

WLAN 과 Bluetooth 를 동시에 지원하는 Access Point 에서의 QoS 보장 연구

박진희*, 김대환*, 윤정미*, 김용호*

*전자부품연구원

e-mail : pjhe@keti.re.kr

A Study on QoS in Access Point supporting WLAN and Bluetooth

Jin-Hee Park*, Dae-Hwan Kim*, Jung-Mee Yun*, Yong-Ho Kim*
*KETI (Korea Electronics Technology Institute)

요 약

초고속 통신망과 모바일 컴퓨팅 디바이스의 발전이 급속도로 이루어지고 있고, 블루투스나 무선랜 등 무선 네트워크 프로토콜들이 등장함에 따라 가정, 캠퍼스, 회사나 공공 지역에서 무선 인터넷 사용자들이 늘고 있다. 이에 따라 많은 대역폭을 요구하는 멀티미디어 어플리케이션과 다양한 레벨의 품질을 요구하는 어플리케이션이나 사용자들의 요구 또한 늘어가고 있어 이에 대한 QoS 제공 방안이 촉구된다. 따라서 본 논문에서는 무선 인터넷을 가능하게 하는 무선랜과 블루투스를 동시에 지원하는 Access Point 에서 QoS 지원 방안에 대해 고찰해보도록 한다.

1. 서론

PDA 나 노트북과 같은 모바일 컴퓨팅 디바이스가 대중화 되면서 무선으로 인터넷에 접속하려는 요구가 증가하고 있다. 또한 한국통신 공중망 무선랜 서비스가 무선 인터넷 사용자를 더욱 증가하게 하였다. 그리고 인터넷 서비스 이용자들은 텍스트화된 서비스 보다는 다양한 멀티미디어가 가미된 인터넷을 사용하고 이의 응용도 계속 발전하고 있다. VDSL, E-벨리 등과 같은 50M 이상의 초고속 통신망의 발전이 급속화 됨에 따라 무선 인터넷 사용자들의 요구도 점차 다양화 되고 있다. 이에 본 논문에서는 무선 인터넷 사용자들을 위한 무선랜과 블루투스를 동시에 지원하는 AP 에서의 QoS 보장을 위한 방안에 대해 고찰해보자 한다. 본 논문은 2 장에서 QoS 연구에 앞서 무선 인터넷 사용을 가능하게 해주는 실제 임베디드 시스템에서 구현한 무선랜과 블루투스가 탑재된 AP 설계에 관해 살펴 볼 것이며, 3 장, 4 장, 5 장에서는 2 장에서 구현된 AP 에 QoS 를 제공하기 위한 방안을 무선랜, 블루투스, 구현 시스템 각각의 측면에서 논할 것이며 6 장에서 결론을 맺도록 하겠다.

2. 무선랜과 블루투스 AP 설계

본 논문에서 무선 네트워크 프로토콜에서의 QoS 구현 방안을 연구하기 위해 무선랜과 블루투스가 동시에 동작하는 AP 를 삼성 ARM9 S3C2510 보드에 구현하였다. OS 는 임베디드 리눅스로 구현하였고, 무선랜은 802.11b PCMCIA 카드형태로 PCMCIA 소켓에 부착되었으며 드라이버와 프로토콜 스택은 리눅스 공개 소스인 linux-wlan-ng 를 사용하였고, 블루투스는 USB 형태로 구현되었으며 드라이버와 프로토콜 스택은 bluez 를 사용하였다. 무선랜과 블루투스 AP 에서 QoS 제공 기법은 MAC 이하의 하위 레이어와 네트워크 이상의 상위 레이어에서 고려해 보도록 하겠다.

3. 무선랜에서의 QoS

무선랜은 IEEE 802.11 워킹 그룹에서 표준안이 제정되고 있으며, 현재 802.11 (1~2Mbps), 802.11a (54Mbps), 802.11b (11Mbps), 802.11g (54Mbps)가 표준으로 채택되어 있다. 그러나 현재 널리 사용되는 버전은 최대 11Mbps 를 지원하는 802.11b 이므로 본 논문에서는 802.11b 만을 고려하기로 한다. 특히 IEEE 802.11E 태

스킹 그룹에서는 MAC 이하의 하위 레이어에서 QoS 보장기법에 대해 연구하고 있는데 이에 대해 살펴보고자 한다.

802.11E는 QoS를 지원할 수 없는 기존의 802.11 MAC 구조에 QoS를 지원하도록 확장한 MAC 메커니즘이다. 802.11E MAC은 HCF (Hybrid Coordination Function)이라고도 불리며, 802.11 MAC의 DCF와 PCF로 불리워지는 controlled와 contention-based channel 메커니즘 모두를 지원한다.

HCF contention-based channel access 방식은 Enhanced DCF (EDCF)라고 불리며, 표 1과 같이 8개로 구분되어진 user priority에 의해 무선 채널 접근 기회를 서로 다르게 부여하여 QoS를 보장하고, Ad-Hoc과 Infrastructure 모드에서 동작한다.

표 1. User Priority와 Access Category 매핑

User Priority (802.1D Priority)	802.1D Designation	Access Category	Designation (Informative)
1	BK	0	Best Effort
2	-	0	Best Effort
0	BE	0	Best Effort
3	EE	1	Video Probe
4	CL	2	Video
5	VI	2	Video
6	VO	3	Voice
7	NC	3	Voice

이에 반해 HCF controlled channel access 방식은 QoS를 AP가 결정하는 것으로, 트래픽을 전송하기 전에 Traffic Stream이라는 가상 회선을 셋업하여 트래픽 특성과 QoS 요구 파라미터를 서로 교환하고 협상하고 AP는 이 QoS 파라미터에 근거하여 QSTA (QoS Station)들의 무선 대역폭을 보장해주기 위해 polling frame을 통하여 트래픽을 스케줄링하는 방식이며 이는 Infrastructure 모드에서 동작한다.

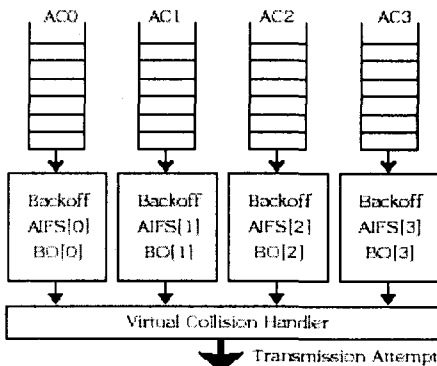


그림 1. EDCF에서의 4개의 AC

EDCF 방식의 상세 설명은 다음과 같다. EDCF는 표 1과 같이 0~7까지의 8개의 user priority를 가진 프레임의 서로 다른 channel access를 제공한다. 상위 레이어에서 내려온 프레임과 user priority는 MAC data frame과 header에 의해 전송된다. 802.11e STA는 4개의 AC (Access Categories)로 구현되는데, 이는 8개의 user priority와 매핑된다. 그림 1과 같이 802.11e는 4개의 transmission queue를 가지고 있는데, 이들은 각각 자신의 AIFS를 가지고 자신의 Backoff Counter를 유지하는 큐가 있는 AC인 EDCF로 동작한다. 하나 이상의 AC가 동시에 backoff를 끝냈을 때 collision 해결 방법은 가장 우선 순위가 높은 프레임이 선택되어 전송되고, 다른 AC의 프레임들은 CW 값을 증가시키고 backoff를 수행한다.

HCF Controlled Channel Access 방식은 다음과 같다. EDCF가 서로 다른 user priority traffic을 사용하여 QoS를 제공한다면 HCF Controlled Channel Access는 AP와 QSTA 간에 서로 계약된 파라미터를 통해서 QoS가 제공된다. QoS 파라미터를 가진 프레임을 전송하기 전에 traffic stream이라 부르는 가상 회선이 연결된다. Traffic stream은 QSTA와 AP 간의 uplink, AP와 QSTA 간의 downlink, QSTA와 QSTA 간의 directlink로 나뉜다. Traffic stream이 setup되기 위해 traffic 특성의 집합과 QoS 요구 파라미터들이 AP와 QSTA 간에 교환되고 협상된다. Traffic stream이 setup되면 AP안의 HC (Hybrid Coordinator)가 HCF controlled channel access에 사용되는 traffic stream에게 요구된 대역폭을 할당하기 위해 서로 계약된 QoS를 공급하여 QoS보장이 이루어지게 된다.

4. 블루투스에서의 QoS

많은 bluetooth device들은 data rate, delay variance, reliability 등 서로 다른 요구사항을 가지고 있다. Specification은 상위 레이어 어플리케이션이나 프로토콜의 요구사항에 의한 링크의 속성을 위해 QoS configuration을 제공한다. 이 속성들은 QoS type, token rate, token rate bucket size, peak bandwidth, latency, delay variation 등을 포함한다.

QoS를 컨트롤하기 위해 Bluetooth protocol stack 내부에서는 QoS flow를 설정하고 setup하는 메시지들은 스택의 레이어 사이에서 상하의 수직적으로 움직이는 반면 Link manager와 Logical Link control and Adaptation layer (L2CAP)은 peer와 peer 협상을 통해서 QoS를 설정한다. Link manager는 QoS 정책을 구현하고, baseband link를 컨트롤하며 L2CAP request의 QoS를 만족시키기 위해서 노력한다.

처음 링크가 setup되면 QoS는 상위 레이어에서 L2CAP으로 요청된다. 그러면 QoS 설정 협상 패킷들은 local과 remote L2CAP 사이에서 보내지게 된다. Link manager는 L2CAP의 request에 의한 QoS capabilities를 제공한다. HCI를 가진 시스템에서 L2CAP와 Link manager 사이의 상호작용은 HCI command와 event를 통해서 수행된다. LMP command

는 poll interval 과 Master 에서 Slave 로 보내지는 패킷 사이의 maximum interval 과 broadcast packet 반복 시간을 설정하는데 사용될 수 있다. QoS setup 은 LMP 의 setting 이 끝났을 때 수행된다. 만약 실패하면 상위 레이어로 시도를 계속 할 것인지 포기할 것인지에 대해 메시지를 보내게 된다. 만약 성공하면 채널은 의도된 QoS 로 데이터를 전송하기 위해 오픈 됨으로써 QoS 보장이 이루어지게 된다.

5. 구현 시스템에서의 QoS

3 장과 4 장은 무선랜과 블루투스에서 MAC 레이어 이하에서 QoS 를 제공하기 위해 실제 Standard Specification 에 규정되어 있는 내용을 살펴보았다. 그러나 본 장에서는 IP 레이어 이상의 상위 계층과 구현 시스템 특히, 리눅스 시스템에서 QoS 를 제공해 줄 수 있는 방안을 살펴보기로 한다. 또한 무선랜과 블루투스를 동시에 지원하는 본 논문에서 고려한 Access Point 는 이를 인터넷망으로 연결시켜주는 분산시스템으로써 ADSL, VDSL, Cable modem, E-밸리 등과 같은 초고속 통신망과 연계된다.

즉 무선랜과 블루투스 RF 로부터 Access Point 로 오는 트래픽은 Access Point 의 TC (Traffic Control) 시스템을 거치게 되는데 이를 통해 각기 다른 QoS 를 적용하고자 하는 트래픽에 따라 적절한 QoS 를 제공할 수 있다. TC 시스템은 본 Access Point 가 리눅스로 구현되었기 때문에 리눅스 Traffic Control 메커니즘을 따르게 된다.

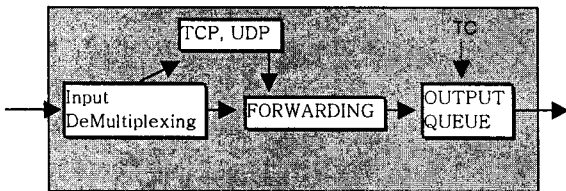


그림 2. TC (Traffic Control)

그림 2 와 같이 커널안에서 출력큐 부분에서 TC 가 적용될 수 있다. 우선 네트워크 인터페이스로 들어오는 패킷의 목적지 주소를 Input DeMultiplexing 을 통해서 판별하게 되는데, 만약 AP 자신이면 TCP 나 UDP 와 같은 상위 계층으로 패킷을 올려주고, 그렇지 않다면 Forwarding 모듈로 전달하여 라우팅 테이블을 검사하여 다음 홉을 결정한다. 그런 후, 이 패킷은 출력큐로 보내지게 되는데, 이 시점에서 클래스와 필터와 큐잉전략으로 이루어진 TC 가 동작하게 된다. 큐잉전략은 Class Based Queue (CBQ), Token Bucket Flow (TBF), Clark-Shenker-Zhang (CSZ), First In First Out (FIFO), Priority, Traffic Equalizer (TEQL), Stochastic Fair Queuing (SFQ), Random Early Detection (RED)등 리눅스에서 많은 기법을 제공하고 있으며 주번호와 부번호로 구성되는 핸들에 의해 클래스와 연결된다. 큐잉전략은 해당 디바이스의 출력 큐에 도착한 패킷을 어떤 방식으

로 처리할 것인가를 결정해 준다. 일단 패킷이 도달하여 큐의 Enqueue 함수를 호출하게 되면 큐잉전략은 필터를 적용하여 어떤 패킷이 어떤 클래스에 속하게 될지 결정하게 되고, 그 클래스들 사이에 서로 다른 우선 순위를 적용하여 패킷의 전송을 차별화 할 수 있게 된다. 이때의 큐잉전략은 패킷의 delay, reliability, bandwidth 등이 될 수 있다. 필터는 패킷을 클래스화 하는데 사용되며 IP 헤더의 TOS 나 IP 주소, 포트 번호등을 사용할 수 있다. 이와 같은 리눅스 커널안의 TC 를 이용하기 위해 사용자 프로그램인 tc (traffic controller)가 사용된다. 이것은 다양한 종류의 큐잉전략을 생성시킬 수 있고 출력 디바이스와 연계시킬 수도 있다. 또한 라우팅 테이블이나 u32 classifier, tcindex classifier, RSVP classifier 에 근거하여 필터를 설정할 수 있다.

또한 TC 는 그림 3 과 같이 인터넷 QoS 프레임워크인 IntServ 와 DiffServ 를 구현할 수 있는 기본 프레임워크를 제공하고 있다. 이로인해 리눅스 오픈소스로 RSVP 나 DiffServ 를 이용한 QoS 를 구현할 수도 있다. 그러나 본 논문에서는 RSVP 나 DiffServ 는 기간망 차원에서 이를 지원하는 네트워크가 이미 구현되어 있어야 그 실제 적용이 가능하므로, 이를 실제 구현하지는 않고, 단지 리눅스의 TC 시스템을 이용하여 트래픽에 서로 다른 우선 순위를 적용하여 QoS 를 제공하는 기법에 대해서만 고려를 하였다. 이에 대한 고려 방안은 차후의 연구로 남기도록 한다.

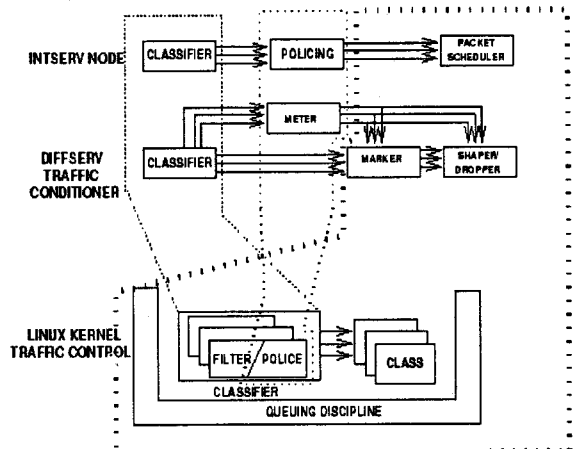


그림 3. IntServ 와 DiffServ 와 연계된 프레임워크

6. 결론 및 향후과제

본 논문에서는 무선랜, 블루투스, 시스템 차원으로 나누어서 각각의 경우에 QoS 지원에 대한 방안에 대해 고찰해 보았다. MAC 레이어 이하의 WLAN MAC 과 Bluetooth MAC 부분에서 생각해 볼 수 있는 QoS 방안과 IP 레이어 이상의 Traffic control 과 QoS 제공 프레임 워크 부분에 대해 생각해 보았다. 무선 인터넷의 사용자는 더욱 증가하게 될 것이고, QoS 제공의 필요성은 크게 대두 될 것이다. 따라서 본 논문에서 고

찰한 내용이 WLAN 과 Bluetooth 로 대표되는 무선 인터넷망에서의 QoS 고려 방안은 이들을 가능하게 하는 Access Point 구현에 있어서 앞으로 중요사항이 될 것이다. 향후과제로 실제 임베디드 장비에서 이들을 구현하여 QoS 보장이 얼마나 다양한 품질의 무선 인터넷을 사용할 수 있게 하는지에 대한 실험을 하는 것이다.

참고문헌

- [1] Sunghyun Choi, "Emerging IEEE 802.11e WLAN for Quality-of-Service (QoS) Provisioning", Telecommunications Review, Vol.12 No.6, 2002.
- [2] IEEE Std 802.11-1999, Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Reference number ISO/IEC 8802-11:1999(E), IEEE Std 802.11, 1999 edition, 1999.
- [3] IEEE Std 802.11b, Supplement to Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-speed Physical Layer Extension in the 2.4GHz Band, IEEE Std 802.11b-1999, 1999.
- [4] IEEE 802.11e/D3.3, Draft Supplement to Part II: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), Oct. 2002.
- [5] Bluetooth SIG Radio Working Group, "Radio Improvements: Proposal for L2CAP Flow and Error Control", QoS Team [BT1-QoS], 2002
- [6] Saravanan Radhakrishnan, "Linux-Advanced Networking Overview Version 1", Univ. of Kansas, 1999.