Explicit and Implicit Pipelining for Wireless Medium Access Control

(Invited Paper)

Xue Yang and Nitin H. Vaidya
Department of Electrical and Computer Engineering, and
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Email: xueyang, nhv@uiuc.edu

***Abstract*— In wireless networks, multiple stations contend for access to the shared channel. In the cases of collisions, the associated collision cost is much higher than wired networks since stations cannot detect the collision without explicit feedback from the receiver. For this reason, more efficient contention resolution algorithms are desired for wireless networks to reduce the collision probability among backlogged stations. With distributed multiple access control, each station usually goes through a contention resolution stage before initiating its transmission. As contention resolution stage consumes channel bandwidth without producing any goodput, ideally, we desire it to be as short as possible while reducing the possibility of collisions to as small as possible. However, in general, it is difficult to achieve an optimum tradeoff between these two desired features. In this paper, we propose to use pipelining techniques to resolve such conflicts and improve the performance of multiple access control in terms of channel utilization. We discuss several pipelining MAC schemes and present their advantages and disadvantages accordingly.**

I. INTRODUCTION

In wireless networks, a station usually can only learn about a collision when the transmission is finished and the expected acknowledgment (in some form) does not come back. Consequently, a collision will last for the entire packet transmission duration instead of propagation delay only, resulting in higher collision cost in wireless networks compared with wired networks. Moreover, wireless networks deliver much lower bandwidth than wired networks, which makes it hard for wireless networks to afford significant loss of channel resource. For the above reasons, more efficient contention resolution algorithms for multiple access control are desired in order to improve channel utilization of wireless networks.

Standards for wireless MAC protocol include the European Telecommunications Standards Institute (ETSI) High Performance European Radio LAN (HIPERLAN/1) [1] and the IEEE 802.11 WLAN [2]. HIPERLAN/1 has two contention resolution stages in series: the "elimination" stage followed by the "yield" stage. In the elimination stage, a contending station transmits bursts (i.e., pulses of energy) 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. The objective of the "elimination" stage is to eliminate as many contending stations as possible. "Yield" stage follows "elimination" stage to further resolves contentions among surviving stations. By selecting a set of fix parameters, e.g., the maximum number of elimination slots and the probability of bursting in an elimination slot, HIPERLAN/1 compromises its peak performance to its stability over a certain range of network sizes.

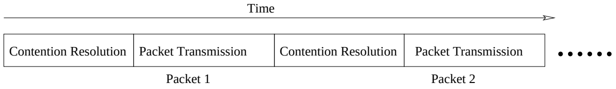

Fig. 1. Serial Packet Scheduling

IEEE 802.11 standard [2] defines a distributed coordination function named DCF, which uses *binary exponential backoff* (BEB) algorithm to resolve channel contention. In DCF, a station wanting to access the channel generates a random backoff counter uniformly distributed over the interval [0, CW] (CW represents the contention window). This backoff counter corresponds to the number of *idle slots* this station has to wait before its transmission. The contention window, CW, is exponentially increased by a factor of 2 each time a collision happens, until it reaches the maximum value denoted by CW_{max}. Once a packet is successfully transmitted by a station, CW at that station is reset to the minimum value CW_{min}. Clearly, the choice of contention window is critical to the performance of 802.11. The collision probability is likely to be small if using large values for CW. On the other hand, unnecessarily large CW will reduce channel utilization.

Notice that in both IEEE 802.11 DCF and HIPERLAN/1, mobile stations go through contention resolution stage and packet transmission stage sequentially, as shown in Figure 1. Since contention resolution stage consumes channel bandwidth without producing any goodput, it is desired to reduce the channel time spent on contention resolution while maintaining a small probability of collision, which is difficult to achieve in general.

Observe that in the above sequential procedure, when a pair of source and destination stations are exchanging packets, stations in the neighborhood remain idle and do nothing but wait. A new round of contention resolution is not started until current transmission finishes. On the other hand, if the cost associated with contention resolution can be hidden (or

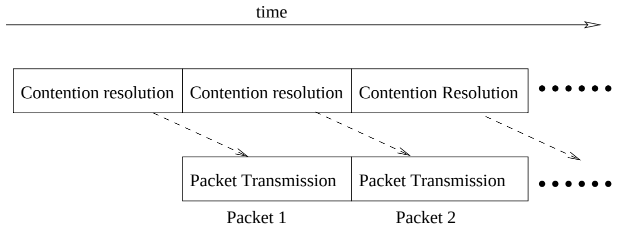

Fig. 2. Pipelined Packet Scheduling

partially hidden) by overlapping channel contention resolution for next packet with current packet transmission, then the tradeoff between channel cost for contention resolution and the collision probability will no longer be a limiting factor for the MAC protocol design. Figure 2 illustrates the basic idea of pipelining the contention resolution stage with the packet transmission stage.

While the basic idea of pipelining can be adapted to different kinds of multiple access control protocols, in this paper, we focus our discussion on the application of pipelining to IEEE 802.11, for the purpose of demonstration.

Some prior research work, e.g., [3], uses one common channel to schedule packets for multiple data channels. The fundamental difference from the pipelining schemes proposed in this paper is that, in the prior schemes, the exchange of control messages to decide which station will transmit on an available data channel occur when at least one of the channels is perceived as IDLE. Contrary to this, the contention resolution of the pipelined schemes proceeds for packets to be transmitted in the future when the channel is currently BUSY.

The rest of this paper is organized as follows. Three pipelining schemes: “Total Pipelining”, “Partial Pipelining” and “Implicit Pipelining” are discussed respectively in Section II, III and IV. The conclusion is presented in Section V.

II. TOTAL PIPELINING WITH TWO CHANNELS

IEEE 802.11 DCF defines a RTS/CTS access method, in which RTS (Request To Send) and CTS (Clear To Send) handshake frames are exchanged before Data/ACK packets, in order to avoid possible collision of data packets and to enable a fast collision detection [1].

One obvious way of pipelining for 802.11 is to divide the channel into two sub-channels: a control channel and a data channel. Control channel is used for random backoff and RTS/CTS handshake (stage 1), and data channel is used for DATA/ACK exchange (stage 2). While current packet is transmitting on the data channel, the contention resolution and RTS/CTS handshake for the next packet can proceed on the control channel, as we illustrate in Figure 3. Ideally, if contention resolution and RTS/CTS handshake can always be finished within one data packet transmission duration, then the full utilization of data channel can be expected. This scheme is named “Total Pipelining” since it attempts to completely resolve channel contention during pipelined stage 1.

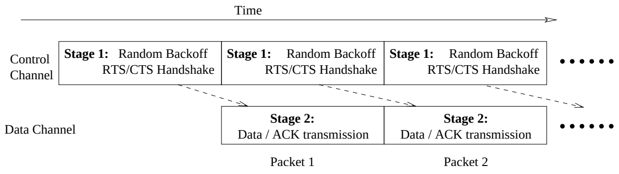

Fig. 3. Total Pipelining Scheme

A basic principle of pipelining is that the pipelined stages must be balanced, otherwise, the efficiency of pipelining will be significantly reduced [4]. We argue below that it is hard for the “Total Pipelining” scheme to achieve balanced pipelining stages in dynamic environments.

Assume the total channel bandwidth is fixed, say W, and the bandwidth for control and data channels is W_c and W_d respectively, where $W_c + W_d = W$.

For “Total Pipelining” scheme, the length of stage 1, T_1, includes the random backoff duration for contention resolution and RTS/CTS exchange duration. T_1 is determined by the following factors:

1) The number of contending stations. It determines how much time will be spent on contention resolution.
2) The control channel bandwidth W_c. It determines how much time will be spent on RTS/CTS transmissions, which in turn, determines how fast the collision can be detected in cases that collisions happen.

On the other hand, the length of stage 2, T_2, is the duration of DATA/ACK exchange on the data channel and is determined by the following two factors:

1) Data packet size;
2) The data channel bandwidth W_d.

Let T represent the average time period between two successful packet transmissions using 802.11 DCF RTS/CTS access method *without pipelining*. To achieve the desired full utilization of data channel bandwidth and have better performance than 802.11, the following constraint needs to be satisfied by the “Total Pipelining” scheme:

$$T > T2 \geq T1 \tag{1}$$

However, to satisfy constraint (1), the required ratio of $\frac{W_c}{W_d}$ changes with the data packet size and channel contention degree. In other words, different distribution of traffic payload sizes and different network sizes will require different bandwidth divisions among control channel and data channel, which limits its use in practice.

III. PARTIAL PIPELINING WITH BUSY TONE

A. *Motivation*

As we see in Section II, the difficulty of applying “Total Pipelining” is that the optimum bandwidth ratio between two channels changes with the data packet size and channel contention degree. To overcome this issue, instead of resolving channel contention completely in the pipelined stage 1, in “Partial Pipelining” scheme, contention resolution procedure

---

[1] Since RTS/CTS frames are usually much shorter than data packets.

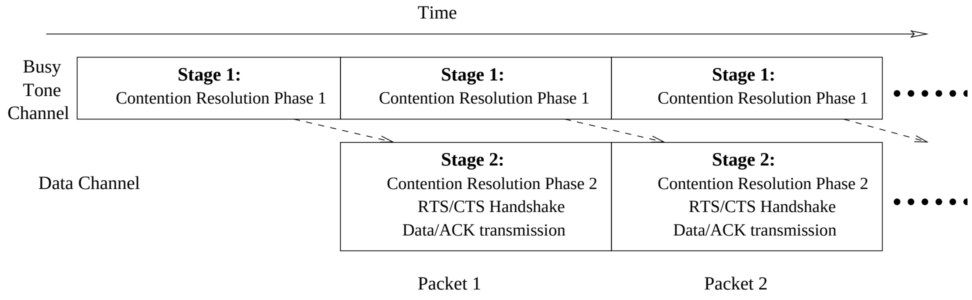

Fig. 4. Partial Pipelining Scheme

is split into two phases. Only contention resolution phase 1 is performed in pipelined stage 1 and uses a narrow-band busy tone channel. Stage 1 reduces the number of contending stations to a smaller number (only these stations compete in stage 2). Contention resolution phase 2, as well as RTS/CTS handshake and Data/ACK transmission, belong to pipelined stage 2 and are performed on the data channel. This scheme is named "Partial Pipelining" in the sense that channel contention is only partially resolved in the pipelined stage 1.

By using pipelining, stage 1 proceeds in parallel to stage 2, as shown in Figure 4. Stage 1 takes advantage of the time period that data channel is busy to reduce the channel contention degree. Notice that the end of stage 2 is marked by the end of a successful data packet transmission, and it can be known from the "Network Allocation Vector (NAV)" used by IEEE 802.11.

B. Protocol Description

To operate stage 1 in parallel to stage 2, "Partial Pipelining" scheme requires a busy tone channel, in addition to the data channel. A station maintains a backoff counters bc_1, a contention window CW1 for contention resolution phase 1, a backoff counter bc_2, a contention window CW2 for phase 2. When a pair of source and destination are exchanging packets on the data channel, the remaining stations begin to count down their phase 1 backoff counter bc_1. Upon bc_1 reaching zero, the station will send out a signal on busy tone channel to claim that it has won stage 1. Other stations will be aware of the wining station's presence through sensed busy tone and freeze their phase 1 backoff counters. Notice that *multiple* winning stations from stage 1 are possible when more than one station count down their bc_1 to zero at the same time. When current packet transmission finishes, the winning stations from stage 1 will compete in stage 2 for the right of next packet transmission. It is possible that, by the end of current packet transmission, no station counts down its bc_1 to zero. When such cases occasionally occur, all backlogged stations enter stage 2 to contend for the channel access.

With backoff counter bc_2 and contention window CW2, stations in stage 2 follow a backoff algorithm similar to IEEE 802.11. However, now that the number of contending stations in stage 2 is small, channel contention can be resolved efficiently. The benefits of "Partial Pipelining" include:

1) Pipelined stage 1 proceeds in parallel to stage 2. Without consuming much channel resource (except for the narrow-band busy tone channel), stage 1 reduces the data channel contention significantly.

2) Only a small number of stations will contend for the data channel in the second stage. The channel contention among them can be resolved efficiently and the collision probability can be reduced significantly.

More details for "Partial Pipelining" scheme can be found in [5]. Notice that the two contention resolution phases used in "Partial Pipelining" are functionally similar to the "elimination" stage and "yield" stage used in HIPERLAN/1. The advantage of "Partial Pipelining" over HIPERLAN/1 is that contention resolution phase 1 is *pipelined* and it consume little channel resource in fulfilling its responsibility of reducing channel contention. Similar pipelining technique can be applied to HIPERLAN/1 as well.

C. Performance Evaluation

The purpose of this paper is to demonstrate the effect of applying pipelining techniques to MAC protocols. The schemes proposed in this paper make simple modifications on IEEE 802.11 DCF backoff algorithm to apply pipelining techniques. Hence, we primarily present their results compared with IEEE 802.11 to show the improved performance. We expect similar pipelining techniques can be adapted to various MAC protocols to gain performance improvement.

Our simulation results are based on a modified version of ns-2 network simulator. Channel bit rate is set to 11 Mbps for 802.11. Since proposed "Partial Pipelining" scheme requires a busy tone channel, for which we assign 2% channel bandwidth, the resulting data channel bit rate we are using in simulations is 10.78 Mbps (98% of 11 Mbps). Physical layer preamble and header length is set to 192 μs according to IEEE 802.11 standard (with Direct Sequence Spread Spectrum) [2]. The packet payload size used is 512 bytes and the RTS/CTS access method is used. We use *Constant Bit Rate* traffic and traffic rate is aggressive enough to keep an active station continuously backlogged.

Increasing the number of active stations from 1 to 256, the aggregate throughput of "Partial Pipelining" and IEEE

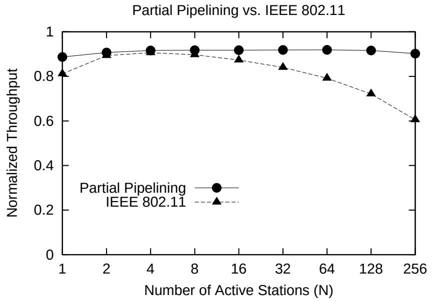

Fig. 5. Aggregate throughput of Partial Pipelining and IEEE 802.11 (normalized to the maximum throughput of 3100.3 Kbps). Horizontal axis is plotted in log-scale.

802.11 DCF in wireless LANs is normalized to the maximum aggregate throughput of 3100.3 Kbps [2] (i.e., without any cost incurred by channel contention resolution) and is presented in Figure 5.

The simulation results[3] show that "Partial Pipelining" is more stable than 802.11 in the sense that its aggregate throughput changes little when the number of active stations (N) increases, and the throughput remains around the peak point of 802.11 up to 256 contending stations. When N is 256, the throughput of "Partial Pipelining" is 1.49 times throughput of 802.11.

IV. IMPLICIT PIPELINING

A. Motivation

Both "Partial Pipelining" and HIPERLAN/1 rely on the signaling mechanisms (energy burst or busy tone), which may make it difficult to be used in multi-hop wireless networks due to hidden terminals.

"Implicit Pipelining" is motivated by the desire to eliminate the dependence on busy tone signaling used in "Partial Pipelining" scheme.

Recall that, in "Partial Pipelining", a station that counts down its phase 1 backoff counter to zero will send a busy tone signal to claim its winning of stage 1. Without the aid of busy tone, other stations cannot learn of existence of the winning stations and will continue to count down their phase 1 backoff counters until the end of current packet transmission. Therefore, when there is no busy tone signaling mechanism, more stations may claim to win stage 1 and enter stage 2 to contend for the channel access.

How much will the negative impact be then? By observing the performance of IEEE 802.11 in Figure 5, we can see that 802.11 backoff algorithm performs reasonably well when the number of contending stations are within the range of [2, 32]. Therefore, if we can design the contention resolution phase 1 backoff algorithm such that the contending stations in stage 2 is controlled within a small range, it is possible to minimize the performance degradation.

Moreover, the length of stage 2 in "Partial Pipelining" scheme limits the number of slots that phase 1 backoff counter (bc_1) can be counted down. Notice that a station simply reduces bc_1 by 1 after each slot in the process of counting down. When there is no busy tone signaling, all stations will count down their bc_1 until the end of current packet transmission, which is equivalent to reducing bc_1 by a fixed amount at the end of current packet transmission. In fact, conceptually, it makes no difference to reduce bc_1 by any amount we desire, which leaves us much flexibility in designing the phase 1 backoff algorithm.

This scheme is named "Implicit Pipelining" since there is no explicit channel associated with stage 1, and the backoff procedure in stage 1 is implicitly performed in parallel to stage 2, as shown in Figure 6.

B. Protocol Description

"Implicit Pipelining" differs from "Partial Pipelining" in the backoff algorithm for the contention resolution phase 1. Instead of reducing phase 1 backoff counter bc_1 by 1 after each slot in the time period parallel to stage 2, in "Implicit Pipelining", stage 1 is implicitly performed in that a station reduces its bc_1 by a quantity F each time when it overhears a successful packet transmission, as shown in Figure 6. Whenever a station's bc_1 becomes *less than or equal to 0*, this station enters stage 2. By adaptively adjusting the distribution of bc_1 among all contending stations and the value of F, the number of contending stations in stage 2 can be controlled.

Among all contending stations in stage 2, only one station will win the channel (following a procedure similar to 802.11 DCF). The winning station transmits its packet, resets its CW1 and returns back to stage 1. Other stations that lose channel will *double* its CW1 and return to stage 1.

Intuitively, the distribution of CW1 changes with the number of contending stations in stage 2. If very few stations enter stage 2, then very few stations will double CW1 upon losing channel contention in stage 2. CW1 of the contending stations tends to be small. On the other hand, if the channel contention is severe in stage 2, many stations (except for the winning one) will lose the channel and double their CW1. As a result, many stations tend to have large values of CW1.

On the other hand, the number of stations entering stage 2 is closely related to the values of CW1. If many stations have large CW1, then bc_1 among all contending stations tends to be widely distributed and a small number of stations may enter stage 2 with an appropriate choice of F. In our implementation, F is reset to a minimum value when a station enters stage 1, and F increases with time so that the longer a station has stayed in stage 1, the more aggressively it will reduce its bc_1, hence, a larger probability of entering stage 2.

---

[2] Taking into account the overhead introduced by data packet header (48 bytes), RTS (20 bytes), CTS (14 bytes), ACK (14 bytes), DIFS ($50 \mu s$), SIFS ($10 \mu s$), physical layer preamble and header ($192 \mu s$) respectively for each of RTS, CTS, DATA and ACK, the total transmission time is $1290.18 \mu s$ for each payload packet (512 byte) using full bandwidth of 11 Mbps. The maximum throughput calculation follows.

[3] The parameters for "Partial Pipelining" are set as follows: $CW1_{min} = 31$, $CW1_{max} = 255$, $CW2_{min} = 15$, $CW2_{max} = 127$.

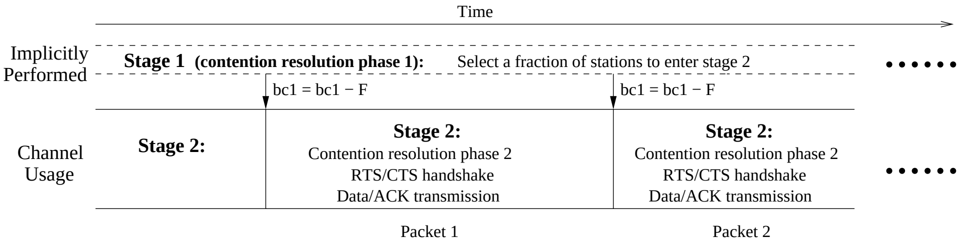

Fig. 6. Implicit Pipelining Scheme

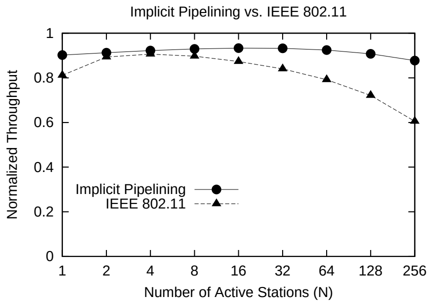

Fig. 7. Aggregate throughput of "Implicit Pipelining" and IEEE 802.11. Horizontal axis is plotted in log-scale.

The interaction between stage 1 and stage 2 helps to construct a feedback system. As a result, channel contention in stage 2 is well controlled. Simulation results show that with up to 256 active stations, the average number of stations contending for the channel is less than 28. Our analysis also confirms this trend [6].

C. Performance Evaluation

We repeat the sets of simulations in Section III for "Implicit Pipelining". Notice that the channel bit rate of "Implicit Pipelining" is set to 11 Mbps since it does *not* require an extra channel. The simulation results[4] are presented in Figure 7 and are compared with 802.11. The aggregate throughput of "Implicit Pipelining" degrades slightly faster than "Partial Pipelining" due to more channel contention in stage 2. However, the performance degradation is small as "Implicit Pipelining" has successfully controlled the contention degree in stage 2. Compared to 802.11, "Implicit Pipelining" remains more stable, and the throughput gain in large networks is significant. With 256 active stations, the throughput of "Implicit Pipelining" is 1.46 times throughput of 802.11, which is only 3% less than "Partial Pipelining". Since "Implicit Pipelining" scheme does not rely on signaling mechanisms, it has the potential to be used in multi-hop networks [7].

V. CONCLUSION

Contention resolution and packet transmission are usually performed sequentially in current MAC protocols. The involved tradeoff between the channel resource used for contention resolution and the resulting collision probability causes design difficulties for these MAC protocols. In this paper, we propose to pipeline multiple access control so that the contention resolution procedures overlap (or partially overlap) in time with packet transmissions. The main benefits can be summarized as follows:

- The channel cost associated with contention resolution is (partially) hidden;
- Pipelined contention resolution helps to reduce the data channel contention degree without consuming data channel bandwidth.
- The collision probability is reduced due to reduced data channel contention degree.

Three pipelining schemes ("Total Pipelining", "Partial Pipelining" and "Implicit Pipelining") proposed in this paper make modifications to IEEE 802.11 in order to apply pipelining techniques. Simulation results of "Partial Pipelining" and "Implicit Pipelining" show significant performance improvement over 802.11, thus, demonstrate pipelining techniques can help to improve the performance of multiple access control protocols.

ACKNOWLEDGMENT

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

The authors would like to thank Vaduvur Bharghavan for the discussions on "Total Pipelining" scheme, and thank Romit Roy Choudhury, Pradeep Kyasanur and Matt Miller for their suggestions on the paper.

REFERENCES

[1] 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.
[2] IEEE, "Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) Specifications," *IEEE Standard 802.11*, June 1999.
[3] Asimakis Tzamaloukas and J.J.Garcia-Luna-Aceves, "A Receiver-Initiated Collision-Avoidance Protocol for Multi-Channel Networks," *IEEE Infocom*, 2001.
[4] John Hennessy and David Patterson, "Computer Architecture: A Quantitative Approach," *3rd Edition, Morgan Kaufmann*, 2002.
[5] Xue Yang and Nitin H. Vaidya, "Pipelined Packet Scheduling in Wireless LANs," *Technical Report, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign*, Aug. 2002.
[6] Xue Yang and Nitin H. Vaidya, "DSCR: A More Stable Wireless MAC Protocol," *Technical Report, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign*, Dec. 2002.
[7] Xue Yang and Nitin H. Vaidya, "DSCR – A Wireless MAC Protocol Using Implicit Pipelining," *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) Poster Session*, June 2003.

---

[4] The parameters for "Implicit Pipelining" are set as follows: $CW1_{min} = 15$, $CW1_{max} = 1023$, and $CW2_{min} = 31$, $CW2_{max} = 1023$.