A Principal-Agent Method to Prevent Selfish MAC Layer Behavior in Wireless Networks

Kai Shi, Yantai Shu, Chunfeng Liu
Department of Computer Science
Tianjin University
Tianjin, 300072, China
{shikai0229, ytshu, cfliu}@tju.edu.cn

Oliver Yang
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, Canada, K1N 6N5
yang@site.uottawa.ca

Abstract

Security is a fundamental prerequisite for the survivability and reliability of wireless networks. In a network where limited wireless resources have to be shared, selfish nodes can manipulate relevant network parameters to gain more access to the resources, and hence obtain a higher performance than their fair share, while the performance of well-behaved nodes will be significantly degraded. This paper considers the environment of an IEEE802.11 WLAN, and proposes a solution from the prospective of a principal-agent system. Our solution uses an incentive and a constraint mechanism to encourage the selfish agent to perform normally. Our method does not modify the IEEE 802.11 protocol, but requires an additional principal node only. Simulation results show that our method can overcome the influence of selfish nodes improve the network fairness performance while maintaining the throughput performance.

Keywords: *network security, principal-agent system, incentive and constraint mechanism.*

1. Introduction

The exponential growth of wireless network in recent years has brought major research issues including the fair share of available bandwidth, QoS (Quality of Service), and the control of misbehaving traffic sources. These issues have been investigated in the context of the Internet (TCP/IP), to certain extent, by using traffic shaping, traffic conditioning, and admission control [1]. However, these issues have not been fully addressed in wireless networks (including cellular, Ad Hoc, and sensor networks) that are based on a shared medium and often contention-oriented protocol.

Many wireless networks nowadays are based on IEEE 802.11x standards that provide public wireless access to the Internet. The MAC (Medium Access Control) in IEEE 802.11 uses DCF (Distributed Contention Function) to share the limited bandwidth of the wireless channel. If the MAC protocol is manipulated or misused, then the consequences can be overwhelming and disrupt the operation of the whole network. For example, a selfish node can manipulate the MAC protocol to gain access to the channel, and resulting in some cases of starvation of other nodes in the same network. The manipulation of the MAC layer protocol is hidden from the upper layers, and can be further enhanced if combined with more violations from these upper layers.

DCF is a distributed contention-based protocol. If a station has a packet to transmit, it checks if the medium is idle for a DIFS (Distributed Inter-Frame Space) time. If idle, it sends an RTS (Request to Send) message to the destination. The destination station acknowledges the message by sending a CTS (Clear to Send) back to the sender. If the destination receives the packet without errors, it sends an ACK (Acknowledgment) back to the sender. Otherwise, the sender assumes that the packet did not arrive and it retransmits it again. If the medium is busy, the station defers until the medium is idle for a DIFS time and the *Binary Exponential Backoff* algorithm is invoked (to be explained in the next section).

A station with data to send selects a backoff value *BO* based on the current contention window (*CW*) as follows: $BO = \text{int}(CW \times rand \times slot\ time)$, where *rand* is a random number uniformly distributed between 0 and 1; *slot-time* is 20μs; and $CWmin < CW < CWmax$, where *CWmin* is the minimum *CW*, which is usually set to 31 and *CWmax* is the maximum *CW* and often set to 1023. At the first transmission $CW = CWmin - 1$, which means that the station chooses a *CW* from the range $[0, \ldots, CWmin - 1]$. After a DIFS idle time, the station senses the medium. If the medium is idle, then the station decrements its backoff value by a *slot time*; otherwise the backoff value stays the same. When the backoff value of a station reaches 0, the station transmits its packet to the destination station. If the sender receives an ACK from the destination, the transmission is assumed to be successful, and the station sets its CW back to $CWmin - 1$. If a collision

978-1-4244-5113-5/09/$25.00 2009 IEEE

happens, both stations back off and increase their CW exponentially to $CW = 2i-1$, where i is the number of collisions. The maximum number of retransmission permitted by a station is 7 times. There are many ways a node can violate the MAC layer protocol. Below is a list of some possibilities:

1. A node can adjust its backoff time to *CWmin* or *less* at all times.
2. A node may *scramble/Interfere* frames sent by other nodes in order to increase their CW [4].
3. A receiver node may *delay* sending CTSs and ACKs, or *reject* RTS and DATA, so the sender would double its *CW* and consequently the selfish node gets to transmit.
4. A node may *increase* its NAV (Network Allocation Vector) time to prevent other nodes from contending during this period.
5. A node may also transmit when it senses the channel idle before waiting DIFS time.

There are two types of misbehaving nodes: *selfish nodes*, and *malicious nodes*. A selfish node is only concerned about improving its performance even at the expenses of other nodes. Malicious nodes like to disrupt normal network operations, like DoS (Denial of Service) attacks, or jamming the wireless channel to prevent communication. Dealing with malicious nodes has been investigated mostly under wireless security e.g. [7], and will not be discussed here.

Unlike the existing works reviewed above which either rely on detection or punishment to resolve the selfish behaviour in the MAC layer, we propose the principal-agent method which does not rely on these two mechanisms to overcome the selfish problem. We have extended the existing 802.11 algorithm to accommodate our mechanism and we would like to evaluate its performance by simulation.

As a contribution to overcome the selfish problem,, our principal-agent method uses two mechanisms to make the selfish node to obey the IEEE 802.11 MAC layer protocols. The first one is an incentive mechanism to encourage the agents to find a small contention window to improve the network performance. The second one is a constraint mechanism to prevent the selfish nodes from degrading the performance of other nodes. Our simulation results validate the method we propose.

2. Related work

Several techniques have been proposed to detect misbehavior on the network layer in wireless networks [2], but little has been done on the MAC layer. MAC layer detection solutions focused on detecting backoff values manipulation because the backoff values are the easiest to manipulate and the hardest to detect.

2.1. Detecting and Handling of MAC Layer Misbehavior in Wireless Networks [3]

The authors proposed a modification to the IEEE 802.11MAC protocol that simplifies the misbehavior detection. They assume that the *receiver* is a trusted node like a BS (Base Station) that selects a backoff value and sends it in the CTS and ACK packets to the sender. The sender is expected to use this backoff value in its next transmission to the receiver. These changes allow the receiver to monitor the sender if it deviates from the protocol by observing the number of idle slots between consecutive transmissions from the sender. If the number of idle slots is less than the assigned backoff, it *may* be an indication of a *deviation* from the protocol. To reduce the throughput of a deviating sender, the receiver would penalize it by assigning a *larger* backoff value to its next transmission (this is called the correction scheme). If the number of its deviations over a number of transmissions exceeds a predefined *threshold*, the sender is judged to be *misbehaving* with a high probability. Simulation results have shown that the scheme can offer an accurate diagnosis of node-misbehavior if the misbehavior *persists*. Note that the correction scheme is to limit the throughput of misbehaving nodes to its fair share of bandwidth. Although the scheme works well on an infrastructure networks, it is not suitable for distributed networks, where receivers can not be trusted. Another potential problem is the *colluding nodes* problem, where the receiver can assign smaller backoff values to a sender in return of favors from the latter (like having some packets to send to it). In addition, IEEE 802.11 has to be changed in order for this scheme to be applied.

2.2 DOMINO [4]

The authors proposed a system to detect greedy behavior in the MAC layer of IEEE 802.11 public networks. DOMINO is a piece of software installed at the Access Point (AP). It detects and identifies greedy stations without any changes of the standard protocol. DOMINO [4] does not only focus on the manipulation of the backoff values, but it has been extended to cover mostly all manipulation techniques mentioned above. The AP gathers enough *statistical data* to detect if the parameters of the MAC protocol have been manipulated.

Traffic traces are collected periodically during short intervals of time called monitoring periods. The information gathered by the AP run through multiple tests to detect misbehavior. Eventually if a node is misbehaving, it has to be caught by one of these tests. Below are some of the tests used in DOMINO [4]:

a) *Scrambled Frames*: In order to gain a significant share of the wireless bandwidth using the usual

RTS/CTS/Data scrambling, the misbehaving node has to scramble a large percentage of CTS, ACK, or DATA frames sent by other stations. Hence, its average number of retransmissions should be less than that of the others.

b) *Shorter than DIFS*: The AP monitors the idle time period after the last ACK received, therefore it can detect if a station waited the DIFS time before retransmitting again or not.

c) *Oversized NAV*: The AP monitors the actual time it takes a station to transmit. Then, it compares it to its NAV to check if the station set the NAV appropriately or not.

d) *Backoff manipulation*: The authors proposed three algorithms, to help with the detection of the backoff values manipulation.

The advantage of this scheme is its simplicity, and its high accuracy to detect a variety of cases. The system is also resilient to several factors, such as traffic types that could affect the performance of other detection systems. Although and as the authors pointed, DOMINO has some open issues. One issue is the security, where a selfish node may impersonate an honest node in order to provoke the punishment function, and gets it possibly disconnected. Adaptive misbehaving is also an issue in DOMINO, where some nodes may exploit the detection system, and avoid being detected by switching enough between several techniques.

2.3. Detection and Prevention of MAC Layer Misbehavior in Ad Hoc Networks [5]

The authors proposed a detection algorithm assuming that at *least one* of the stations involved in the transmission is honest. They also proposed a statistical detection scheme to limit the throughput of colluding nodes. The idea is that both the sender and the receiver agree on a random backoff value through a public discussion. An honest party will always make sure that the backoff value is truly random. To detect a *dishonest sender*, they proposed the use of the detection algorithm of [3], with minor modifications. To detect a *misbehaving receiver* the algorithm works as follows:

1. A receiver R send a sender S a random backoff value r in $[0, \ldots, CWmin - 1]$ and commits to it by binding to it, and hiding it from the sender. r is a binary number of length m, where $m = \log_2^{CWmin}$.

2. After receiving the commitment, S selects a random value r′ in $[0, \ldots, CWmin - 1]$, where r′ is also a binary number of length m.

3. Finally R opens its commitment to S, which enables it to verify if the value sent by the receiver R was correct or not. If it is correct, both S and R compute the backoff value by *applying the* $\oplus$ *operation on* r *and* r′. Otherwise, the receiver R is assumed misbehaving, and consequently reported to a reputation management system.

This detection algorithm does not solve the problem of colluding nodes where both the sender and the receiver pretend to select backoff value of *zero*. Therefore, to further tests the randomness of a backoff value, the authors used a well known statistical test called *Entropy Estimation*. Although, the Entropy Estimation test improves the detection of colluding nodes, it does not solve it completely. Another disadvantage of the scheme is that the assumption of an already existing reputation management system that processes the results which are not always true.

Finally, a SWN-CUSUM (Sliding Window Nonparameter Cumulative Sum) mechanism [8] is proposed for MAC layer selfish behavior detection, based on backoff time measurement between consecutive successful transmissions. The technique is a statistically robust selfish detector that can operate without modifying the protocol implementation and can be used to any random access MAC layer protocol. However, there are no proposals on how to penalize the selfish behavior.

3. Preliminaries

In this section, we shall first introduce the principal-agent theory, and then explain how to map the selfish MAC layer behavior into a principal-agent problem.

The principal-agent problem [9] originates from the study of political science and economics where it is desirable if one can motivate a party to act on behalf of another. The problem arises when a principal compensates an agent for performing certain useful but costly acts, e.g., some performance measures are costly to observe. To some extent all contracts are written in a world of information asymmetry, uncertainty and risk. Here, a principal is not certain whether (or to what extent) a contract has been satisfied. The solution of this information problem (which is closely related to the moral hazard problem [10]) is to ensure the provision of appropriate incentives so agents act in the way principals wish. In terms of game theory, it involves changing the rules of the game so that the self-interested rational choices of the agent coincide with what the principal desires. Even in the limited arena of employment contracts, the difficulty of doing this in practice is reflected in a multitude of compensation mechanisms (‘the carrot’) and supervisory schemes (‘the stick’). Another distinct and relatively new application of the principal-agent problem describes the landlord-tenant relationship as a barrier to energy savings. The problem is also discussed in terms of "agency theory".

In our problem, there exists similar information

asymmetry between the network provider (the principal) and the network users (the agents). Here, the network provider wishes the whole network performance to be optimal where the network utilization is high and all the uses are satisfied. In other words, both high throughput and fairness need to be considered. On the other hand, the network users want to maximize their own throughput. This information asymmetry arises because the principal and each agent do not know the contention window information of other agents. So the selfish nodes have a chance to manipulate the contention window to get a better performance while degrading the performance of other agents and the principal. In the next section, we shall give a detailed description of an incentive and constraint mechanism to constrain such behavior.

4. The proposed scheme

Our proposed scheme requires an extra node to function as the principal and to monitor the agents. Since a single principal may not be able to monitor the whole network, we may appoint at least one principal to each region. The selection of principal is a problem itself which will be studied in a future paper. We shall focus on the formation and implementation of an effective mechanism instead.

As mentioned before, we have to consider both the network utilization and the fairness performance in order to maximize the network performance. They are described as follows.

A. Network Utilization Function

As discussed in Section 1, each node in IEEE 802.11 DCF protocol has to wait for a time before transmitting. This means that the channel may be idle sometimes when all nodes are in their backoff stage. From the view point of a principal, the shorter the idle time the better the network utilization. So a principal would like each agent to transmit with the shortest backoff time. One candidate utilization function ($Funu(u)$) can be defined as:

$$Funu(u) = T/(T+u), \tag{1}$$

where T is the transmitting time of each successful transmission; it also depends on the transmitting rate of the wireless channel and the MAC frame length. From expression (1), one can see that each agent will need to select a minor contention window in order to improve the network utilization, and also to improve its own performance. Therefore this function is also referred to as an *incentive function/mechanism*. But the selection of small window may influence the performance of other well behaved nodes, so we have to constrain the agent at the same time using the fairness function to be described below.

B. Fairness Function

Fairness is an important factor for the network performance. Here, we choose the fairness index $Funf(x)$ from [6] which is defined as follows.

$$Funf(x) = (\sum_{i=1}^{N} x_i)^2 \ / \ n\sum_{i=1}^{N} x_i^2, \tag{2}$$

where x_i is the number of data transmission of node *i*, and N is the number of agents managed by this principal. One can see that this fairness function will be degraded if the agents do not obey the same rule. So this function is called the *effective constraint function*.

4.1 The Utility Functions

The utility functions allow the principal to perform an effective incentive and constraint mechanism. Both the principal and the agents have their own utility functions as follows:

C. Principal's utility function

The utility function of a principal consists of both the network utilization function and the fairness function described earlier. Let x_i be the total transmission time of node *i* (note that the transmission time for each transmission may be different), and u_i be the current idle time of the channel if node *i* is selected for the next transmission. Furthermore, let $Funu(\bar{u})$ and $Funf(\bar{x})$ be the current network utilization and the fairness functions, where $\bar{u}$ is the average idle time between two transmissions, and $\bar{x}$ is the average transmission time. Then the updated utility function of a principal can be obtained as

$$\begin{aligned} Func(u_i, x_i) = {} & \alpha(Funu(u_i) - Funu(\bar{u})) \\ & + (1-\alpha)(Funf(x_i) - Funf(\bar{x})), \end{aligned} \tag{3}$$

where the parameter α reflects the performance preference of the principal. If the value of α is large, it means the principal prefers network utilization more. In this paper, we let α =0.5, which gives both the utilization and fairness the same importance.

D. Agent's Utility Function

For each agent, the only benefit is to improve their throughput performance. So they will try to decrease their waiting time (w_i), One candidate utility function of the agent is $Fun(w_i) = 1/w_i$.

To improve the value of its utility function, the agent may seek to decrease the backoff time by choosing a shorter contention window. However, as this misbehavior may be constraint by the principle, its performance may be worse. If the agent is rational, it will find out that it can achieve the optimal equilibrium point by obeying the IEEE 802.11 mechanism.

4.2 The Algorithm

Our algorithm consists of the behaviors at the principal node and the agents. Some assumptions are:

i) The principal is preset in each region/domain in order for us to focus on the study of the incentive and constraint mechanisms only.
ii) All nodes are rational who want to improve their own performance, e.g., the selfish node.

4.2.1 The Principal Behavior

When an agent requests to access to the channel through a RTS, the principal will compute the utility function. If this value is smaller than zero, the principal will let the receiver not to send CTS to the sender to reject the sender to access to the channel. Otherwise, the requested node will be admitted to access to the channel. From this utility function, we can see that the mechanism can encourage the agents obey the rule to select an optimal contention window.

4.2.2 The Agent Behavior

Note that a receiver agent is required to exchange information with the principal before replying CTS. But it does not modify the 802.11 protocol. A sender can transmit frames only on receiving CTS.

The detailed operation process of a principal and agents is described as follows:

i) When an agent requests to access to the channel, it sends an RTS to the receiver;

ii) On receiving the RTS, the receiver sends a request (RTS1) to the principal, to enquire whether the request of the sender is permitted.

iii) The principal computes the utility function and sends its decision to the receiver agent through a CTS1.

iv) The receiver checks the principal's decision from CTS1. If the request is rejected, it will not reply CTS to the sender agent. Otherwise,

v) The receiver agent replies CTS to the sender agent, and then the sender starts sending data frames to the receiver.

The Step 1 and Step 5 are the same as IEEE 802.11 MAC protocol. While Step 2 to Step 4 are added by our mechanism.

5. Simulation results

We conducted our simulations using Qualnet 3.7 simulation tool. We consider an Ad Hoc network with 36 wireless nodes managed by 4 principal nodes. There are 20 FTP traffic randomly set among these 36 nodes. The MAC layer protocol of IEEE 802.11 b is used. The simulated time is 500 seconds.

We compare the performance our mechanism with the default IEEE 802.11 b mechanism when there are no selfish nodes at first. The fairness performance measure has been defined in Eq. (2). Our throughput here is measured by the total amount of correctly received packets in bits divided by the simulated time.

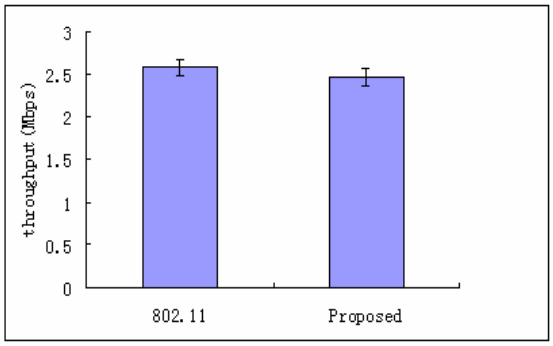

Fig. 1: Throughput comparison without selfish nodes

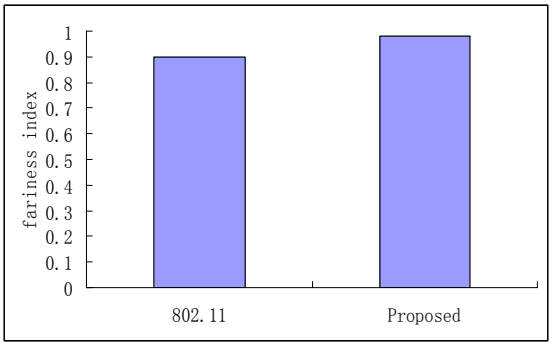

Fig. 2: Fairness comparison without selfish nodes

The simulation results show that our mechanism is fairer than 802.11b (Fig.2) while maintaining almost the same network throughput (Fig. 1).

Then we study the performance when there are selfish nodes. Fig. 3 shows the aggregate throughput of all selfish nodes (numbered from 1 to 10 out of 36 nodes in total). One can see that fairness can still be maintained (Fig. 4), while the throughput is degraded greatly (Fig. 3). This means that the misbehavior of the selfish nodes does not get extra performance improvement. On the other hand, the network performance (including itself) is degraded. So if selfish nodes are rational (from our assumption), it would be in their interest to obey the network rules in order to obtain the same network performance.

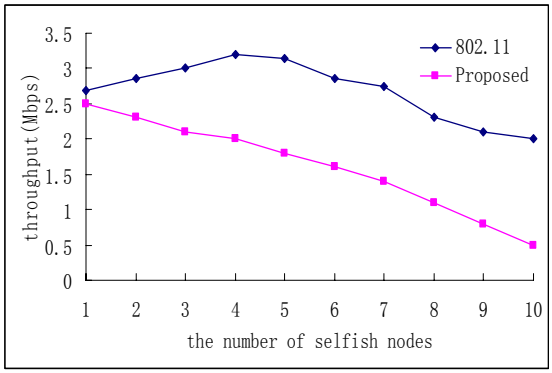

Fig. 3: Throughput comparison with selfish nodes

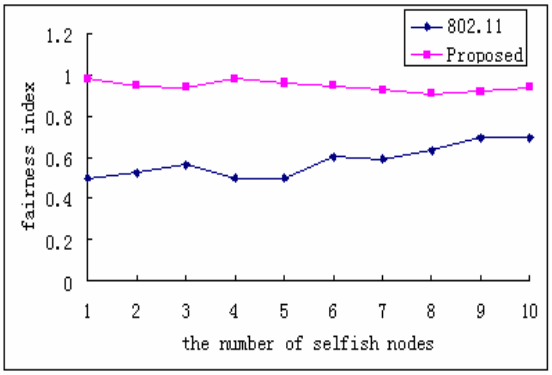

Fig. 4: Fairness comparison with selfish nodes

The false alarm rate is an important factor to evaluate a selfish behavior detection algorithm. The false alarm is defined as the rejection of the principal to a normal agent. Fig 5 shows the false alarm rate when the number of selfish nodes from 1 to 10. From the figure, one can see that the false alarm rate in our mechanism is low, with the maximum about 0.06 percent. As the number of selfish nodes increases, the false alarm rate is decreasing. This validates the accuracy of our mechanism.

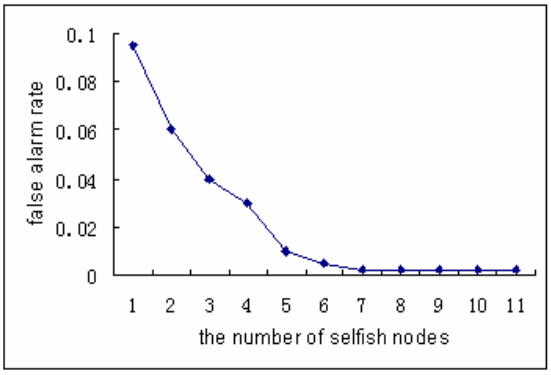

Fig. 5 The false alarm rate

6. Conclusion

This paper proposed a method to prevent the selfish behavior in the MAC layer of an IEEE802.11 WLAN. The proposed mechanism is based on the incentive and constraint method of the principal-agent theory. Simulation results show that our mechanism can effectively motivate the selfish nodes in the network to adopt a normal behavior, or its performance would suffer. The innovation in this paper is that the proposed mechanism does not rely on the detection and punishment of the selfish nodes adopted in other algorithms. Instead, we use an effective incentive and constraint mechanism to encourage the selfish nodes to obey the rules. Our algorithm need not modify the original 802.11 algorithm. Our idea may provide a new approach or theory to tackle the selfish behavior in the future.

Note that the mechanism of this paper is not perfect, because that some simple assumptions: The selection of a principal; and the management of agents need further study. Moreover, the deployment of principal nodes will cause some overhead, and the node mobility may take further impact. We will study these problems in our future work, to make our mechanism practical in real networks.

Acknowledgment

This research was supported in part by the National NSFC (Grant Nos. 90604013 and 60702038), the National 863 Program of China (Grant No. 2007AA01Z220) and by the Ministry of Education of China (Grant No. 708024). This is also supported in part by an NSERC Discovery Grant of Canada.

References

[1] S. Capkun J. Hubaux, L. Buttyan. The Quest for Security in Mobile Ad Hoc Networks. ACM, October 2001.

[2] J. Kong and X. Hong. ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad Hoc Networks. In Proceedings of ACM MOBIHOC, 2003.

[3] N.H. Vaidya P. Kyasanur. Detecting and Handling of MAC Layer Misbehavior in Wireless Networks. Dependable Systems and Networks, June 2003.

[4] I. Aad M. Raya, J. Hubaux. DOMINO: A System to Detect Greedy Behavior in IEEE 802.11 Hotspots. MobiSys, pages 84–97, 2004.

[5] S. Radosavac, A.Cardenas and J. S. Baras, “Detection and Prevention of MAC Layer Misbehavior in Ad Hoc Networks”. ACM, pages 17–22, October 2004.

[6] R. Jain, G. Babic and et al, “Fairness, Call Establishment Latency and Other Performance Metrics,” Technical Report ATM_Forum/96-1173, ATM Forum Document, Aug. 1996.

[7] W.R. Pires Júnior, T.H. de Paula Figueiredo, "Malicious Node Detection in Wireless Sensor Networks," vol. 1, pp.24b, Proceed IPDPS'04.

[8] Chunfeng Liu, Yantai Shu, Oliver Yang, A New Mechanism to Detect Selfish Behavior in IEEE 802.11 Ad Hoc Networks, In Proceedings of ICC 2008, Beijing China.

[9] J. Mirrlees, “An exploration in the theory of optimum income taxation,” Rev. Econ. Studies, vol. 38, pp. 175–208, 1971.

[10] Summers, Lawrence. "Beware moral hazard fundamentalists". Financial Times. 2008. 1.