인터페이스간 동적 부하 분배를 고려한 다중 인터페이스 액세스 포인트 설계 및 구현

준회원 김 태 근*, 서 형 윤* 종신회원 김 종 덕*°

Design and Implementation of a Multi-Interface Access Point with Inter-interface Dynamic Load Balancing

Tae-keun Kim*, Hyung-Yoon Seo* *Associate Members*, Jong-Deok Kim*° *Lifelong Member*

요 약

최근 스마트폰, 노트북 등 무선 네트워크를 사용하는 단말들이 널리 보급되면서 이러한 무선 단말들이 액세스 포인트(AP)에 접속하여 주고받는 트래픽의 양도 증가하였다. 이 때 하나의 AP를 통해 전달되는 트래픽이 증가하면 성능 저하가 발생하는 문제점이 생겨났다. 이를 해결하기 위해 서비스 제공자들은 트래픽의 부하가 큰 지역에 추가적으로 AP를 설치하여 이를 방지하고자 한다. 하지만 기존 시스템에서는 단말이 접속할 AP를 선택할 때 AP의 부하를 알 수 없고, AP에서도 부하를 측정하고 부하 분배를 수행하는 매커니즘이 존재하지 않는다. 따라서 AP를 추가로 설치하여도 성능 저하를 근본적으로 방지할 수 없다. 본 논문에서는 이를 해결하기 위해 멀티 인터페이스 액세스 포인트(MIAP)를 제안한다. 여러 개의 인터페이스를 통해서 하나의 AP가 여러 개의 AP와 같이 동작하고 AP에서 각 인터페이스별 실시간 부하를 측정하여 동적으로 부하 분배를 수행한다. 실험을 통해 기존 시스템의 문제점과 그에 따른 성능 저하를 확인하였고 MIAP를 직접 비교하여 성능 비교 실험을 수행하였다. 그 결과 전체 성능을 약 72% 향상시켰고 딜레이 지터도 안정화 시키는 것을 확인하였다.

Key Words : Multi-Interface, Access Point, Dynamic Load Balancing, Traffic Monitoring

ABSTRACT

Recently, smartphone, notebook, PC and other supporting wireless LAN device have come into wide use. By increasing user that use wireless LAN device, wireless traffic also increased. If wireless traffic through one AP is increase, it causes throughput decrease. To solve this problem, wireless LAN service provider install more AP where overload occurred. But this is not enough. Because stations can't know AP's load factor, and APs do nothing for load balancing. In this paper, we propose Multi-Interface Access Point(MIAP) to solve this problem. MIAP operate same as multiple APs with multi-interface, and MIAP measure each interface's load periodically. If MIAP detect overloaded interface, MIAP transfer station from overloaded interface to under-loaded interface. We conducted an experiment for verifying existing problem, and we found this problem occurred. We plan an experiment scenario for a comparison between existing AP and MIAP, and excute these experiment. In the result, we show MIAP with load balancing can improve total throughput about 72% and stabilize delay jitter than existing AP.

Ⅰ. 서 론

최근 스마트폰, 노트북, PDA, 태블릿 PC등과 같

※ "이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(No. 2012-0001578)."
* 부산대학교 컴퓨터공학과 이동통신연구실 (all@mobile.re.kr), (° : 교신저자)
논문번호 : KICS2011-12-601, 접수일자 : 2011년 12월 4일, 최종논문접수일자 : 2012년 5월 10일

은 무선 통신을 사용하는 단말들이 널리 보급되면서 이러한 단말들이 접속하여 서비스를 제공받을 수 있는 액세스 포인트(AP)의 수요도 크게 증가하였다. 사용자들은 개방된 AP가 설치된 지역 근처에서 AP에 접속하여 단말을 통해 인터넷, VoIP, SNS, VOD 서비스 등을 제공받을 수 있다. 이러한 AP가 설치되어 무선 네트워크 사용이 가능한 지역을 핫스팟이라 한다.

AP가 존재하지 않는 환경에서도 단말들은 3G, WIBRO, LTE 등을 통해 위와 같은 서비스를 제공받을 수 있지만 3G망은 속도가 느리고, WIBRO, LTE 는 속도는 빠르지만 사용할 수 있는 지역이 제한적이고, 설치를 위해서는 많은 비용이 든다는 문제점이 있다. 따라서 사용자들은 비교적 넓은 지역에 걸쳐 서비스가 제공되고 쉽게 접속하여 안정적이고 빠른 서비스를 제공받을 수 있는 Wi-Fi를 선호한다.

무선 네트워크 특성 상 단말들과 AP는 무선 채널을 사용하기 위해 경쟁을 통해서 권한을 얻어 데이터를 전송한다. 무선 채널의 대역폭은 한정되어 있으므로 하나의 무선 채널이 수용할 수 있는 트래픽의 양은 제한적이고, 따라서 AP와 단말들은 같은 무선 채널 내에서 경쟁을 하게 된다. 이로 인해 특정 채널에 트래픽이 증가하여 하나의 채널이 수용 가능한 한계를 넘어설 경우 성능 저하가 발생한다.

이러한 특성 때문에 서비스 공급자들은 서비스 품질 향상을 위해 트래픽이 많이 발생하는 지역에 AP를 추가적으로 설치하여 문제를 해결하고자 한다. AP를 추가로 설치하여 전체 트래픽 수용 용량을 증가시키고, 이를 통해 하나의 채널에 집중된 트래픽을 분산시키려고 한다.

하지만 이것은 문제를 완벽하게 해결하지 못한다. 서로 다른 무선 채널을 사용하는 인접한 두 개의 AP가 있다고 가정할 때, 사용자들은 먼저 단말을 통해 AP에 접속을 시도한다. 이 때 사용자들이 얻을 수 있는 AP의 정보는 AP의 SSID(Service Set Identifier)와 RSSI(Received Signal Strength Identifier)뿐이다. SSID는 AP의 이름을 나타내고 RSSI는 AP의 신호 세기를 나타내는데 이것은 AP와 단말의 거리에 의해 크게 좌우된다. 즉, 사용자는 AP들이 사용하는 채널의 부하를 알 수 없으므로 최적의 AP를 찾아 접속하기가 불가능하다. 이러한 문제 때문에 사용자는 특정 AP에 집중적으로 접속할 수 있고 이는 무선 채널의 과부하를 발생시킬 위험이 있다. 또한, 기존의 AP는 자신과 주변에 있는 AP들의 부하를 측정하지 않고, 부하 분배도 수행하지 않는다. 따라서 단말들의 수가 AP간에 적절하게 분배가 되었다고 쳐도 특정 AP를 통해 전송되는 트래픽의 양이 급증할 경우 무선 채널 과부하로 인해 성능 저하가 나타날 수 있다.

본 논문에서는 이러한 문제점을 해결하기 위해 부하 분배를 고려한 멀티 인터페이스 액세스 포인트(MIAP)를 제안한다. MIAP란 하나의 AP에서 여러 개의 인터페이스를 사용하여 하나의 AP가 여러 개의 AP가 있는 것과 같이 동작한다. 이는 여러 개의 AP를 설치하는 것 보다 비용 측면에서 더 저렴하고 부하 분배를 위해 AP들의 상태를 수집할 때 별도의 추가 트래픽이 필요하지 않으므로 더 효과적이다. MIAP를 이용해 여러 개의 인터페이스들이 서로 중첩되지 않는 무선 채널을 통해 전체 트래픽 수용 용량을 증가시키고, AP 내부에서 각 인터페이스들이 서비스하는 무선 채널의 부하를 지속적으로 모니터링 하여 과부하가 발생할 경우 그림 1과 같이 부하 분배를 위해 접속한 단말을 다른 인터페이스로 이동시킨다.

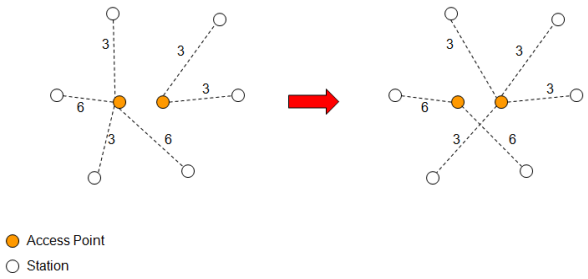

Fig 1. Dynamic Load Balancing with Station Transfer

본 논문의 구성은 다음과 같다. 2장에서는 AP의 부하를 줄이기 위한 기존 연구들을 조사하고 본 논문과의 차이점을 알아본다. 3장에서는 무선 채널에 과부하가 발생할 경우 나타나는 성능 저하 현상을 확인하고 그 원인과 해결책을 찾아본다. 4장에서는 본 논문에서 제시하는 MIAP의 구조를 설명하고 MIAP의 부하 분배 과정을 상세히 기술한다. 5장에서는 기존 AP와 MIAP의 성능 비교를 위한 실험 시나리오를 구성하고 실험을 수행하여 결과를 통해 MIAP의 성능을 검증한다. 6장에서는 논문의 결론을 맺고 향후 연구 방향을 소개한다.

Ⅱ. 관련 연구 분석

AP의 부하를 분배하기 위한 기존 연구로 중앙

서버를 통해 AP들 간에 부하 분배를 수행하는 연구가 있다[3]. 이 논문에서는 중앙 서버에서 주기적으로 AP들의 부하를 측정하고 이를 통해서 각 단말들이 어떠한 AP에 접속할지 결정해주고 이를 통해 전체 네트워크의 사용률을 높여 성능 향상을 보였다.

다른 연구로는 따로 중앙 서버를 두지 않고 같은 네트워크 내에 존재하는 AP 각각에서 부하 분배를 수행하는 연구가 있다[4]. 이 논문에서는 각 AP들이 같은 네트워크 내에서 유선으로 연결된 백본 네트워크를 통해서 부하 정보를 주고받아 부하 분배를 수행한다. 이를 통해 전체 네트워크의 성능을 높이고 딜레이를 줄이는 것을 목표로 하였다.

그 밖에도 AP와 단말의 bit-rate를 최적화하여 전체 네트워크의 성능을 향상시키기 위한 연구가 진행됐다[5]. 이 논문에서는 AP와 단말의 거리에 따라 bit-rate가 설정되는 데 여러 개의 AP와 단말들이 존재하는 환경에서 각 단말들이 접속할 AP를 지정해 줌으로써 전체 네트워크의 성능을 향상시키고 단말들 사이에 공평성을 고려하려고 하였다.

AP의 부하 분배를 위한 기존 연구들은 구현을 위해 별도의 중앙 서버를 두거나[3] 또는 단말들에게 추가적인 모듈 설치가 필요하다는 단점이 있다[5]. 또한, 실시간으로 무선 트래픽을 측정하지 않고 백본을 통해 전달되는 트래픽 량을 통해서 무선 트래픽 량을 추정하거나[4] 실제 사용 트래픽 량이 아닌 bit-rate를 측정하였다[5]. 이는 백그라운드 트래픽을 측정할 수 없다는 문제점과 실시간 트래픽 사용량을 반영하지 못한다는 단점을 가진다. 그 밖에도 부하 분배를 수행하기 위해 단말을 이동시킬 경우 이동시킬 AP들의 위치정보 또한 고려하여야 한다는 문제점이 있다.

본 논문에서 제안하는 MIAP는 실시간으로 무선 채널 전체의 트래픽 량을 측정하므로 실시간 트래픽 량에 따라 좀 더 동적이고 효율적인 부하 분배를 수행할 수 있다는 장점을 가진다. 그리고 하나의 AP 내에 여러 개의 인터페이스를 사용하여 여러 개의 AP를 사용할 때 보다 비용에서 이점을 가지고 같은 AP내에 존재하는 인터페이스 간에 단말 이동을 수행하므로 위치 정보를 고려하지 않아도 된다. 또한 부하 분배를 위해 단말을 수정하거나 별도의 서버를 구축하지 않아도 되므로 구현에 필요한 비용을 절약할 수 있다.

Ⅲ. 문제 분석 및 해결 방안

IEEE 802.11 표준은 무선 네트워크의 구조와 프로토콜을 정의한다. 그리고 802.11a/b/g/n과 같은 추가적인 표준은 확장된 세부 사항을 기술한다. 우리가 흔히 사용하는 802.11g에는 총 1~13까지 13개의 무선 채널이 정의되어 있고 각 채널은 20MHz의 대역폭을 가진다. 최근 빠른 속도로 서비스가 확장되고 있는 802.11n은 20MHz 또는 40MHz의 채널 대역폭을 가진다. 이처럼 무선 채널 자원은 한정되어 있으므로 하나의 무선 채널이 수용 가능한 트래픽의 양도 제한적이다.

따라서 서비스 제공자는 과부하를 방지하기 위해 여러 개의 AP를 설치하여 사용 가능한 채널을 늘려 트래픽을 분산시키려 한다. 하지만 여러 개의 AP를 통해 여러 개의 무선 채널을 제공하여도 특정 채널에 트래픽이 집중될 수 있지만 기존의 표준과 AP는 이것을 고려하지 않으므로 문제가 발생한다.

우리는 특정 무선 채널에 트래픽이 집중되었을 때와 적절하게 트래픽이 분배되었을 때 성능 차이를 알아보기 위해서 실험을 수행하였다. 실험을 위해서 서로 중첩되지 않는 다른 무선 채널을 사용하는 두 개의 AP를 설치하였고 그림 2와 같이 임의로 단말을 두 개의 AP에 접속시켜 각 단말들이 트래픽을 발생시키도록 하였다.

Table 1. The existing Load-Balancing research comparison

관련 연구	구조	측정 요소	부하 분배 방법	검증 방법	단말 수정
Hot-Spot Congestion[3]	Centralized	Wireless Traffic	Dynamic	Simulation	O
Load Balancing in Overlapping Cells[4]	Distributed	Wired Traffic	Semi-Dynamic	Experiment	X
Association Control[5]	Centralized	bit-rate	Static	Simulation	O
Multi-Interface Access Point	Hybrid	Wireless Traffic	Dynamic	Experiment	X

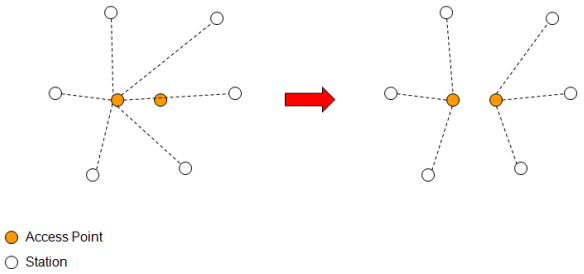

Fig. 2. Network Configuration

위와 같은 네트워크를 구성하여 실험을 수행하여 전체 성능과 딜레이 지터를 측정하였다. 그림 3은 두 개의 시나리오에 대한 전체 성능을 나타낸 그래프이고 그림 4는 각 시나리오에 대한 패킷별 평균 딜레이 지터를 나타낸 것이다. 그림 3을 보면 단말이 특정 AP에 집중된 Unbalanced의 경우 두 개의 무선 채널 중 트래픽이 하나의 채널에 몰려 일정 수준 이상의 성능을 보여주지 못하는 반면 적절히 단말이 분배가 된 Balanced의 경우 전송한 데이터를 거의 다 수신하는 것을 확인할 수 있다.

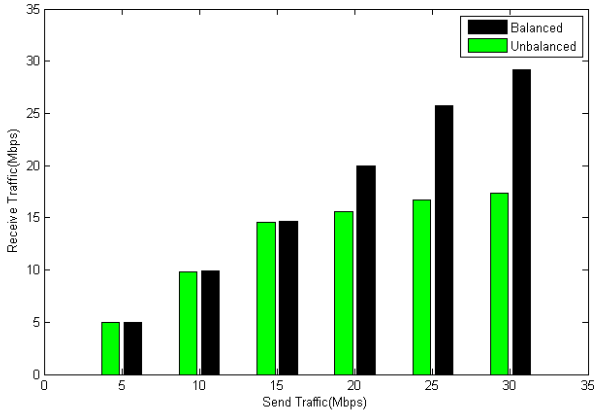

Fig. 3. The Overall Performance for Two Scenarios

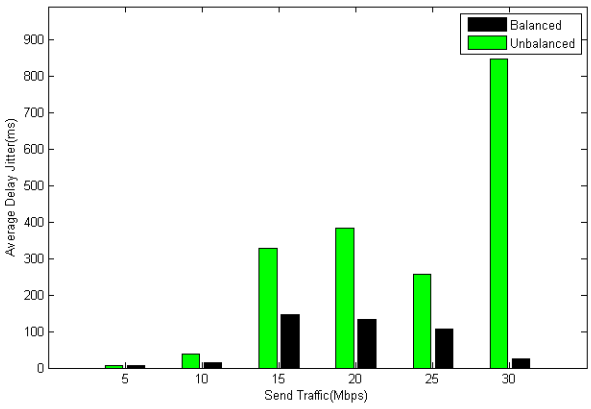

Fig. 4. The Inter-Arrival Jitter for Two Scenarios

그림 4는 두 시나리오에 대한 패킷 별 딜레이 지터의 평균을 나타낸 것이다. 그림 3에서 약 15Mbps 이상의 트래픽이 발생할 경우 트래픽을 제대로 처리하지 못하는 현상이 나타났는데 그림 4를 살펴보면 약 15Mbps 이상의 트래픽이 발생했을 때 딜레이 지터가 증가하는 현상이 나타나는 것을 볼 수 있다.

본 논문에서는 이를 해결하기 위해 멀티 인터페이스 액세스 포인트(MIAP)를 제안하고 MIAP에서 부하 분배를 수행하도록 한다. 하나의 AP에서 여러 개의 인터페이스를 통해 여러 개의 무선 채널을 서비스하고 각 인터페이스가 제공하는 채널의 트래픽을 실시간으로 측정한다. 측정값을 토대로 무선 채널간 트래픽 분배를 위해 인터페이스간 단말 이동을 수행하고 이를 통해 전체 성능 향상을 보이고자 한다.

Ⅳ. MIAP 설계 및 구현

4.1. MIAP 설계

우리는 무선 채널의 부하를 측정하여 인터페이스간 부하 분배를 수행하도록 MIAP를 설계하였다. MIAP는 여러 개의 인터페이스들이 사용하는 무선 채널의 트래픽을 실시간으로 측정하고 측정값을 바탕으로 과부하 여부를 판단한다. 그리고 과부하 현상이 나타날 경우 이를 해결하기 위해 부하 분배를 수행한다.

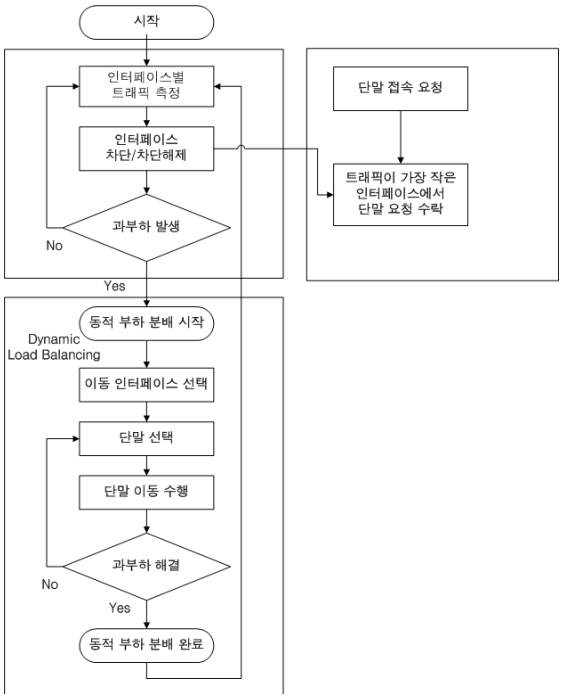

Fig. 5. MIAP Flow Diagram

그림 5는 MIAP 내부에서 트래픽을 모니터링하고 부하 분배를 수행하는 과정을 그림으로 나타낸 것이다. MIAP 내부 동작은 인터페이스를 통해 무선 채널의 부하를 측정하고 과부하 여부를 판단하는 모니터링 과정, 과부하 상태를 해결하기 위해 인터페이스간 단말 이동을 통해 부하를 분배하는 부하 분배 과정으로 나눌 수 있다.

4.2. MIAP 구성 요소

4.2.1. 트래픽 모니터링

MIAP의 인터페이스들은 자신이 사용하는 무선 채널의 트래픽을 실시간으로 모니터링 한다. 무선 네트워크 카드에서 패킷을 수신하면 인터럽트가 발생하고 이 때 지정된 인터럽트 핸들러가 호출된다. 호출되는 핸들러에서 수신한 패킷의 크기를 측정하고 디바이스 드라이버에서 이를 저장한다. 실시간 트래픽 량을 측정하기 위해 디바이스 드라이버에서 타이머를 통해 기존에 저장하고 있는 통계 자료를 바탕으로 매 초당 발생한 트래픽 량을 알아내었다. 이렇게 기존 디바이스 드라이버의 통계 자료를 통해서 트래픽 모니터링을 구현함으로써 별도의 부하 없이 트래픽 모니터링을 할 수 있도록 하였다.

인터페이스들은 무선 채널 전체 트래픽 뿐만 아니라 자신에게 접속한 단말들이 사용하는 트래픽 양도 별도로 측정한다. 동적 부하 분배를 위해 단말을 이동시킬 때 이동시킬 단말을 선택하여야 하는데 이 때 단말의 실시간 트래픽 량을 기준으로 이동시킬 단말을 선택하므로 해당 정보를 별도로 저장한다.

4.2.2 부하 분배 결정

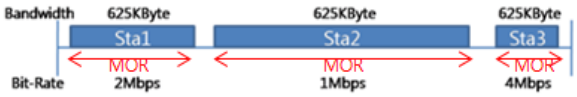

Fig. 6. MOR Concept

MIAP는 무선 채널의 과부하 여부를 판단하여 동적 부하 분배를 수행한다. 이 때 무선 채널의 부하를 나타내는 기준이 필요한데 미디어 점유율(MOR : Medium Occupation Ratio)을 통해 무선 채널의 부하를 나타낸다.

MOR은 무선 채널이 트래픽에 의해 점유되는 비율을 나타내고 이것이 높을 수록 무선 채널의 부하가 높다는 것을 의미한다. MOR의 계산은 각 Packet의 크기와 해당 Packet의 bit-rate를 통해서 계산한다. Packet의 크기를 bit-rate로 나누면 해당 Packet이 전송되는데 점유한 미디어의 시간을 구할 수 있다. 각각의 단말들과 연결된 링크의 bit-rate가 다 다를 수 있으므로 단순히 데이터의 크기가 아닌 미디어 점유율을 바탕으로 하여 채널의 혼잡도를 계산한다.

이와 같은 방법으로 단위 시간 내 수신한 모든 Packet에 대해서 MOR을 구하고 MOR이 Threshold 이상일 경우 과부하 상태라고 판단하도록 한다. 하지만 과부하 상태에 있다고 판단하여도 즉시 부하 분배를 수행하지는 않는다. 트래픽이 Burst하게 발생하였거나 특정 단말에 의해서 일시적으로 많은 트래픽이 발생하여 무선 채널이 과부하상태가 되었다면 이는 자연스럽게 해결될 수 있으므로 지속적인 모니터링을 통해 매 초마다 과부하 여부를 판단하여 특정 무선 채널에 과부하 현상이 지속적으로 나타날 때 부하 분배를 수행하도록 하였다.

4.2.3 부하 분배 수행

MIAP에서 수행하는 부하 분배는 단말 접속 제어, 동적 단말 이동 두 가지로 나눌 수 있다. 단말 접속 제어는 MIAP에 접속하는 단말이 어떤 인터페이스로 접속할 지를 AP에서 지정해주는 것이다. MIAP의 모든 인터페이스들은 모두 같은 SSID를 사용하므로 단말은 MIAP를 하나의 AP로 인식한다. 단말이 접속을 시도하면 단말은 MIAP의 모든 인터페이스들에게 접속 요청을 하게 되는데 이 때 MIAP는 무선 채널 모니터링 결과를 참조하여 가장 트래픽 량이 적은 무선 채널을 사용하는 인터페이스가 요청을 받아들이도록 한다. 이를 통해서 단말이 가장 트래픽 량이 적은 무선 채널을 사용할 수 있도록 한다.

동적 단말 이동은 과부하가 발생한 무선 채널을 사용하는 인터페이스의 단말을 다른 인터페이스로 이동시키는 것이다. 특정 무선 채널에 부하 분배가 필요하다고 판단될 때 MIAP는 부하 분배를 위해 해당 채널을 사용하는 인터페이스에 접속한 단말을 이동시킴으로써 과부하가 발생한 무선 채널의 트래픽을 다른 무선 채널로 분배한다. 먼저 MIAP는 어떤 단말을 이동시킬 것인지를 선택해야 한다. 단말 선택을 위해 우리는 아래 표2와 같이 각 요소들을 정리하였다.

Table 2. Element for Dynamic Load-Balancing (n : Number of Interface, k : Number of Station)

의미	기호
MOR Threshold	M_{th}
Each Channel MOR	M_i
Overloaded Channel MOR	M_o
Target Channel MOR	M_t
Min Balancing MOR	M_{min}
Max Balancing MOR	M_{max}
Station's MOR	M_s
Sum of Candidate Sta's MOR	M_{sum}

M_{th}는 하나의 무선 채널 점유율 Threshold이다. 이 값을 통해 특정 무선 채널의 점유율이 Threshold를 넘을 경우 과부하 상태라고 판단한다. M_i은 MIAP의 각 인터페이스들이 사용하는 무선 채널들의 점유율이다. M_o는 현재 과부하가 발생한 채널의 점유율, M_t은 이동시킬 인터페이스가 사용하는 채널의 점유율을 나타낸다. M_{min}는 과부하 상태를 벗어나기 위해 분배해야 하는 최소 채널 점유을 뜻하고 M_{max}는 이동시킬 인터페이스가 사용하는 채널의 남은 용량을 의미한다. M_{min}과 M_{max}는 아래 수식 (1)을 통해 구할 수 있다.

$$Mmin = Mo - Mth$$
$$Mmax = Mth - Mt \quad (1)$$

만약 M_{max}를 넘는 단말을 이동시킬 경우 이동시킨 인터페이스가 사용하는 무선 채널에 과부하가 발생하여 해당 인터페이스에서 다시 단말을 이동시켜야 하는 문제가 발생할 수 있다. 따라서 이동시킬 단말의 점유율은 M_{max}를 넘지 않아야 한다. 과부하가 발생한 인터페이스에 접속한 단말들의 실시간 채널 점유율 M_s를 통해 $M_s < M_{max}$ 를 만족하는 단말들을 추려낸다. 그 후 추려낸 단말 리스트에서 단말을 선택할 수 있는 모든 경우의 수를 구해 각각의 경우에 대한 선택된 단말들의 M_s의 합을 계산하고 이를 M_{sum}라 한다. M_{sum}는 선택된 단말들이 채널을 점유하는 정도를 나타낸다. M_{sum}중 M_{min}보다 큰 값이 존재한다면 그 중 가장 작은 M_{sum}를 선택한다. 그 이유는 단말을 이동시킬 때 단말과 AP 사이에 연결이 끊어지는데 그로 인해 발생하는 데이터 손실을 최소화하기 위해서이다. 만약 M_{min}보다 큰 M_{sum}가 존재하지 않는다면 그 중 가장 큰 경우를 선택한다. 이 상황은 단말을 이동시켜도 M_o를 M_{th}아래로 만들 수 없는 경우를 뜻하므로 가능한 한 많은 트래픽을 이동시키도록 한다. 즉, M_{sum}중 가장 M_{min}와 가까운 것을 선택하는 것이 가장 적절하다. 모든 M_{sum}를 구하는 것은 후보 단말의 수가 n이라고 했을 때 $O(2^n)$의 시간복잡도를 가지므로 M_{sum}를 구성하는 단말의 수를 최대 10개로 제한한다.

그림 7은 단말이 이동하는 과정을 나타낸 것이다. 1번에서 MIAP에서 이동시킬 단말의 정보를 삭제하고 강제로 연결을 끊는다. 2번에서 해당 단말은 접속이 종료된 것을 확인하고 기존에 접속한 AP와 같은 SSID를 가진 주변의 모든 AP를 검색한다. MIAP의 모든 인터페이스는 같은 SSID를 가지므로 re-association 절차에 따라 단말은 MIAP의 모든 인터페이스를 검색하게 되고 3번에서 모든 인터페이스에게 접속 요청을 보낸다. MIAP에서는 가장 트래픽 량이 적은 무선 채널을 사용하는 인터페이스만 접속을 수락하므로 4번과 같이 단말은 해당 인터페이스로 이동하게 된다. 이와 같이 단말을 이동시킬 때 기존의 단말들이 사용하는 단말의 재접속 요청에 대한 AP의 응답을 사용하므로 추가적인 단말의 수정 없이 인터페이스간 단말 이동 구현이 가능하다. 따라서 기존에 사용중인 모든 단말에 추가적인 설치 없이 적용이 가능하므로 구현에 드는 비용이 적다.

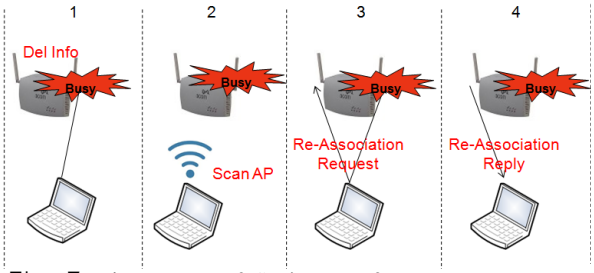

Fig. 7. The Process of Station Tranfer

4.3. MIAP 구현

우리는 실험을 위해 MIAP를 직접 설계하고 구현하였다. 그림 8은 MIAP의 구조를 그림으로 나타낸 것이다. 수정된 madwifi 디바이스 드라이버를 통해 무선 채널 트래픽을 모니터링하고 각 인터페이스에 접속한 단말들이 전송하는 트래픽 량을 측정한다. AP내부에 존재하는 관리 프로그램에서 각 인터페이스에서 측정한 무선 채널들의 트래픽 량을 수집하고 이를 통해 부하 분배를 수행한다. 관리 프로그램이 동작하는 유저 공간과 디바이스 드라이버가 존재하는 커널 공간 사이의 통신을 위해 IOCTL을 사용하였다.

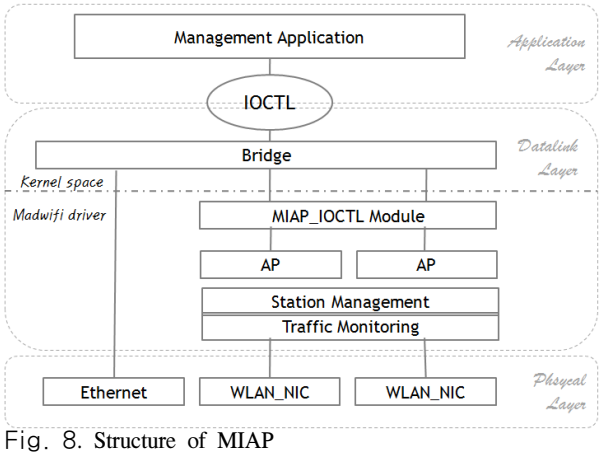

Fig. 8. Structure of MIAP

그림 9은 실제 하드웨어를 통해 MIAP를 구현한 것이다. Alix(AMD Geode LX based system boards) 임베디드 보드에 피라미드 리눅스 OS를 설치하였고 구현을 위해 수정한 madwifi 디바이스 드라이버와 부하 분배를 위한 모듈을 설치하였다. 2개의 무선 네트워크 인터페이스 카드를 장착하여 2개의 채널을 서비스하도록 하였고 유선 네트워크 인터페이스를 백본 네트워크와 연결하였다.

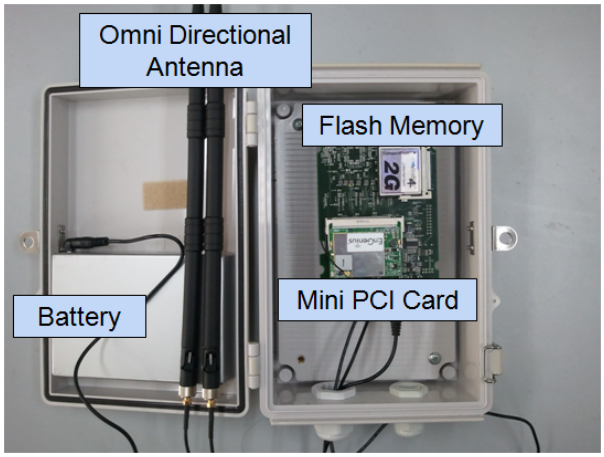

Fig. 9. The Hardware Configuration of MIAP

V. 실험 및 분석

5.1. 실험 시나리오

구현한 MIAP의 성능 검증을 위해 우리는 Simulation이 아닌 실제 실험을 준비하였다. 평가를 위한 네트워크를 구성하고 패킷 생성기를 통해 AP에 접속한 단말들이 트래픽을 발생시키도록 했고, 단말들이 보낸 패킷을 AP를 통해 서버가 수신하여 수신 로그를 바탕으로 성능을 평가하였다. 기존의 AP와 MIAP, 여러 단말들과 서버 PC를 준비하여 성능 비교 평가를 위한 3개의 시나리오를 준비하여 각 시나리오별 네트워크를 구성하였다. 아래 그림 10과 같이 네트워크를 구성한 후 AP를 바꾸면서 성능을 측정하였다.

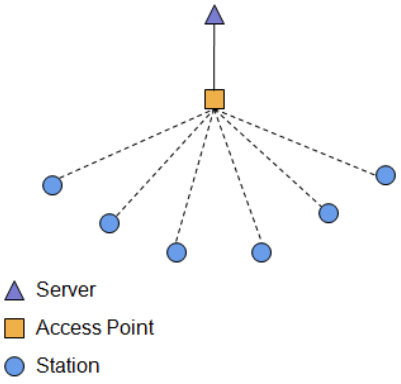

Fig. 10. Experiment Network Configuration

5.1.1 단일 인터페이스 AP 성능 평가

이 시나리오는 기존 AP의 성능을 평가하기 위해 구성하였다. 단일 인터페이스 AP 하나를 설치하여 AP에 6개의 단말을 접속시켰다. AP에 유선으로 서버를 연결하였고 AP에 접속한 각 단말들이 시간에 따라 트래픽의 양을 변경하면서 발생시켰다. 해당 트래픽을 서버가 수신하여 성능을 측정하였다.

5.1.2 동적 부하 분배를 수행하지 않는 멀티 인터페이스 AP 성능 평가

두 번째 시나리오는 첫 번째 시나리오에서 기존 AP 대신 동적 부하 분배를 수행하지 않는 멀티 인터페이스 AP를 설치하여 성능을 측정하였다. 사용 가능한 무선 채널을 확장시켰을 때 기존 환경보다 어느 정도 성능 향상이 있는지 알아보기 위해 이 시나리오를 구성하였다.

5.1.3 동적 부하 분배를 수행하는 멀티 인터페이스 AP 성능 평가

세 번째 시나리오는 동적 부하 분배를 수행하는 멀티 인터페이스 AP를 설치하여 성능을 측정하였다. 사용 가능한 무선 채널을 늘리고 무선 채널별 부하를 측정해 동적으로 부하 분배를 수행할 때 어느 정도 성능이 향상되는지 알아보았다.

Table 3. The Amount of Bandwidth for Each Station over the time(Mbps)

단말 / 시간	T1	T2	T3	T4	T5	T6
A	0	1	2	4	0	0
B	0	1	2	4	4	6
C	0	1	2	4	4	4
D	0	1	2	2	6	8
E	0	1	2	2	2	4
F	0	1	2	2	6	4

위의 세 가지 시나리오에 대해서 단말에서 동일한 트래픽을 발생시켜 실험을 수행하였다. 6개의 단말을 통해 트래픽을 위의 표 3과 같이 시간이 지남에 따라 발생시키는 트래픽 량을 다르게 하여 실험을 수행하였다.

5.2. 실험 결과 분석

세 가지 시나리오에 따라 각각 네트워크를 구성하여 실험을 진행하였다. 세 가지 시나리오 각각에 대해 서버의 수신 로그를 분석하여 시나리오별 전체 성능과 딜레이 지터를 측정하였다.

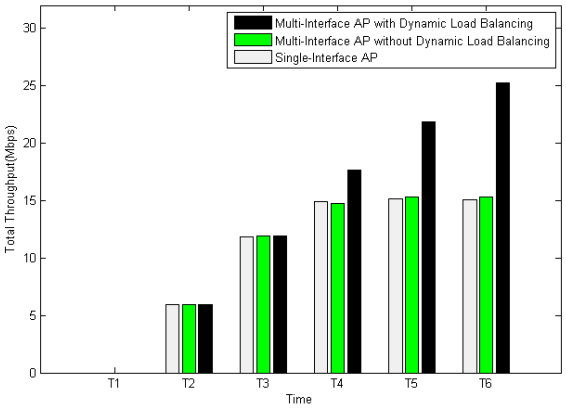

Fig. 11. The Overall Performance for Each Scenario over the time

그림 11을 통해 트래픽 발생량 증가에 따른 성능을 확인할 수 있다. 일정 수준 이하의 트래픽이 발생할 때에는 성능의 차이를 거의 보이지 않지만 그 이상 트래픽이 발생하면 단일 인터페이스 AP의 경우 성능 저하 현상이 나타난다. 그리고 멀티 인터페이스 AP를 설치하였지만 동적 부하 분배를 수행하지 않을 경우 역시 실제 2개의 무선 채널을 사용가능하도록 설치했음에도 불구하고 단말들이 사용하는 무선 채널은 하나이고 이를 AP에서 해결하지 않으므로 단일 인터페이스 AP 하나가 있는 환경과 거의 비슷한 성능을 보여준다. 마지막 시나리오인 부하 분배를 수행하는 멀티 인터페이스 AP를 설치한 경우에는 앞의 두 시나리오에서 성능 저하가 발생하는 트래픽 량 이상을 원활하게 처리하는 것을 확인할 수 있다.

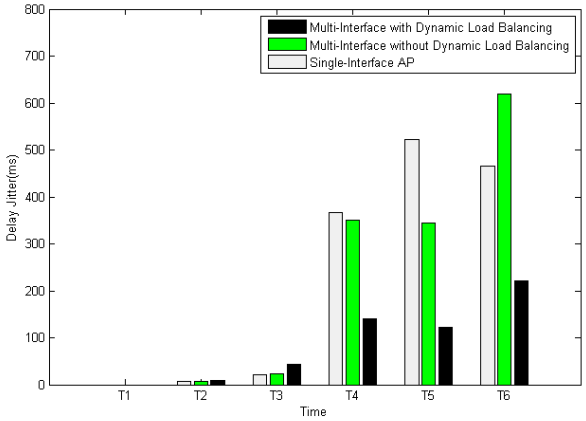

Fig. 12. The Inter-Arrival Jitter for each Scenario over the time

그림 12는 위의 실험에 대한 패킷 별 딜레이 지터를 나타낸 것이다. 단일 인터페이스 AP와 부하 분배를 수행하지 않는 멀티 인터페이스 AP의 경우 T3 이후부터 딜레이 지터가 증가하는데 이는 단말과 AP사이 링크의 안정성이 떨어진다는 것을 의미한다. 반면 부하 분배를 수행하는 멀티 인터페이스 AP의 경우 딜레이 지터가 앞의 두 시나리오보다 낮게 나타나는 것을 볼 수 있다.

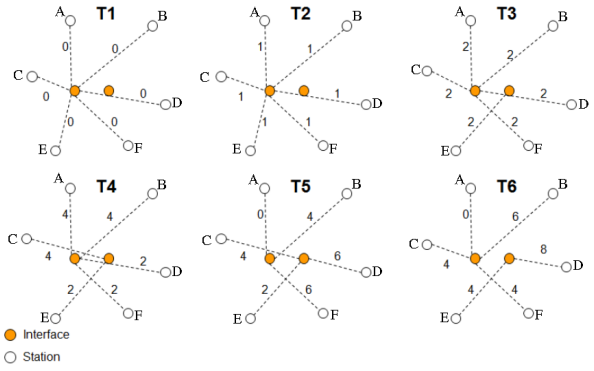

Fig. 13. The Network Topology for Each Time

그림 13은 발생시키는 트래픽 량의 변경에 따라 동적 부하 분배로 인한 단말의 이동을 그림으로 나타낸 것이다. 트래픽을 발생시키기 전에 6개의 단말이 MIAP에 동시에 접속하였고 더 적은 MOR을 가진 하나의 인터페이스에 모두 접속하여 T1과 같은 형태를 이루었다. T2에서는 각 단말이 트래픽을 발

생시켰지만 MOR이 Threshold를 넘지 않아 단말 이동을 수행하지 않았고, T3이 되면서 MOR이 Threshold를 초과하여 E단말을 다른 인터페이스로 이동시켰다. T4에서는 C단말이 추가로 이동하였고 T5에서는 D단말이 이동하였다. T6에서는 두 번째 인터페이스의 부하가 증가하여 C단말이 다시 처음 인터페이스로 이동하였다. 위의 그림과 같이 MIAP에서 동적 부하 분배를 통해 실시간으로 부하를 분배하여 두 개의 무선 채널을 효율적으로 사용하도록 하였고 이를 통해 전체 성능을 약 72%정도 향상시켰고 딜레이 지터의 증가도 방지하였다.

Ⅵ. 결 론

본 논문에서는 AP에서 자신이 사용하는 무선 채널을 실시간으로 감시하고 동적으로 부하 분배를 수행하는 멀티 인터페이스 AP를 설계하였고, 직접 구현하여 실험을 수행해 성능을 검증하였다.

한정된 무선 채널 자원 특성을 고려하여 트래픽이 많이 발생하는 지역에 AP를 추가로 설치하는데 본 논문에서 제안한 멀티 인터페이스 AP는 하나의 AP에서 여러 인터페이스를 통해 여러 개의 AP를 설치하는 효과를 얻고 이는 비용 측면에서 이점이 있다.

뿐만 아니라 멀티 인터페이스 AP 내부에서 실시간으로 각 인터페이스들이 사용하는 무선 채널의 트래픽 량을 감시하여 이를 통해 특정 무선 채널에 부하가 집중될 경우 인터페이스간 단말 이동을 수행하여 동적으로 부하를 분배하도록 하였다.

기존 AP와 비교 평가를 위해 시나리오를 구성하여 직접 실험을 수행하였고 그 결과 동적 부하 분배를 수행하는 멀티 인터페이스 AP를 사용하면 기존 환경보다 약 72%의 성능 향상을 이끌어낼 수 있고 딜레이 지터 또한 매우 안정화시킬 수 있다는 것을 확인하였다.

향후 연구에서는 본 논문에서 제안한 부하 분배 알고리즘과 다른 부하 분배 알고리즘을 비교하여 성능을 평가하는 것이 필요할 것이다. 그리고 더 나아가 동적 부하 분배를 위해 단말을 이동시킬 때 약간의 연결 단절로 인한 데이터 손실과 일시적인 딜레이 지터 증가 현상이 발생하는데 좀 더 빠르고 효율적인 인터페이스간 단말 이동 방법을 고려하여 성능 저하 현상을 방지할 것이다.

참 고 문 헌

[1] IEEE 802.11 Working Group, Part 11: wireless LAN medium access control(MAC) and physical layer(PHY) specifications, ANSI/IEEE Std. 802.11, March 2007.

[2] IEEE 802.11g Working Group, Part 11: wireless LAN medium access control(MAC) and physical layer(PHY) specifications Amendment 4: Further Higher Data Rate Extension in the 2.4GHz Band, June 2003.

[3] A. Balachandran, P. Bahl, G. M. Voelker, "Hot-Spot Congestion Relief in Public-area Wireless Networks", *Proc. of 4th IEEE Workshop on Mobile Computing Systems and Applications*, June 2002.

[4] H. Velayos, V. Aleo, G. Karlsson, "Load Balancing in Overlapping Wireless LAN Cells", *IEEE Communications Society*, July 2004.

[5] Y. Bejerano, S. J. Han, L. Li, "Fairness and Load Balancing in Wireless LANs Using Association Control", *IEEE/ACM Transaction on Networking*, Vol. 15, No. 3, June 2007.

[6] Y. Bejerano, S. J. Han, "Cell Breathing Techniques for Load Balancing in Wireless LANs", *25th IEEE International Conference on Computer Communications*, 2006.

[7] A. Mishra, V. Brik, S. Benerjee, A. Srinvasan, W. Arbaugh, "A Client-driven Approach for Channel Management in Wireless LANs", *25th IEEE International Conference on Computer Communications*, 2006.

[8] J. Yeo, M. Youssef, A. Agrawala, "A Framework for Wireless LAN Monitoring and its Applications", *3rd ACM workshop on Wireless security*, 2004.

[9] P. Bhagwat, P. Bhattacharya, A. Krishna, S. K. Tripathi, "Enhancing throughput over Wireless LANs using Channel State Dependent Packet Scheduling", *Fifth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation*, 1996.

[10] Y. Shu, M. Yu, J. Liu, "Wireless Traffic Modeling and Prediction Using Seasonal ARIMA Models", *Communications ICC '03.*

IEEE International Conference, 2003.

[11] F. Hernandez-Campos, M. Papadopouli, "Assessing the real impact of 802.11 WLANs: a Large-scale Comparison of Wired and Wireless Traffic", *Local and Metropolitan Area Networks*, 2005.

김 태 근 (Tae-keun Kim) 준회원

2010년 2월 부산대학교 정보컴퓨터공학부 졸업

2010년 3월~현재 부산대학교 컴퓨터공학과 석사과정

<관심분야> 무선통신, 링크측정

서 형 윤 (Hyung-Yoon Seo) 준회원

2002년 2월 한서대학교 컴퓨터정보공학과 졸업

2011년 2월 부산대학교 컴퓨터공학과 석사

2011년 2월~현재 부산대학교 컴퓨터공학과 박사과정

<관심분야> 무선 통신, 방송통신융합

김 종 덕 (Jong-Deok Kim) 종신회원

1994년 2월 서울대학교 계산통계학과 졸업

1996년 2월 서울대학교 전산과학과 석사

2003년 2월 서울대학교 컴퓨터공학과 박사

2004년 2월~현재 부산대학교 정보컴퓨터공학부 부교수

<관심분야> 무선통신, 라우팅