

IEEE 802.11 무선 네트워크에서 GPS와 SNR을 이용한 핸드오버 메커니즘

(A Handover Mechanism for IEEE 802.11 Wireless Networks using GPS and SNR)

윤인수[†] 정상화^{**} 김정수^{***}
(In-Su Yoon) (Sang-Hwa Chung) (Jeong-Soo Kim)

요약 본 논문에서는 IEEE 802.11 링크 계층의 핸드오버 성능을 증가시킬 수 있는 메커니즘을 제시한다. 본 메커니즘은 GPS에 기반한 AP 맵을 참조하여 스캐닝할 채널 수를 줄인다. 또한 모바일 노드와 주변 AP들의 SNR값들을 모니터링 함으로써 주어진 경계값보다 높은 SNR값을 유지하도록 핸드오버한다. 실험 결과, 본 메커니즘은 6.7%의 통신 단절률을 가지며 평균 16.8 dB의 SNR값을 가진다. 이는 기존의 MadWifi가 사용하는 핸드오버 방식에 비해서는 4.1% 낮은 통신 단절률이며, 26% 높은 평균 SNR값이다.
키워드 : IEEE 802.11, 핸드오버, 링크계층 핸드오버, GPS, SNR

Abstract In this paper, we propose a mechanism for increasing the handover performance of the IEEE 802.11 link layer. The mechanism reduces the number of scanning channels by referencing an AP map based on GPS. Also, by monitoring the SNRs of the mobile node and neighbor APs, it enables the handover to maintain a higher SNR than a given threshold. The experimental results establish that it has a disconnection ratio of 6.7% and an average SNR of 16.8 dB. It is 4.1% lower disconnection ratio and 26% higher SNR than the mechanism used by MadWifi.

Key words : IEEE 802.11, Handover, Link-layer Handover, GPS, SNR

1. 서론

IEEE 802.11에 기반한 다수의 AP(Access Point)들로 구성된 무선 네트워크망이 대학교 캠퍼스, 대형 오피스 빌딩, 항만, 공항 터미널등과 같은 장소에 설치되어 널리 사용되고 있다. 또한 도시 혹은 메트로폴리탄 범위

로 이를 확장하여 사용자들에게 seamless하게 무선 네트워크 서비스를 제공하고자 하는 것이 전세계적인 추세이다. 무선 네트워크망의 광범위한 보급과 더불어 이를 사용하는 PDA 혹은 스마트폰과 같은 모바일 노드들은 인터넷 접속을 기본적으로 제공할 뿐만 아니라, 저렴한 GPS칩을 장착하여 여러 가지 위치 기반 서비스(Location Based Service, LBS)들을 제공하는 방향으로 진화하고 있다.

무선 네트워크 서비스를 광범위하게 제공하기 위해서는 약 100미터의 커버리지를 가지는 IEEE 802.11 기반 AP들을 대규모로 설치하여야 하며, 이러한 환경에서는 모바일 노드들이 이동함에 따라 핸드오버가 빈번히 발생한다. 그러나 현재 IEEE 802.11 네트워크에서 핸드오버를 수행하는 데는 수백 밀리초에서 수 초 정도의 시간이 소요된다. 이는 잦은 통신 두절과 VoIP등과 같은 응용 프로그램의 성능저하 문제를 야기한다. 이러한 문제를 해결하기 위해 핸드오버를 보다 빨리 그리고 지능적으로 수행하려는 연구들이 활발히 진행되고 있다 [1-5].

본 논문에서는 기존의 방식들보다 핸드오버를 빨리

* 이 논문은 2009년 교육과학기술부로부터 지원받아 수행된 연구임 (지역거점연구단육성사업/차세대물류IT기술연구사업단)

† 정 회 원 : 삼성전자
isyoon@gmail.com

** 종신회원 : 부산대학교 컴퓨터공학과 교수
shchung@pusan.ac.kr
(Corresponding author임)

*** 학생회원 : 부산대학교 컴퓨터공학과
mankindy@pusan.ac.kr

논문접수 : 2008년 12월 18일
심사완료 : 2009년 4월 15일

Copyright©2009 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 정보통신 제36권 제3호(2009.6)

수행하며, 일정수준 이상의 SNR값을 유지하여 신뢰성 있는 무선 통신성능을 가지는 핸드오버 메커니즘을 제안한다. 2장에서는 관련 연구들과 각각의 장, 단점들을 살펴보며, 3장에서는 GPS와 SNR을 사용하여 핸드오버 시간을 줄이고, 주어진 경계값보다 높은 SNR값을 유지하는 방법에 대해 설명한다. 4장에서는 본 논문에서 제안하는 메커니즘의 구현 방법과 기존의 방식들과 비교한 실험 결과를 나타낸다. 끝으로 5장에서 결론을 나타낸다.

2. 관련 연구

핸드오버는 크게 링크 계층과 IP 계층 핸드오버로 나뉘는데, 본 논문에서는 링크 계층 핸드오버에 초점을 맞춘다. 링크 계층 핸드오버는 스캐닝(scanning), 인증(authentication) 그리고 결합(association) 과정으로 구성된다. 이들 중에서 핸드오버에 소요되는 전체 시간의 90% 이상을 차지하는 것이 스캐닝 과정이다[1]. 2.4 GHz 대역을 사용하는 IEEE 802.11g와 같은 경우, 모바일 노드는 11~14개 채널에 걸쳐 프로브 요청(probe request) 패킷을 브로드캐스팅하고, 이에 대한 응답이 AP들로부터 오기를 기다리는 작업을 반복적으로 수행한다.

M. Shin 의 저자들은 무선망을 구성하고 있는 AP들과 모바일 노드가 핸드오버하는 패턴을 가지고 Neighbor Graph를 만들었다[2]. 모바일 노드는 이 그래프를 사용하여 현재 AP와 이웃한 AP들이 사용하는 채널만을 스캐닝하여 핸드오버하는 시간을 줄였다. 그러나 그래프를 통해 스캐닝하는 채널 수는 줄였으나 모바일 노드가 언제 핸드오버를 수행할 것인가에 대한 고려는 하지 않고 있다.

H. Wu 의 저자들은 시간 소모가 많은 스캐닝을 핸드오버할 때 수행하는 것이 아니라, 핸드오버에 앞서 미리 수행하는 Proactive Scan[3]을 기법을 제안하였다. 스캐닝을 언제 수행할 것인지 그리고 실제 핸드오버는 언제 수행할 것인지 결정하기 위해 전송률과 신호 세기를 이용하였다. 그러나 이 기법은 모든 채널을 스캐닝하기 때문에 스캐닝에 소모되는 시간이 크다. 그리고 스캐닝을 주기적으로 수행하기 때문에 데이터를 전송하는 응용프로그램의 성능을 감소시킨다는 단점이 있다.

AP들은 자신의 존재를 알리기 위해 일반적으로 100 ms마다 비콘(beacon)을 보낸다. I. Ramani와 S. Savage는 무선망을 구성하는 모든 AP들의 시간을 동기화하여 일정한 시간 간격으로 비콘을 보내는 SyncScan 기법을 제안하였다[4]. 예를 들면, 채널 1번을 쓰는 AP들이 t 시간에 비콘을 보내고, 채널 2번을 쓰는 AP들은 $t+d$ ms에 비콘을 보내는 것이다. 모바일

노드는 주기적으로 수신되는 비콘들을 통해 주변 AP들이 사용하는 채널과 신호 대 잡음비(Signal-to-Noise Ratio, SNR)를 알 수 있다. 따라서 핸드오버 시의 스캐닝 과정이 불필요하게 되어 핸드오버 시간을 단축할 수 있다. 그러나 무선망을 구성하고 있는 모든 AP들의 시간을 정밀하게 동기화하는데 어려움이 있으며, 스캐닝을 안 하는 대신 비콘들을 계속 모니터링하는 오버헤드를 가지고 있다.

J. Montavont와 T. Noel은 모바일 노드에 GPS를 장착하여 핸드오버하는 메커니즘을 제안하였다[5]. 모바일 노드와 현재 AP간의 거리가 경계값(threshold)보다 멀어지면, 모바일 노드는 자신의 현재 좌표를 담아 서버에 전송한다. 그 후 서버는 모바일 노드의 좌표와 무선망을 구성하고 있는 다른 AP들의 좌표들을 비교하여 가장 가까운 AP의 MAC 주소를 모바일 노드에게 알려준다. 이 방식은 스캐닝 없이, 서버로부터 보고받은 AP로 바로 핸드오버하기 때문에 핸드오버에 소모되는 시간을 단축시킨다. 그러나 SNR을 고려하지 않고 좌표에 따른 거리만으로 핸드오버하기 때문에 거리는 가까워도 신호가 약한 AP로 핸드오버할 수 있다는 문제점이 있다.

본 논문에서는 GPS와 SNR을 통해 핸드오버하는 메커니즘을 제안한다. 이는 GPS에 기반한 AP 맵을 참조하여 스캐닝할 채널 수를 줄인다. 또한 모바일 노드와 주변 AP들의 SNR값들을 모니터링 함으로써 주어진 경계값보다 높은 SNR값을 유지하도록 핸드오버한다.

3. GPS와 SNR을 사용한 핸드오버 메커니즘

본 논문의 메커니즘은 IEEE 802.11 모니터 모드로 동작하는 모니터 노드가 테스트베드 무선망을 이동하면서 AP map을 만들고, 이를 모바일 노드가 다운로드 하여 핸드오버에 사용하는 방식이다. GPS를 장착한 모니터 노드는 테스트베드 무선망을 이동하면서 주변 AP들의 네트워크 트래픽을 패킷 스니퍼(sniffer) 툴을 사용하여 모니터링한다. 이 때, SNR값이 경계값 이상인 패킷들에 대한 좌표를 그림 1과 같이 기록한다. 이를 기반으로 우리는 표 1과 같이 각 AP들의 신호가 관측된 위도, 경도 범위와 AP가 사용하는 채널 번호로 구성된 테이블을 생성하였다. 이를 본 논문에서 AP map이라 부르겠다.

GPS를 장착한 모바일 노드가 테스트베드 무선망에 들어오면 서버로부터 AP map을 다운로드 받는다. 모바일 노드와 현재 통신하고 있는 AP와의 SNR값이 경계값보다 낮아지면, 자신의 현재 위치와 AP map을 참조하여 특정 채널만을 스캐닝한다. 예로 그림 1의 회색 지역에서 모바일 노드가 스캐닝할 필요가 있으면, 모바일 노드는 AP map을 참조하여 AP1, AP2, AP3가 주변에 있다는 것을 알고 1, 5, 9번 채널에 대해서만 스캐닝을

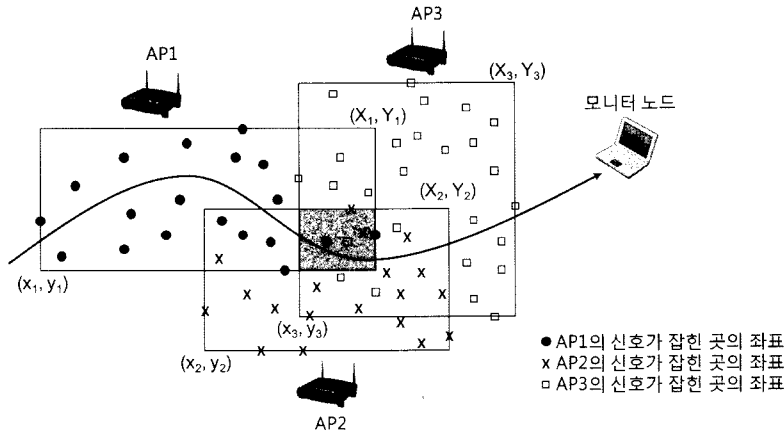


그림 1 AP의 신호가 잡힌 곳의 좌표

표 1 AP map

ID	위도 범위	경도 범위	채널 번호
AP1	$x1 < AP1의\ 신호 < X1$	$y1 < AP1의\ 신호 < Y1$	1
AP2	$x2 < AP2의\ 신호 < X2$	$y2 < AP2의\ 신호 < Y2$	5
AP3	$x3 < AP3의\ 신호 < X3$	$y3 < AP3의\ 신호 < Y3$	9
...

```

7.         end if
8.         end if
9.     else
10.        (new bssid, new SNR) = scanning()
11.        if new SNR > association threshold then
12.            associate(new bssid)
13.        end if
14.    end if
15. end while
    
```

수행한다.

J. Montavont와 T. Noel이 제안한 방법[5]은 가장 가까운 거리에 있는 AP를 찾기 위해 모바일 노드와 모든 AP들과의 거리를 계산하는데 반해, 본 방법은 거리 계산 없이 모바일 노드의 현재 위치가 AP map의 위도, 경도 범위 안에 있는지만 확인하기 때문에 간단하다. Neighbor Graph[2] 역시 모바일 노드의 핸드오버 패턴에 따라 그래프를 생성하여 스캐닝하는 채널 수를 줄이겠다는 목적은 AP map과 같으나 그래프 생성이 AP map에 비해 복잡하고 언제 핸드오버를 수행할 것인지에 대한 언급이 없다. 또한 SyncScan[4]은 주변 AP들이 보내는 비콘들을 모니터링하여 적절한 핸드오버 시점을 결정하는데, 이것 또한 모니터링으로 인한 오버헤드가 있다. 본 논문에서는 모니터링 오버헤드를 줄이고 무선 통신을 수행하는데 지장이 없는 경계값 이상으로 SNR을 유지하는 방법을 다음 알고리즘을 통해 설명한다.

모바일 노드가 결합(association)되어 있을 때는 SNR

값이 셀 탐색 경계값(cell search threshold) 이하인 경우에만 스캐닝을 수행한다. 이 때, AP map을 참조하여 특정 채널만을 스캐닝한다. 스캐닝한 결과로 나온 주변 AP들의 SNR값들 중에서 가장 큰 값과 현재 SNR값을 비교해서 δ -경계값보다 크면 새로운 AP로 핸드오버한다. δ -경계값 설정 없이 단순히 현재 SNR보다 큰 SNR을 가지는 AP로 핸드오버하면 그림 2의 t1, t2 시간 사이에 AP1과 AP2를 ping-pong하는 문제가 발생한다.

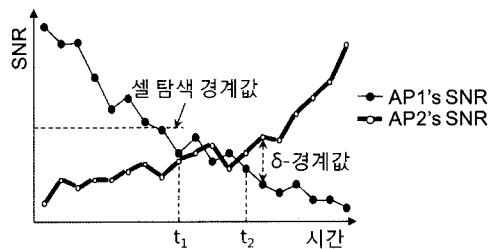


그림 2 셀 탐색 경계값과 δ -경계값

알고리즘

```

1. while ( 1 ) do
2.     if current status == associated then
3.         if current SNR < cell search threshold then
4.             (new bssid, new SNR) = scanning()
5.             if new SNR - current SNR >  $\delta$ -threshold then
6.                 handover(new bssid)
    
```

모바일 노드가 어떠한 AP와도 결합되어 있지 않을 때는 AP map을 참조하여 즉시 해당 채널을 스캐닝한다. 이 때, 결합 경계값(association threshold)보다 높은 SNR을 가진 새로운 AP로만 결합하도록 하였다. 결합 경계값보다 낮은 SNR을 가진 AP로 결합하면 연결

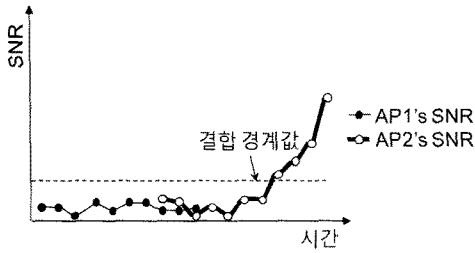


그림 3 결합 경계값

이 다시 끊어지거나 낮은 SNR로 인해 통신이 안 되기 때문이다. 이를 그림 3에 나타냈다.

4. 구현 및 실험 결과

본 연구에서는 AP map을 만들기 위해 GPS 수신기와 무선랜 카드를 장착한 노트북을 모니터 노드로 이용하였다. GPS는 SiRFstarIII 칩셋을 사용하고, 무선랜 카드는 Atheros 칩셋 기반의 IEEE 802.11g를 지원한다. Kismet[6]이라는 오픈소스 프로그램을 사용하여 IEEE 802.11 링크 계층 네트워크 트래픽을 모니터링해서 GPS좌표와 함께 기록하였다. 이들을 이용해 캠퍼스 내의 17개 건물 주위를 이동하면서 측정하였다. 이동거리는 약 900미터이며, AP 개수는 약 200개 정도가 분

포해 있다. 그림 4는 200개의 AP들 중에서 확장 서비스 셋 아이디(Extended Service Set Identifier, ESSID)가 같은 AP들이 주고 받는 패킷들을 kismet을 통해 캡처하여, 그 SNR값과 GPS좌표를 기록하고 이를 구글 어스를 통해 나타낸 그림이다. 그림 4에서 막대 그래프는 SNR값을 나타내며, 같은 AP에서 캡처한 패킷들은 같은 색을 가진다. 이러한 데이터를 사용하여 각 AP들의 신호가 관측된 위도, 경도 범위와 AP가 사용하는 채널 번호로 구성된 AP map을 생성하였다.

모바일 노드로는 리눅스로 구동되는 UMPC를 이용하였고, 모니터 노드와 동일한 GPS 수신기 및 무선랜 카드를 장착하였다. 앞서 만든 AP map은 모바일 노드에 저장되어 있다. 모바일 노드의 현재 좌표를 GPS로부터 얻기 위해 gpsd[7]라는 GPS 데몬 프로그램과 libgps[7]라는 라이브러리를 이용하였다. 스케닝과 핸드오버를 수행하기 위해 무선랜 카드에 접근하여 명령을 내릴 수 있는 iwlib[8]라는 라이브러리를 이용하였다. 이러한 라이브러리들을 이용하여 GPS와 SNR에 기반한 본 논문의 메커니즘을 C언어로 구현하였다.

본 논문에서 제안하는 메커니즘의 비교대상으로는 실제 리눅스 모바일 노드에서 널리 쓰이는 MadWifi[9]드라이버가 핸드오버하는 방식을 이용하였다. MadWifi는 모바일 노드와 AP간의 연결이 끊어졌을 때만 주변 AP

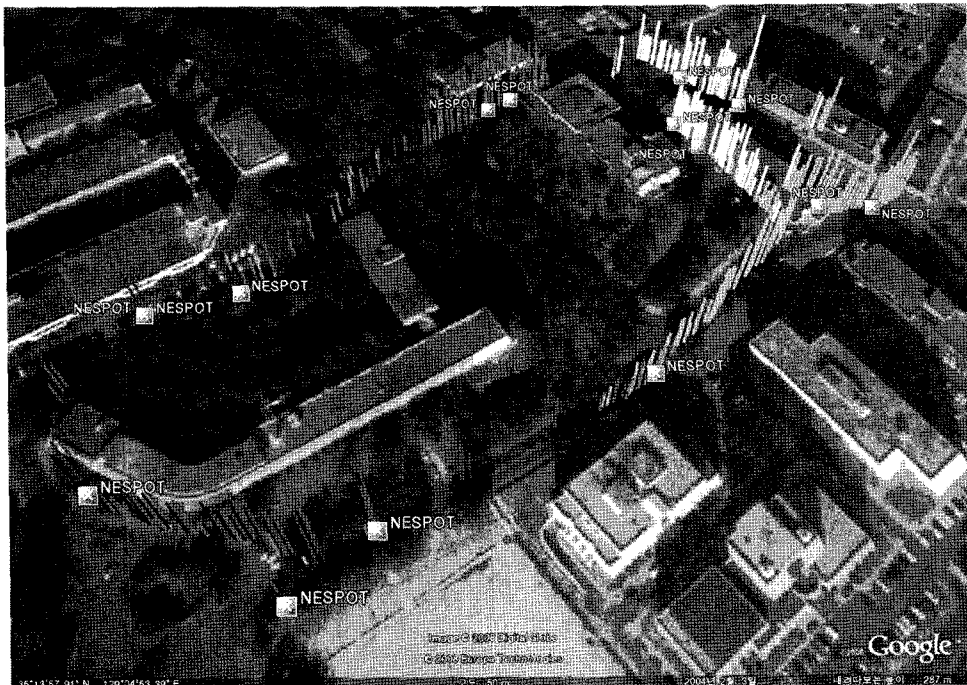


그림 4 Kismet을 통해 캡처한 패킷들의 SNR값과 GPS좌표를 구글 어스를 통해 표현

들을 스캐닝하고 핸드오버한다. 본 논문의 메커니즘처럼 모바일 노드와 현재 통신하고 있는 AP의 SNR이 낮아졌을 때, 주변 AP들을 스캐닝하여 보다 높은 SNR을 가지는 AP로 핸드오버하지 않는다. SyncScan[4]은 시간에 동기화된 비콘을 수신하는 수정된 무선랜 드라이버가 공개되어 있지 않기 때문에 본 실험의 비교 대상에서 제외하였다.

각 메커니즘별로 다섯 번의 반복 실험을 했으며, 각 실험에서 모바일 노드를 가진 사람이 약 15분간 캠퍼스를 걸으면서 통신 단절률과 평균 SNR값을 기록하였다. 본 논문에서 제안하는 메커니즘의 셀 탐색 경계값, δ-경계값 그리고 결합 경계값은 각각 10, 6, 6 dB로 설정하였다.

핸드오버에 소요되는 전체 시간의 90% 이상을 차지하는 것이 스캐닝 과정[1]이며, MadWifi를 포함한 일반적인 무선 환경에서는 아래 식에 의해 구해진 시간 범위 내에서 스캐닝을 완료하게 된다.

$$\sum_{c=1}^{c=\text{채널수}} \{(1-P(c)) \times \text{MinChannelTime} + P(c) \times \text{MaxChannelTime}\}$$

위 식에서 P(c)는 채널 c에서 하나 이상의 AP가 있을 확률을 나타낸다. 모바일 노드는 프로브 요청 (probe request) 패킷을 브로드캐스팅하고 MinChannelTime 시간 동안 AP로부터 프로브 응답 (probe response) 패킷을 하나도 못 받으면 다음 채널로 변경하여 다시 프로브 요청 패킷을 보낸다. 만약 MinChannelTime 시간

동안에 하나 이상의 AP로부터 프로브 응답 패킷을 받으면 모바일 노드는 MaxChannelTime 시간만큼 시간을 연장하여 해당 채널을 사용하는 AP가 더 있는지 기다린다. 현재 리눅스의 MadWifi 드라이버에서 MinChannelTime은 20 ms, MaxChannelTime은 200 ms로 설정되어 있다. 따라서 실험에 사용된 11개의 채널을 가지고 있는 IEEE 802.11g와 같은 경우에는 최소 220 ms에서 최대 2200 ms안에 스캐닝을 완료하게 된다.

이에 반해 본 논문에서 제안하는 메커니즘은 3장에서 설명하였듯이 모바일 노드가 자신의 현재 위치와 AP map을 참조하여 스캐닝하는 채널 수를 줄이는 방법이다. 또한 MadWifi를 포함한 일반적인 무선 환경에서는 연결이 끊어진 후, 스캐닝을 수행하는데 반해, 본 메커니즘은 현재 통신하고 있는 AP와의 SNR값이 셀 탐색 경계값보다 낮아지면 주기적으로 스캐닝을 수행하여 핸드오버할 AP의 정보를 미리 알고 있다. 따라서 핸드오버할 때 스캐닝이 필요하지 않기 때문에 실험 결과 약 20 ms내에 핸드오버를 완료하였다.

그림 5와 6은 각 메커니즘의 통신 단절률을 나타내고 그림 7은 평균 SNR값을 나타낸다. 그림 5에 비해 그림 6이 더 자주 핸드오버하는 것을 볼 수 있다. 이는 MadWifi가 연결이 끊어진 후에 핸드오버를 수행하는데 반해, 그림 6은 SNR이 셀 탐색 경계값보다 낮아지면 스캐닝을 통해 기존의 SNR보다 강한 SNR을 가진 AP

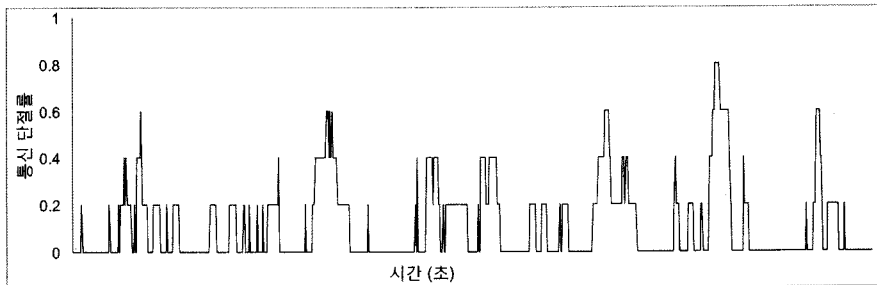


그림 5 MadWifi의 통신 단절률

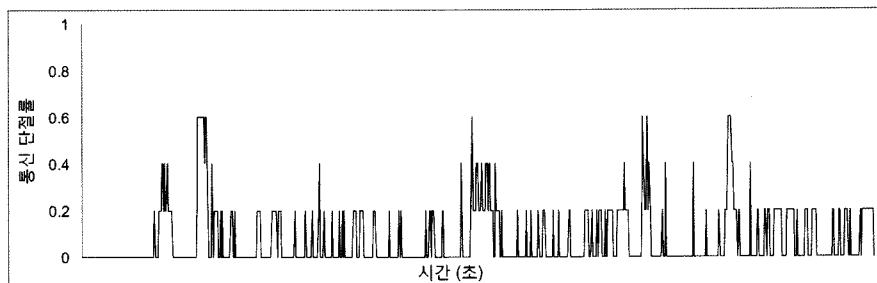


그림 6 제안하는 메커니즘의 통신 단절률

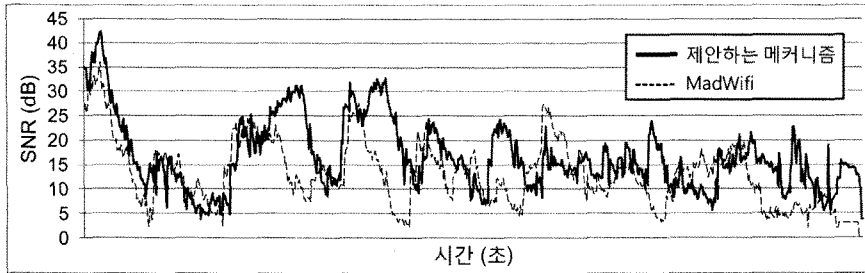


그림 7 두 메커니즘의 SNR 비교

로 핸드오버하기 때문이다. 본 논문에서 제안하는 메커니즘이 MadWifi보다 더 자주 핸드오버를 수행하지만 약 6.7%의 통신 단절률을 가지는데 반해 MadWifi는 약 10.8%의 높은 통신 단절률을 가진다.

또한 15분간의 실험을 5회 반복한 총 4500초의 실험 시간에서 1초 이상의 통신 단절을 기록한 구간을 비교, 분석해보면, MadWifi는 약 486초의 통신 단절 시간에 92%인 약 447초가 1초 이상 통신이 단절된 시간이었다. 이에 반해 제안하는 메커니즘은 약 300초의 통신 단절 시간 중에 75%인 약 225초가 1초 이상 통신이 단절된 시간이었다. 따라서 MadWifi를 사용할 때보다 본 메커니즘을 사용했을 때, 1초 이상의 긴 통신 단절을 약 50% 정도 감소시킬 수 있었다.

이는 MadWifi가 신호가 끊어진 후에 스캐닝을 수행하고, 그 당시에 SNR값이 가장 높은 AP로 핸드오버하는데 비해, 본 메커니즘은 세 가지 경계값들에 기반하여 주기적인 스캐닝을 함으로써 핸드오버 발생 전에 가장 핸드오버하기에 적합한 AP를 미리 알고 있기 때문이다. 또한 이러한 주기적인 스캐닝을 GPS와 AP map을 사용하여 빠르게 수행한다.

그림 7에서 MadWifi가 핸드오버하는 방식이 평균적으로 13.3 dB의 SNR을 가지는데 반해, 제안하는 메커니즘은 평균적으로 약 16.8 dB의 보다 높은 SNR을 가진다. 또한 셀 탐색 경계값이 10 dB로 설정된 본 논문의 메커니즘이 MadWifi가 핸드오버하는 방식에 비해 10 dB 이하로 SNR이 떨어지는 경우가 더 적다는 것을 볼 수 있다.

5. 결론

본 논문에서 GPS를 장착한 모바일 노드가 AP map을 참조하여 간단한 비교 연산을 통해 스캐닝할 채널 수를 줄이는 메커니즘을 보였다. 또한 이렇게 줄여진 스캐닝 시간을 이용해, 주어진 경계값보다 높은 SNR값을 유지할 수 있는 핸드오버 메커니즘을 보였다. 실험 결과, 본 메커니즘은 6.7%의 통신 단절률을 가지며 평균

16.8 dB의 SNR값을 가진다. 이는 기존의 MadWifi가 사용하는 핸드오버 방식에 비해서는 4.1% 낮은 통신 단절률이며, 26% 높은 평균 SNR값이다.

현재 캠퍼스 내에 설치된 무선망이 모바일 IP 혹은 글로벌 DHCP등을 지원하지 않아 실제 응용 프로그램과 연동된 핸드오버 성능을 측정하지 못하였다. 앞으로 핸드오버를 수행하여도 IP를 유지할 수 있는 무선 테스트베드를 구축하고, 본 논문에서 제안한 메커니즘을 적용하여 실제 응용 프로그램과 연동된 핸드오버 성능을 비교, 분석할 예정이다.

참고 문헌

- [1] A. Mishra, M. Shin, and W. Arbaugh, "An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process," ACM SIGCOMM Computer Communication Review, Vol.33, No.2, pp. 93-102, April, 2003.
- [2] M. Shin, A. Mishra, and W. Arbaugh, "Improving the latency of 802.11 hand-offs using neighbor graphs," Proc. of the 2nd International Conference on Mobile systems, Applications, and Services, Boston, MA, USA, pp. 70-83, June, 2004.
- [3] H. Wu, K. Tan, Y. Zhang, and Q. Zhang, "Proactive Scan: Fast Handoff with Smart Triggers for 802.11 Wireless LAN," Proc. of the 26th IEEE International Conference on Computer Communications (INFOCOM 2007), Anchorage, Alaska, USA, pp. 749-757, May, 2007.
- [4] I. Ramani and S. Savage, "SyncScan: practical fast handoff for 802.11 infrastructure networks," Proc. of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005), Miami, USA, pp. 675-684, March, 2005.
- [5] J. Montavont and T. Noel, "IEEE 802.11 Handovers Assisted by GPS Information," Proc. of the 2nd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Montreal, Canada, pp. 166-172, June, 2006.
- [6] Kismet [online]. Available: <http://www.kismetwireless.net>

less.net/

- [7] gpsd, a GPS service daemon [online]. Available: <http://gpsd.berlios.de/>
- [8] iwlib, wireless tools for Linux [online]. Available: <http://wiki.openwrt.org/iwlib>
- [9] MadWifi [online]. Available: <http://madwifi-project.org/>



윤 인 수

2001년 부산대학교 컴퓨터공학과 학사
 2001년~현재 부산대학교 컴퓨터공학과 석박사통합과정. 2009년~현재 삼성전자 DMC부문 무선사업부 책임연구원. 관심 분야는 이동통신, 임베디드 시스템, 메쉬 네트워크, 핸드오버.

정 상 화

정보과학회논문지 : 정보통신
 제 36 권 제 2 호 참조



김 정 수

2006년 부산대학교 전자전기정보컴퓨터 공학부 학사. 2008년 부산대학교 컴퓨터 공학과 석사. 2008년~현재 부산대학교 컴퓨터공학과 박사과정. 관심분야는 임베디드 시스템, 메쉬 네트워크, 이동통신