

# 버퍼를 장착한 스위치로 구성된 네트워크들의 성능분석

(Performance Evaluation of Networks with Buffered Switches)

신 태 지 <sup>†</sup>      남 창 우 <sup>†</sup>      양 명 국 <sup>\*\*</sup>  
 (Tae-zi Shin)    (Chang-woo Nam)    (Myung-kook Yang)

**요 약** 본 논문은 출력 버퍼를 장착한 크로스바 스위치로 구성된 다양한 네트워크들의 성능 예측 모델을 제안하고, 스위치에 장착된 버퍼의 개수 증가에 따른 성능 향상 추이를 분석하였다. 스위치 내부에 버퍼를 장착하는 기법은 네트워크 내부의 데이터 충돌 문제를 효과적으로 해결하고, 네트워크 성능 및 신뢰도를 높이는 방법으로 널리 알려져 있다. 또한, 크로스바 스위치를 이용하여 네트워크를 구성할 경우 네트워크 내부의 스위치들 간의 연결 형태 그리고, 각 스위치 내부의 데이터 이동 패턴에 따라 네트워크 특성이 결정된다. 본 논문에서는 크로스바 스위치로 구성된 세 가지 서로 다른 형태의 네트워크 : 다단 연결망(MIN), Fat-tree 망, 그리고 일반 통신망 등의 성능 분석모형을 제안하였다.

제안한 분석 모형은 네트워크 내부 스위치에 장착된 버퍼의 개수와 무관하게 네트워크 성능 평가의 두 가지 주요 요소인 네트워크 정상상태 처리율(Normalized Throughput, NT)과 네트워크 지연시간을 예측한다. 제안한 수학적 성능 분석 연구의 실효성을 검증하기 위하여 병행된 시뮬레이션 결과는 상호 미세한 오차 범위 내에서 모형의 예측 데이터와 일치하는 결과를 보여 분석 모형의 타당성을 입증하였다. 또한 분석 결과 네트워크 내부 스위치에 많은 버퍼를 장착 할수록 상대적으로 정상상태 처리율의 증가율은 감소하고, 네트워크 지연시간은 증가하는 것으로 나타났다.

**키워드** : 크로스바 스위치, 출력 버퍼, 정상상태 처리율, 네트워크 지연시간, 성능분석

**Abstract** In this paper, a performance evaluation model of Networks with the multiple-buffered crossbar switches is proposed and examined. Buffered switch technique is well known to solve the data collision problem of the switch networks. The characteristic of a network with crossbar switches is determined by both the connection pattern of the switches and the limitation of data flow in a each switch. In this thesis, the evaluation models of three different networks : Multistage interconnection network, Fat-tree network, and other ordinary communication network are developed. The proposed evaluation model is developed by investigating the transfer patterns of data packets in a switch with output-buffers. Two important parameters of the network performance, throughput and delay, are evaluated.

The proposed model takes simple and primitive switch networks, i.e., no flow control and drop packet, to demonstrate analysis procedures clearly. It, however, can not only be applied to any other complicate modern switch networks that have intelligent flow control but also estimate the performance of any size networks with multiple-buffered switches. To validate the proposed analysis model, the simulation is carried out on the various sizes of networks that uses the multiple buffered crossbar switches. It is shown that both the analysis and the simulation results match closely. It is also observed that the increasing rate of Normalized Throughput is reduced and the Network Delay is getting bigger as the buffer size increased.

**Key words** : crossbar switch, buffer, throughput, delay, analysis, simulation

<sup>†</sup> 정 회 원 : 울산대학교 전기전자정보시스템공학부 교수  
 newtaiji@mail.ulsan.ac.kr  
 cwnam@ulsan.ac.kr

<sup>\*\*</sup> 종신회원 : 울산대학교 전기전자정보시스템공학부 교수  
 mkyang@ulsan.ac.kr

논문접수 : 2006년 7월 6일  
 심사완료 : 2007년 1월 15일

## 1. 서 론

크로스바 스위치는 넓은 bandwidth, 우수한 상호 연결 기능 등의 장점으로 인해 다양한 병렬 컴퓨터의 상호 연결망으로 널리 사용되고 있으며, 최근 컴퓨터 통신

기술의 발전과 함께 각종 네트워크 스위칭 소자로 활용되고 있다[1].

크로스바 스위치를 이용하여 네트워크를 구성할 경우, 스위치 소자를 네트워크 특성에 맞게 나열하고 이를 상호 연결하여 전체 네트워크를 형성하게 된다. 이때 네트워크의 특성에 따라 스위치의 크기나 형태, 내부 이동 패턴이 결정된다.

이와 같이 크로스바 스위치로 구성된 네트워크의 경우 네트워크 내부 트래픽이 증가하거나 hot-spot이 발생하면, 데이터 충돌 현상이 불가피하게 발생한다. 데이터 충돌 현상은 스위치의 성능 저하를 유발함은 물론이고 네트워크 전체의 신뢰도에도 큰 영향을 미치게 된다. 이를 해결하기 위한방법으로 네트워크 내부 스위치의 모양을 변형[2-4]하거나, 데이터 충돌 시 우회경로를 설정하는 방법[5], 스위치 소자에 버퍼[6-12]를 장착하는 방법 등이 다각적으로 연구되어 왔다. 이들 가운데 스위치 소자에 버퍼를 장착하는 기법은 각 스위칭 소자 터미널에 버퍼를 장착하여 데이터 충돌로 인하여 소실될 데이터 패킷을 버퍼의 여유 공간에 저장함으로써, 네트워크 내부의 데이터 충돌 문제를 효과적으로 해결하고 네트워크의 성능을 향상시키는 방법으로 널리 알려져 있다.

본 논문에서는, 데이터 충돌 문제를 해결하기 위한 방법으로 제안된 buffered 스위치의 성능 분석 모형을 제안하고, 스위치 크기를 일반화하여 스위치에 장착된 버퍼의 개수 혹은 스위치의 크기 등에 제약받지 않고 성능 분석을 할 수 있도록 설계되었다. 또한 스위치의 변형을 고려하여 3가지의 서로 다른 형태의 스위치들로 구성된 네트워크들의 성능 분석 모형을 제안하고, 이를 검증하였다.

먼저, 다단 연결 망(Multistage Interconnection Network : MIN)의 경우 작은 스위치 소자를 단계(stage) 별로 나열하고 이를 주어진 패턴으로 연결하여 스위치 소자간의 연결 루트를 형성하고, 네트워크의 크기에 따라 스테이지 수를 조정하여 전체 상호 연결 네트워크를 형성한다. 이 경우 네트워크 내부 스위치들은 각각 입력 단과 출력 단이 서로 분리되고 같은 크기를 갖는  $a \times a$  형태의 스위치들로 구성된다.

Fat-tree[13] 네트워크의 경우 네트워크의 특성 상 다수의 루트 노드가 존재하는 tree의 구조를 가지므로 child 포트와 parent 포트의 개수가 서로 다르고 입출력 포트가 명확하게 구분되지 않는다. 즉 child 포트를 통해 유입된 데이터 패킷은 parent 포트뿐만 아니라 자신을 제외한 다른 child 포트를 지향 할 수 있다. 따라서, Fat-tree 네트워크의 경우 입력 단과 출력 단이 서로 다른 크기를 갖는  $a \times b$  형태의 스위치들로 구성

된다.

마지막으로 일반 통신 네트워크의 중간 노드로 사용되는 스위치 라우터[14-16]의 경우 입력 단과 출력 단의 구분이 없이  $n$ 개의 입출력 포트를 가지며 입력된 데이터 패킷은 자신을 제외한 다른 포트를 통해 출력 가능하다.

이들 네트워크들은 크로스바 스위치들로 구성되지만, 전체 네트워크 구조상 각각 스위치의 내부 데이터 이동 패턴이 서로 상이하다.

본 논문에서 제안된 성능분석 모형은 스위치에 장착된 버퍼의 개수 혹은 스위치 크기와 변형된 형태 등에 제약받지 않고 성능을 분석할 수 있도록 설계되었다. 또한, 분석 과정에서 수식 이해를 돕기 위하여 간단한 데이터 충돌 처리 기법 및 steady state probability 개념을 도입하였다. 제안한 수학적 성능 분석 연구의 실효성 검증을 위하여 병행된 시뮬레이션 결과는 상호 미세한 오차 범위 내에서 모형의 예측 데이터와 일치하는 결과를 보였다.

본 논문의 구성은 다음과 같다. 먼저 2절에서는 다양한 네트워크에 사용되는 스위치들의 여러 가지 구조를 설명하고, 각 스위치 내부의 데이터 이동 패턴을 분석하고, 이를 이용한 네트워크의 성능분석 모형을 제시하고, 3절에서는 성능 분석 모형과 성능평가를 하였으며, 끝으로 본 연구의 성과와 결과를 요약·기술하였다.

## 2. 네트워크의 성능분석

본 절에서는 네트워크의 스위치에 장착된 buffer가 네트워크 성능에 미치는 영향을 분석 예측할 수 있는 수학적 분석 모형을 기술하였다. 먼저 분석 모형 개발에 적용된 네트워크 환경에 대한 일반적인 가정을 정리하고, 각 buffered 스위치 내부의 데이터 이동 패턴을 확률적으로 해석하였다. 이를 기반으로 하여 각 스위치 내부의 데이터 이동 확률을 토대로 buffered 스위치를 장착한 다단 연결 망 성능 평가의 주요 요소로 거론되어지는 네트워크 정상상태 처리율(NT : Normalized Throughput)과 네트워크 지연시간(Network Delay)에 대한 성능 분석 기법을 기술하였다.

### 2.1 네트워크 환경에 대한 일반적인 가정

복수 buffered 다단 연결 망의 분석 모형 개발과 시뮬레이션을 위해 본 논문에 적용된 일반적인 가정을 정리하면 다음과 같다.

- multiple-buffered crossbar 스위치들로 구성된 네트워크를 분석대상으로 한다.
- 네트워크는 스위치 클럭 싸이클,  $\Delta t$ ,에 따라 동기적으로 작동한다. 즉, 네트워크 내부 데이터 패킷은 스위치 클럭 동안 임의의 스위치 클럭 단을 출발, 다음 스테

이지 스위치를 통과하여 해당 출력 단에 도달한다.

- 스위치에 장착된 buffer는 스위치 출력 단에 위치하고, buffer 공간 하나는 한 개의 데이터 패킷을 수용할 수 있다.
- 데이터 패킷은 네트워크 입력 단의 각 source 노드에서 같은 확률로 발생한다. 네트워크 내부 임의의 스테이지  $i$ 에 위치한 스위치 입력 단으로 데이터가 유입될 확률은  $\zeta_i$ 라 한다. 따라서 매 사이클마다 네트워크 각 입력 단에 한 개씩의 데이터 패킷이 유입될 경우,  $\zeta_i$ 는 1이 된다.
- 네트워크 입력 단으로 유입되는 데이터 패킷의 네트워크 최종 출력 단 행선지는 무작위 선택 방식에 의거 결정된다.
- 데이터 충돌 발생 시 무작위 중재 방식에 의거 데이터 처리 우선 순위를 결정한다.

**2.2 스위치 형태에 따른 스위치 내부에서의 데이터 이동 패턴**

네트워크에 사용되는 크로스바 스위치는 네트워크의 특성에 따라 3가지 형태로 변형 되어 사용 되어 질수 있다.

첫 번째는 다단 연결망과 같이 입력 단과 출력 단이 서로 분리되고 같은 크기를 갖는  $a \times a$  형태의 스위치들로 구성된다.

Fat-tree 네트워크의 경우 입력 단과 출력 단이 엄격한 구분이 없고 서로 다른 크기를 갖는  $a \times b$  형태의 스위치들로 구성된다.

일반 통신 네트워크의 중간 노드로 사용되는 스위치 라우터의 경우 입력 단과 출력 단의 구분이 없이  $n$ 개의 입출력 포트를 가지며 입력된 데이터 패킷은 자신을 제외한 다른 포트를 통해 출력 가능하다.

**2.2.1 다단 연결 망의  $a \times a$  형태의 스위치**

네트워크 내부 임의의  $a \times a$  crossbar 스위치 입력 단에 유입된 데이터 패킷은 데이터가 지향하는 행선지에 따라 스위치의  $a$ 개 출력 단 중 어느 한 출력 단으로 향하

게 된다. 2.1.의 가정에 의거, 네트워크 입력 단에 처음 데이터 패킷이 유입될 때 최종 출력 단 행선지가 무작위 선택 방식에 의거 주어짐으로, 네트워크 스테이지  $i$ 에 위치한 임의의 스위치 입력 단에 데이터 패킷이 유입될 확률이  $\zeta_{stage\ i}$ 로 주어지면 해당 스위치의 어느 한 출력 단으로 데이터 패킷이 향할 확률은  $\frac{\zeta_{stage\ i}}{a}$ 가 된다. 그림 1은 네트워크 내부 임의의 스위치에서 데이터 패킷이 이동하는 패턴을 확률적으로 해석하여 도식화한 것이다.

네트워크 내부 스테이지  $i$ 에 위치한 임의의 스위치 입력 단  $I_0$ 에 데이터 패킷이 유입될 확률이  $\zeta_{stage\ i}$ 로 주어지면, 그로 인하여 해당 스위치의 특정 출력 단  $D_0$ 로 데이터 패킷이 향할 확률은  $\frac{\zeta_{stage\ i}}{a}$ 가 되고, 특정 출력 단  $D_0$ 로 데이터 패킷이 향하지 않을 확률은  $(1 - \frac{\zeta_{stage\ i}}{a})$ 가 된다. 따라서, 스테이지  $i$ 에 위치한 임의의 스위치 출력 단  $D_0$ 로  $r$ 개의 데이터 패킷이 지향할 확률,  $P(h=r)_i$ ,은

$$P(h=r)_i = \binom{a}{r} \times \left(\frac{\zeta_{stage\ i}}{a}\right)^r \times \left(1 - \frac{\zeta_{stage\ i}}{a}\right)^{a-r} \quad (1)$$

로 계산된다. 여기서  $0 \leq r \leq a$  이다.

**2.2.2 Fat tree 네트워크의  $a \times b$  형태의 스위치**

Fat-tree 네트워크는 복수 루트 노드를 가지는 트리 형태를 띄며, 각각의 스위치 노드는 하나 이상의 상위 노드들에 연결되어 있다. 따라서 임의의 말단 노드로부터 루트 노드에 이르기까지 단일 경로만을 제공하는 일반적인 트리 네트워크와 달리, 상위 노드로 접근 과정에 복수 경로를 제공하여 네트워크의 대역폭을 증가시킴과 동시에 임의의 스위치 노드 및 링크 고장 시, 우회경로를 제공함으로써 네트워크의 가용성을 향상시켰다.

Fat-tree 네트워크는 스위치 노드의 크기와 말단 노드의 수에 따라 다양한 구성을 가진다. 그림 2의 (a),

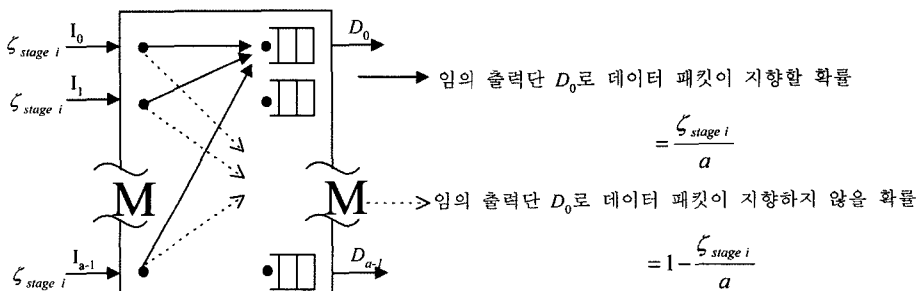


그림 1 네트워크 내부 임의의 스위치에서 데이터 이동 패턴

(b)는 27개의 말단 노드를 가지는 두 가지 형태의 일반적인 fat-tree 네트워크의 구성을 보여주고 있다. 그림 2의 (a)는 네트워크의 높이는 3이고 각 스위치 노드는 3개의 child 포트와 3개의 parent 포트를 가지고, 그림 2의 (b)는 네트워크 높이는 3이고 각 스위치 노드는 3개의 child 포트와 2개의 parent 포트를 가지고 있다. 일반적으로  $a \times b$  스위치로 구성된 높이가  $h$ 인 fat-tree 네트워크는  $FT(h, a, b)$ 로 표현한다. Fat-tree 네트워크  $FT(h, a, b)$ 는 레벨 0에서  $a^h$ 개의 말단 노드(processor)와 상위 각 레벨  $l$ 에서  $a^{h-l}b^{l-1}$ 개의 스위치 노드들로 구성된다. 이때,  $b=1$ 이면  $a$ -ary 트리가 되고,  $a=b$ 인 경우 BMIN(Bidirectional MIN)과 같은 구조를 가지게 된다.

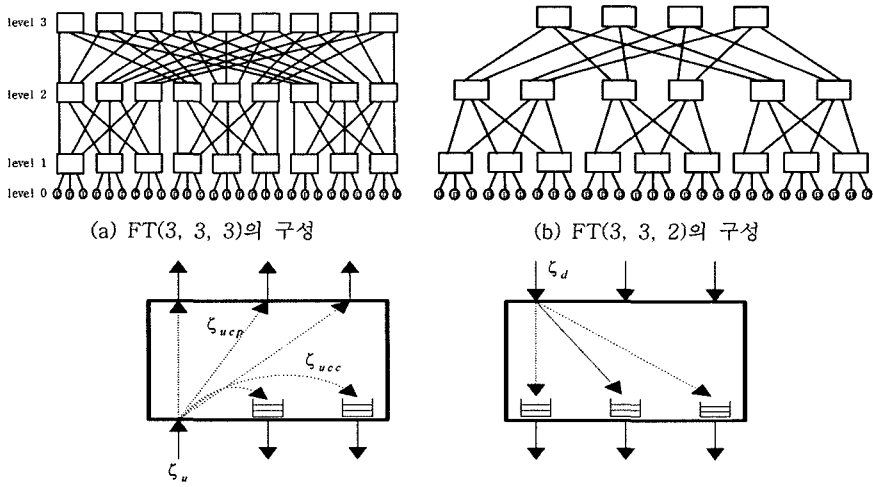
Fat tree를 구성하는 각 스위치에서 데이터의 라우팅 형태는 그림 2의 (c)와 같이 데이터 패킷이 스위치의 어느 포트에 유입되어 어느 포트에 출력되는가에 따라 세 가지 형태로 나타난다. 첫 번째는 스위치의 parent 포트에 유입되어  $a$ 개의 child 포트에 다운 라우팅하는 경우이고, 두 번째는 스위치의 child 포트에 유입되어  $b$ 개의 parent 포트에 업 라우팅하는 경우, 세 번째는  $a-1$ 개의 다른 child 포트에 회귀 라우팅을 하는 경우이다. 레벨 0의 processor  $P(0, i)$ 에서 생성된 데이터 패킷은 목적지  $P(0, d)$ 를 포함하는 최소 sub-fat 트리의 루트까지, parent 포트에 업 라우팅한다. 데이터 패킷이 업 라우팅하는 경우 fat tree 네트워크가 다중 루트를 가짐으로, 스위치의 parent 포트  $b$ 개 중 어느 포트에 출력되어도 원하는 목적지에 도달할 수 있다. 따라서 임의의 선택된 parent 포트가 이미 다른 데이터 패킷에 의해 사용 중

이라면 다른 유용한 포트를 통해 출력된다. 그러나,  $b$ 개의 모든 포트가 다른 데이터 패킷들에 의해 이미 사용 중이라면 해당 데이터 패킷은 탈락하게 된다.

데이터 패킷이 목적지  $P(0, d)$ 를 포함하는 sub-root 임의의 스위치 노드  $S(l, x)$ 에 도달하면, 여기서  $l = \lfloor \frac{i}{a^l} \rfloor = \lfloor \frac{d}{a^l} \rfloor$ 을 만족하는 최소 레벨, 스위치 내에서  $a-1$ 개의 child 포트에 회귀하여 다운 라우팅을 시작하게 된다. 데이터 패킷이 회귀하거나 다운 라우팅을 하는 경우는 업 라우팅과는 달리 데이터 패킷의 경로가 하나로 결정되게 된다. 따라서, 데이터 패킷들 간에 충돌현상이 발생하면 일부 해당 데이터 패킷이 다른 경로를 택할 수 없으므로 탈락되게 된다. 여기서, 일시적 Hot spot 등의 현상으로 야기되는 데이터 충돌과 이로 인한 데이터 손실을 방지하기 위하여 네트워크 출력 단에 버퍼를 장착하게 된다.

$FT(h, a, b)$  레벨  $l$ 의 임의의 스위치  $S(l, x)$ 의 child 포트에 유입된 데이터 패킷은 데이터가 지향하는 최종 행선지에 따라 업 라우팅, 회귀 라우팅, 혹은 다운 라우팅을 하게된다. 먼저 스위치 임의의 child 포트에 유입된 데이터 패킷은 자신을 제외한  $(b+a-1)$ 개의 출력 단 중 어느 한 출력 단으로 진행 가능하고, parent 포트에 유입된 데이터 패킷은  $a$ 개의 child 포트들 중 어느 한 출력 단으로 향하게 된다.

2.1절의 가정에 의거, 네트워크 입력 단에 처음 데이터 패킷이 유입될 때 최종 출력 단 행선지가 무작위 선택 방식에 의해 주어짐으로, 임의의 스위치 입력 단에 도착한 데이터 패킷이 어느 출력 단으로 향하게 되는가는



(c) 스위치 내에서의 데이터 이동 패턴  
 그림 2 Fat-tree 구조와 스위치 내부에서 데이터 이동

다음과 같은 데이터 진행 확률 분석으로 수식화할 수 있다.

먼저 레벨  $l$ 에 위치한 임의 스위치의 child 포트 입력 단으로 데이터 패킷이 유입될 확률이  $\zeta_{u, level\ l}$ 로 주어지면, 데이터 패킷이 스위치 child 포트의 출력 단으로 지향할 확률은 다음과 같이 계산할 수 있다.

Fat-tree 네트워크 구조상  $FT(h, a, b)$ 의 임의의 레벨  $l$ 에 위치한 스위치 노드는  $a$ 개의 child 포트를 통하여 스위치 노드 또는 말단 노드와 연결된다. 따라서 계귀적 연결 구조를 갖는 fat-tree  $FT(l, a, b)$ 는  $a^l$ 개의 말단 노드를 포함한다. 또한 최상위 스위치 노드는  $FT(h, a, b)$ 의 루트 노드로 전체 말단 노드  $a^h$ 개의 말단 노드를 포함하게 된다. 따라서, 레벨  $l$ 의 임의 스위치에서 데이터 패킷이 회귀 라우팅할 경우는 현 스위치 노드가 child 포트에 유입된 데이터 패킷의 최종 행선지를 포함하는 sub fat-tree의 루트 노드가 되는 경우이다. 이 경우 데이터 패킷은 자신이 유입된 child 포트에 회귀하는 경우가 없으므로, 해당 child 포트와 연결된 레벨  $(l-1)$ 의 스위치 노드가 루트가 되는 sub fat-tree  $FT(l-1, a, b)$ 에 포함된  $a^{l-1}$ 개의 말단노드는 경로 설정에서 제외된다. 따라서, 임의 레벨  $l$ 의 임의 스위치 노드의 child 포트 입력 단으로 유입된 데이터 패킷이 유입된 child 포트를 제외한 임의 child 포트 출력 단으로 지향할 확률 즉, 회귀 라우팅할 확률,  $\zeta_{uc, level\ l}$ ,은 다음과 같이 구할 수 있다.

$$\zeta_{uc, level\ l} = \zeta_{u, level\ l} \times \frac{a^l - a^{l-1}}{a^h - a^{l-1}} = \zeta_{u, level\ l} \times \frac{a-1}{a^{h-l+1} - 1} \quad (2)$$

식 (2)에서 분모항  $(a^h - a^{l-1})$ 은 해당 스위치 child 포트 입력 단으로 유입된 데이터 패킷이 도달할 수 있는 모든 말단 노드의 수이고, 분자항  $(a^l - a^{l-1})$ 은 해당 스위치의  $(a-1)$ 개의 child 포트를 통하여 도달할 수 있는 말단 노드의 수를 나타낸다. 따라서 식 (2)는 데이터 패킷의 행선지가 무작위 방식으로 주어지는 가정 하에, 레벨  $l$ 의 임의 스위치 child 포트에 도달한 데이터 패킷의 최종 행선지가 해당 스위치를 sub-root로 하는 말단 노드에 존재할 확률을 나타낸다. 즉, 식 (2)는 해당 데이터 패킷의 최종 행선지가 해당 스위치내에서 회귀 라우팅하여야만 도달할 수 있는  $(a^l - a^{l-1})$ 개의 말단 노드들 중에 존재할 확률을 수식화한 것이다.

유사한 방법으로  $FT(h, a, b)$  레벨  $l$ 의 임의 스위치  $S(l, x)$ 의 child 포트 입력 단으로 유입된 데이터 패킷이 parent 포트 출력 단을 지향하게 될 확률,  $\zeta_{cp, level\ l}$ ,은

$$\zeta_{cp, level\ l} = \zeta_{u, level\ l} \times \frac{a^h - a^l}{a^h - a^{l-1}} \quad (3)$$

와 같이 나타낼 수 있다. 여기서,  $l < h$ 이다. 식 (3)에서 분자항  $(a^h - a^l)$ 은 해당 스위치 child 포트 입력 단으로 유입된 데이터 패킷이 parent 포트를 통해야만 도달할 수 있는 모든 말단 노드의 수를 나타낸다. 식 (3)는 해당 데이터 패킷의 최종 행선지가 상위 레벨의 스위치로 이동하여야만 도달할 수 있는  $(a^h - a^l)$ 개의 말단 노드에 존재할 확률을 수식화한 것이다.

데이터 패킷이 스위치의 parent 포트를 지향할 경우, parent 포트  $b$ 개 중 임의의 포트를 선택해서 출력하게 되고, 만약 선택된 포트에서 충돌이 발생할 경우, 다른 가능한 parent 포트를 선택해서 출력하게 된다. 일반적으로, parent 포트를 통해 데이터 패킷이 출력될 경우, 다중 경로의 선정이 가능하므로 parent 포트에는 버퍼를 장착하지 않게 된다. 따라서, 만일 모든 parent 포트가 사용 중이라면, 해당 데이터 패킷은 제거될 수 있다. 네트워크 구조상 parent 포트를 통과하여 나온 데이터 패킷들은 다음 레벨의 스위치  $S(l+1, y)$ 의 child 포트 입력으로 유입되게 된다. 따라서, 다음 레벨  $l+1$ 의 임의 스위치  $S(l+1, y)$ 의 child 포트에 데이터 패킷이 유입될 확률,  $\zeta_{u, level\ l+1}$ ,은

$$\zeta_{u, level\ l+1} = \sum_{r=1}^a \left\{ a C_r (\zeta_{ucp, level\ l})^r (1 - \zeta_{ucp, level\ l})^{a-r} \times \left(\frac{r}{b}\right)^r \right\} \quad (4)$$

로 계산된다. 식 (4)은 먼저 레벨  $l$ 의 임의 스위치 child 포트에 도달한 데이터 패킷들 중  $r$ 개의 데이터 패킷이 parent 출력 포트에 지향할 확률을 구하고, 이를 토대로 해당 스위치의 임의 parent 출력 포트에 데이터 패킷이 출력될 확률을 계산하고 있다. 식 (4)의 마지막 곱셈항  $\left(\frac{r}{b}\right)^r$ 는 parent 출력 포트에 지향하는  $r$ 개 데이터 패킷 중 어느 하나가 임의 parent 출력 포트를 통하여 출력될 가능성을 나타내고 있다. 여기서  $r \geq b$ 이면 모든 parent 출력 포트를 통하여 데이터가 출력되어  $\left(\frac{r}{b}\right)^r$ 는 1로 계산된다.

따라서, 네트워크 입력 단으로 데이터 패킷이 유입될 확률,  $\zeta_{u, level\ 1}$ ,이 주어지면 레벨 1에서부터 루트 노드까지 식 (2), (3), (4)을 레벨별로 반복 계산하여 각 레벨의 스위치 노드에 데이터 패킷이 유입될 확률, 업 라우팅할 확률, 그리고 회귀 라우팅할 확률 등을 계산할 수 있다.

일단 루트 스위치에 도달한 데이터 패킷들은 모두 회귀하여 다운 라우팅을 시작하게 된다. 이 경우 임의 루트 스위치의 child 포트 출력 단으로  $r$ 개의 데이터 패킷이 지향할 확률,  $P(\tilde{h}_c = r)_h$ ,을 계산하면 다음과 같이 주어진다.

$$P(h_c = r)_h = {}_{a-1}C_r \times \left(\frac{\zeta_{succ, level h}}{a-1}\right)^r \times \left(1 - \frac{\zeta_{succ, level h}}{a-1}\right)^{a-r-1} \quad (5)$$

루트 스위치의 임의 child 포트에 도착한 데이터 패킷은 자신이 유입된 child 포트를 제외한 나머지 (a-1)개의 다른 child 포트에 지향하게 된다. 따라서 해당 스위치의 특정 child 포트 출력 단으로 데이터 패킷이 지향할 확률은  $\left(\frac{\zeta_{succ, level h}}{a-1}\right)$ 가 되고, 지향하지 않을 확률은  $\left(1 - \frac{\zeta_{succ, level h}}{a-1}\right)$ 이 된다. 이를 이용하여 루트 스위치의 임의 child 포트 출력 단을 지향하는 데이터 패킷의 수가 r개일 확률은 식 (4)와 같이 나타낼 수 있다. 식 (5)의 확률과 같이 루트 노드에 도착하여 다운 라우팅 되는 데이터 패킷의 경우 child 포트에서 충돌 발생 시, 다중 경로를 설정할 수 없으므로 데이터 패킷의 유실을 막기 위해 child 포트에 버퍼를 장착하게 된다. 따라서, 버퍼의 유효 공간이 있는 한 데이터 패킷의 유실은 일어나지 않게 된다. 그러므로, 식 (5)의 확률로 루트 노드의 스위치에서 child 포트에 회귀한 데이터 패킷은 버퍼에 저장되었다가 출력되거나 또는 바로 출력되어 (h-1)레벨의 임의 스위치의 parent 포트에 유입되어 다운 라우팅을 시작하여, 다음 레벨의 다운 라우팅 확률,  $\zeta_{d, level h-1}$ ,로 주어진다.

같은 방법으로, 임의 레벨 l의 스위치 S(l, x)의 임의 child 포트 출력 단으로 데이터 패킷이 지향할 경우는 먼저 해당 스위치 상위 레벨로부터 데이터가 다운 라우팅 하여 parent 포트 입력 단으로 입력된 경우와 해당 스위치 child 포트 입력 단에 도착된 데이터 패킷이 회귀 라우팅을 시작한 경우이다. 따라서, a개의 child 포트와 b개의 parent 포트를 가지는 스위치에서 임의의 child 포트 출력 단으로 r개의 데이터 패킷이 지향할 확률,  $P(h_c = r)_l$ ,은 데이터 패킷이 스위치 내부에서 회귀할 확률,  $\zeta_{succ, level l}$ ,과 parent 포트로부터 다운 라우팅할 확률,  $\zeta_{d, level l}$ ,을 이용하여 다음과 같이 나타낼 수 있다.

$$P(h_c = r)_l = \sum_{\omega=0}^r \left\{ {}_{a-1}C_{\omega} \times \left(\frac{\zeta_{succ, level l}}{a-1}\right)^{\omega} \times \left(1 - \frac{\zeta_{succ, level l}}{a-1}\right)^{a-\omega-1} \times {}_bC_{r-\omega} \times \left(\frac{\zeta_{d, level l}}{a}\right)^{r-\omega} \times \left(1 - \frac{\zeta_{d, level l}}{a}\right)^{b-r+\omega} \right\} \quad (6)$$

데이터 패킷이 스위치 내부에서 회귀하는 경우; 식 (6)과 같이 자신을 제외한 (a-1)개의 child 포트 중 하나를 지향하게 되고, parent 포트 입력 단으로 유입된 데이터 패킷의 경우; 최종 행선지에 따라 a개의 child 포트 중 하나를 지향하게 된다. 식 (6)은 임의 child 포트 출력 단으로 회귀 라우팅 하는 데이터 패킷과 다운

라우팅하는 데이터 패킷의 수를 합해서 모두 r개 데이터 패킷이 지향할 확률을 수식화 한 것이다.

식 (6)에서 상위 레벨로부터 데이터 패킷이 다운 라우팅하여, 임의 레벨 l에 위치한 스위치 노드의 parent 포트 입력 단으로 유입될 확률로 정의된,  $\zeta_{d, level l}$ 은 다음과 같이 계산될 수 있다. Fat-tree 네트워크의 구조상 임의 레벨 l에 위치한 스위치의 child 포트 출력 단은 하위 레벨의 임의 스위치의 parent 포트 입력 단으로 연결된다. 따라서, 상위 레벨 스위치의 child 포트에 데이터 패킷이 출력 확률,  $P(D_c = 1)_l$ ,은 현재 레벨 스위치의 parent 포트에 데이터 패킷이 유입될 확률,  $\zeta_{d, level l-1}$ ,이 된다. 즉,  $P(D_c = 1)_{level l} \equiv \zeta_{d, level l-1}$ 이다. 여기서 임의 사이클 (j-1)에 레벨 l에 있는 스위치의 임의 child 포트 출력 단,  $D_c$ ,로 데이터 패킷이 출력될 확률,  $P(D_c = 1)_{l, cycle j}$ ,을 살펴보면, 해당 출력 단 버퍼가 사이클 (j-1)에 데이터 패킷을 저장하고 있는 경우, 혹은 사이클 (j-1)에 스위치의 child 포트 입력 단으로 새로 유입된 데이터 패킷이 해당 출력 단으로 지향할 경우이다. 반대로 스위치 출력 단으로 데이터 패킷이 출력되지 않는 경우는 사이클 (j-1)에 해당 출력 단 버퍼가 데이터 패킷을 저장하지 않은 상태에서, 사이클 j에 해당 출력 단으로 지향하는 데이터 패킷이 없을 경우이다. 이를 수식화하여 레벨 l에 위치한 임의 스위치 child 포트에 데이터 패킷이 출력될 확률,  $P(D_c = 1)_{l, cycle j}$ , 즉 레벨 (l-1)의 임의 스위치 parent 포트 입력 단으로 데이터 패킷이 유입될 확률,  $\zeta_{d, level l-1}$ , 구하면

$$\zeta_{d, level l-1} = P(D_c = 1)_{l, cycle j} = 1 - P(h=0)_{l, cycle j} \times P(\epsilon=0)_{l, cycle (j-1)} \quad (7)$$

와 같이 계산된다. 여기서,  $P(h=0)_{l, cycle j}$ 는 식 (5)를 통하여 확률적으로 계산할 수 있으며,  $P(\epsilon=0)_{l, cycle (j-1)}$ 는 다음의 2.3절의 확률적 계산 방법을 통하여 얻게 된다.

2.2.3 n 개의 포트를 갖는 스위치

네트워크 내부 임의 스위치의 한 포트에 유입된 데이터 패킷은 데이터 패킷이 지향하는 행선지에 따라 자신을 제외한 다른 포트에 지향하게 된다. 네트워크 입력 단에 처음 데이터 패킷이 유입될 때 최종 행선지가 무작위 선택 방식에 의해 주어짐으로, n개의 포트를 갖는 스위치의 경우 자신을 제외한 n-1개의 포트를 지향하게 되므로 스위치 임의 포트 D<sub>0</sub>의 입력 포트에 데이터 패킷이 유입될 확률이  $\zeta_{D_0}$ 로 주어지면, 해당 스위치의 특정 포트D의 출력 단으로 데이터 패킷이 지향할 확률은  $\left(\frac{\zeta_{D_0}}{(n-1)}\right)$ 이 되고, 지향하지 않을 확률은  $\left(1 - \frac{\zeta_{D_0}}{(n-1)}\right)$ 가 된다.

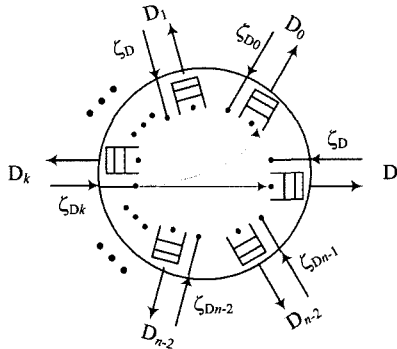


그림 3 n개의 포트를 가지는 스위치 라우터 내부의 데이터 이동 패턴

→  $D_k$ 로 유입된 데이터 패킷이 포트 D를 지향할 확률  $\frac{\zeta_{Dk}}{3}$

→  $D_0$ 로 유입된 데이터 패킷이 포트 D를 지향하지 않을 확률  $1 - \frac{\zeta_{Dk}}{3}$

그림 3과 같은 n개의 포트를 가진 스위치의 경우, 스위치의 특정 포트 D의 출력 포트로 0개의 데이터 패킷이 지향할 확률,  $P(h_D=0)_{cycle j}$ ,은 다음과 같이 나타낼 수 있다.

$$P(h_D=0)_{cycle j} = \left(1 - \frac{\zeta_{D_0}}{n-1}\right) \cdot \left(1 - \frac{\zeta_{D_1}}{n-1}\right) \cdot \dots \cdot \left(1 - \frac{\zeta_{D_{n-1}}}{n-1}\right) \quad (8)$$

위 식은 자기 자신을 제외한 다른 포트로 유입된 데이터 패킷이 모두다 포트 D를 제외한 다른 포트를 지향함을 수식화 한 것이다. 따라서, 포트 D를 지향하는 데이터 패킷이 한 개일 확률,  $P(h_D=1)_{cycle j}$ ,은 포트 D를 제외한 임의의 포트로 유입된 데이터 패킷이 포트 D로 지향할 경우 나머지 다른 두개의 포트로 유입된 데이터 패킷은 포트 D를 출력 지향하지 않음을 수식화하여 다음과 같이 나타낼 수 있다.

$$P(h_D=1)_{cycle j} = \left(\frac{\zeta_{D_0}}{n-1}\right) \cdot \left(1 - \frac{\zeta_{D_1}}{n-1}\right) \cdot \left(1 - \frac{\zeta_{D_2}}{n-1}\right) \cdot \dots \cdot \left(1 - \frac{\zeta_{D_{n-1}}}{n-1}\right) + \left(1 - \frac{\zeta_{D_0}}{n-1}\right) \cdot \left(\frac{\zeta_{D_1}}{n-1}\right) \cdot \left(1 - \frac{\zeta_{D_2}}{n-1}\right) \cdot \dots \cdot \left(1 - \frac{\zeta_{D_{n-1}}}{n-1}\right) + \dots + \left(1 - \frac{\zeta_{D_0}}{n-1}\right) \cdot \left(1 - \frac{\zeta_{D_1}}{n-1}\right) \cdot \left(1 - \frac{\zeta_{D_2}}{n-1}\right) \cdot \dots \cdot \left(\frac{\zeta_{D_{n-1}}}{n-1}\right) \quad (9)$$

같은 방법으로 스위치 임의의 포트 D를 지향하는 데이터 패킷의 개수가 2개일 확률,  $P(h_D=2)_{cycle j}$ ,는

$$P(h_D=2)_{cycle j} = \left(\frac{\zeta_{D_0}}{n-1}\right) \cdot \left(\frac{\zeta_{D_1}}{n-1}\right) \cdot \left(1 - \frac{\zeta_{D_2}}{n-1}\right) \cdot \dots \cdot \left(1 - \frac{\zeta_{D_{n-2}}}{n-1}\right) \cdot \left(1 - \frac{\zeta_{D_{n-1}}}{n-1}\right) + \left(\frac{\zeta_{D_0}}{n-1}\right) \cdot \left(1 - \frac{\zeta_{D_1}}{n-1}\right) \cdot \left(\frac{\zeta_{D_2}}{n-1}\right) \cdot \dots \cdot \left(1 - \frac{\zeta_{D_{n-2}}}{n-1}\right) \cdot \left(1 - \frac{\zeta_{D_{n-1}}}{n-1}\right) + \dots$$

$$\dots + \left(1 - \frac{\zeta_{D_0}}{n-1}\right) \cdot \left(1 - \frac{\zeta_{D_1}}{n-1}\right) \cdot \left(1 - \frac{\zeta_{D_2}}{n-1}\right) \cdot \dots \cdot \left(\frac{\zeta_{D_{n-2}}}{n-1}\right) \cdot \left(\frac{\zeta_{D_{n-1}}}{n-1}\right) \quad (10)$$

와 같이 나타낼 수 있다. 따라서 포트 D를 제외한 모든 포트로 유입된 데이터 패킷이 포트 D를 지향할 확률  $P(h_D=n-1)_{cycle j}$ 는

$$P(h_D=n-1)_{cycle j} = \left(\frac{\zeta_{D_0}}{n-1}\right) \cdot \left(\frac{\zeta_{D_1}}{n-1}\right) \cdot \left(\frac{\zeta_{D_2}}{n-1}\right) \cdot \dots \cdot \left(\frac{\zeta_{D_{n-1}}}{n-1}\right) \quad (11)$$

와 같이 나타낼 수 있다.

따라서, n개의 포트를 갖는 스위치의 경우 위 수식 (8)~(11)과 같은 방법을 통해서 특정 포트로 출력 지향하는 데이터 패킷의 개수가 0~n-1개가 될 확률  $P(h_D=0)_{cycle j} \sim P(h_D=n-1)_{cycle j}$ 들을 구할 수 있다.

### 2.3 정상상태 처리율 분석

네트워크 내부에 위치한 임의의 출력 버퍼 스위치 내의 데이터 이동 패턴의 확률적 분석을 토대로 네트워크의 성능 분석을 위하여 사용될 변수는 다음과 같다.

- $\beta$  : 스위치에 장착된 버퍼가 저장할 수 있는 데이터 패킷 수
- $\epsilon$  : 버퍼에 저장된 데이터 패킷 수
- $P(\epsilon=k)_i$  : 버퍼에 저장된 데이터 패킷 수가 k개일 확률
- $P(D_d=1)_i$  : 출력 단  $D_d$ 로 데이터 패킷이 출력될 확률
- $P(D_d=0)_i$  : 출력 단  $D_d$ 로 데이터 패킷이 출력되지 않을 확률

네트워크 성능 분석의 두 가지 주요 요소는 네트워크 정상상태 처리율과 네트워크 지연시간이다. 네트워크 정상상태 처리율( $NT$ )은 네트워크 마지막 스위치 출력 단으로 데이터 패킷이 출력될 확률,  $P(D_d=1)_{last\ stage}$ ,을

네트워크 첫 번째 스위치 입력 단으로 데이터 패킷이 유입될 확률을 나누어서 식 (12)와 같이 계산된다.

$$NT = \frac{P(D_d = 1)_{last\ stage}}{\zeta_{stage\ 0}} \quad (12)$$

임의 사이클  $j$ 에 네트워크 내부  $i$ 에 위치한 스위치 출력 단  $D_d$ 로 데이터 패킷이 출력되는 경우를 살펴보면, 먼저 사이클  $(j-1)$  종료 시 해당 출력 단 버퍼가 데이터 패킷을 저장하고 있는 경우, 혹은 스위치 입력 단으로 새로이 유입된 데이터 패킷이 해당 출력 단으로 지향할 경우이다. 반대로 스위치 출력 단  $D_d$ 로 데이터 패킷이 출력되지 않는 경우는 사이클  $(j-1)$ 에 해당 출력 단 버퍼가 데이터 패킷을 저장하지 않은 상태에서, 스위치 입력 단에서 해당 출력 단으로 지향하는 데이터 패킷이 없을 경우이다. 따라서 임의 사이클  $j$ 에 네트워크 내부  $i$ 에 위치한 스위치 출력 단  $D_d$ 로 데이터 패킷이 출력되지 않을 확률,  $P(D_d = 0)_{i, cycle\ j}$ ,을 구하면

$$P(D_d = 0)_{i, cycle\ j} = P(\epsilon = 0)_{i, cycle\ (j-1)} \times P(\bar{h} = 0)_{i, cycle\ (j-1)} \quad (13)$$

이 된다. 여기서  $j \geq \beta$ 이다. 또한, 임의 사이클  $j$ 에 스위치 출력 단  $D_d$ 로 데이터 패킷이 출력 될 확률,  $P(D_d = 1)_{i, cycle\ j}$ ,은

$$P(D_d = 1)_{i, cycle\ j} = 1 - P(D_d = 0)_{i, cycle\ j} \\ = 1 - \{P(\epsilon = 0)_{i, cycle\ (j-1)} \times P(\bar{h} = 0)_{i, cycle\ (j-1)}\} \quad (14)$$

로 계산된다.

따라서, 임의 사이클  $j$ 에 네트워크 내부 임의 스위치 출력 단을 통해 데이터 패킷이 출력될 확률은 스위치 내부로 유입된 데이터 패킷이 특정 출력 단을 지향할 확률,  $P(\bar{h} = 0)_{i, cycle\ (j-1)}$ , 해당 출력 단 버퍼가 데이터 패킷을 저장하지 않고 있을 확률,  $P(\epsilon = 0)_{i, cycle\ (j-1)}$ ,로 나타낼 수 있다. 여기서, 스위치 확률 식  $P(\bar{h} = 0)_{i, cycle\ (j-1)}$ 은 2.2 절의 스위치 형태에 따른 스위치 내부에서의 데이터 이동 패턴 분석을 통하여 네트워크를 구성하는 스위치의 형태에 따라 식 (1)~(11)을 이용하여 계산할 수 있고,  $P(\epsilon = 0)_{i, cycle\ (j-1)}$ , 즉 사이클  $(j-1)$  종료 시점에 버퍼가 비어있을 확률은 다음과 같이 계산된다.

- ① 사이클  $(j-2)$  종료 시 버퍼에 저장된 데이터 패킷의 수가 하나이고, 사이클  $(j-1)$ 에 해당 출력 단으로 향하는 데이터 패킷이 없는 경우
- ② 사이클  $(j-2)$  종료 시 버퍼에 저장된 데이터 패킷이 없고, 사이클  $(j-1)$ 에 해당 출력 단으로 향하는 데이터 패킷이 하나인 경우
- ③ 사이클  $(j-2)$  종료 시 버퍼에 저장된 데이터 패킷이 없고, 사이클  $(j-1)$ 에 해당 출력 단으로 향하는 데이터 패킷이 없는 경우

따라서, 임의 사이클  $(j-1)$ 에 버퍼에 저장된 데이터 패킷의 수가 0일 확률,  $P(\epsilon = 0)_{i, cycle\ (j-1)}$ ,은

$$P(\epsilon = 0)_{i, cycle\ (j-1)} = \\ P(\epsilon = 1)_{i, cycle\ (j-2)} \times P(\bar{h} = 0)_{i, cycle\ (j-1)} \\ + P(\epsilon = 0)_{i, cycle\ (j-2)} \times P(\bar{h} = 1)_{i, cycle\ (j-1)} \\ + P(\epsilon = 0)_{i, cycle\ (j-2)} \times P(\bar{h} = 0)_{i, cycle\ (j-1)} \quad (15)$$

로 계산된다. 여기서,  $b \geq 1$ 이고,  $a \geq 2$ 이다. 식 (15)의  $P(\epsilon = 1)_{i, cycle\ (j-2)}$ 는 사이클  $(j-2)$  종료 시 버퍼에 1개의 데이터 패킷이 저장될 확률은  $P(\epsilon = 0)_{i, cycle\ (j-1)}$  분석과 유사한 과정을 거쳐 확률 식으로 표현하면,

$$P(\epsilon = 1)_{i, cycle\ (j-2)} = P(\epsilon = 2)_{i, cycle\ (j-3)} \times P(\bar{h} = 0)_{i, cycle\ (j-2)} \\ + P(\epsilon = 1)_{i, cycle\ (j-3)} \times P(\bar{h} = 1)_{i, cycle\ (j-2)} \\ + P(\epsilon = 0)_{i, cycle\ (j-3)} \times P(\bar{h} = 2)_{i, cycle\ (j-2)} \quad (16)$$

이다. 같은 방법으로, 식 (16)을 일반화하여 임의 사이클  $(j-k-1)$ 에 버퍼에 저장된 데이터 패킷의 수가  $k$ 일 확률,  $P(\epsilon = k)_{i, cycle\ (j-k-1)}$ ,은

$$P(\epsilon = k)_{i, cycle\ (j-k-1)} = \\ P(\epsilon = k+1)_{i, cycle\ (j-k-2)} \times P(\bar{h} = 0)_{i, cycle\ (j-k-1)} \\ + P(\epsilon = k)_{i, cycle\ (j-k-2)} \times P(\bar{h} = 1)_{i, cycle\ (j-k-1)} \\ + P(\epsilon = k-1)_{i, cycle\ (j-k-2)} \times P(\bar{h} = 2)_{i, cycle\ (j-k-1)} \\ \dots \\ + P(\epsilon = k+1-a)_{i, cycle\ (j-k-2)} \times P(\bar{h} = a)_{i, cycle\ (j-k-1)} \\ = \sum_{x=k+1-a}^{k+1} P(\epsilon = x)_{i, cycle\ (j-k-2)} \quad (17)$$

$$\times P(\bar{h} = k+1-x)_{i, cycle\ (j-k-1)}$$

이다. 여기서  $\beta \geq 1$ ,  $a \geq 2$ 이고  $1 \leq k < \beta$ 이다. 식 (17)은 사이클  $(j-k-1)$ 에 버퍼가 저장하고 있는 데이터 패킷의 수가  $k$ 일 경우는 사이클  $(j-k-2)$ 에 버퍼에 저장된 데이터 패킷의 수와 사이클  $(j-k-1)$ 에 해당 출력 단으로 지향하는 데이터 패킷의 수의 합이  $(k+1)$ 임을 보여주고 있다. 이때 사이클  $(j-k-1)$  동안 하나의 데이터 패킷은 다음 스테이지로 이동하고 나머지  $k$ 개 데이터 패킷은 버퍼에 저장된다. 식 (17)에서  $x$ 가 0보다 적게 되면,  $P(\epsilon = x)_{i, cycle\ (j-b-1)} \Big|_{x < 0} = 0$ 가 된다. 마지막으로 사이클  $(j-\beta-1)$ 에서 버퍼에 저장된 데이터 패킷의 수가  $\beta$ 일 확률,  $P(\epsilon = \beta)_{i, cycle\ (j-\beta-1)}$ 을 구하면,

$$P(\epsilon = \beta)_{i, cycle\ (j-\beta-1)} = \\ P(\epsilon = \beta)_{i, cycle\ (j-\beta-2)} \times \sum_{y=1}^{\beta} P(\bar{h} = y)_{i, cycle\ (j-\beta-1)} \\ + P(\epsilon = \beta-1)_{i, cycle\ (j-\beta-2)} \times \sum_{y=2}^{\beta} P(\bar{h} = y)_{i, cycle\ (j-\beta-1)} \\ \dots \\ + P(\epsilon = \beta+1-a)_{i, cycle\ (j-\beta-2)} \times \sum_{y=a}^{\beta} P(\bar{h} = y)_{i, cycle\ (j-\beta-1)}$$



$$= \sum_{x=b+1-a}^{\beta} \left\{ P(\varepsilon = x)_{i, \text{cycle}(j-\beta-2)} \times \sum_{y=\beta+1-x}^{\alpha} P(\bar{h} = y)_{i, \text{cycle}(j-\beta-1)} \right\} \quad (18)$$

이 된다. 여기서, 만약 이전 사이클에서 버퍼에 저장된 데이터 패킷의 수와 현재 사이클에서 해당 출력 단으로 지향하는 데이터 패킷의 합이  $(\beta+1)$ 보다 큰 경우 데이터 충돌에 연루된 모든 데이터 패킷을 저장할 버퍼 공간이 부족하므로 데이터 패킷의 손실이 일어난다. 즉, 버퍼의 최대저장 할 수 있는 데이터 패킷의 수가  $\beta$  이므로,  $(x+y+1-\beta)$ 개의 데이터 패킷은 손실된다.

식 (15), (16), (17) 그리고 (18) 등의 식에서, 확률적으로 임의 버퍼가 사이클  $j$ 에  $k$ 개의 데이터 패킷을 저장할 확률과 사이클  $(j+1)$ 에  $k$ 개의 데이터 패킷을 저장할 확률은 같다고 볼 수 있다. 즉, 이들 식에 정상 상태 확률(steady state probability) 개념 적용이 가능하고,  $P(\varepsilon = k)_{i, \text{cycle } j} = P(\varepsilon = k)_{i, \text{cycle}(j+1)}$ , 그리고  $P(\bar{h} = x)_{i, \text{cycle } j} = P(\bar{h} = x)_{i, \text{cycle}(j+1)}$ 이 된다. 정상상태 확률 개념을 이용하여 식 (15)를 다시 쓰면

$$\begin{aligned} P(\varepsilon = 0)_i &= P(\varepsilon = 1)_i \times P(\bar{h} = 0)_i \\ &+ P(\varepsilon = 0)_i \times P(\bar{h} = 1)_i \\ &+ P(\varepsilon = 0)_i \times P(\bar{h} = 0)_i \end{aligned} \quad (19)$$

이 된다.  $P(\varepsilon = 0)_i$ 를 얻기위해 식 (19)를 정리하여  $P(\varepsilon = 1)_i$ 를  $P(\varepsilon = 0)_i$ 의 식으로 구하면

$$\begin{aligned} P(\varepsilon = 1)_i &= P(\varepsilon = 0)_i \times \frac{(1 - P(\bar{h} = 0)_i - P(\bar{h} = 1)_i)}{P(\bar{h} = 0)_i} \\ &= P(\varepsilon = 0)_i \times \frac{1}{P(\bar{h} = 0)_i} \times \sum_{y=2}^{\alpha} P(\bar{h} = y)_i \\ &= P(\varepsilon = 0)_i \times \Omega_0 \\ &= P(\varepsilon = 0)_i \times \Phi_1 \end{aligned} \quad (20)$$

이다. 여기서  $\Omega_0 = \frac{1}{P(\bar{h} = 0)_i} \times \sum_{y=2}^{\alpha} P(\bar{h} = y)_i$ 이고,  $\Phi_1 = \Omega_0$

이다.  $P(\bar{h} = y)_i$ 는 2.2절에서 네트워크를 구성하는 스위치의 형태에 따른 분석을 통하여 구할 수 있다. 또한,  $P(\varepsilon = 1)_i$ 는 다음과 같이 나타낼 수도 있다.

$$\begin{aligned} P(\varepsilon = 1)_i &= P(\varepsilon = 0)_i \times \sum_{y=2}^{\alpha} P(\bar{h} = y)_i \\ &+ P(\varepsilon = 1)_i \times \sum_{y=1}^{\alpha} P(\bar{h} = y)_i \end{aligned} \quad (21)$$

같은 방법으로 식 (16)의  $P(\varepsilon = 1)_i$ 은 다음과 같이 나타내고

$$\begin{aligned} P(\varepsilon = 1)_i &= P(\varepsilon = 2)_i \times P(\bar{h} = 0)_i \\ &+ P(\varepsilon = 1)_i \times P(\bar{h} = 1)_i \\ &+ P(\varepsilon = 0)_i \times P(\bar{h} = 2)_i \end{aligned} \quad (22)$$

와 같이 정리된다.

여기서  $P(\varepsilon = 2)_i$ 는 식 (21)과 식 (22)를 이용하여 다음과 같이 두 가지 형태로 정리할 수 있다.

$$\begin{aligned} P(\varepsilon = 2)_i &= P(\varepsilon = 0)_i \times \frac{1}{P(\bar{h} = 0)_i} \times \sum_{y=3}^{\alpha} P(\bar{h} = y)_i \\ &+ P(\varepsilon = 1)_i \times \frac{1}{P(\bar{h} = 0)_i} \times \sum_{y=2}^{\alpha} P(\bar{h} = y)_i \\ &= P(\varepsilon = 0)_i \times \Omega_1 + P(\varepsilon = 1) \times \Omega_0 \\ &= P(\varepsilon = 0)_i \times \{\Omega_1 + \Phi_1 \times \Omega_0\} \\ &= P(\varepsilon = 0)_i \times \Phi_2 \end{aligned} \quad (23)$$

또는

$$\begin{aligned} P(\varepsilon = 2)_i &= P(\varepsilon = 0)_i \times \sum_{y=3}^{\alpha} P(\bar{h} = y)_i \\ &+ P(\varepsilon = 1)_i \times \sum_{y=2}^{\alpha} P(\bar{h} = y)_i \\ &+ P(\varepsilon = 2)_i \times \sum_{y=1}^{\alpha} P(\bar{h} = y)_i \end{aligned} \quad (24)$$

$$\text{여기서 } \Omega_0 = \frac{1}{P(\bar{h} = 0)_i} \times \sum_{y=2}^{\alpha} P(\bar{h} = y)_i,$$

$$\Omega_1 = \frac{1}{P(\bar{h} = 0)_i} \times \sum_{y=3}^{\alpha} P(\bar{h} = y)_i, \quad \Phi_1 = \Omega_0,$$

$\Phi_2 = \Omega_1 + \Phi_1 \times \Omega_0$ 이다. 같은 방법으로 식 (19)~(24)를 일반화하여 버퍼가 임의 사이클 종료 시  $(k-1)$ 개의 데이터 패킷을 저장하고 있을 확률,  $P(\varepsilon = k-1)_i$ 은

$$P(\varepsilon = k-1)_i = \sum_{x=k-a}^k P(\varepsilon = x)_i \times P(\bar{h} = k-x)_i \quad (25)$$

이 되고, 이 식으로부터 버퍼가 임의 사이클 종료 시  $k$ 개의 데이터 패킷을 저장하고 있을 확률,  $P(\varepsilon = k)_i$ 을 구하면

$$\begin{aligned} P(\varepsilon = k)_i &= P(\varepsilon = 0)_i \times \frac{1}{P(\bar{h} = 0)_i} \times \sum_{y=k+1}^{\alpha} P(\bar{h} = y)_i \\ &+ P(\varepsilon = 1)_i \times \frac{1}{P(\bar{h} = 0)_i} \times \sum_{y=k}^{\alpha} P(\bar{h} = y)_i \\ &\dots \\ &+ P(\varepsilon = k-1)_i \times \frac{1}{P(\bar{h} = 0)_i} \times \sum_{y=2}^{\alpha} P(\bar{h} = y)_i \\ &= \sum_{x=0}^{k-1} \left\{ P(\varepsilon = x)_i \times \frac{1}{P(\bar{h} = 0)_i} \times \sum_{y=k+1-x}^{\alpha} P(\bar{h} = y)_i \right\} \\ &= P(\varepsilon = 0)_i \times \Omega_{k-1} + P(\varepsilon = 1)_i \times \Omega_{k-2} + \dots \\ &\quad + P(\varepsilon = k-1)_i \times \Omega_0 \\ &= P(\varepsilon = 0)_i \times \{\Omega_{k-1} + \Phi_1 \times \Omega_{k-2} + \dots + \Phi_{k-1} \times \Omega_0\} \\ &= P(\varepsilon = 0)_i \times \Phi_k \end{aligned} \quad (26)$$

또는

$$P(\varepsilon = k)_i = \sum_{x=0}^k \left\{ P(\varepsilon = x)_i \times \sum_{y=k+1-x}^{\alpha} P(\bar{h} = y)_i \right\} \quad (27)$$

로 정리할 수 있다.

$$\text{여기서 } \Omega_l = \frac{1}{P(\bar{h}=0)_i} \times \sum_{y=l+2}^a P(\bar{h}=y)_i,$$

$$\Phi_k = \Omega_{k-1} + \sum_{x=0}^{k-2} \Phi_{k-x-1} \times \Omega_x \text{ 이다.}$$

마지막으로, 버퍼에  $b$ 개의 데이터 패킷이 저장될 확률은, 즉 버퍼가 완전히 차게 될 확률,  $P(\epsilon=\beta)_i$ ,는 다음과 같이 구할 수 있다.

$$P(\epsilon=\beta-1)_i = \sum_{x=b-a}^{\beta} P(\epsilon=x)_i \times P(\bar{h}=\beta-x)_i \quad (28)$$

이고,

$$\begin{aligned} P(\epsilon=\beta)_i &= \sum_{x=0}^{\beta-1} \left\{ P(\epsilon=x)_i \times \frac{1}{P(\bar{h}=0)_i} \times \sum_{y=\beta+1-x}^a P(\bar{h}=y)_i \right\} \\ &= P(\epsilon=0)_i \times (\Omega_{\beta-1} + \Phi_1 \times \Omega_{\beta-2} + \dots \\ &\quad + \Phi_{\beta-2} \times \Omega_1 + \Phi_{\beta-1} \times \Omega_0) \\ &= P(\epsilon=0)_i \times \Phi_{\beta} \end{aligned} \quad (29)$$

이다. 여기서  $\Omega_l = \frac{1}{P(\bar{h}=0)_i} \times \sum_{y=l+2}^a P(\bar{h}=y)_i$ ,  $\Phi_b = \Omega_{\beta-1} + \sum_{x=0}^{\beta-2} \Phi_{\beta-x-1} \times \Omega_x$  이다. 식 (20), (23), (26), 그리고 (29)식으로부터 임의의  $k$ 에 대한  $P(\epsilon=k)_i$ 는  $P(\epsilon=0)_i$ 와  $\Omega_l$ ,  $\Phi_k$ 를 이용하여 계산이 가능하다. 이때  $P(\epsilon=0)_i$ 는 다음과 같이 계산할 수 있다. 스위치에 장착한 버퍼의 개수가  $b$ 개인 경우 임의의 싸이클 종료 시 버퍼에 저장된 데이터 패킷의 개수는 0에서  $\beta$ 개 중 하나가 된다. 즉,  $\sum_{x=0}^{\beta} P(\epsilon=x)_i = 1$ . 따라서,

$$\sum_{x=0}^{\beta} P(\epsilon=x)_i = P(\epsilon=0)_i \times \sum_{x=0}^{\beta} \Phi_x = 1 \quad (30)$$

이 된다. 따라서, 정상 상태 처리율 계산의 주요 변수로 정의된  $P(\epsilon=0)_i$ 은

$$P(\epsilon=0)_i = \frac{1}{\sum_{x=0}^{\beta} \Phi_x} \quad (31)$$

로 얻어진다. 여기서  $\Phi_x = \Omega_{x-1} + \sum_{k=0}^{x-2} \Phi_{x-k-1} \times \Omega_k$ ,

$$\Omega_l = \frac{1}{P(\bar{h}=0)_i} \times \sum_{y=l+2}^a P(\bar{h}=y)_i \text{ 이다.}$$

일단 네트워크 내부 임의의  $i$ 번째에 위치한 스위치 입력 단의 데이터 패킷이 유입율이 주어지면 네트워크를 구성하는 스위치의 형태에 따라 2.2절의 식 (1)~(10)과 (13), (14), 그리고 (31)를 이용하여 출력 단  $D_d$ 로 데이터 패킷이 출력될 확률,  $P(D_d=1)_i$ 을 구하게 된다. 네트워크 구조 상  $i$ 에 위치한 스위치 출력 단은  $(i+1)$ 에 위치한 임의의 스위치의 입력 단으로 연결됨으로,  $i$ 의 임

의 스위치 출력 단으로 데이터 패킷이 출력될 확률,  $P(D_d=1)_i = \zeta_{stage\ i+1}$ 이 된다. 따라서 네트워크 입력 단의 데이터 유입율,  $\zeta_0$ ,이 주어지면  $P(D_d=1)_0$ 을 구하고, 다시  $P(D_d=1)_0$ 을  $\zeta_1$ 로 하여  $P(D_d=1)_1$ 을 구하는 과정을 반복하여 네트워크의 스위치 출력 단으로 데이터 패킷이 출력될 확률,  $P(D_d=1)_{last\ stage}$ 을 계산하게 된다. 마지막으로, 네트워크 정상상태 처리율은  $\zeta_{stage\ 0}$ 와  $P(D_d=1)_{last\ stage}$ 을 이용하여 식 (12)에 의해 구할 수 있다.

## 2.4 네트워크 지연시간 분석

네트워크 성능 평가에 있어 정상상태 처리율과 함께, 또 다른 주요 평가 지표로 네트워크 지연시간(Network Delay,  $J$ )을 들 수 있다. 임의의 데이터 패킷이 네트워크 입력 단에 유입된 후, 각 스위치 스테이지를 지나, 최종 출력 단을 통과하기까지 소요되는 스위치 클럭의 평균 개수로 측정되는 네트워크 지연시간은 데이터 패킷 이동 경로의 트래픽에 따라 결정된다. 임의의 데이터 패킷의 이동 경로에 위치한 버퍼의 데이터 저장 상태, 그리고 함께 도착된 데이터 패킷의 개수에 따라 데이터 패킷을 특정 위치의 버퍼에 저장하게 되고, 일정 기간 동안 스위치에 머무르도록 한다. 문제는 “임의의 데이터 패킷이 스위치 버퍼의 어느 위치에 저장되는가?” 이다. 일단, 스위치 버퍼  $k$ 번째 위치에 저장되면, 이는 해당 스위치에서  $(k+1) \times \Delta t$ 의 시간만큼 머물고 다음 스위치로 이동하게 된다.

먼저, 네트워크 스테이지  $i$ 에 위치한 스위치 소자를 성공적으로 통과한 데이터 패킷  $\delta$ 가 해당 스위치의  $k$ 번째 버퍼에 저장 될 경우를 살펴보면 다음과 같다: ‘이전 싸이클 종료 시  $\rho$ 개 데이터 패킷이 저장된 스위치 버퍼에 현 싸이클에 데이터 패킷  $\delta$ 를 포함한  $(y+1)$ 개의 새로운 데이터 패킷들이 도착한다. 이들 새로 도착한 데이터 패킷 가운데 패킷  $\delta$ 가  $(k-\rho+1)$ 번째 순서로 버퍼에 저장될 경우, 데이터 패킷  $\delta$ 는 해당 스위치의  $k$ 번째 버퍼에 저장된다.’ 여기서,  $0 \leq \rho \leq k \leq b$ , 그리고  $(k-\rho+1) \leq (y+1) \leq a$  이다.

따라서, 네트워크 내부  $i$ 에 위치한 스위치를 통과하여 다음 스테이지로 이동에 성공한 데이터 패킷이 해당 스위치에 체류한 시간,  $\tau_{s,stage\ i}$ ,을 구하면

$$\begin{aligned} \tau_{s,stage\ i} &= \sum_{k=0}^b \left[ \sum_{\varphi=k+1-a}^k \left\{ P(\epsilon=\rho)_{i,cycle(j-1)} \times \sum_{y=k-\rho}^{a-1} \frac{1}{y+1} P(\bar{h}=y)_{i,cycle\ j}^* \right\} \right. \\ &\quad \left. \times (k+1) \Delta t \right] \end{aligned} \quad (32)$$

으로 얻어진다. 여기서,  $P(\bar{h}=y)^* = \left( \frac{a-1}{y} \right) \times \left( \frac{\zeta_{stage\ i}}{a} \right)^y$

$\left(1 - \frac{\zeta_{stage\ i}}{a}\right)^{a-1-y}$  그리고,  $P(\epsilon < 0) = 0$ 이다. 식 (32)는 임의 데이터 패킷  $\delta$ 가 스테이지  $i$ 의 스위치를 성공적으로 통과하는데 요구되는 평균 지연 시간을 수식화한 것이다. 식 (32)의  $\sum_{y=k-\rho}^{a-1} \frac{1}{y+1} \times P(\bar{h}=y)_{i,cycle\ j}^*$ 는 데이터 패킷  $\delta$ 를 포함한  $(y+1)$ 개의 데이터 패킷이 해당 버퍼에 새로 도착되고, 이들 가운데 데이터 패킷  $\delta$ 가  $k$ 번째 버퍼 공간에 저장될 확률을 나타낸다.

만약 해당 출력 단 버퍼가 빈 상태에서 새로 도착된  $(y+1)$ 개 데이터 패킷 가운데 스위치를 통과할 첫 번째 패킷으로 선정되어진다면 의심의 여지없이 데이터 패킷은  $\delta$ 는 단 하나의 스위치 사이클,  $\Delta t$ , 동안만 해당 스위치에 머무르고 다음 스테이지로 이동 할 것이다. 이를 식 (32)를 이용하여 다시 쓰면

$$\tau_{s,stage\ i} = \left\{ P(\epsilon=0)_{i,cycle\ (j-1)} \times \sum_{y=0}^{a-1} \frac{1}{y+1} \times P(\bar{h}=y)_{cycle\ j} \right\} \times \Delta t \quad (32-1)$$

이 된다.

네트워크를 성공적으로 통과한 데이터 패킷의 전체 네트워크 지연시간,  $\tau_s$ ,는 각 스테이지 별 지체시간을 합하여

$$\tau_s = \sum_{i=0}^{n-1} \tau_{s,stage\ i} \quad (33)$$

과 같이 나타낼 수 있다. 여기서  $n$ 은 네트워크 유입 단에서부터 출력 단까지의 스위치의 개수를 나타낸다.

한편, 일부 데이터 패킷들은 한정된 버퍼공간으로 인하여 전송 중, 네트워크 내부에서 유실될 수 있다. 이들 중도 유실된 데이터 패킷들은 소정의 “중도 유실 감지” 과정을 거쳐 최초 데이터 패킷이 유입된 입력 단에서 재전송 되게 된다. 그림 4는 네트워크를 성공적으로 통과한 데이터 패킷들의 네트워크 지연 시간과 함께, 전송 과정에서 중도 소실된 데이터 패킷들의 재 전송 시간을 고려한 총 네트워크 지연시간에 관한 상태도이다. 여기

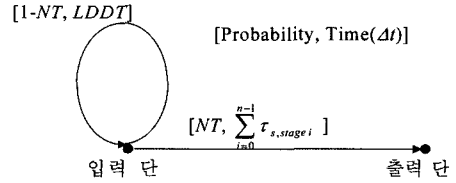


그림 4 네트워크 지연 시간에 관한 상태도

서, 임의 데이터 패킷  $\delta$ 가 네트워크를 성공적으로 통과할 확률은 정상상태 처리율(NT)로 볼 수 있고, 이때 네트워크 지연시간은 식 (32)와 (33)에서 얻은 식으로 구할 수 있다.

반면에, 데이터 패킷  $\delta$ 가 네트워크 내부에서 소실될 확률은  $(1-NT)$ 로 계산되고, 이들 중도 소실 데이터 패킷은 중도 유실 감지 시간(Lost Data Detection Time, LDDT) 만큼의 오류 검사 과정을 거쳐 재전송된다.

따라서, 임의 데이터 패킷이 전체 네트워크를 통과하는데 걸리는 평균 시간,  $J$ ,는 그림 4으로 부터  $J = NT \times \tau_s + (1-NT) \times (LDDT + J)$  (34)

와 같은 식으로 얻어진다. 식 (34)를  $J$ 에 관하여 풀면

$$J = \tau_s + \frac{(1-NT)}{NT} \times LDDT \quad (35)$$

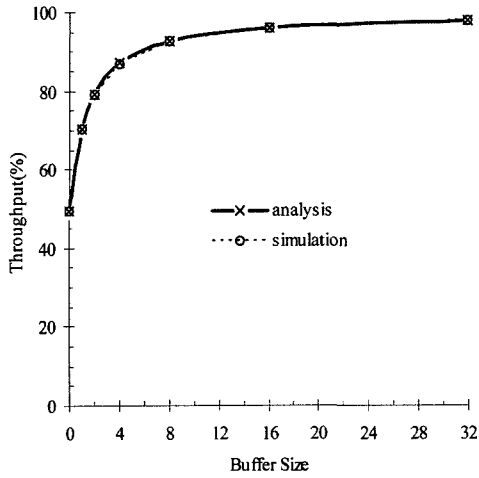
과 같이 계산된다. 여기서,  $NT$ 는 2.2.3절을 통해서 구할 수 있고,  $\tau_s$ 는 식 (32) 그리고 (33)로부터 구할 수 있고, LDDT는 네트워크 특성에 따라 상수로 주어진다.

### 3. 네트워크들의 성능 평가

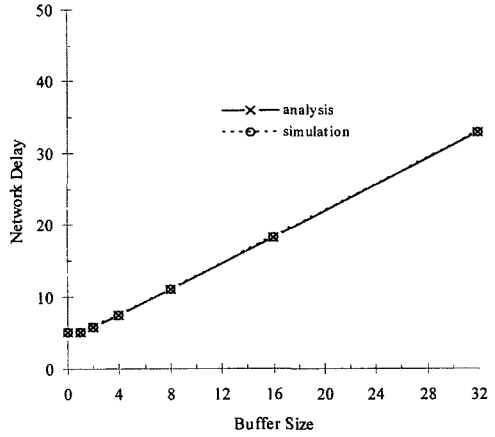
다음의 표 1~표 4, 그림 5~그림 8은 각각 8×8 buffered crossbar 스위치로 구성된 64×64 MIN과 Fat-tree(3, 3, 2)와 4개의 포트를 가진 스위치 라우터의 스위치에 장착된 버퍼의 크기에 따른 네트워크 정상상태 처리율과 지연시간에 관한 분석 결과를 비교한 표와 그래프이다. 본 논문에서 기술한 분석 모형을 적용하여 각 네트워크 입력 단으로 데이터 패킷이 유입될 확률을 변화시켜가면서 예측한 정상상태 처리율과 네트워크

표 1 8×8 buffered crossbar 스위치로 구성된 64×64 MIN의 성능 ( $\zeta_{stage\ 0} = 1$ )

buffer size	Normalized Throughput (NT,%)		Network Delay ( $\Delta t$ )			
	Analysis	Simulation	Delay for success packets ( $\Delta t$ )		Discarded 확률 (%)	
			Analysis	Simulation	Analysis	Simulation
0	49.59	49.60	2	2	50.415	50.403
1	70.33	70.24	2.975	2.975	29.673	29.758
2	79.21	79.15	3.925	3.923	20.792	20.846
4	87.01	86.88	5.785	5.786	12.994	13.118
8	92.58	92.51	9.472	9.483	7.422	7.493
16	96.00	95.95	16.825	16.821	3.995	4.046
32	97.92	97.92	31.518	31.380	2.077	2.078



(a) 정상상태 처리율

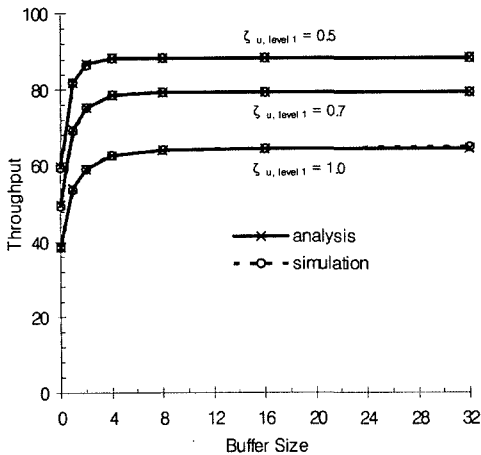


(b) 지연시간( $LDDT = (b \times n + 1)\Delta t$ )

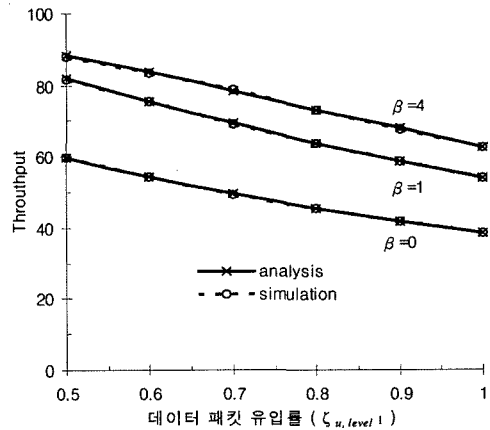
그림 5 다중 buffered 8x8 스위치로 구성된 64x64 MIN,  $\zeta_{stage 0} = 1$

표 2 FT(3, 3, 2)의 성능(정상 상태 처리율)

Buffer size	정상상태 처리율 (NT, %)					
	데이터 패킷 입력률 ( $\zeta_{u, level 1}$ )					
	$\zeta_{u, level 1} = 0.5$		$\zeta_{u, level 1} = 0.7$		$\zeta_{u, level 1} = 1.0$	
	해석	시뮬레이션	해석	시뮬레이션	해석	시뮬레이션
0	59.78	59.59	49.49	49.43	38.34	38.36
1	82.00	81.73	69.45	69.12	53.96	53.78
2	86.86	86.68	75.41	75.19	59.22	59.02
4	88.35	88.25	78.57	78.60	62.73	62.71
8	88.46	88.53	79.34	79.27	64.29	64.10
16	88.46	88.44	79.36	79.41	64.63	64.56
32	88.46	88.34	79.36	79.26	64.65	64.84



(a) 버퍼 사이즈에 따른 Throughput 변화

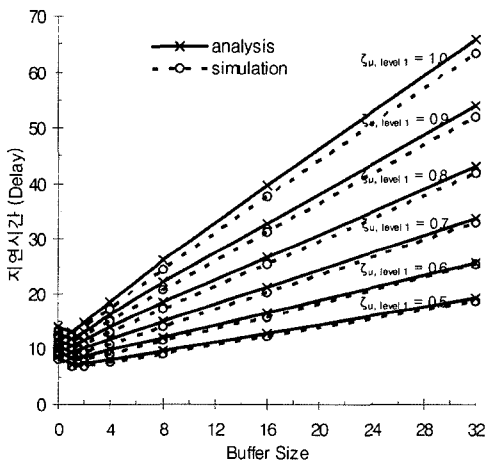


(b) 데이터 유입률에 따른 Throughput의 변화

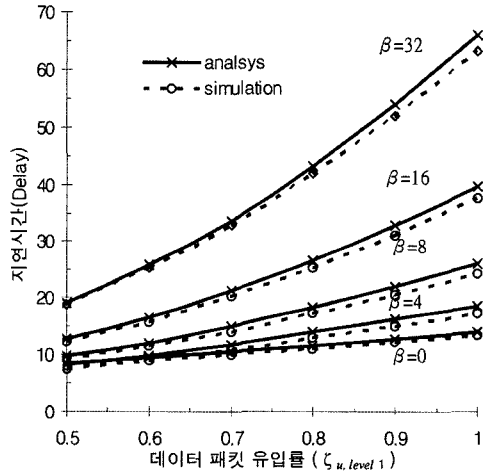
그림 6 버퍼를 장착한 FT(3, 3, 2) 성능 분석 결과와 시뮬레이션 결과의 비교

표 3 FT(3, 3, 2)의 성능(네트워크 지연 시간)

네트워크 지연시간								
Buffer size	$\zeta_{u, level 1} = 0.7$				$\zeta_{u, level 1} = 1.0$			
	네트워크를 통과한 패킷의 지연시간 ( $\Delta t$ )		데이터 패킷 탈락 확률 (%)		네트워크를 통과한 패킷의 지연시간 ( $\Delta t$ )		데이터 패킷 탈락 확률 (%)	
	해석	시뮬레이션	해석	시뮬레이션	해석	시뮬레이션	해석	시뮬레이션
0	4.406	3.946	50.51	50.57	4.311	3.781	61.66	61.64
1	5.432	4.881	30.55	30.89	5.498	4.860	46.04	46.22
2	6.139	5.429	24.59	24.81	6.487	5.594	40.78	40.98
4	6.852	5.965	21.43	21.40	7.906	6.584	37.27	37.29
8	7.180	6.233	20.67	20.73	9.370	7.576	35.71	35.90
16	7.211	6.277	20.63	20.59	10.06	8.036	35.37	35.44
32	7.212	6.293	20.63	20.74	10.12	8.011	35.35	35.16



(a) 버퍼 사이즈에 따른 네트워크 지연시간 변화



(b) 데이터 유입률에 따른 네트워크 지연시간 변화

그림 7 버퍼를 장착한 FT(3, 3, 2) 성능 분석 결과와 시뮬레이션 결과의 비교 (네트워크 지연시간)

표 4 4개의 포트를 가진 스위치 라우터에서의 포트 3의 트래픽 분석 ( $\zeta_{D1} = 0.7$ )

buffer size	Normalized Throughput (NT,%)		Network Delay( $\Delta t$ )			
	Analysis	simulation	Delay for success packets ( $\Delta t$ )		Discarded 확률 (%)	
			Analysis	Simulation	Analysis	Simulation
0	73.09	73.22	1.00	1.00	26.91	26.78
1	87.78	87.85	1.49	1.49	12.22	12.15
2	93.04	93.16	1.90	1.88	6.96	6.84
4	97.07	97.12	2.54	2.53	2.93	2.88
8	99.28	99.28	3.35	3.33	0.72	0.72
16	99.94	99.89	3.89	3.84	0.06	0.11
32	100	100	3.99	3.98	0	0

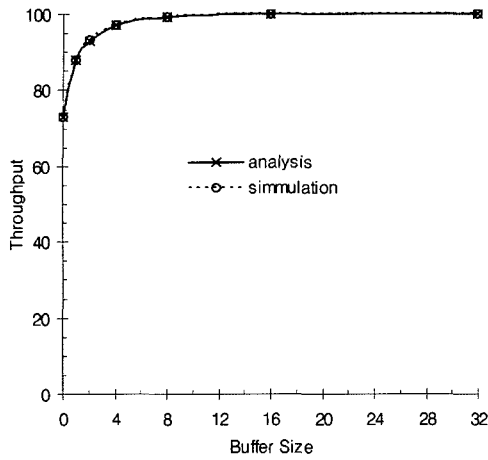
크 지연시간을 나열하고, 이를 다시 시뮬레이션을 통하여 얻은 데이터들과 비교하여 보여주고 있다. 시뮬레이션 과정에서는, 초기에 버퍼가 비어있는 상태에서 네트워크 성능 측정을 피하기 위하여, 충분한 예비 동작시간을 준 후 본격적으로 데이터를 수집하여 처리하였다. 표에서 보인 바와 같이, 네트워크 성능 관련 분석 결과는 다양한 버퍼 크기 및 네트워크 트래픽에 대하여 시뮬레

이션 결과와 매우 근접한 값을 갖는 것으로 밝혀졌다.

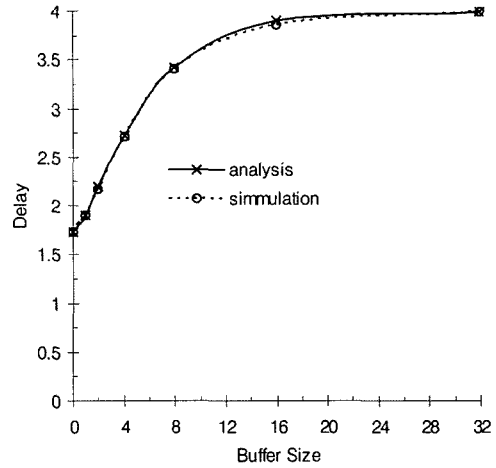
#### 4. 결론

본 논문에서는 스위치 출력 단에 복수 버퍼를 장착한 크로스바 스위치들로 구성된 네트워크의 성능 분석 모형을 제안하고, 실효성을 입증하였다.

분석 모형은 네트워크를 구성하는 크로스바 스위치의



(a) Buffer Size vs. Throughput



(b) Buffer Size vs. Delay

그림 8 4개의 포트를 가진 스위치 라우터에서의 포트 3의 트래픽 분석

변형을 고려하여 크게 3가지 형태의 네트워크,  $a \times a$  크로스바 스위치들로 구성된 다단 연결망과  $a \times b$  스위치들로 구성된 Fat-tree 네트워크,  $n$ 개의 포트를 가진 스위치 라우터의 성능 분석 모형을 개발하였다.

제안된 분석 모형은 네트워크 스위치 내부에서 데이터 패킷의 이동 상태를 관찰하여 확률 식으로 정리하고, 이를 토대로 네트워크 전체의 정상상태 처리율 및 네트워크 지연시간을 예측하였다. 분석 모형의 수립 단계에서 정상상태 확률 개념을 도입하여 간단한 근사화(approximation)을 시도하고, 모형의 해석과 확률 식 전개를 용이하게 하였다. 또한 본 논문에서는 모형의 이해를 돕기 위하여 네트워크 트래픽 제어 및 중도유실 패킷에 대한 처리기능 등 최근 개발되는 스위치 네트워크의 부가기능을 배제하고 수식을 정리하였다. 그러나, 제안된 분석모형은 이들 다양한 성능 향상 기술이 적용된 네트워크, 그리고 다양한 형태의 스위치를 사용하는 네트워크 성능분석에도 쉽게 적용이 가능하다. 모형의 실효성 검토를 위하여 병행된 시뮬레이션 결과는 분석 모형에 의하여 얻은 결과와 상호 미세한 오차 범위 내에서 일치하여, 제안된 분석 기법의 우수성을 입증하였다.

### 참고 문헌

- [1] S. H. Byun, D. K. Sung, "The UniMIN Switch Architecture for Large-Scale ATM Switches," *IEEE Trans. on Networking*, Vol.8, No.1, pp.109-120, Feb. 2000.
- [2] V. P. Kumar and S. M. Reddy, "Augmented Shuffle-Exchange Multistage Interconnection Networks," *IEEE Computer*, Jun. 1987.
- [3] A. Landin, E. Haggersten and S. Haridi, "Race-free interconnection network and multiprocessor consistency," Proceedings of the 18th Annual Symposium on Computer Architecture, Vol.19, No.3, Toronto, Canada (May 1991), pp.106-115.
- [4] Sabine R. Ohring, Maximilian Ibel, Sajal K. Das, Mohan J. Kumar "On Generalized Fat trees," *Parallel Processing Symposium*, 1995, Proceedings, 9th International, 1995, Page(s): 37-44.
- [5] Ronald I. Greenberg, Lee Guan, "An Improved Analytical Model for Wormhole Routed Networks with Application to Butterfly Fat-trees," *Parallel Processing*, 1997., Proceedings of the 1997 International Conference on, 1997, Page(s): 44-48.
- [6] H. Yoon, K. Y. Lee, and M. T. Liu, "Performance Analysis of Multibuffered Packet-Switching Networks in Multiprocessor Systems," *IEEE Trans. on Computers*, Vol.C-39, No.3. pp.319-327, Mar. 1990.
- [7] Y. Mun and H. Y. Yoon, "Performance Analysis of Finite Buffered Multistage Interconnection Networks," *IEEE Trans. on Computers*, Vol.43, No.2, pp.153-162, Feb. 1994.
- [8] Chita R. Das and Prasant Mohapatra, "Performance Analysis of Finite-Buffered Asynchronous Multistage Interconnection Networks," *IEEE Trans. on Parallel and Distributed systems*, Vol.7, No.1, pp. 18-25, Jun 1996.
- [9] H. Y. Yoon and H. S. Choo, "Performance Enhancement of Multistage Interconnection Networks with Unit Step Buffering," *IEEE Trans. on Communications*, Vol.47, No.4. pp.618-630, APR. 1999.
- [10] Myung K Yang and Tae Z Shin, "Performance Evaluation of the Buffered MIN with  $a \times a$  Switches," *KISS Conf. on Parallel Processing*,

- pp.244-246, Nov. 2000.
- [11] Alunweiri H.M, Aljunaidi H, Beraldi R, "The Buffered Fat-Tree ATM switch," Global Telecommunication Conference, 1995. GLOBECOM '95., IEEE Volume: 2, 1995, Page(s): 1209-1215 vol.2.
- [12] Youngsik Kim, Oh-Young Kwon, Tack-Don Han, Youngsong Mun, "Design and performance analysis of the Practical Fat Tree Network using a butterfly network," Journal of systems Architecture 43, pp.355-363, 1997.
- [13] C.E. Leiserson, "Fat trees : universal networks for hardware-efficient supercomputing," IEEE Trans. on Computers Vol.c-34, No.10. pp.892-901, Oct. 1985.
- [14] Cisco Systems Inc., "Catalyst 8500 Campus Switch Router Architecture," [http://www.cisco.com/warp/public/cc/cisco/mkt/switch/cat/8500/tech/8510\\_wp.htm](http://www.cisco.com/warp/public/cc/cisco/mkt/switch/cat/8500/tech/8510_wp.htm)
- [15] Cisco Systems Inc., "Next Generation ClearChannel Architecture for Catalyst 1900/2820 Ethernet Switches," [http://www.cisco.com/warp/public/cc/cisco/mkt/switch/cat/c1928/tech/n\\_wgen\\_wp.htm](http://www.cisco.com/warp/public/cc/cisco/mkt/switch/cat/c1928/tech/n_wgen_wp.htm)
- [16] F. Tobagi, "Fast Packet Switch Architectures for Broadband Integrated Networks," Proc. of the IEEE, Vol.78, No.1, pp.133-167, Jan 1990.



신 태 지

1998년 울산대학교 전기공학과 졸업(학사). 2000년 울산대학교 전기전자정보시스템 공학부 졸업(공학석사). 2004년 전기전자정보시스템 공학부 졸업(공학박사). 2005년~현재 울산대학교 전기전자정보시스템공학부 강의전담 교수. 관심분야는 컴퓨터 네트워크, 병렬 처리시스템



남 창 우

1984년 연세대학교 전기공학과 졸업(학사). 1989년 University of Florida 졸업(석사). 1994년 The Pennsylvania State University 졸업(박사). 1995년~현재 울산대학교 전기전자 및 정보시스템공학부 교수



양 명 국

1983년 한양대학교 전자 공학과 졸업(학사). 1992년 The pennsylvania State university, Electrical and Computer Engineering 졸업(공학 박사). 1993년~현재 울산대학교 전기전자 및 정보시스템 공학부 교수. 관심분야는 컴퓨터 네트워크, 병렬 처리 시스템, 고장 적응 시스템

워크, 병렬 처리 시스템, 고장 적응 시스템