

RTP를 이용한 CBQ기반의 QoS 성능에 관한 연구

하미숙* · 박승섭**

*부경대학교 전산교육학과, **부경대학교 전자컴퓨터정보통신공학부 교수

A Study on QoS Performance Based on CBQ Using Real-time Transport Protocol

Mi-Sook Ha* · Seung-Seob Park**

*Dept. of Computer education of Pukyong National University

**Division of Electronics, Computer and Telecommunication Engineering, Pukyong National University

요 약 : 인터넷 환경에서 실시간 서비스에 대한 제공을 위해 제안된 RTP(Real-time Transport Protocol)는 실시간 전송 프로토콜로서 스트림형 데이터 전송을 목적으로 한 프로토콜이다. RTP는 기본적으로 RTCP(Real-time Transport Control Protocol)와 쌍을 이루어 동작하며, RTCP는 현재 네트워크의 상태 정보를 전달한다. RTP는 종단 시스템에서 실행되고, 디멀티플렉싱을 제공하는 전송 프로토콜의 중요한 특성을 가지고 있다. 또한, TCP 같은 전송 프로토콜에서 제공하지 못하는 신뢰성과, 프로토콜에 정의된 흐름/혼잡제어를 제공한다.

본 논문에서는 RTP상에서 동작하는 Differentiated Service의 개념과 구조를 살펴보고, Differentiated Service상에서 효율적인 패킷 전송을 위해 CBQ(Class-Based Queuing)을 사용한 패킷 전송 모델의 파라미터를 설정함으로써 각각의 서비스 큐를 적절하게 제어하고, WRR(Weighted Round Robin)과 PRR(Packet-by-packet Round Robin) 같은 패킷 스케줄링 기법을 통해서 모든 서비스 클래스들이 기아현상을 경험하지 않고 공평한 스케줄링이 이루어지도록 컴퓨터 모의실험을 통해 성능을 확인한다.

핵심용어 : 실시간 전송 프로토콜(RTP), Class-Based Queuing(CBQ), 서비스품질(QoS), 차별화된 서비스(Differentiated Services)

ABSTRACT : RTP that is proposed supplement of real-time services on internet environment, as Real-time Transport Protocol, is the protocol that for the purpose of sending data of stream type. RTP and RTCP(Real-time Transport Control Protocol) basically work at the same time, RTCP serves with state information of network at present. RTP has important properties of a transport protocol that runs on end-to-end systems and provides demultiplexing. It also offer reliability and protocol-defined flow/congestion control that transport protocol like TCP can not provides. In this paper, we look around concept and construction of Differentiated service that run on RTP and by setting parameters of packet transfer method be used CBQ(Class-Based Queuing) for packet transfer on Differentiated service, each service queue controls properly through packet scheduling method, such as WRR(Weighted Round Robin) and PRR(Packet-by-packet Round Robin). all service classes do not experience the starvation and confirm the performance through computer simulation to achieve fairly scheduling.

KEY WORDS : RTP, Class-Based Queuing (CBQ), Quality of Service (QoS), DiffServ(Differentiated Services)

1. 서 론

컴퓨터의 보급과 인터넷 사용자의 증가에 따라 네트워크 환경이 일반화되었다. 또, 인터넷상으로 오디오와 비디오 콘텐츠를 전송하고 수신하는 네트워크 응용들이 상당히 많이 개발되어 왔고 또한 사용되고 있다. 그래서 영상이나 화상회의 같이 높은 대역폭과 실시간 전송을 요구하는 서비스가 많이 도입되

고 있다.[3] 새로운 멀티미디어 네트워킹으로 오락 비디오, 인터넷 라디오, 멀티미디어 WWW사이트, 원격회의, 대화형 게임, 가상 세계, 원격 교육 등 이러한 응용 서비스의 요구사항은 웹 텍스트나 이미지, FTP, DNS응용과 같은 기존의 데이터 중심의 응용들과는 상당히 다르다. 특히 멀티미디어 응용은 종단간 지연(delay) 및 지연변이(delay variation)에 매우 민감하나, 약간의 데이터 손실에는 둔감한 편이다. 이와 같이 근본적으로 양이

한 서비스 요구사항 때문에, 주로 데이터 통신을 위해 설계된 네트워크 구조가 멀티미디어 응용을 지원하기에 적합하지 않을 수 있다. 따라서, 많은 사용자가 요구하는 멀티미디어 데이터의 전송 서비스를 지원하고, 멀티미디어 데이터의 전송을 프레임화하고 제어하는 새로운 응용계층 표준인 RTP/RTCP 프로토콜이 개발되었다.[6]

RTP는 실시간 데이터 전송을 위한 표준 패킷형식이며, 종단간 네트워크 전송 프로토콜로써 오디오, 비디오 그리고 시뮬레이션 데이터 등의 실시간 데이터를 전송하기에 적합한 기능을 제공해주며, RTCP는 RTP데이터의 피드백 정보를 제공하며, 피드백 정보는 최소한의 QoS를 제공하는데 사용된다.

실시간 데이터의 전송에 대한 보장은 요구하는 서비스(QoS)를 제공하기 위해서 최선의 서비스 이상의 서비스가 필요한데, 인터넷 표준화 기구인 IETF에서는 IntServ(Integrated Services)모델과 DiffServ(Differentiated Services)모델 등과 같이 인터넷상에서 QoS를 제공하는 방법에 대하여 표준화 작업을 진행하고 있다.

본 논문에서는 실시간 멀티미디어 데이터의 효과적인 전송을 위해서 다양한 형태의 데이터를 네트워크에 적용하여 데이터의 QoS를 보장하기 위한 방법으로 CBQ 기반에서의 패킷 스케줄링 기법에 대해 성능평가를 해보고자 한다. 2장에서는 RTP/RTCP와 IntServ, DiffServ 등의 현재 인터넷에서 사용되고 있는 QoS방법과 스케줄링 기법에 대해 알아보고, 3장에서는 CBQ에 관한 성능분석을 하고, 끝으로 4장에서는 결론과 향후 연구과제로 맺도록 한다.

2. 관련연구

2.1 RTP/RTCP

RTP는 IETF에서 인터넷 제안표준으로 설계되어진 RFC 1889(RTP: A Transport Protocol for Real-Time Applications)와 RFC 1890(RTP Profile for Audio and Video Conferences with Minimal Control)에 발표되었다.[6] RTP는 보통 사용자 데이터그램 프로토콜 (UDP)의 상위통신 규약으로 사용되며, 전형적으로 RTP는 멀티플렉싱, 체크섬 그리고 멀티캐스팅과 같은 서비스를 제공한다. 송신측은 타임 스탬프를 근거로 재생 동기를 위해서 지연이 큰 패킷을 포기할 수 있다. RTP의 주요 기능은 네트워크 종단에 오디오, 비디오 및 시뮬레이션 데이터와 같이 실시간 특성을 갖는 데이터를 송수신 하기 위한 트랜스포트층 통신규약으로 RFC 1889에 RTCP와 함께 규정되었고, 라우터 등의 통신망 기기에 의지하지 않고 단말간에 실행되는 것이 특징이다.

또한, 옵션 메커니즘을 첨가함으로써 부가적인 기능을 확장할 수 있게 하는 전통적인 프로토콜과는 달리 필요한 만큼 RTP 패킷의 헤더를 수정 또는 첨가함으로써 특정 애플리케이션에 유연하게 적용할 수 있게 한다. 그리고 또한 RTP는 버전 확인, 패킷의 순서 번호화, 타임 스탬핑 등의 기능을 제공한다.

그러나 RTP 그 자체로는 시간에 따르는 순서화 배달, 패킷의 실시간 서비스, 기타 QoS보장을 위한 어떠한 메커니즘도 제공하지 않는다. 따라서 세션의 QoS를 모니터 하거나 세션에 참가 중인 참가자들에 대한 식별 정보와 같은 여러 가지 정보들을 전달하는 실시간 트랜스포트 제어 프로토콜인 RTCP가 부가적 확장 프로토콜로 사용된다. RTCP는 분실된 패킷 수, 지터, 직전 패킷과의 지연 시간 등의 QoS 정보를 교환하도록 하여, 애플리케이션이 적절한 QoS를 설정할 수 있게 해 준다.[4][6]

RTP는 기본적으로 다자간 멀티미디어 회의의 필요에 의해 설계되기는 하였지만, 특정 애플리케이션에 제한을 두기보다는 스트림 데이터의 저장, 대화식 분산 멀티미디어 데이터, 그리고 제어 및 측정 프로그램 등 다양한 애플리케이션에 적용될 수 있다. 또 수신 측에서 전송지연이나 대역폭들을 점검, RTCP를 사용해서 송신측의 상위층 애플리케이션에 통지하는 것으로 부호화 속도 등의 조정을 통해 QoS제어를 실현할 수 있다.

RTP 패킷과 RTCP 패킷은 하위 망의 지원 여부에 따라 유니캐스트 또는 멀티캐스트 환경에서 모두 사용될 수 있다.

Fig.1은 다른 프로토콜과 RTP와의 관계를 나타낸 것이다.

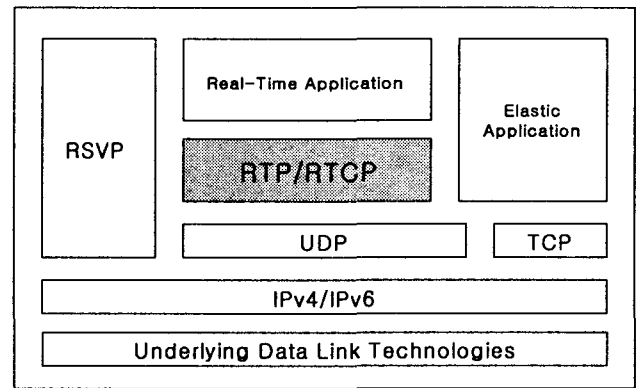


Fig.1 RTP와 관련 프로토콜

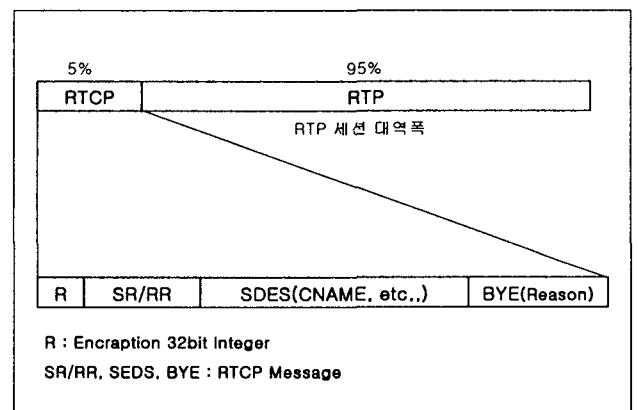


Fig.2 RTCP 패킷의 구조

RTCP는 RTP 데이터 제어 프로토콜로서 네트워크 환경에서 실시간 미디어 서비스의 QoS를 보장하기 위한 피드백 제어 방법을 SR와 RR 메시지에 의해 제공한다.[2] RTP 소스의 식별

을 위해 지속적인 식별자를 수송하게 된다. RTCP는 세션 대역폭의 5%이내에 생성되어야 하며, 최소 간격은 RTCP 5초 이상으로 설정해야 한다. RTCP 패킷사이의 간격은 모든 참가자들이 원하지 않는 동기화를 피할 수 있도록 계산된 범위에서 랜덤하게 변해야 한다. 이와 같이 RTCP의 전송간격은 RTCP패킷으로 인한 네트워크 상의 폭주 유발을 방지하기 위한 것이다. Fig.2는 패킷구조와 RTP 패킷과의 관계를 보여준다.

2.2 IntServ 와 DiffServ

인터넷 표준화 기구인 IETF에서는 인터넷상에서 QoS를 제공하는 방법에 대하여 표준화 작업으로 IntServ(Intergrated Services)와 DiffServ(Differentiated Services)를 설계했다. IntServ(Intergrated Services)는 자원예약을 위해 모든 라우터에 자원예약 상태에 대해 저장하고 있어야 한다. 인터넷 백본망에는 수많은 개별적인 패킷흐름이 존재하고 이런 흐름의 상태를 유지하기 위해서 큰 메모리 공간이 필요하게 된다. 자원예약이나 각 흐름에 대한 flowspec의 생성, 관리 등을 수행하기 위해 RSVP(resource reservation protocol) 프로토콜을 사용하게 된다. 이 수많은 흐름에 의한 자원 예약을 처리하기 위해서 라우터는 빠른 처리능력을 가져야 하며, IntServ(Intergrated Services) 도메인 내의 모든 라우터에 RSVP 프로토콜을 처리할 수 있는 능력이 있어야 된다. 또한, 각각의 라우터에서는 복잡한 패킷 분류 작업이 수행된다.[3][8] IntServ(Intergrated Services)모델에서는 서비스에 따른 QoS를 보장하기 위해서 망의 모든 라우터에는 패킷헤더 정보에 의해서 패킷 분류를 수행할 고도의 처리능력을 요구하게된다. 그러나 현재 이 기술은 라우터 제조 회사들 중 일부에서 이를 구현하고 있고, policy를 지원할 수 있는 수준에 이르렀으나 IntServ(Intergrated Services) 모델의 확장성 문제 때문에 DiffServ(Differentiated Services) 방식으로 많이 구현되고 있다.

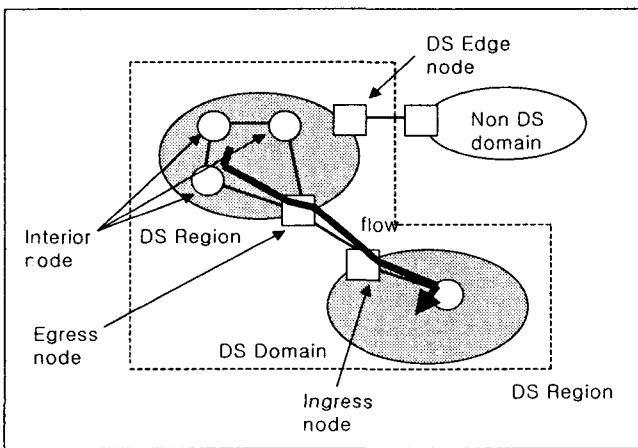


Fig.3 Differentiated Service 망 기본 구조

Fig.3은 DiffServ(Differentiated Services)의 망에 대한 기본 구조를 보여 주고 있다. DiffServ(Differentiated Services)는

IPv4헤더에 QoS를 보장하기 위해 TOS(Type of Service) 필드를 가지고 있고, IPv6패킷의 클래스 필드 바이트 DSCP(DS Code Point)의 값에 의해서 결정되는 PHB(Per-Hop Behavior) 정책과 트래픽 조절기, 네트워크의 관리정책의 복합적인 구조를 통한 다양한 서비스를 실현할 수 있는 구조를 가진다. 그러나 기존 대부분의 라우터들에는 이 필드를 무시하고 모든 패킷을 동일하게 처리를 한다. DiffServ(Differentiated Services)는 TOS필드의 이름을 DS필드로 재 명명하고 DSCP(DS Code Point)값을 할당하여 서비스 클래스를 구분하도록 하였다. Fig.4는 현재 RFC에서 정의하고 있는 DS 바이트 포맷형식이다.[1][5][7]

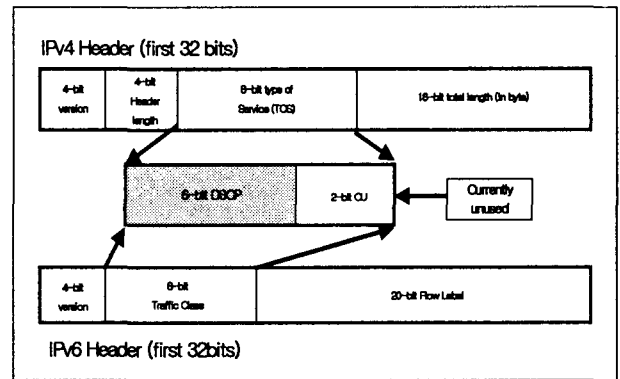


Fig.4 IPv4 와 IPv6헤더의 DS byte

Fig.4에는 DSCP로 6비트가 할당되어 있다. 이 코드 값은 패킷이 경유하는 라우터에서 패킷이 전달되는 순서와 바퍼 할당과 같은 패킷 전달 방식을 결정하게 된다.

DiffServ(Differentiated Services)에서는 이러한 망 내부 라우터에서 패킷전달 방식을 PHB(Per-Hop Behavior)라고 부른다. DiffServ(Differentiated Services)모델의 패킷 전달 방식인 PHB(Per-Hop Behavior)는 DSCP의 값에 따라 서로 다른 패킷 전송방법을 제공함으로써 다양한 버퍼 관리 및 패킷 스케줄링 기법을 제공하며 QoS를 보장하는 상대적인 우선 순위 기법이다.

이렇게 정의된 PHB들이 집합을 형성하여, 큐 관리 방식과 같은 공통적인 규약에 따라 서비스를 받을 수 있는데, 이러한 PHB들의 집합을 PHB 그룹이라 한다. 지금까지 DiffServ WG에서 정의하고 있는 PHB로는 세 가지가 있는데 그중 DE PHB는 Codepoint '000000'으로 정의 되어있고 현재 인터넷에서 서비스 되고있는 최선의 서비스 형태이다. EF PHB는 Codepoint가 '101110'으로 우선 순위가 가장 높은 전달 방식으로서 확실한 대역폭뿐만 아니라 낮은 지연과 손실률도 보장하며, 다른 트래픽의 부하에 의한 영향을 받지 않는다. EF 트래픽에 대한 서비스는 두 부분으로 이루어져 있는데 첫째로 각 노드의 EF PHB는 적절한 최소 전송률을 설정해야 하며, 둘째로 DS 도메인의 경계노드에서는 트래픽을 조절하여 내부 노드에서 패킷의 도착율이 전송률을 넘어서지 않도록 고려해야 한다. AF PHB

그룹은 Table.1과 같이 4개의 클래스로 구성되어 있으며, 각 클래스는 다른 클래스와 독립적으로 대역폭이나 버퍼와 같은 자원이 할당된다. 또한 각 클래스별로 3등급의 폐기 순서가 구분되어 총 12개의 PHB로 나뉘어져 있다.[8][9]

구분	Class 1	Class 2	Class 3	Class 4
Low Drop Prec.	001010	010010	011010	100010
Medium Drop Prec.	001100	010100	011100	100100
High Drop Prec.	001110	010110	011110	100110

Table.1 AF PHB와 Codepoint

2.3 CBQ

실시간 트래픽의 성능에 중요한 영향을 미치는 패킷 스케줄링 알고리즘은 각 흐름에 할당된 만큼의 대역폭을 다른 트래픽의 영향을 받지 않게 보장해 주는 독립성과 모든 흐름들이 공평하게 여분의 대역폭을 공유할 수 있는 공평성을 제공해야 한다. 이러한 조건을 만족시키기 위하여 많은 패킷 스케줄링 알고리즘이 제안되었으며, 이들은 크게 우선 순위 제어방식(Priority mechanism)과 대역 보장형 스케줄링 방식으로 나뉘어진다. 우선 순위 제어방식은 높은 우선 순위를 갖는 패킷이 언제나 우선 순위가 낮은 패킷들보다 서비스 순서에 차이를 두어 먼저 처리되는 기법이다. 그리고 대역 보장형 스케줄링 알고리즘은 우선 순위 제어방식의 문제점을 극복하기 위해 각 사용자 흐름마다 QoS를 보장해 줄 수 있는 보다 복잡한 패킷 스케줄링 기법이다. CBQ는 Priority queuing의 변형으로 서비스 클래스에 따라 서비스 큐를 정의하고 있으며, 특정 서비스 클래스가 낮은 우선 순위로 인해 자원을 할당받지 못하여 생기는 기아현상을 막기 위해 고안되었다. CBQ알고리즘은 연결공유를 기반으로 하는 패킷 스케줄링 알고리즘이다.

먼저, 사용자 트래픽은 트리나 계층적으로 클래스화 되어지며, 이런 클래스는 어떠한 어플리케이션을 총칭하는 집합의 flow를 의미하게 된다. 따라서, 이러한 클래스의 트래픽에 맞는 적절한 QoS를 위해서는 그것의 각각에 대해 Queue를 해주어야 한다. 따라서, 각각의 트래픽의 클래스에 맞게 bandwidth rate(per-cent), priority, average packet size, borrow, maxburst, minburst들의 파라미터를 가지고 있다.

CBQ의 구현에 사용된 스케줄링 방법으로 PRR(packet-by-packet round robin)패킷 스케줄러와 WRR(weighted round robin)패킷 스케줄러를 사용한다. 이러한 패킷 스케줄링방법은 우선 순위에서 패킷을 스케줄링하는 방법에 따라 구분되는데, PRR 패킷 스케줄러는 같은 우선 순위를 갖는 서비스 클래스 내에서 자신의 스케줄링 라운드동안 패킷단위로 서비스된다. 그러나 WRR패킷 스케줄러는 서비스 클래스에 할당된 대역폭의 비중에 따라 각 서비스 큐에 가중치(weight)를 두어 서비스한다. 지연과 지터, 데이터 손실 등과 같은 응용들의 요구 사항

에 기초하여 패킷들 사이에서 서비스를 차별화하는데 사용된다. 각 큐의 무게는 스케줄링 하는 동안에 큐가 전송할 수 있는 트래픽의 양으로 표현될 수 있다.

CBQ는 일반적으로 패킷 분류, 패킷 스케줄러와 대역폭의 처리가 필요하다. CBQ 스케줄러는 트래픽의 혼잡이 없는 경우 패킷의 스케줄링하는 priority based scheduler(general-scheduler)와 혼잡이 있을 시 사용되는 link-sharing 스케줄러를 가지고 있다. 이러한 연결공유 스케줄러의 2가지 기능을 가지고 있는데, 첫째로 각각의 클래스에서 적절한 시간간격으로 대역폭을 할당하는 것과, 두 번째로는 적당한 guideline에 의한 초과 대역폭을 나누는 역할을 한다. 따라서, 이러한 CBQ는 패킷이 도착하면, 먼저 클래스를 나누고, 그 트래픽 클래스가 underlimit 클래스이면 general scheduler에 의해 output link로 나가게 되고, overlimit 클래스이면, 연결공유 스케줄러를 이용한다.[8][10]

Fig.5는 일반적으로 사용되는 CBQ를 보여주고 있다.

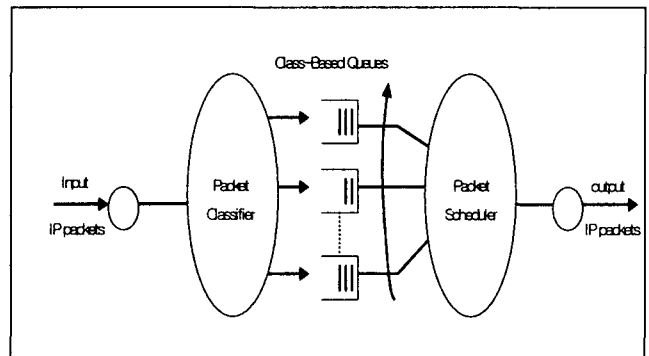


Fig.5 CBQ의 일반적 구조

3. 시뮬레이션 환경 및 성능분석

본 논문에서 실시한 성능분석을 위해 사용한 시뮬레이터는 Network Simulator Version 2를 사용하여 Hancorn Linux 3.0 서버 상에 설치하여 실험하였다.

CBQ 구현을 실험하기 위해서 서버와 라우터 사이의 연결은 10Mbps 스위칭 이더넷을 사용하였다. 또한 라우터에 연결된 4대의 노드는 각각 1.5Mbps 속도로 연결하였고, 실시간 패킷의 송수신을 위해 각각 노드에 1개의 클래스를 생성하였다. A 노드에 비디오 클래스를 생성하고 연결 대역의 22%를 할당하고 우선 순위는 1로 설정하였다, B노드에 오디오 클래스를 생성하고 연결 대역의 3%를 할당하고 우선 순위는 1로 설정하였다, C노드에는 ftp클래스를 생성하고 연결 대역의 65%를 할당하고 우선 순위는 2로 설정하였다, D노드에는 e-mail 클래스를 생성하고 연결 대역의 10%를 할당하고 우선 순위는 1로 설정하였다. 또한 비디오클래스의 데이터 소스는 0.001초 당 190byte의 패킷을 보내는 CBR 흐름이고, 오디오클래스의 데이터 소스는 0.002초 당 500byte의 패킷을 보내는 CBR 흐름이고, ftp클래스의 데이터 소스는 0.005초 당 1000byte의 패킷을 보내는 CBR 흐름이고, mail클래스의 데이터 소스는 0.005

초 당 1000byte의 패킷을 보내는 CBR 흐름이다. Fig.6은 실험을 위한 망의 구성도를 나타내며, Fig.7은 우선 순위와 연결-공유를 위한 대역폭의 할당을 나타내고 있다.

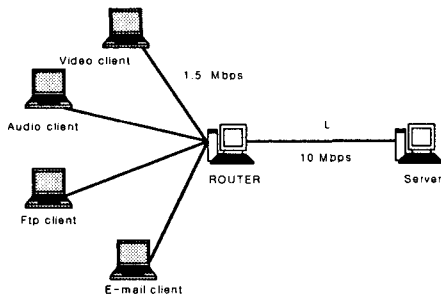


Fig.6 망구성도

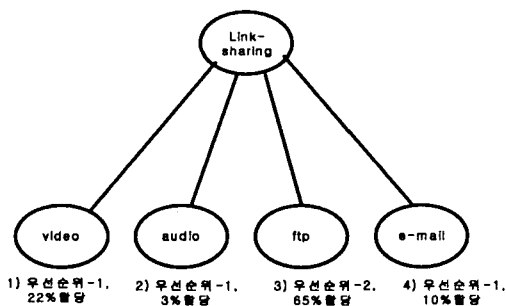


Fig.8 우선순위와 연결-공유할당

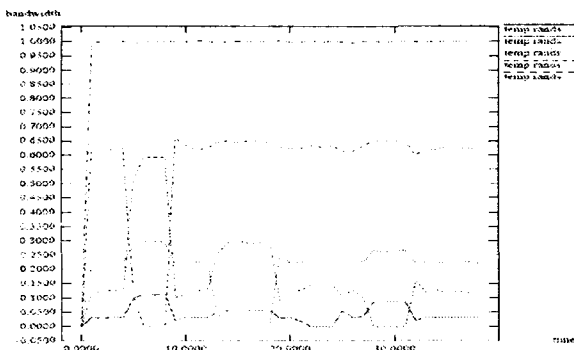


Fig.8 PRR 스케줄링 방법에 의한 대역폭 할당

Fig.8과 Fig.9의 결과를 보면 PRR과 WRR은 우선 순위가 낮은 클래스가 전송할 데이터가 없을 때, 다른 클래스들에게 여분의 대역을 할당하는 방법에 차이가 있다. PRR은 여분의 대역이 생겼을 때, 우선순위가 같은 경우 다른 클래스의 패킷 길이에 비례해서 여분의 대역을 배분한다. 반면에 WRR은 할당된 대역에 비례해서 각 클래스에 여분의 대역을 분배한다.

본 논문에서는 초기 대역의 할당 값과 우선 순위, 패킷의 길이, 큐의 길이를 각각 바꿔가며 시뮬레이션을 하였다. 결과적으로 PRR은 패킷의 길이에 비례하여 여분의 대역을 할당을 받고, WRR은 초기 할당 값에 비례하여 여분의 대역을 할당받는다. PRR의 문제점은 같은 우선 순위라도 클래스들의 패킷의 크기가 다양할 때, 서비스하는 순서가 달라지게 된다. 그래서

원하지 않은 트래픽이 발생하게 되고 서비스의 품질이 떨어지는 경우도 발생하게 된다.

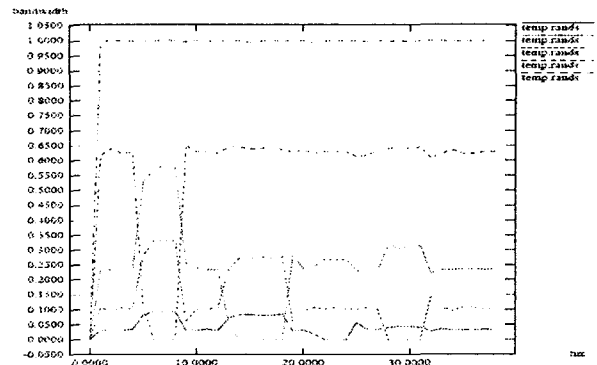
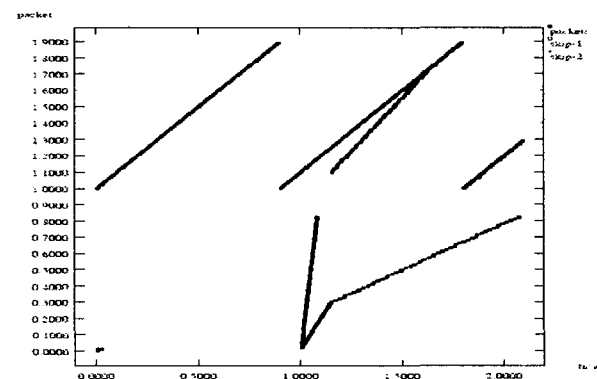


Fig.9 WRR 스케줄링 방법에 의한 대역폭 할당

WRR은 각 클래스마다 다른 가중치를 가질 경우 할당받는 여분의 대역은 가중치에 비례하기 때문에 초기에 설정한 대역의 설정 값이 크다면 여분의 대역이 생겼을 때, 다른 클래스보다 이 클래스가 더 많은 대역을 할당받기 때문에, 다른 클래스들이 기아현상을 경험하게 될 것이다.

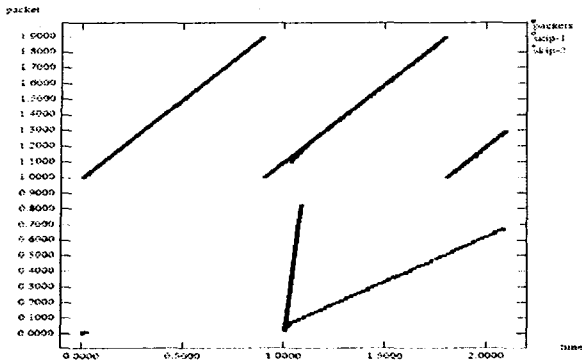
각 서비스들이 이런 불공평한 현상을 경험하지 않고 최적의 서비스를 받기 위해서는 연결-공유 스케줄링 방법에 의해 발생될 수 있는 기아문제를 해결할 수 있다. 또한 각 클래스들이 사용하는 연결의 사용 한계 값을 정의하고 제한하는 파라미터를 두어 연결의 사용량을 일정한 값으로 유지시키게 되어 새롭게 생성되는 응용들이 발생시키는 실시간 트래픽들을 망에서 수용할 수 있는 응용성을 제공할 수 있다. 또한, 본 논문에서는 CBQ의 파라미터 중에서 평균-휴지 시간에 대한 (양수) 상한 값으로 요구시간과 현재활동한 시간이 너무 길지 않게 어떤 값을 주어 그 값 이상 기다리지 않게 하는 값과(Fig.10), 클래스에 할당된 시간을 모두 소비했을 때 클래스에 대한 안정된 상태로 만들어 주기 위한 값(Fig.11)에 대해 할당될 값들을 변화시켜 시뮬레이션을 해 보고 결과를 분석해 보았다.



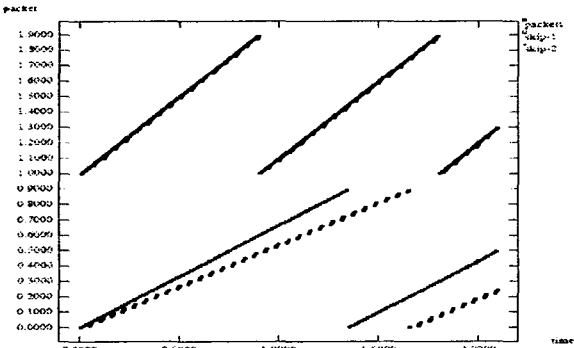
(a) 패킷의 크기가 25일 때 최대 대기시간

Fig.10에서 (a)는 요구 시간과 현재 활동한 시간 간격이 너무 길지 않게 설정 값을 주어 그 값 이상 기다리지 않게 하도록 한다. 또한 (b)는 최근 시간에 아무런 패킷도 보내지 않은 클래스에게 전송을 허락할 때 버스트의 최대 크기를 설정한다.

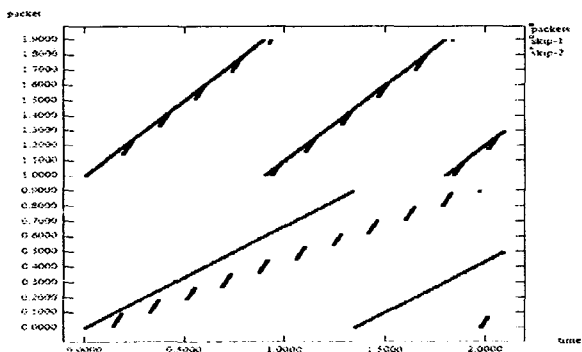
Fig.10의 시뮬레이션에서 버스트값을 25패킷과 5패킷으로 설정할 때의 결과를 보여준다.



(b) 패킷의 크기가 5일 때 최대 대기시간
Fig.10 패킷의 크기 변화에 따른 최대 대기시간



(a) 버스트 크기가 2일 때 신호 지연시간



(b) 버스트의 크기가 8일 때 신호 지연시간
Fig.11 버스트 크기에 따른 신호 변화의 지연시간

Fig.11은 leaf 클래스를 위해 규정된 클래스에 대한 안정된-상태 버스트의 크기를 결정한다. 이것은 overlimit 클래스가 다른 패킷을 전송하기 전에 기다려야 하는 시간 간격을 결정한다. Fig.11의 (a)는 버스트 크기를 2로 주고 (b)는 8을 주었을 때의 시뮬레이션 결과를 보여준다.

4. 결론 및 향후 연구과제

네트워크상에서는 다양한 종류의 스트림이 송 수신되고 있다. 이러한 스트림의 효율적인 전송을 위해 트래픽 제어 기능의

핵심 요소인 패킷 스케줄링은 각 패킷의 흐름마다 고유의 서비스 품질을 제공하기 위한 방법으로 매 순간마다 이용 가능한 여러 흐름 중에서 서비스 순서를 결정하여야 한다. 기존의 논문에서는 어떠한 상황에서 한 스케줄링 기법이 다양한 스트림의 전송을 모두 만족시키는 효과를 볼 수 없었다.

본 논문에서는 DiffServ상에서 효율적인 패킷전송을 위해 CBQ를 이용하였고, 기본적으로 라운드 로빈 스케줄링을 기본으로 한 PRR과 WRR을 사용하여 스케줄링을 했다. 여러 가지 환경을 생성하여 실험해 보면서 각 스케줄링 기법의 성능을 평가하고 좀더 나은 성능을 갖게 하기 위해 여러 가지 파라미터 값을 변화시켜서 결과 값을 알아보았다. 이렇게 파라미터 값을 적절히 변화시켜 기아상태를 상당히 줄 일 수 있었다. 그러나 각 상황에 따라 최적의 성능을 나타나게 하기 위해서는 파라미터를 상황에 맞게 변화시켜야 하는 수동적인 작업이 필요했다.

따라서 앞으로의 연구는 실제 DiffServ상에서 좀더 다양하고, 안정적이며, 사용자들이 요구하는 최적의 전송에 상태에 대한 만족을 지속적으로 적용하기 위해서 각 파라미터들의 표준화 작업을 통한 모듈의 설계와, 성능보장을 위한 최적의 QoS에 대한 기법 연구가 이루어져야 하겠다.

참 고 문 헌

- [1] Diffserv를 이용한 인터넷 QoS 보장 기술
한국통신학회지(정보통신) 제18권 9호, 2001
- [2] RTP/RTCP를 이용한 인터넷상의 QoS 제어방법
한국통신학회 논문집 제 16권 3호 1997
- [3] 인터넷 QoS 제공을 위한 CBQ 기반의 스케줄링 기법
정보처리학회 논문지, 제8-C권 제4호 2001
- [4] 인터넷상의 주요 QoS 관리 기술 동향
한국통신학회지(정보통신) 제18권 9호, 2001
- [5] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," RFC 2475, December, 1998.
- [6] H.Schulzrinne, s.Casner, R. Frederick, and V.Jacobson, "RTP: A Transport Protocol for Real-time Applications." RFC 1889, January 1996.
- [7] K. Nichols, S. Blake, "Definition of the Differentiated Services Field(DS Byte) in the Ipv4 and IPv6 Headers", RFC 2474, December 1998.
- [8] Next Internet QoS Control
"http://multinet.inha.ac.kr/main/lecture/comnet/term-proj"
- [9] S. Brim, B. Carpenter, F. Le Faucheur, "Per Hop Behavior Identification codes." RFC 2836, May, 2000.
- [10] S. Floyd, and V. Jacobson, "link-sharing and resource management models for packet networks," IEEE/ACM Transaction on Networking, Vol.3, August, 1995.