MAC Layer's Misbehavior Handling in Wireless Network

Gollagi S.G.[*], Ankaliki S.G. and Zinage H.R.

[*]HiraSugar Institute of Technology, Nidasoshi-591236, Karnataka,India, shantesh1973@rediffmail.com,
sankaliki@rediffmail.com, hema_zinage@rediffmail.com

Abstract- The 802.11 family uses a MAC layer known as CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) contention resolution mechanism for sharing the wireless medium. In this environment, hosts can not trusted and selfish host that fails to adhere to the medium access policies may obtain an unfair throughput share. For instance, IEEE 802.11 requires host competing for access to the medium to wait for backoff interval, randomly selected from the specified range, before initiating a next transmission. Selfish station may wait for smaller backoff interval than well-behaved station, thereby obtaining an unfair advantage of channel.

In this paper, an attempt is made to capture the behavior of misbehaving station and diagnosis such hosts in the network. Further, we discover the penalty scheme for punishing selfish hosts. Simulation results under this misbehavior model have indicated that, proposed scheme provides fairly accurate diagnosis and effective in restricting the throughput of selfish station to a fair share and hence successful in handling Medium Access Control (MAC) misbehavior in the wireless Network. The NetSim-2(simulator) had used for the simulation: CBR flow with rate 2Mbps, Packet size: 512 bytes and simulation time per run is 50 second.

Keywords: Contention resolution, Coordination function, Percentage Misbehavior (PM), diagnosis scheme, Penalty scheme, Network Simulator and MAC layer.

I. Introduction

Wireless Medium Access Control protocol such as IEEE 802.11 use distributed contention resolution mechanism for sharing the radio medium. This mechanism is typically based on cooperative policy (e.g. random backoff before transmission) that ensures a reasonably fair throughput share for all participating mobile stations [1,2]. In an environment where stations in the network are untrusted, some stations may misbehave by failing to adhere to the network protocol, with an intention of obtaining an unfair share of the channel. The presence of selfish stations that deviate from the contention resolution protocol can reduce the throughput share received by well-behaving stations. Thus, development of mechanism for detecting and handling such selfish misbehavior is needed. IEEE 802.11 is packed with two contention resolution mechanisms:

1) Point Coordination Function (PCF) for synchronous and time bound services (Centralized)

2) Distributed Coordination Function (DCF) for asynchrous services.

PCF is an optional feature in IEEE802.11 and suitable for only infrastructure based networks. DCF is the mechanism that can be used with ad-hoc networks and infrastructure based wireless networks. In this paper, we address misbehavior possible in the DCF mode, in infrastructure- based networks. However, using PCF, instead of DCF, may alleviate the misbehavior that we can identify, but PCF may offer lower performance then DCF during normal network operation.

II. Possible Misbehavior

Some strategies that misbehaving hosts may use for obtaining channel include:

1. Selecting backoff values from a different distribution with smaller backoff value then the distribution
2. specified by DCF (e.g. selecting backoff values from the range [0 - CW/2] instead of [0-CW] or by always selecting a fixed backoff of one slot)
3. Using a different retransmission strategy that does not double the CW value after collision.

Such selfish misbehavior can seriously degrade the throughput of well-behaved stations. Therefore, it is required to provide scheme for detecting and penalizing such stations in the network.

Most research [2, 3] addressing selfish misbehavior assumes that selfish hosts misbehave basically to improve their own performance (throughput, latency, energy etc…). But, this assumption may degrade the performance of well-behaving hosts in the network. Malicious misbehavior, on the other hand aimed at disrupting normal network operation, possibly with no performance gain to the misbehaving host. Malicious misbehavior includes: denial of services, Jamming etc. Many approaches have been proposed for addressing selfish misbehavior at the Network layer and are based on game theory. This paper addresses selfish misbehavior management at MAC layer in DCF mode for infrastructure based wireless network.

III. Proposed Approach: An overview.

We define the following terminology used in presenting the proposed scheme:

Sender: Host which wants to transmit a date packet to another host.

Receiver: Host which receives a date packet from sender. The receiver monitors the sender host to detect sender's misbehavior. Host can assume different role (sender/receiver) at different times. Sender sends DATA packet to receiver after an optional RTS/CTS exchange.

The proposed scheme allows the receiver to detect sender misbehavior identified. In public wireless network (infrastructure based) the base station is well-behaved; there is no misbehavior when it is sending. But other hosts sending data to base station using DCF mode are untrusted and may misbehave to gain higher throughput share then well-behaving hosts. Therefore, we assume that the receivers are well-behaved while presenting the scheme. It is also assumed that there is no collision between the receiver and sender. All these assumptions are justifiable for infrastructures-based wireless network.

The proposed scheme is designed to handle selfish MAC layer misbehavior by hosts in using IEEE 802.11 DCF mode. In IEEE 802.11, a sender transmits an RTS after waiting for a randomly selected number of slots in the range [0 – CW]. Consequently, the time interval between consecutive transmissions by sender can be any value within the above range. Our approach use new backoff scheme that enables receiver to identify sender misbehavior within a small observation interval. How? Instead of sender selecting random backoff, the receiver selects a random backoff value and sends it in the CTS packet and ACK packet to the sender. The sender uses this assigned backoff to initialize backoff counter" With these modifications, receiver knows the exact backoff values that sender is expected to use. Hence, receiver can identify a sender deviating from protocol by observing the number of idle slots between consecutive transmissions from the sender. If this observed number of idle slots is less then the assigned backoff, then the sender may have deviated from the protocol. A small history of deviation over received packets can be used in diagnosis with high probability. Deviating senders are penalized, thereby discouraging misbehavior. The proposed scheme has three components. 1] The receiver decides, at the end of transmission from the sender, whether sender has deviated from the protocol for that particular transmission. 2] If receiver has identified the deviation for transmission from the sender, it penalizes the sender on the magnitude of the perceived deviation for that transmission (Penalty Scheme). 3] Based on the magnitude of the received deviation over multiple transmissions from the sender, the receiver identifies senders that are indeed misbehaving (Diagnosis scheme).

IV. Identifying Deviation from the Protocol:

In our scheme the receiver R dictate the backoff value to be used by sender S. The first time S sends a packet to R, S may use an arbitrarily selected backoff. For all subsequent transmissions, the sender has to use the backoff values provided by R. Fig. (1) below depicts receiver and sender interaction in our scheme.

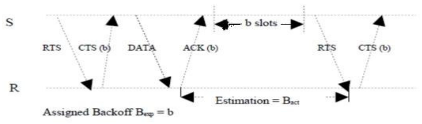


Fig.(1): Receiver sender Interaction.

When the receiver R receives RTS from S, R assign backoff value $B_{exp} = b$ to S in CTS and ACK packets. Receiver select $B_{exp} = b$ from the range [0, CW]. The R may misbehave by backing off for a smaller duration then B_{exp}. The R observes the channel status during the interval between the sending of an ACK by R and reception of the next RTS from S. The R records the length of this interval in slots, K, as well as number of slots that were idle B_{act} during this interval. The S is designated as deviating from the protocol if the observed number of idle slots B_{act} is smaller then a specified fraction α of the assigned backoff B_{exp}. ie. $B_{act} < \alpha * B_{exp}$, where $0 < \alpha \leq 1$

We now describe sender misbehavior during packet retransmissions. Every RTS sent by sender has an attempt number included in a new field in the RTS header. The sender sets the attempt number to 1 after successful transmission, and increment it by 1 after every unsuccessful transmission. The contention window CW maintained by the sender is set to CW_{min} after successful transmission and, after an unsuccessful transmission $CW = \min \{ (CW_{min} + 1)*2^{i-1} - 1 , CW_{max} \}$ for i^{th} transmission attempt as in IEEE802.11. Fig. (2) demonstrates the protocol operation after a collision. In figure, the number in the parenthesis next to RTS indicate attempt number.

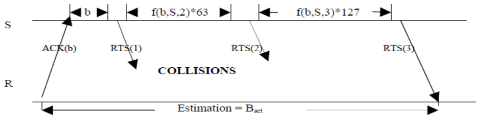


Fig. (2): Receiver sender interaction during retransmission

Assign backoff: b, CWmin = 31

Backoff value Expected by receiver: $B_{exp} = b + f(b,S,2)*63 + f(b,S,3)*127$

When collision is detected sender(S) need to choose the new backoff value using deterministic function f as : newbackoff = f(backoff, SenderId, attempt) * CW, where backoff(b) is previously assigned by receiver, senderId is the unique sender Identifier and attempt is attempt number maintained by the sender. The f used by S for computing newbackoff for retransmission attempt is given by:

$$\mathbf{f\,(backoff,senderId,attempt) = \{\, \{aX + c)\ mod\ (\,CW_{min} + 1)\,\}\,/\,CW_{min}}$$

Where a = 5, c = 2*attempt + 1 and X = (backoff + senderId) mod (CW_{min} + 1), The f generates a uniform random number between 0 & 1 and f is chosen carefully to ensure that, after collision, the colliding sender will select different backoff with high probability. When RTS is received by receiver (possibly, after multiple transmission attempts), the receiver can then estimate B_{exp} by applying same function f as sender using attempt number in RTS.

ie. $B_{exp} = \text{backoff} + \sum f(\text{backoff}, \text{senderId}, i) * CW_i$ for $i = 2$ to attempt.

Where CWi is the contention windows for the i^{th} transmission attempt as in IEEE802.11. This estimated backoff is then used in checking for possible deviation as explained before. It may possible for the sender to provide incorrect attempt number values in the RTS. To ensure that senders provide correct attempt number, the receiver can sense channel to identify high collision intervals. During these intervals, the receiver can analyze the traffic to identify any sender achieving a large number of successful transmissions than other hosts or having smaller average attempt number values than other hosts. It will be really hard for the misbehaving sender to persistently send incorrect attempt number without being detected.

V. Penalty Scheme

Host deviating from the protocol may obtain a larger throughput share then well-behaved hosts.

The penalty scheme penalizes deviating host by assigning larger backoff values than those assigned to well-behaved hosts. We use principle that hosts deviating more should be assigned larger penalty. After detecting a host deviating from the protocol, we measure the deviation as:

$$D = \max \{\alpha * B_{exp} - B_{act} \, . \, 0\}$$

and assign this measured deviation as a penalty to the sender. Penalty scheme adds a penalty for every perceived deviation. Since the penalty added is proportional to the amount of deviation. This will be small for a well-behaved host, if scheme incorrectly detect well-behaving hosts as deviating from the protocol. Simulation results show that the throughput obtained by well-behaved hosts when the penalty scheme is enabled is comparable to that obtained when using IEEE802.11 MAC protocol.

VI. Diagnosis Scheme

We use two metrics W and Threshold. The receiver maintains a moving window containing information about the last W packets received from each sender. When a new packet is received, the difference (B_{exp} - B_{act}) is stored in moving window. A positive difference indicates that the sender waited for less than the expected backoff. A negative indicates that the sender waited more than expected backoff by receiver. If sum of this difference in the previous W packets is greater than a threshold, then the sender is designated as “Misbehaving “. A constantly misbehaving host will have positive for the most packets and is likely to be diagnosed. The choice of W and Threshold does not affect the penalty scheme. Hence, a sender adapting to W and Threshold will still have penalty added for every perceived deviation, even if the host is not immediately diagnosed to be misbehaving. Threshold may be adaptively selected, based on the channel conditions, to maximize the probability of correct diagnosis while minimizing the probability of misdiagnosis.

Thus, penalty and diagnosis schemes together ensure that a misbehaving host can not obtain a larger fair of the channel without being diagnosed as misbehaving. The MAC layer may refuse to accept packets from the misbehaving host (after diagnosed to be misbehaved). Alternately, higher layer can be informed of the misbehavior. The proposed scheme can be augmented with authentication mechanisms provided by higher layers to identify such misbehaving hosts.

VII. Simulation Results

Netsim-2 has been used for simulation of network using proposed scheme. In the simulations, all the sender hosts are backlogged. The traffic from S to R (base station) is a CBR flow with rate 2Mbps and packet size is 512bytes. Simulation time per run is 40second. The results are averaged over 50 runs of the simulations. Hosts are stationary in all simulations.

CASE-I: We have first evaluated performance of protocol in the absence of misbehavior. The number of sender communicating with Receiver(R) is varied from 1 to 128 and all senders are well-behaved. As we can see in the Fig. (3) the average throughput obtained when using new scheme is comparable with IEEE802.11 across different network sizes (Two curves almost overlap). Hence, our penalty scheme does not degrade the aggregate throughput of the network.

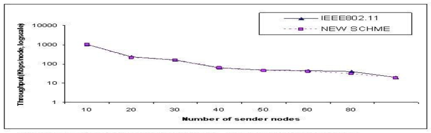
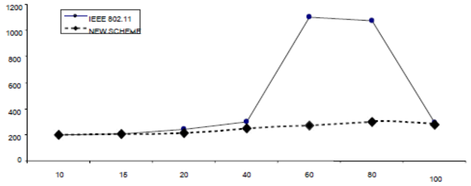


Fig.(3): Performance analysis in the absence of misbehavior

CASE-II: Fig. (4) Compares the throughput obtained by a misbehaving host using proposed scheme with that obtained using IEEE802.11 protocol. We define fair share as the throughput obtained by a host when it is using New scheme and fully confirming to the protocol (ie. PM = 0%, while for the 802.11 the misbehaving host obtained a large throughput share even when the extent of misbehavior is not too high.



X-axis: Percentage Misbehavior (PM).
Y-axis: Throughput (Kbps/nodes, logscale).

Fig. 4-Performance analysis in the presence of misbehavior

Hence proposed penalty scheme is fairly successful in ensuring reasonable throughput for well-behaved host in the presence of misbehaving hosts. When PM(Percentage Misbehavior) is close to 100% the misbehaving host backoff for small fraction of assigned backoff and consequently the our scheme can not restrict the throughput to the misbehaving hosts.

VIII. Conclusion

MAC layer‘s Misbehavior is an important requirement in ensuring a reasonable throughput share for well – behaved host in the presence of misbehaving host. In this paper, we have presented new backoff scheme that simplifies misbehavior detection. Simulation results have indicated that this scheme provides fairly accurate misbehavior diagnosis. In future, we will extend the work for diagnosis scheme using Neural Networks tools and work on handling misbehaving host in Ad-hoc wireless networks.

References

[1] Bhargavan V., Demers A., Shenker S. and Zhang L. Media Access protocol for Wireless LANS, proc. ACM SigmComm, Sept. 1994

[2] Karn P. , ‘MACA-a new channel access method for packet radio’

[3] N.Nisan and A. Ronen, ‘Algorithmic Mechanism design”

[4] Savege S., Cardwell N., Wetherall D. and Anderson T. (1999) *ACM Computer communication* Rev., 71-78.

[5] Mackenzie A.B. and Wickr S.B. (2000*)* *IEEE Communication Maga*zine, 39(11), 126 -131, 2000.

[6] IEEE transaction on ‘Mobile Computing’ Volume 4, No. 5, oct-2005

[7] Burroughs D.J., Willson L.F. (2002) *Proc. IEEE Intel performance computing and communication, Confrence*.

[8] Jonathan Bredin, David Kotz and Daniela Rus. Utility Driven Mobile-Agent Scheduling. Technical Report PCS-TR98-331, Dept. of Computer Science, Dartmouth College, May, 1998. Revised October 3, 1998.

[9] Jonathan Bredin, Rajiv T. Maheswaran, Çagri Imer, Tamer Basar, David Kotz and Daniela Rus. (2000) *In Proceedings of the Fourth International Conference on Autonomous Agents, ACM Press,* 349-356.

[10] Qun Li and Daniela Rus. (2002) In *IEEE MASCOTS Workshop on Mobility and Wireless Access*.