

# 무선 메시 네트워크를 위한 다중 인터페이스 MAC 프로토콜에 관한 연구

A Study on Multi-Interface MAC Protocols in Wireless Mesh Network

김준호, 김형진, 전보규, 김영범  
(건국대학교 정보통신대학 전자공학부)

현재 무선 랜 기반의 WMN은 도심지와 같은 통화량이 많은 지역('핫존')에서 서비스 되고 있다. 다수 핫존을 서비스하는 액세스 포인트(AP)별로 인터넷 망에 유선 접속할 경우, 망 접속 비용이 높지만, WMN 기술을 이용하면 핫존별로 산재한 트래픽의 집중을 통해 자원을 대폭 줄일 수 있다. 또한, IEEE 802.11 표준은 가용한 채널이 다양한 직교주파수로 되어있어 IEEE 802.11b에서는 2.4GHz에서 간섭 없는 3개의 채널을 제공하고 IEEE 802.11a는 5GHz에서 간섭 없는 12개의 채널을 제공한다. 근접한 access point (AP)주변에는 다른 채널을 할당함으로써 근접한 AP사이에서의 간섭을 막을 수 있다. IEEE 802.11 표준은 동시 사용에 1개의 채널만 사용하게 되어있어 가용한 채널을 동시에 간섭 없이 사용함으로써 전체적인 throughput을 증대시킬 수 있어 IEEE 802.11 표준 기반의 MAC 프로토콜의 수정만으로 상, 하 레이어의 수정 없이 향상된 멀티채널을 위한 MAC 프로토콜을 제시한다.

Key Words : MAC, MMAC, WMN, IEEE 802.11, ATIM WINDOW, INTERFRAME SPACING

## 목 차

- I. 서론
- II. 제안방식
- III. 결론

### I. 서론

#### 1. Interframe spacing

Interframe spacing이란 패킷을 전송하기 전에 한 노드는 채널이 idle 상태여도 작은 시간의 기간 동안 기다리는 것을 말한다. 4가지 각기 다른 기간이 각각의 패킷을 채널을 사용하기 위한 다른 우선순위를 갖도록 해주는 SIFS, PIFS, DIFS, EIFS는 4가지 interframe spacing이 있다. 노드는 RTS 패킷을 전송하기 전에 DIFS 동안 기다리고 CTS나 ACK 패킷을 전송하기 전에 SIFS를 기다린다. ACK 패킷은 채널을 RTS 패킷을 데이터 패

킷과 경쟁을 할 때 채널을 사용할 수 있는데 이것은 SIFS 기간이 DIFS보다 기간이 작기 때문이다.

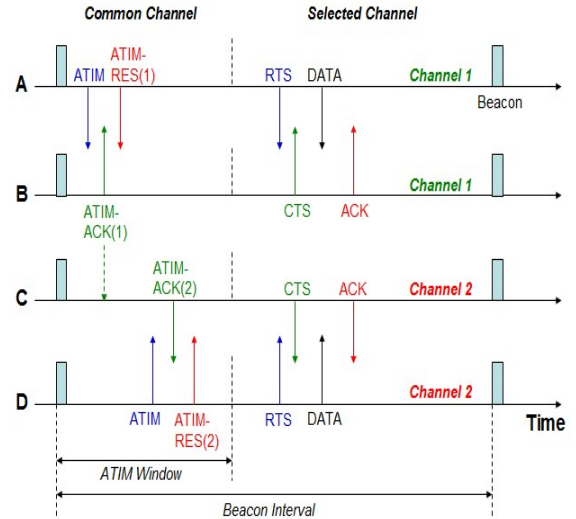
#### 2. MMAC

MMAC은 link-layer에 있는 한 노드에 하나의 인터페이스를 가지는 Multi-channel 프로토콜이다. 한 개의 인터페이스를 가지고 있는 한 노드는 한 번에 한 개의 채널만을 감지 할 수 있으므로 여러 채널을 사용하기 위해서는 인터페이스를 채널에 스위칭 해줘야 한다.

이러한 통신을 하기 전에 서로 control message 를 사용해서 어느 채널을 사용할지 어느 노드와 통신을 할지 정해야 한다. 먼저 MMAC에 서는 시간을 beacon이라는 interval들로 나뉘어 진다. Beacon interval이 시작되면 ATIM window가 시간동안 각 노드들은 각각 어느 노드와 어떤 채널을 사용할 것을 정하게 된다. 그러기 위해서는 모든 노드들이 이 시간 동안에는 한 채널에 모여서 모든 노드들이 control message를 받을 수 있어야 한다. 다음 ATIM window가 끝나면 각각의 노드는 서로 어느 노드와 어느 채널을 사용할지가 결정되었기 때문에 각각의 노드는 각기 자신이 사용할 채널에 인터페이스를 스위칭하고 나머지 beacon interval 동안 서로 데이터를 주고받게 되고 통신이 끝나게 되면 다시 새로운 beacon interval이 시작되게 된다.

기존의 단일 채널, 한 개의 인터페이스를 사용한 프로토콜에 비해 단 1개의 인터페이스만을 사용하여 많은 채널을 스위칭 할 수 있게 사용할 수 있으므로 비용적인 면에서 경제적인 효과가 있고 한 번에 동시에 서로 다른 노드들이 통신을 할 수 있는 장점을 가지게 된다.

ATIM window 기간 동안 여러 노드가 한 채널에 모여서 데이터를 주고받게 된다. 이러한 노드의 수가 늘어나게 되면 채널이 하나이기 때문에 병목현상이 일어나는 단점을 가지는데 이유는 한 노드가 control message를 보내게 되는 동안 다른 노드들은 아무것도 할 수 없이 기다려야 하기 때문이다.



<그림 1>

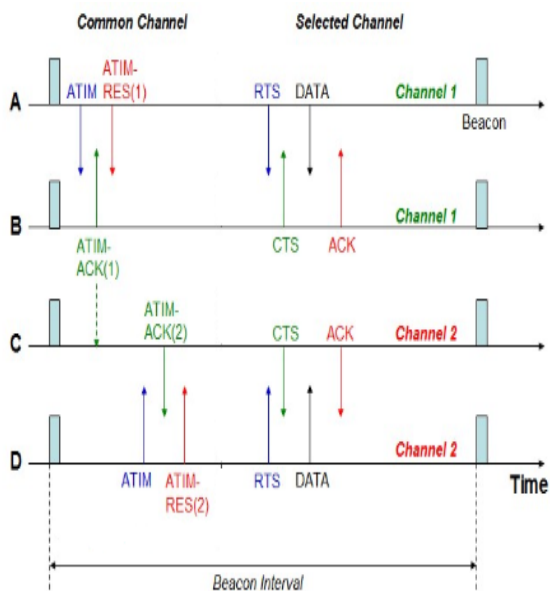
<그림 1>에서 보는 MMAC은 노드 A, B, C, D가 있고 각각의 노드는 하나의 인터페이스를 갖고 있으며, 채널은 채널1과 채널2가 있다. 먼저 beacon interval이 시작되면서 노드 A, B, C, D들이 모두 채널1에 모이게 된다. 그리고 노드 A가 ATIM message를 노드 B와 통신하겠다고 보내게 된다. 이 때 노드 A는 충돌을 피하기 위해 랜덤 딜레이 후에 메시지를 보내게 된다. 이 message를 노드 B가 받게 되고, 노드 B는 채널 1을 하겠다는 ATIM-ACK message를 보내게 된다. 노드 A는 다시 ATIM-RES라는 confirm을 위한 message를 노드 B에 보내게 된다. 노드 A와 B가 교섭을 끝내게 되면 이번에는 노드 D가 같은 방식으로 노드 C와 통신을 하기 위한 교섭을 하게 된다. 그리고 ATIM window가 끝나게 되고 각각의 선택된 채널에서 노드 A와 B는 채널 1에서 노드 C와 D는 채널2에서 각각 통신을 하게 된다.

통신을 하기 위해서 노드 A는 노드 B에 노드 D는 노드 C에 각각 RTS(request to send)라는 메시지를 보내서 데이터를 보내기 전에 수신측에 데이터를 보낸다는 신호를 보내주면 수신측에서

는 송신측에 데이터를 보내라는 CTS (clear to send)라는 메시지를 보내게 되고, 송신측이 메시지를 받게 되면 데이터를 보내게 된다. 수신측에서 데이터 전송이 끝나게 되면 ACK라는 확인 메시지를 보내게 된다. 송신측에서 ACK라는 메시지를 받지 못하면 다시 데이터를 전송하게 된다.

## II. 제안 방식

기존의 MMAC 프로토콜은 노드의 수에 관계없이 일정한 ATIM WINDOW를 사용했다. 노드의 수가 많아지게 되면 채널을 설정할 시간이 부족하게 되고 노드의 수가 적어지면 ATIM WINDOW의 시간이 남게 되어 비효율적이게 된다. 이러한 문제점을 보완하기 위해 ATIM WINDOW를 제거하는 방법에 대해 생각해 보았다.

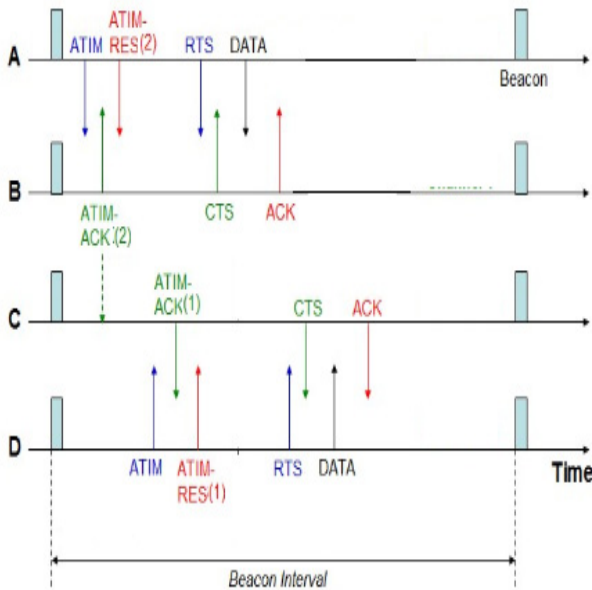


<그림 2>

첫 번째 방법은 <그림 2>에서 ATIM WINDOW를 없애고 ATIM 패킷들에 RTS/CTS, DATA 패킷보다 우선순위를 가질 수 있도록 INTERFRAME SPACING을 두는 방법이다.

그렇게 되면 노드들이 패킷을 보낼게 있는 경우 각 노드에 사용할 채널을 할당해주기 위해 ATIM 패킷을 주고받는 채널에서 ATIM 패킷을 주고받는데 <그림2>을 예를 들어 설명하면 먼저 노드 A가 B에 ATIM 패킷을 보내게 된다. 여기에는 ATIM 패킷을 주고받는데 걸리는 예상시간과 현재 노드A의 PCL(Preferable Channel List)정보를 포함하고 있다. 그럼 C와 D도 그 패킷을 확인하고 데이터 충돌이 일어나지 않도록 그 기간 동안 데이터 전송을 하지 않고 NAV상태로 있게 된다. 노드 B는 ATIM 패킷을 받고 노드A의 PCL과 노드B의 PCL의 상태를 확인한 뒤 더 좋은 상태의 채널을 사용하게 된다. 여기서 채널1과 2가 둘 다 medium 상태이므로 랜덤하게 채널1을 선택하게 되었고 노드A는 ATIM-RES 패킷을 보내게 되고 데이터를 보내게 될 채널로 스위칭을 하게 되고 노드B는 패킷을 받은 때 스위칭을 한다. 이때 채널이 비어있는 것을 확인하고 노드 C와 노드 D가 ATIM패킷을 주고 받게 되는데 여기서 ATIM패킷이 데이터 패킷이나 RTS/CTS 패킷보다 우선순위를 가지게 되므로 충돌할 가능성은 적어진다. 만약에 C와D가 보낼 패킷이 없는 경우는 노드 A와 노드B가 데이터 패킷을 보내게 된다. 그렇기 때문에 무조건 일정한 ATIM 윈도우만큼의 시간을 기다린 후에 데이터 전송을 하는 경우보다 더 효율적임을 알 수 있다. 여기의 예에서는 노드C와 노드D가 ATIM패킷을 주고 받게 되고 이 노드들은 채널1을 노드A와 노드B가 사용한다는 것을 ATIM패킷을 통해 알았기 때문에 채널 1은 PCL이 LOW 상태로 낮아져서 채널2를 사용하게 된다. 역시 ATIM-RES 패킷을 받고 채널2로 스위칭 한 뒤 CSMA-CA 방식에 의해 데이터를 주고받게 된다. 여기서 ATIM 패킷이 노드 A와 노드B가 데이터 전송 시에 노드C와 노드D가 보내야 된다면 이미 채널

은 사용 중임을 알게 되고 딜레이 되므로 문제가 발생하지 않는다. 만약에 채널이 비게 된다면 데이터를 전송하게 되고 beacon interval 안에 보내지 못하게 되면 송신측에서 ACK 패킷을 받지 못했으므로 다음 인터벌에 데이터를 재전송하게 된다.

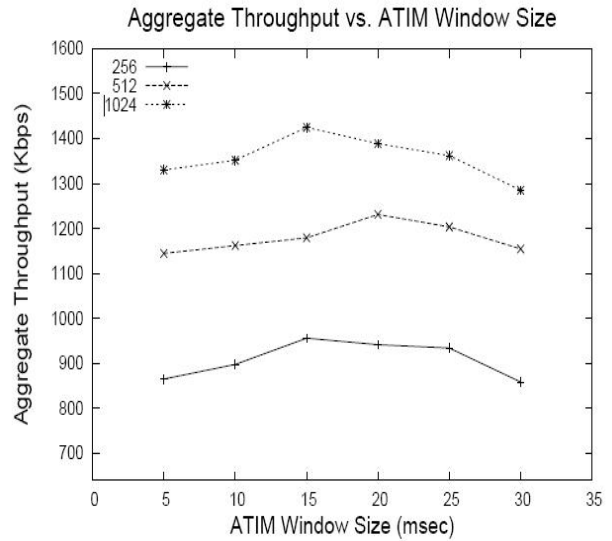


<그림 3>

두 번째 방법은 <그림 3>에서 보는 것과 같이 첫 번째 방법과 동일하지만 첫 번째 방법에서 노드 A와 B처럼 먼저 ATIM패킷을 주고받는 경우 채널을 ATIM패킷을 주고받는 채널을 선택할 경우 다른 노드들이 ATIM패킷을 주고받을 때까지 데이터 전송을 기다려야 하기 때문에 비효율적이다.

이런 단점을 보완하기 위해 리셋 상태의 PCL을 ATIM패킷을 주고받는 채널의 경우 LOW상태로 만들어 좀 더 효율적이게 된다. 그러나 알고리즘 단계에서 좀 더 복잡해지는 단점이 있지만 현재의 기술력으로 이것을 극복할 수 있겠다.

#### IV. 결론



<그림 4>

본 논문에서는 기존 MMAC 프로토콜의 ATIM의 고정적인 시간 간격을 제거함으로써 정해진 beacon interval 안에서 통신 노드간 좀 더 효율적인 데이터전송이 가능한 프로토콜을 제안하였다. 위의 <그림 4>는 MMAC와의 기타 동일 조건에서 ATIM window 길이를 조정하여 위의 시뮬레이션 결과로부터 ATIM window 길이에 따라 Throughput이 변화하는 것을 볼 수 있다. 하지만 제안된 프로토콜은 ATIM window가 없으므로 가장 좋은 throughput을 가질 수 있음을 알 수 있다.

#### 참고문헌

- [1] IEEE Standard for Wireless LAN-Medium Access Control and Physical Layer Specification, P802.11, 1999.
- [2] J. So and N. H. Vaidya, "Multi-channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals using a Single Transceiver," in Mobihoc, 2004.
- [3] Multi-Channel Mesh Networks: Challenges and Protocols
- [4] Data Communications and Networking , Behrouz A. Forouzan