

USB방식을 적용한 MIN기반 교환기 구조의 모델링 및 성능평가

홍유지, 추현승, 윤희용

Modeling and Performance Evaluation of Multistage Interconnection Networks with USB Scheme

Yuiee Hong, Hyunseung Choo, Hee Yong Youn

Abstract

One of the most important things in the research for MIN-based switch operation is the management scheme of network cycle. In the traditional MIN, when the receiving buffer module is empty, the cell has to move forward the front-most buffer position by the characteristic of the conventional FIFO queue. However, most of buffer modules are almost always full for practical amount of input loads. The long network cycle of the traditional scheme is thus a substantial waste of bandwidth. In this paper, we propose the modeling method for the input and multi-buffered MIN with unit step buffering scheme. In spite of simplicity, simulation results show that the proposed model is very accurate comparing to previous modeling approaches in terms of throughput and the trend of delay.

Key Words: 지연, 다중패킷전송, 상호연결망, 패킷교환, 처리율

1. 서론

다단계 상호연결 네트워크(Multistage Inter-connection Network : MIN)는 지난 몇 년 동안 텔레커뮤니케이션과 병렬 컴퓨터 시스템[1]과 같은 분야에서 중요한 응용으로 이용되었다. MIN은 ATM 스위치를 포함한 다양한 네트워크 스위치나 라우터의 교환구조이고, 또한 병렬 컴퓨터 내부에 상호 연결망을 구성하는데 널리 사용되었다. ATM[2]은 셀(cell)이라고 불리는 짧고 고정된 크기(53바이트)의 패킷을 가지고 있는 고속의 패킷 스위칭 기술이고, 종합 정보 통신망(ISDN)에 매우 효과적이며, 또한 화상 통신과 그래픽 관련 응용들을 위한 광대역 서비스들을 지원하는 통신방식이다.

MIN의 동작 연구에 있어서 가장 중요한 것 중의 하나가 네트워크 사이클의 운영 방법이라 할 수 있다. 일반적인 MIN에서 수신 버퍼 모듈이 비어있어서 셀이 가장 앞쪽의 버퍼 위치로 움직여야만 하는 상황의 발생이 최악의 경우라 할 수 있다. 결과적으로 네트워크 사이클은 임의의 상태로의 완전한 셀 이동을 보장할 정도로 충분히 길어야만 한다. 그러나 실제적으로는 입력 로드의 실질적인 크기에 대해서 대부분의 버퍼 모듈들은 거의 항상 꽉 차게 된다. 그렇기 때문에 일반적인 체계의 긴 네트워크 사이클은 대역폭의 낭비를 가져오게 된다[3]. 그러나 만약 더욱 짧은 네트워크 사이클 동안에 각 셀이 하나의 단위 버퍼 위치로만 움직이게 한다면, 네트워크 사이클의 대역폭은 보다 효율적으로 사용하게 된다. 이런 기존의 개념인 단위 단계 버퍼링(Unit Step Buffering : USB)체계는 [4]에서 소개되었다.

일반적인 체계[5-12]와 비교한 단위 단계 버퍼링(USB) 체계의 성능은 ATM 스위칭 환경의 보편적인 트래픽 부하 상에서 실제 크기의 MIN에서 상당히 개선되었을 뿐만 아니라 큰 패킷 크기를 가지는 일반적인 병렬 컴퓨터 시스템에서도 더욱 크게 개선되었다. 하지만 더욱 중요한 가치를 부여할 수 있는 것은 어떤 하드웨어나 동작의 오버헤드 없이 단지 스위칭 구조를 변형함

로서 이루어졌다는 것이다. 단위 단계 버퍼링 체계의 성능이 개선되었음에도 불구하고, 이 체계의 모델링 방법이 매우 복잡하기에 다른 유사한 MIN기반 구조에 적용하기가 어렵고 처리율과 지연시간에서 정확하지 못한 결과를 나타내었다.

본 논문에서는 이런 기존의 문제점을 수정하여 기존에 연구된 네트워크 사이클에서 단위 단계 버퍼링 체계의 접근 방식을 따르는 입력 버퍼에 관한 분석 모델을 제안하였다. 이 제안된 모델은 매개변수의 수와 상태 전이 다이어그램이 기존의 모델링 기법들보다 매우 단순하여 계산상의 오버헤드를 줄였을 뿐만 아니라 이런 단순함에도 불구하고 분석 결과가 시뮬레이션을 통해 얻은 처리율과 지연 시간에 있어서 상당히 유사한 결과를 보여주었다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 논문에서 제안한 모델과 관련된 핵심 내용을 살펴보고, 3장에서는 제안된 분석 모델을 설명한다. 4장에서는 제안된 분석 모델과 시뮬레이션에 대한 결과를 살펴봄, 5장에서는 본 논문에 대한 결론을 내린다.

2. 관련연구 및 배경설명

이 절에서는 기존에 연구된 네트워크 사이클 운영방법과 단위 단계 버퍼링에 대해서 살펴보고, MIN의 스위칭 요소(switching element: SE) 내부에 위치하고 있는 버퍼는 MIN의 성능을 현저히 향상시키며, 기존에 연구된 네트워크 사이클의 패킷 이동은 마지막 스테이지로부터 처음 스테이지로 필요한 요소들을 재귀적으로 계산된 전체 스테이지의 버퍼 가용성 정보를 이용하여 이동한다. 이런 MIN내의 버퍼에서 패킷이 블록 되는 경우는 다음 스테이지에 있는 패킷의 목적지 버퍼에 가용한 공간이 없거나, 블록킹 스위치의 특성에 의해 경쟁관계에 있는 버퍼에 있는 패킷과 출력에 대한 경쟁에서 지는 경우이다. 일단 한번 블록된 패킷은 원래 목적지를 다음 네트워크 사이클에서도 그대로 향하려고 하는 기억 특성을 가지는데, 이러한 기억 특성이 있기 때문

에 MIN의 모델링을 정확히 하는데 어려움이 있다[13].

일반적인 MIN에 존재하는 버퍼에서의 셀은 한번의 네트워크 사이클 동안 다음 단계의 SE로 전진하기 때문에 수신 버퍼 모듈의 꼬리에 위치하게 된다. 여기서 '꼬리(tail)'란 버퍼 모듈에서의 마지막 엔트리를 지칭하며, 반면에 '끝(end)'은 물리적으로 마지막 버퍼 엔트리를 나타낸다. 그렇기 때문에 꼬리는 버퍼 모듈 안에 존재하는 셀들의 수에 따라 변화하지만 끝은 그렇지 않다. 일반적인 MIN에서 수신 버퍼 모듈이 비어 있어서 셀이 가장 앞쪽의 버퍼 위치로 움직여야만 하는 최악의 상황이 발생한다. 결과적으로 네트워크 사이클은 임의의 상태로의 완전한 셀 이동을 보장할 정도로 충분히 길어야만 하지만 실제적으로는 입력 로드의 실질적인 크기에 대해서 대부분의 버퍼 모듈은 거의 항상 꽉 차게 되기 때문에 일반적인 체계에서의 긴 네트워크 사이클은 대역폭의 낭비를 가져오게 된다[4].

단위 단계 버퍼링 체계에서는 더욱 짧은 네트워크 사이클 동안에 각 셀이 하나의 단위 버퍼 위치로만 움직이게 되어 네트워크 사이클의 대역폭이 완전하게 사용되기 때문에 대역폭의 낭비를 줄일 수 있게 된다. 이처럼 단위 단계 버퍼링 체계에서는 들어오는 셀들은 그 앞에 아무런 빈 버퍼가 있다 하더라도 항상 버퍼 모듈의 끝에 위치한다[4].

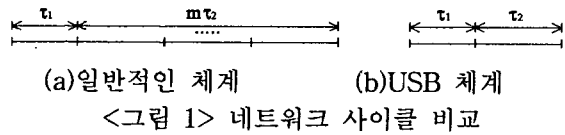
단위 단계 버퍼링 체계에서는 일반적인 MIN보다 짧은 네트워크 사이클 동안에 각 셀들이 하나의 단위 버퍼로 이동하기 때문에 프론트 셀과 헤드 셀이 다를 수 있다는 것이다. 프론트 셀(front cell)이란 앞 버퍼에 위치한 셀이라는 뜻이며, 반면에 헤드 셀(head cell)은 버퍼의 가장 앞쪽에 있는 셀을 뜻한다. 버퍼 모듈이 셀들을 갖고 있다 할지라도 프론트 셀을 가지지 않을 수 있다. 다시 말해 프론트 셀은 항상 헤드 셀이지만, 그 역이 항상 참인 것은 아니다. 매년 네트워크 클럭 사이클마다 프론트 셀은 서로 경쟁하기 때문에, 버퍼 모듈의 프론트 셀은 블록 되거나 블록 되지 않을 수 있다는 점은 중요한 사실이다[4].

프론트 셀이 블록되었을 때, 블록된 셀의 직접적으로 뒤에 있는 인접한 모든 연속된 셀들은 전진할 수 없으며 이것은 head of line blocking이라 불리어 진다. Head of line blocking에 의해 블록된 셀은 프론트 셀이 될 때까지는 경쟁에 참여할 수 없기 때문에 어떤 미리 정해진 목적지 노드를 갖지 않는다. 다르게 표현하자면, head of line blocking을 경험한 셀들과 새로이 도착한 셀들을 구별할 필요가 없다. 블록된 셀은 다음의 네트워크 클럭 사이클에 다시 원래의 목적지로 향하게 된다. 만약 블록된 셀이 아니라면 노말 셀(normal cell)로 간주되어 진다.

MIN에서 네트워크 성능의 레벨을 달성하기 위해 버퍼 모듈은 채널 충돌에 의한 패킷(혹은 셀)의 손실을 피하고, 블록된 패킷을 유지하고 있으므로 실제적인 네트워크 성능의 레벨을 달성할 수 있다.

버퍼 모듈은 first-in first-out(FIFO)동작을 갖고 있는 큐로 설명되며, 버퍼는 각각의 버퍼 엔트리로 언급되어 진다. m 을 버퍼 모듈의 크기로 하고, r 를 버퍼의 내용물을 한 위치만큼 상대적으로 움직이는데 필요한 시간이라고 하자. 만약 버퍼 모듈에 $b(0 \leq b \leq M)$ 만큼의 셀들이 있다고 한다면, 버퍼 모듈은 기본적으로 쉬프트-레지스터이기 때문에, 들어오는 셀을 버퍼 모듈의 꼬리에 놓기 위해서는 $(m - b)r$ 만큼의 시간이 걸린다. 만약 버퍼 모듈이 비어있다면 새로운 들어오는 셀이 버퍼 모듈의 머리(head)로 전진하는데 걸리는 시간은 mr 가 되며, 이는 셀이 하나의 SE로부터 다른 SE로 움직이는데 걸리는 기간 중 가장 긴 시간이 된다. 그러므로 보내는 버퍼 모듈과 받는 버퍼 모듈 사이에서 움직이는 한 완전한 셀의 보증 기간은 mr 가 된다. 그러나 정상 상태에서 현실적인 트래픽량을 갖고 있는 MIN의 버퍼 모듈들은 항상 거의 꽉 차기 때문에 대부분의 셀 움직임은 가장 긴 지연시간인 mr 시간보다는 훨씬 작을 것이다. 그러므로 네트워크 사이클 대역폭의 낭비로 인한 성능 저하가 발생할 것이다. 일반적인 셀 이동 체계에서 이런 커

다란 대역폭의 손실을 피하기 위해서 매번의 사이클마다 한 단계 이동만을 허용하는 짧은 클럭 사이클을 이용한다. 일반적인 체계에서의 네트워크 사이클 τ_c 는 아래의 <그림 1>에서 보인 두 시기의 합인 $\tau_c = \tau_1 + m\tau_2$ 이다. 전송 버퍼 모듈은 첫째 시기에서는 τ_1 동안 수신 버퍼 모듈들의 공간 가용성을 검사하며, 둘째 시기에서 만약 그 공간이 사용 가능하다면 $m\tau_2$ 동안에 헤드셀을 보낸다[4].

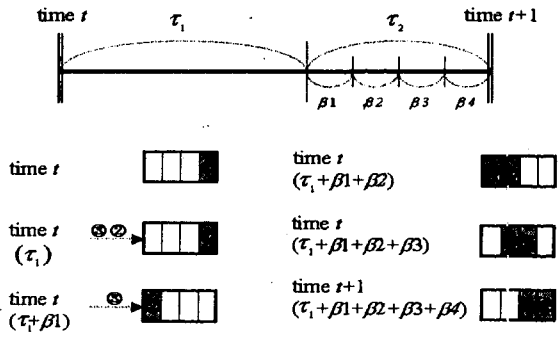


단위 단계 버퍼링 체계에서의 네트워크 사이클인 τ_u 는 $\tau_1 + \tau_2$ 로 된다. 먼저 첫째 시기를 살펴보도록 하자. 일반적인 MIN에서 버퍼 가용성의 정보를 전달하는 신호들은 마지막 스테이지로부터 첫 번째 스테이지로 첫째 시기에 전파되어 진다. 그러므로 첫째 시기의 길이는 n 이 단계들의 개수라고 했을 때, 단계들 사이에서 한 셀의 움직임을 위한 유닛 사이클 시간을 소비하는 시간인 n 이 된다. 이처럼 대부분의 버퍼 모듈은 거의 꼭 찬다는 사실과 짧은 네트워크 사이클 시간으로 인해서 단위 단계 버퍼링은 일반적인 체계보다 네트워크 사이클 시간을 효율적으로 이용함으로써 인해서 성능이 뛰어나다.

다음 <그림 2>에서는 <그림 1>에서 살펴본 것에 대한 이해를 돕기 위해서 버퍼 모듈에서의 패킷의 동작에 대한 하나의 간단한 실례를 통해서 살펴보도록 한다.

여기서 τ_1 는 전송하는 패킷의 라우팅 정보에 대한 시간 τ_2 는 패킷의 전송 시간을 나타낸다. 버퍼 모듈은 시프트 레지스터로서, 첫 번째 패킷(②)은 버퍼모듈의 끝에 도달한 시간($a + \beta 1$)후에, 버퍼 모듈의 끝에 있는 패킷은 버퍼 모듈의 헤드 쪽으로 움직이는 시간($\beta 2 + \beta 3 + \beta 4$)이 보장되어야 한다. 두 번째 패킷(③)이 시간($\tau_1 + \beta 1 + \beta 2$)에 버퍼

모듈로 들어갈때, 그 패킷은 첫 번째 패킷(②)의 보장된 시간($\beta 3 + \beta 4$)안에 움직이게 된다. 그러므로 버퍼모듈은 네트워크 사이클의 변화없이 두개의 패킷을 받을 수 있다.



<그림 2> 버퍼 모듈에서의 패킷의 동작

3. USB방식에서 입력버퍼 분석모델

이 절에서는 기존에 연구된 네트워크 사이클에서 단위 단계 버퍼링 체계를 기반으로 하는 유한개의 입력 버퍼를 가진 기존의 모델[4]보다 간단한 모델링 방법을 사용하면서도 정확한 성능을 나타내는 MIN의 성능 분석을 위한 모델을 제시한다.

3.1 정의와 가정

우선 제안된 분석 모델에서 사용되는 가정은 다음과 같다. 네트워크 사이클과 버퍼링 전략은 일반적인 MIN 체계를 따르지 않으며, 입력 트래픽 로드는 균등하게 분산되었다. 제안된 분석 모델에서 사용되는 정의는 다음과 같다.

- 정의:
- m : 버퍼 모듈에 있는 버퍼의 개수
 - n : MIN에 존재하는 스테이지의 개수
 - Q_i : 스위치 내에서 스테이지 i 번째 버퍼 모듈
 - $E_i(t)$: 시간 t 에서 $t+1$ 동안에, $i-1$ 번째 스테이지에 있는 SE로부터 Q_i 로 들어오려는 패킷이 존

재하는 확률

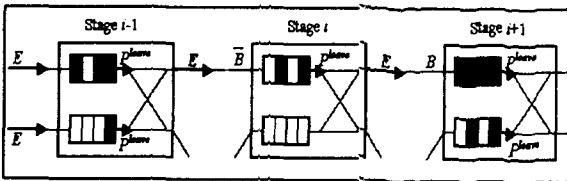
$B_i(t)$: 시간 t 에서 $t+1$ 동안에, Q_i 에 있는 패킷이 블록되는 확률. (편의상 앞으로 $E_i(t)$ 는 E 로, $B_i(t)$ 는 B 로 나타낸다.)

$P_i(l, t)$: 시간 t 에서 $t+1$ 동안에, Q_i 의 버퍼 모듈에 l 개의 패킷이 있는 확률, 여기서 $0 \leq l \leq M$

$P_i^{blocked}(t)$: 시간 t 에서 $t+1$ 동안에, Q_i 의 버퍼 모듈에서의 서버가 블록 상태에 있는 안정 상태 (steady state)의 확률

$P_i^{leave}(t)$: 시간 t 에서 $t+1$ 동안에, Q_i 의 버퍼 모듈에서 떠나려는 패킷의 확률.

다음의 <그림 3>은 E 와 B 의 관계를 나타내었다.



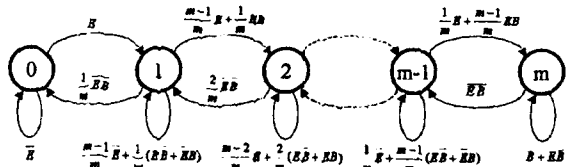
<그림 3> 스위칭내 버퍼에서의 E 와 B 의 관계

위의 <그림 3>에서 나타낸바와 같이 $Stage i$ 에서의 E 와 B 그리고 P^{leave} 를 나타내었다. 이 논문에서 나오는 첨자 i 는 스테이지를 나타내며 E 는 그 이전의 $Stage i-1$ 에서 $Stage i$ 의 버퍼 모듈로 들어오는 패킷이 존재하는 확률을 의미하는 것이며, B 는 한 네트워크 사이클 동안에 $Stage i$ 의 버퍼 모듈에 있는 패킷이 블록되는 확률을 나타내는 것이다. 마지막으로 P^{leave} 는 $Stage i$ 의 버퍼 모듈에서 다음의 스테이지 $Stage i+1$ 로 떠나려는 패킷이 존재하는 확률이다. 즉, 헤드패킷이 존재하는 확률을 나타내는 것이라 할 수 있다. 다음에서는 이를 바탕으로 각 버퍼의 상태에 따른 상태식에 대해 살펴보도록 한다.

3.2 버퍼의 상태와 상태식

각각의 매개변수 사이의 관계는 <그림 2>에 나타내었으며, 버퍼 모듈 Q_i 에 대한 마르코프 체인 (Markov Chain)은 <그림 3>에 나타내었다. 다음의 <그림 3>은 각 패킷의 개수에 따른 각 상태에서 E 와 B 에 의해 전이되는 특징이 있다. $Stage i$ 에서는 버퍼 모듈에 i 개의 셀들이 있는 것을 나타낸다. 예를 들어 $Stage 1$ 로부터 $Stage 2$ 로 상태가 전이한다고 할 때, 현재 상태가 $Stage 1$ 이라고 가정하자. $Stage 2$ 로 전이하기 위해서는 버퍼 모듈의 하나의 셀이 프론트 셀 ($1/m$)이면서 현재 네트워크 사이클에서 블록 되어야만 하며, 하나의 셀($m-1/m$)이 들어와야만 한다. 그러므로 $Stage 1$ 로부터 $Stage 2$ 로 전이되는 확률은 $\frac{m-1}{m} E + \frac{1}{m} EB$ 이 된다.

여기에서 프론트 셀은 프론트 버퍼에 존재하는 셀을 나타내는 것으로 상태 방정식은 마르코프 체인에 의해 쉽게 나타낼 수 있다.



<그림 4> 버퍼모듈 Q_i 의 상태전이에 따른 마르코프 체인

<그림 4>에서 각 상태의 상태식은 다음과 같다.

$$P_i(0, t+1) = \bar{E} \cdot P_i(0, t) + \frac{1}{m} \cdot \bar{E} \cdot \bar{B} \cdot P_i(1, t)$$

$$P_i(1, t+1) = E \cdot P_i(0, t) + \left\{ \frac{1}{m} (\bar{E} \cdot B + E \cdot \bar{B}) + \frac{m-1}{m} \cdot \bar{E} \right\} \cdot P_i(1, t) + \frac{2}{m} \cdot \bar{E} \cdot \bar{B} \cdot P_i(2, t)$$

$$P_i(k, t+1) = \left\{ \frac{m-k+1}{m} \cdot E + \frac{k-1}{m} \cdot E \cdot B \right\} \\ \cdot P_i(k-1, t) \\ + \left\{ \frac{k}{m} (\bar{E} \cdot B + E \cdot \bar{B}) + \frac{m-k}{m} \cdot \bar{E} \right\} \\ \cdot P_i(k, t) + \frac{k+1}{m} \bar{E} \cdot \bar{B} \cdot P_i(k+1, t)$$

($2 \leq k \leq m-2$)

$$P_i(m-1, t+1) = \left\{ \frac{2}{m} \cdot E + \frac{m-2}{m} \cdot E \cdot B \right\} \\ \cdot P_i(m-2, t) \\ + \left\{ \frac{m-1}{m} (\bar{E} \cdot B + E \cdot \bar{B}) + \frac{1}{m} \cdot \bar{E} \right\} \\ \cdot P_i(m-1, t) + \bar{E} \cdot \bar{B} \cdot P_i(m, t)$$

$$P_i(m, t+1) = \left\{ \frac{1}{m} \cdot E + \frac{m-1}{m} \cdot E \cdot B \right\} \\ \cdot P_i(m-1, t) + (B + E \cdot \bar{B}) \cdot P_i(m, t)$$

3.3 성능측정

MIN의 성능을 평가하기 위해서 정규화 된 처리율과 셀당 평균 지연이라는 두 가지 측정 요소에 의해 측정된다. 정규화된 처리율은 한 네트워크 사이클 동안에 출구 당 네트워크에서 나가는 셀들의 평균 개수이며, 평균 지연은 셀 당 네트워크에서 보낸 총 시간으로서, 이들 측정에 관한 절차는 다음과 같은 특징을 나타내고 있다.

3.1.1 계산순서

1. $t=0$ 일때, 첫번째 스테이지에서의 E 는 입력 로드로 초기화되며 각 버퍼 모듈의 $P_i(0, t)$ 는 1로 초기화된다. 다른 모든 변수들은 0으로 초기화 된다.

2. $t=t+1$

3. 각 버퍼 모듈에서

3.1 $P_i^{blocked}(t)$ 와 $P_i^{ave}(t)$ 를 계산한다.

3.2 $E_i(t)$ 와 $B_i(t)$ 를 계산한다.

3.3 $P_i(l, t)$ ($0 \leq l \leq M$)를 상태식을 이용하여 구한다.

4. 처리율 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 작을지를 비교하여 참이면 종료하고, 거짓이면 다시 2번 단계

로 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

셀이 블록될 때마다, 블록된 셀은 같은 출력 포트[14]로 가려고하는 기억 특성을 유지하며 정확한 MIN모델링은 이런 기억 특성에 의해 매우 복잡하다. 기억 특성을 고려한 근사화 모델은 다음과 같다. 먼저 버퍼 모듈의 서버가 안정 상태에서 블록 상태에 있는 확률을 근사화 방법으로 구한 $P_i^{blocked}(t)$ 가 설명된다.

$$P_i^{blocked}(t) = B_i(t-1) \times B_i(t-2)$$

위의 식은 패킷이 이전 네트워크 사이클 동안에 한번 이상 연속적으로 블록 되었다면 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 대부분의 목적지 버퍼 모듈이 차있거나, 패킷이 경쟁에서 연속적으로 지는 경우에 발생한다. 전자의 경우는 현실적인 입력 트래픽 상황에서 버퍼 모듈들은 거의 차 있다는 사실에 기반하며, 후자의 경우는 대부분의 버퍼 모듈이 차 있지 않은 상황에서는 거의 발생하지 않는다는 컴퓨터 시뮬레이션 분석 결과[15]에 의거해서 고려하지 않는다.

$P_i^{blocked}(t)$ 값을 이용해서 기억 특성으로 발생하는 잦은 블록킹 상황을 근사적으로 모델링할 수 있다. 일반적으로 기존의 방법에서는 안정화 상태의 확률 값을 얻는 것은 매우 어렵지만 제안된 모델에서의 근사화 안정 상태 확률은 위의

식과 같이 매우 간단하다. 다음은 버퍼 모듈을 떠나려는 패킷이 존재하는 확률은 다음과 같다.

$$P_i^{\leq ave}(t) = \overline{P_i^{blocked}(t)} \times \sum_{k=1}^m \frac{k}{m} P_i(k, t)$$

만약 버퍼 모듈 $P_i(k, t)$ 에 k 개의 셀들이 있다면, 그것들 중에서 하나는 프론트 셀 (k/m)이고 버퍼 모듈에서의 서버는 블록되지 않은 상태 ($\overline{P_i^{blocked}(t)}$)이면 셀은 버퍼 모듈을 떠날 수 있게 된다. 그러므로 $P_i^{\leq ave}(t)$ 는 위에서와 같이 나타낼 수 있게 된다. 여기에서 구한 $P_i^{\leq ave}(t)$ 을 통해서 E 와 B 를 쉽게 구할 수 있다.

$$E_i(t) = P_{i-1}^{\leq ave}(t) - 0.25 \cdot P_{i-1}^{\leq ave}(t)^2$$

$E_i(t)$ 는 Stage-($i-1$)에서의 셀들이 시간 t 에서 버퍼 모듈 Q_i 로 들어갈 준비가 되어 있을 확률을 나타낸다. 이런 경우에, 이전 단계에서 두개의 셀들은 $0.5(2 \times 0.5 \times P_{i-1}^{\leq ave}(t) = P_{i-1}^{\leq ave}(t))$ 의 확률을 가지고 Q_i 로 들어갈 수 있다. 그러나 동시에 $(0.25 \cdot P_{i-1}^{\leq ave}(t)^2)Q_i$ 로 들어갈 수는 없다. 따라서 위에서와 같이 $E_i(t)$ 는 나타낼 수 있다. 그러면 마지막으로 블록되는 확률 $B_i(t)$ 은 다음과 같다.

$$B_i(t) = \overline{P_i^{blocked}(t)} + \overline{P_i^{blocked}(t)} \times \{P_{i+1}(m, t) \cdot B_{i+1}(t) + \overline{P_{i+1}(m, t)} \cdot 0.25 \cdot P_{i+1}^{\leq ave}(t)\}$$

만약 Q_i 에서의 서버가 블록된 상태 ($P_i^{blocked}(t)$)이면 Q_i 에서의 셀들도 역시 블록된다. 또한 Q_i 에서의 서버가 블록되지 않은 상태 ($\overline{P_i^{blocked}(t)}$)이면, Q_i 에서의 셀들은 목적지 버퍼 모듈이 꽉찬 상태 ($P_{i+1}(m, t)$)이면서 블

록되어 있는 상태이거나 목적지 버퍼 모듈이 꽉 차지 않았을 경우 ($\overline{P_{i+1}(m, t)}$)는 경쟁에서 졌을 경우 ($0.25 \cdot P_{i+1}^{\leq ave}(t)$)에 블록된다.

정규화된 MIN의 처리율은 MIN이 정상 상태에 있을 때, 네트워크 사이클 당 네트워크에서 나가는 셀들의 수로 정의되는 것으로 아래의 식과 같이 마지막 스테이지에서의 패킷이 전송되어지는 확률로서 정의된다. 따라서 다음의 식은 SE의 위쪽 포트의 경우를 나타내며 아래쪽 포트의 경우는 r 대신 $(1-r)$ 을 사용한다.

$$T_n(t) = P_n^{\leq ave}(t) - 0.25 \cdot P_n^{\leq ave}(t)^2$$

평균 지연시간은 셀 당 네트워크에서 보낸 총 시간의 평균으로서 정의된다. 제안된 체계에서 버퍼 모듈의 작동 구조는 전통적인 FIFO체계가 아니기 때문에 직접 Little의 공식을 적용할 수 없다[14].

셀이 앞으로 한 위치를 전진하는데 필요한 시간인 $D_s(n_c)$ 를 정의함으로써, 버퍼 모듈에서의 지연을 얻는다. D_s 는 명백히 버퍼에 있는 셀의 수인 n_c 의 함수이다. 지연은 버퍼에 있는 셀의 수에 선형적으로 의존적이지 않음을 나타냄에 주의하라. 이는 숫자가 작을수록, head of the line blocking에 의한 지연의 기회가 줄어들기 때문이다. 그러므로, 우리는 양의 기울기를 가진 2차 함수 $D_s(n_c) = an_c^2 + 1$ 를 가정하자. 위의 사실과 두개의 조건 상태 $D_s(0) = 1, D_s(m) = 1/\overline{B}$ 을 사용해서 $D_s(n_c)$ 을 다음과 같이 나타낼 수 있다.

$$D_s(n_c) = \left(\frac{1}{\overline{B}} - 1\right) \frac{1}{m^2} n_c^2 + 1$$

i 번째 스테이지에서의 지연 시간은 다음과 같이 나타낼 수 있다.

$$D(i) = \lim_{i \rightarrow \infty} \left\{ (m-1)D_s(N_i) + \frac{1}{\overline{B}} \right\}$$

여기서 N_i 는 $N_i = \sum_{k=1}^m k \cdot P_k(t)$ 로 나타낸 i 번째 스테이지 버퍼 모듈에서 셀의 평균수이다. 위의 식에서, 첫 번째 식은 버퍼의 끝에서 프론트 버퍼에 도달하기 위해 새롭게 출발하는 셀에 대한 지연 시간인 반면, 두 번째 식은 프론트 셀이 i 번째 스테이지의 버퍼를 떠나는데 따른 시간을 나타낸다. 마지막으로, 전체 네트워크를 통해서 움직이기 위한 평균 지연 시간(Delay)은 다음과 같다.

$$\nabla ay = \sum_{i=1}^m D(i)$$

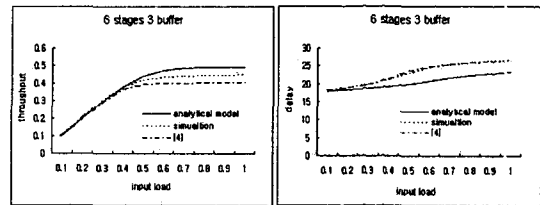
다음에서는 기존에 연구된 네트워크 사이클에서 단위 단계 버퍼링 체계 방식을 따르는 입력 버퍼에 관한 분석 모델과 컴퓨터 시뮬레이션을 사용하여 제안된 체계의 성능을 평가한다.

4. 성능평가

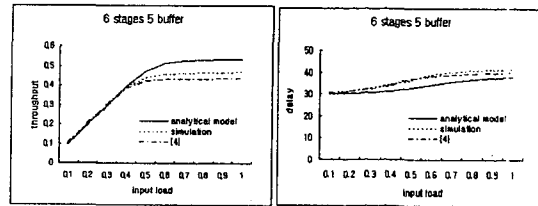
이 절에서는 높은 성능의 MIN 스위칭을 위해 제안된 모델이 평가된다. 제안된 모델과 시뮬레이션의 처리율 및 지연시간의 비교를 통해서 제안된 모델의 정확성을 검증하였다. 시뮬레이션에서 각 입력모듈은 네트워크 사이클마다 패킷을 입력 트래픽 로드(input traffic load)의 비율로 생성하며, 각 패킷의 목적지는 랜덤 생성기에 의해 랜덤하게 설정된다. 만약 SE안에서 셀들간에 경쟁이 있다면 경쟁에 참여하는 각 패킷이 이길 확률은 같으며 경쟁은 랜덤하게 수행된다.

다음에 제시되는 <그림 5>~<그림 8>은 트래픽 부하가 0.1에서 1까지 변화하며 제공되었을 때 3개의 셀을 저장할 수 있는 (3-버퍼)버퍼 모듈과 5개의 셀을 저장할 수 있는 (5-버퍼)버퍼 모듈을 가진 6단계(64×64)와 10단계(1024×1024) MIN의 처리율과 지연 시간을 제안된 분석 모델과 시뮬레이션의 결과를 비교 하였다. 여기서 각 분석 모델과 시뮬레이션의 데이터는 펜티엄 III 윈도우 2000 사용환경 상에서 MS Visual

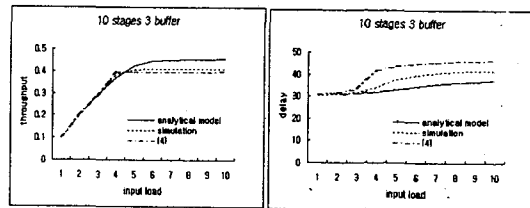
C++ 6.0 컴파일러를 사용해서 10⁶번 반복 수행하여 얻은 결과로 이전에 발표된 이 분야 논문의 시뮬레이션 프로그램과 같은 환경에서 실험했을 경우 같은 결과를 보여준다.



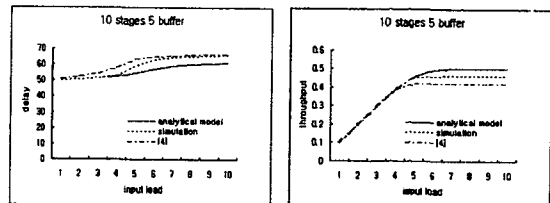
<그림 5> 균등 트래픽에서 6스테이지, 3버퍼를 가진 MIN의 처리율과 지연시간 비교



<그림 6> 균등 트래픽에서 6스테이지, 5버퍼를 가진 MIN의 처리율과 지연시간 비교



<그림 7> 균등 트래픽에서 10스테이지, 3버퍼를 가진 MIN의 처리율과 지연시간 비교



<그림 8> 균등 트래픽에서 10스테이지, 5버퍼를 가진 MIN의 처리율과 지연시간 비교

위의 <그림 5>~<그림 8>을 살펴보면 제안된 모델의 처리율은 시뮬레이션 결과보다 약간 큰 값에서, 지연시간은 작은 값으로 일정하게 수렴됨을 알 수 있다.

이 제안된 모델은 매개변수의 수와 상태 전이 다이어그램이 이전의 것보다 매우 단순하다. 이런 단순함에도 불구하고, 제안된 모델의 분석 결과는 시뮬레이션을 통해 얻은 처리율과 지연시간은 상당히 유사한 결과를 보여준다. 또한 단계와 버퍼의 수가 증가할수록 더욱더 시뮬레이션 결과에 유사한 것으로 나타났다. <그림 5>~<그림 8>에서 나타났듯이 단계나 버퍼가 증가함에 따라 더욱 정확한 값을 나타내는 것을 볼 수 있듯이 10 단계 5 버퍼에서 더욱더 시뮬레이션 결과와 유사한 정확한 값을 가지는 것을 볼 수 있다.

위에서 살펴보았듯이 제안된 모델은 일반적인 체계보다는 네트워크 사이클을 보다 효율적으로 사용하면서 기존에 제시된 모델들과 비교했을 때 보다 단순한 구조이면서도 정확한 값을 가지는 결과를 나타내는 것을 입증해 보였다. 더욱이 어떤 하드웨어나 동작의 오버헤드 없이 고안된 체계의 모델링 및 그 정확성은 더욱 큰 의미를 지닌다고 할 수 있다.

5. 결론

본 논문에서는 기존에 연구된 네트워크 사이클에서 단위 단계 버퍼링 체계의 접근방식을 따르는 입력 버퍼에 관한 분석 모델을 제안하였다. 기존에 연구된 네트워크 사이클의 경우는 셀이 가장 앞쪽의 버퍼 위치로 이동해야만 하기 때문에 네트워크 사이클이 셀 이동을 보장할 정도로 충분히 길어야 하는데 대부분의 버퍼 모델은 거의 꽉 차게 되어 대역폭의 낭비를 가져오게 되는 단점이 있다. 이런 단점을 보완한 단위 단계 버퍼링 체계를 이용할 뿐만 아니라 이전 모델의 방법이 매우 복잡하기에 다른 구조의 MIN에 적용하기가 어렵고 처리율과 지연시간에서 정밀하지 못한 결과를 나타낸 반면 이 제안

된 모델은 매개변수의 수와 상태 전이 다이어그램에 있어서 기존의 것보다 매우 단순함에도 불구하고 분석 결과가 시뮬레이션을 통해 얻은 처리율과 지연시간에 있어서 상당히 유사함을 보여주었다.

참고문헌

- [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] H. Rudin, "The ATM-Asynchronous Transfer Mode," Computer Networks and ISDN Systems, Vol.24, pp.277-278, 1992.
- [3] H.Y. Youn and C. Chevli, "A Local Blocking Scheme for Performance Enhancement of MINs," in 991 Int'l Symp. on Applied Computing, pp. 273-282, April 1991.
- [4] 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.
- [5] D.M. Dias and J.R. Jump, "Analysis and simulation of buffered delta networks," IEEE Trans. on Computers, vol. c-30, pp.273-282, Apr. 1981.
- [6] Y.C. Jenq, "Performance analysis of a packet switch based on single-buffered banyan network," IEEE J. Select. Areas Commun., vol. SAC-3, pp. 1014-1021, Dec. 1983.
- [7] H.S. Yoon, K.Y. Lee and M.T. Liu, "Performance analysis of multibuffered packet-switching networks in multi-processor system s," IEEE Trans. on Computers, vol. c-39, pp. 319-327, March 1990.

-
- [8] T.H. Theimer, E.P. Rathgeb and M.N. Huber, "Performance analysis of buffered banyan networks," IEEE Trans. on Commun. vol. c-39, pp. 269-277, Feb. 1991.
- [9] S.H. Hsiao and C.Y.R. Chen, "Performance analysis of single-buffered multistage interconnection networks," in Proc. Third IEEE Symp. on Parallel and Distributed Processing. pp. 864-867, Dec. 1991.
- [10] Y. Mun and H.Y. Youn, "Performance Analysis of Finite Buffered Multistage Interconnection Networks," IEEE Trans. on Computers, pp. 153-162, Feb. 1994.
- [11] J. Ding and L.N. Bhuyan, "Performance evaluation of multistage interconnection networks with finite buffers," in Proc. Int'l. Conf on Parallel Processing, pp.592-595 1991.
- [12] 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.
- [13] 최원제, 추현승, 문영성, "MIN기반 교환기 구조를 분석하기 위한 간단한 근사화 방법 연구", 정보처리 학회 논문지, 1999.
- [14] 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.
- [15] W. Choi, H. Choo, and S-M. Yoo, "Analytical Approaches for Modeling characterized MIN", ISCA 15th Int'l Conf. on Computers and Their Applications, pp.46-49, March 2000.

● 저자소개 ●



홍유치

2000년 관동대학교 컴퓨터공학과 졸업(학사)

2000년~현재 성균관대학교 정보통신공학부 석사과정

관심 분야: ATM 네트워크



추현승

1988년 성균관대학교 수학과 졸업(학사)

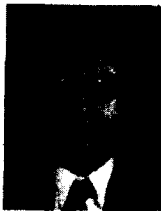
1990년 텍사스 주립대(달라스) 전자계산학과(공학석사)

1996년 텍사스 주립대(알링턴) 전산공학과(공학박사)

1997년 특허청 심사4국 컴퓨터심사 담당관실 심사관(사무관)

1998년~현재 성균관대학교 정보통신공학부 조교수

관심 분야: 병렬 및 분산처리, 스토리지 시스템, 이동 컴퓨팅 등



윤희복

1977년 서울대학교 전기공학과 졸업(학사)

1979년 서울대학교 전기공학과 대학원(공학석사)

1988년 Univ. of Massachusetts at Amherst, 컴퓨터공학과대학원 공학박사

1988년~1991년 Univ. of North Texas, 전산학과 조교수

1991년~2000년 Univ. of Texas at Arlington, 컴퓨터공학과 부교수

(1999년~2000년 한국정보통신대학원대학교 교수)

2000년~현재 성균관대학교 정보통신공학부 교수(특별 우대 교수)

관심 분야: 병렬 및 분산처리, 저장시스템, 이동네트워크 등