

Asymmetric Cylinder : 비트 레벨 독자경로 라우팅 다중홉 네트워크

정회원 서승우*, 유지연**

Asymmetric Cylinder: A Bit-Level Self-Routing Multihop Network with Deflection

Ji Yon Youe*, Seung-Woo Seo** *Regular Members*

요 약

독자경로 라우팅과 같은 단순한 라우팅 알고리즘은 초고속 패킷망에 있어서 매우 중요한 역할을 한다. 이러한 독자경로 라우팅을 위하여 규칙적인 연결형상(topology)을 가진 여러 가지 네트워크 구조가 제안되었다. 이중 특히 셔플넷(Shufflenet)은, 구현 가능한 노드수에 대한 제약을 완화하기 위한 많은 연구가 수행되었다. 본 논문에서는 셔플넷에 기초한 새로운 네트워크 구조를 제안한다. 제안된 네트워크는 구현 가능한 노드수의 제약을 크게 완화하면서도 독자경로 라우팅을 가능하게 한다. 또한, 가교자 하는 경로에 여러 패킷이 몰릴 경우 우회 경로를 이용하는 디플렉션 라우팅(deflection routing)도 구현할 수 있으며, 이 때 패킷을 우회시키더라도 헤더의 정보를 변화시키지 않고 그대로 사용할 수 있도록 한다.

ABSTRACT

A simple routing algorithm such as self-routing plays a critical role in an ultra-fast multihop packet-switching network. To achieve this property, a variety of regular network topologies has been proposed. Among them, the Shufflenet has been modified and generalized by several researchers to alleviate the constraint on the number of realizable nodes. In this paper, a new network architecture based on the Shufflenet is presented. This network relieves the constraint on the number of nodes realizable greatly, but still exploits the simple self-routing algorithm. Deflection routing can be combined with self-routing in the proposed network, and even when a packet is deflected, the header information can still be reused without modification.

I. 서 론

광섬유는 매우 큰 대역폭(저손실 영역에서 약 25THz)을 가지고 있음에도 불구하고 지금까지 다중 사용자(혹은 노드) 네트워크 구성시 그 대역폭이 충분히 활용되지 못하고 있는데, 그 이유는 각 노드(node)에서의 전광/광전 변환에 따른 제약(opto-electronic conversion bottleneck) 때문이다^[1]. 이러한 문제점을 완화시키는 방안 중의 하나가 정형화

된 형상(regular topology)을 가지는 다중홉(multihop) 네트워크에서의 규칙적 연결 구조를 이용하여 라우팅 알고리즘을 간소화하고 노드의 처리 시간(processing time)을 줄임으로써 전체적인 통신망의 속도 향상을 얻는 것이다. 특히 전광 패킷 통신망(all-optical packet-switching network)에서는 복잡한 계산을 하는 것이 용이하지 않으므로, 단순한 라우팅 알고리즘이 필수적이다. 셔플넷(Shufflenet)^[2], de Bruijn graph^[3], Manhattan street network(MSN)^[4] 등이 정

* 서울대학교 전기공학부(sseo@plaza.snu.ac.kr)

** 서울대학교 전기공학부(jiyon@stargate.snu.ac.kr)

논문번호 : 98117-0316, 접수일자 : 1998년 3월 16일

※ 이 연구는 과학재단 특정기초과제(97-01-00-04-01-3) 지원하에 수행되었음.

형화된 형상을 가진 다중홉 네트워크의 예로 들 수 있다. 그러나 연결 형상의 규칙성은 구현 가능한 노드의 수에 제약을 주게되는 단점이 있다. 셔플넷의 경우, 구현 가능한 노드의 수 N 은 $N = kb^k$ 로 주어지며, 여기서 k 는 열(column), 또는 스테이지(stage)의 개수, b 는 노드의 입/출력 포트의 개수이다. de Bruijn graph의 경우, 노드 수 N 은 $N = b^D$ 로 주어지며, D 는 네트워크의 diameter로 정의된다. MSN의 노드 수는 자연수의 제곱, 즉 $N = n^2$ 이다. 이와 같이, 정형화된 형상의 네트워크에서는 구현 가능한 노드의 수가 매우 이산적인 분포를 가지게 된다. 이중 특히 셔플넷의 노드 수에 대한 제약을 완화하기 위한 많은 연구가 이루어졌는데, Shuffle ring^[5], Generalized Shuffle Network(GSN)^[6], GEMNET^[7] 등이 구현 가능한 노드수의 유연성에 초점을 두고 셔플넷을 개선한 것들이다.

본 논문에서는 Asymmetric Cylinder(AC)라 명명한 새로운 네트워크 구조를 제안한다. 제안된 네트워크 구조는 구현 가능한 노드수의 제한을 크게 완화하여, 기존의 셔플넷보다 훨씬 유연한 구조를 가질뿐 아니라, 비트 단위 독자경로 라우팅(bit-level self-routing)^[10]을 지원할 수 있다. 비트 단위 독자경로 라우팅을 사용하면 송신자와 수신자 사이의 중간 노드들은 패킷 헤더의 일부분만을 읽고도 패킷을 라우팅하여 목적지까지 보낼 수 있다. [10]에 제안된 독자 경로 라우팅 알고리즘은 구현 가능한 노드의 수가 $N = kb^k$ 로 정의되는 셔플넷에서만 적용이 되나, 본 논문에서는 이를 확장하여 제안된 네트워크 구조(AC)에서도 적용이 가능하도록 하였다. 또한 디플렉션 라우팅(deflection routing)^[8]을 구현할 수 있으며, 비트 단위 독자경로 라우팅과 병행하여 사용이 가능하다. 디플렉션 라우팅의 장점은, 패킷을 버퍼(buffer)에 저장하는 대신 우회시킴으로써, 매우 적은 양의 버퍼만으로 네트워크 운용이 용이하다는 것이다.

II. 셔플넷 기반 네트워크에서의 독자경로 라우팅

1. 노드가 패킷을 라우팅하기 위해 필요한 정보의 최소량

비트 단위 독자경로 라우팅에서는 노드가 패킷의 최종 목적지를 모르는 채, 패킷 헤더에 담긴 정보중 일부분만을 이용하여 라우팅을 하게 된다. 만약 p 개의 입/출력 포트를 가지는 노드들로 구성된 다중

홉 네트워크를 가정하면, 노드는 p 개의 출력 포트중 하나로 패킷을 보내면 된다. 따라서 노드가 필요로 하는 최소의 정보량을 H_{\min} 이라 표시하고, 이를 비트 단위로 표시하면, $H_{\min} = \lceil \log_2 p \rceil$ 비트가 된다. 즉, H_{\min} 비트만큼의 정보가 있으면 p 개의 포트중 하나를 지정할 수 있다. 반얀(banyan)망을 예로 들어보면, 반얀망의 스위칭 소자는 두 개의 입/출력 포트를 가지며, $\lceil \log_2 2 \rceil = 1$ 비트만을 가지고 패킷을 다음 단(stage)으로 라우팅할 수 있다. 셔플넷의 노드는 반얀망의 스위칭 소자와는 달리, 송신기와 수신기가 추가되어 있으므로 수신기에 대응되는 포트가 필요하다. 또한 패킷이 거쳐가는 중간 노드들 중 일부 노드에서는 패킷이 p 개의 포트중 어느 것을 선택해도 무방한 경우가 생긴다. 이러한 노드들을 don't-care노드라 하자. 이에 대해서는 후에 자세히 논의될 것이다. 이처럼, 포트를 선택하는 것 외에 추가적인 고려사항이 생기므로, 노드는 p 개의 포트중 하나를 선택해서 패킷을 보낼 것인지, 임의의 한 포트를 택하여 패킷을 보낼 것인지(don't-care노드), 아니면 패킷을 수신할 것인지를 결정해야 한다. 따라서, 셔플넷에서 노드가 필요로 하는 정보의 최소량 $H_{Shufflenet}$ 은 아래와 같이 주어진다.

$$H_{Shufflenet} = \lceil \log_2(p+2) \rceil \text{ 비트}$$

2. 비트 단위 독자경로 라우팅과 디플렉션 라우팅의 병행

비트 단위 독자경로 라우팅과 디플렉션 라우팅을 병행하려면 패킷 헤더에 있는 라우팅 태그(routing tag)가 송신자와 무관해야 한다. 노드가 패킷 헤더의 일부분만을 읽기 때문에, 패킷이 우회되더라도 패킷 헤더의 내용이 변하지 않도록 하기 위해서는 라우팅 태그는 수신자의 주소에 의해 결정되어야 한다. 그리고, 패킷 헤더에 있는 정보 중에서 노드가 읽어야 할 정보를 지정하기 위해 일종의 포인터(pointer)가 필요하다. 노드는 포인터가 지정하는 $H_{Shufflenet}$ 비트만을 패킷 헤더에서 읽어 내어 패킷을 라우팅한다. 이를 구현하기 위해서는 노드들의 연결 형상에 제약이 필요하게 된다. 임의의 한 노드를 목적지(수신자)로 하고 여기서부터 패킷 진행방향의 반대방향으로 자라는 트리를 만들고, 이를 역방향 스패닝 트리(spanning tree)라 하자. 이 스패닝 트리에서 가지(branch)와 깊이(depth)를 아래와 같이 정의한다.

정의 1 :스패닝 트리에 속해있는 노드에 연결되어 있는 링크를 가지(branch)라 한다.

정의 2 :역방향 트리의 루트(root), 즉 수신자로부터의 홉 거리(hop distance)를 깊이(depth)라 한다.

우선 임의의 송신자수신자에 대해서 패킷 송수신이 가능해야 하므로, 역방향 스패닝 트리는 아래와 같은 조건을 만족시켜야 한다.

조건 1 :임의의 노드를 수신자로 정했을 때, 네트워크의 모든 노드를 포함할 수 있는 역방향 스패닝 트리가 존재해야 한다.

노드의 출력 포트에 연결된 링크들을 각각 0, 1, 2, ..., (p-1)까지 번호를 붙이고 이 번호를 링크 인덱스(link index)라 하자. 송신자와 무관한 라우팅 태그를 얻기 위해서는 다음과 같은 조건이 추가적으로 만족되어야 한다.

조건 2 :깊이 d와 깊이 d+1 사이의 가지(branch)들이 모두 같은 링크 인덱스를 가지거나, 또는 깊이 d와 깊이 d+1 사이의 모든 링크들이 역방향 스패닝 트리의 가지가 되어야 한다.

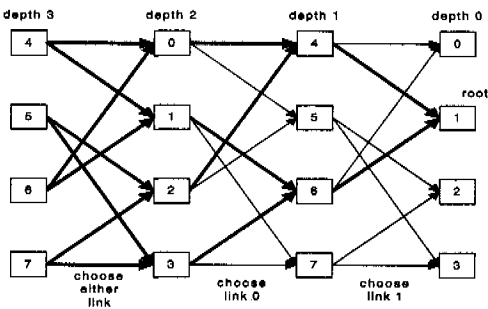


그림 1. 8-노드 서플넷의 역방향 스패닝 트리

만약 네트워크 위의 두 조건을 만족하면, 패킷이 우회되더라도 포인터의 위치만 변경함으로써 라우팅 태그의 내용을 그대로 유지하면서 목적지까지 도달할 수 있다. 서플넷이 이러한 조건을 만족하는 예이다. 그림 1에 8개의 노드($k=2, p^k=2^2=4$)로 이루어진 서플넷의 역방향 스패닝 트리가 도시되어 있다. 여기서 수신자는 노드 1이다. 굵은 선으로 표시된 링크들이 가지를 나타낸다. 그림에서 볼 수 있듯이, 깊이 0과 깊이 1 사이의 가지들은 모두 링크 인덱스가 1이다. 마찬가지로, 깊이 1과 깊이 2 사이에서는 가지들의 링크 인덱스가 0, 깊이 2와 깊이 3

사이에서는 모든 링크가 역방향 스패닝 트리의 가지이다. 이 역방향 스패닝 트리는 노드 0부터 노드 7까지 네트워크의 모든 노드를 포함한다. 따라서, 어느 노드에서 패킷이 출발하든지 노드 1로 도착할 수 있다. 그리고 깊이 2와 깊이 3 사이의 모든 링크가 역방향 스패닝 트리의 가지가 되므로, 깊이 3인 열(column)의 노드에서는 패킷이 두 개의 출력 포트중 어느 것을 택해도 무방하다. 이 경우, 깊이 3인 열에 위치한 노드는 don't-care 노드이다.

서플넷에서는 한 열에 속하는 노드의 수가 항상 p의 멱승(a power of p)이 되어야 한다 ($N=bp^k$). 그러나 다음의 정리는 서플넷이 가지는 이러한 제약이 반드시 독자 경로 라우팅과 디플렉션 라우팅을 위해 필수 불가결한 것은 아니라는 것을 시사해 준다.

정리 1 :서플넷 기반의 네트워크에서 비트 레벨 독자경로 라우팅과 디플렉션 라우팅이 가능하면, 한 열에 속하는 노드의 수는 p의 배수이다.

증명 :앞에서도 언급했듯이, 깊이 d와 깊이 d+1사이의 가지들의 링크 인덱스가 같거나 모든 링크가 가지가 되어야 한다. 이것은 한 노드의 입력 포트에 연결된 링크들의 인덱스가 모두 같아야 한다는 것을 의미한다. 한 열에 속한 노드의 수를 n이라 하면, 이웃한 두 개의 열 사이에는 같은 링크 인덱스를 가지는 링크가 각각 n개씩 있게 된다. 즉, 링크 인덱스가 0인 링크가 n개, 1인 링크가 n개, 이런 식으로 총 np개의 링크가 있다. 이중 링크 인덱스가 q($0 \leq q \leq p-1$)인 n개의 링크를 고려하자. 하나의 노드가 p개의 입력 포트를 가지고 있고, 여기에 연결된 링크의 인덱스가 모두 같아야 하므로, 한 열에 있는 n개의 노드중 입력 포트에 연결된 링크의 인덱스가 q인 노드는 n/p개이며, 따라서 n/p는 정수가 되어야 한다. 즉, n은 p의 배수가 되어야 한다. Q. E. D.

위의 정리는 서플넷 기반 네트워크에서 비트 레벨 독자경로 라우팅과 디플렉션 라우팅을 병행하기 위한 필요 조건을 제시하고 있는데 이는 서플넷보다 구현 가능한 노드수의 제약이 적은 유연한 네트워크 구조가 존재할 가능성이 있다는 것을 보여준다.

III. Asymmetric Cylinder(AC)

Asymmetric Cylinder(AC)는 입/출력 포트의 수

가 각각 2개인 노드로 구성되며, 구현 가능한 노드의 수 N 은 $N=km$ 으로 주어진다. 여기서 k 는 열의 수이고, n 은 하나의 열에 속하는 노드의 수이다. AC에서 n 은 $n=(2^x-1) \cdot 2^{y+1}$ 로 주어지며, x 는 자연수이고 y 는 0이상의 정수이다. 만약 $x=1$ 이면, n 의 값은 2의 멱이 되어, 셔플넷의 변형인 Shuffle ring 또는 GSN과 같은 구조가 된다. 따라서 Shuffle ring이나 GSN은 AC의 특별한 경우라고 볼 수 있다.

열의 수가 k 개이고, 각 열에 n 개씩의 노드가 이루는 AC를 (n, k) AC로 표시한다. AC의 열은 왼쪽으로부터 0, 1, 2, ..., $(k-1)$ 로 번호를 붙이고, 이를 열 주소(column address)로 정한다. 그리고 행은 위에서 아래로 0, 1, 2, ..., $(n-1)$ 로 번호를 붙이고, 이를 행 주소(row address)라 한다.

1. Interconnection

주소가 i 인 열에 속한 노드의 출력 포트는 주소가 $(i+1) \bmod k$ 인 열에 속한 노드의 입력 포트에 연결이 되어, 네트워크의 연결형태는 실린더(cylinder)와 같은 모습이 된다. 서로 이웃한 두 열사이의 연결 규칙을 제시하게 전에 편의상 아래와 같이 두 개의 변수를 정의한다.

$$m = n/2$$

$$a = 2^{\lceil \log_2 m \rceil} = 2^{x+y}$$

열 주소가 i 이고 행 주소가 j 인 노드의 출력 포트는 열 주소가 $(i+1) \bmod k$ 이고 행 주소가 각각 $j, j+1$ 인 두 노드에 연결되며, j 의 값은 아래와 같다.

$$j = \begin{cases} (j \bmod a) \cdot 2 & (j \bmod a) < m, \text{ 즉, } j) m \text{ 이거나 } j \geq a \text{ 일때} \\ (j+m-a) \cdot 2 & \text{그 외의 경우.} \end{cases}$$

$(j \bmod a) < m$ 인 경우, 연결 규칙은 한 열당 $2a$ 개의 노드를 가지는 셔플넷의 그것과 동일하다. 따라서, AC에서 n 이 2의 멱인 경우에 임의의 j 에 대하여 $(j \bmod a) < m$ 이므로, 이때는 AC의 연결 형상은 GSN이나 Shuffle ring과 동일하게 된다.

그림 2에 36개의 노드로 이루어진 AC의 연결 형상이 제시되어 있다. 여기서 $k=3$ 이고, $n=(2^x-1) \cdot 2^{y+1} \mid_{x=2, y=1} = 12$ 이다. 열 주소가 2인 노드들의 출력 포트는 열 주소가 0인 노드들의 입력 포트에 연결되어, 실린더와 같은 형태를 이룬다.

다.

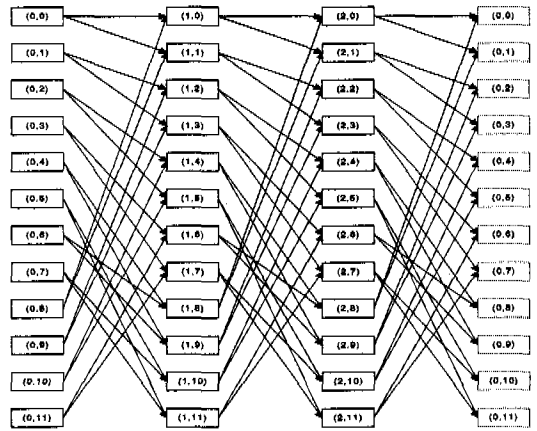


그림 2. (12,3) Asymmetric Cylinder

2. 스패닝 트리(Spanning Tree)

역방향 스패닝 트리의 깊이가 증가할수록 트리에 속하는 노드의 수는 증가한다. 그림 1의 예에서도 볼 수 있듯이, 깊이 1에서는 두 개의 노드가 트리에 속하고, 깊이 2에서는 네 개의 노드가 트리에 속한다. 행 주소가 $\xi (0 \leq \xi \leq n-1)$ 인 노드로부터 역방향 스패닝 트리를 만든다고 가정하자. 열(column) 내의 모든 노드가 역방향 스패닝 트리에 포함되는 깊이의 최소값을 $k(\xi)$ 라 하면, n 이 2의 멱일 경우, 이웃하는 두 열간의 연결은 셔플 연결(shuffle connection)이 되고, $k(\xi)$ 는 다음과 같이 ξ 와 무관하게 주어진다.

$$k(\xi) = \log_2 n$$

n 이 2의 멱이 아닌 경우에는 $k(\xi)$ 는 ξ 의 값에 따라 변하게 된다. 우선 $x \geq 2$ 이고 $y=0$ 인 경우, 즉 $n=(2^x-1) \cdot 2^{y+1} \mid_{y=0} = 2^{x+1}-2$ 인 경우, 다음과 같은 정리가 성립한다.

정리 2 : $n=2^{x+1}-2(x \geq 2)$ 인 AC에서 행 주소가 ξ 인 노드로부터 역방향 스패닝 트리를 만들면, $\xi < a$ 인 경우 $k(\xi)$ 는 다음과 같다.

$$k(\xi) = \log_2 2a = \lceil \log_2 n \rceil$$

증명 : 행 주소가 ξ 인 노드에서 만든 역방향 스패닝 트리의 깊이 d 에 있는 노드중에 이 스패닝 트리에 속하는 노드의 수를 $\alpha(\xi, d)$ 라 하자. 열 사

이의 연결 규칙이 서플넷 연결에 기반을 두고 있기 때문에, 만약 $\alpha(\xi, d) \geq n/2$ 이면, 깊이 $d+1$ 의 모든 노드가 역방향 스페닝 트리에 속하게 된다. 그렇기 때문에, $\alpha(\xi, k(\xi)-1) \geq n/2$ 이면, 깊이 $k(\xi)$ 의 모든 노드가 스페닝 트리에 속한다. AC에서 일부 노드는 서플넷 연결 규칙이 아닌 다른 연결 규칙으로 연결되기도 하나, 여기서는 서플넷 연결 규칙으로 연결된 것만을 고려해도 증명에 무리가 없으므로, 서플넷 연결만을 고려한다. 우선 앞에서 이미 언급한 서플넷 연결 규칙을 역방향으로 적용한다. 즉, 열 주소가 $(i+1) \bmod k$ 이고 행 주소가 ξ 인 노드의 입력 포트는 열 주소가 i 인 두 개의 노드의 출력 포트와 인덱스가 같은 링크로 연결되어 있고, 이 두 노드의 행 주소는 각각 $\lfloor \xi/2 \rfloor$ 와 $\lfloor \xi/2 \rfloor + a$ 이다. 이 관계로부터, 깊이 d 에서 트리에 속할 수 있는 가능한 행 주소의 집합 $\{\xi_d\}$ 를 구하면 다음과 같다.

$$\{\xi_d\} = \{ \lfloor \xi/2^d \rfloor + ba/2^{d-1}, b=0, 1, \dots, 2^{d-1} \}$$

그리고 $\xi < a$ 이므로, $\lfloor \xi/2^d \rfloor$ 의 범위는 다음과 같다.

$$0 \leq \lfloor \xi/2^d \rfloor \leq a/2^{d-1} - 1$$

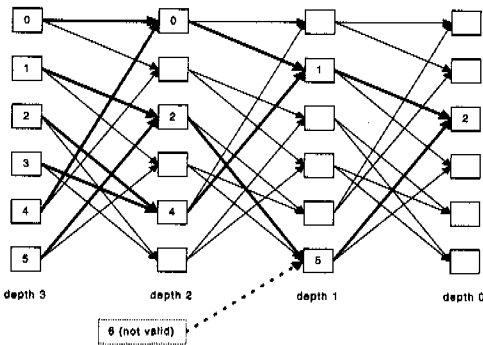


그림 3. n=6인 AC에서의 역방향 스페닝 트리

우선 $x=2$ 인 경우를 생각해 보자. 이 경우 $n=6$ 이고 $a=4$ 이다. 그림 3에 역방향 스페닝 트리의 예가 도시되어 있다. 깊이 1에서, 스페닝 트리에 속하는 노드의 행 주소는 $\lfloor \xi/2 \rfloor$ 와 $\lfloor \xi/2 \rfloor + a$ 이다. $\xi < a$ 이므로, $\lfloor \xi/2 \rfloor + a$ 는 n 보다 작고, 따라서 위의 두 행 주소는 모두 유효하다. 깊이 2에서 스페닝 트리에 속할 수 있는 행 주소는 $\lfloor \xi/4 \rfloor$, $\lfloor \xi/4 \rfloor + a/2$, $\lfloor \xi/4 \rfloor + a$, $\lfloor \xi/4 \rfloor + 3a/2$ 이다. 여기서 $\lfloor \xi/4 \rfloor + 3a/2$ 는

$\lfloor \xi/4 \rfloor + 3a/2 = \lfloor \xi/4 \rfloor + 3 \cdot 4/2 = \lfloor \xi/4 \rfloor + 6 < n$ 이므로, 유효한 주소가 아니다(그림 3). 따라서, 깊이 2에서 유효한 행 주소는 $\lfloor \xi/4 \rfloor$, $\lfloor \xi/4 \rfloor + a/2$, $\lfloor \xi/4 \rfloor + a$ 가 된다. $\alpha(\xi, 2) = 3 \geq n/2$ 이므로, 깊이 3의 모든 노드는 스페닝 트리에 속하게 되고, $k(\xi)$ 의 값은 다음과 같이 주어진다.

$$k(\xi) \mid_{x=2, y=0} = 3 = \lceil \log_2 6 \rceil$$

$n > 6$ 또는 $a > 4$ 인 경우, $\{\xi_a\}$ 의 최대값을 생각해 보면, $\xi = a-1$ 로부터 만든 역방향 스페닝 트리에서 최대값이 나타나며, 그 값은 다음과 같다.

$$\begin{aligned} \max(\xi_d) &= \lfloor (a-1)/2^d \rfloor + (2^d - 1)a/2^{d-1} \\ &= a/2^d - 1 + (2^d - 1)a/2^{d-1} \end{aligned}$$

a 의 정의($a=2^{x+y}$)와

$$n = (2^x - 1) \cdot 2^{y+1} \mid_{y=0} = 2^{x+1} - 2$$

로부터 $n = 2a - 2$ 라는 관계식을 얻을 수 있다. 이 식을 적용하여 n 에서 $\max(\xi_d)$ 를 빼면, 다음과 같은 결과를 얻는다.

$$\begin{aligned} \Delta &= n - \max(\xi_d) = 2a - 2 - (a/2^d - 1 + (2^d - 1)a/2^{d-1}) \\ &= a/2^{d-1} - a/2^d - 1 \\ &= a/2^d - 1 \end{aligned}$$

위의 결과로부터, Δ 가 0보다 크기 위한 조건을 구하면, $0 \leq d \leq \log_2 2a - 2$ 가 된다. 이 범위내의 깊이에서는 $\{\xi_d\}$ 의 원소는 항상 그 값이 n 보다 작으므로, 모두 유효한 원소이다. 그러므로, 깊이 $(\log_2 2a - 1)$ 에서는, $\{\xi_d\}$ 의 원소들 중 그 값이 n 보다 큰 것들을 제외한 모든 원소가 다 유효하다. 이 깊이에서 값이 n 보다 큰 $\{\xi_d\}$ 의 원소는 $\max(\xi_d)$ 뿐이다. 두 번째로 큰 값을 가지는 원소는 항상 n 보다 작다. 다음의 식이 이를 보여준다.

$$\begin{aligned} n - \lfloor (a-1)/2^d \rfloor + (2^d - 2)a/2^{d-1} \\ = 2a - 2 - (a/2^d - 1 + (2^d - 2)a/2^{d-1}) \\ = 3a/2^d - 1 > 0 \end{aligned}$$

깊이 $(\log_2 2a - 1)$ 에서는 $\max(\xi_d)$ 를 제외한 $\{\xi_a\}$ 의 모든 원소가 유효하므로, 이 깊이에서 스페닝 트리에 속하는 노드의 수 $\alpha(\xi, \log_2 2a - 1)$ 는 다음과 같다.

$$c(\xi, \log_2 2a - 1) = 2^d - 1 = 2^{\log_2 2a - 1} - 1 = a - 1 = n/2$$

따라서, 깊이 $(\log_2 2a - 1) + 1 = \log_2 2a = \lceil \log_2 n \rceil$ 에 있는 모든 노드는 스페닝 트리에 속하게 된다.

Q. E. D

$a \leq \xi < n$ 일 때는 다음의 정리가 성립한다.

정리 3 : $n = 2^{x+1} - 2$ ($x \geq 2, y = 0$) 인 AC에서 행 주소가 ξ 인 노드로부터 만든 역방향 스페닝 트리를 가정하면, $a \leq \xi < n$ 인 경우에 대하여 $k(\xi)$ 의 최대값 $\max(k(\xi))$ 는 다음과 같다.

$$\max(k(\xi)) = \lceil \log_2 n \rceil + 1$$

증명 : 깊이가 1에서 스페닝 트리에 속하는 것이 가능한 열 주소는 $\lfloor \xi/2 \rfloor$ 와 $\lfloor \xi/2 \rfloor + a$ 이다. 여기서 $\lfloor \xi/2 \rfloor$ 는 n 보다 항상 작으므로 항상 유효하고, $\lfloor \xi/2 \rfloor + a$ 는 n 과 같거나 클 수도 있으므로, 항상 유효한 값은 아니다. 정리 2에 의하여, 깊이 $\lceil \log_2 n \rceil + 1$ 의 모든 노드는 $\lfloor \xi/2 \rfloor$ 에 의해 만들어진 트리에 의하여 모두 트리에 속하게 되며, 이것은 $\lfloor \xi/2 \rfloor + a$ 가 유효한지의 여부에 무관하다. 따라서, $\max(k(\xi)) = \lceil \log_2 n \rceil + 1$ 이다. Q. E. D.

$y = 1$, 즉 $n = (2^x - 1) \cdot 2^2$ 인 AC의 경우, 두 개의 이웃하는 노드를 한 그룹으로 묶으면, 이웃한 열 사이의 연결 형태는 같은 x 에 $y = 0$ ($n = (2^x - 1) \cdot 2$)인 AC의 연결 형태와 같게 된다. 그림 4에 $n(x, y) = (2^x - 1) \cdot 2^{y+1}$ | $x=2, y=1$ | $n = 12$ 의 값이 x 와 y 의 함수이므로, n 대신 $n(x, y)$ 로 표시하였다.)인 경우의 예가 제시되어 있다. 그림 4에서도 나타나 있듯이, 노드를 두 개씩 묶으면 연결 형태가 $n = 6$ ($x = 2, y = 0$)인 AC와 동일하다. 이러한 성질로부터, 깊이가 a 가

$$\lceil \log_2 (2^x - 1) \cdot 2 \rceil = \lceil \log_2 n(x, 0) \rceil \text{ 이하}$$

($d \leq \lceil \log_2 n(x, 0) \rceil$)일 경우에는 $c(\xi, d) |_{y=1}$ 의 값과 $c(\xi, d) |_{y=0}$ 의 값이 같음을 알 수 있다. 일반적으로, $i \geq 1$ 이고, $d \leq \lceil \log_2 n(x, i-1) \rceil$ 일 때, 아래와 같은 관계가 성립한다.

$$c(\xi, d) |_{y=i} = c(\xi, d) |_{y=i-1}$$

위의 관계를 이용하면, 정리 2를 $y \geq 1$ 인 경우가 지 일반화시킬 수 있다.

정리 4 : $n = (2^x - 1) \cdot 2^{y+1} = n(x, y)$ 인 AC에서

행 주소가 ξ 인 노드로부터 만든 역방향 스페닝 트리를 가정하면, $\xi < a$ 인 경우에 $k(\xi)$ 의 값이 다음과 같이 주어진다.

$$k(\xi) = \log_2 2a = \lceil \log_2 n(x, y) \rceil$$

증명 : 이 정리는 수학적 귀납법으로 증명할 수 있다. 정리 2의 증명과 마찬가지로, 여기서도 $c(\xi, d)$ 의 값이 $n(x, y)/2$ 이상임을 증명하면, 깊이 $d+1$ 의 모든 노드가 스페닝 트리에 속함을 보일 수 있다.

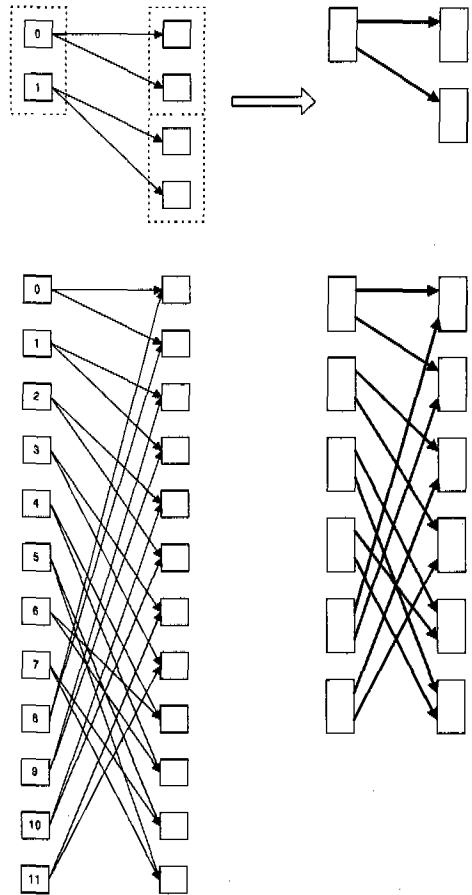


그림 4. n=12인 AC의 연결 모양

우선, $y = 1$ 인 경우를 고려해 보자. 이 때, $c(\xi, \log_2 2a - 1)$ 은 다음과 같은 관계를 만족한다.

$$\begin{aligned} c(\xi, \log_2 2a - 1) &= c(\xi, \lceil \log_2 n(x, 1) \rceil - 1) \\ &= c(\xi, \lceil \log_2 n(x, 0) \rceil) \end{aligned}$$

정리 2에 의하여, $c(\xi, \lceil \log_2 n(x, 0) \rceil) = n(x, 0)$ 이므로, 다음의 식이 성립한다.

$$c(\xi, \lceil \log_2 n(x, 1) \rceil - 1) = n(x, 0)$$

$n(x, y)$ 의 정의에 의하여, $n(x, y) = n(x, y+1)/2$ 이므로, 위의 식에 $n(x, 0)$ 대신 $n(x, y+1)/2$ 를 대입하면 다음과 같은 결과를 얻는다.

$$c(\xi, \lceil \log_2 n(x, 1) \rceil - 1) = n(x, 0) = n(x, 1)/2 \geq n(x, 1)/2$$

그러므로, $k(\xi)$ 는 다음과 같다.

$$k(\xi) = \lceil \log_2 n(x, 1) \rceil - 1 + 1 = \lceil \log_2 n(x, 1) \rceil$$

이제, $c(\xi, \lceil \log_2 n(x, i) \rceil) \geq n(x, i)/2$ 이라 가정하면, $c(\xi, \lceil \log_2 n(x, i) \rceil) = n(x, i)$ 가 되고,

$c(\xi, \lceil \log_2 n(x, i+1) \rceil - 1)$ 은 다음과 같이 주어진다.

$$\begin{aligned} c(\xi, \lceil \log_2 n(x, i+1) \rceil - 1) &= c(\xi, \lceil \log_2 n(x, i) \rceil) \\ &= n(x, i) = n(x, i+1)/2 \end{aligned}$$

따라서, $c(\xi, \lceil \log_2 n(x, i+1) \rceil - 1) \geq n(x, i+1)/2$ 이다. Q. E. D.

정리 4로부터, 정리 3도 $y \geq 1$ 인 경우로 확장시킬 수 있다.

정리 5 : $n = (2^x - 1) \cdot 2^{y+1} = n(x, y)$ 이고, $x \geq 2$ 인 AC에서 행 주소가 ξ 인 노드로부터 만든 역방향 스페닝 트리를 가정하면, $a \leq \xi < n$ 인 경우에 $k(\xi)$ 의 최대값은 다음과 같다.

$$\max(k(\xi)) = \lceil \log_2 n(x, y) \rceil + 1$$

증명 : 이 정리는 정리 4를 이용하여 정리 3의 증명과 같은 방법으로 증명할 수 있다. Q. E. D.

위에서 제시한 정리들로부터, 깊이가 $\lceil \log_2 n \rceil - 1$ 이하이면 ($d \leq \lceil \log_2 n \rceil - 1$), 깊이 d 와 깊이 $d+1$ 사이의 가지들의 링크 인덱스가 같고, 깊이가 $\lceil \log_2 n \rceil - 1$ 보다 크면

($d > \lceil \log_2 n \rceil - 1$), 깊이 d 와 $d+1$ 사이의 모든 링크가 역방향 스페닝 트리의 가지가 된다. 따라서 비트 단위 독자경로 라우팅과 다플렉션 라우팅을 구현하기 위한 조건을 만족한다.

3. 라우팅 태그(Routing Tag)와 라우팅 알고리즘(Routing Algorithm)

앞에서도 언급했듯이, 노드는 패킷을 라우팅하기

위해 $H_{shuffle} = \lceil \log_2(p+2) \rceil$ 비트를 읽어야 한다. AC에서는 $p=2$ 이므로, $\lceil \log_2(2+2) \rceil = 2$ 비트가 하나의 정보 블록(block)이 된다. 이 블록의 내용은 다음과 같이 정의할 수 있다.

- [00] - 패킷을 링크 인덱스 0에 해당하는 출력 포트에 보낸다.
- [01] - 패킷을 링크 인덱스 1에 해당하는 출력 포트에 보낸다.
- [10] - 패킷을 두 출력 포트 어디로 보내도 무방하다. (don't-care 노트)
- [11] - 패킷을 수신한다.

라우팅 태그는 위와 같은 블록들로 이루어지며, 각 블록은 목적지로부터의 역방향 스페닝 트리의 깊이에 대응한다. 패킷 헤더는 이러한 블록들로 이루어진 라우팅 태그와, 포인터로 이루어지며, 노드는 포인터가 지정하는 하나의 블록을 읽고 패킷을 라우팅한다. 라우팅 태그의 내용은 변하지 않으며, 노드는 포인터의 위치만을 변화시킨다. AC의 역방향 스페닝 트리에서, 편의상 k 를 다음과 같이 정의한다.

$$k = \max(k(\xi))$$

앞에서 제시한 정리들에 의하여, k 의 값은 다음과 같이 주어진다.

$$k = \lceil \log_2 n(x, y) \rceil + 1$$

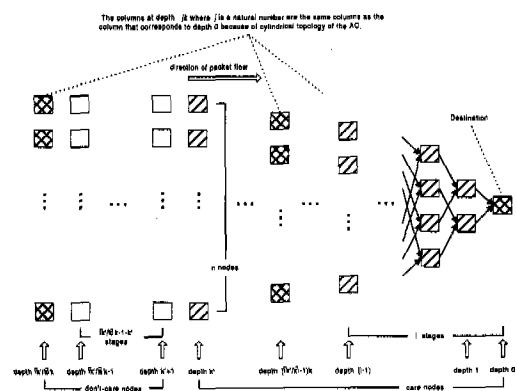


그림 5. AC의 역방향 스페닝 트리(일반적인 경우)

깊이 k 에서는 모든 노드가 스페닝 트리에 속한다. 그림 5에 역방향 스페닝 트리의 모습이 제시되어 있다. 이 스페닝 트리의 루트(root)가 목적지인 패킷

이 깊이 k 에 있는 임의의 노드에 있으면, 우회되지 않을 경우 스페닝 트리의 가지들을 따라서 목적지까지 라우팅 될 수 있다. 만약 패킷이 깊이가 k 보다 큰 노드에 있으면, 이 패킷은 깊이 k 에 있는 한 노드에 도착하기 위해 어느 출력 포트를 택해도 무방하다.

라우팅 태그 T 는 다음과 같은 모양을 가지게 된다.

$$T = [10][10]\dots[10] \\ [[00]과 [01]로 이루어진 k 개의 블록] [11]$$

T 의 각 블록은 스페닝 트리의 깊이에 대응된다. 맨 오른쪽의 [11]은 깊이 0에 대응되고, 오른쪽에서 두 번째 블록은 깊이 1에 대응된다.

패킷의 진행방향을 왼쪽에서 오른쪽으로 가정하면, 패킷의 진행방향과 블록의 배열을 일치시키기 위해 T^R 을 정의한다. 이는 T 의 블록들을 역순으로 배열한 것이다.

$$T^R = [11] \\ [[00]과 [01]로 이루어진 k 개의 블록] [10][10]\dots[10]$$

T^R 의 중간 부분에 있는 k 개의 블록은 패킷 목적지의 행 주소에 의해 결정되는데, 만약 행 주소가 $(\delta_{k-1}\delta_{k-2}\dots\delta_0)_2$ 이면, 이 블록들은 T 에서 다음과 같다.

$$[0\delta_{k-1}][0\delta_{k-2}]\dots[0\delta_0]$$

예를 들어 $k=4$ 이고 목적지 노드의 행 주소가 $7=(0111)_2$ 이면, 라우팅 태그 T 의 중간 부분의 블록들은 다음과 같은 모양이 된다.

$$[00][01][01][01]$$

T^R 에서는 이들이 역순으로 배열되므로 [01][01][01][00]이 된다.

포인터의 초기 위치는 송신자와 수신자 사이의 최소 홉 거리(hop distance)에 해당하는 블록이 된다. 만약 패킷이 우회되지 않으면, 한 노드를 지날 때마다 포인터가 뒤로(왼쪽으로) 한 블록씩 이동한다. 패킷이 목적지에 도착하면, 포인터는 맨 왼쪽의 [11]을 가리키게 되고, 목적지 노드는 이를 읽고 패킷을 수신하게 된다. 만약 깊이 i ($0 \leq i \leq k$)에서 패킷이 우회되면, 패킷은 깊이 k 에 있는 임의의 한 노드에 도착할 때까지 여러 노드를 지나가게 되

며, 이때는 어느 출력 포트를 택해도 무방하다 (don't-care 노드). 그림 5에서도 볼 수 있듯이, 깊이 k 에 있는 노드에 도착하기 위해 패킷은 $\lceil k/k \rceil k-1-k+i$ 개의 열을 지나가야 한다. 패킷이 우회되면, 포인터는 T^R 에서 앞으로(오른쪽으로) 이동한다. 몇 블록만큼을 이동할 것인지는 다음의 정리에 의해 정해진다.

정리 6 : AC에서 패킷이 우회되면, 포인터는 T^R 에서 $\lceil k/k \rceil k-1$ 블록만큼 앞으로 이동한다.

증명 : 포인터가 γ 개의 블록만큼 앞으로 이동한다고 가정하자. 그러면 패킷이 지나가는 don't-care 노드의 수 D 는 다음과 같다.

$$D = \gamma - (k - i)$$

깊이 i 에서 패킷이 우회되면, 패킷이 깊이 k 에 이르기 위해 $\lceil k/k \rceil k-1-k+i$ 개의 don't-care 노드를 지나가야 하므로, 다음의 등식이 성립해야 한다.

$$\lceil k/k \rceil k-1-k+i = \gamma - (k - i)$$

위의 식으로부터, γ 는 다음과 같이 주어지며, i 에 무관하다.

$$\gamma = \lceil k/k \rceil k-1$$

따라서, 패킷이 우회되면, 포인터는 $\lceil k/k \rceil k-1$ 블록만큼 이동한다. Q. E. D.

패킷이 우회될 경우, 포인터가 $\lceil k/k \rceil k-1$ 블록만큼 움직이므로, 라우팅 태그의 [10]블록의 수는 $\lceil k/k \rceil k-1$ 개만 있으면 충분하다.

그림 6은 $m=12$, $k=3$ 인 AC에서의 패킷 라우팅을 보여준다. 여기서 k 의 값은

$$k = \lceil \log_2 12 \rceil + 1 = 5$$

이며, 패킷이 우회될 경우, 포인터는 $\lceil k/k \rceil k-1 = \lceil 4/3 \rceil \cdot 3-1 = 5$ 블록만큼 이동한다. 노드의 열 주소와 행 주소를 (A_c, A_r) 로 표시하면, 그림 6에서 송신자의 주소는 (0,6)이고, 목적지는 (2,3)이다. 목적지의 열 주소가 $3=(00011)_2$ 이므로, T 와 T^R 은 다음과 같다.

$$T = [10][10][10][10][10] [00][00][00][01][01] [11] \\ T^R = [11] [01][01][00][00][00] [10][10][10][10][10]$$

그림 6의 예에서는 패킷이 노드 (1,9)에서 우회된

것으로 가정하였다.

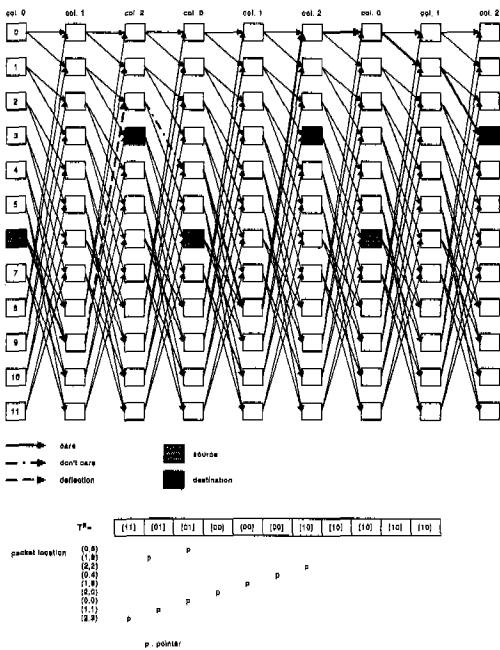


그림 6. (12,3) AC에서 라우팅의 예

IV. 성능 평가(Performance Evaluation)

1. 홉 수의 기대값(The Expected Number of Hops)

송신자로부터 h 홉만큼 떨어진 곳에 위치한 노드의 수를 $\eta(h)$ 라 하자. 네트워크 내에서 송신자와 수신자 사이의 최대 홉 거리를 h_{max} 라 하면, 홉 수의 기대값 \bar{h} 는 다음과 같이 정의된다.

$$\bar{h} = \frac{\sum_{h=1}^{h_{max}} h \cdot \eta(h)}{N-1}$$

AC의 경우, $x=1$ 이면, n 은 2의 멱이 되어 망 구조가 GSN과 동일하게 된다. GSN에서의 \bar{h} 는 이미 [6]에 제시되어 있으므로, 여기서는 $x \geq 2$ 인 경우, 즉 n 이 2의 멱이 아닌 경우만을 고려하도록 한다. $x \geq 2$ 인 AC에서는 $\eta(h)$ 가 송신자에 따라 변하게 된다. 따라서 \bar{h} 를 하나의 값으로 얻기는 어렵다. 그러므로, 여기서는 \bar{h} 의 upper bound 와 lower bound를 제시하였다.

편의상, AC의 연결 형상을 두 가지로 분류하였는

데, 하나는 열의 수 k 가 k' 이상인 경우이고, 나머지는 k 가 k' 보다 작은 경우이다.

1.1 $k \geq k'$ 인 경우

표 1에 $k \geq k'$ 인 경우에 $\eta(h)$ 의 값이 주어져 있다. $\eta(h)$ 의 best case와 worst case를 각각 $\eta_b(h)$ 와 $\eta_w(h)$ 로 표기하였다.

표 1을 요약하면, $\eta_b(h)$ 와 $\eta_w(h)$ 는 다음과 같다.

$$\eta_b(h) = \begin{cases} 2^h & 1 \leq h < k' - 1 \\ n & k' - 1 \leq h < k \\ n - 2^{h-k} & k \leq h \leq k' + k - 2 \end{cases}$$

$$\eta_w(h) = \begin{cases} 2^h & 1 \leq h < y + 1 \\ 2^{h-1} & y + 2 \leq h < k' \\ n & k' \leq h < k \\ n - 2^{h-k} & k \leq h \leq k' + y + 1 \\ n - 2^{h-k-1} & k' + y + 2 \leq h \leq k' + k - 1 \end{cases}$$

표 1. $k \geq k'$ 일때의 $\eta(h)$

| h | $\eta_b(h)$ (best case) | $\eta_w(h)$ (worst case) |
|-------------------------------------------|----------------------------|----------------------------------|
| 1 | 2^1 | 2^1 |
| 2 | 2^2 | 2^2 |
| 3 | 2^3 | 2^3 |
| ... | ... | ... |
| $y+1$ | 2^{y+1} | 2^{y+1} |
| $y+2$ | 2^{y+2} | 2^{y+2} |
| $y+3$ | 2^{y+3} | 2^{y+2} |
| ... | ... | ... |
| $k-2$ | 2^{k-2} | 2^{k-3} |
| $k-1 = \lceil \log_2 n \rceil$ | n | $2^{\lceil \log_2 n \rceil - 1}$ |
| $k' = \lceil \log_2 n \rceil + 1$ | n | n |
| $k'+1$ | n | n |
| $k'+2$ | n | n |
| ... | ... | ... |
| $k-1$ | n | n |
| k | $n-1$ | $n-1$ |
| $k+1$ | $n-2^1$ | $n-2^1$ |
| $k+2$ | $n-2^2$ | $n-2^2$ |
| ... | ... | ... |
| $k+y+1$ | $n-2^{y+1}$ | $n-2^{y+1}$ |
| $k+y+2$ | $n-2^{y+2}$ | $n-2^{y+2}$ |
| $k+y+3$ | $n-2^{y+3}$ | $n-2^{y+2}$ |
| ... | ... | ... |
| $k+k'-2 = k + \lceil \log_2 n \rceil - 1$ | $n-2^{k-2}$ | $n-2^{k-3}$ |
| $k+k'-1$ | 0 | $n-2^{k-2}$ |

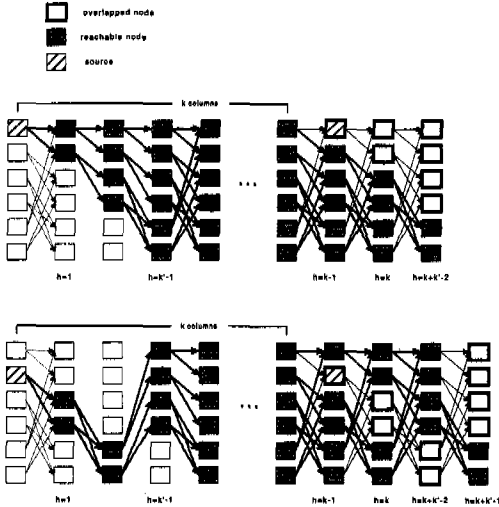


그림 7. $n=6, k \geq k'$ 일 때 패킷이 도달할수 있는 노드

그림 7에 $n=6$ 인 AC에서의 $\eta_b(h)$ 와 $\eta_w(h)$ 가 제시되어 있다. lower bound와 upper bound를 각각 $\bar{h} |_{k \geq k', L}$, $\bar{h} |_{k \geq k', U}$ 라 하면, 그 값은 다음과 같이 주어진다.

$$\bar{h} |_{k \geq k', L} = \frac{\sum_{h=1}^{k+k'-1} h \cdot \eta_b(h)}{kn-1}$$

$$= \frac{\sum_{h=1}^{k-2} h \cdot 2^h + \sum_{h=k-1}^{k+k'-1} hn + \sum_{h=k}^{k+k'-2} h(n-2^{k-h})}{kn-1}$$

$$\bar{h} |_{k \geq k', U} = \frac{\sum_{h=1}^{k+k'-1} h \cdot \eta_w(h)}{kn-1}$$

$$= \frac{\sum_{h=1}^{k-2} h \cdot 2^h + \sum_{h=k-1}^{k+k'-1} h \cdot 2^{h-1} + \sum_{h=k}^{k+k'-1} hn + \sum_{h=k}^{k+k'-1} h(n-2^{k-h}) + \sum_{h=k+1}^{k+k'-1} h(n-2^{k-h-1})}{kn-1}$$

만약 n 이 2의 멱이면, 네트워크 구조가 isotropic하게 되어 $\eta(h)$ 가 송신자에 무관한 하나의 값으로 구해지는데, 이때 $\eta(h)$ 는 $\eta_b(h)$ 와 같고, \bar{h} 는 $\bar{h} |_{k \geq k', L}$ 이 된다. 이 결과는 [6]에서 제시된 결과와 같은 값이다.

1.2 $k < k'$ 인 경우

이 경우는 $k < k'$ 인 영역에서 일부 노드가 중복(overlap)될 수 있으므로, $\eta(h)$ 를 정확하게 기술하는 것이 매우 어렵다. 여기서는 계산을 간소화하고 해석적인 bound를 얻기 위해 몇 가지 가정을 적용한다.

$\bar{h} |_{k < k'}$ 의 lower bound를 계산하기 위한 가정은 다음과 같다.

- (1) $k < k' - 1$ 에서 중복되는 노드가 없다.
- (2) $k < k' - 1$ 일 때, 송신자로부터 h 홉만큼 떨어진 노드의 수(중복된 노드 포함)는 같은 n 에서의 $\eta_b(h)$ 의 값과 같다. 여기서 중복되는 노드는 무시한다.

$\bar{h} |_{k < k'}$ 의 upper bound를 계산하기 위한 가정은 다음과 같다.

- (1) 송신자 노드는 $h=k$ 일 때 중복된다.
- (2) $h \leq k$ 이면, 송신자에서 h 홉만큼 떨어진 노드의 수(중복된 노드 포함)는 같은 n 에서의 $\eta_w(h)$ 와 같다.

위의 가정들로부터 다음과 같은 결과를 얻는다. $\bar{h} |_{k < k'}$ 의 lower bound를 위한 $\eta(h)$ 의 값은 다음과 같다.

$$\eta(h) |_{k < k', lower} = \begin{cases} 2^h & 1 \leq h < k-1 \\ n - \sum_{i=0}^{h/k-1} 2^{ik+(h \bmod k)} & k-1 \leq h \leq k'+k-2 \end{cases}$$

$\bar{h} |_{k < k'}$ 의 upper bound를 위한 $\eta(h)$ 는 다음과 같다.

$$\eta(h) |_{k < k', upper} = \begin{cases} \eta_w(h) & 1 \leq h < k \\ \eta_w(h) - \eta_w(h-k) & k \leq h < k' \\ n - \eta_w(h-k) & k' \leq h \leq k'+k-1 \end{cases}$$

그러므로, $\bar{h} |_{k < k'}$ 의 bound는 다음과 같이 주어진다.

$$\bar{h} |_{k < k', L} = \frac{\sum_{h=1}^{k-2} h \cdot 2^h + \sum_{h=k-1}^{k+k'-2} h(n - \sum_{i=0}^{h/k-1} 2^{ik+(h \bmod k)})}{kn-1}$$

$$\bar{h} |_{k < k', U} = \frac{\sum_{h=1}^{k-1} h \eta_w(h) + \sum_{h=k}^{k+k'-1} h(\eta_w(h) - \eta_w(h-k)) + \sum_{h=k+1}^{k+k'-1} h(n - \eta_w(h-k))}{kn-1}$$

그림 8에 AC의 여러 가지 구성에 따른 홉수의 기대값이 제시되어 있다. U(analytic)와 L(analytic)은 각각 해석적으로 얻어진 \bar{h} 의 upper bound와 lower bound를 나타내고, U(numerical)와 L(numerical)은 수치적으로 계산한 값들을 나타낸다. 그림 8에서 볼 수 있듯이, $k \geq k'$ 일 경우에는 ($N=60$ 일 때 $n=(4, 6, 12)$, $N=90$ 일 때 $n=(4, 6, 8, 12, 16, 24)$, $N=300$ 일 때

$n=(4, 6, 12, 30)$ 해석적으로 얻은 값과 수치적으로 얻은 값이 완전히 일치한다. 그 외의 경우, 즉 $k \neq k'$ 일 경우에, 해석적으로 얻은 값과 수치적으로 얻은 값은 매우 비슷하며, 위에서 제시한 가정이 실제와 근접함을 보여준다. 그림 8로부터 AC의 성능은 n 이 클수록 좋을 것이라고 생각하기 쉬우나, 다음에 제시될 컴퓨터 시뮬레이션 결과는 이러한 예상이 반드시 옳은 것은 아니라는 것을 보여준다.

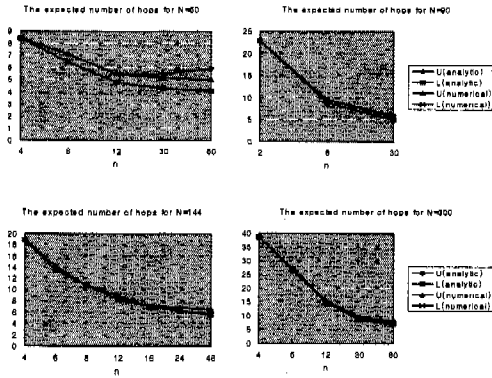


그림 8. $N=60, 90, 144, 300$ 일 때 홉수의 기대값의 범위

2. 컴퓨터 시뮬레이션(Computer Simulations)
 여기서는 AC의 여러 가지 구성에 대해, 패킷이 목적지에 도달하기 위해 필요한 평균 홉수(average number of hops)와 수율(throughput)을 제시한다. 각 노드는 하나의 송신기와 수신기를 가지고 있으며, 내부의 버퍼는 없다고 가정하였다. 따라서, 두 패킷이 하나의 포트에 나가고자 할 경우, 하나는 우회되어야 한다.

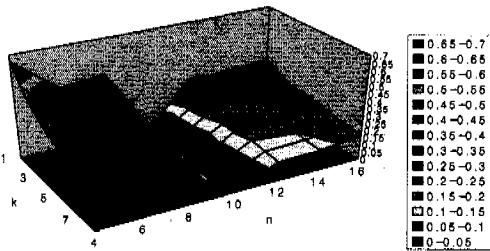


그림 9. n 과 k 의 값에 대한 수율의 변화

그림 9는 여러 가지의 n 과 k 에 대해서 부하(패킷이 네트워크로 들어올 확률, offered load)가 0.8

일 때의 수율을 도식한 것이다. $n=10$ 인 경우는 AC로 구성이 불가능하므로, 수율을 0으로 처리하였다. 같은 k 값에 대하여, n 이 2의 멱이면 수율이 상대적으로 높음을 볼 수 있다. 같은 n 에 대해서는 k 가 증가함에 따라 수율이 감소하는 경향을 보인다. 따라서, 주어진 $N=kn$ 에서, n 과 k 사이에 일종의 trade-off가 있다. $N=60$ 인 경우를 예로 들면, $n=30, k=2$ 로 구성하거나 $n=4, k=15$ 로 구성할 수 있다. 전자의 경우는 k 가 작으나 n 이 2의 멱이 아니고, 후자의 경우 n 은 2의 멱이지만 k 가 큰 값을 가진다.

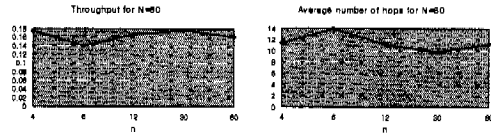


그림 10. $N=60$ 일 때 수율과 평균 홉수

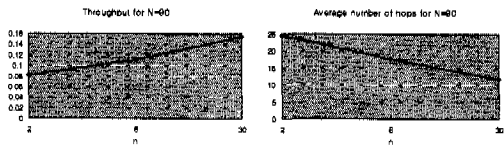


그림 11. $N=90$ 일 때 수율과 평균 홉수

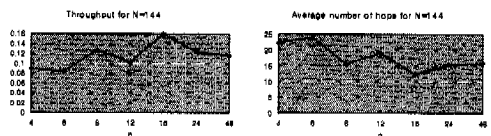


그림 12. $N=144$ 일 때 수율과 평균 홉수

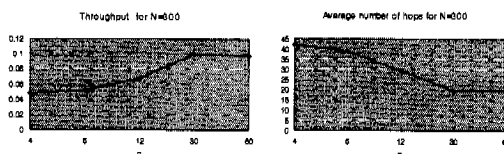


그림 13. $N=300$ 일 때 수율과 평균 홉수

그림 10은 부하가 0.8일 때 $N=60$ 인 AC의 수율과 평균 홉수를 보여준다. $n=30$ 일 때와 $n=4$ 일 때 피크(peak)가 생기는데, 이 경우 $n=30$ 인 경우가 더 성능이 우수함을 볼 수 있다.

그림 11, 12, 13은 각각 $N=90$, $N=144$, $N=300$ 인 AC의 수율과 평균 홉수를 보여준다. 부하는 0.8로 가정하였다. $n=90$ 인 경우, 기존의 셔플 연결만으로 네트워크를 구성하는 방법은 $n=2$, $k=45$ 를 택하는 방법뿐이다(GSN). 수율은 $n=30$ 일 때 가장 높다. $N=144$ 인 경우는 $n=16$ 일 때 최대의 수율을 보여주는데, 이는 $n=16$ 일 때의 k 의 값이 $n=24$ 나 $n=48$ 의 경우에 비해 그리 크지 않기 때문이다. $N=300$ 일 때는 $n=30$ 일대의 수율이 $n=60$ 일 때의 수율보다 조금 높으며, 최대치이다. 그림 10에서 13을 보면, 수율은 n 이 2의 멱이 되거나 2의 멱에 가까운 수일때($n|_{y=0} = 2^{x+1} - 2$) 성능이 상대적으로 우수한 것을 볼 수 있다.

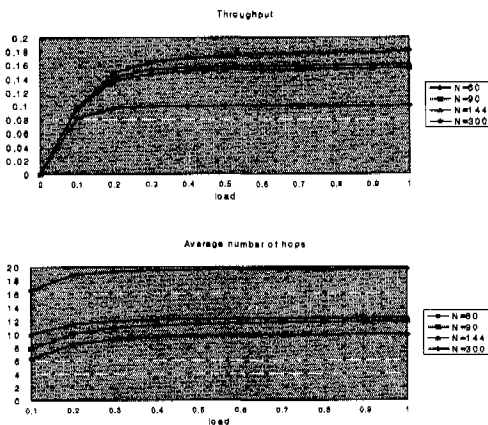


그림 14. 부하(offered load)의 변화에 따른 수율과 평균 홉수의 변화

그림 14는 부하의 변화에 따른 수율과 평균 홉수를 도시한 것이다. 각각의 N 에 대하여 최대의 수율을 얻을 수 있는 n 값을 가정하였다. 즉 $N=60, 90, 300$ 인 경우 $n=30$ 이고, $N=144$ 일 때는 $n=16$ 이다. 노드 수 N 이 증가함에 따라 얻을 수 있는 수율이 감소하고, 평균 홉수는 증가하는 것을 볼 수 있다.

비록 AC가 내부 버퍼가 없이 동작이 가능하지만, 버퍼를 추가하여, 버퍼에 빈자리가 있을 경우 패킷을 우회시키지 않고 버퍼에 저장했다가 보냄으로써 성능을 향상시킬 수 있다. 셔플넷에서 버퍼에 의한 성능향상은 [9]에 제시되어 있다. [9]에서 제시된 결과에 따르면, 한개 또는 두 개의 버퍼는 성능을 크게 향상시켜주지만, 그 이상으로 많이 사용하면 사용한 버퍼에 비해 성능 향상은 미미하다. AC도 기

본적으로 셔플넷과 유사한 구조이므로, 이 결과를 guideline으로 사용할 수 있으리라 추정된다.

V. 결론

본 논문에서는 구현 가능한 노드수의 제약이 상대적으로 적고, 비트 레벨 독자경로 라우팅과 디플렉션 라우팅을 병행할 수 있는 새로운 네트워크 구조를 제안하였다. 구현 가능한 노드수의 측면에서 볼 때, 제안된 네트워크, AC는 셔플넷, GSN, 또는 de Bruijn graph보다 유연하며, 이들은 모두 AC의 특별한 경우로 볼 수 있다. 가장 구현 가능한 노드수가 많은 네트워크 구조는 GEMNET이나, 이 네트워크 구조는 비트 레벨 독자경로 라우팅과 디플렉션 라우팅을 병행하기에는 부적합한 구조이다.

본 논문에 제시된 라우팅 알고리즘은 셔플넷, GSN, de Bruijn graph에도 그대로 적용이 가능하다. 셔플넷에서의 비트 레벨 독자경로 라우팅은 [10]에 제시되어 있다.

2장에서 제시된 비트 레벨 독자경로 라우팅과 디플렉션 라우팅을 병행하기 위한 필요조건을 고려해 볼 때, AC가 반드시 이러한 조건을 만족하는 가장 유연한(flexible) 구조의 네트워크라고 볼 수는 없다. 비트 레벨 독자경로 라우팅과 디플렉션 라우팅을 병행할수 있는 보다 유연한 네트워크 구조가 있을 가능성은 아직도 존재한다.

참고 문헌

- [1] A. S. Acampora and M. J. Karol, "An overview of lightwave packet networks," *IEEE Network Mag.*, pp. 29-40, Jan. 1989.
- [2] M. G. Hluchyj and M. J. Karol, "Shufflenet: An application of generalized perfect shuffles to multihop lightwave networks," in *Proc. IEEE INFOCOM*, pp. 4B.4.1-4B.4.5, 1988.
- [3] K. Sivarajan and R. Ramaswami, "Multihop lightwave networks based on de Bruijn graphs," in *Proc. IEEE INFOCOM*, pp. 1001-1011, 1991.
- [4] N. F. Maximchuk, "The Manhattan street network," in *Proc. IEEE GLOBECOM*, pp. 255-261, 1986.
- [5] A. Krishna and B. Hajek, "Performance of shuffle-like networks with deflection," in *Proc.*

IEEE INFOCOM, pp. 473-480, 1990.

- [6] S. Seo, P. R. Prucnal, and H. Kobayashi, "Generalized multihop shuffle networks," *IEEE Trans. Commun.*, Vol. 44, No. 9, pp. 1205-1211, Sep. 1996.
- [7] J. Iness, S. Banerjee, and B. Mukherjee, "GEMNET: A generalized shuffle-exchange-based, regular, scalable, modular, multihop, WDM lightwave network," *IEEE/ACM Trans. Networking*, Vol. 3, No. 4, pp. 470-476, Aug. 1995.
- [8] P. Baran, "On distributed networks," *IEEE Trans. Commun. Sys.*, pp. 1-9, Mar. 1964.
- [9] N. F. Maximchuk, "Comparison of deflection and store-and-forward techniques in the Manhattan street and shuffle-exchange networks," in *Proc. IEEE INFOCOM*, pp. 800-809, 1989.
- [10] S. Seo, B. Y. Yu, and P. R. Prucnal, "Bit-level packet-switching multihop shuffle networks with deflection routing," *Applied Optics*, Vol. 36, pp. 3142-3146, May 1997.

유 지 연(Ji Yon Youe)

정회원

1997년 2월 : 서울대학교 전기공학부 학사 졸업
 1997년 3월~현재 : 서울대학교 전기공학부 석사과정
 <주관심 분야> optical networks

서 승 우(Seung-Woo Seo)

정회원



1987년 : 서울대학교 전기공학과 (학사)
 1989년 : 서울대학교 대학원 전기공학과(석사)
 1993년 : 미국 펜실베니아 주립대학 전기공학과(박사)
 1993년 8월~1994년 7월 : 미국 펜실베니아 주립대학 전산기공학과 조교수

1994년 8월~1996년 7월 : 미국 프린스턴 대학 전기공학과 및 OPEM 연구소 연구원 (미국 과학 재단 선발 연구원)
 1996년 9월~현재 : 서울대학교 공과대학 전기공학부 조교수

<주관심 분야> 고속 통신망, 광통신망, 유/무선 교환 시스템