출력 버퍼형 $a \times b$ 스위치로 구성된 Fat-tree 망의 성능 분석

(Performance Evaluation of a Fat-tree Network with Output-Buffered $a \times b$ Switches)

신 태 지 † 양 명 국 ††
(Tae-zi Shin) (Myung-kook Yang)

요 약 본 논문에서는, $a \times b$ 출력 버퍼 스위치로 구성된 fat-tree 망의 성능 예측 모형을 제안하고, 스위치에 장착된 버퍼의 개수 증가에 따른 성능 향상 추이를 분석하였다. Buffered 스위치 기법은 스위치 네트웍 내부의 데이타 충돌 문제를 효과적으로 해결할 수 있는 방법으로 널리 알려져 있다. 제안한 성능 예측 모형은 먼저 네트웍 내부 임의 스위치 입력 단에 유입되는 데이타 패킷이 스위치 내부에서 전송되는 유형을 확률적으로 분석하여 수립되었다. 제안한 모형은 스위치에 장착된 버퍼의 개수와 무관하게 출력 버퍼를 장착한 $a \times b$ 스위치의 성능, 즉 네트웍 성능 평가의 두 가지 주요 요소인 네트웍 정상상태 처리율(Steady state Throughput, ST)과 네트웍 지연시간(Network Delay)의 예측이 가능하다. 또한 모형의 이해를 도모하기 위하여 지능형 네트워크 트래픽 제어 및 중도 소실 패킷에 대한 다양한 처리 기능 등 최근 개발되는 스위치 네트워크의 부가기능을 배제하고 수식을 정리하였다. 그러나, 제안된 분석 모형은 이들 다양한 성능 향상 기술이 적용된 네트워크, 그리고 다양한 크기의 네트워크 성능분석에도 쉽게 적용이 가능하다. 제안한 수학적 성능 분석 연구의 실효성 검증을 위하여 병행된 시뮬레이션 결과는 상호 미세한 오차 범위 내에서 모형의 예측 데이타와 일치하는 결과를 보여 분석 모형의 타당성을 입증하였다.

키워드 : 팻트리 네트워크, 버퍼, 정상상태 처리율, 네트워크 지연시간, 해석, 시뮬레이션

Abstract In this paper, a performance evaluation model of the Fat-tree Network 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 network. 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 then 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 Fat-tree networks that uses the multiple buffered crossbar switches. Less than 2% differences between analysis and simulation results are observed.

Key words : Fat-tree Network, Buffer, Throughput, Delay, Analysis, Simulation

1. 서 론

Fat-tree 네트워크[1]는 넓은 대역폭, 구조적 유연성, 그리고 스위치 고장적응 특성 등의 장점으로 인해 Think Machine CM-5[2,3] 및 Meiko 슈퍼컴퓨터 CS-2[4], 그리고 Kendall Square Research KSR-1[5] 등의 다양한 대규모 고성능 병렬 컴퓨터의 상호 연결망으로 널리 사용되고 있으며, 최근 컴퓨터 통신기술의 발전과 함께 네트워크 스위칭 기술로 활용되고 있다. Fat-tree 네트워크는 일반 트리 네트워크와는 달리 복수 루트들을 가지는 트리 형태로 구성됨으로써, 말단노

† 정 회 원 : 울산대학교 전기전자및정보시스템공학부
shintaezi@bcline.com
†† 종신회원 : 울산대학교 전기전자및정보시스템공학부 교수
mkyang@mail.ulsan.ac.kr
논문접수 : 2002년 12월 24일
심사완료 : 2003년 3월 26일

드로부터 상위 노드로 갈수록 채널의 대역폭을 증가시켜서 병목현상을 완화시킴과 동시에 특정 노드 고장 시 우회 경로를 제공하여 네트워크 고장적응 기능을 제공한다.

Fat-tree 네트워크는 스위치 내부 구조상 두 개 이상의 데이타들이 하나의 이동경로를 통해 진행하고자 하는 경우가 빈번히 발생하고, 이때 데이타의 충돌현상이 발생한다. 데이타 충돌 현상은 네트워크 성능저하를 유발할 뿐만 아니라 네트워크 신뢰도에도 큰 영향을 미치게 된다. 이러한 네트워크 내부의 데이타 충돌현상을 막고 성능 향상을 위한 방법으로 채널 대역폭을 증가시킨 다양한 변형된 fat-tree topology연구[6,7], 경로 배정 알고리즘에 관한 연구[8,9], 그리고 스위치 소자에 버퍼를 장착[10,11]하는 등의 연구가 진행되고 있다. 이들 가운데 스위치 소자에 버퍼를 장착하는 기법은 데이타 충돌로 인하여 소실될 데이타 패킷을 버퍼의 여유공간에 저장함으로써, 네트워크 내부의 데이타 충돌로 인한 데이타 손실을 막고, 나아가서 네트워크의 성능을 증가시키는 방법으로 널리 알려져 있다.

Alunweiri[10] 등은 buffered fat-tree ATM 스위치를 제안하고, 성능을 분석하였다. 제안된 네트워크 모형은 출력 버퍼를 장착한 스위치들로 구성된 fat-tree의 버퍼 사용률을 최대화하기 위하여 스위치의 내부에 concentrator를 장착시키고, 스위치 내부의 모든 데이타 패킷들이 버퍼를 공유하도록 하였다. 하지만 제안된 분석 모형의 네트워크 내부 라우팅 형태가 최상위 노드로부터 말단노드로 데이타를 전달하는 구조로 단순화되어 분석결과의 신뢰도를 저하시키고 있다. Kim, Kwon[11] 등은 실질적 fat-tree 네트워크를 제안하고 분석하였다. 제안된 분석 모형은 입력 버퍼를 장착한 $c \times c$ 스위칭 소자로 구성된 fat-tree 네트워크를 대상으로 설계되어 BMIN(Bidirectional MIN) 형태의 fat-tree 네트워크를 제외한 $a \times b$ 스위치로 구성된 일반적인 fat-tree 네트워크의 성능분석으로의 확대적용이 어렵다.

본 논문에서는 버퍼를 장착한 양 방향성 $a \times b$ 스위치들로 구성된 fat-tree 네트워크의 성능 분석 기법을 제안하고, 분석 모형의 타당성을 검증하였다. 스위치에 버퍼를 장착하여 네트워크 내부의 데이타 충돌로 인한 손실을 줄이는 방법은 이미 잘 알려져 있으나, 이들 성능 분석은 비교적 제한된 범위 내에서 시행되고 있다. 본 논문에서는 이를 보완하여 Leiserson의 이론적 fat tree 망 조건을 만족시키는 $a \times b$ crossbar 스위치로 구성된 fat-tree 네트워크 성능분석을 위한 새로운 수학적 기법을 제시하였다.

먼저 신태지와 양명국[12]에 의해 제안된 $a \times a$ crossbar 스위치 성능분석 기법을 확장하여, 양방향 $a \times b$ 스위치 분석 모형을 설정하고, 이를 fat-tree 망의 경로 배정 알고리즘에 적용하여 네트워크 성능을 분석하였다. 분석모형의 신뢰성 검증을 위하여 시행된 시뮬레이션 결과는 분석기법에 의해 얻어진 결과와 미세한 오차 범위 내에서 일치하여 분석모형의 우수성을 입증하였다. 제안된 분석 모형은 fat-tree 네트워크뿐만 아니라, BMIN(bidirectional MIN)에도 적용 가능하다.

본 논문의 구성은 다음과 같다. 먼저 2절에서는 fat-tree 네트워크의 일반적인 구조를 설명하고, 경로 배정 형태를 기술하였다. 3절에서는 fat-tree 네트워크 내부 버퍼를 장착한 $a \times b$ 스위치 소자의 성능분석 모형을 제시하고, 이를 토대로 fat-tree 네트워크의 성능을 분석하였다. 끝으로 마지막 절에는 본 연구의 성과와 결과를 요약·기술하였다.

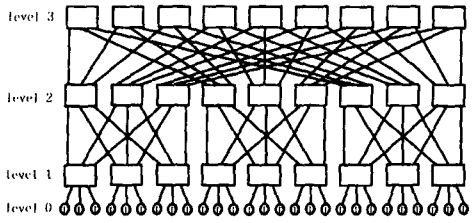
2. Fat-tree 네트워크

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

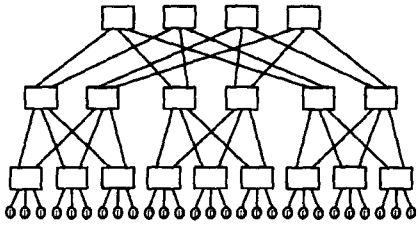
2.1 네트워크 구조

Fat-tree 네트워크는 스위치 노드의 크기와 말단 노드의 수에 따라 다양한 구성을 가진다. 그림 1은 27개의 말단 노드를 가지는 두 가지 형태의 일반적인 fat-tree 네트워크의 구성을 보여주고 있다. 그림 1의 (a)는 네트워크의 높이는 3이고 각 스위치 노드는 3개의 child 포트와 3개의 parent 포트를 가지고, 그림 1의 (b)는 네트워크 높이는 3이고 각 스위치 노드는 3개의 child 포트와 2개의 parent 포트를 가지고 있다. 일반적으로 $a \times b$ 스위치로 구성된 높이가 h인 fat-tree 네트워크는 FT(h, a, b)로 표현한다. Fat-tree 네트워크 FT(h, a, b)는 레벨 0에서 a^h개의 말단 노드(processor)와 상위 각 레벨 l에서 $a^h\ ^l b^{l-1}$개의 스위치 노드들로 구성된다.

네트워크 내부 임의 스위치 노드는 위치에 따라 S(l, x)로 나타낸다. 여기서, l은 스위치 노드가 위치한 레벨, 즉 말단 노드로부터의 높이를 나타내고, x는 같은 레벨의 좌로부터 우로 계수하여 얻어진 해당 스위치 노드의 위치를 나타낸다. 레벨 l의 스위치 노드 S(l, x)의



(a) FT(3, 3, 3)의 구성



(b) FT(3, 3, 2)의 구성

그림 1 Fat-tree 네트워크의 구성

parent 포트 P_p는 레벨 (l+1)의 스위치 노드 S(l+1, y)의 child 포트 P_c와 연결된다. 여기서, $0 \le p < b$, $0 \le c = \lfloor \frac{x \bmod (a \times b^{l-1})}{b^{l-1}} \rfloor < a$, 그리고 $y = \lfloor \frac{x}{a \times b^{l-1}} \rfloor \times b^l + (x \bmod b^{l-1}) \times b + p$이다.

반면에 S(l, x)의 child 포트 P_c는 노드 S($l-1$, w)의 parent 포트 P_p와 연결된다. 여기서, $0 \le c < a$, $0 \le p = x \bmod b < b$, $w = \lfloor \frac{x}{b^{l-1}} \rfloor \times a \times b^{l-2} + \lfloor \frac{x \bmod b^{l-1}}{b} \rfloor + c \times b^{l-2}$이다. 이때, b=1이면 a-ary 트리가 되고, a=b인 경우 BMIN(bidirectional MIN)과 같은 구조를 가지게 된다.

2.2 Fat tree 네트워크 내부 스위치에서의 라우팅 형태

Fat tree를 구성하는 각 스위치에서 데이타의 라우팅 형태는 그림 2와 같이 데이타 패킷이 스위치의 어느 포트로 유입되어 어느 포트로 출력되는가에 따라 세 가지 형태로 나타난다. 첫 번째는 스위치의 parent 포트로 유입되어 a개의 child 포트로 다운 라우팅하는 경우이고, 두 번째는 스위치의 child 포트로 유입되어 b개의 parent 포트로 업 라우팅하는 경우, 세 번째는 a-1개의 다른 child 포트로 회귀 라우팅을 하는 경우이다. 레벨 0의 processor P(0, i)에서 생성된 데이타 패킷은 목적지 P(0, d)를 포함하는 최소 sub-fat 트리의 루트까지, parent 포트로 업 라우팅한다. 데이타 패킷이 업 라우팅하는 경우 fat tree 네트워크가 다중 루트를 가짐으로,

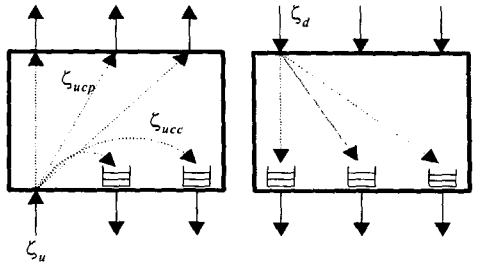


그림 2 스위치 내에서의 데이타 이동 패턴

스위치의 parent 포트 b개 중 어느 포트로 출력되어도 원하는 목적지에 도달할 수 있다. 따라서 임의 선택된 parent 포트가 이미 다른 데이타 패킷에 의해 사용 중이라면 다른 유용한 포트를 통해 출력된다. 그러나, b개의 모든 포트가 다른 데이타 패킷들에 의해 이미 사용 중이라면 해당 데이타 패킷은 탈락하게된다.

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

3. Fat-tree 네트워크의 성능 분석 및 평가

3.1 Fat-tree 네트워크의 성능 분석

3.1.1 네트워크 환경에 대한 일반적인 가정

Buffered fat-tree 네트워크의 분석 모형 개발과 시뮬레이션을 위해 본 논문에 적용된 일반적인 가정을 정리하면 다음과 같다.

- 네트워크는 스위치 클럭 싸이클, Δt,에 따라 동기적으로 동작한다.
- 스위치에 장착된 버퍼는 스위치의 child 포트의 출력 단에 위치하고, 버퍼 공간 하나는 한 개의 데이타 패킷을 수용할 수 있다.
- 데이타 패킷은 네트워크 입력 단의 각 소스 노드에서 같은 확률로 발생한다. 네트워크 내부 임의 레벨 l에 위치한 스위치 입력 단으로 데이타가 유입될 확

률은 $\zeta_{u,\ level\ 1}$라 한다. 따라서 매 싸이클마다 네트워크 각 입력 단에 한 개씩의 데이타 패킷이 유입될 경우, $\zeta_{u,\ level\ 1}$는 1이 된다.

- 네트워크 입력 단으로 유입되는 데이타 패킷의 네트워크 최종 출력 단 행선지는 무작위 선택 방식으로 주어진다.
- 데이타 충돌 발생시 무작위 중재 방식에 의거 데이타 처리 우선 순위를 결정한다.

본 절에 기술한 가정은 기존의 네트워크 성능 평가 연구에 보편적으로 적용되고 있다.

3.1.2 스위치 내부에서의 데이타 이동 패턴

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

3.1절의 가정에 의거, 네트워크 입력 단에 처음 데이타 패킷이 유입될 때 최종 출력 단 행선지가 무작위 선택 방식에 의해 주어짐으로, 임의 스위치 입력 단에 도착한 데이타 패킷이 어느 출력 단으로 향하게 되는가는 다음과 같은 데이타 진행 확률 분석으로 수식화할 수 있다.

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

2.1절에 보인바와 같이 fat-tree 네트워크 구조상 FT(h, a, b)의 임의의 레벨 l에 위치한 스위치 노드는 a개의 child 포트를 통하여 스위치 노드 또는 말단 노드와 연결된다. 따라서 재귀적 연결구조를 갖는 fat-tree FT(l, a, b)는 a^l개의 말단 노드를 포함한다. 또한 최상위 스위치 노드는 FT(h, a, b)의 루트 노드로 전체 말단 노드 a^h개의 말단 노드를 포함하게 된다. 따라서, 레벨 l의 임의 스위치에서 데이타 패킷이 회귀 라우팅할 경우는 현 스위치 노드가 child 포트로 유입된 데이타 패킷의 최종 행선지를 포함하는 sub fat-tree의 루트 노드가 되는 경우이다. 이 경우 데이타 패킷은 자신이 유입된 child 포트로 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 l의 임의 스위치 노드의 child 포트 입력 단으로 유입된 데이타 패킷이 유입된 child 포트를 제외한 임의 child 포트 출력 단으로 지향할 확률 즉, 회귀 라우팅할 확률, $\zeta_{ucc,\ level\ l}$은 다음과 같이 구할 수 있다.

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

식 (1)에서 분모항 $(a^h - a^{l-1})$은 해당 스위치 child 포트 입력 단으로 유입된 데이타 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 (1)은 데이타 패킷의 행선지가 무작위 방식으로 주어지는 가정 하에, 레벨 l의 임의 스위치 child 포트에 도달한 데이타 패킷의 최종 행선지가 해당 스위치를 sub-root로 하는 말단 노드에 존재할 확률을 나타낸다. 즉, 식 (1)은 해당 데이타 패킷의 최종 행선지가 해당 스위치내에서 회귀 라우팅 하여야만 도달할 수 있는 $(a^l - a^{l-1})$개의 말단 노드들 중에 존재할 확률을 수식화한 것이다.

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

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

와 같이 나타낼 수 있다. 여기서, $l < h$이다. 식 (2)에서 분자항 $(a^h - a^l)$은 해당 스위치 child 포트 입력 단으로 유입된 데이타 패킷이 parent 포트를 통해야만 도달할 수 있는 모든 말단 노드의 수를 나타낸다. 식 (2)는 해당 데이타 패킷의 최종 행선지가 상위 레벨의 스위치로 이동하여야만 도달할 수 있는 $(a^h - a^l)$개의 말단 노드에 존재할 확률을 수식화한 것이다.

데이타 패킷이 스위치의 parent 포트를 지향할 경우, parent 포트 b개 중 임의의 포트를 선택해서 출력하게 되고, 만약 선택된 포트에서 충돌이 발생할 경우, 다른 가용한 parent 포트를 선택해서 출력하게 된다. 일반적으로, parent 포트를 통해 데이타 패킷이 출력될 경우, 다중 경로의 선정이 가능함으로 parent 포트에는 버퍼를 장착하지 않게 된다. 따라서, 만일 모든 parent 포트가 사용 중이라면, 해당 데이타 패킷은 제거될 수 있다. 네트워크 구조상 parent 포트를 통과하여 나온 데이타 패킷들은 다음 레벨의 스위치 S(l+1, y)의 child 포트 입력으로 유입되게 된다. 따라서, 다음 레벨 l+1의 임의

스위치 S(l+1, y)의 child 포트로 데이타 패킷이 유입될 확률, $\zeta_{u,\ level\ l+1}$,은

$$\zeta_{u,\ level\ l+1} = \sum_{r=1}^{a} \left\{ {}_aC_r(\zeta_{ucp,\ level\ l})^r(1-\zeta_{ucp,\ level\ l})^{a-r} \times(\frac{r}{b})^{\dagger}\right\} \quad (3)$$

로 계산된다. 식 (3)은 먼저 레벨 l의 임의 스위치 child 입력 포트에 도달한 데이타 패킷들 중 r개의 데이타 패킷이 parent 출력 포트로 지향할 확률을 구하고, 이를 토대로 해당 스위치의 임의 parent 출력 포트로 데이타 패킷이 출력될 확률을 계산하고 있다. 식 (3)의 마지막 곱셈항 $(\frac{r}{b})^{\dagger}$는 parent 출력 포트로 지향하는 r개 데이타 패킷 중 어느 하나가 임의 parent 출력 포트를 통하여 출력될 가능성을 나타내고 있다. 여기서 $r \geq b$ 이면 모든 parent 출력 포트을 통하여 데이타가 출력되어 $(\frac{r}{b})^{\dagger}$는 1로 계산된다. 따라서, 네트워크 입력 단으로 데이타 패킷이 유입될 확률, $\zeta_{u,\ level\ 1}$,이 주어지면 레벨 1에서부터 루트 노드까지 식 (1), (2), (3)을 레벨 별로 반복 계산하여 각 레벨의 스위치 노드에 데이타 패킷이 유입될 확률, 업 라우팅할 확률, 그리고 회귀 라우팅할 확률 등을 계산할 수 있다.

일단 루트 스위치에 도달한 데이타 패킷들은 모두 회귀하여 다운 라우팅을 시작하게 된다. 이 경우 임의 루트 스위치의 child 포트 출력 단자로 r개의 데이타 패킷이 지향할 확률, $P(\hbar_c = r)_h$,을 계산하면 다음과 같이 주어진다.

$$P(\hbar_c = r)_h = {}_{a-1}C_r \times \left(\frac{\zeta_{ucc,\ level\ h}}{a-1}\right)^r \times \left(1-\frac{\zeta_{ucc,\ level\ h}}{a-1}\right)^{a-r-1} \quad (4)$$

루트 스위치의 임의 child 포트로 도착한 데이타 패킷은 자신이 유입된 child 포트를 제외한 나머지 (a-1)개의 다른 child 포트로 지향하게 된다. 따라서 해당 스위치의 특정 child 포트 출력 단으로 데이타 패킷이 지향할 확률은 $\left(\frac{\zeta_{ucc,\ level\ h}}{a-1}\right)$가 되고, 지향하지 않을 확률은 $\left(1-\frac{\zeta_{ucc,\ level\ h}}{a-1}\right)$이 된다. 이를 이용하여 루트 스위치의 임의 child 포트 출력 단을 지향하는 데이타 패킷의 수가 r개일 확률은 식 (4)와 같이 나타낼 수 있다. 식 (4)의 확률과 같이 루트 노드에 도착하여 다운 라우팅 되는 데이타 패킷의 경우 child 포트에서 충돌 발생 시, 다중 경로를 설정할 수 없으므로 데이타 패킷의 유실을 막기 위해 child 포트에 버퍼를 장착하게 된다. 따라서, 버퍼의 유효 공간이 있는 한 데이타 패킷의 유실은 일어나지 않게 된다. 그러므로, 식 (4)의 확률로 루트 노드의 스위치에서 child 포트로 회귀한 데이타 패킷은 버퍼에 저장되었다가 출력되거나 또는 바로 출력되어 (h-1) 레벨의 임의 스위치의 parent 포트로 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 데이타 패킷이 지향할 경우는 먼저 해당 스위치 상위 레벨로부터 데이타가 다운 라우팅 하여 parent 포트 입력 단으로 입력된 경우와 해당 스위치 child 포트 입력 단에 도착된 데이타 패킷이 회귀 라우팅을 시작한 경우이다. 따라서, a개의 child 포트와 b개의 parent 포트를 가지는 스위치에서 임의의 child 포트 출력 단으로 r개의 데이타 패킷이 지향할 확률, $P(\hbar_c = r)_l$,은 데이타 패킷이 스위치 내부에서 회귀할 확률, $\zeta_{ucc,\ level\ l}$,과 parent 포트로부터 다운 라우팅할 확률, $\zeta_{d,\ level\ l}$,을 이용하여 다음과 같이 나타낼 수 있다.

$$P(\hbar_c = r)_l = \sum_{\omega=0}^{r}\left\{ {}_{a-1}C_{\omega} \times \left(\frac{\zeta_{ucc,\ level\ l}}{a-1}\right)^{\omega} \times \left(1-\frac{\zeta_{ucc,\ level\ l}}{a-1}\right)^{a-\omega-1} \times {}_hC_{r-\omega} \times \left(\frac{\zeta_{d,\ level\ l}}{a}\right)^{r-\omega} \times \left(1-\frac{\zeta_{d,\ level\ l}}{a}\right)^{b-r+\omega}\right\} \quad (5)$$

데이타 패킷이 스위치 내부에서 회귀하는 경우; 식 (4)와 같이 자신을 제외한 (a-1)개의 child 포트 중 하나를 지향하게되고, parent 포트 입력 단으로 유입된 데이타 패킷의 경우; 최종 행선지에 따라 a개의 child 포트 중 하나를 지향하게 된다. 식 (5)는 임의 child 포트 출력 단으로 회귀 라우팅 하는 데이타 패킷과 다운 라우팅하는 데이타 패킷의 수를 합해서 모두 r개 데이타 패킷이 지향할 확률을 수식화한 것이다.

식 (5)에서 상위 레벨로부터 데이타 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 포트로 데이타 패킷이 출력 확률, $P(D_c=1)_l$,은 현재 레벨 스위치의 parent 포트로 데이타 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 싸이클 ($j$-1)에 레벨 l에 있는 스위치의 임

의 child 포트 출력 단, D_c,로 데이타 패킷이 출력될 확률, $P(D_c=1)_{l,\ cycle\ j}$,을 살펴보면, 해당 출력 단 버퍼가 싸이클 $(j-1)$에 데이타 패킷을 저장하고 있는 경우, 혹은 싸이클 $(j-1)$에 스위치의 child 포트 입력 단으로 새로이 유입된 데이타 패킷이 해당 출력 단으로 지향할 경우이다. 반대로 스위치 출력 단으로 데이타 패킷이 출력되지 않는 경우는 싸이클 $(j-1)$에 해당 출력 단 버퍼가 데이타 패킷을 저장하지 않은 상태에서, 싸이클 j에 해당 출력 단으로 지향하는 데이타 패킷이 없을 경우이다. 이를 수식화하여 레벨 l에 위치한 임의 스위치 child 포트로 데이타 패킷이 출력될 확률, $P(D_c=1)_{l,\ cycle\ j}$, 즉 레벨 $(l-1)$의 임의 스위치 parent 포트 입력 단으로 데이타 패킷이 유입될 확률, $\zeta_{d,\ level\ l-1}$, 구하면

$$\zeta_{d,\ level\ l-1} = P(D_c=1)_{l,\ cycle\ j} = 1-P(\hbar=0)_{l,\ cycle\ j}\times P(\varepsilon=0)_{l,\ cycle\ (j-1)} \quad (6)$$

와 같이 계산된다. 여기서, $P(\hbar=0)_{l,\ cycle\ j}$는 식 (5)를 통하여 확률적으로 계산할 수 있으며, $P(\varepsilon=0)_{l,\ cycle\ (j-1)}$는 다음의 3.3절의 확률적 계산 방법을 통하여 얻게 된다.

3.1.3 정상상태 처리율 분석

네트워크 내부 레벨 l에 위치한 임의 $a\times b$ crossbar 스위치 내부 데이타 이동 패턴의 확률적 분석을 토대로, buffered fat-tree 네트워크의 성능 분석을 위하여 사용될 변수는 다음과 같다.

- β : 스위치에 장착된 버퍼가 저장할 수 있는 데이타 패킷 수
- ε : 버퍼에 저장된 데이타 패킷 수
- $P(\varepsilon=k)_l$: 버퍼에 저장된 데이타 패킷 수가 k개일 확률
- $P(D_c=1)_l$: 스위치의 child 포트 출력 단 D_c로 데이타 패킷이 출력될 확률
- $P(D_c=0)_l$: 스위치의 child 포트 출력 단 D_c로 데이타 패킷이 출력되지 않을 확률

네트워크 성능 분석의 두 가지 주요 요소는 네트워크 정상상태 처리율과 네트워크 지연시간이다. 네트워크 정상상태 처리율은 레벨 1의 스위치 출력 단으로 데이타 패킷이 출력될 확률, $P(D_c=1)_{level\ 1}$,을 네트워크 입력 단으로 데이타 패킷이 유입될 확률로 나누어서 식 (7)과 같이 계산된다.

$$ST=\frac{P(D_c=1)_{level\ 1}}{\zeta_{u,\ level\ 1}} \quad (7)$$

임의 싸이클 j에 레벨 l에 있는 스위치의 child 포트 출력 단 D_c로 데이타 패킷이 출력되는 경우를 살펴보면, 먼저 싸이클 $(j-1)$ 종료시 해당 출력 단 버퍼가 데이타 패킷을 저장하고 있는 경우, 혹은 스위치의 child 포트나 parent 포트의 입력 단으로 새로이 유입된 데이타 패킷이 해당 출력 단으로 지향할 경우이다. 반대로 스위치 출력 단 D_c로 데이타 패킷이 출력되지 않는 경우는 싸이클 $(j-1)$에 해당 출력 단 버퍼가 데이타 패킷을 저장하지 않은 상태에서, 해당 출력 단으로 지향하는 데이타 패킷이 없을 경우이다. 따라서 임의 싸이클 j에 레벨 l에 위치한 스위치의 child 포트 출력 단 D_c로 데이타 패킷이 출력되지 않을 확률, $P(D_c=0)_{l,cycle\ j}$,을 구하면

$$P(D_c=0)_{l,cycle\ j}=P(\varepsilon=0)_{l,cycle(j-1)}\times P(\hbar_c=0)_{l,cycle\ j} \quad (8)$$

이 된다. 여기서 $j\geq b$이다. 또한, 임의 싸이클 j에 스위치의 child 포트 출력 단 D_c로 데이타 패킷이 출력될 확률, $P(D_c=1)_{l,cycle\ j}$,은

$$P(D_c=1)_{l,cycle\ j}=1-P(D_c=0)_{l,cycle\ j} = 1-\{P(\varepsilon=0)_{l,cycle(j-1)}\times P(\hbar_c=0)_{l,cycle\ j}\} \quad (9)$$

로 계산된다. 식 (9)에서 $P(\hbar_c=0)_{l,cycle\ (j-1)}$은 식 (5)에서 얻을 수 있고, child 포트 D_c의 버퍼가 싸이클 $(j-1)$ 종료 시점에 비어있을 확률, $P(\varepsilon=0)_{l,cycle(j-1)}$, 은 다음과 같이 분석된다.

① 싸이클 $(j-2)$ 종료 시 해당 출력 단 버퍼에 저장된 데이타 패킷의 수가 하나이고, 싸이클 $(j-1)$에 해당 출력 단으로 향하는 데이타 패킷이 없는 경우

② 싸이클 $(j-2)$ 종료 시 해당 출력 단 버퍼에 저장된 데이타 패킷이 없고, 싸이클 $(j-1)$에 해당 출력 단으로 향하는 데이타 패킷이 하나인 경우

③ 싸이클 $(j-2)$ 종료 시 해당 출력 단 버퍼에 저장된 데이타 패킷이 없고, 싸이클 $(j-1)$에 해당 출력 단으로 향하는 데이타 패킷이 없는 경우

따라서, 임의 싸이클 $(j-1)$에 버퍼에 저장된 데이타 패킷의 수가 0일 확률, $P(\varepsilon=0)_{l,cycle(j-1)}$,은

$$P(\varepsilon=0)_{l,cycle(j-1)}= P(\varepsilon=1)_{l,cycle(j-2)}\times P(\hbar_c=0)_{l,cycle(j-1)} + P(\varepsilon=0)_{l,cycle(j-2)}\times P(\hbar_c=1)_{l,cycle(j-1)} + P(\varepsilon=0)_{l,cycle(j-2)}\times P(\hbar_c=0)_{l,cycle(j-1)} \quad (10)$$

로 계산된다. 식 (10)의 $P(\varepsilon=1)_{l,cycle(j-2)}$는 싸이클 $(j-2)$ 종료 시 버퍼에 1개의 데이타 패킷이 저장될 확률로 $P(\varepsilon=0)_{l,cycle(j-1)}$ 분석과 유사한 과정을 거쳐 확률 식으로 표현하면,

$$P(\varepsilon=1)_{l,cycle(j-2)}$$
$$= P(\varepsilon=2)_{l,cycle(j-3)} \times P(\hbar_c=0)_{l,cycle(j-2)}$$
$$+ P(\varepsilon=1)_{l,cycle(j-3)} \times P(\hbar_c=1)_{l,cycle(j-2)}$$
$$+ P(\varepsilon=0)_{l,cycle(j-3)} \times P(\hbar_c=2)_{l,cycle(j-2)} \quad (11)$$

와 같다. 같은 방법으로, 식 (11)을 일반화하여 임의 싸이클 ($j-k-1$)에 버퍼에 저장된 데이타 패킷의 수가 k일 확률, $P(\varepsilon=k)_{l,cycle(j-k-1)}$,은

$$P(\varepsilon=k)_{l,cycle(j-k-1)} =$$
$$P(\varepsilon=k+1)_{l,cycle(j-k-2)} \times P(\hbar_c=0)_{l,cycle(j-k-1)}$$
$$+ P(\varepsilon=k)_{l,cycle(j-k-2)} \times P(\hbar_c=1)_{l,cycle(j-k-1)}$$
$$+ P(\varepsilon=k-1)_{l,cycle(j-k-2)} \times P(\hbar_c=2)_{l,cycle(j-k-1)}$$
$$\cdots$$
$$+ P(\varepsilon=k-a-b+2))_{l,cycle(j-k-2)}$$
$$\times P(\hbar_c=a+b-1)_{l,cycle(j-k-1)}$$
$$= \sum_{x=k-a-b+2}^{k+1} P(\varepsilon=x)_{l,cycle(j-k-2)}$$
$$\times P(\hbar_c=k+1-x)_{l,cycle(j-k-1)} \quad (12)$$

이다. 여기서 $1 \le k < \beta$이다. 식 (12)은 싸이클 ($j-k-1$)에 버퍼가 저장하고 있는 데이타 패킷의 수가 k일 경우는 싸이클 ($j-k-2$)에 버퍼에 저장된 데이타 패킷의 수와 싸이클 ($j-k-1$)에 해당 출력 단으로 지향하는 데이타 패킷의 수의 합이 ($k+1$)임을 보여주고 있다. 이때 싸이클 ($j-k-1$) 동안 하나의 데이타 패킷은 다음 스테이지로 이동하고 나머지 k개 데이타 패킷은 버퍼에 저장된다. 마지막으로 싸이클 ($j-\beta-1$)에서 버퍼가 완전히 차 있을 확률, 즉 버퍼에 저장된 데이타 패킷의 수가 β일 확률, $P(\varepsilon=\beta)_{l,cycle(j-\beta-1)}$을 구하면,

$$P(\varepsilon=\beta)_{l,cycle(j-\beta-1)} =$$
$$P(\varepsilon=\beta)_{l,cycle(j-\beta-2)} \times \sum_{y=1}^{a+b-1} P(\hbar_c=y)_{l,cycle(j-\beta-1)}$$
$$+ P(\varepsilon=\beta-1)_{l,cycle(j-\beta-2)} \times \sum_{y=2}^{a+b-1} P(\hbar_c=y)_{l,cycle(j-\beta-1)}$$
$$\cdots$$
$$+ P(\varepsilon=\beta-a-b+2)_{l,cycle(j-\beta-2)}$$
$$\times \sum_{y=a+b-1}^{a+b-1} P(\hbar_c=y)_{l,cycle(j-\beta-1)}$$
$$= \sum_{x=\beta-a-b+2}^{\beta} \left\{ P(\varepsilon=x)_{l,cycle(j-\beta-2)} \times \sum_{y=\beta-x+1}^{a+b-1} P(\hbar_c=y)_{l,cycle(j-\beta-1)} \right\} \quad (13)$$

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

식 (10), (11), (12) 그리고 (13)의 식에서, 확률적으로 임의 버퍼가 싸이클 j에 k개의 데이타 패킷을 저장할 확률과 싸이클 ($j+1$)에 k개의 데이타 패킷을 저장할 확률은 같다고 볼 수 있다. 즉, 이들 식에 정상 상태 확률(steady state probability) 개념 적용이 가능하고, $P(\varepsilon=k)_{l,cycle\,j} = P(\varepsilon=k)_{l,cycle(j+1)}$, 그리고 $P(\hbar_c=x)_{l,cycle\,j} = P(\hbar_c=x)_{l,cycle(j+1)}$이 된다. 정상 상태 확률 개념을 이용하여 식 (10)를 다시 쓰면

$$P(\varepsilon=0)_l = P(\varepsilon=1)_l \times P(\hbar_c=0)_l$$
$$+ P(\varepsilon=0)_l \times P(\hbar_c=1)_l$$
$$+ P(\varepsilon=0)_l \times P(\hbar_c=0)_l \quad (14)$$

이 된다. 식 (14)를 정리하여 $P(\varepsilon=1)_l$를 $P(\varepsilon=0)_l$의 식으로 구하면

$$P(\varepsilon=1)_l = P(\varepsilon=0)_l \times \frac{(1-P(\hbar_c=0)_l - P(\hbar_c=1)_l)}{P(\hbar_c=0)_l}$$
$$= P(\varepsilon=0)_l \times \frac{1}{P(\hbar_c=0)_l} \times \sum_{y=2}^{a+b-1} P(\hbar_c=y)_l$$
$$= P(\varepsilon=0)_l \times \Omega_0$$
$$= P(\varepsilon=0)_l \times \Phi_1 \quad (15)$$

이다. 여기서 $\Omega_0 = \frac{1}{P(\hbar_c=0)_l} \times \sum_{y=2}^{a+b-1} P(\hbar_c=y)_l$이고, $\Phi_1 = \Omega_0$, $P(\hbar_c=y)_l$는 식 (5)에서 구할 수 있다. 또한, $P(\varepsilon=1)_l$는 다음과 같이 나타낼 수도 있다.

$$P(\varepsilon=1)_l = P(\varepsilon=0)_l \times \sum_{y=2}^{a+b-1} P(\hbar_c=y)_l$$
$$+ P(\varepsilon=1)_l \times \sum_{y=1}^{a+b-1} P(\hbar_c=y)_l \quad (16)$$

같은 방법으로 식 (11)의 $P(\varepsilon=1)_l$은 다음과 같이 나타내고

$$P(\varepsilon=1)_l = P(\varepsilon=2)_l \times P(\hbar_c=0)_l$$
$$+ P(\varepsilon=1)_l \times P(\hbar_c=1)_l$$
$$+ P(\varepsilon=0)_l \times P(\hbar_c=2)_l \quad (17)$$

와 같이 정리된다.

여기서 $P(\varepsilon=2)_l$는 식 (15)과 식 (16)를 이용하여 다음과 같이 두 가지 형태로 정리할 수 있다.

$$P(\varepsilon=2)_l = P(\varepsilon=0)_l \times \frac{1}{P(\hbar_c=0)_l} \times \sum_{y=3}^{a+b-1} P(\hbar_c=y)_l$$
$$+ P(\varepsilon=1)_l \times \frac{1}{P(\hbar_c=0)_l} \times \sum_{y=2}^{a+b-1} P(\hbar_c=y)_l$$
$$= P(\varepsilon=0)_l \times \Omega_1 + P(\varepsilon=1)_l \times \Omega_0$$
$$= P(\varepsilon=0)_l \times \{\Omega_1 + \Phi_1 \times \Omega_0\}$$

$$= P(\varepsilon=0)_l \times \Phi_2 \quad (18)$$

또는

$$P(\varepsilon=2)_l = P(\varepsilon=0)_l \times \sum_{y=3}^{a+b-1} P(\hbar_c=y)_l + P(\varepsilon=1)_l \times \sum_{y=2}^{a+b-1} P(\hbar_c=y)_l + P(\varepsilon=2)_l \times \sum_{y=1}^{a+b-1} P(\hbar_c=y)_l \quad (19)$$

여기서

$$\Omega_0 = \frac{1}{P(\hbar_c=0)_l} \times \sum_{y=2}^{a+b-1} P(\hbar_c=y)_l,$$

$$\Omega_1 = \frac{1}{P(\hbar_c=0)_l} \times \sum_{y=3}^{a+b-1} P(\hbar_c=y)_l,$$

$\Phi_1=\Omega_0$ $\Phi_2=\Omega_1+\Phi_1\times\Omega_0$이다. 같은 방법으로 식 (14)~(19)를 일반화하여 버퍼가 임의 싸이클 종료 시 $(k-1)$개의 데이타 패킷을 저장하고 있을 확률, $P(\varepsilon=k-1)_l$은

$$P(\varepsilon=k-1)_l = \sum_{x=k-a-b+1}^{b} P(\varepsilon=x)_l \times P(\hbar_c=k-x)_l \quad (20)$$

로 되고, 이 식으로부터 버퍼가 임의 싸이클 종료시 k개의 데이타 패킷을 저장하고 있을 확률, $P(\varepsilon=k)_l$을 구하면

$$P(\varepsilon=k)_l = P(\varepsilon=0)_l \times \frac{1}{P(\hbar_c=0)_l} \times \sum_{y=k+1}^{a+b-1} P(\hbar_c=y)_l + P(\varepsilon=1)_l \times \frac{1}{P(\hbar_c=0)_l} \times \sum_{y=k}^{a+b-1} P(\hbar_c=y)_l + \cdots + P(\varepsilon=k-1)_l \times \frac{1}{P(\hbar_c=0)_l} \times \sum_{y=2}^{a+b-1} P(\hbar_c=y)_l$$

$$= \sum_{x=0}^{k-1} \left\{ P(\varepsilon=x)_l \times \frac{1}{P(\hbar_c=0)_l} \times \sum_{y=k+1-x}^{a+b-1} P(\hbar_c=y)_l \right\}$$

$$= P(\varepsilon=0)_l \times \Omega_{k-1} + P(\varepsilon=1)_l \times \Omega_{k-2} + \cdots + P(\varepsilon=k-1)_l \times \Omega_0$$

$$= P(\varepsilon=0)_l \times \{\Omega_{k-1} + \Phi_1 \times \Omega_{k-2} + \cdots + \Phi_{k-1} \times \Omega_0\}$$

$$= P(\varepsilon=0)_l \times \Phi_k \quad (21)$$

또는

$$P(\varepsilon=k)_l = \sum_{x=0}^{k} \left\{ P(\varepsilon=x)_l \times \sum_{y=k+1-x}^{a+b-1} P(\hbar_c=y)_l \right\} \quad (22)$$

로 정리할 수 있다. 여기서

$$\Omega_m = \frac{1}{P(\hbar_c=0)_l} \times \sum_{y=m+2}^{a+b-1} P(\hbar_c=y)_l,$$

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

마지막으로, 버퍼에 β개의 데이타 패킷이 저장될 확률은, 즉 버퍼가 완전히 차게 될 확률, $P(\varepsilon=\beta)_l$은 다음과 같이 구할 수 있다.

$$P(\varepsilon=\beta-1)_l = \sum_{x=\beta-a-b+1}^{\beta} P(\varepsilon=x)_l \times P(\hbar_c=\beta-x)_l \quad (23)$$

이고,

$$P(\varepsilon=\beta)_l = \sum_{x=0}^{\beta-1} \left\{ P(\varepsilon=x)_l \times \frac{1}{P(\hbar_c=0)_l} \times \sum_{y=\beta+1-x}^{a+b-1} P(\hbar_c=y)_l \right\}$$

$$= P(\varepsilon=0)_l \times \{\Omega_{\beta-1} + \Phi_1 \times \Omega_{\beta-2} + \cdots + \Phi_{\beta-2} \times \Omega_1 + \Phi_{\beta-1} \times \Omega_0\}$$

$$= P(\varepsilon=0)_l \times \Phi_\beta \quad (24)$$

이다. 여기서

$$\Omega_m = \frac{1}{P(\hbar_c=0)_l} \times \sum_{y=m+2}^{a+b-1} P(\hbar_c=y)_l,$$

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

식 (15), (16), (21), 그리고 (24)식으로부터 임의의 k에 대한 $P(\varepsilon=k)_l$는 $P(\varepsilon=0)_l$와 Ω_m 그리고 Φ_k를 이용하여 계산이 가능하다. 이때 $P(\varepsilon=0)_l$는 다음과 같이 계산할 수 있다. 스위치에 장착한 버퍼의 개수가 β개인 경우 임의의 싸이클 종료 시 버퍼에 저장된 데이타 패킷의 개수는 0에서 β개 중 하나가 된다.

$$\sum_{x=0}^{\beta} P(\varepsilon=x)_l = P(\varepsilon=0)_l \times \sum_{x=0}^{\beta} \Phi_x = 1 \quad (25)$$

이다. 따라서, 정상 상태 처리율 계산의 주요 변수로 정의된 $P(\varepsilon=0)_l$은

$$P(\varepsilon=0)_l = \frac{1}{\sum_{x=0}^{\beta} \Phi_x} \quad (26)$$

로 얻어진다. 여기서 $\Phi_x=\Omega_{x-1}+\sum_{k=0}^{x-2}\Phi_{x-k-1}\times\Omega_k$, $\Omega_m = \frac{1}{P(\hbar_c=0)_l} \times \sum_{y=m+2}^{a+b-1} P(\hbar_c=y)_l$이다.

일단 데이타 패킷들이 각 소스 노드에서 데이타 유입률 $\zeta_{node,l}$로 생성되어 fat tree 네트워크 FT(h, a, b)로 유입되면, 각 레벨별로 식 (1)과 식 (2), (3)을 반복 계산하여 각 레벨에서 데이타 패킷의 유입될 확률과 회귀 라우팅할 확률, 업 라우팅할 확률을 계산할 수 있다. 따라서, 최상위 루트 노드에서의 데이타 패킷의 유입될 확률과 회귀 확률을 이용하여 루트 노드의 child 포트 출력 단으로 데이타 패킷이 출력될 확률, $P(D_l=1)_{down,l}$은 식 (4), (8), (9) 그리고 (26)을 이용하여 계산할 수 있다. Fat tree 네트워크의 구조상 임의 레벨 l에 위치한 스위치의 child 포트 출력 단은 하위

레벨의 임의 스위치의 parent 포트 입력 단으로 연결됨으로, 레벨 l의 스위치 child 포트 출력은 레벨 $(l-1)$의 임의 스위치 parent 포트의 입력이 된다. 즉, $P(D_c=1)_l \equiv \zeta_{d,\ level\ l-1}$이 된다. 따라서 레벨 h의 child 포트 출력 단으로 데이타 패킷이 출력될 확률, $P(D_c=1)_h$,은 다음 레벨 $(h-1)$의 스위치의 parent 포트 입력 단으로 데이타 패킷이 유입될 확률, $\zeta_{d,\ level\ h-1}$,이 된다. 이와 같은 과정을 반복하여 패킷의 목적지에 따라 각 레벨의 스위치들을 통과하여 목적지 노드의 child 포트의 데이타 패킷이 출력될 확률, $P(D_c=1)_1$,을 구하게 된다.

네트워크 정상상태 처리율은 레벨 1의 스위치 출력 단으로 데이타 패킷이 출력될 확률, $P(D_c=1)_1$,을 네트워크 입력 단으로 데이타 패킷이 유입될 확률, $\zeta_{u\ level\ 1}$, 을 나누어서 식 (7)과 같이 계산된다.

3.1.4 네트워크 지연시간 분석

네트워크 성능 평가에 있어 정상 상태 처리율과 함께, 또 다른 주요 평가 지표로 네트워크 지연시간(Network Delay, T)을 들 수 있다. 임의 데이타 패킷이 네트워크 입력 단에 유입된 후, 각 레벨의 스위치를 지나 최종 출력 단을 통과하기까지 소요되는 스위치 클럭의 평균 개수로 측정되는 네트워크 지연시간은 데이타 패킷 이동 경로와 네트워크 트래픽에 따라 결정된다.

Fat tree의 경우 데이타 패킷의 목적지에 따라 이동 경로가 결정되고, 이동 경로에 따라 네트워크 지연 시간이 달라진다. 따라서, 네트워크를 통과하는 데이타 패킷의 평균 지연시간은 구체적으로 어느 레벨의 스위치에서 회귀 라우팅을 시작하는가 그리고, 회귀 라우팅이나 다운 라우팅 시 child 포트를 통과할 때 버퍼의 어느 위치에 유입되는가에 따라 결정된다. 임의 레벨 l에 위치한 스위치의 child 포트로 성공적으로 출력되는 데이타 패킷이 네트워크에 머무른 평균지연시간, $\tau_{s,\ level\ l}$,은 임의 데이타 패킷이 네트워크에 유입된 후 해당 스위치 입력 단에 도달하기까지의 시간, $\tau_{i,\ level\ l}$,과 해당 스위치 체류 시간, $\tau_{b,\ level\ l}$,을 합하여

$$\tau_{s,\ level\ l} = \tau_{i,\ level\ l} + \tau_{b,\ level\ l} \quad (27)$$

로 계산된다. 여기서 $\tau_{i,\ level\ l}$은 해당 스위치에서 회귀하는 데이타 패킷이 해당 스위치에 도달하기까지 소요된 평균 네트워크 지연시간 혹은 상위 레벨에서 다운 라우팅한 데이타 패킷이 해당 스위치 parent 포트 입력 단에 도달하기까지 소요된 평균 네트워크 지연시간 등을 이용하여 다음과 같이 계산된다.

$$\tau_{i,\ level\ l} = \frac{\zeta_{ucc,\ level\ l} \times (l-1) \times \Delta t + \zeta_{d,\ level\ l} \times \tau_{s,\ level\ (l+1)}}{\zeta_{ucc,\ level\ l} + \zeta_{d,\ level\ l}} \quad (28)$$

식 (28)에서 분자항의 첫 번째 곱셈 항은 네트워크 레벨 l까지 업 라우팅한 데이타 패킷이 해당 스위치에서 다운 라우팅을 시작할 경우 데이타 패킷이 해당 스위치에 도달하기까지 네트워크에 머무른 시간을 계산한다. 여기서 각 스위치 노드의 parent 포트에는 버퍼가 장착되지 않으므로, 업 라우팅에 성공한 데이타 패킷은 레벨 당 Δt의 지연시간을 가진다. 따라서 임의 레벨 l까지 업 라우팅하여 올라온 데이타 패킷은 $(l-1)\times\Delta t$ 만큼의 시간 동안 네트워크에 머무른 것이 된다. 또한 식 (28) 분자항의 두 번째 곱셈 항은 상위 레벨에서 다운 라우팅하여 해당 스위치에 도달한 데이타 패킷이 네트워크에 머무른 시간의 평균값을 순환식으로 표현하고 있다. 여기서 최상위 스위치에 도달한 데이타 패킷의 경우, 식 (28)은 다음과 같이 계산된다.

$$\tau_{i,\ level\ h} = \frac{\zeta_{ucc,\ level\ h} \times (h-1) \times \Delta t}{\zeta_{ucc,\ level\ h}} = (h-1) \times \Delta t \quad (28\text{-}1)$$

일단 식 (28)과 같은 평균 지연시간을 가진 임의 데이타 패킷이 child 포트의 버퍼에 유입되면 해당 버퍼의 데이타 저장 상태, 그리고 함께 도착된 데이타 패킷의 개수에 따라 임의 데이타 패킷은 특정 위치의 버퍼에 저장 되고, 일정 기간 동안 스위치에 머무르게 된다. 문제는 "임의 데이타 패킷이 스위치 버퍼의 어느 위치에 저장되는가?"이다. 일단, 스위치 버퍼 k번째 위치에 저장되면, 이는 해당 스위치에서 $(k+1)\times\Delta t$의 시간만큼 머물고 다음 스위치로 이동하게 된다.

먼저, 데이타 패킷 δ가 해당 스위치의 k번째 버퍼에 저장 될 경우를 살펴보면 다음과 같다 : '이전 싸이클 종료 시 해당 버퍼는 ρ개 데이타 패킷을 저장하고 있는 상태에서, 현 싸이클에 데이타 패킷 δ를 포함한 $(y+1)$개의 새로운 데이타 패킷들이 도착한다. 이들 새로 도착한 데이타 패킷들 가운데 패킷 δ가 $(k-\rho+1)$번째 순서로 버퍼에 저장될 경우, 데이타 패킷 δ는 해당 스위치의 k번째 버퍼에 저장된다.' 여기서, $0\le\rho\le k\le b$, 그리고 $(k-\rho+1)\le(y+1)\le a+b-1$이다.

따라서, 임의 출력 단을 지향한 데이타 패킷이 해당 출력 단에서 탈락하지 않고 해당 버퍼에 저장될 경우, 버퍼를 통과하는데 걸리는 시간, $\tau_{b,\ level\ l}$,은

$$\tau_{b,\ level\ l} = \sum_{\rho=0}^{b} \Big[\sum_{y=k-\rho}^{a+b-2} \Big\{ P(\varepsilon=\rho)_{l,\ cycle(j-1)} \times \sum_{y=k-\rho}^{a+b-2} \frac{1}{y+1} P(h_l=y)^{*}_{l,\ cycle\ j} \Big\} \times (k+1)\Delta t \Big] \quad (29)$$

으로 얻어진다. 여기서, $P(\hbar_c=y)_l^*$는

$$\begin{aligned}P(\hbar_c=y)_l^* = \sum_{\omega=0}^{y}\{ & {}_{a\cdots 2}C_\omega \times \left(\frac{\zeta_{ucc,\ level\ l}}{a-1}\right)^{\omega} \\ & \times \left(1-\frac{\zeta_{ucc,\ level\ l}}{a-1}\right)^{a\cdots\omega\cdot 2} \times {}_bC_{y\ \omega} \times \left(\frac{\zeta_{d,\ level\ l}}{a}\right)^{y-\omega} \\ & \times \left(1-\frac{\zeta_{d,\ level\ l}}{a}\right)^{b\ \ y+\omega} \times P(\hbar_c=1)_a^* \\ & + {}_{a\ \ 1}C_\omega \times \left(\frac{\zeta_{ucc,\ level\ l}}{a-1}\right)^{\omega} \times \left(1-\frac{\zeta_{ucc,\ level\ l}}{a-1}\right)^{a\ \ \omega\ \ 1} \\ & \times {}_{b\ \ 1}C_{y\ \ \omega} \times \left(\frac{\zeta_{d,\ level\ l}}{a}\right)^{y\ \ \omega} \times \left(1-\frac{\zeta_{d,\ level\ l}}{a}\right)^{b\ \ y+\omega\ \ 1} \\ & \times P(\hbar_c=1)_b^*\}\end{aligned}$$

$$P(\hbar_c=1)_a^* = {}_{a\ \ 1}C_1 \times \left(\frac{\zeta_{ucc,\ level\ l}}{a-1}\right)^{1} \times \left(1-\frac{\zeta_{ucc,\ level\ l}}{a-1}\right)^{a\ \ 2} / P(\hbar_c=1)_l$$

$$P(\hbar_c=1)_b^* = {}_bC_1 \times \left(\frac{\zeta_{d,\ level\ l}}{a}\right)^{1} \times \left(1-\frac{\zeta_{d,\ level\ l}}{a}\right)^{b\ \ 1} / P(\hbar_c=1)_l \quad (30)$$

이다. 이때 $P(\hbar_c=1)_a^*$는 데이타 패킷 δ가 회귀라우팅 하는 패킷들 중 하나일 확률이고, $P(\hbar_c=1)_b^*$는 데이타 패킷 δ가 다운라우팅 패킷들 중에 하나일 확률이다. 또한 $P(\hbar_c=y)_l^*$는 임의의 데이타 패킷 δ를 제외한 y개의 데이타 패킷이 해당 출력 단을 지향하는 경우를 수식화 한 것이다. 따라서 식 (29)의 $\sum_{v=k}^{\cdot} \frac{1}{{}_{\rho}\ y+1} \times P(\hbar_c=y)_l^*$는 데이타 패킷 δ를 포함한 $(y+1)$개의 데이타 패킷이 해당 버퍼에 새로 도착되고, 이들 가운데 데이타 패킷 δ가 k번째 버퍼 공간에 저장될 확률을 나타낸다.

네트워크 전체를 성공적으로 통과한 데이타 패킷의 평균 지연시간, $\tau_{s,\ level\ 1}$,은 네트워크의 구조상 루트 노드를 통과한 데이타 패킷의 평균 지연시간이 다음 레벨의 평균 지연시간에 영향을 미치므로, 루트 노드로부터 최하위 레벨로 식 (27), (28), (29)을 반복 계산하여 구할 수 있다.

한편, 일부 데이타 패킷들은 업 라우팅 시 데이타 패킷의 충돌과 다운 라우팅 시 한정된 버퍼 공간으로 인하여 전송 중, 네트워크 내부에서 유실될 수 있다. 이들 중도 유실된 데이타 패킷들은 소정의 "중도 유실 감지 과정"을 거쳐 최초 데이타 패킷이 유입된 입력 단에서 재전송 되게 된다. 그림 3은 네트워크를 성공적으로 통과한 데이타 패킷들의 네트워크 지연 시간과 함께, 전송 과정에서 중도 소실된 데이타 패킷들의 재전송 시간

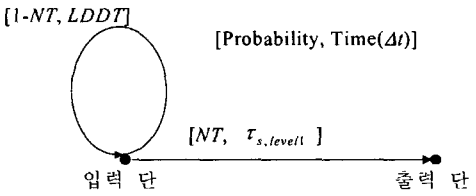


그림 3 네트워크 지연 시간에 관한 상태도

을 고려한 총 네트워크 지연시간에 관한 상태도이다. 여기서, 임의 데이타 패킷 δ가 네트워크를 성공적으로 통과할 확률은 정상상태 처리율(ST)로 볼 수 있고, 이때 네트워크 지연시간, $\tau_{s,\ level\ 1}$,은 식 (27), (28) 그리고 (29)로 구할 수 있다.

반면에, 데이타 패킷 δ가 네트워크 내부에서 유실될 확률은 $(1-ST)$로 계산되고, 이들 중도 유실 데이타 패킷은 중도 유실 감지 시간(Lost Data Detection Time, LDDT) 만큼의 오류 검사 과정을 거쳐 재 전송 된다.

따라서, 임의 데이타 패킷이 전체 네트워크를 통과하는데 걸리는 평균 시간, $\mathcal{T}$는 그림 3으로부터

$$\mathcal{T} = ST\times\tau_{s,\ level\ 1} + (1-ST)\times(LDDT+\mathcal{T}) \quad (31)$$

와 같은 식으로 얻어진다. 식 (29)를 $\mathcal{T}$에 관하여 풀면

$$\mathcal{T} = \tau_{s,\ level\ 1} + \frac{(1-ST)}{ST}\times LDDT \quad (32)$$

과 같이 계산된다. 여기서, ST와 τ_s는 식 (5), (26) 그리고 (29)로부터 구할 수 있고, LDDT는 네트워크 특성에 따라 상수로 주어진다.

3.2 Fat-tree 네트워크의 성능 평가

표 1과 표 2, 그리고 그림 4와 그림 5는 높이가 3이고 3×3 스위치로 구성된 FT(3, 3, 3) 네트워크를 시험 대상으로, 표 3와 표 4, 그림 6과 그림 7은 높이가 3이고 3×2 스위치로 구성된 FT(3, 3, 2) 네트워크를 시험 대상으로 스위치에 장착된 버퍼의 크기에 따른 네트워크 정상 상태 처리율과 지연시간에 관한 분석 결과를 비교한 표와 그래프이다. 시뮬레이션 과정에서는, 초기에 버퍼가 비어 있는 상태에서 네트워크 성능 측정을 피하기 위하여, 충분한 예비 동작시간을 준 후 본격적으로 데이타를 수집하여 처리하였다. 네트워크 입력 단으로 유입되는 데이타 패킷의 목적지는 무작위 방식으로 주어지며, 이때 네트워크로 유입되는 트래픽에 따른 네트워크의 성능 평가를 위하여 유입률을 달리하여 시뮬레이션 하였다. 네트워크 입력 단의 데이타 패킷 유입률이 1.0 인 경우는 매 스위치 싸이클마다 각 입력 단으로

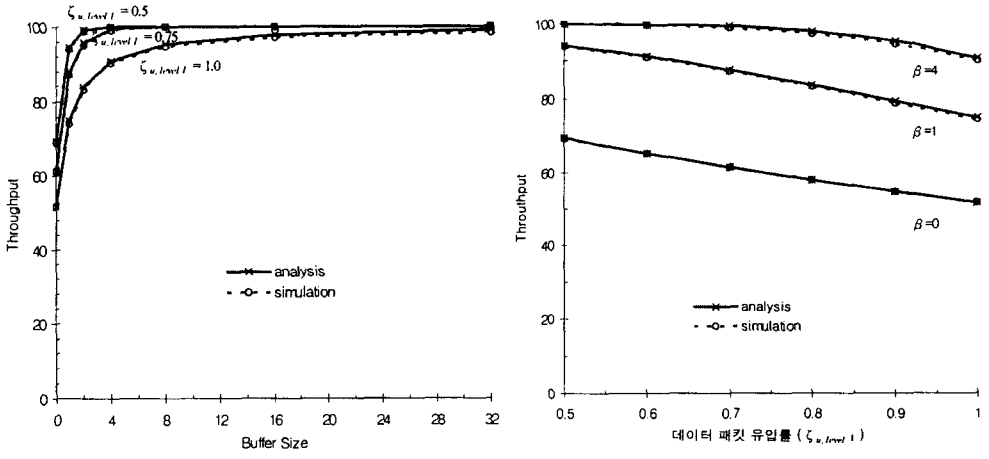
각 각 1개씩의 데이타 패킷이 유입되도록 하였다. 데이타 패킷 유입률 ζ가 1.0 보다 작을 경우 네트워크 입력단으로 유입될 확률이 떨어져 네트워크 내부의 데이타 트래픽이 줄어들게 된다. 예를 들어 ζ가 0.8 인 경우 각 네트워크 입력 단에는 평균적으로 매 10개 스위치 싸이클마다 8개 데이타 패킷이 유입되게 된다.

그림 4와 그림 6의 (a)는 스위치에 장착된 버퍼 개수(β)에 따른 네트워크 정상상태 처리율의 변화와 네트워크로 패킷이 유입될 확률($\zeta_{u,\ level\ 1}$)에 따른 정상상태 처리율을 각각 나타낸 것이다. 그림에서와 같이 버퍼가 1~4개 데이타 패킷을 저장할 수 있을 때까지 극적인 증가 양상을 보이고, 이후 정상상태 처리율이 포화상태에 이르게 됨을 보여주고 있다. 그림 4, 그림 6의 (b)의 경우 같은 수의 버퍼를 장착한 경우 네트워크로 유입되는 데이타 패킷의 유입률이 증가함에 따라 정상상태 처리율이 선형적으로 낮아짐을 보여주고 있다. 또한 그림 4의 FT(3, 3, 3) 네트워크가 그림 6의 FT(3, 3, 2) 네트워크의 정상상태 처리율보다 전반적으로 높은 것으로 나타났다. 이는 FT(3, 3, 2) 네트워크의 경우, 3×2 스위치의 특성상 child 포트에서 parent 포트로의 업 라우팅 과정에서 입·출력 포트 개수의 불균형으로 데이타 패킷의 탈락이 불가피 하게 되고, 이로 인하여 정상상태 처리율의 저하를 가져오는 것으로 조사되었다.

네트워크 지연시간의 경우 스위치에 장착된 버퍼의 크기의 증가와 네트워크으로 데이타 유입률의 증가는 전반적인 네트워크 지연시간의 증가를 초래하는 것으로 나타났다. 그림 5와 그림 7에서 데이타 중도 소실 감지 시간(LDDT)은 최소 소요시간을 $(\beta \times h + h)\Delta t$로 놓고 내

표 1 FT(3, 3, 3)의 성능(정상 상태 처리율)

Buffer size	정상상태 처리율(ST, %)					
	데이터 패킷 입력률($\zeta_{u,\ level\ 1}$)					
	$\zeta_{u,\ level\ 1}$ = 0.5		$\zeta_{u,\ level\ 1}$ = 0.75		$\zeta_{u,\ level\ 1}$ = 1.0	
	해석	시뮬레이션	해석	시뮬레이션	해석	시뮬레이션
0	69.53	69.18	61.36	61.44	51.79	51.72
1	94.30	93.98	87.61	87.08	74.93	74.52
2	98.95	98.67	95.53	95.02	83.74	83.09
4	99.96	99.93	99.34	98.96	90.79	90.19
8	100	100	99.99	99.95	95.34	94.72
16	100	99.98	100	100	97.79	94.29
32	100	100	100	99.99	98.92	98.56



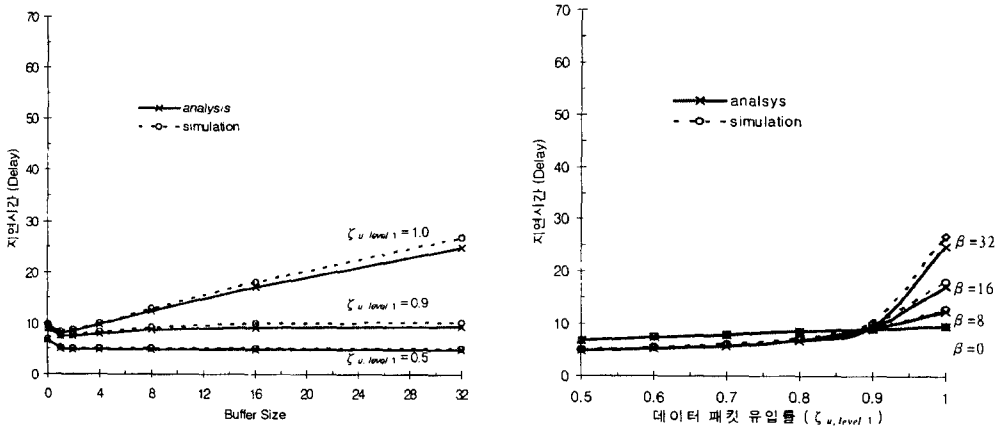
(a) 버퍼 사이즈에 따른 Throughput 변화 (b) 데이타 유입률에 따른 Throughput의 변화

그림 4 버퍼를 장착한 FT(3, 3, 3) 성능 분석 결과와 시뮬레이션 결과의 비교

트워크 지연시간을 구하였다. 여기서 β는 스위치에 장착된 버퍼의 크기이고, 성공적으로 네트워크를 통과한 데이타 패킷이 네트워크에 체류할 수 있는 최대 지연시간은 $(\beta \times h + (h-1))\Delta t$로 계산된다. 이때까지 네트워크 출력 단에 데이타 패킷이 도달하지 못하면 패킷이 중도 유실된 것으로 간주하게 된다. 데이타 중도 유실이 확인되면 바로 입력 단으로 사실이 알려지고, 해당 입력 단에서 재전송 하게 된다. 그림 5의 FT(3, 3, 3) 네트워크의 지연시간은 그림 7의 FT(3, 3, 2) 네트워크의 지연시간보다 낮게 나타나는 것으로 조사되었다. 이는 FT(3, 3, 2) 네트워크 보다 FT(3, 3, 3) 네트워크의 정상 상태 처리율이 전반적으로 높아 데이타 패킷의 유실 확률이 낮기 때문으로 해석할 수 있다. 또한 데이타 유입률이 낮을 경우 버퍼공간이 커지더라도 데이타 패킷이 활용하는 버퍼공간이 적어 스위치 체류 시간이 일정하게 유지된다. 그리고 중도 유실되는 데이타 패킷이 거의 없기 때문에 중도 유실 감지에 소요되는 시간의 증가가 전체 지연시간에 거의 영향을 미치지 못하므로 네트워크 지연 시간은 거의 일정하게 유지되게 된다. 반면에 데이타 유입률이 1에 가까워지면 버퍼 공간이 커지면서 데이타 패킷 유실 확률은 부분적으로 줄 일수 있으나 스위치 체류시간의 증가로 인하여 전체적인 네트워크 지연시간이 증가하는 것으로 나타났다. 또한 그림 7의 (a)는 버퍼 공간이 커짐에 따라 네트워크 지연시간이 선형적으로 증가함을 보여 주고 있다. 이는 다음 두 가지 원인으로 인한 결과로 설명된다. 먼저 각 스위치에 장착된 버퍼공간이 커지면서 데이타 손실은 줄일 수 있으나, 성공적으로 네트워크를 통과한 데이타 패킷의 각

표 2 FT(3, 3, 3)의 성능(지연 시간)

네트웍 지연시간								
Buffer size	$\zeta_{u,\ level\ 1} = 0.7$				$\zeta_{u,\ level\ 1} = 1.0$			
	네트웍을 통과한 패킷의 지연시간 (Δt)		데이터 패킷 탈락 확률(%)		네트웍을 통과한 패킷의 지연시간 (Δt)		데이터 패킷 탈락 확률(%)	
	해석	시뮬레이션	해석	시뮬레이션	해석	시뮬레이션	해석	시뮬레이션
0	4.093	4.117	38.64	38.56	4.028	4.070	48.21	48.28
1	4.890	4.989	12.39	12.92	5.230	5.309	25.07	25.48
2	5.319	5.434	4.47	4.98	6.273	6.311	16.26	16.91
4	5.646	5.864	0.66	1.04	8.017	8.046	9.21	9.81
8	5.739	6.015	0.02	0.05	11.001	11.144	4.66	5.28
16	5.742	6.058	0	0	15.927	16.511	2.21	2.71
32	5.742	6.039	0	0	23.769	25.436	1.08	1.44



(a) 버퍼 사이즈에 따른 네트워크 지연시간 변화 (b) 데이타 유입률에 따른 네트워크 지연시간 변화

그림 5 버퍼를 장착한 FT(3, 3, 3) 성능 분석 결과와 시뮬레이션 결과의 비교

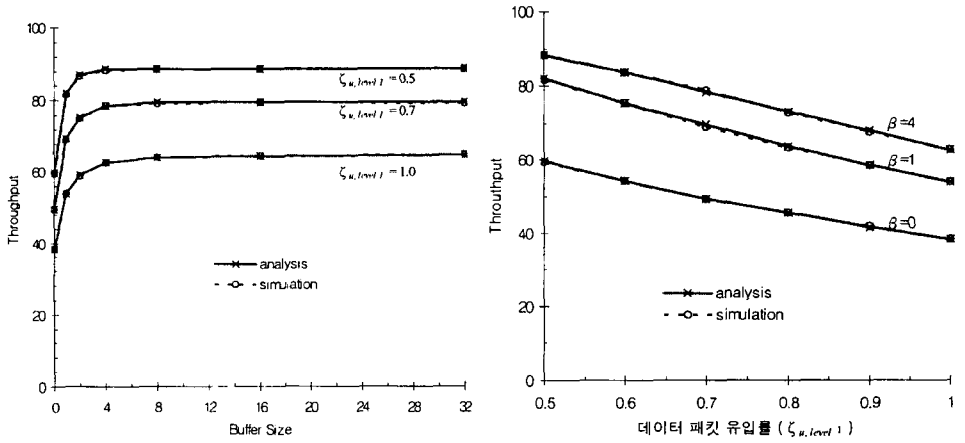
스위치 체류 시간은 증가하게 된다. 두 번째 원인으로 데이타 중도 유실 감지에 소요되는 시간이 버퍼의 크기 (β)의 증가와 함께 커지게 됨을 들 수 있다. 그림 7 (b)의 경우 일정한 개수의 버퍼를 장착한 네트워크에서의 입력부하 증가에 따른 네트워크 지연시간의 변화를 보여준다. 이 경우 네트워크로 데이타 유입률이 증가하면 데이타 패킷의 유실 확률이 커짐으로써 전체 네트워크의 지연시간이 증가하게 됨을 알 수 있다.

4. 결 론

본 논문에서는 스위치 출력 단에 복수 버퍼를 장착한 $a \times b$ 스위치들로 구성된 fat-tree 네트워크 FT(h, a, b)의 성능을 확률식으로 분석하는 새로운 성능 분석 모형을 제안하고, 실효성을 입증하였다. 제안된 분석 기법은 네트워크 스위치 내부에서 데이타 패킷의 이동 상태를 관찰하여 확률식으로 정리하고, 이를 토대로 네트워크 전체의 정상 상태 처리율 및 네트워크 지연시간을 예측한다. 분석 모형의 수립 단계에서 정상상태 확률 개념을 도입하여 간단한 근사화(approximation)를 시도하여 모형의 해석과 확률식 전개를 용이하게 하였다. 또한 본 논문에서는 모형의 이해를 도모하기 위하여 지능형 네트워크 트래픽 제어 및 중도 소실 패킷에 대한 다양한 처리 기능 등 최근 개발되는 스위치 네트워크의 부가기능을 배제하고 수식을 정리하였다. 그러나, 제안된 분석 모형은 이들 다양한 성능 향상 기술이 적용된 네트워크, 그리고 다양한 크기의 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.

표 3 FT(3, 3, 2)의 성능(정상 상태 처리율)

Buffer size	정상상태 처리율(ST, %)					
	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

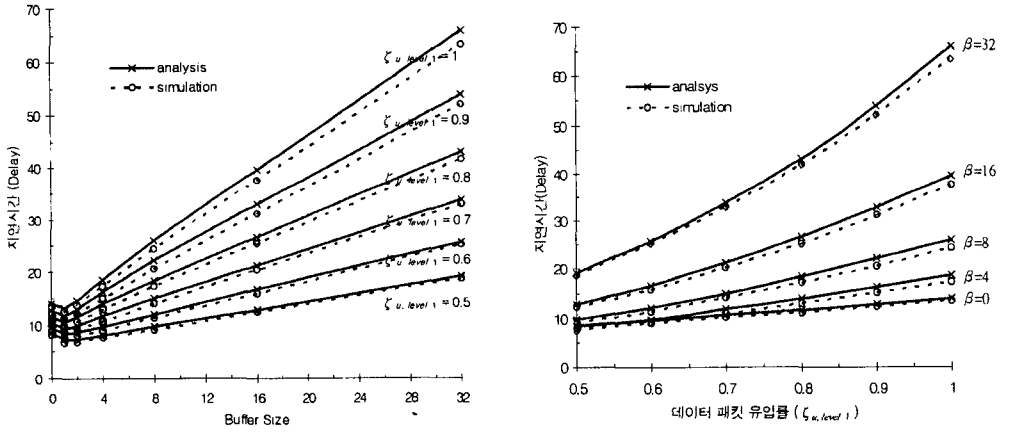


(a) 버퍼 사이즈에 따른 Throughput 변화 (b) 데이타 유입률에 따른 Throughput의 변화

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

표 4 FT(3, 3, 2)의 성능(지연 시간)

네트워 지연시간								
Buffer size	$\zeta_{u,\ level\ 1} = 0.7$				$\zeta_{u,\ level\ 1} = 1.0$			
	네트워을 통과한 패킷의 지연시간 (Δt)		데이터 패킷 탈락 확률(%)		네트워을 통과한 패킷의 지연시간 (Δt)		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) 데이타 유입률에 따른 네트워크 지연시간 변화

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

참 고 문 헌

[1] 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.

[2] C.E. Leiserson, "The network architecture of the connection machine CM 5," 4th *Annual ACM Symp. on Parallel Algo. and Arch.*, pp. 272 285, June 1992.

[3] Thinking Machine Corporation. "The Connection Machine System CM 5," Technical Summary, November 1993.

[4] Klaus E. Schauser and Chris J. Scheiman. "Experiments with active message on the Meiko CS 2," the Proceedings of the 9th *International Parallel Processing Symposium, Santa Barbara,* April, 1995.

[5] S.Frank, J.Rothmie, and H.Burkhardt. "The KSR1 : Bridgeing the gap between shared memory and mpps," *In Proceedings Compcon* '93, San Fran cisco, CA, February 1993.

[6] 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.

[7] 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.

[8] R.I. Greenberg and C.E. Leiserson. "Randomized routing on fat trees," In Silvio Micali, editor, Advances in Computing Research, Book 5:

Randomness and Computation, pages 345-374, JAI Press, Greenwich, CT, 1989.
[9] Ronald I.Greenberg, Lee Guan, "An Improved Analytical Model for Wormhole Routed Networks with Application to Buterfly Fat-trees," Parallel Processing, 1997, Proceedings of the 1997 International Conference on, 1997, Page(s): 44-48.
[10] 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.
[11] 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.
[12] 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.

신 태 지

1998년 울산대학교 전기전자 및 정보시스템 공학부 졸업(학사). 2000년 울산대학교 전기전자 및 정보시스템 공학부 졸업(석사). 2000년~현재 울산대학교 전기 전자 및 정보 시스템 공학부 박사과정 관심분야는 컴퓨터 네트웍, 병렬 처리시스템

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 부교수. 관심분야는 컴퓨터 네트웍, 병렬 처리 시스템, 고장 적응 시스템