

Buffered Fat-tree Network의 성능분석

조성래*, 신태지*, 양명국*
 *울산대학교 전기전자공학부

Performance Evaluation of a Buffered Fat-tree Network

Sung-Lae Cho*, Tai Z. Shin*, Myung K. Yang*

*Dept. of Electrical and Electronic Engineering, University of Ulsan

Abstract - 본 논문에서는 buffer를 장착한 양 방향성 $a \times b$ switch들로 구성된 fat-tree network의 성능 분석 기법을 제안하고, 분석 모형의 타당성을 검증하였다. 제안한 분석 기법은 먼저 스위치 내부의 데이터 이동 패턴을 확률식으로 표현하고, 나아가서 buffer를 장착한 $a \times b$ switch의 buffer 크기에 따른 정상상태 throughput을 간단한 수식으로 구할 수 있도록 하였다. 이를 토대로 buffer를 장착한 $a \times b$ switch로 구성된 fat-tree network의 성능을 분석하고, 제안한 분석모형의 실효성 입증을 위하여 simulation을 시행한 후, 결과를 비교·분석하였다.

1. 서론

Fat-tree network은 넓은 bandwidth, 구조적 유연성, 그리고 스위치 고장적용 특성 등의 장점으로 인해 다양한 대규모 고성능 병렬 컴퓨터의 상호 연결망으로 널리 사용되고 있으며, 최근 컴퓨터 통신기술의 발전과 함께 네트워크 스위칭 요소기술로 fat-tree network이 활용되고 있다. Leiserson[1]에 의해 제안된 fat-tree network은 Think Machine CM-5[2] 및 Meiko supercomputer CS-2, 그리고 Kendall Square Research KSR-1 등의 상호연결망으로 사용되었다. Fat-tree network은 복수 root들을 가지는 tree 형태를 띠고, 각각의 vertex는 하나 이상의 root를 가질 수 있다. Fat-tree 네트워크는 일반적인 트리 상호연결망과 달리 말단노드로부터 root로 갈수록 채널의 대역폭을 증가시켜서 병목현상을 완화시킨다. 다중 연결망(MIN)에서는 목적지 주소와 상관없이 통신경로의 길이가 고정되지만, Fat-tree 네트워크에서는 목적지 주소에 따라서 거쳐야 하는 통신 경로의 길이가 달라진다.

Ohring과 Ibel[3] 등은 fat-tree의 일반화에 대한 연구를 진행하였으며, Greenberg와 Lee 등은 wormhole routed network의 성능모형을 butterfly fat-tree에 적용하였다. Alunweiri[4] 등은 shared buffer를 사용하는 buffered fat-tree ATM switch를 제안하였다. 김영식, 권오영[5] 등은 2×2 스위칭 소자로 구성된 fat-tree 네트워크의 성능분석을 시도하였다.

본 논문에서는 buffer를 장착한 양 방향성 $a \times b$ switch들로 구성된 fat-tree network의 성능 분석 기법을 제안하고, 분석 모형의 타당성을 검증하였다. 스위치에 buffer를 장착하여 네트워크 내부의 데이터 충돌로 인한 손실을 줄이는 방법은 이미 잘 알려져 있으나, 이의 성능 분석은 비교적 제한된 범위 내에서 시행되고 있다. 제안한 분석 모형은 Leiserson의 이론적 fat-tree 망의 조건을 만족시키는 $a \times b$ crossbar switch로 구성된 fat-tree 망의 성능분석을 위한 새로운 수학적 기법을 제시하였다. 먼저 신태지와 양명국[6]에 의해 제안된 $a \times a$ crossbar 스위치 성능분석 기법을 확장하여, 양방향 $a \times b$ 스위치 분석 모형을 설정하고, 이를 fat-tree 망의 routing algorithm에 적용하여 네트워크 성능을 분석하였다. 분석모형의 신뢰성 검증을 위하여 시행된

simulation 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, routing algorithm을 기술하였다. 3절에서는 fat-tree 네트워크 내부 buffer를 장착한 $a \times b$ switch 소자의 성능분석 모형을 제시하고, 이를 토대로 fat-tree network의 성능을 분석하였다. 끝으로 마지막 절에는 본 연구의 성과와 결과를 요약·기술하였다.

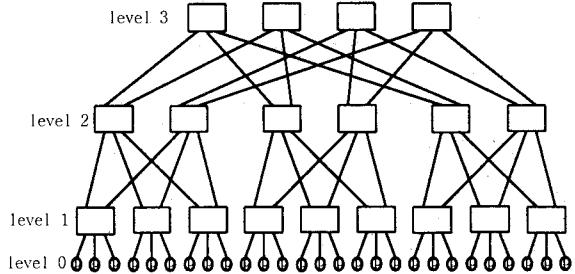
2. Fat tree network**2.1 구조**

그림 1 buffered fat-tree BFT(3,3,2)의 구성

Fat-tree는 parent node의 개수 w 와 child node의 개수 m 에 따라 다양한 구성을 가진다. 그림 1은 일반적인 fat-tree의 구조를 보여주고 있다. Buffered 스위치로 구성된 높이 h 인 fat-tree, BFT(h, m, w),는 level 0에 m^h 개의 processor와 상위 level에서 router와 스위치로 동작하는 $m^{h-1}w^{h-1}$ 개의 스위치 node로 구성된다. 중간 level의 node들은 각각 w 개의 parent node 그리고 m 개의 child node와 연결된다. 각 스위치 node는 위치에 따라 (l, x) 로 색인된다. 이때, l 은 node의 level 즉, leaf로부터의 거리를 나타내고, x 는 동일 level에서 node의 위치를 나타낸다. 스위치 node $S(l, x)$ 는 ($1 \leq l \leq h-1$) w 개의 parent ports($P_0, P_1, \dots, P_{(w-1)}$)와 m 개의 child ports($C_0, C_1, \dots, C_{(m-1)}$)를 가진다. 각각의 child port는 데이터 패킷 충돌 처리를 위하여 buffer를 장착하고 있다. level 0의 processor $P(0, x)$ 는 level 1의 스위치 node $S(1, \lfloor x/m \rfloor)$ 의 child port C_i (여기서, $i = (a \bmod m)$)에 연결된다. level 1의 스위치 node $S(l, x)$ 의 parent port P_i 는 level $(l+1)$ 의 스위치 $S(l+1, \lfloor (x_i)/(mw^{l-1}) \rfloor \times w^l + (x_i \bmod w^{l-1}) \times w + i)$ 의 child port C_i ($i = \lfloor (x \bmod mw^{l-1})/w^{l-1} \rfloor$)에 연결된다. level 1의 스위치 node $S(l, x)$ 의 child port i 는 level $(l-1)$ 의 스위치 $S(l-1, \lfloor x/w^{l-1} \rfloor \times mw^{l-2} + \lfloor (x \bmod w^{l-1})/w \rfloor + i \times w^{l-2})$ 의 parent port P_i (여기서, $i = (x \bmod w)$)에 연결된다.

특별한 경우에 $w=1$ 이면 m -ary tree가 되고, $m=w$ 인 경우 BMIN(bidirectional MIN)과 유사한 그래프를 가지게 된다.

2.2 Routing algorithm

Fat-tree 네트워크에서 데이터 패킷은 level 0의 임의의 프로세서에서 생성되어 level 0의 특정 프로세서 $P(0, d)$ 를 지향한다. 데이터 패킷은 임의 level 1에 위치한 스위치 $S(l, x)$ 의 subtree에 목적지가 포함될 때까지 up routing 한다. Up routing에서 스위치 node는 child port C_i 로부터 가능한 parent port로 데이터 패킷 전송을 시도한다. 이때 random하게 parent를 선택하게 된다. 만약 random하게 선택된 parent가 block되었을 경우, 다른 parent port로 전송을 시도하게 되고, 모두 block되어 있을 경우, 데이터 패킷은 버려지게 된다.

데이터가 $S(l, x)$ 에서 $\lfloor x/(w^{l-1}) \rfloor = \lfloor d/m^l \rfloor$ 을 만족하는 경우 회귀 routing을 하게 되는데, 회귀 routing의 경우 스위치 node 내에 child port C_i 로부터 다른 child port C_k (여기서, $k = (\lfloor d/(m^{l-1}) \rfloor \bmod m)$, $k \neq i$)로 packet을 전송한다. 이때 $S(l, x)$ 는 데이터 패킷이 지향하는 목적지 $P(0, d)$ 를 포함하는 subtree의 root이다.

데이터가 level 1의 임의 스위치 $S(l, x)$ 에서 회귀 routing을 시작한 후, 데이터 패킷은 목적지 주소를 라우팅 태그(tag)로하여 down routing 한다. down routing에서 데이터 패킷은 parent port P_j 로부터 child port C_i (여기서, $i = (\lfloor d/(m^{l-1}) \rfloor \bmod m)$)로 전송된다.

Up routing에서 스위치 node 내에서 모든 가능한 parent로 데이터 패킷을 전송하는 반면, down routing과 회귀 routing에서는 유일한 경로만을 가진다. 데이터 패킷이 down 또는 회귀 routing을 하는 경우 데이터 패킷은 목적지 주소를 이용하여 child port를 결정하게 된다. 스위치의 child port는 데이터의 충돌을 처리하기 위해 buffer를 가지고 있으며, 데이터의 충돌 발생 시 무작위 중재방식에 의거 데이터 처리(통과 혹은 buffer에 저장) 우선 순위를 결정한다. 그림2는 스위치 node 내에서 전송 가능한 경로를 보여주고 있다.

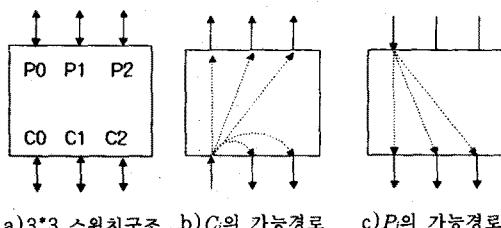


그림2 스위치 node 내에서 전송 가능 경로

3. Fat tree network의 성능분석

Level 1에 위치한 임의 스위치 node i 의 child port C_a 로 유입된 데이터 패킷이 동일 스위치 node 내부 다른 child port C_b 로 회귀할 확률, ζ_{ucc} , 는 child port C_a 에 데이터가 입력될 확률 ζ_u 과 해당 패킷이 level 1에서 선회할 확률의 곱으로 다음과 같이 구할 수 있다.

$$\zeta_{ucc} = \zeta_u \frac{m^l - m^{l-1}}{m^h - m^{l-1}} = \zeta_u \frac{m-1}{m^{h-l+1}-1} \quad (1)$$

Level 1에 위치한 임의 스위치 node i 의 child port C_a 로 유입된 데이터 패킷이 동일 스위치 node의 임의 parent port로 진행할 확률, ζ_{ucp} , 는 다음과 같이 구할 수 있다.

$$\zeta_{ucp} = \zeta_u \frac{m^h - m^l}{m^h - m^{l-1}} \quad (2)$$

임의 스위치 node의 데이터 패킷이 parent port를 지향하는 경우, 데이터 패킷은 랜덤하게 parent port를 선택하고, 선택된 parent port가 block되었을 경우 다른 parent port를 선택하여 액세스한다. 만약 모든 parent port가 block되었을 경우 해당 데이터 패킷은 discard된다. 스위치 node 내 child port들로부터 r 개의 데이터 패킷이 parent를 지향하는 경우에 특정 parent node로 패킷이 지향할 확률은 r/w 이다. 만약 $r \geq w$ 인 경우에 패킷은 항상 임의 parent로 지향하게 된다.

임의 parent port로 데이터 패킷이 통과하여 출력될 확률, $P(P_r=1)_i$, 은 해당 스위치 child port로 유입된 데이터 패킷이 parent port로 지향할 확률, ζ_{ucp} , 을 이용하여 다음과 같이 얻어진다.

$$P(P_r=1)_i = \sum_{r=1}^w \left\{ {}_w C_r (\zeta_{ucp})^r (1 - \zeta_{ucp})^{w-r} \times (r/w) \right\} \quad (3)$$

여기서, r/w 이면 $r/w = 1$ 이다.

임의 child port로 r 개의 데이터 packet이 지향할 확률, $P(h_c=r)_i$, 은 스위치 node 내부의 데이터 패킷 회귀율, ζ_{ucc} , 와 parent port에서 child port로 하강하는 packet rate, ζ_d , 를 이용하여 다음과 같이 구할 수 있다.

$$P(h_c=r)_i = \sum_{x=0}^{m-1} \left\{ {}_{m-1} C_x \left(\frac{\zeta_{ucc}}{m-1} \right)^x \left(1 - \frac{\zeta_{ucc}}{m-1} \right)^{m-x-1} \right. \\ \left. \times {}_w C_{r-x} \left(\frac{\zeta_d}{m} \right)^{r-x} \left(1 - \frac{\zeta_d}{m} \right)^{w-r+x} \right\} \quad (4)$$

임의 child port C_i 로 데이터 패킷이 출력되지 않는 경우는 해당 출력 단 buffer에 데이터 패킷이 저장되지 않은 상태에서, 스위치 입력 단에서 해당 출력 단으로 지향하는 데이터 패킷이 없을 경우이다. 따라서 임의 레벨 1에 위치한 스위치 node의 child port C_i 로 데이터 패킷이 출력되지 않을 확률, $P(C_i=0)_i$, 을 구하면

$$P(C_i=0)_i = P(\epsilon=0)_i \times P(h_c=0)_i, \quad (5)$$

이 된다.

Buffer에 저장된 데이터 패킷의 수가 0일 확률, $P(\epsilon=0)_i$, 은

$$P(\epsilon=0)_i = P(\epsilon=1)_i \times P(h_c=0)_i + P(\epsilon=0)_i \times P(h_c=1)_i + P(\epsilon=0)_i \times P(h_c=0)_i \quad (6)$$

이다. 식(5)에서 $P(\epsilon=1)_i$ 를 $P(\epsilon=0)_i$ 의 식으로 구하면

$$P(\epsilon=1)_i = P(\epsilon=0)_i \times \frac{1}{P(h_c=0)_i} \times \sum_{r=2}^{w+m-1} P(h_c=r)_i \\ = P(\epsilon=0)_i \times Q_0 \quad (7)$$

여기서 $Q_0 = \frac{1}{P(h_c=0)_i} \times \sum_{r=2}^{w+m-1} P(h_c=r)_i$ 이다. 식 (6)

과 같은 방식으로 buffer에 하나의 데이터 패킷이 저장될 확률, $P(\epsilon=1)_i$, 을 계산하면,

$$P(\epsilon=1)_i = P(\epsilon=2)_i \times P(h_c=0)_i + P(\epsilon=1)_i \times P(h_c=1)_i + P(\epsilon=0)_i \times P(h_c=2)_i \quad (8)$$

가 된다. 식(8)와 식(7)을 $P(\epsilon=2)_i$ 에 대해 정리하면 다음과 같은 식을 유도 할 수 있다.

$$P(\epsilon=2)_i = P(\epsilon=0)_i \times \frac{1}{P(h_c=0)_i} \sum_{r=3}^{w+m-1} P(h_c=r)_i \\ + P(\epsilon=1)_i \times \frac{1}{P(h_c=0)_i} \sum_{r=2}^{w+m-1} P(h_c=r)_i \\ = P(\epsilon=0)_i \times (Q_1 + Q_0^2) \quad (9)$$

$$\text{여기서 } Q_1 = \frac{1}{P(h_c=0)_i} \sum_{r=3}^{w+m-1} P(h_c=r)_i.$$

$$\Omega_0 = \frac{1}{P(\zeta_c=0)} \sum_{r=2}^{m+w-1} P(\zeta_c=r)$$

같은 방법으로 buffer가 임의 싸이클 종료 시 ($k-1$)개의 데이터 패킷을 저장하고 있을 확률을 이용하여, $P(\epsilon=k)$ 를 구하면

$$\begin{aligned} P(\epsilon=k) &= P(\epsilon=0) \times \frac{1}{P(\zeta_c=0)} \times \sum_{r=k+1}^{m+w-1} P(\zeta_c=r) \\ &+ P(\epsilon=1) \times \frac{1}{P(\zeta_c=0)} \times \sum_{r=k}^{m+w-1} P(\zeta_c=r), \\ &\dots \\ &+ P(\epsilon=k-1) \times \frac{1}{P(\zeta_c=0)} \times \sum_{r=2}^{m+w-1} P(\zeta_c=r), \\ &= P(\epsilon=0) \times (\Omega_{k-1} + \Omega_{k-2}^2 + \dots + \Omega_1^{k-1} + \Omega_0^k) \\ &= P(\epsilon=0) \times \sum_{m=0}^{k-1} \Omega_m^{k-m} \end{aligned} \quad (10)$$

여기서 $\Omega_m = \frac{1}{P(\zeta_c=0)} \times \sum_{r=m+2}^{m+w-1} P(\zeta_c=r)$ 가 된다.

그리고 Ω_m 은 ζ_{level_1} 가 주어질 때 식 (4)로 얻을 수 있다. 또한, $P(\epsilon=0)$ 는 스위치가 수용할 수 있는 데이터 패킷의 수가 b 개로 주어질 때, 임의 싸이클 종료 시 buffer에 저장된 데이터 패킷의 수는 0개에서 b 개 중 하나라는 것으로부터 구할 수 있다. 즉

$$\sum_{k=0}^b P(\epsilon=k) = P(\epsilon=0) \times \sum_{k=0}^b \sum_{m=0}^{k-1} \Omega_m^{k-m} = 1 \quad (11)$$

이고, 따라서 $P(\epsilon=0)$ 는 식(12)와 같이 계산된다.

$$P(\epsilon=0) = \frac{1}{\sum_{k=0}^b \sum_{m=0}^{k-1} \Omega_m^{k-m}} \quad (12)$$

Fat tree network 내부 level l 에 위치한 $a \times b$ crossbar 스위치의 child port C_r 로 데이터 패킷이 출력될 확률, $P(C_r=1)_l$ 은 식 (5), (12)로부터 $P(C_r=0)_l$ 을 계산한 후, 다음 식(13)에 의해서 얻을 수 있다.

$$P(C_r=1)_l = 1 - P(C_r=0)_l \quad (13)$$

Fat tree network의 구조상 level l 의 child port로 데이터 패킷이 출력될 확률, $P(C_r=1)_l$,은 level $(l-1)$ 에 위치한 해당 스위치의 parent port로 데이터 패킷이 유입될 확률, $\zeta_{level(l-1)}$,이 된다. 또한, level l 의 parent port로 데이터 패킷이 출력될 확률, $P(P_r=1)_l$,은 level $(l+1)$ 의 child port로 데이터 패킷이 유입될 확률, $\zeta_{level(l+1)}$,이 된다.

따라서 네트워크 입력 단의 ζ_{level_1} 가 주어지면 이로부터 $P(P_r=1)_1$ 을 구하고, $P(P_r=1)_1$ 을 다시 ζ_{level_2} 로 놓고 $P(P_r=1)_2$ 를 구하는 과정을 반복하여 parent port로의 패킷 출력 확률을 구할 수 있다. 또한 정상상태에서 임의 싸이클 j 에서 parent port로 패킷이 출력될 확률, $P(P_r=1)_{j, cycle, l}$,는 임의 싸이클 j 에서 parent로 패킷이 출력될 확률, $P(P_r=1)_{j, cycle, l}$,과 동일하므로, fat-tree network의 최종 출력 단(level 0)에서의 패킷이 출력될 확률, $P(C_r=1)_{level 1}$,을 재귀적으로 구해낼 수 있다.

BFT(h, m, w)의 경우 전체 네트워크 출력 단으로 출력되는 데이터 패킷의 수, OP ,는 식(14)로 계산된다.

$$OP = m^h \times P(C_r=1)_{level 1} \quad (14)$$

또한 다중 연결 망 입력 단으로 매 싸이클마다 유입되는 총 데이터 패킷의 수를 $IP (= m^h \times \zeta_{level_1})$ 라 하면, fat-tree 네트워크 정상상태 Throughput, NT (Normalized Throughput),은

$$NT = \frac{OP}{IP} = \frac{P(C_r=1)_{level 1}}{\zeta_{level 1}} \quad (15)$$

와 같이 얻어진다.

4. 결론

본 연구에서는 양방향성 $a \times b$ 스위치로 구성된 buffered fat-tree 네트워크 성능분석 모형을 제시하였다. 제시된 분석모델은 스위치에 장착된 buffer의 개수와 무관하게 적용 가능하고, 분석과정에서 간단한 데이터 충돌 처리 기법을 도입하여 모델의 수식 이해가 용이하다. 제안한 수학적 성능 분석 연구의 실효성을 검증하기 위한 시뮬레이션 처리 결과는 상호 미세한 오차 범위 내에서 모형의 예측 데이터와 일치하는 결과를 보여 분석 모형의 타당성을 입증하였다. 또한 제안된 본 연구의 분석모델은 변형된 팻 트리밍 뿐만 아니라 다양한 양방향성 다단계 상호 연결 망의 성능 분석에 확대 적용 가능하다.

표 1 BFT(3, 4, 4)의 buffer size와 ζ 에 따른 성능

ζ	1	0.9	0.8	0.7				
b	imratio	analysis	imratio	analysis	imratio	analysis		
0	48.289	48.076	51.402	50.930	54.558	54.102	58.197	57.642
1	71.526	71.651	76.244	76.079	80.698	80.577	84.997	84.979
2	80.965	81.303	86.203	86.180	90.463	90.600	94.015	94.222
3	85.867	86.270	91.078	91.188	94.813	95.083	97.438	97.662
4	88.838	89.252	93.909	94.059	97.068	97.328	98.839	99.034
5	90.808	91.225	96.674	96.853	98.236	98.520	99.474	99.598
6	92.236	92.619	96.863	97.041	98.991	99.172	99.754	99.832
7	93.267	93.655	97.650	97.858	99.396	99.535	99.883	99.930
8	94.085	94.454	98.272	98.433	99.644	99.738	99.941	99.971

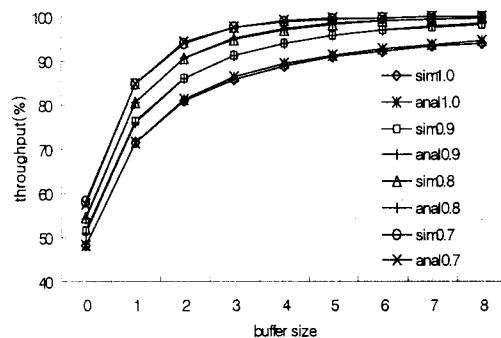


그림 3 BFT(3,4,4)의 buffer size와 ζ 에 따른 throughput(%)

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] 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
- [4] Alunweiri H.M. Aljunaidi H. Beraldí R, " The Buffered Fat-Tree ATM switch ", Global Telecommunication Conference, 1995. GLOBECOM '95.. IEEE Volume:2, 1995, Page(s): 1209-1215
- [5] Youngsik Kim, Oh-Young Kwon, Tack-Don Han, Youngho Mun, " Design and performance analysis of the Practical Fat Tree Network using a butterfly network ", Journal of systems Architecture 43, pp 355-363, 1997
- [6] 신태지, 양명국, "Buffered-MIN의 성능분석", 한국정보과학회 가을 학술발표논문집, 제 26권 2호, pp244-246, 1999