Collision Reduction Random Access Using *m*-ary Split Algorithm in Wireless Access Network

You-Chang Ko[1], Eui-Seok Hwang[2], Jeong-Jae Won[3], Hyong-Woo Lee[4], and Choong-Ho Cho[5]

[1] LG Electronics Inc. UMTS Handset Lab.
ycko@lge.com
[2] Univ. of Washington Dept. of Electrical Engineering
eui@ee.wahsington.edu
[3] Univ. of British Columbia Dept. of Electrical and Computers Engineering
wonjj@ece.ubc.ca
[4] Korea Univ. Dept. of Electronics & Information Engineering
hwlee@korea.ac.kr
[5] Korea Univ. Dept. of Computer & Information Science
chcho@korea.ac.kr

Abstract. In the high performance radio access networks, the number of random access channels can be used for mobile stations to transmit their bandwidth requests in contention mode via multiple random access. In this paper a collision reduced random access scheme based on *m*-art split algorithm in the centralized medium access control protocol is presented. In this method the splitting algorithm is used in two fold. On one hand the whole random access channels are exclusively separated into two parts; one is for initial contention where only initial access mobile stations can be served and the other is for retransmit contention where the mobile stations whose initial random access was not successful can join for their retransmission. On the other hand the *m*-ary split algorithm applied in the retransmit contention area to resolve the collisions. By doing so the proposed scheme achieves considerably greater performance in terms of maximum throughput, mean access delay, and delay jitter, which is one of the important criteria for real-time traffic. Through numerical examples and computer simulations the effect of the various parameters of the algorithm, initial number of random access report(s) and the split size *m*, on the system performance is examined.

Key words: Wireless Access Network, Random Access, *m*-ary split algorithm, Multimedia QoS

1. Introduction

ETSI broadband radio access network (BRAN) project is currently developing standard for various types of wireless broadband access networks. High performance radio local area network type 2 (HiperLAN/2) is one of these proposed to operate in the 5GHz band which provides high-speed communications between mobile terminals and various broadband infrastructure networks [1,2]. The medium access control (MAC) protocol is based on dynamic time division multiple access/time division duplexing (TDMA/TDD). In a HiperLAN/2, fixed-sized messages are transmitted based on the scheduling performed by an access point (AP) or a central controller (CC) either in centralized mode or direct mode, respectively. Without loss of generality, we will only concern with the AP for the remainder of this paper.

Mobile terminals (MTs) report their uplink transmission requirements in Resource Request (RR) messages to the AP. The AP then allocates the resources according to the RRs and sends the allocation of the resources via Resource Grant (RG) messages. When an MT does not have an uplink transport channel allocated to transmit its RR, it transmits its RR in one of the available Random Channels (RCHs) in a contention mode. The outcome whether the MT's access attempt in a frame is successful or not is informed by the AP via Access Feedback Channel (ACH) in the subsequent frame. The number of RCHs may vary according to traffic load in the RCHs. A fixed-sized frame consists of downlink and uplink transport channels among which are the RCHs. Although an RCH slot is only 9 bytes long, because it uses the most robust modulation/coding (BPSK, 1/2 rate) and because the guard time between successive RCHs is usually not negligible [1,4], unnecessarily large number of RCHs can waste the channel resources. On the other hand, insufficient number of RCHs cause prolonged access delay and channel instability common to most contention schemes.

In [1], the problem of channel instability is partially solved using a version of the binary exponential backoff algorithm for retransmissions. Roughly, the contention window size (CW) of an MT in number of RCH slots is doubled after every collision experienced by the MT until the corresponding RR succeeds or the CW reaches the maximum size, thereby, spreading the retransmission attempts. The number of RCHs in a frame is recursively determined in [3] using the number of idle and successful RCHs in the previous frame. Compared with a scheme which uses a fixed number of RCHs, their scheme reduces access delay and possibly achieves higher maximum channel throughput. However, according to their simulations in [3], the maximum normalized channel throughput which is defined as the fraction of successful RCHs is still limited to $1/e$. Moreover, the access delay variation is rather large.

In this paper, we propose an m-ary split algorithm for dynamically adapting the number of RCHs in the current frame based on the observation of the RCHs in the previous frame. Essentially, by allocating m RCHs for each collided RCHs in addition to the RCHs allocated for RRs generated by newly arriving MTs, the maximum normalized throughput can be increased to approximately 0.43 and the delay variation, which is one of the important system criteria for real-time traffic are substantially reduced compared with the schemes using the binary backoff algorithm [1,3]. It is, however, necessary to modify the content of the ACH slot such that the outcome as to whether an RCH contains a collision or not is informed to the MTs instead of the outcome as to whether it contains a success or not. Also, retransmission algorithm is simplified by making the binary exponential backoff unnecessary.

This paper is organized as follows. In section 2, we describe the proposed m-ary split algorithm. Analysis of the mean access delay and throughput of the proposed scheme is presented in section 3. Since the analysis is exact only if the maximum number of RCHs allocated in a frame is not limited and does not provide the delay variance, we performed computer simulations. In section 4, we present numerical examples and computer simulation results demonstrating the superiority of the proposed algorithm compared with the previous algorithm based on the binary exponential backoff algorithm in terms of mean and variance of delay and throughput. We also examine the effect of the degree of split, m, and the number of RCHs allocated for initial access on the performance. Finally we end with conclusions in section 5.

2. *M*-ary Split Algorithm

As described in the introduction, the AP allocates a number of RCHs for random access transmission of Resource Requests (RRs) by mobile terminals (MTs). The set of RCHs which are placed at the end of a frame is referred to as ***RCH train***. In the proposed scheme an RCH train as shown in Fig. 1 consists of $N_a (\geq 1)$ RCHs for transmission of the RRs generated by newly arriving MTs and N_c RCHs for retransmission of the collided RRs attempted in the previous frame.

$$\underbrace{\boxed{\mathrm{R}_1}\boxed{\cdots}\boxed{R_{N_a}}}_{N_a}\;\underbrace{\boxed{R_{N_a+1}}\boxed{\cdots}\boxed{\mathrm{R}_i}\boxed{\cdots}\boxed{R_{N_a+N_c}}}_{N_c}$$

Fig. 1. The RCH train structure of the proposed algorithm

According to [4], which evaluates throughput of HiperLAN/2 MAC protocol taking all guard time spaces into account, the maximum available number of RCHs in an AP would be much less than those defined in [1]. We, therefore, limit the maximum RCH train size to R_{MAX}. Then, the number of RCHs in the $(t+1)^{th}$ frame, $r(t+1)$, is given by equation 1.

$$r(t+1) = \min[\, N_a + m * c(t),\ R_{MAX}], \qquad \textbf{(1)}$$

where $m (\geq 2)$ is the degree of split and $c(t)$ is the number of collided RCHs in the t^{th} frame. For a simple description of the proposed algorithm, we will assume that $R_{MAX} - N_a$ is divisible by m. The algorithm is as follows.

1. An RR arriving during the t^{th} frame will be transmitted in one of the N_a RCHs chosen at random in the subsequent frame, $(t+1)^{th}$ frame.
2. An MT which transmitted an RR in the t^{th} frame is notified of the outcome of its attempt in the ACH. Moreover, it is able to find the number of RCHs involved in a collision prior to the RCH in which it transmitted its RR. If its contention is unsuccessful one of the followings are performed.
   a) If $N_a + m(\theta_i + 1) \leq R_{MAX}$, it retransmits its RR during the $(t+1)^{th}$ frame in one of the RCHs chosen at random in $[R_{N_a + m\cdot\theta_i + 1}, R_{N_a + m(\theta_i + 1)}]$, where i is the index of RCH which the MT has accessed in the previous MAC frame and θ_i is the number of collided RCH indices which are less than i, $\theta_i \geq 0$.

b) Otherwise, it attempts retransmission in one of the N_a RCHs during the (t+1+δ) frame, where δ is a uniform random variate whose interval is chosen appropriately depending on the traffic intensity.

In the following equations RF(i) is the retransmission function which determines its position of RCH in the subsequent frames.

Initial attempt: random access within [1, N_a] **(2)**

RF(i) = random access within $[R_{N_a+m\cdot\theta_i+1}, R_{N_a+m(\theta_i+1)}]$ **(3)**

if $N_a + m(\theta_i + 1) \le R_{MAX}$ or,

random access within [1, N_a] after random delay

if $N_a + m(\theta_i + 1) > R_{MAX}$

Figure 2 shows an example how the proposed method works based on m-ary split algorithm with N_a=2, m=3, and R_{MAX}=11. In the figure S, C, and I denote success, collision, and idle respectively. Suppose that collisions occur at R_1, R_3, R_5, and R_6 in the t^{th} MAC frame and Q_1, Q_3, Q_5, and Q_6 are the set of the MTs which have collided at R_1, R_3, R_5, and R_6 respectively. Then $|Q_i| \ge 2$ and the MTs in Q_1, Q_3, Q_5, and Q_6 in the t^{th} MAC frame will contend again in the $(t+1)^{th}$ MAC frame within N_c area. R^m_i is the split size in $(t+1)^{th}$ MAC frame for the MTs in Q_i. The MTs in set Q_6 cannot retransmit their RRs within the $(t+1)^{th}$ MAC frame due to the limitation of the maximum RCH size. They retransmit their RRs in one of the N_a RCHs in the $(t+1+\delta)^{th}$ MAC frame. Their RRs are treated as if they are newly arrived RR in the $(t+\delta)^{th}$ MAC frame

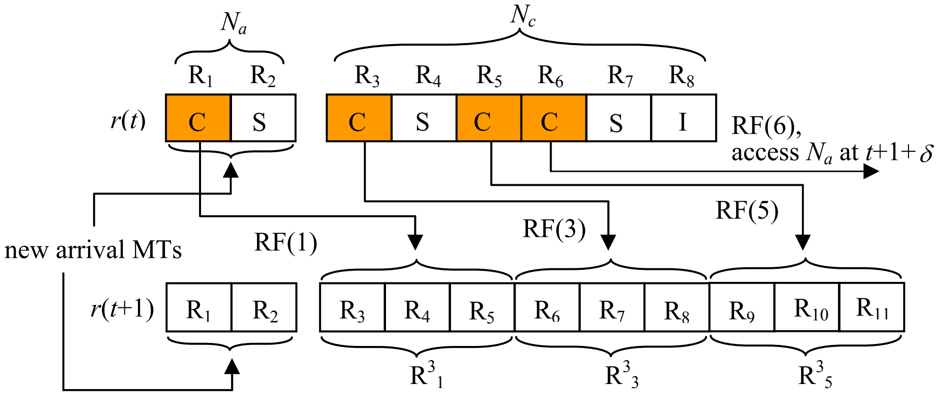

Fig. 2. RCH structure in the t^{th} and the $(t+1)^{th}$ MAC frame for N_a=2, m=3, R_{MAX}=11.

3. Analysis of the Mean Access Delay and Throughput

The following notations and assumptions are used.

1. Let A_t denote the number of RR packets arriving during the t^{th} frame. We assume that the sequence of A_t's are independent and identically distributed (iid) random variables. Although it is not essential for the analysis, we assumed that A_t is Poisson distributed with mean λ_t RR packets/frame. λ_t denotes the offered load in the AP's perspective whereas λ represents the offered load by each MT. As stated in the previous section, all the newly arriving RRs are transmitted in an RCH during the subsequent frame.
2. For the sake of tractability, we assume that there is no limit on the number of RCHs in a frame. That is, $R_{MAX}=\infty$. This restriction will be relaxed in the computer simulations.
3. Let $D(n)$ denote the sum of mean delays in number of frames experienced by the RRs conditioned on that they belong to a group of n RRs initially transmitted in one of the N_a RCHs. We note that $D(0)=D(1)=0$.
4. Let $N(n)$ denote the mean of total number of RCHs required to resolve a collision involving n RRs. We note that $N(0)=N(1)=0$.

3.1. Mean Access Delay

We have the following recursive equation for $n \geq 2$.

$$D(n) = \sum_{s_1+\ldots+s_m=n} \left[\alpha(s_1, s_2, \ldots, s_m; n) \cdot \left\{ n + \sum_{i=1}^{m} D(s_i) \right\} \right].$$

Solving for $D(n)$ in the above equation, we obtain

$$D(n) = \frac{n + \sum_{0\leq s_1\leq n-1,\ldots,0\leq s_m\leq n-1} \left[\alpha(s_1, s_2, \ldots, s_m; n) \cdot \left\{ \sum_{i=1}^{m} D(s_i) \right\} \right]}{1 - m^{-n+1}}, \quad \text{for } n \geq 2.$$

where $\alpha(s_1, s_2, \ldots, s_m; n)$ is the multinomial probability function. That is, if n items are placed randomly among m bins, the probability that the first bin contains s_1 items, the second bin contains s_2 items, …, and the m^{th} bin contains s_m items is $\alpha(s_1, s_2, \ldots, s_m; n)$. We have

$$\alpha(s_1, s_2, \ldots, s_m; n) = \frac{n!}{s_1! \cdot s_2! \cdot \ldots \cdot s_m!} (p_1^{s_1} \cdot p_2^{s_2} \cdot \ldots \cdot p_k^{s_m}) \quad = \quad \frac{n! \cdot m^{-n}}{s_1! \cdot s_2! \cdot \ldots \cdot s_m!} \qquad (4)$$

Here, p_i is the probability that an RR chooses the i^{th} RCH among the m RCHs allocated for the group of RRs involved in a collision. Since an RR chooses one of the m RCHs at random, we have $p_i=1/m$, and the last equality results. We recursively obtain $D(n)$ (n=2,3,…) starting from the initial conditions $D(0)=D(1)=0$. We define, $\overline{D}$, the mean delay as the time from the first access attempt to the beginning of the frame in which the access is successful. Finally, we have

$$\overline{D}= \frac{N_a}{\lambda_t}\sum_{i=0}^{\infty} D(i)\cdot P(i), \quad \text{where } p(i)=e^{-\frac{\lambda_t}{N_a}}\cdot\frac{\left(\frac{\lambda_t}{N_a}\right)^i}{i!},\ i=0,1,2,...$$

In the above equation $p(i)$ is the probability that i RRs attempt transmission in one of the N_a RCHs

3.2. Throughput

Recalling that $N(n)$ is defined as the additional RCHs needed to resolve a collision involving n RRs, we have $N(0)=N(1)=0$. A similar recursive equation can be obtained for $N(n)$ as that for $D(n)$.

$$N(n)=\sum\left[\alpha(s_1,s_2,...,s_m\,;n)\cdot\left\{m+\sum_{i=1}^{m}N(s_i)\right\}\right].$$

Solving for $N(n)$ using the above equation, we have for $n\geq 2$,

$$N(n)=\frac{m+\sum_{0\leq s_1\leq n-1,...,0\leq s_m\leq n-1}\left[\alpha(s_1,s_2,...,s_m\,;n)\cdot\left\{\sum_{i=1}^{m}N(s_i)\right\}\right]}{1-m^{-n+1}}. \tag{5}$$

We define, $\overline{N}$, as the mean number of RCHs in an RCH train. Then, we have

$$\overline{N}= N_a\cdot\left(1+\sum_{i=0}^{\infty}D(i)\cdot P(i)\right), \quad \text{where } p(i)=e^{-\frac{\lambda_t}{N_a}}\cdot\frac{\left(\frac{\lambda_t}{N_a}\right)^i}{i!},\ i=0,1,2,... \tag{6}$$

We will use the resource utilization (normalized throughput), ρ, as defined in equation 11 of reference [3]. We, finally, obtain the normalized throughput as

$$\rho = \frac{total\ number\ of\ successful\ access\ attempts}{total\ number\ of\ allocated\ RCHs} = \frac{\lambda_t}{N} \tag{7}$$

4. Numerical Results and Simulations

In this section, we present the numerical results of the proposed method in terms of mean access delay, delay variance, and the normalized throughput. The results here are based on the assumptions that transmission failures due to wireless channel errors are negligible and can be ignored and RR arrivals obey Poisson distribution with mean λ packets/frame/MT. We also assumed that the number of MTs belonging to an AP is 50.

Figure 3 and Figure 4 show the effect of varying N_a and/or RCH split size, m, on the normalized throughput. As seen in the figures the proposed algorithm can achieve maximum normalized throughput of about 0.43 which is considerably higher than about 0.36 as reported in the previous work [3]. In Figure 3, we fix N_a=1 or N_a=5 and vary the split size from 2 to 5. It is observed that the maximum utilization does not improve beyond $m=2$. However, compared with the case when $m=2$ the decrease in the throughput when the offered load is greater than the optimum load at which the maximum throughput is achieved is more gradual when m is greater than 2. We note that the gradual decrease in throughput has a desirable effect on channel stability.

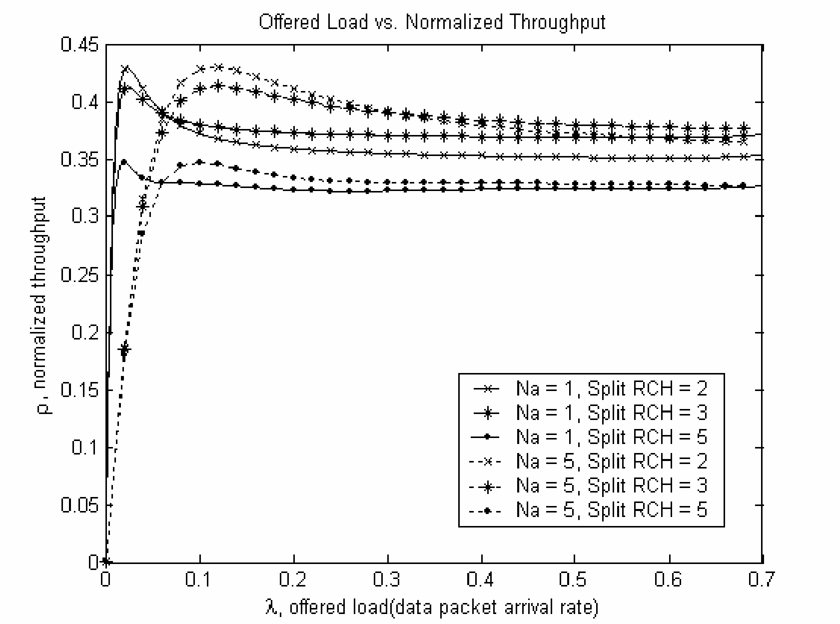

Fig. 3. Offered load vs. resource utilization(throughput) as m varies(numerical results)

The results also show that the maximum throughput does not depend on the value of N_a; however, the peak of the throughput curves occur at higher offered load when N_a is larger. At a region where the offered traffic is high, increasing the number of initial RCHs, N_a, is more effective for enhancing system performance than increasing RCH split size, *m*. Although the system suffers from performance degradation in terms of maximum normalized throughput as RCH size increases in general, the system throughput becomes greater after a certain offered load for the case of split sizes 2 and 3. For example, when N_a=1, this threshold of the offered load is approximately 0.07, while it is about 0.32 when N_a=5.

In Figure 4, we fix *m*=2 or *m*=5 and vary the N_a from 1 to 5. We have a similar observation as in Figure 3.

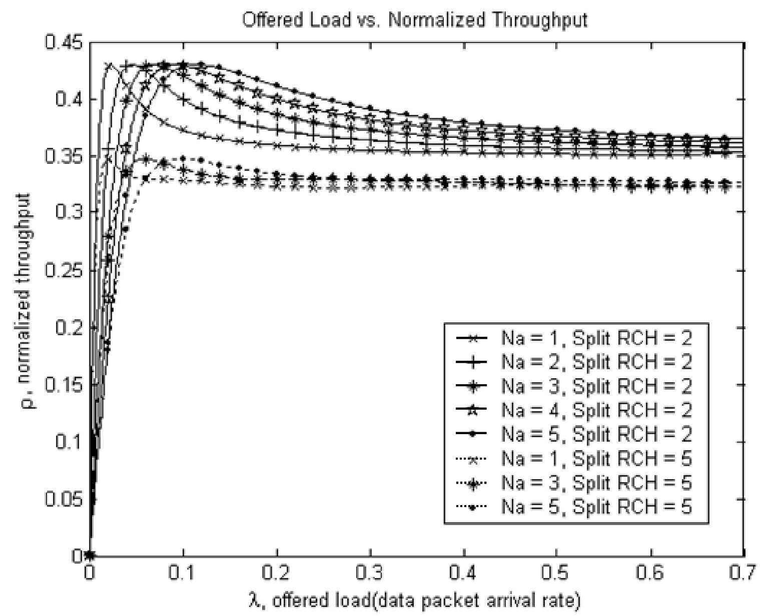

Fig. 4. Offered load vs. resource utilization(throughput) as N_a varies(numerical results)

Figure 5 presents the mean access delay in number of MAC frames required for the users until they successfully transmit their RRs. With various RCH split size, *m,* from 2 to 5, we plot the mean access delay vs. offered load, λ, when initial number of RCHs, N_a is between 1 and 5. The numerical results show that the mean access delay can be decreased by increasing the split size *m* as well as the initial number of slots N_a. By comparing the 5th curve(N_a=1 & Split RCH=5) and the 6th curve(N_a=5 & Split RCH=2), we note that appropriate choice of parameter values (N_a and *m*) has a significant influence on the delay vs. load performance. We can take advantage of adaptively varying N_a and *m*. Experiments also show that the biggest mean delay decrease can be achieved when *m* is increased from two to three. However, increasing *m* size to reduce mean delay would not always be available due to the fact that the number of available RCHs in MAC frame could be limited by the system parameters such as guard times which can be imposed in many places in a MAC frame[4].

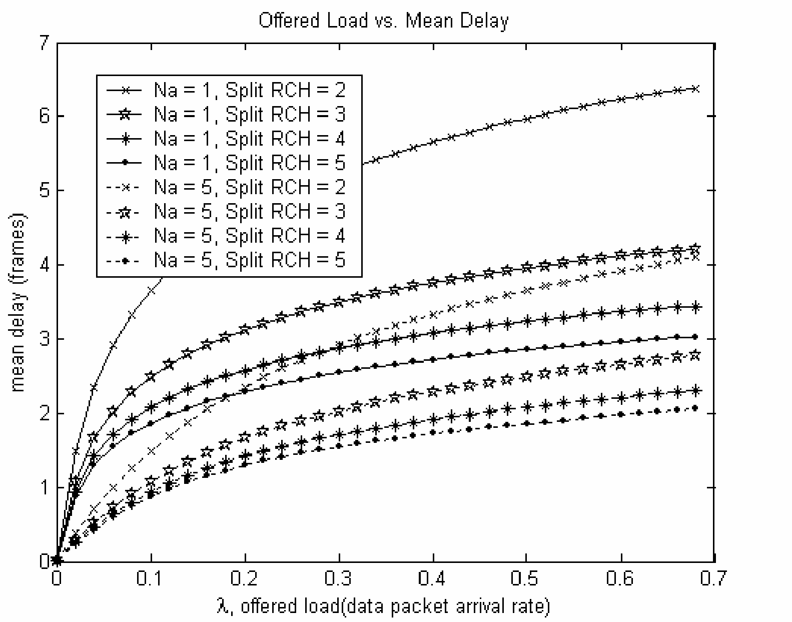

Fig. 5. Offered load vs. mean delay(numerical results)

The graphs in Figure 6 shows mean delay variance in number of MAC frames and are plotted by computer simulation when N_a is increased from 1 to 5 as m varies. As seen in the graphs, by increasing m the mean delay variance can be significantly reduced in general. A similar behavior as in mean delay graphs in Figure 5 is observed such that increasing m from 2 to 3 is most significant in decreasing the mean delay variance. Simulation results also show that increasing N_a is still effective to reduce mean delay variance for arbitrary m. However, the amount of decrease in the mean delay variance by increasing N_a reduces as offered load increases. When referred to [5], which shows delay variance simulation result of the scheme proposed by [3]; it requires approximately 150 frames when offer load is about 0.3, our proposed method substantially reduce the jitter time which is one of important criteria for real-time traffic.

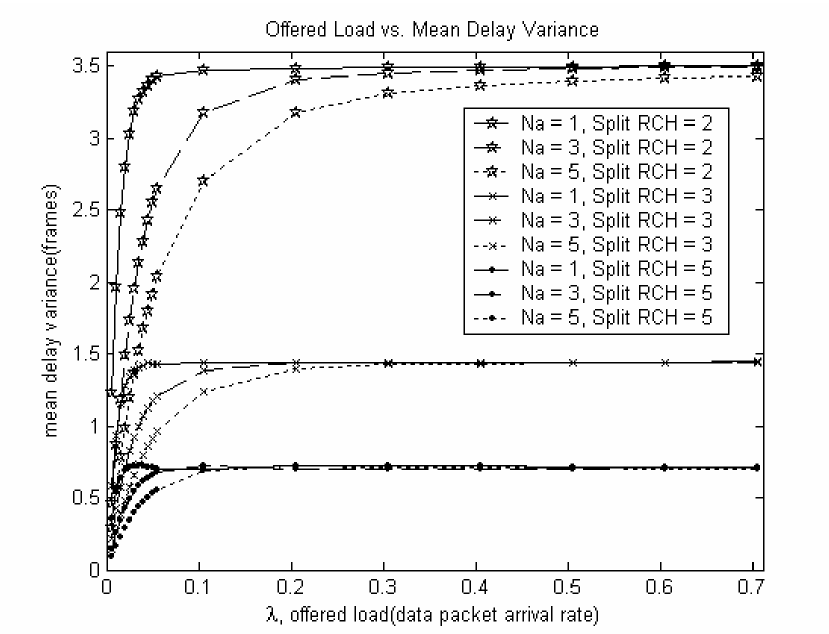

Fig. 6. Offered load vs. mean delay variance(simulation results)

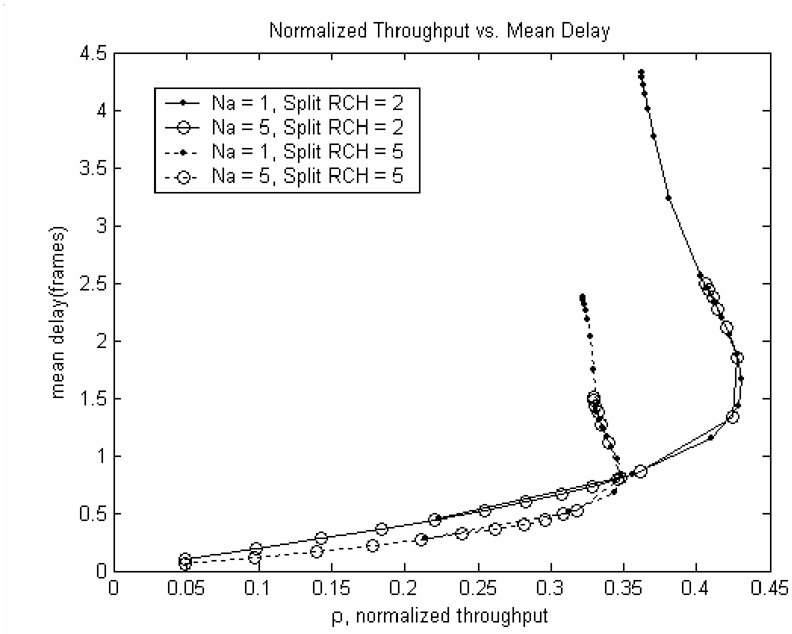

Fig. 7. Resource utilization(throughput) vs. mean delay(simulation results)

In the normalized throughput's perspective Figure 7 illustrates the mean delay characteristics in MAC frames by computer simulation. As seen the two cases when m=2 and m=5, delays are at most less than 4.5 and 2.5 frames respectively. We see generally that the more N_a or m size is the lower mean delay is. We also note the delay difference between minimum and maximum N_a size for each m size; when m=2 there are approximately 2 frames difference between the end of circle mark(N_a=5) and the end of dot mark(N_a=1), whereas when m=5 there are approximately 1 frames difference between the end of circle mark(N_a=5) and the end of dot mark(N_a=1). When compared to the results in [3], with the proposed method when N_a=m=5 we can achieve nearly 10 times less delay frames while preserving the same throughput. Moreover the proposed method can increase throughput up to approximately 0.44 when m=2, which is actually 26% increase compared to the previous method in [3], at the expense of slightly additional delays but remaining more than 6 times less delay frames still.

5. Conclusions

In this paper, we proposed a new method based on m-ary split algorithm to control the number of RCHs dynamically in an RCH train of a HiperLAN/2 MAC frame. The structure of RCH train consists of N_a RCHs for those MTs who access the system for the first time and N_c RCHs for those MTs who retransmit their request after experiencing collision in the previous frame. By letting N_c to be m times number of collided RCHs in the previous slot, we were able to increase the maximum normalized throughput. We also analyzed the mean access delay, delay variance and the normalized throughput. Through numerical examples using mathematical analyses and computer simulations with various values of N_a and the RCH split size, m, we have shown that the proposed method can achieve much better performance in terms of throughput and delay performance compared with the previous work in [3], and that the proposed scheme is suitable for guaranting QoS for not only non-real-time traffic but also real-time traffic, that is, QoS of multi-media traffic in wireless LAN. We observed that as the traffic load increases increasing N_a area is generally more effective than increasing m. It was also observed that the best performance is achieved with m=2 or 3 regardless of N_a and that it is effective to reduce mean delay frame and its variance by increasing either m or N_a. It will be interesting to find an adaptive scheme for switching the value of m as well as for balancing the size of N_a and m depending on the observed traffic intensity for optimum performance.

Although the proposed algorithm is proposed to Hiperlan2 system in this paper to compare with the previous applied method in [3] it can be able to apply to most of wirelss access Networks where resources are allocated by MT's random access such as IEEE802.16, IEEE802.20, and so on.

References

[1] HIPERLAN Type2; Data Link Control(DLC) Layer; Part1: Basic Data Transport Functions, Broadband Radio Access Networks(BRAN), ETSI TS 101 761-1 V1.3.1(2001-12).

[2] HIPERLAN Type2; Packet based Convergence Layer; Part1: Common Part, Broadband Radio Access Networks(BRAN), ETSI TS 101 493-1 V1.1.1(2000-4).
[3] Gyung-Ho Hwang and Dong-Ho Cho, “Adaptive Random Channel Allocation Scheme in HIPERLAN Type2”, IEEE Communications Letters, Vol.6, No.1 Jan 2002.
[4] You-Chang Ko, Eui-Seok Hwang, Jeong-Shik Dong, Hyong-Woo Lee and Choong-Ho Cho, “Throughput Analysis of ETSI BRAN HIPERLAN/2 MAC Protocol taking Guard Timing Spaces into Consideration”, 9th International Conference on Personal Wireless Communications 2004, Deft Netherlands, September 2004.
[5] Eui-Seok Hwang, Jeong-Jae Won, You-Chang Ko, Hyong-Woo Lee and Choong-Ho Cho, “Random Channel Allocation Scheme in HIPERLAN/2”, to appear Pacific-Rim Conference on Multimedia 2004, Tokyo Japan, November 2004.
[6] Bernhard H. Walke, et al. “IP over Wireless Mobile ATM-Guaranteed Wireless QoS by HiperLAN/2”, Proceedings of the IEEE, vol.89, no.1, JANUARY 2001.
[7] Andreas Hettich, Arndt Kadelka, “Performance Evaluation of the MAC Protocol of the ETSI BRAN HIPERLAN/2 Standard”, Proc. European. Wireless’99, Munich Germany, Oct. 1999.
[8] Andreas Hettich, Matthias Schrother, “IEEE 802.11 or ETSI BRAN HIPERLAN/2: Who will win the race for a high speed wireless LAN standard?”, Proc. European. Wireless’99, Munich Germany, Oct. 1999.
[9] Alberto Leon-Garcia, “Probability and Random Processes for Electrical Engineering”, Chapter 2 Basic Concept and Probability Theory, Addison-Wesley, 1994 Second Edition.