

고속 패킷망에서 효율적인 가상 시작 시간 기반 패킷 스케줄링 알고리즘

정희원 고 남 석*, 꺾 동 용*

Efficient Packet Scheduling Algorithm using Virtual Start Time for High-Speed Packet Networks

Nam-Seok Ko*, Dong-Yong Kwak* Regular Members

요 약

본 논문에서는 Minimum Possible Packet Fair Queueing (MPSFQ)로 명명되는 효율적이면서 구현의 관점에서 용이한 패킷 스케줄링 알고리즘을 제안한다. MPSFQ 알고리즘은 $O(1)$ 의 가상 시간 계산의 복잡도를 가지며 WFQ (Weighted Fair Queueing) 알고리즘과 동일한 지연 특성 및 우수한 fairness 특성을 가지고 있다. MPSFQ의 중요한 특징은 시스템 가상 시간은 패킷의 종료 시간에 현재 전송될 패킷이 있는 세션의 최 선두에 있는 패킷들의 가능한 가상 시간의 값 중 최소의 값으로 시스템 가상 시간의 재조정이 이루어지거나 시스템 가상 시간의 재조정 시간의 중간에 도착한 패킷의 시스템 가상 시간은 이전 시스템 가상 시간에 두 시간 사이의 실시간을 더함으로써 계산된다. 제안하는 알고리즘이 위와 같은 간단한 알고리즘을 이용하면서도 우수한 지연 특성 및 fairness 특성을 가짐을 수학적 분석을 통해 증명하고 시뮬레이션을 통하여 그 결과를 확인한다.

ABSTRACT

In this paper, we propose an efficient and simple fair queueing algorithm, called Minimum Possible Virtual Start Time Fair Queueing (MPSFQ), which has $O(1)$ complexity for the virtual time computation while it has good delay and fairness properties. The key idea of MPSFQ is that it has an easy system virtual time recalibration method while it follows a rate-proportional property. MPSFQ algorithm recalibrates system virtual time to the minimum possible virtual start time of all backlogged sessions. We will show our algorithm has good delay and fairness properties by analysis.

I. 서 론

고속 통합 서비스 패킷 망은 각 스위치 및 라우터의 패킷 스케줄링 알고리즘을 통하여 다양한 Quality-of-Service (QoS)를 제공한다. 스케줄링 알고리즘으로 제공할 수 있는 QoS의 측정 변수로는 단대단 지연 (end-to-end delay), 지연 지터 (delay jitter), fairness 등이 존재한다. 스케줄링 알고리즘의 기본적인 기능은 스

위치의 각 출력 링크에 대하여 동일한 출력 링크를 공유하는 세션에 속하는 가용한 패킷 중 다음에 전송될 패킷을 선택하는 것이다.

지금까지 많은 패킷 스케줄링 알고리즘이 제안되어 왔다^{[1]-[7]}. 그들 중 Weighted Fair Queueing (WFQ)^[2]이 지연 특성 및 fairness 특성에 있어서 이상적인 트래픽 스케줄링 알고리즘으로 알려져 있다. 그러나 N 개의 세션을 서비스하는 WFQ 스케줄러의 timestamp를 계산하

* 한국전자통신연구원 네트워크기술연구소 (nsko@etri.re.kr),
논문번호 : 020361-0822, 접수일자 : 2002년 8월 7일

는 복잡도는 $O(N)$ 으로서 구현상의 많은 어려움이 존재한다. 따라서 최근에는 Self-Clocked Fair Queueing (SCFQ)^[3], Frame-based Fair Queueing (FFQ)^[4], Starting Potential Fair Queueing (SPFQ)^[4], Minimum Delay Self-Clocked Fair Queueing (MD-SCFQ)^[6] 등의 알고리즘들이 WFQ 알고리즘의 복잡도를 해결하면서 우수한 성능을 유지하고자 하는 목적에서 제안되어 왔다.

SCFQ는 가상 시간 계산이 시스템 내에 내재되어 있기 때문에 훨씬 간단하게 구현이 가능하지만 각 세션들 간에 독립성을 제공하지 못하기 때문에 지연 특성이 좋지 않은 단점을 가지고 있다. SPFQ는 WFQ와 동일한 지연 특성을 가지며 WFQ와 유사한 fairness 특성을 가지지만 매 패킷의 서비스가 끝나는 시점마다 각 세션에 속하는 패킷의 가상 시작 시간이 가장 작은 세션을 선택하여야 하기 때문에 각 세션의 Head-of-Line (HOL) 패킷에 대한 정렬을 필요로 한다. 결국 부가적으로 $O(\log N)$ 의 복잡도를 유발한다. MD-SCFQ 역시 좋은 지연 특성 및 fairness 특징을 가지고 있지만 매 패킷이 전송될 때마다 각 세션의 HOL에 있는 패킷의 가상 시작 시간의 가중 평균을 구해야 하는 단점을 가지고 있다.

본 논문에서는 Minimum Possible Packet Fair Queueing (MPSFQ)로 명명된 새로운 패킷 스케줄링 알고리즘을 제안한다. MPSFQ 알고리즘은 $O(1)$ 의 가상 시간 계산의 복잡도를 가지며 WFQ와 동일한 지연 특성을 가지며 우수한 fairness 특성을 가진다.

본 논문은 다음과 같이 구성되어 있다. II장에서는 기존의 패킷 스케줄링 알고리즘을 비교 분석한다. III장에서는 제안된 알고리즘의 개념 및 동작 과정을 설명하며 지연 특성 및 fairness 특성을 분석한다. IV장에서는 지연 특성에 대하여 시뮬레이션을 통하여 확인한다. 마지막으로 V장에서 결론을 맺는다.

II. 패킷 스케줄링 알고리즘

1. 배경

본 논문에서 다루는 패킷 스케줄링 알고리즘은 기본적으로 각 패킷이 시스템에 도착할 때

각 알고리즘 별로 유지하고 있는 시스템 가상 시간 (또는 시스템 포텐셜)에 기반 하여 계산된 timestamp를 이용하여 서비스의 순서를 결정한다. 이 때 기준 시간으로 사용하는 시스템 가상 시간은 시스템이 한 일의 양을 나타내기 위해서 사용되며 시스템의 기준 시간으로서 지연 특성 및 fairness 특성을 결정하는 중요한 역할을 한다. Timestamp는 가상 종료 시간 (virtual finish time)이라고도 하며 일반적으로 시스템에서 각 세션이 예약한 속도로 서비스를 하게 될 경우 서비스를 마칠 수 있는 시간을 의미한다. 패킷에 대한 서비스 순서는 timestamp의 증가 순으로 결정된다. 다시 말해서, 패킷은 서비스를 위해서 큐에 놓이기 전에 timestamp를 할당 받으며 서버는 보낼 패킷이 있는 (이하 백로그라는 용어를 사용) 각 세션의 HOL 패킷들 중 가장 작은 timestamp를 가지는 패킷을 서비스 하게 된다.

일반적으로 각 패킷 스케줄링 알고리즘의 성능을 비교하기 위한 척도로서 지연 특성과 fairness 지수를 사용한다. 지연 특성으로는 latency와 단대단 지연(end-to-end delay)을 사용한다. Latency는 전송할 패킷이 존재하지 않는 세션에 새로운 패킷이 들어왔을 때 그 패킷이 서비스를 마치는데 까지 걸리는 시간을 의미하며 단대단 지연은 동일한 스케줄러를 가지는 N 개의 노드로 구성되는 망에서 송신 노드에서 수신 노드까지 패킷이 전송되는데 걸리는 시간을 의미한다. 이 때 단대단 지연의 경우는 일반적으로 leaky-bucket에 의해서 패킷이 조절되는 상황에 대한 지연을 계산한다. Fairness 지수는 임의의 두 세션이 계속적으로 백로그되어 있는 임의의 시간 구간에서 두 세션간에 두 세션의 각 예약된 속도에 의해서 정규화 (이하 정규화된다는 의미는 예약 속도에 의한 정규화를 의미)된 서비스 양의 차이 값으로 정의된다^[3].

본 논문에서 연구하는 스케줄링 알고리즘들의 근간이 되는 알고리즘으로 Generalized Packet Sharing (GPS) 알고리즘이 존재한다. 이 알고리즘은 실제 패킷 시스템에서는 적용이 불가능한 이상적인 알고리즘으로 패킷이 아주 작은 단위로 나누어져 현재 전송할 패킷이 있는 모든 세션들이 동시에 서비스를 할 수 있도록 하고 있다. 이 때 각 세션들은 각 세션이 예약한 속도에 비례하여 서비스가 이루어지며 임의

의 시점에서 지속적으로 전송할 패킷이 있는 두 세션간에 각 세션의 예약 속도에 의해서 정규화된 서비스의 양을 동일하게 유지하는 이른바 완벽한 fairness를 제공하는 알고리즘이다.

Packet-by-packet GPS (PGPS) 로 불리기도 하는 WFQ 알고리즘^[2]은 GPS의 패킷 시스템 버전이다. WFQ는 GPS에서 이루어지는 시스템 가상 시간을 그대로 유지하기 위하여 GPS를 시뮬레이션하며 다음과 같은 기율기를 가진다.

$$\frac{d\tau(t)}{dt} = r / \sum_{i \in B(t)} r_i, \quad (1)$$

여기서 B(t)는 임의의 시점 t에서 백로그된 세션들의 집합이며 r은 서버의 속도이다.

이와 같이 WFQ는 GPS를 시뮬레이션하여 시스템 가상 시간을 유지하기 때문에 지연 특성 및 fairness특성은 GPS의 특성에 매우 근접하게 된다. 하지만 하나의 패킷이 전송되는 시점에서 동시에 N개의 세션이 GPS의 시스템에서 서비스를 마치고 세션을 끊거나 또는 새로운 세션으로 들어올 수 있기 때문에 시스템 가상 시간을 유지하는 최악의 경우의 복잡도는 O(N)이 되어 연결이 많은 실제적인 알고리즘의 구현에 많은 문제점을 가지고 있다. WFQ의 latency는 다음과 같다.

$$\Theta_i^{(WFQ)} = \frac{L_i}{r_i} + \frac{L_{max}}{r} \quad (2)$$

여기서 L_i and L_{max} 는 각각 세션 i로 부터의 패킷의 최대 길이와 시스템 내의 패킷의 최대 길이를 나타낸다.

WFQ의 fairness 지수는 다음과 같다.

$$FI_{WFQ} = \max \left(\frac{L_i}{r_i} + \frac{L_{max}}{r_j} + C_i, \frac{L_j}{r_j} + \frac{L_{max}}{r_i} + C_j \right) \quad (3)$$

여기서 $C_i = \min \left((N-1) \frac{L_{max}}{r_i}, \max_{1 \leq k \leq N} \frac{L_k}{r_k} \right)$

위와 같이 WFQ는 아주 우수한 성능을 보이는 알고리즘이지만 O(N)의 복잡도가 보여 주는 것처럼 구현상에 많은 어려움을 가지고 있다. 따라서 지금 까지 많은 논문들에서 WFQ의 우수한 성능을 유지하면서 WFQ가 가지는 복잡도를 해결하기 위한 많은 시도가 있었다. 그 대표적인 알고리즘으로 SCFQ, FFQ, SPFQ, MD SCFQ 등을 들 수 있다. SCFQ는 시스템

가상 시간으로 현재 전송중인 패킷의 가상 종료 시간을 사용한다. 이는 시스템 가상 시간의 유지를 매우 간단하게 만들었으며 결과적으로 실제적인 구현에 있어서 아주 용이하게 되었다. 또한 fairness 측면에서도 다음과 같이 아주 우수한 성능을 보였다.

$$FI_{SCFQ} = \frac{L_i}{r_i} + \frac{L_j}{r_j} \quad (4)$$

하지만 위의 우수한 fairness 특성과 구현의 용이성에 반해서 SCFQ가 가지는 단점으로 세션들 간의 독립성을 제공하지 못하면서 지연 특성이 세션의 수에 비례를 하는 단점을 가지게 되었다. N개의 세션을 가지는 SCFQ 시스템의 latency는 N에 의해서 선형적으로 증가하는 다음과 같은 값을 갖는다.

$$\Theta_i^{(SCFQ)} = \frac{L_i}{r_i} + (N-1) \frac{L_{max}}{r}, \quad (5)$$

FFQ, SPFQ, MD-SCFQ 알고리즘은 모두 Rate Proportional Server (RPS) 클래스에 속하는 알고리즘이다^[4]. 다음절에서 설명하는 RPS의 기본적인 특징으로부터 알 수 있는 바와 같이 패킷 스케줄링 알고리즘이 RPS 클래스에 속하게 되면 그 알고리즘은 WFQ의 지연 특성과 동일한 지연 특성을 가지게 되며 각 알고리즘마다 시스템 가상 시간을 어떻게 재조정하느냐에 따라서 fairness 지수는 달라지지만 기본적으로 공정한 스케줄링을 하게 된다. FFQ는 모든 백로그된 세션의 가상 시간이 고정된 frame 시간 T를 초과할 때 시스템 가상 시간을 재조정하게 된다. 따라서 fairness의 특성은 frame 시간 T에 의해서 영향을 받게 된다. 하지만 시스템 가상 시간의 계산에 대한 복잡도는 세션의 수와 상관없이 일정한 O(1)을 가진다. FFQ의 fairness 지수는 다음과 같다.

$$FI_{FFQ} = 2T + \max \left(\frac{L_i}{r_i} + \frac{L_j}{r_j} \right) \quad (6)$$

SPFQ는 모든 백로그된 세션들의 HOL 패킷의 가상 시작 시간 값 중 최소 값으로부터 시스템 가상 시간을 재조정한다. WFQ에 가까운 fairness 특성을 가지고 있다. 하지만 모든 세션

의 가상 시작 시간에 의한 정렬 구조를 유지해야 한다는 단점이 있다. 결과적으로 시스템 가상 시간을 유지하는 복잡도는 $O(\log N)$ 이 된다. SPFQ 알고리즘의 fairness 지수는 다음과 같다.

$$FI_{SPFQ} = \max \left(\frac{L_i}{r_i} + \frac{L_j}{r_j} \right) + \max_{1 \leq k \leq N} \frac{L_k}{r_k} + \frac{L_{\max}}{r} \quad (7)$$

MD-SCFQ 알고리즘은 시스템 가상 시간을 백로그된 모든 세션의 가상 시작 시간의 가중 평균으로 제조정을 한다. 이 알고리즘 역시 fairness 특성에 있어서도 우수한 특성을 보이고 있다. 하지만 모든 세션의 가상 시작 시간에 대한 가중 평균을 구해야 한다는 부가적인 복잡도가 존재한다. MD-SCFQ 알고리즘의 fairness 지수는 다음과 같다.

$$FI_{MD-SCFQ} = \max(f_{i,j}, f_{j,i}) \quad (8)$$

여기서

$$f_{i,j} = \frac{L_i}{r_j} + \max \left(\frac{L_{\max}}{r_j}, \max_{1 \leq k \leq N} \frac{L_k}{r_k} - \frac{r_j}{r - r_j} \left(\max_{1 \leq k \leq N} \frac{L_k}{r_k} - \frac{L_i}{r_i} \right) - \frac{L_j}{r} \right)$$

이며 이 값이 의미하는 바는 세션 i 가 받은 서비스 양 보다 세션 j 가 더 많이 받은 서비스의 양을 의미한다.

위에서 살펴본 바와 같이 대부분의 시스템 가상 시간 계산 복잡도를 줄이는 알고리즘은 시스템 성능에서 저하를 가져왔으며 좋은 시스템 성능을 가지는 알고리즘들은 구현 관점에서 복잡하다는 단점을 가지고 있다.

본 논문에서는 우수한 지연 및 fairness 특징을 가지면서도 구현 관점에서 복잡도가 낮은 알고리즘을 제시하고자 한다. 다음 절에서는 제안되는 알고리즘을 바로 설명하기에 앞서 MPSFQ 알고리즘이 기반을 두고 있는 스케줄링 알고리즘의 하나의 프레임워크인 Rate Proportional Server에 관해서 살펴보기로 한다.

2. Rate Proportional Server

Rate-Proportional Servers (RPS)는 Stiliadis와 Varma에 의해서 체계화된 스케줄링 알고리즘의 유체 흐름 모델에 대한 프레임워크이다^[4]. 즉, RPS는 앞에서 설명한 GPS와 같이 실제적

으로 적용이 될 수 없는 알고리즘이다. 하지만 GPS와는 달리 백로그된 세션의 모든 패킷이 동시에 각 예약된 속도에 비례하여 서비스를 받는 것이 아니라 최소 세션 가상 시간을 가진 세션들만이 동시에 각 예약된 속도에 비례하여 서비스를 받게 된다. RPS는 이와 같이 서비스를 여러 개의 세션이 동시에 서비스를 받을 수 있다는 이상적인 알고리즘으로 이에 대한 패킷 시스템 버전이 존재하는데 이를 Packet by Packet RPS (PRPS)라 한다. 패킷 스케줄링 알고리즘이 RPS의 범주에 속하게 되면 그의 PRPS 스케줄러의 지연 특성은 WFQ의 특성과 같게되며 기본적으로 공정한 스케줄링을 한다^[4]. 물론 fairness 지수는 각 알고리즘마다 특별한 가상 시간을 제조정하는 방법에 따라서 차이가 있다.

본 논문에서 제시하는 MPSFQ 알고리즘 역시 RPS의 기본 규칙을 따르고 있기 때문에 먼저 RPS에 대한 기본적인 특징을 알아보도록 한다. RPS는 시스템 가상 시간, 세션 가상 시간, 서비스 규칙의 관점에서 설명될 수 있다.

RPS 시스템에서 세션 가상 시간은 다음 세 가지 특징을 만족해야 한다.

- ① $v_i(t)$ 는 세션 i 가 백로그되어 있지 않은 동안에는 값이 변하지 않는다.
- ② 세션 i 가 τ 에 백로그되면 세션 i 의 가상 시간은 다음과 같다.

$$v_i(\tau) = \max(v_i(\tau^-), v(\tau^-)) \quad (9)$$

여기서 τ^- 는 시간 τ 바로 이전 시간을 나타낸다.

- ③ 세션 i 는 백로그되어 있을 때 그 세션의 세션 가상 시간은 $(\tau, t]$ 구간동안 그 세션에 제공된 정규화된 서비스의 양에 의해서 증가한다.

$$v_i(t) = v_i(\tau) + \frac{W_i(\tau, t)}{r_i} \quad (10)$$

RPS 시스템에서 시스템 가상 시간은 다음 2 가지 조건을 만족해야 한다.

- ① 시스템 바쁜 구간의 임의의 구간 $(t_1, t_2]$ 에서 다음이 성립한다.

$$v(t_2) - v(t_1) \geq t_2 - t_1 \quad (11)$$

- ② 임의의 시점 t 에서 시스템 가상 시간은 결코

모든 백로그된 세션들의 최소 가상 시간을 초과할 수 없다.

$$v(t) \leq \min_{i \in B(t)} v_i(t), \quad (12)$$

여기서 $B(t)$ 는 시점 t 에서 백로그된 세션들의 집합이고 $v_i(t)$ 는 세션 i 의 가상 시간이다.

임의의 시점 t 에서 백로그된 세션들은 다음과 같은 규칙에 따라서 서비스된다.

- ① 백로그된 세션들 중 가장 적은 가상 시간을 가지는 세션들의 집합이 서비스된다.
- ② 그 집합의 각 세션은 각 예약된 속도에 비례하여 서비스되어 각 세션의 가상 시간은 동일한 기울기로 증가된다.

위의 RPS조건을 만족하는 알고리즘으로 SPFQ와 MD-SCFQ 등이 제안되었다. 이 알고리즘들은 기본적인 RPS의 특징을 준수하기 위하여 다음과 같은 절차를 가진다.

□ 시스템 가상 시간은 실시간에 선형적으로 증가를 하며 $\tau_1, \tau_2, \dots, \tau_k$ ($\tau_0 < \tau_1 < \dots < \tau_k$ 이며 τ_0 는 시스템 바쁜 구간의 시작 시간)에 재조정이 이루어진다.

실제 시스템에서 재조정은 각 패킷이 서비스가 끝나는 시점에 해당한다. 즉, 시스템 가상 시간은 매 패킷의 서비스가 끝나는 시점마다 재조정이 이루어진다.

□ 시스템 가상 시간은 임의의 시스템 가상 시간 재조정 시간인 τ_j 에서 다음과 같이 계산된다.

$$v(\tau_j) = \max(v(\tau_j^-), SP(\tau_j))$$

여기서 $v(\tau_j^-) = v(\tau_{j-1}) + \tau_j - \tau_{j-1}$, $v(\tau_0) = 0$ 이고 $SP(\tau_j)$ 는 각 알고리즘마다 특정한 non-decreasing 함수이다.

□ 기본적으로 위의 규칙은 시스템 바쁜 구간에서 정의된다. 따라서 시스템이 idle한 상태가 되면 시스템 가상 시간은 0으로 초기화되며 시스템 바쁜 구간이 다시 시작되면 시스템 가상 시간은 위의 규칙에 따라서 다시 계산된다.

III. MPSFQ 알고리즘

MPSFQ 알고리즘의 기본 아이디어는 기본적인 절차는 PRPS의 절차를 준수하면서 시스템 가상 시간이 각 패킷의 전송이 끝나는 시점에서 백로그된 세션들이 가질 수 있는 최소 가능한 가상 시작 시간으로 재조정되도록 한다는 것이다. 여기서 최소 가능한 가상 시작 시간을 사용하는 이유는 SPFQ에서와 같이 현재 백로그되어 있는 세션으로부터 실제 가장 작은 값을 구하기 위해서 별도의 정렬 구조를 필요로 하는 단점을 해결하기 위함이다. 패킷 전송이 방금 끝나고 새로운 패킷을 선택하게 되면 그 패킷은 현재 백로그되어 있는 세션의 최소 가상 종료 시간이 되므로 그 값에서 각 세션의 최소 가능한 값은 쉽게 얻을 수 있다.

위와 같이 복잡하지 않은 알고리즘을 사용하지만 우수한 성능을 보이고 있다. MPSFQ는 PRPS의 특성을 가진다. 패킷 스케줄링 알고리즘이 PRPS 특성을 가지며 입력 트래픽이 리키 버킷에 의해서 제어될 경우 지연 특성은 이미 분석이 되었으며 그 한계 값은 WFQ의 지연 한계 값과 같다. 또한 시스템 가상 시간을 재조정하는 방법에 따라서 다르지만 기본적으로 fairness가 어느 정도 보장이 된다. 다음 이어지는 절에서 MPSFQ 알고리즘의 상세 절차를 설명하고 MPSFQ 알고리즘의 RPS 특성을 증명하여 지연 특성을 설명하도록 하고 fairness를 분석하여 우수한 fairness 지수를 가짐을 보인다.

1. MPSFQ 알고리즘의 서비스 방식

MPSFQ 알고리즘에서 GPS와 연관된 스케줄링 알고리즘과 같이 패킷의 도착 시에는 timestamp를 할당 받는다. Timestamp를 할당 받는 절차는 다음과 같다.

$$F_i^k = \max(F_i^{k-1}, v(a_i^k)) + \frac{l_i^k}{r_i} \quad (13)$$

여기서 F_i^k 는 세션 i 의 k 번째 패킷의 길이이고 r_i 는 세션 i 의 예약된 속도이다.

시스템 가상 시간을 유지하는 방법에 있어서는 기본적인 PRPS의 규칙을 따르며 시스템 가상 시간의 재조정은 현재 설정된 세션들의 가상 시간 중 최소 가능한 값으로 이루어진다. 최소 가능한 값은 각 세션의 최대 패킷 길이를 세션의 예약된 속도로 나누어진 값 즉, 각 세

선의 최대 패킷을 각 세션의 예약된 속도로 전송을 할 경우 걸리는 시간을 전송을 위해 방금 선택한 패킷의 가상 종료 시간에서 빼주면 쉽게 얻을 수 있다. 패킷의 도착 할 때 및 전송이 끝날 때 MPSFQ 알고리즘의 처리 절차는 다음과 같다.

[패킷의 도착 시 절차]

- 패킷이 t 시점에 도착하게 되면 가장 최근에 재조정된 시스템 가상 시간에 그 시간 이후로 경과한 실시간을 더함으로써 시스템 가상 시간을 계산한다.

$$temp_v(t) = v(\tau_{j-1}) + (t - \tau_{j-1}),$$

여기서 $\tau_{j-1} \leq t < \tau_j$, $\tau_0 = 0$ and $v(\tau_0) = 0$.

- 패킷의 가상 종료 시간(timestamp)을 계산한다.

$$\square F_j^k = S_j^k + L_j^k / r_j$$

여기서 $S_j^k = \max(F_j^{k-1}, temp_v(t))$

- 패킷을 timestamp를 붙여서 해당 큐에 넣는다.

[패킷 전송의 완료 시 절차]

- 방금 전송이 끝난 시점까지 패킷 도착 시와 같이 바로 이전 시스템 재조정 시간 이후로 경과한 실시간을 이용하여 시스템 가상 시간을 계산한다.

$$\square v(\tau_j^-) = v(\tau_{j-1}) + (\tau_j - \tau_{j-1})$$

각 세션의 HOL 패킷들 중 가장 작은 가상 종료 시간(F_{cur})을 가지는 패킷을 선택한다.

다음과 같이 시스템 가상 시간을 재조정한다.

$$\square v(\tau_j) = \max(v(\tau_j^-), F_{cur} - \max_{k \in B(\tau_j)} \frac{L_j^k}{r_j})$$

여기서 $F_{cur} - \max_{k \in B(\tau_j)} \frac{L_j^k}{r_j}$ 가 의미하는 바는 현재 설정된 세션들의 최소 가능한 가상 시작 시간을 의미하게 된다.

위에서 설명한 절차에서 알 수 있는 바와 같이 MPSFQ 알고리즘의 시스템 가상 시간을 계산하는 복잡도는 현재 설정되어 있는 세션의

수와 상관없는 $O(1)$ 이 되며 시스템 가상 시간을 재조정하기 위해서 유지해야 하는 정보는 각 백로그된 세션에서 그 세션이 예약한 속도로 최대 패킷을 전송하기 위해서 소요되는 시간이며 이 정보는 각 세션이 설정 및 해제될 때 쉽게 얻을 수 있는 값이다. 따라서 MPSFQ 알고리즘은 WFQ나 기타 다른 알고리즘에 비하여 적은 정보만을 필요로 한다.

다음 절에서 이와 같이 단순한 방법을 가지며 지연 및 fairness특성이 우수함을 수학적 분석을 통하여 증명하도록 한다.

2. MPSFQ 알고리즘의 성능 분석

2.1 MPSFQ의 RPS 특성

이번 장에서는 MPSFQ 알고리즘의 성능에 대한 수리적인 분석을 제시한다. 일반적으로 스케줄링 알고리즘의 유체 흐름 모델이 RPS의 특징은 그 스케줄링 알고리즘의 성능에 많은 영향을 준다. 본 논문에서 제시하는 MPSFQ 알고리즘 역시 RPS의 범주에 포함되는 알고리즘이기 때문에 먼저 RPS의 기본적인 특징을 살펴 본 후 MPSFQ 알고리즘의 지연 및 fairness에 있어서 성능을 분석하도록 한다.

Lemma 1: τ_n 과 τ_{n+1} 이 두 개의 연속된 시스템 가상 시간 재조정 시간이라고 하면 다음이 성립한다.

$$v_j(\tau_n) \geq v(\tau_n), \forall j \in B(t) \text{ 이면}$$

$$v_j(t) \geq v(t), \forall j \in B(t), \tau_n \leq t < \tau_{n+1} \text{ 이다.}$$

Proof :

MPSFQ 알고리즘의 정의로부터 시스템 가상 시간은 경과한 실시간의 양에 의해서 증가한다. 하지만 임의의 세션의 세션 가상 시간은 그 세션이 받은 서비스의 양에 의해서 증가하게 된다. 이 경우 임의의 시점에서 가장 적은 가상 시간을 가지는 세션들만이 서비스된다는 것을

고려할 때 $v_j(\tau_n) \geq v(\tau_n), \forall j \in B(t)$, 대신에

$v_j(\tau_n) \geq v(\tau_n), \forall j \in M(t)$ 를 고려하면 된다. 여

기서 $M(t)$ 는 시점 t에서 최소 세션 가상 시간을 가지는 세션들의 집합이다.

최소의 가상 시간을 가지는 세션들의 가상 시

간은 그 세션이 $M(t)$ 의 집합에 머무는 한 $r/r_{M(t)}$ 의 기울기로 증가한다. 집합 $M(t)$ 가 다음 시스템 가상 시간의 재조정이 이루어지기 이전에 변경이 생길 수 있지만 그 기울기는 항상 시스템 가상 시간의 기울기보다 크다. 이러한 조건을 가질 때 두 개의 연속된 시스템 가상 시간의 재조정시간 사이에서 시스템 가상 시간은 결코 임의의 세션 가상 시간을 초과할 수 없다. 따라서 Lemma 1의 결과를 얻을 수 있다.

Theorem 1: MPSFQ의 유체 흐름 모델은 Rate Proportional Server이다.

Proof :

패킷 스케줄링 알고리즘의 유체 흐름 모델이 RPS가 되기 위해서는 식 (11)과 식 (12)가 만족되어야 한다. 식 (11)의 조건은 MPSFQ 알고리즘의 정의에 의해서 바로 만족이 되기 때문에 식 (12)에 대한 증명만이 필요하다. .

단계 1: 먼저 식 (12)가 시스템의 바쁜 구간(system busy period)의 시작(τ_0)으로부터 시스템 가상 시간의 첫 번째 재조정시간(τ_1)까지 만족함을 보인다. 즉, 다음에 대한 증명이 필요하다. :

$$v_j(t) \geq \nu(t) \quad \forall j \in B(t), \tau_0 \leq t \leq \tau_1$$

$v_j(\tau_0) = \nu(\tau_0) = 0, \forall j \in B(\tau_0)$ 이 성립하기 때문에 Lemma 1 으로부터 다음과 같은 식이 성립한다.

$$v_j(t) \geq \nu(t), \quad \forall j \in B(t), \tau_0 \leq t < \tau_1.$$

이제 시스템 가상 시간의 첫 번째 재 조정이 일어나는 시간인 τ_1 에서도 여전히 수식이 성립함을 보이면 된다. $\mathcal{S}(\tau_1)$ 을 τ_1 시점에 첫 번째 패킷이 전송을 마치는 세션이라고 하자. 전송된 패킷의 가상 종료 시간을 $F_{\mathcal{S}(\tau_1)}$ 이라고 하면 그 값은 또한 그 시점에서 $\mathcal{S}(\tau_1)$ 의 세션 가상 시간이 된다. 세션 $\mathcal{S}(\tau_1)$ 이 τ_1 시점에서 서비스를 받고 있었기 때문에 $F_{\mathcal{S}(\tau_1)}$ 은 백로그된 세션들의 가상 시간의 최소값이며 모든 다른 세션의

HOL 패킷의 가상 시작 시간보다 값이 크다. 즉 다음이 성립하게 된다.

$$va_j \leq F_{\mathcal{S}(\tau_1)} \leq v_j(\tau_1) \quad \forall j \in B(\tau_1^-) \setminus \{\mathcal{S}(\tau_1)\}$$

여기서 va_j 는 세션 j의 가상 시작 시간이다.

MPSFQ의 정의로부터 시스템 가상 시간은 τ_1 시간까지는 경과된 실시간에 의해서 선형적으로 증가하게 되며 τ_1 에서 경과된 실시간 값과 그 시점에서 세션들의 가상 시간의 가능한 최소의 값 중 더 큰 값으로 시스템 가상 시간이 재조정된다. Lemma 1으로부터 선형적으로 증가한 시간은 결코 세션의 가상 시간을 초과하지 못한다. 따라서 시스템 가상시간은 세션 가상 시간의 최소 가능한 값으로 재조정되게 된다.

따라서 $v_j(t) \geq \nu(t) \quad \forall j \in B(t), \tau_0 \leq t \leq \tau_1$ 이 성립하게 된다.

단계 2: 이제 식 (12)가 τ_{n-1} 까지 만족이 될 경우 τ_n 까지도 여전히 성립함을 보이면 된다. 즉, 다음에 대한 증명을 하면 된다.

$$v_j(\tau_{n-1}) \geq \nu(\tau_{n-1}) \quad \forall j \in B(\tau_{n-1})$$

$$\Rightarrow v_j(t) \geq \nu(t) \quad \forall j \in B(t), \tau_{n-1} < t \leq \tau_n$$

Lemma 1으로부터 이미 $\tau_{n-1} < t < \tau_n$ 에 대하여 위의 수식이 성립함을 알 수 있기 때문에 우리는 다음이 성립함을 보여야 한다.

$$v_j(t) \geq \nu(t) \quad \forall j \in B(\tau_n)$$

$\mathcal{S}(\tau_n)$ 을 τ_n 에서 바로 하나의 패킷에 대한 전송을 마치는 세션이라고 하고 $F_{\mathcal{S}(\tau_n)}$ 을 그 전송된 패킷의 가상 종료 시간이라고 하자.

단계 1에서와 같이 세션 $\mathcal{S}(\tau_n)$ 은 시점 τ_n 에서 서비스를 받고 있었기 때문에 백로그된 세션들 중 최소의 가상 시간을 가지는 세션이다. 즉 다음이 성립한다.

$$F_{\mathcal{S}(\tau_n)} \leq v_j(\tau_n) \quad \forall j \in B(\tau_n)$$

τ_n 시점에서 세션들의 가상 시작 시간이 $F_{\mathcal{S}(\tau_n)}$ 보다 더 클 수 있지만 그 값들은 결코 $v_j(\tau_n)$ 을 초과할 수 없다. MPSFQ 알고리즘이 최소 가능한 가상 시간으로 재조정된다는 것을 상기하면

다음의 수식이 성립함을 분명하게 알 수 있다.

$$v(\tau_n) \leq va_j(\tau_n) \leq v_j(\tau_n)$$

따라서 $v(t) \forall j \in B(t), \tau_{n-1} \leq t \leq \tau_n$ 이 성립하게 된다.

단계 1과 단계 2를 통하여 Theorem 1을 증명하였다.

2.2 MPSFQ의 지연 특성

2.1절에서 MPSFQ 알고리즘의 유체 흐름 모델이 RPS 임을 증명하였다. 따라서 MPSFQ 알고리즘은 WFQ의 지연 특성과 동일한 PRPS의 모든 지연 특성을 가지게 된다. 결과적으로 PRPS의 지연 특성으로부터 MPSFQ 스케줄러의 latency는 다음과 같다.

$$\Theta_i^{(MPSFQ)} = \frac{L_i}{r_i} + \frac{L_{\max}}{r} \quad (14)$$

세션 i 가 r_i 의 요구 속도(reserved rate)를 가지고 σ_i 의 버스트 크기를 가지는 (σ_i, r_i) -리키 버킷에 의해서 제어가 될 때 서버에 도착하는 정보 단위의 양 a_i 는 $\alpha_i(\tau, t) = \alpha_i(t) - \alpha_i(\tau)$ 가 성립하는 구간 $(\tau, t]$ 에서 다음과 같이 한계를 가진다.

$$\alpha_i(\tau, t) \leq \sigma_i + r_i(t - \tau) \quad (15)$$

MPSFQ 서버로 구성된 임의의 망에서 망의 K 번째 노드에서의 최대 지연 D_i^K 은 다음과 같은 한계 값을 가진다.

$$D_i^K \leq \frac{\sigma_i}{r_i} + \sum_{j=1}^K \Theta_j^i - \frac{L_i}{r_i} \quad (16)$$

2.3 MPSFQ의 Fairness 특성

PRPS 시스템에서는 시스템 가상 시간의 재조정으로 인하여 시스템 가상 시간이 세션 가상 시간으로부터 차이가 일정한 값 이상으로 발생하지 않도록 한다. MPSFQ 알고리즘 역시 PRPS 알고리즘으로 역시 시스템 가상 시간이 세션 가상 시간으로부터 일정한 차이 이상이 발생하지 않도록 하는 공정한 스케줄링 알고리즘이다. 본 장에서는 Golestani^[3]가 정의한 fairness의 정의에 의해서 MPSFQ 알고리즘의 fairness를 증명하도록 한다. Fairness 지수는 다음과 같다.

$$\left| \frac{W_i(t_1, t_2)}{r_i} - \frac{W_j(t_1, t_2)}{r_j} \right| \leq F_{i,j} \quad (17)$$

여기서 $W_i(t_1, t_2)/r_i$ 는 세션 i 가 지속적으로 백로그되는 구간인 (t_1, t_2) 에서 받은 정규화된 서비스 양을 의미한다.

Lemma 2 PRPS 시스템의 임의의 시점 t 에서 시스템 가상 시간 $v(t)$ 와 현재 시스템 내에 존재하는 세션 i 의 임의의 패킷 P_i^k 의 가상 종료 시간 F_i^k 의 차이는 다음과 같은 한계 값을 가진다.

$$v(t) - F_i^k \leq \frac{L_{\max}}{r_i} \quad \forall i \in B(t)$$

여기서 r_i 는 세션 i 의 예약된 속도이다.

Proof:

임의의 PRPS 시스템과 그에 대한 RPS 시스템은 다음과 같은 규칙을 가진다^[4].

$$t_i^k \leq \hat{t}_i^k + \frac{L_{\max}}{r} \quad (18)$$

여기서 t_i^k 는 패킷이 PRPS 시스템에서 전송을 마치는 시간이며 \hat{t}_i^k 는 동일한 패킷이 RPS에서 패킷의 서비스를 마치는 시간이다.

시스템 가상 시간의 n 번째 재조정 τ_n 이 t_i^k 에 발생한다. 세션 가상 시간 $v_i(t_i^k)$ 은

$$v_i(t_i^k) = v_i(\hat{t}_i^k) + \frac{r}{r_{M(t)}}(t_i^k - \hat{t}_i^k) \quad (19)$$

여기서 $M(t) = \{j | v_j(t) = \min_{k \in B(t)} v_k(t)\}$ 이다.

RPS 시스템에서 패킷 P_i^k 가 전송을 마칠 때 그 패킷의 가상 종료 시간은 시스템 가상 시간보다 크거나 같다. :

$$v(\hat{t}_i^k) \leq F_i^k \quad (20)$$

따라서 식 (19)와 식 (20) 그리고 RPS 시스템의 두 번째 조건인 $v(t) \leq \min_{k \in B(t)} v_k(t)$ 에 의해서 t_i^k 시점에 시스템 가상 시간은 다음과 같은 한계 값을 가진다.

$$\begin{aligned}
 v(t_i^k) &\leq v(\hat{t}_i^k) + \frac{r}{r_i}(t_i^k - \hat{t}_i^k) \\
 &\leq F_i^k + \frac{L_{\max}}{r_i} \quad (21)
 \end{aligned}$$

두 개의 시스템 가상 시간의 재조정 시간 사이에서 시스템 가상 시간은 경과한 실시간에 의해서 증가하기 때문에 다음 corollary를 유도할 수 있다.

Corollary 1 PRPS 시스템에서 패킷 p_i^k 가 전송을 시작하는 시간 s_i^k 에 시스템 가상 시간과 세션의 가상 종료 시간 F_i^k 의 차이는 다음과 같은 관계를 가진다.

$$v(s_i^k) - F_i^k \leq \frac{L_{\max}}{r_i} - \frac{l_i^k}{r} \quad (22)$$

위의 lemma 1과 corollary 1을 이용하여 [6]에서의 절차와 동일한 방식으로 MPSFQ 알고리즘의 fairness 지수를 증명한다.

Theorem 2 MPSFQ의 fairness 지수는 다음과 같다.

$$FI_{(MPSFQ)} = \max(f_{i,j}, f_{j,i}),$$

여기서 $f_{i,j} = \frac{L_i}{r_i} - \frac{L_j}{r} + \max(\frac{L_j + L_{\max}}{r_j}, \max_{1 \leq n \leq N} \frac{L_n}{r_n})$ 이며 세션 i 가 받은 서비스 양 보다 세션 j 가 더 많이 받은 서비스의 양을 의미한다.

Proof:

앞에서 언급한 바와 같이 두 개의 세션이 지속적으로 서비스할 패킷을 가지는 임의의 시간 구간 (t_1, t_2) 에서 세션 i 와 j 가 받은 정규화된 서비스 양의 차이로부터 $FI_{(MPSFQ)}$ 를 구할 수 있다. 고려할 수 있는 모든 가능한 경우는 다음과 같다.

case 1) 시간 t_1 에 하나의 세션(i)에 패킷이 백로그될 때 이미 다른 세션(i)는 t_1 이전에 백로그되어 있다.

case 2) 두 세션이 모두 t_1 에 백로그 된다. 실제적으로 고려할 수 있는 하나의 경우로 시

간 t_1 이전에 두 세션 모두가 백로그되는 경우를 고려할 수 있지만 이 경우는 $t' < t_1$ 이 성립하는 t' 에서 case 1의 경우와 같다.

따라서 case 1과 case 2에 대하여 $FI_{(MPSFQ)}$ 를 구하면 된다.

Case 1: 세션 j 가 t_1 시점에 백로그되기 시작할 때 이미 세션 i 는 t_1 이전에 백로그되어 있다. F_i^k 를 세션 i 의 HOL 패킷인 p_i^k 의 가상 종료 시간이라고 하자. 여기서 다시 2개의 다른 경우를 고려할 수 있다.

Subcase a) 세션 j 의 패킷 p_j^m 의 가상 종료 시간이 세션 i 의 패킷 p_i^k 의 가상 종료 시간 보다 큰 경우($F_j^m > F_i^k$)

이 경우 세션 j 가 첫 번째 서비스를 받기 전에 세션 i 에 제공되는 정규화된 서비스의 양이 최대가 되는 경우는 패킷 p_j^m 이 p_i^k 가 전송을 위해 선택이 되는 바로 그 순간에 도착하는 경우이다.

Lemma 2와 Corollary 1에 의해서 위의 조건을 만족하는 바로 그 시점에 세션 j 의 가상 종료 시간 F_j^m 은 다음과 같다.

$$F_j^m \leq F_i^k + \frac{L_{\max}}{r_i} - \frac{l_i^k}{r} + \frac{l_j^m}{r_j} \quad (23)$$

여기서 l_i^k 와 l_j^m 은 각각 패킷 p_i^k 와 p_j^m 의 패킷 길이이다.

패킷 p_i^k 에 제공되는 서비스의 양인 l_i^k/r_i 및 p_j^m 가 서비스 되기 이전에 F_j^m 보다 크지 않은 가상 종료 시간을 가지는 세션 i 의 모든 패킷에 의한 서비스 양이 결국 정규화된 서비스 양의 차이를 만들게 된다.

식 (23)으로부터 다음이 성립한다.

$$\begin{aligned}
 F_j^m - S_i^k &= F_j^m - (F_i^k - \frac{l_i^k}{r_i}) \\
 &\leq \frac{L_{\max}}{r_i} + \frac{l_j^m}{r_j}
 \end{aligned}$$

여기서 S_i^k 는 패킷 p_i^k 의 가상 시작 시간이다.

따라서 패킷 P_j^m 이 전송을 시작하기 이전에 세션 i 와 정규화된 서비스 양의 차이는 다음과 같은 한계 값을 가지게 된다.

$$f_{j,i}^{(1a)} = \frac{L_j + L_{\max}}{r_j} - \frac{L_i}{r} + \frac{L_j}{r_j} \quad (24)$$

Subcase b) 세션 j 의 패킷 P_j^m 의 가상 종료 시간이 세션 i 의 패킷 P_i^k 의 가상 종료 시간 보다 작은 경우 ($F_j^m < F_i^k$)

세션 i 가 다시 서비스를 받기 전에 세션 j 주어지는 정규화된 서비스 양을 최대화 하기 위해서 패킷 P_j^m 은 패킷 P_i^k 가 전송을 마치는 바로 그 순간에 시스템에 도착해야 된다. I_i^k 를 패킷 P_i^k 의 길이라 하고 I_i^{k+1} 를 패킷 P_i^{k+1} 의 길이라 하자. 패킷 P_i^k 가 시간 $t_s = t - I_i^k/r$ 에서 전송을 시작하였으며 그 시점에서 MPSFQ의 정의에 따라서 시스템 가상 시간의 최소 가능한 값은 다음과 같다.

$$v(t_s) = F_i^k - \max_{1 \leq n \leq N} \frac{L_n}{r_n}$$

패킷 P_i^{k+1} 가 $F_i^{k+1} = F_i^k + I_i^{k+1}/r$ 의 가상 종료 시간을 가지게 된다. 패킷 P_j^m 이 P_i^k 가 전송을 마치는 바로 이전 시간인 t 에 시스템에 도착하게 되면 시스템 가상 시간의 최소 가능한 값은 다음과 같다.

$$v(t) = v(t_s) + \frac{I_j^m}{r} \quad (25)$$

따라서 세션 j 와 i 간의 정규화된 서비스의 차이는 다음과 같이 표현될 수 있다.

$$F_i^{k+1} - v(t) = \frac{I_i^{k+1}}{r_i} + \max_{1 \leq n \leq N} \frac{L_n}{r_n} - \frac{I_j^m}{r} \quad (26)$$

위의 식 (26)에서 전체적인 정규화된 서비스 양의 차이는 I_i^k 와 I_i^{k+1} 의 값에 따라서 증가를 하게 되기 때문에 결국 다음을 유도할 수 있다.

$$f_{i,j}^{(1b)} = \frac{L_j}{r_j} + \max_{1 \leq n \leq N} \frac{L_n}{r_n} - \frac{L_i}{r} \quad (27)$$

Case 2: 두 세션 i 와 j 모두 시간 t 에 동시에

백로그된다.

이 경우 두 세션은 가상 종료 시간을 계산하기 위하여 동일한 시스템 가상 시간을 사용한다. 따라서 그 두 세션 사이에서 수신된 정규화된 서비스 양의 차이는 가상 종료 시간에 의해서 결정된다. 일반성에 대한 손실 없이 $I_i^k/r_i > I_j^m/r_j$ 라고 할 때 세션 i 는 세션 j 이전에 서비스를 하게 된다. 결과적으로 임의의 시점 t 에서 그 두 세션 사이에 정규화된 서비스 양의 차이는 P_i^k 가 전송을 마치는 시점이 되며 다음과 같은 값을 얻을 수 있다.

$$f_{i,j}^{(2)} \leq \frac{L_j}{r_j} \quad (28)$$

따라서 식 (24), 식 (25), 식 (28)로부터 theorem 1에 대한 증명을 마무리한다.

IV. 시뮬레이션 결과

이번 장에서는 위의 III장에서 분석된 MPSFQ의 성능을 시뮬레이션을 통하여 확인하도록 한다. 성능을 비교하기 위하여 WFQ, SCFQ, SPFQ, 알고리즘과 비교를 하였다.

1. 시뮬레이션 모델 및 트래픽 소스

단일 노드 환경에서 알고리즘에 대한 성능을 분석하였다. 8개의 세션이 동일한 출력 링크를 공유하며 그 중 하나의 세션이 할당된 대역 이상으로 보내고 있다. 노드에서는 스케줄러가 패킷을 큐잉할 수 있는 무한 버퍼를 가지고 있다고 가정한다.

소스 트래픽으로는 ON-OFF 모델이 사용되었다. 패킷 스트림은 T ms의 간격으로 도착하고 있으며 이때 T는 한 패킷 시간으로 정의하였다. OFF 상태에 있을 때는 패킷이 존재하지 않으며 ON 상태에 있을 때에는 평균 $1/(\alpha T)$ 를 가지는 geometric distribution으로 모델링되며 OFF 상태의 지속 시간은 평균 $1/\beta$ 를 가지는 exponential distribution으로 모델링 된다.

Stilliadis의 SPFQ의 시뮬레이션과 동일한 환경으로 비교를 하기 위하여 $1/\alpha$ 를 $100 \cdot r_i$ 로 설정하였으며 $1/\beta$ 를 $100 \cdot (1 - r_i)$ 로 설정하였다. 여기서 r_i 는 세션의 대역을 나타낸다. 비정상적인 하

나의 세션을 제외하고 정상적으로 동작하는 세션들은 leaky bucket에 의하여 웨이핑되어 트래픽이 입력된다.

2. 시뮬레이션 결과

각 세션에 대한 leaky bucket에 대한 버스트의 크기를 2로 설정하였다. 또한 세션 1이 할당 대역 이상의 비 정상적인 트래픽을 발생하도록 하였다. 표 1 과 2에서 각각 한 패킷 전송 시간 관점에서 알고리즘의 평균 지연과 최대 지연 시간을 보여 주고 있다. 앞에서 예상했던 결과처럼 SCFQ의 지연 특성은 좋지 않은 결과를 보여 주고 있으며 MPSFQ의 지연 특성은 WFQ나 SPFQ의 평균 및 최대 지연 특성과 유사함을 쉽게 알 수 있다. 이 결과로서 MPSFQ는 부가적인 복잡도를 요구하지 않으면서도 WFQ나 SPFQ와 유사한 지연 특성을 가진다는 것을 알 수 있으며 이는 앞장에서 수학적으로 증명한 결과를 뒷받침하게 된다.

표 1. 평균 지연 비교

| | Reserved Rate | Arrival Rate | WFQ | SCFQ | SPFQ | MPSFQ |
|---|---------------|--------------|--------|---------|--------|--------|
| 0 | 0.500000 | 0.498 | 1.5913 | 3.08040 | 1.5917 | 1.5826 |
| 1 | 0.062500 | 0.100 | N/A | N/A | N/A | N/A |
| 2 | 0.062500 | 0.062 | 8.5502 | 14.5631 | 5.4680 | 5.8823 |
| 3 | 0.062500 | 0.061 | 8.9832 | 14.4902 | 5.7585 | 5.8112 |
| 4 | 0.078125 | 0.076 | 6.3890 | 11.9415 | 3.8515 | 4.0739 |
| 5 | 0.078125 | 0.076 | 6.4524 | 11.7837 | 3.5661 | 3.5801 |
| 6 | 0.078125 | 0.076 | 5.8597 | 11.6815 | 3.2028 | 3.0361 |
| 7 | 0.078125 | 0.076 | 6.5652 | 11.8675 | 4.1405 | 4.5099 |

V. 결론

본 논문은 Minimum Possible Virtual Start Time Fair Queueing (MPSFQ)으로 명명된 새로운 패킷 스케줄링 알고리즘을 제안하였다. MPSFQ 알고리즘은 SPFQ와 유사하지만 구현의 용이성 측면에서 간단한 스케줄링 알고리즘

표 2. 최대 지연 비교

| | Reserved Rate | Arrival Rate | WFQ | SCFQ | SPFQ | MPSFQ |
|---|---------------|--------------|---------|---------|---------|---------|
| 0 | 0.500000 | 0.498 | 2.0000 | 7.6000 | 5.0000 | 2.0000 |
| 1 | 0.062500 | 0.100 | N/A | N/A | N/A | N/A |
| 2 | 0.062500 | 0.062 | 26.0000 | 31.6000 | 26.0000 | 26.0000 |
| 3 | 0.062500 | 0.061 | 28.0000 | 26.6000 | 28.0000 | 28.0000 |
| 4 | 0.078125 | 0.076 | 19.6000 | 25.9984 | 18.6000 | 18.6000 |
| 5 | 0.078125 | 0.076 | 19.6000 | 24.6000 | 19.4000 | 19.4000 |
| 6 | 0.078125 | 0.076 | 21.4000 | 23.9984 | 21.4000 | 21.4000 |
| 7 | 0.078125 | 0.076 | 22.4000 | 24.6000 | 22.4000 | 22.4000 |

이다. SPFQ에서와 같이 가상 시작 시간의 정렬을 위한 별도의 정렬 구조를 유지할 필요가 없다. MPSFQ 알고리즘은 PRPS에 속하는 패킷 스케줄링 알고리즘이기 때문에 PRPS가 가지는 기본적인 특징으로 WFQ와 동일한 지연 특성을 가진다. 또한 수식을 통하여 연속적으로 서비스할 패킷을 가지는 두 세션에 제공되는 서비스 양의 차이를 통하여 MPSFQ의 fairness 특성이 아주 우수함을 보였으며 지연 특성을 시뮬레이션을 통하여 검증하였다.

참고문헌

- (1) H. Zhang, "Service disciplines for guaranteed performance service in packet switching networks", *Proceedings of the IEEE*, vol.83, no.10, pp. 1374-1396, October, 1995.
- (2) A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single node case," *Proceedings of IEEE INFOCOM'92*, vol. 2, pp. 915-924, May 1992.
- (3) S. J. Golestani, "A self-clocked fair queueing scheme for broadband applications," *Proceedings of IEEE INFOCOM '94*, pp. 636-646, April 1994.

- [4] D. Stiliadis and A. Varma, "Efficient fair queueing algorithms for packet-switched networks," *IEEE/ACM Trans. on Networking*, vol.6, no.2, pp. 175-185, April 1998.
- [5] D. Stiliadis and A. Varma, "A general methodology for designing efficient traffic scheduling and shaping algorithms," *Proceedings of IEEE INFOCOM '97*, vol. 1, pp. 326-335, April 1997.
- [6] F. M. Chiussi and A. Francini, "Minimum-delay self-clocked fair queueing algorithm for packet-switched networks," *Proceedings of IEEE INFOCOM '98*, vol. 3, pp. 1112-1121, March 1998.
- [7] F. M. Chiussi and A. Brizio et al, "A family of ASIC devices for next generation distributed packet switches with QoS supports for IP and ATM," *Hot Interconnects 9*, pp. 145- 149, August 2001.
- [8] C. Dovrolis and D. Stiliadis, "Proportional differentiated services: delay differentiation and packet scheduling," *IEEE/ACM Trans. on Networking*, vol. 10, no. 1, pp. 12-26, February. 2002.
- [9] Byung-Hwan Choi and Hong-Shik Park, "Rate Proportional SCFQ (RP-SCFQ) Algorithm for High-Speed Packet-Switched Networks," *ETRI Journal*, vol. 22, no. 3, pp. 1-9, September 2000.

고 남 석(Nam-Seok Ko)

정회원



1998년 2월 : 전북대학교 컴퓨터공학과 학사
2000년 2월 : 한국정보통신대학원대학교 공학부 석사
2000년 3월~현재 : 한국전자통신연구원 네트워크기술연구소

<주관심분야> IP QoS, Network Processor, 패킷 스케줄링 알고리즘, 네트워크 보안

곽 동 용(Dong-Yong Kwak)

정회원



1983년 2월 : 동국대학교 전산학과 학사
1985년 9월 : 동국대학교 전산학과 석사
1985년 9월~현재 : 한국전자통신연구원 네트워크기술연구소 NP 기술팀 팀장

<주관심분야> IP QoS, Network Processor, 패킷 스케줄링 알고리즘, 네트워크 보호 및 복구, 네트워크 보안