

QoS가 보장된 멀티캐스트 서비스를 위한 Blocking Avoidance 셀 할당 기법과 Virtual Threshold 기법을 이용한 CICQ 스위치 구조

정희원 김 경 민*, 정 현 덕*, 종신회원 이 재 용*

A CICQ Switch Architecture Using Blocking Avoidance & Virtual Threshold Scheme for QoS Guaranteed Multicast Service

Kyungmin Kim*, Hyunduk Jung* *Regular Members*, Jaiyong Lee** *Lifelong Member*

요 약

최근 다양한 멀티미디어 서비스의 등장으로 멀티캐스트를 통한 콘텐츠의 전송이 빠르게 늘어나고 있으며 이를 처리하기 위한 스위치 기술의 중요성이 높아지고 있다. CICQ 스위치 구조는 cross point의 버퍼를 이용하여 HoL blocking 확률을 낮출 수 있으며 스케줄링이 간단하다는 장점이 있지만 트래픽의 부하가 높아짐에 따라 멀티캐스트 트래픽의 처리율이 급격히 떨어질 수 있다는 단점이 있다. 이를 해결하기 몇몇의 멀티캐스트 셀 할당 기법들과 입력 포트와 cross point 사이의 스케줄링 기법들이 제안 되었지만 여전히 멀티캐스트 트래픽의 충분한 처리율을 보장 하지 못한다. 따라서 본 논문에서는 BA 셀 할당 기법과 VT 기법을 제안하여 멀티캐스트 트래픽의 HoL blocking 확률을 낮추고 cross point 버퍼의 전용 공간을 확보하여 멀티캐스트 트래픽의 처리율을 보장 할 수 있도록 하였다. 또한 시뮬레이션을 통해 제안된 방법을 이용하여 QoS를 향상시킬 수 있음을 확인 하였다.

Key Words : Multicast Switch, CICQ, Cell Assignment, Blocking Avoidance, Virtual Threshold,

ABSTRACT

Recently the multicast based contents transmission is rapidly increasing due to the various multimedia services and the importance of switching technology to handle it is increasing as a consequence. Though the CICQ architecture has advantages that reduction of HoL blocking probability and simple scheduling using cross point buffer, it has disadvantage that the processing rate of multicast traffic can be significantly degraded corresponds to the traffic load increment. Several schemes have been proposed to solve this problem however they still can't provide enough processing ratio for multicast traffic. Therefore this paper proposes the BA cell assignment scheme and the VT scheme, and the processing rate of multicast traffic can be guaranteed by reducing the HoL blocking probability of multicast traffic and reservation of cross point buffer. Also simulation results verify that using the proposed scheme, the QoS of multicast service can be improved.

I. 서 론

최근 스마트폰, PMP (Portable Multimedia Player)

등과 같은 휴대용 멀티미디어 장비가 빠르게 보급되고 있으며 WiFi (Wireless Fidelity) 및 3G (3rd Generation) 이동통신망과 같이 이들 장비를 이용하여

※ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(No. 2010-0016636)

* 연세대학교 전기전자공학과 유비쿼터스 네트워크 연구실 (gamesety@yonsei.ac.kr, psychblue@yonsei.ac.kr, jyl@yonsei.ac.kr)

논문번호 : KICS2010-10-482, 접수일자 : 2010년 10월 11일, 최종논문접수일자 : 2011년 1월 27일

시간과 장소에 구애받지 않고 인터넷에 접속할 수 있는 환경이 빠르게 보급되고 있다. 이에 따라 실시간 방송, 텔레컨퍼런스, 온라인 게임, 온라인 강의 등과 같은 다양한 실시간 멀티미디어 서비스에 대한 수요와 공급이 나날이 늘어나고 있으며 서비스와 기술은 서로간의 발전을 촉진시키고 있다. 한편 소리 및 영상 처리 기술의 발달에 따라 제공되는 콘텐츠의 품질이 높아지고 있으며 이에 따라 인터넷상에서 전송되는 콘텐츠의 크기가 계속해서 증가하고 있다. 그 결과 1990년 이후 인터넷망의 발달로 많이 전송할 수 있는 데이터의 양이 매우 빠르게 증가 하였지만 전송되는 데이터의 평균적인 크기 또한 증가하였기 때문에 인터넷망의 자원은 여전히 충분하지 못하며 모든 사용자들에게 충분한 QoS (Quality of Service)를 보장하기는 어려운 있는 실정이다.

실시간 멀티미디어 서비스는 다수의 사용자를 대상으로 동시에 동일한 콘텐츠를 전달하는 경우가 많기 때문에 다수의 사용자들에게 동시에 콘텐츠를 전송할 수 있는 멀티캐스트 기술을 이용 하면 인터넷 망에서 전송되는 전체적인 데이터의 양을 크게 줄일 수 있다. 그림 1은 하나의 서버에서 4명의 사용자에게 콘텐츠를 전송 할 때 유니캐스트 전송 방법과 멀티캐스트 전송 방법의 사용에 따라 인터넷 망에서 전송되는 트래픽의 양을 비교한 것이다. 유니캐스트의 경우 동일한 패킷들이 동일한 경로를 통해 중복되어 전송되기 때문에 전체적으로 전송되는 패킷의 양이 매우 많아지지만 멀티캐스트의 경우 서버에서는 하나의 패킷만이 전송되며 전송 경로에 위치한 라우터들 중 패킷의 복사가 필요한 라우터에서만 패킷이 복사되어 전송되기 때문에 인터넷 망 전체에서 전송되는 패킷의 양을 크게 줄일 수 있다.

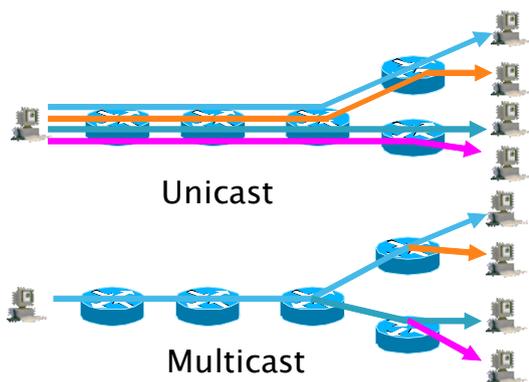


그림 1. 유니캐스트와 멀티캐스트 전송 비교
Fig. 1. Comparison of multicast and unicast transmission

이러한 멀티캐스트 전송기술을 실제 콘텐츠 전송에 이용하기 위해서는 유니캐스트 트래픽과 멀티캐스트 트래픽이 공존하는 상황에서 이들을 효율적으로 처리 하면서 동시에 공정성을 보장 할 수 있는 스위치 기술이 뒷받침 되어야 한다. 이를 위해 VOQ (Virtual Output Queueing) 구조와 이를 발전시킨 CICQ (Combined Input Crosspoint Queueing) 구조를 이용한 기법들이 제안 되었으며 CICQ 구조는 현재 가장 발전된 CMOS (Complementary Metal - Oxide - Semiconductor)기술을 바탕으로 하여 만들어진 구조로 가장 높은 효율을 보이기 때문에 현재 가장 널리 사용되고 있다¹¹⁾.

CICQ 구조의 스위치에서 유니캐스트 트래픽과 멀티캐스트 트래픽을 동시에 처리하는 경우 제한된 수의 멀티캐스트 큐에 멀티캐스트 셀 (스위치의 입력 포트에 들어온 패킷은 스위치 내부에서 작은 단위로 나누어져 처리되는데 이 단위를 셀 이라고 한다) 을 할당하는 방법 그리고 유니캐스트 큐와 멀티캐스트 큐에 대기 중인 셀들을 cross point 버퍼로 스케줄링 하는 방법이 스위치의 효율을 높이고 공정한 서비스를 제공하기 위한 핵심적인 요소가 된다. 먼저 멀티캐스트 셀 할당 기법의 경우 module 셀 할당 기법¹⁷⁾ 과 eigen vector를 이용한 방법¹⁸⁾ 등이 제안 되었지만 modulo 셀 할당 기법의 경우 셀을 제한된 숫자의 멀티캐스트 큐에 균등하게 할당하기 위해 제안된 방법이 기 때문에 HoL blocking 문제를 고려하지 않았으며 eigen vector 방법의 경우 입력 포트에 들어온 멀티캐스트 셀을 동일한 목적지를 가지는 것 들끼리 분류해 놓은 것으로 특정 출력 포트에 향하는 cross point 버퍼가 가득 찼을 경우 해당 목적 포트를 나타내는 eigen vector를 가진 큐를 스케줄링에서 제외하더라도 여전히 다른 큐로 할당된 셀들이 해당 포트를 목적 포트에 가질 수 있기 때문에 HoL blocking 문제를 충분히 해결 하지는 못한다. 다음으로 입력포트의 큐와 cross point 버퍼 사이의 스케줄링 방법으로는 유니캐스트와 멀티캐스트 큐를 순차적으로 처리하는 MURS (Multicast Unicast Round Robin) 방법이 가장 널리 사용되고 있지만 멀티캐스트 셀의 높은 HoL blocking 확률로 인해 멀티캐스트 패킷의 처리율이 빠르게 감소하는 단점을 가지고 있다⁶⁾.

이와 같은 문제를 해결하기 위해 본 논문에서는 입력 포트에서 멀티캐스트 셀을 제한된 멀티캐스트 큐에 할당하는데 있어 HoL blocking 확률을 줄이기 위해 BA (Blocking Avoidance) 셀 할당 기법을 제안하였으며 입력 포트와 cross point 버퍼 사이의 스케줄

링에 있어 멀티캐스트 셀의 처리율을 보장하기 위해 VT (Virtual Threshold) 기법을 제안하였다. 제안된 BA 셀 할당 기법을 이용하면 멀티캐스트 셀의 HoL blocking 확률을 낮출 수 있으며 VT 기법을 이용하면 멀티캐스트 트래픽의 처리율을 보장 하면서 동시에 유니캐스트 트래픽과의 공정성을 확보 할 수 있다.

본 논문의 II장에서는 관련연구를 검토하며 제 III장에서는 제안된 BA 셀 할당 기법 및 VT 기법을 설명한다. 제 IV장에서는 분석을 통해 기존의 방법과 제안된 방법의 성능을 평가하고 제안된 방법의 우수성을 검증하며 마지막으로 제 V장에서는 본 논문의 결론을 맺는다.

II. 관련 연구

2.1 CICQ 스위치 구조

스위치의 구조는 기본적으로 OQ(Output Queuing) 스위치와 IQ(Input Queuing) 스위치로 나누어 볼 수 있다. OQ 스위치의 경우에는 큐가 출력 포트에 존재하기 때문에 입력포트로 들어온 패킷이 별도의 버퍼 없이 바로 처리된다. 따라서 HOL (Head of Line) blocking 문제가 발생하지 않는다는 장점을 가지지만 입력 포트에 별도의 버퍼가 존재하지 않기 때문에 switching fabric의 처리 속도가 입력 포트의 수의 배수만큼 빨라야 한다는 제약이 있어 실제 구현에 어려움이 따른다. IQ 스위치의 경우에는 큐가 입력포트에 존재하기 때문에 switching fabric의 속도가 입력 패킷이 들어오는 속도와 같으면 입력 패킷을 처리하는 것이 가능하여 구현이 간단하다는 장점을 가지지만 HoL blocking 문제로 인해 성능이 저하되는 단점이 있다¹¹⁾.

VOQ (Virtual Output Queuing) 스위치는 IQ 스위치에서 발생하는 HoL blocking 문제를 해결하기 위해 입력포트의 큐를 각 셀의 목적 포트에 따라 나누어 논리적으로 독립된 큐를 가지도록 한 스위치 구조이다²⁾. 유니캐스트 셀의 경우에는 스위치가 N개의 출력 포트를 갖는 경우 각각의 입력 포트에 N개의 큐를 두어 HoL blocking 문제를 완전히 해결 할 수 있지만 멀티캐스트 셀의 경우에는 하나의 셀이 가질 수 있는 출력포트의 조합이 2^N-1 개로 매우 많아지기 때문에 실제로 각각의 입력포트마다 2^N-1 개의 독립된 큐를 두는 것은 비용을 고려할 때 합리적이지 못하며 실제 구현에 있어서 어려움이 따르며 이에 따라 멀티캐스트 셀을 효율적으로 처리하지 못한다²⁾.

CICQ 스위치는 VOQ 구조를 기반으로 하며 CMOS

기술의 발전에 따라 switching fabric의 crosspoint에 버퍼를 둘 수 있게 됨에 따라 개발 되었다. 그림 2와 같이 VOQ 구조를 기반으로 하여 switching fabric의 cross point에 버퍼가 추가된 구조로 입력 포트에서 멀티캐스트 셀을 저장하기 위해 2^N-1 개의 독립된 큐를 사용하지 않고 이보다 훨씬 적은 k개의 ($1 < k \ll 2^N-1$) 큐만을 사용 한다.이 때 멀티캐스트 셀을 위해서 k개의 제한된 숫자의 큐를 사용하게 때문에 HoL blocking이 발생 하게 되지만 cross point에 버퍼에 의해 기존의 VOQ 스위치에 비해 HoL blocking 확률을 크게 낮출 수 있으며 이에 따라 현재까지 제안된 스위치 구조들 중에서 가장 효율이 높은 구조로 각광받고 있다⁴⁾.

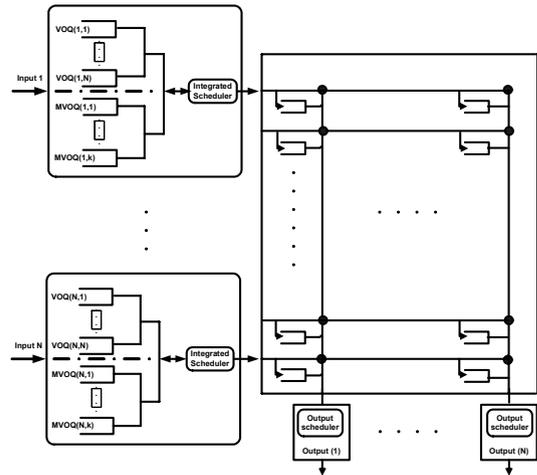


그림 2. CICQ 스위치 구조
Fig. 2. CICQ switch architecture

2.2 멀티캐스트 셀 할당기법

Modulo 셀 할당 기법은 가장 잘 알려진 셀 할당기법으로 각각의 멀티캐스트 셀의 목적지의 수를 바탕으로 큐를 할당한다. 즉 식 1과 같이 각각의 멀티캐스트 셀의 출력 포트의 숫자 N과 멀티캐스트 큐의 수 k를 이용하여 modulo 연산을 수행하며 그 결과 j의 값에 따라 각각의 멀티캐스트 셀을 k개의 큐에 할당한다. 그림 3은 modulo 셀 할당기법의 예시로 N=8, k=4인 경우이며 큐 안의 숫자들은 목적 포트의 번호를 나타낸다. 맨 위에 위치한 큐에는 modulo 연산의 결과 j가 1 즉, 목적포트의 수가 1개 또는 5인 셀이 할당되며, 두 번째 큐에는 j가 2 즉, 목적 포트의 수가 2개 또는 6개인 셀이, 세 번째 큐에는 j가 3 즉, 목적 포트의 수가 3개 또는 7개인 셀이 그리고 네 번째 큐에는 j가 0 즉 목적 포트의 수가 4개 또는 8개인 셀이 저장

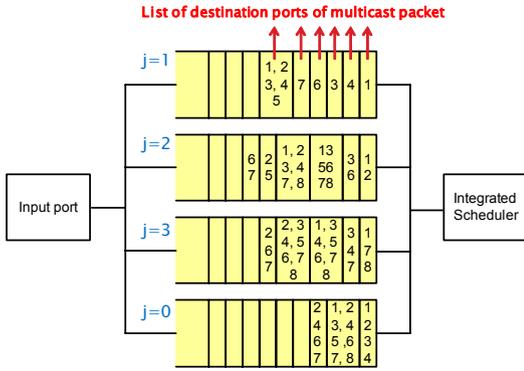


그림 3. Modulo 셀 할당기법
Fig. 3. Module cell assignment scheme

된다⁷⁾.

$$j = N \text{ Modulo } (k) \quad (1)$$

이와 같은 Modulo 셀 할당기법의 경우 구현이 간단하며 각각의 멀티캐스트 큐에 균등하게 멀티캐스트 셀이 할당 된다는 장점이 있지만 각각의 멀티캐스트 셀들의 목적지들끼리는 서로 어떠한 연관 관계가 없기 때문에 HoL blocking 문제로 인해 쉽게 성능이 저하되는 단점이 있다.

Eigen vector를 이용한 셀 할당기법의 경우에는 k 개의 멀티캐스트 큐에 서로 직교하는 각각의 eigen vector를 할당하고 각각의 멀티캐스트 셀의 출력 포트를 vector로 표현하여 그 내적의 값이 최대가 되는 큐에 셀을 할당하는 방법이다. 그림 4는 eigen vector 셀 할당기법의 예시로 각각의 큐가 가지고 있는 eigen vector와 각각의 멀티캐스트 셀의 목적 포트의 vector와의 내적을 계산하여 그 최대값을 갖는 큐에 셀을 할당 한다⁸⁾.

Eigen vector 셀 할당기법의 경우에도 구현이 간단

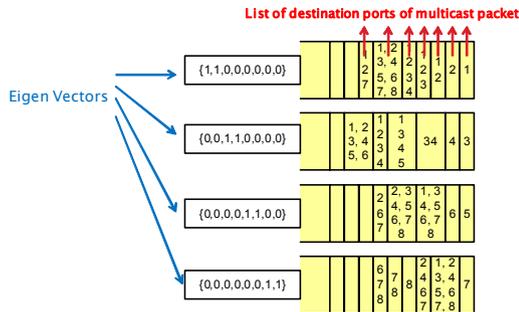


그림 4. Eigen vector 셀 할당기법
Fig. 4. Eigen vector cell assignment

하며 각각의 큐에 셀을 공정하게 할당 할 수 있다는 장점이 있으며 각각의 멀티캐스트 셀 들을 공통의 목적 포트를 가지는 셀들끼리 모아 놓았기 때문에 어느 정도 HoL blocking 확률을 낮출 수 있다. 그러나 eigen vector 방법의 경우 특정 cross point 버퍼가 가득 차 있을 경우 망의 사정 등으로 인해 특정 포트에서 blocking이 계속적으로 발생할 경우 하나의 큐에만 셀이 계속적으로 쌓이게 되어 이로 특정 목적 포트에 대해 처리율이 급격하게 떨어질 수 있다. 또한 eigen vector와의 내적이 최소가 되는 것을 기준으로 셀을 할당하였기 때문에 특정 포트에 연결된 cross point 버퍼가 가득 차 있을 때 해당 포트를 나타내는 eigen vector를 가진 큐를 스케줄링에서 제외한다고 하여도 여전히 다른 큐에 할당된 셀들이 해당 포트를 목적지로 가질 수 있으며 따라서 여전히 HoL blocking 문제가 발생할 수 있는 확률이 있다.

2.3 유니캐스트와 멀티캐스트 셀 스케줄링 방법

유니캐스트 셀과 멀티캐스트 셀을 스케줄링 하는 방법으로는 유니캐스트 셀과 멀티캐스트 셀을 번갈아가면서 순차적으로 처리하는 MURS 방법이 가장 대표적인 방법이다. 그러나 MURS 방법의 경우 스위치에 가해지는 트래픽의 양이 증가함에 따라 멀티캐스트 셀의 HoL blocking 확률이 매우 빠르게 증가하여 멀티캐스트 셀의 처리율이 빠르게 감소하는 단점이 있다⁶⁾.

이와 같은 현상을 수학적으로 분석하기 위해 먼저 유니캐스트 셀의 blocking 확률을 P_{bu} 라고 하자. 이 때 유니캐스트 셀의 blocking은 해당 목적지의 cross point 버퍼가 가득 차 있을 때 발생하며 이 때 각각의 cross point 버퍼의 크기를 L 라고 가정하면 각각의 cross point 버퍼의 blocking 확률은 식 2와 같이 M/M/1/L 큐잉 모델로 정의 할 수 있다⁹⁾.

$$P_{bu} = \frac{1 - \rho_{cb}}{1 - \rho_{cb}^{L+1}} \rho_{cb}^N \quad (2)$$

이 때 ρ_{cb} 는 각 버퍼에 가해지는 traffic load가 되며 입력 포트로 들어오는 유니캐스트 패킷의 traffic load를 ρ 라고 가정하고 목적 포트가 N개의 출력 포트에 균등하게 배분되어 있다고 가정하면 ρ 와 ρ_{cb} 의 관계는 식 3과 같다.

$$\rho_{cb} = \frac{\rho}{N} \quad (3)$$

한편 멀티캐스트 셀의 blocking 확률 P_{bm} 은 유니캐스트 셀의 blocking 확률인 P_{bu} 를 이용해서 구할 수 있다. 하나의 멀티캐스트 셀은 2개~ N 개 사이의 목적지를 가질 수 있으며 어떤 멀티캐스트 셀이 d 개의 목적지를 가진다고 할 때 존재 할 수 있는 경우의 수는 조합을 이용하여 $N C_d$ 과 같이 나타낼 수 있다. 따라서 하나의 멀티캐스트 셀이 d 개의 목적지를 가질 확률은 식 4와 같이 나타낼 수 있다. 이 때 d 개의 목적지를 갖는 멀티캐스트 셀이 blocking이 발생 하지 않을 확률은 목적지로 지정된 d 개의 cross point 버퍼에서 모두 blocking이 발생하지 않을 확률인 $(1-P_{bu})^d$ 와 같게 되므로 이를 종합해 보면 하나의 멀티캐스트 셀이 d 개의 목적지를 가질 경우의 blocking 확률은 식 5와 같이 나타 낼 수 있다.

$$P_{num_dest}(d) = \frac{N C_d}{2^N - 1} \quad (4)$$

$$P_{bm}(d) = \frac{N C_d}{2^N - 1} (1 - (1 - P_{bu})^d) \quad (5)$$

결과적으로 유니캐스트 셀의 blocking 확률을 이용해서 멀티캐스트 셀의 blocking 확률을 나타 낼 수 있으며 또한 유니캐스트 셀의 blocking 확률을 traffic load ρ 를 이용해서 나타낼 수 있기 때문에 traffic load ρ 의 변화에 따른 유니캐스트 셀과 멀티캐스트 셀의 blocking 확률을 비교 해 볼 수 있다.

그림 5는 4개의 입력포트와 4개의 출력 포트를 갖는 4×4 스위치에서 cross point 버퍼의 크기가 3인

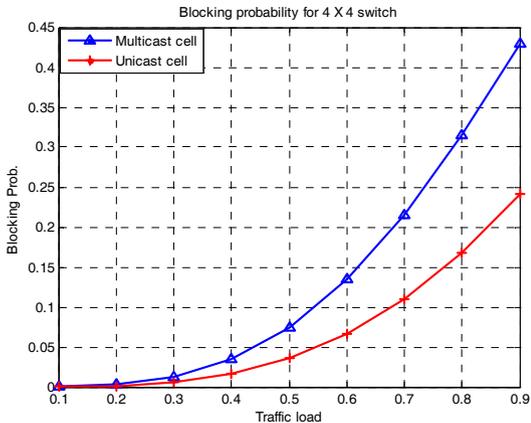


그림 5. 4×4 스위치에서 유니캐스트 셀과 멀티캐스트 셀의 blocking 확률 비교
Fig. 5. Unicast & multicast cell blocking probability comparison for 4×4 switch

경우에 유니캐스트 셀과 멀티캐스트 셀의 HoL blocking 확률을 traffic load의 변화에 따라 나타낸 것이며 그림 6은 8개의 입력포트와 8개의 출력 포트를 갖는 8×8 스위치에서 cross point 버퍼의 크기가 5 인 경우에 유니캐스트 셀과 멀티캐스트 셀의 HoL blocking 확률의 변화를 traffic load의 변화에 따라 나타낸 그림이다. 결과로부터 알 수 있듯이 traffic load가 증가함에 따라 멀티캐스트 셀의 blocking 확률이 매우 빠르게 증가하는 것을 확인 할 수 있으며 스위치의 출력 포트의 수가 많아짐에 따라 그 차이가 더욱 커지는 것을 알 수 있다.

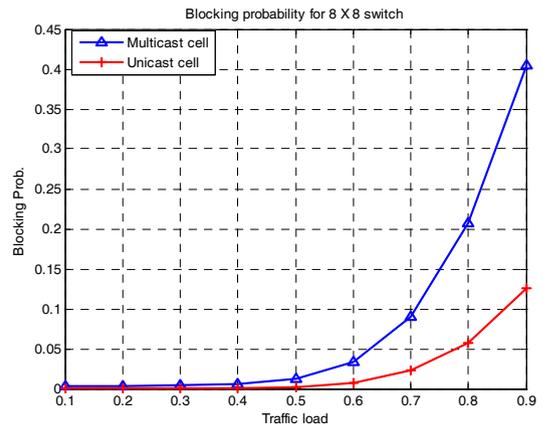


그림 6. 8×8 스위치에서 유니캐스트 셀과 멀티캐스트 셀의 blocking 확률 비교
Fig. 6. Unicast & multicast cell blocking probability comparison for 8×8 switch

III. 제안하는 방법

본 장에서는 멀티캐스트 셀의 HoL blocking 확률을 낮추고 멀티캐스트 셀의 처리율을 높이면서 동시에 유니캐스트 셀과 멀티캐스트 셀의 처리에 있어서 공정성을 보장할 수 있는 기법들을 제안한다. 먼저 입력 포트의 각 큐에 멀티캐스트 셀을 할당하는데 있어 HoL blocking 확률을 낮추기 위한 BA 셀 할당기법과 스위치의 cross point 버퍼에 멀티캐스트 셀을 위한 일정 공간을 확보하여 멀티캐스트 셀의 HoL blocking 확률을 낮추고 멀티캐스트 셀의 처리율을 보장하기 위한 VT 기법을 설명한다.

3.1 BA 셀 할당 기법

본 논문에서 제안하는 BA 셀 할당 기법은 eigen vector를 이용한 방법과 같이 멀티캐스트 셀을 위한 각각의 큐에 고유한 vector를 할당하고 이를 이용하여

각각의 큐에 셀을 할당하는 방법이다. Eigen vector 방법의 경우 이들 k개의 큐의 각각 하나의 vector를 할당 하였으며 vector와의 내적의 값이 최대가 되는 큐에 셀을 할당한다. 반면에 제안한 BA 셀 할당기법의 경우 eigen vector 방법보다 하나 많은 k+1개의 큐를 사용하며 k개의 큐에는 각각 고유한 BA vector를 할당하여 각 큐의 vector와의 내적의 값이 0이 되는 큐에 각 멀티캐스트 셀을 할당하도록 하고 맨 마지막에 위치한 k+1 번째 큐에는 멀티캐스트 셀 중 어느 큐의 BA vector와도 내적의 값이 0이 되지 않는 셀 들을 할당하게 된다. 이 때 입력 포트의 큐들은 VOQ 구조가 사용되어 물리적으로 분리된 큐가 아니라 논리적으로 분리된 큐이기 때문에 하나의 큐를 더 사용 하는 것이 스위치의 구조적인 변화를 필요로 하지는 않는다. 또한 BA 셀 할당 기법의 경우 특정 출력 포트에 향하는 cross point 버퍼가 가득 차 있을 경우 해당 포트에 해당하는 BA vector를 가진 큐를 제외한 다른 모든 큐의 HoL에서 셀의 전송이 가능하다. Eigen vector 방법과 비교하면 eigen vector 방법의 경우에는 특정 포트에 해당하는 cross point 버퍼가 가득 찼을 경우 해당 포트를 나타내는 eigen vector를 가지는 큐를 제외하고 스케줄링 한다고 하여도 여전히 다른 큐의 셀들이 특정 포트를 목적지로 가질 수 있기 때문에 HoL blocking 확률이 여전히 남아있게 되지만 BA 셀 할당 기법의 경우 이러한 HoL blocking을 스케줄링에서 해당 큐를 제외하는 방법을 통해 회피할 수 있다.

그림 7은 BA 셀 할당 기법을 사용한 예를 나타내 것으로 총 출력 포트의 수 N=8, k=4 인 경우에 셀을 할당하는 방법을 나타낸다. BA 셀 할당기의 경우 Modulo 셀 할당 기법이나 eigen vector 방법보다 하나 많은 5개의 큐가 사용되며 이 중 위에서부터 4개의 큐에는 v₁부터 v₄까지 4개의 BA vector가 식 6과 같이 할당된다.

$$\begin{aligned}
 v_1 &= \{1, 1, 0, 0, 0, 0, 0, 0\} \\
 v_2 &= \{0, 0, 1, 1, 0, 0, 0, 0\} \\
 v_3 &= \{0, 0, 0, 0, 1, 1, 0, 0\} \\
 v_4 &= \{0, 0, 0, 0, 0, 0, 1, 1\}
 \end{aligned}
 \tag{6}$$

이 때 각각의 멀티캐스트 셀 들은 그림 6의 셀 할당 결과와 같이 각 큐의 BA vector와 내적의 값이 0이 되는 곳에 할당된다. 이 때 하나 이상의 vector와의 내적이 0이 되는 멀티캐스트 셀은 내적의 값이 0이 되는 큐 중에서 무작위로 하나의 큐가 선택이 되며 어느 BA vector와도 내적이 0이 되지 않는 멀티캐스트 셀은 마지막 5번째 큐에 저장된다.

3.2 VT 방법

본 논문에서 제안하는 VT 기법은 HoL blocking 문제로 인한 멀티캐스트 셀의 처리율의 급속한 저하를 막고 멀티캐스트 셀에게 일정 수준 이상의 처리율을 보장 하면서 동시에 유니캐스트 셀과의 공정한 처리를 보장할 수 있는 기법이다. 그림 8과 같이 CICQ 스위치의 switching fabric에 위치한 cross point 버퍼에 가상의 임계값 V_{th}를 설정하여 버퍼에 대기중인 패킷이 V_{th} 이상으로 늘어나게 되면 전체 버퍼 공간중 남은 공간에 해당 하는 L-V_{th} 만큼의 공간을 멀티캐스트 셀만이 사용할 수 있도록 한 방법이다. 즉 cross point의 버퍼의 공간 중 일부 공간을 멀티캐스트 셀만이 사용할 수 있는 공간으로 설정함으로써 멀티캐스트 셀의 HoL blocking 확률을 낮추어 처리율을 높일 수 있는 방법이다.

VT 기법에서 가상의 임계값 V_{th}의 사용에 있어서 유니캐스트 셀과 멀티캐스트 셀의 양의 변화에 따라 동적으로 변화하는 V_{th} 값을 사용할 수 있다. 우선 초기의 V_{th}값을 버퍼의 최대 크기 L보다 1 작은 L-1 로 설정하며 멀티캐스트 셀의 HoL blocking이 발생 할 때 마다 V_{th}값을 1씩 줄여 나간다. 또한 HoL blocking이 발생 하지 않고 멀티캐스트 셀이 성공적으로 처리

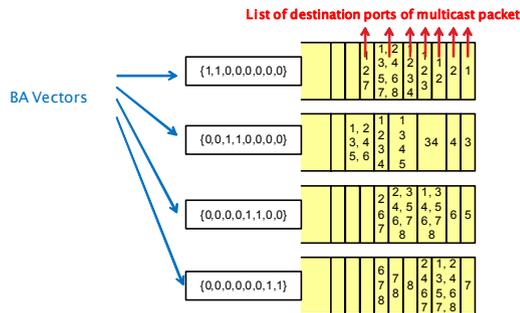


그림 7. BA 셀 할당 기법
Fig. 7. BA cell assignment scheme

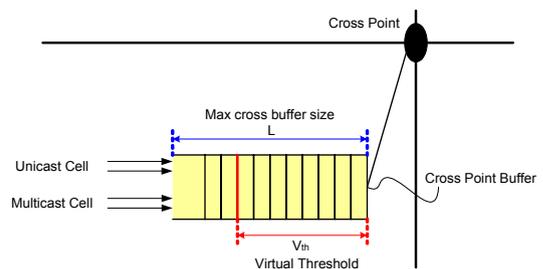


그림 8. VT 기법
Fig. 8. VT scheme

될 때 마다 V_{th} 값은 다시 L-1이 될 때 까지 1씩 증가하게 되며 이에 따라 V_{th} 가 가질 수 있는 최대값은 L-1, 최소값은 1이 된다. 이 때 V_{th} 값이 작아질수록 멀티캐스트 셀을 위한 전용 공간이 늘어나는 것을 의미하며 V_{th} 값이 커질수록 멀티캐스트 셀의 전용 공간이 줄어드는 것을 의미한다. 이와 같은 V_{th} 값의 조정을 통해 멀티캐스트 셀의 blocking이 많이 발생하면 멀티캐스트 셀의 전용 공간을 늘려서 멀티캐스트 셀의 처리율을 보장 할 수 있으며 blocking이 발생하지 않으면 전용 공간을 줄여서 유니캐스트 셀의 처리율을 저하 시키는 것을 막아 유니캐스트 셀과의 처리에 있어서 공성을 확보 할 수 있다.

한편 VT 기법을 사용하는 스위치의 동작을 간단히 하기 위해 동적으로 변하는 값이 아닌 고정된 V_{th} 값을 사용하는 스위치를 구현 할 수 있다. 이 때 고정된 V_{th} 값을 사용하기 위해서는 수학적 분석을 통한 최적화된 V_{th} 값을 찾아내는 과정이 필요하며 다음 장에서 그 과정을 설명한다.

IV. 성능 분석 및 평가

4.1 고정된 V_{th} 값을 사용을 위한 최적화

VT 기법을 사용하는 스위치의 동작을 간단히 하기 위해 고정된 V_{th} 값을 사용 할 수 있으며 고정된 V_{th} 값을 사용하기 멀티캐스트 셀의 처리율을 높이면서 동시에 유니캐스트 셀과의 공정성을 보장하기 위한 최적화 과정이 필요하다. 최적화를 위해 입력 포트에 위치한 큐로부터 N개의 cross point 버퍼로 전달되는 유니캐스트 셀의 전체 arrival rate를 λ_u , 멀티캐스트 셀의 전체적인 arrival rate를 λ_m , cross point 버퍼의 크기는 L이라고 하자. 먼저 유니캐스트 셀의 경우 입력 포트의 멀티캐스트 셀이 N개의 큐에 균등하게 나누어진다고 가정하면 평균 arrival rate는 λ_u/N 가 된다. 다음으로 멀티캐스트 셀의 경우는 1개~N개 사이의 목적 포트를 가질 수 있기 때문에 평균적인 목적지의 수를 m이라고 하면 m을 식 7과 같이 구할 수 있으며 목적지의 수에 비례하여 cross point 버퍼에 도착하는 arrival rate가 증가하게 된다. 따라서 하나의 cross point 버퍼에 대한 유니캐스트 셀과 멀티캐스트 셀의 전체적인 arrival rate는 식 8과 같이 나타낼 수 있다. 또한 이 때 k는 하나의 입력 포트에서 가지고 있는 멀티캐스트 큐의 수를 나타낸다. 또한 평균적인 service rate μ 는 식 9와 같이 정의 한다.

$$ave(m) = \sum_{i=1}^k \frac{i}{N} \tag{7}$$

$$\lambda = \frac{\lambda_u}{N} + \frac{\lambda_m}{N} \sum_i \frac{i}{N} = \frac{\lambda_u}{N} + \frac{(N+1)}{2N} \lambda_m \tag{8}$$

$$\mu = \frac{(\lambda_u + m\lambda_m)}{N} \tag{9}$$

식 8, 9의 평균적인 arrival rate와 service rate를 이용하면 VT 기법을 적용한 cross point 버퍼의 상태를 그림 9와 같은 M/M/1/L 큐잉 시스템의 state transition 다이어그램으로 나타낼 수 있다. 그림에서 L은 cross point 버퍼의 전체 크기가 되며 cross point 버퍼에 패킷이 가상의 임계값인 V_{th} 값 이상으로 쌓이게 되면 유니캐스트 패킷은 더 이상 들어올 수 없기 때문에 이때의 arrival rate λ' 는 식 10과 같이 정의 된다.

$$\begin{aligned} \lambda' &= \frac{\lambda_m}{N} \sum_{i=1}^k \frac{i}{N} \\ &= \frac{(N+1)}{2N} \lambda_m \end{aligned} \tag{10}$$

다음으로 그림 9의 state transition 다이어그램에 global balance equation을 적용 하면 식 11과 같이 각 state의 확률 P_i 를 구할 수 있으며 모든 state의 확률을 합하면 1 이라는 것을 이용 하면 식 12와 같이 초기 확률 P_0 를 구할 수 있다. 계속해서 이 결과를 이용하면 유니캐스트 셀의 blocking 확률을 식 13과 같이 구할 수 있다.

$$\begin{aligned} P_i &= \left(\frac{\lambda}{\mu}\right)^i P_0 = \rho^i P_0, \quad (i \leq V_{th}) \\ P_i &= \left(\frac{\lambda}{\mu}\right)^{V_{th}} \left(\frac{\lambda}{\mu}\right)^{(i-V_{th})} P_0 \\ &= \rho^{V_{th}} \rho^{(i-V_{th})} P_0, \quad (i > V_{th}) \end{aligned} \tag{11}$$

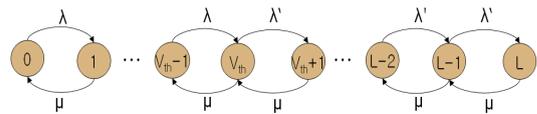


그림 9. Crosspoint 큐의 state transition 다이어그램
Fig. 9. State transition diagram of crosspoint queue

$$P_0 = \frac{(1-\rho)(1-\rho')}{\rho(1-\rho^{V_{th}})(1-\rho') + (\rho^{V_{th}}(1-\rho)\rho'(1-\rho'^{(L-V_{th})}))} \quad (12)$$

$$P_{Bu} = P_{V_{th}} + P_{V_{th}+1} + \dots + P_{L-1} + P_L$$

$$= \sum_{i=V_{th}}^L P_i \quad (13)$$

다음으로 멀티캐스트 경우에는 모든 목적 포트의 cross point 버퍼가 사용 가능한 상태일 때에만 blocking이 아니므로 멀티캐스트 셀에 대한 blocking 확률 P_{Bm} 은 식 14와 같이 나타낼 수 있다.

$$P_{Bm} = 1 - \frac{1}{N} \sum_{n=1}^N (1 - P_L)^n \quad (14)$$

이 때 제 2장에서 설명한 것과 같이 멀티캐스트 셀의 blocking 확률이 유니캐스트 셀에 비해 훨씬 크기 때문에 이들 둘 사이에서 공정성을 보장하기 위해서는 멀티캐스트 셀의 HoL blocking 확률을 줄이는 것이 필수적이다. 그러나 VT 기법을 적용하는데 있어서 유니캐스트 셀과 멀티캐스트 셀의 blocking 확률은 서로 간에 trade-off 관계가 성립한다. 즉 V_{th} 값이 작아지면 멀티캐스트 셀의 blocking 확률이 낮아지고 유니캐스트 셀의 blocking 확률이 높아지게 되며 V_{th} 값이 커지면 멀티캐스트 셀의 blocking 확률이 높아지고 유니캐스트 셀의 blocking 확률이 낮아지게 된다. 따라서 V_{th} 값을 이용하여 멀티캐스트 셀의 blocking 확률을 낮추더라도 유니캐스트 셀의 blocking 확률이 멀티캐스트 셀의 blocking 확률보다 커지지는 않도록 해야 한다. 따라서 이와 같은 최적의 V_{th} 값을 구하는 문제는 식 15와 같은 최적화 문제로 정의 할 수 있다.

$$\min V_{th}$$

$$\text{subject to} \quad (15)$$

$$0 \leq V_{th} \leq L, \quad P_{Bu} \leq P_{Bm}$$

식 9~14에 적당한 값을 대입하면 식 15의 최적화 문제를 풀 수 있지만, 각각의 값들은 정수 값을 가지기 때문에 closed-form으로 된 해답을 찾기는 어렵다. 따라서 본 논문에서는 8개의 입력과 8개의 출력 그리고 4개의 멀티캐스트 VOQ를 갖는 8×8 스위치에 대해 위의 식을 적용하여 numerical result를 구하였으며 그 결과 그림 10과 같은 결과를 얻을 수 있었다. 그림 10의 결과는 cross point 버퍼의 크기 L 이 4인 경우에

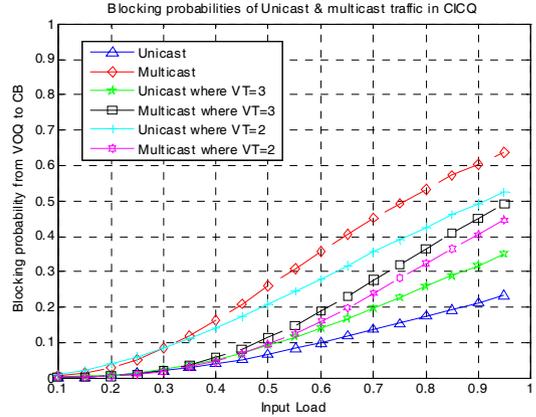


그림 10. Input load의 변화에 따른 blocking 확률의 변화
Fig. 10. Blocking probability corresponds to input load

V_{th} 값을 변화시키면서 얻은 결과로 V_{th} 값이 3인 경우에는 멀티캐스트 셀의 blocking 확률이 유니캐스트 셀의 blocking 확률보다 높지만 V_{th} 값이 2로 감소할 경우에는 멀티캐스트 셀의 blocking 확률보다 유니캐스트 셀의 blocking 확률이 더 커지는 것을 확인할 수 있다. 따라서 V_{th} 값이 2인 경우는 멀티캐스트 트래픽의 처리율을 높이기 위해 유니캐스트 셀의 처리율을 과도하게 희생한 경우라고 볼 수 있으며 이 경우 유니캐스트 셀과 멀티캐스트 셀의 공정성이 위배되었다고 할 수 있기 때문에 공정한 처리를 위해서는 V_{th} 값으로 3을 사용 하는 것이 바람직하다.

4.2 BA 셀 할당기법 시뮬레이션 결과

BA 셀 할당 기법의 시뮬레이션을 8개의 입력 포트와 8개의 출력 포트를 갖는 8×8 CICQ 스위치가 사용되었으며 멀티캐스트 셀은 8개의 출력 포트를 각각 동인한 확률로 목적지로 가질 수 있도록 하였다. 입력 포트의 큐의 크기는 무한하다고 가정 하였으며 Module 셀 할당 기법과 Eigen vector 셀 할당 기법의 경우 4개의 멀티캐스트 큐를, BA 셀 할당기법에서는 5개의 멀티캐스트 큐를 사용 하도록 하였다. 입력 traffic model은 Bernoulli traffic model이 사용 되었으며 전체 시뮬레이션 시간은 20,000 time-slot 동안 반복해서 측정 하였으며 결과 그래프에서의 delay는 하나의 셀이 cross point buffer 까지 입력되는데 걸린 time-slot의 수, throughput은 총 시뮬레이션 시간동안 처리된 셀의 수가 된다.

그림 11은 각각의 셀 할당 기법들을 적용 했을 때의 delay를 비교한 것으로 modulo 기법과 eigen vector 방법의 경우 비슷한 결과를 보이지만 BA 셀

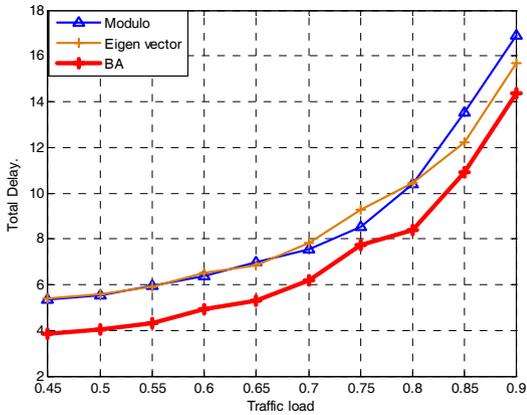


그림 11. BA 셀 할당기법 delay 비교
Fig. 11. Delay comparison for BA cell assignment

할당 기법의 경우 적어도 10%이상의 delay의 감소를 확인할 수 있다. 또한 eigen vector 방법이 0.8 이상의 매우 높은 traffic load에서 단순한 modulo 방법보다 더 좋은 성능을 보인다는 것을 확인할 수 있었다. 그림 12는 각각의 셀 할당기법들을 적용 했을 때의 throughput을 비교한 결과로 전체적으로 3가지 방법이 throughput에 있어서는 큰 차이가 나지 않으며 제안한 BA 셀 할당기법을 사용 했을 때 약간의 성능 향상이 있다는 것을 확인할 수 있다.

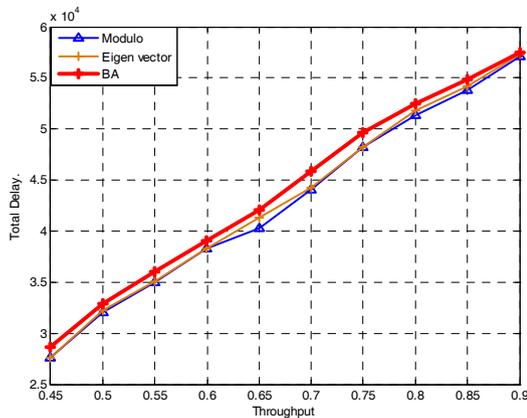


그림 12. BA 셀 할당 기법 throughput 비교
Fig. 12. Throughput comparison for BA cell assignment

4.3 VT 방법 시뮬레이션 결과

VT 방법의 시뮬레이션을 위해 8개의 입력 포트와 8개의 출력 포트를 갖는 위해 8x8 CICQ 스위치가 사용되었다. 입력포트의 VOQ에서 crosspoint 버퍼 사이의 스케줄러는 MURS_mix^[6] 방법이 사용 되었으며

cross point 버퍼의 크기는 4, V_{th} 값은 동적 할당을 사용 하였으며, cross point 버퍼에서 출력 포트에 셀을 스케줄링 하는 방법으로는 round-robin 방법이 사용 되었다. Traffic model은 Bernoulli traffic model이 사용 되었으며 유니캐스트 셀과 멀티캐스트 셀의 비율은 2:1로 가정하여 시뮬레이션을 진행 하였다. 또한 시뮬레이션은 20,000 time-slot 동안 반복해서 실행 하였다.

그림 13은 VT 방법과 일반적인 CICQ 구조를 사용 했을 때 delay를 비교하 것이다. 일반적인 CICQ 방법의 경우에는 유니캐스트 셀의 delay가 작은 반면에 멀티캐스트 셀의 delay가 매우 크다. 그러나 VT 방법을 사용 했을 경우에는 유니캐스트 셀 delay에 약간의 증가가 있지만 멀티캐스트 셀에 대해서 큰 폭의 delay 감소 효과가 있기 때문에 평균적인 delay에서는 VT 방법을 사용 하면 더 좋은 성능을 얻을 수 있으며 또한 traffic load가 높을수록 더 높은 성능의 향상을 확인할 수 있다. 그림 14는 VT 방법과 일반적인 CICQ 방법을 사용한 경우에 achieved bandwidth의 비율을 비교한 결과로 VT 방법을 사용할 경우 유니캐스트 트 셀의 achieved bandwidth에는 약간의 감소가 생기지만 멀티캐스트 셀의 achieved bandwidth가 크게 향상

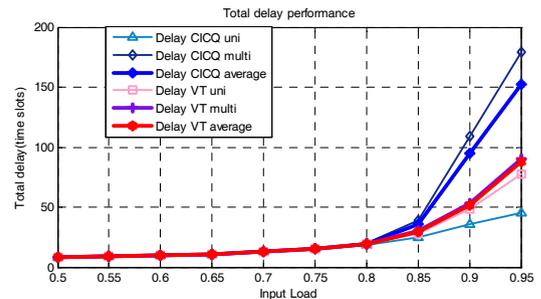


그림 13. VT 방법을 적용 했을 때 delay 비교
Fig. 13. Delay comparison for VT method

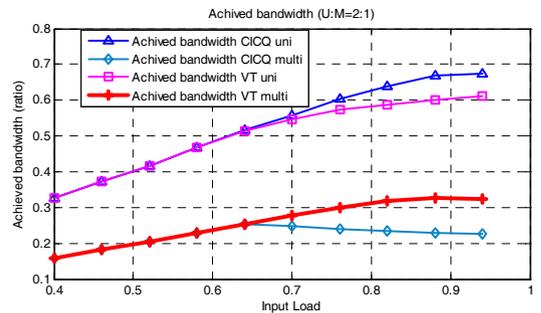


그림 14. VT 방법을 적용했을 때 achieved bandwidth 비교
Fig. 14. Achieved bandwidth comparison for VT method

되어 결과적으로 유니캐스트와 멀티캐스트 셀의 처리에 있어서의 공정성을 크게 향상 시킨 것을 확인 할 수 있다.

그림 15는 VT 기법의 시뮬레이션을 위해 사용한 것과 동일한 조건의 스위치에 BA 셀 할당기법과 VT 기법을 각각 적용 했을 경우와 두 가지 방법을 동시에 적용 했을 경우에 achieved bandwidth를 비교한 결과이다. BA 셀 할당 기법과 VT 기법을 각각 따로 적용 했을 때에도 일반적인 CICQ 구조보다 좋은 성능을 얻는 다는 것을 확인 할 수 있으며 두 가지 기법을 모두 사용 했을 때 가장 큰 achieved bandwidth를 얻는 다는 것을 확인 할 수 있다.

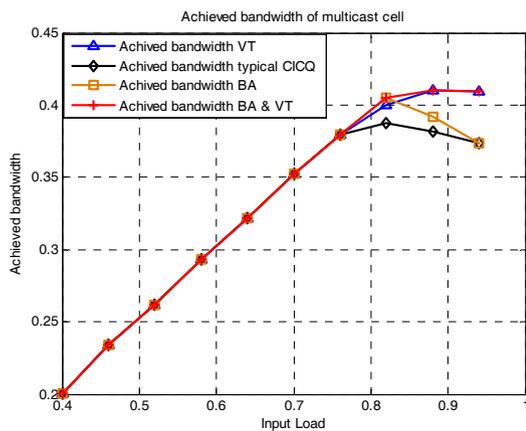


그림 15. BA 셀 할당 기법과 VT 방법의 적용에 따른 achieved bandwidth 비교
 Fig. 15. Achieved bandwidth comparison for integrated applying of BA scheme and VT method

V. 결론

본 논문에서는 HoL blocking 문제로 인해 스위치에서 멀티캐스트 트래픽의 처리율이 급격히 떨어지는 문제를 해결하기 위해 BA 셀 할당 기법과 VT 기법을 제안 하였다. BA 셀 할당 기법은 스위치의 입력 포트 로 들어오는 멀티캐스트 트래픽을 제한된 숫자의 큐 에 효율적으로 할당하는 방법으로 HoL blocking 문제를 완화 시킬 수 있다. 또한 VT 기법은 cross point 버퍼에서 멀티캐스트 셀을 처리하기 위한 전용 공간을 확보하는 방법으로 멀티캐스트 트래픽의 처리율을 크게 향상 시키면서 동시에 유니캐스트 트래픽과의 공정성을 보장 할 수 있다. 시뮬레이션을 통해 제안된 기법의 우수성을 검증 하였으며 뚜렷한 delay의 감소와 throughput의 향상을 확인하였다. 따라서 본 논문

에서 제안된 BA 셀 할당 기법과 VT 기법을 이용하여 멀티캐스트 트래픽의 처리율 향상 시키면서 동시에 유니캐스트 트래픽과의 공정성을 보장하여 멀티캐스트 서비스에서의 QoS를 향상 시킬 수 있다는 결론을 얻을 수 있었다.

참고 문헌

- [1] A.M. Karol, M. Hluchyj and S. Morgan, "Input Versus Output Queuing on a Space-Division Packet Switch," IEEE Transactions on Communications Vol.35, No.09, Dec 1987, pp.1337-1356.
- [2] N. McKeown, "Scheduling algorithms for input-queued cell switches," Ph.D. thesis, University of California at Berkeley, May 1995.
- [3] M. Ajmone Marsan, A. Bianco, P. Giaccone, E. Leonardi and F. Neri, "Multicast traffic in input-queued switches : optimal scheduling and maximum throughput," IEEE/ACM Transactions on Networking, Vol.11, No.3, June 2003, pp. 456-477
- [4] L. Mhamdi and M. Hamdi, "Scheduling Multicast Traffic in Internally Buffered Crossbar Switches," IEEE ICC, June 2004, pp.1103-1107
- [5] Fong-Cheng Lee, Wen-Fong Wang, Ming-Da Liu "An Efficient Buffered Crossbar Switching Architecture with Multiple Multicast Virtual Output Queues for Mixed Uniand Multicast Traffic," IEICE APCC, October 2008.
- [6] Hongchao Hu, Lin Pang, Peng Yi, and Yunfei Guo, "Integrated Uni- and Multicast Traffic Scheduling in Buffered Crossbar Switches," CHINACOM'08, August 2008, pp.66-72
- [7] L. Mhamdi, "On the Integration of Unicast and Multicast Cell Scheduling in Buffered Crossbar Switches," IEEE Transactions on parallel and distributed systems. Vol.20, No.6, JUNE 2009, pp. 818-830
- [8] A. Bianco, P. Giaccone, E. Leonardi, F. Neri, and C. Piglione, "On the Number of Input Queues to Efficiently Support Multicast Traffic in Input Queued Switches," IEEE Workshop on High Performance Switching and Routing

(HPSR '03), June 2003, pp.111-116

- [9] L. Kleinrock, "QUEUEING SYSTEM Vol.1 Theory," Wiley, New York, 1975

김 경 민 (Kyungmin Kim)

정회원



2006년 2월 연세대학교 전기전자공학부
 2008년 2월 연세대학교 전기전자공학과 석사
 2008년 3월~현재 연세대학교 전기전자공학과 박사과정

<관심분야> 차세대 통신망, 핸드오버, 휴대방송, 멀티캐스트, QoS, 스케줄링

이 재 용 (Jaiyong Lee)

중신회원



1977년 2월 연세대학교 전자공학과
 1984년 5월 IOWA State University 공학석사
 1987년 5월 IOWA State University 공학박사
 1987년 6월~1994년 8월 포항공과대학 교수

1994년 5월~현재 연세대학교 전자공학과 교수
 <관심분야> Protocol Design for Wired/Wireless QoS Management, Ubiquitous Sensor Network, Wireless Multihop Network

정 현 덕 (Hyunduk Jung)

정회원



2004년 2월 연세대학교 전기전자공학부
 2006년 2월 연세대학교 전기전자공학과 석사
 2006년 3월~현재 연세대학교 전기전자공학과 박사과정

<관심분야> 차세대 통신망, 핸드오버, 휴대방송, 유무선 통합망, Femtocell