

# Spin을 이용한 근사 정렬된 우선 순위 큐 스케줄러 알고리즘의 명세

김병철, 김태운

고려대학교 컴퓨터학과

e-mail:bckim@netlab.korea.ac.kr

## Formal Specification for Priority Queue Schedulers with Approximate Sorting Algorithm using Spin

Byoung-Chul Kim, Tai-Yun Kim

Dept of Computer Science and Engineering, Korea University

### 요약

본 논문에서는 정형 명세 도구인 Spin을 이용한 근사 정렬된 우선 순위 큐 스케줄러 알고리즘에 대한 정형 명세 방법론을 제시하였다. 최근에 제안된 패킷 스케줄링 알고리즘은 우선 순위(마감 순위, 가상 종료 시간, 시간 스템프 등)에 따라 QoS를 지원한다. 그러나 QoS를 지원하기 위한 우선 순위를 유지하는데는 많은 오버헤드가 요구된다. 따라서 근사된 우선 순위 큐 스케줄러 알고리즘은 낮은 계산상의 오버헤드를 통해 근사된 우선 순위 큐를 유지함으로써 정확한 우선 순위 큐를 유지하기 위한 오버헤드와의 trade off를 고려한다. 큐는 주기적으로 회전을 하며 최소한의 포인터 오퍼레이션을 통해 근사된 우선 순위 큐를 유지한다. 이러한 스케줄러 알고리즘의 동작 과정을 정형 기법을 이용하여 패킷 스케줄링상에 기아 현상등이나 데드락 현상등의 발생여부를 검증하는 방법등의 연구가 전무한 상태이다. 정형 명세 도구인 Spin을 이용하여 제안된 알고리즘을 명세하는 방법론을 기술한다.

### 1. 서론

오늘날 통신 기술의 발달과 인터넷 사용량의 급증으로 사용자의 요구가 다양화, 세분화, 고품질화됨에 따라 인터넷을 이용한 다양한 응용 서비스의 창출과 고도의 통신망 기술 및 정보 처리 기술이 요구되고 있다. quality-of-service(QoS) 분야에서 주요한 분야중에 하나는 우선 순위에 입각한 패킷 전송이다. 대부분의 QoS 네트워크상에서 패킷전송은 우선 순위를 고려한 알고리즘이다. 대표적인 예로서 공정한 큐잉 알고리즘인 generalized processor sharing(GPS)는 우선 순위는 가상 종료 시간과 가상 시작 시간, 시간 스템프에 의해서 결정된다. earliest-deadline-first(EDF) 스케줄러는 패킷을 마감시간에 따라 전송한다. 이러한 알고리즘들은 모두 일반적으로 정렬된 우선 순위 큐를 필요로 한다. 고속의 네트워크상에서는 이러한 정렬된 우선 순위 큐

를 유지하기 위한 오버헤드는 병목현상을 발생 시킨다. 일반적으로 우선 순위 큐를 유지하기 위한 오버헤드는 최악의 경우 N개의 엔트리에 대해  $O(\log N)$ 으로 알려져 있다. 본 논문에서 제시한 알고리즘은 우선 순위 큐를 유지하기 위한 계산상의 오버헤드를 근접한 우선 순위 큐로 유지하면서 줄이는 접근 방법을 제시한다. 스케줄러는 전송할 패킷을 가지고 있는 가장 높은 우선 순위를 가지는 FIFO큐에서 패킷 전송을 시작한다. 스케줄러에게 전송되는 패킷은 시간 스템프가 찍어지며 이러한 시간 스템프는 마감 시간이나 가상 종료 시간, 다른 값등이 될 수 있다. 우선 순위가 낮은 패킷들의 기아 현상을 막기 위해 큐는 주기적으로 재 명명되는데, 모든 FIFO(p) 큐는 FIFO(p-1)로 명명되고 FIFO(1) 큐는 FIFO(q)로 명명된다. 이러한 큐의 주기적인 재명명은 패킷들의 시간 스템프에 따른 근사한 우선 순위를 유지하게

하여 주며 우선 순위가 낮은 패킷들의 기아 현상을 막아 준다. 근사된 우선 순위 큐라는 개념은 첫 번째로 패킷은 시간 스템프가 아닌 큐에 도착한 순서대로 순서가 정해지기 때문이며, 두 번째로 시간 스템프로서  $p$ 를 가지는 패킷이 큐 회전 바로 전에 도착하였다면 큐의 회전에 의해 시간 스템프  $p$ 를 가지는 패킷은  $p-1$ 의 순위를 가지고 시간 스템프  $p-1$ 을 가지는 패킷이 바로 뒤에 도착하였다면 시간 스템프  $p-1$ 을 가지는 패킷은 시간 스템프  $p$ 를 가지는 패킷 뒤에 위치하기 때문이다. 이러한 근사한 우선 순위 큐를 유지하기 위한 복잡도는 패킷의 수와는 무관하며 FIFO 큐의 개수  $P$ 에 따른다는 장점이 있다. 근사한 우선 순위를 가지며 회전하는 큐 알고리즘을 rotating priority queues(RPQ)라고 부르며 이 알고리즘은 회전하지 않는 우선 순위 큐보다 성능상 나쁘지 않아야 한다는 상황을 만족 시켜 주지 못한다. RPQ보다 2배의 큐의 개수를 가지면서 이러한 문제점을 해결한 rotating priority queues<sup>+</sup>(RPQ<sup>+</sup>) 알고리즘을 제시한다. 제시된 RPQ, RPQ<sup>+</sup> 알고리즘을 명세하기 위해 정형 명세 도구인 Spin을 이용하여 명세하는 방법론을 제시한다. 본 논문의 구성을 살펴보면 2장에서는 RPQ, RPQ<sup>+</sup> 알고리즘의 동작 과정을 알아보고, 3장에서는 현재 시스템의 안전성을 보장하기 위한 연구와 사용되는 방법인 정형기법에 대해서 간단히 설명하고, 4장에서는 정형 명세 도구 Spin 및 정형 명세 언어 Promela에 대해 설명한다. 5장에서 결론과 향후 과제를 제시한다.

2 RPQ, RPQ<sup>+</sup> 알고리즘  
2.1 RPQ 알고리즘

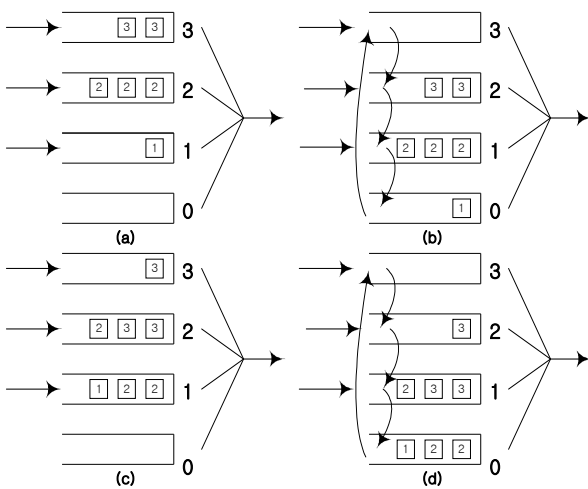


그림 1 RPQ 스케줄러 (P = 3)

그림 1은 3의 우선 순위를 가지면서 동작하는 RPQ 스케줄러를 보여준다. (a)는 시간  $0 \leq t < \Delta$ 에서

의 FIFO 큐의 상태이다. (b)는  $\Delta$  시간에서 큐가 회전한뒤의 상태를 나타낸다. RPQ 알고리즘에서 만약 가장 우선 순위가 높은 FIFO(1)에 패킷이 있는 상태에서 큐가 회전하면 FIFO(1)에 패킷들은 회전뒤에 가장 낮은 우선 순위를 가지는 FIFO(p)가 되므로 이러한 문제점을 해결하기 위해서 FIFO(0)을 따로 두고 FIFO(0)에 패킷이 없을때만 회전이 가능한 제약점을 둔다. 또한 FIFO(0)에는 새로 스케줄러에 들어온 패킷이 바로 삽입되는 것을 막으므로 낮은 우선순위를 가지는 패킷이 큐가 회전하지 못하여 서비스 받지 못하는 기아 현상을 해결한다. 이러한 RPQ 스케줄러의 문제점은 그림 2와 같다.

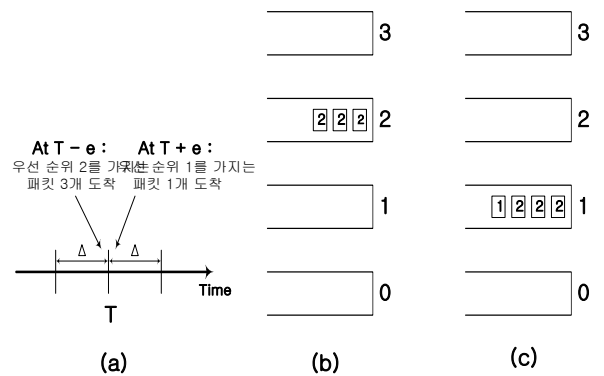


그림 2 RPQ에서 변칙 회전

그림 2의 (a)와 같이 시간  $\Delta$  바로 직전에 우선 순위 2인 패킷이 3개 들어오고 시간  $\Delta$  바로 다음에 우선 순위 1인 패킷 1개 들어온 상황이라면 FIFO(1)에서의 패킷 순서는 (c)와 같이 우선 순위 1인 패킷이 우선 순위 2인 패킷 뒤에 위치하게 된다. 이러한 문제점을 해결하기위해 RPQ<sup>+</sup> 스케줄러 알고리즘은 RPQ 스케줄러가 사용하는 큐보다 2배 많은 큐를 사용하여 스케줄링을 실시한다. RPQ<sup>+</sup> 스케줄러의 동작은 그림 3과 같다.

2.2 RPQ<sup>+</sup> 알고리즘

그림 3의 (a),(b),(c)는 각각 시간  $\Delta, 2\Delta, 3\Delta$ 에서의 큐의 회전 상황을 보여준다. RPQ<sup>+</sup> 알고리즘은 Concatenation step과 Promotion step으로 나뉜다. Promotion step에서는 FIFO(p)는 FIFO(p-1)<sup>+</sup>로 명명된다 그림 3의 (iii). Concatenation step에서는 새로들어온 패킷들이 있는 FIFO(p)와 기존에 있던 패킷들이 있는 FIFO(p)<sup>+</sup>와 합병하는 단계이다. RPQ 알고리즘과 비교하여 큐의 개수를 2배로 늘려 사용함으로써 RPQ 알고리즘에서 제기된 패킷 순서의 변칙을 해결하였다.

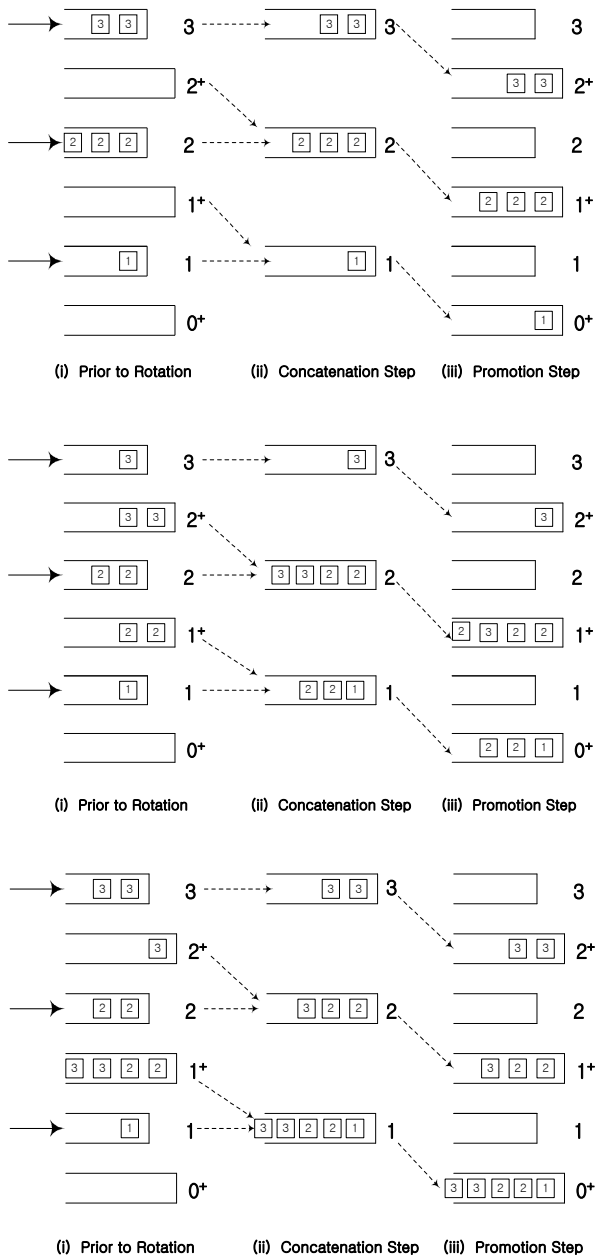


그림 3 RPQ<sup>+</sup> 스케줄러

그림 3의 (a),(b),(c)는 각각 시간  $\Delta, 2\Delta, 3\Delta$ 에서의 큐의 회전 상황을 보여준다. RPQ<sup>+</sup> 알고리즘은 Concatenation step과 Promotion step으로 나뉜다. Promotion step에서는 FIFO(p)는 FIFO(p-1)<sup>+</sup>로 명명된다 그림 3의 (iii). Concatenation step에서는 새로들어온 패킷들이 있는 FIFO(p)와 기존에 있던 패킷들이 있는 FIFO(p)<sup>+</sup>와 합병하는 단계이다. RPQ 알고리즘과 비교하여 큐의 개수를 2배로 늘려 사용함으로써 RPQ 알고리즘에서 제기된 패킷 순서의 변칙을 해결하였다.

### 3 정형 기법 (Formal Methods)

정형 기법은 수학과 논리학에 기반을 두 방법으로

하드웨어나 소프트웨어 시스템을 명세하거나 검증하는 것이다. 수학적 기호를 사용하여 시스템의 명세를 작성하고 검증할 특성도 논리로 기술하여 시스템이 특성을 만족하는지를 수학적 성질을 이용하여 검증함으로써 자연어가 내포하는 애매모호함이나 불확실성을 최소화할 수 있다. 정형 기법에는 크게 정형 명세와 정형 검증으로 나누어 진다.

#### 3.1 정형 명세 (Formal Specification)

정형 명세에는 요구 명세와 설계 명세로 구성되는데 요구 명세는 시스템이 무엇을 해야 하는지를 정의해 놓은 명세로서 간결하게 표현되고, 설계 명세는 요구 사항을 완성할 수 있는 설계를 묘사하는데 목적이 있으므로 설계 명세는 시스템의 형태를 구현할 목적으로 작성되며 세부적인 사항을 포함한다. 정형 명세 언어는 자연어가 내포하고 있는 애매모호함이 없기 때문에 명세 하고자 하는 시스템을 명확하게 기술할 수 있다.

#### 3.1 정형 검증 (Formal Verification)

정형 검증은 명세가 정확한지를 검증하는 것을 의미한다. 정형 명세한 요구 명세와 설계 명세가 일치하는지를 검증한다. 정형 기법을 사용하여 시스템을 개발하면 개발 초기에 시스템의 정확성을 검증함으로써, 완성된 시스템을 시험하고 검사하는데 소요되는 시간과 비용을 경감시킬 수 있다는 장점이 있다.

### 4. Spin과 Promela

#### 4.1 Promela 문법

정형 명세와 검증을 원하는 시스템은 Promela를 이용하여 명세될 수 있다. Spin은 랜덤 하거나 인터랙티브한 시스템의 수행을 시뮬레이션 할 수 있으며, 시스템 상태를 빠르게 검증할 수 있다. 시뮬레이션과 검증단계에서 Spin은 데드락 및 수행되지 않는 코드등의 발생 여부를 체크한다. 또한 시스템의 불변성 및 진행 사이클에서의 진행 될 수 없는 부분의 검사를 실시할 수 있다. Promela 프로그램은 프로세스와 메시지 채널, 변수로서 구성되어 있다. 문법의 구성은 C와 비슷하다. 몇가지 특징적인 Promela의 문법을 살펴 보면 다음과 같다.

- Message channels  
ex) chan Transfer = [2] of {mtype, bit, short, chan}  
Transfer는 2개의 메시지를 저장할 수 있으며, 메시지 형태는mtype,bit,short,chan이 가능하다.
- Atomic  
ex) atomic { statements }  
다른 프로세스들의 간섭없이 statements이 한 번

에 수행된다.

● Selection

ex) if :: statements

....

:: statements

fi

:: 로서 시작되는 statemets들중에서 조건에 맞는 것을 실행한다. 한가지 이상의 조건에 맞는 ::로서 시작하는 statements가 있으면 random하게 선택하여 수행한다.

● Repetition

ex) do :: statements

...

:: statements

od

if 문과 수행동작은 비슷하며 차이점은 반복하여 수행한다.

4.2 Spin을 이용한 RPQ 및 RPQ<sup>+</sup> 명세

큐의 명세는 byte 채널로서 큐를 명세한다. byte 패킷을 랜덤하게 생성하며 생성된 패킷은 byte 값에 따라 우선 순위가 정해지며 우선 순위에 따라 각각의 채널에 입력되게 된다. 패킷 전송부분은 우선 순위가 높은 채널부터 삭제 오퍼레이션을 통해 전송 과정을 시뮬레이션하며 정해진 개수 만큼의 패킷이 전송되면 채널을 회전하는 루틴이 수행되게 된다. 큐를 회전 시키는 루틴은 입력 채널과 출력 채널을 함수의 인자로 받아 입력 채널의 모든 값들을 출력 채널로 out시키므로서 수행한다. RPQ<sup>+</sup> 알고리즘의 명세에서는 채널의 사용 개수를 2배로 하면서 채널을 회전하기 전에 채널을 합치는 concatenation step을 수행한다.

5. 결론 및 향후 과제

Spin을 이용한 RPQ와 RPQ<sup>+</sup> 알고리즘의 명세는 일반적인 프로그래밍 언어로 시뮬레이션 과정을 기술한다면 상당한 양의 프로그램 라인이 필요하겠지만, Promela를 이용하면 약 50라인 수준에서 시뮬레이션에 필요한 소스 코드 작성이 끝났으며 Spin을 이용한 시뮬레이션 과정에서 각 채널의 상태를 시각적으로 표현함으로써 패킷의 우선순위가 뒤바뀌어 스케줄되는 RPQ 알고리즘의 문제점을 시각적으로 확인할 수 있었다. 추가적으로 RPQ와 RPQ<sup>+</sup> 알고리즘을 이용할 시 낮은 우선 순위를 가지는 패킷들의 서비스 기아 현상을 검증하기 위해서는 시간 개념이 필요하였으나 시간의 개념이 들어간 시스템 명세의

불현함으로 인해 우선 순위가 낮은 큐의 패킷 개수를 측정함으로써 특정 시간에 한계값 이상의 패킷 개수를 기아가 발생한 것으로 판단하였다.

향후 과제로서 RPQ<sup>+</sup> 알고리즘이 패킷의 우선순위가 뒤바뀌는 현상을 해결하기 위해 큐의 개수를 RPQ에 비해 2배 많이 사용하는 오버헤드 문제를 해결하기 위해 사용하는 큐의 개수를 줄이면서 패킷의 우선 순위가 뒤바뀌지 않는 개선된 알고리즘을 Spin을 이용하여 명세하여 본다.

6. 참고 문헌

[1] Jorg Liebeherr, Dallas E. Wrege "Priority Queue Schedulers with Approximate Sorting in Output-Buffered Swithes" IEEE trans comm., vol. 17. NO. 6. June 1999

[2] A. K. Parekh and R. G. Gallager, " A generalized processor sharing approach to flow control in integrated services networks: The single-node case," IEEE/ACM Trans. Networking, vol. 1, pp 344-357, June 1993.

[3] F. M. Chiussi and V. Sivraman, "Achieving high utilization in guaranteed services networks using early-deadline-first scheduling," in Proc. 6th IEEE IWQoS'98, pp. 209-217

[4] Y. Lim and J. E. Kobza, "Analysis of a delay-depenent priority discipline in an integrated multiclass traffic fast packet switch," IEEE Trans. Commun., vol. 38, pp. 659-665, May 1990.

[5] D. E. Wrege and J.Liebeherr. "A near-optimal packet scheduler for QoS networks", in Proc.IEEE Infocom '97 Conf., pp. 576-583

[6] 유희준, 이성권, 김우곤, 최진영 “Z를 이용한 보안기능의 설계 방법 연구” vol. 2 pp. 32-56 Dec 1999.