

엔터프라이즈 환경에서 다양한 서비스 요구 사항을 지원하는 패킷 스케줄링 알고리즘

(Packet Scheduling Algorithms that Support Diverse Performance Objectives in Enterprise Environment)

김 병 철[†] 김 태 윤^{**}

(Byoung-Chul Kim)(Tai-Yun Kim)

요 약 네트워크에서 QoS를 보장하기 위해 최근에 제안되는 패킷 스케줄링 알고리즘은 대부분 우선 순위에 입각한 패킷 전송 서비스를 한다. 이러한 우선 순위를 유지하기 위한 큐의 관리에는 많은 비용이 들므로 QoS를 보장하는 네트워크에서 우선 순위 큐의 관리 비용을 줄이는 노력이 필요하다. 패킷 스케줄링 알고리즘 중 RPQ+(Rotate Priority Queue)는 우선 순위 FIFO(First in first out) 큐를 사용하여 주기적으로 재명명되는 패킷 스케줄링 알고리즘이다. FIFO 큐에 패킷들을 근사 정렬하여 패킷의 우선 순위를 유지하므로 계산 복잡도를 줄이지만, 패킷 우선 순위를 유지하기 위해 2배(2P)의 큐를 필요로 한다[1].

본 논문에서는 필요한 큐의 개수를 P개의 큐로 제한하여 큐에 대한 관리 비용을 줄였으며 엔터프라이즈 환경에서 애플리케이션이 요구하는 서비스 특성에 따라 클래스로 구분하여 적합한 패킷 스케줄링 서비스를 제공하는 알고리즘을 제시한다. 본 기법은 추가적인 오버플로우 큐를 관리하고 패킷 어드미션 컨트롤러를 통해 패킷 전송 지연 시간을 제한함으로써 다양한 애플리케이션의 네트워크 QoS 요구를 보장하고 패킷 전송 효율을 높였다.

Abstract Most recently proposed packet scheduling algorithms that support network QoS transmit packets in some priority order. Since maintaining a sorted priority queue introduces significant overhead, they need effort to reduce management overhead of priority queue that supports network QoS. One of the packet scheduling algorithm RPQ+(Rotate Priority Queue) uses priority FIFO(First in First out) queues that are periodically relabeled. RPQ+ algorithm maintains packet priority order by less accurate sorting for lower computational overhead. But it needs twice the number of priority queues for the purpose of keeping packet priority order[1].

In this paper, we propose improved packet scheduling algorithm that reduces queue management overhead by the number of P queues and divides multimedia applications as demand on their service in enterprise environment. In this method, the scheduler meet multimedia application network QoS which provides deterministically bounded delay service and improve packet transmission efficiency, by managing additional overflow queue and packet admission controller.

1. 서 론

인터넷 사용자의 급속한 증가와 전자 상거래의 발달로 기업의 컴퓨터 시스템 및 네트워크를 설계하고 관리하는 일은 매우 복잡하게 변화하고 있다. 더욱이 수만명

의 인터넷 사용자와 인터넷으로 연결 되어 있는 지구촌 곳곳의 수억명의 사용자를 고려해야 하는 엔터프라이즈 환경에서 네트워크는 성능, 신뢰성, 보안성 등을 요구하는 멀티미디어 애플리케이션의 요구 사항을 만족시켜야 한다. 이러한 네트워크 QoS를 지원하기 위해서는 패킷 전송 순위를 결정하는 패킷 스케줄링 알고리즘과 요구되는 패킷 서비스 시간을 보장하여 주는 알고리즘이 필요하다.

엔터프라이즈 환경에서의 네트워크 QoS 요구는 VOD 서비스와 같이 한가지 서비스를 제공하는 멀티미

[†] 비 회 원 : 고려대학교 컴퓨터학과
bckim@netlab.korea.ac.kr

^{**} 종 신 회 원 : 고려대학교 컴퓨터학과 교수
tykim@netlab.korea.ac.kr

논문접수 : 2000년 2월 17일
심사완료 : 2000년 7월 18일

디어 애플리케이션에서의 QoS 요구와 상당한 차이를 보이고 있는데, 무엇 보다도 임계작업(mission critical)의 애플리케이션을 우선적으로 처리하고 나서 VOD 서비스나 대화형 애플리케이션의 지원을 요구하고 있다. 따라서 엔터프라이즈 환경에서 네트워크 스케줄링 알고리즘은 멀티미디어 애플리케이션을 비슷한 서비스 목적과 특성을 가지는 클래스로 나누고 각 클래스별로 요구되는 지연 시간과 가능한 패킷 손실률(loss rate)을 고려한 스케줄링 알고리즘이 필요하다. 엔터프라이즈 환경하에서 멀티미디어 애플리케이션의 서비스 요구 사항은 애플리케이션 특성에 따라 다양한데 예를 들면, 대화형 애플리케이션에게는 낮은 평균 지연시간을 요구할 수 있고, 비디오나 음성을 서비스하는 애플리케이션에게는 적절한 패킷 손실율을 통한 서비스 품질의 보장이 요구 사항이 될 수 있다. 기업 인사 관리 애플리케이션 같은 경우에는 패킷 손실이 없는 신뢰성 있는 전송 서비스를 요구 할 수 있다.

멀티미디어 애플리케이션의 네트워크 서비스 목적을 달성하기 위해서는 패킷 전송의 지연 시간을 제한(bound)하는 알고리즘이 필요하며, 최근에는 대부분 우선 순위에 입각한 알고리즘들이 제안되었다[3],[4],[5]. CBS(Cost Based Scheduling) 알고리즘은 패킷의 서비스 목적에 따라 값(cost)을 정하여 이 값에 따라 스케줄링을 하는 기법이다. 이 기법은 모든 패킷에 대해 값을 계산해야 하는 단점이 있다[4]. 폴링(polling)에 기반한 MARS 스케줄링 기법은 패킷들을 3분류의 클래스로 구분하여 클래스 1에 해당하는 패킷은 지연 시간이 지정된 최대값을 넘지 않는 것을 목표로 하며, 클래스 2에 해당하는 패킷은 마감 시간을 넘길 수도 있는 패킷들을 목표로 한다. 클래스 3에 해당하는 패킷은 클래스 1과 2에 패킷 전송 효율을 떨어뜨리지 않는 범위에서 최대한 빨리 서비스 되는 것을 목표로 한다. MARS 기법은 수행 목표에 따라 스케줄링을 하는 효과가 있지만 다양한 서비스 목표를 고려하지 못한 단점이 있다[5]. 선점 기반(occupancy-based) 알고리즘은 다수의 클래스의 패킷이 존재하고 각 클래스에 얼마나 많은 패킷들이 큐에서 서비스 대기 중이냐에 따라 스케줄링이 정해지는 기법이다. 즉 음성 패킷은 어떤 경계값을 넘지 않는 한 일반 데이터 패킷보다 우선 순위가 높다. 이 기법은 음성 패킷이 어떤 경계값 이하일때는 효과적인 스케줄링 기법이지만, 패킷의 도착율의 변동이 심할때는 좋지 못한 성능을 보인다[6]. 이러한 알고리즘들은 대부분 정렬된 우선 순위 큐의 관리를 요구하므로, 이러한 우선 순위 큐의 유지에 드는 비용을 줄이면서 최적의 성능을 보이

는 EDF(Earliest Deadline First) 스케줄러에 근접한 성능을 보이는 알고리즘이 연구되었다[1]. [1]은 우선 순위 큐 관리에 있어 근사 정렬(approximate sort)된 우선 순위 큐 알고리즘을 사용하여 낮은 계산 복잡도를 가지는 장점과, 일정 시간 간격(Δ)으로 FIFO 큐를 회전함으로써 패킷 지연 시간을 제한하는 특성을 가진다. 그러나 [1]은 패킷 스케줄링에 평균 지연 시간만을 고려하여 엔터프라이즈 환경에서 멀티미디어 애플리케이션들이 요구하는 각각의 특성이 고려되지 않았으며, 큐의 회전후에 발생하는 패킷 우선 순위의 역전 현상을 해결하기 위해 기존의 큐의 개수에 2배 만큼의 큐를 사용하여 큐의 관리에 많은 비용이 든다.

따라서 본 논문에서는 엔터프라이즈 환경에서 애플리케이션 특성에 따른 스케줄링 알고리즘 및 큐의 주기적인 회전에 따른 패킷 우선 순위 역전 현상을 해결하기 위해 큐의 개수를 2배(2P) 늘리지 않고 P개의 큐를 사용하므로 큐를 관리하기 위한 비용을 줄인 알고리즘을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로서 엔터프라이즈 환경과 기존의 패킷 스케줄링 알고리즘을 분석한다. 3장에서는 엔터프라이즈 환경에서 애플리케이션 특성에 따른 패킷 스케줄링 알고리즘을 제안한다. 4장에서는 기존의 관련 알고리즘과의 성능을 비교 분석하고, 5장에서는 결론과 향후 과제를 제시한다.

2. 관련 연구

본 장에서는 엔터프라이즈 환경에 대한 실험과 멀티미디어 애플리케이션을 서비스 특성에 따라 클래스로 구분하여 분석하고, 근사한 우선 순위로서 계산 복잡도를 줄인 RPQ 알고리즘을 분석한다.

2.1 엔터프라이즈 환경

엔터프라이즈 환경이란 수백 대의 서버와 초고속 백본 네트워크를 수만 명의 사용자가 여러 애플리케이션에 우선 순위를 두어 시스템 자원을 공유하는 것으로 인터넷이나 통신 환경에서는 다양한 애플리케이션의 복합적인 고려가 필요하다. 예를 들면 비디오 스트림이나 음성 데이터뿐만 아니라, 전자 상거래, 원격 교육 등에 필요한 보안 데이터와 다른 서비스 수준을 요구하는 애플리케이션 데이터가 함께 고려되어야 하는데, 대부분의 네트워크 QoS를 보장하는 패킷 스케줄링 알고리즘은 하나의 데이터 스트림을 기준으로 삼고 있다. 이러한 네트워크에서 스위칭 및 평균 응답 시간, 서비스 지연 시간의 제한등의 보장 및 QoS를 요구하지 않는 애플리케이션

이션에서의 패킷 손실, 재전송 및 기아(starvation) 현상 등의 부작용을 최소화 하는 것도 스케줄링 시에 고려되어야 한다.

표 1 엔터프라이즈 환경에서 애플리케이션의 분류

구분	특성
클래스 1 (임계 작업)	<ul style="list-style-type: none"> ● 연성 실시간(Soft real time)또는 경성 실시간(hard real time) ● 애플리케이션에서 데이터 손실을 허용하지 않는다. ● 패킷 크기는 100byte 정도로 크지 않다. ● 연결당 1Kbps 정도의 적은 대역폭을 요구.
클래스 2 (대화형 작업)	<ul style="list-style-type: none"> ● 주기적으로 데이터를 보내는 경성 실시간. ● 재전송과 같은 메커니즘이 의미가 없으며, 전진 에러 수정과 같은 메커니즘이 요구. ● 음성성의 경우 패킷 크기가 100byte 정도이며, 비디오의 경우는 수 Kbyte 이다. ● 대역폭은 10-64Kbps 정도가 요구.
클래스 3 (고용량 고품질 작업)	<ul style="list-style-type: none"> ● 경성 실시간을 요구. ● 높은 정밀도의 오디오/비디오 스트림 처리를 요구. ● 전진 에러 수정이 필요. ● 패킷 크기가 매우 크다. ● 대역폭을 많이 요구.
클래스 4 (일반 작업)	<ul style="list-style-type: none"> ● 실시간을 요하지 않으며 적절한 응답 시간 요구. ● 보통의 파일, 텍스트, 그래픽 이미지 데이터이다. ● 패킷 크기와 요구되는 대역폭은 애플리케이션에 따라 다양하다.

엔터프라이즈 인터넷/인트라넷 환경에서 컴퓨터 및 통신 네트워크를 활용하는 서비스에 따라 멀티미디어 애플리케이션의 속성을 구분하면 표 1과 같이 4가지의 클래스로 구별된다[11]. 클래스 1의 임계 작업은 재무 시스템이 대표적으로 볼 수 있으며, 금융 시스템, 항공 예약 시스템, 전자 상거래에서의 전자 결제 시스템 등이 그 예라 볼 수 있다. 데이터의 양은 작지만 실시간 처리를 요하며 분실이나 지연이 발생해서는 시스템 전체의 신뢰성이 크게 손상된다. 사용자 입장에서의 가치가 매우 크고 중요한 데이터이다. 클래스 2의 대화형 작업은 인터넷 전화, 웹 비디오 전화, 주문형 뉴스, 인터넷 비디오 광고 등이 해당되며, 실시간 처리 및 스트림 동기화를 필요로 한다. 품질에 크게 지장을 주지 않는 범위의 지연이나, 데이터의 손실은 허용 될 수도 있다. 클래스 3의 고용량 고품질 작업은 디지털 방송, VOD 등과 같이 MPEG2 수준의 데이터를 실시간으로 전송하는 경우에 발생하는 데이터 유형이다. 이러한 클래스의 데이터는 데이터 손실 및 엄격하지 않은 에러 제어가 허용 될 수 있으나, 실시간성 및 연속성을 요구하므로 빠른 전송

을 필요로 할 뿐 아니라 각 데이터 간의 동기화에도 신경을 써야 한다. 또한 실시간으로 인코딩과 디코딩을 수행함에 따라 많은 시스템 자원을 사용하며 주기적인 패킷 서비스 요청의 특성을 보인다. 클래스 4의 일반적인 작업을 ftp, 전자 우편(email), 그래픽 이미지 전송 등의 데이터들로서 동기화나 지연 시간 제어와 같은 복잡한 메커니즘을 필요로 하지 않는다.

2.2 기존의 패킷 스케줄링 알고리즘 및 RPQ+ 알고리즘 분석

기존의 패킷 스케줄링 방식에는 FCFS(First Come First Served), SP(Static Priority), EDF 방식이 있다. FCFS 알고리즘은 패킷의 서비스 목적과는 상관없이 모든 패킷에게 동일한 수준의 서비스를 제공하는 방식으로 알고리즘의 간단성으로 많은 네트워크에서 사용되는 방식이다. SP 알고리즘은 FCFS 알고리즘이 모든 패킷을 동일한 수준으로 서비스하는 단점을 극복하고자 패킷에 정적인 우선 순위를 부여하는 알고리즘이다. SP 알고리즘은 패킷이 네트워크로 전송되기 전에 우선 순위가 부여되고, 서비스를 대기하고 있는 각각의 큐에서 가장 높은 우선 순위를 가지는 패킷부터 선택하여 전송한다. 이 알고리즘이 애플리케이션의 서비스 요구 사항 및 목적에 따라 패킷 스케줄링 서비스를 달리 할 수 있는 장점이 있지만, 마감 시한을 가지는 긴급(urgent)한 패킷의 스케줄링 서비스를 처리하지 못하는 단점이 있다. EDF 알고리즘은 패킷이 큐에 삽입되기 전에 마감 시한이 정해 지면서 큐내에서 서비스를 대기하는 패킷 중 가장 마감 시한이 임박한 패킷부터 전송 하도록 한다. 마감 시한을 넘긴 패킷은 애플리케이션 특성에 따라 계속적으로 우선 순위가 낮아져 가장 낮은 우선 순위를 가지거나 폐기 처리 될 수 있다. 이 알고리즘은 대부분의 패킷 전송에 대해 최적의 지연 시간을 제한하는 성능을 보이지만, 새로운 패킷이 스케줄러에 전송 될 때마다 패킷들이 마감 시한에 따라 정렬되어야 하므로 패킷 우선 순위 유지에 비용이 많이 들고, 애플리케이션 특성에 따라 패킷 손실이 미치는 영향을 고려하지 못하는 단점이 있다. RPQ 스케줄러는 큐에서 대기중인

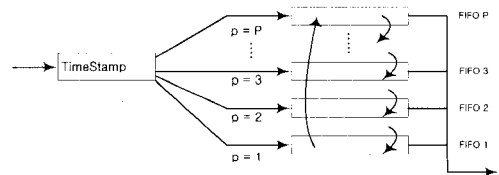


그림 1 FIFO 큐의 근사된 정렬 기법

패킷의 우선 순위를 정확하게 유지하지 않는 근사된 우선 순위 큐로 관리 비용을 줄이면서 주기적인 큐의 회전을 통해 패킷 지연 시간을 제한하여 EDF 스케줄러에 근사한 성능을 보이는 특성이 있는데 스케줄러의 동작은 그림 1과 같다.

RPQ 스케줄러의 큐가 근사된 우선 순위 큐라는 개념은 첫째로 2개의 패킷이 동일한 큐에 삽입되면 패킷의 우선 순위가 FIFO 큐에 도착한 순서대로 정해지기 때문이며, 두 번째로 주기적인 시간 간격으로 큐를 회전함으로써 FIFO 큐의 우선 순위는 증가하고 패킷은 타임스탬프에 따라 근사 정렬되는 효과를 보이기 때문이다. RPQ 알고리즘은 간단한 포인트 연산을 통해 큐를 주기적으로 회전하면서 패킷 전송 지연 시간을 제한하는 성능을 보이지만, 다양한 멀티미디어 애플리케이션마다 요구되는 지연 시간은 고려되지 못했고 그림 2 (a)와 같이 큐의 회전 시간(Δ) 바로 직전에 우선 순위 2인 패킷이 3개 들어오고 회전 시간(Δ) 바로 다음에 우선 순위 1인 패킷 1개 들어온 상황이라면 FIFO(1)에서의 패킷 순서는 그림 2 (c)와 같이 우선 순위 1인 패킷이 우선 순위 2인 패킷 뒤에 위치하게 된다. RPQ 스케줄러의 패킷 우선 순위 역전 현상은 큐의 회전 시간을 0에 가깝게 줄임에 따라 큐를 회전 시키지 않는 SP 알고리즘보다 성능이 좋지 않는 문제점을 보였다[2]. 이러한 문제점을 해결하기 위해 RPQ' 알고리즘은 2배 만큼의 큐의 개수를 사용하여 스케줄링을 한다.

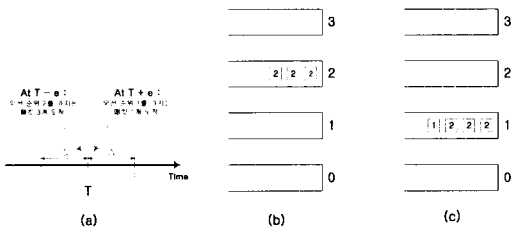


그림 2 우선 순위의 역전

RPQ' 알고리즘은 기존의 큐의 개수가 P개라 하면 $F(0), \dots, F(P-1)$ 의 큐에 새로이 $F(0'), \dots, F(P-2')$ 의 큐를 두어 스케줄링을 한다. RPQ' 알고리즘에서는 2 단계로 스케줄링이 실시되는데 각각 연쇄(concatenation) 스텝, 승진(promotion) 스텝으로 구성된다.

그림 3의 (i)에서 새로이 우선순위 p인 패킷이 스케줄러에 도착하면 $F(p)$ 로 명명된 큐에 삽입된다. 그림 3의 (ii) 연쇄 스텝에서 $F(p')$ 큐와 $F(p)$ 인 큐가 통합되어,

새로 삽입되는 패킷은 $F(p')$ 의 패킷들 보다 앞서서 서비스 받도록 스케줄링 된다. 그림 3의 (iii) 승진 스텝에서는 기존의 RPQ와 같이 큐의 회전을 통한 재명명이 이루어진다. 이와 같은 메커니즘을 통해 RPQ' 알고리즘은 기존의 P개의 큐보다 2배 많은 2P개의 큐를 사용하여 패킷의 우선 순위가 역전되는 문제점을 해결한다.

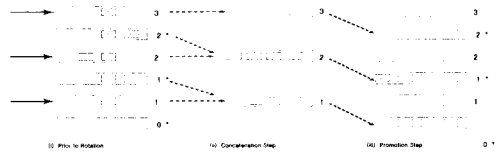


그림 3 RPQ+ 스케줄러의 동작 과정

3. 제안된 패킷 스케줄링 알고리즘

3.1 제안된 스케줄러의 개요

엔터프라이즈 환경에서 각 애플리케이션 클래스별로 요구되는 QoS 파라미터에 따른 패킷 지연 시간이 고려된 스케줄링 알고리즘 및 오버플로우 버퍼를 이용하여 스케줄러에 의해 서비스가 거부된 패킷들을 버퍼 관리 정책에 따라 효과적으로 패킷 스케줄링을 처리하는 방법을 고려한다. 또한 RPQ' 알고리즘이 2P개의 큐를 사용하는 오버헤드를 줄이기 위해 P개의 큐로서 기존의 방법과 동일한 큐의 개수를 사용하면서 우선 순위의 역전이 발생하지 않도록 하는 방법을 설명한다. 제안된 스케줄러의 전체적인 개요는 그림 4와 같다.

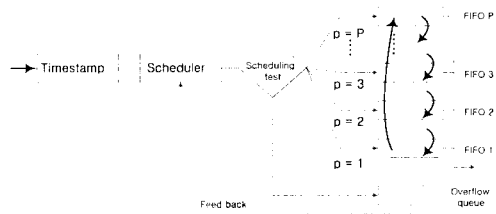


그림 4 패킷 스케줄러 구조

그림 4에서와 같이 스케줄러에 도착하는 패킷은 타임스탬프가 기록되는데 이러한 타임 스텝프는 스케줄러가 패킷에 부여하는 우선 순위의 정책에 따라 마감시한이나, 가상 종료 시간 등의 값을 가질 수 있다. 패킷 전송은 우선 순위가 가장 높은 0번 큐부터 패킷이 서비스를 대기하고 있는 큐를 전송한다.

제안된 스케줄링 알고리즘의 의사코드는 그림 5와 같다. 엔터프라이즈 환경에서의 애플리케이션은 그 특성별

1. for all queue
2. allot queue each class and calculate schedule time;
3. calculate rotate time(Δ);
4. check overflow queue;
5. if available slack time then
6. reschedule;
7. else
8. class 1, 4 packets allot into overflow queue;
9. class 2, 3 packets discard;
10. check scheduling test;
11. if available scheduling time then
12. allot into proper queue;
13. else
14. allot into overflow queue;
15. queue_rotate;
16. for all queue head
17. head = NULL;

그림 5 제안된 스케줄링 알고리즘

로 2.1절에서와 같이 4가지의 클래스로 크게 구분하면 클래스의 번호가 다수의 FIFO 큐가 존재할 때 FIFO 큐에 배치되는 범위를 결정하며 동일 클래스의 큐에서의 다수개의 큐간에 우선 순위는 타임 스텝에 따라 정해지게 된다. 그림 5의 알고리즘에서 스케줄러는 각 클래스별 애플리케이션이 요구하는 QoS 파라미터에 따라 각 큐의 패킷 서비스 시간 $Q(t)$ 를 할당한다(라인 1-2). 동일한 클래스의 애플리케이션 패킷은 가장 적은 지연 시간의 제한을 가지는 애플리케이션의 패킷이 우선 순위가 높은 큐에 할당된다. 스케줄러가 클래스별 큐의 개수와 각 큐에 패킷 서비스 시간의 할당은 기본적으로 가용한 큐의 개수를 클래스 1과 4, 클래스 2와 3에 $\frac{1}{2}$ 씩을 할당 한 후 시스템의 애플리케이션 부하 여부에 따라 클래스 1과 4, 클래스 2와 3은 패킷이 할당되지 않은 빈 큐의 개수의 $\frac{1}{2}$ 씩을 교환하여 사용한다. 스케줄러는 클래스 1에 해당하는 임계 작업 애플리케이션에 대해 패킷 지연 시간을 우선적으로 고려하여 큐의 회전 시간(Δ)을 동적으로 정하므로 응답 시간을 서비스 특성으로 요구하는 애플리케이션을 지원한다(라인 3). 제안된 스케줄러가 P개의 큐를 관리한다고 가정하면 패킷 스케줄링이 가능하기 위해서는 (1)의 조건을 만족해야 한다.

$$\Delta \geq \sum_{i=1}^P Q_i(t) \quad (1)$$

(1)의 조건을 만족하여야 가장 높은 우선 순위를 가지는 큐(P[0])가 전송할 패킷이 남은 채로 큐가 회전하여 0에 해당하는 우선 순위의 패킷이 가장 낮은 p-1의

우선 순위를 가지는 문제가 발생하지 않으며 각 클래스별 패킷들의 지연 시간이 제한 될 수 있다.

큐의 회전 시간이 정해 짐에 따라 p번 큐에 삽입되는 패킷의 지연 시간은 (2)와 같이 제한된다.

$$\sum_{i=1}^p \Delta \leq S_p \leq \sum_{i=1}^p \Delta \quad (2)$$

이러한 패킷의 지연 시간이 제한되는 상황이 지켜지기 위해서는 스케줄러는 새로 도착하는 패킷에 대해 (3)과 같은 스케줄링 가능성 여부를 테스트하여 각각의 큐에 삽입시켜야 한다(라인 10-14). 새로 도착한 패킷이 p번 큐에 삽입되기 위해 (3)의 조건을 만족하면 스케줄링이 가능하다.

$$A_{new}(t) \leq Q_p(t) - \sum_{i \in R(p)} A_i(t) \quad (3)$$

(3)에서 $A(t)$ 는 패킷 전송 시간이고, $Q_p(t)$ 는 큐에 할당된 패킷 서비스 시간이다. 스케줄 가능성 테스트에 따라 큐에 삽입되지 못한 패킷은 오버 플로우 큐에 삽입되어 큐의 회전 직후에 우선적으로 스케줄링을 서비스 하여준다. 스케줄링 컨트롤러를 통해 패킷 전송 서비스가 거부된 패킷들을 위해 추가적으로 오버 플로우 큐를 관리하여 엔터프라이즈 환경에서 각 멀티미디어 애플리케이션 특성에 따른 오버 플로우 패킷 처리를 수행한다(라인 4-9). 스케줄러가 오버 플로우 큐를 추가적으로 관리함으로써 패킷 손실율을 서비스 특성으로 요구하는 애플리케이션의 서비스를 지원한다. 일정한 패킷 손실율을 허용하는 애플리케이션의 패킷들은 시스템이 과부하 상태로 인해 패킷 서비스가 거부되면 QoS 파라미터로 제공되는 패킷 손실율 정보에 따라 스케줄러가 오버 플로우 큐내의 패킷을 폐기 처리한다. 또한 큐의 회전 직후에 오버 플로우 큐내에 패킷을 우선적으로 스케줄링함으로써 우선 순위가 더 낮은 큐들에게 할당된 서비스 시간을 남긴 큐들의 여유 시간(slack time)을 이용하여 효율적인 스케줄링 서비스를 지원하여 시스템 자원을 충분히 활용하는 효과를 나타낸다.

3.2 제안된 스케줄러의 동작 과정

그림 6은 제안된 스케줄러의 한 단계 큐의 회전 동작 과정을 보인 것 이다. 스케줄러는 패킷의 삽입 위치를 나타내는 trail 포인터를 큐의 회전 후에 조정하여 RPQ가 2배 만큼의 큐를 사용하여 패킷의 우선 순위가 역전 되는 현상을 해결하는 비용을 기존의 큐의 개수로써 해결하는 방법을 보여준다.

그림 6 (a)는 Δ 를 큐회전 주기로 가지는 스케줄러의 큐의 회전 전 상태이다. F(p-1)의 마지막 패킷은 F(p)의 최초 패킷과 포인터로 연결이 되어 있으므로 패킷 전송은 우선 순위가 가장 높은 F(0)부터 스케줄러와는

독립적으로 실행될 수 있다. 스케줄러는 추가적으로 현재 진행되는 스텝전까지의 단계에서 패킷 전송 서비스가 거부된 패킷들을 위해 오버 플로우 큐를 관리한다.

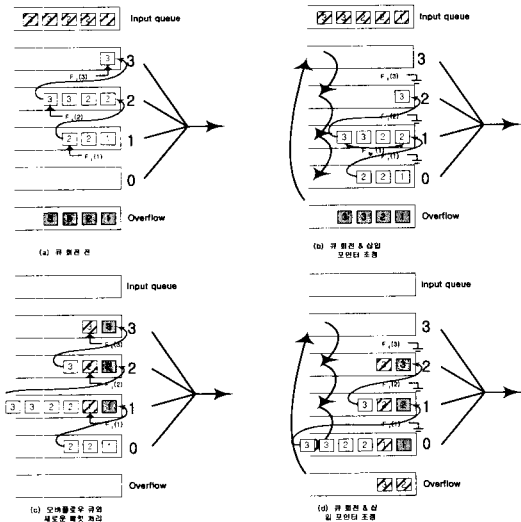


그림 6 제안된 스케줄러의 동작 과정

그림 6 (b)의 큐의 회전 및 삽입 포인터 조정 단계에서는 $F(p)$ 의 큐는 $F(p-1)$ 의 큐로 재명명되고 큐의 trail로 표시되는 삽입 포인터 $F_i(p)$ 는 큐의 회전 뒤에 우선 순위 p 를 가지는 패킷들의 우선 순위 $p-1, \dots, P-1$ 를 가지는 패킷들보다 먼저 서비스를 받도록 스케줄링 되기 위하여 NULL로서 표시된다.

그림 6 (c)에서는 새로이 우선 순위 1을 가지는 패킷 1개와 우선 순위 2를 가지는 패킷 2개, 우선 순위 3을 가지는 패킷 2개가 스케줄러에 도착된 상황에서 오버 플로우 큐 및 새로운 패킷의 스케줄링을 나타낸다. 오버 플로우 큐에 패킷을 새로운 패킷의 스케줄링보다 우선적으로 실시한다. 그림 6 (b)에 $F_i(p)$ 로 나타내어 지는 시간은 $\sum_{i \in F(p)} A_i(t)$ 이고 오버 플로우 큐에 있는 각 클래스에 있는 패킷들이 스케줄링 되기 위해서는 (4)의 조건을 만족해야 한다.

$$A_1(t) \leq \sum_{i=1}^{p-1} Q_i(t) - \sum_{i \in F(p)} A_i(t) \quad (4)$$

오버 플로우 큐의 클래스 1의 패킷의 전송시간이 (4)를 만족하면 $F(1)$ 큐에 패킷을 삽입하여 스케줄링 서비스를 수행하고 스케줄링이 불가능 할 시에는 애플리케이션 클래스의 특성에 따라 다시 오버 플로우 큐에 삽입되어 다음 스텝에서 서비스를 실시한다. 이는 엔터프

라이즈 환경에서 클래스 1에 해당하는 애플리케이션이 패킷 전송상의 손실이 없는 서비스를 요구하기 때문이며, 클래스 2나 3과 같은 애플리케이션은 애플리케이션 특성에 따라 일정한 패킷 손실에 대한 포용율(loss tolerance rate)에 따라 패킷을 폐기한다. 그림 6 (c)에서는 오버플로우 큐에서의 우선 순위 패킷 3을 패킷 손실 포용율에 따라 폐기하여 스케줄링한 결과이다. 스케줄러는 오버 플로우를 관리하는 추가적인 오버헤드가 있지만, 각각의 큐에 할당된 패킷 서비스 시간에 따라 패킷 스케줄링 컨트롤러에 의해 서비스가 거부된 패킷들을 다음 스케줄링 스텝에서 가능한 서비스 시간을 고려하여 최대한 스케줄링 서비스하여 주는 동시에 애플리케이션별로 요구하는 서비스 특성에 따라 패킷을 폐기하거나, 계속하여 패킷전송을 위해 오버 플로우 큐를 관리 할 수 있다.

그림 6 (d)는 오버 플로우 큐의 패킷들을 처리하고, 새로운 패킷들을 스케줄링한 이후의 큐의 상태를 나타낸다.

4. 시스템 성능 평가 분석

표 2는 제안된 스케줄링 알고리즘과 기존 SP, EDF, RPQ⁺의 특성을 비교 분석한 것이다.

표 2에서와 같이 EDF와 SP는 정렬된 우선 순위 큐를 유지하기 위해 힙(heap) 알고리즘을 사용하면 복잡도는 $O(\log N)$ 으로 나타내어 진다[6]. RPQ⁺ 알고리즘과 제안된 알고리즘의 복잡도는 $O(N)$ 으로 동일하며 필요

표 2 제안된 스케줄러와 기존 스케줄링 알고리즘과의 비교

알고리즘 구분	SP	EDF	RPQ ⁺	제안된 스케줄러
알고리즘 복잡도	$O(N \log N)$	$O(N \log N)$	$O(N)$	$O(N)$
큐정렬비용(N개 패킷)	$O(\log N)$	$O(\log N)$	$O(1)$	$O(1)$
필요한 큐	1	1	2P	P + 1
기아 현상	○	×	×	×
마감시한 고려	×	○	○	○
애플리케이션 특성 고려	△	×	△	○
패킷 사이즈 고려	×	○	×	○

한 큐의 개수에서 제안된 스케줄러는 추가적으로 오버플로우 큐를 관리하므로 P + 1개 이고, RPQ⁺ 알고리즘이 2P개의 큐를 사용하므로 알고리즘의 주요 부분인 우선 순위 큐 관리 비용이 $\frac{(P+1)O(N)}{(2P)O(N)}$ 으로 줄어드는 개선 효과가 있다. 제안된 스케줄러는 엔터프라이즈 환경하에서 애플리케이션을 클래스로 구분하여 스케줄링을 제공한다. 제안된 스케줄러의 스케줄링 어드미션 컨트롤러는 각 큐에 패킷을 스케줄링할 때 패킷의 처리 시간을 고려한 스케줄링 서비스를 제공한다.

표 3 실험 조건

클래스	지연 시간 제한	손실율
1	20ms	손실율 0%
2	50ms	손실율 1%
3	100ms	손실율 5%
4	∞	

실험 환경을 구성하기 위해 패킷 i는 패킷 클래스, 마감시간, 처리시간, 클래스 정보의 4가지 QoS 파라미터를 갖는다. 패킷 클래스 C_i는 엔터프라이즈 환경에서 애플리케이션 특성별로 구분한 클래스 ID로서 본 논문에서는 크게 4가지로 구분하였다. 마감시간 D_i는 패킷이 마감 시간을 넘기지 않으면서 전송되어야 할 최만 시간(latest time)을 나타낸다. 클래스 4와 같이 마감 시간이 없는 애플리케이션의 패킷들에게는 D_i값은 ∞를 가진다. 처리시간 P_i는 패킷을 전송하는데 소요되는 시간을 나타내며 패킷 크기의 전송에 필요한 유휴 시간을 포함한다. 클래스 정보를 나타내는 I_i는 엔터프라이즈 환경에서 애플리케이션이 요구하는 특성을 나타내며 패킷 손실이 없는 신뢰성 있는 스케줄링 어드미션 컨트롤, 전진에러 수정, 연성 실시간, 경성 실시간, 패킷 손실 포용률 값을 가질 수 있다. 예를 들어 패킷 손실 포용률은 인코딩 기법과 압축 기법에 따라 다르지만 음성 스트림 패킷은 서비스 품질을 크게 떨어뜨리지 않는 최대 패킷 손실 포용률은 1%~5% [7]로 설정할 수 있다.

스케줄링 테스트를 위해 고려한 환경은 다음과 같다. 출력 링크의 전송율은 100Mbit/s이며, 각 애플리케이션 클래스별 지연 시간의 제한은 표 3과 같다. 클래스 1의 임계 작업은 지연 시간의 제한이 엄격히 준수되는 경성 실시간의 특성을 요구하고 패킷의 손실은 허용되지 않는다. 클래스 2와 3의 애플리케이션은 50~100ms의 지연 시간 제한의 경성 실시간 또는 연성 실시간의 특성

을 가지면서 패킷 손실율은 1~5%를 허용한다. 클래스 4의 일반적인 작업에는 지연 시간의 제한이 준수되어야 하는 특성은 없다.

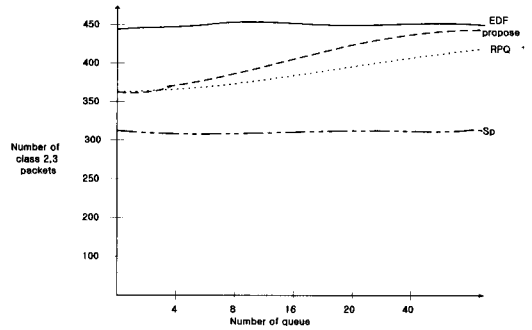


그림 7 큐의 개수 변화에 따른 스케줄링 효과 비교

그림 7은 큐의 회전 시간(Δ)를 10ms로 하여 큐의 개수를 변화 시켰을때의 클래스 2와 3에 해당하는 패킷들의 스케줄링 가능한 양을 나타낸 것이다. 큐의 개수가 8개를 기준으로 하여 RPQ⁺ 알고리즘보다 개선된 성능을 보이기 시작하며 이는 RPQ⁺ 알고리즘이 기본적으로 스케줄링을 위해 2P개의 큐를 사용하고 제안된 스케줄러는 P+1개의 큐를 사용하므로 큐 관리 비용의 감소 효과이다. 그림 5에서 볼 수 있듯이 큐의 개수를 30개 이상으로 사용하면 EDF 알고리즘의 성능에 근접한 효과를 보임을 알 수 있으므로, 스케줄러의 큐의 회전 시간(Δ) 변화에 따른 스케줄링 성능을 평가하기 위해서는 30~60개의 큐를 사용하여 스케줄링의 영향을 측정한다.

그림 8은 SP, EDF, RPQ⁺, 제안된 스케줄러의 패킷 전송 효율을 비교한 것이다. Y축으로 나타내어지는 스

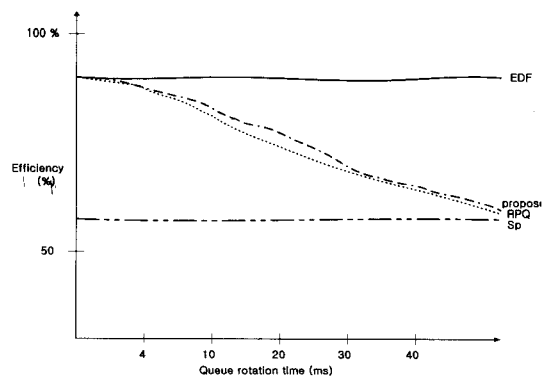


그림 8 스케줄링 효율 비교

케줄러의 효율은 총 전송 패킷량에 대한 애플리케이션 클래스의 요구 사항에 따라 마감 시간을 넘기지 않거나 패킷 손실 없이 효과적으로 스케줄링된 패킷량의 비율을 나타낸다. X축은 RPQ 스케줄러와 제안된 스케줄러의 회전 시간(Δ)에 따른 변화를 나타낸다. 결과를 보면 $\Delta \leq 4ms$ 에서부터 EDF와 근접한 성능을 보이면서 $\Delta \leq 10ms$ 에서 RPQ보다 우수한 효율을 보임을 알 수 있다. 이는 RPQ와는 달리 오버플로우 큐에서 패킷 손실 포용률에 따라 총 전송 패킷량이 줄어들며 따라 전체적인 스케줄링 효율의 개선 결과이다.

5. 결론 및 향후 과제

네트워크 대역폭이 커지고 다양한 요구 사항을 가지는 애플리케이션이 증가함에 따라 네트워크는 패킷 전송 지연 시간을 제한하여 주는 기능이 필수적으로 필요하다. 이러한 상황에서 네트워크 버퍼에서 스위치로나 네트워크 접근 포인트로의 패킷 전송 순서를 결정하는 서비스 정책이나 스케줄링 기법이 패킷의 지연 시간을 조절하여 애플리케이션의 다양한 요구 조건을 만족시키는 네트워크 알고리즘으로서 중요한 역할을 수행한다.

본 논문에서는 패킷의 지연 시간을 제한하는 알고리즘중 근사 정렬된 우선 순위 큐 알고리즘이 큐의 회전에 따른 패킷의 우선 순위 역전 현상을 해결하기 위해 2P개의 큐를 사용하므로, 큐를 관리하기 위한 오버헤드를 증가시키는 문제점을 제기하고, 패킷의 우선 순위 역전 현상을 해결하기 위해 필요한 큐의 개수를 P개로 줄였다. 또한 엔터프라이즈 환경에서 애플리케이션을 그 특성별로 클래스로 구분하여 패킷 서비스 응답 시간, 패킷 손실율 등의 서비스 특성을 요구하는 애플리케이션의 스케줄링 서비스를 제공한다. 추가적인 오버플로우 큐를 관리함으로써 패킷 전송 효율을 높였으며, 패킷 어드미션 컨트롤러를 통해 각 클래스 별로 요구되는 서비스 지연시간의 제한을 만족 시킬 수 있었다. 향후 연구 과제로는 애플리케이션의 특성별로 분류한 클래스를 좀더 세분화하고 각 클래스의 네트워크 부하 상태에 따라 각 큐마다의 패킷 서비스 시간을 관리하는 패킷 어드미션 컨트롤러의 동작을 동적으로 조절하며 스케줄러가 좀더 세분화된 애플리케이션의 네트워크 QoS 파라미터를 지원하고 오버플로우 큐 관리 정책에 대한 사용자와의 상호 작용에 의한 지원이다.

참고 문헌

[1] J. Liebeherr, and D. E. Wrege, "Priority queue schedulers with approximate sorting in output-

buffered swithes," IEEE Trans. Commun., vol. 17, NO. 6. June 1999

- [2] J. Liebeherr, D. E. Wrege, and D. Ferrari, "Exact admission control in networks with bounded delay services," IEEE/ACM Trans. Networking, vol. 4, pp. 885-901, Dec. 1996
- [3] J. M. Peha and F. A. Tobagi, "Cost-based scheduling and dropping algorithms to support integrated services" IEEE Trans Commun., vol. 44. NO. 2. Feb 1996
- [4] J. Hyman, A. A. Lazar, and G. Pacifici, "MARS: The magnet II real-time scheduling algorithm," in Proc. ACM. Sigcomm, sept. 1991, pp. 285-93.
- [5] R. Chipalkatti, J. F. Kurose, and D. Towsley, "Scheduling policies for real-time and nonreal-time traffic in a statistical multiplexer," in Proc. IEEE Infocom. Apr. 1989, pp. 774-83.
- [6] Donald E. Kunth, The Art of Computer Programming. Volume 3: Sorting and Searching, Reading, MA, Addison-Wesley, 1975
- [7] Junji Suzuki and Mashiro Taka, "Missing Packet Recovery Techniques for Low-Bit-Rate Coded Speech," IEEE Journal on Selected Areas in communications, vol. 7, No. 5, pp. 707-717, June 1989
- [9] J. M. Peha and F. A. Tobagi, "Evaluating scheduling algorithms for traffic with heterogeneous performance objectives," in Proc. IEEE Globecom-90. Dec. 1990, pp. 21-27
- [10] K. Sriram, "Dynamic bandwidth allocation and congestion control schemes for voice and data multiplexing in wideband packet technology," in Proc. IEEE ICC-90, Apr 1990, pp. 1003-9
- [11] 유항재, 김태윤 "엔터프라이즈 환경에서의 멀티미디어 QoS 프레임워크," 고려대학교 석사학위논문, 1999

김 병 철

1999년 고려대학교 컴퓨터학과 졸업(이학사). 1999년 ~ 현재 고려대학교 컴퓨터학과. 석사과정 재학중. 관심분야는 컴퓨터 네트워크, 실시간 시스템, 이미지 프로세싱.

김 태 윤

정보과학회논문지:정보통신
제 27 권 제 1 호 참조