

MIN기반 교환기 구조를 분석하기 위한 간단한 근사화 방법 연구

최 원 제[†] · 추 현 승^{††} · 문 영 성^{†††}

요 약

다단계 상호연결 네트워크(multistage interconnection network MIN)는 병렬 컴퓨터 시스템에서의 효율적인 상호연결 구조로서 인식되어 왔으며, 최근에는 차세대 통신 기술인 기가비트 이더넷 스위치와 테라비트 라우터 그리고 ATM(Asynchronous Transfer Mode)에서의 전형적인 스위칭 구조로 인식되고 있다. MIN의 성능을 분석하고자 하는 많은 모델 연구들이 논문에서 보고되었으나 이들 기존의 모델들에는 대부분 정확하지 않거나, 정확하더라도 매우 복잡한 모델링 기법이 사용되었다. 본 논문에서는 Small Clock Cycle(SCC) 환경에서 스위칭 요소 내에 다중 입력 버퍼를 가진 MIN의 성능을 정확히 계산하는 모델을 근사화한 방법에 기인하여 제안한다. 새로이 제안되는 모델은 기존의 모델들보다 내단화 간단하여 계산 오버헤드를 줄일 수 있다. 종합적인 컴퓨터 시뮬레이션은 제안된 모델이 성능 척도인 처리율과 지연의 측면에서 매우 정확한 결과를 나타냄을 보여준다.

A Simple Approximation Method for Analyzing MIN Based Switching Architecture

Wonjae Choi[†] · Hyunseung Choo^{††} · Youngsong Mun^{†††}

ABSTRACT

Multistage interconnection networks (MINs) have been recognized as an efficient interconnection network for high-performance computer systems and also have been recently identified to be effective for a switching fabric of new communication structures - gigabit ethernet switch, terabit router, and ATM (asynchronous transfer mode). While lots of models analyzing the performance of MINs have been proposed, they are either inaccurate or, even if accurate, very complex for the analysis. In this paper, we propose an extremely simple model for evaluating the multibuffered MIN with small clock cycles based on the approximation approach. Comprehensive computer simulation shows that the proposed model is very accurate in terms of the throughput and mean delay. Furthermore, it significantly reduces the computing overhead due to its simplicity.

1. 서 론

여러 스테이지(stage)에서 간단한 스위칭 요소(switching element : SE)들을 연결함으로써 구성되는 패킷 스위칭 다단계 상호연결 네트워크(multistage intercon-

nection network : MIN)는 병렬 컴퓨터 시스템[1]과 빠른 속도의 통신 시스템[2,3]을 위한 효율적인 상호연결 구조로서 인식되어 왔다. 이들의 정규성과 모듈성은 매우 큰 규모의 통합이 적절히 배열되도록 만들어졌으며, 또한 입력으로부터 출력으로의 빠른 통신에 중요한 "자신의 경로매정(self-routing)" 속성을 지닌다. 최근에 많은 관심을 끄는 기가비트 이더넷 스위치[4]와 테라비트 라우터에서의 가능한 교환구조로, 또 새로운

[†] 준 회원 : 성균관대학교 대학원 전기전자 및 컴퓨터공학부
^{††} 중신회원 : 성균관대학교 전기전자 및 컴퓨터공학부 교수
^{†††} 중신회원 : 숭실대학교 컴퓨터학부 교수
논문접수 : 1999년 8월 11일, 심사완료 : 1999년 10월 11일

통신 방법인 ATM의 스위치로 MIN은 중요하게 인식 되어 왔고, 현재 널리 사용되는 크로스바(cross-bar) 구조와 함께 차세대 교환구조로 기대된다.

ATM[5]은 셀(cell)이라고 불리는 고정된 길이의 짧은 패킷을 가지고 있는 통신 시스템에서의 고속 패킷 스위칭 기술이며, 간략화된 기능성들로 인해 전형적인 패킷 스위칭보다 빠른 속도의 스위칭을 허용한다. ATM을 위한 다수의 스위칭 시스템들이 제안되어오는 동안에도, MIN은 오늘날 종합 정보 통신망(integrated service digital network : ISDN)에 매우 효과적임이 입증되었으며, 또한 화상 통신과 그래픽 관련 응용들을 위한 광대역 서비스들을 지원하는 미래 네트워크의 ATM 스위칭을 위한 뛰어난 후보들 중의 하나임에 틀림없다.

MIN의 동작연구에 있어서 가장 중요한 개념중의 하나가 네트워크 사이클의 운영 방법이다. 여기에는 big clock cycle(BCC)[6]과 small clock cycle(SCC)[7, 8], unit step buffering(USB)[9]등을 통하여 네트워크 사이클을 효과적으로 운영함으로써 성능을 향상시키는 다양한 방법들이 논문에 제안되었다. 이전 연구에서 SCC가 BCC보다 성능면에서 우수함[8]이 입증되었고, 여기에 USB[9] 개념의 도입으로 더 한층 향상된 성능을 얻을 수 있었다. 이런 개념들의 성능비교를 위한 이전에 설계되었던 다른 모델들에서도 비교적 정확한 결과를 갖는 모델링 방법이 사용되었지만, 이전 모델들은 그 방법이 매우 복잡하며 여러 가지 다른 형태의 MIN 구조나 작동법에 적용되기 힘든 단점이 있었다. 본 논문에서는 이전 모델보다 훨씬 간단한 모델링 기법을 사용하면서도 정확한 결과를 나타내는 새로운 MIN 모델을 제안한다. 이 모델은 SCC 환경에서 각 스위칭 요소가 다중 입력 버퍼 모듈을 가진 MIN의 모델링이다. 제안된 모델의 정확성 검증은 위하여 종합적인 컴퓨터 시뮬레이션이 수행되고, 처리율과 셀 지연을 성능평가의 척도로 사용하여 모델의 타당성을 입증한다. 또한 이전의 MIN 모델링 중에서 가장 정확한 모델로 알려진 [8]과 그 결과를 비교함으로써 새로운 모델의 간결성 및 우수성을 성능평가 측면에서 논의한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 이전 모델들에 대한 소개와 함께 MIN의 모델링 기법에 있어서 고려해야 하는 중요한 논점들에 대해서 서술한다. 3장에서는 제안된 새로운 모델이 설명되고, 이 모델의 결과에 대한 성능평가가 4장에서 이루어진다. 마지막으로 5장에서 본 논문을 결론 짓는다.

2. MIN의 모델링

MIN의 SE내부에 위치하고 있는 버퍼는 MIN의 성능을 현저히 향상시키며, 버퍼의 위치에 따라서 입력 버퍼의 MIN과 출력 버퍼의 MIN으로 나눌 수 있다. 논문 [11]에서 제시된 바와 같이 출력 버퍼의 MIN은 스위칭 요소 내부의 보다 빠른 스위칭이 전제된 경우에 입력 버퍼의 MIN 보다 나은 성능을 나타낸다. Ding과 Bhuyan[7]은 패킷 이동시에 각 인접한 다음 스테이지의 정보만을 이용한다면 MIN의 성능이 급격히 향상될 수 있다는 사실을 보였다(SCC). 이에 반해서 이전의 기존 방법에서는 패킷 이동은 마지막(맨 오른쪽) 스테이지로부터 처음(맨 왼쪽) 스테이지로 필요한 요소들을 재귀적으로(recursively) 계산된 전체 스테이지의 버퍼 가용성 정보를 이용하여 이동한다(BCC)

실제 MIN 내의 버퍼에서 패킷이 블록 되는 경우는 다음 스테이지에 있는 패킷의 목적지 버퍼에 가용한 공간이 없거나, 블러킹 스위치(blocking switch)의 특성에 의해 경쟁관계의 버퍼에 있는 패킷과 출력에 대한 경쟁에서 지는 경우, 이 두 가지의 경우이다. 일단 블록된 패킷은 원래 목적지를 다음 네트워크 사이클에서도 그대로 향하려고 하는 기억 특성(memory characteristics)을 가지는데, 이러한 기억 특성이 있기 때문에 MIN의 모델링을 정확히 하는데 어려움이 있다. 블록된 패킷의 목적지 버퍼 모듈이 차있는 경우, 그 블록된 패킷은 SCC의 특성상 다음 네트워크 사이클에서 반드시 블록된다. 이러한 현상은 트래픽 부하가 커질수록 자주 관찰됨을 알 수 있으며, 본 논문에서는 이러한 사실에 착안하여 패킷이 지속적으로 블록되는 확률을 계산하여 모델링에 고려함으로써 정확한 성능을 예측하는 근사화 방법을 제안한다. 이러한 기억 특성을 분석 모델에 적용하기 위해 다양한 방법이 사용되어 왔는데 다음에 소개하는 기존의 두 가지 대표적인 모델에서는 이런 기억 특성을 나타내기 위해서 각각 다른 방법이 사용되었다.

- Lin과 Kleinrock의 모델 [10]: Lin과 Kleinrock은 SCC환경에서 출력 버퍼 MIN의 모델에 대해 연구하였다. 그들의 모델은 버퍼 모듈에 들어오려고 하는 패킷이 존재하는 확률과, 버퍼 모듈에서 나가려고 하는 패킷이 블록되는 확률의 두 확률 값을 주로 이용한 모델이다. 그들은 기억 특성으로 나타나

는 높은 블럭킹 상태를 효과적으로 나타내기 위해서 버퍼 모듈 서버의 상태를 new와 blocked로 나누고, 서버가 각 상태에 존재하는 안정 상태 확률(steady state probability)를 계산하였다 이 확률을 이용하여 버퍼 모듈에 패킷이 존재하는 확률을 근사화 시켜서 줄였으며, 그들은 이 방법을 근사화 방법(approximation method) 이라고 불렀다 그들의 모델은 간단하면서도 정확한 결과를 나타내었으며 EFOS(Even-First Odd-Second)와 같은 비균등 입력 트래픽 패턴(nonuniform input traffic pattern)에서도 마찬가지로 정확한 결과를 나타냈다. 그러나 그들의 모델은 출력 버퍼 MIN의 모델이며, 입력 버퍼 MIN에서는 안정 상태 확률을 구하기가 매우 어려우므로 이 모델을 입력 버퍼 MIN으로 적용시키기에는 매우 어리우며 복잡하다.

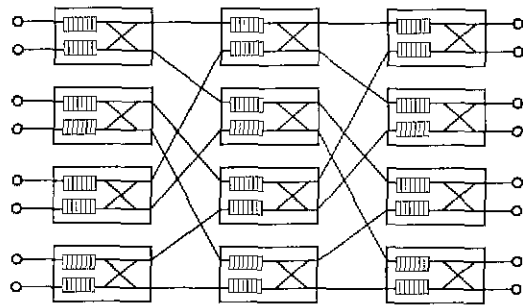
- *Youn과 Mun의 모델* [8] : Youn과 Mun은 SCC 상황에서 입력 버퍼의 MIN을 두 연결된 네트워크 사이클 간의 패킷 이동의 상호관계를 고려해서 모델링 하였다. 이들이 사용한 방법은 위의 Lin과 Kleinrock의 모델이 기억 특성을 나타내기 위해서 버퍼에 있는 서버의 상태를 new/blocked의 안정화 상태(steady-state)로 나눈 것과는 달리, 버퍼 모듈에 있는 패킷의 수와 선두 패킷(leading packet)의 상태(normal/blocked)에 따라서 버퍼 모듈의 상태를 나누었다. 이 모델에서 사용된 버퍼 모듈 상태의 수는 Lin과 Kleinrock의 모델보다 두 배 많으며, 이로 인해서 모델링은 매우 복잡해졌다. 이 모델은 매우 정확한 분석 결과를 보인다. 그러나 이들의 모델링 방법은 비균등 입력 트래픽 패턴에 적용하기가 매우 힘들다는 단점이 있다.

마지막으로, 제안된 모델에서는 비균등 입력 트래픽에서 성능 평가를 위해 비균등 트래픽의 한 종류인 EFOS(Even-First Odd-Second) 트래픽 패턴이 사용된다 여기서 EFOS 트래픽 패턴이란 MIN의 입력 포트 중 짝수번째 포트에 들어온 패킷은 MIN의 출력 포트 중 위쪽 반의 포트들로 균등하게 나가며, 반대로 홀수 번째 포트에 들어온 패킷은 아래쪽 반의 출력 포트들로 균등하게 나가는 트래픽 패턴이다. 제안된 모델과는 다른 경우인 버퍼 모듈 내에 출력 버퍼를 가진 MIN에서의 EFOS 트래픽에 대한 성능 평가는 [10]에서 이루어졌으며, EFOS 트래픽이 가진 패킷의 비균등성으로 인해 심한 패킷 경쟁이 발생하여 MIN의 성능

이 급격히 떨어짐이 보여졌다.

3. 입력 버퍼의 MIN을 위한 제안된 모델

이 절에서는 SCC상황에서 유한개의 입력 버퍼를 가진 기존의 모델보다 간단한 모델링 방법을 사용하면서도 정확한 성능을 나타내는 MIN의 성능 분석을 위한 모델을 제시한다. 먼저 제시된 모델에서 사용된 가정과 정의를 살펴보면 다음과 같으며 (그림 1)은 세 스테이지 입력 버퍼 MIN를 나타낸 그림이다.



(그림 1) 입력 버퍼를 가진 세 스테이지 MIN

3.1 정의와 가정

균등 트래픽(uniform traffic)의 가정이 이전 모델들 [6-11]에서와 같이 사용되며, 더욱 특별한 가정은 다음과 같다. 우선 각 프로세서는 모든 메모리 모듈에 대해 균등하게 분산된 확률을 사용해서 패킷을 생성하며, 2x2 스위칭 요소 내에 있는 두 버퍼는 통계적으로 독립적이다. 또한 각 패킷은 경쟁에서 이길 확률이 동등하며, 블럭된 패킷은 원래의 목적지로 재전송된다. 이러한 가정으로부터, 스위칭 요소 내의 버퍼 상태를 고려해서 전체 스테이지의 상태(state)를 나타내었다 즉 스위칭 요소 내의 버퍼에 의하여 그 요소 내의 상태가 나타나며, 이는 결과적으로 그 스테이지의 상태가 된다. 이제 다음의 변수들이 제안된 분석 모델을 설명하기 위해서 정의된다.

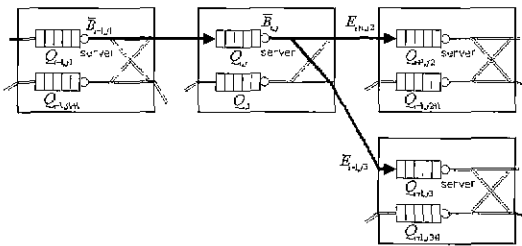
정의 :

- M : 버퍼 모듈에 있는 버퍼의 개수.
- n : MIN에 존재하는 스테이지의 개수. N 이 입력 또는 출력 포트 수일 때, $N \times N$ MIN에서의 스테이지 수는 $n = \log_2 N$ 으로 나타난다

- r : MIN을 구성하는 2×2 SE에서, 버퍼에 있는 패킷이 위쪽 출력 포트를 목적지로 향하고 있는 확률. 균등 입력 트래픽(uniform input traffic)의 경우에 r 의 값은 0.5이다.
- $Q_{i,j}$: 스위치 내에서 i 번째 스테이지, j 번째 포트에 위치하는 버퍼모듈.
- $E_{i,j}(t)$: 시간 t 에서, $i-1$ 번째의 스테이지에 있는 SE로부터 $Q_{i,j}$ 로 들어오려는 패킷이 존재하는 확률.
- $B_{i,j}(t)$: 시간 t 에서, $Q_{i,j}$ 에 있는 패킷이 블록되는 확률.
(편의상 앞으로 $Q_{i,j}$ 에서의 $B_{i,j}(t)$ 를 B 로, $E_{i,j}(t)$ 를 E 로 표기한다)
- $P_{i,j}(l, t)$: 시간 t 에서, $Q_{i,j}$ 의 버퍼 모듈에 l 개의 패킷이 있는 확률, 여기서 $0 \leq l \leq M$.
- $P_{i,j}^{blocked}(t)$: 시간 t 에서, $Q_{i,j}$ 의 버퍼 모듈에서 버퍼 모듈의 서버가 블록 상태에 있는 안정 상태(steady state)의 확률.
- $P_{i,j}^{extended}(0, t)$: 시간 t 에서, $P_{i,j}(0, t)$ 의 확장된 의미로 $Q_{i,j}$ 에서 나가려는 패킷이 없는 확률. 즉 $Q_{i,j}$ 가 비이 있거나 서버가 블록된 확률

3.2 버퍼의 상태와 상태식

앞에서 정의된 E 와 B 의 관계가 (그림 2)에 나타나 있으며, $Q_{i,j}$ 의 상태는 (그림 3)과 같이 각 패킷의 개수에 따른 각 상태에서 E 와 B 에 의해 전이되는 마르코프 체인(Markov Chain)으로 나타난다.

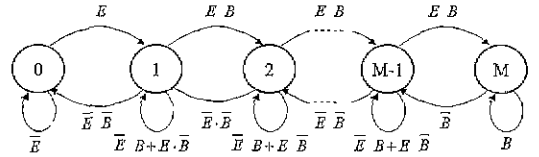


(그림 2) 스위칭 모듈 내에 있는 버퍼 모듈에서의 E 와 B 의 관계

이 상태도에서 각 상태의 상태식들은 다음과 같다.

$$P_{i,j}(0, t+1) = \bar{E} \cdot P_{i,j}(0, t) + \bar{E} \cdot \bar{B} \cdot P_{i,j}(1, t) \quad (1)$$

$$P_{i,j}(1, t+1) = E \cdot P_{i,j}(0, t) + (\bar{E}B + E\bar{B}) \cdot P_{i,j}(1, t) + \bar{E} \cdot \bar{B} \cdot P_{i,j}(2, t) \quad (2)$$



(그림 3) 버퍼 $Q_{i,j}$ 의 상태 전이에 따른 마르코프 체인

$$P_{i,j}(l, t+1) = E \cdot B \cdot P_{i,j}(l-1, t) + (\bar{E}B + E\bar{B}) \cdot P_{i,j}(l, t) + \bar{E} \cdot \bar{B} \cdot P_{i,j}(l+1, t) \quad (2 \leq l \leq M-2) \quad (3)$$

$$P_{i,j}(M-1, t+1) = E \cdot B \cdot P_{i,j}(M-2, t) + (\bar{E}B + E\bar{B}) \cdot P_{i,j}(M-1, t) + \bar{B} \cdot P_{i,j}(M, t) \quad (4)$$

$$P_{i,j}(M, t+1) = E \cdot B \cdot P_{i,j}(M-1, t) + B \cdot P_{i,j}(M, t) \quad (5)$$

3.3 계산 순서

1. $t=0$ 일 때, 첫번째 스테이지에서의 E 는 입력 로드로 초기화되며 모든 SE의 $P_{i,j}(0, t)$ 와 $P_{i,j}^{extended}(0, t)$ 는 1로 초기화된다. 다른 모든 변수들은 0으로 초기화된다.
2. $t = t+1$
3. 각 버퍼 모듈에서
 - 1) $P_{i,j}^{blocked}(t)$ 와 $P_{i,j}^{extended}(0, t)$ 를 계산한다.
 - 2) $E_{i,j}(t)$ 와 $B_{i,j}(t)$ 를 계산한다.
 - 3) $P_{i,j}(l, t) (0 \leq l \leq M)$ 를 상태식을 이용하여 구한다.
4. 처리를 T 와 지연시간 D 를 계산한다. 시간 t 에서 얻은 T 와 D 값과, 시간 $t+1$ 에서 계산된 T 와 D 값의 차이가 각각 10^6 보다 작은지를 비교하여 그러하면 종료하고, 그렇지 않으면 다시 2번 단계로 돌아간다.

T 와 D 의 닫힌 형태(closed form)의 계산식은 네트워크와 패킷 이동 구조의 복잡성으로 인해 계산하기가 매우 복잡하다. 그러므로 일반적으로 이들은 시스템이 안정화 상태(stable condition)에 도달할 때까지 반복적으로 계산함으로써 얻는다.

3.4 분석

기억 특성을 고려한 근사화 모델은 다음과 같다. 먼저 버퍼 모듈의 서버가 안정상태에서 블록 상태에 있는 확률을 근사화 방법으로 구한 $P_{i,j}^{blocked}(t)$ 가 설명된다.

$$P_{i,j}^{blocked}(t) = B_{i,j}(t-1) \times B_{i,j}(t-2) \quad (6)$$

식 (6)은 패킷이 이전 네트워크 사이클동안에 한번 이상 연속적으로 블록되었다면 그 패킷은 다음 네트워크 사이클에 다시 블록되는 확률이 상대적으로 높기 때문에, 이런 연속적인 블록킹 확률을 안정상태에서 서버의 지속적인 블록킹 확률로 근사화시켜서 나타내었다. 이러한 연속적인 블록킹의 경우는 ① 대부분의 목적지 버퍼 모듈이 차있거나, ② 패킷이 경쟁에 연속적으로 지는 경우에 생긴다 ①의 경우는 현실적인 입력 트래픽 상황에서 버퍼 모듈들은 거의 차있다는 사실[9]에 기반하며, ②의 경우는 대부분의 버퍼 모듈이 차있지 않은 상황에서는 거의 발생하지 않는다는 컴퓨터 시뮬레이션 분석 결과에 의거해서 고려하지 않는다. <표 1>은 균등 입력 트래픽에서 입력 로드 변화에 따른 각 스테이지에서의 $P_{i,j}^{blocked}$ 값의 변화를 보인다. 이 표에서 알 수 있듯이 $P_{i,j}^{blocked}$ 은 입력 로드가 크거나 비균등 트래픽 패턴의 상황에서와 같은 잦은 블록킹이 발생하는 경우에서만 0이 아닌 연속적인 블록킹 확률값을 갖게 되어 시스템 성능에 영향을 끼친다.

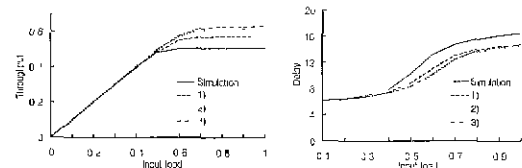
<표 1> 6-스테이지 3-버퍼 MIN에서의 입력로드 변화에 따른 $P_{i,j}^{blocked}$ 값

	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Stage 6
Load = 0.1	0.00069	0.00069	0.00069	0.00069	0.00065	0.0
Load = 0.3	0.00797	0.00796	0.00794	0.00791	0.00663	0.0
Load = 0.5	0.03367	0.03292	0.03199	0.02991	0.02013	0.0
Load = 0.6	0.07187	0.06398	0.05667	0.04713	0.02737	0.0
Load = 0.8	0.11824	0.09045	0.07361	0.05579	0.03029	0.0
Load = 1.0	0.12308	0.09269	0.07379	0.05638	0.03047	0.0

또한 이 식은 이전 네트워크 사이클에 두 번 연속적인 블록킹이 일어나는 경우의 확률 값으로 서버의 지속적인 블록킹 상태의 확률로 근사화하는데 그 이유는 다음과 같다 만약 이전 한번의 블록킹 값($B(t-1)$) 반으로 $P_{i,j}^{blocked}$ 값을 나타낸다면 단 한번의 블록킹으로 서버의 지속적인 블록킹 상황을 나타내게 되어 실제 상황보다 너무 많은 블록킹이 생기며 이로 인해서 시스템 모델의 결과는 수렴하지 못한다. 이전 사례들에 두 번 블록킹의 고려보다 많은 세 번 이상의 글록킹으로 $P_{i,j}^{blocked}$ 을 계산하는 경우는 계산된 값이 실제 상황보다 너무 작은 값을 갖게 되어 기억특성을 제대로 나타내지 못하게 되며, 고려 횟수를 더할수록 $P_{i,j}^{blocked}$ 은 0이 되어 기억특성을 전혀 고려하지 않은 모델 값에 수렴한다. (그림 4)에

서 $P_{i,j}^{blocked}$ 계산 시에 고려된 연속적인 블록킹 횟수에 따른 처리율 및 지연시간 결과를 보이는데 횟수를 더할수록 기억특성을 고려하지 않은 결과에 수렴함을 알 수 있다.

일 때 다음과 같다



- 1) $P_{i,j}^{blocked}(t) = B_{i,j}(t-1) \times B_{i,j}(t-2)$ 의 계산식을 적용한 모델
- 2) $P_{i,j}^{blocked}(t) = B_{i,j}(t-1) \times B_{i,j}(t-2) \times B_{i,j}(t-3)$ 의 계산식을 적용한 모델
- 3) $P_{i,j}^{blocked}(t) = 0$ (기억특성을 고려하지 않음)의 계산식을 적용한 모델

(그림 4) $P_{i,j}^{blocked}$ 계산시 블록킹 횟수에 따른 6-스테이지 3-버퍼 MIN의 처리율 및 지연시간

이 값을 이용해서 기억특성으로 발생하는 잦은 블록킹 상황을 근사적으로 모델링할 수 있다. 일기적인 기존의 방법에서는 안정화 상태의 확률 값을 이용하여 매우 어렵지만, 제안된 모델에서의 근사화 인접 상태 확률은 식 (6)과 같이 매우 간단하다. 다음으로 $P_{i,j}(0)$ 의 확장된(extended) 확률인 $P_{i,j}^{extended}(0)$ 를 설명한다

$$P_{i,j}^{extended}(0, t) = 1 - (1 - P_{i,j}^{blocked}(t)) \cdot (1 - P_{i,j}(0, t)) \\ = P_{i,j}^{blocked}(t) + P_{i,j}(0, t) - P_{i,j}^{blocked}(t) \cdot P_{i,j}(0, t) \quad (7)$$

만약 서버가 블록 상태에 있다면 버퍼 모듈의 패킷은 이동할 수 없으며, 이는 버퍼 모듈이 비어 있는 것과 같은 상태로 확장된 의미에서 고려할 수 있다. 즉 E 와 B 를 구할 때 $P_{i,j}(0)$ 대신 $P_{i,j}^{extended}(0)$ 를 사용한다. 다음은 $P_{i,j}^{blocked}$ 및 $P_{i,j}^{extended}(0)$ 를 사용한 E 와 B 의 식이다

$$E_{i,j}(t) = r \cdot (1 - P_{i-1,j}^{extended}(0, t)) \\ + r \cdot (1 - P_{i-1,j+1}^{extended}(0, t)) \\ - r^2 \cdot (1 - P_{i-1,j}^{extended}(0, t)) \\ \cdot (1 - P_{i-1,j+1}^{extended}(0, t)) \quad (8)$$

식 (8)은 시간 t 에서 $i-1$ 번째 스테이지에서 r 의 확률로 위쪽 버퍼 모듈 $Q_{i,r}$ 로 들어올려지는 패킷이 존재하는 확률이다. 아래쪽 버퍼 모듈로 들어오는 다른 경우는 r 대신 $(1-r)$ 을 사용함으로써 쉽게 구할 수 있다. 마지막으로 블록킹 확률 B 는 다음과 같이 얻는다.

$$B_{i,r}(t) = P_{i,r}^{blocked}(t) + (1 - P_{i,r}^{blocked}(t)) \times \left[\begin{array}{l} r \cdot P_{i+1,r2}(M, t) + \frac{1}{2} r^2 \\ \cdot (1 - P_{i,r}^{extended}(0, t)) \\ \cdot (1 - P_{i+1,r2}(M, t)) \\ + (1-r) \cdot P_{i+1,r3}(M, t) + \frac{1}{2} (1-r)^2 \\ \cdot (1 - P_{i,r}^{extended}(0, t)) \\ \cdot (1 - P_{i+1,r3}(M, t)) \end{array} \right] \quad (9)$$

서버가 블록 상태에 있다면($P_{i,r}^{blocked}$), 패킷은 블록된다. 서버가 활성(active) 상태에 있다면($1 - P_{i,r}^{blocked}$), 목적지 버퍼 모듈이 차있거나, 또는 목적지 버퍼 모듈이 차있지 않고 패킷이 경쟁에서 지는 경우에 패킷 블록킹이 발생한다.

정규화된 처리률은 다음과 같이 마지막 스테이지에서의 패킷이 전송되어지는 확률로서 정의된다. 식 (10)은 SE의 위쪽 포트의 경우를 나타내며 아래쪽 포트의 경우는 r 대신 $(1-r)$ 을 사용한다.

$$T_{n,r}(t) = r \cdot (1 - P_{n,r}(0, t)) + r \cdot (1 - P_{n,r+1}(0, t)) - r^2 \cdot (1 - P_{n,r}(0, t)) \cdot (1 - P_{n,r+1}(0, t)) \quad (10)$$

각 SE에서의 지연시간은 식 (11)과 같이 리틀의 식(Little's formula)을 사용해서 얻는다.

$$D_{i,r}(t) = \lim_{t \rightarrow \infty} \frac{\sum_{l=1}^M l \times P_{i,r}(l, t)}{T_{i,r}(t)} \quad (11)$$

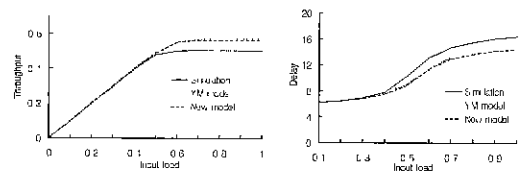
여기서 l 은 버퍼 모듈 내에 존재하는 패킷의 개수를 나타낸다. 다음으로 n 이 스테이지 수이며 N 이 포트 수일 때, 네트워크 전체의 패킷 평균 지연은 다음과 같이 구한다

$$D(t) = \frac{\sum_{i=1}^n \sum_{j=1}^N D_{i,r}(t)}{N} \quad (12)$$

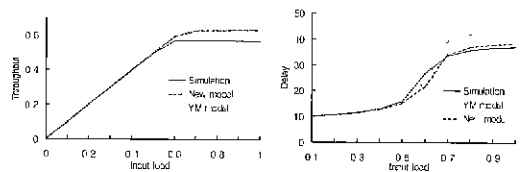
4. 제안된 모델의 검증

제안된 모델의 정확성을 검증하기 위해서 네트워크 처리율과 지연의 관점에서 얻어진 컴퓨터 시뮬레이션 결과와의 비교를 통해서 그 정확성을 나타낸다. 시뮬레이션을 위한 가정으로는 ① 각 프로세스는 네트워크 사이클마다 패킷을 입력 트래픽 로드(input traffic load)의 비율로 생성하며, ② 각 패킷의 목적지는 랜덤 생성기에 의해 랜덤하게 설정된다. ③ SE안에서 경쟁이 있다면 경쟁에 참여하는 각 패킷이 이길 확률은 같으며, ④ 버퍼는 FCFS로 작동하는 가정이 사용된다.

(그림 5)는 3개의 셀을 저장할 수 있는(3-버퍼) 버퍼 모듈을 가진 6-스테이지(64×64) MIN의 처리율과 지연을 나타낸다. 제공된 트래픽 로드는 0.1에서 1까지 변화한다. 제안된 분석 모델과 컴퓨터 시뮬레이션에서의 결과가 데이터로 쓰였으며, 기존의 모델과 비교하기 위해서, SCC개념의 입력 버퍼 모듈들 중 가장 잘 모델링 되었다고 평가 받는 [8]의 결과도 같이 비교되었다. 여기서 각 분석 모델과 시뮬레이션의 데이터는 10년 실행한 후의 평균값이다. 각 실행은 신뢰적인 데이터를 얻기 위해 1 000 000번 반복되었으며 마지막 1 000 000번째에서의 편차는 0.1% 미만이다.



(그림 5) 6-스테이지, 3-버퍼를 가진 MIN의 입력 로드 에 대한 처리율과 지연시간



(그림 6) 10스테이지, 6 버퍼를 가진 MIN의 입력로드에 대한 처리율과 지연시간

(그림 5)는 제안된 모델이 기존 모델[8]보다 훨씬 간단함에도 불구하고 비슷한 정확도를 가짐을 나타낸다. 보다 큰 크기 MIN의 결과는 (그림 6)에서 보이며, 이

경우 제안된 모델이 기존 모델보다 정확하다. 보통 MIN과 버퍼 크기가 증가할수록 제안된 모델이 기존 모델보다 좀 더 낮은 결과를 보인다.

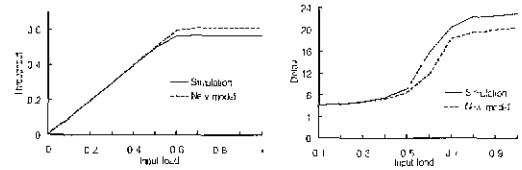
다음으로 미퍼 개수의 변화에 따른 처리율과 지연시간 결과를 설명한다. <표 2>는 6스테이지 MIN에서 입력 로드가 1일 때의 미퍼 크기 변화에 따른 처리율 및 지연시간을 나타낸다. 이를 살펴보면 제안된 모델의 처리율은 시뮬레이션 결과보다 약간 큰 값에서, 지연시간은 작은 값으로 일정하게 수렴됨을 알 수 있다. 분석 모델에서 지연시간을 나타내는 식은 안정 상태에서 처리율에 따른 버퍼 모듈 내의 패킷 개수로 나타나는 리틀의 식을 사용했으므로, 처리율이 시뮬레이션 결과보다 클 때 지연 시간은 작은 값을 갖는 결과는 당연하다. 그러나 [8]에서는 버퍼 모듈의 크기가 3일 때는 제안된 모델과 같이 지연 시간이 시뮬레이션의 결과보다 작은 값으로 수렴되지만, 버퍼 모듈의 크기가 5이상 일 때는 시뮬레이션의 결과보다 큰 값으로 수렴된다. 이러한 결과는 다른 크기의 MIN에서도 비슷하게 관찰되며, 이는 제안된 모델이 기존의 모델보다 버퍼 크기의 변화에 따른 지연 시간에서 좀 더 정확한 결과를 나타냄을 보여준다.

<표 2> 입력 로드가 1일 때, 6스테이지 MIN에서 버퍼 크기의 변화에 따른 처리율 및 지연시간

Buffer Size	Throughput			Delay		
	Simulation	[8]	New	Simulation	[8]	New
3	0.508	0.578	0.576	16.44	15.11	14.79
5	0.583	0.628	0.628	23.38	23.79	25.56
7	0.620	0.646	0.640	30.32	33.33	26.96
9	0.645	0.655	0.643	37.01	46.62	30.94

비균등 입력 트래픽의 한 종류인 EFOS(Even-First Odd-Second) 트래픽 패턴에 대한 제안된 모델의 결과가 (그림 7)에서 보여지며, 위의 균등 트래픽의 경우와 마찬가지로 정확한 결과를 나타냄을 알 수 있다. 버퍼 모듈 내에 출력 버퍼를 가진 MIN의 경우에서의 EFOS 트래픽에 대한 결과[10]는 패킷의 심한 경쟁이 발생하여 성능이 급격히 떨어짐이 보여졌지만, 제안된 모델과 같이 버퍼모듈 내에 입력 버퍼를 가진 MIN에서의 EFOS 트래픽 패턴의 경우에는 성능이 떨어지지 않는다. 이는 EFOS 트래픽은 그 특성상 첫번째 스테이지에서는 버퍼 모듈간의 패킷 경쟁이 발생하지 않기 때문이다 더욱이 작은 크기의 버퍼(3-buffer)인 경우는 균등 트래픽의

경우에 비해서 성능이 조금 증가하며(4%), 이러한 성능 증가분은 버퍼 크기가 커짐에 따라 점점 감소하여 큰 크기의 버퍼인 경우는 1%의 성능 증가를 나타낸다



(그림 8) 6스테이지, 4 버퍼를 가진 MIN의 EFOS 입력 트래픽에 대한 처리율과 지연시간

결국 이와 같은 제안된 모델의 분석 결과에 대한 검증은 통해서, 제안된 모델에서 사용된 방법인 안정 상태에서 버퍼 모듈 서버의 블록킹 상태를 나타내는 확률인 $P_{i,j}^{blocked}$ 를 이전 네트워크 사이클에서의 연속적인 블록킹 확률 값으로 근사화 시키고, 이 값에 기반하여 지속적인 블록킹 상태의 버퍼 모듈을 $P_{i,j}^{extended}(0)$ 으로 나타내는 방법이 정확하다는 점이 입증된다. 이러한 근사화 모델링 방법을 통해서 제안된 모델은 높은 정확성을 유지하면서도 매우 간단하다.

5. 결 론

본 논문에서는 Small Clock Cycle(SCC) 환경에서 스위칭 요소 내에 다중 입력 버퍼를 가진 MIN의 성능을 정확하게 나타내는 모델이 소개되었다. 기존 모델들은 대부분 정확하지 않거나, 정확하더라도 매우 복잡한 모델링 기법이 필요한 반면에, 본 논문에서 제안된 모델은 기존의 모델들과 비교해서 훨씬 간단한 모델링 방법이 사용되어 많은 양의 계산 오버헤드를 줄일 수 있었다. 제안된 모델의 결과는 컴퓨터 시뮬레이션을 통한 처리율과 지연시간의 관점에서 검증되었으며 또한 기존 모델 중 가장 정확한 값을 갖는 모델과 비교되었다. 기존 모델 및 시뮬레이션과의 비교 결과로써, 제안된 모델은 MIN의 크기와 MIN에서의 스위칭 모듈의 버퍼 크기가 증가할수록 기존 모델보다 정확한 결과를 나타냈으며 비균등 입력 트래픽의 한 종류인 EFOS 트래픽에서도 정확한 결과를 나타냈다. 기가비트 이더넷 스위치와 같은 다른 구조의 스위치나, 다른 형태의 비균등(nonuniform) 입력 트래픽에서의 MIN 모델링 방법 또한 연구중이다

참 고 문 헌

[1] C.L. Wu and T.Y. Feng, "On a class of Multistage Interconnection Networks," IEEE Trans. on Computers, Vol.C-29., pp.694-702, August 1980.

[2] CCITT Recommendation L121, "Broadband aspects of ISDN" Blue Book, Vol.III.7, Geneva, Switzerland, 1989.

[3] J.S. Turner. "Design of and integrated services packet networks," Ninth Data Commun. Symp., in ACM SigComm Comput. Commun. Rev., Vol.15, pp 124-133, Sept. 1985

[4] Muh-rong Yang and GnoKou Ma, "BATMAN . A New Architectural Design of a Very Large Next Generation Gigabit Switch," IEEE International Conference on Communications. Vol.2/3. pp 740-744, May 1997.

[5] H. Rudm. "The ATM-Asynchronous Transfer Mode," Computer Networks and ISDN Systems, Vol.24, pp.277-278, 1992.

[6] Y. Mun and H.Y. Youn, "Performance Analysis of Finite Buffered Multistage Interconnection Networks," IEEE Trans. on Computers, pp 153-162, Feb 1994.

[7] J.Ding and L.N. Bhuyan, "Performance evaluation of multistage interconnection networks with finite buffers," Int'l. Conf. On Parallel Processing, pp.592-595, August 1991.

[8] H.Y. Youn and Y. Mun, "On Multistage Interconnection Networks with Small Clock Cycles," IEEE Trans. on Parallel and Distributed Systems, pp.86-93, Jan. 1995.

[9] H.Y. Youn and H. Choo. "Performance Enhancement of Multistage Interconnection Networks with Unit Step Buffering," IEEE Trans on Commun. Vol. 47, No.4, April 1999.

[10] T. Lin and L Kleinrock, "Performance Analysis of Finite-Buffered Multistage Interconnection Networks with a General Traffic Pattern," ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, San Diego, CA. pp. 68-78. May 21-24. 1991.

[11] H.Y. Youn and H. Choo. "Packet Switched Multistage Interconnection Networks with Multiple Packet Movement Per Network Cycle," Inter-

national Conf. Parallel and Distributed Computing Systems. pp.183-188, Sept. 1995.



최 원 제

e-mail : lance@ece.skku.ac.kr
 1992년 성균관대학교 공과대학
 정보공학과 졸업(학사)
 1992년~현재 성균관대학교 전기
 전자 및 컴퓨터공학부
 석사과정

관심분야 : ATM, 고속통신망, 망 디자인



추 현 승

e-mail choo@ece.skku.ac.kr
 1988년 성균관대학교 이과대학
 수학과 졸업(학사)
 1990년 텍사스 주립대(달라스)
 전자계산학과(공학석사)
 1996년 텍사스 주립대(알링턴)
 전산공학과(공학박사)

1997년~1998년 특허청 심사4국 컴퓨터심사담당관실
 심사관

1998년~1999년 성균관대학교 전기전자 및 컴퓨터공학
 부 정인강사

1999년~현재 성균관대학교 전기전자 및 컴퓨터공학부
 조교수

관심분야 : ATM, 병렬 및 분산 처리, 알고리즘 해석,
 고속통신망



문 영 성

e-mail . mun@computing.soongsil.ac.kr
 1983년 연세대 전자공학과 학사
 1986년 캐나다 알버타대 전기공학
 과 석사
 1993년 텍사스 주립대(알링턴) 컴
 퓨터 공학 박사

1992년 Supercomputing'92 최우수학생논문상 수상

1987년~1994년 한국통신 연구원

1994년~현재 숭실대 컴퓨터학부 부교수

1999년~현재 Journal of Supercomputing 편집위원

1999년 정보처리학회 추계학술대회 최우수논문상 수상

2000년~현재 정보처리학회 국제공동학술대회 위원장