Detecting and Preventing Selfish Behaviour in Mobile Ad Hoc Network

Tao Lei[1], Shangguang Wang[1], Jinlin Li[1], Ilsun You[2], Fangchun Yang[1]
[1] State Key Laboratory of Networking and Switching Technology,
[1] Beijing University of Posts and Telecommunications,
[1] Beijing, China
[2] Department of Information Security Engineering
[2] Soonchunhyang University
[2] Asan-si, Republic of Korea
{leitao; sgwang; jlli }@bupt.edu.cn; isyou@sch.ac.kr; fcyang@bupt.edu.cn

Abstract

In mobile ad hoc networks for big data transmission, nodes communicate with each other via intermediate nodes. However, some intermediate nodes may behave selfishly, resulting in reduced throughput and increased delay in the network. Thus, the existence of selfish behaviour in a mobile ad hoc network degrades the performance of the network. In this paper, we propose a scheme that detects selfish behaviour and prevents its occurrence in mobile ad hoc network. The proposed scheme uses an adaptive threshold algorithm to detect selfish behaviour, and prevents it based on a repeated games scheme. Extensive simulations using NS-2 showed that our scheme can effectively detect and prevent selfish behaviour.

Keywords: Mobile ad hoc network, Repeated games, Adaptive threshold algorithm, Selfish behaviour

1 Introduction

Mobile ad hoc network (MANET) [1] are unlike traditional infrastructure networks. They are established solely by mobile devices, called mobile nodes, without any fixed infrastructure such as control centres and base stations. Thus, in MANET, network communication relies on mutual cooperation among the nodes for big data transmission. A node is not only a mobile terminal, but also acts as a router. This is an advantage in MANET, but the MANET is based on the assumption that each node is cooperative and trust. However, some of nodes may have selfish behaviour in reality.

The selfish behaviour occurs in the form of nodes trying to preserve their own resources, such as battery life and bandwidth, in an effort to receive the greatest benefits from the MANET [2]. In order to preserve their resources and maintain a longer survival time in MANET, nodes exhibiting selfish behaviour may refuse to cooperate with other nodes. For example, they may drop routing requests and data packets, or refuse to retransmit routing request packets in which they

have no interest. Consequently, selfish behaviour can potentially reduce throughput, increase delay, and degrade the performance of a MANET [3]. Therefore, if every node of the network decides to act selfishly, then the entire network could be collapsed.

The problem of selfish behaviour in MANET has been studied extensively throughout the years [4],with several schemes being proposed to solve problem. These schemes can be classified into three categories: credit-based schemes, reputation-based schemes, and game-based schemes.

Credit-based schemes use either a virtual or a real currency to pay for self-originated data retransmitted by other nodes. Credit is also used to compensate for the utilization of resources in the relaying process [4]. Nodes can also gain credit by retransmitting other nodes' packets or by exchanging real money. Buttyán and Hubaux [5] designed a credit-based system that provides incentives for forwarding packets in the form of virtual money. In their system, nodes earn credit by providing a forwarding service to others and, in exchange, have to pay for services from other nodes. Zhong et al.[6] proposed another system in which mobile nodes are provided with incentives to cooperate and report actions honestly. The proposed system does not require any tamper-proof hardware at any node. Yoo and Agrawal [7] investigated these credit-based systems and stated that they have the disadvantage of requiring the complete path information from the source to the destination, resulting in them being compatible only with source routing protocols.

Reputation-based schemes have also been proposed to prevent selfish behaviour in mobile ad hoc networks [4]. These schemes rely on building a reputation metric for each node according to its behavioural pattern. Detection and reaction are the two modules of reputation based on protocols. Nodes use the detection module to observe whether neighbour nodes retransmit packets from other nodes, and use the reaction module to change or update the reputation table. The most famous reputation-based scheme is the watchdog scheme [8], which detects instances of non-forwarding of data packets by listening to the transmission of its neighbour node. He et al. [9] proposed a secure and objective reputation-based incentive scheme that encourages packet forwarding and disciplines selfish.

Many theories for preventing selfish behaviour in MANET based on a combination of credit and reputation systems are also existence. Of these, one of the most popular approaches is based on game theory. The game theory model simulates a game in which each mobile node can choose whether to retransmit other nodes' data or not. Srivastava et al. [10] described how various interactions in wireless ad hoc networks can be modelled as a game, including credit and reputation-based game theory, and analysed selfish packet forwarding behaviour. Pandana et al. [11] proposed a self-learning repeated-game framework in which each distributed node studies the cooperation points and develops protocols for maintaining cooperation. Han and Poor [12] proposed an approach based on coalition games, in which the boundary nodes can use cooperative transmission to help the backbone nodes in the middle of the network. Debjist Das et al. [13] proposes a new

game theoretic scheme for selfish node detection with guarantying secure low cost data transfer and smallest amount of idle time in MANET.

Most of the schemes [4-13] used for selfish node detection can isolate those nodes exhibiting selfish behaviour. However, if the nodes are always isolated, their resources are not available for use. Consequently, in order to fully utilize the resources in MANET, this paper proposes a novel scheme that detects and prevents selfish behaviour. Different from our previous work [14], this paper focuses on the selfish behaviour detection and prevention. In our proposed scheme, we count the number of packets receiving and forwarding for each node, and use an adaptive threshold algorithm to ascertain whether those nodes demonstrate selfish behaviour or not. Then, we prevent selfish behaviour by forcing the nodes to forward data packets in MANET based on a repeated games scheme. The results of simulations conducted on NS-2 and numerical analysis indicate that our proposed scheme is very effective at the aspects of throughput and delay in MANET.

The remainder of this paper is organized as follows. Section 2 describes our proposed scheme for detecting and preventing selfish behaviour in MANET. Section 3 presents the results of performance evaluations conducted. Finally, Section 4 concludes this paper.

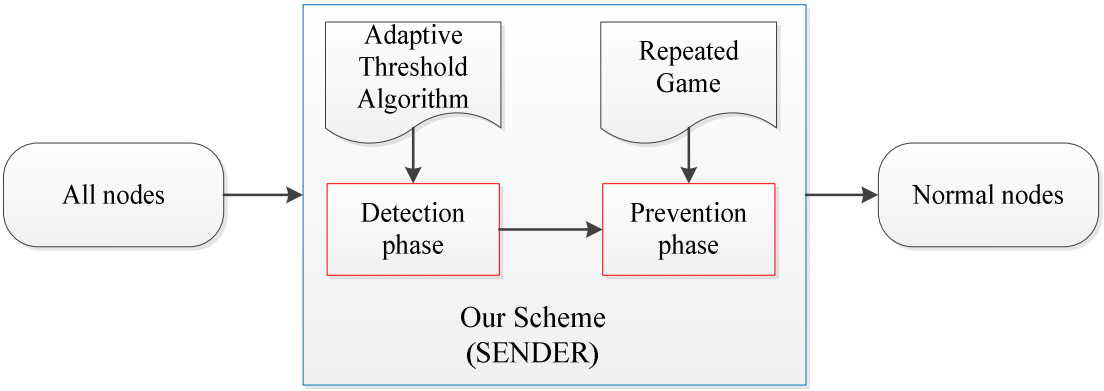

Fig. 1. Our proposed scheme

2 Our Proposed Scheme

It is well known that selfish behaviour affects throughput and delay in networks. Hence, if some nodes exhibit selfish behaviour in a MANET, the throughput of the MANET decreases sharply and its delay increases rapidly. In order to solve this problem, as shown in Fig.1, we proposed a selfish node detection and prevention scheme, called SENDER. This scheme contains two phases: a detection phase and a prevention phase. In the detection phase, we use an adaptive threshold algorithm to detect all nodes whether have selfish behaviour. In the prevention phase, we prevent the selfish behaviour based on repeated games.

2.1 The Detection Phase

In this phase, we detect and ascertain whether nodes exhibit selfish behaviour. In MANET, node often not to forward the packets when it has selfish behaviour, hence the number of packets forwarding with the selfish behaviour is far less than normal. To detect the selfish behaviour, the number of packet should be compared between present and the normal behaviour. An adaptive threshold algorithm [15] is often used to detect anomaly changes by comparing a present value with a previous value. This feature can be applied to detect selfish behaviour by nodes in MANET. Therefore, we use the adaptive threshold algorithm to detect selfish behaviour in SENDER. The procedure utilized in this phase comprises three steps. Firstly, a threshold value is set in accordance with former observation values. Next, we compute packet forwarding ratio (PFR), which denotes the ratio of the number of packets forwarding and the number of packets receiving in the present time interval. Finally, we use the adaptive threshold algorithm to compare the present PFR with a threshold value to determine whether the present node is exhibiting selfish behaviour. If PFR is lower than the threshold value, the node is selfish and an alarm is raised. Otherwise, the threshold value is updated according to the present PFR and the new threshold value for the next time interval.

2.1.1 The Adaptive Threshold Algorithm

We assume that x_n indicates the number of data packets in the *n-th* time interval, and $\overline{\mu}_n$ represents the rate estimated from prior measurements; thus, the alarm condition is as follows:

If $x_n \geq (1+\alpha)\overline{\mu}_{n-1}$, then an alarm is triggered at time interval *n*, where $\alpha > 0$ is a parameter that we consider to be an indication of anomalous behaviour and that indicates the percentage above the mean value. We can compute $\overline{\mu}_n$ using two methods: through some past time interval window, or using an exponentially weighted moving average (EWMA) of previous measurements, as in (1):

$$\overline{\mu}_n = \beta\overline{\mu}_{n-1} + (1-\beta)x_n, \tag{1}$$

where $\beta(0 \leq \beta \leq 1)$ is the EWMA factor.

If we directly apply the algorithm to detect selfish behaviour, this algorithm may lead to a high ratio of false alarms. Hence, we propose an improved adaptive threshold algorithm in which we set a value *k(k=1,2,...)*. Then, if the number of consecutive violations of the threshold is more than *k*, an alarm is raised. Consequently, the alarm condition is changed to the following:

$$\sum_{i=n-k+1}^{n} 1_{\{x_i \geq (1+\alpha)\overline{\mu}_{i-1}\}} \geq k, \tag{2}$$

where $k > 1$ is a parameter that indicates the number of consecutive time intervals the threshold is violated; if selfish behaviour occurs, then the alarm is raised at time interval *n*.

Nodes that exhibit selfish behaviour in MANET are not willing to forward data packets, and may drop packets which have no relationship with these nodes. Thus, we can adopt the adaptive threshold algorithm to detect selfish behaviour by computing PFR in each node of a MANET.

2.1.2 Application for Detection

As described in Section 3.1, we use the adaptive threshold algorithm to detect selfish behaviour. For this we need statistics x_n and y_n, where x_n signifies the number of packets that enter a node, and y_n indicates the number of packets forwarded by the node at the *n*-th time interval. Parameter z_n represents the ratio of y_n and x_n at the *n*-th time interval, i.e. z_n is PFR. In the ideal case, assuming that the PFR can be accurately counted in MANET, we have the following theorem:

Theorem 1. A node exhibiting selfish behaviour can be detected with high probability by computing the *n-th* ratio z_n and estimating the value μ_n with the adaptive threshold algorithm, assuming that the PFR can be accurately counted.

Proof: Under normal conditions, the PFR z_n of normal nodes changes over a small range. If one node exhibits selfish behaviour, its PFR must therefore be far from the normal, so we can set a threshold parameter $\alpha\left(\alpha\epsilon\left[0,1\right]\right)$ to monitor whether the PFR of the nodes is within the normal range. Selfish behaviour can be detected when the PFR is within an estimated range. Therefore, the threshold value's condition is $z_n < \alpha\overline{\mu}_{n-1}$, where $\overline{\mu}_{n-1}$ is the rate estimated from measurements prior to *n*. If $z_n > \alpha\overline{\mu}_{n-1}$ represents the normal state of the node at the *n*-th time interval, we can calculate the $\overline{\mu}_{n-1}$ using (1). Conversely, when one node's PFR is less than the threshold value at the *n*-th time interval, this value does not change, i.e., $\overline{\mu}_n = \overline{\mu}_{n-1}$. In accordance with the above description, the alarm condition equation changes as follows:

$$\sum_{i=n-k+1}^{n} 1_{\{x_i < a\,\overline{\mu}_{i-1}\}} \geq k \ . \tag{3}$$

Here, if the value is higher than *k*, then the alarm occurs at the *n*-th time interval. Under this new condition, when a certain node's PFR is continuously less than the threshold value more than *k* time intervals, the alarm could be at the *n*-th time interval. Hence, the value *k* is another threshold value, which we can accurately find with high probability experimentally. In other words, we can detect the selfish behaviour with high probability via the adaptive threshold algorithm.□

2.2 The Prevention Phase

Most of the mechanisms proposed to prevent selfish behaviour in Section 1 isolate nodes that are identified as demonstrating selfish behaviour immediately to ensure that the MANET becomes normal. However, such actions result in the resources possessed by those nodes being unavailable for use by the MANET. Resources are crucial in MANET, therefore, it is a

huge wastage for a MANET to isolate a node that exhibits selfish behaviour occasionally. Hence, it is important to prevent selfish behaviour in MANET. In order to induce all nodes to participate in the forwarding of data packets in MANET, any instance of selfish behaviour must be prevented. As we known, the repeated games always used to induce the two parties achieving a new balance which can be satisfied for the both parties in a game. Therefore, in this phase, repeated games can be used to punish nodes which exhibit selfish behaviour in the MANET.

In this section, we discuss the details of the prevention phase, in which repeated games are used to prevent selfish behaviour in MANET. In Section 2.2.1, we introduce the repeated games utilized in this paper. Next, we show in detail how selfish behaviour is prevented using the repeated games in Section 2.2.2.

2.2.1 Repeated Games

Repeated games [16] refers to scenario in which the same structure in a game is repeated in many time intervals with each game being called a 'stage game'. The repeated game can be defined as the following way. First, the players' strategy spaces and payoff functions must be specified. Then, all players observe the realized actions at the end of each discrete time period in which the stage game played. And the repeated games have three common strategies: trigger strategy, limited punishment strategy, and tit-for-tat strategy. In the trigger strategy, assuming the two game players choose integrity at the *t-1(t=2,3,...)* stage, and one of them chooses the cheat strategy at the *t(t=2,3,...)* stage, then the other players could choose the cheat strategy immediately. Limited punishment strategy, punishment will last *k (k=0,1,2,...)* time intervals, but not forever, in this strategy. In the tit-for-tat strategy, if one player chooses to cheat at one time interval, the other player can select cheat immediately during the next time intervals to punish his opponent.

When the game only occurs at one time interval, each participant just cares about its disposable payment. If the game is repeated many times, participants may consider sacrificing the disposable income for long-term interests, and then select a different strategy. Thus, the number of games will affect the result of game equilibrium. In the repeated games, a discount factor is introduced which helps to explain the repeated games as limited repeated games; however, the times are random before the game is closed.

Definition 1 (Discount Average Income). Let δ be the discount factor, then the discount average value of income stream $(R_1, R_2, R_3 \ldots)$ can be calculated by the following:

$$R=(1-\delta)\sum_{t=1}^{\infty}\delta^{t-1}R_t, \tag{4}$$

where R_t *(t=1,2, ...)* indicates the income of each stage game, and $\delta^{t-1}R_t$ *(t=1,2...)* is the discount value of each stage of the game.

Because the discount average income R is $(1-\delta)$ times that of the sum of all the discount values, the maximization of R is equal to the maximization of the sum of all discount values. Therefore, we can compare R at different stages of the game to the decision as to which strategy is selected.

Definition 2 (Game Type). The strategy game type is $G=\{N,S,u\}$, where $N=\{1,2,...,n\}$ represents the participants set, $S=\{S_1,S_2,...,S_n\}$ is a set of strategies, and $u=\{u_1,u_2,...,u_n\}$ is the set of the participants' incomes at each stage of the game. If G is repeated *T(T=1,2,...)* times, $G(T)$ indicates the repeated games during T time intervals. v_i indicates the expected income of participant i in repeated games $G(T)$, and can be obtained by the following:

$$v_i = u_i(s^1) + \delta u_i(s^2) + \ldots + \delta^{T-1} u_i(s^T) = \frac{R_i}{1-\delta} \quad , \tag{5}$$

where $u(s^t)$ is the Bernoulli return function, $s^t(T \geq t \geq 1)$ signifies the combination of actions at the *t-th* stage in repeated games, δ is the discount factor, R_i is the discount average income value of participant i. The income of each strategy can consequently be indicated by the following:

$$R_i = (1-\delta)\sum_{t=1}^{T} \delta^{t-1} u_i(s^t) \, . \tag{6}$$

In this paper, we set the selfish game as a four tuple, $G=\{N,S_{SN},S_{OS},u\}$, where $N=\{Nodes,OS\}$ represents all the participants in the games, including all nodes and our scheme in MANET. S_{SN} and S_{OS} represent the two participants in repeated games. $S_{SN}=\{Normal,Selfish\}$ and $S_{OS}=\{Trust,Isolate\}$ are defined as $\{N,S\}$ and $\{T,I\}$, respectively. The repeated games pay off matrix u indicates the preference of the selection strategy.

According to the preference of participants, both gains can be determined. $a \prec b$ is defined as b obtaining a greater gain than a. We will discuss both participants' income in repeated games on the basis of the four combinations of strategies. In our proposed scheme, nodes have maximum income when the selfish behaviour has succeeded and there is no isolation. When a node is normal and our scheme trusts it, this is an optimal selection in our work. On the other hand, when a node is isolated immediately when it exhibits selfish behaviour, the income is lowest in this condition. From the above description, we obtain the nodes' preference set as

$$SI \prec NI \prec NT \prec ST \, .$$

We use the same method to analyse our scheme. The income is maximized when the node is normal and our scheme trusts the node. The next scenario is to isolate the node when it is exhibiting selfish behaviour. The worst case occurs when a node is in the selfish state and our scheme does not discover it. Thus, our scheme's preference set is

$$TS \prec IN \prec IS \prec TN .$$

The incomes statement of pure strategy for both sides in the game is given in Table 1. Further, we know that: $e > a > c > g$ about nodes and $b > h > d > f$ about our scheme.

Table 1. Income statement of strategy in game

Strategy	Trust	Isolate
Normal	*(a,b)*	*(c,d)*
Selfish	*(e,f)*	*(g,h)*

2.2.2 Preventing Selfish Behaviour

When nodes that exhibit selfish behaviour are found in the detection phase, they are punished in this phase. Naturally, when a node is punished, then it must check its income. If it finds that the income is less than normal, it may become normal in the next time interval. Of course, some nodes may at times select a selfish strategy, but after a certain time interval, they will tend to select the normal strategy. Therefore, in the prevention phase, we can prevent selfish behaviour by forcing nodes to select the normal strategy in MANET. In order to do this, we propose an algorithm based on repeated games, which uses the limited punishment strategy.

On detecting the selfish behaviour, the next step is to identify the nodes that demonstrate selfish behaviour and prevent them from selecting a selfish strategy. In order to achieve this, we first determine which nodes exhibit selfish behaviour by comparing PFR and threshold value. Then, assuming that the selfish behaviour can be accurately detected using the adaptive threshold algorithm, we have the following theorem.

Theorem 2. In the game context of our scheme, if the discount factor $\delta > max\left(\frac{e-a}{e-c}, \frac{d}{b}\right)$, we can always determine a couple of suitable values of *(k, δ)* to dissuade the nodes from choosing the selfish strategy based on the limited punishment strategy.

***Proof*:** In a game situation, both sides have different income when they select different strategies. We must let both players in the game obtain maximization incomes when the nodes select normal strategy and our scheme choose trust strategy. Therefore, we can calculate the incomes at different stages, as introduced in Section 4.1, of each side in the repeated games, according to Table 1. We set the nodes' incomes as R_{N1}, R_{N2} and R_{N3}, where R_{N1} indicates the nodes' income when it chooses the normal strategy and our scheme chooses trust strategy in the k time intervals stages. R_{N2} and R_{N3} indicate the incomes of the nodes that choose the selfish strategy and our scheme, which selects the trust strategy at the first stage. The nodes then choose normal and selfish strategy but our scheme chooses isolate strategy in the following

k(k=1,2…) time intervals, respectively. According to (5) and (6), and Table 1, we can calculate those incomes as follows:

$$R_{N1} = a + a\delta + a\delta^2 + a\delta^3 + \ldots + a\delta^k + \ldots = \frac{a}{1-\delta} \quad , \tag{7}$$

$$R_{N2} = e + g\delta + g\delta^2 + \ldots + g\delta^k + \ldots = e + \frac{g}{1-\delta} \quad , \tag{8}$$

$$R_{N3} = e + c\delta + c\delta^2 + \ldots + c\delta^k + \ldots = e + \frac{c}{1-\delta} \quad , \tag{9}$$

where δ indicates discount factors. To prevent selfish behaviour, we need the nodes to choose the normal strategy in the games. In other words, we need to maximize the nodes' incomes. Hence, we need the result of δ when $R_{N1} > R_{N2}$ and $R_{N1} > R_{N3}$. Then, we calculate the result of the discount factor value from $\delta > \frac{e-a}{e-c}$. Using the same approach to calculate the income of our scheme, we can obtain another discount factor value as $\delta > \frac{d}{b}$. Consequently, the maximum discount factor is as follows:

$$\delta > max\left(\frac{e-a}{e-c}, \frac{d}{b}\right). \tag{10}$$

This value can satisfy the maximization condition for both incomes in the repeated games. Therefore, we can find a discount factor value that lets the nodes select the normal strategy. □

In the real environment, the nodes may sometimes choose a selfish strategy to escape detection, but when our scheme detects the selfish behaviour, it can start the punishment strategy in the subsequent *k(k=1,2, …)* stages. From Theorem 2, we know that the income of nodes is lowest when nodes select the selfish strategy; therefore, the nodes tend to select the normal strategy after a finite number of time intervals.

3 Performance Evaluations

We evaluated the performance of our scheme via simulation on NS-2[1]. The results show that our scheme can effectively detect and prevent selfish behaviour.

3.1 Simulation setup

In our simulation, we assume that a node has selected the normal strategy before the *t(t=2,3,…)* stage. Because our scheme

[1] http://www.isi.edu/nsnam/ns/ns-documentation.html.

selects its strategy in accordance with the last strategy which the node selected, our scheme can choose the trust strategy in the first stage. If the node chooses the selfish strategy at the *t-th* stage, the prevention phase uses a limited punishment strategy at the *(t+1)-th* stage to punish it.

We utilized the expected utility theory [17] to evaluate the value of the preferences set. The values were initialized as follows: *SI=0, NI=0.2, NT=0.4,* and *ST=0.5*, then multiplied by frequency span 10 to obtain every strategy's value. We also initialized *TN=0.3, IS=0.2, IN=0.1,*and *TS=0*, which resulted in the values in Table 1 changing to those shown in Table 2 after initialization.

Further, in accordance with Theorem 2, we calculated the discount factor $\delta = 1/3$, which indicates that the normal strategy of income is maximization in the *k(k=1,2,...)* time intervals punishment whatever the value of *k*. However, in real situations, we cannot take the value of *k* as infinite. We therefore applied the initialization values in Table 2 to (7), (8), and (9). The calculation results from those equations are as follows:

$$f_{N1}(k) = 4(1 + \delta + \delta^2 + \ldots + \delta^k) . \tag{11}$$

with $f_{N2} = 5$

$$f_{N3}(k) = 5 + 2\delta(1 + \delta + \delta^2 + \ldots + \delta^k) . \tag{12}$$

In order to find a pair of suitable values for *(k, δ),* we can calculate the value of δ_k, which indicates the value of discount factor δ on the condition of different *k(k=1,2,...)*. Then, if the condition is

$$f_k(\delta) = (f_{N1}(k) - f_{N3}(k)) > 0 . \tag{13}$$

We can define the deviation as follows:

$$\Delta\delta = \left|\delta_k - \delta_{k+1}\right| \quad . \tag{14}$$

Here, (14) indicates the change in δ about each pair of adjacent *k*. We consider that the pair of values of *(k, δ)* is effective when $\Delta\delta < 0.001$. We know that $\lim_{k\to\infty} \delta_k = 1/3$, and hence we set $\delta = 0.334$. We then apply this value in (3) to find a minimum value for *k(k=1,2...)*. In so doing, we obtain the minimum value *k=5* through calculation, giving us the pair of values as (5, 0.334).

The results obtained from the simulation in NS-2 indicate that our scheme is effective. Table 3 shows the parameters used in the NS-2 simulations. We simulated a network with a field size of 1000m× 1000m and 20 nodes. The nodes moved within the network space according to the random waypoint mobility model [18]. In this model, each node moves to a random location within the specified network area. When a node arrives at the target location, it stays in the position for a

period of time (pause time) before moving to another random location. In our simulation, the pause time was set to 0.5s. We evaluated an IEEE 802.11 wireless ad hoc network with the 20 nodes spreading out randomly in a given area. Among them, five nodes had selfish behaviour. The transmission radius of all nodes was 250 m.

Table 2. Initialization income statement

Strategy	Trust	Isolate
Normal	(4,3)	(2,1)
Selfish	(5,0)	(0,2)

In our simulation, we set three scenarios in accordance with selfish behaviour whether existed in the MANET. The first is SENDER within unselfish behaviour scenario, in which there is no selfish behaviour in the MANET but we used our scheme to detect and prevent selfish behaviour. And the second is SENDER within selfish behaviour scenario, in which some nodes were selfish in the MANET and we used our scheme to detect and prevent selfish behaviour. The last is NDP scheme within selfish behaviour scenarios, in which some nodes were selfish in the MANET but no detection and prevention scheme was used. We then compared the throughput and delay obtained in the three scenarios.

Table 3. Simulation parameters

Parameter	**Value**
No. of nodes	*20*
Type of channel	*Wireless*
Type of propagation	*Two Ray Ground*
Type of network interface queue	*Phy/wireless Phy*
Type of interface queue	*Queue/Drop Tail/PriQueue*
Type of antenna	*Antenna/OmniAntenna*
Type of protocol	*AODV*
Simulation time	*60min*
Packet size	*512B*
Terrain dimensions	*1000m×1000m*

3.2 Simulation results

In the simulation, we compare the throughput for the three scenarios and the results shown in Fig.2. From the figure, it is clear that the throughput in SENDER within unselfish behaviour scenario is lower than other two scenarios. In SENDER within selfish behaviour scenario, however, the throughput is close to SENDER within unselfish behaviour scenario and

lower than NDP scheme within selfish behaviour scenarios, because our scheme punishes any node in *k(k=1,2,...)* time intervals when it chooses the selfish strategy in one time interval. Weighing the gain and loss, the nodes choose the normal strategy in the repeated games. Hence, in our scheme within selfish behaviour scenario, the throughput approaches that of our scheme within unselfish behaviour scenario. So the throughput in SENDER within selfish behaviour scenario is effective.

Fig.3 shows the performance of SENDER within selfish behaviour scenario compared with SENDER within unselfish behaviour scenario and NDP scheme within selfish behaviour scenarios for varying delays in packet transmission. It can be seen that the delay for the NDP scheme within selfish behaviour scenarios is greater than that for the other two scenarios, whereas the delay with SENDER within selfish behaviour scenario is close to that of SENDER within unselfish behaviour scenario. In our scheme within selfish behaviour scenario, the delay also fluctuates because some nodes may only select selfish behaviour occasionally. They vary their strategy between normal and selfish. In comparison, the delay tends to be relatively stable in the presence of selfish behaviour in SENDER within selfish behaviour scenario. SENDER immediately punishes any node that exhibits selfish behaviour and the nodes compute their income under the selfish and normal strategy. When the nodes discover that the income under the normal strategy is more than that under the selfish strategy, they tend to select the normal strategy in the ensuing stages. Consequently, the delay is less than that of the NDP scheme within selfish behaviour scenarios, and similar to that of SENDER within unselfish behaviour scenario.

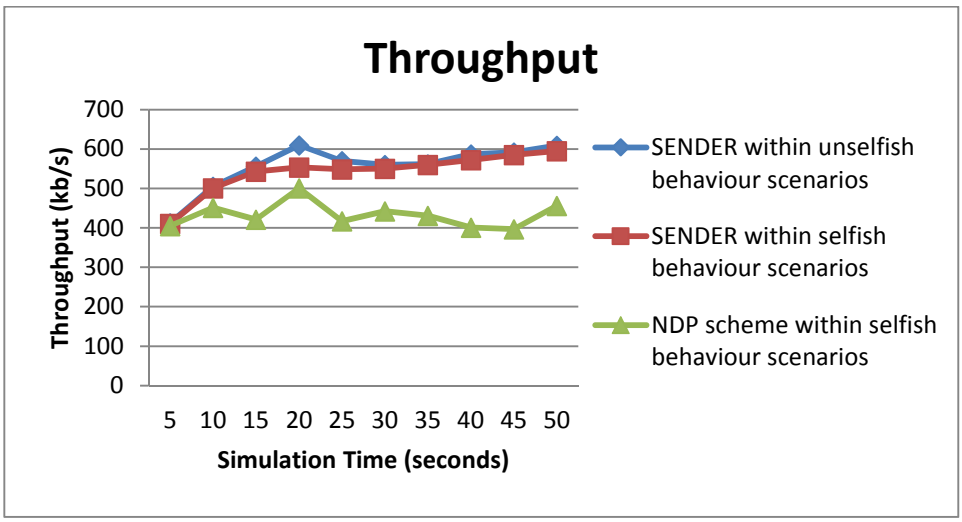

Fig. 2. Throughput in the MANET for the three scenarios. And it is clear that the throughput in SENDER within selfish behaviour scenario is effective.

4 Conclusion

In this paper, we proposed a scheme that detects and prevents selfish behaviour in mobile ad hoc networks by combining two techniques, adaptive threshold algorithm and repeated games. We detected selfish behaviour based on an adaptive

threshold algorithm by comparing each node's present PFR with a threshold value estimated according to the former statistics packets forwarding rate. Then, we used repeated games which punished nodes that selected a selfish strategy. When a node chose selfish behaviour in one stage, we punished it in the next *k* stages, resulting in the node's income becoming less than the normal condition. Consequently, nodes tended to choose the normal strategy after a certain time interval. The results of simulation of our scheme show that it is highly accurate in both selfish behaviour detection and prevention. However, the proposed scheme is less efficient when the network topology changes frequently. Thus, our future work will focus on methods of adapting our proposed scheme to dynamic network environments.

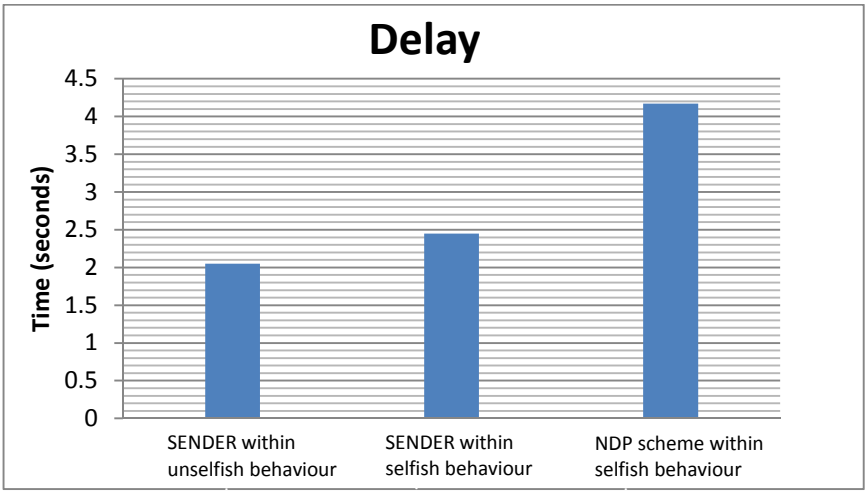

Fig. 3. Delay in the MANET for the three scenarios. It can be seen that the delay for SENDER within selfish behaviour scenario NDP is close to SENDER within unselfish behaviour scenario.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant No. 61202435 and 61272521 and the Natural Science Foundation of Beijing under Grant No.4132048.

References

[1] B. Sun, L. Osborne, Y. Xiao, and S. Guizani, "Intrusion detection techniques in mobile ad hoc and wireless sensor networks," *IEEE Wireless Communications,* vol. 14, pp. 56-63, 2007.

[2] S. Yokoyama, Y. Nakane, O. Takahashi, and E. Miyamoto, "Evaluation of the impact of selfish nodes in ad hoc networks and detection and countermeasure methods," in *7th International Conference on Mobile Data Management, (MDM 2006)* , pp. 95-100, 2006.

[3] S. Padiya, R. Pandit, and S. Patel, "Survey of Innovated Techniques to detect selfish nodes in MANET," *International Journal of Computer Networking, Wireless and Mobile Communications (IJCNWMC),* pp. 2250-1568, 2013.

[4] A. Rodriguez-Mayol and J. Gozalvez, "Reputation based selfishness prevention techniques for mobile ad-hoc networks," Springer *Telecommunication Systems,* pp. 1-15, 2013.

[5] L. Buttyán and J.-P. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," *Mobile Networks and Applications,* vol. 8, pp. 579-592, 2003.

[6] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *2003 IEEE 22th International Conference on Computer Communication (INFOCOM 2003)*, pp. 1987-1997, 2003.

[7] Y. Yoo and D. P. Agrawal, "Why does it pay to be selfish in a MANET?," *IEEE Wireless Communications,* vol. 13, pp. 87-97, 2006.

[8] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," *in 2000 ACM 6 th annual international conference on Mobile computing and networking*, pp. 255-265, 2000.

[9] Q. He, D. Wu, and P. Khosla, "SORI: a secure and objective reputation-based incentive scheme for ad-hoc networks," in *2004 IEEE Conference on Wireless Communications and Networking (WCNC 2004),*vol. 2, pp. 825-830.

[10] V. Srivastava, J. O. Neel, A. B. MacKenzie, R. Menon, L. A. DaSilva, J. E. Hicks*, et al.*, "Using game theory to analyze wireless ad hoc networks," *IEEE Communications Surveys and Tutorials,* vol. 7, pp. 46-56, 2005.

[11] C. Pandana, Z. Han, and K. R. Liu, "Cooperation enforcement and learning for optimizing packet forwarding in autonomous wireless networks," *IEEE Trans. on Wireless Communications,* vol. 7, pp. 3150-3163, 2008.

[12] Z. Han and H. V. Poor, "Coalition games with cooperative transmission: a cure for the curse of boundary nodes in selfish packet-forwarding wireless networks," *IEEE Trans. on Communications,* vol. 57, pp. 203-213, 2009.

[13] D. Debjit, K. Majumder, and A. Dasgupta. "Selfish Node Detection and Low Cost Data Transmission in MANET using Game Theory."*Procedia Computer Science,* vol. 54, pp. 92-101,2015.

[14] L. Tao and X. Ming-Sen, "Research of Avoid the Selfish Behavior in Mobile Ad Hoc Networks Based on Repeated Game," in *2012 4th International Conference on Multimedia Information Networking and Security (MINES 2012),* pp. 835-838, 2012.

[15] V. A. Siris and F. Papagalou, "Application of anomaly detection algorithms for detecting SYN flooding attacks," *Computer communications,* vol. 29, pp. 1433-1442, 2006.

[16] G. J. Mailath and L. Samuelson, "Repeated games and reputations: long-run relationships," *OUP Catalogue,* 2006.

[17] Yu-rui L. I. N., "Application of Desired Utility Function Theory to Library Administration," *College Mathematics,* 2: 039, 2008.

[18] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," *IEEE Trans. Mobile Computing,* vol. 2, pp. 257-269, 2003.