Efficient Polling Scheduler for IEEE 802.11 WLAN

Taekon Kim[*★] , Hyungkeun Lee[**] and Jinhwan Koh[***]

Abstract

Although the Distributed Coordination Function is the fundamental access protocol of IEEE 802.11, it cannot meet the Quality of Service (QoS) requirements in general. So, the Point Coordinate Function is provided to support QoS related services. However, it has inherent problems. Access point (AP) has no knowledge of the queue status and instantaneous channel condition of stations in the system. In this paper we propose an efficient and versatile polling scheduler that shows excellent throughput and fairness performance. Comparison with well known polling schemes is provided through computer simulation under various channel situations including error prone environments.

Key words: IEEE 802.11 Wireless LAN, Quality of Service (QoS), MAC protocol, Point Coordination Function (PCF), polling scheduler

I. Introduction

The IEEE 802.11 MAC protocol includes two operational modes characterized by the Distributed Coordination Function (DCF), the Point Coordination Function (PCF), respectively. The DCF provides contention services to use for either infrastructure or ad-hoc network configuration while the PCF provides contention free services and is only usable on infrastructure network configuration [1].

Although DCF known as a Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) is the fundamental access protocol of IEEE802.11, it is not designed for Quality of Service (QoS) related application.

As the number and the traffic of active stations increase, overall throughput of the system drops significantly and medium access delay of each station grows fast. To make matters worse, when the channel conditions of active stations are not good and when there are lots of broadcast/multicast frames, the contention in the system rises seriously. Also it cannot guarantee stations to send data in a given time. For these reasons IEEE 802.11 provides the centralized polling-based access mechanism PCF to support QoS related services. However, to use PCF efficiently the fundamental question of scheduling algorithms, "which stations and how many polls AP should assign at one time?" should be answered.

In fact, PCF has inherent problems costing system performance. First, Point Coordinator (PC) has no mechanism to know the queue status of stations. PC may give transmission opportunities to stations without any data. Also, it is hard for the PC to determine the transmission opportunity time given to a polled station. The PC may keep polling a station until there is no data to transmit, whereas in some cases other stations would hardly get an opportunity to be polled. Second, PC has no instantaneous information about channel conditions of stations in the polling list. If the PC frequently polls stations in erroneous situation, the throughput

[*] Electronics and Information Engineering, Korea University
[**] Dept. of Computer Engineering. Kwangwoon University
[***] Electronical Engineering, KyungSang University
★ Corresponding author

※ Acknowledgment
This research was supported by the MIC, under the ITRC support program supervised by the IITA and by a Korea University Grant

of the system would drop significantly. Due to these kinds of problems, 802.11 MAC does not specify how the AP determines polling schemes. This explains why 802.11 products with the PCF function are hardly found in the market.

With the strong demand for wireless multimedia data transmission, the study of WLAN systems that supports QoS requirements has been done. However, most of the research has focused on improvement of the DCF and as reviewed these approaches have limitation [2-4]. Moreover, research on polling schemes has been done in restricted environment, ideal channel condition [5]. Also, the research on fair scheduling algorithms in WLAN usually assumes that the scheduler exactly know the buffer status and instantaneous channel condition of each station in the system [6-7]. In this case the schedulers focus on fair polling to stations in good channel condition and on compensation for stations previously in bad channel. In fact, the performance of scheduling algorithms greatly depends on the system configurations, traffics and channel conditions. In this paper to solve the discussed shortcomings of 802.11 PCF polling, we do not assume such exact system information is available.

Instead of assuming the exact system information, it is estimated by using available parameters from the received packets. According to the frames from stations, we can classify the polling list into four classes. Then we will discuss on polling priorities and policies of each class for either throughput or fairness those are the most important elements in polling scheduler design. Based on the polling priorities and policies, we propose a polling scheduler that can allow either throughput or fairness oriented polling which also guarantees good performance to some extent regardless of the system environment.

In this paper we will prove the performance of the proposing polling scheduler. Simulation on performance comparison with Non-preemptive (*NP)*, Round-Robin (*RR)* and First-In-First-Out (*FIFO)* polling schemes varying channel environments has been done.

II. Efficient PCF Polling Scheme

2.1 Popular Polling Schemes

While joining a BSS (Basic Service Set) a station may request to be placed on the polling list. Then AP put the association identifier (AID) on the list. In the following we explain and make a short analysis of three popular polling schemes.

Non-preemptive (*NP*) polling scheme: Unless the PC receives the null frame or data frame with more field set to 0, it keeps polling the same station. In short PC polls a station on the polling list in order and keeps polling the same station as long as the station has data to transmit or to receive.

Round-Robin(*RR*) polling scheme: PC polls a station on the polling list in order. This time, however, the PC immediately turns into the next station in the polling list after receiving the appropriate feedback from the polled station.

FIFO polling scheme: PC follows the order of data frames in the buffer queue. It first checks if there is a packet in the queue. If any, it sees the index of the packet and then polls the station accordingly. Therefore, this scheme follows the order of frames in the PC buffer and always piggybacks.

RR seems to guarantee the best fairness. However, PC may give polls to stations with no data or to stations in bad channel. *NP* and *FIFO* generally show good throughput performance. However, channel may be allocated to a few greedy stations while other stations suffer from starvation. Moreover, if the greedy stations do not use channel efficiently (with small packet lengths and low data rates) the overall system throughput will be decreased significantly.

2.2 Polling List Management

To improve the performance of polling schemes, PC should not poll stations with neither any downlink nor uplink data and it should not poll stations in error environment. Also, PC may issue more than one polls to stations with data buffered in good channel condition. However, as we discussed, PC do not have the exact instantaneous channel conditions and the uplink queue status of stations in the BSS. So, we will discuss about how those information can be approximately estimated by using available parameters of the received data packets and how the polling list shall be managed. When PC polls a station, in general there could be three types of response from the station:

1. PC receives data correctly without error. The polled station sends one data frame when it has some uplink data. Otherwise it sends a null frame. In both cases, the channel condition between AP and the station can be considered in good situation.
2. PC receives an erroneous packet.
3. The station does not responses.

According to responses from a polled station, we can classify the polling list into four classes: good channel with data (GD), good channel with no data (GN), bad channel (BC) and no response (*NR*). The Fig. 1 shows the new polling list with the four classes and the queue status of AP. The polling list shall be updated by the most recent frames received from stations in either the PCF polling or the DCF contention.

2.3 Polling Priority

When it comes to the polling scheduler design, throughput performance and fairness are most important elements. So, we will discuss on the polling priority for either throughput or fairness.

1. Obviously, PC may give the highest priority to the stations in GD class.

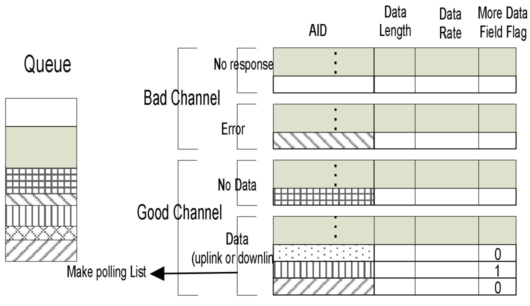

Fig. 1. New polling list with four classes.

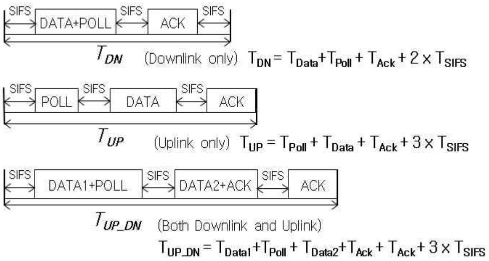
SIFS
DATA+POLL
ACK
T_{DN} (Downlink only) $T_{DN} = T_{Data} + T_{Poll} + T_{Ack} + 2 \times T_{SIFS}$
POLL
DATA
T_{UP} (Uplink only) $T_{UP} = T_{Poll} + T_{Data} + T_{Ack} + 3 \times T_{SIFS}$
DATA1+POLL
DATA2+ACK
T_{UP_DN} (Both Downlink and Uplink)
$T_{UP\_DN} = T_{Data1} + T_{Poll} + T_{Data2} + T_{Ack} + T_{Ack} + 3 \times T_{SIFS}$

Fig. 2. New polling list with four classes.

The stations in this class are supposed to be in good channel situation and to have some uplink or downlink data. When there is any downlink data, PC can piggyback. The question is that what the polling order in GD is. As we have reviewed, inefficient channel usage in *NP* and *FIFO* can cause significant throughput decrease. So, we introduce the channel efficiency of *i* station and GD list is sorted according to its value. The stations in GD can be classified into three cases (Fig. 2); stations with downlink only, with uplink only or with both data. Time durations for one poll chance are calculated and channel efficiencies are define as follows.

$$E_i = \frac{MSDUs\ size\ of\ i\ station.}{T_{UP}\ or\ T_{DN}\ or\ T_{UP\_DN}} \qquad (1)$$

Of course, uplink data length and rate are assumed to the same as those of the data packet received recently from *i* station. Now the question is how many polls we give to a station in a row. It is obvious that applying *NP* to the new GD list shows the excellent throughput performance. However, for fairness between stations in GD *RR* can be used.

2. PC should assign less priority to stations in GN class. There could be two policies: Firstly, PC just does not poll them at all. When either any downlink data is happened in AP or they send the packet with more data field flag 1 using DCF contention, PC starts to poll again. Secondly, PC may estimate their mean uplink data rates and poll them intermittently adjust to their estimated one. This policy increase fairness. But it may waste resource and introduce lots of computational complexity.
3. PC should assign a little priority to stations in BC class. Similar to the above 2 case, there could be two policies:

4. PC should assign little priority to stations in *NR* class. PC can apply the same policies as one with stations in BC.

Based on the discussed polling priorities specific polling schedulers can be designed aiming for either throughput or fairness performance.

2.4 Efficient Polling Scheduler

In this paper, we propose an efficient polling scheduler (*EP*). Different from other polling schemes the *EP* decides the polling order and iteration numbers for each station before the CFP(Contention Free Period) begins. This improves not only fairness but also throughput performance. The scheduler first plans to give one poll to stations in GD class using *RR*. After one round PC do not have any uplink information and it uses only downlink. So, it gives more chance to station with more downlink using *RR* until the channel is available. So far the scheduler aims both for throughput improvement and for fairness of stations in good condition. In fact, the *EP* uses only the advantages of *RR*, *FIFO* and *NP*.

Efficiency Polling Algorithm

Step 1 (*Initialization*): initialize T_i, P_i (poll iteration weight) and T_{poll} where n is the number of stations in GD and $0 \le i \le n$. Calculate T_{CFP} [1].

$$T_i,\ P_i,\ T_{Poll} = 0, \tag{2}$$

Step 2 (*Sorting GD list*): Calculate T_i and E_i (channel efficiency), sort the GD in descending order and make a new polling list.

Step 3 (*Poll Scheduling1*): Schedule poll in *RR* with the polling list until the CFP is available.

Do While

$$T_{poll} = T_{poll} + T_i \le T_{CFP},\ P_i = P_i + 1, \tag{3}$$

If $T_{poll} \ge T_{CFP}$ *Go Step* 6,

Step 4 (*Poll Scheduling 2*): Find stations $(m \le n)$ with more downlink data, calculate E_j and make a new polling list. Calculate new T_j, schedule poll in *RR* until the CFP is available and update P_j.

Do While

$$T_{poll} = T_{poll} + T_i \le T_{CFP},\ P_i = P_i + 1, \tag{4}$$

If $T_{poll} \ge T_{CFP}$ *Go Step* 6,

Step 5 (*Iteration of Step 4 or Step 1*): If there is enough time to transmit at least one packet and if there are stations with more downlink data, then go back to Step 4, otherwise go Step 1. If there is no more time left, go Step 6.

Step 6 (*Poll*): Poll the stations according to P_j.

Before applying Step 4, PC may poll the stations in GN, BC and NR intermittently to improve fairness costing throughput decrease.

III. Performance Evaluation

3.1Efficient Polling Scheduler (*EP*)

In this section we will prove the performance of the proposed polling scheduler. For simulation simplicity we introduce three weighting factors that characterize *EP* scheduler accurately. First, no-traffic weighting factor is introduced for stations in GN class. Secondly, error-occurrence weighting factor is considered to count for stations in BC and *NR* classes. The stations with no traffic and/or the station in error environment are/is skipped next polling cycles for the duration of given weights, thereby allocating other stations more transmission bandwidth[2].

[1] T_{CFP} is the available CFP duration in a given superframe.

[2] In this experiment appropriate values for the *EP* performance are assumed to be obtained. Note that the appropriate value would be changed in different environment and configuration.

Table 1. number of Stations for each environment

Environment Type	Under 1% PER	Around 10% PER	Around 40% PER
Good	20	0	0
Bad	4	8	8

3.2 Channel Environment

To reflect the different error environments into simulation the simplified radio channel model is in use for the simulation [8-9]. The channel quality may be determined by PER(Packet Error Rate) perceived at the receiver. However in IEEE 802.11 the each type of frame suffers different PER. Therefore we present the channel model that reflects such characteristics, frame length, modulation and coding schemes.

In the simulation for the purpose of comparison with polling schemes, PER is used as criterion to differentiate the type of the error environments as in the Table 1. It lists the defined environment type and the number of stations placed in different locations classified by PER, which is on the basis with frame size of 2304Byte running at 54Mbps.

3.3 Simulation Results and Analysis

In the simulation, we created an infrastructure basic service set (BSS) including an AP and 20 stations associated with it, which remain stationary all along. A station generates a fixed 2304 bytes long data frame which is maximum length allowed in the IEEE 802.11 MAC and transmits over the medium with a 54 Mbps data rate. The superframe composed of CFP and CP repeats itself every 100msec, i.e., AP sends a beacon every 100msec.

To measure the maximum throughput and poll interval time, we generate the traffic so that it surpasses the amount of network bandwidth then buffer at each station always has data ready to transmit. The case where destination address of the traffic is assigned only for the limited group of stations is considered herein to simulate the situation where stations without any data buffered to send out is inefficiently polled so it decreases overall system performance. Therefore we consider that the half of the stations in the BSS has no traffic available in their buffers.

Fig. 3 compares the throughput of *EP* using iteration. Also the effectiveness using piggybacking makes *FIFO* similar performance with *EP*. The performance for non-preemptive scheme degrades a bit due to no-piggyback. This is because each of traffic buffered at stations and PC becomes unbalanced so improvement in terms of piggyback disappears. *RR* shows poor performance because under this scheme it wastes poll attempts to stations without traffic.

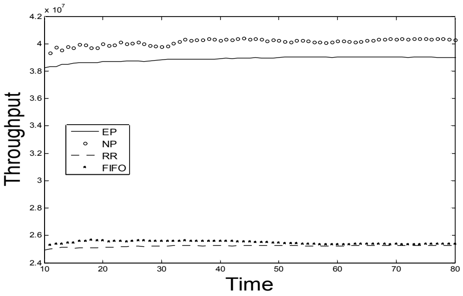

Fig. 3. Throughput comparison under "Good" environment.

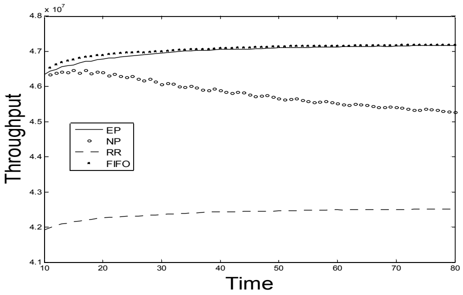

Fig. 4. Throughput comparison under "Bad" environment.

Throughput of *EP* and *NP* schemes are outstanding in "Bad" environment as shown in Fig. 4 because when the PC polls the station in good condition. If station in bad environment comes across, the next station in the polling list takes turns to be polled. Difference between *EP* and *NP* comes from gains via piggyback.

Fig. 5 shows poll interval time for the polling schemes measured at a specific station located in "under 1% PER" range as defined in Table 1. In fact, polling interval time is defined as one between two consecutive polling transmission opportunities of a station. *RR* scheme shows the shortest interval time due to its nature of guaranteed fairness and *EP* also shows comparative performance in this aspect. Note that *EP* shows better throughput performance in both good and bad channel environments and good fairness. In fact, the proposing *EP* scheme overcomes the weakness of other schemes.

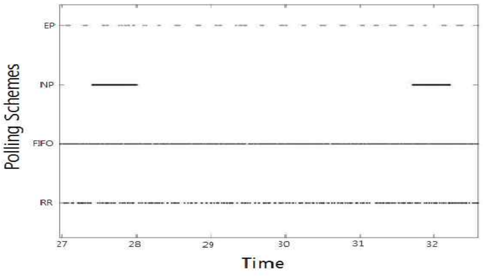

Fig. 5. Poll interval times of four schemes.

IV. Conclusion

In this paper we addressed the problems of PCF polling in the IEEE 802.11 MAC protocol and presented an improved polling scheme, an efficient polling scheduler (EP). The system information is estimated by using available parameters and in accordance with it the polling list is classified into four classes. Based on polling priorities and policies of each class, the EP provides versatile polling procedures for either throughput performance or fairness and it shows excellent throughput to some extent regardless of the system environment. The simulation results show that *EP* has performance improvement over the popular polling schemes, *NP*, *RR* and *FIFO* while preserving the fairness.

References

[1] IEEE "Information Technology" Telecommunications and Information Exchange Between Systems LAN/MAN Specific Requirements Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications" Std 802.11, 1999

[2] Frederico Cali, Marco Conti and Enrico Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," {\it IEEE/ACM Trans. on Networking}, Vol.8, No.6, December 2000

[3] Kanghee Kim, Aftrab Ahmad and Kiseon Kim, "A Wireless Multimedia LAN Architecture Using DCF with Shorten Contention Window for QoS Provisioning," Communications Letters, IEEE , Volume: 7 Issue: 2 , Feb. 2003

[4] Anders Lindgren, Andreas Almquist, Olov Schelen, "Evaluation of Quality of Service Schemes for IEEE 802.11 Wierless LANs," Local Computer Networks, 2001. Proceedings. LCN 2001. 26th Annual IEEE Conference on , 14-16 Nov. 2001

[5] Jing-Yuan Yeh, Chienhua Chen, "Support of Multimedia Services with the IEEE 802.11 MAC Protocol," ICC 2002. IEEE International Conference on Communications, Volume: 1 , 28 April-2 May 2002

[6] T. S. Eugene Ng, Ion Stoica, Hui Zhang, "Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors," Infocom 1998

[7] Oran Sharon, Eitan Altman, "An Efficient Polling MAC for Wireless LANs," IEEE/ACM Transactions on Networking, Vol. 9, No. 4, Aug. 2001

[8] J. M, Gilbert, "802.11a Wireless Networks: Principles and Performance," Atheros Communications Inc., May 2002

[9] Gregory D. Durgin, Theodore S. Rapport, Hao Xu, "Partition-based Path Loss Analysis for In-Home and Residential Areas at 5.85 GHz, " Global Telecommunications Conference, 1998. Volume: 2 , 8-12 Nov. 1998

BIOGRAPHY

Kim, Taekon (Member)

2001 : PhD degree in EE, Pennsylvania State University 2001~2002: Intel Corp. 2003 ~2004: Samsung Digital Media Research Lab. 2005~present: Assistant Prof. in Electronics and Information Engineering, Korea University

<major interests> Multimedia and Communications. Signal Processing, Networks.

Lee, Hyungkeun

1987 : BS degree in Electronic Engineering, Yonsei University.

1998 : MS degree in Computer Engineering, Syracuse University.

2006 : PhD degree in Computer Engineering, Syracuse University.

<major interests> Wireless Networks, Sensor Networks, Mesh Networks.

Koh, Jinhwan

1999 : PhD degree in Electrical Engineering, Syracuse University.

2003~present: Associate Prof. in Electronic Engineering, Gyeongsang University.

<major interests> Communications, Signal Processing, Radar, Micro-processor.