가변 길이 패킷을 지원하는 스위칭 패브릭의 설계

(Design of Switching Fabric Supporting Variable Length Packets)

류 경 숙 † 김 무 성 †
(Kyoungsook Ryu) (Musung Kim)

최 병 석 ††
(Byeongseog Choe)

요 약 최근 인터넷 망에서 고속 스위칭을 위하여 입출력 인터페이스 간 패킷 전송에 있어서 스위칭 패브릭이 적용되고 있다. 기존의 구조들은 가변 길이 IP 패킷의 처리에 ATM 스위칭 패브릭을 그대로 적용하기 위해 패킷을 일정 크기로 분할 및 재조립하거나 크로스포인트에 버퍼를 두는 방식을 고려하고 있어 시스템에 부하를 가져온다. 본 논문에서는 데이타 메모리 평면과 스위칭 평면을 분리하여 패킷 데이타는 독립된 메모리 구조에 저장하고 동시에 메모리 주소 포인터 부분만 스위칭 패브릭을 통과하도록 하는 새로운 스위치 구조를 제안한다. 스위칭 패브릭은 주소 포인터와 기본적인 정보를 포함하는 아주 작은 미니 패킷이 통과하게 되는데 이것은 가변길이 패킷들이 경쟁하는 스위칭 패브릭과 비교할 때 탁월한 스위칭 속도를 가진다.

키워드 : 초고속 스위치, 스위칭 패브릭, 가변 길이 IP 패킷, 입력 큐잉, 출력 큐잉

Abstract The switching fabric used to make high speed switching for packet transfer between input and output interface in recent internet environments. Without making any changes in order to remain ATM switching fabric, the existing structures should split/reassemble a packet to certain size, set aside cross-point buffer and will put loads on the system. In this paper, we proposed a new switch architecture, which has separated data memory plane and switching plane packet data will be stored on the separate memory structure and simultaneously only the part of the memory address pointers can pass the switching fabric. The small mini packets which have address pointer and basic information would be passed through the switching fabric. It is possible to achieve the remarkable switching performance than other switch fabrics with contending variable length packets.

Key words : High-speed switch, Switching fabric, Variable length IP packet, Input queuing, Output queuing

1. 서 론

최근 인터넷 환경에서의 초고속 스위치의 발전 방향은 기가 비트 스위칭이 가능하도록 하고자 스위치 내부의 입/출력 인터페이스간 패킷 전송에 있어서 ATM 스위치와 같은 고속의 하드웨어 스위칭 기법을 도입하고 있다[1,2]. Abacus와 같은 ATM 스위치는 53 옥텟(Octet)의 고정 셀 스위칭을 하는데 반하여 인터넷 망에서는 가변 길이 패킷을 서비스 해야 하는 차이점이 있다. 일반적으로 이더넷 프레임은 64 바이트에서 1518 바이트의 길이를 가진다. 스위칭 패브릭(switching fabric)에서 이러한 가변 길이 패킷을 효율적으로 스위칭 하기 위해서는 새로운 형태의 스위치 구조가 필요하다.

현재까지 스위칭 패브릭을 적용하여 가변 길이 IP 패킷을 서비스하기 위한 연구에는 두 가지 접근 방법이 있다. 하나는 패킷들이 입력 단에서 스위칭 패브릭에 진입하기 전에 패킷들을 일정한 셀 단위로 분할하여 전송하고 출력 단에서 재조립하는 방법이며 다른 하나는 가변 길이 패킷을 별도의 가공 없이 서비스 할 수 있도록 스위칭 패브릭 내에 가변 길이 패킷을 저장하기 위한 저장 공간(cross-point buffer)을 확보하는 방법이다[3-6]. 전자는 패킷을 분할하고 재조립 하기 위한 시간들이 시스템에 부하로 작용할 수 있고 후자는 단편화(fragmentation) 문제는 해결할 수 있지만 패킷이 스위칭 패브릭을 통과하는 동안에도 지속적으로 가변 길이 패킷들을 서비스해야 하므로 스위칭 패브릭의 속도 면에서 역시 문제가 된다. 최근 고속 스위치에서 하드웨어적 구현이 용이한 입력 버퍼 형태의 다양한 VOQ (Virtual Output Queued) 방식들이 제시되고 있으나 공유 메모리 기반의 스위치 구조로 인해 복잡한 중재 알고리즘들과 스케쥴링 기법들을 필요로 하는 단점이 있다[7-9]. 본 논문에서는 가변 길이 패킷 스위칭을 효

· 이 논문은 제34회 추계학술대회에서 '가변 길이 패킷을 고려한 스위칭 패브릭의 설계'의 제목으로 발표된 논문을 확장한 것임

† 정 회 원 : 동국대학교 정보통신공학과
ksryu@dongguk.edu
moosung@dongguk.edu
†† 정 회 원 : 동국대학교 정보통신공학과 교수
bchoe@dongguk.edu
(Corresponding author임)

논문접수 : 2007년 12월 7일
심사완료 : 2008년 2월 14일

정보과학회논문지 : 컴퓨팅의 실제 및 레터 제14권 제3호(2008.5)

율적으로 하기 위한 새로운 스위치 구조를 제안 한다. 제안한 구조는 독립적 메모리를 사용하도록 하여 기존의 입력 버퍼형 스위치의 복잡한 중재 알고리즘(arbitration algorithm)을 필요로 하지 않으며 현재 출력 큐 스위치에서만 적용되고 있는 QoS를 입력 큐 방식에 제공하는 부분을 고려한다.

본 논문의 구성은 다음과 같다. 2장에서는 제안한 스위치 구조에 대해 설명하고 3장에서는 출력 포트에서의 동작을 자세히 설명한다. 4장에서는 제안한 구조의 성능을 평가하고 마지막으로 5장에서는 결론을 맺는다.

2. 제안한 구조

제안한 스위치는 크게 두 개의 평면(plane)으로 구성된다. 하나는 데이타 제어 평면(data control plane) 부분이고 다른 하나는 주소 포인터 스위칭 평면(address pointer switching plane)이다.

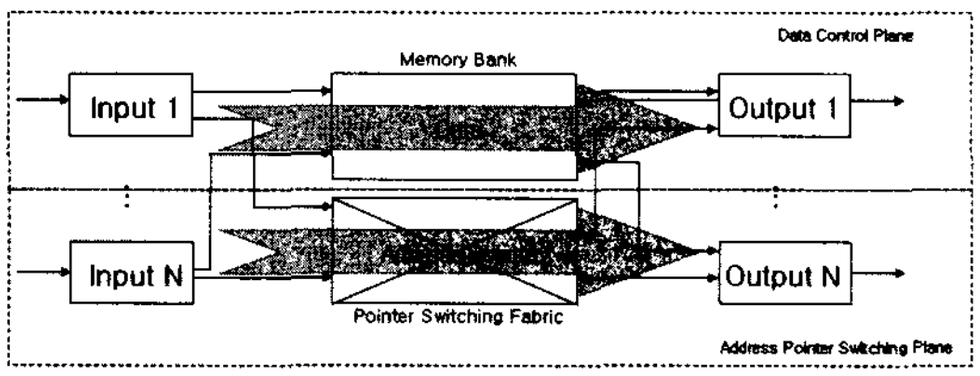

그림 1 제안한 스위치 구조

그림 1과 같이 입력 인터페이스에 입력된 패킷은 라우팅 테이블 룩업 후 메모리 뱅크에 저장되며 동시에 저장된 패킷의 주소 포인터(Address Pointer) 부분만을 스위칭 패브릭을 사용해서 스위칭 하도록 한다. 스위칭 패브릭은 구현이 간단하면서도 확장성이 우수한 ATM 스위치의 스위칭 패브릭을 사용한다. 패킷은 입력 포트와 출력 포트 번호로 참조되는 독립적인 메모리들의 집합인 메모리 뱅크(Memory Bank)에 저장한다. 각 라인 카드에 라우팅 테이블과 포워딩 엔진을 두어 라우팅 기능을 각 라인 카드에 분산하여 병렬적인 처리가 가능하도록 하고 있다. 성능 향상을 위해 출력 큐는 ASIC 형태의 자동 정렬 큐(ASQ: Auto Sorting Queue)로 구현하여 스케줄링 알고리즘의 복잡성을 회피하고자 한다[10]. 기존의 스위치 구조들에서는 패킷 데이타 전체가 스위칭 패브릭을 통과하도록 하였으나 제안한 구조에서는 패킷의 메모리 뱅크 내 주소 포인터 값을 가지는 미니 패킷(mini packet)들만이 스위칭 패브릭을 통과하도록 설계하였다.

2.1 포인터 스위칭 패브릭

스위치에 진입한 모든 패킷은 입력 단에서 라우팅 테이블 룩업을 위해서 헤더 정보를 읽히게 되는데 이 과정에서 필요한 계산들을 미리 하게 되면 효과적이기 때문에 제안한 스위치 구조에서는 QoS 처리를 위한 스케줄링과 출력 큐에서의 정렬을 위한 정보는 입력 포트에서 처리하도록 한다. 출력 포트에서 조절할 경우 보다 정확하고 안정적일 수 있지만 헤더 정보를 한 번 더 읽어야 하고 고속 스위치에서 QoS를 복잡하게 계산하는 것은 지연을 초래할 수 있다.

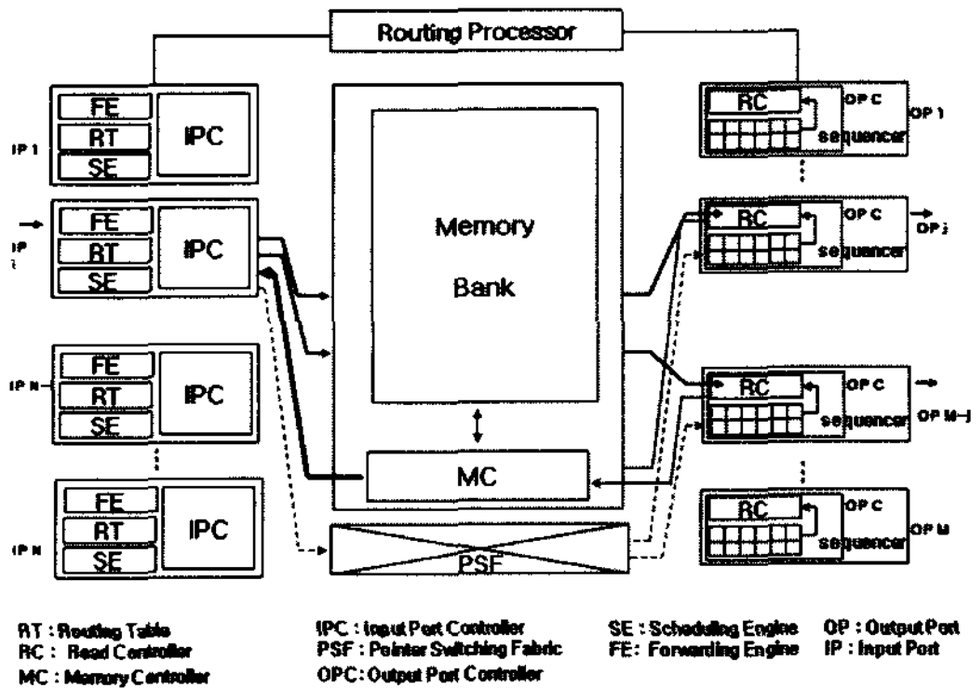

그림 2 포인터 스위칭 패브릭

제안한 스위치의 동작을 간략하게 살펴보면 다음과 같다.

① 스위치의 입력 포트에 패킷이 들어오면 라우팅 테이블을 룩업(lookup)한다. 이때 목적지 주소에 따라서 출력 포트가 결정된다.

② 입력 포트와 출력 포트로 참조되는 메모리에 패킷을 저장하고, 패킷의 메모리 뱅크 내 메모리 주소 포인터와 ToS필드를 이용한 우선순위, 스위칭 패브릭 내부 라우팅을 위한 기본적인 정보들로 구성된 미니 패킷을 생성하여 주소 포인터 스위칭 패브릭으로 보낸다.

③ 주소 포인터 스위칭 패브릭을 통과해 출력 포트에 도착한 미니 패킷은 출력 큐인 ASQ에 의해서 입력 포트에서 지정한 우선순위에 따라 시퀀서의 오른쪽부터 정렬(sorting) 된다.

④ ASQ에서 미니 패킷이 전송될 순서(HOL: Head Of Line)가 되면 메모리 포인터 주소를 참조해서 메모리 뱅크에서 패킷을 읽어 오고 참조한 메모리 인덱스 부분은 해제(free) 된다.

⑤ 메모리 뱅크에서 읽어온 패킷을 다음 홉(hop)으로 전송한다.

2.2 미니 패킷

제안한 구조의 내부에서 사용하는 미니 패킷은 그림 3과 같이 구성된다.

입력 포트로 유입된 패킷은 IPC(Input Port Controller)에서 내부 전용 패킷인 미니 패킷을 생성하여 제안한 포인터 스위칭 패브릭(PSF, Pointer Switching Fabric)을 통해서 해당 전송 큐로 전송된다. 미니 패킷은

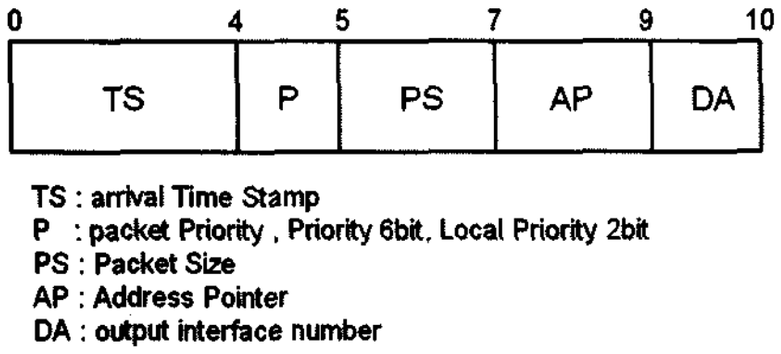

그림 3 미니 패킷

도착 시각을 나타내는 타임 스탬프(TS: arrival Time Stamp) 4 바이트, 우선 순위(Priority) 6 비트와 로컬 우선 순위(Local Priority) 2 비트를 포함하는 패킷 우선 순위(P: Packet Priority) 1 바이트, 패킷의 크기(PS: Packet Size) 2 바이트, 메모리 뱅크 내의 주소 포인터(AP: Address Pointer) 2 바이트, 출력 포트 주소(DA: output interface number) 1 바이트로 구성된다. 여기서 우선 순위(P) 6 비트는 노드에서의 클래스별 가중치를 적용하고 로컬 우선 순위(LP) 2 비트는 drop된 횟수를 고려하여 우선 순위를 높여 주는 역할을 한다.

2.3 메모리 뱅크

가변 길이 패킷을 저장하기 위해 제안한 메모리 뱅크(Memory Bank)의 구조는 그림 4와 같다. 각 입력 포트에 연결된 독립적인 메모리 슬롯들은 출력 포트의 개수만큼의 메모리들이 순서대로 버스로 연결되어 있으며 동일한 출력 포트에 속하는 메모리들은 다시 해당 출력 포트로 향하는 버스로 연결되어 있다.

메모리 뱅크의 접근에 있어서 메모리에 쓸 경우에는 입력 포트 번호(제안한 구조에서는 i로 참조)를 기준으로 저장되며 메모리에서 읽어낼 경우에는 출력 포트 주소(제안한 구조에서는 j로 참조)를 기준으로 읽어낸다. 예를 들어서 입력 포트 i로부터 들어온 패킷 k는 헤더 정보를 읽힌 후 데이타 부분이 추출되어 메모리 뱅크에 저장 된다. 입력 i로부터 들어 오는 패킷의 데이타는 입력 인터페이스 i에 연결된 메모리 슬롯 내에서 패킷의 출력 포트 j에 해당하는 위치의 메모리에 저장 된다. 즉, 비어 있는 메모리의 위치에 해당 패킷의 데이타를 저장하고 각 메모리에 할당된 메모리 주소(index) i, j에 대한 포인터를 미니 패킷에 넘겨준다. 출력 포트에서 전송될 패킷은 출력 포트 j에 연결된 버스에서 자신이 들어왔던 입력 포트 번호에 해당하는 메모리에서 메모리 포인터를 참조하여 패킷을 읽어 내면 된다. 이 과정에서 입력 i로 들어오는 패킷에 할당되는 메모리 셀 내의 주소는 Memory Controller 내의 Idle Address Controller에 의해서 수행된다. 전체 메모리 셀을 논리적으로 하나의 메모리 뱅크(Memory Bank)로 보고 할당된 메모리 주소를 관리하며 입력된 패킷에 대한 새로운 메모리에 할당과 전송된 패킷에 의해서 반납된 메모리 공간의 제어를 Memory Controller가 수행한다. IPC에 의해서 메모리에 write 시그널이 들어오면 Idle Address Controller에 유지되는 비어 있는 메모리 주소 리스트(Idle Address List)에서 메모리 주소를 할당하고 만약 OPC에 의해서 read 시그널이 도착하면 Idle Address Controller에 유지되는 비어 있는 메모리 주소 리스트에 메모리 주소를 반납한다.

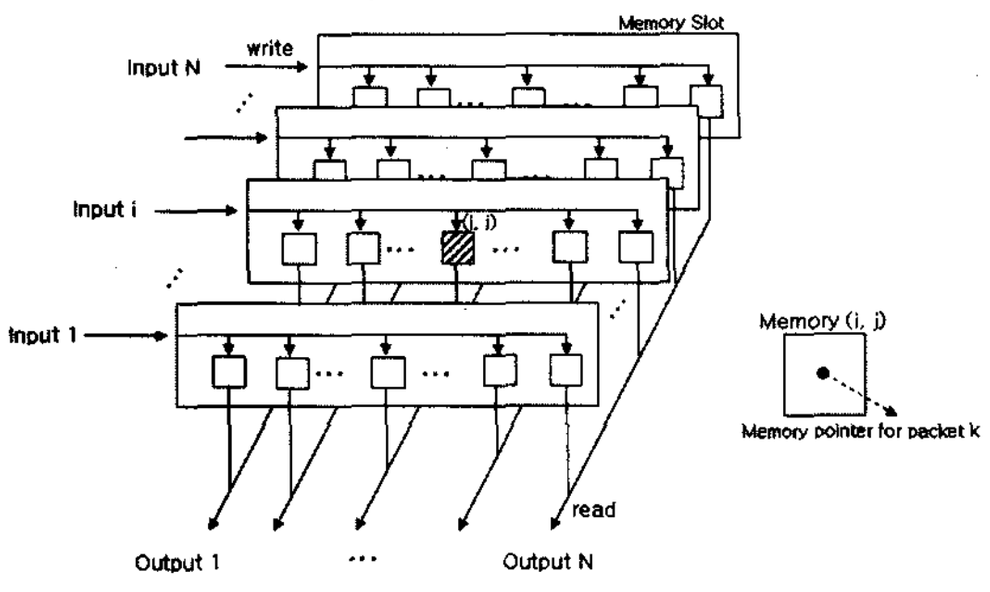

그림 4 메모리 뱅크

3. 출력 포트

PSF를 통해서 출력 포트로 스위칭 된 미니 패킷은 ASQ에 정렬 필드로 작용하는 타임 스탬프와 우선 순위, 패킷 크기를 기준으로 자동 정렬 되어 오른쪽에서 왼쪽으로 오름차순 저장이 되며 전송 차례가 되면 ASQ의 가장 오른쪽 쌍의 주소(HOL)부터 RC(Read Controller)에 넘겨준다. 높은 우선 순위의 미니 패킷들이 항상 낮은 우선 순위의 미니 패킷들보다 오른쪽에 저장되며 RC에 의해서 가장 먼저 접근된다. 출력 포트 제어기(OPC, Output Port Controller)에서의 동작은 그림 5와 같다.

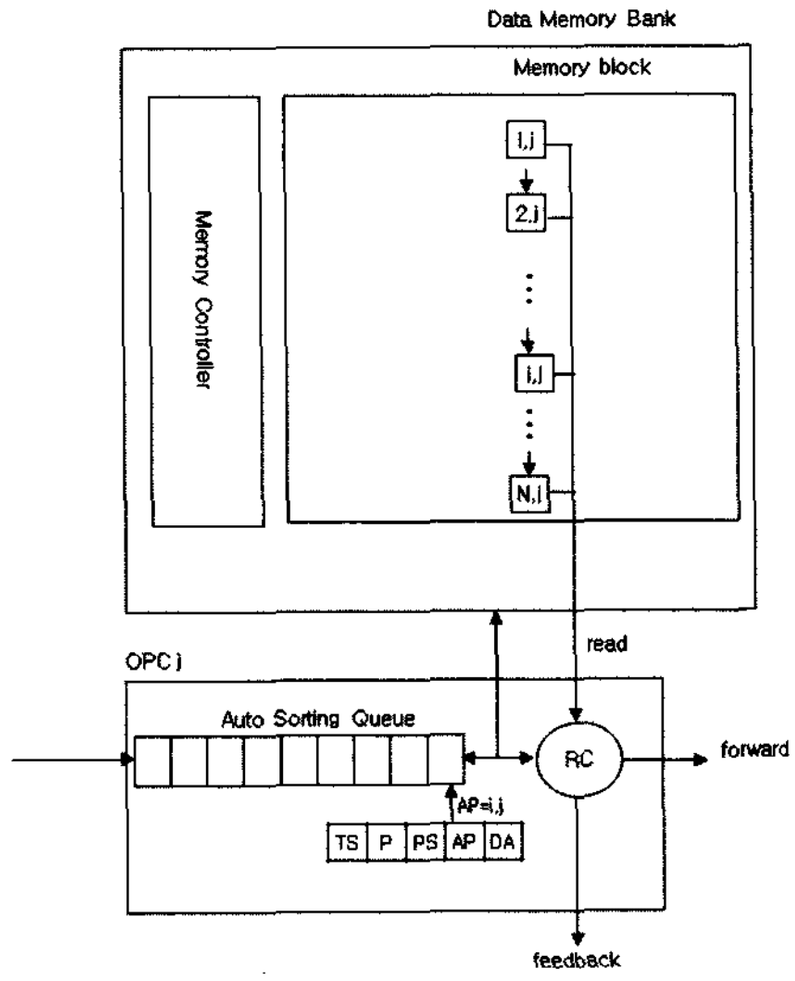

그림 5 출력 포트에서의 동작

스위칭 패브릭을 통하여 출력 포트로 온 미니 패킷들은 입력 포트에서 결정된 우선 순위에 의해 ASQ에서 자동 정렬되고 이를 RC에 의해서 HOL부터 꺼내서 메모리 뱅크의 데이타를 참조하여 전송하며 자세한 동작 과정은 다음과 같다.

출력 포트 j에서 RC에 의해서 선택된 HOL 미니 패킷의 주소 포인터를 통해서 OPC는 read 시그널을 Memory Controller에 보내고 해당하는 메모리 (i,j)에서 메모리 포인터(index)를 통해 데이타를 읽어온다. 메모리에서 읽어 온 데이타를 새로운 헤더 정보와 결합하여 출력 인터페이스로 전송한다. 전송이 정상적으로 끝나면 메모리 뱅크의 해당 메모리 포인터를 Memory Controller 내의 Idle address controller에 반환한다. 폐기된 패킷이 발생하면 local priority 값을 설정해서 IPC로 피드백 시키며 우선 순위를 재조정한다. 출력 포트의 ASQ에서 미니 패킷이 우선 순위(TS_{n+1}, P, PS)로 정렬되는 동작은 결과적으로 WFQ를 사용하는 것과 같은 효과를 가진다. 따라서 제안한 구조에서는 별도의 소프트웨어적인 스케줄링 알고리즘을 사용하지 않는다.

4. 성능 평가

4.1 시뮬레이션 환경

모든 입력 포트와 출력 포트의 링크 용량은 동일하고 모든 버퍼의 속도는 외부 회선 속도와 동일하다. 출력 포트에 사용되는 ASQ의 크기는 256이다. 각 입력 포트로 들어오는 패킷들은 베르누이(Bernoulli) 확률 분포를 따르며 각 트래픽 원은 모든 출력 포트에 대해 동등하게 패킷을 발생시키는 uniform traffic을 가정한다. 각 입력 포트에서 패킷이 들어올 확률은 λ이고 μ 확률로 전송한다고 가정한다.

표 1 트래픽 클래스 별 구성 비율

Class	Traffic type	Portion (%)
1	Voice	18
2	Video	15
3	Call-Signaling	5
4	Network Control	5
5	Critical Data	27
6	Bulk Data	4
7	Best-Effort	25
8	Scavenger	1

4.2 시뮬레이션 결과

그림 6은 제안한 구조에서 각 패킷 클래스 별 평균 지연 시간에 대한 실험 결과이다. 모든 트래픽 클래스에 대해서 만족할 만한 평균 지연 시간을 가짐을 알 수 있으며 특히 우선 순위가 높은 트래픽에 대한 부하의 최대치에서도 탁월한 성능을 보여 준다.

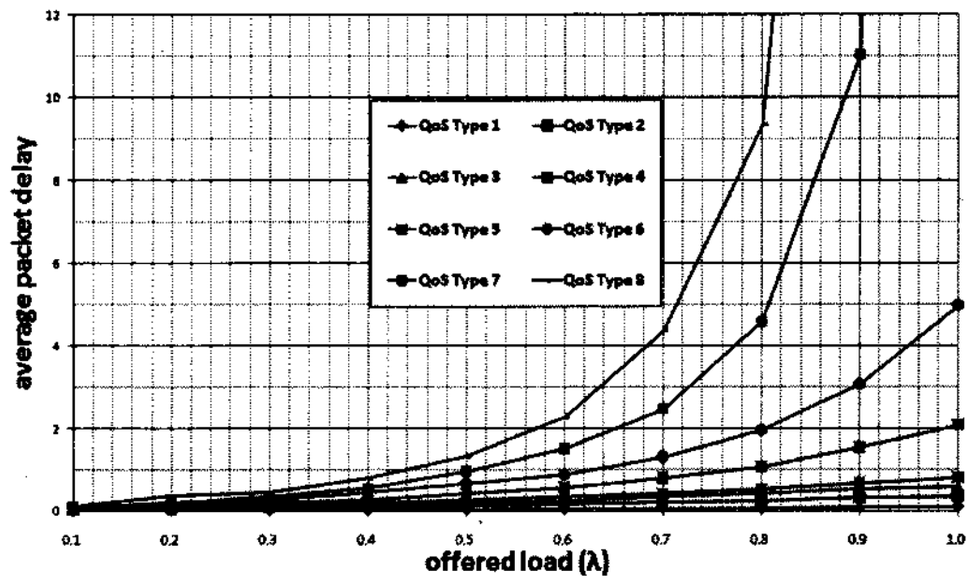

그림 6 트래픽 클래스 별 평균 지연 시간

제안한 구조를 적용하여 포화 상태에서 입력 부하 λ가 0.9일 때 메모리 뱅크의 임의의 입력 i, 출력 j에 해당하는 독립적인 메모리의 크기에 따른 각 트래픽 클래스 별 패킷 손실률은 그림 7에 나타나 있다.

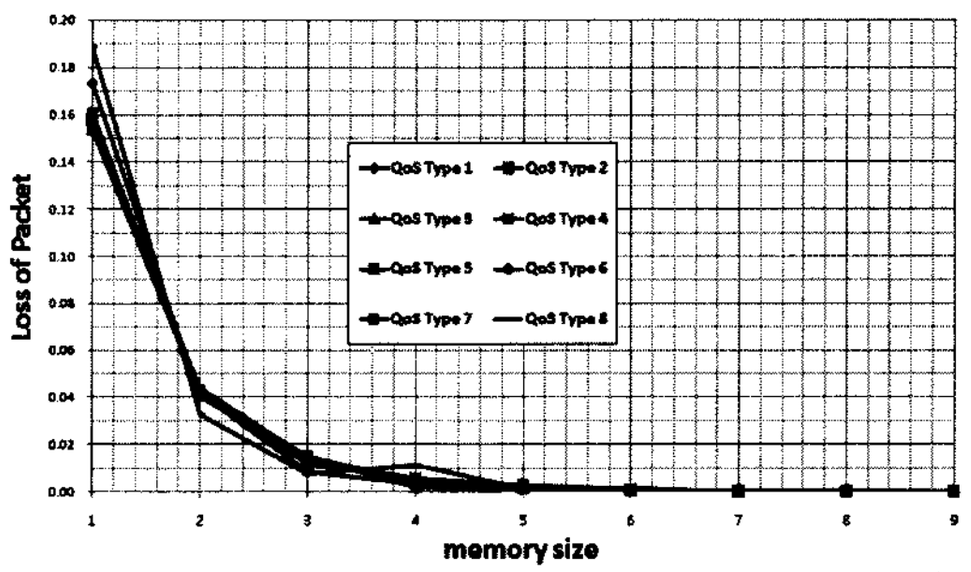

그림 7 메모리 크기에 따른 패킷 손실율

그림 8은 제안한 구조에서 입력 부하가 0.8로 일정하고 스위치 크기 N이 2, 4, 8, 16, 32, 64로 증가함에 따라 변화하는 평균 지연 시간을 보여 준다. 결과적으로 제안한 구조는 스위치 크기가 증가 하더라도 일정한 지연 시간을 보장하는 우수한 확장성(Scalability)을 가지고 있음을 확인할 수 있다.

그림 9는 주어진 시뮬레이션 환경에서 제안한 구조(VOIQ)와 기존 공유 메모리 기반의 MIQ(Multiple Input

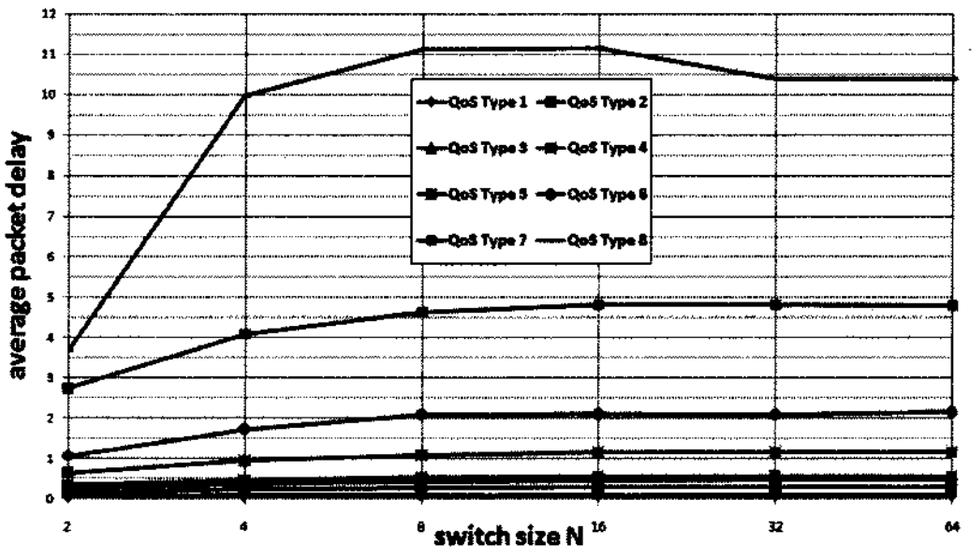

그림 8 스위치 크기에 따른 평균 지연 시간

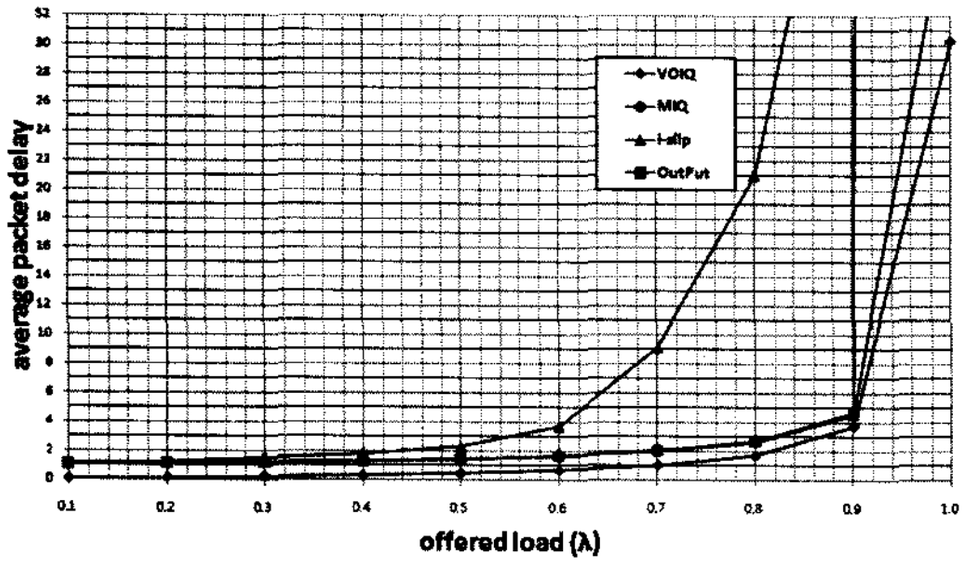

그림 9 평균 지연 시간 비교

Queue), Output, i-slip 방식들을 평균 지연 시간의 관점에서 비교한 실험 결과이다.

제안한 구조에서는 기존의 구조들에서 사용된 복잡한 중재 알고리즘이나 스케줄링 알고리즘을 별도로 구현하지 않았음에도 불구하고 모든 입력 부하(λ)의 조건에서 기존 공유 메모리 기반의 구조들에 비해 상대적으로 낮은 평균 지연 시간을 보여 주고 있다. 이는 제안한 구조에서는 입력 포트에서 패킷 헤더를 읽어서 필요한 작업을 하는 동안 우선 순위 값을 할당하여 미니 패킷에 삽입 하는 작업을 수행하므로 출력 포트에서는 할당된 우선 순위 값에 따라 ASQ에서 자동 정렬 되기 때문에 별도의 중재 알고리즘이나 스케줄링 없이도 우수한 성능을 얻을 수 있는 것이다.

5. 결 론

본 논문에서는 인터넷 망에서 가변길이 IP 패킷 스위칭과 고속 포워딩을 고려한 스위치 구조를 제안하였다. 입력 포트에서 IP 패킷의 데이타 부분은 독립적인 메모리에 저장하고 스위칭 패브릭으로는 메모리 주소 포인터만이 통과하도록 하여 데이타 저장과 스위칭이 병렬적으로 처리될 수 있도록 하였다. 입/출력 포트 각각에 독립적인 메모리를 사용하여 스위칭 패브릭 진입을 위한 분할 및 재조립 과정이 필요 없으며 공유 메모리 방식에서의 VOQ와 같은 복잡한 중재 알고리즘을 사용할 필요가 없다. 출력 포트 제어기 부분에 ASQ를 사용하여 하드웨어 속도의 공평 스케줄링(Fair Scheduling)이 가능하다. 성능 평가 결과 제안한 구조는 초고속 인터넷 망의 고속 패킷 스위칭과 사용자 요구를 동시에 만족할 수 있는 구조로 적합함을 확인하였다.

참 고 문 헌

[1] H. Chao and B. Choe, "Design and Implementation of Abacus Switch: A Scalable Multicast ATM Switch," IEEE JSAC. Vol.15, No.5, Jun., 1997.

[2] H. Chao and B. Choe, "Design and Analysis of a Large-Scale Multicast Output Buffered ATM Switch," IEEE/ACM Trans. Networking, Vol.3, No.2, Apr., 1995.

[3] E. Oki and N. Yamanaka, "Scalable Crosspoint Buffering ATM Switch Architecture Using Distributed Arbitration Scheme," in Proc. IEEE ATM '97 Workshop, 1997.

[4] C. Minkenberg and T. Engbersen, "A Combined Input and Output Queued Packet-Switched System Based on PRIZMA Switch-on-a-Chip Technology," IEEE Commun. Mag., pp. 70-77, 2000.

[5] K. Yoshigoe and K. Christensen, "An Evolution to Crossbar Switches with Virtual Output Queuing and Buffered Cross Points," IEEE Network, pp. 48-56, Sep./Oct., 2003.

[6] "Cisco 12000 Series - Gigabit Switch Routers," http://www.cisco.com/warp/public/cc/pd/rt/12000/prodlit/gsr_ov.pdf

[7] J. Garcia, L. Cerda, J. Corbal and M. Valero, "A conflict-free memory banking architecture for fast VOQ packet buffers," in Proc. IEEE GLOBECOM '03, pp. 4158-4162.

[8] J. Wang and K. Nahrstedt, "Parallel IP packet forwarding for tomorrow's IP routers," IEEE Workshop on High Performance Switching and Routing, pp. 353-357, 2001.

[9] C.Kolias and L. Kleinrock, "The Odd-Even input queueing ATM switch: performance evaluation," in Proc. IEEE ICC '96, Vol.3, pp. 1674-1679, 1996.

[10] H. Jonathan Chao, "A VLSI Sequencer Chip for ATM Traffic Shaper and Queue Manager," IEEE JSAC, Vol.27, No.11, Nov, 1992.