

# 무선 네트워크에서 신뢰성 있는 MAC Layer 멀티캐스트에 대한 성능 분석

송유덕 성우경 김보성 장주욱  
서강대학교 전자공학과

## Analysis of Reliable MAC Layer Multicast in Wireless Networks

Yoodoc Song Wookyung Sung Bosung Kim Juwook Jang  
Sogang Univ.

### 요약

현재 IEEE 802.11 무선 네트워크에서 무선의 특성을 유용하게 이용하기 위한 멀티캐스트/브로드캐스트 표준이 존재하지 않는다. IEEE 802.11에서는 CSMA/CA방식을 기본으로 지원하며, 이에 따라 새로운 프로토콜이 요구된다. 현재 BMW[2], BMMM[1]등의 프로토콜이 제안 되었으며, 이러한 프로토콜 성능에 대한 비교, 분석이 필요하다. M. T. Sun등이 제안한 BMMM[1]의 논문에서 BMW와 비교하여 성능이 우수한 BMMM 프로토콜을 제안하였다. 하지만 이 논문에서는  $n$ 개의 수신 노드에 하나의 데이터만을 전송하는 상황을 고려하였으며, 이에 따라 BMW에 대한 성능 평가가 제대로 이루어 지지 않았다. 본 논문에서는 BMW 프로토콜의 성능을 새로이 분석하여 BMMM에 대해 분석한 논문과 비교한다.

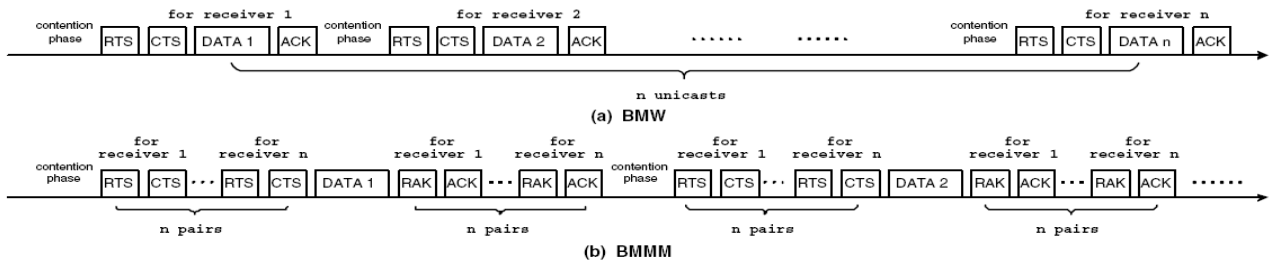
## 1. 서론

IEEE 802.11 무선 네트워크에서 멀티캐스트를 위한 신뢰성 있고, 효율적인 프로토콜의 연구가 계속되고 있다. IEEE 802.11 무선 네트워크에서의 멀티캐스트를 위한 프로토콜은 최초 IEEE 802.11의 표준에 따르는 CSMA/CA 방법이 존재하였다. 이 방법은 ACK의 유실시 전송실패로 판단하여 재전송을 하는 방법으로, hidden node problem 과 ACK 충돌문제 등이 존재한다. 이러한 문제를 해결하고 보다 효율적인 멀티캐스트, 브로드캐스트 통신을 위하여, BMW[2]와 BMMM[1]등이 제안되었다.

M. T. Sun 등은 BMMM에 관한 논문[1]에서 contention phase가 BMW 비해 적고, 효율적인 BMMM을 제안하고 이에 대한 성능을 평가하였다. 하지만, [1]의 논문에서는 BMW의 성능을 제대로 파악하지 못하였다. [2][3]등에 나타난 것과는 다르게 BMW프로토콜을 정의하였으며, 이를 이용해 BMMM의 효율성과, 신뢰성을 말하였다.

먼저, [1]의 논문은 하나의 데이터가 존재하는 상황에서의 contention phase의 수를 가정하였다. 즉, BMW의 경

우 하나의 데이터가 존재할 때 노드의 수  $n$ 만큼의 contention phase가 존재하고  $n$  contention phases/ a data가 됨을 말하고 있다. 하지만 실제 전송환경에서는 연속된 데이터가 전송되는 경우가 많고, 이러한 경우 BMW의 contention phase 수는  $n$  노드에 대해  $n$ 개의 contention phase 수를 가지고 이는 contention phase 수가 데이터 당  $n$ 번에서 1번으로 줄어드는 분석이 된다. 다음으로 [1]의 논문은 RTS / CTS를 주고받는 해당 수신 노드에 대한 데이터 전송실패 시 또 다시 contention phase를 가진다고 말하고 있다. 하지만 실제 BMW의 경우, (그림 2) 과 같이 하나의 데이터에 대해  $n$ 개의 노드 중 하나의 노드만이 한번의 contention phase를 가진다. 이러한 두 가지의 분석 요인의 차이로 인해 [1]의 논문은 BMW의 성능 분석을 제대로 하지 못하였으며, BMW와 BMMM의 성능 비교 또한 문제점이 있다. 본 논문에서는 BMW의 최초 제안 논문 및 BMW관련 논문의 내용에 따라 BMW 프로토콜을 다시 고려하며, 하나의 데이터만을 전송하는 상황이 아닌 실제 환경과 흔히 일어날 수 있는 다수의 데이터를 연속적으로 전송하는 상황을 고려한다. 이에 따라 새로이 BMW 프로토콜의 성능을 수식적으로 분석하



( 그림 1 ) BMW and BMMM

여, BMW의 성능을 BMMM과 비교하고자 한다.

## 2. BMW와 BMMM의 기본개념

### 2.1 BMMM

BMMM(Batch Mode Multicast MAC)은 IEEE802.11에서 신뢰성 있는 멀티캐스트를 지원하기 위해 M. T. Sun, 외 4명 등에 의해 제안된 프로토콜이다[1]. BMMM(그림1-b)는 기본적으로 하나의 데이터를 n개 노드에 전송할 때 n개의 RTS/CTS 프레임 쌍과 n개의 RAK(Request for ACK)/ACK 프레임 쌍을 이용한다. 하나의 데이터를 전송하기 위해서 먼저 송신자는 각각의 노드에게 RTS를 송신하고 각각에 대한 CTS를 받음으로써 수신할 모든 노드들의 상태를 준비시키고 데이터를 전송한다. 즉 1, ... n까지의 수신노드가 존재할 때, 송신자는 1에게 RTS를 보내고 1로부터 CTS를 받음으로써 1에 대한 준비상태를 약속하고 순차적으로 2, 3, ... n 노드들과 RTS/CTS를 주고받음으로써 데이터를 전송할 모든 노드들과 준비상태를 확인한다. 이러한 RTS/CTS교환이 완료되면 송신자는 모든 노드들에게 데이터를 송신하고, RAK/ACK를 모든 수신 노드와 주고받음으로써 모든 수신 노드의 데이터 전송결과를 확인한다.

이러한 BMMM의 장점은 모든 노드가 첫 번째 데이터를 받기까지 걸리는 딜레이가 짧아진다는 점이다. 하지만 BMMM에서는 n쌍의 RTS/CTS, RAK/ACK에 의해 컨트롤 오버헤드가 커지는 단점을 가지고 있다.

### 2.2 BMW

BMW(Broadcast Medium Window) 프로토콜은 IEEE 802.11에서 브로드캐스트 서비스의 신뢰성을 높이기 위해 Tang 과 Gerla 에 의해 제안된 알고리즘이다.[2]

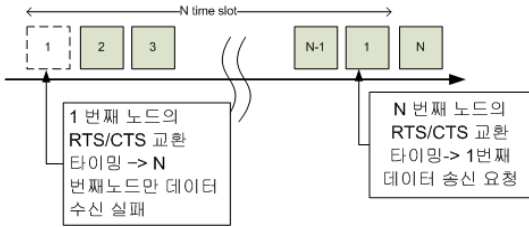
BMW (그림 1-a)은 RTS/ CTS / DATA / ACK 로 구성된다. 1-hop 간격의 이웃 노드에 대해 각각 유니캐스트를 함으로서 신뢰성 있는 브로드 캐스트를 하게 된다. BMW 프로토콜에서는 각각의 이웃 노드에 유니캐스트를 하고 다른 노드들이 데이터 프레임을 엿들음으로서 전송이득을 높인다. 한 이웃 노드에게 데이터 프레임이 완전하게 전송되게 되면, 노드는 ACK를 통해 수신 성공을 알리고, 송신자는 데이터 프레임의 시퀀스 수를 하나 증가 시킨 후 이웃 노드와 유니캐스트 통신을 시작한다. 반대로 이웃 노드가 데이터 프레임을 받지 못하였을 때는 송신자가 보낸 RTS에 대해, 받지 못한 데이터 프레임의 시퀀스 수를 CTS에 포함시켜 송신자에게 보낸다. 이러한 BMW는 논문[1]에서 언급한 것과는 달리 전체 노드 수와 재전송 횟수와는 관계없이 하나의 데이터에 대해서는 한번의 contention phase 가 존재하며(그림2), 구현이 간단하다는 장점이 있다.

[1]의 논문에는 BMW의 단점으로 하나의 데이터를 전송하는 상황에서 딜레이가 크게 발생할 수 있다는 점들을 들 수 있다. 이러한 예로, n개의 노드가 존재하고 첫 번째 데이터에 대하여 n번째 노드만 그 데이터를 받지 못했을 때, n 번째 노드가 첫 번째 데이터를 수신하기까지 1부터 n-1 노드가 통신하는 시간 동안 지연된다는 것이다. 여기서의 지연은 하나의 데이터가 존재할 때를 가정함이다. 하지만 실제 시스템에서는 하나의 데이터만 존재하는 상황이 아니라 연속적인 데이터를 전송하는 상황을 가정하여 하여야 한다. 즉 n개의 노드가 존재하고 n 개 이상의 데이터가 연속적으로 전송되는 상황이라면,



( 그림 2 ) BMW 프로토콜에서 재전송 발생 시 프레임 구조

(그림 3)과 같이 n번째 노드에서 첫 번째 데이터의 손실은 전체 n중에 1만큼의 손실에 불과하다. 즉, 1 delay / n time slot의 지연이 발생하게 되어 전체 데이터 전송률에는 큰 영향을 미치지 않는다.



(그림 3) n번째 노드의 데이터 수신 상황  
- 첫 번째 데이터를 n번째 노드만 수신 실패 시 -

### 3. BMMM, BMW의 성능 분석 및 비교

#### 3.1 평균 Contention phase 수

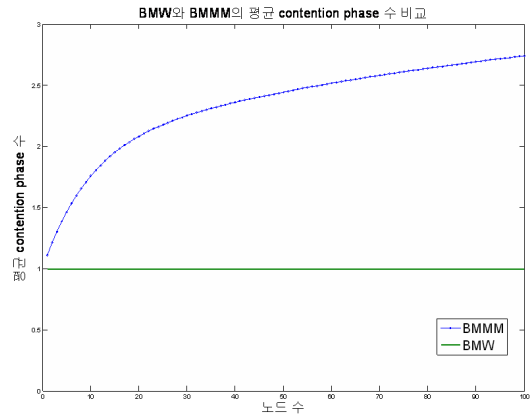
Contention phase는 데이터 전송을 원할 때 medium의 상태를 확인하기 위한 단계이다. 만약 medium 이 비어 있다면, 프레임을 전송하고, medium이 사용 중이라면, medium이 빌 때까지 대기 후 랜덤 시간 slot동안 back-off 한다. back-off timer가 만기 되었을 때, 채널이 비어있다면 프레임을 전송하고 back-off 시간 전에 채널이 사용 중이라면 timer를 멈추고 다시 채널의 상태를 확인한다.

[1]의 논문은 실제 무선 통신에서의 상황이 아닌 하나의 데이터가 존재할 때만을 가정한다. 그에 따라 BMW 프로토콜에서의 contention phase 수는 하나의 데이터에 대해 노드 수 n만큼의 contention phase 수를 가짐을 말하고 있다. 즉 n contention phase / a data를 말하고 있다.

하지만 실제 무선 통신 환경에서는 한 시간 slot의 데이터가 아닌 다수의 time slot에 걸쳐 연속적으로 데이터가 전송되는 상황이 발생하게 된다. BMW에서 모든 노드와 한번 씩 RTS/CTS를 주고받는 시간을 1 사이클이라고 했을 때, 1 사이클은 n개의 노드에 대해 n개의 데이터를 전송하는 상황이 된다. 한 사이클 안에서는 (그림 2)과같이 재전송 여부와 관계없이 각 노드 당 한번의 contention phase 만이 발생한다. 즉 n개의 노드에 대해 n개의 데이터가 전송되고, 이때에는 n번의 contention phase만이 존재한다. 이러한 경우에 BMW의 contention phase 수는 1 contention phase / a node이 되고 노드 수에 비해 데이터 수가 더 많을 경우, 경우 1 contention

phase / a data가 된다.

BMW의 평균 contention phase 수를 [1]에서 BMMM의 contention phase 수와 수식적으로 비교 해볼 때 (그림 4)와 같은 그래프를 나타낸다. 즉 BMW가 평균 contention phase의 수에서 더 나은 성능을 나타냄을 알 수 있다. 또한 BMMM의 경우 각 시간 slot마다 노드 수만큼의 RTS / CTS를 가지게 되는데 이는 모두 데이터에 대한 오버헤드가 된다.



(그림 4) 하나의 데이터에 대한 BMW 와 BMMM의 평균 contention phase 수 비교 (p = 0.9)

#### 3.2 평균 데이터 전송 시간

[1]의 논문은 전체 n개의 노드에 대해 하나의 데이터를 전송하는 상황을 고려하였다. 하지만 이러한 가정은 실제 통신 상황을 고려하지 않은 상황이다. 실제 통신 환경에서 프로토콜의 성능을 분석하기 위해서는 하나의 데이터 프레임이 아닌 다수의 데이터 프레임이 존재하는 경우를 가정하여야 한다.

[1]에서는 실제 통신 상황과 달리 하나의 데이터 프레임만 존재하는 상황을 고려하였다. 이럴 경우 BMW는 하나의 데이터 프레임만 존재하는 경우에도 통신을 완료하기 위해 모든 수신노드와 RTS / CTS 를 주고 받아야 한다. 따라서 연속된 데이터의 수가 적고 노드의 수가 많을 때 contention phase로 인해 좋지 않은 성능을 나타낼 수 있다. BMMM 프로토콜은 각 데이터 전송에서 모든 노드와 RTS / CTS를 주고받기 때문에 하나의 데이터만 존재할 때 하나의 시간 slot에 통신을 끝낼 수 있게 된다. 이러한 특정한 상황에서 BMMM은 우수한 성능을 나타낸다. 하지만 실제 멀티캐스트 통신환경에서는 한 번의 통신에 하나의 데이터 프레임만을 보내는 경우는 거의 없으므로 본 논문에서는 다수의 데이터 프레임

이 생성되는 환경을 고려하여 프로토콜 성능을 분석한다. 프로토콜의 성능을 분석하기 위해 두 프로토콜의 평균 데이터 전송시간을 비교한다. 평균 데이터 전송 시간은 n개의 수신 노드가 존재할 때 하나의 노드가 n개의 데이터를 모두 수신할 때까지 걸리는 시간의 평균값으로 정의한다. 이때 n개의 수신 노드가 n개의 데이터를 모두 수신할 때까지의 과정을 1 사이클이라 하고 1 사이클의 상황에서 평균 데이터 전송 시간을 고려한다.

### 3.2.1 BMW의 평균 데이터 전송 시간

평균 데이터 전송 시간에서 가장 큰 영향을 미치는 요소는 재전송에 의한 오버헤드가 될 것이다. BMW의 평균 데이터 전송 시간을 분석하기 위해 모든 수신 노드가 한 번씩 RTS / CTS를 주고받기까지의 시간 1 사이클 고려하겠다. 평균 재전송 시간을 고려함에 있어 고려되는 데이터 프레임의 각 시간 slot을 contention phase 가지는 최초 데이터 프레임 전송과 contention phase를 가지지 않는 재전송 데이터 프레임 전송으로 나누어 생각할 수 있다. 최초 데이터 프레임 전송은 contention phase를 가지는 특징이외에 각각의 데이터 프레임은 각각 하나의 노드가 담당하고 있다는 특징을 가지고 있다. 그리하여 최초 데이터 프레임 전송 횟수는 모든 수신 노드가 한 번씩 RTS / CTS를 주고받기까지의 시간 사이클 동안에는 전체 수신 노드수와 관계없이 한 노드에 한 번만을 가지게 된다. 재전송 프레임의 경우에는 각각의 노드에 대해 1 사이클 동안 n개의 데이터 재전송을 고려해 주어야 하며, 전체 수신 노드 수에 영향을 받게 된다.

하나의 수신 노드만이 존재하고 데이터의 전송성공 확률이 p일 때, 한 데이터 프레임을 재전송하기 위해서는  $1 + (1-p) + (1-p)^2 + (1-p)^3 + (1-p)^4 + \dots = 1/p$  번의 재전송횟수를 가져야한다. 한 수신노드에 대해 고유하게 부여된 한 번의 시간 slot 즉, 최초 데이터 프레임 전송에 대해서는 contention phase + RTS + CTS + 데이터 + ACK 만큼의 시간이 한번 씩 소모되며, 추가적인 재전송에 대해서는 contention phase 없이 각각 RTS + CTS + 데이터 + ACK 만큼의 시간이 소모된다. 이에 따라 평균 데이터 전송 시간은 아래와 같아진다.

$$T_d(1) = T_{CP} + T_{frame} + (1-p)T_{frame}1/p \quad (1)$$

BMW에서 두개의 수신 노드만이 존재하는 경우, 한 사이클 안에서 하나의 노드가 두 개의 데이터를 모두 받기

까지의 시간은 1번의 contention phase를 가진다. 두 개의 수신 노드가 존재하는 상황에서 송신자는 각각의 노드에 대해 최초의 CTS송신 전에만 contention phase를 가지고, 각 노드의 재전송 데이터 프레임에 대해서는 contention phase없이 CTS를 바로 송신하기 때문이다. 두 개의 수신 노드가 존재하는 상황에서 평균 전송횟수는 한 수신 노드가 1 사이클(두개의 수신 노드를 가정한 상황에서는 두 개의 데이터가 모두 전송되는 과정)동안의 평균 전송 횟수를 고려한다. 이 값은 세부적으로 구성된다. 먼저, 첫 번째 수신 노드의 경우 첫 번째 데이터 전송만을 최초 데이터 전송으로 고려한다. 두 번째 데이터는 두 번째 노드에서 최초 데이터가 되며, 따라서 첫 번째 노드에서는 두 번째 데이터의 최초 전송은 over-hear 함으로써 수신가능하게 되지만, 첫 번째 노드에서 전송 시간을 고려할 때는 제외하게 된다. 다음으로 두 개의 노드중 하나의 노드에서 두 개의 데이터 중 하나를 재전송해야 할 경우를 고려한다. 첫 번째 수신 노드가 첫 번째 데이터 혹은 두 번째 데이터의 최초 전송을 수신하지 못하였을 때, 이러한 형태는 하나의 데이터를 수신하지 못할 확률  $(1-p)$ 과 두 개의 노드중 하나가 선택될 조합  ${}_2C_1$ 의 값이 곱해진  $(1-p){}_2C_1$ 의 확률이 된다. 이러한 경우 재전송을 반복하여 데이터를 완전히 수신할 평균 전송 횟수는  $1/p$ 가 된다. 다음으로 첫 번째 노드가 두 개의 데이터 프레임의 두 번의 최초 전송 모두를 수신하지 못하였을 경우는  ${}_2C_2(1-p)^2$ 의 확률로 나타나며, 두 개의 재전송 데이터를 성공적으로 전송하기 위한 평균 전송 횟수는  $2/p$ 가 된다. 한 노드의 평균 데이터 전송 횟수는 이 세 가지 확률에 따른 평균 재전송 횟수를 더한 형을 모두 더한  $f_2$ 의 형태가 된다.

$$f_2 = 1 + {}_2C_1p(1-p)1/p + {}_2C_2(1-p)^22/p \quad (2)$$

평균 재전송 횟수를 이용해 평균 데이터 전송시간을 구하기 위해 최초 데이터 전송인 1 에서는 contention phase를 가지는 데이터 프레임 시간을 곱하고, 재전송 시에는 BMW 프로토콜의 특징에 따라 contention phase가 없다. 평균 재전송 횟수에서 최초 전송 부분과 재전송 부분에 곱하여 주는 값을 달리하여 평균 데이터 전송 시간을 구하면

$$T_d(2) = T_{CP} + T_{frame} + \sum_{r=1}^2 C(2,r)p^{2-r}(1-p)^r T_{frame}(r/p) \quad (3)$$

가 된다. 이와 같은 방식으로 평균 데이터 전송 시간은 아래 식 (4)과 같이 나타나게 된다.

$$T_d(N) = T_{CP} + T_{frame} + \sum_{r=1}^N N C_r p^{N-r} (1-p)^r T_{frame}(r/p) \quad (4)$$

여기에서  $r$ 은 하나의 노드가 자신이 RTS/CTS를 주고 받는 타이밍에 재전송해야 할 데이터의 수이며  $p$ 는 그 타이밍에 데이터 전송이 성공될 확률이다.  $N$ 은 전체 수신 노드의 수이며, 1 사이클 안에서 고려해야 할 데이터의 수가 되고  $T_{CP}$ 은 contention phase의 시간,  $T_{frame}$ 은 RTS+CTS+데이터+ACK의 시간이 된다.

위 식에서  $T_{CP} + T_{frame}$ 의 시간은 각 노드에서 하나의 최초 데이터 프레임 전송에 소요되는 시간이며,  $\sum_{r=1}^N N C_r p^{N-r} (1-p)^r$ 은 재전송이 일어날 확률이 된다.  $p^{N-r} (1-p)^r$ 은  $r$  개의 데이터를 받지 못할 확률이며, 이러한 형태가 나올 수 있는 조합을 모두 고려하여 하나의 데이터를 수신하기까지의 재전송 횟수  $1/p$ 를 곱한 값이  $\sum_{r=1}^N N C_r p^{N-r} (1-p)^r$ 로서 전체 평균 재전송 횟수가 된다. 이러한 재전송 데이터 프레임에 대해서는 contention phase 없이  $T_{frame}$ 의 시간만을 고려한다. (4)의 식에서의 모든 가정은  $n$ 개 수신 노드가 존재하고  $n$ 개 데이터가 전송된다고 가정하므로 평균 데이터 전송 시간에  $n$ 을 곱하여 줌으로써  $n$ 개 노드에 대한  $n$ 개의 데이터 전송시간  $nT_d(n)$ 을 구할 수 있다.

$$T_d(N) = n(T_{CP} + T_{frame} + \sum_{r=1}^N N C_r p^{N-r} (1-p)^r T_{frame}(r/p)) \quad (5)$$

### 3.2.1 BMW의 평균 데이터 전송 시간

BMMM에서는 하나의 데이터를 모든 노드에게 전송하는 시간을 통해 평균 데이터 전송 시간을 유도한다. BMMM의 경우에 평균 데이터 전송 시간을 BMW와 같이 전체  $n$ 노드 중 한 개의 노드에 대해  $n$ 개 데이터가 데이터를 수신하는 시간을 구하는 방식으로 직접 유도할 수 없다. BMMM은 하나의 데이터를 모든 노드에게 전송한 후 다음 데이터를 모든 노드에게 전송하는 방식을 가지기 때문이다. 대신 하나의 데이터를 모든 노드에게 전송하는 시간을 통해 평균 데이터 전송 시간을 유도하여야 한다. 데이터를 순차적으로 전송하는 특징 때문에  $n$ 개 데이터를  $n$ 개 노드에게 전송하는 시간은 하나의 데이

터를  $n$ 개 노드에 전송하는 시간에 단순히  $n$ 을 곱하여서 구할 수 있다. 평균 데이터 전송 시간은 전체  $n$ 개 노드가  $n$ 개의 데이터를 수신하는 시간에  $n$ 을 나눈 값이므로 BMMM에서 하나의 데이터를  $n$ 개 노드에게 전송하는 시간을 BMW에서의 평균 데이터 전송 시간과 직접 비교할 수 있게 된다.

BMMM에서 하나의 데이터를  $n$ 개 노드에게 전송하는 시간 역시 재전송 확률을 이용하여 유도한다. BMMM에서는 하나의 데이터에 대해서 한번의 contention phase를 가진다. 그리고 하나의 노드만 존재할 경우 하나의 데이터를 완전히 수신하기까지의 평균 횟수는  $1 + (1-p) + (1-p)^2 + (1-p)^3 + (1-p)^4 + \dots = 1/p$ 의 값을 가지게 된다. 이에 따른 전송 시간은 아래와 같은 값을 가진다.

$$T_d(1) = (T_{CD} + T_{frame}(1))1/p \quad (6)$$

다음으로 두 개의 노드가 존재할 때 전송 시간  $T_d(2)$ 을 고려하고자 한다. 두 개의 노드가 존재하는 경우 세 가지 경우로 생각할 수 있다. 먼저, 두 노드가 기본적인 최초의 데이터 전송을 가지는 시간, 두 번째로 최초 전송 후에 두 노드 중 한 노드만 재전송 과정을 거치는 경우, 마지막으로 두 노드 모두가 재전송 과정을 거치는 경우로 나누어 전송 시간을 구할 수 있다. 최초 데이터 전송 과정에서 contention phase( $T_{CD}$ )와 두 개의 데이터가 포함된 데이터 전송 단계( $T_{frame}(2)$ )의 값을 가진다. 두 번째의 경우는  $2 * p(1-p)$ 의 확률로 발생하게 되며 이때에는  $(T_{CD} + T_{frame}(1))1/p = T_d(1)$  만큼의 시간이 추가적으로 소요된다. 세 번째 경우는  $(1-p)^2$ 의 확률로 발생하게 되며 재귀적으로  $T_d(2)$ 의 값을 가진다. 모든 경우를 고려하였을 때 두 개의 노드가 하나의 데이터를 전송 완료하는 시간은 다음과 같다.

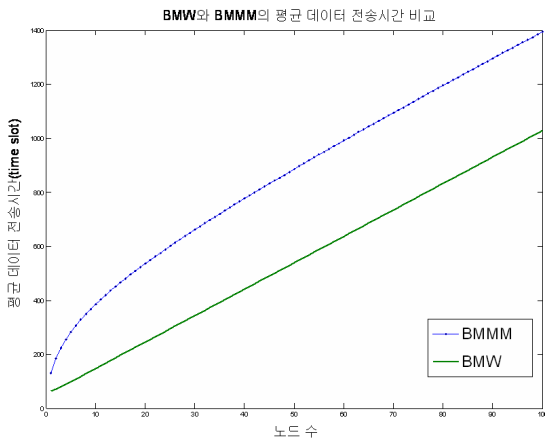
$$T_d(2) = T_{CD} + T_{frame}(2) + {}_2C_1 p(1-p) T_d(1) + {}_2C_2 (1-p)^2 T_d(2) \quad (7)$$

이와 같은 방식으로  $n$ 개의 노드가 존재할 때의 값,  $T_d(n)$ 을 구하려면, 두 개의 노드만이 존재할 경우와 같이, 최초의 데이터 전송을 고려하고, 다음으로 재전송이 일어날 수 있는 경우를 재전송 대상이 되는 노드의 수들로 나누어 생각하여 구할 수 있다. 이 때 최초의 데이터 전송의 경우는 한번의 contention phase와  $n$ 개의 RTS/CTS/ACK, 하나의 데이터를 포함하는 시간 값,  $T_{frame}(n)$

의 합의 형태인  $T_{CD} + T_{frame}(n)$ 가 된다. 다음으로 재전송이 일어나는 노드의 수가 1,2,3,...,n일 경우, 각각의 경우는 데이터 전송을 완료하기 위해  $T_d(1), T_d(2), \dots, T_d(n-1), T_d(n)$ 만큼의 시간이 필요하게 되고, 각각의 경우는  ${}_n C_1 p^{n-1}(1-p), {}_n C_2 p^{n-2}(1-p)^2, {}_n C_3 p^{n-3}(1-p)^3, \dots, {}_n C_n (1-p)^n$ 의 확률로 발생한다. 이를 식으로 나타내면

$$T_d(n) = T_{CD} + T_{frame}(n) + \sum_{r=1}^n ({}_n C_r p^{n-r} (1-p)^r T_d(r)) \quad (8)$$

이 된다. 이 값은 n개의 수신 노드에게 하나의 데이터를 전송하게 될 때의 평균 시간이 되며, n개 데이터를 전송하게 될 경우는 이러한 상황의 n번 반복이므로  $nT_d(n)$ 은 n개 노드로 n개의 데이터를 전송하는데 걸리는 평균 시간 값이 된다. BMMM의  $nT_d(n)$ 과 BMW에서  $nT_d(n)$ 은 n개 노드로 n개의 데이터를 전송하는데 걸리는 평균 시간 값이 된다.



( 그림 5 ) BMW, BMMM, BMMM-en의 평균 데이터 전송시간 비교(  $p = 0.9$  )

(그림5)에서는 전체 노드 수 n값에 따른 BMMM, BMMM-en, BMW의 평균 전송 시간을 나타내었으며, 전송 성공 확률(p)는 BMMM 논문의 시뮬레이션에서 사용한 값과 같은  $p = 0.9$ 를 사용하였다. 그 외 contention phase, RTS, CTS, DATA, RAK, ACK의 값은 [4]의 값을 참조하였다.

#### 4. 결론

[1]에서 언급한 것과는 달리 위의 분석에서 BMW가 BMMM과 비교했을 때, 하나의 데이터에 대한 contention수가 훨씬 적음을 알 수 있다. 이는 BMW에

대한 다른 논문 [2],[3]에서도 확인할 수 있다.

이러한 contention phase에 대한 잘못된 분석을 바로 잡고 데이터 전송의 상황을 새로이 고려함으로써, 평균 데이터 전송시간을 분석하였다. BMW와 BMMM의 평균 데이터 전송시간을 비교함으로써 연속된 데이터가 생성되는 환경에서 BMW 프로토콜이 BMMM 프로토콜에 비해 간단한 프레임 구조를 가지면서도, 더욱 효율적임을 알 수 있다.

#### 5.참고문헌

[1] M. T. Sun, L. Huang, A. Arora, and T. H. Lai. Reliable MAC layer multicast in IEEE 802.11 wireless networks. In *Proc. the International Conference on Parallel Processing (ICPP) 2002*

[2] K. Tang and M. Gerla, "Random Access MAC for Efficient Broadcast Support in Ad Hoc Networks," *Proc. IEEE WCNC 2000*, pp. 454-459, Sep. 2000.

[3] Min-Te Sun, Lifei Huang, Anish Arora, Ten-Hwang Lai. RMAC: A Reliable Multicast MAC Protocol for Wireless Ad Hoc Networks in *Proceedings of the 2004 International Conference on Parallel Processing (ICPP'04)*.

[4] Vikram Shankar. ENHANCING THE RELIABILITY OF MEDIUM ACCESS CONTROL LEVEL WIRELESS MULTICAST A Dissertation Presented in Partial Fulfillment. of the Requirements for the Degree. Doctor of Philosophy. ARIZONA STATE UNIVERSITY