큐의 상태를 고려한 VOQ 스케쥴링 알고리즘

송은봉[0] 최문철 조한성 신상호 최재원 안순신
고려대학교 전자공학과
{seb, cmc, chos.feagle, zealot,sunshin}@dsys.korea.ac.kr

VOQ Scheduling Algorithm considering queue state

Eun-Bong Song Moon-Chul Choi Han-Sung Cho Sang-Ho Shin Jae-Won Choi
Sun-Shin An
Dept. of Electronics, Korea University

요 약

VOQ 방법은 입력 큐잉의 저 비용과 출력 큐잉의 고성능이라는 장점을 결합한 효과적인 큐잉 방법이다. VOQ를 효과적으로 사용하기 위해서는 셀의 순서를 스케쥴 해주는 스케쥴링 알고리즘이 필요하다.

본 논문에서는 기존의 iSlip, RRM 의 round-robin based priority 방식을 따르면서 큐의 현재 상태에 따라서 스케쥴링을 달리하는 알고리즘을 제시하였다. 이 알고리즘에서는 큐에 대한 threshold값을 설정하고 큐에 저장하고 있는 셀의 수가 threshold 값 보다 클 경우 그 큐의 셀을 연속해서 포워딩 하도록 하였다.

시뮬레이션을 통해 본 논문에서 제시한 알고리즘과 기존의 iSlip 알고리즘과의 성능을 패킷의 지연 시간 측면에서 비교해 보았다.

1. 서론

최근 대부분의 고속 스위치나 고속 라우터는 일정한 크기의 셀을 전송 단위로 하는 crossbar switch를 기반으로 만들어진다. Crossbar switch에서는 서로 다른 입력 단에 존재하는 여러 개의 셀들이 같은 목적지를 가질 수 있으며 셀의 손실을 피하기 위하여 셀들을 저장하기 위한 큐잉이 필요하다.

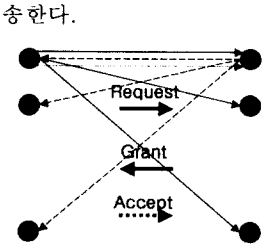
일반적으로 큐잉 방법에는 입력 큐잉, 출력 큐잉의 두가지 방법이 있다. 입력 큐잉은 HOL(Head Of Line) 충돌 현상에 의한 Throughput 저하의 단점이 있으며 출력 큐잉은 N*N 의 switch 일 경우 switch fabric의 N배의 속도 향상을 요구한다는 단점이 있다. 최근에는 입력 큐잉의 단점인 HOL 충돌 현상을 없애기 위하여 VOQ 방법을 많이 이용한다.

현재까지 VOQ를 이용한 switch의 스케쥴링 알고리즘이 많이 제시 되었다. 그러나 큐에 존재하는 셀의 수에 따라서 스케쥴링을 달리하는 알고리즘은 현재 제시 되지 않았다. 본 논문에서는 기존의 RRM, iSlip 알고리즘에 기반을 두고 큐에 존재하는 셀의 수에 대한 threshold를 설정하여 이에 따라서 스케쥴링을 하는 알고리즘을 제시하였으며 시뮬레이션을 통해서 알고리즘의 성능을 분석해 보았다.

2. 관련 연구

대표적인 VOQ 스케쥴링 알고리즘인 PIM, RRM, iSlip 알고리즘은 [그림 1]과 같이 request, grant, accept라는 세가지 단계로 구성된다. 세가지 단계는 다음과 같다.

1) Request: 모든 입력 단에서 출력 단으로 request를 신호를 브로드캐스트 한다.
2) Grant: 각각의 출력 단에서 하나의 request를 선택하고 grant 신호를 선택된 입력 단으로 보낸다.
3) Accept: 각각의 입력 단은 하나의 grant를 선택하고 선택된 출력 단으로 하나의 셀을 전송한다.



[그림 1] VOQ 스케쥴링 알고리즘 절차

위의 세가지 알고리즘의 차이점은 grant할 request와

accept할 grant 신호를 선택하는 방법이다. PIM에서는 모든 request 신호에 대한 grant 신호를 보내며 accept는 random하게 이루어진다. 이에 반해 RRM 과 iSlip에서는 request 신호 선택과 grant 신호 선택이 round-robin based priority 방식으로 이루어 진다.

3. 알고리즘 소개

앞 절에서 설명한 RRM, iSlip 알고리즘과 같이 본 논문에서 제시하고자 하는 알고리즘도 request, grant, accept 세 가지 상태를 가지며 round-robin based priority 방식에 근거한 알고리즘이다. 그러나 본 논문에서 제시하고자 하는 알고리즘은 VOQ의 상태를 고려하여 스케쥴링을 수행한다는 점에서 기존의 알고리즘과 다르다.

본 논문에서는 만약 VOQ 의 길이가 우리가 정한 threshold보다 길 경우 그 입력 단을 **lengthy input**이라고 정의하겠다. 만약 입력 단이 lengthy input 으로 설정되면 다음 time slot 시 우선적으로 accept 된다.

본 논문에서 제시하고자 하는 알고리즘의 pseudo-code는 다음과 같다.

```
/* i: number of inputs ports, j: number of output port */
for(all i) request{ i }{ /* Request phase */
if(lengthy_input[i] == YES) { /* is a lengthy input */
        accept(i, j); /* use the same matching in this time slot */
lengthy_input[i] = NO; } /* change to not lengthy input */
else for(all j){
        if(num_cells[i,j] >0){ /* queued cells for output j */
        request(i,j); /* request to output j */
}

for(all j) grant(j) { /* Grant phase */
if(num_request[j] >0){ /* received any request? */
        i = select_request(j) /* select request */
        send_grant(i,j); /* send grant from j to i */
}}

for(all i)accept(i){ /* accept phase */
if(num_grant[i]>0){
        j = select_grant(i); /* select the grant j for acceptance */
        accept_grant(i,j); /* accept the grant j */
        prioirty[i] = (j+1); /* set the highest priority of input I */
        priority[j] = (i+1); /* set the highest priority ofouput j */
        if(num_cells[i,j] > threshold) {
                lengthy_input[i] = YES;
}}}
```

[그림 2] VOQ 상태를 고려한 스케쥴링 알고리즘

이 알고리즘은 request, grant, accept 세가지 상태를 가진다. 그리고 세가지 상태는 다음과 같다.

1) Request

Lengthy-input이 아닌 각각의 입력 단은 자신이 저장하고 있는 셀에 대한 모든 출력 단으로 request 신호를 보낸다.

2) Grant

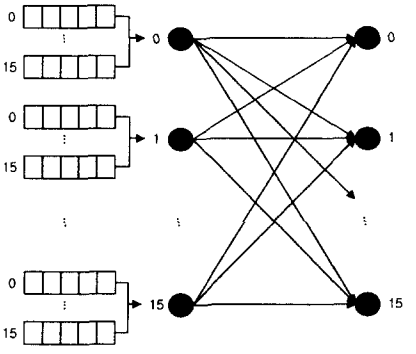
만약 출력 단에서 request 신호를 받았을 경우에는 가장 priority 가 높은 입력 단에서부터 시작하는 round-robin 방식으로 grant 할 입력 단을 선택하고 grant 신호를 보낸다. 만약 grant 신호가 accept되면 출력 단에서는 현재 priority가 가장 높은 입력 단의 다음 입력 단에 가장 높은 priority를 부여한다.

3) Accept

만약 입력 단에서 grant 신호를 받았을 경우에는 priority가 가장 높은 출력 단에서 시작하는 round-robin 방식으로 accept 할 출력 단을 선택한다. Grant 신호의 accept가 이루어 지면 accept한 입력 단에서 grant한 출력 단으로 셀을 전송 시킨다. 만약 입력 단의 큐가 threshold 보다 많은 셀을 가지고 있을 경우 입력단과 그것을 accept 한 출력 단은 다음 time slot 때 까지 lengthy-input 상태로 남아 있다.

4. 성능 평가

본 논문에서 제시한 알고리즘의 성능을 분석하기 위해서 기존의 iSlip 알고리즘과 시뮬레이션을 통하여 비교해 보았다.



[그림 3] 시뮬레이션 switch 구조

시뮬레이션 시 사용할 switch 구조를 [그림 3]에 나타내었다. 16*16 크기의 VOQ 방법을 사용하는 switch 이다.

시뮬레이션 시 다음과 같은 것을 가정하였다.

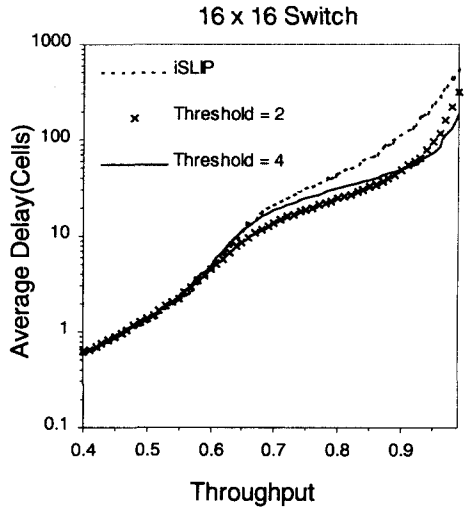
- 모든 입력 단과 출력 단의 링크 capacity는 동일하다.
- 입력 단의 패킷은 independent, identically, distributed(i.i.d) Bernoulli 분포를 가지며 각각의 출력 단에 대해서 같은 비율로(uniform distribution) 도달하도록 하였다.
- 시뮬레이션은 50,000 time slot 동안 수행되었다.

시뮬레이션의 결과는 이 알고리즘의 iSlip에 대한 패킷의 평균 지연 시간에 대해서 나타내었다.

앞에서 제시한 조건에서 이루어진 시뮬레이션에 대한 결과를 [그림 4]에 나타내었다.

1) Threshold가 2일 때는 Throughput이 0.6보다 클 때에는 iSlip보다 더 좋은 성능을 보였으면 Throughput이 0.6 이하 일 때에는 iSlip 이 더 좋은 성능을 보였다. 그리고 Throughput 이 0.6 에서 0.95 사이에서는 지연 시간의 차가 점점 커졌으며 Throughput이 0.95보다 클 때에는 그 차이가 점점 줄어 들었다.

2) Threshold가 4 일 때는 Throughput 이 약 0.67 보다 클 때 iSlip 보다 좋은 성능을 보이며 Throughput 이 0.67 보다 작을 때 에는 iSlip 이 더 좋은 성능을 보였다.
3) Throughput 이 0 에서 0.6 사이에서는 Threshold 가 2 인 경우와 Threshold 가 4 인 경우의 성능이 거의 동일하다. 그러나 Throughput 이 0.6 에서 0.9 사이인 경우에는 Threshold 가 2 일 때 더 좋은 성능을 보였으며 Throughput 이 0.9 이상 일 때는 Threshold 가 4 일 때 더 좋은 성능을 보였다.



[그림 4] 시뮬레이션 결과

위의 결과에서 보듯이 낮은 Throughput 에서는 본 논문에서 제시한 알고리즘의 성능은 iSlip 알고리즘보다 떨어진다. 그러나 높은 Throughput 에서는 iSlip 알고리즘보다 좋은 성능을 보였다. 그리고 낮은 Throughput 에서는 threshold 값이 작을 때 이 알고리즘은 좋은 성능을 보이며 높은 Throughput 에서는 threshold가 높을 때 더 좋은 성능을 보였다.

5. 결론

본 논문에서는 VOQ 방식을 사용하는 스위치에서 패킷의 지연 시간을 줄이기 위한 새로운 알고리즘을 제시하였다. 이 알고리즘은 큐에 존재하는 셀의 수에 대한 threshold를 설정하고 threshold 보다 셀의 수가 많은 큐와 그렇지 않은 큐의 처리를 다르게 하는 알고리즘이다. 시뮬레이션 결과에서 보듯이 이 알고리즘은 기존의 알고리즘의 패킷 지연 시간을 단축 시켰다.

[참고 문헌]

1. M ischa Schwartz, " Broadband Integrated Networks" , Department of Electrical Engineering Columbia Univ
1. Hui,J. , Arthurs, E., " A broadband packet switch for integrated transport " , IEEE J. Selected Areas Communications, vol.5, no. 8 1987.10
2. Karol, M., Hluchyj, M., " Queueing in high - performance packet -switching" , IEEE J.Selected Areas Communications, vol.6 1998.12
3. Chen, M., Georganas, N.D., " A fast algorithm for multi-channel/port traffic scheduling " , Proc IEEE Supercom/ICC' 94
4. Anderson,T., Owicki,S., Saxe,J. and Thacker, C., " High speed switch scheduling for local area network " , ACM Trans. On Computer System. 1993.11
5. Cisco Systems GSR 12000 Technical Product Description, http://www.cisco.com/warp/public/733/12000/index.html