Increasing WiFi Performance Through Packet Collisions

Grant Ayers
GAyers@smu.edu

Quentin Morris
QMorris@smu.edu

Daniel W. Engles
dwe@alum.mit.edu

***Abstract*—In this paper we show that added functionality of intentional collisions over a simulated ad hoc wireless mesh network, referred to herein as the GQ simulator, can decrease slowdown. Using the IEEE 802.11 existing MAC protocols we simulate the added functionality of nodes with less packets to send will intentionally cause collisions in the transmission slots before their transmission slots to optimize their own throughput. In current 802.11 when a collision occurs both transmitting nodes experience exponential backoff and adjust their contention window and choose a new transmission slot, the GQ simulator's simulation of 802.11 causes there to be no unintentional collisions and ensures that all collisions are caused by a node with less packets. The GQ simulator takes advantage of the Distributed Control Function and MAC protocol to decrease its slow down up to 29% dependent on the various data distributions and nodes transmitting.**

***Index Terms*—Intentional Collisions, 802.11, WiFi, WLAN, MAC, DCF**

I. Introduction

TThe IEEE 802.11 standard encompasses a number of functions and algorithms used to establish Wireless Local Area Networks (WLANs). Traditionally, Medium Access Control (MAC) for 802.11 has taken an approach that treats all nodes and their traffic in the same, 'fair' manner. MAC for 802.11 provides medium access control to ensure that nodes sending packets over the network get to utilize network resources as efficiently as possible and so data retransmission is kept to a minimum. The current MAC flow control utilizes a slotted Aloha algorithm with exponential back off in combination with Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to avoid collisions. The 'fair' approach being used in 802.11 protocol creates network congestion which could be alleviated through an 'unfair' approach to accessing the medium in WLAN. An alternate method for handling the flow control of packets, such as utilizing collisions to add 'unfairness' to the MAC protocol for 802.11 may offer a promising new outlook for the future of data transmission via IEEE 802.11 based WLAN.

Some assumptions needed to be made before simulating the IEEE 802.11 network. One of the notions we assumed is that all communication observed and analyzed happens on one channel. All nodes in a WiFi network that seek to communicate with one another do so at the same frequency, to model current 802.11 and how their protocols handle collisions. Another assumption is that MAC protocol for 802.11 does not support intentional collisions. Anytime a transmitting node in a channel experiences a collision, both the node transmitting and the node that caused the collision do exponential back off as defined by the current DCF protocol. To introduce intentional collisions as a method to increase efficiency of communication, we also assume that nodes can transmit packets that cause a transmitting node to experience backoff without the colliding node being forced into backoff.

We found the slow down under our proposed MAC protocol had the highest percentage of decrease for nodes that followed a Pareto distribution. For the even and normal distributions, the performance was worse than the original MAC protocol where all nodes that experience collision perform exponential back off. There was not a significant difference between the slow down of the even and normal distributions. [Results TBD]

According to our results, if the flow size distribution of network traffic follows a Pareto distribution then our simulated 802.11 WLAN provides increased efficiency by using MAC protocols to cause intentional collisions with nodes of larger flow sizes to favor nodes with smaller flow sizes. With many nodes transmitting small numbers of packets versus large numbers of packets per node, wait time was minimized due to the prioritized transmission of these small packets with minimal exponential back off which in turn minimized their slow down. Large packets that were intentionally collided with had a longer wait time with their contention window size constantly increasing due to exponential back off, but the overall slowdown of the simulated WLAN was minimized. Minimal wait time for nodes with small flow sizes and forcing nodes with larger flow sizes to have longer wait times ensured that overall the slow down is kept to a minimum.

In a WLAN consisting of 2 or more nodes, collisions are inevitable. Instead of focusing efforts on collision avoidance of packets sent over 802.11 networks to increase efficiency of data transmission, causing packets to intentionally collide during communications could provide a more effective solution to minimizing total transmission time. Under the current MAC protocol for WiFi, the Distributed Coordination Function (DCF) handles all packet collisions the same. All packets that experience collision are dropped and the nodes involved perform exponential back off to help clear channels of communication and increase the probability of successful transmission. Each node increases its Contention Window (CW) size by a factor of two and randomly transmits at a time inside of this window. In the case of multiple collisions, this process is repeated until a maximum size is reached. In

this paper we seek to show that adjusting the DCF to be unfair, favoring nodes with smaller flow size will be most efficient for communication in a large IEEE 802.11 based WLAN. Nodes with large flow sizes, meaning they have a high number of packets to send, will be collided with so that nodes with smaller flows can access the channel. Short data transmissions allow the channel to be used by many nodes with small flows and have nodes with the largest flows continuously increase their CW. Although nodes with large flow sizes experience more wait time, mean slowdown is held to a minimum because their large flow size makes their wait time more negligible than nodes with small flow sizes.

II. IEEE 802.11 MAC

The 802.11 [1] Wireless Local Access Network which is the basis of all current WiFi has the sub-layer protocol medium-access control (MAC) [2] that dictates how the router or access point should best be shared so that the most devices can use it efficiently. One of the key responsibilities of the MAC sublayer is collision resolution and retransmitting in instances where collisions occur. The MAC sublayer uses Acknowledgements and Non-Acknowledgements to let nodes know when a packet was not received so that the node can reuse the medium of transmission to retransmit its packet. The MAC sublayer primarily uses the DFC protocol [3] which is responsible for assigning transmitting nodes a random transmission slot within the contention window and when collisions occur the DFC protocol is responsible for increasing the contention window to twice what it is currently and then randomly assigning a transmission slot within the new contention window. The DCF protocol is fair as it uses randomness to assign nodes transmission slots regardless of their data size, however intentional collisions could make the DCF protocol unfair and decrease slow down by utilizing the exponential backoff function to force nodes transmitting more packets to wait until nodes that have to transmit fewer packets transmit their data.

III. IEEE 802.11 DCF

Transmitting-Nodes that abide by IEEE 802.11 standards determine Medium Access Control through Distributed Coordination Function (DCF). Using DCF all nodes operate independently and assume all other nodes are equipped with DCF. DCF contains protocols on avoiding collisions through Carrier Sensing Multiple Access Collision Avoidance, which requires the use of Request to Send (RTS) and Clear to Send (CTS). All of these protocols in culmination result in collision avoidance without coordination between nodes and without being able to detect collisions when they occur.

When a node has data to transmit it begins by deciding its backoff interval which is a randomly selected number between 0 and its Contention Window (CW). Every time the medium that the node wishes to transmit on is not busy the node will decrement the backoff interval by one. Once the node's backoff interval reaches zero it will transmit. A node will transmit the entirety of data regardless of potential collisions after it is finished transmitting it will receive an Acknowledgement (ACK) that the packet was successfully received. If the package was not successfully transmitted it will receive a Not Acknowledged (NACK) or no response, due to collision or a noisy network. If a node receives a NACK or no response it will double its CW size, while being less than a network specific maximum, recalculate its backoff interval and begin the process again. Once a node receives an ACK it will set its CW to the network specific minimum.

The DCF methods, utilizing randomness, allows for equal chances for nodes to transmit. When a large number of nodes are attempting to transmit through the same medium collisions occur, which doubling the CW allows for nodes that experience collisions or noise to transmit with less frequency, which increases their chance of successful transmissions. If there are two Nodes on a WLAN that abide by 802.11 the first thing that will happen is the DCF protocol of the MAC sublayer will assign each node a unique transmission slot which will be a random number between 1 and the contention window; the contention window is determined by the minimum contention window of the WLAN. Note, an increase in devices that are connected to the WLAN should correlate with an increase in contention window size. After those two Nodes are assigned their first transmission slot they will decrement their own transmission slot counter for each time the medium is available to take a transmission until the node with the smallest transmission slot's counter reaches 0. At that time the node will transmit its data, if two nodes have the same transmission slot then the channel that they are transmitting over will be too noisy to decipher either transmission and they will not receive an ACK. At this time the two nodes will double their contention window size and reselect a transmission slot from 1 to their new contention window, this process is referred to as exponential backoff. They will continue this process until they successfully transmit during a time where another node is not transmitting this data and receive an ACK. Every time a node successfully transmits its contention window will return to its original size.

IV. GQ Simulator

There are a number of scenarios that needed to be looked at to determine if an unfair processing time based DCF would be the best implementation of MAC in IEEE 802.11 based WLANs and what situations and node distributions result in the most efficient data transmission. To test these scenarios we have created a logic level simulation of the MAC for an 802.11 network. The simulator models an 802.11 network with nodes trying to send packet data in the same channel. The network setup is a P2P mesh network in which all nodes have the ability to sense noise over the channel.

A. Node Distribution

Depending on the network distribution, nodes are created according to Algorithms 1-3 with each node given a number of packets with the appropriate size corresponding to the first parameter in the function setFS. The number of packets for each parameter is outlined in figure 1. Algorithm 1 creates a

Pareto distribution of nodes in which 80% of the packets make up 20% of the total packets in the network while the remaining 20% of packets make up 80% of the total packets. Algorithm 2 creates an even distribution of nodes where each number of packets is allotted to 20% of the nodes in the network. Algorithm 3 simply draws packet numbers at random between 1 and 1000 to create a Normal (Gaussian) distribution. This algorithm is dependent on the assumption that drawing random numbers from a uniform distribution (the std::rand function in C++) will produce a normal distribution of numbers.

```
if(nPercent < 0.5)
{
    node->setFS(1,seed);
    nodes.push_back(node);
    seed++;
    continue;
}
else if(nPercent > 0.5 && nPercent < 0.6)
{
    node->setFS(2,seed);
    nodes.push_back(node);
    seed++;
    continue;
}
else if(nPercent > 0.6 && nPercent < 0.8)
{
    node->setFS(3,seed);
    nodes.push_back(node);
    seed++;
    continue;
}
else if(nPercent > 0.8 && nPercent < 0.9)
{
    node->setFS(4,seed);
    nodes.push_back(node);
    seed++;
    continue;
}
else if(nPercent > 0.9 && nPercent < 0.95)
{
    node->setFS(5,seed);
    nodes.push_back(node);
    seed++;
    continue;
}
else
{
    node->setFS(6,seed);
    nodes.push_back(node);
    seed++;
}
```

Algorithm 1

B. Data transmissions

Maintaining the integrity of the current MAC used in 802.11 networks, the simulator assigns a transmission slot within the original contention window set when each node in the network is created. Nodes are transmitted according to their time slots chosen. To model the traditional MAC, if nodes share a transmission each node has to perform backoff and choose a new slot within their larger contention window. To simulate the unfair MAC, nodes that share a transmission slot compare the number of packets they have to send. Whichever has the larger number of packets performs backoff (up to the maximum contention window size), allowing the other node to transmit packets. When the network is created, a total count of packets is taken to determine when transmission has finished. Nodes transmit data until they reach a slot where another node is set to send data. The packets sent successfully (the difference between the next transmission slot and the number of packets a node has to send) is used to decrement the total number of packets in the network and the contention window is reset to the original size. If the difference in the number of packets a node has to send and the current transmission slot is less than the next node's transmission slot, the wait time is set to the current transmission slot. Algorithm 4 shows this process which continues until the total number of packets in the network is zero. The difference between Algorithm 4 for traditional and unfair MAC is the comparison done on lines 316-322. In the case of traditional MAC, both nodes perform backoff.

```
if(nPercent < 0.2)
{
    node->setFS(1,seed);
    nodes.push_back(node);
    seed++;
    continue;
}
else if(nPercent > 0.2 && nPercent < 0.4)
{
    node->setFS(2,seed);
    nodes.push_back(node);
    seed++;
    continue;
}
else if(nPercent > 0.4 && nPercent < 0.5)
{
    node->setFS(3,seed);
    nodes.push_back(node);
    seed++;
    continue;
}
else if(nPercent > 0.5 && nPercent < 0.7)
{
    node->setFS(4,seed);
    nodes.push_back(node);
    seed++;
    continue;
}
else if(nPercent > 0.7 && nPercent < 0.9)
{
    node->setFS(5,seed);
    nodes.push_back(node);
    seed++;
    continue;
}
else
{
    node->setFS(6,seed);
    nodes.push_back(node);
    seed++;
    continue;
}
```

Algorithm 2

```
for(int i = 0;i < nSize; i++)
{
    NNode* node = new NNode();
    node->setFS(7,seed);
    nodes.push_back(node);
    seed++;
}
```

Algorithm 3

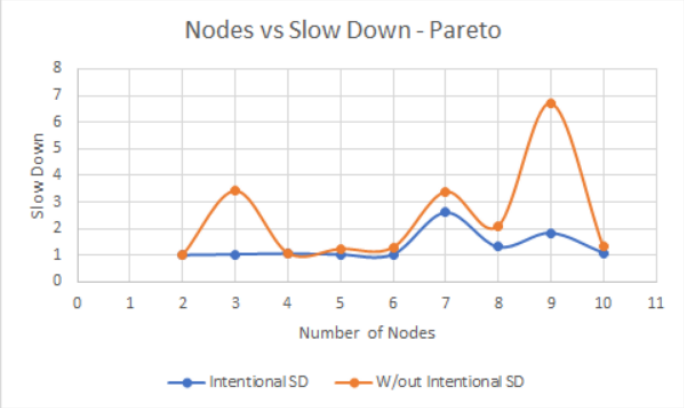
```
if(size == 1)
{
    //generate number between 1 and 10
    fs = (rand() % 10) + 1;
}
else if(size == 2)
{
    //generate number between 11 and 20
    fs = 11 + (rand() % (20-11+1));
}
else if(size == 3)
{
    //generate number between 21 and 50
    fs = 21 + (rand() % (50-21+1));
}
else if(size == 4)
{
    //generate number between 51 and 100
    fs = 51 + (rand() % (100-51+1));
}
else if(size == 5)
{
    //generate number between 101 and 500
    fs = 101 + (rand() % (500-101+1));
}
else if(size == 6)
{
    //generate number between 501 and 1000
    fs = 501 + (rand() % (1000-501+1));
}
else if(size == 7)
{
    //generate number between 1 and 1000
    fs = (rand() % 1000) + 1;
}
```

Figure 1

```
300             if(a->getFS() + a->getTS() <= b->getTS() && a!=b)
301             {
302                 totalFlow -= a->getFS();
303                 a->setFS(0);
304                 a->setWT(a->getWT() + a->getTS());
305
306             }
307             else if((a->getFS() + a->getTS() >= b->getTS()) && a!=b)
308             {
309                 if(a->getTS() > b->getTS())
310                 {
311                     continue;
312                 }
313                 a->setCW(5);
314                 a->setFS(a->getFS()-(b->getTS()-a->getTS()));
315                 totalFlow -= (b->getTS() - a->getTS());
316                 if(a->getFS() <= b->getFS())
317                 {
318                     backOff(b);
319                 }
320                 else
321                 {
322                     backOff(a);
323                 }
324             }
```

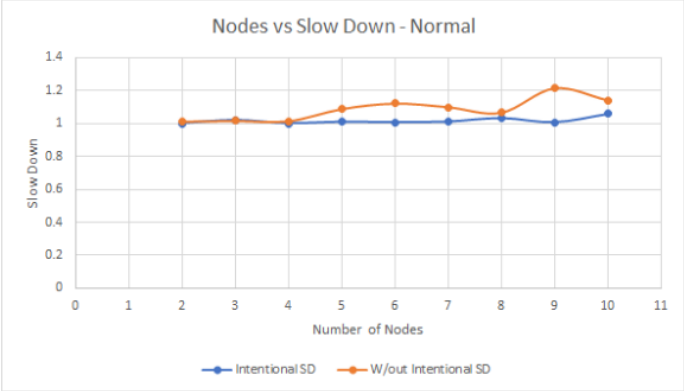
Figure 2

V. Results



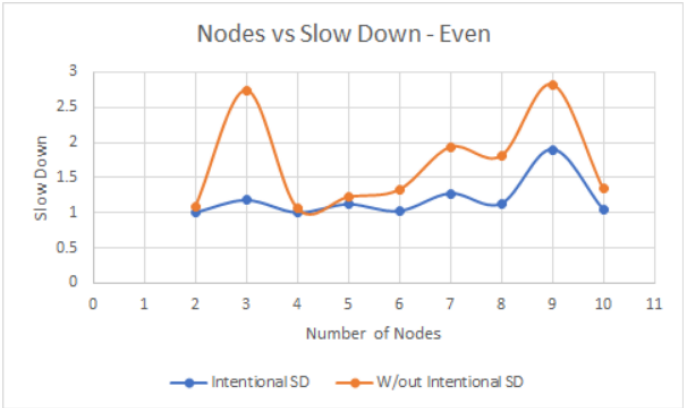
Result 1

The Pareto distribution modeled the network traffic that is seen in most IEEE 802.11 based WLANs today. The slow down in the simulation that uses intentional collisions is consistently lower than the simulation that uses the traditional MAC. The average decrease in slow down was 29.20% for WiFi networks with 2-10 nodes.



Result 2

The normal distribution showed marginal improvement of slowdown over 2-10 nodes in a WiFi network. The average decrease in slow down using intentional collisions was 3.83% with some simulations showing slightly worse with intentional collisions.



Result 3

Nodes that have an even distribution of number of packets showed improvement when intentional collisions were used. There was an average of 20.45% decrease in slow down in WiFi networks with 2-10 nodes.

VI. Conclusion

An unfair approach to the MAC of the IEEE 802.11 protocol could improve slow down in small networks by up to 30%. WiFi networks that have a large number of nodes with small flow sizes (number of packets to send) as well as those with an even spread of flow sizes stand to see the most significant improvement in slow down compared to those with a normal distribution of flow sizes, according to this simulation. Further experiments would include increasing the number of nodes that a simulated WiFi network consists of to test the effects of intentional collisions on a larger scale. Different packet number distributions as well as adjusting the range of numbers of packets assigned to nodes is another area for further experimentation. The simulator used acts on a logic level to model WiFi networks with the introduction of intentional collisions, but more intricacies of the 802.11 protocol could alter results of the experiment.

References

[1] Bianchi, G., "Performance analysis of the IEEE 802.11 distributed coordination function", *IEEE Journal on Selected Areas in Communications*, pp. 535-547, 2000.

[2] "IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1-3534, 2016

[3] Cheng. R, Chung-Ming, Huang "Collision detect and avoidance media access mechanism for next generation 802.11ax networks", *2015 11th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE)*, pp. 189-193, 2015