

# 멀티프로세서 멀티쓰레드 기반의 네트워크 시스템에서 패킷 처리 태스크의 스케줄링 알고리즘 성능 연구

김 창 경, 강 윤 구  
아주대학교 전자공학부

전화 : (031) 219-1786 / 팩스 : (031) 219-1798

H.P 번호 : 016-9596-6079

## A Study of tasks scheduling algorithms for packet processing on network system with multi-processor multi-threaded architecture

Chang Kyoung Kim, Yoon Gu Kang  
School of Electronics Engineering Ajou University  
E-mail : moohon@csl.ajou.ac.kr

### Abstract

In this paper, we modelize several scheduling algorithms for real-time packet filtering tasks based on the multi-threaded multi-processor architecture for the network security system like the firewall and compare the performance of the algorithms by implementing the algorithms and doing a number of empirical tasks. As the matrices of the performance we use the idle factor and the packet transfer rate. We get the idle factors and the packet transfer rates according to the transfers of the packet sizes from 64 bytes to 1500 bytes.

### I. 서론

최근까지 대부분의 network device들은 단일의 범용 프로세서에 소프트웨어를 구현함으로써 사용해 왔

다. 하지만, 최근 인터넷의 성장은 기하급수적으로 증가하여 기존의 단일 프로세서로서는 그 한계를 드러냈다. 단일의 프로세서의 처리 속도나 용량의 향상을 꾀하는 것보다 여러 개의 프로세서를 사용함으로써 같은 성능을 꾀하는 것이 훨씬 쉬울 뿐 아니라들이 Processing Element를 효율적으로 운영하도록 지원하기 위해서 하드웨어적으로 한 프로세서 안에 여러 개의 쓰레드들을 지원하는 멀티 쓰레드의 구조를 가진 프로세서들이 나타나게 되었다.

방화벽이나 침입 탐지 시스템 등은 일반적으로 개방된 인터넷으로부터 내부의 사설망을 보호하기 위하여 들어오는 모든 패킷에 대한 적법성 여부를 판단한다. 특히 방화벽은 입력 패킷의 적법성 여부에 따라 패킷을 내부의 사설망으로 보내거나 폐기하는 등의 대처를 한다.

본 논문은 방화벽 같은 필터링 엔진을 구성하는 데 있어서의 멀티 프로세서 멀티 쓰레드의 패킷에 대한 스케줄링을 구현하고 실험하여 성능 평가를 하였다.

본 논문의 구성은 서론에 이어서, 2장에서는 연구의 동기가 되는 멀티 프로세서 멀티 쓰레드의 구성에 대한 여러 스케줄링 알고리즘의 구성 및 구현과 이에 따른 문제점을 짚어보고, 3장에서는 여러 스케줄링 알고리즘의 성능 이득을 실험한 실험 결과에 대한 분석을 한다. 마지막 4장에서는 본 논문에서 도출된 결론에 대하여 논한다.

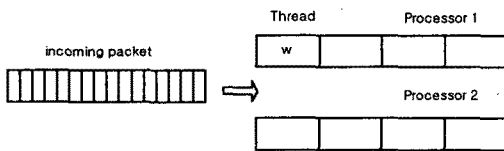
## II. 패킷 처리 태스크 스케줄링

고속망에서 패킷을 처리하기 위해서는 멀티 프로세서 멀티 쓰레드 구조에서의 패킷에 대한 프로세서의 스케줄링을 이용해서 빠르게 패킷 처리를 할 수 있다.

각 프로세서들에 대한 스케줄링 방법으로는 FCFS (First Come First Served), Round-Robin 방법 등 여러 가지가 있다[5]. 그 중에서 Round-Robin은 구현의 간편함과 스케줄링의 적은 overhead 덕분에 고속망에 적합한 스케줄링으로 알려져 있다[1]. 우리는 이 스케줄링 방법들을 멀티 프로세서 멀티 쓰레드 모델에 적용시킬 것이다.

멀티프로세서를 가지는 분산환경에서 사용되는 버퍼 모델은 크게 단일 버퍼 모델(Centralized Queuing Model)과 분산 버퍼 모델(Distributed Queuing Model)이 있는데, 단일 버퍼 모델이 분산 버퍼 모델보다 빠른 것으로 알려져 있다[2].

따라서 구현하려는 시스템의 모델은 단일 버퍼 모델을 사용하고 멀티프로세서 멀티쓰레드의 구조를 가지고 동작하도록 구성하였다.

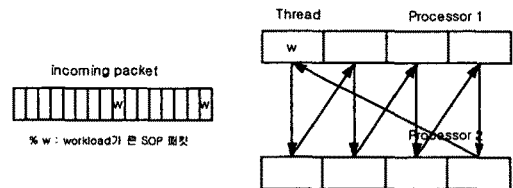


멀티프로세서 시스템에서 패킷에 대하여 프로세서를 할당하는 스케줄링 방법에는 병렬 블록 처리 (Parallel Block Processing)라는 방법이 있다. 이 방법은 연속된 입력 데이터를 일정한 크기로 나눈 다음, 이들을 각각 다른 프로세서에 할당하여 처리하게 하는 방법이다. 이 병렬 블록 처리 방법을 사용하여 입력되는 패킷을 일정한 크기로 나누어 한 프로세서의 하나의 쓰레드에 할당하여 처리하게 하고, 블록을 할당받아 처리하는 쓰레드가 memory operation 등을 수행할 경우, context switching 되어 같은 프로세서의 다른 쓰레드가 패킷의 다른 블록을 받아 처리할 수 있게 함으로서 빠른 패킷의 처리가 가능하게 한다.

인터넷 패킷은 일정 패킷 블록으로 나누어 SOP,

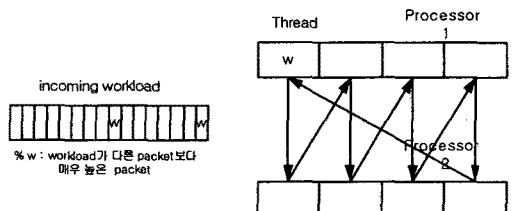
EOP, nSOP\_nEOP, SOP\_EOP의 4가지로 분류할 수 있다. SOP(Start Of Packet)은 하나의 패킷 프레임은 일정 크기로 나누었을 때의 첫 번째 패킷 블록으로 패킷의 헤더부분이 존재한다. 이 SOP는 패킷 헤더를 이용하여 적법성 검사를 하는 필터링 엔진 등에서 그 처리에 있어서의 복수개의 조건과의 매칭으로 인하여 다른 패킷 블록에 비해 많은 실행 시간을 요구한다. EOP(End Of Packet)은 패킷의 마지막 부분이고, nSOP\_nEOP는 패킷의 첫 부분도 아니고 마지막 부분도 포함하지 않는 패킷의 중간부분이고, SOP\_EOP는 패킷의 시작과 끝 부분이 공존하는 부분으로 64 byte의 길이를 가지고 있는 패킷들이 이에 속한다.

프로세서의 스케줄링을 위해 많이 사용하는 방식은 Round-Robin 스케줄링 방식이다. 아래는 멀티 쓰레드 구조의 프로세서가 두개일 경우의 Round-Robin 스케줄링을 표현한 그림이다. 한 프로세서에는 각각 4개의 쓰레드를 가지고 있다.



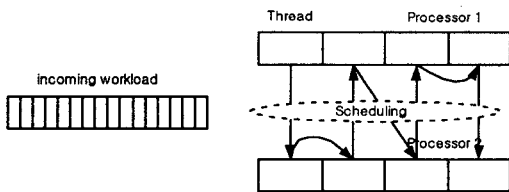
한 프로세서의 하나의 쓰레드가 패킷 블록을 할당 받으면 다음의 쓰레드에 실행 시작 신호를 줌으로서 다음의 쓰레드가 깨어나서 실행할 수 있도록 하고, 다음의 쓰레드를 결정하는 방법은 각기 다른 프로세서의 하나의 쓰레드가 정적으로 자신의 다음 쓰레드로 선택되도록 구성하였다.

이 방법은 다음의 쓰레드를 결정하는 쓰레드 선택에 있어서 정적으로 정의해 놓았기 때문에 빠르게 후행 쓰레드를 결정할 수 있어서 그에 따른 오버헤드가 없다. 하지만, Round-Robin 방법은 패킷 블록의 크기가 고려되지 않은 방법으로 실행시간이 매우 큰 헤더를 포함하는 SOP를 처리하는 경우 이 패킷 블록이 한 프로세서로 몰려있는 경우엔 해당 패킷의 처리가 늦어짐에 따라 전체적인 패킷 처리 속도의 저하를 가져올 수 밖에 없다.



위의 그림에서 w로 표시되는 부분은 방화벽 같은

시스템의 헤더 프로세싱을 하는 SOP패킷을 나타내고 있다. 위와 같이 Round-Robin 방식은 SOP 패킷의 처리와 같은 많은 실행시간을 요구하는 패킷에 대한 처리를 고려하지 않았으므로 실질적으로 SOP 패킷을 처리하는 쓰레드는 이전 쓰레드로부터 새로운 패킷을 받더라도 이전 패킷에 대한 처리가 끝나지 않을 상황이 존재하게 된다. 이 같은 문제점을 해결하기 위해서 FCFS 스케줄링을 구현하여 먼저 패킷 블록을 처리하고 기다리고 있는 상태의 쓰레드를 찾아 후행 쓰레드로 결정하고 그렇게 결정된 후행 쓰레드에게 다음 패킷 블록을 처리하도록 signal을 주도록 하였다.



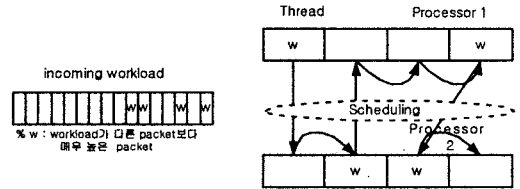
위의 그림에서와 같이 하나의 쓰레드는 자신이 선택되면 스케줄링을 통해서 후행 쓰레드를 결정한다. 이 스케줄링 방식은 스케줄링 오버헤드가 있는 반면 정적으로 정해진 Round-Robin 방식보다는 후행 쓰레드를 선택하는데 있어서의 SOP 패킷을 처리하고 있는 쓰레드에 대한 이전 쓰레드의 패킷 블록 할당 측면에서 해당 패킷이 전송되지 못함으로 인한 그 이후의 쓰레드들이 SOP패킷을 처리하고 있는 쓰레드가 이전 패킷을 다 처리하고 새로운 패킷을 처리할 수 있을 때까지 기다리고 있는 현상을 줄일 수 있다.

이 방법 또한 SOP에 대한 문제점을 가지고 있다. 멀티 프로세서 시스템에서 하나의 프로세서에 SOP 패킷이 물리게 되는 경우에 상대적으로 처리시간이 긴 SOP 패킷을 하나의 프로세서에서 처리하게 되면 처리되어 전송되어야 할 패킷이 SOP 패킷의 처리가 늦어짐에 따라 다른 패킷 블록들이 처리되었음에도 불구하고 패킷이 전송되지 못하고 패킷 버퍼에 남아 있게 된다. 그렇게 되면 역시 패킷의 처리 속도가 떨어지게 되므로 이들 SOP 패킷들에 대한 분산 정책이 필요하다.

그리하여, FCFS를 수정하여 SOP 패킷이 한 프로세서에 할당됨으로서 패킷 헤더가 늦게 처리됨에 따라 생기는 bottleneck을 줄여줌으로써 고속망에서 filtering engine 내부에서 패킷이 처리되는 지연시간이 줄어들게 되고 이에 따라 빠르게 패킷을 처리할 수 있다. 이를 위하여 SOP 패킷이 어느 프로세서에 얼마나 있는가를 체크하는 정보 필드를 이용하여 적절히 SOP 패킷을 다른 프로세서의 쓰레드에 분산시켜 할당한다. 이를 modified FCFS라고 명명하고 구현

하였다.

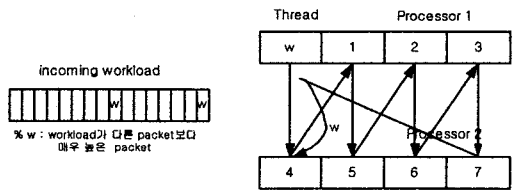
아래의 그림은 w로 표시되는 SOP 패킷의 프로세서에 분산을 고려하여 할당한 모습을 나타낸다.



이 스케줄링 역시 문제점이 있다. SOP 분산을 위해서는 수행 중인 쓰레드에서 다음의 패킷 블록이 SOP 패킷인가를 안 후에 후행 쓰레드를 결정할 수 있다. 다음의 패킷 블록이 SOP 인가를 알기 위해서는 자신이 처리하는 패킷이 EOP라고 인지하면 가능하므로 자신의 패킷에 대한 정보를 읽고 난 후에 SOP 패킷을 프로세서에 분산시킬 수 있다. 따라서 mail을 보내는 시점이 이전의 두 가지 방법 보다 상당히 뒤로 미뤄지게 된다. 그렇게 되면 SOP 패킷을 분산시키는 이득보다 오히려 mail을 늦게 보냄으로 인한 오버헤드로 인해 성능의 효과를 보지 못할 경우가 발생할 수 있다.

그리하여 Round-Robin에서 SOP 패킷을 처리함으로써 인한 패킷 처리의 블로킹 상태를 배제하기 위하여 후행 쓰레드가 working 상태일 때는 자신의 후행 쓰레드가 아닌 그 다음의 쓰레드에 패킷 블록을 할당함으로써 블로킹 상태를 없애주는 방법을 생각하고 구현하였다.

이와 같은 방식은 SOP 패킷에 의한 패킷 처리 지연을 줄이게 되어 처리 속도의 향상이 기대된다.



### III. 실험 결과 및 분석

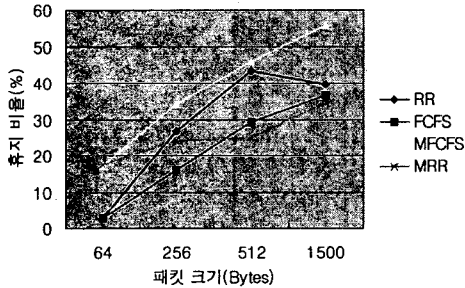
본 장에서는 Round-Robin, FCFS 스케줄링 알고리즘과 이들 패킷의 특성에 맞게 수정한 Modified FCFS, Modified Round-Robin 스케줄링 알고리즘에 대한 성능 평가를 실험하기 위한 실험 환경과 내용, 그리고 결과에 대하여 기술한다.

본 논문의 실험에 사용한 환경으로는 멀티 프로세서 멀티 쓰레드의 구조인 ixpi200을 사용하였다.

ixp1200은 시스템의 전반적인 관리를 위한 하나의 주 프로세서가 존재하고, 주 프로세서와 공유메모리를 통하여 연결된 6개의 칩 레벨 RISC 멀티 프로세서를 가지고 있고, 각각의 프로세서는 멀티 쓰레드 구조로 되어 있다.

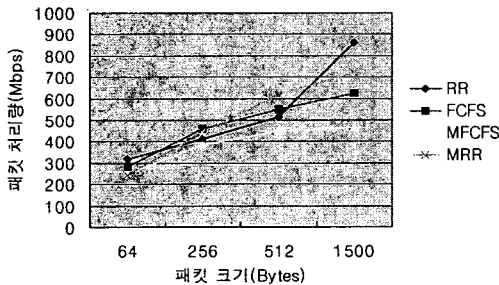
본 논문의 실험에는 64, 256, 512, 1500 패킷에 대해서 4가지의 스케줄링 모델에 대해 휴지시간(idle time)과 전송률(transfer rate)에 대해서 측정하였다.

프로세서의 휴지시간은 한 프로세서의 어떠한 쓰레드도 동작하지 않는 상태이므로 휴지 시간이 많을 수록 패킷의 처리에 사용되는 시간이 적으므로 휴지 시간을 줄여주어야 한다. 아래의 그래프는 각 패킷 크기에 따른 휴지 시간을 모델별로 보여준다.



Round-Robin 방법은 64 byte의 패킷의 경우 후행 쓰레드를 결정하는데 대한 오버헤드가 적어서 가장 적은 휴지시간을 가지고 있지만, 256, 512 byte에서는 SOP패킷의 처리에 따른 지연과 bottleneck으로 인하여 휴지시간이 급격히 증가함을 볼 수 있다.

그래프에 따르면 MFCFS 방법이 가장 많은 휴지 시간을 가지고 있는 것으로 나타났다. 이는 패킷의 정보를 확인하고 후행 쓰레드를 결정하고 신호를 주는 데에 대한 오버헤드가 크므로 인해 다른 스케줄링 알고리즘에 비해 휴지시간이 많은 것으로 나타났다.



64, 256, 512 byte 패킷의 경우 휴지시간이 패킷의 처리량에 영향을 준다는 것을 알 수 있다. 하지만

1500 byte 패킷의 결과를 살펴보면 처리량이 제일 높은 Round-Robin 방식의 휴지시간 또한 많다는 것을 알 수 있다. 이는 FCFS나 Modified RR 스케줄링에서의 프로세서 사용이 후행 쓰레드를 결정하는 데에도 많이 사용됨을 알 수 있다.

## VI. 결론 및 추후 연구

본 논문에서 멀티 프로세서 멀티 쓰레드 구조의 프로세서 및 쓰레드의 여러 스케줄링 알고리즘을 구현하고 실험하였다.

패킷의 크기에 따라 각 스케줄링의 휴지시간의 차이가 패킷의 전체 처리에 영향을 주는 것을 알 수 있지만, 스케줄링에 있어서 후행 쓰레드 결정에 따른 오버헤드가 패킷의 처리에 사용되는 시간에 비해 그 크게 나타나는 크기가 큰 패킷의 경우에는 후행 쓰레드를 정적으로 정해놓은 Round-Robin 방식이 후행 쓰레드를 결정하는 다른 스케줄링 방식보다 좋다는 것을 알 수 있다.

멀티 프로세서 멀티 쓰레드의 네트워크 시스템을 구현함에 있어서 시스템이 사용될 네트워크의 전반적인 사항을 고려해서 적합한 스케줄링 방식을 사용하는 것이 좋다.

## 참고문헌(또는 Reference)

- [1] Hui Zhang, "Service disciplines for guaranteed performance services in packet-switching networks", Proceeding of IEEE, Vol. 83, No.10, pp.1374-1396, 1995
- [2] Qiwei Huang, Errol Lloyd, "Analysis of Queuing Delay in Multimedia Gateway Call Routing", CSREA, IC 2002
- [3] IXP1200 Network Processor Programmer's reference, Intel
- [4] IXP1200 Network Processor Hardware Reference Manual, Intel
- [5] William Stallings, "Operation Systems 3rd edition", pp389-394, Prentice Hall, 1998