

완전 적응 자기 경로제어 알고리즘을 사용하는 사이클릭 베니안 네트워크의 성능 분석

정회원 박재현*

Performance Analysis of the Cyclic Banyan Network using a Fully-Adaptive Self-routing Algorithm

Jae-Hyun Park* *Regular Member*

요약

본 논문에서는 완전 적응 자기 경로 제어 알고리즘을 사용하는 고장 감내 셀 스위치인 사이클릭 베니안 네트워크의 성능을 분석한다. 본 스위치 구조는 기존 스위치보다 입력단과 출력단 사이에 더 많은 다중 경로(Path)들을 제공함으로써, 높은 신뢰성을 제공함과 동시에, 짧은 시간의 스위치 내부 경로상의 혼잡 문제를 해결한다. 본 논문에서는 사이클릭 베니안 네트워크의 성능을 분석하기 위하여, 적응 라우팅에 따라 스위칭되는 셀들을 모형화하는, 대기행렬 모델에 기반을 둔 분석적 모델을 제시한다. 이 모델의 정확성을 시뮬레이션을 통해서 검증한다. 또한 인터넷에서 보편적이라 할 수 있는 비균일한 주소 분포를 가진 트래픽 부하에 대한 성능 분석을 시뮬레이션을 통해 보인다. 본 논문은 정량적인 스위치 성능 분석을 제공함으로써, 사이클릭 스위치 구조가 베니안 네트워크나 다른 네트워크보다 성능과 확장성이 우수함을 보인다.

ABSTRACT

In this paper, we develop the performance analysis of the cyclic Banyan network, which is a fault tolerant cell switch architecture using a fully-adaptive self-routing algorithm. By providing more multiple paths than the related previous switches between an input/output pair of a switch, the switching architecture imparts high reliability, and resolves the short-term congestion problem of the internal switching paths. To analyze the performance of the cyclic Banyan network, we present an analytical model based on a queuing network model. We expressed the cells being switched in accordance with the adaptive self-routing in this model. We show the correctness of the analytical model through comparing the results to simulation ones. We also ran a simulation analysis under the traffic loads having the non-uniform address distributions that is usual in Internet. This analysis shows that proposed switching network architecture is suitable to Internet traffic. We present an analytic performance analysis to provide quantitative comparisons with other switches, which shows that the cyclic switch architecture has a far better performance and scalability than the other networks.

I. 서론

베니안 네트워크는 입력력 단들 간에 병렬 연결을 제공하는 수단으로써, 비용 대 효과비 측면에서 크로스바 스위치의 좋은 대체 스위치이다. 베니안 네

트워크는 이미 상용화 되어왔지만^[13], 네트워크상의 각 입출력 쌍에 대해서 각각 유일한 경로만을 갖기 때문에 고장 감내성이 없다는 문제점을 가지고 있다. 결과적으로 어떠한 하나의 고장도 이러한 종류의 네트워크를 동작불가능 상태로 만든다.

* 영남대학교 전자정보공학부 (jaehyun.park@ieee.org),
논문번호:00439-1115, 접수일자: 2000년 11월 15일

그들이 고장 감내성을 갖게하기 위해서 몇몇 기법들이 개발되었는데, 이들 방법들은 고장이 생긴 경로들(paths)를 돌아서 가는 여분의 경로들을 제공하는데 중점을 두고 있다^[1,4,5,11,14]. 이들 작업들은 추가된 링크를 통해서 신뢰성을 높이는데 성공했으나, 기존의 링크가 우회 경로로 사용될 수 있음을 간과 했다.

사이클릭-베니안 네트워크(Cyclic Banyan network) [7, 8, 9, 20]은 고장난 링크가 생겼을 때, [6, 7, 8, 9, 10]의 방법과 같이, 추가된 링크 뿐만 아니라 기존의 링크도 패킷의 경로를 위해서 사용한다. 본 스위치는 간단한 라우팅 알고리즘을 사용하며, 스위칭 요소의 복잡도도 기존의 스위치와 같게 유지한다. 이러한 편향 자기 경로제어(Deflection Self-routing)는 고장 감내성의 개선뿐만 아니라, 성능의 증가도 가져온다.

대형 셀 교환 시스템을 구성하기 위한, 다단 스위칭 네트워크는 스위치 페브릭들로 구성되는데, 이들은 링크들로 상호 연결되거나, 링크를 통해서 정합 장치들과 연결된다. 이러한 셀 교환 시스템의 주요 수요처로 예상이 되는 인터넷 트래픽은 폭발적이어서, 하나의 스위치 페브릭에 다른 입력 링크들을 통해 동시에 도착하는 셀들이 동일한 출력 링크를 통해 전달될 가능성이 매우 크다.

사이클릭-베니안 네트워크는 많은 우회 경로를 제공함으로써, 이러한 폭발적 트래픽 상황에서 적합한 구조인데, 본 논문에서는 분석적 모델을 통해서 본 스위칭 네트워크 구조의 성능을 분석한다. 그리고 이 모델의 정확성을 시뮬레이션을 통해서 검증한다.

2절에서 사이클릭-베니안 스위칭 망과 완전 적응 자기 경로제어에 대해 간략히 설명한다. 3절에서는 균일한(Uniform) 트래픽 모델을 사용하는 성능 분석을 보인다. 그리고 비균일한 트래픽 모델을 적용한 성능 분석을 보인다. 끝으로 4절에서 결론을 맺는다.

II. 사이클릭-베니안 네트워크와 셀 스위칭

1. 사이클릭-베니안 스위치

사이클릭 베니안 네트워크는 베니안 네트워크에 링크들 추가하고, 이러한 링크들과 기존의 모든 링크들을 우회경로로 사용하는 자기 경로제어 알고리즘을 모두 채용한 다단 스위칭 네트워크 이다^[8,20]. 이 네트워크는 베니안 네트워크에 기반을 두기 때문에, 본 네트워크의 형식적 명세는 베니안 네트워크의 위상 기술법

을 확장하여 만든다. 베니안 네트워크에 속하는 많은 다단 스위칭 망은 위상적으로 동일하다는 것이 널리 알려져 있기 때문에^[15], 본 논문에서는 델타 네트워크를 예로 들어 설명한다. 본 논문에서는 C. Wu^[15]의 것과 동일한 형태의 명명법(naming scheme)을 확장하여 사이클릭 베니안 네트워크의 형태(configuration)와 경로제어 알고리즘을 기술하기 위해서 사용한다. 각 단계는 첫 단계를 1로 하며, 연속으로 $\log_2 N$ 까지 번호를 붙인다. N 개의 입출력 단들은 $\log_2 N$ 자리수의 이진수로 번호를 붙인다: $link(p_{\log_2 N}, p_{(\log_2 N)-1}, \dots, p_1)$. 각 단계에서, 한 SE는 $(\log_2 N)-1$ 자리수의 이진수들로, $(p_{(\log_2 N)-1}, p_{(\log_2 N)-2}, \dots, p_1)$, 표현되는데, 이는 꼭대기부터 시작하여 단계내에서의 위치를 이진수로 표현한 것이며, 수준(level)이라고도 부른다. 입력 패킷의 목적 주소는 $A = (a_1, a_2, \dots, a_{\log_2 N})$ 로 표현 된다. 이제 SE와 링크를 기술하기 위한 약식 기호를 다음과 같이 정의한다. 기호 n_{ij} 는 단계 i , 수준 j 의 SE이다. 여기서 $1 \leq i \leq \log_2 N$ 이고 $0 \leq j < N/2$ 이다. 그러면, 사이클릭-베니안 네트워크는 다음과 같이 정의된다.

정의 1 사이클릭-베니안 네트워크의 각 출력 링크에 대한 위상 기술방법은 다음과 같다^[8,20].

$$\beta_0(p_1, p_{l-1}, \dots, p_1)_i = (p_{l-1}, p_{l-2}, \dots, p_1, 0)_{i+1}, \quad (1)$$

link($p_1, p_{l-1}, \dots, p_1, 0$) _{i} 로 연결

$$\beta_l(p_1, p_{l-1}, \dots, p_1)_i = (p_{l-1}, p_{l-2}, \dots, p_1, 1)_{i+1}, \quad (2)$$

link($p_1, p_{l-1}, \dots, p_1, 1$) _{i} 로 연결

여기서 $l = (\log_2 N) - 1$, $1 \leq i \leq l$ 이다.

연쇄 링크들에 대한 위상 기술방법은 다음과 같다.

$$\gamma_l[n_{ij}] = n_{i, (j+1) \bmod (N/2)}, \quad (3)$$

$$\gamma_{-l}[n_{ij}] = n_{i, (j+N/2-1) \bmod (N/2)} = \gamma_{N/2-1}[n_{ij}]. \quad (4)$$

식 (1)과 (2)의 연산 β 는 단계 i 의 한 SE로부터 단계 $i+1$ 의 두 SE들로의 사상(mapping)으로 상호 연결을 표현한다. 연산 β_0 는 SE의 상위 출력 링크를 통해서 도달 가능한 SE를 기술하기 위해서 사용하고, 연산 β_l 은 하위 출력 링크를 위해서 사용한다. 식 (3)과 (4)의 연산 γ_l 과 연산 γ_{-l} 은 연쇄-출력 링크들에 대해서 한 SE를 동일한 단계내의 다른 SE로 사상한다. 일반적인 16×16 델타 네트워크로부터 사이클릭 베니안 네트워크를 구성하는 예를 그림 1에 나타내었다.

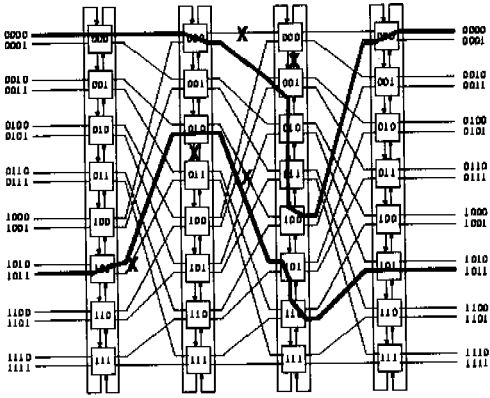


그림 1. 사이클릭 베니안 네트워크와 적응 자기 경로제어 알고리즘

2. 적응 자기 경로제어 알고리즘

많은 다른 MIN들과 같이, 사이클릭 베니안 네트워크의 경로제어는 목적지 꼬리표(tag)를 가지고 제어된다. 여기에 덧붙여서, 패킷에 덧붙여지는 고정된 크기의 $\log_2 N$ 이탈 꼬리표(deviation tag) k 가 각 단계에서 원래 목적이 된 SE나 이것의 동치 SE로부터의 위상적 거리를 나타내기 위해서 사용된다. k 값은 그것의 현재 값과 선택된 링크 그리고 원래 목적된 링크를 가지고 계산할 수 있다. 따라서, 행선지 주소는 (A, k) 쌍으로 표현된다. 단계 1의 입력 링크에 한 패킷이 도착했을 때, k 값은 0으로 설정된다. 즉 행선지 주소는 $(A, 0)$ 이다. 사용되는 적응 경로제어 알고리즘은 다음과 같다. (여기서 i 는 해당 SE의 현재의 단계 번호이다.)

정의 2 사이클릭 베니안 네트워크의 각 SE의 적응 자기 경로제어 알고리즘은 다음과 같다^[8,20].

(여기서 $s \in \{0, 1\}$ 이고, $\alpha_i[(p_0, p_1, \dots, p_i)_s, (a_1, a_2, \dots, a_{i+1})] = (p_1, p_2, \dots, p_i, a_i)_{i+1}$ 이다. $(a_1, a_2, \dots, a_{i+1})$ 는 각 셀의 주소이다.)

각 입력 셀에 대하여,

1. $k = 0$ 인 경우,
 - (a) $\beta_s[\cdot]$ ($= \alpha_i[\cdot]$)로 보낸다.
 - (b) 만약 실패하면, $\beta_s[\cdot]$ ($\neq \alpha_i[\cdot]$)로 보낸다.
 - (c) 만약 실패하면, $\gamma_{s0}[\cdot]$ 로 보낸다.
 - (d) 만약 실패하면, $\gamma_{s(s+1)}[\cdot]$ 로 보낸다.
2. $k \neq 0$ 인 경우,
 - (1) $\gamma_{s0}[\cdot]$ 로 보낸다.
 - (2) 만약 실패하면, $\gamma_{s0(1)}[\cdot]$ 로 보낸다.
 - (3) 만약 실패하면, $\beta_{s(0)}[\cdot]$ 로 보낸다.

(4) 만약 실패하면, $\beta_{s(1)}[\cdot]$ 로 보낸다.

k 를 수정하는 규칙은 다음과 같다.

1. $k = 0$ 일때,
 - (1) 만약 $\beta_s[\cdot] = \alpha_i[\cdot]$ 그리고 $\beta_s[\cdot]$ 가 선택되었다면, $k \leftarrow 2^{i-1}$.
 - (2) 만약 $\beta_s[\cdot] = \alpha_i[\cdot]$ 그리고 $\beta_s[\cdot]$ 가 선택되었다면, $k \leftarrow 1$.
 - (3) 만약 $\gamma_{s1}[\cdot]$ 가 선택되었다면, $k \leftarrow 2^{i-1} - 1$.
 - (4) 만약 $\gamma_{s-1}[\cdot]$ 가 선택되었다면, $k \leftarrow 1$.
2. $k \neq 0$ 일때,
 - (1) $\beta_s[\cdot] = \alpha_i[\cdot]$ 그리고 $\beta_s[\cdot]$ 가 선택되었다면, $k \leftarrow k \times 2 - 1$.
 - (2) 만약 $\beta_s[\cdot] = \alpha_i[\cdot]$ 그리고 $\beta_s[\cdot]$ 가 선택되었다면, $k \leftarrow k \times 2 + 1$.
 - (3) 만약 $\beta_j[\cdot] = \alpha_i[\cdot]$ 그리고 $\beta_j[\cdot]$ 가 선택되었다면, $k \leftarrow k \times 2$, 여기서, $j \in \{0, 1\}$.
 - (4) 만약 $\gamma_{s1}[\cdot]$ 가 선택되었다면, $k \leftarrow k - 1$.
 - (5) 만약 $\gamma_{s-1}[\cdot]$ 가 선택되었다면, $k \leftarrow (k + 1) \bmod 2^{i-1}$.

여기서 $\delta(j) = \begin{cases} 1, & j \bmod 2 = 1 \text{ 경우} \\ -1, & \text{나머지 경우} \end{cases}$ 이고

$$\theta(m) = \begin{cases} (m+1) \bmod 2, & k \leq 2^{i-2} \text{ 경우} \\ (m) \bmod 2, & \text{나머지 경우} \end{cases} \text{ 이다.}$$

적응 자기 경로제어 알고리즘의 두가지 예를 그림 1에서 볼 수 있다. (여기서 충돌은 X로 표시되어 있다.) 한 셀이 입력단 (0000)에서 출력단 (0000)으로 전달 되어야 할 경우, 만약 혼잡인 링크를 통해 스위칭되도록 원래 경로가 설정되었다면, 그림 1에서 보인 것 처럼, 원래 목적된 출력 링크가 혼잡인 SE는 본 경로제어 알고리즘에 의해, 셀을 다른 출력 링크로 보낸다. 그 셀은 본 경로제어에 의해 연쇄 링크를 통해서 원래 의도된 SE의 동치 SE중 하나로 후에 보내 질 것이고, 여기서부터 목적지 출력 정합으로 올바르게 보내질 수 있을 것이다. 그리고 정상적인 경로에 3개의 혼잡들을 가지고 있는 (1011)로부터 (1011)의 경로제어 경우에 대해서도, 본 스위치와 경로제어 알고리즘을 사용하여, 올바르게 경로를 설정할 수 있음을 그림 1에서 예시하였다.

III. 균일한 트래픽 하에서의 사이클릭 베니안

사이클릭-베니안 네트워크의 성능 향상은 모든 링크들을 대체 경로들로 사용하는 완전 적응, 편향 자기 경로제어 방법의 사용에서 나온다. 이 절에서는 균

일한 트래픽 하에서의 사이클릭-베니안 네트워크의 성능을 분석한다. 사이클릭-베니안 망의 성능 분석을 위해서 일종의 대기 행렬 모델을 사용한다.

1. 균일한 트래픽 모델

본 절에서는 균일한 트래픽 모델 하에서의 스위칭 네트워크를 분석한다. 이를 위해 다음과 같은 일반적으로 사용되는 가정들을 도입한다.

가정 1

- 새로운 셀들은 입력 포트들에 도착률 λ 를 갖는 포아송 확률 과정에 따라 도착한다. 여기서 λ 는 $0 \leq \lambda \leq 1$ 이다.
- 입력 셀들은 모든 출력 포트들에 대해서 균일하게 분산된다. 또한, 네트워크에 하나의 셀이 도착할 확률은 모든 입력포트에 대해 동일하다.
- 각 셀이 경쟁시 이길 확률이 동일하다.

이들 가정들은 스위칭 네트워크의 각 스위칭 단들에 대해서, 셀 분포 패턴이 모든 단위 스위치들에 대해서 동일하고 통계적으로 독립적이라는 것을 의미한다. 그러므로 각 스위칭 단들은 하나의 단위 스위치로 간주될 수 있고, 이러한 사실은 망의 분석을 매우 쉽게 한다.

다중 경로의 스위칭 네트워크의 성능 분석을 위한 많은 관련된 작업들 [16, 17]에서, 목적지에서의 충돌은 무시되었다. 그러한 가정은 셀 주기 마다 여러 개의 요청들을 받아들일 수 있는 빠른 출력 큐를 사용하고, 큐의 앞쪽에 멀티플렉서를 사용함으로써 정당화 될 수 있다.

예들을 들어, 목적지에서의 충돌을 피하기 위해, 네트워크의 속도의 2배로 동작하는 2×1 멀티플렉서와 하나의 주기에 두 셀들을 받아들일 수 있는 빠른 큐가 중복-2 (Replication-2) 베니안에 일반적으로 사용된다. 그리고 네트워크의 속도의 4배로 동작하는 4×1 멀티플렉서와 하나의 주기에 네개의 셀들을 받아들일 수 있는 빠른 큐가 중복-4 (Replication-4) 베니안에 사용된다. 중복-8의 베니안을 위해서, 네트워크의 속도의 8배로 동작하는 8×1 멀티플렉서와 하나의 주기에 여덟 셀들을 받아들일 수 있는 빠른 큐가 사용된다.

제안된 스위칭 네트워크를 위해, 네트워크의 속도의 4 배로 동작하는 4×1 멀티플렉서와 하나의 주기에 네개의 셀들을 받아들일 수 있는 빠른 큐를 사용한다. 따라서, 본 절에서의 다양한 종류의 네트워크들의 성능 분석들을 위해 다음과 같은 가정을 적용할 수 있다.

가정 2 네트워크의 출력 포트에 있어서 경쟁은 없다. 제안된 편향 자기 경로제어를 사용하는 사이클릭-베니안 네트워크는 단위 스위치들 안에서 버퍼링 메커니즘과 백프레서 메커니즘을 사용하지 않는다. 따라서, 마지막 단에 도착하며 편향 꼬리표 k 의 값으로 영이 아닌 값을 갖는 셀들은, 가능한 연쇄 링크가 없으면 상실된다. 그러므로 우리는 다음과 같이 가정한다.

가정 3 마지막 단의 단위 스위치들은 편향 꼬리표 K 의 값으로 영이 아닌 값을 갖고, 가능한 연쇄 링크가 없으면 입력 셀들을 버린다.

사이클릭 베니안 네트워크의 성능 분석을 위해서 멀티 컴퓨터 네트워크의 분석^[18]에 사용된 것과 유사한 대기행렬 모델을 사용할 것이다. 본 모델에서 각 셀들은 하나의 클래스와 연관지어지고, 이 연관에 의해 우리는 성능 분석 관점에서 사이클릭 베니안 네트워크의 편향 자기 경로제어를 모형화 할 것이다. 셀은 원래 의도된 단위 스위치와 (혹은 원래 의도된 단위스위치와 동일한 단위스위치와) 실제로 셀이 도착한 단위스위치 간의 위상적 거리에 따라 하나의 클래스로 분류한다. 따라서 제 i 단에서의 셀을 $2^{i-2} + 1$ 클래스들 중 하나로 분류 한다. 그리고 위상적 거리가 0인 셀을 클래스 0로 분류한다.

셀과 연관된 클래스는 셀이 하나의 단위 스위치에서 동일한 단 또는 다음 단의 다른 단위 스위치로 보내짐에 따라 다른 클래스로 변한다. 하나의 클래스의 셀이 다른 클래스의 셀로 변환 할률은 그 셀이 하나의 단위 스위치에서 다른 셀들과 경쟁해서 이길 확률에만 의존적이다.

우리는 먼저 [18]에서와 비슷한 방법으로, 상태방정식을 간략하게 기술하기 위한 변수들을 다음에서 정의한다. 그리고 이러한 변수들과 연관된 상태 방정식들의 집합을 유도한다.

정의 3 기호 $x_{i,j}^{(l)}$ 는 특정 단 i 의 단위 스위치에서 특정 단 j 의 단위 스위치로 옮겨지면서 새로운 클래스 l 이 되는 셀들의 수를 나타낸다.

또한 $x_{i,j}$ 를 이전에 특정 단 i 의 단위 스위치에서 특정 단 j 의 단위 스위치로 옮겨지는 모든 셀들의 수를 나타내는 기호로 정의 한다.

정의 4 제 j 단에 있는 이전에 i 단에 있었던 셀들의 총 개수 $x_{i,j}$ 는 다음과 같이 정의 된다.

$$x_{i,j} = \sum_{l=0}^{2^{j-2}} x_{i,j}^{(l)} \tag{5}$$

이제 제시될 상태 방정식들의 표현을 간략화하는 데 쓰일 약간의 약어들을 정의하겠다.

정의 4 기호 L_i 는 하나의 SE의 모든 입력 링크들의 개수를 나타내고, 기호 L_c 는 하나의 SE의 모든 연쇄-입력 링크들의 개수를 나타낸다. 제 $i-1$ 단을 떠나 제 i 단의 SE의 입력 링크들에 도착하는 클래스 0의 셀들을 제외한 모든 셀들의 개수 $\bar{\chi}_{i-1,i}^{(0)}$ 는 다음과 같다.

$$\bar{\chi}_{i-1,i}^{(0)} \equiv \chi_{i-1,i} - \chi_{i-1,i}^{(0)} \quad (6)$$

제 i 단을 떠나 제 i 단의 SE의 연쇄-입력 링크들에 도착하는 클래스 0의 셀들 ($\chi_{i,i}^{(0)}$)를 제외한 모든 셀들의 개수를 다음과 같이 정의되는 기호로 정의한다.

$$\bar{\chi}_{i,i}^{(0)} \equiv \chi_{i,i} - \chi_{i,i}^{(0)} \quad (7)$$

기호 $\hat{\lambda}_{i,j}^{(k)(l)}$ 는 하나의 클래스의 셀이 다른 클래스의 셀로 변하는 천이 확률을 정의한다.

정의 5 기호 $\hat{\lambda}_{i,j}^{(k)(l)}$ 는 제 i 단의 클래스 k 의 셀이 제 j 단의 한 SE로 전달되면서 클래스 l 의 셀로 변하는 천이 확률이다.

그림 5에서 보이는 바와 같이, 기호 $\hat{\lambda}_{i,j}^{(k)(l)}$ 를 사용하여 제안된 성능 모델에서 사용되는 상태 천이도를 손쉽게 표현할 수 있다. 예시된 것은 16×16 사이클릭 베니안 네트워크의 경우이다. 각 상태는 단의 수와 셀이 실제 도착한 SE와 원래 의도한 SE (또는 그것과 동일한 SE)와의 위상적 거리차의 쌍으로 식별 되어 있다.

이러한 종류의 상태도와 다음에서 제시될 상태 천이 확률 $\hat{\lambda}_{i,j}^{(k)(l)}$ 를 사용하여 안정 상태에서의 사이클릭 베니안 네트워크의 통과율(Throughput)을 얻을 수 있다.

정리 1 상태 천이가 안정 상태에 있을 때, 첫 번째 단에서 클래스 0의 셀이 두 번째 단에 전달되면서 클래스 l 로 변하는 상태 천이 확률 $\hat{\lambda}_{1,2}^{(0)(l)}$ 는 다음과 같다.

$$\hat{\lambda}_{1,2}^{(0)(l)} = \sum_{\chi_{0,1}^{(0)}=0}^{L_1} \left[\binom{L_1}{\chi_{0,1}^{(0)}} (1 - \lambda_{0,1})^{L_1 - \chi_{0,1}^{(0)}} \cdot (\lambda_{0,1}^{(0)})^{\chi_{0,1}^{(0)}} \cdot P_{1,2}^{(0)(l)}(\chi_{0,1}^{(0)}, 0, 0, 0) \right] \quad (8)$$

제 i 번째 단 (여기서 $i \geq 2$)의 클래스 k 의 셀이 제 j 번째 단의 한 SE에 전달되면서, 클래스 l 로 변환될 확률 $\hat{\lambda}_{i,j}^{(k)(l)}$ 는 다음과 같다.

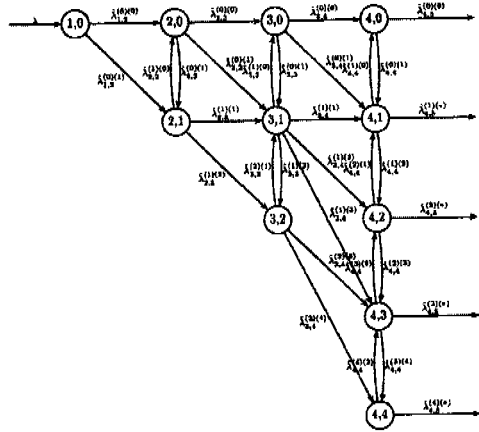


그림 2. 사이클릭 베니안 네트워크를 통해서 스위치되는 셀들의 상태 천이들

$$\hat{\lambda}_{i,j}^{(k)(l)} = \sum_{\chi_{i-1,i}^{(0)}=0}^{L_i} \sum_{\chi_{i-1,i}^{(1)}=0}^{L_i - \chi_{i-1,i}^{(0)}} \dots \sum_{\chi_{i-1,i}^{(2^{i-2})}=0}^{L_i - \chi_{i-1,i}^{(0)} - \dots - \chi_{i-1,i}^{(2^{i-2}-1)}} \left[\binom{L_i}{\chi_{i-1,i}^{(0)}, \chi_{i-1,i}^{(1)}, \dots, \chi_{i-1,i}^{(2^{i-2})}} \cdot (1 - \lambda_{i-1,i})^{L_i - \chi_{i-1,i}} \prod_{l=0}^{2^i-2} (\hat{\lambda}_{i-1,i}^{(l)})^{\chi_{i-1,i}^{(l)}} \cdot \sum_{\chi_{i,i}^{(0)}=0}^{L_c - \chi_{i,i}^{(1)} - \dots - \chi_{i,i}^{(2^{i-2}-1)}} \binom{L_c}{\chi_{i,i}^{(0)}, \chi_{i,i}^{(1)}, \dots, \chi_{i,i}^{(2^{i-2})}} (1 - \lambda_{i,i})^{L_c - \chi_{i,i}} \cdot \prod_{l=0}^{2^i-2} (\hat{\lambda}_{i,i}^{(l)})^{\chi_{i,i}^{(l)}} \cdot P_{i,j}^{(k)(l)}(\chi_{i-1,i}^{(0)}, \bar{\chi}_{i-1,i}^{(0)}, \chi_{i,i}^{(0)}, \bar{\chi}_{i,i}^{(0)}) \right]$$

여기서 $\hat{\lambda}_{h,i}^{(l)} = \sum_{a=0}^{2^h-2} \hat{\lambda}_{h,i}^{(k)(l)} h \in \{i-1, i\}, j \in \{i, i+1\}$

이고, $l \in \{0, 1, \dots, 2^i-2\}$ 이다. ■

식 (8)을 식의 구성 요소별로 설명하면 다음과 같다. 식에 나타난 다음의 이항 확률 함수는

$$\binom{L_i}{\chi_{0,1}^{(0)}} (1 - \lambda_{0,1})^{L_i - \chi_{0,1}^{(0)}} \cdot (\lambda_{0,1}^{(0)})^{\chi_{0,1}^{(0)}}$$

주어진 셀 시간에 첫 단의 한 SE의 L_1 개의 입력 링크들 중 " $\chi_{0,1}^{(0)}$ " 입력 링크들에 모두 " $\chi_{0,1}^{(0)}$ " 셀들이 도착할 확률이다. 그리고 확률 $P_{1,2}^{(0)(l)}(\chi_{0,1}^{(0)}, 0, 0, 0)$ 은 첫 번째 단의 하나의 SE안의 클래스 0의 " $\chi_{0,1}^{(0)}$ " 셀중 적어도 하나가 클래스 l 로 변하며, 출력 링크를 통하여 두 번째 단에 전달될 확률이다. 따라서 천이 확

를 $\hat{\lambda}_{1,2}^{(0)(l)}$ 은 이 두 확률 함수들의 곱들의 합이다.

식 (9)를 구성 요소별로 설명하면 다음과 같다. 다음의 다항(Multinomial) 확률 함수는

$$\binom{L_i}{\chi_{i-1,i}^{(0)}, \chi_{i-1,i}^{(1)}, \dots, \chi_{i-1,i}^{(2^{i-2})}} (1 - \hat{\lambda}_{i-1,i})^{L_i - \chi_{i-1,i}} \cdot \prod_{l=0}^{2^i-2} (\hat{\lambda}_{i-1,i}^{(l)})^{\chi_{i-1,i}^{(l)}} \quad (10)$$

제 $i-1$ 번째 단으로부터 클래스 0의 " $\chi_{i-1,i}^{(0)}$ "개의 셀들이, 클래스 1의 $\chi_{i-1,i}^{(1)}$ 개의 셀들, 그리고 클래스 2^{i-2} 의 $\chi_{i-1,i}^{(2^{i-2})}$ 개의 셀들이 제 i 번째 단의 하나의 SE의 " L_i "개의 입력 링크들 중 " $\chi_{i-1,i}$ "개의 입력 링크에 도착할 결합 확률 함수이다.

식 (9)에 있는 다음의 다항 확률 함수는

$$\binom{L_c}{\chi_{i,i}^{(0)}, \chi_{i,i}^{(1)}, \dots, \chi_{i,i}^{(2^{i-2})}} (1 - \hat{\lambda}_{i,i})^{L_c - \chi_{i,i}} \cdot \prod_{l=0}^{2^i-2} (\hat{\lambda}_{i,i}^{(l)})^{\chi_{i,i}^{(l)}} \quad (11)$$

제 i 번째 단으로부터 클래스 0의 " $\chi_{i,i}^{(0)}$ "개의 셀들과 클래스 1의 $\chi_{i,i}^{(1)}$ 개의 셀들, 그리고 클래스 2^{i-2} 의 $\chi_{i,i}^{(2^{i-2})}$ 개의 셀들이 제 i 번째 단의 하나의 SE의 " L_c "개의 연결-입력 링크들 중 " $\chi_{i,i}$ "개의 연결-입력 링크에 도착할 결합 확률 함수이다.

식 (9)의 확률 함수 $P_{i,j}^{(k)(l)}(\chi_{i-1,i}^{(0)}, \bar{\chi}_{i-1,i}^{(0)}, \chi_{i,i}^{(0)}, \bar{\chi}_{i,i}^{(0)})$ 는 제 i 번째 단의 한 SE안의 클래스 k 의 " $\chi_{i,j}^{(k)(l)}$ "개의 셀들의 적어도 하나가 클래스 l 로 변하며, 특정 하나의 출력 링크를 통하여 제 j 번째 단에 전달될 확률이다. 이는 다음에서 유도할 것이다. 그러면 천이 확률 $\hat{\lambda}_{i,j}^{(k)(l)}$ 은 입력 링크들에 도착하는 모든 가능한 셀들의 수에 관한 위에서 설명한 세 확률 함수들의 곱들의 합이다.

이제 확률 함수 $P_{i,j}^{(k)(l)}$ 를 유도해보자. 편향 자기 경로제어에 따라 SE 사이에 스위칭 되는 입력셀들의 클래스의 변환을 모델링하기 위해서, 지금부터 공식들을 유도한다. 먼저 확률 변수 $P_{i,j}^{(0)(l)}$, 다시 말해서 하나의 SE에 도달하는 모든 입력셀들이 클래스 0인 경우, 즉 $\bar{\chi}_{i-1,i}^{(0)} = 0$ 이고 $\chi_{i,i} - \chi_{i,i}^{(0)} = 0$ 인 경우를 다음 보조 정리 1에서 정의해보자.

보조 정리 1 제 i 번째 단의 하나의 SE안의 클래스 0의 모든 셀들 중 적어도 하나가 클래스 0를 유지하며, 특정 하나의 출력 링크를 통하여 제 $i+1$ 번째 단에 전달될 확률 함수 $P_{i,i+1}^{(0)(0)}$ 는 다음으로 정의된다.

$$P_{i,i+1}^{(0)(0)}(\chi_{i-1,i}^{(0)}, \bar{\chi}_{i-1,i}^{(0)}, \chi_{i,i}^{(0)}, \bar{\chi}_{i,i}^{(0)}) = 1 - \left(\frac{1}{2}\right)^{(\chi_{i-1,i}^{(0)} + \chi_{i,i}^{(0)})} \quad (12)$$

제 i 번째 단의 하나의 SE안의 클래스 0의 모든 셀들 중 적어도 하나가 클래스 1로 되며, 특정 하나의 출력 링크를 통하여 제 $i+1$ 번째 단에 전달될 확률 함수 $P_{i,i+1}^{(0)(1)}$ 은 다음으로 정의된다.

$$P_{i,i+1}^{(0)(1)}(\chi_{i-1,i}^{(0)}, \bar{\chi}_{i-1,i}^{(0)}, \chi_{i,i}^{(0)}, \bar{\chi}_{i,i}^{(0)}) = \left(\frac{1}{2}\right)^{(\chi_{i-1,i}^{(0)} + \chi_{i,i}^{(0)})} \quad (13)$$

제 i 번째 단의 하나의 SE안의 클래스 0의 모든 셀들 중 적어도 하나가 클래스 1로 되며, 특정 하나의 연쇄-출력 링크를 통하여 제 i 번째 단에 전달될 확률 함수 $P_{i,i}^{(0)(1)}$ 는 다음으로 정의된다.

$$P_{i,i}^{(0)(1)}(\chi_{i-1,i}^{(0)}, \bar{\chi}_{i-1,i}^{(0)}, \chi_{i,i}^{(0)}, \bar{\chi}_{i,i}^{(0)}) = \begin{cases} \frac{1}{2}, & \text{if } \chi_{i-1,i}^{(0)} + \chi_{i,i}^{(0)} = 3 \\ 1, & \text{if } \chi_{i-1,i}^{(0)} + \chi_{i,i}^{(0)} = 4 \end{cases} \quad (14)$$

다음으로, 동일한 단 안에서의 라우팅을 통해서 편향 정도를 한 수준 줄이는 경우의 확률 함수를 정의한다. 확률함수의 수학적 표현식들을 간략화 하기 위해서, 다음 정의 6에서의 행렬들을 도입한다.

정의 6 행렬 \bar{P}_i 와 행렬 \bar{C}_i 를 각각 다음과 같이 정의 하자.

$$\bar{P}_i = (\hat{\lambda}_{i-1,i}^{(k)(2k+j)} \hat{\lambda}_{i-1,i}^{(0)(1)}) \quad (15)$$

$$\bar{C}_i = (\hat{\lambda}_{i,i}^{(k)(k-1)} \hat{\lambda}_{i,i}^{(k)(k+1)} \hat{\lambda}_{i,i}^{(0)(1)}) \quad (16)$$

여기서

$$\hat{\lambda}_{i-1,i}^{(k)(2k+j)} = \sum_{s=1}^{2^i-3} \hat{\lambda}_{i-1,i}^{(s)(2s+j)} \quad (17)$$

$$\hat{\lambda}_{i,i}^{(k)(k-1)} = \sum_{s=2}^{2^i-2} \hat{\lambda}_{i,i}^{(s)(s-1)} \quad (18)$$

$$\hat{\lambda}_{i,i}^{(k)(k+1)} = \sum_{s=1}^{2^i-2} \hat{\lambda}_{i,i}^{(s)(s+1)} \quad (19)$$

$$j \in \{-1, 0, 1\}.$$

추가로, 다음과 같이 세가지의 행렬 연산자 기호들을 정의 하자. 먼저 기호 " \odot "는 벡터의 스칼라 곱과 유사한 두 행렬의 곱을 나타낸다.

$$\begin{pmatrix} a_{00} & \dots & a_{i0} \\ \vdots & & \vdots \\ a_{0y} & \dots & a_{xy} \end{pmatrix} \odot \begin{pmatrix} b_{00} & \dots & b_{x0} \\ \vdots & & \vdots \\ b_{0y} & \dots & b_{xy} \end{pmatrix} \equiv \sum_{i,j} a_{ij} \cdot b_{ij} \quad (20)$$

기호 " \otimes "는 벡터의 데카르트 곱과 유사한 두 행렬의 곱을 나타낸다.

$$\begin{pmatrix} a_{00} & \cdots & a_{x0} \\ \vdots & & \vdots \\ a_{0y} & \cdots & a_{xy} \end{pmatrix} \otimes (b_0 \cdots b_w)$$

$$\equiv \begin{pmatrix} a_{00}b_0 & \cdots & a_{00}b_w \\ a_{10}b_0 & \cdots & a_{10}b_w \\ \vdots & & \vdots \\ a_{x0}b_0 & \cdots & a_{x0}b_w \\ a_{01}b_0 & \cdots & a_{01}b_w \\ \vdots & & \vdots \\ a_{xy}b_0 & \cdots & a_{xy}b_w \end{pmatrix} \quad (21)$$

마지막으로 기호 “ \cdot ”는 벡터의 구성요소들의 합과 비슷한 다음과 같은 행렬의 평가를 나타낸다.

$$\left| \begin{pmatrix} a_{00} & \cdots & a_{x0} \\ \vdots & & \vdots \\ a_{0y} & \cdots & a_{xy} \end{pmatrix} \right| \equiv \sum_{i,j} a_{ij} \quad (22)$$

위에서 정의한 두개의 행렬들과 세 가지의 연산자들을 사용하여, 우리는 천이 확률함수 $P_{i,i}^{(k)(k-1)}$ 를 다음 보조 정리 2 에서와 같이 정의할 수 있다.

보조 정리 2 제 i 단의 하나의 SE 안에 있는 클래스 k 의 입력 셀들 모두 중 적어도 하나가 특정 연쇄-출력 링크를 통해서 전달되며, 클래스 $k-1$ 이 될 확률 함수 $P_{i,i}^{(k)(k-1)}$ 는 다음과 같이 정의 된다.

$$P_{i,i}^{(k)(k-1)}(\chi_{i-1,i}^{(0)}, \bar{\chi}_{i-1,i}^{(0)}, \chi_{i,i}^{(0)}, \bar{\chi}_{i,i}^{(0)}) = \begin{cases} \frac{1}{2}, & \text{if } \bar{\chi}_{i-1,i}^{(0)} + \bar{\chi}_{i,i}^{(0)} = 1 \\ \left| \left(\begin{pmatrix} \frac{1}{2} & \frac{3}{4} \\ \frac{3}{4} & \frac{1}{2} \end{pmatrix} \otimes (\bar{P}_i \otimes \bar{P}_i) \right) \right|, & \text{if } \bar{\chi}_{i-1,i}^{(0)} = 2 \text{ and } \bar{\chi}_{i,i}^{(0)} = 0 \\ \left| \left(\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{4} \end{pmatrix} \otimes (\bar{P}_i \otimes \bar{C}_i) \right) \right|, & \text{if } \bar{\chi}_{i-1,i}^{(0)} = 1 \text{ and } \bar{\chi}_{i,i}^{(0)} = 1 \\ \left| \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & 1 \\ \frac{1}{2} & 1 & 1 \end{pmatrix} \otimes (\bar{C}_i \otimes \bar{C}_i) \right|, & \text{if } \bar{\chi}_{i-1,i}^{(0)} = 0 \text{ and } \bar{\chi}_{i,i}^{(0)} = 2 \\ \left| \left(\left(\begin{pmatrix} \frac{1}{2} & \frac{7}{8} \\ \frac{7}{8} & \frac{1}{2} \end{pmatrix} \otimes (\bar{P}_i \otimes \bar{P}_i) \right) \otimes \bar{C}_i \right) \right|, & \text{if } \bar{\chi}_{i-1,i}^{(0)} = 2 \text{ and } \bar{\chi}_{i,i}^{(0)} = 1 \\ \left| \left(\left(\begin{pmatrix} 1 & \frac{3}{4} & \frac{3}{4} \\ \frac{3}{4} & 1 & 1 \\ \frac{3}{4} & 1 & 1 \end{pmatrix} \otimes (\bar{C}_i \otimes \bar{C}_i) \right) \otimes \bar{P}_i \right) \right|, & \text{if } \bar{\chi}_{i-1,i}^{(0)} = 1 \text{ and } \bar{\chi}_{i,i}^{(0)} = 2 \\ \left| \begin{pmatrix} 1 & \alpha_i & \alpha_i \\ \alpha_i & 1 & 1 \\ \alpha_i & 1 & 1 \end{pmatrix} \otimes (\bar{C}_i \otimes \bar{C}_i) \right|, & \text{if } \bar{\chi}_{i-1,i}^{(0)} = 2 \text{ and } \bar{\chi}_{i,i}^{(0)} = 2 \text{ and } \alpha_i \\ 0, & \text{if } \bar{\chi}_{i-1,i}^{(0)} = 0 \text{ and } \bar{\chi}_{i,i}^{(0)} = 0 \end{cases} \quad (23)$$

여기서

$$\alpha_i = \left| \begin{pmatrix} \frac{3}{4} & \frac{7}{8} \\ \frac{7}{8} & \frac{3}{4} \end{pmatrix} \otimes (\bar{P}_i \otimes \bar{P}_i) \right|. \quad (24)$$

마지막으로, 우리는 스위칭 경쟁에서 저서 편향의 정도가 증가하는 경우의 확률 함수를 정의 한다.

보조 정리 3 제 i 단의 하나의 SE 안에 있는 클래스 k 의 입력 셀들 모두 중 적어도 하나가 특정 연쇄-출력 링크를 통해서 전달되며, 클래스 $k+1$ 이 될 확률 함수 $P_{i,i}^{(k)(k+1)}$ 는 다음과 같이 정의 된다.

$$P_{i,i}^{(k)(k+1)}(\chi_{i-1,i}^{(0)}, \bar{\chi}_{i-1,i}^{(0)}, \chi_{i,i}^{(0)}, \bar{\chi}_{i,i}^{(0)}) = 1 - P_{i,i}^{(k)(k-1)}(\chi_{i-1,i}^{(0)}, \bar{\chi}_{i-1,i}^{(0)}, \chi_{i,i}^{(0)}, \bar{\chi}_{i,i}^{(0)}), \quad \text{만약 } \bar{\chi}_{i-1,i}^{(0)} + \bar{\chi}_{i,i}^{(0)} > 1. \quad (25)$$

제 i 단의 하나의 SE 안에 있는 클래스 k 의 입력 셀들 모두 중 적어도 하나가 특정 출력 링크를 통해서 전달되며, 클래스 $2k+j$ 이 될 확률 함수 $P_{i,i+1}^{(k)(2k+j)}$ 는 다음과 같이 정의 된다.

$$P_{i,i+1}^{(k)(2k+j)}(\chi_{i-1,i}^{(0)}, \bar{\chi}_{i-1,i}^{(0)}, \chi_{i,i}^{(0)}, \bar{\chi}_{i,i}^{(0)}) = \begin{cases} \frac{1}{2} \cdot \left(\frac{1}{2}\right)^{|j|+1}, & \text{if } \bar{\chi}_{i-1,i}^{(0)} + \bar{\chi}_{i,i}^{(0)} = 3 \\ 1 \cdot \left(\frac{1}{2}\right)^{|j|+1}, & \text{if } \bar{\chi}_{i-1,i}^{(0)} + \bar{\chi}_{i,i}^{(0)} = 4 \end{cases} \quad j \in \{-1, 0, 1\}. \quad (26)$$

제 i 단의 하나의 SE 안에 있는 클래스 2^{i-2} 의 입력셀들 모두 중 적어도 하나가 특정 출력 링크를 통해서 전달되며, 클래스 2^{i-1} 이 될 확률 함수 $P_{i,i+1}^{(2^{i-2})(2^{i-1})}$ 는 다음과 같이 정의 된다.

$$P_{i,i+1}^{(2^{i-2})(2^{i-1})}(\chi_{i-1,i}^{(0)}, \bar{\chi}_{i-1,i}^{(0)}, \chi_{i,i}^{(0)}, \bar{\chi}_{i,i}^{(0)}) = \begin{cases} \frac{1}{2} \cdot \frac{1}{2}, & \text{if } \bar{\chi}_{i-1,i}^{(0)} + \bar{\chi}_{i,i}^{(0)} = 3 \\ 1 \cdot \frac{1}{2}, & \text{if } \bar{\chi}_{i-1,i}^{(0)} + \bar{\chi}_{i,i}^{(0)} = 4. \end{cases} \quad (27)$$

알고리즘 상태 천이의 복잡성 때문에 반복적 계산에 의해서 스위칭 망의 성능을 측정한다. 각 반복 단계는 첫번째 단 부터 시작해서 마지막 단을 끝으로 한다. 각 계산 단계들은 안정 상태에 도달할 때 까지 계산을 반복한다. 정규화된 통과율 $T(N)$ 은 가능한 셀 도착의 최대 수로 목적지에 도달한 셀들의 수를 나누어서 얻는다.

$$T(N) = \frac{\lambda_{\log_2 N-1, \log_2 N}^{(0)(0)} + \lambda_{\log_2 N, \log_2 N}^{(1)(0)}}{\lambda}. \quad (28)$$

정규화된 지연 $D(N)$ 은 스위칭 네트워크의 단들의 라우팅 횟수와 단 안에서의 라우팅의 기대치의 합을 단들의 수로 나누어 얻는다.

$$D(N) = 1 + \frac{1}{\log_2 N} \sum_{i=2}^{\log_2 N} \frac{|\vec{C}_i|}{|\vec{P}_{i+1}| + \lambda_{i,i+1}^{(0,0)}} \quad (29)$$

시뮬레이션: 제시한 분석을 검증하기 위하여, 우리는 사이클릭-베니안 네트워크의 시뮬레이션을 행했다. 분석의 기본 가정들은 다음과 같이 시뮬레이터에서 구현 되었다.

- N 개의 근원들은 셀들을 비율 λ 의 포아송 확률 과정에 따라서 생성한다. (여기서 $0 \leq \lambda \leq 1$)
- 각 셀의 목적지는 정규 분포 트래픽을 시뮬레이션하기 위해, 0 부터 $N - 1$ 사이의 수를 생성하는 임의의 수 생성기를 사용하여 임의로 설정한다.
- 하나의 링크를 위해 셀들간에 충돌이 있으면, 하나의 셀이 임의로 선택되어 링크로 보내지고, 다른 셀들은 다른 링크들로 우회되어 진다.
- 통과율과 지연시간은 각 출력 포트에서 측정되어지고, 네트워크의 크기로 평균되어진다. 시뮬레이션의 시간은 정규화된 통과율과 정규화된 지연시간을 얻는데까지 진행된다.

표본 시뮬레이션 결과들과 분석적 결과들을 그림 3과 그림 4에 보였다. 그림 3은 입력 부하 1.0에서의 사이클릭 베니안 네트워크의 크기의 변화에 대한 정규화된 통과율을 보인다. 그림 3에서 보여지는 괄목할만한 사실은 정규화된 통과율이 크기 N 이 2^6 에 도달했을 경우부터 감소하기 시작한다는 것이다. 이것은 사이클릭-베니안 스위치의 교차점의 수 ($N/2 \times \log_2 N \times 16$)가 크로스바의 그것 (N^2) 보다 작아지기 시작하기 때문이다. 그림 4는 입력 부하가 1.0 인 동 네트워크의 크기 변화에 대한 정규화된 지연을 보인다. 분석 결과들과 시뮬레이션 결과들간에는 차이가 있으나, 이는 해석적 분석이 근사법으

로 만들어졌고, 몇가지 균일성과 독립 가정들에 근거를 두고 만들었기 때문에 불가피하게 발생한 것이다. 근사법과 가정은 분석을 간단하고, 이해하기 쉬우며, 결과를 산출하는 것을 간단하게 만들어 주는 대신 이러한 오차를 발생시킨다.

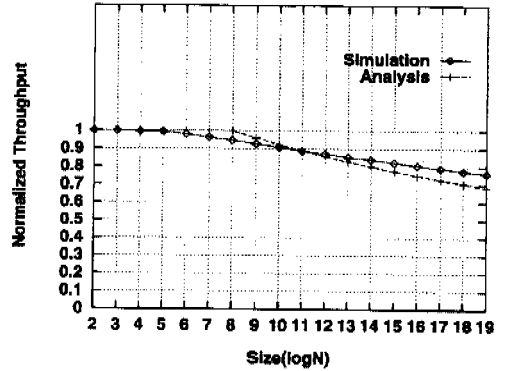


그림 3. 사이클릭 베니안 네트워크의 네트워크 크기의 변화에 대한 정규화된 통과율

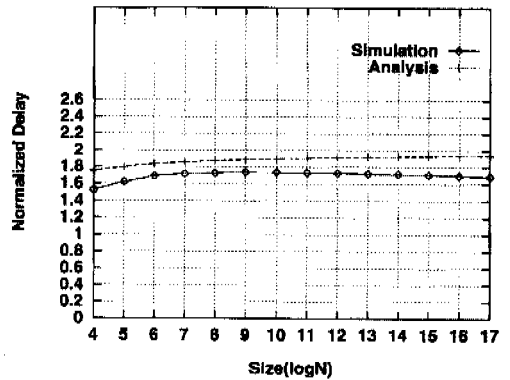


그림 4. 사이클릭 베니안 네트워크의 네트워크 크기의 변화에 대한 정규화된 지연

이제, 사이클릭-베니안 네트워크의 성능과 일반 베니안 네트워크 및 중복 베니안 네트워크들간의 성능비교

표 1. 베니안, 중복 베니안들, B-network, 사이클릭-베니안 ($N = 1024$)의 정규화된 하드웨어 복잡도

Component	Banyan	Replicated-2	Cyclic banyan	B-Network	Replicated-4	Replicated-8
Input Cell Buffer	7168	14336	14336	21504	28672	57344
Complete Interconnection	454	908	1812	2040	1816	3632
Output Cell Latch	224	448	448	672	896	1792
Contention Control	54	108	196	204	216	432
Subtotal	7900	15800	16792	24480	31600	63200
Deviation-tag Processing	0	0	1948	0	0	0
Additional Contention Control	0	0	448	672	0	0
Total	7900	15800	19188	25152	31600	63200

를 하자. 네트워크들 간의 공정한 비교를 위해서, 먼저 네트워크들의 하드웨어 비용을 분석한다. 하드웨어 비용은 구현결과 사용된 게이트들의 수에 비례한다고 가정한다. 그러면, 일반적인 베니안 네트워크의 하드웨어 비용은 $(N/2 \times \log_2 N \text{ SE 개수}) \times (\text{SE 당 게이트들의 수})$ 가 되고, 중복2 베니안 네트워크는 $(N/2 \times \log_2 N \text{ SE 개수}) \times (\text{SE 당 게이트들의 수}) \times (2 \text{ 중복})$ 이 되고, 사이클릭-베니안 네트워크는 $(N/2 \times \log_2 N \text{ SE 개수}) \times (\text{SE 당 게이트들의 수})$ 가 되고, B-network는 $(N \times \log_2 N \text{ SE 개수}) \times (\text{SE 당 게이트들의 수})$ 가, 중복4 베니안 네트워크는 $(N/2 \times \log_2 N \text{ SE 개수}) \times (\text{SE 당 게이트들의 수}) \times (4 \text{ 중복})$ 이 된다. 또한 중복8 베니안 네트워크는 $(N/2 \times \log_2 N \text{ SE 개수}) \times (\text{SE 당 게이트들의 수}) \times (8 \text{ 중복})$ 이 된다.

따라서 우리는 이러한 하드웨어 비용에 관한 수식들을 일반 $N \times N$ 베니안 네트워크의 모든 SE들의 개수인 $(N/2 \times \log_2 N)$ 으로 나누어 정규화할 수 있다. 이러한 정규화된 하드웨어 비용은 결국 일반적인 베니안 네트워크의 하나의 SE에 대응하는 기능을 만드는 데 필요한 게이트들의 수를 의미한다.

일반적인 베니안 네트워크를 만들기 위한 정규화된 하드웨어 비용, 그리고 중복2 베니안 네트워크를 위한 비용, 사이클릭-베니안 네트워크를 위한 비용, 중복4 베니안 네트워크를 위한 비용, 중복8 베니안 네트워크를 위한 비용은 각각 (SE당 게이트들의 수), (SE당 게이트들의 수 \times 2), (SE당 게이트들의 수), (SE당 게이트들의 수 \times 2), (SE당 게이트들의 수 \times 4), 그리고 (SE당 게이트들의 수 \times 8)로 산출할 수 있다.

$N = 1024$ 크기의 네트워크를 위한 하드웨어 비용의 상세한 값들은 표 1에 보이고 있다. 한 SE의 게이트들의 수를 계산하기 위해서, 우리는 다음과 같은 가정들을 사용한다. 우리는 NAND 논리회로를 위해서 한개의 게이트가 필요하고, 배타적 OR 회로를 위해서는 3개의 게이트들이 필요하며, 하나의 플립플롭을 위해서는 7개의 게이트들이 필요하다고 가정한다. 우리는 또한 각 SE들이 입력 셀 버퍼들과, 출력 셀 래치들, 그리고 입력셀 버퍼들에서 출력셀 래치들로 셀을 전송할 수 있는 완전 상호연결로 구성된다고 가정한다. 이들은 플립플롭들로 구성되며, 또한 일반적인 충돌 제어 기능을 가졌다고 가정한다.

입력 셀 버퍼는 64 비트들을 저장하기 위한 512개의 플립플롭들로 구성된다. 출력 셀 래치는 2 비트들을 저장하기 위한 16개의 플립플롭들로 구성

된다. 완전 상호연결을 위해 필요한 게이트들의 수는 (입력셀 버퍼의 수) \times (2 비트 대역의 입력셀 래치의 플립플롭수) \times 플립플롭당 게이트수 + (출력 셀 래치의 수) \times (출력 멀티플렉서의 게이트의 수) \times (2 비트의 대역)의 수식으로부터 구할 수 있다. 정의 2에 따라 편향 꼬리표 k 를 처리하고 갱신하기 위해서는 추가적으로 약간의 게이트들이 필요하고, 또한 충돌을 제어하기 위해서도 약간의 추가적인 게이트들이 필요하다. 편향 꼬리표 k 를 위해 추가적으로 필요한 게이트들의 수는 $N \times N$ 네트워크에 경우에 대해서 $O(\log_2 N)$ 에 비례한다. 네트워크의 크기와는 상관없이, SE를 위한 게이트들의 수는 거의 불변이다. 그러므로 다양한 크기의 네트워크들의 정규화된 하드웨어 비용들은 거의 모두 동일하다.

표 1에 따르면, 사이클릭-베니안 네트워크를 만들기 위해서, 중복2 베니안 보다 약 20%의 추가적인 게이트들이 필요하다. 사이클릭-베니안의 하드웨어 비용은 중복2 베니안 네트워크의 1.2배가 되고, B-network의 0.8 배가 된다. 또한 중복4 베니안 네트워크의 0.6배, 그리고 중복8 베니안 네트워크의 0.3 배이다.

그림 5에서 보는 바와 같이, 1024×1024 사이클릭-베니안 네트워크의 통과율은 일반적인 베니안 네트워크의 3.51 배, 중복2 베니안 네트워크의 2.15배가 되고, 또한 중복4 베니안 네트워크의 1.51배, 그리고 중복8 베니안 네트워크의 1.20배이다. 그림 4에서 보는 바와 같이, 정규화된 지연은 모든 크기에 있어서, 일반적인 베니안 네트워크의 1.7배 이다. 더 나아가, 그림 6에서 보듯이, 입력 부하를 줄이면, 혹은 스피드업을 하면, 우리는 통과율이 거의 1인 스위칭 네트워크를 얻게 된다.

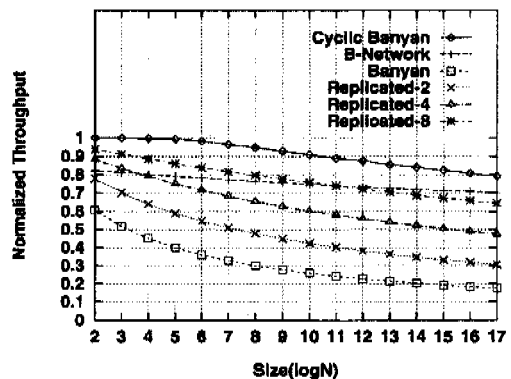


그림 5. 베니안, 중복베니안들, B-Network, 사이클릭 베니안의 크기 변화에 대한 정규화된 통과율

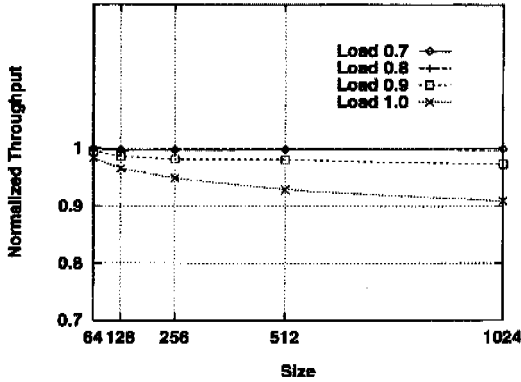


그림 6. 다양한 입력부하들 하에서 사이클릭 베니안 네트워크 크기의 변화에 대한 정규화된 통과율

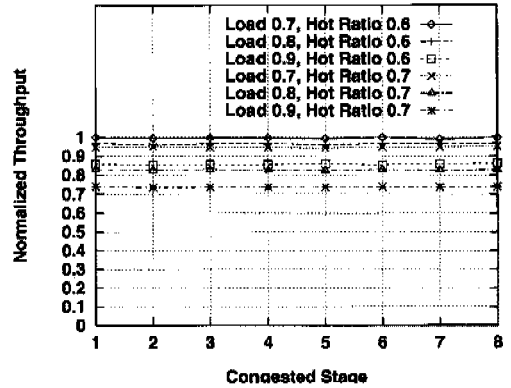


그림 7. 비균일 트래픽하에서의 512 × 512 사이클릭 베니안 네트워크의 혼잡 단계에 대한 정규화된 통과율

네트워크들의 크기가 커짐에 따라, 사이클릭 베니안 네트워크의 성능은 중복 베니안 네트워크보다 좋아진다. 따라서 사이클릭 베니안 네트워크는 중복 베니안보다 더 대용량 셀 스위치에 적합한 확장성을 가진다.

2. 비균일 트래픽 모델

본 절에서는 사이클릭 베니안 네트워크를 비균일 (Nonuniform) 트래픽 하에서 분석한다. 성능분석을 위해서, 우리는 시뮬레이션을 행하였다. 우리는 가정 1을 제외하고는 균일한 트래픽 하의 분석과 동일한 가정들을 사용한다. 비균일 트래픽은 $N \times N$ 부하 행렬 $D = \{\lambda_{ij}\}$ 로 나타낼 수 있는데, 여기서 λ_{ij} 는 출력 포트 j 가 목적지인 하나의 셀이 입력 포트 i 에 도달할 확률을 의미한다. 따라서, 행렬의 제 i 행의 합은 입력포트 i 에 부가되는 전체 부하를 나타내고, 제 j 열의 합은 출력 포트 j 에 부가되는 총 입력 부하를 나타낸다. 비균일 트래픽 패턴들은 그 가짓수가 무한대이다.

이러한 패턴들 중 실제적인 트래픽 패턴들을 대표하는 hot-group model [19] 이라고 하는 특정 비균일 트래픽 패턴에 주목한다. 입력 셀들은 모든 출력 포트들에 다음과 같이 비균일적으로 분포 한다. 즉 출력 포트는 “격렬한 출력 집단”과 “대체적으로 쉬는 출력 집단”의 두개의 그룹들로 나뉜다. 부하 행렬 D 는 $D = [D_h(\lambda_h) D_c(\lambda_c)]$ 로 나뉜다. D_h 의 행의 합은 λ_h 이고 D_c 의 행의 합은 λ_c 이다. 여기서 $\lambda_h + \lambda_c = \lambda$ 이다. 다른말로 하면, 동일한 크기의 두개의 출력 포트 집단이 있는데, 셀들이 목적지 집단 D_h 로 가는 확률은 λ_h 이고, 목적지 집단 D_c 로 갈 확률은 λ_c 이다. Hot ratio를 r 이라고 정의하면, $\lambda_h = r\lambda$ 이고, $\lambda_c = (1-r)\lambda$ 이다.

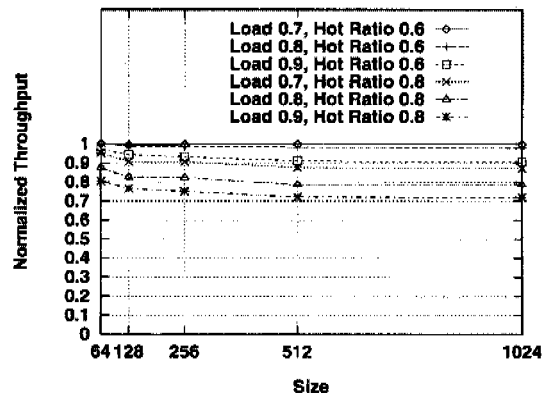


그림 8. 비균일 트래픽하에서의 사이클릭 베니안 네트워크의 네트워크 크기의 변화에 대한 정규화된 통과율

여기서 r 은 $0.5 \leq r \leq 1.0$ 이다.

시뮬레이션 결과들로부터 얻은 특기할 사항은 비균일 부하 하에서의 사이클릭-베니안 네트워크의 성능은 혼잡한 단계의 위치와 무관하다는 것이다. 그림 7 에서 이 사실을 볼 수 있다. 따라서 만약 두 출력 포트들의 그룹이 동일한 크기라면, 어떠한 비균일한 트래픽 패턴을 적용하더라도, 통과율을 측정하기 위한 시뮬레이션의 결과는 동일하다. 따라서 우리는 비균일 트래픽하에서의 모든 성능을 측정하기 위한 모형을 입력부하 λ 와 Hot Ratio r 로 표현할 수 있다.

이제 우리는 비균일 트래픽 하에서의 사이클릭 베니안 네트워크의 성능을 살펴보자. 그림 8 에 보이는 바와 같이, 입력 부하 $\lambda = 0.7$ 을 사이클릭 베니안 네트워크에 걸었을때, 만약 Hot Ratio r 을 0.6으로 했을때, 512 × 512 네트워크의 통과율은 단지 0.022%만 감소한다. Hot Ratio를 0.7로 했을때,

통과율은 2.871% 떨어진다. 심지어는 Hot Ratio가 0.8일때, 성능이 12.321%만이 떨어진다. 만약 입력 부하 $\lambda = 0.8$ 을 사이클릭 베니안 네트워크에 걸었을때, 만약 Hot Ratio r 을 0.6으로 했을때, 512×512 네트워크의 통과율은 단지 1.668%만 감소한다. Hot Ratio를 0.7로 했을때, 통과율은 11.230% 떨어진다. 만약 입력 부하 $\lambda = 0.9$ 을 사이클릭 베니안 네트워크에 걸었을때, 만약 Hot Ratio r 을 0.6으로 했을때, 512×512 네트워크의 통과율은 6.840%만 감소한다. Hot Ratio를 0.7로 했을때, 통과율은 17.502% 떨어진다. 입력 부하 $\lambda = 0.8$ 그리고 Hot Ratio r 을 0.6으로 했을때, 128×128 네트워크의 통과율은 0.99임을 그림 8이 잘 보여주고 있다. 이 경우 통과율은 균일한 트래픽하에서의 통과율에 비해서 단지 0.666%가 적다.

IV. 결론

본 논문에서는 완전 적응 자기 경로 제어 알고리즘을 사용하는 고장 감내 셀 스위치인 사이클릭 베니안 네트워크의 성능을 분석하였다. 이 분석을 통해, 이 스위치 구조가 기존 스위치보다 입력단과 출력단 사이에 더많은 다중 경로(Path)들을 제공함으로써, 짧은 시간의 스위치 내부 경로상의 혼잡 문제를 해결할 수 있다는 사실을 보였다. 제시된 분석적 모델은 대기행렬 모형에 기반을 두고 있으며, 이 모델의 정확성은 시뮬레이션을 통해서 보였다. 또한 인터넷에서 보편적이라 할 수 있는 비균일한 주소 분포를 가진 트래픽 부하에 대한 성능 분석을 시뮬레이션을 통해 보였다.

일반적인 베니안 네트워크와 본 사이클릭 베니안 네트워크, 그리고 다른 관련된 네트워크들의 균일한 트래픽 패턴과 비균일 트래픽 패턴 하에서의 정량적인 성능 비교를 행하여 다음과 같은 결과를 얻었다. 균일한 트래픽 하에서의 스위칭 네트워크의 분석으로부터, 크기가 1024×1024 인 사이클릭 베니안 네트워크의 통과율, 즉 입력 셀들의 수 대비 출력 셀들의 수의 비율이 정규 베니안의 것보다는 3.51 배 좋고, 중복-2 베니안 네트워크 보다는 2.15배 좋음을 발견했다. 또한 중복-4 베니안의 것 보다는 1.51배, 심지어 중복-8 베니안의 성능보다 1.20배 좋음을 알 수 있었다. 한편 하드웨어 복잡도는 중복-2 베니안에 비해 1.2배 높다. 모든 크기들의 사이클릭 베니안 네트워크의 지연은 모든 크기들의 정규 베니안 네트워크 각각의 지연의 약 1.7배에 불과하다. 네트워크의

크기가 커질수록 사이클릭 베니안 네트워크의 성능과 중복 베니안 네트워크의 성능차이는 더욱 벌어진다.

비균일 트래픽 하에서의 네트워크의 분석을 통해 다음과 같은 결과를 얻었다. 사이클릭 베니안 네트워크에 0.7의 부하를 걸었을 때, Hot-Ratio가 0.6이면, 512×512 네트워크는 단지 0.022% 만큼 출력이 떨어진다. Hot-Ratio가 0.7인 경우는, 출력이 2.871% 떨어지고, 심지어 Hot-Ratio가 0.8인 경우에도, 제안된 네트워크의 통과율은 12.321% 만큼만 떨어진다. 부하를 0.8을 걸었을 경우에는, Hot-Ratio가 0.6이면, 512×512 네트워크는 단지 1.668% 만큼 출력이 떨어지고, Hot-Ratio가 0.7인 경우는, 출력이 11.230% 떨어진다. 부하를 0.9을 걸었을 경우에는, Hot-Ratio가 0.6이면, 512×512 네트워크는 단지 6.840% 만큼 출력이 떨어지고, Hot-Ratio가 0.7인 경우는, 출력이 17.502% 떨어진다.

결과적으로 분석적 성능분석을 통해서, 본 스위치 구조가 완전 적응 자기 경로제어를 사용하여, 기존 스위치보다 입력단과 출력단 사이에 더많은 다중 경로(Path)들을 제공함으로써, 짧은 시간의 스위치 내부 경로상의 혼잡 문제를 해결할 수 있다는 사실을 보였고, 균일한 트래픽 패턴과 비균일 트래픽 패턴 하에서의 정량적인 성능 비교를 행하여, 사이클릭 스위치 구조가 베니안 네트워크이나 다른 네트워크보다 성능이 우수함을 보였다. 또한 본 논문에서 제안된 성능분석 방법은 우회 라우팅을 하는 다른 다단 스위칭 네트워크의 성능분석에도 적용이 가능함을 밝혀둔다.

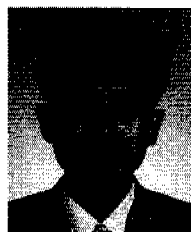
참고 문헌

- [1] G. Adams III, D.P. Agrawal, and H. J. Siegel, "A Survey and Comparison of Fault-Tolerant Multistage Interconnection Networks," *IEEE Computer*, Vol.20, No.6, pp. 14-27, June 1987.
- [2] G. B. Adams III and H. J. Siegal, "Modifications to Improve the Fault Tolerance of the Extra Stage Cube Interconnection Network," *Proc. 1984 Int'l Conf. Parallel Processing*, pp. 169-173, August, 1984.
- [3] G. R. Goke and G. J. Lipovski, "Banyan networks for partitioning multiprocessor systems," *Proc. 1st Annu. Symp. Computer Architecture*, pp. 21-28, 1973.

- [4] V. P. Kumer and S. M. Reddy, "Augmented Shuffle-Exchange Multistage Interconnection networks," *IEEE Computer*, Vol.20, No.6, pp. 30-40, June 1987.
- [5] C.-T. A. Lea, "Load-Sharing Banyan Network," *IEEE Trans. on Comp.*, Vol.C-35, No.12, pp. 1025-1034, Dec. 1986.
- [6] J.-H. Park, H. Yoon, H. Lee, and S. Eun, "The Ring-Banyan Network: A Fault-Tolerant Multistage Interconnection Network for Multiprocessor Systems," *International Journal of High Speed Computing*, pp. 557-577, World Scientific Publishing, U.S.A., 1994.
- [7] J.-H. Park, H. Yoon, and H. Lee, "The Cyclic Banyan Network: A Fault Tolerant Multistage Interconnection Network with the Fully-Adaptive Self-routing," *Seventh IEEE Symposium on Parallel and Distributed Processing*, pp. 702-710, October 1995, Texas, U.S.A.
- [8] J.-H. Park, H. Yoon, and H. Lee, "The Deflection Self-routing Banyan Network: A Large-Scale ATM Switch using the Fully-Adaptive Self-routing and its Performance Analyses," *IEEE/ACM Trans. on Networking*, Vol.7, No.4, pp. 588-604, IEEE, U.S.A., August 1999.
- [9] J.-H. Park and H. Lee, "A Fault-Tolerant Multistage Interconnection Network and Its Fault Diagnosis: Ring-Banyan Network," *International Journal of Computer Systems*, CRL Publishing Ltd., U.K., 1999, *Accepted*.
- [10] J.-H. Park, H. Yoon, and H. Lee, "Algebraic Properties of the Banyan Networks and Its Application to Fault Tolerant Fully-Adaptive Self-Routing in an Augmented Banyan Network," *Journal of Parallel and Distributed Computing*, Academic Press, U.S.A., 2000, *In minor revision*.
- [11] K. Y. Lee and H. Yoon, "The B-Network: A Multistage Interconnection Network with Backward Links," *IEEE Trans. on Comp.*, Vol.39, No.7, pp. 966-969, April 1990.
- [12] K. Padmanabhan, *Fault Tolerance and Performance Improvement in Multiprocessor Interconnection networks*, Ph.D. Thesis, Univ. of Illinois at Urbana-Champaign, Dept. of C.S Report No. UIUCDCS-R-84-1156, May, 1984.
- [13] A. Trew and G. Wilson, Eds., *Past, Present, Parallel: A Survey of Available Parallel Computing Systems*, Springer-Verlag, pp. 64-75, 1991.
- [14] N. Tzeng, P. Yew, and C. Zhu, "A Fault-Tolerant Scheme for Multistage Interconnection Networks," *Proc. 12th Int'l Symp. Computer Architecture*, pp. 368-375, June, 1985.
- [15] C. Wu and T. Feng, "On a Class of Multistage Interconnections Networks," *IEEE Trans. on Comp.*, vol.C-29, pp. 694-702, August, 1980.
- [16] R. Venkatesan and H. T. Mouftah, "Balanced gamma network-- A new candidate for broadband packet switch architectures, in *Proc. IEEE INFOCOM'92*, pp. 2482-2488., 1992.
- [17] T. D. Morris and E. F. Gehringer, "A cost-effective reliable multipath interconnection network," *ACM Comput. Architecture News*, pp. 45-65, June 1991.
- [18] M. Harchol and P. E. Black, "Queuing theory analysis of greedy routing on arrays and tori," Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, CA 94720, Tech. Rep. UCB/CSD 93/756, June 1993.
- [19] S. Gianatti and A. Pattavina, Performance analysis of shared-buffered Banyan networks under arbitrary traffic patterns, in *Proc. INFOCOM93*, pp. 943-952, 1993.
- [20] 박재현, "완전 적응 자기 경로제어 알고리즘을 사용하는 고장 감내 ATM 스위치 -- 사이클릭 베니안 네트워크," *한국통신학회 논문지*, Vol. 24, No. 9B, 9월, 1999.

박재현 (Jae-Hyun Park)

정회원



1988년 2월 : 중앙대학교 전자계산학과 졸업
 1991년 2월 : 한국 과학기술원 전산학과 석사
 1995년 8월 : 한국 과학기술원 전산학과 박사

1995년 8월~2000년 2월 : 삼성전자 정보통신 본부
데이터네트워크 개발팀 MPLS/ATM 개
발담당

2000년 3월~현재 : 영남대학교 전자정보공학부 교수
<주관심 분야> ATM Switch Arch., 상호연결 네트워크,
Multiprotocol Label Switching System,
Network Processors