

HIPERLAN Type 2에서 Split 알고리즘에 기반한 랜덤채널 할당 기법

정희원 황 의 석*, 고 유 창**, 이 승 규***, 윤 철 식***, 이 형 우****, 조 충 호*

Random Channel Allocation Scheme Based on Split Algorithm in HIPERLAN 2

Eui-Seok Hwang*, You-Chang Ko**, Seung-Que LEE***, Chul-Sik Yoon***
Hyong-Woo Lee****, Choong-Ho Cho* *Regular Members*

요 약

HIPERLAN/2(High PERFORMANCE Local Area Network Type2)에서의 MAC 프로토콜은 TDMA/TDD에 기반을 두고 있으며 무선 채널은 Access Point(AP)에 의해서 할당되어진다. 상위 방향의 데이터 전송이 필요한 단말은 랜덤채널(RCH)을 통하여 AP에게 자원을 요청하게 된다. 각 MAC 프레임마다 RCH 수는 변경될 수 있는데, 이때 필요 이상의 RCH 수는 무선 채널의 낭비를 초래하게 된다. 또한 필요 이하의 RCH 수는 단말들간의 충돌을 야기하여 단말들이 AP에 접속하는 시간을 지연시킬 수 있다. 따라서 RCH 수는 트래픽 양에 따라 적절히 변화되는 것이 바람직하다. 본 논문에서는 Split 알고리즘에 기반을 두어 RCH 수를 제어하는 기법을 제안하였다. 제안된 기법에 대하여 시뮬레이션 및 수학적 분석 모델을 통하여 고정된 수의 RCH를 사용하는 방식과 기존에 제안된 방식에 비하여 처리율, 지연 및 지연에 대한 분산에서 성능이 향상됨을 보여준다.

Key Words : HIPERLAN Type 2, Access Point(AP), Random Channel(RCH), Multiple Access Control(MAC), Split Algorithm

ABSTRACT

The HIPERLAN/2(High PERFORMANCE Local Area Network Type2) is one of the wireless LAN standards for providing raw data rates of up to 54 Mbps. The MAC protocol of HIPERLAN/2 is based on TDMA/TDD, and resources in one MAC frame can be allocated dynamically by Access Point(AP). The random channel(RCH) is defined for the purpose of giving a mobile terminal the opportunity to request transmission resources in the uplink MAC frames. It is desirable that the number of RCHs is dynamically adapted by the AP depending on the current traffic situation. Allocation of excessive RCHs may waste radio resources and insufficient RCHs compared to traffic loads may result in many collisions in access attempts. We propose an RCH allocation scheme based on split algorithm in HIPERLAN/2. The simulation and analytic results show that the proposed scheme achieves a higher channel throughput, lower access delay and delay jitter than previously proposed RCH allocation schemes.

* 고려대학교 컴퓨터 정보학과({ushwang,chcho}@korea.ac.kr), **** 고려대학교 전자 및 정보공학부(hwlee@korea.ac.kr),

** LG 전자(ycko@lg.com), ***한국전자통신연구원({sqlee,csyoon}@etri.re.kr)

논문번호: #030200-0513, 접수일자: 2003년 5월 14일

※ 본 논문은 한국과학재단 목적기초연구(R01-2002-000-00489-0(2002))지원 및 2003년도 고려대학교 박사수료 후 연수과정 연구비 지원에 의하여 수행되었음.

I. 서론

현재 5GHz 대역폭에서 50Mbps 이상을 지원하기 위한 무선 LAN 표준으로는 ETSI BRAN의 HIPERLAN/2 와 IEEE의 802.11a가 제시되고 있다. 두 방식의 가장 큰 차이점은 MAC 프로토콜에 있다[3-7]. 802.11a는 CSMA/CA에 기반을 둔 분산 방식의 MAC 프로토콜을 이용하는 반면 HIPERLAN/2에서는 2ms의 MAC 프레임[2]을 이용한 TDMA/TDD 방식에 기반을 두고 있다. 각각의 MAC 프레임은 Access Point(AP)에 의해서 구성되어진다. 상위 방향의 데이터 전송이 필요한 단말은 랜덤채널(RCH)를 이용하여 AP에게 자원을 요청한다. 이때 RCH의 수는 프레임 마다 변경될 수 있다. 즉, 필요 이상의 RCH 수는 단말간의 충돌을 완화할 수 있는 반면 무선 채널의 낭비를 초래한다. 필요 이하의 RCH 수는 단말들간의 충돌을 야기하여 단말들이 AP에 접속하는 시간을 지연시키고 채널 처리율의 급격한 감소를 일으킨다. 따라서 RCH 수는 트래픽 양에 따라 효과적으로 변하는 것이 바람직하다. 고정된 개수의 RCH를 사용하는 방식에서는 최대 35%의 처리율을 보이고 있다. [1]에서는 이전 프레임에서 충돌이 발생한 RCH 수와 성공이 발생한 RCH 수의 차를 다음 프레임에 더해 주는 방식으로 RCH 수를 변화 시켰다. 그 결과 처리율이 최대 37%까지 증가하였다. 본 논문에서는 같은 RCH를 액세스하여 충돌이 발생한 단말들에 대하여 각각의 단말이 성공적으로 서비스를 받을 수 있을 때까지 작은 그룹으로 나누어져서 서로 다른 RCH를 액세스하도록 하는 Split 알고리즘[8]을 이용하여 RCH 수를 제어한다. 결과적으로 처리율이 최대 44% 까지 증가하였다. 또한 기존의 방식에 비하여 지연시간이 줄어들고 지연시간의 분산이 현격히 감소하여 실시간 트래픽을 전송하는데 적합함을 볼 수 있었다. 본 논문에서는 제안된 알고리즘에 대하여 시뮬레이션 및 수학적 분석 모델을 통하여 성능을 분석하였다. 본 논문의 구성은 다음과 같다. 2장에서는 HIPERLAN/2의 MAC 프로토콜 및 기존연구를 살펴보고 3장에서 새로운 랜덤채널 할당 기법을 제안하였다. 4장에서는 제안된 모델에 대한 수학적 분석을 제시하고 5장에서는 수학적 분석 및 시뮬레이션 결과를 통하여 제안된 기법의 성능을 분석하였다. 그리고 마지막으로 6장에서 결론을 맺는다.

II. HIPERLAN/2의 MAC 프로토콜

HIPERLAN/2의 하나의 MAC 프레임의 길이는 2ms이고 그림 1과 같은 구조를 갖는다.

- BCH(Broadband CHannel: downlink) : 단말에게 필요한 정보 제공(power level, FCH와 RCH 시작 위치 및 길이 등)
- FCH(Frame control CHannel: downlink) : MAC 프레임에서 각 단말에 대한 자원 할당 정보 제공
- ACH(Access feedback CHannel: downlink) : 이전 프레임에서 RCH 액세스 시도에 대한 성공 여부 전송
- RCH(Random access CHannel) : 무선 LAN에 접속을 시도하는 단말의 자원 할당 요구 사항 전달
- DL(DownLink), UL(UpLink), DiL(Direct Link)

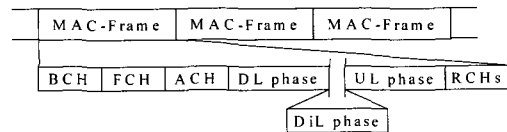


그림 1 MAC 프레임 구조

HIPERLAN/2는 AP에서 모든 전송을 관리하고 자원 할당을 스케줄링하는 중앙제어방식을 택하고 있다. 상위 방향의 데이터 전송이 필요한 단말은 RCH를 통하여 자원 요청을 할 수 있다. 그리고 이전 프레임에서 자원 요청을 시도하였다면 ACH를 통하여 그 결과를 확인한다.

RCH에 대한 액세스는 각 단말에서 관리하는 Contention Window(CW)에 의해서 제어된다. 각 단말은 재전송 횟수 a에 의해서 CW_a의 크기를 결정하며 그 방식은 다음과 같다.[2]

· initial attempt : $a = 0, CW_0 = n$ (1)

· retransmission : $a \geq 1,$

$$CW_a = \begin{cases} 256, & 2^a \geq 256 \\ 2^a, & n < 2^a \leq 256 \\ n, & n \geq 2^a \end{cases} \quad (2)$$

- CW_a : Contention Window
- a : 액세스 시도 횟수
- n : 현재 프레임의 RCH 수

단말은 CW_a 를 결정한 후 $[1, CW_a]$ 사이의 균일분포 랜덤정수 r 을 선택한다. 선택한 r 은 단말이 액세스를 시도할 RCH의 번호가 된다. 액세스할 RCH의 번호는 ACH를 통하여 액세스 실패를 수신한 후 그 이후의 첫 번째 RCH에서부터 계산을 시작한다. RCH의 개수가 r 에 도달하기 이전에 단말은 RCH를 액세스 하지 않고 r 번째 RCH에 재전송을 시도한다.

AP는 MAC 프레임의 구성하는데 있어서 RCH의 개수를 프레임마다 변경할 수 있다. RCH의 개수는 하나의 MAC 프레임에서 하나 이상 31개 이하이다 [2]. 과도한 RCH는 단말간의 충돌을 완화할 수 있는 반면 채널의 낭비를 초래할 수 있다. 또한 RCH의 개수가 적으면 단말간의 빈번한 충돌로 인하여 지연이 발생할 수 있다. 따라서 트래픽 양에 따라 채널의 낭비를 최소화 하면서 충돌을 줄일 수 있도록 RCH의 개수가 변경되는 것이 바람직하다. [1]에서는 이전 MAC 프레임의 RCH 액세스 결과에 따라 RCH 개수를 적응적으로 변화시켰다. 충돌이 발생한 RCH 수만큼 다음 프레임의 RCH의 수를 증가시키고 성공한 RCH 수만큼 다음 프레임의 RCH 수를 감소시키는 방법을 사용하였다. 만약 이전 프레임에서 RCH에 대한 액세스가 없었다면 RCH의 수를 하나 감소시켰다. 이와 같은 방법은 (식 3) ¹으로 나타내어 질 수 있다.

$$r(t+1) = r(t) + a(N_f(t) - N_s(t))(1 - I_{idle}(t)) - I_{idle}(t) \quad (3)$$

- $r(t)$: t 시간에서 할당된 RCH 수
- a : 가중치
- $N_f(t)$: t 시간에서 충돌이 발생한 RCH 수
- $N_s(t)$: t 시간에서 성공한 RCH 수
- $I_{idle}(t)$: t 시간에서 RCH에 대한 액세스가 없었다면 1, 그렇지 않다면 0

한 프레임에서 RCH의 수에 대한 범위를 최소한 개에서 최대 R_{MAX} 까지로 제한한다(식 4).

$$r(t+1) = \min\{\max\{r(t+1), 1\}, R_{MAX}\} \quad (4)$$

III. 새로운 랜덤채널 할당 기법

본 연구에서는 Split 알고리즘[8]에 기반을 두어 RCH의 수를 변경한다. 같은 RCH를 액세스하여 충돌이 발생한 단말들에 대하여 각각의 단말이 성공적으로 서비스를 받을 수 있을 때까지 작은 그룹으로 나누어져서 서로 다른 RCH를 액세스한다. 현재 프레임의 RCH 수는 새로운 단말을 위해 고정된 개수의 RCH와 이전 프레임에서 충돌이 발생한 단말을 위해 프레임마다 개수가 변하는 RCH의 합으로 구해질 수 있다(식 5). 이전 프레임에서 충돌이 발생한 하나의 RCH마다 현재 프레임에서는 두개의 RCH가 생성된다.

· $r(t+1) = N_a + 2 \times N_f(t)$ (5)

- $r(t)$: t 시간에서 할당된 RCH 수
- $N_f(t)$: t 시간에서 충돌이 발생한 RCH 수
- N_a : 새로이 상향 방향 채널 자원을 요청하는 단말을 위한 고정된 RCH 수

한 프레임에서 RCH 수에 대한 범위를 최소 N_a 에서 최대 R_{MAX} 까지로 제한한다.

$$r(t+1) = \min\{r(t+1), R_{MAX}\} \quad (6)$$

처음으로 RCH 액세스를 시도하는 단말은 새로운 단말을 위해 고정된 개수의 RCH 중에서 하나를 랜덤하게 선택하여 액세스를 시도한다(식 7). 이전 프레임에서 RCH 액세스에 실패한 단말은 몇 번째에서 충돌이 발생했는가의 정보를 바탕으로 액세스할 RCH를 결정한다. 즉, 충돌이 발생한 RCH 중 i 번째 RCH를 액세스한 단말은 새로운 단말을 위한 RCH를 제외한 RCH 중 $2 \times i - 1$ 또는 $2 \times i$ 번째 RCH를 랜덤하게 선택하여 액세스한다(식 8). 액세스할 RCH의 범위가 R_{MAX} 를 초과하는 단말은 프레임 단위로 랜덤하게 지연을 발생한 후 새로운 단말을 위한 RCH 중 하나를 랜덤하게 선택하여 액세스한다(식 9).

- Initial attempt : random access within $[1, N_a]$ (7)

- Retransmission : random access either $i \times 2 - 1 + N_a$ or $i \times 2 + N_a$ if $2 \times i + N_a \leq R_{MAX}$ (8)

- random access within $[1, N_a]$ after random delay if $2 \times i + N_a > R_{MAX}$ (9)

· i : 충돌이 발생한 RCH 위치

¹ 참고문헌 [1]의 식에서 $I_{idle}(t)-1$ 항은 $1 - I_{idle}(t)$ 로 바뀌어야 한다.

그림 2는 새로운 단말을 위한 RCH의 개수가 2 이고 R_{MAX} 가 8인 경우의 예를 보이고 있다. S는 성공을 C는 충돌을 의미한다. 앞선 MAC 프레임에서 RCH의 수가 8개이고 이중 첫 번째, 세 번째, 다섯 번째 그리고 여섯 번째 RCH에서 충돌이 발생 하였다면 다음 프레임에서의 RCH 개수는 새로운 단말을 위한 두 개의 RCH와 충돌이 발생한 네 개의 RCH로 인한 8개의 RCH의 합으로 총 10개가 되어야 한다. 그러나 R_{MAX} 값이 8이므로 8개가 된다. 이때 이전 프레임의 첫 번째 RCH에서 충돌이 발생한 단말은 다음 프레임에서 세 번째와 네 번째 RCH 중 하나를 액세스한다. 이전 프레임의 첫 번째 RCH는 충돌이 발생한 네 개의 RCH 중 첫 번째로서 이곳에서 충돌이 발생한 단말은 다음 프레임에서 새로운 단말을 위한 RCH를 제외한 첫 번째와 두 번째, 즉 전체 RCH 중 세 번째와 네 번째 RCH 중 하나를 액세스한다. 이전 프레임의 세 번째 RCH에서 충돌이 발생한 단말은 다음 프레임에서 다섯 번째와 여섯 번째 RCH 중 하나를 액세스 한다. 이전 프레임의 세 번째 RCH는 충돌이 발생한 네 개의 RCH 중 두 번째에 해당한다. 따라서 새로운 단말을 위한 RCH를 제외한 세 번째와 네 번째 중 하나를 액세스 하여야 한다. 즉, 전체 RCH 중 다섯 번째와 여섯 번째 RCH 중 하나를 액세스 한다. 그리고 이전 프레임의 다섯 번째 RCH는 충돌이 발생한 네 개의 RCH 중에서 세 번째 RCH에 해당한다. 따라서 새로운 단말을 위한 RCH를 제외한 여섯 번째와 일곱 번째, 즉 전체 RCH 중 일곱 번째와 여덟 번째 RCH 중 하나를 액세스 한다. 마지막으로 이전 프레임의 여섯 번째 RCH에서 충돌이 발생한 단말은 다음 프레임에 액세스 할 수 없다. 이전 프레임의 여섯 번째 RCH는 충돌이 발생한 네 개의 RCH 중 네 번째에 해당한다. 따라서 다음 프레임의 아홉 번째나 열 번째 중 하나를 액세스하여야 한다. 그러나 다음 프레임의 최대 RCH 개수는 8개로서 이를 초과한다. 따라서 이전 프레임의 여섯 번째를 액세스한 단말은 프레임 단위로 랜덤하게 지연을 경험한 후 새로운 단말을 위한 RCH를 액세스한다.

제안된 기법은 각 단말이 액세스 시도 횟수 및 CW_0 을 유지할 필요가 없어 단말의 복잡도를 낮출 수 있을 뿐만 아니라 RCH 액세스에 실패한 단말은 바로 다음 프레임에서 RCH 액세스를 시도함으로써 지연을 줄일 수 있다. 또한 실시간 트래픽의 성능지표 중 하나인 지연에 대한 분산을 효과적으로 줄일

수 있다. 그리고 처리율에 있어서 RCH의 수를 고정된 기법이나 [1]에서의 기법보다 평균 10%에 가

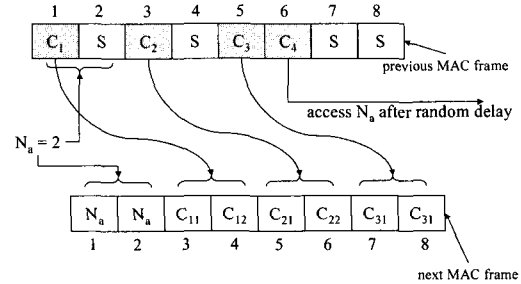


그림 2 새로운 단말을 위한 RCH 수가 2이고 최대 RCH의 개수가 8인 경우의 예

까운 증가를 보인다. ACH는 각각의 RCH에 대한 성공여부를 각 단말에게 알려준다. 이때 0으로 설정된 비트는 해당 RCH가 성공하지 못함을 의미한다. 즉, 충돌 또는 액세스가 없었음을 나타낸다. 1로 설정된 비트는 해당 RCH에 액세스한 단말이 성공적으로 무선 LAN에 접속했음을 의미한다[2]. 제안된 기법에서는 RCH 액세스에 실패한 단말은 충돌이 발생한 RCH 중 자신이 액세스한 RCH가 몇 번째에 해당하는가를 알아야 한다. 이를 위해 ACH의 비트의 의미를 0인 경우는 충돌을 의미하도록 한다. 1인 경우는 해당 RCH에 대한 액세스가 없었거나 성공을 의미하도록 하면 된다. 하나의 RCH에 대하여 하나의 단말이 액세스할 경우에 한하여 성공적으로 무선 LAN에 접속하게 된다. 또한 각 단말은 자신이 액세스한 RCH의 번호를 알고 있기 때문에 [2]에서처럼 명시적으로 성공을 나타낼 필요는 없다.

IV. 처리율 및 지연에 대한 수학적 분석 모델

분석을 위하여 각 단말의 도착은 평균 λ 인 포아송 분포를 따르고 시스템의 RCH 수에는 제한이 없다고 가정한다.

분석을 위하여 $D(n)$, $N(n)$,과 $b(i;m)$ 을 다음과 같이 정의한다.

- $D(n)$: 현재 프레임의 N_a (새로운 단말을 위한 고정된 RCH)에 n 개의 단말이 접속을 시도 하였을 때 모든 단말이 성공적으로 서비스를 받기 위하여 전체 n 개의 단말 측면에서 지연되어야 하는 프레임의 수

- $N(n)$: 현재 프레임의 N_a 에 n 개의 단말이 접속을 시도 하였을 때 모든 단말이 성공적으로 서비스를 받기 위하여 추가로 요구되는 RCH 수
- $b(i;m)$: 하나의 RCH에서 충돌이 발생한 m 개의 단말 중 i 개의 단말이 2개의 RCH가 생성되는 다음 프레임에서 앞쪽의 RCH를 선택할 확률

$$b(i;m) = \binom{m}{i} \left(\frac{1}{2}\right)^i \left(\frac{1}{2}\right)^{m-i} = \binom{m}{i} 2^{-m} = \frac{m!}{i!(m-i)!} 2^{-m} \quad (10)$$

4.1 단말 당 평균 지연에 대한 분석

N_a 가 1인 경우에 대하여 우선적으로 분석한다. $D(0)=D(1)=0$ 이다. 단말의 도착이 없는 경우에는 지연이 없다. 그리고 하나의 단말이 도착한 경우에는 지연 없이 바로 서비스를 받을 수 있다.

$$D(2) = b(0;2)[2+D(2)] + b(1;2)(2) + b(2;2)[2+D(2)] \quad (11)$$

즉, 두 개의 단말이 동시에 도착하였을 경우에 두 단말이 다음 프레임에서 서로 다른 RCH를 액세스 할 경우(식 11의 두 번째 항)에는 각 단말 당 하나의 프레임의 지연이 발생한 후 각 단말은 서비스를 받을 수 있다. 그러나 두 단말이 다음 프레임에도 같은 RCH를 액세스 한다면(식 11의 첫 번째와 세 번째 항) 각 단말 당 하나의 프레임 지연이 발생한 후 두 단말은 다시 경쟁 상태를 유지한다. (식 11)에서 $D(2)$ 를 구할 수 있다.

$$D(3) = b(0;3)[3+D(3)] + b(1;3)[3+D(2)] + b(2;3)[3+D(2)] + b(3;3)[3+D(3)] \quad (12)$$

즉, 세 개의 단말이 동시에 도착하였을 경우에 세 단말이 다음 프레임에서도 같은 RCH를 액세스(식 12의 첫 번째와 네 번째 항) 한다면 각 단말 당 하나의 프레임 지연이 발생한 후 세 단말은 다시 경쟁 상태로 돌아간다. 그러나 다음 프레임에서 세 단말 중 두개의 단말이 같은 RCH를 액세스하고 다른 하나의 단말이 나머지 하나의 RCH를 액세스 한다면(식 12의 두 번째와 세 번째 항) 두 단말은 각각 하나의 프레임 지연을 경험한 후 다시 경쟁 상태를 유지하게 되고 나머지 하나의 단말은 하나의 프레임 지연이 있는 후 성공적으로 서비스를 받을 수 있다.

(식 11)에서 구해진 $D(2)$ 를 기반으로 (식 12)에서 $D(3)$ 를 구할 수 있다. 이와 같은 방법으로 $D(n)$ 을 구할 수 있다(식 13).

$$\begin{aligned} D(0) &= D(1) = 0 \\ D(n) &= \sum_{i=0}^n b(i;n)[n + D(n-i) + D(i)] \\ &= n \sum_{i=0}^n b(i;n) + \sum_{i=0}^n b(i;n)[D(n-i) + D(i)] \\ &= n + b(0;n)D(n) + b(n;n)D(n) + \sum_{i=1}^{n-1} b(i;n)[D(n-i) + D(i)] \\ &= n + 2^{-n}D(n) + 2^{-n}D(n) + \sum_{i=1}^{n-1} b(i;n)[D(n-i) + D(i)] \\ &= \frac{n + \sum_{i=1}^{n-1} \frac{n!2^{-n}}{i!(n-i)!} [D(n-i) + D(i)]}{1 - 2^{-n+1}} \quad \text{if } n \geq 2 \quad (13) \end{aligned}$$

따라서 각 단말 당 평균 지연 프레임은 다음과 같이 구할 수 있다.

$$\bar{D} = \frac{1}{\lambda} \sum_{n=0}^{MAX_MT} D(n) \cdot P(n), \quad P(n) = e^{-\lambda} \frac{\lambda^n}{n!} \quad (n = 0, 1, 2, \dots) \quad (14)$$

MAX_MT는 시스템에 있는 전체 단말의 수를 의미한다.

N_a 가 2인 경우는 각각의 RCH의 도착율을 $\frac{\lambda}{2}$ 로 하여 식을 유도 할 수 있다. N_a 에 대한 일반적인 식은 (식15)와 같이 표현 될 수 있다.

$$\bar{D} = \frac{N_a}{\lambda} \sum_{n=0}^{MAX_MT} D(n) \cdot P(n), \quad P(n) = e^{-\frac{\lambda}{N_a}} \frac{\left(\frac{\lambda}{N_a}\right)^n}{n!} \quad (n = 0, 1, 2, \dots) \quad (15)$$

4.2 처리율에 대한 분석

N_a 가 1인 경우에 대하여 우선적으로 분석한다. $N(0)=N(1)=0$ 이다. 단말의 도착이 없었다면 추가적으로 요구되는 슬롯은 없다. 또한 하나의 단말이 도착한 경우에는 슬롯의 추가 없이 바로 서비스를 받을 수 있다.

$$N(2) = b(0;2)[2+N(2)] + b(1;2)(2) + b(2;2)[2+N(2)] \quad (16)$$

즉, 두 개의 단말이 동시에 도착하였을 경우에 두 단말이 다음 프레임에서 서로 다른 RCH를 액세스 할 경우(식 16의 두 번째 항)에는 두 개

의 슬롯만이 추가된 후 두 단말은 성공적으로 서비스를 받을 수 있다. 그러나 두 단말이 다음 프레임에도 같은 RCH를 액세스 한다면(식 16의 첫 번째와 세 번째 항) 두 개의 슬롯이 추가된 후 다시 경쟁 상태를 유지한다. (식 16)에서 $N(2)$ 를 구할 수 있다.

$$N(3) = b(0;3)[2+N(3)] + b(1;3)[2+N(2)] + b(2;3)[2+N(2)] + b(3;3)[2+N(3)] \quad (17)$$

즉, 세 개의 단말이 동시에 도착하였을 경우에 세 단말이 다음 프레임에서도 같은 RCH를 액세스 한다면(식 17의 첫 번째와 네 번째 항) 두 개의 슬롯이 추가된 후 각 단말은 다시 경쟁 상태로 돌아간다. 그러나 다음 프레임에서 세 단말 중 두개의 단말이 같은 RCH를 액세스하고 다른 하나의 단말이 다른 하나의 RCH를 액세스 한다면(식 17의 두 번째와 세 번째 항) 두개의 슬롯이 추가된 후 두 단말은 다시 경쟁 상태를 유지하게 되고 다른 하나의 단말은 성공적으로 서비스를 받을 수 있다.

(식 16)에서 구해진 $N(2)$ 를 기반으로 (식 17)에서 $N(3)$ 을 구할 수 있다. 이와 같은 방법으로 $N(n)$ 을 구할 수 있다(식 18).

$$\begin{aligned} N(0) &= N(1) = 0 \\ N(n) &= \sum_{i=0}^n b(i;n)[2 + N(n-i) + N(i)] \\ &= 2 \sum_{i=0}^n b(i;n) + \sum_{i=0}^n b(i;n)[N(n-i) + N(i)] \\ &= 2 + b(0;n)N(n) + b(n;n)N(n) + \sum_{i=1}^{n-1} b(i;n)[N(n-i) + N(i)] \\ &= 2 + 2^{-n}N(n) + 2^{-n}N(n) + \sum_{i=1}^{n-1} b(i;n)[N(n-i) + N(i)] \\ &= \frac{2 + \sum_{i=1}^{n-1} \frac{n!2^{-n}}{i!(n-i)!}[N(n-i) + N(i)]}{1 - 2^{-n+1}} \quad \text{if } n \geq 2 \quad (18) \end{aligned}$$

따라서 현재 프레임에 n 개의 단말이 접속을 시도 하였을 때 모든 단말이 성공적으로 서비스를 받기 위하여 요구되는 평균 RCH의 수는 (식 19)와 같이 구할 수 있다.

$$\bar{N} = 1 + \sum_{n=1}^{MAX_MT} N(n)P(n), \quad P(n) = e^{-\lambda} \frac{\lambda^n}{n!} (n = 0, 1, 2, \dots) \quad (19)$$

MAX_MT 는 시스템에 있는 전체 단말의 수를 의미한다.

N_a 가 2인 경우는 각각의 RCH의 도착율을 $\frac{\lambda}{2}$

로 하여 식을 유도 할 수 있다. N_a 에 대한 일반적인 식은 (식 20)과 같이 표현 될 수 있다.

$$\bar{N} = N_a \left(1 + \sum_{n=1}^{MAX_MT} N(n)P(n) \right), \quad P(n) = e^{-\frac{\lambda}{N_a}} \frac{\left(\frac{\lambda}{N_a}\right)^n}{n!} (n = 0, 1, 2, \dots) \quad (20)$$

(식 20)의 첫 번째 항은 새로운 단말을 위해 할당 되어진 RCH의 수를 의미하고 두 번째 항은 충돌이 발생한 단말을 위해 추가적으로 요구되는 RCH의 수를 의미한다.

할당된 자원이 얼마나 효율적으로 사용되어지는가에 대한 지표인 처리율(ρ)은 (식 21)과 같이 정의되어진다[1].

$$\begin{aligned} \rho &= \frac{\text{total number of successful access attempt}}{\text{total number of allocated RCHs}} \\ &= \frac{\lambda}{\text{Mean number of RCHs}} \\ &= \frac{\lambda}{N_a \left(1 + \sum_{n=0}^{MAX_MT} N(n)P(n) \right)}, \\ P(n) &= e^{-\frac{\lambda}{N_a}} \frac{\left(\frac{\lambda}{N_a}\right)^n}{n!} (n = 0, 1, 2, \dots) \quad (21) \end{aligned}$$

V. 수학적 분석 및 시뮬레이션 결과

본 논문에서는 수학적 분석 모델 및 시뮬레이션을 통하여 제안된 기법의 성능을 분석하였다. 제안된 기법의 성능 분석을 위한 지표로 (식 21)에서 정의한 처리율과 실시간 트래픽에 있어서 중요한 성능 지표인 평균 지연과 지연에 대한 분산을 고려하였다. 평균 지연 및 지연에 대한 분산은 프레임 단위로 계산하였다. 각각의 성능지표에 대하여 새로운 단말을 위해 할당된 RCH 값 및 액세스할 RCH의 범위가 R_{MAX} 를 초과하는 단말에 대하여 프레임 단위로 랜덤하게 발생하는 지연이 시스템의 성능에 어떠한 결과를 미치는가를 살펴보았다.

본 논문에서는 하나의 AP 시스템을 기준으로

시뮬레이션을 수행하였으며 무선 채널의 영향으로 인한 전송에러는 고려하지 않았다. 전체 단말의 수는 50개로 가정하였으며 각 단말의 자원요청은 포아송 프로세스를 따른다고 가정하였다.

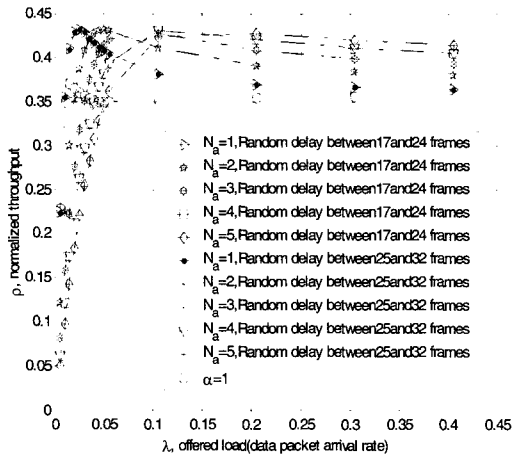


그림 3 도착율 대 처리율(시뮬레이션 결과, $R_{MAX}=31$)

전체 시뮬레이션 시간은 1000초로서 5×10^5 개의 MAC 프레임을 포함한다.

[1]에서는 RCH 수를 고정된 기법과 제안한 기법에 대하여 처리율 및 평균 지연에 대하여 시뮬레이션을 통한 성능 분석을 하였다. [1]에서는 (식 3)의 α (가중치)가 1일 때 가장 좋은 성능을 나타내었다. RCH의 수가 고정된 경우 최고 35%까지 처리율이 나타나는 반면 [1]에서 제안한 방법에서는 α 가 1일 때 처리율이 최고 37%까지 증가되었다(그림 3). 그러나 그 차이는 2% 정도로 크지 않음을 알 수 있다. 그림 3과 4는 본 논문에서 제안된 기법의 처리율에 대한 시뮬레이션 및 수학적 분석 결과를 보여주고 있다. 본 논문에서 제안된 기법을 통하여 처리율은 최고 44%까지 증가하였다. 제안된 기법은 평균적으로 10%정도의 처리율 향상을 가져왔으며 결과적으로 기존의 방법 보다 효율적으로 무선 자원을 이용하고 있음을 알 수 있다. 일반적으로 단말의 도착율이 낮은 경우에는 N_a 값이 작을 때에 높은 처리율을 보인다. 즉, 단말의 도착율이 낮은 경우에 새로운 단말을 위해 많은 RCH를 할당하는 것은 자원의 낭비를 의미한다. 단말의 도착율이 높은 경우에는 N_a 값이 높을 때에 처리율의 향상을 보인다. 즉, 단말의 도착율이 높은 경우에 새

로운 단말을 위해 적은 RCH를 할당하는 것은 단말간의 충돌로 인해 자원을 효율적으로 사용하지 못함을 알 수 있다. 하나의 MAC 프레임에서의 최대 RCH의 개수는 31개 까지 가능하다[2]. 그림 3에서는 R_{MAX} 의 값을 31로 설정한 후 R_{MAX} 를 초과하는 단말에 대하여 17에서 24프레임 또는 25에서 32 프레임 동안

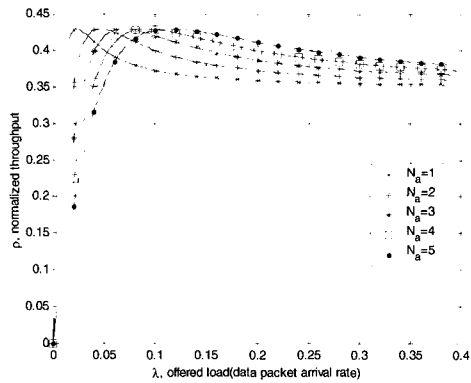


그림 4 도착율 대 처리율(수학적 분석 결과)

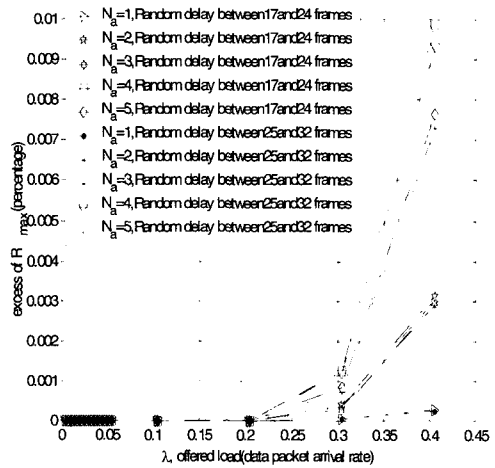


그림 5 R_{MAX} 값을 초과 할 확률($R_{MAX} = 31$)

랜덤하게 지연을 발생시킨 시뮬레이션 결과이다. 이 경우에는 R_{MAX} 를 초과하는 단말에 대한 랜덤지연의 차이는 거의 없음을 알 수 있다. R_{MAX} 값이 31일 경우 시스템에서 필요한 RCH의 개수가 R_{MAX} 를 초과하는 경우가 거의 없기 때문이다(그림 5). 그림 6과 7에서는 처리율에 대한 평균 지연시간을 프레임단위로 나타내고 있다. 예를 들어 처리

율이 3.5에서 제안된 기법은 하나의 프레임 이내의 지연을 보인 반면 [1]에서의 방법은 3개 이상의 프레임의 지연이 발생되었다. 지연에 민감한 실시간 트래픽에 있어서 제안된 기법을 사용함으로써 서비스 품질을 만족시킬 수 있다. 실시간 트래픽에 대해서 지연과 더불어 지연에 대한 분산 또한 중요한 성능지표가 된다. [1]에서

만족시킬 수 있는 기법이다.

하나의 MAC 프레임에 대하여 최대 RCH의 개수를 31로 설정한 경우에 R_{MAX} 를 초과한 단말에 대한 랜덤지연이 시스템의 성능에 대하여 거의 영향을 미치지 않음은 위에서 살펴보았다. 그러나 하나의 MAC 프레임에 대하여 RCH의 개수를 8개로 설정한 경우에는 랜덤지연은 시스템의

표 1 지연에 대한 분산($N_a=2, R_{MAX}=31, \text{Random delay}=[17,24]$)

도착율	[1]에서 =1	$N_a=2, R_{MAX}=31, \text{Random delay}=[17,24]$
0.005	23.3911	0.676124
0.01	409.495	1.18526
0.015	1467.6	1.60542
0.02	1969.09	1.95896
0.025	2066.16	2.23938
0.03	1871.38	2.46129
0.035	1721.5	2.6278
0.04	1506.57	2.76634
0.045	1329.39	2.88813
0.05	1203.47	2.96508
0.055	1057.31	3.06026
0.105	484.129	3.38835
0.205	223.639	3.47065
0.305	150.943	3.5624

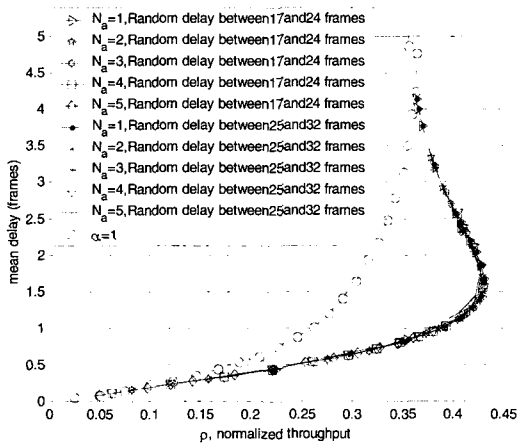


그림 6 처리율 대 평균 지연(시뮬레이션 결과, $R_{MAX}=31$)

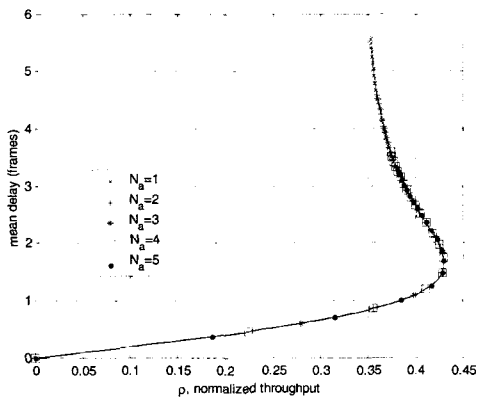


그림 7 처리율 대 평균 지연(수학적 분석 결과)

α 가 1인 경우와 본 논문에서 제안한 기법에 대한 지연에 대한 분산은 표 1과 같다. 제안된 기법은 지연 및 지연에 대한 분산에 대하여 기존의 방법에 비해 상당히 향상된 결과를 보이고 있다. 앞으로의 무선 LAN은 비실시간 트래픽 뿐만 아니라 실시간 트래픽 등 멀티미디어 트래픽에 대한 서비스 품질(QoS)을 보장해 주어야 한다. 본 논문에서 제안된 기법은 앞으로의 무선 LAN에서 요구되는 멀티미디어 트래픽에 대한 QoS를

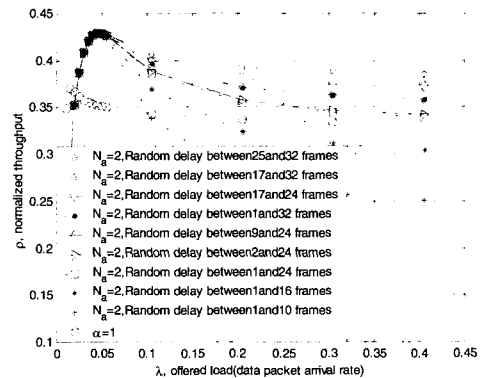


그림 8 도착율 대 처리율(시뮬레이션 결과, $R_{MAX}=8$)

성능에 중요한 요소가 된다.

그림 8은 새로운 단말을 위한 RCH 값이 2일 때 R_{MAX} 의 값을 8로 설정한 경우 도착율 대 처리율에 대한 시뮬레이션 결과이다. R_{MAX} 를 초과한 단말이 경험하게 될 랜덤지연의 값을 여러 가지로 변화시켜 가면서 시스템의 성능에 어떠한

영향을 미치는가를 살펴보았다. 여러 가지의 랜덤지연에 대하여 제안된 기법의 처리율은 최대 44% 정도를 보여주고 있다. 그러나 시스템의 안정화 측면에서 랜덤지연의 값이 작을 경우(1부터 10 또는 16)는 바람직하지 않음을 볼 수 있다. 일반적으로 시스템 성능은 랜덤지연의 범위를 크게 하기 보다는 랜덤지연의 값

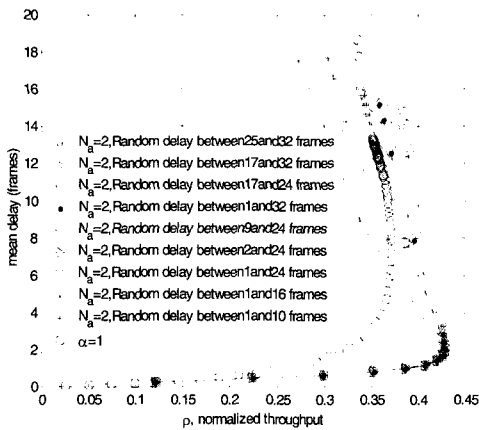


그림 9 처리율 대 평균 지연(시뮬레이션 결과, $R_{MAX} = 8$)

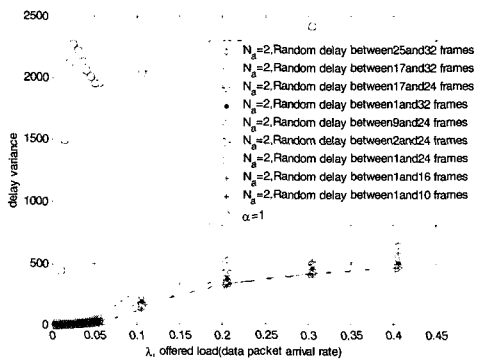


그림 10 도착율 대 지연에 대한 분산($R_{MAX} = 8$, 시뮬레이션 결과)

을 크게 하는 것이 향상됨을 볼 수 있다.

그림 9는 여러 가지 랜덤지연에 대하여 새로운 단말을 위한 RCH 값이 2일 때 R_{MAX} 의 값을 8로 설정한 경우 처리율 대 평균 지연에 대한 시뮬레이션 결과이다. 평균 지연의 경우 [1]의 방법에서는 처리율이 30% 정도에서 두 프레임 정도의 평균지연이 발생한 반면 제안된 기법에서는 처리율이 44% 정도에서 두 프레임 정도의 평균지연이 발생하였다. 그림 10은 여러 가지 랜덤지연에 대

하여 새로운 단말을 위한 RCH 값이 2일 때 R_{MAX} 의 값을 8로 설정한 경우 도착율 대 지연에 대한 분산의 시뮬레이션 결과이다. R_{MAX} 의 값이 31일 때와 비교하여 R_{MAX} 의 값이 작아짐에 따라 지연 및 지연에 대한 분산이 커짐을 볼 수 있다. 그러나 [1]의 결과와 비교할 때 충분히 효과적인 결과임을 알 수 있다. 본 논문에서 제안된 기법은 앞으로의 무선 LAN에서 요구되는 실시간 트래픽에 대한 QoS를 보장해 주는데 있어서 매우 좋은 성능을 보여주고 있다.

VI. 결론

본 논문에서는 무선 LAN 표준 중 하나인 HIPERLAN/2의 랜덤채널 할당 기법을 제안하였다. 제안된 기법은 각 단말이 액세스 시도 횟수 및 CW_0 을 유지할 필요가 없어 단말의 복잡도를 낮출 수 있는 동시에 처리율, 지연 및 지연에 대한 분산에서 기존의 기법과 비교하여 더욱 좋은 성능을 보여주고 있다. 본 논문에서는 제안된 기법에 대한 시뮬레이션 및 수학적 분석을 통하여 시스템 성능을 보여주고 있다. 각 프레임마다 고정된 개수의 RCH를 사용하는 기법과 [1]에서의 기법에 대하여 제안된 기법은 평균적으로 10% 정도의 처리율 향상을 가져왔다. 결과적으로 기존의 방법 보다 효율적으로 무선 자원을 이용하고 있음을 알 수 있다. 새로운 단말을 위한 RCH 개수는 일반적으로 트래픽의 양에 비례하여 증가시키는 것이 처리율의 향상을 가져올 수 있다. 하나의 MAC 프레임의 최대 RCH를 초과한 단말이 경험하게 될 랜덤지연의 값은 지연의 크기가 클수록 처리율에 있어서 효과적이었다. 그러나 지연에 대한 분산의 값이 높아짐을 볼 수 있다. 앞으로의 무선 LAN은 멀티미디어 트래픽에 대한 QoS를 보장해 주어야 한다. 즉, 비실시간 트래픽뿐만 아니라 실시간 트래픽에 대한 QoS를 만족하기 위해서는 실시간 트래픽에서 요구되는 지연 및 지연에 대한 분산을 낮출 수 있어야 한다. 그러나 기존의 방법에서는 이러한 요구를 충족시키지 못하고 있다. 결국 본 논문에서 제안된 기법은 앞으로의 무선 LAN에서 요구되는 멀티미디어 트래픽에 대한 QoS를 만족시킬 수 있는 기법이다.

참 고 문 헌

[1] Gyung-Ho Hwang, Dong-Ho Cho, "Adaptive Random Channel Allocation Scheme in HIPERLAN Type 2", *IEEE COMMUNICATIONS LETTERS*, VOL. 6, NO. 1, JAN. 2002, pp. 40-42

[2] *HIPERLAN Type 2; Data Link Control (DLC) Layer: Part 1, Basic Data Transport Functions*, Broadband Radio Access Networks(BRAN), ETSI Standard TS 101 761-1 V1.3.1, Nov. 2001

[3] Angela Doufexi, et al., "A Comparison of the HIPERLAN/2 and IEEE 802.11a Wireless LAN Standards", *IEEE Communications Magazine*, May 2002, pp. 172-180

[4] R. Van Nee et al., "New High-Rate Wireless LAN Standards", *IEEE Communications Magazine*, Vol. 37, No. 12, Dec. 1999, pp. 82-88

[5] A. Doufexi et al., "Throughput Performance of WLANs Operating at 5GHz Based on Link Simulations with Real and Statistical Channels", *IEEE VTC '01 Spring*

[6] G. Anastasi, L. Lenzini, and E. Mingozzi, "MAC Protocols for Wideband Wireless Local Access: Evolution Toward Wireless ATM", *IEEE Pers. Commun.*, Oct. 1999, pp. 169-74

[7] A. Hettich and M. Schrother, "IEEE 802.11a or ETSI BRAN HIPERLAN/2: Who Will Win the Race for a High-Speed Wireless LAN Standard", *Euro. WL Conf.*, Germany, Oct. 1999, pp 169-74

[8] Bertsekas D, Gallager R. *Data Networks*. New Jersey: Prentice Hall, 2nd edition, 1992

황 의 석(Eui-Seok Hwang)

정회원

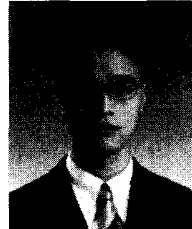


1998년 : 고려대학교 전산학과
이학사
2000년 : 고려대학교 전산학과
이학석사
2000년~현재 : 고려대학교 전
산학과 박사 과정

<주관심분야> 이동통신, 네트워크 보안

고 유 창(You-Chang Ko)

정회원



1994년 : 고려대학교 전산학과
이학사
1996년 : 고려대학교 전산학과
이학석사
1998년~현재 : 고려대학교 전
산학과 박사과정
LG 전자 GSM 단말 연구소

선임 연구원

<주관심분야> 무선이동통신망 트래픽 제어 및 Qos
관리, 무선 이동망에서의 핸드오프,

윤 철 식(Chul-Sik Yoon)

정회원

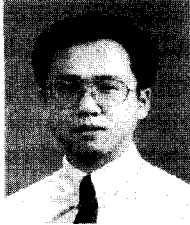


1988년 : 서울대학교 물리교육과
학사
1990년 : 포항공과대학 물리학과
석사
1998년~현재: 서강대학교 전자
공학과 박사과정
한국전자통신연구원 선임연구원

<주관심분야> 표준화연구, 시스템 엔지니어링

이 승 규(Seung-Que Lee)

정회원



1988년 : 중앙대학교 전자계산학과 학사
1990년 : 중앙대학교 전자계산학과 석사
2002년~현재: 고려대학교 전산학과 박사과정
한국전자통신연구원 선임연구원

<주관심분야> OFDM, MAC 프로토콜, QoS, 스케줄링

이 형 우(Hyong-Woo Lee)

정회원



1979년 : University of British Columbia Electrical Engineering (학사)
1983년 : University of Waterloo, Electrical Engineering(박사)
1983년~1988년 : Univ. of British Columbia, 조교수

1988년~1991년 : Univ. of Carleton, 조교수
1992년~1995년 : Univ. of Waterloo, 조교수
1995년~현재 : 고려대학교 전자 및 정보공학부, 교수

<주관심분야> 통신망 설계 및 성능분석, ATM 트래픽제어, MAC프로토콜

조 충 호(Choong-Ho Cho)

정회원



1981년 : 고려대학교 공과대학 산업공학과 학사
1983년 : 고려대학교 공과대학 산업공학과 석사
1986년 : 프랑스 Institute National des Sciences Appliques de Lyon 전산학과 석사

1989년 : 프랑스 INSA de Lyon 전산학과 박사
1990~1994년 : 순천향대학교 전산통계학과 조교수
1994~현재 : 고려대학교 전산학과 교수

<주관심분야> 통신망 트래픽관리기술, 무선통신 시스템, 멀티미디어통신