

논문 2007-44TC-10-13

# 멀티 채널 멀티 홉 무선 네트워크에서 부하 인지 채널 변경 기술

(Load-Aware Channel Switching Algorithm for Multi-Channel  
Multi-Hop Wireless Network)

강민수\*, 이영석\*, 강남희\*\*, 김영한\*\*\*

(Minsu Kang, Youngsuk Lee, Namhi Kang, and Younghan Kim)

## 요약

멀티 홉 무선 네트워크에서 멀티 채널의 사용은 무선노드의 채널 간섭 및 채널 경쟁을 줄임으로써 네트워크 성능을 향상시킨다. 최근 이를 위한 다양한 기법들이 제안되고 있다. 특히 하이브리드 인터페이스 할당 방식을 사용하는 MCR(Multi channel routing protocol)은 간단한 채널 할당 방법을 통해 채널 간섭을 줄여주는 대표적인 라우팅 기술이다. 본 논문에서는 MCR의 링크 계층에서 사용하고 있는 채널 스위칭 알고리즘의 문제점을 지적하고 이를 개선할 수 있는 방안으로 트래픽 부하를 고려한 채널 스위칭 알고리즘인 LCS(Load-aware channel switching) 방안을 제안한다. LCS에서는 이웃 노드의 큐 상태 정보를 수집하여 네트워크 트래픽 상황을 파악하고 이를 바탕으로 최선의 채널을 선택한다. 또한 본 논문에서는 시뮬레이션과 테스트 베드 실험을 통해 LCS의 성능을 시험한다. 시험 결과 LCS를 적용한 MCR이 기존의 채널 스위칭 알고리즘 보다 향상된 성능을 보였다.

## Abstract

In multi-hop wireless network, multi-channel makes it possible to enhance network performance because it reduces channel interferences and contentions. Recently several schemes have been proposed in the literatures to use multi-channel. Especially, MCR(Multi channel routing protocol), which utilize hybrid interface assignment, is a prominent routing protocol. MCR uses simple way to change channel but efficiently reduce channel interferences. In this paper, we propose a load-aware channel selection algorithm called LCS that enhances the channel switching algorithm used in MCR protocol. In LCS, channel of a node is assigned based on collected information about queue length of neighbors. Moreover, this paper evaluates the performance of LCS by using simulation test and testbed demonstration. Test results show that the MCR with LCS outperforms the naive MCR.

**Keywords :** Wireless network, Multi channel, Load-aware, Channel assignment, Testbed

## I. 서론

현재 IEEE 802.11 무선 MAC 기술은 무선 LAN에서

널리 사용되고 있다. IEEE 802.11 표준은 서로 중첩되지 않아 서로 간섭을 받지 않는 다수 개의 주파수 채널을 제공한다. 예를 들어 IEEE 802.11b의 경우 중첩되지 않는 3개의 채널, IEEE 802.11a는 12개의 채널을 가지고 있다. WLAN 과 같은 인프라 기반 무선 네트워크에서는 근접한 AP(Access point)간에 이와 같은 채널을 할당 하여 사용한다. 하지만, 멀티 홉 무선 네트워크에서는 인접한 두 개의 노드는 같은 채널을 사용해야 통신이 가능하다. 인프라 기반 무선 네트워크에서 사용되는 채널 할당 방안을 멀티 홉 무선 네트워크에 적용했을 시에는 채널 간섭으로 인해 네트워크 성

\* 정회원, \*\*\* 정회원-교신저자, 송실대학교 정보통신 전자공학부

(Dept. I&C, Soongsil University)

\*\* 정회원, 카톨릭대학교 컴퓨터 정보공학부  
(The Catholic University of Korea)

※ 본 논문은 21세기 프론티어 연구개발사업의 일환으로 추진되는 정보통신부의 유비쿼터스 컴퓨팅 및 네트워크 원천기반기술개발사업의 지원에 의한 것임.

접수일자: 2007년8월27일, 수정완료일: 2007년10월18일

능을 현저히 떨어지게 만든다. 하지만 멀티 채널을 사용하면 이웃 노드들 간의 채널 간섭을 줄일 수 있기 때문에 네트워크 성능을 높일 수 있다. 이러한 이유로 멀티 채널을 위한 프로토콜이 많이 제안 되었다. 특히 IEEE 802.11 기반의 멀티채널 MAC 프로토콜<sup>[1~2]</sup>이 많이 연구 되었다.

각각의 노드가 서로 다른 채널의 인터페이스를 가지고 있고, 두 노드가 통신할 때 한가지의 채널을 결정하는 방식의 멀티 채널 MAC 프로토콜이 일반적인 멀티 채널 프로토콜이다<sup>[3]</sup>. 채널을 결정할 때 여러 가지 기준에 근거하여 채널을 결정할 수 있다. 사용하지 않는 채널을 기준으로 하거나<sup>[1]</sup>, 송신 노드의 전송 파워<sup>[2]</sup>, 또는 수신노드의 수신 파워<sup>[3]</sup> 같은 정보를 이용할 수 있다. 두 개의 인터페이스를 이용한 MAC 프로토콜에서 한 개의 인터페이스는 컨트롤 정보를 교환하는 공통 채널로 사용하고, 나머지 한 개의 인터페이스는 실제 데이터를 교환하는 채널로 사용 한다<sup>[4]</sup>. 앞서 소개한 프로토콜과 다르게 한 개의 인터페이스를 사용한 멀티채널 MAC 프로토콜도 연구되었다<sup>[5]</sup>. 위와 같은 프로토콜은 IEEE 802.11을 수정해야 한다. 하지만 SSCH(Slotted Seeded Channel Hopping)와 같은 프로토콜은 링크 계층 솔루션이기 때문에 IEEE 802.11 규격의 수정이 필요 없다<sup>[6]</sup>. 반면 지금까지 언급한 채널 할당 방법에 대한 접근이 아닌 라우팅 메트릭을 이용한 LQSR(Link Quality Source Routing)과 같은 멀티채널 라우팅 프로토콜도 제안 되었다<sup>[7]</sup>. LQSR은 여러 인터페이스를 사용하지만 채널을 바꾸지 않고 처음 할당된 채널을 계속 사용한다. 또한 WCETT(Weighted Cumulative Expected Transmission Time)라는 라우팅 메트릭을 사용하여 링크의 상태가 좋은 라우팅 경로를 찾는다. LQSR처럼 단일 채널 라우팅이 아닌 멀티 채널 라우팅 프로토콜을 제안한 다른 연구도 있다<sup>[8]</sup>. 한 개의 인터페이스를 사용하여 멀티 채널 네트워크의 구조를 제안한 연구에서는 노드마다 기본 수신 채널을 할당 하였다. 데이터를 보낼 노드는 해당 노드의 수신채널로 채널을 스위칭 한 뒤 데이터를 송신한다<sup>[9]</sup>. 또 한 무선 메쉬 네트워크를 위한 멀티 채널 라우팅 프로토콜과 채널 할당에 대한 방안도 제안 되고 있다<sup>[10~11]</sup>.

이와 같은 멀티 채널 프로토콜은 동적 채널 할당 방식과 정적 채널 할당 방식중 하나의 방식을 사용한다. MCR의 링크 계층 프로토콜은 동적 채널 할당 방식과 정적 채널 할당 방식을 적절히 섞어 이전에 제안된 멀티 채널 할당 방식보다 간단하고, 채널 간섭도 적은 메

카나즘을 사용 하였다<sup>[12~13]</sup>. 하지만 MCR은 이웃 노드의 고정 인터페이스 채널 개수 정보만을 가지고 채널 스위칭을 하기 때문에 특정 채널의 트래픽이 증가하면 효율적인 채널 할당을 할 수 없다. 본 논문은 채널 개수에 따라 채널을 할당하는 MCR의 채널 할당 방식과 달리 네트워크 상황을 고려하여 채널에 따른 트래픽 부하를 반영한 채널 할당 방식(LCS)을 제안 하고자 한다.

본 논문의 구성은 다음과 같다. II장에서는 제안 하는 채널 할당 방식의 근간이 되는 MCR의 링크 계층 프로토콜과 채널 할당 방식을 소개하고, III장에서는 MCR의 채널 할당 방식의 문제점을 제시한다. IV장에서 문제점 해결을 위한 채널 할당 방식을 제안한다. 이어서 V장에서는 시뮬레이션과 테스트 베드의 실험을 통하여 LCS를 검증하고 VI장에서 결론을 맺는다.

## II. 배경 지식

MCR은 멀티 채널, 멀티 인터페이스를 위한 링크 계층 프로토콜과 라우팅 프로토콜로 나뉘어져 있다. 링크 계층은 MAC 계층과 인터넷 계층 사이를 연결해 주고, 상황에 따라 인터페이스의 채널을 바꾸어 주는 역할을 한다. 라우팅 프로토콜은 ETT(Expected Transmission Time)를 적용하기 위해 수정된 DSR(Dynamic Source Routing)을 사용하지만 링크 계층 프로토콜과 라우팅 프로토콜이 독립적으로 설계되었기 때문에 어떠한 애드혹 라우팅 프로토콜을 사용해도 무방하다. 따라서 본 논문에서는 MCR의 라우팅 프로토콜에 대한 자세한 내용은 언급하지 않는다.

### 1. MCR 링크 계층 프로토콜

MCR 링크 계층 프로토콜의 중요한 기능은 사용 가능한 채널을 인터페이스에 적절히 분배하여 채널을 효율적으로 사용하게 하는 것이다. 이를 위해 링크 계층 프로토콜은 고정 인터페이스와 스위칭 인터페이스를 제어한다. 고정 인터페이스는 데이터를 받기만 하는 인터페이스로 비교적 오랜 시간동안 채널을 바꾸지 않고 유지한다. 반면 스위칭 인터페이스는 데이터를 보내기만 하는 인터페이스로 보내고자 하는 데이터가 있을 때 해당 노드의 고정 인터페이스 채널로 채널을 바꾸어 데이터를 보내게 된다. 그래서 스위칭 인터페이스는 채널 변화가 많고 그로 인해서 채널 스위칭 오버헤드가 많이 발생한다. 구체적인 링크 계층 동작은 그림 1에서 설명하고 있다.

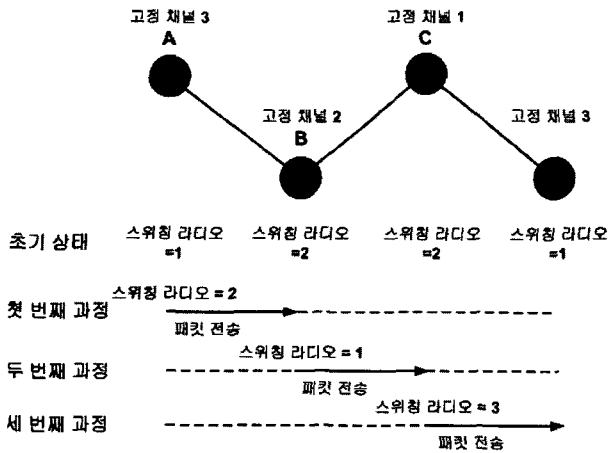


그림 1. 링크 계층 프로토콜 동작  
Fig. 1. Process of Link layer Protocol.

그림 1은 노드 A가 노드 D에게 데이터를 보내려고 하는 상황이다. 노드 D에게 데이터를 보내려면 노드 B와 노드 C를 거쳐야한다. 먼저 노드 A는 자신의 스위칭 인터페이스 채널을 B의 고정 인터페이스 채널 2로 바꾸어 데이터를 보내게 되고, 데이터를 받은 노드 B도 역시 자신의 스위칭 인터페이스 채널을 노드 C의 고정 인터페이스 채널 1로 바꾼 후 데이터를 보낸다. C도 역시 위와 같은 동작을 하여 노드 A에서 노드 D까지 데이터 전달을 가능 하게 한다.

2. MCR 채널 스위칭 알고리즘

채널 스위칭 판단을 하려면 노드의 채널 정보가 필요한데 이 정보는 헬로우 메시지를 통하여 얻는다. MCR의 고정 인터페이스 채널 스위칭 알고리즘은 헬로우 메시지를 보내기 전에 수행 된다. 이것은 최신 채널 정보를 알기 위해서 이다. 그림 2의 코드는 MCR의 채널 스위칭 알고리즘을 나타낸다.

```

MCR channel switching algorithm
-----
Calculate number of channel with channel list
if (Own's number of channel > average number of channel) then
    p = rand()
    if (p > switching probability) then
        switching channel to min(number of channel)
    end if
end if
Send Hello message
    
```

그림 2. MCR 채널 스위칭 알고리즘  
Fig. 2. Channel switching algorithm of MCR.

III. MCR 채널 스위칭 방식의 문제점 및 해결방안

MCR의 채널 스위칭 알고리즘은 주변노드가 사용하고 있는 고정 인터페이스 채널의 개수만을 정보로 수집하여 가장 적게 사용되는 채널로 스위칭 하는 방식이다. 이와 같은 채널 스위칭 알고리즘은 비교적 간단하고 채널의 간섭영향을 간접적으로 반영하지만 그림 3와 같은 상황에서는 성능을 저하 시킨다. 이런 상황일 때 A 노드가 MCR의 채널 스위칭 알고리즘을 따른다면 고정 인터페이스의 채널을 이웃 노드의 채널 개수가 가장 적은 채널 0으로 변경한다. 그림 3에서와 같이 3개의 트래픽 플로우가 있다고 가정 한다면 노드 A의 고정 인터페이스를 채널 0으로 변경하는 것은 최선의 방법이 아니다. 두 개의 트래픽 플로우가 채널 0에 집중되는 상황이기 때문에 만약 A의 고정 인터페이스를 채널 0으로 변경한다면 채널 간섭이 많아 질 것이다. 오히려 A노드 주변의 트래픽 플로우에서 사용되지 않는 채널 1이나 채널 4를 사용한다면 채널 간섭을 줄일 수 있다.

이러한 문제를 해결하기 위한 다음과 같은 두 가지 방안이 있다.

1. 채널 스위칭을 통해 네트워크 상황을 악화 시키는 것 보다 채널 스위칭을 하지 않고 초기 상태를 유지한다.
2. 채널 개수 계산 시 2 hop 이웃노드 보다 1 hop 이웃 노드에 비중을 두어 계산한다.

위 두 가지 방안도 문제점을 갖고 있다. 첫 번째 방안은 새로운 노드가 나타나거나 사라질 때 능동적으로

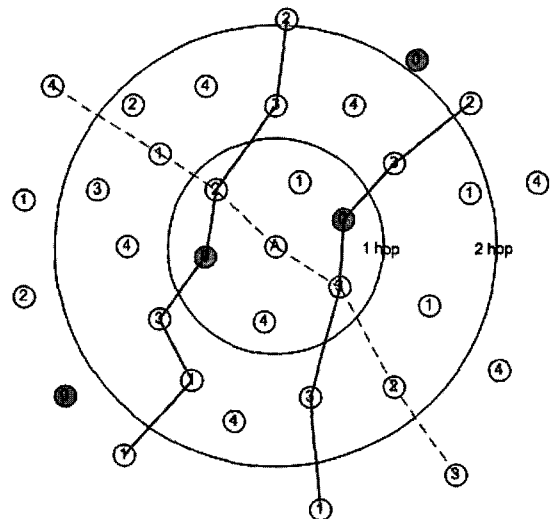


그림 3. 노드 A 주변의 트래픽  
Fig. 3. Traffic of node A.

채널을 스위칭 할 수 없어서 한 곳에 중복 채널이 집중 될 수 있는 단점이 있고, 두 번째 방안은 데이터를 받는 노드를 중심으로 계산하는 것이 아니라 보내는 노드를 중심으로 계산하여야 의미가 있기 때문에 노드 개수 계산의 복잡도가 높고 동시에 다른 이웃노드에게 데이터를 받을 때 복잡도의 문제가 더 커진다. 따라서 위의 두 가지 방안은 해결책이 될 수 없다.

LCS는 위의 문제점과 단점을 모두 해결 할 수 있다. 이웃 노드의 채널 큐 정보를 수집하여 채널 스위칭하는 LCS는 트래픽이 많은 채널 할당을 피할 수 있고, 기존의 MCR 채널 스위칭 알고리즘처럼 간단하며 능동적으로 채널 스위칭을 한다. 다음 장에서는 이 방안에 대한 자세한 알고리즘을 소개 한다.

IV. LCS-MCR

기존의 MCR의 고정 인터페이스 채널 스위칭 알고리즘은 이웃 노드의 채널 개수 정보를 이용하여 개수가 적은 채널을 선택하는 것이었다면 LCS-MCR은 각 채널의 큐 상태 정보를 바탕으로 채널 큐의 크기가 가장 작은 채널을 선택하여 고정 인터페이스 채널을 선택한다.

1. 채널 부하 정보 수집

LCS는 이웃노드의 채널 큐의 상태 정보를 필요로 하기 때문에 헬로우 메시지를 전송할 때 채널 큐의 정보도 포함되어 있어야 한다. MCR의 링크 계층은 그림 4 과 같이 고정 인터페이스와 스위칭 인터페이스를 제어 하며 채널 마다 서로 다른 큐를 두어 관리하기 때문에 채널 큐의 상태를 쉽게 파악할 수 있다. 하지만 큐의 상태 정보가 버스트한 성격을 띄고 시간에 따라 빠르게 변하기 때문에 각 노드는 EWMA(Exponential Weighted Moving Average)를 사용하여 각 채널 큐 별로 상

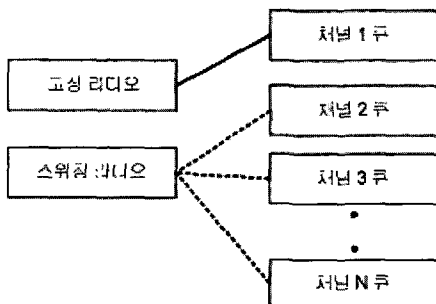


그림 4. 채널 큐와 인터페이스  
Fig. 4. Channel queue and interface.

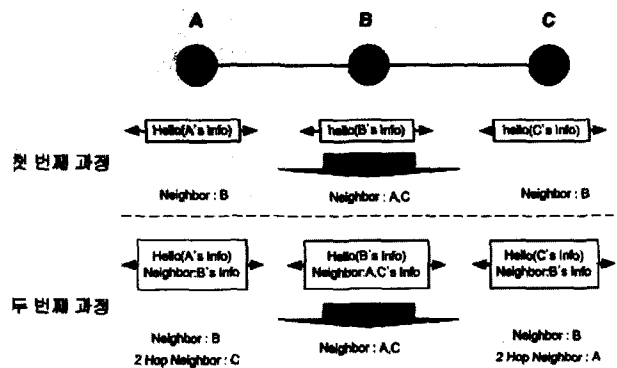


그림 5. 채널 부하 정보 수집  
Fig. 5. Collecting load of channel.

대 값을 관리한다<sup>[14]</sup>. 이렇게 저장한 각각의 채널 큐 정보는 헬로우 메시지에 포함되어 전달된다.

헬로우 메시지를 통한 채널 큐 정보의 교환은 그림 5 와 같이 두 단계로 이루어 진다. 첫 번째 단계에서는 모든 노드가 헬로우 메시지를 브로드 캐스트하여 자신을 이웃 노드에 알리고, 다음 단계에서는 헬로우 메시지를 브로드 캐스트 할 때 자신의 이웃노드의 정보도 같이 알려준다. 헬로우 메시지에 이웃노드의 정보도 포함되기 때문에 자신의 이웃노드 뿐 아니라 2 hop 떨어져 있는 이웃노드의 정보도 알 수 있다.

2. 초기 채널 할당 방안

그림 6은 무선노드의 초기 채널 할당 방법을 나타낸다. 새로운 무선 노드가 새로 네트워크에 참여하거나 시스템을 새로 시작할 때 일정 시간 동안 MCR의 채널 스위칭 알고리즘에 따라 채널을 할당 한다. 이와 같은

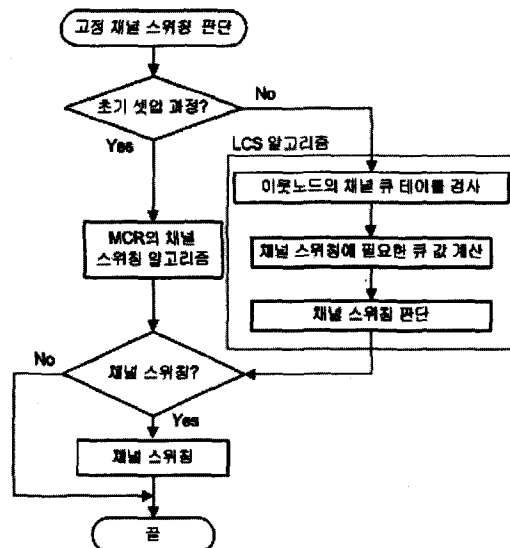


그림 6. 초기 채널 할당 방안  
Fig. 6. Initial channel assignment.

초기 설정이 지나면 LCS 알고리즘에 따라 고정 인터페이스의 채널을 스위칭 한다. MCR의 채널 스위칭 알고리즘은 네트워크의 모든 노드들의 고정 인터페이스 채널을 골고루 할당하기 때문에 채널 분산 효과가 있다. 이와 같은 상태에서 LCS 알고리즘 적용은 초기 임의의 채널 할당 방안 후 LCS 알고리즘을 적용 하는 것보다 효과적이다.

### 3. 채널 스위칭 알고리즘

부하를 고려한 채널 스위칭 알고리즘은 각 채널의 큐 상태 정보를 이용한다. 채널 스위칭 알고리즘의 기본 골격은 MCR의 채널 스위칭 알고리즘과 같다. MCR과 비교하여 다른 점은 스위칭 기준이 채널 큐라는 것과 스위칭 하기 전 현재 노드의 큐 임계값( $C_{th}$ )을 두어 스위칭을 결정하는 것이다. 이것은 전달해야 하는 데이터가 없는 노드의 불필요한 채널 스위칭을 막고, 너무 작은 채널 스위칭을 막는다. LCS-MCR의 채널 스위칭 알고리즘은 그림 7과 같다.

그림 7에서 사용되는 용어의 의미는 다음과 같다.

- $CL_i$  - 이웃노드 채널 i의 큐 크기 합
- $CL_{average}$  -  $CL_i$ 의 평균
- $CL_{current}$  - 현재 노드의 송신 채널의  $CL$ 값
- $C_{current}$  - 현재 노드의 큐 사용 비율
- $N$  - 사용 중인 채널의 개수

그림 8과 같은 상황일 때 A 노드에서 수행되는 예를

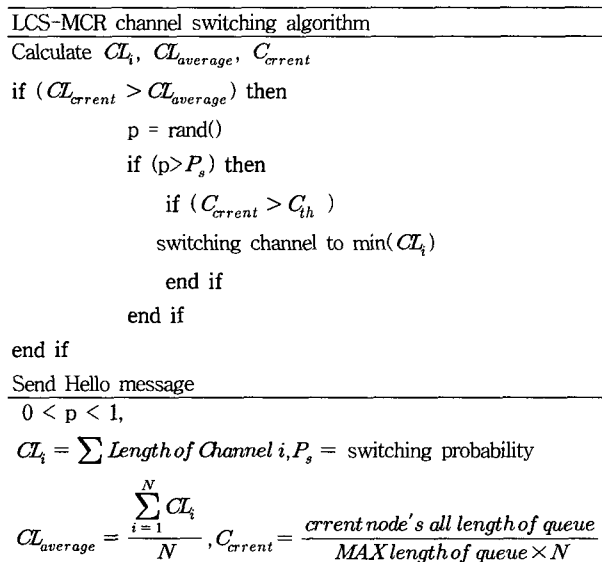


그림 7. LCS-MCR 채널 스위칭 알고리즘  
Fig. 7. Channel switching algorithm of LCS-MCR.

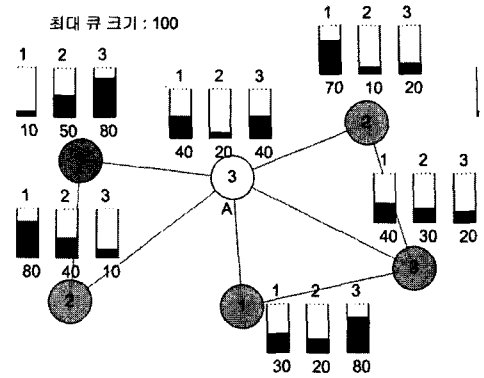


그림 8. 이웃 노드의 채널 정보  
Fig. 8. Channel information of neighbors.

들면 다음과 같다. 노드안의 숫자는 고정 인터페이스의 채널이고, 박스 위의 숫자는 채널, 아래 숫자는 큐잉 양을 나타낸다.  $p$ 가  $P_s$ 보다 크고  $C_{th}$ 가 0.25 라고 가정한다. 채널 1인 모든 이웃노드의 채널 큐 크기 값  $CL_1$ 은 230, 채널 2의 큐 크기 값  $CL_2$ 은 150, 채널 3의 큐 크기 값  $CL_3$ 은 210이고. 이 값들의 평균값  $CL_{average}$ 은 196이다. A 노드의 현재 채널3인 이웃 노드의 채널 큐 크기  $CL_3$ (210)가 평균 값  $CL_{average}$ (196)보다 크고, A 노드의 큐 사용 비율  $C_{current}$ (0.33)가  $C_{th}$ (0.25)보다 크기 때문에 알고리즘에 따라 A 노드는  $CL$ 이 가장 작은 채널 2로 스위칭 된다.

## V. 성능 평가

### 1. 시뮬레이션

시뮬레이션 실험은 MCR의 채널 스위칭 알고리즘을 사용했을 때와 LCS를 사용했을 때의 패킷 전달율과 단대 단 전송지연을 비교 하였다. 시뮬레이션 툴은 Qualnet3.9를 사용하였다<sup>[15]</sup>. 모든 실험의 시뮬레이션 타임은 90초, 네트워크 크기는 500x500m 이고 시나리오마다 노드 토폴로지는 랜덤하게 구성하였다. PHY와 MAC 은 IEEE 802.11a 모델을 사용하였고, 중첩 되지 않는 채널 5개를 사용하였다. 모든 시나리오는 트래픽 패턴과 토폴로지를 달리하여 실험하였다.

첫 번째 실험은 50개의 CBR 트래픽을 네트워크에 골고루 발생하여 일반적인 네트워크 환경 일 때 MCR과의 성능을 비교 하였다. 실험결과는 그림 9과 그림 10이다. 그림 9는 패킷 전달율을 나타내고, 그림 10은 전송 지연을 나타낸다. 두 가지 알고리즘 모두 전송지연에서 큰 편차를 보이지만, 전반적으로 비슷한 성능을 보임을 알 수 있다. LCS-MCR은 로드 상황에 따라 고

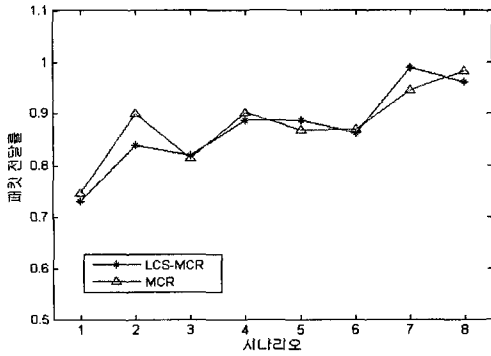


그림 9. 패킷 전송률  
Fig. 9. Delivery ratio.

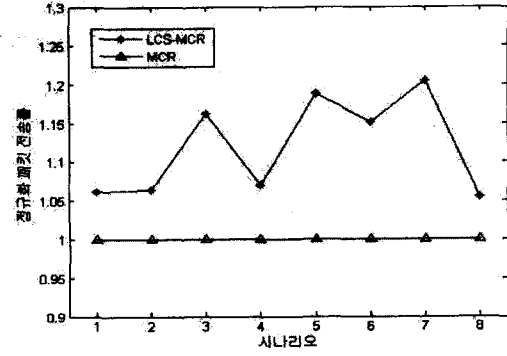


그림 11. 패킷 전송률  
Fig. 11. Delivery ratio.

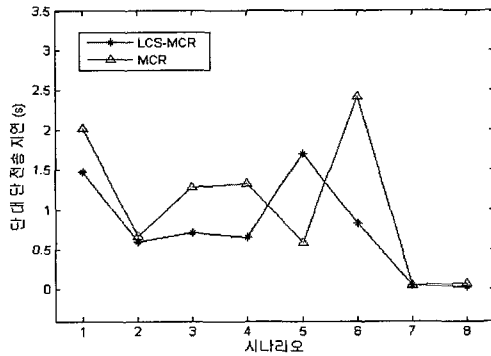


그림 10. 단대단 전송 지연  
Fig. 10. Average end to end delay.

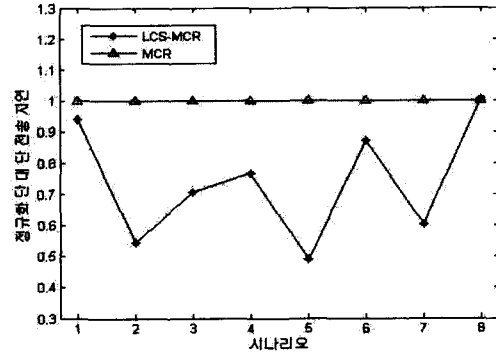


그림 12. 단대단 전송 지연  
Fig. 12. Average end to end delay.

정 인터페이스의 채널을 스위칭 한다. 첫 번째 실험은 네트워크 전체에 트래픽을 골고루 발생 시켰기 때문에 모든 노드의 채널 스위칭 빈도는 MCR의 경우와 비슷하게 된다. 따라서 MCR과 LCS-MCR은 비슷한 성능을 보인다. MCR과 LCS-MCR 채널 스위칭 알고리즘은 메트릭이 최소인 채널로 스위칭 하는 알고리즘이기 때문에 특정 노드들의 채널 스위칭 핑퐁 현상을 유발 할 수 있다. 단대단 전송 지연의 편차가 크이유가 이 때문이다. 특정노드가 채널 스위칭을 자주하게 되면 특정 노드에게 데이터를 전달해 주어야 하는 노드의 스위칭 딜레이가 커지게 되고 그 만큼 전송 지연도 커진다.

두 번째 실험은 그림 3과 같은 상황을 만들기 위해 대부분의 CBR 트래픽을 그림 13의 표시된 부분에 발생 시켰다. MCR과의 성능 비교를 위해 패킷 전달율과 전송 지연을 MCR에서 나타난 값을 1로 하여 정규화 하였다. 그림 11과 그림 12는 그 결과를 보여준다. LCS-MCR의 패킷 전달율은 대체적으로 MCR보다 10~20퍼센트 향상 되었고, 단 대 단 전송지연도 20~30퍼센트 향상되었다. LCS-MCR은 채널의 로드상황 정보를 수집하여 채널 스위칭을 하기 때문에 특정 지역에 로드가

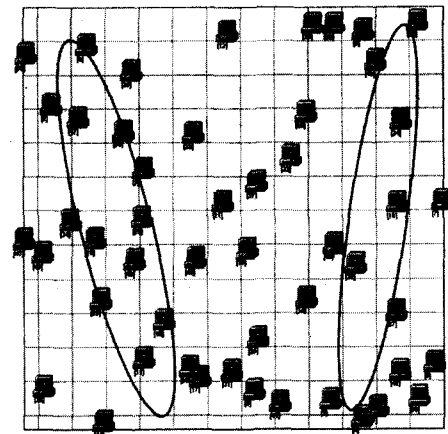


그림 13. 시뮬레이션 CBR 트래픽 환경  
Fig. 13. CBR traffic in simulation.

높아지면 채널 로드 분산 효과를 얻을 수 있다. 하지만 MCR은 특정한 지역에 로드가 높아지는 상황에서도 채널 개수 정보만으로 채널 스위칭을 하기 때문에 특정 채널에 트래픽이 집중 될 수 있다. 이는 채널 간섭과 채널 경쟁을 높이게 하여 성능 저하를 가져 오게 된다.

시뮬레이션 결과로 알 수 있듯이 LCS-MCR은 일반적인 네트워크 상황일 때 MCR과 비슷한 성능을 보이

며 그림 8과 같이 트래픽이 특정한 곳에 많이 발생하는 상황 일 때 MCR보다 좋은 성능을 보임을 알 수 있다.

2. 테스트베드

테스트 베드는 실제 환경과 비슷하게 구성 하였고 용이한 테스트베드 구성을 위해 VTC(Virtual Topology Controller)를 구현하여 이를 이용하였다. VTC는 실제로 멀티 홉 상황을 만들어 실험 할 때 용이하게 실험하

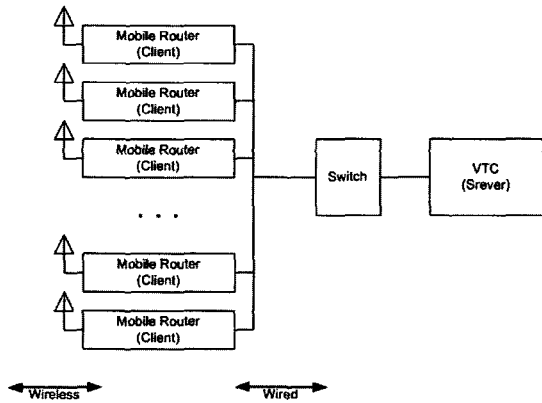


그림 14. VTC의 구성  
Fig. 14. Architecture of VTC.

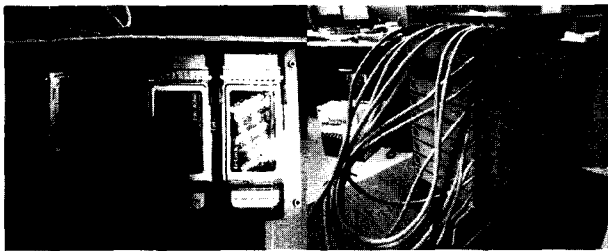


그림 15. 테스트 베드의 모바일 라우터  
Fig. 15. Mobile router of testbed.

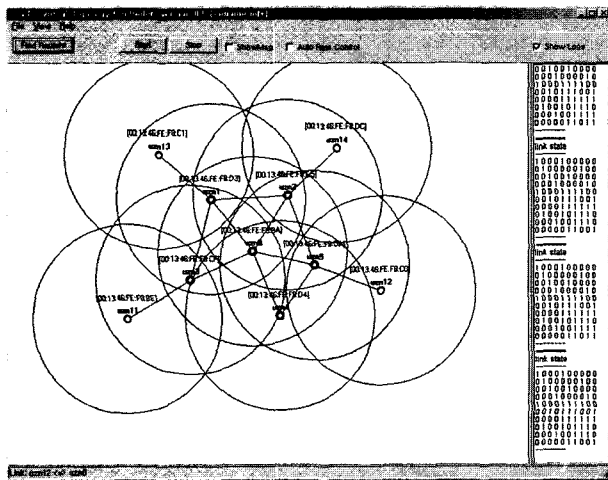


그림 16. 테스트 베드의 서버  
Fig. 16. Server of testbed.

기 위해 개발된 애드 혹 네트워크 평가 도구이다.

VTC는 MAC 필터를 사용하여 좁은 실내 환경에서 실제와 같은 멀티 홉 상황을 그림 16에서 보여주는 GUI를 이용하여 손쉽게 구성할 수 있게 해 준다. VTC는 그림 14와 같이 모바일 라우터(클라이언트), VTC(서버)로 구성되어 있다. 서버는 그림 16과 같은 GUI 컨트롤 프로그램이고, 클라이언트는 실제 애드 혹 네트워크를 구성할 모바일 라우터에 설치된 프로그램이다. 클라이언트와 서버는 유선으로 연결되어 정보를 교환한다. 클라이언트는 서버에서 발생하는 토폴로지 정보를 이용하여 MAC 필터를 구성하여 서버에서 발생한 정보와 같은 토폴로지를 만들 수 있다. 토폴로지가 구성되면 실제 라우팅 때문에 의해 처리되는 컨트롤 패킷이나 데이터 패킷은 무선 인터페이스를 통해 전달된다. 그림 15는 실제 테스트 베드의 모바일 라우터 모습을 나타낸다.

그림 16은 VTC 서버의 GUI 화면을 나타내며 실험 환경의 토폴로지를 보여준다. 각 노드는 두 개의 IEEE 802.11a 인터페이스가 장착 되어 있고, 실험을 위해 노드 각각의 인터페이스에 고정된 임의의 채널을 할당 하였다. 그림 17에서 원안의 숫자는 각 노드사이 링크의 채널을 나타내고, 화살표는 CBR트래픽의 방향을 가르킨다. CBR 트래픽은 총 2개이며, 노드 10에서 생성한 CBR 트래픽이 첫 번째 트래픽이고, 노드 7에서 생성한 CBR트래픽이 두 번째 트래픽이다. 각 CBR 트래픽의 전송 속도는 4Mbps이다.

실험 시나리오는 다음과 같다. 첫 번째 CBR 트래픽을 발생시킨 후 50~60초 후 두 번째 CBR 트래픽을 발생 시킨다. 초기 토폴로지가 그림 17의 왼쪽 그림과 같을 때 LCS-MCR의 경우 표시한 부분의 채널에서 트래픽의 로드가 높아지기 때문에 노드 6이 그림 14의 오른쪽 상황과 같이 고정 인터페이스의 채널을 60에서 161로 바꾼다. 반면에 MCR같은 경우 이웃 노드의 채널 개수 정보만으로 채널 스위칭을 결정하기 때문에

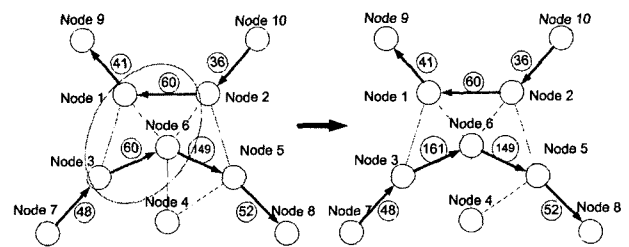


그림 17. 실험 토폴로지  
Fig. 17. Test topology.

VI. 결 론

본 논문은 트래픽 상황을 고려한 채널 스위칭 알고리즘을 제안하여 MCR의 성능을 개선 시켰다. 기존의 MCR의 채널 스위칭 알고리즘은 단순히 주변 노드들이 사용하고 있는 채널 개수 정보를 이용하기 때문에 특정한 지역에 많은 트래픽이 발생 하였을 때 효과적인 채널 할당을 할 수 없다. 하지만 LCS는 주변노드의 채널 큐 정보를 바탕으로 채널 할당을 하기 때문에 트래픽이 분산 되는 효과를 얻어 네트워크의 성능을 높일 수 있다. 일반적인 환경과 특정 지역에 많은 트래픽이 발생하는 환경을 만들어 시뮬레이션과 테스트 베드를 통하여 검증해 본 결과, 일반적인 환경에서는 MCR과 LCS-MCR이 비슷한 성능을 보였고 특정 지역에 많은 트래픽이 발생하는 환경에서는 LCS-MCR이 MCR보다 좋은 성능을 보였다.

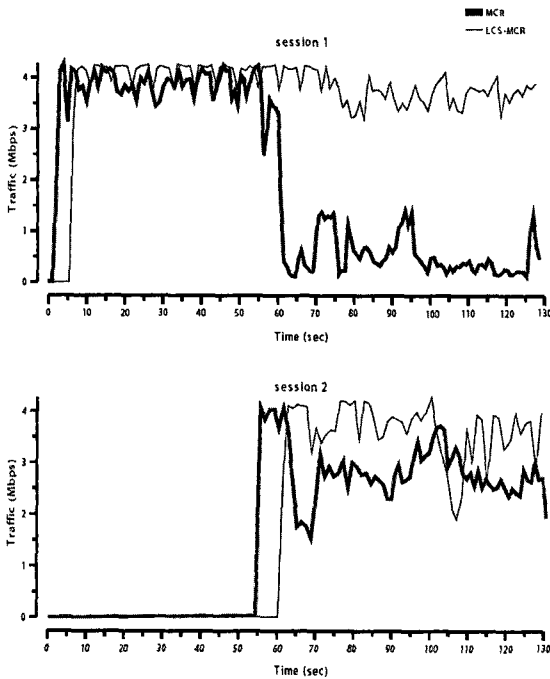


그림 18. MCR과 LCS-MCR의 처리율  
Fig. 18. Throughput of MCR and LCS-MCR.

그림 14의 왼쪽 토폴로지를 계속 유지 한다. LCS-MCR은 로드가 높아지면 고정 인터페이스의 채널을 바꾸기 때문에 그림 14의 오른쪽 토폴로지처럼 두 개의 트래픽이 채널간섭을 받지 않게 된다. 하지만 MCR은 그림 14의 왼쪽 상황을 계속 유지하기 때문에 두 개의 트래픽이 서로 채널 간섭을 일으키게 된다. 그림 18는 실험 처리율 결과를 보여준다. 첫 번째 CBR 트래픽이 4Mbps를 유지하고 있는 가운데 50~60초 사이에 두 번째 4Mbps의 CBR 트래픽을 발생 시켰다. LCS-MCR의 경우 두 번째 트래픽이 발생된 후에도 두 개의 트래픽 모두가 약3.5Mbps 처리율을 유지함을 알 수 있다. 반면에 MCR은 두 번째 트래픽이 발생 한 후에 두 개의 트래픽 모두가 원래 처리율을 유지하지 못하고 급격히 저하됨을 알 수 있다. 이는 특정 채널에 트래픽이 많아지게 되면 그 만큼의 채널 간섭과 채널 경쟁이 많아지기 때문이다. LCS-MCR 측정 결과에서 처리율이 4Mbps를 유지하지 못하고 약간 저하하는 이유는 실내에서의 실험을 위해 완전히 중첩 되지 않는 채널을 사용하였기 때문이다.

이와 같은 테스트 베드 실험을 통해 실제로 로드가 많은 환경에서 LCS를 사용한 MCR의 성능향상을 확인하였다.

참 고 문 헌

- [1] A. Nasipuri, J. Zhuang, and S.R. Das, "A Multichannel CSMA MAC Protocol for Multihop Wireless Networks," in WCNC, September 1999.
- [2] A. Nasipuri and S.R. Das, "Multichannel CSMA with Signal Powerbased Channel Selection for Multihop Wireless Networks," in VTC, September 2000.
- [3] N. Jain, S. Das, and A. Nasipuri, "A Multichannel CSMA MAC Protocol with Receiver-Based Channel Selection for Multihop Wireless Networks," in IEEE International Conference on Computer Communications and Networks (IC3N), October 2001.
- [4] Shih-Lin Wu, Chih-Yu Lin, Yu-Chee Tseng, and Jang-Ping Sheu, "A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks," in International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN), 2000.
- [5] Jungmin So and Nitin H. Vaidya, "Multi-channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals using a Single Transceiver," in Mobihoc, 2004.
- [6] Paramvir Bahl, Ranveer Chandra, and John Dunagan, "SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks," in ACM Mobicom, 2004.



[7] Richard Draves, Jitendra Padhye, and Brian Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks," in ACM Mobicom, 2004.

[8] Jungmin So and Nitin H. Vaidya, "A Routing Protocol for Utilizing Multiple Channels in Multi-Hop Wireless Networks with a Single Transceiver," Tech. Rep., University of Illinois at Urbana-Champaign, October 2004.

[9] N. Shacham and P. King., "Architectures and Performance of Multichannel Multihop Packet Radio Networks," IEEE Journal on Selected Area in Communications, vol. 5, no. 6, pp. 1013 - - 1025, July 1987.

[10] Ashish Raniwala, Kartik Gopalan, and Tzi-cker Chiueh, "Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks," Mobile Computing and Communications Review, vol. 8, no. 2, pp. 50 - -65, April 2004.

[11] Ashish Raniwala and Tzi-cker Chiueh, "Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network," in Infocom, 2005, To Appear.

[12] P. Kyasanur and N. H. Vaidya, "Routing and Link-layer protocols for Multi-Channel Multi-interface Ad hoc Wireless Networks," Mobile Computing and Communications Review, Jan 2006.

[13] P. Kyasanur and N. H. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," in Wireless Communications and Networking Conference, 2005.

[14] S. Floyd and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance," IEEE/ACM Transactions on Networking, vol. 1 no. 4, p. 397-413, August 1993.

[15] QualNet Simulator, <http://www.scalable-networks.com/>, Online Link.

저 자 소 개



강 민 수(정회원)  
 2005년 숭실대학교 정보통신전자공학부 학사 졸업  
 2006년~현재 숭실대학교 정보통신전자공학부 석사과정  
 <주관심분야 : 컴퓨터 통신, 라우팅, MANET>



강 남 희(정회원)  
 1999년 숭실대학교 정보통신공학과 학사  
 2001년 숭실대학교 정보통신공학과 석사  
 2005년 Siegen University(독일) 컴퓨터공학 박사  
 2006년 다산네트웍스 선임연구원  
 2007년~현재 카톨릭 대학교 컴퓨터정보공학부 초빙교수  
 <주관심분야 : 컴퓨터 통신, 인터넷 보안>



이 영 석(정회원)  
 2004년 숭실대학교 정보통신전자공학과 학사  
 2006년 숭실대학교 전자공학과 석사  
 2006년~현재 숭실대학교 정보통신공학과 박사과정  
 <주관심분야 : IMS, BcN, SIP, VoIP, QoS>



김 영 한(정회원)-교신저자  
 1984년 서울대학교 전자공학 학사  
 1986년 한국과학기술원 전기전자공학 석사  
 1990년 한국과학기술원 전기전자공학 박사  
 1990년~1994년 디지콤정보통신연구소 연구부장  
 2006년~현재 숭실대 정보통신공학과 교수  
 2006년~현재 통신학회 인터넷연구회위원장  
 2006년~현재 VoIP포럼 차세대분과 위원장  
 <주관심분야 : BcN, IMS, VoIP, QoS, MANET>