Ad-Hoc Network을 위한 QoS 프레임웍
(A QoS Framework for Ad-Hoc Networks)

김 준 형 † 모 상 덕 †† 정 광 수 †††
(Junhyung Kim) (Sangdok Mo) (Kwangsue Chung)

요 약 Ad-Hoc Network의 실용화에 대한 기대가 커짐에 따라 다양한 응용의 안정적인 서비스를 위해 QoS의 지원을 위한 연구가 필요하게 되었다. Ad-Hoc Network에서 QoS를 지원하기 위한 기존의 연구들은 정량적인 서비스를 보장하기 어렵거나, 네트워크 오버헤드를 발생시키는 문제점을 가지고 있었다. 본 논문에서는 이러한 문제를 해결하기 위해 새로운 QFAN(QoS Framework for Ad-hoc Networks)을 제안한다. 본 논문에서 제안하는 QFAN은 네트워크 오버헤드를 최소화하면서 실시간 트래픽의 최소 대역폭을 보장할 수 있다. 그리고 실시간 트래픽과 최선형 트래픽간의 잉여 대역폭에 대한 공정분배를 통해 대역폭 불균형 문제를 해결할 수 있다. 제안된 프레임웍을 실제 적용하기 위해서 TiRe(Tiny Reservation)과 ADR(Adaptive Drop Rate) 컨트롤 알고리즘을 설계하였다. 시뮬레이션을 통해 실시간 트래픽의 최소 요구 대역폭을 만족시키면서, 실시간 트래픽과 최선형 트래픽이 잉여 대역폭을 공정 분배하는 것을 확인 할 수 있었다.

키워드 : Ad-Hoc Network, QoS, 대역폭 보장, 공정성, QFAN, TiRe, ADR

Abstract Research about QoS in the ad-hoc networks for stable service of various applications has been needed as the expectation about the realization of the ad-hoc networks grows bigger. Existing researches about QoS in the ad-hoc network had the problems which can not guarantee the quantitative services or create the overhead. In this paper, we propose a novel algorithm of QFAN(QoS Framework for Ad-hoc Networks) the framework to resolve such problems and considered application of the proposed algorithm into the ad-hoc networks. Our model can guarantee the minimum bandwidth of the real-time traffic as minimized the overhead. And, disproportionate distribution of bandwidth problem can resolve by the proposed algorithm through the fair share between real-time traffic and best-effort traffic about available bandwidth. We design both the TiRe(Tiny Reservation) and the ADR(Adaptive Drop Rate) control algorithm to apply the proposed QFAN. Using simulation, we confirm fair share of available bandwidth between real-time traffic and best-effort traffic as guarantee minimum required bandwidth of real-time traffic.

Key words : Ad-Hoc Network, QoS, Bandwidth Guarantee, Fairness, QFAN, TiRe, ADR

1. 서 론

Ad-Hoc Network이란 무선통신 장비를 가진 두 개 이상의 장치가 액세스 포인트(access point)나 기지국과 같은 인프라스트럭처(infrastructure)의 도움 없이 시간과 장소의 구애를 받지 않고 통신이 가능한 네트워크이다. 향후 많은 응용 어플리케이션들이 이러한 네트워크를 이용하게 될 것이다. 특히 멀티미디어 컨텐츠나 화상회의와 같은 스트림 데이타 서비스의 요구에 따라 Ad-Hoc Network은 단순한 통신의 기능을 뛰어 넘어 좀 더 안정적이고 고품질의 서비스를 필요로 하게 될 것이다[1]. 따라서 최소 대역폭이나 지연 시간의 보장을 필요로 하는 서비스를 위해 QoS의 지원이 필요하다[2]. 현재 IETF(Internet Engineering Task Force)의 MANets(Mobile Ad-hoc Networks) 그룹에서 이에 대한 연구가 활발히 진행 중이다[3]. 다양한 서비스 중 특정 트래픽에 대한 대역폭 보장은 중요하면서도 보편적인 QoS 지원 방법이다. 대역폭 보장을 위한 QoS 지원 연구 중 많은 부분이 유선 환경에서의 QoS 기술을 무

· 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 육성·지원 사업과 2003년도 광운대학교 연구년에 의하여 지원되었음
† 비 회 원 : 광운대학교 전자공학부
junhyung77.kim@samsung.com
†† 학생회원 : 광운대학교 전자공학부
sdmo@adams.kw.ac.kr
††† 종신회원 : 광운대학교 전자공학부 교수
kchung@kw.ac.kr
논문접수 : 2004년 1월 7일
심사완료 : 2004년 11월 22일

선 환경에 적용하는 방법으로 시도되고 있다. 그 중 대표적인 방법이 시그널링(signaling)을 이용하는 자원 예약(Resource reservation) 방법과 서비스 클래스(service class)에 따라 우선순위를 부과하는 DiffServ (differentiated Service) 방법이다[4,5].

자원 예약 방법은 예약 상태에 대한 유지가 필요하므로 주기적인 갱신 메시지를 필요로 한다. 잦은 갱신 메시지는 좁은 대역폭을 가지는 무선 환경에 큰 오버헤드로 작용하여 실제 데이타의 전송 성능을 저하시킬 수 있다. 반면 DiffServ를 이용하는 경우는 주기적인 갱신 메시지를 필요로 하지 않는다. 그러나 단말의 움직임에 따라 코어 노드(core node)와 에지 노드(edge node)의 경계가 불분명하므로 코어 노드와 에지 노드간의 역할 분담이 어렵다[6,7]. 또한 상대적인 우선순위에 의해 서비스를 제공하므로 정량적인 서비스 제공에는 무리가 있어 실시간 트래픽이 요구하는 최소 대역폭에 대한 보장이 어렵다. 따라서 실시간 트래픽의 최소 대역폭을 만족시키면서 다른 트래픽의 전송 성능을 저하시키지 않도록 하는 새로운 QoS 메커니즘이 필요하다.

본 논문에서는 이러한 문제를 해결하기 위해 Ad-Hoc Network의 특성에 적합하도록 경량화된 새로운 QoS 프레임웍인 QFAN(QoS Framework for Ad-hoc Networks)을 제안하였다. QFAN은 대역폭의 보장을 주요한 서비스 지표로 하여, 실시간 트래픽의 최소 대역폭을 보장하도록 설계되었다. 일반적으로 실시간 트래픽은 UDP 연결을, 최선형 트래픽은 TCP 연결을 가지는데, 본 논문에서도 실시간 트래픽은 UDP 연결을, 최선형 트래픽은 TCP 연결을 가지는 것으로 가정하였다. UDP와 TCP 트래픽이 경쟁할 때 공격적으로 대역을 점유하는 UDP 트래픽에 의해 TCP 트래픽의 전송률이 현저히 낮아지는 문제점이 발생할 수 있다[8]. 이를 해결하기 위해 본 논문에서 제안하는 QFAN은 최선형 트래픽의 성능 저하 현상을 방지하고 잉여 대역폭에 대한 실시간 트래픽과 최선형 트래픽간의 공정한 분배를 실현하도록 설계되었다. 그리고 QoS 프레임웍을 실제 적용하기 위해 TiRe(Tiny Reservation) 알고리즘과 ADR (Adaptive Drop Rate) 컨트롤 알고리즘의 두 가지 방법을 제안하였다. TiRe 알고리즘은 실시간 트래픽의 대역폭 보장을 위한 자원 예약 방법으로 주기적인 시그널링(signaling) 메시지에 따르는 오버헤드를 제거하면서 최소 대역폭의 보장을 수행한다. ADR 컨트롤 알고리즘은 UDP 트래픽의 유입에 따라 TCP 트래픽의 성능저하가 발생하는 것을 방지하여 잉여 대역폭에 대한 공정한 분배를 위한 알고리즘이다. 이 두 가지 메커니즘을 통해 실시간 트래픽의 서비스 보장과 기타 트래픽에 대한 공정한 자원 분배에 대한 문제를 해결하였다.

본 논문의 구성은 다음과 같다. 2장에서는 Ad-Hoc Network에서 QoS를 지원하기 위해 고려할 점을 기술하고, 그에 따른 관련 연구와 기존 연구들의 문제점에 대해서 기술하였다. 3장에서는 2장에서 제시된 문제점을 해결하기 위해 본 논문에서 제안하는 프레임웍과 알고리즘에 대해 기술하였다. 4장에서는 시뮬레이터를 통한 성능 검증과 마지막으로 5장에서 결론 및 향후 과제에 대해 기술하였다.

2. 관련연구

Ad-Hoc Network를 위한 QoS지원 방법을 살펴보기 전에 기존의 유선망에서의 QoS지원 방안으로 IETF에서 표준으로 채택한 IntServ와 DiffServ에 대한 연구를 간략히 살펴본다.

IntServ는 특정 사용자 플로우의 서비스 품질을 제공하기 위해 그 플로우가 경유하는 모든 라우터(router)에 자원 예약을 하고, 이를 기반으로 특정 서비스의 품질을 제공하는 서비스이다. IntServ는 플로우가 지나는 각 라우터에 자원 예약을 수행하기 위해 시그널링(signaling) 프로토콜인 RSVP(Resource Reservation Protocol)을 사용한다. 따라서 모든 노드는 자신을 지나는 플로우에 대한 정보를 저장하고 있어야 하므로 플로우의 수가 증가할 경우 라우터는 각 플로우의 정보를 저장하기 위해 큰 비용을 지불해야 한다. 또한 모든 라우터가 IntServ를 지원해야 하므로 확장성이 어렵다. DiffServ는 플로우의 수가 증가함에 따라 발생하는 비용을 줄이기 위해 각 플로우를 몇 개의 클래스로 나누어 처리한다. 즉 IntServ는 플로우 관리 서비스인데 반해 DiffServ는 클래스 관리 서비스라 할 수 있다. 클래스 관리 서비스는 서비스 레벨이 비슷한 몇 개의 플로우를 묶어 하나의 클래스로 정의하고, 제한된 수의 클래스만을 관리하는 방법이다. DiffServ는 다양한 플로우의 요구사항을 정확하게 만족시킬 수는 없지만, 클래스간의 상대적인 차등을 두어 서비스를 제공함으로써 IntServ의 한계를 극복하고자 하였다. DiffServ 네트워크의 노드는 에지 노드(edge node)와 코어 노드(core node)로 분류하고, 복잡한 기능은 에지 노드가 처리하고 코어 노드는 단지 전달만을 해주기 때문에 확장성의 측면에서도 매우 유리하다고 할 수 있다.

위와 같은 유선 환경하의 QoS 지원 기술을 Ad-Hoc Network에 그대로 적용할 경우, 여러 제약 사항이 존재한다. IntServ의 경우 각 노드는 자신을 지나는 모든 플로우에 대한 정보를 보관하고 있어야 하므로 저장공간의 제약이 있는 Ad-Hoc Network에는 적합하지 않다. 또한 경로 설정 시 자원 예약 프로토콜인 RSVP를 이용하는데, 무선 환경의 좁은 대역폭의 일부를 상태유지

에 필요한 갱신 메시지가 차지하게 되어 실제 전송해야 할 데이타의 전송률을 저하시키게 된다. 따라서 일반적인 자원 예약 프로토콜을 Ad-Hoc Network에서 사용하기 어렵다. DiffServ의 경우 클래스 별로 차등을 두며, 코어 노드에서는 플로우의 정보를 저장하지 않아도 되므로 저장공간의 제약을 크게 받지 않는다. 그러나 노드의 이동성으로 인해 코어 노드와 에지 노드 간의 경계가 불분명해지므로 역할 분담이 어렵다. 또한 계약을 체결한 노드의 움직임이 유동적이므로 각 클래스의 서비스 레벨을 명시하는 SLA(Service Level Agreement)의 적용이 어렵다. 위와 같은 이유로 유선망의 QoS 기술을 Ad-Hoc Network에 그대로 적용하는 것에는 무리가 있다.

2.1 FQMM

FQMM(Flexible QoS Model for Manets)은 노드의 개수가 50개미만의 Ad-Hoc Network에 적용하기 위한 QoS 지원 프레임웍이다[9]. FQMM은 기존의 DiffServ 모델을 기반으로 하여 노드를 인그레스 노드(ingress node), 인테리어 노드(interior node), 이그레스 노드(egress node)로 구분한다. 각각의 노드는 패킷의 시작, 전달, 도착점의 역할을 하게 된다. 노드의 이동성을 고려하여 각 노드는 연결 설정마다 그 역할이 달라질 수 있다. FQMM은 QoS를 위해 필요한 모든 구성 요소를 모아 프레임웍으로 만들고, 노드의 역할에 따라 필요한 구성 요소만을 선택하여 사용하도록 하였다. 그러나 무선 환경의 특성에 민감한 라우팅이나 시그널링 프로토콜에 대한 고려가 없고, Ad-Hoc Network의 특성을 고려한 구체적인 방법이나 기술의 제안 없이 프레임웍만을 제공하므로 실제 적용에 어려움이 있다.

2.2 SWAN

SWAN(Stateless Wireless Ad-Hoc Networks)은 Ad-Hoc Network에서 차등적 서비스를 제공하기 위한 무상태정보 네트워크 모델로 콜롬비아 대학 내의 COMET 그룹에서 SWAN 프로젝트로 진행이 되었다[10,11]. SWAN은 기본적으로 DiffServ를 기반으로 설계되었으며, 멀티미디어와 같은 실시간 플로우에 QoS를 지원하도록 하였다. 이를 위해 네트워크상의 트래픽을 실시간 플로우(real-time flow)와 최선형 플로우(best-effort flow) 나누어 실시간 플로우에는 마킹(marking)을 하여 상대적인 QoS를 제공하도록 하였다. 전달해야 할 실시간 데이타가 있을 때, 송신단은 프로브(probe)를 수신단까지 전달하고 다시 돌려받음으로써 경로상의 대역폭을 확인하고, 서비스가 가능하다고 판단되면 실시간 플로우에 마킹을 하여 전달하게 된다. 그러나 SWAN은 과도한 실시간 트래픽이 유입될 경우 어드미션이 연속적으로 발생할 수 있으며, 그에 따라 모든 실시간 트래픽의 전송률이 저하된다. 따라서 실시간성을 보장하기 위한 최소 대역폭을 만족시킬 수 없는 문제가 발생한다.

그림 1은 2Mbps의 대역폭을 가지는 링크에서, 700 Kbps의 최소 대역폭을 요구하는 세 개의 실시간 트래픽의 대역폭 점유율을 나타낸 것이다. 20초에서 세 번째 트래픽이 유입될 경우, 세 개의 트래픽이 모두 경쟁상태에서 최소 대역폭을 만족시키지 못하고 평균 전송률이 500Kbps로 떨어지는 것을 확인할 수 있다.

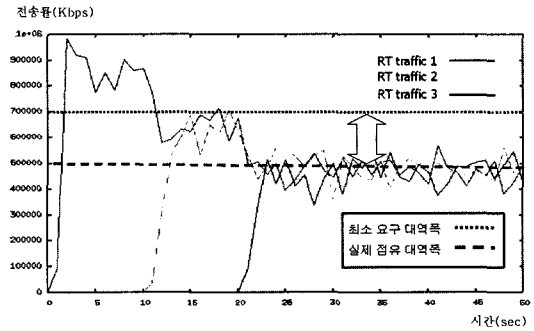


그림 1 실시간 트래픽의 경쟁 상태

그림 2는 700Kbps의 최소 요구 대역폭을 가지는 두 개의 실시간 트래픽과 두 개의 최선형 트래픽을 이용한 실험 결과 중, 하나의 실시간 트래픽의 전송률 곡선을 나타내고 있다. 일반적으로 UDP 연결을 가지는 실시간 트래픽은 TCP 연결을 가지는 최선형 트래픽에 비해 공격적으로 대역을 점유하기 때문에, 그림 2와 같이 최선형 트래픽의 성능 저하를 가져오게 된다. 즉 실시간 트래픽의 영향으로 실제 측정된 최선형 트래픽의 전송률은 대역을 공정 분배했을 때의 이상적인 최선형 트래픽의 전송률에 비해 큰 폭으로 떨어지는 것을 볼 수 있다.

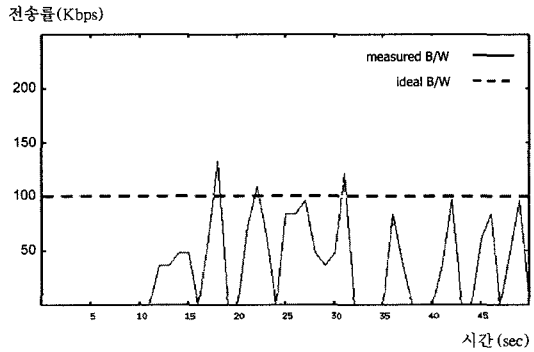


그림 2 최선형 트래픽의 성능 저하

2.3 INSIGNIA

INSIGNIA는 Ad-Hoc Network를 위해 설계된 인밴

드(in-band) 시그널링 프로토콜이며 콜롬비아 대학의 COMET 그룹에 의해 제안되었다[12]. RSVP와 같은 시그널링 프로토콜은 아웃오브밴드(out-of-band) 시그널링으로 시그널링을 위한 별도의 컨트롤 패킷을 전송하게 된다. 그러나 무선 네트워크의 제한된 대역폭 중 일부를 컨트롤 패킷이 점유함으로 인해 실제 데이타의 전달에 지장을 받게 된다. Ad-Hoc Network에서의 아웃오브밴드 시그널링은 컨트롤 패킷의 오버헤드가 크기 때문에 실제 적용에 무리가 있다. INSIGNIA는 이를 해결하기 위해 기존의 IP 헤더에 자원 예약을 위한 INSIGNIA 옵션을 붙여 보내는 방식을 취한다. 그러나 INSIGNIA는 시그널링을 위한 컨트롤 패킷을 없애는 대신 INSIGNIA 옵션을 추가하기 위해 IP 헤더를 수정해야 한다. 따라서 매 패킷마다 추가되는 옵션 헤더의 크기도 큰 오버헤드로 작용할 수 있으며, IP 헤더를 수정함에 따라 기존의 네트워크와의 연동이 어려워지는 문제점을 안고 있다.

2.4 dRSVP(Dynamic RSVP)

dRSVP는 기존의 자원 예약 프로토콜인 RSVP를 동적인 네트워크 환경에 맞도록 확장한 방법이다[13-15]. dRSVP는 자원 예약을 기존의 RSVP처럼 경로 설정 시 고정된 값으로 정하는 것이 아니라, 특정 범위 안에서 네트워크 상황에 맞게 유동적으로 변경할 수 있도록 하였다. 이를 위해 *Path, Resv* 메시지 외에 *ResvNotify* 메시지를 새로 정의하고, 각 메시지에 새로운 필드를 추가하였다. 또한 어드미션 컨트롤을 수정하여 대역의 범위 내에서 자원 예약이 가능하도록 하였다. 이러한 동적인 자원 할당이 가능하려면 응용 프로그램도 네트워크 계층과 통신을 통해 QoS 지원방법을 이해하고 있어야 한다. 그러나 dRSVP는 유선망에서의 RSVP의 문제점을 그대로 계승하고 있다. dRSVP는 아웃오브밴드 시그널링 방법을 사용하므로 주기적인 갱신 메시지가 필요하다. 또한 예약한 플로우마다 별도의 상태정보 관리가 필요하므로 이에 따르는 오버헤드는 여전히 존재하게 된다.

3. QoS 프레임웍

본 장에서는 2장에서 지적한 Ad-Hoc Network에서의 QoS 지원을 위한 기존연구들의 문제점을 해결하기 위해 새롭게 제안하는 QFAN(QoS Framework for Ad-Hoc Networks)에 대해 기술한다. 그리고 이를 지원하기 위해 설계된 두 가지 알고리즘에 대해 설명한다. 경량화된 예약 프로토콜인 TiRe(Tiny Reservation) 알고리즘과 적응적인 폐기율 제어 방법으로 ADR(Adaptive Drop Rate) 컨트롤 알고리즘을 설계하여 QFAN 프레임웍에 적용함으로써, 실시간 트래픽의 전송률을 보장하면서 잉여 대역폭에 대해서는 실시간 트래픽과 최선형 트래픽 간에 대역폭 균등 분배를 수행할 수 있다.

3.1 QFAN의 구조 및 동작

Ad-Hoc Network에서 QoS를 지원하기 위한 많은 연구가 진행되고 있다. 특히 실시간 트래픽 또는 특정 트래픽을 보호하기 위한 관점에서의 연구가 활발히 진행되고 있다. 이러한 연구들은 2.3절에서 기술한 것처럼 실시간 트래픽의 보호를 위해 DiffServ 모델을 이용하거나, 시그널링을 통한 자원 예약 방법을 이용하고 있다. 자원 예약 방법을 이용할 경우 정확한 요구 대역폭을 만족시킬 수 있다. 그러나 예약 상황을 유지하기 위해 주기적인 갱신 메시지가 필요하므로 좁은 대역폭을 가지는 Ad-Hoc Network의 대역폭의 낭비를 가져올 수 있으며, 각 플로우의 상태정보를 모두 관리해주어야 하는 문제가 있다. DiffServ 모델을 이용하는 경우는 자원 예약 방법과는 반대로 예약 상태 유지를 위한 오버헤드는 없지만, 실시간 트래픽이 요구하는 정확한 대역폭의 보장이 어렵다는 단점이 있다. 또한 그림 2에서 확인한 것처럼 실시간 트래픽의 보호를 위해 최선형 트래픽의 전송률이 제한을 받게 되는 문제가 발생한다. 그러므로 실시간 트래픽의 전송률을 보장하면서 최선형 트래픽의 전송률 또한 제한을 받지 않고 잉여 대역폭을 공정 분배할 수 있는 새로운 방법이 필요하다. 본 논문에서는 실시간 트래픽의 전송률 보장과 잉여 대역폭의 공정 분배를 위한 새로운 프레임웍을 제안하고, 실제 서비스를 제공할 수 있도록 TiRe 알고리즘과 ADR 컨트롤 알고리즘을 제안하였다.

본 논문에서 제안하는 QFAN은 에지 노드(edge node)에서 각각의 패킷에 마킹을 하고 코어 노드(core node)에서는 마킹된 패킷의 우선순위에 근거해서 서비스를 제공하는 DiffServ 모델을 따르고 있다. 그리고 실시간성을 요구하는 트래픽에 대해서는 최소 대역폭을 만족시키기 위해서 자원 예약 프로토콜을 이용하므로 부분적인 IntServ 모델을 함께 채용하고 있다. 따라서 특정 플로우에 대해서만 자원 예약을 수행하되 자원 예약과 그 유지에 필요로 하는 비용을 최소화시킴으로써 좁은 대역폭을 가지는 Ad-Hoc Network에 적합한 QoS 모델을 제시하였다. Ad-Hoc Network의 특성상 코어와 에지의 경계가 불분명 하므로 모든 노드는 코어 노드나 에지 노드의 역할을 동시에 수행할 수 있어야 한다. 그러므로 소스 노드(source node)가 에지 노드의 역할을 동시에 수행하며, 트래픽을 전달하는 모든 노드는 코어 노드가 된다. 이러한 요구사항을 만족시키기 위해서 네트워크 전체 노드에 동시에 적용될 수 있는 그림 3과 같은 프레임웍을 설계하였다.

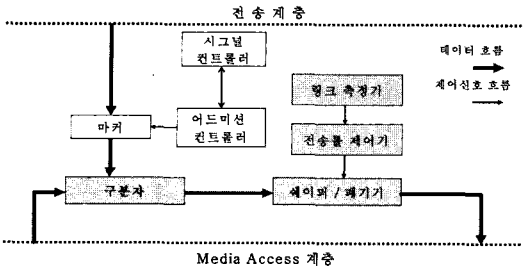


그림 3 QFAN의 구조

그림 3에 나타낸 QFAN의 구조는 노드의 역할에 따라 사용되는 구성요소가 다르다. 에지 노드에서는 자원 예약을 위해 마커(marker)와 시그널 컨트롤러(signaling controller), 어드미션 컨트롤러(admission controller)를 이용한다. 시그널 컨트롤러는 전송할 데이타의 전체 경로에 대한 대역폭의 확인을 위해 사용된다. 시그널링을 통해 대역폭을 확인한 후, 어드미션 컨트롤러는 실시간 트래픽이 요구하는 대역폭과의 비교를 통해 자원 예약 가능 여부를 판단하게 된다. 시그널링을 위해 본 논문에서는 TiRe 알고리즘을 설계하였으며 3.2절에서 자세히 알아본다. 마커는 예약이 된 트래픽에 대해 IN 또는 OUT으로 마킹하는 역할을 한다. 실제 데이타의 전달을 위해 구분자(classifier)와 링크 측정기(link measurer), 전송률 제어기(rate controller), 그리고 쉐이퍼/폐기기를 사용한다. 코어 노드에 패킷이 유입되었을 때 구분자는 실시간 트래픽과 최선형 트래픽을 다르게 처리하기 위해 마킹된 패킷의 종류를 분류한다. 링크 측정기는 코어 노드에 들어오는 트래픽의 유입량을 측정한다. 링크 측정기는 실시간 트래픽과 최선형 트래픽의 유입량을 구분하여 측정한다. 전송률 제어기는 실시간 트래픽과 최선형 트래픽의 전송률의 비를 이용해 패킷 폐기 확률을 결정하여 전송률을 제어한다. 본 논문에서는 전송률 제어를 위해 ADR 컨트롤 알고리즘을 제안하였다. ADR 컨트롤 알고리즘은 3.3절에서 자세히 기술한다. 쉐이퍼/폐기기는 전송률 제어기에서 결정된 폐기율에 따라 실제로 패킷을 폐기하는 역할을 한다.

그림 4는 QFAN을 적용한 시스템 모델의 동작을 나타낸 것이다. 시그널링을 통해 대역폭을 확인하고 어드미션 컨트롤러에 의해 자원 예약이 이루어지면, 에지 노드에서는 실시간 트래픽에 대해 예약된 양만큼의 패킷은 IN으로 나머지 패킷은 OUT으로 마킹을 하여 전송한다. 코어 노드에서는 마킹된 패킷을 확인하여 우선순위를 결정해 패킷을 전달한다. 특히 IN으로 마킹된 실시간 트래픽에 대해서는 가장 높은 우선순위로 서비스한다. 일반적으로 UDP 트래픽은 TCP 트래픽과는 달리 네트워크 상황에 따른 전송률 제어를 수행하지 않으므로 결과적으로 TCP 트래픽에 대한 전송률을 제한하고, UDP 트래픽간의 경쟁으로 네트워크 전체에 악영향을 미치게 된다. 따라서 QFAN은 두 트래픽간의 폐기율을 서로 다르게 적용함으로써 UDP 트래픽과 TCP 트래픽간에 잉여 대역폭을 공정히 분배하도록 하였다.

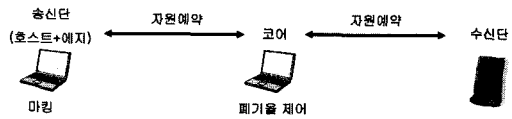


그림 4 시스템 모델의 동작

3.2 경량화 자원 예약 알고리즘

3.2.1 TiRe 알고리즘

TiRe 알고리즘은 실시간 트래픽에 대해 정량적인 대역폭을 보장하기 위한 자원 예약 방법이다. TiRe 알고리즘은 Ad-Hoc Network의 유동적인 토폴로지(topology)의 변화와 좁은 대역폭을 가지는 무선 링크의 특성을 고려하여 기존의 RSVP 메커니즘을 수정, 보완하였다.

RSVP 메커니즘은 각각의 플로우마다 자원예약을 수행하므로 경로상의 모든 노드는 자신을 지나는 모든 플로우의 상태 정보를 관리해야 한다. 또한 플로우 단위의 서비스를 제공해야 하므로 플로우마다 각각 독립된 큐를 관리해야 하며, 예약 상태의 유지를 위해서 일정 주기로 계속적인 갱신 메시지를 주고받아야 한다. 따라서 좁은 대역폭을 가지는 무선링크에서 전달해야할 데이타와 시그널 메시지와의 경합이 발생하여 데이타의 전송률 저하가 발생하게 된다. 이러한 문제점을 해결하기 위해서 TiRe 알고리즘은 코어 노드와 에지 노드의 역할을 분리하고, 마킹을 통해 관리해야 할 정보를 최소화한다. 그리고 패킷 간의 지연시간에 근거한 타이머를 이용함으로써 주기적인 갱신 메시지를 제거하였다.

3.2.2 컨트롤 메시지

TiRe 알고리즘은 전송단에서 어드미션 컨트롤(source-based admission control)을 수행하는데, 이를 위해 컨트롤 메시지인 프로브(probe) 메시지를 이용한다. 프로브 메시지는 네트워크 상황을 확인하고, 자원 예약을 하며, 네트워크 상황에 따라 채널 상태를 전송단에 보고하기 위한 목적으로 사용된다. 프로브 메시지는 기능에 따라 *path probe, resv probe, readmission probe*로 정의하였다. 프로브 메시지는 그림 5와 같은 필드를 가지며 타입 필드 값에 따라 프로브 메시지의 종류를 구분할 수 있다.

Path probe 메시지는 초기 연결 설정 시 어드미션 컨트롤을 위해, 경로 상의 병목대역폭을 측정하기 위한 컨트롤 신호로 사용된다. *Resv probe* 메시지는 실제

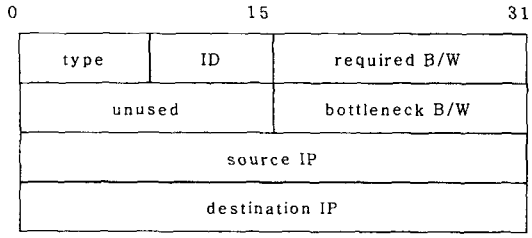


그림 5 프로브 메시지

자원을 예약하기 위한 메시지이다. *Path probe* 메시지가 수신단까지 도착했을 때 다시 전송단에게 같은 경로를 통해 *resv probe* 메시지를 전달하여 경로상에 자원 예약이 이루어지게 된다. *Readmission probe* 메시지는 실시간 트래픽의 요구 대역폭에 비해 병목 대역폭이 좁거나, 채널 상황이 악화되어 패킷 전달이 어려울 경우 자원을 회수하고 이를 전송단에 알리기 위한 메시지이다. 그러므로 송신 단말이 *readmission probe* 메시지를 수신했을 경우, 두 가지 방법으로 동작하게 된다. 먼저 *readmission probe* 메시지에 대한 플로우를 전송하는 어플리케이션이 적응적으로 요구 대역폭을 바꿀 수 있는 경우, 다시 *path probe* 메시지를 전송하여 새로운 대역폭을 할당받도록 한다. 만일 *readmission probe* 메시지에 대한 플로우를 전송하는 어플리케이션이 적응적으로 요구 대역폭을 바꿀 수 없다면, 전송을 중단하거나 별도의 *probe* 메시지 전송없이 최선형 서비스로 패킷을 전달하게 된다.

3.3.3 TiRe 알고리즘의 동작

TiRe 알고리즘의 동작은 그림 6과 같다. 전송단에서 실시간 트래픽의 최소 요구 대역폭을 *path probe* 메시지의 *required B/W* 필드에 마킹을 하여 전송한다. *Path probe* 메시지는 두 개를 전송하여 경로상의 병목 대역폭을 확인하고 메시지 사이의 패킷간 지연시간을 이용해서 타이머를 세팅한다. 경로상의 코어 노드에서는 *required B/W* 필드 값과 코어 노드의 가용 대역폭과 비교하여 목적지까지 자원 예약 여부를 확인한다. *Path probe* 메시지에 정의된 요구 대역폭이 전체 경로에 걸쳐 만족이 되면 목적지에서는 *resv probe* 메시지를 전송하여 경로 상에 자원 예약이 이루어진다.

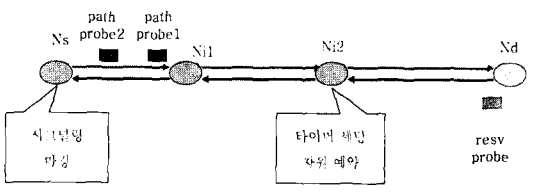


그림 6 TiRe 알고리즘의 동작

만일 예약이 불가능할 경우 *readmission probe* 메시지를 통해 자원 예약이 실패하였음을 알린다. *Path probe* 메시지가 수신단까지 도착했을 때 수신단에서는 *resv probe* 메시지는 같은 경로를 통해 전송단에게 전달하며, 이 과정에서 실제 자원 예약이 이루어진다. 트래픽의 전송 중 채널 상태에 이상이 생길 경우, 현재 할당된 자원을 회수 하고 자원이 회수된 트래픽의 전송단에게 *readmission probe* 메시지를 이용하여 상황을 보고한다. 성공적으로 *resv probe* 메시지를 받은 전송단은 실시간 트래픽에 대해서 예약된 대역만큼의 패킷은 IN으로 마킹하고 나머지 패킷에 대해서는 OUT으로 마킹한다. 코어 노드에서는 IN으로 마킹된 패킷을 최우선으로 서비스하므로 RSVP에서와 같은 복잡한 상태정보의 관리가 필요 없다.

TiRe 알고리즘은 주기적인 갱신 메시지를 제거하기 위해 타이머를 동작시켜 그 시간 안에 새로운 패킷의 유입이 없을 경우 자원 예약 상태를 해지하는 soft-state 예약 방법을 이용한다. 상태정보 타이머는 초기 자원 예약 시 두 개의 프로브 메시지 사이의 패킷간 지연 시간을 이용해서 설정한다. RSVP의 경우 주기적인 메시지를 경로상의 모든 노드가 주고받아야 하며, 채널의 상태가 바뀌거나 네트워크 토폴로지(topology)가 변하는 경우, 다음 갱신 메시지를 받을 때 까지 상황 변화에 대한 대처가 미흡한 단점이 있다. 그러나 TiRe 알고리즘은 상태정보 타이머를 사용함으로써, 네트워크 오버헤드로 작용하는 갱신 메시지를 제거하였다. 그리고 IP 헤더에 채널 노이즈 상태나 연결단절 상태를 나타낼 수 있는 CN(channel noisy) 필드를 정의하고, 이를 통해 채널 상황을 보고 하므로 네트워크의 변화에도 유동적으로 반응할 수 있다.

TiRe 알고리즘의 soft-state 예약 알고리즘은 그림 7과 같다. 코어 노드는 CN 비트가 마킹되었을 경우 정상적인 서비스가 불가능한 것으로 간주하고 예약되었던 자원을 회수한다. 만일 CN 비트는 마킹되어 있지 않지만, 타임아웃의 시간이 지나서 패킷이 전달되었을 경우 채널 상태가 좋지 않은 것으로 간주하고 패킷에 CN 비트를 마킹하고 자원을 회수한다. TiRe 알고리즘의 자원 예약은 각 노드마다 토큰 버킷(token bucket) 방식으로 이루어 진다. 즉 토큰 버킷의 사이즈와 토큰이 생성되는 속도에 의해 예약되는 자원의 양이 결정되므로 자원의 회수는 할당된 플로우에 대한 토큰과 버킷을 제거하는 방식으로 수행된다.

D_{rt} 값은 실시간 트래픽의 패킷간 지연 시간으로, D_{rt} 값이 상태정보 타이머보다 큰 경우 네트워크 상황이 악화되는 것으로 판단하고 *hit_count*를 증가시킨다. 만일 *hit_count*가 연속적으로 3회까지 증가하면 CN 비트를

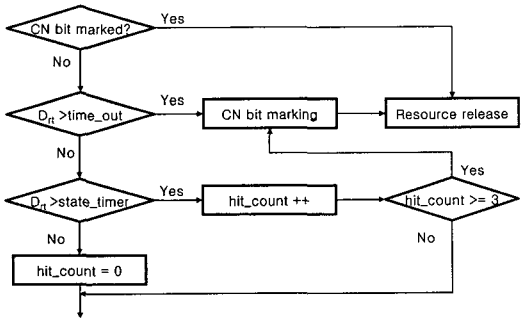


그림 7 Soft-State 예약 알고리즘

마킹하고 자원을 회수하므로, 일시적인 지연 시간의 증가에 의해 예약이 해지되는 상황에 대처할 수 있다. 상태정보 타이머와 타임아웃 값의 설정은 식 (1), (2)와 같다. D_{probe}는 프로브 메시지간의 패킷간 지연시간, 즉 *path probe1*과 *path probe2* 간의 도착 지연시간을 의미하며 경로상의 매 노드가 각각 timer를 세팅하게 된다. a는 *time_out*값을 결정하는 인자이다.

이때, *time_out* 값은 다음과 같은 경우에 갱신될 수 있다. 먼저 새로운 노드가 삽입되거나 기존 노드가 사라져서 망의 토폴로지가 변할 경우, 라우팅 패스를 다시 설정하므로 *path probe* 메시지도 재전송 되므로 *time_out* 값이 갱신된다. 만일 토폴로지의 변화 없이 노드가 이동할 경우, 거리또는 신호 감쇄에 의해 *state_timer*를 위반하여 *hit_count*가 증가하므로 결과적으로 다시 *path probe* 메시지에 의해 *time_out* 값이 갱신된다.

$$state\_timer_i = D_{probe} + 2 \times \left(\frac{packet\_length}{reserved\_rate}\right), \quad (1)$$

$$time\_out_i = a \times state\_timer_i \quad (2)$$

$\frac{packet_length}{reserved_rate}$는 하나의 패킷이 서비스 받는데 걸리는 시간이며, D_{probe}는 두 패킷간의 지연시간이므로 *state_timer*는 2개의 패킷이 도착해서 서비스를 받는데까지 걸리는 총 시간이라 할 수 있다. 그러므로 *state_timer*의 값이 계속적으로 증가하는 것은 네트워크 상태가 악화된다는 것을 의미한다. 따라서 *state_timer*의 값을 연속적으로 위반하는 것을 감지하고, *time_out*을 발생시킨다. 즉 *time_out*은 네트워크 상태가 악화되어 새롭게 자원 예약을 하거나 기존의 자원 예약을 해지하도록 하며, 이로 통해 조기에 네트워크 상황에 대처하여 자원의 낭비를 막을 수 있다. 그리고 *time_out*값은 측정된 *state_timer*값과 미리 결정된 a값에 의해 결정된다. a값이 감소함에 따라 *time_out*값이 작아지므로 지연시간에 민감하게 반응하여 작은 패킷 지연에도 자원 예약을 해지하게 된다. 반대로 a값이 커질수록 *time_out*값이 커지므로 지연시간 증가에 따른 빠른 자원 예약 해지를 완화시킬 수 있다.

3.3 잉여 대역폭 공정 분배 알고리즘

3.3.1 ADR 컨트롤 알고리즘

ADR 컨트롤 알고리즘은 공격적으로 대역을 점유하는 UDP 트래픽의 전송률을 제한함으로써 상대적으로 TCP 트래픽의 전송률 저하를 막고, 잉여 대역폭을 실시간 트래픽과 최선형 트래픽사이에 공정하게 분배한다. 실시간성을 요구하는 대부분의 트래픽은 UDP 연결 특성을 가지게 된다. 본 논문에서도 실시간 트래픽은 모두 UDP 연결로 가정하였다. 실시간 트래픽이 대역을 점유함에 따라 TCP 연결을 가지는 최선형 트래픽의 전송률을 제한하므로 최선형 트래픽이 정상적으로 서비스 받지 못하는 상황이 발생할 수 있다. 본 논문에서 제안한 ADR 컨트롤 알고리즘은 실시간 트래픽의 요구 대역폭을 만족시켜 주면서 잉여 대역폭에 대해서는 실시간 트래픽과 최선형 트래픽간의 공정 분배를 수행한다. ADR 컨트롤 알고리즘은 실시간 트래픽의 과다한 대역폭 점유를 방지하기 위해서 폐기율을 조정하는 방법을 사용하는데, 실시간 트래픽의 폐기 확률은 최선형 트래픽의 폐기 확률에 따라 적응적으로 변하게 된다.

3.3.2 ADR 컨트롤 알고리즘의 동작

ADR 컨트롤 알고리즘은 코어 노드에서 OUT으로 마킹된 실시간 트래픽의 패킷 폐기율을 조정함으로써 최선형 트래픽의 성능 저하 현상을 방지하고, 잉여 대역폭에 대해서 OUT으로 마킹된 실시간 트래픽과 최선형 트래픽 간에 공정 분배를 하도록 설계되었다. 이를 위해서 OUT으로 마킹된 트래픽과 최선형 트래픽간의 차등화된 폐기 절차(drop precedence)를 적용하고 있다[16]. 전체 링크 대역 중 이미 자원이 예약된 대역을 제외한 나머지 잉여 대역폭 BW_{avail},을 식 (3)과 같이 정리할 수 있다. BW_{total}과 BW_{rsv}는 각각 링크의 대역폭과 예약된 대역폭을 의미한다.

$$BW_{avail} = BW_{total} - BW_{rsv}. \quad (3)$$

R_{out}, R_{be}는 각각 OUT으로 마킹된 트래픽과 최선형 트래픽의 전송률이며 P_{out}, P_{be}는 각각 OUT으로 마킹된 트래픽과 최선형 트래픽의 폐기 확률 값이다. OUT으로 마킹된 트래픽의 타깃 전송률 T_{out}과 최선형 트래픽의 타깃 전송률 T_{be}는 식 (4), (5)와 같이 각각 나타낼 수 있으며,

$$T_{be} = \left(\frac{\sum R_{out}}{\left(\sum R_{be} + \sum R_{out}\right)}\right) \times BW_{avail}, \quad (4)$$

$$T_{out} = \left(\frac{\sum R_{be}}{\left(\sum R_{out} + \sum R_{be}\right)}\right) \times BW_{avail}. \quad (5)$$

비례식을 통해 식 (6)으로 유도할 수 있다.

$$T_{out} = \left(\frac{\sum R_{be}}{\sum R_{out}} \right) \times T_{be} \quad (6)$$

폐기 확률은 전송률의 제곱에 반비례하므로 식 (7)로 정리하면, 최선형 트래픽의 폐기 확률에 대한 적응적인 실시간 트래픽의 폐기 확률을 구할 수 있다[17].

$$P_{out} = \left(\frac{\sum R_{out}}{\sum R_{be}} \right)^2 \times P_{be} \quad (7)$$

최선형 트래픽을 위한 버퍼관리 기법으로 RED (Random Early Detection)을 적용하면, 두 개의 버퍼 임계 값과 최대 폐기 확률 값에 의해 버퍼점유에 따른 폐기 확률 값을 구할 수 있다[18]. 이에 근거해 실시간 트래픽 중 OUT으로 마킹된 패킷의 폐기 확률 값을 구하게 된다. 두 트래픽의 폐기 확률 곡선은 그림 8과 같이 나타낼 수 있다.

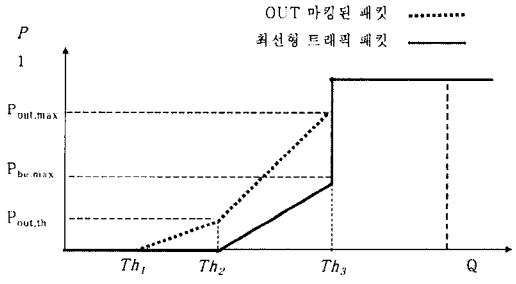


그림 8 폐기 확률 곡선

when

$$P_{be} \begin{cases} 0 & Q < Th_2 \\ P_{be,max} * Q - Th_2 * P_{be,max} / (Th_2 - Th_1) & Th_2 < Q < Th_3 \\ 1 & Q > Th_3 \end{cases}$$

Estimated every T seconds period

If ($Q < Th_1$)

$P_{out} = 0$

Else if ($Th_1 < Q < Th_2$)

$P_{out} = P_{out,th} * Q - Th_1 * P_{out,th} / (Th_2 - Th_1)$

Else if ($Th_2 < Q < Th_3$)

$P_{out} = (\Sigma R_{out} / \Sigma R_{be})^2 * P_{be}$

Else

$P_{out} = 1$

그림 9 패킷 폐기 확률 계산 알고리즘

그림 8처럼 OUT 마킹 패킷의 폐기 확률은 최선형 트래픽의 폐기 확률에 근거해서 결정할 수 있다. 즉 최선형 트래픽의 폐기 확률에 따라 OUT 마킹 패킷의 폐기 확률이 적응적으로 변할 수 있다. 그림에서 Q는 패킷의 큐 점유율이며, 점유율에 따라 패킷의 폐기 확률을 결정하기 위해서 Th_1, Th_2, Th_3의 세 개의 임계 값을 설정하고, 세 부분으로 영역을 나누었다. 먼서 Th_1과 Th_2 사이의 구간에서는 OUT 마킹된 패킷을 우선적으로 폐기한다. Th_2와 Th_3 구간에서는 식 (7)에 의해 최선형 패킷의 폐기 확률에 근거한 OUT 마킹 패킷의 폐기 확률을 계산할 수 있다. 마지막으로 사이즈가 Th_3 이상일 경우 새로 들어오는 모든 패킷을 폐기한다. 이에 따라 실제 그림 9와 같이 폐기 확률을 적용한다.

4. 실험 및 성능평가

4.1 실험 환경

TiRe(Tiny Reservation) 메커니즘과 ADR(Adaptive Drop Rate) 컨트롤 알고리즘의 성능 평가를 위해 본 장에서는 LBNL(Lawrence Berkeley National Laboratory)의 ns-2(network simulator-2)를 사용하여 시뮬레이션 하였다[19].

표 1 실험 환경

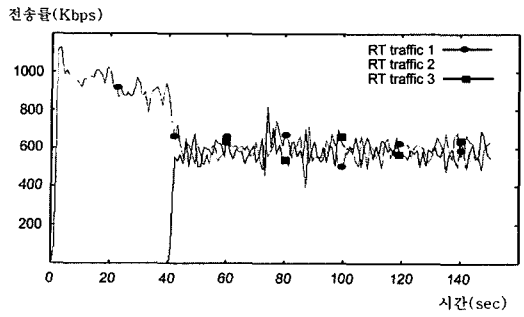
항목	값	항목	값
라우팅 프로토콜	AODV	전송 매체	2Mbps 무선 랜
네트워크 구역	670m×670m	노드 개수	20 개
무선 전송 범위	200m	시뮬레이션 시간	150 초
패킷 크기	1024 bytes		

실험을 위한 Ad-Hoc Network의 환경은 표 1과 같이 설정하였다. 본 실험은 670m × 670m 크기의 중·소 규모의 네트워크 구역을 가정하였다. 라우팅 프로토콜은 AODV를 사용하였으며 네트워크 노드의 개수는 20개로 네트워크 구역 내에 무작위로 분산되어 있다[20]. 전송 매체는 2Mbps의 대역폭을 가지는 무선 랜이며 200m의 전송 범위를 가진다. 20개의 노드 중 무작위로 노드를 선출하여 TCP 연결을 가지는 최선형 트래픽과 UDP 연결을 가지는 실시간 트래픽을 생성하였다. 실시간 트래픽은 1Mbps 전송률을 가지는 CBR(constant bit rate) 트래픽이며, 500Kbps의 최소 대역폭을 요구한다.

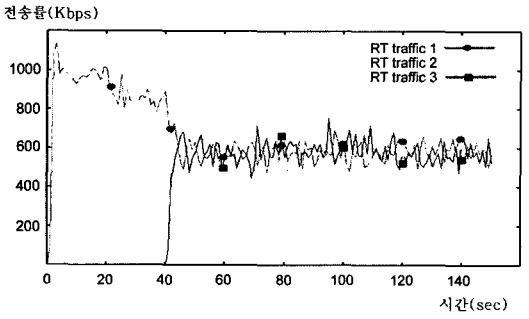
4.2 실시간 트래픽의 요구 대역폭 보장

본 절에서는 ns-2 시뮬레이터에 구현한 TiRe 알고리즘을 이용하여 실시간 트래픽의 요구 대역폭 보장에 관한 실험 결과를 기술한다.

SWAN을 이용한 그림 10의 결과처럼 것처럼 Diff-Serv 메커니즘을 적용한 QoS 모델의 경우, 과다한 실시간 트래픽이 유입되면 최소 요구 대역폭을 만족시키지 못하여 전체 트래픽의 성능이 모두 저하되는 것을 확인할 수 있었다. 실시간 트래픽의 전송률 보장을 확인하기 위해서 동일한 조건에서 DiffServ 모델을 적용한 SWAN과 TiRe의 비교 실험을 하였다. 실험 결과 그림



(a) SWAN의 전송률

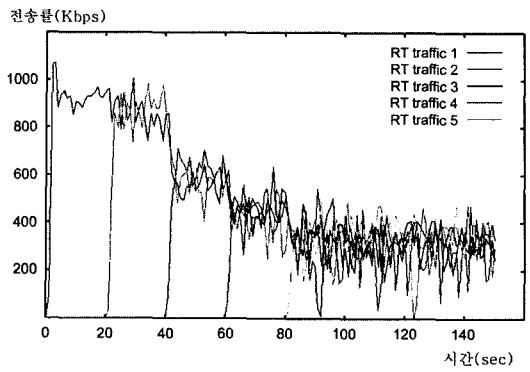


(b) TiRe의 전송률

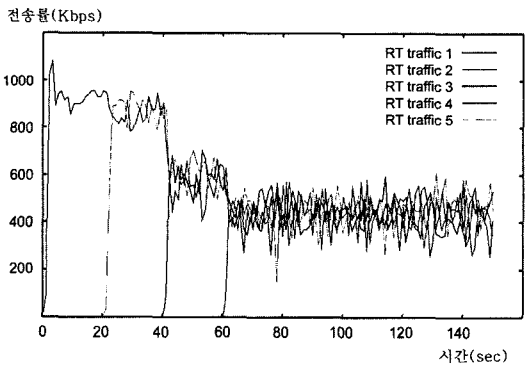
그림 10 실시간 트래픽에 대한 SWAN과 TiRe의 전송률 비교

10과 같이 전체 링크 대역폭은 세 개의 실시간 트래픽의 최소 요구 대역폭을 모두 만족시킬 수 있으므로 SWAN과 TiRe 모두 비슷한 결과를 얻을 수 있었다.

같은 특성을 가지는 실시간 트래픽이 2개가 더 유입될 경우, SWAN은 새로운 실시간 트래픽의 서비스를 위해 기존의 실시간 트래픽의 전송률을 제한하게 된다. 따라서 그림 11(a)와 같이 5개의 트래픽 모두 500Kbps의 최소 요구 대역폭을 만족시키지 못하게 된다. TiRe는 연결 설정 시 가용 대역폭을 확인하고 최소 요구 대역폭을 만족시키지 못할 경우 연결을 거부하므로, 새로운 트래픽의 유입에 의해 기존 트래픽의 성능이 저하되는 것을 막을 수 있다. 즉 2Mbps의 링크 대역폭을 이미 4개의 실시간 트래픽이 점유하고 있으므로, 5번째 실시간 트래픽은 연결이 거부되어 실제 4개의 트래픽만



(a) SWAN의 전송률



(b) TiRe의 전송률

그림 11 과다한 실시간 트래픽에 대한 SWAN과 TiRe의 전송률 비교

존재하게 되지만, 4개의 실시간 트래픽에 대해서는 그림 11(b)와 같이 전송률이 보장되는 것을 확인할 수 있다.

TiRe는 대역폭의 보장을 위해 자원 예약 방법을 사용하지만, 컨트롤 메시지를 제거하여 기존의 자원 예약 방법에서 발생하는 오버헤드를 줄이고 있다. 표 2에서 DiffServ 모델을 적용한 SWAN과의 비교를 통해 적은 오버헤드로 서비스를 제공하는 것을 확인할 수 있다.

2개의 실시간 트래픽을 통한 실험에서는 두 방법 모두 비슷한 성능을 보이고 있으며, SWAN에 비해 TiRe의 시그널 패킷의 비율이 조금 높은 것을 확인할 수 있다. 그러나 전체 패킷의 양에 비해 데이타 패킷 대 시그

표 2 SWAN과 TiRe의 데이타 패킷 대 시그널 패킷의 비율 비교

2개의 실시간 트래픽			5개의 실시간 트래픽		
	SWAN	TiRe		SWAN	TiRe
총전송 패킷	148297 개	148615 개	총전송 패킷	228491 개	209313 개
데이타 패킷	95804 개	95797 개	데이타 패킷	99509 개	93050 개
시그널 패킷	244 개	445 개	시그널 패킷	528 개	844 개
비율	0.26%	0.46%	비율	0.53%	0.9%

널 패킷의 비율은 1% 미만의 작은 값이므로, DiffServ 모델을 적용한 SWAN에 비해 큰 오버헤드의 증가 없이 요구 대역폭을 만족시킬 수 있었다. 5개의 실시간 트래픽을 발생시켰을 때, 2Mbps의 링크 대역폭은 5개의 실시간 트래픽의 최소 요구 대역폭을 만족 시킬 수 없다. 그림 11(b)와 같이 TiRe는 4개의 트래픽만 실제 전송되므로 총 전송 패킷의 개수는 SWAN에 비해 적다. 그러나 총 전송 패킷의 개수에 비해 폐기되는 패킷의 개수는 상대적으로 적어, 실제 전달된 데이타 패킷의 개수는 SWAN과 비슷한 양을 보인다. 그러므로 폐기된 패킷 개수에 대한 전달된 데이타 패킷 개수의 비율을 보면 SWAN은 약 77%이고 TiRe는 약 80%로 TiRe가 약간 좋은 성능을 보이는 것을 확인할 수 있다.

4.3 실시간 트래픽과 최선형 트래픽의 대역폭 공정 분배 성능 검증

본 절에서는 잉여 대역폭에 대한 실시간 트래픽과 최선형 트래픽간의 공정 분배에 관한 실험을 하였다. 2.3.2절에서 기술한 것처럼 최선형 트래픽은 실시간 트래픽의 영향으로 전송률이 저하되는 현상을 보이는데, 이를 해결하기 위해서 ADR 컨트롤 알고리즘을 구현하였다. 링크의 모든 대역을 실시간 트래픽이 예약하는 것을 방지하기 위해 전체 링크 대역 2Mbps 중, 예약 가능 대역폭을 1.5Mbps로 제한하였다. 표 3과 같이 트래픽을 생성시켜 실험을 하였다.

실험 1은 두 개의 500Kbps의 최소 요구 대역폭을 가지는 실시간 트래픽과 두 개의 최선형 트래픽을 만들어 잉여 대역폭에 대한 공정 분배 정도를 측정하였다. 그림 12에서 두 개의 실시간 트래픽은 최소 요구 대역폭 500Kbps를 만족하고, 남는 대역폭을 최선형 트래픽과 나누어 점유하고 있는 것을 확인할 수 있다. 실시간 트래픽은 최소 요구 대역폭만큼은 IN으로 마킹되어 서비스 되고, 남은 대역에 대해서는 OUT으로 마킹되어 최선형 트래픽과 경쟁하고 있다. ADR 컨트롤 알고리즘은 최선형 트래픽의 폐기 확률에 의해 OUT으로 마킹된 패킷의 폐기 확률을 결정하므로 공격적으로 대역을 점

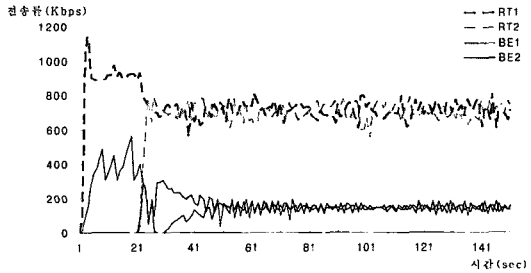


그림 12 두 개의 실시간 트래픽과 두 개의 최선형 트래픽간의 공정성 비교

유하려 하는 실시간 트래픽의 전송률을 제한할 수 있다.

표 4는 실험 결과를 수치화하여 나타낸 것으로, 각 트래픽의 평균 전송률을 구하고 타깃 전송률을 제외한 잉여 대역의 점유율을 구하였다. 실시간 트래픽과 최선형 트래픽이 각각 점유하고 있는 총 잉여 대역폭을 산출한 결과 각각 369,757bps와 319,539bps의 값을 얻었다. 두 개의 실시간 트래픽의 최소 요구 대역폭의 합이 1Mbps이므로 남은 1Mbps는 잉여 대역폭이 된다. 표 4의 총 잉여 대역 점유율을 볼 때, 전체 잉여 대역을 모두 사용하지는 못했지만 두 트래픽이 잉여 대역폭을 공정히 분배하는 것을 확인할 수 있다.

실험 2는 두 개의 실시간 트래픽과 다섯 개의 최선형 트래픽간의 공정성 실험을 하였다. 두 개의 실시간 트래픽은 500Kbps의 최소 요구 대역폭을 가지고 있으며 최선형 트래픽은 20초마다 각각 생성시켰다. 실험 결과를 표 5와 같이 정리하였다. TiRe 알고리즘을 통해 자원 예약을 한 두 개의 실시간 트래픽은 최소 대역폭 500 Kbps를 모두 만족시키고 있다. UDP연결을 가지는 실시간 트래픽이 대역을 점유함에 따라 TCP 연결을 가지는 최선형 트래픽의 전송률은 저하되어 정상적인 서비스가 어려워 질 수 있다.

그러나 ADR 컨트롤 알고리즘은 OUT으로 마킹된 패킷의 폐기 확률을 최선형 트래픽에 비해 급격하게 증가시키므로, 실시간 트래픽의 전송률을 억제할 수 있다.

표 3 실험 시나리오

실험	실험 1	실험 2	실험 3
트래픽 종류	실시간 트래픽 2개 최선형 트래픽 2개	실시간 트래픽 2개 최선형 트래픽 5개	실시간 트래픽 4개 최선형 트래픽 3개

표 4 두 개의 실시간 트래픽과 두 개의 최선형 트래픽간의 공정성 비교

단위(bps)	실시간 트래픽		최선형 트래픽	
트래픽	RT1	RT2	BE1	BE2
평균 전송률	684,480	685,277	154,009	165,530
최소 전송률	500,000	500,000		
잉여 대역 점유율	184,480	185,277	154,009	165,530
총 잉여 대역 점유율	369,757		319,539	

표 5 두 개의 실시간 트래픽과 다섯 개의 최선형 트래픽간의 공정성 비교

단위(bps)	실시간 트래픽		최선형 트래픽				
트래픽	RT1	RT2	BE1	BE2	BE3	BE4	BE5
평균 전송률	694,283	693,613	58,200	60,467	52,638	65,220	86,469
최소 전송률	500,000	500,000					
잉여대역점유율	194,283	193,613	58,200	60,467	52,638	65,220	86,469
총 잉여 대역 점유율	387,896		322,994				

즉 식 (7)과 같이 OUT으로 마킹된 패킷의 폐기 확률은 두 트래픽간의 전송률에 의해 결정되므로, 실시간 트래픽이 증가할 경우 OUT으로 마킹된 패킷의 폐기 확률이 더욱 높아진다. 결과적으로 두 트래픽간은 비슷한 대역을 점유하여, 대역폭 공정 분배를 수행할 수 있다.

실험 3은 네 개의 실시간 트래픽과 세 개의 최선형 트래픽을 이용해서 실험을 하였다. 자원 예약 가능 대역폭이 1.5Mbps이므로 그림 13에서 볼 수 있듯이 TiRe 알고리즘에 의해 네 번째 유입되어야 할 실시간 트래픽은 예약 가능 대역폭이 부족하므로 어드미션에 실패하여 연결 거부당했다. 결과적으로 세 개의 실시간 트래픽과 세 개의 최선형 트래픽이 경쟁하는 상황을 보이고 있다.

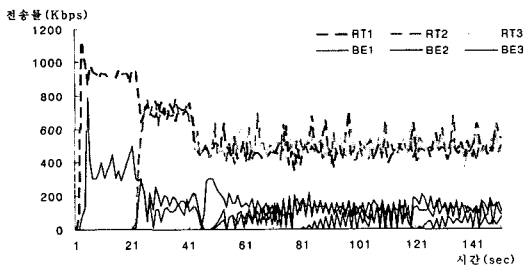


그림 13 네 개의 실시간 트래픽과 세 개의 최선형 트래픽간의 공정성 실험

표 6의 전송률 결과를 보면 잉여 대역에 대한 분배가 최선형 트래픽에 치우쳐 있는 것을 확인할 수 있다. 이것은 ADR 컨트롤 알고리즘에 의해 실시간 트래픽의 패킷 폐기가 상대적으로 많이 일어났기 때문이다. ADR 컨트롤 알고리즘은 식 (7)과 같이 유입되는 트래픽의 전송률에 따라 실시간 트래픽의 폐기율을 결정하여 전송률을 제한하고 있다. 그러므로 유입되는 실시간 트래픽의 패킷이 많아지게 되면, 상대적으로 실시간 트래픽에 대해 더욱 높은 값으로 폐기 확률이 적용된다. 따라서 실제 잉여 대역폭의 대부분을 최선형 트래픽이 차지하게 되었다. 실험 조건에서, 실시간 트래픽이 모든 링크 대역폭을 예약하여 최선형 트래픽이 서비스 되지 못하는 상황을 막기 위하여 예약 가능 대역폭을 1.5Mbps로 한정하였다. 실시간 트래픽이 예약 가능 대역폭을 모두 점유하고 있으므로, 대부분의 잉여 대역폭을 최선형 트래픽이 점유함으로써 최선형 트래픽의 성능 저하를 막을 수 있다.

5. 결론 및 향후 과제

Ad-Hoc Network에서 QoS 지원을 위한 연구가 활발히 진행되고 있다. 특히 특정 트래픽의 대역폭을 보장하기 위한 연구는 앞으로의 멀티미디어 컨텐츠나 화상회의와 같은 응용을 위해 매우 중요하다. 본 논문에서는 Ad-Hoc Network에서 QoS를 지원하기 위해 고려할 점들을 살펴보았다. 그리고 QoS를 지원하기 위해 선행된 기존의 연구들에 대해 살펴보고, 이 들의 문제점에 대해 기술하였다. DiffServ 모델을 적용한 기존의 연구들은 과도한 트래픽의 유입 시 실시간 트래픽에 대하여 정량적인 대역폭을 보장할 수 없는 문제가 있었다. 자원 예약 프로토콜을 이용하는 방법은 요구하는 대역폭을 보장할 수 있지만 컨트롤 신호에 의한 오버헤드가 발생하므로 좁은 대역폭을 가지는 무선 환경에 적용하기 어렵다.

본 논문에서는 특정 트래픽의 대역폭 보장을 위해서 새로운 QoS 프레임웍을 설계하였다. 특히 좁은 대역폭을 가지는 Ad-Hoc Network의 특성을 고려하여 경량화된 QoS 지원 방법을 제시하였다. 새로운 QoS 프레임

표 6 네 개의 실시간 트래픽과 세 개의 최선형 트래픽간의 공정성 비교

단위(bps)	실시간 트래픽			최선형 트래픽		
트래픽	RT1	RT2	RT3	BE1	BE2	BE3
평균 전송률	503,204	510,218	501,690	105,792	96,485	84,903
최소 전송률	500,000	500,000	500,000			
잉여 대역 점유율	3,204	10,218	1690	105,792	96,485	84,903
총 잉여 대역 점유율	15,112			287,180		

웍을 적용하여 실시간 트래픽의 전송률을 보장하고, 실시간 트래픽과 최선형 트래픽간의 공정성 문제를 해결하였다. 제시된 QoS 프레임웍을 실제 적용하기 위한 방법으로 TiRe(Tiny Reservation) 알고리즘을 제안하였다. TiRe 알고리즘은 패킷간 지연 시간을 이용해서 갱신 메시지 없이도 자원 예약 상황을 유지하면서 실시간 트래픽의 최소 요구 대역폭을 만족시킬 수 있도록 설계되었다. SWAN(Stateless Wireless Ad-hoc Network)과의 비교 실험을 통해 추가적인 오버헤드를 줄이면서 최소 요구 대역폭을 만족시키는 것을 확인할 수 있었다. 그리고 실시간 트래픽에 의해서 최선형 트래픽의 전송률이 제한되는 것을 막기 위해 ADR(Adaptive Drop Rate) 컨트롤 알고리즘을 제안하였다. ADR 컨트롤 알고리즘은 실시간 트래픽의 폐기 확률을 최선형 트래픽에 적응적으로 조정함으로써 잉여 대역폭에 대해서 공정 분배를 수행하도록 하였다. 실시간 트래픽과 최선형 트래픽의 경쟁 실험을 통해 잉여 대역폭을 공정 분배하는 것을 확인할 수 있었다.

향후 연구 과제로는 다양한 요구 조건을 가지는 트래픽을 만족시킬 수 있도록 세분화된 차등화 서비스가 필요할 것이다. 그리고 실시간 트래픽의 중요한 특성인 대역폭 보장 뿐 아니라 지연 시간이나 지연 시간의 편차와 같은 다양한 지표에 의한 서비스 차등화에 대한 연구도 함께 진행되어야 할 것이다.

참 고 문 헌

[1] S. Da and M. Chatterjee and N. K. Kakani, "QoS Provisioning in Wireless Multimedia Networks," *Wireless Communications and Networking Conference*, October 1999.

[2] M. S. Corson, "Issues in Supporting Quality of Service in Mobile Ad Hoc Networks," *Workshop on Quality of Service '97*, May 1997.

[3] Mobile Ad-Hoc Networks Discussion Archive-Data, http://www1.ietf.org/mail-archive/working-groups/manet/current/maillist.html

[4] R. Braden, and D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," *Intrnet RFC 1633*, June 1994.

[5] S. Blake and D. Black and M. Carlson and E. Davies and Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," *RFC 2475*, December 1998.

[6] X. Xiao, and L. M. Ni, "Internet QoS: A Big Picture," *IEEE Network Magazine*, March/April 1999.

[7] D. H. Cansever and A. H. Levesque, "A Framework for QoS Support in Mobile Ad-Hoc Networks," *Internet Draft, draft-ietf-manet-qos-frame work-00.txt*, February 1999.

[8] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet," *IEEE/ACM Transactions on Networking*, 1999.

[9] H. XIAO, "A Flexible Quality of Service Model for Mobile Ad-Hoc Networks," *IEEE VTC2000*, Spring 2000.

[10] G. Ahn and A. T. Campbell, and A. Veres and S. Li-Hsiang, "SWAN: Service Differentiation in Stateless Wireless Ad-Hoc Networks," *IEEE INFOCOM'2002*, June 2002.

[11] G. Ahn and A. T. Campbell, Andras Veres and Li-Hsiang Sun, "Supporting Service Differentiation for Real-Time and Best Effort Traffic in Stateless Wireless Ad-Hoc Networks (SWAN)," *IEEE Transactions on Mobile Computing*, September 2002.

[12] S. Lee, "INSIGNIA: In-Band Signaling Support for QoS in Mobile Ad-Hoc Networks," *International Workshop on MoMuC'98*, 1998.

[13] M. Mirhakkak and N. Schult and D. Thomson, "Dynamic Quality-of-Service for Mobile Ad-Hoc Networks," *MobiHoc 2000*, 2000.

[14] R, Braden and L. Zhang and S. Berson and S. Herzog and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification," *IETF RFC 2205*, September 1997.

[15] R. Braden and L. Zhang and S. Berson and S. Herzog and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Message Processing Rules," *IETF RFC 2209*, September 1997.

[16] B. Nandy and N. Seddign and P. Pieda and J. Ethridge, "Intelligent Traffic Conditioners for Assured Forwarding Based Differentiated Services Networks," IFIP High Performance Networking. June 2000.

[17] S. Floyd and M. Handley, J. Pdhye, "A Comparison of Equation-Based and AIMD Congestion Conrol," ACM SIGCOMM, October 2000.

[18] S. Floyd and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, August 1993.

[19] The Network Simulator ns-2, http://www.isi.edu/nanam/ns/

[20] C. E. Perkins and E. M. Royer and S. R. Das, "Ad-Hoc On-Demand Distance Vector(AODV) Routing," *IEEE Workshop on Mobile Computing Systems and Applications*, February 1999.

김 준 형

2002년 광운대학교 전자통신공학과 학사

2004년 광운대학교 전자통신공학과 석사

모 상 덕

1998년 광운대학교 전자통신공학과 학사
2000년 광운대학교 전자통신공학과 석사
2000년~현재 광운대학교 전자통신공학과 박사 과정

정 광 수

1981년 한양대학교 전자공학과 학사 1983년 한국과학기술원 전기 및 전자공학과 석사. 1991년 미국 University of Florida 전기공학과 박사(컴퓨터공학전공). 1983~1993년 한국전자통신연구원 선임연구원 1991~1992년 한국과학기술원 대우 교수. 1993년~현재 광운대학교 전자공학부 교수 (정보통신 연구원)