Detection, penalization and handling of misbehavior in ad hoc wireless networks

Revoti Prasad Bora, Dheeraj Harihar and Saurabh Sehrawat *

Abstract

Devices in mobile ad hoc networks depend on the cooperation of other nodes for relaying of packets in the network. Medium Access Control protocols such as IEEE 802.11 are efficient with upper layer protocols only if the nodes adhere to the protocol and cooperate. Non-cooperating nodes may delay forwarding of frames or drop the frames altogether. This might be advantageous for individual nodes (from the point of view of saving energy) but it hampers the network as a whole. Such non-cooperating nodes disrupt communication between the cooperating nodes. We present a solution with a part modification of the IEEE 802.11 protocol to detect and penalize such non-cooperating nodes and thus making it unattractive to deny cooperation. For this we associate each node with an index, the Fairness Index, which is dynamically mapped to the behavior of the associated node. We not only penalize them but also ensure reliable routing throughout. In our scheme we also allow a convicted node to return to the mainstream if it shows its eagerness to cooperate. We prove the same using simulation results.

Keywords: *Ad Hoc Network, Fairness Index, Input Token Counter, Output Token Counter, Correction Factor, IEEE 802.11.*

1 Introduction

Mobile Ad-hoc Networks have witnessed enormous interest from the research community because of its ability to support communication without any infrastructure. The devices in such a network are battery powered and hence have to make the most efficient use of the limited energy available. For this various energy aware schemes have been formulated that make the optimal use of the limited energy available.

The concept of forwarding of packets, formulated to conserve energy, enables a device to send its data to another device without using its full transmission power. This saves a lot of energy but this scheme needs the cooperation of the various intermediate nodes. Although forwarding of packets is a healthy sign for the overall performance of the network yet it might not be that advantageous for individual nodes from the point of view of saving energy. A node that is running low in battery might not be able to transmit the packets with enough energy. Moreover a misbehaving node might purposely transmit the packets with low energy, delay the forwarding or even drop them. Thus the general assumption of a cooperating environment may help some misbehaving nodes to save their energy at the cost of the other cooperating nodes' energy. This would create serious problem for the achievement of the global predicate of the network. It would be advantageous for overall performance of the network to detect such nodes and take some steps to prevent such a misbehavior.

In this paper we address this problem and propose a solution by a part modification of the IEEE 802.11 protocol. We achieve this by making misbehavior costlier than cooperation, thus compelling the nodes to adhere to cooperation. We not only detect such nodes but also penalize them by temporal suspension. As soon as the node tries to return to the mainstream (i.e. co-operate) it is given a chance again.

The paper is organized as follows. In Section 2 we discuss previous work done in this area. We explain the preliminary concepts involved in section 3, and the overview of our idea in section 4. Our detailed explanation is opresented in section 5 followed by our evaluation and conclusions in section 6 and section 7.

*Department of Computer Science and Engineering, Department of Electronics and Communications Engineering, Motilal Nehru National Institute of Technology, India. Email: rebathip@gmail.com, dheeraj.mnnit@gmail.com, saurabhsehrawat@gmail.com. The order of the authors do not convey their contribution to this paper.

2 Related Work

There has been a lot of research in the various issues of misbehavior in the network layer but not much of the papers deal with MAC layer misbehavior. A common misbehavior for a node is to use more bandwidth than its fair share. Many proposals have been made to tackle such a problem. One approach is to assume one of the sender or the receiver to be co-operating and then to evaluate the behavior of the other side. Kyasanur et al.[3] discusses the issues of contention window size manipulation and proposes a solution assuming the receiver to be co-operating. Savage et al.[7] deals with the problem of misbehaving receiver and proposes a solution by eliminating the vulnerabilities in TCP.

Another approach is to detect the misbehaving nodes and formulate protocols that select routes avoiding these malicious nodes. Marti et al.[9] uses watchdogs and pathraters for this purpose. Buttyan et al.[7] uses the concept of a tamper resistant security module which maintains a *nuglet* counter at each node. Hubaux in his terminode project [3], solves this problem by enforcing a barter system in the network using *Beans*. A node must have a reservoir of beans in order to initiate a packet transmission.

One of the issues in designing protocols for ad hoc wireless networks is to make the scheme energy efficient. Srinivasan et al.[10] deals with the investigation and study of the ability of the network at guaranteeing a low power node the right to behave selfish and then its impact on the overall network performance.

3 Preliminaries

The concepts to be known are the Ad-Hoc Networks and the IEEE 802.11 specification. We define the following terminology used in presenting the proposed scheme.

Sender: Sender is a node which transmits a data packet to a receiver node.
Receiver: Receiver is a node which receives a data packet from a sender node.
Agent: An agent is a node that forwards packets from other nodes. An agent should cooperate for a network to sustain and this is our focus.

Sender, receiver and agent are different roles a node can perform at any time. Another aspect to be known is the reason why the nodes fail to cooperate. Each node is wireless and hence they tend to conserve energy when they are neither the sender nor the receiver and this is what results in their misbehavior. There is no direct check for the disobedience of a node.

3.1 Assumptions

A node is assumed not to shut down its interface as this would result in losing the benefits that it derives by participating in the network. The maximum extent of non-cooperation that a node could display is by dropping frames that are not of its direct interest. At start the network would be considered to have only co-operating nodes with Fairness Index values above the minimum value. A malicious node cannot take the advantage of this assumption as the Fairness Index, that is assigned to it at the start of the network, would drop with its misbehavior.

Though the problem dealt with can be handled at the upper layers yet the ability of the nodes to manipulate the system parameters also increases up the protocol stack. Hence we present a solution at the MAC layer itself. Moreover the overhead of adding an 8-bit header is compensated by the gain in throughput.

Moreover the nodes are assumed to be active participants in the network i.e. all the nodes are senders and receivers once in a way. Due to power constraints at the MAC layer, we have considered the series expansions which do not affect the performance of our algorithm.

3.2 Overview

In this paper we introduce a new variable, the Fairness Index(FI), and asociate it with each node participating in the network.The Fairness Index is mapped dynamically to the behavior of the corresponding node. We calculate this number by considering the behavior of the node for the time it behaves as an agent.

Nodes cooperate with any node only if it has an Fairness Index above a certain predetermined value called the Co-operation Threshold(CT). We detect and penalize nodes that fail to co-operate beyond a certain extent. When a node does not co-operate its Fairness Index drops. Its Fairness Index is sent in every data frame that leaves the node. Once a node has fallen into the category of misbehaving nodes, our algorithm prevents the other nodes that receive its frames, from accepting them. This forms

the penalization part as a node is not able to send any frame until its Fairness Index rises beyond the Co-operation Threshold.It would be able to send frames only if it tries to return to the mainstream by showing its eagerness to co-operate.

4 Proposed Scheme

Our proposed scheme consists of the introduction of the variable Fairness Index(FI) which is calculated according to a function f(x) as shown below:

$$f(x) = \begin{cases} log(e^{f(x-1)} + 1) - k & FrameForwarded \\ \\ -log(e^{f(x-1)} + 1) + t & FrameReceived \end{cases}$$

The values of k and t are used for scaling and depend on the choice of how fast the oscillations of FI could be tolerated. For calculating the FI we needed a function that dampens the effect of sending frames when the node is already above the CT. Similarly the function should also make sure that misbehavior by a cooperating node is quickly detected when above the CT. As it can be seen from the equation, the calculation of FI takes either the negative logarithmic or the positive logarithmic path depending on whether it receives or forwards frames. Figure 1 illustrates the two different curves choosen for our algorithm.

$$NodeDecision = \begin{cases} FI \geq CT & FrameAccepted \\ \\ FI \ll CT & FrameDropped \end{cases}$$

The modification to the MAC header is the introduction of an 8-bit field for the Fairness Index. The modification to the node description includes this variable to be stored in the node also.

Working of the algorithm in a node:
1. When a frame enters and is not for the node.
2. If FI ¡ Co-operation Threshold, drop frame.
3. FI is decremented using scaled negative logarithmic funtion.
4. When a frame leaves and the node is not the sender,
FI is incremented using scaled positive logarithmic funtion.
5. When frame is to be released FI is appended to the Frame Header.

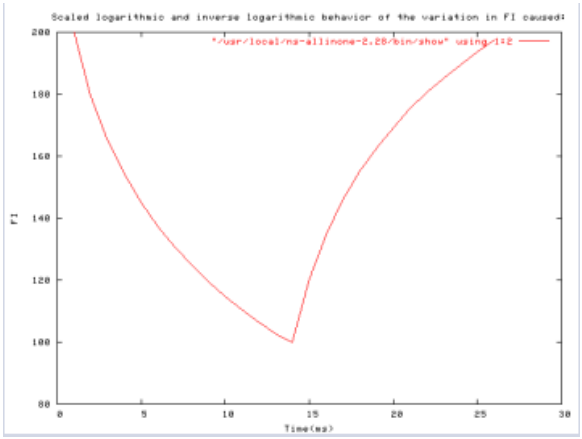

Figure 1: Sample simulation snapshots

We take up the cases of the nodes one by one.

1. When a node is cooperating and its FI is above the CT This is how a node will start and its value remains above the CT. This is the case when the node is forwarding the packets promptly. The value of its FI is also passed on to the other nodes along with its packets. Thus when it has to send any data, the other nodes also cooperate. Looking from the third layer, the DSR route requests and replies pass through the node properly and hence it is an active agent in the network.

2. When a node is non-cooperating and its FI is above the CT This is the situation a node might get tempted to be in. We assume that a node will never want to be perfectly prompt, but just enough prompt to escape being penalized. When a node starts disobeying and saving its energy at the cost of the network's performance, then its FI drops quickly. As per the rule, any node can drop frames which have a FI below the CT without its own FI being affected. Hence a node affecting the network in this phase can do so until its FI drops below the CT.

3. When a node is cooperating and its FI is below the CT Anode that has disobeyed can get back to normal operation if it shows its eagerness to cooperate. The problem that a node with a low FI faces, is that it does not become an active agent as nodes don't prefer its use. Hence it faces problem of inability to send information until it wastes a certain amount of energy trying the same. Thus the node

suffers(is penalised) inspite of being cooperating due to its earlier misbehavior, until its FI rises back to the minimum required value.

4. When a node is non-cooperating and its FI is below the CT Wrong When a node is below the CT and still disobeys it further spoils its FI and this is to its own disadvantage. This is because whenever it rquires to send any information in future, it will have to waste a lot of energy trying to get to its CT, apart from facing a time lag. Hence we tempt a node to instead cooperate normally and not face this delay and energy loss when it needs to send information.

4.1 Detection of Misbehavior

When a node misbehaves its FI goes below the CT which is passed on with every frame. So, the other nodes know the current behavior of the concerned node. This is the easiest way of detecting the misbehavior as every frame coming out of that node carries its FI.

4.2 Handling

When a node misbehaves it is isolated from the network by other nodes by dropping its packets and this act does not affect the FI of other nodes. A node must face dropped packets for a while in order to resume normal functioning in the network thus making misbehavior less attractive. The handling of the misbehavior of a node is local and does not affect the overall network. This is proved by the following two reasons:

1)When a third node tries to establish a path to some node through this misbehaving node, the packets of route request are dropped by the node that follows this misbehaving node. This is justified because the misbehaving node has already displayed an overall characteristic that is less beneficial to use it as an agent, even if it provides a shorter path.
2)When the misbehaving node tries to send data, it is blocked by the neighbours thus handling the misbehaving node locally. The rest of the network need not worry about the handling of this misbehaving node. Once the node gets its FI above the CT, then it is again a mornal member of the network and its frames can pass through its neighbours.

We do not worry about flooding of packets, which would block the neighbours from their normal operation. This is because the very reason of misbehavior of a node that we have considered is to save energy.

4.3 Other Issues

This scheme does not affect the network layer as when a DSR broadcast is made and there is a misbehaving node, the route requests and replies are blocked by the nodes after the misbehaving node. Hence the path that is formed does not include the misbehaving node thus maintaining the throughput. Also, the penalty in the form of dropped packets is an equal punishment for the misbehavior and the node thus prefers to behave with cooperation.

5 Simulation Results

We have performed simulations in NS2 by mainly modifying the MAC files for the 802.11 protocol. A scenario of 20 mobile nodes was created. The sender sends data at a constant bit ratio of 64kbps with each packet of size 160 bytes. In all the simulations the simulation time was 100s and DSR was used as the routing protocol. The results are averaged from 50 simulations. The number of misbehaving nodes were varied and the performance was analyzed. In figure-2 we plot the variation of the FI of a node that misbehaves in phases. The node has to spend some energy everytime its FI goes below CT, which was taken as 150. We have plotted graphs for varying percentage of misbehaving nodes and throughput in selected cases and the FI variation for a node. All these results are averaged in figure-3. In figure-3 we see that when 5 percent of nodes misbehave, the performance of the proposed scheme is somewhat lower than 802.11 but when the percentage of misbehaving nodes go above that our algorithm clearly performs better than the standard 802.11 algorithm. If the percentage of misbehaving nodes rises beyond 50 percentage, both the schemes fail because many routes are hampered. As per our previous assumption [3.1], if all the nodes are senders and receivers once in a way, our modification to the protocol performs better. This is mandatory for the network to survive. Incase this scheme is used in a network where there could be nodes that may transmit negligible amounts of data in their entire lifespan and just do the forwarding for others to communicate, then such a node could misbehave and yet live in the network for a long time after which it would have to spend comparatively lesser energy to send data due to the nature of the function used to vary the FI. Hence our scheme is more suitable for networks in which all nodes are active participants.

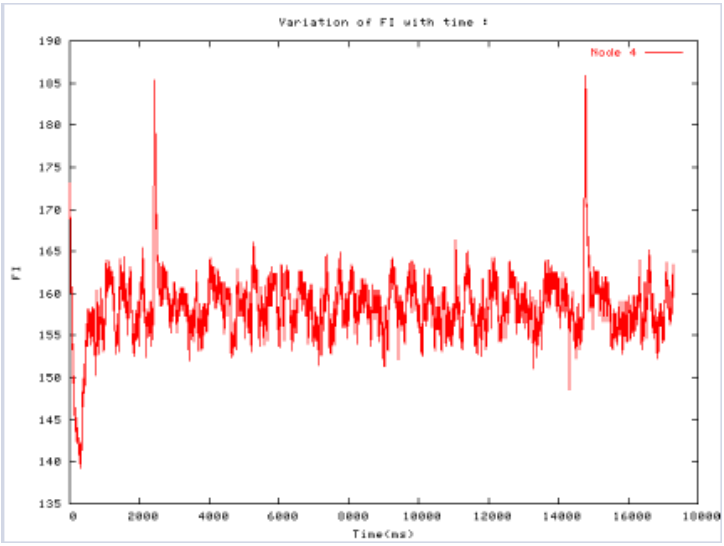

Figure 2: Sample simulation snapshots

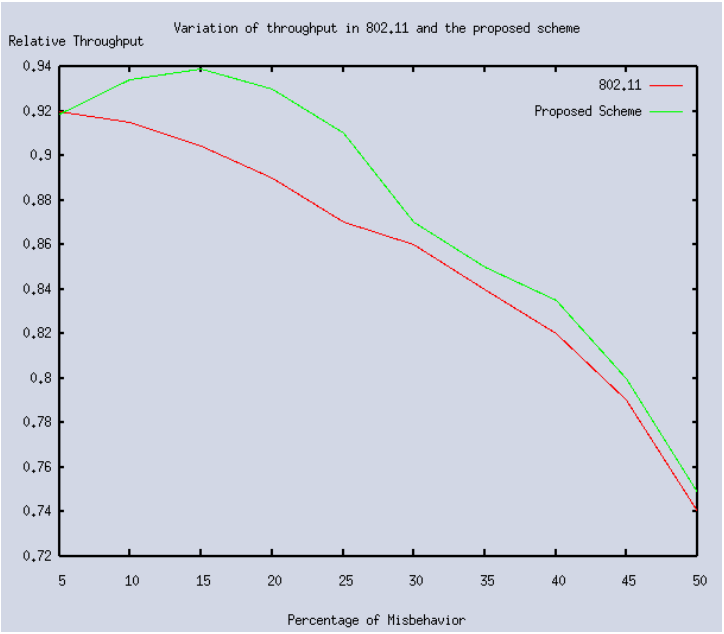

Figure 3: Sample simulation snapshots

6 Conclusions and Future Work

We have addressed the problem of packet dropping by misbehaving nodes while trying to save energy. We have proposed a part modification of the 802.11 protocol and explore the possibility of calculating the fairness of a node using its last fairness value with a function that performs the increment or decrement according to the events in a node. Our results have shown us that our scheme works well if the percentage of misbehaving nodes is not high. We plan to extend the proposed scheme by making it handle higher percentage of misbehaving nodes and making it work in special scenarios. Clubbing our solution with network layer solutions could be more effective, which has to be verified. Our scheme has been tested for the DSR protocol and the results are satisfactory. This scheme must be tested for other protocols and topologies.

References

[1] P. Michiardi and R. Molva. "Game theoretic analysis of security in mobile ad hoc networks." *Technical Report RR-02-070*, Institut Eurecom, April 2002.

[2] P. Kyasanur and N. H. Vaidya. "Detection and Handling of MAC Layer Misbehavior in Wireless Networks." *Technical report, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign*, August 2002

[3] J-.P. Hubaux. "Terminodes : Toward Self-organized MobileNetworks." *Technical Report "SSC/1999/022", EPFL-ICA*, June 1999

[4] S. Savage, N. Cardwell, D. Wetherall, T. Anderson. "TCP Congestion Control with a Misbehaving Receiver." *ACM Computer Communications Review, pages 71-78, October 1999.*

[5] S. Marti, TJ Giuli, K. Lai, M. Baker. "Mitigating Routing Misbehavior in Mobile Ad hoc Networks." *Mobile Computing and Networking, pages 255-265,2000.*

[6] Vikram Srinivasan, Pavan Nuggehalli, Carla F.Chiasserini, Ramesh R. Rao."Energy Efficiency of Ad Hoc Wireless Networks with Selfish Users"

[7] Levente Buttyan, Jean-Pierre Hubaux. "Stimualting Cooperation in Self-Organising Mobile Ad hoc Networks".