버퍼를 장착한 $a \times b$ 스위치들로 구성된 Fat-tree 망의 성능분석

신태지 설춘룡 신종균 양명국

울산대학교 전기전자 정보시스템공학부

shintaezi@bcline.com paulimda@bcline.com love1317@hanmail.net mkyang@mail.ulsan.ac.kr

Evaluation of a Fat-tree Network with Buffered $a \times b$ Switches

Tae-Zi Shin[0] Choon-Ryong Seol Jong-Kyun Shin Myung-Kook Yang

Dept. of Electrical & Computer Engineering University of Ulsan

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. 제안한 성능 예측 모형은 먼저 네트웍 내부 임의 스위치 입력 단에 유입되는 데이터 패킷이 스위치 내부에서 전송되는 유형을 확률적으로 분석하여 수립되었다. 성능분석 모형은 스위치에 장착된 버퍼의 갯수와 무관하게 버퍼를 장착한 $a \times b$ 스위치의 성능, 네트웍 정상상태 처리율(Steady state Throughput, *ST*)과 네트웍 지연시간(Network Delay)을 간단한 확률식으로 구할 수 있다. 제안한 수학적 성능 분석 연구의 실효성 검증을 위하여 병행된 시뮬레이션 결과는 상호 미세한 오차 범위 내에서 모형의 예측 데이터와 일치하는 결과를 보여 분석 모형의 타당성을 입증하였다.

1. 서 론

Fat-tree 네트워크[1]는 넓은 대역폭, 구조적 유연성, 그리고 스위치 고장적응 특성 등의 장점으로 인해 Think Machine CM-5[2], Kendall Square Research KSR-1[3] 등의 다양한 대규모 고성능 병렬 컴퓨터의 상호 연결 망으로 널리 사용되고 있으며, 최근 네트워크 스위칭 기술로 활용되고 있다. Fat-tree 네트워크는 일반 트리 네트워크와는 달리 복수 루트들을 가지는 트리 형태로 구성됨으로써, 말단노드로부터 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 빈번한 데이터의 충돌현상이 발생한다. 데이터 충돌 현상은 네트워크 성능저하를 유발할 뿐 아니라 네트워크 신뢰도에도 큰 영향을 미치게 된다. 이러한 네트워크 내부의 데이터 충돌현상을 막고 성능 향상을 위한 다양한 연구가 진행되고 있다.[4-8] 이들 가운데 스위치 소자에 버퍼를 장착하는 기법은 데이터 충돌로 인하여 소실될 데이터 패킷을 버퍼의 여유공간에 저장하고, 네트워크 내부의 데이터 충돌로 인한 데이터 손실을 막고, 나아가서 네트워크의 성능을 증가시키는 방법으로 널리 알려져 있다.

본 논문에서는 버퍼를 장착한 양 방향성 $a \times b$ 스위치들로 구성된 fat-tree 네트워크의 성능 분석 기법을 제안하고, 분석 모형의 타당성을 검증하였다. 본 논문에서는 Leiserson의 이론적 fat-tree 망 조건을 만족시키는 $a \times b$ crossbar 스위치로 구성된 fat-tree 네트워크 성능분석을 위한 새로운 수학적 기법을 제시하였다.

2. Fat-tree 네트워크의 성능 분석

2.1 스위치 내부에서의 데이터 이동 패턴

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

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

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 확률, $\zeta_{ucc,\ level\ l}$은

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

와 같다. 유사한 방법으로 FT(*h*, *a*, *b*) 레벨 *l*의 임의 스위치 S(*l*, *x*)의

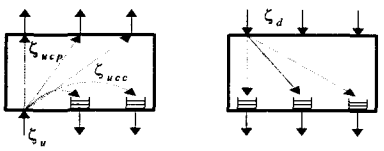

그림 1. 스위치 내에서의 데이터 이동 패턴

child 포트 입력 단으로 유입된 데이터 패킷이 parent 포트 출력 단을 지향하게 될 확률, $\zeta_{ucp,\ level\ l}$은

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

와 같이 나타낼 수 있다. 여기서, $l<h$이다. 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+1의 임의 스위치 S(l+1, y)의 child 포트로 데이터 패킷이 유입될 확률, $\zeta_{u,\ level\ l+1}$은

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

로 계산된다. 여기서 $r \geq b$ 이면 $\left(\frac{r}{b}\right)^{\dagger}$는 1로 계산된다. 따라서, 네트워크 입력 단으로 데이터 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 데이터 패킷이 유입될 확률, 업 라우팅할 확률, 그리고 회귀 라우팅할 확률 등을 계산할 수 있다.

임의 레벨 *l*의 스위치 S(*l*, *x*)의 임의 child 포트 출력 단으로 r개의 데이터 패킷이 지향할 확률, $P(\hbar_c = r)_l$은 데이터 패킷이 스위치 내부에서 회귀할 확률, $\zeta_{ucc,\ level\ l}$과 parent 포트로부터 다운 라우팅 할 확률, $\zeta_{d,\ level\ l}$을 이용하여 다음과 같이 나타낼 수 있다.

$$P(\hbar_c = r)_l = \sum_{w=0}^{r} \left\{ {}_{a-1}C_w \times \left(\frac{\zeta_{ucc,\ level\ l}}{a-1}\right)^w \times \left(1 - \frac{\zeta_{ucc,\ level\ l}}{a-1}\right)^{a-w-1} \right.$$
$$\left. \times\ {}_bC_{r-w} \times \left(\frac{\zeta_{d,\ level\ l}}{a}\right)^{r-w} \times \left(1 - \frac{\zeta_{d,\ level\ l}}{a}\right)^{b-r+w} \right) \qquad (4)$$

임의 레벨 *l*에 위치한 스위치의 child 포트 출력 단은 하위 레벨의 임의 스위치의 parent 포트 입력 단으로 연결되므로, 상위 레벨 스위치의 child 포트로 데이터 패킷이 출력 확률, $P(D_c=1)_l$은 현재 레벨 스위치의 parent 포트로 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 레벨 *l*에 있는 스위치의 임의 child 포트 출력 단, D_c로 데이터 패킷이 출력될 확률, $P(D_c=1)_l$은 2.2절의 확률적 계산을 통하여 얻게된다.

2.2 정상상태 처리율 분석

Buffered fat-tree 네트워크의 성능 분석을 위하여 사용될 변수는 다음과 같다.

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

네트워크 정상상태 처리율은 레벨 1의 스위치 출력 단으로 데이터 패킷이 출력될 확률, $P(D_c=1)_{level\ 1}$,을 이용하여 식 (5)과 같이 계산된다.

$$ST=\frac{P(D_c=1)_{level\ 1}}{\zeta_{u,\ level\ 1}} \tag{5}$$

먼저 임의 레벨 l에 있는 스위치의 child 포트 출력 단 D_c로 데이터 패킷이 출력되지 않는 경우를 살펴보면, 출력 단 버퍼가 데이터 패킷을 저장하지 않은 상태에서, 지향하는 데이터 패킷이 없을 경우이다. 레벨 l에 위치한 스위치의 child 포트 출력 단 D_c로 데이터 패킷이 출력되지 않을 확률, $P(D_c=0)_{l,cycle}$,는

$$P(D_c=0)_l=P(\varepsilon=0)_l\times P(\hbar_c=0)_l \tag{6}$$

이 된다. 따라서 임의 스위치의 child 포트 출력 단 D_c로 데이터 패킷이 출력 될 확률, $P(D_c=1)_l$,은

$$P(D_c=1)_l=1-P(D_c=0)_l=1-\{P(\varepsilon=0)_l\times P(\hbar_c=0)_l\} \tag{7}$$

로 계산된다. 식 (7)에서 $P(\hbar_c=0)_l$은 식 (4)에서 얻을 수 있고, child 포트 D_c의 버퍼가 비어있을 확률, $P(\varepsilon=0)_l$,은

$$\begin{aligned}P(\varepsilon=0)_{l,cycle(j-1)}=&\ P(\varepsilon=1)_{l,cycle(j-2)}\times P(\hbar_c=0)_{l,cycle(j-1)}\\&+P(\varepsilon=0)_{l,cycle(j-2)}\times P(\hbar_c=1)_{l,cycle(j-1)}\\&+P(\varepsilon=0)_{l,cycle(j-2)}\times P(\hbar_c=0)_{l,cycle(j-1)}\end{aligned} \tag{8}$$

이 된다. 식 (8)를 정리하여 $P(\varepsilon=1)_l$를 $P(\varepsilon=0)_l$의 식으로 구하면

$$\begin{aligned}P(\varepsilon=1)_l&=P(\varepsilon=0)_l\times\frac{(1-P(\hbar_c=0)_l-P(\hbar_c=1)_l)}{P(\hbar_c=0)_l}\\&=P(\varepsilon=0)_l\times\frac{1}{P(\hbar_c=0)_l}\times\sum_{y=2}^{a\cdot b-1}P(\hbar_c=y)_l\\&=P(\varepsilon=0)_l\times\Omega_0=P(\varepsilon=0)_l\times\Phi_1\end{aligned} \tag{9}$$

이다. 여기서 $\Omega_0=\frac{1}{P(\hbar_c=0)_l}\times\sum_{y=2}^{a\cdot b-1}P(\hbar_c=y)_l$이고, $\Phi_1=\Omega_0$, $P(\hbar_c=y)_l$는 식 (4)에서 구할 수 있다. 같은 방법으로 식 (9)~(10)를 일반화하여 버퍼가 $(k-1)$개의 데이터 패킷을 저장하고 있을 확률, $P(\varepsilon=k-1)_l$,은

$$P(\varepsilon=k-1)_l=\sum_{x=a\cdot b-1}^{k}P(\varepsilon=x)_l\times P(\hbar_c=k-x)_l \tag{10}$$

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

$$\begin{aligned}P(\varepsilon=k)_l=&\ P(\varepsilon=0)_l\times\frac{1}{P(\hbar_c=0)_l}\times\sum_{y=k+1}^{a\cdot b-1}P(\hbar_c=y)_l\\&+P(\varepsilon=1)_l\times\frac{1}{P(\hbar_c=0)_l}\times\sum_{y=k}^{a\cdot b-1}P(\hbar_c=y)_l\\&\cdots\\&+P(\varepsilon=k-1)_l\times\frac{1}{P(\hbar_c=0)_l}\times\sum_{y=2}^{a\cdot b-1}P(\hbar_c=y)_l\\=&\sum_{x=0}^{k-1}\left\{P(\varepsilon=x)_l\times\frac{1}{P(\hbar_c=0)_l}\times\sum_{y=k-1}^{a\cdot b-1}P(\hbar_c=y)_l\right\}\\=&\ P(\varepsilon=0)_l\times\Omega_{k-1}+P(\varepsilon=1)_l\times\Omega_{k-2}+\cdots+P(\varepsilon=k-1)_l\times\Omega_0\\=&\ P(\varepsilon=0)_l\times\{\Omega_{k-1}+\Phi_1\times\Omega_{k-2}+\cdots+\Phi_{k-1}\times\Omega_0\}\\=&\ P(\varepsilon=0)_l\times\Phi_k\end{aligned} \tag{11}$$

로 정리할 수 있다. 여기서 $\Omega_m=\frac{1}{P(\hbar_c=0)_l}\times\sum_{y=m-2}^{a\cdot b-1}P(\hbar_c=y)_l$,

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

식 (9), (11)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 데이터 패킷의 개수는 0에서 β개 중 하나가 된다.

$$\sum_{x=0}^{\beta}P(\varepsilon=x)_l=P(\varepsilon=0)_l\times\sum_{x=0}^{\beta}\Phi_x=1 \tag{12}$$

이다. 따라서, $P(\varepsilon=0)_l$은

$$P(\varepsilon=0)_l=\frac{1}{\sum_{x=1}^{\beta}\Phi_x} \tag{13}$$

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

일단 데이터 패킷들이 각 소스 노드에서 데이터 유입률 $\zeta_{u\ level\ 1}$로 생성되어 fat-tree 네트워크 FT(h, a, b)로 유입되면, 각 레벨별로 식 (1)과 식 (2), (3)을 반복 계산하여 각 레벨에서 데이터 패킷의 유입 확률과 회귀 라우팅할 확률, 업 라우팅할 확률을 계산할 수 있다. 따라서, 최상위 루트 노드에서의 데이터 패킷의 유입될 확률과 회귀 확률을 이용하여 루트 노드의 child 포트 출력 단으로 데이터 패킷이 출력될 확률, $P(D_c=1)_{level\ h}$,은 식 (4), (6), (7) 그리고 (14)를 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 연결되므로, 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 데이터 패킷이 출력될 확률을 구하고 이를 레벨별로 같은 과정을 반복하여 네트웍의 최종 출력 단으로 데이터 패킷이 출력될 확률, $P(D_c=1)_1$,을 구하면, 네트워크 정상상태 처리율은 식 (5)와 같이 계산된다.

2..3 네트워크 지연시간 분석

임의 데이터 패킷이 네트워크 입력 단에 유입된 후, 각 레벨의 스위치를 지나 최종 출력 단을 통과하기까지 소요되는 스위치 클럭의 평균 개수로 측정되는 네트워크 지연시간은 데이터 패킷 이동 경로와 네트워크 트래픽에 따라 결정된다.

임의 레벨 l에 위치한 스위치의 child 포트로 성공적으로 출력되는 데이터 패킷이 네트워크에 머무른 평균지연시간, $\tau_{s,\ level\ l}$,은 임의 데이터 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

$$\tau_{s,\ level\ l}=\tau_{i,\ level\ l}+\tau_{b,\ level\ l} \tag{14}$$

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

$$\tau_{i,\ level\ l}=\frac{\zeta_{ucc,\ level\ l}\times(l-1)\times\Delta t+\zeta_{d,\ level\ l}\times\tau_{s,\ level\ (l-1)}}{\zeta_{ucc,\ level\ l}+\zeta_{d,\ level\ l}} \tag{15}$$

일단 식 (16)과 같은 평균 지연시간을 가진 임의 데이터 패킷이 child 포트의 버퍼에 유입되면 특정 위치의 버퍼에 저장되고, 일정 기간 동안 스위치에 머무르게 된다. 일단, 스위치 버퍼 k번째 위치에 저장되면, 이는 해당 스위치에서 $(k+1)\times\Delta t$의 시간만큼 머물고 다음 스위치로 이동하게 된다.

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

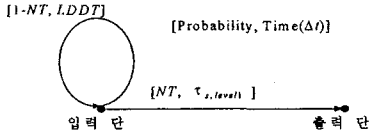

그림 2. 네트워크 지연 시간에 관한 상태도

된다.' 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 데이터 패킷이 해당 출력 단에서 탈락하지 않고 해당 버퍼에 저장될 경우, 버퍼를 통과하는데 걸리는 시간, $\tau_{b,\ level\ l}$은

$$\tau_{b,\ level\ l} = \sum_{k=0}^{b}\Big[\sum_{\rho=k-a-b+2}^{k}\Big\{ P(\varepsilon=\rho)_{l,\ cycle(j-1)} \times \sum_{y=k-\rho}^{a+b-2} \frac{1}{y+1} P(\hbar_c=y)^{*}_{l,\ cycle\ j} \Big\} \times (k+1)\Delta t\Big] \quad (16)$$

으로 얻어진다. 여기서, $P(\hbar_c=y)^{*}_{l}$는

$$P(\hbar_c=y)^{*}_{l} = \sum_{w=0}^{y}\{ {}_{a-2}C_w \times \Big(\frac{\zeta_{ucc,\ level\ l}}{a-1}\Big)^{w} \times \Big(1-\frac{\zeta_{ucc,\ level\ l}}{a-1}\Big)^{a-w-2} \times {}_{b}C_{y-w} \times \Big(\frac{\zeta_{d,\ level\ l}}{a}\Big)^{y-w} \times \Big(1-\frac{\zeta_{d,\ level\ l}}{a}\Big)^{b-y+w} \times P(\hbar_c=1)^{*}_{a} + {}_{a-1}C_w \times \Big(\frac{\zeta_{ucc,\ level\ l}}{a-1}\Big)^{w} \times \Big(1-\frac{\zeta_{ucc,\ level\ l}}{a-1}\Big)^{a-w-1} \times {}_{b-1}C_{y-w} \times \Big(\frac{\zeta_{d,\ level\ l}}{a}\Big)^{y-w} \times \Big(1-\frac{\zeta_{d,\ level\ l}}{a}\Big)^{b-y+w-1} \times P(\hbar_c=1)^{*}_{b}\}$$

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

$$P(\hbar_c=1)^{*}_{b} = {}_{b}C_1 \times \Big(\frac{\zeta_{d,\ level\ l}}{a}\Big)^{1} \times \Big(1-\frac{\zeta_{d,\ level\ l}}{a}\Big)^{b-1} / P(\hbar_c=1)_l$$

이다. 이때 $P(\hbar_c=1)^{*}_{a}$는 데이터 패킷 δ가 회귀라우팅 하는 패킷들 중 하나일 확률이고, $P(\hbar_c=1)^{*}_{b}$는 데이터 패킷 δ가 다운라우팅 패킷들 중에 하나일 확률이다. 또한 $P(\hbar_c=y)^{*}_{l}$는 임의의 데이터 패킷 δ를 제외한 y개의 데이터 패킷이 해당 출력 단을 지향하는 경우를 수식화한 것이다.

네트워크 전체를 성공적으로 통과한 데이터 패킷의 평균 지연시간, $\tau_{s,\ level\ 1}$,은 루트 노드로부터 최하위 레벨로 식 (14), (15), (16)을 반복 계산하여 구할 수 있다.

한편, 중도 유실된 데이터 패킷들은 소정의 "중도 유실 감지 과정"을 거쳐 최초 데이터 패킷이 유입된 입력 단에서 재전송 되게 된다. 그림 2는 네트워크를 성공적으로 통과한 데이터 패킷들의 네트워크 지연 시간과 함께, 전송 과정에서 중도 소실된 데이터 패킷들의 재전송 시간을 고려한 총 네트워크 지연시간에 관한 상태도이다. 여기서, 임의 데이터 패킷 δ가 네트워크를 성공적으로 통과할 확률은 정상상태 처리율(ST)로 볼 수 있고, 반면에 데이터 패킷 δ가 네트워크 내부에서 유실될 확률은 $(1-ST)$로 계산되고, 이들 중도 유실 데이터 패킷은 중도 유실 감지 시간(Lost Data Detection Time, $LDDT$) 만큼의 오류 검사 과정을 거쳐 재 전송된다. 따라서, 임의 데이터 패킷이 전체 네트워크를 통과하는데 걸리는 평균 시간, $\Im$,는 그림 2로 부터

$$\Im = ST \times \tau_{s,\ level\ 1} + (1-ST) \times (LDDT + \Im) \quad (17)$$

와 같은 식으로 얻어진다. 식 (17)를 $\Im$에 관하여 풀면

$$\Im = \tau_{s,\ level\ 1} + \frac{(1-ST)}{ST} \times LDDT \quad (18)$$

과 같이 계산된다. 여기서, ST와 τ_s는 식 (5), (14)로부터 구할 수 있고, $LDDT$는 네트워크 특성에 따라 상수로 주어진다.

표 1과 그림 3는 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. 중도 소실 감지 시간(LDDT)은 최소 소요시간을 $(\beta \times h + h)\Delta t$로 놓고 네트워크 지연시간을 구하였다.

4. 결 론

본 논문에서는 스위치 출력 단에 복수 버퍼를 장착한 $a \times b$ 스위치

표 1. FT(3, 3, 2)의 성능

Buffer size	데이터 패킷 입력률($\zeta_{w,\ level\ 1}$= 1.0)					
	정상상태 처리율(ST, %)		네트웍을 통과한 패킷의 지연시간 (Δt)		데이터 패킷 탈락 확률(%)	
	해석	시뮬레이션	해석	시뮬레이션	해석	시뮬레이션
0	59.78	38.36	4.311	3.781	61.66	61.64
1	82.00	53.78	5.498	4.860	46.04	46.22
2	86.86	59.02	6.487	5.594	40.78	40.98
4	88.35	62.71	7.906	6.584	37.27	37.29
8	88.46	64.10	9.370	7.576	35.71	35.90
16	88.46	64.56	10.06	8.036	35.37	35.44
32	88.46	64.84	10.12	8.011	35.35	35.16

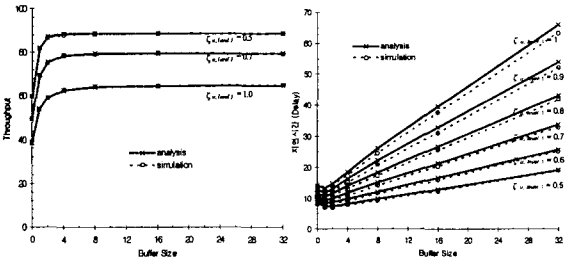

그림 3. 버퍼를 장착한 FT(3, 3, 2) 성능 분석

들로 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 데이터 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. 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.

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

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

[4] 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.

[5] 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

[6] 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.

[7] 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

[8] 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