FITS: A Finite-Time Reputation System for Cooperation in Wireless Ad-Hoc Networks

Tingting Chen[1] Fan Wu[2] Sheng Zhong[1]
1 Computer Science and Engineering Department, SUNY Buffalo
{tchen9, szhong}@cse.buffalo.edu
2 Department of Electrical and Computer Engineering, UIUC
fwu@crhc.illinois.edu

Abstract—A wireless ad hoc network does not have an infrastructure and thus needs the cooperation of nodes in forwarding other nodes' packets. Reputation system is an effective approach to give nodes incentives to cooperate in packet forwarding. However, existing reputation systems either lack rigorous analysis, or have analysis in unrealistic models. In this paper, we propose FITS, the first reputation system that has rigorous analysis and *guaranteed* incentive compatibility in a *practical model*. FITS has two schemes: the first scheme is very simple, but needs a Perceived Probability Assumption (PPA) ; the second scheme uses more sophisticated techniques to remove the need for PPA. We show that both of these two FITS schemes have a subgame perfect Nash equilibrium in which the packet forwarding probability of every node is 1. Experimental results verify that FITS provides strong incentives for nodes to cooperate.

Index Terms—Keywords Ad hoc networks; Incentive Compatibility; Routing; Packet Forwarding.

1 Introduction

A wireless ad hoc network does not have an infrastructure. In such a network, the cooperation of nodes is needed for forwarding other nodes' packets. If nodes do not forward each other's packets, the entire wireless ad hoc network cannot function properly. Nevertheless, in civilian ad hoc networks, nodes belong to different users and thus have their own interests. Consequently, we need to give nodes incentives to make them cooperative.

There are mainly two approaches to give nodes incentives: reputation-based systems and credit-based systems (see Section 7 for examples of these two types of systems). In this paper, we focus on the reputation-based approach, which is highly efficient and has been effectively applied to wireless ad hoc networks.

The existing works on reputation systems suffer from one of two problems. First, most of them (e.g., [4], [5], [12], [15], [16]) do *not* have rigorous analysis of incentive-compatibility. Hence, it is not clear what guarantee for cooperation these reputation systems can provide. Second, although some other reputation systems [14], [19] do have rigorous analysis, their analyses are in *unrealistic models*. Therefore, in practice, their work cannot guarantee cooperation as well.

Specifically, the existing reputation systems which have rigorous analyses study the interaction between each pair of neighbor nodes, as an *infinite* repeated reputation game. Hence, their analyses of incentive compatibility can be valid *only if* the reputation game between each pair of nodes continues infinitely. However, in reality, the games are finite. Therefore, even though their analyses are mathematically correct for an infinite repeated game, they do not really guarantee cooperation in a finite repeated game, i.e., in reality.

To see more clearly the problem caused by a finite repeated game, let us consider a simple example of two selfish neighbor nodes, v_0 and v_1. Suppose v_0 knows that his game will end, say, after two hours (e.g., he is going to move out of the transmission range of his neighbors at that time). Recall that a finite repeated game consists of a finite number of stages. As a selfish node, his best strategy is to figure out the total number of stages in the repeated game and refuse to cooperate in the last stage, because no punishment can be made after that. Unfortunately, his neighbor v_1 is as smart as him. She is also aware of his best strategy. Consequently, given the strategy of v_0 that drops all packets of v_1 in the last stage, the best strategy for v_1 is not to cooperate in the last *two* stages, because no matter v_1 cooperates or not, her own packets are not forwarded by v_0 in the last stage, and if she chooses to not cooperate in the last two stages, no more punishment she will receive. This sounds pretty bad, but is still not the end of story. Considering the strategy of v_1, v_0 decides that he had better change his strategy to no cooperation in the last *three* stages ... Therefore, we will have a *cascade* of no cooperation, until finally both v_0 and v_1 decide not to cooperate in the entire game.

In fact, we can prove that, not only the existing reputation systems, but also any reputation system designed in the traditional way, will fail to provide incentive compatibility in the realistic model (see Theorem 1).

This paper was partly done while Fan Wu was a student at SUNY Buffalo. This work was supported by NSF CNS-0845149.

Digital Object Indentifier 10.1109/TC.2010.162

To overcome the difficulty in the realistic model, in this paper we propose FITS (FInite-Time reputation System). It is the first reputation system that has a proof of incentive compatibility in a practical model. FITS uses a novel technique called *Threat To Interfere* (TTI). The idea of TTI is very simple. We allow a node to threaten to interfere with his neighbor's communications after a finite reputation game if his neighbor does not cooperate in the last stage of the game. In this way, the cascade of no cooperation is prevented from its very beginning. Hence, all nodes will cooperate in the entire reputation games, and thus *no real interference is needed*. (For more details, see Section 4.)

Our contributions are briefly summarized below:

- First, we show that traditional reputation systems *cannot* provide any Subgame Perfect Nash Equilibrium (SPNE) solution in a finite reputation game. Therefore, it is necessary to enhance the reputation system approach.
- Second, we enhance the reputation system approach by introducing the novel TTI technique. Using this technique, we design the FITS-D scheme and show that, when the FITS-D scheme is used, under an assumption called Perceived Probability Assumption (PPA), there is a SPNE in which the forwarding probabilities of all nodes are close to 1.
- Third, we also propose the FITS-I scheme, which does not need the PPA. Without the PPA, we show that it also has a SPNE in which the forwarding probabilities of all nodes are close to 1.
- Finally, experiments verify that our FITS schemes provide strong incentives for nodes to cooperate.

The rest of this paper is organized as follows. Section 2 presents technical preliminaries. In Section 3, we show that traditional reputation systems cannot provide SPNE solutions when the reputation game has a finite number of stages. In Sections 4 and 5, we present the FITS-D and FITS-I schemes, respectively. Evaluation results are presented in Section 6, while related works are discussed in Section 7. We conclude in Section 8.

2 Technical Preliminaries

In this paper, we study a wireless ad hoc network in which the nodes are in promiscuous mode. That is, each node can hear the transmissions of its neighbors. Furthermore, each link in this network is bidirectional. That is, if a node can receive packets from another node, then the latter node can also receive packets from the former node. Like in [14], we assume that nodes are selfish *not malicious*. That is, each node is interested only in maximizing its own utility.

Same as in a lot of previous works [12], [14], [15], [19], we consider a *hop-by-hop* reputation system and isolate each pair of neighbors (v_0, v_1). We study the *reputation game* between them. (What we describe in the rest of this section is the *basic model* of the reputation game, which is used to model *traditional reputation systems* in this paper. In Section 4, the model is extended so that it can model reputation systems using a newly introduced technique.) The reputation game is divided into T stages, where $T > 0$ is a finite number. In each stage, a sufficiently large number of packets are transmitted between v_0 and v_1.

Formally, our reputation game is a finite repeated game. The players of this game are the pair of neighbor nodes, v_0 and v_1. In each stage of the game, there is an action set available to player v_i ($i \in \{0, 1\}$): $A_i = \{a_i | 0 \leq a_i \leq 1\}$. In stage t, player v_i chooses an action $a_{i,t}$ from A_i. Intuitively, $a_{i,t}$ is the probability that v_i forwards v_{1-i}'s packets in stage t. The utility of player v_i in stage t is decided by the actions of both players:

$$u_{i,t} = a_{1-i,t}u - a_{i,t}c,$$

where u is the amount of benefit player v_i can receive if all its packets are forwarded by v_{1-i} in the entire stage, and c is the amount of cost needed by v_i for forwarding all v_{1-i}'s packets in the entire stage. In this paper, we assume that all nodes have the same (u, c) in all stages. Clearly, $u > c > 0$.

The utility of player v_i in the entire reputation game is the sum of its utilities in all stages:

$$u_i = \sum_{t=1}^{T} u_{i,t}.$$

Note that u_i is a function of the two players' actions in all stages, sometimes written as $u_i((a_{0,1}, a_{1,1}), \ldots, (a_{0,T}, a_{1,T}))$. However, unlike in a standard repeated game, in our reputation game, the actions of each player are *invisible* to the other player. That is, player v_i cannot directly see player v_{1-i}'s actions $a_{1-i,t}$ (for $t = 1, \ldots, T$), because there are collisions. Suppose p_c is the probability that a packet forwarded by v_i cannot be overheard by v_{1-i} due to collision. (As in previous work, we assume all nodes have the same p_c in all stages.) Then, what player v_i can see in stage t is a *perceived probability* that player v_{1-i} forwards its packets:

$$\hat{a}_{1-i,t} = (1 - p_c)a_{1-i,t}.$$

In the basic model, we also call $\hat{a}_{1-i,t}$ the perceived action of player v_{1-i}. Previous work [14] has introduced an assumption that we call the *Perceived Probability Assumption* (PPA): both players (v_0 and v_1) can see both perceived actions ($\hat{a}_{0,t}$ and $\hat{a}_{1,t}$) in every stage. Note that here the *actions* refer to the actions (i.e., forwarding probabilities) defined in the finite repeated game. In this paper, our FITS system has two schemes, FITS-D which uses the PPA, and FITS-I which does not.

Consequently, our definition of strategy is also different from that in a standard repeated game. Based on whether the PPA assumption is used or not, we distinguish two types of strategies: PPA-dependent strategies and PPA-independent strategies. Here a PPA-dependent strategy s_i for player v_i is a function defined on all possible histories of both players' perceived actions in stages 1 through t, where $t \leq T - 1$, together with other information v_i obtains in these stages. For each such history, s_i specifies an action for player v_i to take in the next stage. Formally, if v_i uses strategy s_i, the action v_i should take in stage $t+1$ is $s_i((\hat{a}_{i,1}, \hat{a}_{1-i,1}, I_{i,1}), \ldots, (\hat{a}_{i,t}, \hat{a}_{1-i,t}, I_{i,t}))$, where $(\hat{a}_{i,1}, \hat{a}_{1-i,1}), \ldots, (\hat{a}_{i,t}, \hat{a}_{1-i,t})$ are

the perceived actions of the two players in the first t stages, and $(I_{i,1}, \ldots, I_{i,t})$ is other information v_i obtains from v_{i-1} about its forwarding probabilities in stage 1 through t. (If there is no other information, $(I_{i,1}, \ldots, I_{i,t}) = (\perp, \ldots, \perp)$, where $\perp$ is a special symbol denoting no information.) In particular, in the first stage, the history is an empty history (also denoted by $\perp$). Thus, $s_i(\perp)$ is the action v_i should take in the first stage if v_i is using PPA-dependent strategy s_i.

Besides PPA-dependent strategies, we also study PPA-independent strategies. Here a PPA-independent strategy s_i for player v_i is a function defined on all possible histories of v_i's own actions and v_{1-i}'s perceived actions in stages 1 through t, where $t \leq T-1$, together with other information v_i obtain in these stages. For each such history, s_i specifies an action for player v_i to take in the next stage. Formally, $s_i((a_{i,1}, \hat{a}_{1-i,1}, I_{i,1}), \ldots, (a_{i,t}, \hat{a}_{1-i,t}, I_{i,t}))$ is the action v_i should take in stage $t+1$ if v_i is using strategy s_i, where $(a_{i,1}, \ldots, a_{i,t})$ are the actions taken by v_i in the first t stages, and $(\hat{a}_{1-i,1}, \ldots, \hat{a}_{1-i,t})$ are the perceived actions of v_{1-i} in the first t stages. Note that, as we have mentioned, $(I_{i,1}, \ldots, I_{i,t})$ is other information v_i obtains in stage 1 through t; it does not include the perceived probabilities. In particular, $s_i(\perp)$ is the action v_i should take in the first stage if v_i is using PPA-independent strategy s_i.

Let $s = (s_0, s_1)$ be a profile of PPA-dependent or PPA-independent strategies. If the two players are using this strategy profile, then the actions they take in all strategies can be easily determined. These actions are called the *outcome* of s, denoted by $O(s)$. The utility of player v_i when s is used by the two players is defined as the utility of v_i when $O(s)$ are taken by the two players. Denote this utility by $u_i(s)$. Consequently, $u_i(s) = u_i(O(s))$.

2.1 SPNE and One Deviation Theorem

The main objective of this paper is to design a system that converges to a SPNE in which both players forward packets with probability 1. To define SPNE, we first briefly review the definition of subgame.

A subgame of our reputation game starts in stage t_0 ($1 \leq t_0 \leq T$) of the reputation game and ends in stage T of the reputation game. Hence a subgame can be viewed as a finite repeated game of $T-(t_0-1)$ stages. We can identify a subgame with its *initial history*, *i.e.*, what happened in stages 1 through t_0-1. Suppose the initial history is $h = ((a_{0,1}, a_{1,1}), \ldots, (a_{0,t_0-1}, a_{1,t_0-1}))$. Then the corresponding subgame is denoted by $\Gamma(h)$.

The players of $\Gamma(h)$ are the two players of the reputation game, v_0 and v_1; the action set available to each player in the subgame is also the same as that in the reputation game. The utility of player v_i in the subgame $\Gamma(h)$ is the sum of v_i's utilities in all stages of $\Gamma(h)$:

$$u_{i,\Gamma(h)} = \sum_{t=t_0}^{T} u_{i,t}.$$

Since $u_{i,\Gamma(h)}$ can be determined by the actions taken in stages t_0 through T, or by the strategy profile the two players use, we can write it as $u_{i,\Gamma(h)}((a_{0,t_0}, a_{1,t_0}), \ldots, (a_{0,T}, a_{1,T}))$ or $u_{i,\Gamma(h)}(s_0, s_1)$.

A strategy (resp., strategy profile) in the reputation game induces a strategy (resp., strategy profile) in any subgame. For simplicity of notation, we often use the same symbol to represent a strategy (resp., strategy profile) and its induced strategy (resp., strategy profile).

In the reputation game, a PPA-dependent (resp., PPA-independent) strategy profile $s^* = (s_0^*, s_1^*)$ is a Nash equilibrium (NE) if for all $i \in \{0,1\}$, for all PPA-dependent (resp., PPA-independent) strategy s_i,

$$u_i(s^*) \geq u_i(s_i, s_{1-i}^*).$$

Similarly, in subgame $\Gamma(h)$, a PPA-dependent (resp., PPA-independent) strategy profile s^* induces a NE if for all $i \in \{0,1\}$, for all PPA-dependent (resp., PPA-independent) strategy s_i,

$$u_{i,\Gamma(h)}(s^*) \geq u_{i,\Gamma(h)}(s_i, s_{1-i}^*).$$

We say a PPA-dependent (resp., PPA-independent) strategy profile s^* is a SPNE of the reputation game if for every subgame $\Gamma(h)$, s^* induces a NE in $\Gamma(h)$.

Since a finite repeated game is also a finite-horizon extensive game, we can apply the famous One Deviation Theorem [20] to our reputation game. In the context of our reputation game, the theorem states that s^* is a SPNE if and only if for each player, a deviation from s^* in any single stage cannot bring more utility to the player in the subgame starting from the stage of deviation. Formally, s^* is a SPNE if and only if for $i \in \{0,1\}$, for every subgame $\Gamma(h)$, for all $a_i \in A_i$,

$$u_{i,\Gamma(h)}(s^*) \geq u_{i,\Gamma(h)}(s_i^*|_{h \to a_i}, s_{1-i}^*),$$

where $s_i^*|_{h \to a_i}$ denotes a strategy in which action a_i is taken right after history h, and the action specified by s_i^* is used after any other history.

3 Impossibility Of SPNE Solution for Traditional Reputation Systems

Given the basic model of finite reputation game, now we show that it is impossible to design a traditional reputation system (in which punishment can only be dropping packets) that provides a SPNE solution in this model. Intuitively, this impossibility result means that we cannot use *any* traditional reputation systems to achieve strong incentive compatibility in finite reputation games. More precisely, if a traditional reputation system is used in a finite reputation game, then in all SPNE, both players of the game drop all packets of each other. A formal theorem and proof are given below.

Theorem 1: In the basic model of reputation game (as defined in Section 2), for all SPNE s^*, and all history h such that its length $|h| < T$, the outcome of s^* in the subgame $\Gamma(h)$ is

$$O_{\Gamma(h)}(s^*) = (0,0)^{T-|h|} = ((0,0), \ldots, (0,0)).$$

In particular, this implies that, $\forall i \in \{0,1\}$, $u_i(s^*) = 0$.

Proof: Here we prove this theorem for the case s^* is PPA-dependent. If s^* is PPA-independent, we can easily obtain a similar proof.

Note that it suffices to show that for all SPNE s^*, all history h such that its length $|h| < T$, and all $i \in \{0,1\}$, we have $s_i^*(h) = 0$.

We prove this theorem by induction on $|h|$. If $|h| = T-1$, by the definition of SPNE, $s^*(h)$ is a NE in stage T. Hence,

$$s_i^*(h) = \arg\max_{a_{i,T}}(a_{1-i,T}u - a_{i,T}c) = 0.$$

Now, suppose that the above proposition is true for $|h| \geq h_0$, where $h_0 \in N^+$ and $1 \leq h_0 \leq T-1$. We show that it is true for $|h| = h_0 - 1$ as well. Let $h' = (h, (\hat{a}_i, \hat{s}_{1-i}^*(h), \perp))$, where $\hat{a}_i$ and $\hat{s}_{1-i}^*(h)$ are the perceived probabilities of a_i and $s_{1-i}^*(h)$, respectively. For all $i \in \{0,1\}$, all a_i, we have

$$\begin{aligned} u_{i,h_0}(a_i, s_{1-i}^*(h)) &= u_{i,\Gamma(h)}(s_i^*|_{h\to a_i}, s_{1-i}^*) \\ &\quad -u_{i,\Gamma(h')}(s^*) \\ &\leq u_{i,\Gamma(h)}(s^*) - u_{i,\Gamma(h')}(s^*) \\ &= u_{i,\Gamma(h)}(s^*) - \sum_{t=h_0+1}^{T}(a_{1-i,t}^{h'}u \\ &\quad -a_{i,t}^{h'}c), \end{aligned}$$

where $((a_{i,h_0+1}^{h'}, a_{1-i,h_0+1}^{h'}), \ldots, (a_{i,T}^{h'}, a_{1-i,T}^{h'}))$ is the outcome of s^* in $\Gamma(h')$.

By the induction assumption, we get that, for all $i \in \{0,1\}$, for $t = h_0+1, \ldots, T$, $a_{i,t}^{h'} = 0$. Therefore,

$$u_{i,h_0}(a_i, s_{1-i}^*(h)) \leq u_{i,\Gamma(h)}(s^*). \quad (1)$$

Let $h'' = (h, (\hat{s}^*(h), \perp))$, where $\hat{s}^*(h)$ is the perceived probabilities of $s^*(h)$. Similar to the above, we can obtain that

$$u_{i,h_0}(s^*(h)) = u_{i,\Gamma(h)}(s^*) - \sum_{t=h_0+1}^{T}(a_{1-i,t}^{h''}u - a_{i,t}^{h''}c),$$

where $((a_{i,h_0+1}^{h''}, a_{1-i,h_0+1}^{h''}), \ldots, (a_{i,T}^{h''}, a_{1-i,T}^{h''}))$ is the outcome of s^* in $\Gamma(h'')$. By the induction assumption, we also have that, for all $i \in \{0,1\}$, for $t = h_0+1, \ldots, T$, $a_{i,t}^{h''} = 0$. Therefore,

$$u_{i,h_0}(s^*(h)) = u_{i,\Gamma(h)}(s^*). \quad (2)$$

Combining (1) and (2), we get that

$$u_{i,h_0}(a_i, s_{1-i}^*(h)) \leq u_{i,h_0}(s^*(h)). \quad (3)$$

Since (3) is true for all a_i, it must be the case that

$$s_i^*(h) = \arg\max_{a_i} u_{i,h_0}(a_i, s_{1-i}^*(h)) = 0,$$

which means the proposition is true for $|h| = h_0 - 1$. □

4 Extended Model and FITS-D Scheme

Since traditional reputation systems cannot provide SPNE solutions, in this section, we introduce a new technique and use it to significantly enhance the incentive compatibility of reputation systems. This technique is called Threat To Interfere (TTI).

The main idea of TTI is that a node can *threaten* to drop a neighbor's packets *and* to interfere with the neighbor's communications. In contrast, in traditional reputation systems, a node can only threaten to drop its neighbor's packets, which is the only way to force its neighbor to forward packets. Because the combined threat (of packet dropping and communication interference) is stronger than a single threat of packet dropping, it can be a more effective method to force the neighbor to forward packets. Note that TTI does not require real interference with communications. In fact, our analysis will show that there is no real interference at all when the system converges to a stable state.

4.1 Extended Model

To allow formal analysis of TTI, we extend the basic model by introducing an additional action ITF to each node v_i, which means v_i drops all packets of v_{1-i} and interferes with the communications of v_{1-i} (using dumb packets containing its own identity). Formally, we replace the action set A_i with $A_i' = A_i \cup \{\mathsf{ITF}\}$. Correspondingly, the perceived action of player v_i in stage t is redefined as

$$\hat{a'}_{i,t} = \begin{cases} \mathsf{ITF} & \text{if } a'_{i,t} = \mathsf{ITF} \\ (1-p_c)a'_{i,t} & \text{otherwise,} \end{cases}$$

(We note that, in the extended model, *perceived actions* are different from *perceived probabilities*.) Furthermore, the utility of player v_i in stage t is redefined as

$$u'_{i,t} = \begin{cases} a_{1-i,t}u - a_{i,t}c & \text{if } a_{i,t} \neq \mathsf{ITF} \\ & \text{and } a_{1-i,t} \neq \mathsf{ITF} \\ a_{1-i,t}u & \text{if } a_{i,t} = \mathsf{ITF} \\ & \text{and } a_{1-i,t} \neq \mathsf{ITF} \\ u_{\mathsf{INT}} & \text{if } a_{1-i,t} = \mathsf{ITF}. \end{cases}$$

where $u_{\mathsf{INT}} < 0$. Here we assume that $u_{\mathsf{INT}} < -u$. Intuitively, this means the loss caused by communication interference is greater than the benefit of forwarding packets. The utility in the entire game is still the sum of utilities in all stages:$u'_i = \sum_{t=1}^{T} u'_{i,t}$.

4.2 FITS-D Scheme

Now we design the FITS-D scheme using the TTI technique.

The first idea of our design is that a node following the scheme can simply choose the "worst" action appeared in the history as its new action. That is, assuming no node has ever taken the action ITF, then a cooperative node should take the lowest forwarding probability ever appeared in the history of the game. (Note that this lowest forwarding probability in the history can be easily computed by a node: it knows all its own forwarding probabilities in the history; by the PPA, it also knows all the perceived probabilities in the history, which allows it to compute the forwarding probabilities of the other node in the history.) The advantage of such a scheme is clear: if a misbehaving node drops packets of its neighbor with a certain probability in a stage, then its cooperative neighbor drops its packets with at least the same probability in all future stages. This threat of punishment is strong enough to prevent a misbehaving node from dropping packets.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

Recall the problem caused by the finite repeated game model, which we have discussed in Section 1. To solve this problem, we introduce an additional stage in which there is *no* data transmission, and use our TTI technique in this stage. Intuitively, we can view this stage as a brief extension period of the reputation game. In this additional stage, a cooperative node examines whether the other node was cooperative in the last stage of data transmission. If so, the cooperative node does nothing; otherwise, the cooperative node interferes with the other node's communications.

In this way, we can effectively prevent packet dropping in the last stage of data transmission (i.e., in the second last stage of the entire game), because communication interference is a strong threat to any misbehaving node. Furthermore, we do not need to worry about misbehavior in the additional stage (i.e., in the last stage of the entire game), because a misbehaving node cannot benefit from cheating in a stage that has no data transmission at all. (For detailed analysis, see Theorem 2 and its proof.)

A detailed description of the FITS-D scheme is given in Figure. 1.

$\triangleright$ (v_i, v_{1-i}) is a pair of neighbors.
$\triangleright$ t is the index of stage in the reputation game.

if $t = 1$,
  then $a_{i,t} \leftarrow 1$;
else if $\exists t' \in \{1, \ldots, t-1\}$ s.t. $\hat{a}_{i,t'} = \mathsf{ITF}$
or $\hat{a}_{1-i,t'} = \mathsf{ITF}$,
  then $a_{i,t} \leftarrow \mathsf{ITF}$;
else if $t < T$,
  then $\rho_{i,t} \leftarrow \frac{\min_{1 \le t' \le t-1}\{\hat{a}_{i,t'}, \hat{a}_{1-i,t'}\}}{\hat{a}_{i,t-1}}$;
    $a_{i,t} \leftarrow \rho_{i,t} a_{i,t-1}$;
else if $\hat{a}_{1-i,t-1} < \min_{1 \le t' \le t-2}\{\hat{a}_{i,t'}, \hat{a}_{1-i,t'}\}$
  then $a_{i,t} \leftarrow \mathsf{ITF}$;
else $a_{i,t} \leftarrow 0$.

Fig. 1. FITS-D scheme

4.3 Analysis of FITS-D Scheme

Now we present a formal analysis of the FITS-D scheme. We show it is a SPNE solution to the packet forwarding problem. More precisely, we show that, in a SPNE, nodes forward packets with probability 1 in all stages except the last stage that does not have any data to forward.

Theorem 2: In the extended model of reputation game (as defined in Section 4.1), if the FITS-D scheme is used, assuming $u > 2c$, then there is a SPNE s^* such that

$$O(s^*) = (1,1)^{T-1}(0,0) = ((1,1), \ldots, (1,1), (0,0)),$$

which implies that, for all $i \in \{0,1\}$, $u_i(s^*) = (T-1)(u-c)$.

Proof: It is easy to see that, if all participants follow the FITS-D scheme, then the outcome is $(1,1)^{T-1}(0,0)$. Hence, denote by s^* the strategy profile defined by our FITS-D scheme; it suffices to show that s^* is a SPNE.

Let $h = ((\hat{a}_{0,1}, \hat{a}_{1,1}, \perp), \ldots, (\hat{a}_{0,|h|}, \hat{a}_{1,|h|}, \perp))$. Let $t = |h|$. Let $s_i = s_i^*|_{h \to a_i}$, where $a_i \neq s_i^*(h)$. We distinguish a number of cases to analyze the relationship between $u_{i,\Gamma(h)}(s^*)$ and $u_{i,\Gamma(h)}(s_i, s_{1-i}^*)$ (We will use notation $u_{\Gamma(h)}(s^*)$ (resp. $u_{\Gamma(h)}(s_i, s_{1-i}^*)$) instead of $u_{i,\Gamma(h)}(s^*)$ (resp. $u_{i,\Gamma(h)}(s_i, s_{1-i}^*)$) in this proof for convenience):

Case A: There exists $t' \in \{1, \ldots, t\}$ s.t. $\hat{a}_{i,t'} = \mathsf{ITF}$ or $\hat{a}_{1-i,t'} = \mathsf{ITF}$. In this case,

$$O_{\Gamma(h)}(s^*) = (\mathsf{ITF}, \mathsf{ITF})^{\mathsf{T-t}}$$
$$\Rightarrow \quad u_{\Gamma(h)}(s^*) = (T-t)u_{\mathsf{INT}},$$

and

$$O_{\Gamma(h)}(s_i, s_{1-i}^*) = (a_i, \mathsf{ITF})(\mathsf{ITF}, \mathsf{ITF})^{\mathsf{T-t-1}}$$
$$\Rightarrow \quad u_{\Gamma(h)}(s_i, s_{1-i}^*) = (T-t)u_{\mathsf{INT}}.$$

Hence, $u_{\Gamma(h)}(s^*) = u_{\Gamma(h)}(s_i, s_{1-i}^*)$.

Case B: $t = 0$. In this case,

$$O_{\Gamma(h)}(s^*) = (1,1)^{T-t-1}(0,0)$$
$$\Rightarrow \quad u_{\Gamma(h)}(s^*) = (T-t-1)(u-c),$$

We have two subcases.

Case B.1: $a_i = \mathsf{ITF}$.

$$O_{\Gamma(h)}(s_i, s_{1-i}^*) = (\mathsf{ITF}, 1)(\mathsf{ITF}, \mathsf{ITF})^{\mathsf{T-t-1}}$$
$$\Rightarrow \quad u_{\Gamma(h)}(s_i, s_{1-i}^*) = (T-t-1)u_{\mathsf{INT}} + u.$$

Since $u_{\mathsf{INT}} < -u < -c$, we can easily obtain that

$$u_{\Gamma(h)}(s^*) \geq u_{\Gamma(h)}(s_i, s_{1-i}^*).$$

Case B.2: $a_i \neq \mathsf{ITF}$.

$$O_{\Gamma(h)}(s_i, s_{1-i}^*) = (a_i, 1)(a_i, a_i)^{T-t-2}(0,0)$$
$$\Rightarrow \quad u_{\Gamma(h)}(s_i, s_{1-i}^*) = (T-t-2)a_i(u-c) + u - a_i c.$$

Since $u > 2c$, we can easily obtain that

$$u_{\Gamma(h)}(s^*) \geq u_{\Gamma(h)}(s_i, s_{1-i}^*).$$

Case C: $\forall t' \in \{1, \ldots, t\}$, $\hat{a}_{i,t'} \neq \mathsf{ITF}$ and $\hat{a}_{1-i,t'} \neq \mathsf{ITF}$, and $0 < t < T-1$. In this case,

$$O_{\Gamma(h)}(s^*)$$
$$= \quad (\tfrac{a_{i,t-1}}{\hat{a}_{i,t-1}} \cdot \min_{1 \le t' \le t}\{\hat{a}_{i,t'}, \hat{a}_{1-i,t'}\},$$
$$\tfrac{a_{i,t-1}}{\hat{a}_{i,t-1}} \cdot \min_{1 \le t' \le t}\{\hat{a}_{i,t'}, \hat{a}_{1-i,t'}\})^{T-t-1}(0,0)$$
$$\Rightarrow \quad u_{\Gamma(h)}(s^*)$$
$$= \quad (T-t-1)(u-c) \cdot a_{i,t-1}$$
$$\cdot \min_{1 \le t' \le t}\{\hat{a}_{i,t'}, \hat{a}_{1-i,t'}\}/(\hat{a}_{i,t-1}).$$

We have four subcases.

Case C.1: $a_i = \mathsf{ITF}$. Thus we have

$$O_{\Gamma(h)}(s_i, s_{1-i}^*)$$
$$= \quad (\mathsf{ITF}, \tfrac{\mathsf{a_{i,t-1}}}{\hat{\mathsf{a}}_{\mathsf{i,t-1}}} \cdot \min_{1 \le t' \le t}\{\hat{\mathsf{a}}_{\mathsf{i,t'}}, \hat{\mathsf{a}}_{\mathsf{1-i,t'}}\})$$
$$(\mathsf{ITF}, \mathsf{ITF})^{\mathsf{T-t-1}}$$
$$\Rightarrow \quad u_{\Gamma(h)}(s_i, s_{1-i}^*)$$
$$= \quad \tfrac{a_{i,t-1}}{\hat{a}_{i,t-1}} \cdot \min_{1 \le t' \le t}\{\hat{a}_{i,t'}, \hat{a}_{1-i,t'}\}u$$
$$+(T-t-1)u_{\mathsf{INT}}.$$

Clearly, $u_{\Gamma(h)}(s_i, s^*_{1-i}) < 0 < u_{\Gamma(h)}(s^*)$.

Case C.2: $a_i \neq \mathsf{ITF}$ and $a_i > s^*_i(h)$.

$$\begin{array}{rc} & O_{\Gamma(h)}(s_i, s^*_{1-i}) \\ = & (a_i, \frac{a_{i,t-1}}{\hat{a}_{i,t-1}} \cdot \min_{1\le t'\le t}\{\hat{a}_{i,t'}, \hat{a}_{1-i,t'}\}) \\ & (\frac{a_{i,t-1}}{\hat{a}_{i,t-1}} \cdot \min_{1\le t'\le t}\{\hat{a}_{i,t'}, \hat{a}_{1-i,t'}\}, \\ & \frac{a_{i,t-1}}{\hat{a}_{i,t-1}} \cdot \min_{1\le t'\le t}\{\hat{a}_{i,t'}, \hat{a}_{1-i,t'}\})^{T-t-2} \\ & (0,0) \\ \Rightarrow & u_{\Gamma(h)}(s_i, s^*_{1-i}) = (T-t-2)(u-c)\cdot a_{i,t-1} \\ & \cdot \min_{1\le t'\le t}\{\hat{a}_{i,t'}, \hat{a}_{1-i,t'}\}/(\hat{a}_{i,t-1}) - a_i c \\ & +u\cdot a_{i,t-1}\cdot \min_{1\le t'\le t}\{\hat{a}_{i,t'}, \hat{a}_{1-i,t'}\}/(\hat{a}_{i,t-1}). \end{array}$$

Hence, $u_{\Gamma(h)}(s^*) > u_{\Gamma(h)}(s_i, s^*_{1-i})$.

Case C.3: $a_i \neq \mathsf{ITF}$ and $a_i < s^*_i(h)$, and $t < T-2$.

$$\begin{array}{rc} & O_{\Gamma(h)}(s_i, s^*_{1-i}) \\ = & (a_i, \frac{a_{i,t-1}}{\hat{a}_{i,t}} \cdot \min_{1\le t'\le t}\{\hat{a}_{i,t'}, \hat{a}_{1-i,t'}\}) \\ & (a_i, a_i)^{T-t-2}(0,0) \\ \Rightarrow & u_{\Gamma(h)}(s_i, s^*_{1-i}) = (T-t-2)(u-c)\cdot a_i - a_i c \\ & +u\cdot a_{i,t-1}\cdot \min_{1\le t'\le t}\{\hat{a}_{i,t'}, \hat{a}_{1-i,t'}\}/(\hat{a}_{i,t-1}). \end{array}$$

Since $u > 2c$ and $t < T-2$, we can easily obtain that $u_{\Gamma(h)}(s^*) > u_{\Gamma(h)}(s_i, s^*_{1-i})$.

Case C.4: $a_i \neq \mathsf{ITF}$ and $a_i < s^*_i(h)$, and $t = T-2$.

$$\begin{array}{rc} & O_{\Gamma(h)}(s_i, s^*_{1-i}) \\ = & (a_i, \frac{a_{i,t-1}}{\hat{a}_{i,t-1}} \cdot \min_{1\le t'\le t}\{\hat{a}_{i,t'}, \hat{a}_{1-i,t'}\})(\mathsf{ITF}, \mathsf{ITF}) \\ \Rightarrow & u_{\Gamma(h)}(s_i, s^*_{1-i}) = u_{\mathsf{INT}} - a_i c \\ & +u\cdot a_{i,t-1}\cdot \min_{1\le t'\le t}\{\hat{a}_{i,t'}, \hat{a}_{1-i,t'}\}/(\hat{a}_{i,t-1}). \end{array}$$

Since $u_{\mathsf{INT}} < -u < -c$, we can easily obtain that $u_{\Gamma(h)}(s^*) > u_{\Gamma(h)}(s_i, s^*_{1-i})$.

Case D: $\forall t' \in \{1,\ldots,t-1\}$, $\hat{a}_{i,t'} \neq \mathsf{ITF}$ and $\hat{a}_{1-i,t'} \neq \mathsf{ITF}$, and $t = T-1$. We have two subcases.

Case D.1: $s^*_{1-i}(h) = \mathsf{ITF}$. Hence, we have

$$\begin{array}{rcl} u_{\Gamma(h)}(s^*) & = & u_{\mathsf{INT}} \\ & = & u_{\Gamma(h)}(s_i, s^*_{1-i}). \end{array}$$

Case D.2: $s^*_{1-i}(h) = 0$. If $s^*_i(h) = \mathsf{ITF}$, then

$$\begin{array}{rcl} u_{\Gamma(h)}(s^*) & = & 0 \\ & \ge & u_{\Gamma(h)}(s_i, s^*_{1-i}). \end{array}$$

If $s^*_i(h) \neq \mathsf{ITF}$, then, since $t = T-1$, from the scheme we know that $s^*_i(h) = 0$. Hence,

$$\begin{array}{rcl} u_{\Gamma(h)}(s^*) & = & -s^*_i(h)\cdot c \\ & = & 0 \\ & \ge & u_{\Gamma(h)}(s_i, s^*_{1-i}). \end{array}$$

Hence, we always have $u_{\Gamma(h)}(s^*) \ge u_{\Gamma(h)}(s_i, s^*_{1-i})$. So, by the One Deviation Theorem of finite repeated game, we know that s^* is a SPNE. □

5 FITS-I SCHEME

The FITS-D scheme provides strong incentive for packet forwarding as long as the PPA is valid. However, in reality there exist scenarios in which the perceived actions of the two nodes can not be seen by both of them, i.e. the PPA is not valid.[1] In these cases, we need to use a scheme that is independent from the PPA. In this section, we develop such a scheme.

5.1 Scheme for PPA-Independence

The main idea to achieve PPA-independence is to use (claimed) real forwarding probabilities instead of perceived forwarding probabilities. We assume that, at the beginning of each stage t, each node v_i claims its real forwarding probability in this stage to the other node v_{1-i}. Denote this claim by $\overline{a}_{i,t}$. If v_i is cooperative, we must have $\overline{a}_{i,t} = a_{i,t}$. (Of course, if v_i is a misbehaving node, then we may have $\overline{a}_{i,t} \neq a_{i,t}$.) Using these claimed probabilities, we can establish a scheme that is similar to the FITS-D scheme but does not need any perceived probability, as long as we can make sure there is no false claim of forwarding probability.

To prevent players from making false claims, we divide each stage into m small time intervals. Each player is responsible for keeping a transcript of the packets it has forwarded in the *current* time interval. At the end of each time interval, with probability p_v node v_i chooses to verify the forwarding probability of v_{1-i} in this time interval. (To reduce the overall overheads of computation and communication, v_i does not verify the forwarding probability in all time intervals. Instead, v_i randomly picks some time intervals and verifies the forwarding probability in these time intervals only.) If v_i chooses to verify the forwarding probability in a time interval, it requests the transcript from v_{1-i}. Then it uses this transcript to decide whether v_{1-i} has really forwarded packets with probability $\overline{a}_{1-i,t}$. (There are various ways to design a verification algorithm for this purpose. We give one example algorithm in Section 5.3.) If cheating is detected, v_i punishes v_{1-i} by dropping its packets and interfering with its communications in all future stages.[2] If no cheating is detected in this time interval, or if this time interval is not chosen for verification of the forwarding probability, then the transcript can be discarded at the beginning of next time interval to save space.

The details of FITS-I scheme are shown in Fig. 2 and 3. In this scheme, we use an algorithm VerProb() to verify the claimed forwarding probabilities. This algorithm works by comparing two transcripts $\overline{X}_{1-i}$ and $X_{1-i,i}$. Here both of these two transcripts are supposed to be the packets forwarded in the current time interval by v_{1-i}. However, they are from different sources: $\overline{X}_{1-i}$ is provided by v_{1-i} (and thus should be consistent with the claimed forwarding probability of v_{1-i}),

1. In such scenarios, a node v_i still knows $\hat{a}_{1-i,t}$, but does not know $\hat{a}_{i,t}$. Note that v_i always knows its own action $a_{i,t}$, but it may have no idea about p_c and thus may not know $\hat{a}_{i,t}$.

2. Note that this strong punishment is to make sure that nodes do not cheat in reporting $\overline{a}_{1-i,t}$. As shown later, when the system converges to the stable state (SPNE) under FITS-I, there is actually *no real intereference* in the system.

while $X_{1-i,i}$ is overheard by v_i (and thus includes collisions it hears in addition to forwarded packets). If v_{1-i} is honest, these two transcripts should be consistent (except for collisions in $X_{1-i,i}$). Even though v_{1-i} can provide a false $\overline{X}_{1-i}$ by including the packets that it has not forwarded, v_i can detect the false claim by comparing $\overline{X}_{1-i}$ and its own record $X_{1-i,i}$. We discuss how to design this algorithm in Section 5.3.

```
▷ (v_i, v_{1−i}) is a pair of neighbors.
▷ t is the index of stage in the reputation game.
▷ r_v = TRUE at the beginning of stage 1.

if r_v = TRUE
    then
      if t = 1,
          then a_{i,t} ← 1;
      else if ∃t' ∈ {1, ..., t − 1} s.t.
          a_{i,t'} = ITF   or â_{1−i,t'} = ITF,
          then a_{i,t} ← ITF;
      else if t < T,
          then a_{i,t} ← min_{1≤t'≤t−1}{a_{i,t'}, ā_{1−i,t'}};
      else if ā_{1−i,t−1} < min_{1≤t'≤t−2}{a_{i,t'}, ā_{1−i,t'}}
          then a_{i,t} ← ITF;
      else
          a_{i,t} ← 0;
else
    a_{i,t} ← ITF.
```

Fig. 2. FITS-I scheme: deciding $a_{i,t}$ at beginning of stage t.

```
▷ (v_i, v_{1−i}) is a pair of neighbors.
▷ X̄_{1−i} is the transcript of packets sent by v_{1−i}
      that is provided by v_{1−i}.
▷ X_{1−i,i} is the transcript of packets sent by v_{1−i}
      that is overheard by v_i.

if r_v = TRUE
      With Probability p_v
          Request the transcript X̄_{1−i} from v_{1−i};
          Verify that X̄_{1−i} is consistent with ā_{1−i,t}.
          r_v ← VerProb(X_{1−i,i}, X̄_{1−i}).
```

Fig. 3. FITS-I scheme: end of a time interval in stage t.

One may notice that the FITS-I scheme also uses the perceived forwarding probability $\hat{a}_{1-i,t'}$. However, this perceived probability of node v_{1-i} is always known to node v_i, regardless of whether the PPA is valid or not. So, the PPA-independence of the scheme is not affected.

5.2 Analysis of FITS-I Scheme

The incentive compatibility of FITS-I scheme is formally stated in the following theorem:

Theorem 3: In the extended model of reputation game, if the FITS-I scheme is used, and if $\forall i \in \{0,1\}$, $\forall t \in \{1,\ldots,T\}$, $\overline{a}_{i,t} = a_{i,t}$, assuming $u > 2c$, then there is a SPNE s^* such that

$$O(s^*) = (1,1)^{T-1}(0,0) = ((1,1),\ldots,(1,1),(0,0)),$$

which implies that, for all $i \in \{0,1\}$, $u_i(s^*) = (T-1)(u-c)$.

The proof is analogous to that of Theorem 2. Due to limitation of space, we omit the proof here.

One may ask whether we can have a similar formal analysis in the case when $\overline{a}_{i,t} = a_{i,t}$ does not always hold. To achieve this goal, we need to redefine the finite repeated game by adding the reports of $\overline{a}_{i,t}$ into the action space. The result would be a much more complex game. We conjecture that, in such a complex game, we will only be able to prove a much weaker variant of SPNE, rather than the standard SPNE we consider in this paper. Hence, it would be very difficult to formally analyze the case when $\overline{a}_{i,t} = a_{i,t}$ does not always hold. In this paper, to guarantee $\overline{a}_{i,t} = a_{i,t}$, we take a lightweight approach for preventing players from making false claims of forwarding probabilities. When this approach is used, with a certain probability making false claims will be detected and punished (as shown in Theorem 4). Consequently, a selfish node has incentives to report $\overline{a}_{i,t}$ that is equal to $a_{i,t}$.

5.3 VerProb: Example Algorithm for Verifying Forwarding Probabilities

Now we design the algorithm VerProb that compares $X_{1-i,i}$ with $\overline{X}_{1-i}$.

Recall $\overline{X}_{1-i}$ is the transcript provided by v_{1-i}. Therefore, it is a sequence of packets. In contrast, because $X_{1-i,i}$ is overheard by v_i, it is a sequence of packets *and collisions*. (Note that, when the packets forwarded by v_{1-i} collide with other transmissions, v_i can hear the collisions, although it cannot determine what are the collided packets.) If v_i is honest, $\overline{X}_{1-i}$ should be identical to $X_{1-i,i}$, except that some segments of $\overline{X}_{1-i}$ correspond to collisions in $X_{1-i,i}$. The restriction on these segments is that each collision of time length τ can only match with a sequence of $\frac{\tau}{\tau_{\mathsf{col}}^{\mathsf{max}}}$ to $\frac{\tau}{\tau_{\mathsf{col}}^{\mathsf{min}}}$ packets, where $\tau_{\mathsf{col}}^{\mathsf{max}}$ (resp., $\tau_{\mathsf{col}}^{\mathsf{min}}$) is the maximum (resp., minimum) length of transmission time for a packet.

The above problem can be easily reduced to the problem of *variable-length gap matching*. Hence, the main idea of the VerProb algorithm is to reduce it to variable-length gap matching and then apply the Rahman-Iliopoulos-Lee-Mohamed-Smyth (RILMS) algorithm [22]. Details of the VerProb algorithm are given in Fig. 4.

Theorem 4: Let N be the total number of packets that need to be forwarded by v_{1-i} in stage t. If the algorithm VerProb is used, and if $\overline{X}_{1-i}$ includes $N\overline{a}_{i,t}$ packets forwarded for v_i while only $Na_{i,t}$ packets are actually forwarded for v_i (where $\overline{a}_{i,t} > a_{i,t}$), then with probability

$$p_d \geq 1 - (1 - p_v(1-p_c))^{\lceil \frac{N(\overline{a}_{i,t} - a_{i,t})}{n_{\mathsf{int}}} \rceil},$$

the algorithm VerProb outputs FALSE, where n_{int} is an upper bound for the number of packets transmitted in a time interval.

```
▷ (v_i, v_{1−i}) is a pair of neighbors.
▷ X̄_{1−i} is the transcript of v_{1−i}'s messages
      provided by v_{1−i}.
▷ X_{1−i,i} is the transcript of v_{1−i}'s messages
      overheard by v_i.
▷ COL is the symbol for an overheard collision.
▷ VLGSearch is the RILMS algorithm.

j ← 1; l ← 1; l' ← 1;
J ← NumberOfPackets(X_{1−i,i});
while j ≤ J do
    X'[l] ← longest prefix of X_{1−i,i}[j ··· J] that
              does not contain COL;
    if X'[l] is not empty
        then l ← l + 1;
             j ← j + NumberOfPackets(X'[l]);
    if j ≤ J
        then X''[l'] ← longest prefix of
             X_{1−i,i}[j ··· J] that contains only COL;
             l' ← l' + 1;
             j ← j + NumberofCOL(X''[l']);
    else
        return VLGSearch(X̄_{1−i}, X'[1 ··· l − 1],
                         {COLPack(X''[j])}_{j=1}^{l'−1});
X'[l] ← empty string;
return VLGSearch(X̄_{1−i}, X'[1 ··· l],
                 {COLPack(X''[j])}_{j=1}^{l'−1});

COLPack(x)
    L_1 ← 0; L_2 ← 0;
    for each COL in x
        τ ← Time of COL;
        L_1 ← L_1 + ⌈ τ / τ_col^max ⌉;
        L_2 ← L_2 + ⌊ τ / τ_col^min ⌋;
    return(L_1, L_2);
```

Fig. 4. VerProb: verification of forwarding probability.

Proof: Let n_j be the number of packets that v_{1-i} falsely claims to be forwarded for v_i in time interval j. Clearly,

$$\sum_{j=1}^{t} n_j = N(\overline{a}_{i,t} - a_{i,t}). \tag{4}$$

Node v_i detects cheating in this time interval if the algorithm VerProb is executed in this time interval, and at least one of these n_j packets is claimed to be sent at a time when v_i does not hear a collision. So the probability of detecting cheating in time interval j is

$$p_{d,j} = p_v(1 - p_c^{n_j}).$$

Recall m is the number of time intervals in stage t. Hence, the probability of detecting cheating in the entire stage is

$$\begin{aligned} p_d &= 1 - \prod_{j=1}^{m}(1 - p_{d,j}) \\ &= 1 - \prod_{j=1}^{m}(1 - p_v(1 - p_c^{n_j})) \\ &\geq 1 - (1 - p_v(1 - p_c))^{\lceil \frac{N(\overline{a}_{i,t} - a_{i,t})}{n_{\text{int}}} \rceil}. \end{aligned}$$

The last inequality is due to (4) and the pigeonhole principle. □

6 EVALUATIONS

In the previous sections, we have presented the two schemes of FITS. To evaluate the performance of FITS, we implement these two schemes in the network layer using wireless network simulator GloMoSim [11] and perform three sets of experiments:

- The first set of our experiments study the punishment of FITS schemes on misbehaving nodes. The objective is to verify that FITS schemes effectively punish misbehaving nodes and thus give nodes incentives to cooperate in packet forwarding.
- The second set of our experiments compare FITS with an existing reputation system, in terms of their convergence to stable states. The objective is to verify that FITS converges to a stable state with significantly higher forwarding probabilities.
- The third set of experiments examine the effect of interference, in case it is really used for punishment.
- The fourth set of our experiments measure the efficiency of FITS schemes in terms of computation and communication overheads.

In the remainder of this section, we describe the settings and results of our experiments.

6.1 Settings

In our experiments, the MAC layer is based on IEEE 802.11; the Dynamic Source Routing protocol (DSR) is used for routing. We use the two-ray propagation path-loss model. The radio transmission power level is at 12 dBm and the radio to noise ratio threshold is set to 8.0 dB. The network has a bandwidth of 2 Mbps.

Within an area of 2000 by 2000 meters, 50 nodes are randomly distributed. The topology of the network is shown in Fig. 5.[3] We generate the traffic of 50 sessions. Every node is a session source and the destination of each session is randomly picked. Throughout the simulation time each source transmits packets at a constant bit rate of 2 packets/s with the packet size being 512 bytes. For FITS-I scheme, we set $m = 20$ (which means there are 20 time intervals in each stage) and $p_v = 0.2$.

3. Here we assume that during each of our data sessions, network topology remains the same.

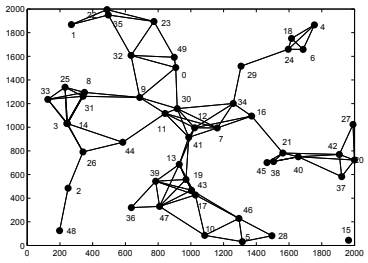

Fig. 5. Topology of the randomly generated network. Nodes are labeled with their IDs. A link between two nodes represents that these two nodes are within each other's transmission range.

6.2 Punishment on Misbehaving Nodes

In the first set of experiments, we study the punishment that misbehaving nodes receive in FITS schemes.

We randomly pick 5 nodes to be the misbehaving nodes, which means these nodes do not implement FITS and drop packets that are not destined to them with a fixed probability. All the other nodes are cooperative, i.e., they use FITS schemes to forward packets for other nodes. There are 8 stages in the reputation game and each stage lasts for 100 seconds. The entire simulation time is 800 seconds. The results described below are the average of 200 runs.

Fig. 6 shows the utilities of misbehaving nodes when they drop packets in the FITS-D scheme (resp., FITS-I scheme) in subfigure (1) (resp., (2)). Here, we use $u = 3.0$ and $c = 1.0$ when calculating the utility. From Fig. 6 we observe that, when the packet dropping probability of misbehaving nodes grows, their utilities obtained in the reputation games decrease quickly. This reflects that the FITS schemes effectively punishes the misbehaving nodes in terms of utilities.

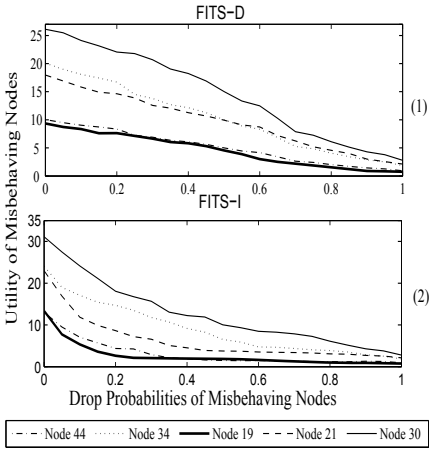

Fig. 6. Utilities of misbehaving nodes in FITS schemes.

Another way to study the punishment on misbehaving nodes is to measure the message success rates (the percentage of packets from the source node that successfully arrive at the destinations) of the misbehaving nodes. Fig. 7 shows the results of our measurements in the two FITS schemes. For comparison, we also include the average message rates of cooperative nodes in these figures. We can see that the message success rates of all misbehaving nodes are fast decreasing (as the packet dropping probability grows), while the message success rates of cooperative nodes only decrease slightly.

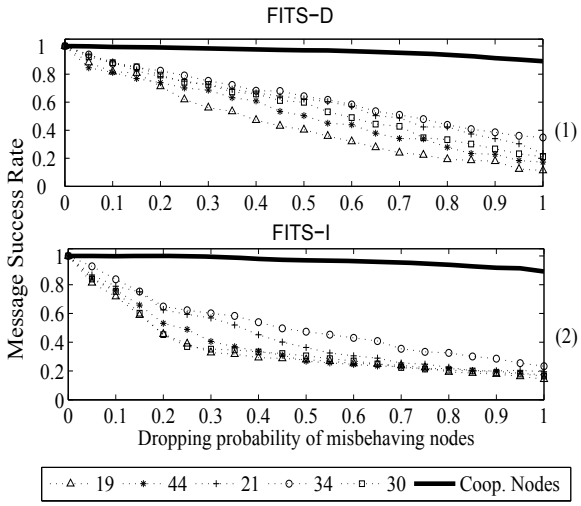

Fig. 7. Message success rate of misbehaving nodes and cooperative nodes in FITS schemes.

6.3 Comparison of Stable States

Our second set of experiments is to compare FITS with an existing reputation system, DARWIN [14], and to observe their convergence to stable states. In this set of experiments, nodes are allowed to choose any stage of the game to start dropping packets. A stable state is a state such that for each node, changing its strategy in any way cannot increase its utility. To find the stable state, in this set of experiments we let nodes go through a sequence of reputation games and allow the nodes to change their strategies between games. A node stops changing its strategy when it finds that there is no way to improve its utility by changing the current strategy. When the involved nodes stop changing their strategies, a reputation system is in its stable state.

We randomly pick a pair of neighbors, Node 24 and Node 29, and observe how their average forwarding probabilities for each other evolve in a sequence of reputation games. These two nodes are selfish so try to maximize their utilities by dropping packets. At the beginning of each experiment, each node forwards packets with probability 1. In the sequence of reputation games, a node can make an attempt to change its strategy as follows: it randomly picks a stage and drops all the packets in and after that stage. Then it compares the utility it gets in this game with the utility in the previous game. If there is a loss in utility, it goes back to the old forwarding strategy, and stays with that strategy for 3 games before it makes another

attempt. If there is a gain in utility, it is happy for that and thus stays with the new forwarding strategy for 20 more games before it makes the next attempt to drop more packets.

We test FITS and DARWIN, respectively, in the above experiments. Each stage is 30 seconds long and each game has 5 stages. Each experiment lasts for 500 minutes, so that both reputation systems are guaranteed to converge to stable states.

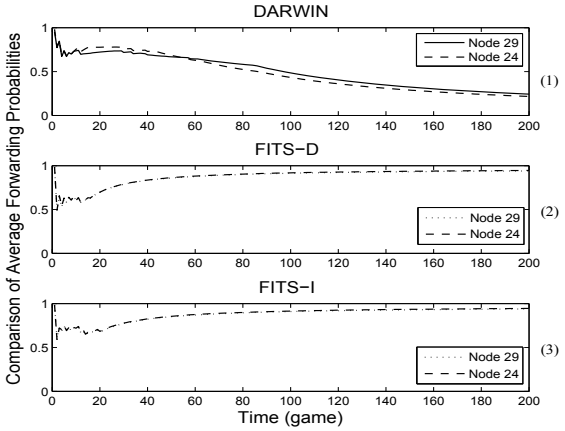

Fig. 8. Comparison of average forwarding probabilities of DARWIN and FITS as system converges.

Fig. 8 shows the results for DARWIN, FITS-D and FITS-I in subfigures (1), (2) and (3) respectively. The forwarding probability of each node is the average forwarding probability in the sequence of games that the node has been through from the beginning. In subfigure (1) we can see that the average forwarding probabilities of both Node 29 and Node 24 decrease as the system evolves. It implies that when the system evolves, the nodes can find better strategies of dropping packets rather than forwarding them with probability 1.

In contrast, in the two FITS schemes, as shown in subfigures (2) and (3), average forwarding probabilities increase to a constant close to 1, after the oscillations in the first few games. In fact, at the beginning the nodes make all possible attempts, and find no way to increase their utilities by dropping packets (because they are punished by FITS). Consequently, in the rest of the experiment they stay with the strategy to forward all packets.

6.4 Effect of Interference

This set of experiments are to examine the effect of interference on cooperative nodes, when interference is used to punish misbehaving nodes. We first measure the number of interference actions taken by cooperative nodes, for different number of misbehaving nodes in the network. Then we measure the average message success rates of the cooperative nodes in their own sessions in the following two settings: In the first setting, our FITS schemes are used; in the second setting, a modification of FITS is used in which there is definitely no interference. We compare the cooperative nodes' average message success rates in these two settings to show the effect of interference on message success rates. In this set of experiments the misbehaving nodes drop packets with probability 0.5.

In Fig. 9, we show the number of interference actions taken by cooperative nodes in the network, when there are 5 to 15 misbehaving nodes. From our schemes, it is easy to see that the number of interference actions taken by the cooperative nodes in FITS-D scheme is the same as in FITS-I scheme. So we only use one figure to illustrate the results. We can see that the number of interferences actions is below 20 when there are no more than 15 misbehaving nodes in the network.

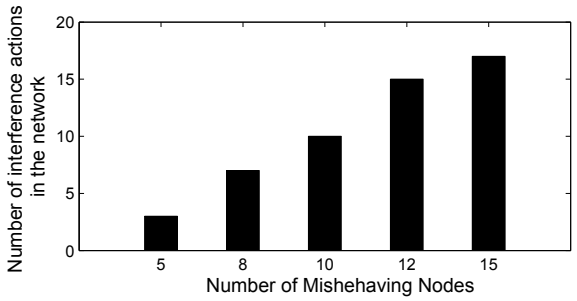

Fig. 9. Number of Interference Actions introduced by FITS vs. Number of Misbehaving Nodes.

Fig. 10 gives a comparison of cooperative nodes' average message success rates in the two settings, i.e., with and without interference introduced by FITS. We vary the number of misbehaving nodes in the network from 5 to 20. From the figure, it is clear that, when there are more misbehaving nodes, the message success rate becomes lower. However, the loss in average message rate caused by interference introduced by FITS is very small.

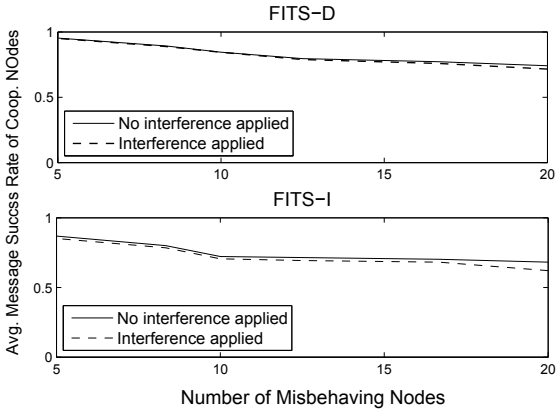

Fig. 10. Message Success Rates of Cooperative Nodes in Their Own Sessions: FITS vs. No Interference.

6.5 Efficiency

In the third set of experiments, we measure the efficiency of FITS.

Fig. 11 shows the computation overheads of the two FITS schemes. Here by computation overhead we mean the average extra computation time needed by FITS for each node in each session, when the system is in the stable state. In this set of experiments, we have 1600 packets in each session. Fig. 11 shows that the computation overhead of FITS-D scheme always remains below 3 milliseconds. The computation overhead of FITS-I scheme is slightly higher, but it is still less than 3.5 milliseconds.

Clearly, in addition to computation overheads, FITS also has communication overheads—the time to send and receive control packets for the FITS schemes. However, for the FITS-D scheme, the communication overheads are only several microseconds per session. For the FITS-I scheme, the communication overheads are several hundred microseconds per session. Consequently, compared with computation overheads, communication overheads can be ignored.

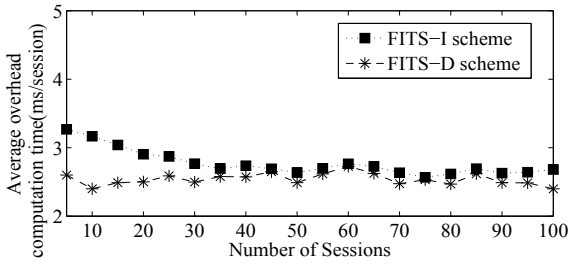

Fig. 11. Computation overheads of FITS schemes.

7 Related Work

There has been extensive study of the incentive problems in routing and packet forwarding for wireless ad hoc networks. Generally, the solutions proposed follow one (or both) of the two approaches: the approach of reputation systems [2], [4], [5], [12], [15]–[18], [21] and the approach of credit-based systems [1], [6]–[8], [13], [23], [24], [26]–[32]. Since we focus on reputation systems in this paper, we only focus on related works using this approach.

In a reputation system, the behavior of a node is observed by other nodes in the network; it gets a bad reputation if it appears to have dropped packets. Reputation systems can be divided into two categories: end-to-end reputation systems and the more recently proposed hop-by-hop reputation systems.

The first solution to the incentive-compatible packet forwarding problem is an end-to-end reputation system proposed by Marti, et al. [16]. Another early solution proposed by Buchegger and Le Boudec, is CONFIDANT [4], [5]. In these two systems each node monitors nodes in the network and calculates their reputations.

In recent years, hop-by-hop reputation systems have been proposed to reduce the overheads of reputation systems. Good example are SORI, proposed by He, et al. [12] and Catch, proposed by Mahajan, et al. [15].

The above reputation systems, along with many other reputation systems (e.g., [2], [17], [18], [21]), have a common limitation: they do not have rigorous analysis of their incentive compatibility. Hence, it is not clear what guarantee they can provide in terms of incentive compatibility.

As far as we know, Milan, et al.'s GTFT [19](which was first studied in a different content by Wu and Axelrod [25]) and Jaramillo and Srikant's DARWIN [14] are two reputation systems in wireless networks that have rigorous analysis. Both of them are shown to provide SPNE solutions. The major difference between FITS and these two reputation systems is that FITS is designed for a reputation game that lasts for a finite amount of time, while in [19] and [14] the reputation games are modeled to last infinite amount of time. In reality, a reputation system most likely runs for a finite amount of time only.

8 Conclusion and Future Work

Reputation system is an important approach to solve the incentive compatibility problem in packet forwarding of wireless ad hoc networks. In this paper, we present the *first* formal study of reputation system in the model of *finite* repeated game. We believe this is a more realistic model for reputation system and thus our results have significant practical implications.

The first result we obtain is the impossibility of building a SPNE solution using traditional reputation systems. This result implies that we must introduce a new technique if we want to establish SPNE solutions for our problem.

Then we introduce the TTI technique and use it to build FITS, our new reputation system. FITS provide strong incentive compatibility for nodes to cooperate in packet forwarding. More precisely, there is a SPNE in which nodes forwards all packets. This is proved theoretically and verified by experiments.

We note that our work has only addressed one aspect of reputation systems, and left out many other aspects. For example, we have assumed a network of fixed topology. In contrast, there are also networks with dynamically changing topologies in reality. In these networks, building a reputation system is much more challegining. It would be non-trivial to establish finite-time reputation systems with provable incentive compatibility in these networks. Hence, this is an interesting topic for future study.

We also note that, besides threat of interference, there are other techniques that can possibly be used in building finite-time reputation systems. One such technique is that a node shuts down its connection with a neighbor whenever it detects that the neighbor is dropping its packets. Again, it is non-trivial to use such techniques to establish a complete reputation system and prove the incentive compatibility. Hence, we also leave this to future explorations.

References

[1] L. Anderegg and S. Eidenbenz. Ad hoc-VCG: A truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In *Proc. MOBICOM'03*, San Diego, CA, Sep. 2003.

[2] S. Bansal and M. Baker. Observation-based cooperation enforcement in ad hoc networks. Technical report, Stanford University, Stanford, CA, Jul. 2003.

[3] N. Ben Salem, L. Buttyan, J. P. Hubaux, and M. Jakobsson. A Charging and Rewarding Scheme for Packet Forwarding. In *Proc. MOBIHOC'02*, Annapolis, MD, Jun. 2003.

[4] S. Buchegger and J.-Y. Le Boudec. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In *Proc. (EUROMICRO-PDP'02)*, Canary Islands, Spain, Jan. 2002.

[5] S. Buchegger and J.-Y. Le Boudec. Performance analysis of the CONFIDANT protocol (Cooperation of nodes: fairness in dynamic ad-hoc networks). In *Proc. MOBIHOC'02*, Lausanne, Switzerland, Jun. 2002.

[6] L. Buttyan and J.-P. Hubaux. Enforcing serviceavailability in mobile ad-hoc WANs. In *Proc. MOBIHOC'00*, Boston, MA, Aug. 2000.

[7] L. Buttyan and J.-P. Hubaux. Stimulating cooperationin self-organizing mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications*, 8(5):579-592, Oct. 2003.

[8] J. Crowcroft, R. Gibbens, F. Kelly, and S.Ostring. Modelling incentives for collaboration in mobile ad hoc networks. In *Proc. WiOpt'03*, France, Mar. 2003.

[9] S. Eidenbenz, V. S. A. Kumar, and S. Zust. Equilibria in topology control games for ad hoc networks. In *Proc. the 2003 Joint Workshop on Foundations of Mobile Computing*, pages 2-11, 2003.

[10] S. Eidenbenz, G. Resta, and P. Santi. Commit: A sender-centric truthful and energy-efficient routing protocol for ad hoc networks with selfish nodes. In *Proc. IPDPS'05*, Denver, Colorado, Apr. 2005.

[11] UCLA GloMoSim project team, "'http://pcl.cs.ucla.edu/projects/glomosim/.'"

[12] Q. He, D. Wu, and P. Khosla. SORI: A secure and objective reputation-based incentive scheme for ad-hoc networks. In *Proc. WCNC'04*, 2:825-830, Atlanta, GA, Mar. 2004.

[13] M. Jakobsson, J. P. Hubaux, and L. Buttyan. A micropayment scheme encouraging collaboration in multi-hop cellular networks. In *Proc. Financial Crypto 2003*, 2742:15-33, 2003.

[14] J.J. Jaramillo and R. Srikant. DARWIN: Distributed and Adaptive Reputation mechanism for WIreless ad-hoc Networks. In *Proc. MOBICOM'07*, Montreal, Quebec, Canada. Sep. 2007.

[15] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Sustaining cooperation in multi-hop wireless networks. In *Proc. NSDI'05*, Boston, MA, May 2005.

[16] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proc. MOBICOM'00*, Boston, MA, Aug. 2000.

[17] P. Michiardi and R. Molva. CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proc. CMS'02* Portoroz, Slovenia, Sep. 2002.

[18] P. Michiardi and R. Molva. Analysis of coalition formation and cooperation strategies in mobile ad hoc networks. *Ad Hoc Networks*, 3(2):193-219, Mar. 2005.

[19] F. Milan, J. J. Jaramillo, and R. Srikant. Achieving cooperation in multihop wireless networks of selfish nodes. In *Proc. GameNets'06*, Pisa, Italy, Oct. 2006.

[20] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, The MIT Press, 1994.

[21] M. T. Refaei, V. Srivastava, L. DaSilva, and M. Eltoweissy. A reputation-based mechanism for isolating selfish nodes in ad hoc networks. In *Proc. MobiQuitous'05*, San Diego, CA, Jul. 2005.

[22] M. S. Rahman, C. S. Iliopoulos, I. Lee, M. Mohamed, and W. F. Smyth, Finding Patterns with Variable Length Gaps or Don't Cares, in *Proc. COCOON'06*, 4112:146-155, Taipei, Taiwan, Aug. 2006.

[23] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao. Energy efficiency of ad hoc wireless networks with selfish users. In *Proc. European Wireless Conference*, Florence, Italy, Feb. 2002.

[24] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao. Cooperation in wireless ad hoc networks. In *Proc. INFOCOM'03*, San Francisco, CA, Mar./Apr. 2003.

[25] J. Wu and R. Axelrod. How to cope with noise in the iterated prisoner's dilemma. *The Journal of Conflict Resolution*, 39(1):183-189, Mar. 1995.

[26] W. Wang, S. Eidenbez, Y. Wang, and X.-Y. Li. OURS- Optimal Unicast Routing Systems in Non-Cooperative Wireless Networks. In *Proc. MOBICOM*, Los Angeles, CA, Sep. 2006.

[27] W. Wang and X.-Y. Li. Low-cost routing in selfish and rational wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(5):596-607, 2006.

[28] W. Wang, X.-Y. Li, and Y. Wang. Truthful Multicast in Selfish Wireless Networks. In *Proc. MOBICOM'04*, Philadelphia, PA, Sep. 2004.

[29] Y. Zhang, W. Lou, W. Liu, and Y. Fang, A secure incentive protocol for mobile ad hoc networks *ACM Wireless Networks*, vol. 13, no. 5, pp. 569-582, October 2007.

[30] S. Zhong, J. Chen, and Y. R. Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *Proc. INFOCOM'03*, San Francisco, CA, Mar./Apr. 2003.

[31] S. Zhong, L. E. Li, Y. G. Liu, and Y. R. Yang. On Designing Incentive-Compatible Routing and Forwarding Protocols in Wireless Ad-Hoc Networks–An Integrated Approach Using Game Theoretical and Cryptographic Techniques, In *Proc. MOBICOM'05*, Cologne, Germany, Aug./Sep. 2005.

[32] S. Zhong and F. Wu. On Designing Collusion-Resistant Routing Schemes for Non-Cooperative Wireless Ad Hoc Networks. In *Proc. MOBICOM'07*, Montreal, Quebec, Canada, Sep. 2007.

Tingting Chen received her B.S. and M.S. degrees in computer science from the department of computer science and technology, Harbin Institute of Technology, China, in 2004 and 2006 respectively. She is currently a Ph.D. candidate at the department of computer science and engineering, the State University of New York at Buffalo, U. S. A. Her research interests include data privacy and economic incentives in wireless networks.

Fan Wu is a post doctoral research associate at Department of Electrical and Computer Engineering of University of Illinois at Urbana-Champaign. He received BS degree in Computer Science from Nanjing University, in 2004, and PhD degree in Computer Science and Engineering from University at Buffalo, the State University of New York, in 2009. His research interests include algorithmic game theory, economic incentives for wireless networks and peer-to-peer computing.

Sheng Zhong is an assistant professor at the computer science and engineering department of the State University of New York at Buffalo. He received his BS (1996), ME (1999) from Nanjing University, and PhD (2004) from Yale University, all in computer science. His research interests include privacy and incentives in data mining and databases, economic incentives in wireless networks.