무선 애드혹 네트워크에서 채널 효율성 향상을 위한 TDMA 기반의 멀티채널 MAC 프로토콜

준회원 김 준 호*, 정회원 최 재 각*, 종신회원 유 상 조**

TDMA based Multi-channel MAC Protocol for Improving Channel Efficiency in Wireless Ad Hoc Networks

Jun-Ho Kim* *Associate Member*, Jae-Kark Choi* *Regular Member*, Sang-Jo Yoo** *Lifelong Member*

요 약

본 논문은 무선 애드혹 네트워크에서 채널 효율성 및 네트워크 성능 향상을 위한 멀티채널 MAC 프로토콜을 제안한다. 멀티채널 MAC 프로토콜 설계에는 랑데부 문제(rendezvous problem)와 멀티채널 히든 노드 문제 등 두 개의 주요한 문제가 있다. 기존에 제안된 여러 멀티채널 MAC 프로토콜에서는 하나의 채널을 공통 컨트롤 채널로 사용하여 컨트롤 패킷 교환을 통해 이러한 문제들을 해결하였는데, 이것은 데이터 채널이 증가할수록 높은 경쟁으로 인한 공통 컨트롤 채널의 병목 현상을 초래하여 비효율적인 데이터 채널 이용의 원인이 된다. 본 논문에서 제안하는 멀티채널 MAC 프로토콜은 멀티채널 히든 노드 문제를 해결하기 위해 TDMA 방식을 사용하였고, 데이터를 동시에 송수신할 수 있도록 하여 네트워크 성능을 높인다. 또한 공통 컨트롤 채널을 사용하지 않기 때문에 공통 컨트롤 채널 병목 현상이 발생하지 않고, 전송 또는 수신할 데이터가 없는 노드는 슬립(sleep) 상태를 유지하도록 하여 에너지 절감(energy savings)이 가능하다. 모의실험결과는 제안한 MAC 프로토콜이 기존의 방법에 비해 네트워크 성능 및 채널 효율성을 향상시키고 에너지를 절감할 수 있다는 것을 보여준다.

Key Words : Multi-channel MAC, Ad Hoc Networks, TDMA, Hidden Node Problem, Energy Savings

ABSTRACT

In this paper, we propose a multi-channel MAC protocol to improve the channel efficiency and network performance in wireless ad hoc networks. There are two main problems encountered in designing multi-channel MAC protocols. The first problem is the rendezvous problem and the second is multi-channel hidden node problem. In order to solve these problems, most of previous researches that have considered multi-channel MAC protocols use a common control channel to exchange control packets. However, they have a bottleneck problem at common control channel as increasing the number of data channels. The proposed MAC protocol solves the multi-channel hidden node problem using a TDMA scheme and increases the network throughput because transmitting and receiving data at the same time is possible. Also, since there is no common control channel, the network does not suffer from the common control channel saturation problem. Moreover, it achieves energy savings by allowing nodes that are not involved in communication to go into sleep mode. Simulation results show that the proposed MAC protocol improves the network throughput and channel efficiency and provides energy savings.

※ 본 연구는 지식경제부 및 정보통신진흥연구원의 대학 IT 연구센터 지원 사업의 연구결과로 수행되었음(IITA-2009-C1090-0902-0019).
※ 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No. 2009-0079228).
* 인하대학교 정보통신대학원 멀티미디어통신망 연구실 (paust1004@naver.com, jkc@inha.edu)
** 인하대학교 정보통신대학원 교수 (sjyoo@inha.ac.kr)
논문번호 : KICS2009-09-420, 접수일자 : 2009년 9월 18일, 최종논문접수일자 : 2010년 1월 25일

Ⅰ. 서 론

멀티채널 환경에서는 인접 노드들이 서로 다른 채널에서 데이터를 전송할 경우 노드 간의 충돌 및 간섭이 일어나지 않기 때문에 다수의 데이터를 동시에 전송하는 것이 가능하여 하나의 채널을 사용하는 것보다 네트워크 성능을 크게 향상시킬 수 있다. 따라서 네트워크 성능을 향상시키기 위해 다수의 채널을 사용하는 여러 MAC 프로토콜들이 제안되었다. [1]과 [2]는 하나의 인터페이스를 사용하는 MAC 프로토콜이다. 인터페이스는 채널을 이동하면서 한 번에 하나의 데이터를 전송하거나 수신한다. 그러나 여러 채널에서 데이터를 동시에 전송하거나 수신하지 못하므로 채널을 효율적으로 이용하지 못한다. 최근 하드웨어 비용의 감소로 인해 다수의 인터페이스를 사용하는 MAC 프로토콜들이 많이 제안되고 있다. [3]과 [4]는 다수의 인터페이스를 사용하는 MAC 프로토콜로, 동시에 데이터를 전송하고 수신하는 것이 가능하다. 그러나 하나의 채널을 공통 컨트롤 채널로 사용하여 컨트롤 패킷 교환을 하기 때문에 데이터 채널이 증가할수록 컨트롤 채널에서 높은 경쟁이 발생하여 공통 컨트롤 채널의 병목 현상을 초래한다. 이것은 데이터 채널을 비효율적으로 이용하는 원인이 된다.

HMCP(Hybrid Multi-channel Protocol)[5],[6] 또한 다수의 인터페이스를 사용한다. 각각의 인터페이스는 데이터를 전송할 때만 사용하는 송신용 인터페이스와 데이터를 수신할 때만 사용하는 수신용 인터페이스로 구분된다. HMCP는 이렇게 다수의 인터페이스를 사용하여 동시에 데이터를 전송하고 받을 수 있기 때문에 하나의 인터페이스를 사용하는 것보다 채널을 효율적으로 이용할 수 있다. 또한 공통 컨트롤 채널을 사용하지 않고, 각 노드마다 여러 채널 중 하나의 채널을 수신 채널로 할당받아 사용하기 때문에 채널 병목 현상도 발생하지 않는다. 그러나 HMCP에서는 공통 컨트롤 채널을 사용하지 않아서 이웃 노드간 수신 채널이 다르기 때문에 데이터 전송을 위해 컨트롤 패킷 교환 시 이웃 노드가 RTS(Request To Send) 또는 CTS (Clear To Send)를 듣지 못하는 경우가 생겨 데이터와 컨트롤 패킷이 충돌하는 멀티채널 히든 노드 문제가 발생한다.

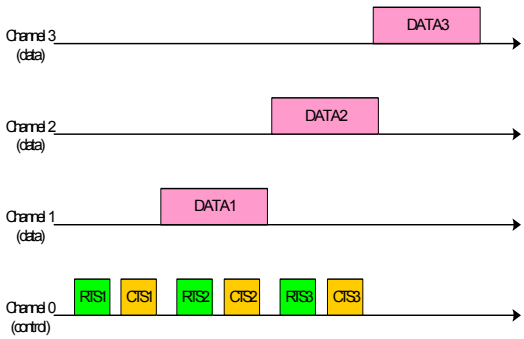
본 논문에서 제안하는 멀티채널 MAC 프로토콜은 채널 병목 현상 및 HMCP에서 발생하는 멀티채널 히든 노드 문제를 해결하여 채널 효율성 및 네트워크 성능을 높이는데 목적을 두고 있다. 제안하는 프로토콜은 노드 당 2개의 인터페이스를 사용하여 동시에 데이터를 전송 및 수신할 수 있도록 하여 네트워크 성능을 높였다. 또한 제안한 MAC 프로토콜은 HMCP와 동일하게 데이터 전송을 위해 공통 컨트롤 채널을 사용하지 않고 각 노드마다 수신 채널을 할당받아 이웃 노드들에게 자신의 수신 채널을 알리는 방식을 사용하여 랑데부 문제를 해결하였고, TDMA 방식을 사용하여 HMCP에서 발생하는 멀티채널 히든 노드 문제를 해결하였다. 마지막으로 데이터를 전송 또는 수신하지 않는 노드는 슬립 상태를 유지하기 때문에 에너지 절약이 가능하다.

본 논문의 구성은 다음과 같다. 제 2장에서는 멀티채널 MAC 프로토콜의 동작 원리 및 여러 채널 할당 방법을 소개하고, 그 중 혼합 채널 할당 방법을 사용한 HMCP와 이 HMCP에서 발생하는 멀티채널 히든 노드 문제에 대해 설명한다. 제 3장에서는 TDMA 방식을 사용한 멀티채널 MAC 프로토콜을 제안하고, 제 4장에서는 NS-2 시뮬레이션을 통해 제안한 MAC 프로토콜과 HMCP를 비교 및 성능을 평가한다. 마지막으로 제 5장에서는 본 연구의 결론을 맺는다.

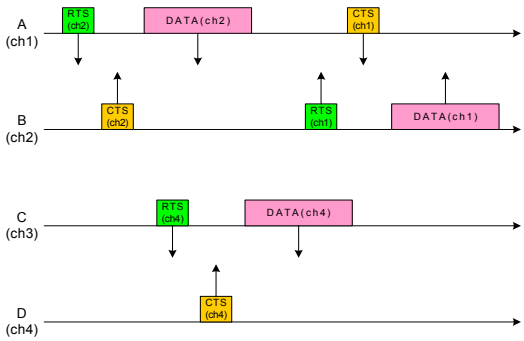
Ⅱ. 관련 연구

2.1 멀티채널 MAC 프로토콜의 동작 원리

멀티채널 MAC 프로토콜에서 각 노드는 데이터 전송을 하기 위해 다른 노드의 수신 채널을 알아야 한다(rendezvous problem). 여기에는 single rendezvous 프로토콜과 multiple rendezvous 프로토콜 이렇게 두 종류의 프로토콜로 나누어진다. 그림 1은 single rendezvous와 multiple rendezvous 프로토콜의 예를 보여준다. 그림 1-(a)는 지정된 컨트롤 채널을 사용하는 프로토콜을 보여준다. 그림 1-(a)에서 채널 0은 컨트롤 채널이고, 채널 1, 2, 3은 데이터 전송 채널이다. 하나의 인터페이스는 컨트롤 채널 0에 대기하고 있고, 다른 인터페이스는 데이터 채널을 변경하면서 데이터를 전송하는데 사용된다. 만일 노드 A가 노드 B에 데이터 전송하기를 원한다면 노드 A와 B는 컨트롤 채널 상에서 RTS와 CTS 교환을 하여 노드 A와 B 모두 이용 가능한 채널 상에서 데이터를 전송한다. 그림 1-(b)는 지정된 컨트롤 채널을 사용하지 않고, multiple rendezvous 채널을 사용한다. 그림 1-(b)에서 노드 A, B, C, D는 각각 channel 1, 2, 3, 4에 대기하고 있다. 노드 A가 B



(a) 2개의 인터페이스를 사용하는 single rendezvous 프로토콜



(b) 1개의 인터페이스를 사용하는 multiple rendezvous 프로토콜

그림 1. single rendezvous와 multiple rendezvous 프로토콜의 예

에게 데이터 전송을 원한다면 노드 A는 노드 B의 수신 채널 2로 채널을 변경 후 RTS와 CTS를 교환하여 데이터를 전송한다. 노드 C가 D에게 전송할 때도 마찬가지로 노드 C는 노드 D의 수신 채널 4로 채널을 변경한 후 RTS와 CTS를 교환하여 데이터를 전송한다.

2.2 멀티채널 MAC 프로토콜에서의 채널 할당 방법

멀티채널 멀티인터페이스 MAC 프로토콜과 관련된 채널 할당 방법은 정적 채널 할당(static channel assignment), 동적 채널 할당(dynamic channel assignment), 혼합 채널 할당(hybrid channel assignment) 이렇게 3가지로 분류된다. 정적 할당 전략은 각 인터페이스를 채널에 영구적 또는 오랜 시간동안 할당하는 것이다[7-9]. 이 전략은 인터페이스가 채널에 고정되어 있어 랑데부 문제(rendezvous problem)와 멀티채널 히든 노드 문제가 발생하지 않지만 채널수만큼 인터페이스를 요구하기 때문에 하드웨어 비용이 증가한다. 동적 할당 전략은 인터페이스가 채널 사이를 스위칭(switching) 하는 것이 가능하여 적은 인터페이스를 가지고도 다수의 채널이 이용 가능하다[10],[11]. 따라서 이 전략을 사용하면 하드웨어 비용이 감소하고, 파워 소비가 줄어들지만 랑데부 문제와 멀티채널 히든 노드 문제를 해결하는데 어려움이 있다.

최근 정적 할당과 동적 할당 방법을 개선하기 위해 혼합 채널 할당 방법이 제안되었다. 혼합 채널 할당 방법에서 각 노드는 다수의 인터페이스를 가지고 있고, 시간 동기화(time synchronization)가 필요하지 않다. 이 방법은 정적 할당과 동적 할당 방법을 결합한 것으로 2개의 인터페이스가 있다고 가정할 때 하나의 인터페이스는 정적 할당 방법을 사용하고, 다른 인터페이스는 동적 할당 방법을 사용하는 것이다. 즉, 혼합 채널 할당 방법은 하나의 인터페이스를 공통 컨트롤 채널 또는 특정 채널에 할당하고, 다른 인터페이스는 채널 사이를 자유롭게 이동하면서 데이터를 전송할 수 있도록 하여 랑데부 문제를 해결하고, 다수의 채널을 효율적으로 이용할 수 있도록 하는 것이다. 혼합 채널 할당 방법은 세부적으로 두 개의 방법으로 분류된다. 하나는 공통 채널 정적 할당 방법이고, 다른 하나는 다중 채널 정적 할당 방법이다.

공통 채널 정적 할당 방법은 모든 이용 가능한 채널을 하나의 컨트롤 채널과 여러 데이터 채널로 나눈다. 각 노드는 하나의 컨트롤 인터페이스와 하나의 데이터 인터페이스 이렇게 두 종류의 인터페이스를 가지고 있다. 컨트롤 채널은 컨트롤 인터페이스에 영구적으로 할당되고, 데이터 인터페이스는 모든 데이터 채널 사이를 이동하면서 데이터를 전송하는데 사용된다. 각각의 노드가 통신하는 동안에 컨트롤 인터페이스는 데이터 전송을 위해 데이터 채널에 대한 접근 권한을 갖기 위해 컨트롤 채널 상에서 컨트롤 패킷을 교환하는데 사용된다. 데이터 채널 상에서 트래픽(traffic) 경쟁을 해결하고 멀티채널 히든 노드 문제를 피하기 위해 데이터 채널은 컨트롤 채널에서 미리 예약된다.

다중 채널 정적 할당 방법은 각 노드가 하나의 고정된 인터페이스(fixed interface)와 하나의 전환 가능 인터페이스(switchable interface) 이렇게 두 종류의 인터페이스를 사용한다. 각 노드의 고정된 인터페이스에 영구적 또는 오랜 시간 동안 하나의 채널이 할당되고, 각 노드마다 고정된 인터페이스에 할당되는 채널이 다르다. 전환 가능 인터페이스는 모든 채널 사이를 이동할 수 있다. 만일 전송자 A

가 수신자 B와 통신하기를 원한다면 A는 자신의 전환 가능 인터페이스를 수신자 B의 고정된 인터페이스에 할당되어 있는 채널로 바꾼 후 통신한다. 이 방법은 랑데부 문제를 해결하고, 시간 동기화 또는 컨트롤 채널을 필요로 하지 않는다.

2.3 HMCP(Hybrid Multi-channel Protocol)

다음은 혼합 채널 할당 방법을 사용하는 HMCP에 대해 설명한다. HMCP에서 각 노드는 고정된 인터페이스와 전환 가능 인터페이스라고 불리는 두 종류의 인터페이스를 사용한다. 고정된 인터페이스는 각 노드가 데이터를 수신하기 위해 수신 채널(fixed channel)이라 불리는 특정 채널에 할당되어 데이터를 수신하는데 사용되고, 전환 가능 인터페이스는 전송할 데이터가 있는 경우 이웃 노드의 수신 채널로 채널을 전환하여 데이터를 전송하는데 사용된다. 만일 N개의 채널과 두 개의 인터페이스가 있다고 한다면 고정된 인터페이스에는 채널 m이 할당되고, 전환 가능 인터페이스에는 채널 m을 제외한 나머지 채널이 할당된다. 그림 2는 N개의 채널과 두 개의 인터페이스를 가졌을 때의 채널 할당 예를 나타낸다. 그림 2에서처럼 고정된 인터페이스에 채널 1이 할당되고 전환 가능 인터페이스에 채널 2부터 N까지 동적으로 할당된 경우, 이 노드는 고정된 인터페이스를 사용하여 채널 1을 통해 데이터를 수신하고 전환 가능 인터페이스를 사용하여 채널 1 이외의 채널로 채널을 변경하면서 데이터를 전송한다.

그림 3은 3개의 채널과 2개의 인터페이스를 가진 HMCP의 동작을 나타낸다. 노드 A와 B는 각각 노드 B와 C에 전송할 데이터를 가지고 있다고 가정한다. 그림 3-(a)는 노드 A, B, C 각각의 수신 채널이 다른 경우의 동작 예를 보여준다. 그림 3-(a)에서 노드 A, B, C 각각의 고정된 인터페이스는

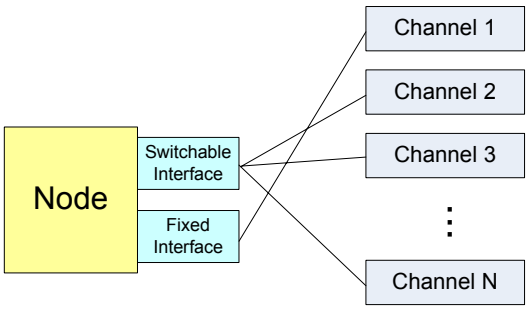
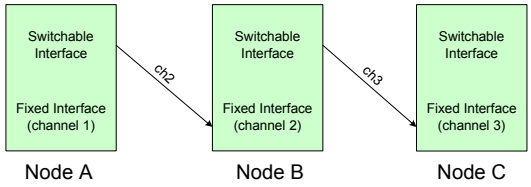
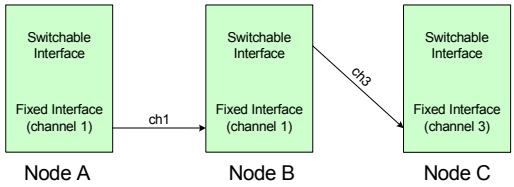


그림 2. 인터페이스의 채널 할당 예



(a) 노드 A, B, C의 수신 채널이 다른 경우



(b) 노드 A, B의 수신 채널이 같은 경우

그림 3. HMCP 동작 예

채널 1, 2, 3에 할당되어 있고, 전환 가능 인터페이스는 채널 3, 2, 1에 할당되어 있다. 노드 A는 노드 B에게 데이터를 전송하기 위해 자신의 전환 가능 인터페이스를 채널 3에서 노드 B의 수신 채널인 채널 2로 전환 후 데이터를 전송한다. 노드 B는 노드 C에게 데이터를 전송하기 위해 전환 가능 인터페이스를 채널 2에서 노드 C의 수신 채널인 채널 3으로 전환 후 데이터를 전송한다. 이 경우 노드 B는 데이터를 전송하는 동시에 수신한다. 그림 3-(b)는 노드 A와 B의 수신 채널이 같은 경우의 동작 예를 보여준다. 그림 3-(b)에서 노드 A와 B는 수신 채널이 1로 동일하기 때문에 그림 3-(a)에서처럼 노드 A가 전환 가능 인터페이스를 채널 1로 전환해서 데이터를 보내지 않고, 노드 A는 자신의 고정된 인터페이스를 사용하여 노드 B에게 데이터를 전송한다. 노드 B는 자신의 전환 가능 인터페이스를 사용하여 채널 3을 통해 노드 C에게 데이터를 전송한다.

HMCP에서 각 노드는 데이터 수신을 위해 자신의 수신 채널을 이웃 노드에게 알려야 한다. 따라서 각 노드는 주기적으로 Hello 패킷을 브로드캐스트(broadcast) 한다. 이 때, HMCP에서 각 노드는 자신의 고정된 인터페이스에 할당된 수신 채널 이외의 채널은 들을 수 없기 때문에 채널을 변경하면서 모든 채널에 Hello 패킷을 브로드캐스트 한다. Hello 패킷에는 자신과 이웃 노드의 수신 채널 정보가 포함되어 있다. 한 노드가 Hello 패킷을 받았을 경우 그 노드는 이웃 노드의 수신 채널 정보를 관리하는 Neighbor-Table을 업데이트 한다. 일정 시

간동안 업데이트 되지 않은 노드는 노드로부터 멀어졌다고 판단하여 테이블에서 제거한다.

그림 4는 HMCP에서의 멀티채널 히든 노드 문제를 보여준다. 3개의 채널과 2개의 인터페이스를 가지고 있고, 노드 A, B, C 각각의 고정된 인터페이스는 채널 1, 2, 3에 할당되어 있다. 노드 A는 자신의 전환 가능 인터페이스를 노드 B의 수신 채널 2로 전환 후 RTS를 보낸다. RTS를 받은 노드 B는 노드 A에게 CTS를 보낸다. 이 때, 노드 C는 채널 3에서 대기하고 있기 때문에 노드 B로부터의 CTS를 듣지 못한다. 즉, 노드 C는 노드 B가 채널 2에서 통신 중인 것을 알지 못한다. 따라서 노드 C는 노드 B와 통신하기 위해 채널 2로 RTS를 보내게 되고, 이것은 노드 B에서 RTS와 DATA의 충돌을 초래한다.

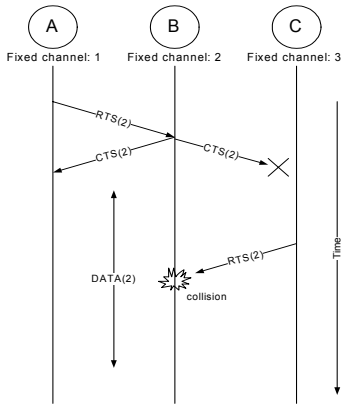


그림 4. HMCP에서의 멀티채널 히든 노드 문제

Ⅲ. TDMA 기반의 멀티채널 MAC 프로토콜

본 논문에서 제안하는 TDMA 기반의 멀티채널 MAC 프로토콜은 혼합 채널 할당 방법을 사용하는 프로토콜로 고정된 인터페이스와 전환 가능 인터페이스 이렇게 2개의 인터페이스를 사용한다고 가정한다. HMCP와 유사하게 고정된 인터페이스는 수신 채널에 할당하여 데이터를 수신하는데 사용하고, 전환 가능 인터페이스는 자신의 수신 채널 이외의 채널로 채널을 변경하면서 데이터를 전송하는데 사용한다. 모든 노드는 sync 구간에 시간 동기화 된다고 가정한다.

3.1 프로토콜 프레임 구조

그림 5는 제안한 TDMA 기반의 멀티채널 MAC 프로토콜 프레임 구조를 보여준다. 그림 5에서 보는 것처럼 슈퍼프레임은 M개의 *communication window*로 구성되어 있고, 각 *communication window*는 synchronous beacon, neighbor discovery, channel and slot negotiation, data transmission으로 구성된다.

Sync 구간에서 모든 노드는 자신의 고정된 인터페이스를 수신 채널로부터 default 채널로 변경하고, 비콘 전송(beacon transmission)에 의해 동기화된다.

Communication window 1의 neighbor discovery 길이는 k개의 슬롯(slot)들로 구성되어 있고, *communication window* 2부터 M까지는 k/2개의 슬롯들로 구성되어 있다. 이것은 최초에는 모든 노드가 자신의 수신 채널을 알리기 위해 Hello 패킷을

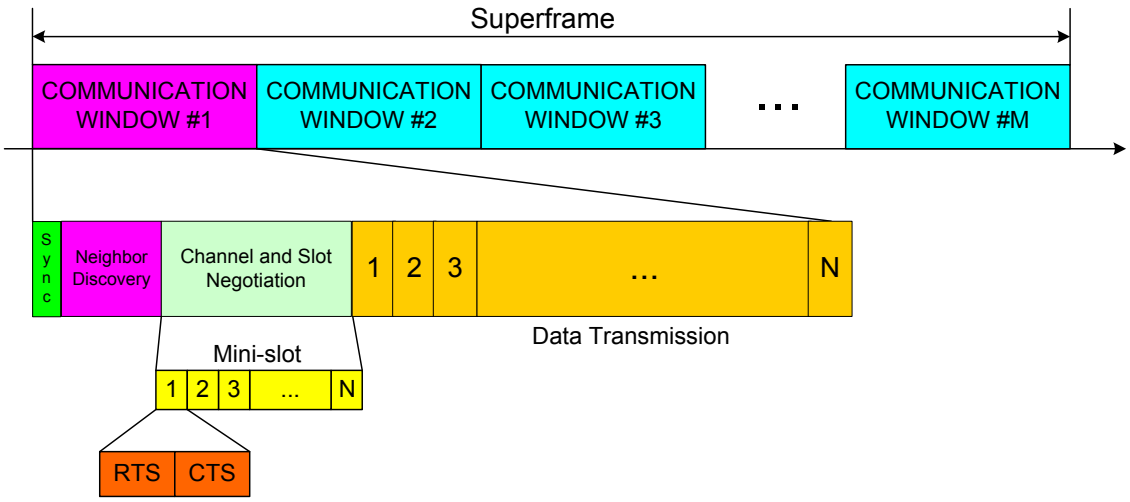


그림 5. 제안한 MAC 프로토콜의 프레임 구조

브로드캐스트 해야 하지만 이후에는 자신의 수신 채널이 바뀌었거나 이웃 노드의 수신 채널이 변경되어 Neighbor-Table을 업데이트 할 필요가 있는 경우, 그리고 새로운 노드가 추가되거나 자신에게서 멀어진 노드를 엔트리(entry)에서 삭제할 경우에만 Hello 패킷 브로드캐스트가 필요하기 때문이다. 따라서 *communication window* 1의 neighbor discovery 구간에서는 모든 노드가 자신의 수신 채널을 알리기 위해 Hello 패킷을 브로드캐스트 하고, *communication window* 2부터 M의 neighbor discovery 구간에서는 자신의 수신 채널이 변경되었거나 Neighbor-Table이 변경된 노드만 Hello 패킷을 브로드캐스트 한다. 이것은 노드의 평균 이동성이 낮을 경우 시간이 지날수록 각 노드가 선택한 수신 채널이 네트워크에 고르게 분포하여 수신 채널 변경이 거의 일어나지 않기 때문에 불필요한 Hello 패킷의 전송을 줄일 수 있다. 그래서 본 논문은 neighbor discovery 크기를 노드 수에 따라 다르게 설정하고, *communication window* M의 반복 주기를 노드의 평균 이동성에 따라 다르게 설정할 수 있게 제안한다.

Channel and slot negotiation 구간은 데이터 전송 구간의 데이터 슬롯(data slot)의 수와 동일한 N개의 mini-slot으로 구성된다. Mini-slot은 RTS와 CTS 구간으로 분리된다. RTS 구간은 backoff 슬롯과 RTS 슬롯으로 구성되는데, 이것은 backoff scheme을 사용하여 충돌을 줄이기 위한 것이다. 트래픽 로드가 적을 경우 channel and slot negotiation 구간의 크기가 크다면 많은 mini-slot이 낭비될 수 있다. 그래서 본 논문은 트래픽 로드에 따라 channel and slot negotiation 구간의 크기를 다르게 설정할 수 있게 제안한다.

데이터 전송 구간은 N개의 데이터 슬롯으로 구성되고, mini-slot을 통해 채널과 슬롯을 예약한 노드들이 데이터를 전송하는데 사용된다.

3.2 Neighbor Discovery

Neighbor discovery 기간 동안 모든 노드는 지정된 default 채널에서 대기한다. 그리고 이 기간에 각 노드는 Hello 패킷을 브로드캐스트 한다. Hello 패킷은 노드의 수신 채널과 이웃 노드들의 수신 채널 정보를 가지고 있는 Neighbor-Table을 포함한다. 한 노드가 이웃 노드로부터 Hello 패킷을 받으면, 그 노드는 이웃 노드의 수신 채널 정보를 바탕으로 Neighbor-Table을 업데이트 하고, 이웃 노드의 Neighbor-Table을 통해 ChannelUsageList를 업데이트 한다. Neighbor-Table은 이웃 노드들의 수신 채널 정보를 포함하고, ChannelUsageList는 2홉(two hop)내에서 각 채널 별 수신 채널로 사용하는 노드들의 수를 나타낸다.

$$Threshold = \left\lceil \frac{\text{2홉 이내의 노드 수}}{\text{총 이용가능한 채널수}} \right\rceil \quad (1)$$

많은 노드들이 동일한 수신 채널을 사용하는 것을 방지하기 위해 각 채널 별 수용할 수 있는 노드들의 수를 식 (1)에 정의된 Threshold 값으로 정의한다. 즉, Hello 패킷을 보내기 전에 노드는 ChannelUsageList를 참고하여 자신과 동일한 수신 채널을 사용하는 노드의 수가 Threshold 보다 클 경우 그 노드는 p_c의 확률로 자신의 수신 채널을 변경한다. 확률 접근은 수신 채널의 빈번한 변경을 피하기 위한 것이다.

그림 6은 노드 A의 수신 채널 변경 예를 보여준다. 노드 A가 속한 채널 2에 할당된 노드 수가 3이고, Threshold는 2이기 때문에 노드 A는 p_c의 확률로 수신 채널 변경을 시도한다. 만일 노드 A가 수신 채널을 변경한다면 채널 3을 사용하는 노드가 없기 때문에 채널 3을 수신 채널로 설정하고, 이것을 이웃 노드들에게 브로드캐스트를 하여 알린다. 만약 수신 채널로 각 채널을 사용하는 노드들의 수가 동일하다면, 임의로 하나의 채널을 선택하여 수신 채널로 설정한다.

그림 7은 Hello 패킷 교환 전과 후 그리고 수신 채널 변경 후의 Neighbor-Table과 ChannelUsageList의 상태를 보여준다. 3개의 채널이 이용가능하고, 5개의 노드가 서로의 1홉 거리에 위치해 있다고 가정한다. 최초에 각 노드는 임의로 자신의 수신 채널을 선택한다(그림 7-(a)). 수신 채널을 선택한 후 각 노드는 Hello 패킷을 브로드캐스트 한다. 그림 7-(b)는 Hello 패킷 교환 후의 Neighbor-Table과 ChannelUsageList의 상태를 나타

Ch	Node
1	B, D
2	A, C, E
3	

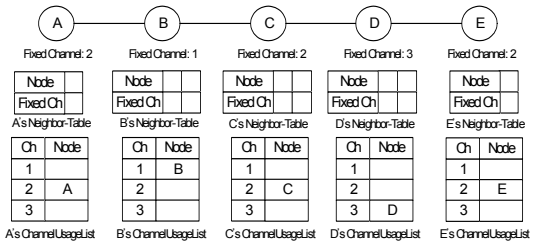
A's ChannelUsageList

fixed channel change: 2->3

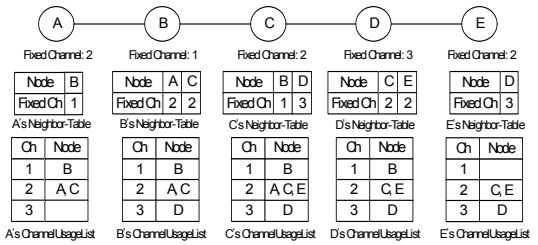
Ch	Node
1	B, D
2	C, E
3	A

A's ChannelUsageList

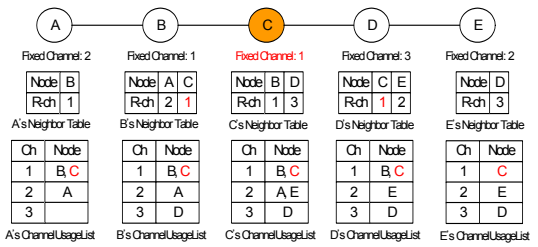
그림 6. 노드 A의 수신 채널 변경 예



(a) Hello 패킷 교환 전



(b) Hello 패킷 교환 후



(c) 노드 C의 수신 채널 변경 후

그림 7. Hello 패킷 교환 및 수신 채널 변경에 따른 Neighbor -Table과 ChannelUsageList의 상태

낸다. 노드 C는 ChannelUsageList를 통해 자신과 동일한 채널을 사용하는 노드의 수가 많다는 것을 알고, 자신의 수신 채널을 채널 2에서 채널 1로 변경 후 이웃 노드들에게 이 정보를 브로드캐스트 한다(그림 7-(c)). 만일 노드 C가 노드 B에게 데이터를 전송하기 원한다면 노드 C는 자신의 Neighbor-Table을 검색하여 노드 B의 수신 채널을 찾은 후 노드 B에게 데이터를 전송하게 된다. 일정 시간동안 업데이트 되지 않은 노드는 자신으로부터 멀어졌다고 생각하여 자신의 Neighbor-Table과 ChannelUsageList에서 제거한다.

3.3 Channel and Slot Negotiation

이 기간에 모든 노드는 자신의 고정된 인터페이스를 수신 채널에 대기한다. 데이터 전송을 원하는 노드는 각 mini-slot의 RTS 구간에서 목적 노드의 수신 채널로 전환 가능 인터페이스를 변경하여 RTS를 전송한다. 만일 한 노드가 mini-slot n에서 RTS를 보내고 CTS를 받았다면, 그 노드는 데이터 전송 구간의 데이터 슬롯 n에서 데이터를 전송한다. 데이터 전송을 위한 채널은 목적 노드의 수신 채널이 된다. 예를 들어, 노드 A의 수신 채널이 1, 노드 B의 수신 채널이 2이고 A가 B에게 전송할 데이터를 가지고 있다고 가정하자. 노드 A는 자신의 전환 가능 인터페이스를 mini-slot n의 RTS 구간에 채널 2로 전환하여 RTS를 보낸다. RTS를 받은 노드 B는 mini-slot n의 CTS 구간에 채널 2로 CTS를 노드 A에게 보낸다. 이 때, 노드 B는 자신의 전환 가능 인터페이스를 사용하여 CTS를 보내는 것이 아니라 RTS를 받은 고정된 인터페이스를 사용하여 CTS를 보내고, 노드 A는 전환 가능 인터페이스를 사용하여 CTS를 받는다. 노드 A와 B는 mini-slot n에서 채널 2를 사용하여 RTS와 CTS를 교환했기 때문에 노드 A는 데이터 전송 구간의 데이터 슬롯 n에서 채널 2를 사용하여 노드 B에게 데이터를 전송한다.

RTS와 CTS를 교환 후 바로 데이터를 전송하는 것이 아니라 채널과 슬롯을 예약한 후 데이터 전송 구간에서 데이터를 전송하는 방식을 사용하기 때문에 데이터를 수신한 모든 노드들이 mini-slot 1에서부터 경쟁을 하게 되어 높은 경쟁으로 인한 충돌 확률이 증가한다. 또한 여러 데이터 전송 경로의 교차 지점에 있는 노드는 다른 노드보다 큐가 그만큼 빨리 차게 되어 큐 오버플로우(queue overflow)로 인해 많은 패킷 손실이 발생한다. 따라서 큐에 임계값을 설정하여 누적된 패킷의 수가 임계값보다 작을 경우에는 각 노드에게 공평한 데이터 전송 기회를 주기 위해 mini-slot의 수가 N일 경우 1/N의 확률로 각 mini-slot에서 채널과 슬롯 예약을 위해 경쟁하도록 하였고, 임계값보다 클 경우에는 모든 mini-slot에서 채널과 슬롯 예약을 위해 경쟁하도록 하였다. 즉, mini-slot에서의 높은 경쟁과 큐 오버플로우로 인한 패킷 손실을 줄이기 위해 각 mini-slot에서 경쟁할 확률 p_s를 다음과 같이 정의한다.

$$p_s = \begin{cases} \frac{1}{N} & q_l < q_{th}, \\ 1 & q_l \geq q_{th} \end{cases} \quad (2)$$

- q_l: 큐에 누적된 패킷의 수.

- q_{th}: 임계값($q_{th}=\alpha \times q_{max}$).
- q_{max}: 각 노드의 큐 크기.
- α: 임계값 변수($0<\alpha \leq 0.5$).
- N: mini-slot의 수.

3.4 Data Transmission

한 노드가 channel and slot negotiation 구간에서 채널과 슬롯을 예약한 후, 그 노드는 데이터 전송 구간에서 예약된 슬롯과 채널을 사용하여 데이터를 전송한다. 데이터 전송 또는 수신하지 않는 노드는 에너지를 절약하기 위해 데이터 전송 구간에서 슬립 상태를 유지한다. 그림 8은 제안한 MAC 프로토콜의 동작 예를 보여준다. 노드 A, B, C, D는 2개의 인터페이스와 4개의 채널이 이용 가능하고, 4개의 노드들은 서로 1홉 거리에 위치한다고 가정한다. 그림 8에서 보는 것처럼 노드 A, B, C, D 각각의 고정된 인터페이스는 채널 1, 2, 3, 4에 할당되어 있다. 노드 B와 C에 데이터 전송을 원하는 노드 A와 B는 노드 B와 C의 수신 채널을 찾기 위해 Neighbor-Table을 검색한다. 노드 B와 C의 수신 채널을 찾은 후 노드 A와 B는 mini-slot 1에서 각각 자신의 전환 가능 인터페이스를 채널 2와 3으로 바꾸고 RTS를 전송한다. RTS를 받은 노드 B와 C는 노드 A와 B에게 CTS를 전송한다. 성공적인 RTS와 CTS의 교환을 통해 노드 A는 채널 2와 데이터 슬롯 1을 예약하고, 노드 B는 채널 3과 데이터 슬롯 1을 예약하게 된다. 이후에 노드 A와 B는 데이터 전송 구간에서 예약된 채널과 슬롯을 사용하여 노드 B와 C에게 데이터를 전송한다. 이 때, 노드 B는 데이터를 보내는 동시에 받는다. 다른 mini-slot에서도 동일한 방식으로 수행된다.

3.5 멀티채널 히든 노드 문제 해결

그림 9는 제안한 MAC 프로토콜이 멀티채널 히든 노드 문제를 해결한 예를 보여준다. 3개의 채널과 2개의 인터페이스가 이용 가능하고, 노드들의 네트워크는 그림 9와 같이 구성되어 있다. 노드 A는 노드 B에게 데이터를 전송하기 위해 mini-slot 1에서 RTS를 보낸다. RTS를 수신한 노드 B는 노드 A에게 CTS를 보낸다. RTS와 CTS 교환을 통해 노드 A는 데이터 슬롯 1과 채널 2를 예약한다. 노드 C도 노드 B에게 데이터를 전송하기 위해 mini-slot 2에서 RTS와 CTS 교환을 통해 데이터 슬롯 2와 채널 2를 예약한다. 노드 C는 채널 3에서 대기하고

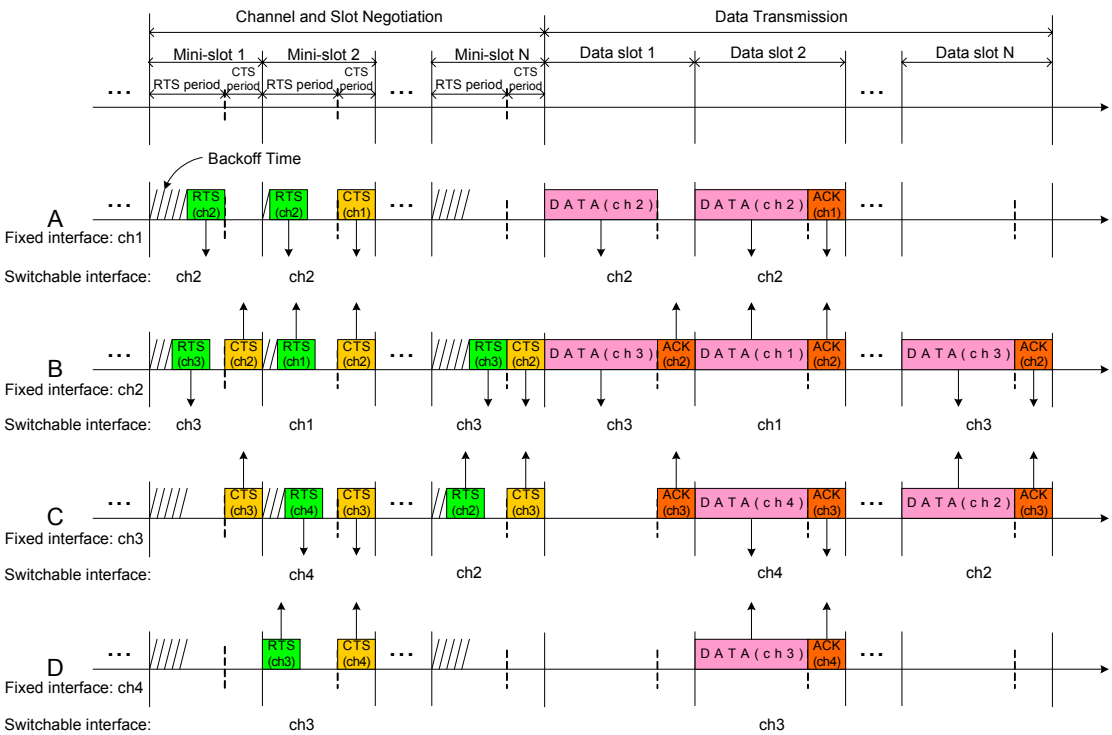


그림 8. 제안한 MAC 프로토콜의 동작 예

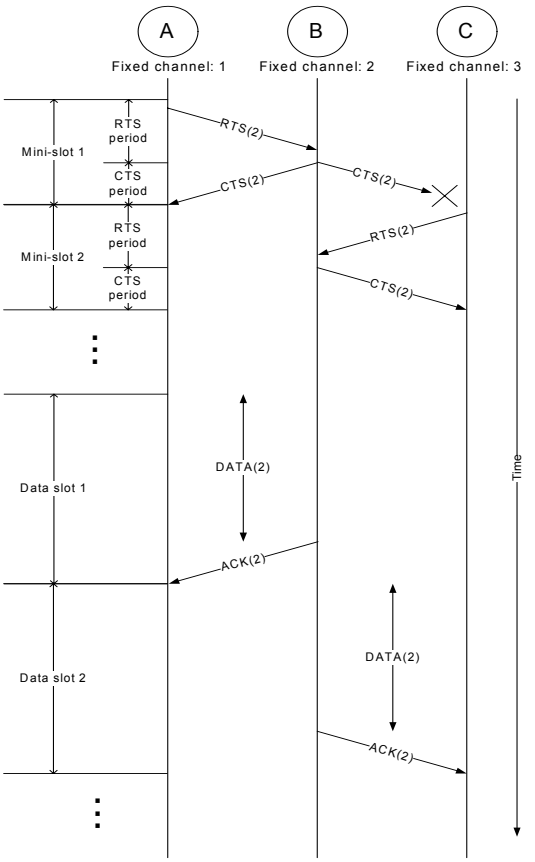


그림 9. 멀티채널 히든 노드 문제를 해결하는 예

있어 mini-slot 1에서 노드 B로부터의 CTS를 듣지 못해 노드 A가 노드 B에게 데이터를 전송한다는 것을 모르지만 노드 A는 노드 B에게 채널 2를 사용하여 데이터 슬롯 1에서 데이터를 전송하고, 노드 C는 노드 B에게 채널 2를 사용하여 데이터 슬롯 2에서 데이터를 전송하기 때문에 노드 B에서 충돌이 발생하지 않는다.

Ⅳ. 모의실험 및 성능 평가

본 논문에서 제안한 MAC 프로토콜을 평가하기 위해 HMCP와 비교를 하였고, 각 노드 당 2개의 인터페이스를 사용하도록 하여 성능 평가를 수행하였다. 이 실험에서 우리는 멀티 홉(multi-hop) 네트워크에서의 데이터 처리량(aggregate throughput), 데이터 전송 성공률, 평균 에너지 소비량 등을 측정하였다. 측정된 데이터 처리량은 네트워크상에서 모든 flow에 대한 총 처리량이다.

4.1 시뮬레이션 설정

우리는 제안한 MAC 프로토콜의 성능을 평가하기 위해 NS-2[12]를 사용하여 시뮬레이션을 수행하였다. 실험은 1000m × 1000m의 토폴로지(topology)에서 진행되었고, 100개의 노드가 사용되었다. 100개의 노드 중 36개의 노드는 토폴로지의 가장자리에 배치되었고, 인접 노드 간의 거리는 100m이다. 나머지 64개의 노드는 토폴로지의 가장자리를 제외한 1000m × 1000m 지역에 임의로 배치되었다. 실험은 각각 15번씩 진행되었고, 실험 결과는 15번 진행된 실험에 대한 평균값이다. 데이터는 토폴로지의 가장 자리에 있는 노드들에 의해 전송되고 수신된다. 이것은 많은 노드가 릴레이(relay)에 참여하게 만들어 네트워크가 포화되도록 하기 위한 것이다. 그림 10은 모의실험을 위해 100개의 노드가 네트워크상에 배치된 모습으로 실험에 사용된 여러 토폴로지 중의 하나이다. 가장 자리에 있는 노드를 제외한 64개의 노드가 임의로 배치되기 때문에 매 실험마다 토폴로지는 달라진다. 실험에 사용된 파라미터(parameter)는 표 1과 같다. 라우팅 프로토콜(routing protocol)은 AODV(Ad hoc On-demand Distance Vector)를 사용하였고, 스케줄링은 FIFO (First In First Out)를 사용하였다. 수신 채널 변경 확률 p_c의 값은 0.4로 설정하였다.

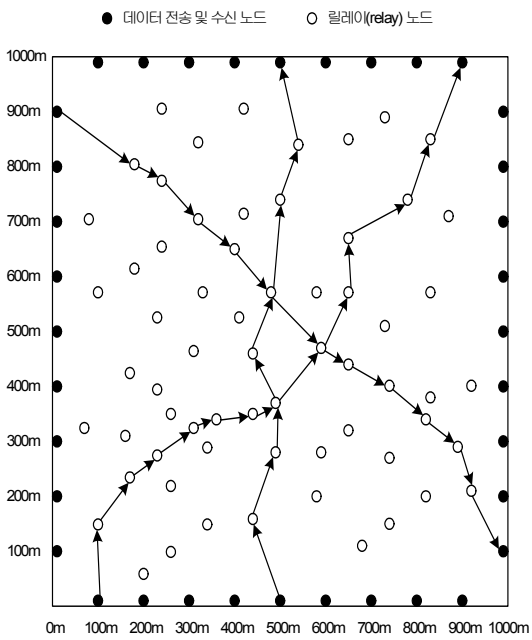


그림 10. 모의실험 토폴로지

표 1. 시뮬레이션 파라미터

시뮬레이션 파라미터	값
시뮬레이션 시간	60초
총 노드 수	100개
시뮬레이션 영역	1000m×1000m
전송 범위	250m
Data Rate	2 Mbps
Data Packet Size	1000 Bytes
전송 파워	1.0 W
수신 파워	1.0 W
Idle 파워	0.1 W

4.2 시뮬레이션 결과

그림 11과 12는 채널수의 증가에 따른 두 프로토콜의 데이터 처리량과 데이터 전송 성공률을 보여준다. 채널의 수는 2에서 10까지 바꾸면서 측정하였고, 10개의 노드가 동시에 데이터를 전송하도록 하였다. Packet arrival rate(packets/sec)는 250이다. 그림 11과 12에서 보는 것처럼 채널수의 증가에 따라 두 프로토콜의 데이터 처리량과 데이터 전송 성공률이 증가하는 것을 알 수 있다. 이것은 다수의 채널이 이용 가능할 때 각 노드는 이웃 노드와 다른 채널을 수신 채널로 사용할 가능성이 높아지게 되고, 따라서 특정 채널을 수신 채널로 사용하는 노드들의 수가 감소하여 각 채널에서 MAC 경쟁으로 인한 오버헤드(overhead)가 줄어들기 때문이다. 그러나 HMCP는 멀티채널 히든 노드 문제로 인한 패킷 충돌과 큐 오버플로우로 인한 패킷 드랍(drop)을 초래하기 때문에 그림에서 보는 것처럼 전반적인 네트워크 처리량이 제안한 MAC과 비교하여 크게 증가하지 못하는 것을 알 수 있다.

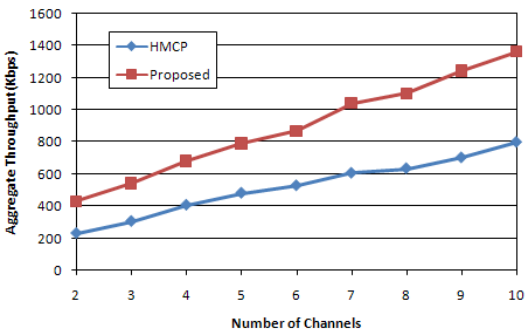


그림 11. 채널수에 따른 데이터 처리량

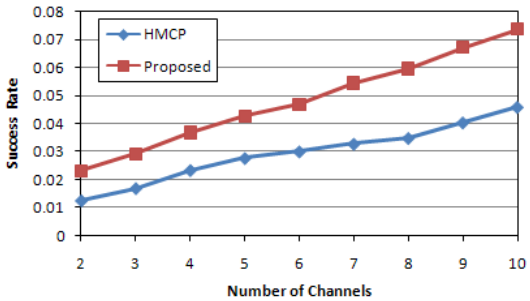


그림 12. 채널수에 따른 데이터 전송 성공률

$$Success\ Rate = \frac{\text{수신한 패킷의 총수}}{\text{전송한 패킷의 총수}} \quad (3)$$

그림 13은 동시에 전송하는 노드 수의 증가에 따른 데이터 처리량을 비교한다. 동시에 전송하는 노드 수를 1부터 10까지 바꾸면서 측정하였고, packet arrival rate(packets/sec)는 25이다. 이용 가능한 채널의 수는 3개다. 그림 13은 동시에 전송하는 노드 수가 증가함에 따라 제안한 MAC 프로토콜과 HMCP의 데이터 처리량이 증가하는 것을 보여준다. 이것은 더 많은 노드가 데이터를 보내고 받기 때문이다. 그러나 네트워크상에서 동시에 전송하는 노드 수가 증가함에 따라 MAC에 대한 경쟁도 증가한다. 따라서 동시에 전송하는 노드 수가 일정 이상 증가할 때 많은 노드들이 채널에서 경쟁해야 하므로 데이터 처리량이 서서히 감소한다. 또한 HMCP는 멀티채널 히든 노드 문제로 인한 패킷 충돌을 초래하기 때문에 제안한 MAC 프로토콜보다 더 낮은 데이터 처리량을 보여준다.

그림 14는 제안한 MAC 프로토콜과 HMCP의 노드 당 평균 에너지 소비를 보여준다. 3개의 채널이 이용 가능하고, 3개의 노드가 동시에 데이터를 전송한다. 제안한 MAC 프로토콜은 노드가 전송하거나 수신하지 않을 경우 데이터 전송 구간에서 슬

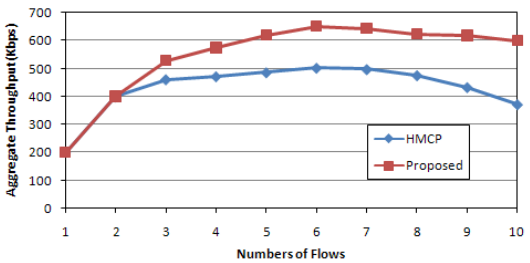


그림 13. flow에 따른 데이터 처리량

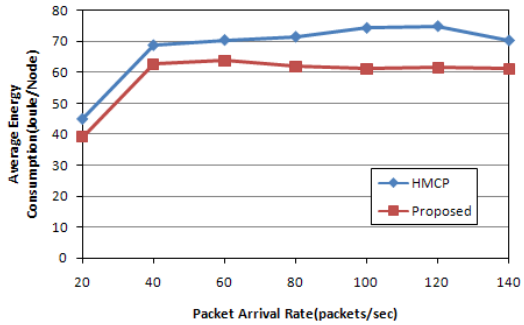


그림 14. 노드 당 평균 에너지 소비

립 상태를 유지하게 된다. 그러나 HMCP의 경우 모든 노드가 데이터 전송 또는 수신하지 않더라도 슬립 상태로 전환하지 않고 항상 대기 상태로 존재한다. 그러므로 그림 14에서 보는 것처럼 제안한 MAC 프로토콜이 HMCP보다 더 적은 에너지를 소모하는 것을 알 수 있다.

그림 15는 제안한 MAC 프로토콜의 데이터 패킷 크기에 따른 데이터 처리량을 비교한다. 동시에 전송하는 노드 수를 1부터 10까지 바꾸면서 측정하였고, packet arrival rate(packets/sec)는 25이다. 이용가능한 채널의 수는 3개다. 그림 15에서 제안한 MAC 프로토콜은 데이터 패킷의 크기가 500 bytes일 때, 1000 bytes일 때보다 데이터 처리량이 더 낮은 것을 볼 수 있다. 또한 데이터 패킷의 크기가 500 bytes일 때, 제안한 MAC 프로토콜이 HMCP보다 데이터 처리량이 더 낮은 것을 볼 수 있다. 이것은 제안한 MAC 프로토콜은 TDMA 방식을 사용하기 때문에 데이터 전송 구간의 크기가 고정되어 있는데, 데이터 패킷의 크기가 500 bytes일 경우 전송하는 패킷의 크기가 데이터 전송 구간보다 작아서 데이터 전송 구간의 낭비로 인해 시스템 성능이 저하되었기 때문이다.

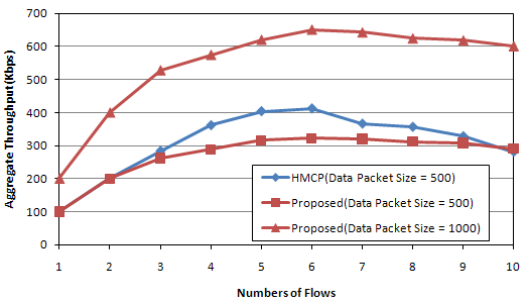


그림 15. 데이터 패킷 크기에 따른 데이터 처리량 비교

V. 결 론

무선 네트워크 환경에서 하나의 채널을 사용하는 MAC 프로토콜의 경우 노드의 수가 증가할수록 채널에 대한 경쟁이 심해져 네트워크 성능이 크게 떨어진다. 따라서 이를 해결하기 위해 다수의 채널을 사용하는 여러 MAC 프로토콜이 제안되었다. 하지만 공통 컨트롤 채널의 병목 현상 및 멀티채널 히든 노드 문제 등 여러 문제들이 나타났다.

제안한 멀티채널 MAC 프로토콜의 주요한 목적은 네트워크 처리량과 채널 효율성을 개선하고, 멀티채널 히든 노드 문제와 공통 컨트롤 채널 사용으로 인한 채널 병목 현상을 해결하는 것이다. 제안한 MAC 프로토콜은 HMCP처럼 각 노드가 수신 채널을 다르게 두는 방식을 사용하여 채널 병목 현상을 줄였고, 2개의 인터페이스를 사용하여 각 노드가 동시에 데이터를 전송하고 수신할 수 있도록 하였다. 또한 TDMA 방식을 사용하여 멀티채널 히든 노드 문제를 해결하고, 에너지를 절감할 수 있다. 모의실험 결과는 제안한 MAC 프로토콜이 HMCP보다 네트워크 성능 및 채널 효율성을 크게 향상시키고, 더 적은 에너지를 소모한다는 것을 보여준다. 그러나 TDMA 방식을 사용하여 데이터 전송 구간의 크기가 고정되어 있기 때문에 전송하는 패킷의 크기가 작을 경우 전송 구간의 낭비로 인해 시스템 성능이 저하되는 것을 알 수 있다. 무선 센서 네트워크에서는 사용 환경에 따라 패킷의 크기를 예측할 수 있기 때문에 전송 구간의 낭비로 인한 성능 저하를 줄일 수 있지만, 애드혹 네트워크에서는 각 노드의 서비스 종류에 따라 패킷의 크기가 다르기 때문에 데이터 전송 패킷의 크기를 예측하기 어렵다. 따라서 애드혹 네트워크에서는 사용 환경에 따라 전송 구간의 낭비로 인한 성능 저하가 발생할 수 있다.

참 고 문 헌

〔1〕 Jungmin So and Nitin H. Vaidya, “Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver,” in *ACM Mobihoc*, pp.222-233, 2004.

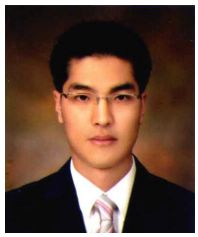
〔2〕 Jinbin Zhang, Gang Zhou, Chengdu Huang, Sang H. Son and John A. Stankovic, “TMMAC: An Energy Efficient Multi- Channel MAC Protocol for Ad Hoc Networks,” in *IEEE ICC*,

pp.3554-3561, June 2007.
〔3〕 S. L. Wu, C. Y. Lin, Y. C. Tseng, and J. P. Sheu, "A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks," in *Proceedings of International Symposium on Parallel Architectures, Algorithm and Networks*, pp.232-237, 2000.
〔4〕 Y. C. Tseng, S. L. Wu, C. Y. Lin, and J. P. Sheu, "A multi-channel MAC protocol with power control for multi-hop mobile ad hoc networks," in *Proceedings of International Conference on Distributed Computing systems Workshop*, pp.419-424, 2001.
〔5〕 Pradeep Kyasanur and Nitin H.Vaidya, "Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, Vol.10, No.1, pp.31-43, January 2006.
〔6〕 Chi-Yu Li, An-Kai Jeng and Rong-Hong Jan, "A MAC protocol for multi-channel multi-interface wireless mesh network using hybrid channel assignment scheme," *Journal of Information Science and Engineering,* Vol. 23, pp. 1041-1055, 2007.
〔7〕 A. Nasipuri, J. Zhuang, and S. R. Das, "A multichannel CSMA MAC protocol for multihop wireless networks," in *Proceedings of IEEE Wireless Communication and Networking Conference*, Vol.3, pp.1402 -1406, 1999.
〔8〕 A. Nasipuri and S. R. Das, "Multichannel CSMA with signal power-based channel selection for multihop wireless networks," in *Proceedings of IEEE Vehicular Technology Conference*, Vol.1, pp.211-218, 2000.
〔9〕 N. Jain, S. R. Das, and A. Nasipuri, "A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks," in *Proceedings of the 9th International Conference on Computer Communications and Networks,* pp.432-439, 2001.
〔10〕 A. Tzamaloukas and J. J. G. L. Aceves, "A receiver-initiated collision-avoidance protocol for multi-channel networks," in *Proceedings of IEEE INFOCOM*, Vol.1, pp.22-26, 2001.
〔11〕 P. Bahl, R. Chandra, and J. Dunagan, "SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking,* pp.216-230, 2004.
〔12〕 http://www.isi.edu/nsnam/ns

김 준 호 (Jun-Ho Kim) 준회원

2007년 8월 인하대학교 컴퓨터공학과(공학사)
2008년 3월~현재 인하대학교 정보통신대학원 석사과정
<관심분야> Wireless Sensor Network, MAC

최 재 각 (Jae-Kark Choi) 정회원



2006년 2월 인하대학교 전자공학과(공학사)
2008년 8월 인하대학교 정보통신대학원(공학석사)
2008년 9월~현재 인하대학교 정보통신대학원 박사과정
<관심분야> Cognitive Radio, Seamless handover, MAC

유 상 조 (Sang-Jo Yoo) 종신회원

1988년 2월 한양대학교 전자 통신학과(공학사)
1990년 2월 한국과학기술원 전기 및 전자공학과(공학석사)
2000년 8월 한국과학기술원 전자전산학과(공학박사)
1990년 3월~2001년 2월 KT 연구개발본부
2001년 3월~현재 인하대학교 정보통신대학원 부교수
<관심분야> 초고속 통신망, 무선 MAC 프로토콜, 인터넷 QoS, Cross-layer 프로토콜 설계, Cognitive Radio Network