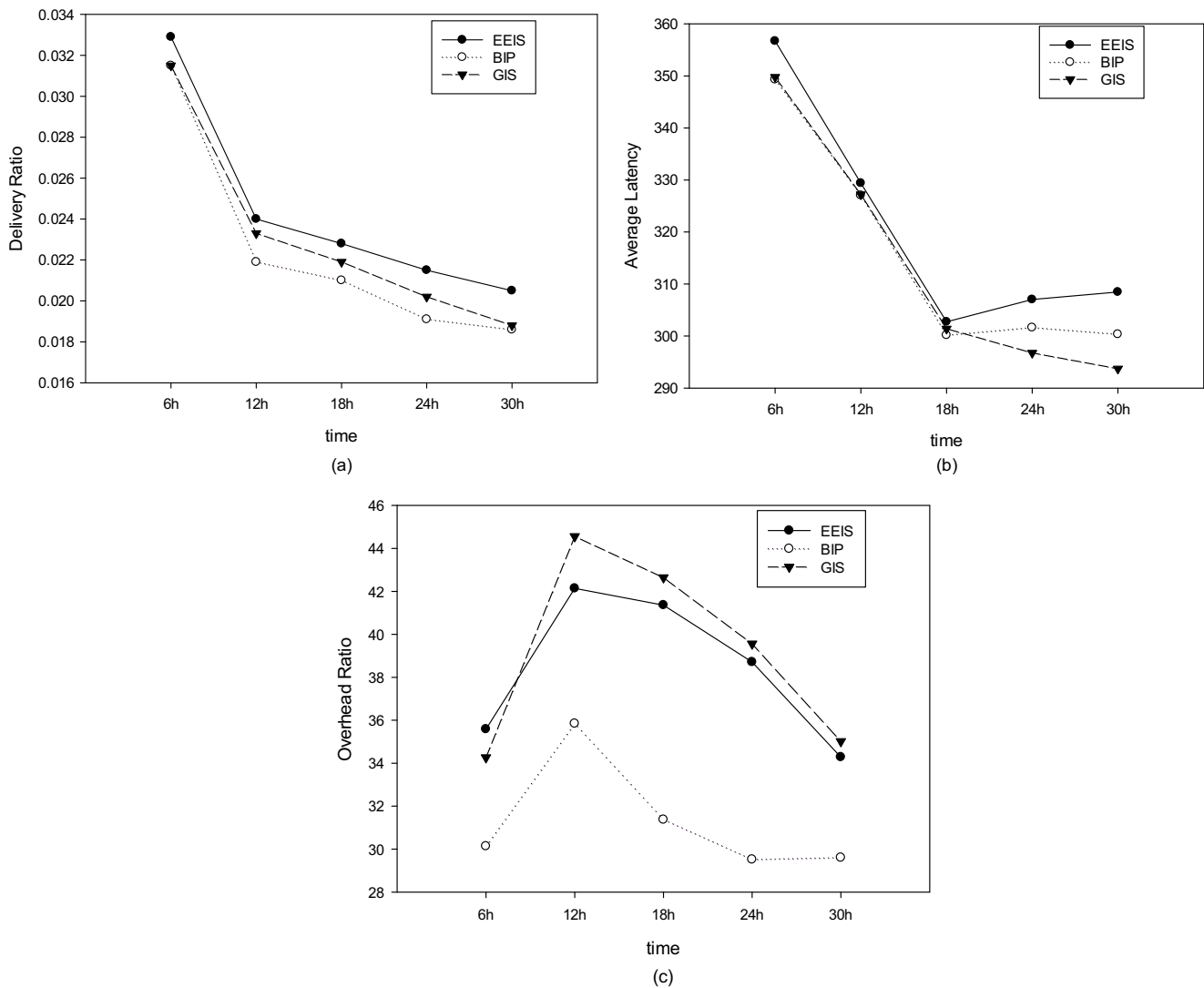
**Figure 6** Performance comparison of various algorithms under different numbers of selfish nodes

on the EEIS’s delivery ratio as the selfish nodes add. Figure 6(b) describes the average latency comparison of two algorithm. The number of selfish nodes keeps increasing, and the average latency of NotWith\_EEIS algorithm is reduced, but the average latency of With\_EEIS algorithms are basically the same as that of NotWith\_EEIS algorithm when the number of selfish nodes is zero. Figure 6(c) compares the network overhead of these two algorithms. With the continuous addition of selfish nodes, the network overhead of NotWith\_EEIS algorithm drops sharply, and the network overhead of With\_EEIS algorithm also continues to drop and is lower than that of NotWith\_EEIS algorithm when the number of selfish nodes is zero. Therefore, after applying the EEIS algorithm, the selfish nodes will actively cooperate with other nodes and forward messages in order to earn virtual currency, so that messages are

successfully forwarded to the destination nodes instead of being rejected or discarded.

(2) The effect of different simulation time on network performance.

Figure 7 describes the performance comparison of EEIS, BIP and GIS these three algorithms in terms of messages delivery ratio, average latency and network overhead as the simulation time changes. It can be seen from Fig. 7(a) that with the change of simulation time, the delivery ratio of EEIS has always been better than that of BIP and GIS, and the reason for the gradual decrease in delivery ratio is that the number of messages generated becomes even larger as time increases, and the energy for sending messages is reduced. When the energy is insufficient, the messages are discarded, which causes the messages delivery ratio of these three algorithms to drop. It can be seen from Fig. 7(b) that the average



**Figure 7** Performance comparison of various algorithms under different simulation times

latency of GIS and BIP are lower than EEIS. When the simulation time is less than 18 h, the average latency of these three algorithms is in a decreasing state. This is because the number of messages that need to be forwarded increases as the simulation time goes by, and the speed at which the messages are successfully delivered is far less than the speed at which they are generated. Therefore, the average latency of three algorithms has been declining. But when the simulation time exceeds 18 h, the average latency of EEIS shows an upward trend, while BIP first rises and then falls, GIS has been in a downward state. This is due to insufficient energy or currency of nodes in GIS over time, resulting in messages being discarded. The reason for the increasing in EEIS and BIP is that the increasing in simulation time causes more messages to be forwarded, so that the time to successfully receive messages will be extended. It can be seen from Fig. 7(c) that the network overhead of EEIS algorithm is lower than that of GIS algorithm, but higher than that of BIP algorithm. This is

because the EEIS algorithm requires two bargaining between nodes which BIP requires one bargaining and GIS requires three bargaining between nodes, so the number of messages forwarding in EEIS is greater than that in the BIP algorithm, and lower than that in GIS algorithm.

### (3) The impact of different TTL on network performance.

Figure 8 describes the performance comparison of EEIS, BIP and GIS these three algorithms in terms of messages delivery ratio, average latency and network overhead as the messages TTL increases. It can be seen from Fig. 8(a) that as the messages TTL changes, the delivery ratio of EEIS algorithm is higher than that of BIP and GIS. From Fig. 8(b), we can see that the average latency of EEIS, BIP and GIS algorithms have little difference. It can be seen from Fig. 8(c) that the network overhead of EEIS algorithm is generally lower than that of the GIS algorithm, but higher than that of BIP algorithm. This is because the EEIS algorithm requires two bargaining between nodes, while GIS requires three bargaining, and BIP only needs one bargaining.

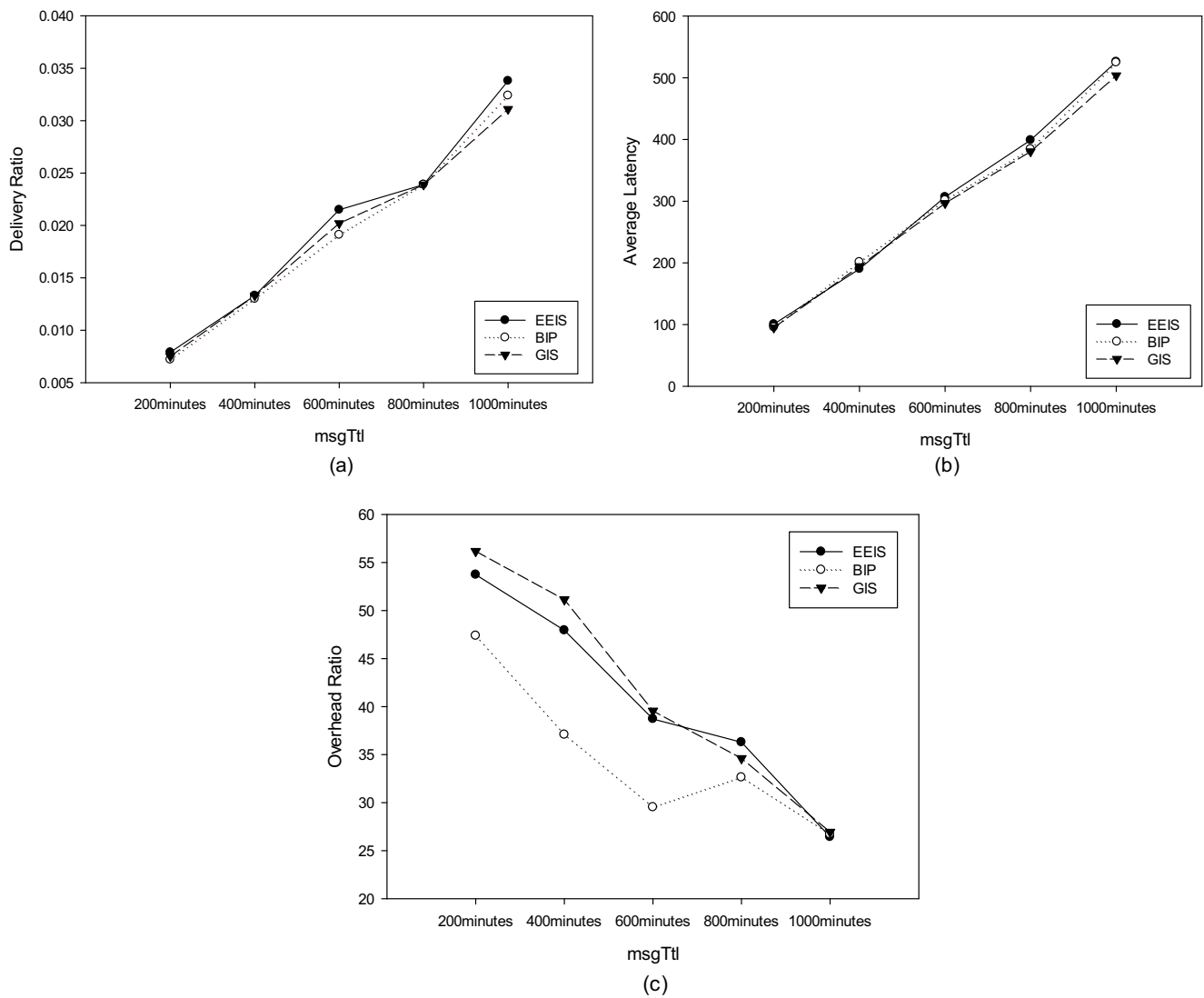


Figure 8 Performance comparison of various algorithms under different messages TTL

Therefore, the number of messages forwarding in EEIS is more than that in BIP and less than that in GIS.

## 6 Conclusion

For the selfishness of nodes in socially aware networking, this paper proposes an incentive scheme EEIS based on equivalent exchange. EEIS mainly abstracts messages forwarding into a two-party transaction process, and introduces an equivalent exchange model, so that both sides in the transaction can obtain maximum benefits through virtual currency, thereby encouraging selfish nodes to participate rationally in the messages forwarding process and forward messages for other nodes. Experimental results show that EEIS algorithm can effectively motivate nodes to participate in cooperation, and it still has a better messages delivery ratio under the condition of limited resources.

Although the EEIS algorithm can effectively incentivize nodes, the paper does not take into account the fact that the messages are discarded due to insufficient currency after the messages are forwarded, and whether the messages are successfully forwarded to the destination node, etc. Therefore, we will further study the issues of whether the currency can be overdrawn first to forward the messages and confirm that the messages is successfully forwarded to the destination node.

**Acknowledgments** This work was supported by the National Natural Science Foundation of China (No. 61972136), MOE (Ministry of Education in China) Project of Humanities and Social Sciences (No.20YJAZH112), Hubei Provincial Department of Education Outstanding Youth Scientific Innovation Team Support Foundation (T2020017, T201410), Hubei Province Higher Education Teaching Research Project (No.2018432), the National Natural Science Foundation of China (No.61971339).

## References

- Gai, K., Qiu, M., & Zhao, H. (2017). Privacy-preserving data encryption strategy for big data in Mobile cloud computing. *IEEE Transactions on Big Data*, 1–1.
- Qiu, M., Jia, Z., Xue, C., Shao, Z., & Sha, E. H. M. (2007). Voltage assignment with guaranteed probability satisfying timing constraint for real-time Multiprocessor DSP. *Journal of VLSI Signal Processing*, 46(1), 55–73.
- Xia, F., Liu, L., Li, J., Ma, J., & Vasilakos, A. V. (2015). Socially aware networking: A survey. *IEEE Systems Journal*, 9(3), 904–921.
- Xia, F., Liu, L., Li, J., Ahmed, A. M., Yang, L. T., & Ma, J. (2015). BEEINFO: Interest-based forwarding using artificial bee colony for socially aware networking. *IEEE Transactions on Vehicular Technology*, 64(3), 1188–1200.
- Ning, Z., Liu, L., Xia, F., Jedari, B., Lee, I., & Zhang, W. (2017). CAIS: A copy adjustable incentive scheme in community-based socially aware networking. *IEEE Trans. Vehicular Technology*, 66(4), 3406–3419.
- Guo, Y., Zhuge, Q., Hu, J., Yi, J., Qiu, M., & Sha, E. H. M. (2013). Data placement and duplication for embedded multicore systems with scratch pad memory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(6), 809–817.
- Xu, F., Xu, Q., Xiong, Z., Xiao, N., Xie, Y., Deng, M., & Hao, H. (2019). Intelligent distributed routing scheme based on social similarity for mobile social networks. *Future Generation Computer Systems*, 96, 472–480.
- Yürür, Ö., Liu, C. H., Sheng, Z., et al. (2016). Context-awareness for mobile sensing: A survey and future directions. *IEEE Communications Surveys & Tutorials*, 18(1), 68–93.
- Xu, F., Xiao, N., Deng, M., Xie, Y., Xiong, Z., & Xu, Q. (2019). Efficient opportunistic routing with social context awareness for distributed mobile social networks. *Concurrency and Computation: Practice and Experience*, 5, e5524.
- Bisio, I., Lavagetto, F., Sciarrone, A., Penner, T., & Guirguis, M. (2017). Context-awareness over transient cloud in D2D networks: Energy performance analysis and evaluation. *Transactions on Emerging Telecommunications Technologies*, 28(2), e3002.
- Song, Y., Li, Y., Jia, L., & Qiu, M. (2020). Retraining strategy-based domain adaption network for intelligent fault diagnosis. *IEEE Transactions on Industrial Informatics*, 16(9), 6163–6171.
- M. Qiu, K. Zhang, M. Huang. An empirical study of web interface design on small display devices. *IEEE/WIC/ACM International Conference on Web Intelligence*. ACM, 2004:29–35.
- Chen, M., Zhang, Y., Qiu, M., Guizani, N., & Hao, Y. (2018). SPHA: Smart personal health advisor based on deep analytics. *IEEE Communications Magazine*, 56(3), 164–169.
- Li, Q., Zhu, S., & Cao, G. (2010). Routing in Socially Selfish Delay Tolerant Networks. *Infocom IEEE International Conference on Computer Communications*. *IEEE*, 1–9.
- Boldrini, C., Conti, M., & Passarella, A. (2010). Design and performance evaluation of ContentPlace, a social-aware data dissemination system for opportunistic networks. *Computer Networks*, 54(4), 589–604.
- Ma, X., Zheng, T., & Li, M. (2019). Analysis and regulation of effects of selfishness on opportunistic network. *Acta Electronica Sinica*, 47(04), 837–847.
- Lv, J., Song, H., He Z., Pan, D. (2017). Survey of research on node's selfish behavior in opportunistic networks. *Computer Engineering and Applications*, 53(18):7–16+50.
- Ren, Z., Chen, M., Kang, J., & Li, X. (2020). Opportunistic network selfish node detection algorithm combined with probabilistic routing. *Journal of Chinese Computer Systems*, 41(05), 1047–1052.
- Nobahary, S., Garakani, H. G., Khademzadeh, A., & Rahmani, A. M. (2019). Selfish node detection based on hierarchical game theory in IoT. *EURASIP Journal on Wireless Communications and Networking*, 1, 1–15.
- Marti, S. (2000). Mitigating routing misbehavior in mobile ad hoc networks. *International Conference on Mobile Computing and Networking*. *ACM*, 255–265.
- Bigwood, G., & Henderson, T. (2011). IRONMAN: Using social networks to add incentives and reputation to opportunistic networks. *IEEE Third International Conference on Privacy, Security, Risk and Trust*. *IEEE*, 65–72.
- Lenin, G. D., Esteve, P. S., Ahmad, M. M., & Jordi, F. (2019). A novel dynamic reputation-based source routing protocol for mobile ad hoc networks. *EURASIP Journal on Wireless Communications and Networking*, 1, 1–14.
- Li, N., & Das, S. K. (2010). RADON: Reputation-Assisted Data Forwarding in Opportunistic Networks. *International Workshop on Mobile Opportunistic Networking*. *ACM*, 8–14.
- Mei, A., & Stefa, J. (2012). Give2get: Forwarding in social mobile wireless networks of selfish individuals. *IEEE Transactions on Dependable and Secure Computing*, 9(4), 369–382.
- Chen, B. B., & Chan, M. C. (2010). MobiCent: A credit-based incentive system for disruption tolerant network. *IEEE Infocom*. *IEEE*, 14–19.
- Zhu, H., Lin, X., Lu, R., et al. (2009). Smart: A secure multilayer credit-based incentive scheme for delay-tolerant networks. *IEEE Transactions on Vehicular Technology*, 58(8), 4628–4639.
- Lu, R., Lin, X., Zhu, H., Shen, X., & Preiss, B. (2010). Pi: A practical incentive protocol for delay tolerant networks. *IEEE Transactions on Wireless Communications*, 9(4), 1483–1493.
- Shevade, U., Song, H. H., Qiu, L., et al. (2008). Incentive-aware routing in DTNs. *International Conference on Network Protocols*. *IEEE*, 238–247.
- Buttayan, L., Dora, L., Felegyhazi, M., et al. (2010). Barter trade improves message delivery in opportunistic networks. *Ad Hoc Networks*, 8(1), 1–14.
- Wu, E., Chen, T., Zhong, S., et al. (2010). A bargaining-based approach for incentive-compatible message forwarding in opportunistic networks. *2012 IEEE International Conference on Communications (ICC)*. *IEEE*, 789–793.
- Wu, F., Chen, T., Zhong, S., Qiao, C., & Chen, G. (2013). A game-theoretic approach to stimulate cooperation for probabilistic routing in opportunistic networks. *IEEE Transactions on Wireless Communications*, 12(4), 1573–1583.
- Wang, E. K., Chen, C. M., Yiu, S. M., et al. (2020). Incentive evolutionary game model for opportunistic social networks. *Future Generation Computer Systems*, 1, 14–29.
- Dai, W., Qiu, L., Wu, A., & Qiu, M. (2018). Cloud infrastructure resource allocation for big data applications. *IEEE Transactions on Big Data*, 4(3), 313–324.
- Li, Y., Yu, J., & You, X. (2013). An incentive protocol for opportunistic networks with resources constraint. *Chinese Journal of Computers*, 36(5), 947–956.
- Liu, Q., Liu, M., & Li, Y. (2015). Novel game-based incentive strategy for opportunistic networks. *Application Research of Computers*, 32(07), 2128–2132.
- Lindgren, A., Doria, A., Olov, S. (2003). Probabilistic Routing in Intermittently Connected Networks. 7(3):19–20.
- Keränen, A., Ott, J., & Kärkkäinen, T. (2009). The ONE Simulator for DTN Protocol Evaluation. *International Conference on Simulation Tools & Techniques*. *ICST*, 1–10.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Zenggang Xiong** received the Ph.D. degree from the University of Science and Technology Beijing, Beijing, China, in 2009. He is currently a Professor with the School of Computer and Information Science and Master Tutor, Hubei Engineering University, Xiaogan, China. His main research interests include cloud computing, big data, pattern recognition, and computer vision.



**Qiong Xu** was born in 1991. She received the B.S. degree in computer science and technology from Hubei University of China, in 2013, and she is currently pursuing the M.S. degree in computer technology with Hubei University of China. She is involved in research on routing algorithms for mobile social networks.



**Nan Xiao** was born in 1995. She received the B.S. degree in computer science and technology from Hubei Engineering University of China, in 2018, and she is currently pursuing the M.S. degree in computer technology with Hubei University of China. She is involved in research on incentives for selfish nodes in Mobile social network.



**Kaibing Zhang** received the M.S. degree in Computer Software and Theory from Xihua University, Chengdu, China, in 2005 and the Ph.D. degree from Xidian University, Xi'an, China, in 2012, respectively. He is currently a Professor at the College of Electrics and Information, Xian Polytechnic University, Xian, China. His main research interests include pattern recognition, computer vision, and image super resolution reconstruction. In these areas, he has published around

20 technical articles in refereed journals and proceedings including IEEE TIP, TNNLS, Signal Processing (Elsevier), CVPR, ICIP, etc.



**Fang Xu** received the M.S. and Ph.D. degrees from Wuhan University, Wuhan, China, in 2009 and 2016, respectively. He is currently a Professor in the School of Computer and Information Science, Hubei Engineering University, Hubei, China. His research interests include social computing, wireless mobile networks, and context aware computing.



**Conghuan Ye** received the B.S. and M.S. degree in computer science from Hubei Normal University, Hubei, China, in 2002, and University of Electronic Science and Technology of China, Chengdu, Sichuan, China, in 2005, respectively. He received the Ph.D. degree in Huazhong University of Science and Technology in 2013, Wuhan, Hubei, China. Since 2018, he has been a professor with the School of Computer and Information Science, Hubei

Engineering University, China. Now, his research interests include digital fingerprinting, digital right management, complex network, and cloud computing.



**Xuemin Zhang** received the Bachelor degree in computer science from Hubei Normal University, China, in 2001, and the MA degree in computer science from Wuhan University of Technology, China, in 2009. She is now a professor in Hubei Engineering University. Her research interests are in the areas of Cloud computing, distributed systems, Service Computing. She is a member of the IEEE and the ACM.