논문 08-33-09-05 한국통신학회논문지 '08-09 Vol. 33 No. 9

Adaptive Wireless Schedulers based on IEEE 802.11e HCCA

Jinoo Joung* ***Regular Member***, Jongho Kim* ***Associate Member***

ABSTRACT

We identify the problem of the current IEEE 802.11e HCCA (Hybrid Coordination Function Controlled Channel Access) scheduler and its numerous variations, that the queue information cannot be notified to the Hybrid Coordinator (HC) timely, therefore the uplink delay lengthens unnecessarily. We suggests a simple solution and a couple of implementation practices, namely the Adaptive Scheduler with RTS/CTS (ASR) and Adaptive Scheduler with Data/Ack (ASD). They are both further elaborated to emulate the Deficit Round Robin (DRR) scheduler. They are finally compared with existing exemplary schedulers through simulations, and shown to perform well.

Key Words : WLAN, HCCA, TXOP, DRR, Wireless DRR

Ⅰ. Introduction

The Wireless Local Area Network (Wireless LAN or WLAN) technology advances with splendid successes both in the areas of standard and market. With the surging popularity of WLAN, multimedia applications such as voice, streaming audio/video, network gaming, teleconferencing are expected to be efficiently supported in WLANs. These applications require certain quality of service (QoS) support in terms of bandwidth and delay requirements. The IEEE 802.11e standard[2,3,4,5,6] aims to offer this QoS support to the 802.11 based WLANs. The 802.11e standard introduces the Hybrid Coordination Function (HCF) that offers channel access mechanism through two MAC mechanisms namely Enhanced Distributed Channel Access (EDCA) for prioritized QoS and HCF Controlled Channel Access (HCCA) for parameterized QoS. Most of the multimedia applications require hard QoS requirements that could only be provided through the HCCA mechanism instead of EDCA which is only able to provide a “soft” or “relative” QoS. HCCA is a polling based mechanism in which the Hybrid Coordinator (HC) collocated with Access Point (AP) grants the transmission opportunities to the contending traffic streams generated from these applications. However, the HC has to properly schedule the granting of the transmission opportunities in order to provide these applications with their pledged QoS. Moreover, the HC should not grant admission to those traffic streams for which it cannot provide the required QoS support. The standard provides a Reference scheduler design along with the design of an admission controller unit to complement the HCCA scheme. The Reference scheduler, however, fails to provide QoS guarantees for Variable Bit Rate (VBR) traffic[8]. Many multimedia applications such as real time multimedia streaming, videoconferencing, network gaming produce VBR traffic. The packets are generated in variable intervals with fluctuating packet sizes. This type of traffic poses significant challenge to the HCCA scheduler because of their time varying nature. Although the transmission opportunities could be granted according to the

---

※ This work was supported by the Korea Research Foundation Grant funded by the Korean Government(MOEHRD).(KRF-2006-331-D00366)

* 상명대학교 컴퓨터과학부 (jjoung@smu.ac.kr)

논문번호 : KICS2007-12-591, 접수일자 : 2007년 12월 28일, 최종논문접수일자 : 2008년 9월 4일

expected bit rate and required service intervals, the traffic might be generated at much higher or lower rates at times. The transmission opportunities should, therefore, be adapted dynamically to take the effect of VBR traffic arrival into consideration.

HCCA in the IEEE 802.11e standard is the medium access method that is designed to provide hard or parameterized QoS guarantee to performance critical applications. It improves over the Point Coordination Function (PCF) of legacy 802.11 MAC. Similar to PCF, HCCA is a polling based mechanism where the access to the medium is arbitrated centrally. However, it resolves the limitation of PCF by eliminating the unpredictability of beacon delays. This problem of PCF made it impossible to guarantee transmission times of the polled stations and contributed to its failure in getting accepted as a viable solution for QoS support in 802.11 based WLANs.[1]

IEEE 802.11e introduces the concept of Traffic Stream (TS) which can be thought of as a set of data units (MSDU) that has to be delivered conforming to a corresponding Traffic Specification (TSPEC). A TSPEC characterizes the traffic streams and its QoS requirements. A TSPEC negotiation takes place between a station (STA) that works as the source of the TS and the HC collocated with the access point, before a TS can be served through the HCCA. The parameters of the TSPEC that are considered during the TSPEC negotiation include nominal MSDU size in octets, mean data rate in bps, Maximum Service Interval (MaxSI) which is the maximum interval in micro seconds (µsec) between two successive polls for the stream, minimum PHY rate in bps and delay bound in µsec. Once the TSPEC negotiation is successful, the TS is admitted and offered transmission opportunities (TXOPs) by the HC in each polling cycle. The length of the TXOPs offered to the stream is decided through a scheduling scheme. The periods in which the HC has the exclusive control of the channel for data transmission are called Controlled Access Periods (CAPs). The HC polls the STAs during these CAPs and offers TXOPs to the admitted streams.

The scheduler first determines a Scheduled Service Interval (SI), which is the time interval used by the AP to periodically poll each non-AP STA that has one or more streams admitted by the admission controller. The SI is calculated as a number which is smaller than all the MaxSIs of the admitted streams and a submultiple of the beacon interval. Any admitted stream will be able to get a TXOP at the end of each SI. The idea is to provide even the most QoS constrained stream with at least one TXOP within its MaxSI time limit. The scheduler then determines the TXOP duration needed for each stream by considering the number of packets that may arrive within an SI. The detailed operation principle is illustrated in the section II.

It was pointed out [8] a number of limitations of the Reference scheduler which fails to meet QoS guarantees for VBR multimedia applications. It has been shown that the QoS guarantees for VBR traffic can not be met by considering only mean values of the traffic statistics or without adapting the transmission opportunities properly. In an ideal scenario, the queue of the VBR stream should become empty at the end of TXOPs, because the scheduler is designed to allocate enough TXOP to transmit all the packets within the scheduled SI. However, the variable traffic intensity of the streams causes backlogs to develop at the end of TXOPs since the TXOPs are not adapted to cope with this phenomenon. This accumulated backlog might further add to the delays of the packets arriving during the next SIs in addition to the delays caused by the insufficient TXOPs. The backlog eases up when packets arrive at a lower rate than expected for some period of time. However, a significant number of packets eventually end up experiencing high access delays due to the queue buildup. Besides, the Reference scheduler treats the admitted streams with equal priority by serving them with similar urgencies in equal intervals. Consequently, streams with stricter delay bounds are penalized more when backlogs buildup at the end of TXOPs and end up with higher percentage of their packets failing to meet the delay requirement.

There is a myriad of solutions suggested to alleviate this problem of the reference scheduler. These are summarized in the next section. In any of these approaches, however, the estimation for the queue size is based on the information gathered at the previous slot, and the actual TXOP allocation occurs at the current slot (this is the lag, which is closely related to the Service Interval). Delay may increase proportionally to this interval. This is serious when SI is large. In the third section we revisit this problem and try to solve as accurately as possible. It is suggested a couple of ways to reduce the lag within a single frame exchanging time. The first one is to notify the queue size of a STA to HC by Request To Send (RTS) frame, and to notify the TXOP to the STA by Clear To Send (CTS) frame. The second idea is to let HC assign a TXOP as usual, but right after the TXOP allocation, the STA informs the current queue size by data frame it sends to the HC. The notified HC then recalculates the TXOP and reassigns it. These simple idea is evaluated through extensive simulations, and proved to perform very well.

Ⅱ. Related Researches

The operation principle within a CAP can be summarized as the following.

- Guaranteed channel access on successful registration.
- Each node will receive a TXOP by means of polls granted to them by the HC.
- TXOP based on negotiated Traffic specification (TSPEC) and observed node activity.
- TXOP is at least the size of one Maximum sized MSDU at the PHY rate.
- Access Point advertises polling list.

What is known as the reference scheduler operates as follows. The service schedule, a defined term in 802.11e, is the polling order and the amount of TXOPs granted to a station for each polling. The schedule for an admitted stream requires two steps. First, the calculation of the scheduled Service Interval (SI), and the next, calculation of TXOPs duration for a given SI. The calculation of the Service Interval (SI) involves the calculation of the minimum of all Maximum Service Intervals for all admitted streams. Let this minimum be ″m″. Choose a number lower than ″m″ that is a submultiple of the beacon interval. This is SI. For example, if MSI1=15ms, MSI2=20ms, and the beacon interval equals to 100ms then the SI becomes 10ms. The calculation of the TXOP follows the equations below.

$$N_i = \left\lceil \frac{SI \cdot \rho_i}{L_i} \right\rceil \text{ and}$$

$$TXOP_i = \max\left[\frac{N_i \cdot L_i}{R_i} + O,\ \frac{M}{R_i} + O\right]$$

where ρ_i is the mean data rate from the STA i, L is the Nominal MSDU Size obtained from the negotiated TSPEC, SI is the scheduled Service Interval, N is the number of MSDUs that arrived at the Mean Data Rate during the SI, R is the Minimum Physical Transmission Rate, M is the Maximum MSDU size, and finally O is the overheads in time units due to IFSs, ACKs, and CF-Polls.

The problem of the reference scheduler is that it is obviously designed for CBR-type traffic, and it does not fit for VBR, where the traffic generation rate (from the above layer) fluctuates. QoS scheduler would be much more efficient if it could schedule TXOPs of variable lengths at arbitrary intervals and calculate the TXOP duration based on actual queue size of each station. The ARROW [9, 10] is an exemplary solution for such VBR traffic environments. In ARROW, with each data frame a station informs the scheduler about its current traffic load. Upon reception of these requests the scheduler will attempt to satisfy the station requirements in the

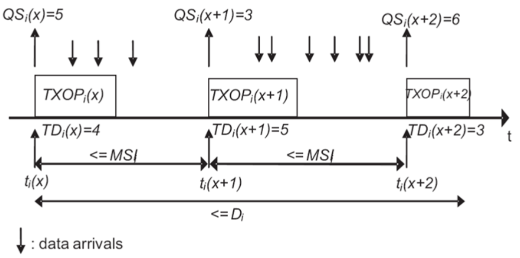


Fig. 1. The ARROW Scheduler

next TXOP by assigning a duration that will be enough to accommodate the pending traffic. Figure 1 depicts the idea of the ARROW scheduler.

To be more precise, there is a QoS Control field in every QoS-Data frame defined in 802.11e, and with the QoS Control field, one can specify the queue size or the requested TXOP value.

Another approach is that, instead of enlarging TXOP, to adapt the Service Interval (SI) itself [11,12]. STAs dynamically update their mSI (min. service interval) & MSI (max. service interval). STAs advertise these values to the HC. The HC then recalculates the SI length. This way the urgent and impending frames can be served more quickly. The SI recalculation, however, can be complex and time-consuming process.

All the approaches so far are based on the information at "the previous state at $(t-1)$" of the STAs. Here, one can easily infer that information at $(t-2)$, $(t-3)$, $(t-3)$, etc. can also be used. This way, a more smooth control is feasible. Then the whole problem can be seen as the classical control theory problem[7]. Let us define as the following. $q_i(n)$ is the queue length at the beginning of nth CAP, $d_i(n)$ is the average input rate, $u_i(n)$ is the average depletion rate of the queue, and T_{CA} is the duration of a CAP. Then the queue length at the beginning of the next CAP can be calculated from the equation,

$$q_i(n+1) = q_i(n) + u_i(n+1) \cdot T_{CA} + d_i(n) \cdot T_{CA}$$

Then the target value for q, q_iT, is obviouly zero, and one can design, for example, a PI controller $G_i(Z)$ that is depicted in Figure 2.

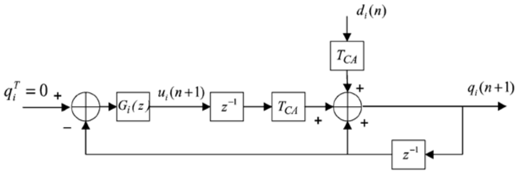


Fig. 2. PI controller-based scheduler

PRO-HCCA scheduler [14] is another example of minimizing the gap between what STA wants and what HC (AP) can respond about. PRO-HCCA is actually a combination of two ideas. The first one is to account for difference delay bounds. It keeps log of "degree of delay" of packets within a TS. TS's can be differentiated according to the delay degree of their packets. This is similar to SI adaptation approach. The second idea is to predict VBR traffic intensities. Again, this is to approximate the amount of traffic that would have been arrived after the last QoS data frame transmission (during current SI). This, in turn, is similar to the Feedback-based approach.

Higuchi et. al [15] introduced the well-known LR server (Latency-Rate server) concept with leaky-bucket for modeling the HCCA uplink scheduler. It is suggested that STA releases any unused TXOP by transmitting QoS Null frame to AP. Any released TXOP can be EDCA utilized. The latency is then obtained by using the LR server model so that the delay bound can be calculated. Based on this equation, it is claimed that the SI length optimization is possible.

An interesting approach can be found in a published patent [16], in which the wireless uplink scheduler emulates a scheduler that is well-known within wired environments. To be more specific, it is motivated to emulate Earliest Deadline First (EDF). STA with flow m provides TSPEC with mean data rate Rm, Average packet size Sm, and the delay requirement δm. tcm is defined to be the creation time of the packet currently at the head of the queue (assuming FIFO for a stream). To meet the deadline, the TXOP must given earlier than $(tcm+\delta m)$. Further

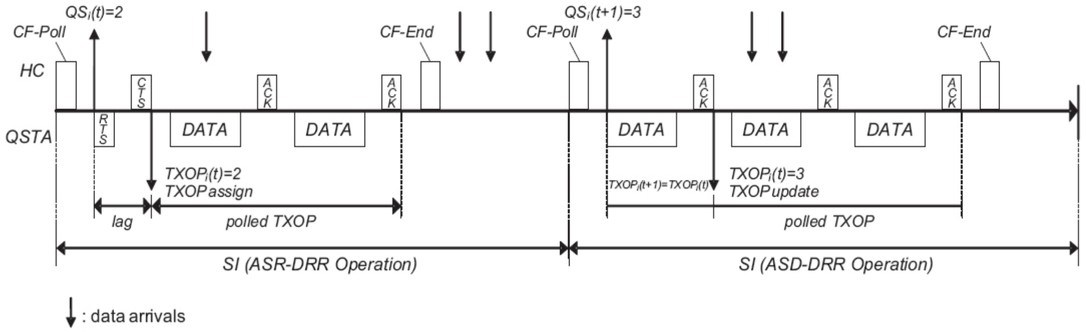


Fig. 3. TXOP assignment with ASR-DRR and ASD-DRR

define minagem and maxagem to be the ages that the oldest packet should reach when the TXOP is granted. The flow is served no earlier than minagem and no later then maxagem, where maxagem is defined to be $(tcm+\delta m)$, and minagem is some time during which the flow wish to gather some packets to transfer in the next TXOP. AP determines which flow to grant TXOP to. tcm, minagem, and maxagem are the important parameters used for the determination. The key idea is still, however, to estimate the number of packets (N_{qp}) in the queue of the selected flow at the time that the TXOP is granted. To be more precise, the (N_{qp}) is estimated with the following equation.

$$N_{pq} = INT(\frac{(t-tcm)\cdot Rm}{Sm})+1$$

where Sm is the average packet size of m. Moreover, even tcm has to be estimated, from the number of remaining packets in the queue.

In any of these approaches, however, the estimation for the queue size is based on the information gathered at the previous slots $(t-1)$, $(t-2)$, etc. and the actual TXOP allocation occurs at the current slot (t). The time difference between the most recent information gathering and the actual TXOP allocation is significant. This is what we call the lag, which is closely related to SI. Delay may increase proportionally to this interval. This is serious when SI is large. In the third section we revisit this problem and try to solve as accurately as possible.

Ⅲ. Adaptive Wireless Scheduler

As it has been observed, the 802.11e HCCA uplink system can be seen as a server, which is defined to be a combination of output queues and a scheduler, within a switch output port. The main difference between wired and wireless servers is as follow. In wired servers, the scheduler is always aware of the queue condition. In wireless servers, in contrast, the queue condition is reported to the scheduler after a certain interval of time. Let's call this interval the *lag*. The lag in 802.11e HCCA system is the SI. The basic idea behind the adaptive scheduler being introduced is that we can minimize the lag, for an instance, by using RTS/CTS which in combination is an optional operation during CFP or CAP but almost a mandatory feature in any WLAN environments practically. Especially the 2 byte duration field in RTS and CTS frames can be used for advertizing the queue length and confirming the TXOP a STA will occupy, respectively. Moreover the duration field is inherently for indicating the transmission duration.

There can be many flavors of possible implementations using the adaptive scheduling. In this method, HC only decides who to transmit at the beginning of a SI. The STA decides the transmission duration, or TXOP, by a scheduling algorithm that are similar to the ones (i.e. DRR) used in wired servers. There can be at least two versions of the implementation practice for the adaptive scheduler as follows. In the first version, the STA advertises its queue size by RTS duration field. The HC may confirm, or for some reason may modify the request with a reduced TXOP, with CTS duration field. In any case, the CTS duration field will indicate the final TXOP the STA will have. In this case the lag is the RTS plus CTS transmission time, which is negligible compared to SI in general. In the second version, HC indicates the TXOP of a STA as does in a conventional HCCA operation, and the STA may request a revised TXOP with the duration field in the QoS data frame, and the HC confirms the request with the duration field in the ACK frame. The first example, which we will call the Adaptive Scheduler with RTS/CTS (ASR), may suffer from the RTS/CTS overhead. The second example, which we will call the Adaptive Scheduler with Data/Ack (ASD), has a slightly longer lag, and may not work as well when the Block-Ack is in use. Figure 3 depicts the idea of the ASR-DRR and ASD-DRR scheduler.

Now consider the first implementation practice with the Deficit Round Robin (DRR) scheduler emulated. A scheduling algorithm selects the next packet to transmit, and decides when it should be transmitted, on the basis of given performance metrics. Two well-established performance metrics of those algorithms are the delay and the fairness. A simple round robin based algorithm, Deficit Round Robin (DRR) [17] can maintain the crucial property of providing the allocated service rates, without the complexity of sorted priority scheduling algorithms. In fact a simple weighted round-robin (WRR) scheduler would provide a rate to a flow if the size of packets is homogenous. When the packet size varies, however, as in the most existing networks, a flow with smaller packets gets disadvantage as the rounds go. Therefore redemptions for flows with shorter packets are necessary, and this is where the idea of registering deficit and redeeming later was introduced, and called DRR. The complexity of the basic DRR can be as low as $O(1)$. A detailed algorithm for DRR can be found in numerous references including[17].

We emulate the DRR in wireless environment, with ASR in IEEE 802.11e HCCA mode in particular as follows. We assume the uplink scheduling (STAs → AP). STAs specify their mean data rate with TSPECs. AP assigns a quantum size (ϕ_i) for a STA i, according the mean data rates of all the STAs. The quantum size can be much greater than the maximum MSDU size. During CFP or CAP, within a Service Interval, AP polls a STA with a CF-Poll frame. The order of the STA polling is round-robin, as in the DRR. Polled STA responds with RTS, specifying the current queued data amount $q_i(t)$. The STA may not respond to the poll at all if there is no data to send. By having no response RTS from the polled STA, the AP figures the STA has no data to send. In the case that a polled STA responds to the AP with an RTS frame indicating the STA has no data to send, the AP can send next polling frame after the AP receives the RTS frame and the channel becomes idle for SIFS. If the queued data amount is zero, the deficit value $D_i(t)$ is reset to zero as well, and the AP sends a poll to another STA, instead of sending CTS. Otherwise, based on the deficit value, $D_i(t-1)$, on the previous SI, AP specifies the TXOP with CTS. The TXOP should not exceed the queued amount. In other words,

$$TXOP_i(t) = \min[\phi_i + D_i(t-1), q_i(t), MBS_i].$$

The deficit value was chosen to not exceed the traffic streams's MBS(Maximum Burst Size) in order to prevent a STA from monopolizing channel. AP updates the deficit value to $D_i(t)$ as follows.

$$D_i(t) = D_i(t-1) + \phi_i - TXOP_i(t-1).$$

This algorithm we call the ASR with DRR emulation (ASR-DRR). As always, within the limit of TXOP, the STA can send multiple frames. There can be a discrepancy between TXOP and the actual amount of transmitted data from a STA, however, because the frame boundary cannot exactly fit to a TXOP. The ASD algorithm is identical to ASR, except that the polled STA responds with a DATA frame, specifying the current queued data amount.

Algorithm 1: ASR-DRR and ASD-DRR

```
Require: Confirm current queue condition through
         RTS or Data frame
$D_i \leftarrow 0$
$Q_i \leftarrow \alpha \cdot [(MEAN\ RATE_i \cdot SI) + MAC\ Header\ Size]$
$MBS_i \leftarrow MAX\ BURST\ SIZE_i$
Ensure: CTS or Ack frame notify updated
        $TXOP_i(t)$ to polled Station
if queue is not empty then
    $TXOP_i(t) \leftarrow \min[\phi_i + D_i(t-1), q_i(t), MBS_i]$
else
    $D_i(t) \leftarrow 0$
    $TXOP_i(t) \leftarrow OVERHEAD$
end if
for $i = 0$ to $n$ do
    $TXOP\ SUM \leftarrow TXOP\ SUM + TXOP_i(t)$
    $i \leftarrow i + 1$
end for
if $TXOP\ SUM > SI$ then
    $TXOP_i(t) \leftarrow TXOP_i(t-1)$
end if
$D_i(t) = D_i(t-1) + \phi_i - TXOP_i(t-1)$
```

Algorithm 1 explains both ASR-DRR and ASD-DRR scheduler's operations. It should be guaranteed that queue size information within RTS frame's duration field be received from STA before this algorithm takes place. Also during call admission process, parameters used in the algorithm shall be initialized. Quantum factor, α is a positive number larger than 1, which is configurable. ASD-DRR uses the same algorithm but with Data/Ack, not RTS/CTS frames, in the process of notifying queue conditions.

Ⅳ. Simulation

We have evaluated the ASR-DRR, ASD-DRR, and existing schedulers such as the Reference HCCA scheduler described in IEEE 802.11e standard[2]. The simulator of choice is the well-known Network Simulator 2(NS-2). To be more precise, we chose the ns-allinone-2.29.3 package, with the ns2hcca patch[18]. This patch for the HCCA implementation is based on the draft amendment 802.11e/D13.0. In simulation scenario, only the uplink scheduling is considered. We composed topology with one AP that functioned as HC and STAs. We did not include admission control part in the simulation, in order to directly compare the various schedulers' performances. It was used G.711 CBR voice and H.261, H.263 VBR traffic for flows from STAs. To make UDP VBR traffic we used real movie clip(The Firm, Starship Troopers)[19]. The mean data rates of simulated flow were 64Kbps, 256Kbps and 256Kbps. To make different flow patterns, we made flows start with different time offsets.

We have simulated scenarios with various flow types like G.711, H.261 and H.263. The data generation rates from different STAs are described in Figure 4. Figure 4 describes traffic specification of encoded real movie clip (The Firm) with H.261, H.263 codec. In Figure 4, it is shown that the rate generated by H.261 codec varies within a small range. The rate generated by H.263 codec varies a lot. In Figure 4, it is shown that even VBR traffic with the same bit rate has differences in the traffic amount created within the unit time. Also, through the above results, it is confirmed that it is important to allocate an accurate TXOP followed by the traffic amount in the process of HCCA.

We now briefly introduce some of major parameters used in the simulation. The PreambleLength is the length of the Preamble, the Physical layer signal that is sent prior to the PLCP(Physical Layer Convergence Protocol) Header. The PLCP is the sublayer between the PHY and MAC layer. The

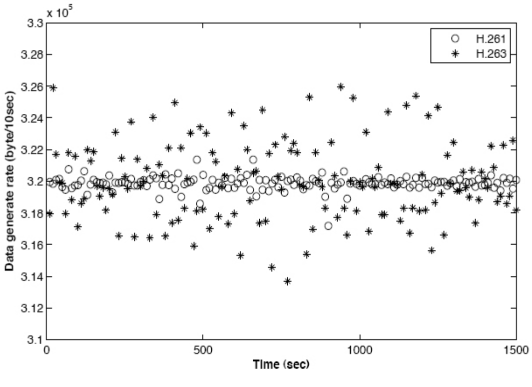


Fig. 4. H.261 and H.263 Data Generation Rate per STAs

Table 1. Parameter values used in the simulation

Parameter	Value
SlotTime	0.000020s (20us)
SIFS	0.000010s (10us)
PreambleLength	144bit
PLCPHeaderLength	48bit
PLCPDataRate	1.0e6 (1Mbps)
dataRate	11.0e6 (11Mbps)
basicRate	1.0e6 (1Mbps)

PLCPHeaderLength is the length of the PLCP header. The PLCPDataRate is the data rate of the PLCP layer. The dataRate is the device-dependent link capacity of the wireless medium. The basicRate is the bandwidth used for the control frame such as RTS/CTS or Poll Frame, which should be less than the dataRate. The values of these parameters used in the simulation are listed in Table 1. The values for the other parameters have chosen accordingly.

First, it was attempted to find out about the optimum quantum size for the suggested schedulers in the wireless environment. It is well understood that in a wired network the smaller quantum size yields a better fairness but more complexity. We implemented the ASR-DRR in the AP to scheduler the uplink flows. To find out the appropriate quantum sizes, we observed the maximum admissible number of STAs with applying various quantum sizes. By the maximum admissible number of STAs, under the condition that STAs are the same type, it is meant the number of STAs, with which the delays are suppressed under an acceptable level.

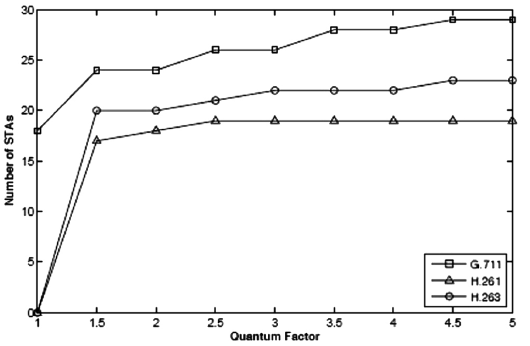


Fig. 5. Number of Station as Quantum value varies

The result is depicted in Figure 5. It is shown that the quantum factor value about 2.5 or more yields the best performance in terms of the throughput. By the above results, we have chosen the quantum size to be MSDU times the quantum factor value 3.0.

Next, we have compared the throughput of each schedulers. To evaluate the scheduler's performance for each traffic type, we examined three kinds of traffics. The result is depicted in Figure 6.

Summarizing the examined results through Figure 6, throughout the difference traffic types, the

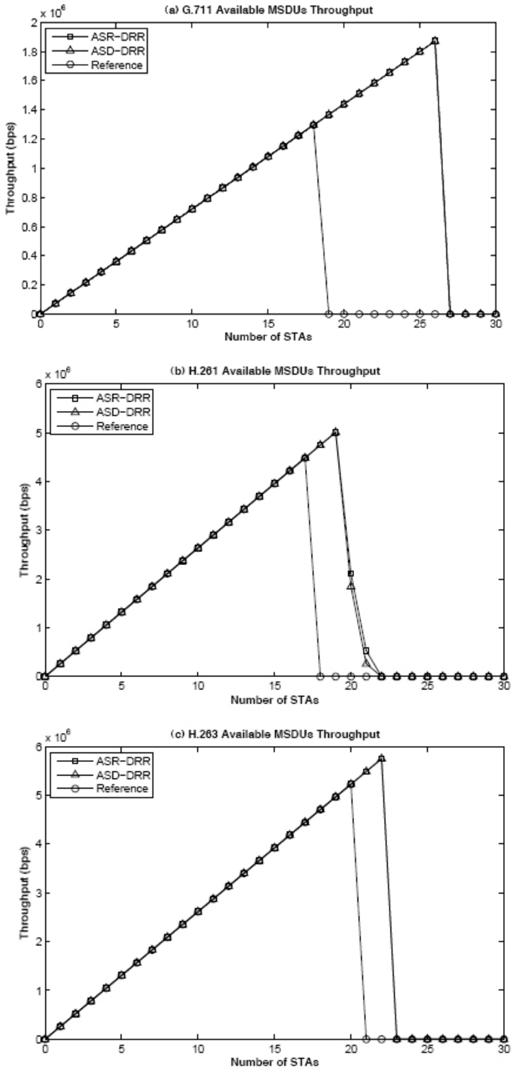
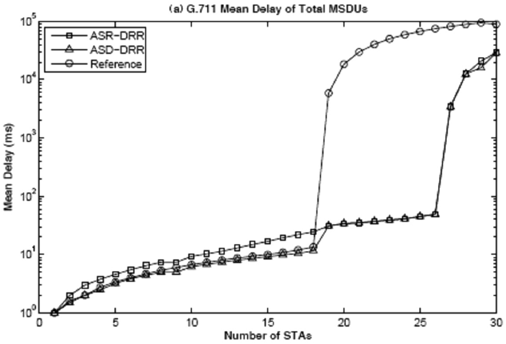


Fig. 6. Throughput of Available MSDUs (a) G.711 voice, (b) H.261 video, (c) H.263 video

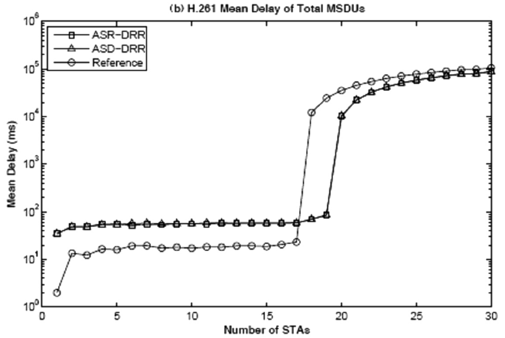
proposed schedulers showed better performance in terms of total throughput. There is a clear gap between the number of admissible STAs of the reference scheduler and the proposed schedulers.

Next we measured the schedulers' mean delays. The mean delay for any scheduler is strictly less than 10^2 milliseconds, under condition that the network is operated with admissible number of STAs. The delay, however, increases dramatically once there are more STAs than maximum admissible number of STAs. The proposed schedulers have larger number of admissible STAs, therefore show better performances. It is depicted in Figure 7.

(a) G.711 Mean Delay of Total MSDUs



(b) H.261 Mean Delay of Total MSDUs



(c) H.263 Mean Delay of Total MSDUs

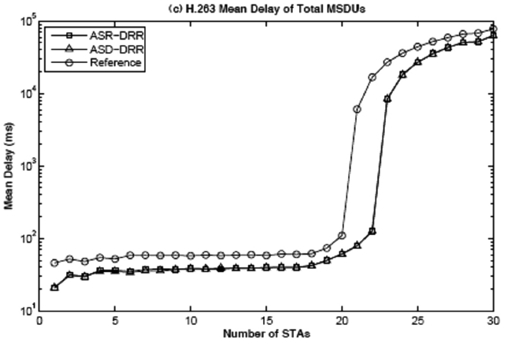


Fig. 7. Mean Delay of total MSDUs (a) G.711 voice, (b) H.261 video, (c) H.263 video

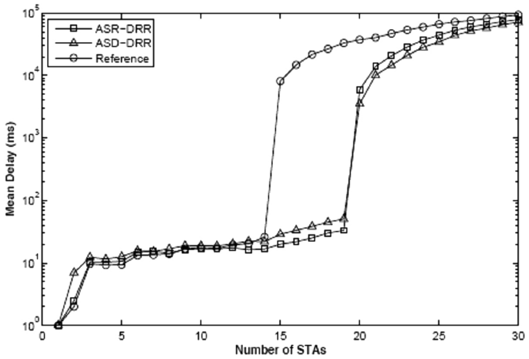


Fig. 8. Mean Delay of total MSDUs with mixed traffic flow

Finally, we simulated the network environment that three kinds of traffics were all mixed. We observed the delay of the total traffic and the delay of each type of traffic with each scheduler. In Figure 8, it is depicted the mean delay of the total traffic by each scheduler. We confirmed the proposed schedulers offer service to more STAs than the existing reference.

In summary, it was shown that ASR-DRR and ASD-DRR outperform Reference scheduler, in term of attained throughput, for both voice and video traffic, with Reference being able to handle relatively light traffic load. ASR-DRR and ASD-DRR, the proposed schedulers allocate TXOP by notifying the queued amount of data immediately. In addition, by reducing an unnecessary allocation of TXOP, more STAs are able to use media at the same time.

Ⅴ. Conclusion

This research has tackled the QoS issue of the WLAN, which is considered to be the weakest spot of the WLAN. During uplink scheduling AP can utilize immediate and accurate information given from STAs. Wireless uplink scheduling can approximate the schedulers (WFQ, DRR, etc.) in wired networks. We can expect tight delay bound and less delay jitter. The proposed schedulers show a great improvement over existing technologies, by introducing a rather simple idea of leveraging wired schedulers' concepts. The complexity has not been grown at all. The first scheduler, ASR-DRR, may requires a slight modification to the IEEE 802.11e standard.

The second one, however, needs not any modification to the standard and may be implemented with only a firmware upgrade for existing WLAN chipsets. A true wired-equivalent QoS scheduler has been suggested.

References

〔1〕 IEEE Std. 802.11, "IEEE standard for wireless LAN medium access control (MAC) and physical layer (PHY) specifications," Nov. 1997.

〔2〕 IEEE Std. 802.11e-2005, "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications Amendment 8: medium access control (MAC) quality of service enhancements," Nov. 2005.

〔3〕 S. Magold, S. Choi, G. R. Hiertz, O. Klein, and B. Walke, "ANALYSIS OF IEEE 802.11E FOR QOS SUPPORT IN WIRELESS LANS," IEEE Wireless Communications, December 2003.

〔4〕 Daquing Gu, jinyun Zhang, "QoS Enhancement in IEEE082.11 Wireless Local Area Networks," IEEE Communications Magazine, pp.120-123, June 2003.

〔5〕 Quang Ni, Lamia Romdhani, Thierry Turletti, "A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN," Wireless Communications and Mobile Computing, Wiley. Volume 4, Issue 5, pp.547-566, 2004

〔6〕 Deyun Gao, Jianfei Cai, King Ngi Ngan, "Admission Control in IEEE 802.11e Wireless LANs," IEEE Network, July/August 2005.

〔7〕 P. Ansel, Q. Ni, and T. Turletti, "FHCF: A simple and efficient scheduling scheme for IEEE 802.11e wireless LAN," ACM/Kluwer J.Mobile Networks Applications (MONET), Special Issue Modeling Opt., Vol.11, No.3, pp. 391-403, June 2006.

〔8〕 M. M. Rashid, E. Hossain, and V. K. Bhargava, "Queueing Analysis of 802.11e HCCA with Variable Bit Rate Traffic," in Proc. IEEE ICC'06, Istanbul, Turkey, June 2006.

〔9〕 Dimitris Skyrianoglou, Nikos Passas, and Apostolis K. Salkintzis, "Support of IP QoS over Wireless LANs" Vehicular Technology Conference, 2004. VTC 2004-Spring.

〔10〕 Dimitris Skyrianoglou, Nikos Passas, and Apostolis K. Salkintzis, "ARROW: An Efficient Traffic Scheduling Algorithm for IEEE 802.11e HCCA", IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, Vol.5, No. 12, DECEMBER 2006.

〔11〕 A. Grilo, M. Macedo, and M. Nunes, "A scheduling algorithm for QoS support in IEEE 802.11e networks," IEEE Wireless Commun. Mag., pp.36-43, June 2003.

〔12〕 Inanc Inan, Feyza Keceli, and Ender Ayanoglu, "An Adaptive Multimedia QoS Scheduler for 802.11e Wireless LANs" IEEE ICC 2006.

〔13〕 Gennaro Boggia, Pietro Camarda, Luigi Alfredo Grieco, and Saverio Mascolo, "Feedback-Based Control for Providing Real-Time Services With the 802.11e MAC", IEEE/ACM TRANSACTIONS ON NETWORKING, Vol.15, No.2, APRIL 2007.

〔14〕 Mohammad M. Rashid, Ekram Hossain, and Vijay K. Bhargava, "HCCA Scheduler Design for Guaranteed QoS in IEEE 802.11e Based WLANs" IEEE WCNC 2007.

〔15〕 Yuhi Higuchi, Augusto Foronda, Chikara Ohta, Masahiko Yoshimoto, and Yoji Okada, "Delay Guarantee and Service Interval Optimization for HCCA in IEEE 802.11e WLANs" IEEE WCNC 2007.

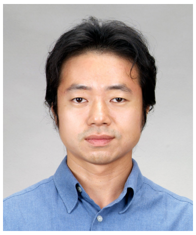
〔16〕 M. Eiger, M. Elaoud, and R. Merera, (Toshiba America Inc. and Telcordia Inc.), "Latency-Aware Service Opportunity Window based (LASO) scheduling", Patent Application US 2007/0036116 A1, published Feb. 15, 2007.

〔17〕 M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round-robin," IEEE/ACM Trans. Networking, Vol.4, No.3, pp.375--385, June 1996.

〔18〕 IEEE 802.11e HCCA simulation using the Network Simulator 2. 〔Online〕 Available : http://info.iet.unipi.it/~cng/ns2hcca/

〔19〕 MPEG-4 and H.263 Video Traces for Network Performance Evaluation. 〔Online〕Available : http://www.tkn.tu-berlin.de/research/trace/trace.html

정 진 우 (Jinoo Joung) 정회원



1992년 KAIST 전기전자공학과 공학사
1997년 Polytechnic Univ., NY, USA, 공학박사 (Ph.D.)
1997년~2001년 삼성전자 중앙연구소
2001년~2005년 삼성종합기술원
2005년~현재 상명대학교 컴퓨터과학부
<관심분야> 네트워크, 유무선 통신, 임베디드 시스템

김 종 호 (Jongho Kim) 준회원

2007년 상명대학교 소프트웨어학과 졸업
2007년~현재 상명대학교 컴퓨터과학과 석사과정 재학 중
<관심분야> 네트워크, 유무선 통신, 임베디드 시스템