WiBi MAC: Biased Medium Access Control for WiFi

Cameron A. Keith
Computer Science and
Engineering Department
Southern Methodist University
Dallas, Texas USA
ckeith@smu.edu

Anna A. Carroll
Computer Science and
Engineering Department
Southern Methodist University
Dallas, Texas USA
aacarroll@smu.edu

Dylan C. Fansler
Computer Science and
Engineering Department
Southern Methodist University
Dallas, Texas USA
dfansler@smu.edu

Ethan Busbee
Computer Science and
Engineering Department
Southern Methodist University
Dallas, Texas USA
ebusbee@smu.edu

ABSTRACT

In this paper, we present WiBi MAC, a biased Medium Access Control (MAC) protocol for WiFi. The standard medium access approach utilized by WiFi relies upon Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) combined with slotted Aloha. This approach achieves a fair packet access scheme that is similar to Processor Sharing (PS). Biased scheduling approaches, such as Shortest Remaining Processing Time (SRPT), have been shown to yield better throughput and faster expected completion times than PS in computer process scheduling. The WiBi MAC for Wifi approximates SRPT in order to achieve greater throughput and utilization of the wireless medium. WiBi MAC operates by having each node choose a maximum potential back off time that is proportional to the number of packets that are ready to send. By favoring packets belonging to short streams over packets belonging to longer streams, we achieve a simulated throughput improvement of XXXX over the standard fair exponential back off approach used by WiFi.

1. INTRODUCTION

Current WiFi MAC protocol is primarily designed around achieving a completely fair protocol where all nodes in the network have the equal probability to interact with the router. This allows for the network to be fair in resource allocation between the nodes at the expense of reducing the overall network throughput performance. The standard WiFi protocol works well in environments where all of the data traffic is fairly similar in size and priority but begins to suffer performance degradation when you have users with starkly different data sizes. A WiFi protocol that instead of optimizing for fairness of access optimizes for near-maximum throughput on the network would cause no negative effects when implemented in a network that has very similar data traffic size, and in the cases where a network has different sizes of data it would increase the throughput of the network by having users that have smaller data sizes will be able to transmit all of their data faster than a user that is sending a larger amount of data.

Current WiFi works with the model of when a node has a data packet to send rolls a random number between 1 and a standard minimum value to establish a back off time for the number of slots to wait to pass before sending its data. This is similar to processor sharing in which programs have a uniformly random chance to be selected for access to the processor. Processor sharing also has functionality similar to WiFi MAC's exponential back off, for when nodes have a collision, which prevent larger programs have an increased delay between times they have access to the processor to ensure that they do not monopolize the processor ensuring that the smaller programs have the ability to use the processor as well.

SRPT achieves the highest throughput, lowest mean completion time, when applied to a single channel processor [2]. It achieves this by always choosing the nest program that has the lowest remaining process time, assuming that the time needed to complete processing is known at the start. A similar process can be implemented in the WiFi MAC protocol with the assumption that a node knows the size of the packet stream it needs to send, the node would then base the upper bound, of the back off random number selector, on the size of the packet stream that the node has left to send. As a node sends data over the network it gradually lowers its upper bound allowing it to have a higher chance of sending its remaining packets in a shorter time frame. This approach bias the network in favor for packet streams that were small in the beginning as well as streams that are near completion. For instances where the size of the packet stream is unknown a different way to bias the network towards streams of a smaller size is to base the upper bound of the random back off of the current frame number of the packet in the stream being sent. This approach has all streams start with the same range for the back off and gradually increasing the range as the node sends more and more packets successfully across the network with larger streams having a higher upper bound than the smaller streams. The final approach to creating a biased protocol that improves the throughput of the network is to dynamically alter the bounds of the back off based on the current level of traffic in the network as well as the size of the stream remaining. If the node detects that the network traffic has increased and that it has a long stream remaining then the node will increase the delay between its own packets to allow other nodes to communicate.

***** Talking with Ethan over the break to discuss the results********

The remainder of this paper is organized as follows. In Section 2 we review related work to medium access control. In Section 3 we discuss the results of the four different simulations: standard WiFi MAC, MAC protocol based on the SRPT algorithm, back off based on the current frame number in the stream, and lastly a dynamic back off window changed based on the traffic level in the network. Finally, we draw the relevant conclusions in Section 4.

2. RELATED WORK

Most research done on unfair MAC protocols has been conducted on how to detect and prevent unfair behavior from a node on the network. Some has focused on merely detecting and blocking the misbehaving node, while others have suggested implementing a punishment of sorts on the misbehaving nodes. Ways that nodes can misbehave in order to increase their own performance is to refuse to forward packets in order to save energy, or to select a smaller backoff in order to increase throughput for their own traffic. These misbehaving nodes can seriously degrade the network throughput for other, well-behaved nodes. [1].

The most research has been done on detecting the manipulation of the random back off [3], detecting if a node is cheating by sending before the end of the guard band This approach only benefits a single user as they crowd out other users as they maximize their bandwidth [**?**]. As a result it also decreases the throughput of the network as other user's data is put on hold they think the communication channel is always busy.

A similar task to creating an unfair MAC protocol that everyone follows is detecting when a single user, or a small group of users, is being unfair and the process of handling them. A few approaches to remediate unfair behavior have been to exclude the misbehaving node from routing operations, encourage nodes to cooperate by penalizing misbehavior, or to incentivise good behavior by paying nodes for cooperating. Another protocol detects unfair behavior by having a sender transmit an RTS (Request to Send) after waiting for a randomly selected number of slots in the range [0; CW].After the initial transmission between hosts, the recieving host sends with their acknowledgement: a random value that the sender then uses as the back off counter for each subsequent transmittion during the stream. [1] With this protocol, if a recieving node recieves a packet before the appropriate number of frames has passed, then the sending host is not obeying the Protocol and can then be handled accordingly.

Our purpose, however, is to prove that implementing an unfair protocol where network traffic is sent according to a certain priority will actually improve throughput for all nodes in the network. The proposed protocol will ideally also maintain its increased performance even when all nodes on the network are not operating under the same protocol.

3. ANALYSIS

A basic simulation was used to analyse and compare the results from the standard fair WiFi protocol versus the previously mentioned biased protocols. The protocols were tested with a network consisting of **** end nodes with the nodes sending data following the distribution of data according to *Citation needed* The standard WiFi protocol is used as the basis of comparison that the other protocols are measured against as a protocol that does not doesn't at least match the throughput of WiFi's current protocol dos not meet the requirements of improving the throughput the network. In the simulation, it is assumed that all end nodes on the network are following the same protocol, everynode behaves in the same manner as other nodes.

4. CONCLUSIONS

5. REFERENCES

[1] P. Kyasanur and N.F. Vaidya. Detection and handling of mac layer misbehavior in wireless networks. In *Dependable Systems and Networks, 2003. Proceedings. 2003 International Conference on*, pages 173–182, June 2003.

[2] Linus E. Schrage and Louis W. Miller. The queue m/g/1 with the shortest remaining processing time discipline. *Operations Research*, 14(4):670–684, 1966.

[3] M. Shanthi and S. Suresh. Detecting mac layer misbehavior in wifi networks by co-ordinated sampling of network monitoring. *International Journal of Innovative Research in Science, Engineering and Technology*, 3(1), Feb 2014.