Pipelined Packet Scheduling in Wireless LANs

Xue Yang and Nitin H. Vaidya
Department of Electrical and Computer Engineering, and
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

Technical Report
March 2002, Revised in August 2002

Abstract— **Channel bandwidth is very scarce resource in wireless networks. However, the way in which IEEE 802.11 DCF uses random backoff time to resolve the channel contention leads to inefficient utilization of this scarce resource. Especially in highly loaded network, the portion of channel bandwidth wasted in idle or collision state is significantly high. Prior work proposes to dynamically tune the parameters of IEEE 802.11 to achieve optimized performance. In this report, we propose a different mechanism which "pipelines" the stage of contention resolution and stage of packet transmission so that the time period in which channel is in idle or collision status is reduced. The protocol performance has been studied using simulations. The promising result is that the utilization of channel bandwidth can be close to peak performance even in a highly loaded network.**

I. Introduction

WIRELESS networks provide lower channel bandwidth in comparison with wired networks. It is because of such scarce bandwidth, improving the utilization of channel resource is important for wireless networks. Given a certain physical wireless channel, the medium control access protocol will introduce some overhead, which leads to suboptimal utilization of available physical channel bandwidth. The time period ("virtual" packet transmission time), during which a station competes for the channel access and finally successfully transmit the packet, is composed of the time period during which channel is idle, the time period during which channel has collisions, and finally the time period spent for packet's successful transmission. The fraction of channel bandwidth used by successful packet transmission is known as the utilization of channel [1].

IEEE 802.11 WLAN [2] standard defines two MAC sublayer functions: the distributed coordination function (DCF) and the point coordination function (PCF). PCF is essentially a polling scheme which needs the coordination of an "Access Point" to determine which station has the right of channel access. On the other hand, DCF is a distributed random access protocol.

Throughout this report, we refer the stations which always have packets backlogged as "active stations". We also borrow the "Saturation Throughput" defined in [3] as our performance metric. "Saturation Throughput" is defined in [3] as the "maximum throughput" the system can carry while the offered load increases. A simulation study using ns2 simulator (Section II) shows that, with the number of active stations increases, the saturation throughput achieved by IEEE 802.11 DCF decreases significantly.

Some prior research work analyzes the performance of IEEE 802.11 DCF [4], [5], [6], [7], [8], [3], [9], [10]. Some improvements over IEEE 802.11 DCF backoff algorithm have been proposed in [11], [9], [7] to increase the protocol performance. In this report, an alternative way to decrease the MAC protocol overhead is proposed. Instead of adjusting backoff algorithm, the backoff stage in the proposed scheme is pipelined with packet transmission, so that the channel idle time and colliding time are minimized. Hence, the channel utilization can be significantly improved.

The rest of the report is organized as follows. Section II summarizes IEEE 802.11 DCF contention resolution procedure based on random backoff time, as well as the related research work which improves IEEE 802.11 DCF backoff algorithm. Section III briefly introduces the basic concept of "pipelined packet scheduling". Section IV discusses one possible way of "piplining" and its drawbacks. In section V, the details of proposed pipelined packet scheduling (PPS) scheme is described. Performance Evaluation is presented in Section VI. In Section VII, we discuss some observations. Finally, the conclusion is presented in section VIII.

This research is supported in part by National Science Foundation grant 01-96410

II. IEEE 802.11 Distributed Coordination Function and Related Work

Distributed Coordination Function (DCF) is the media access method used in IEEE 802.11 [2]. Since a half-duplex wireless transceiver cannot listen to channel for collisions while transmitting, the collision detection technique used in wired Ethernet cannot be applied here. Instead, IEEE 802.11 DCF is a random access protocol based on carrier sense multiple access with collision avoidance (CSMA/CA). Combined with random backoff time, CSMA/CA is designed to reduce the collision probability among multiple stations. A station wanting to access the channel needs to perform carrier sense to determine the status of the medium. If the medium is busy, the station should defer until the medium is idle for a period of time equal to DIFS (DCF Interframe Space). After this DIFS idle period, the station generates a random backoff counter, which corresponds to the number of slots this station has to wait additionally before its transmission. When medium is idle, the backoff counter is decremented by 1 after each idle slot[1]. The backoff counter will be frozen once the medium becomes busy, and will be decremented when the medium is idle again for DIFS duration. Once the backoff counter reaches zero, this station is allowed to transmit. The backoff counter is uniformly distributed over the interval [0, CW]. CW is the "Contention Window", which will take an initial value of CWMin. If more than one station happen to start their transmissions in the same slot, collision will happen. In case of a collision, the colliding packets may be retransmitted. When the number of retransmissions is less than retry limit, CW will be exponentially increased with the factor of 2 until it reaches the maximum value of CWMax. When the number of retransmissions reaches retry limit, CW will be reset to CWMin.

The carrier sense in IEEE 802.11 includes physical carrier sense and virtual carrier sense. The virtual carrier sense mechanism is implemented using a network allocation vector (NAV). RTS (request to send) and CTS (clear to send) frames contain a Duration/ID field that defines the period of time when the medium is to be reserved to transmit the actual data frame and the returning ACK frame. Through exchange of RTS/CTS frames prior to the actual exchange of data, the NAV maintains a prediction of future traffic on the medium.

IEEE 802.11 DCF defines two access methods, respectively, basic access method and RTS/CTS access method. Basic access method involves only DATA/ACK exchange while RTS/CTS access method exchanges RTS/CTS handshake frames before the actual DATA/ACK packets. The RTS/CTS exchange can perform a fast collision detection, therefore, achieves better performance in high contention network [3], [4]. However, due to the increased probability of collisions among active stations in highly loaded network, even using RTS/CTS access method, the portion of time in which channel is in collision status increases significantly and the performance of IEEE 802.11 DCF degrades abruptly.

IEEE 802.11b defines four different bit rates supported at physical layer, respectively, 1 Mbps, 2Mbps, 5.5 Mbps and 11 Mbps. Because of higher channel error probability in high channel bit rate situation, the maximum packet size at MAC layer should not be increased further. With the increase of bit rate, a packet with same packet size will cost less time to transmit. On the other hand, the slot time remains unchanged, which implies that using the same random backoff algorithm, the time period used to resolve channel contention for each packet's transmission remains unchanged. Hence, of the virtual packet transmission time, the fraction of actual time period spent in successful packet transmission is even less with the higher channel bit rate. Therefore, the performance degradation of IEEE 802.11 DCF in highly loaded networks can be seen more severely in the high bit rate situation.

We study the IEEE 802.11 DCF saturation throughput with 11 Mbps channel bit rate through simulation. The simulation results are obtained using ns2 simulator and are presented in Figure 1. The CWMin is set to 31, the number of active stations is increased from 1 to 128 and various packet sizes are used. The performance of 802.11 DCF begins to drop when there are 16 active stations. The saturation throughput with 128 active stations is 20% less than the peak point with 8 active stations when packet size is 1024 bytes. With packet size as 128 bytes, the saturation throughput with 128 active stations is 30% less than the peak point with 8 active stations.

[8] reveals the fact that the increase of the number of active stations and decrease of packet size will have significant negative impact on the performance due to an increase in the probability of collisions. Hence, Weinmiller et al. [8] propose to decrease the probability of collisions by adjusting the backoff counter to be uniformly distributed.

When there are few active stations in the network, a smaller "Contention Window(CW)" will reduce the channel idle time and enable a better usage of channel bandwidth. When the number of active stations increases, however, a larger CW value is perferred to reduce the collision probability among active stations. Clearly, CW value should be closely related to the number of active stations. [9] proposes to dynamically modify the value of CW based on the estimated number of active stations. [10] suggests a

[1] A slot is a fixed duration of time defined in IEEE 802.11

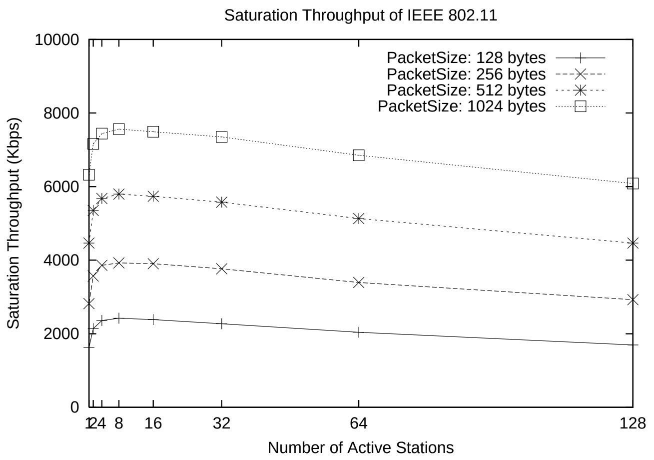

Fig. 1. Saturation Throughput of IEEE 802.11 using CWMin=31

similar idea and exploits an analytical model [11]. In [10] it is argued that the network can operate at a point close to the optimum if the time wasted on idle periods is equal to the time spent on collisions. Based on this argument, [10] has proposed a way to estimate the number of active stations via observed average number of empty slots in a "virtual" transmission time (Recall that "virtual" transmission time includes channel idle, colliding and successful transmission time for each transmitted packet). This estimated number of active stations is then applied to dynamically tune the value of CW at run time. The past work has thus shown that the performance of IEEE 802.11 can be improved through modified backoff algorithms.

HIPERLAN/1 [12] is a MAC protocol using two stage contention resolution. It adopted an "Elimination-Yield Non-preemptive Priority Multiple Access"(EY-NPMA) to resolve channel contention. The contention phase consists of two stages: the "elimination" stage and the "yield" stage. In the elimination stage of HIPERLAN/1, a contending station transmits black bursts[2] for a random duration and then listens to the channel in the *elimination survival verification interval*. A contending station survives the elimination stage if and only if the channel is sensed idle in its *elimination survival verification interval*; otherwise, this station is eliminated and withdraws from the competition for the right of transmission in the current channel access cycle. The yield stage then further resolves contention and reduces the number of stations allowed to transmit to 1 with high probability.

Different from prior research work, in this report, we consider a machanism overlapping the contention resolution stage with packet transmission stage. Specifically, these two stages are "pipelined".

[2]Black bursts are pulses of energy.

III. Basic concept of pipelined packet scheduling

In IEEE 802.11 DCF and the related research work described in previous section, all stations begin their contention resolution stage (i.e., random backoff stage) after the channel becomes idle. Therefore, the chosen range of random backoff period is critical to the performance of the MAC protocol. If the range is too large, much bandwidth will be wasted in idle state. If the range is too small, collisions are likely to happen and waste the bandwidth as well.

However, if contention resolution stage can overlap with packet transmission stage, then we can possibly avoid the dilemma above. Specifically, in this way, the contention resolution stage does not really cost channel bandwidth to resolve contention. A suboptimal choice of random backoff interval can be tolerated to some extent. Ideally, while some station is transmitting packets, all other backlogged stations will decide which station is going to transmit next. Then after current transmission, the winning station will successfully access the channel. Just like the traditional benefit of "pipelining", the cycle for each packet's transmission has been reduced. Therefore, the network throughput can be improved.

Figure 2 gives a conceptual demonstration of "Pipelined Packet Sceduling".

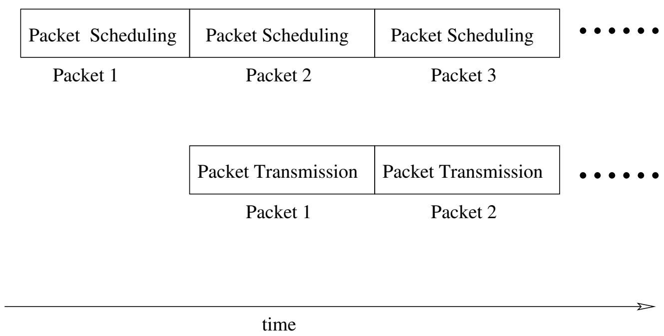

Fig. 2. Pipelined packet scheduling

IV. Pipelined scheduling with two channels

Our initial idea about pipelined packet sheduling is motivated by using two channels. Specifically, one of the two channels (control channel) can be used for RTS/CTS handshake, the other channel (data channel) used for DATA/ACK exchange. While the current data packet is transmitting on the data channel, the RTS/CTS handshake for the next packet can proceed on the control channel. Ideally, if the channel contention resolution and RTS/CTS handshake can always be finished within one data packet transmission duration, then we can expect the full utilization of data channel.

Bharghavan [13] has also considered the possibility of pipelining RTS/CTS. However, no results are available.

Our analysis and simulation results (not reported here) show that this scheme fails to work well because of following reasons:

• The two channels have to share the limited channel bandwidth. Given the total channel bandwidth of W, we represent bandwidth share for control and data channels respectively as W_c and W_d. One reason for using the RTS/CTS access method of IEEE 802.11 rather than the basic access method is that RTS/CTS have much shorter packet transmission duration than data packets. Consequently, the collisions can be detected faster using RTS/CTS access method. However, since now we divide the channel bandwidth into two parts (W_c and W_d), the smaller W_c is, the longer it will take RTS/CTS to transmit. The benefits of using RTS/CTS may no longer hold when W_c is not large enough. On the other hand, we do not want W_c to be too large since it is W_d that is used for actual data transmission.

Let T represent one packet transmission duration using IEEE 802.11 DCF standard RTS/CTS access method.

$$T = T_{idle} + T_{collision}(W) + T_{data/ack}(W) + T_{rts/cts}(W) \quad (1)$$

where T_{idle} is the channel idle time for resolving contention, $T_{collision}$ is the time duration in which collisions happen, $T_{data/ack}$ represents the duration of successful DATA/ACK exchange, and $T_{rts/cts}$ represents the duration of successful RTS/CTS handshake.

Let T_d be the duration of DATA/ACK exchange using the two channel scheme, then we have:

$$Td = Td_{data/ack}(W_d) \quad (2)$$

where $Td_{data/ack}$ is the transmission duration of DATA/ACK on the data channel. Its length will increase with the data packet size increases or W_d decreases.

The duration (T_c), in which the channel contention resolution and RTS/CTS handshake is completed on control channel, consists of:

$$Tc = Tc_{idle} + Tc_{collision}(W_c) + Tc_{rts/cts}(W_c) \quad (3)$$

where Tc_{idle} is the control channel idle time for resolving contention. $Tc_{collision}$ is the time duration in which collisions happen on the control channel. $Tc_{rts/cts}$ represents the duration of succesful RTS/CTS handshake. Both $Tc_{collision}$ and $Tc_{rts/cts}$ will increase with the decrease of W_c.

To achieve the desired full utilization of data channel bandwidth and have better performance than IEEE 802.11, the following equation needs to be statisfied by two channel scheme:

$$T > Td \geq Tc \quad (4)$$

From (1), (2), (3) and (4), we can see that the appropriate ratio of $\frac{W_c}{W_d}$ depends on the data packet size and channel contention degree. In other words, different distribution of packet size and different network size will require different bandwidth division among control channel and data channel, which is unfeasible practically.

• The motivation for pipelined packet scheduling is to save channel idle time. However, by using the two channel scheme, if the RTS/CTS handshake cannot be finished within one data packet transmission duration, then data channel idle time cannot be avoided. Furthermore, we have the extra penality from dividing channel bandwidth. Further analysis and simulation results (not presented in this report) show that without appropriate bandwidth division among control channel and data channel, the gain of this scheme is marginal or it even performs worse than IEEE 802.11 DCF in some cases.

So, instead of above two channel scheme, we propose to use one busy tone channel combined with data channel to achieve pipelined packet scheduling. As a narrow band signal, busy tone does not need much channel bandwidth. However, busy tone can only partially resolve channel contention since collisions cannot be detected. Therefore, our proposed scheme actually includes two contention resolution stages, one is on the busy tone channel, another is on the data channel. The details are presented in the next section.

V. Pipelined Packet Scheduling with busy tone (PPS)

Pipelined packet scheduling (PPS) is currently designed for wireless LANs or infrastructure wireless networks. In a wireless LAN, all stations are within each other's transmission range, and in an infrastructure based network, each mobile host only exchanges packets with base station.

In addition to the data channel, PPS scheme requires a busy tone channel. In this protocol, only stations which are not transmitting or receiving on the data channel need to transmit or receive busy tone signal. Therefore, it does not require the stations to have the capability of transmitting while receiving, or transmitting to more than one channel at the same time. With the help of busy tone signal, the contention resolution stage can be partitially "piplined" with packet transmission stage. Consequently, the contention among competing stations is resolved in two

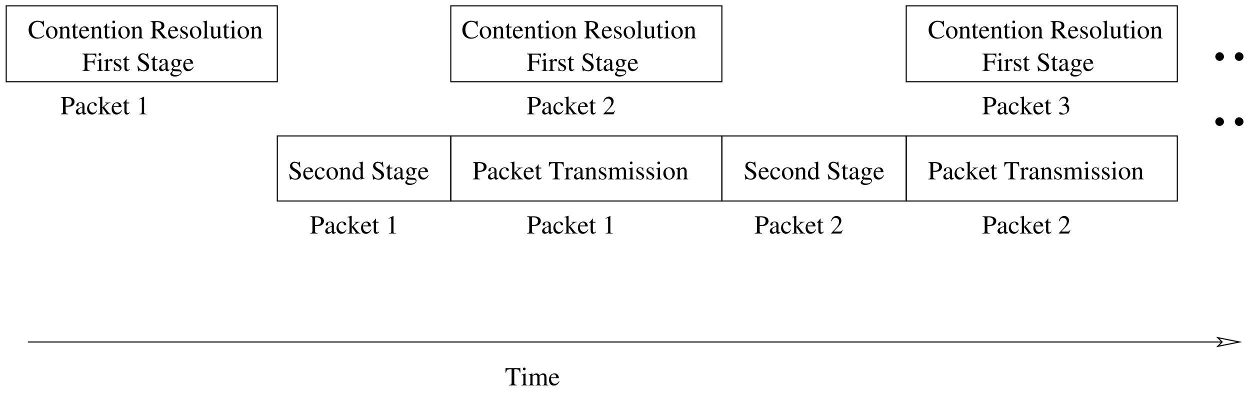

Fig. 3. Pipelined packet scheduling with busy tone

stages. The first stage is performed by backlogged stations while some other station is transmitting on the data channel. Once a station finishes the first stage, it will send a busy tone signal to claim that it has won the channel access of next packet. Other stations will be aware of the winner's presence through sensed busy tone. Only the stations who win the first stage can proceed to the second stage. Like IEEE 802.11, the second stage of contention resolution will start only after data channel becomes idle. However, through the filtering of first stage, only small number of stations will be able to start their second stage each time. This fact will allow us to decrease the collission probability. In addition, a small range of random backoff interval will be sufficient to resolve contention among stations in the second stage. The benefits are two-fold:

1. The first stage is parallel in time to the transmission of other packet. It saves channel idle time.
2. Only the stations which claim to win in the first stage will enter the second stage. Hence, the collision probability in the second stage is significantly reduced.

By "pipelining", we mean that contention resolution procedure proceeds while data channel is busy, Figure 3 illustrate PPS scheme concetually.

The major difference between PPS and IEEE 802.11 DCF is the contention resolution algorithm. Unless stated explicitly, other parts of 802.11 remain unchanged in PPS.

Both the first and second stages of contention resolution in PPS adopt the random backoff strategy of 802.11 DCF. Let BC_1 represent the backoff counter of first stage, while BC_2 represents that of second stage.

A backlogged station will start the contention resolution from the first stage using BC_1. BC_1 is randomly chosen from the range of $[0, CW_1]$ unless it already has a non-zero value, where CW_1 is the contention window for the first stage. If this station does not receive any busy tone signal and its NAV expires (recall that NAV is used for virtual carrier sense to indicate the data channel status), it will decrease BC_1 by 1 after each slot when data channel is *idle*. Once BC_1 reaches zero, this station will begin its data channel access. In this case, PPS behaves same as 802.11.

One the other hand, if the NAV setting indicates that some other station is transmitting on the data channel, PPS begins to behave differently from 802.11. This station will start pipelining and BC_1 is decreased by 1 after each slot (even when data channel is busy). The stations whose BC_1 reaches zero first win the first stage and will send a busy tone signal on the busy tone channel. All other stations which sense this busy tone will freeze their BC_1. These stations are said to be frozen in the first stage.

Once the data channel becomes idle, the winner of the first stage will begin their second stage and BC_2 is generated randomly from the range of $[0, CW_2]$, where CW_2 is the contention window for the second stage. BC_2 is decreased after each slot and the first station which counts down its BC_2 to zero will access the channel.

Other stations which have their BC_1 frozen will stay in the first stage and keep BC_1 frozen until an RTS or CTS is overheard. On overhearing an RTS or CTS, a frozen station in the first stage will know whether it is the intended receiver of current transmission. As a result, a station that is not the intended recipient will unfreeze and resume to count down BC_1 while data channel is busy (pipelining). The procedure described above is then repeated. An unfreezing timer could be used to force a maximum duration in which stations may stay in the frozen status of the first stage in order to avoid deadlock when transmission errors happen.

A station which just finished a successful transmission will return back to the first stage.

Collisions in the first stage will result in more than one station entering the second stage. The actual channel collisions happen in the second stage. Naturally, a collision in the second stage implies a collision in the first stage.

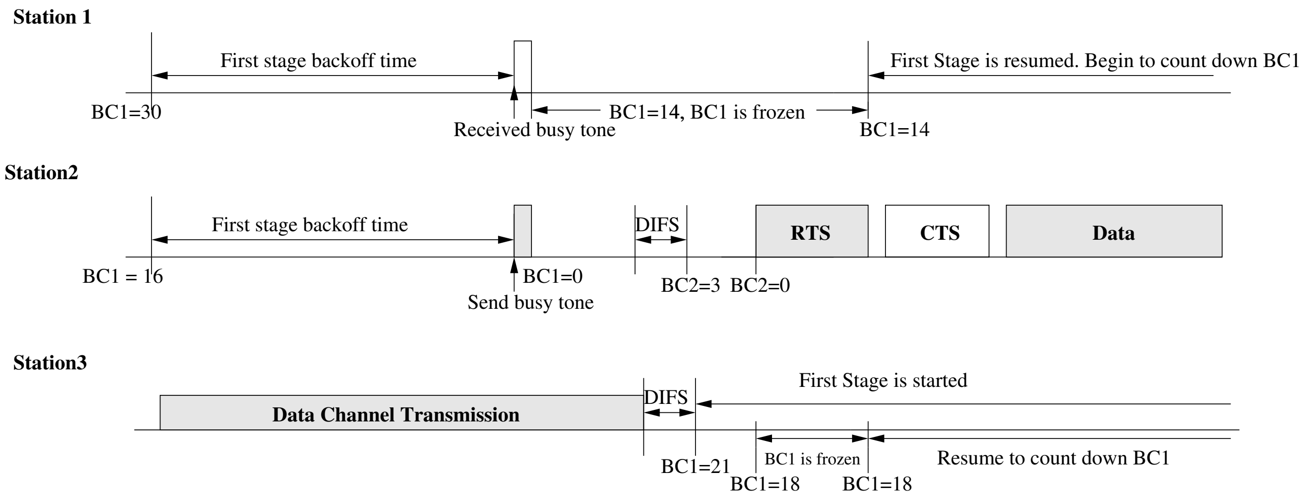

Fig. 4. Pipelined Packet Scheduling

Therefore, whenever a collision happens in the second stage, both CW_1 and CW_2 are exponentially increased by a factor of 2 until they reach their respectively maximum values. The colliding stations will remain in the second stage and continue to contend for the channel using the backoff counter BC_2, where BC_2 is regenerated from the interval $[0, CW2]$.

The value of CW_1 is in the range of $[CW1_{min}, CW1_{max}]$, and the value of CW_2 is in the range of $[CW2_{min}, CW2_{max}]$. After a successful transmission, CW_1 is reset to $CW1_{min}$ so that it can adapt to large variation of network size. However, the CW_2 is decreased more slowly and just divided by 2. The idea behind this is that, through adjusting of the first stage and under a certain network load, the number of stations entering into the second stage should remain relatively stable. Therefore, the slow decrement of CW_2 will tend to have better performance. In the simulations, we set $CW1_{min}$ as 31 and $CW1_{max}$ as 1023, while $CW2_{min}$ is set to 3 and $CW2_{max}$ is set to 255. In the next section, we will show through simulation results that this set of parameters is enough to maintain high performance of MAC protocol even in networks with 128 active stations.

If a station in the first stage has not counted down its BC_1 to zero when its NAV expires, and it has not received any busy tone signal by then, this station will keep on counting down BC_1 when the data channel is idle. Once BC_1 reaches zero, this station will access the channel. This will handle the situation in which all stations happen to choose a large backoff interval and no station has finished the first stage during current packet's transmission. If this is the case, we still save some channel idle time by using "Pipelining".

Initially, when a station desires to transmit and channel is idle, this station will start its random backoff stage using BC_1. If the channel remains idle until BC_1 reaches zero, this station will begin its data channel access. Therefore, in the network with only one flow, PPS will behave exactly same as IEEE 802.11. However, when there are two backlogged stations, say A and B, both of them begin to count down BC_1 initially. In the case that A wins, A will begin to transmit on data channel. In IEEE 802.11 DCF, B will freeze BC_1 until A finishes its transmission. Using PPS, once B knows that it is not the receiver of A's packet, B will continue to count down BC_1 while A is transmitting.

A more specific example of PPS with three backlogged stations is given in Figure 4. Initially, station 3 transmits on data channel and stations 1 and 2 are in the first stage of contention resolution. When BC_1 of station 2 reaches zero, it sends a busy signal out. On receiving this busy tone signal, station 1 freezes its BC_1 at value of 14. After station 3 finishes its transmission, and the channel has been idle for "DIFS" duration, station 2 is then in its second stage of contention resolution. A value for BC_2 , say 3, is randomly chosen from the range of [0, 3]. Assuming that station 3 is continuously backlogged, it would then return back to the first stage and start to count down BC_1 since it did not receive any busy tone signal and channel now is idle. In this example, the BC_1 of station 3 is set to 21, which is randomly chosen over the interval [0, 31]. When station 2 counts down its BC_2 to zero in the second stage and transmits RTS frame, station 3 freezes its BC_1 since it senses the channel becomes busy. Once the RTS frame from station 2 has been received by stations 1 and 3, both of them would then resume their first stage of contention resolution and the BC_1 continues to be counted down. In this particular example, the channel idle time used by PPS for resolving contention is three slots, but it would be 16 slots if using IEEE 802.11 DCF instead.

VI. Performance Evaluation

In this section, we study the performance of "Pipelined Packet Scheduling" (PPS) using simulations. All the simulation results are based on a modified version of ns-2 network simulator from USC/ISI/LBNL [14], with wireless extensions from the CMU Monarch project [15]. The physical channel bit rate is set to 11 Mbps for IEEE 802.11 DCF. Since proposed PPS scheme requires a busy tone channel, for which we assign 2% channel bandwidth, the resulting data channel bit rate we are using in simulation is 10.78 Mbps. According to [16], the time period of 5 μs is sufficient for the busy tone signal to be detected. Therefore, in our simulations, the transmission of each busy tone signal lasts for 5 μs. According to IEEE 802.11 standard, the physical layer preamble and header (totally 192 bits) should be transmitted using 1 Mbps bit rate. However, the simulations in this report are conducted by using ns-2 version 2.1b8a. In this version of ns-2, the physical layer preambler and header are also transmitted at highest channel bit rate (i.e., 11 Mbps for 802.11, 10.78 Mbps for PPS). Therefore, the presented throughput results from simulations in this report could be higher for both 802.11 and PPS than that in real systems.

We are interested in studying the channel utilization by using "Saturation Throughput". Recall that "Saturation Throughput" is defined as the maximum throughput the system can carry while the offered load increases. In our simulations, each flow's source station has very aggressive sending rate and is always backlogged. The traffic sources are chosen to be constant bit rate (CBR) sources with a fixed packet size. We vary the packet size in different simulations.

A. Saturation Throughput

The number of active stations in our simulated scenario is varied from 1 to 128, and the packet size is varied from 128 bytes to 1024 bytes. As we mentioned in section II, due to the increased probability of collisions, the performance of IEEE 802.11 DCF drops abruptly with the increase of active stations. However, in proposed PPS scheme, after the first stage of contention resolution, it is expected to be only a few stations which actually contend for the data channel in the second backoff stage. Therefore, the collision probability will be largely reduced. Specifically, under the ideal condition of busy tone channel (i.e., busy tone signal can always be sensed correctly), only stations which count down BC_1 to 0 simultaneously will enter the second stage at the same time. Since the number of active stations in the second stage is small, through a small random backoff duration, we can effectively resolve the channel contention among them. Hence, we expect the performance of PPS to be less sensitive to the number of active stations.

The simulation results presented in Figure 5, 6, 7 and 8 affirm our expectation. These four figures with different packet sizes (128, 256, 512, 1024 bytes) demonstrate the same point: PPS scheme reaches the peak performance when there are 4 active stations since they can interleave their transmissions perfectly. After that, the saturation throughput degrades slightly, however it remains around the peak point. IEEE 802.11 achieves its peak performance when there are 8 active stations, at this point, the performance gap between PPS and IEEE 802.11 is minimum. Beyond this point, while the performance of IEEE 802.11 degrades with the increase of active stations, the performance of PPS is less sensitive to the network size. Therefore, the performance gap between 802.11 and PPS gets larger when the number of active stations increases. Specifically, with 128 active stations, PPS achieves 57% more saturation throughput than 802.11 when packet size is 128 bytes; and when packet size is 1024 bytes, PPS achieves 29% more saturation throughput than 802.11.

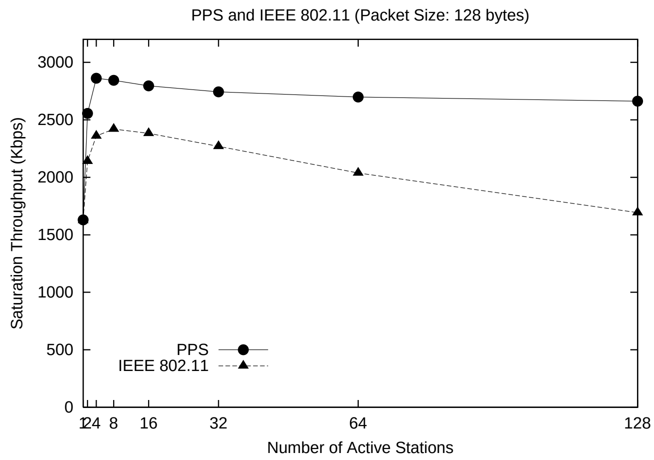

Fig. 5. Saturation Throughput of PPS vs. IEEE 802.11 DCF (PacketSize:128)

B. Performance impact of imperfect reception of busy tone

Since busy tone channel has very narrow channel bandwidth, it is more vulnerable to the fading compared to data channel. In this section, we study the performance impact of such cases by varying the probability of detecting busy tone. The probability of busy tone detection is varied from 100% to 50% and packet size is set to 512 bytes. The obtained simulation results are presented in Figure 9.

When a station does not receive a busy tone sent by other stations, it will continue with its first backoff stage

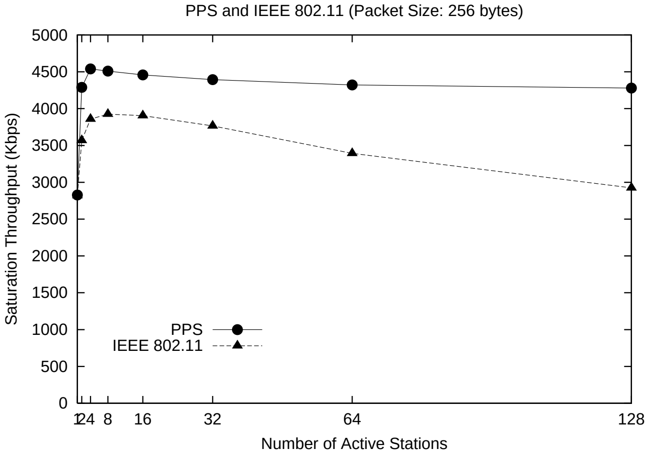

Fig. 6. Saturation Throughput of PPS vs. IEEE 802.11 DCF (PacketSize:256)

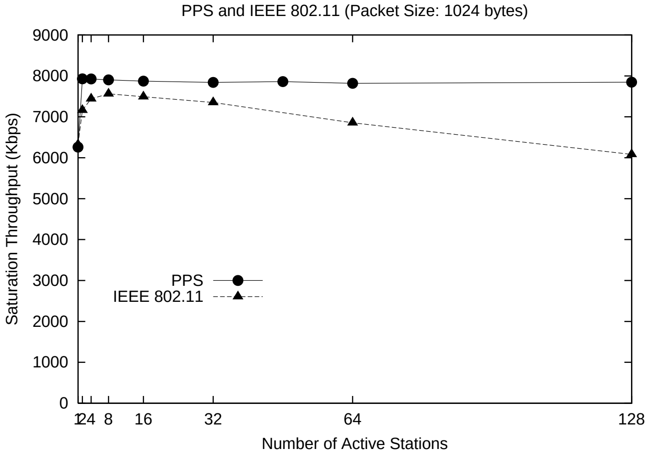

Fig. 8. Saturation Throughput of PPS vs. IEEE 802.11 DCF (PacketSize:1024)

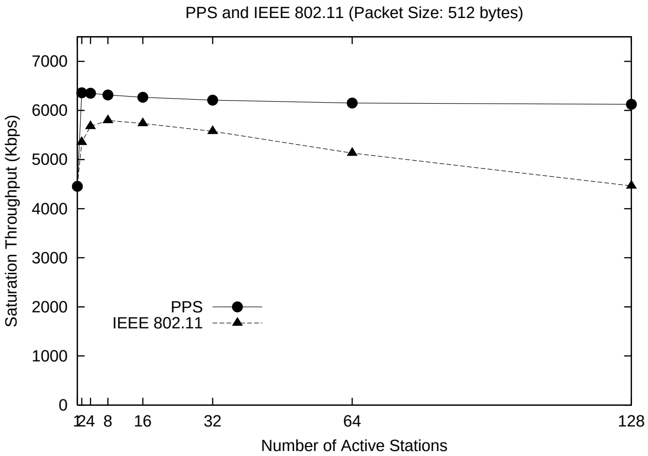

Fig. 7. Saturation Throughput of PPS vs. IEEE 802.11 DCF (PacketSize:512)

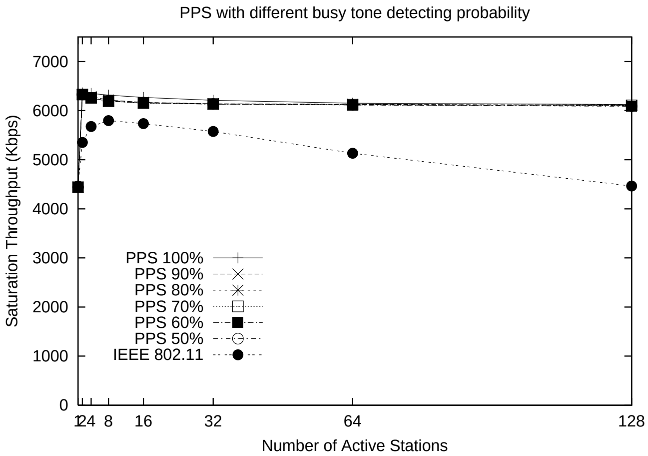

Fig. 9. Performance of PPS with various probability of busy tone detection(PacketSize:512)

until its backoff counter BC_1 eventually reaches zero. Then it will believe it has won the first backoff stage and send another busy tone. The direct result is that more stations will regard themselves as the winners of the first stage and enter into the second backoff stage. Then we will have more contending stations in the second stage compared to the case when busy tone can be perfectly received. However, this number does not increase significantly. Moreover, the second backoff stage has contention window (CW_2) in the range from 3 to 255. After each collision, CW_2 is doubled and it is only divided by half after each successful transmission. Slightly more stations in the second stage will not have much impact on the performance. This explains why the PPS protocol performance changes little when the probability of busy tone detection is decreased from 100% and 50%.

VII. Potential improvements over PPS

A. *Unfairness for common receiver*

In the proposed PPS scheme, the channel access of next packet will be partially decided during current packet's transmission. Consider a wireless LAN with 4 stations in which we have three flows. Both stations 1 and 2 send packets to station 3 while station 3 sends packets to station 4. Suppose station 1 is transmitting to station 3 which has a frozen backoff counter BC_1 of 17. During current transmission, station 2 will win the first backoff stage. Once station 1 finishes current transmission to station 3, station 2 will obtain the channel access after the backoff duration of second stage, say 2 slots, and begin to transmit to station 3. Station 3 now has frozen backoff counter BC_1 of 15. Similarly in the next time, station 1 will obtain the channel access and transmit to station 3 again. Continuing in such way, even though eventually station 3's backoff counter BC_1 will reach zero and obtain channel access, it has less chances of transmitting compared to stations 1 and 2. On average, station 3 can transmit one packet after stations 1 and 2 each transmit 3-4 packets. From the example we given here, it can be seen clearly that this kind of unfairness issue only appears when the number of active stations is small. When the number of active stations is large, even if most stations send to the same receiver, this common receiver can still get relatively fair chances to transmit. One possible solution for the potential unfairness issue is that, if a receiver receives two packets from the same source station during the time period it has packet backlogged, this receiver will directly enter into the second backoff stage after receiving current packet.

B. *Pipelined Packet Scheduling without Busy Tone*

There is an interesting observation for the simulation results in Section VI. With the decreased probability of busy tone detection, the performance of proposed PPS scheme does not change much. As we mentioned before, this attributes to the fact that there are less contending stations in the second stage compared to the total number of competing stations in the entire network.

From this observation, we expect the idea of "pipelining" can be applied to improve the packet sheduling even without using busy tone signal. The result for PPS with 0% probability of detecting busy tone are presented in Figure 10.

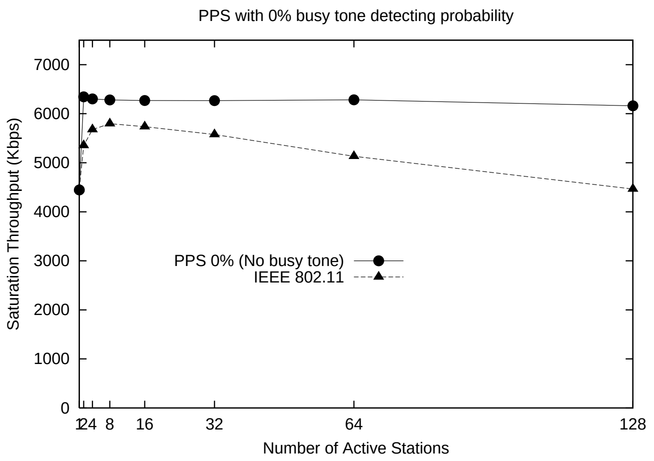

Fig. 10. Pipelining with 0% probability of detecting busy tone (Packet Size: 512 bytes)

Here we give some intuitive explanations. A further study has been conducted and a new MAC protocol - "DSCR" - is proposed in [17].

Let us observe the performance of IEEE 802.11 DCF presented in Figure 1 again. Notice when the number of active stations is between 4 to 16, IEEE 802.11 stays around its peak performance. Only when this number increases further, the performance of IEEE 802.11 degrades abruptly. Therefore, if we can effectively control the actual number of contending stations to be within some optimum range, the protocol's performance should remain around peak performance and be less sensitive to the increased number of active stations. To achieve this objective, our scheme without busy tone aggressively increases active stations' backoff counter in the first stage such that they are sparsely distributed. Hence, the number of stations can finish the first stage before current transmission completes will be relatively small and stable. Only these stations actually contend for the channel in the second stage. Therefore, the performance may remain stable over a large variation in network size. This scheme can also be further extended to ad hoc networks since no busy tone is needed.

VIII. Conclusion

The channel utilization of IEEE 802.11 DCF is low in a highly loaded network. The inefficiency is primarily caused by the random backoff stage of IEEE 802.11 DCF, which cannot balance the fraction of channel idle time and channel collision time quite well during contention resolution procedure. In this report, we proposed a "Pipelined Packet Scheduling (PPS)" scheme in which the contention resolution procedure consists of two random backoff stages. The first stage (using busy tone) is parallel in time to current packet's transmission. Only winners of the first stage can enter the second stage and possibly win the channel access. The main benefits are, the channel idle time has been reduced because of "pipelining", and the probability of channel collision is decreased after contention resolution of the first stage. Therefore, the PPS scheme avoids the dilemma of choosing an optimum random backoff algorithm for IEEE 802.11 DCF dynamically to balance channel idle time and channel collision time [10]. Instead, both the channel idle time and colliding time have been minimized as the direct results of "Pipelining". The simulation results demonstrates the effectiveness of the proposed PPS scheme, and also suggest an alternative mechanism without using busy tone [17].

References

[1] Leonard Kleinrock and Fouad A. Tobagi, "Packet Switching in Radio Channels: Part I- Carrier Sense Multiple-Access Modes and Their Throughput-Delay Charateristics," *IEEE Transactions on Communications*, vol. COM-23, no. 12, December 1975.

[2] IEEE, "Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) Specifications," *IEEE Standard 802.11*, June 1999.

[3] G. Bianchi, "Performance Analysis of the IEEE 802.11 Dis-

tributed Coordinated Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, March 2000.

[4] H. S. Chhaya and S. Gupta, "Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol," *Wireless Networks*, pp. 217–234, 1997.

[5] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Communication Magazine*, pp. 116–126, Sep 1997.

[6] Y. C. Tay and K. C. Chuan, "A Capacity Analysis for the IEEE 802.11 MAC Protocol," *ACM/Baltzer Wireless Networks*, , no. 2, pp. 159–171, March 2001.

[7] J. Weinmiller, H. Woesner, J. P. Ebert, and A. Wolisz, "Analyzing and tuning the distributed coordination function in the IEEE 802.11 DFWMAC draft standard," *Proc. MASCOT, San Jose, CA*, Feb 1996.

[8] J. Weinmiller, M. Schlager, A. Festag, and A. Wolisz, "Performance study of access control in wireless LANs IEEE 802.11 DFWMAC and ETSI RES 10 HIPERLAN," *Mobile Networks and Applications*, pp. 55–67, 1997.

[9] G. Bianchi, L. Fratta, and M. Oliveri, "Performance Evaluation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LANs," *Proceedings of PIMRC, Taipei, Taiwan*, pp. 392–396, Oct 1996.

[10] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 9, pp. 1774–1786, September 2000.

[11] F. Cali, M. Conti, and E. Gregori, "Dynamic tunning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Trans. on Networking*, vol. 8, no. 6, pp. 785–799, December 2000.

[12] ETSI TC-RES, "Radio Equipment and Systems(RES); HIgh PErformance Radio Local Area Network(HIPERLAN) Type 1; Functional specification," *European Telecommunication Standard ETS 300 652*, Oct 1996.

[13] Vaduvur Bharghavan, personal communication, 2002.

[14] VINT Group, "UCB/LBNL/VINT network simulator ns (version 2)," *http://www.isi.edu/nsnam/ns*.

[15] The CMU Monarch Project, "Wireless and mobility extension to ns," *http://www.monarch.cs.cmu.edu*.

[16] Fouad A. Tobagi and Leonard Kleinrock, "Packet Switching in Radio Channels: Part II- The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution," *IEEE Transactions on Communications*, vol. COM-23, no. 12, December 1975.

[17] Xue Yang and Nitin H. Vaidya, "DSCR: A More Stable MAC Protocol for Wireless Networks," *Technical Report, University of Illinois at Urbana-Champaign*, August 2002.