

論文2002-39TC-6-1

# Topology Aggregation 분석을 위한 PNNI 라우팅 시뮬레이터 구현

## (Implementation of the PNNI Routing Simulator for Analyze Topology Aggregation)

金 辯 坤 \* , 金 觀 雄 \* , 丁 光 日 \* , 申 鉉 順 \*\* , 鄭 炅 澤 \*\*\* ,  
田 炳 實 \*\*\*\*

(Byun-Gon Kim, Kwan-Woong Kim, Kwang-il Jeong, Hyun-soon Shin,  
Kyung-Taek Chung, and Byoung-Sil Chon)

### 요 약

본 논문은 기존의 Topology Aggregation 알고리즘에 따른 성능을 비교 평가하는데 초점을 두었다. 이를 위해 PNNI 라우팅 시뮬레이터를 설계 및 구현하여 TA 알고리즘의 성능을 비교 분석하였다. ATM Forum에서는 계층으로 구성된 네트워크에서 계층적 라우팅 프로토콜과 토폴로지 정보를 요약하기 위한 PNNI 1.0 규격을 권고하였다. 토폴로지 정보를 요약하는 일련의 과정을 Topology Aggregation 이라 하며, 라우팅과 네트워크의 확장성 및 보안에 중요한 역할을 하게 된다. 따라서 라우팅 알고리즘과 Topology Aggregation 알고리즘이 PNNI 네트워크의 성능에 중요한 변수가 된다. 이를 비교·분석하기 위해 구현된 PNNI 라우팅 시뮬레이터는 라우팅 알고리즘 및 Topology Aggregation 알고리즘 개발에 적용되어 빠른 기간 내에 보다 개선된 알고리즘 개발에 사용될 수 있다.

### Abstract

In this paper, we focus on comparison and analysis of performance for existing Topology Aggregation algorithm. For these, we designed and implemented PNNI routing simulator which contain various TA schemes, and evaluate performance of TA schemes by this simulator. The PNNI 1.0 specification of the ATM Forum is recommended that hierarchical routing protocol and topology information is aggregated in the network constructed hierarchically. Aggregating topology information is known as TA(Topology Aggregation) and TA is very important for scalability and security in network. Therefore, the performance of PNNI network would vary with TA schemes and routing algorithm. PNNI routing simulator can be applied to develop Routing algorithm and TA algorithm and can be develop these algorithms in short period.

**Keywords :** PNNI, ATM, Routing

\* 正會員, 全北大學校 工科大学 電子工學科

(Dept. of Electronic Eng., Chonbuk Nat'l Univ.)

\*\* 正會員, 韓國電子通信研究員

(Electronics and Telecommunications Research Institute)

\*\*\* 正會員, 群山大學校 工科大学 電子情報工學部

(Division of Electronic and Information Eng., Kunsan Nat'l Univ.)

\*\*\*\* 正會員, 全北大學校 工科大学 電子情報工學部

(Division of Electronic and Information Eng., Chonbuk Nat'l Univ.)

接受日字:2001年8月21日, 수정완료일:2002年5月27日

## I. 서 론

ATM 네트워크의 노드에서 라우팅 경로를 결정하기 위해서는 전체 토폴로지 정보를 가지고 있어야 한다. 그러나 네트워크의 대형화에 따라 네트워크의 라우팅을 위한 전체 토폴로지의 데이터 양은 네트워크의 노드 수와 링크 수에 비례하여 기하학적으로 커지므로, 네트워크 토폴로지 정보를 방송하려면, 방대한 양의 대역과 전달시간과 토폴로지 정보를 저장할 공간이 요구된다. 또한 내부 네트워크 정보의 보안상 이유로 TA(Topology Aggregation)이 해법으로 제시되었다. ATM PNNI (Private Network Node Interface)는 대규모의 ATM 기반 사설 망을 구축하기 위한 계층적이고 동적인 링크 상태 라우팅 프로토콜인 동시에, 점 대 점, 점 대 다중 점 연결 설정을 위한 신호 프로토콜이다. PNNI에서는 계층적 라우팅 방법을 채택함으로써 대규모 망으로의 확장을 가능하게 하였으며, 라우팅 테이블의 규모, 주기적인 갱신 비용을 줄였다. 또한, 라우팅 테이블 프로세스의 자동화 및 QoS를 보장한다<sup>[1]</sup>.

PNNI 토폴로지의 요약정보를 표현하기 위한 방법은 full-mesh, spanning tree, star 방식이 대표적이다. 이 방식들 중 full-mesh가 가장 정확히 토폴로지 정보를 표현하지만 경계 노드 수가  $n$ 이면 링크정보량은  $n \times (n - 1)$ 으로 데이터 양이 상당히 크므로 비실용적이다. spanning tree나 star의 경우  $N$ 에 비례하기 때문에 정보량은 적으나, 토폴로지 정보가 왜곡될 가능성이 있으므로 호 설정할 때 부정확한 정보로 인하여 호 설정이 거부될 가능성이 높다<sup>[2-8]</sup>. 따라서 본 연구에서는 full-mesh, star, spanning tree, simple node방식의 TA 방식의 성능을 측정할 수 있는 PNNI 시뮬레이터를 구현하여, 컴퓨터 시뮬레이션을 통하여 각 방식을 비교 분석하였다.

구현된 PNNI 시뮬레이터는 대규모 망에서 이루어지는 라우팅 프로토콜 등에 적용되어 최적의 경로를 선택하는 라우팅 알고리즘 개발에 적용되어 사용할 수 있고, TA 기법의 분석은 점차 대형 망으로 발전해 가는 현재의 네트워크 발전 추세에 맞추어 계층적으로 네트워크를 구성하여 네트워크 정보를 비교적 정확하고 작은 정보량으로 함축적인 논리 노드를 표현한 다른 네트워크에 전달할 수 있는 TA 알고리즘 개발에 적용되어 사용할 수 있다. 또한 개발된 시뮬레이터는 이러한 라우팅

알고리즘 및 TA 알고리즘 개발에 적용되어 빠른 기간 내에 보다 개선된 알고리즘 개발에 큰 기여를 할 것이다.

## II. PNNI 라우팅 시뮬레이터

시뮬레이터는 TA 알고리즘 및 라우팅 성능 실험 측정을 위해 구현하였으며, ANSI/C++로 작성되어 윈도우즈 NT나 유닉스 시스템에서 사용될 수 있다. 기존의 NIST에서 개발한 APRoPS 시뮬레이터는 PNNI의 라우팅 프로토콜의 성능을 연구하기 위해 개발되어 네트워크 토폴로지를 생성해서 안정 상태에 도달하는데 걸리는 시간과 요구되는 데이터의 양을 측정하는데 사용 가능한 시뮬레이터이다<sup>[9-10]</sup>. 따라서 기존의 TA 알고리즘을 사용하여 경로설정과 crack-back등의 기능은 제공되지 않습니다. 또한 TA은 2레벨이상은 지원하지 못하고 네트워크의 노드수도 최대 50개로 제한되어 WAN과 같은 대규모의 망에서 TA알고리즘의 성능 평가는 불가능하다.

본 논문에서 구현된 PNNI 시뮬레이터는 네트워크 노드 수는 제한이 없으며 2 레벨이상의 토폴로지 구성도 가능하며, 대규모의 네트워크에서 다양한 TA 알고리즘의 라우팅 성능을 측정이 가능하다. 즉, 구현된 시뮬레이터는 ATM Forum의 PNNI 1.0 Spec에 근거하여, 2 계층이상의 TA 및 라우팅, crank-back을 지원하며, Generic-CAC, QoS 라우팅이 포함되어 있다. 구현된 시뮬레이터는 크게 네트워크 토폴로지 생성기, 이벤트 모델, 네트워크 컴포넌트, 라우팅 컴포넌트, 호 처리 컴포넌트로 구성이 되며, TA 방식은 full-mesh, spanning-tree, star와 simple node 방식이 구현되었다.

### 2.1 네트워크 토폴로지 생성기

본 시뮬레이터는 네트워크 토폴로지 생성을 위해 두 가지 버전의 토폴로지 생성기를 구현하였다. 하나는 GUI방식의 토폴로지 생성기로 사용자가 마우스로 그래픽 환경에서 네트워크를 구성할 수 있는 프로그램이고, 하나는 키보드 입력을 받아서 자동으로 네트워크 토폴로지를 생성하는 프로그램이다. PNNI 네트워크는 계층이 커질수록 네트워크의 노드와 링크가 많이 필요하므로, 거대한 네트워크를 구성하기 용이하게 자동 생성 프로그램을 작성했다.

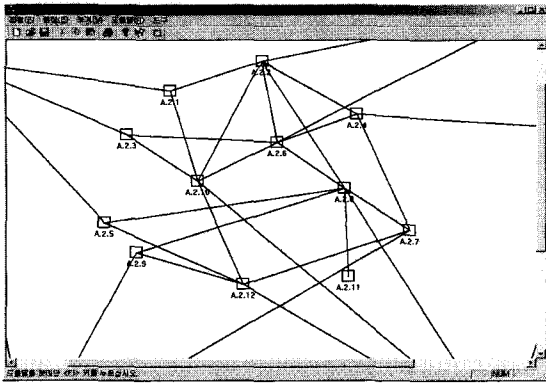


그림 1. PNNI 네트워크 토폴로지 편집기  
Fig. 1. PNNI network topology editor.

2.2 이벤트 모델

PNNI 시뮬레이터는 DES(Discrete Event Simulation) 기법을 적용하여 구현되었다. DES는 각 이벤트의 발생 시점을 이산적으로 발생하지만 시뮬레이션의 시간추이는 연속적인 분포를 가지게 된다. DES 모델의 각 컴포넌트는 프로세스를 실행한 후 다음 발생할 프로세스를 결정한 후 이벤트에 프로세스가 실행할 시점, 즉, 실행 시간을 기록하여 이벤트를 관리하는 이벤트 스케줄러에 이벤트를 삽입한다. 이벤트 관리 모듈은 이벤트가 삽입 되면 이벤트의 실행 시간에 따라 내부 이벤트 리스트를 시간 순서로 정렬한다. 따라서 불필요한 검색과 실행이

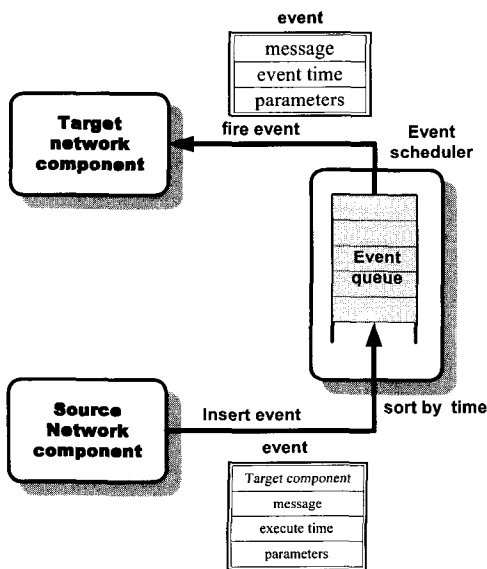


그림 2. 이벤트 스케줄러  
Fig. 2. The event scheduler.

제외되어 CPU 사이클을 줄일 수 있으므로 수행시간이 빠르며, 이벤트를 관리하는 스케줄러가 필요하고 각 객체가 FSM(Finite State Machine)으로 구현된다.

2.3 PNNI 시뮬레이터 구조

2.3.1 객체지향 모델링

본 시뮬레이터는 계층적 네트워크의 물리적인 요소를 컴포넌트로 모델링한다. 네트워크의 물리적인 요소는 크게 스위치(노드), 링크 그리고 ES(end system)으로 구성되어 있다. 물리적인 링크는 링크 객체로 추상화되고, 스위치는 스위치 객체로 추상화되어, 모든 물리적인 네트워크 구성요소는 1:1 객체로 모델링된다. 그림 3은 구현된 시뮬레이터의 구조이다.

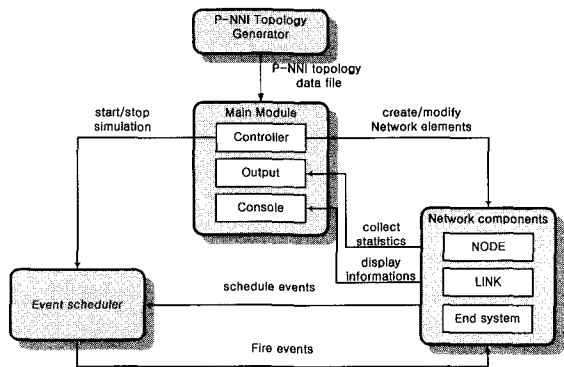


그림 3. PNNI 라우팅 시뮬레이터 블록도  
Fig. 3. The building block of PNNI Routing simulator.

2.3.2 네트워크 컴포넌트

시뮬레이터는 링크, 스위치(노드), ES로 구성된 ATM 네트워크를 각각의 물리요소에 대응하는 객체로 구현하여 재구성한다. 각 객체는 대응하는 물리요소의 동작을 메소드로 구현되어 있다.

i) 링크

링크는 노드와 노드를 연결하는 물리적 링크를 추상화하며, 물리적 링크와 같이 대역폭, 지연과 같은 QoS parameter로 특성화된다.

ii) 스위치

스위치는 ATM 네트워크의 물리적인 스위치를 추상화하며, 스위치가 PNNI 계층적 네트워크의 PG(Peer Group)에 속하여 있으며, 스위치가 PGL(Peer Group Leader)이면, 스위치 객체는 LGN(Logical Group Node)을 포함한다. 또한 스위치 객체는 ATM 스위치의 셀 스

위칭 기능과 CAC, Source-routing 기능이 구현되어 있다.

스위치는 PGL 노트나 Border 노트가 될 수 있으며, Call processing, PTSP(PNNI Topology Summary Packet) flooding, crank-back을 수행한다.

#### iii) LGN(Logical Group Node)

LGN 객체는 하위 PG 리더 노트에 포함되어 있으며, 하위 PG(Peer Group)의 Link Aggregation과 Nodal Aggregation, PTSP flooding을 수행하는 기능을 포함한다.

#### iv) ES(End-System)

물리적인 ES를 추상화한 컴포넌트로서 SES (Source ES)와 DES(Destination ES)로 구성이 된다. 하나의 ES는 각각 스위치에 연결된다.

ES의 SES 객체는 설정된 호 생성률에 따라 포아송 프로세스에 의해 호 생성을 하며, 호가 발생하면 호를 연결하기 위해 연결절차를 실행한다. 발생된 호는 목적지인 DES 객체로 연결되며, Call processing은 스위치나 LGN 컴포넌트가 담당한다. 호는 호의 유일한 번호(ID)와 요구 대역폭, 지연시간을 요구하며, 호의 지속 시간이 미리 설정된 평균 지속 시간에 따라 지수적으로 분포한다.

#### v) Control module

시뮬레이션 환경변수와 네트워크 토폴로지 파일을 읽어서, 네트워크 컴포넌트를 생성하고, 컴포넌트의 데이터를 설정하거나 갱신하며, 메인 모듈의 Output 컴포넌트를 생성한다. 또 시뮬레이션을 수행하고 종료하는 제어 역할과 Output 컴포넌트에 결과를 수집하고, Console에 이벤트나 시뮬레이션 수행 정보를 출력하도록 제어하는 기능을 수행한다.

#### vi) Output 컴포넌트

Output 컴포넌트는 성능평가를 위한 시뮬레이션 결과를 수집하고, 분석하여 출력하는 기능을 수행한다. 결과의 수집은 주기적으로 네트워크 컴포넌트들의 상태를 수집하고, 또 이벤트에 따라 결과를 수집하기도 한다. 또한 네트워크 컴포넌트들도 이벤트에 따라 자율적으로 Output 컴포넌트에 정보를 보고할 수 있다.

### 2.4 라우팅 프로토콜과 시그널링 프로토콜

PNNI 라우팅 프로토콜과 시그널링 프로토콜은 각각 라우팅 프로토콜 컴포넌트와 호처리 컴포넌트에 의해

구현된다. 이 두 개의 컴포넌트는 스위치와 LGN 컴포넌트에 포함되며, 스위치와 LGN에 따라 다르게 동작한다.

#### 2.4.1 라우팅 프로토콜 컴포넌트

라우팅 프로토콜 컴포넌트는 크게 두 가지 컴포넌트로 구성된다. 토폴로지 데이터베이스와 라우팅 경로 결정 알고리즘으로써 토폴로지 데이터베이스는 네트워크의 스위치와 링크의 상태 정보가 저장되어 있고, 토폴로지 데이터베이스를 이용하여 라우팅 경로 결정은 요구된 QoS를 만족하는 라우팅경로를 계산한다. 라우팅 경로 결정은 Dijkstra 알고리즘과 SPF(shortest path first)이 구현되어 있다. 또한 PTSP flooding 과 Hello 프로토콜이 포함한다. LGN 컴포넌트의 라우팅 프로토콜은 TA을 수행하며, 기존의 full-mesh, spanning tree, star, simple node 방식의 알고리즘이 구현되어 있다.

#### 2.4.2 호처리 컴포넌트

PNNI 시그널링은 UNI(User Network Interface) 신호 방식에 기초한다. PNNI 라우팅 프로토콜에 의해서 망의 모든 노드의 토폴로지 데이터 베이스가 동기화 되면, 사용자로부터 연결 요청을 받아 이를 위한 경로를 결정하고 네트워크 자원을 할당한다.

호 처리 시, PNNI 시그널링은 PNNI 라우팅으로부터 하나의 경로를 요구할 수 있는데, PNNI는 이 경로를 DTL(Designated Transit List) 스택으로 표현한다. 하나의 DTL은 각 피어 그룹을 거처가는 하나의 완전한 경로를 나타내며, 일련의 노드 ID와 Port로 구성된다. DTL은 소스 노트나 PG으로의 entry border 노트에 의해 제공된다. 계층적으로 완전한 source routing은 최하위 피어 그룹 계층으로부터 최상위 피어 그룹 계층의 순서로 일련의 DTL로 표현되고, DTL은 경로를 저장하는 스택으로 구성된다(그림 4).

가장 아래에는 최상위 계층 PG에 대한 경로가 포함되어 있고, 가장 위에는 최하위 계층 PG에 대한 경로가 포함된다. 또한 SVC 설정 요청이 현재 어느 노트에 위치하는지를 알 수 있도록 포인터 정보가 포함된다. 포인터가 DTL의 마지막 노트에 도달하게되면 이 DTL은 스택에서 제거된다. 그리고 다음 DTL에 대한 처리가 진행된다. SVC 설정 요청이 경로를 따라 새로운 하위 계층 PG에 들어가게 되면 새로운 DTL이 생성되어 스택에 삽입된 후 처리가 진행된다. DTL을 생성 할 때, 노트는 현재 이용 가능한 자원 및 연결 정보를 이용하

는데, 호를 DTL에 따라 처리할 수 없는 경우, crank-back 발생 이유를 부가하여 DTL의 생성 노드로 되돌려 보내 해제한다. 이 DTL 생성 노드는 차단 경로를 피해 다른 경로를 선택하고, 그 호를 처리하거나, 그럴 수 없는 경우 그 이전 DTL 생성 노드로 다시 되돌려 보내어 다른 경로를 찾기 위한 crank-back과 alternation 라우팅이 사용된다.

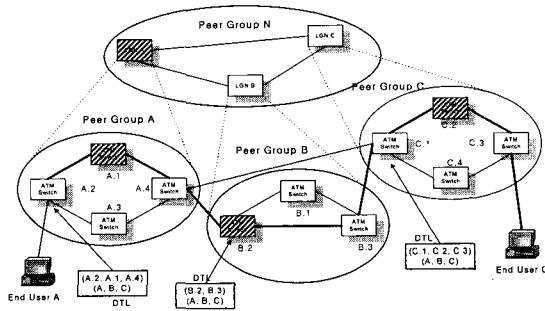


그림 4. 경로 리스트  
Fig. 4. Designated transit list.

### III. 토폴로지 요약 알고리즘

#### 3.1 Simple node 방식

Simple node 방식에서는 PG의 내부 토폴로지는 하나의 LGN으로 집약된다. 즉 n개의 논리 포트를 연결하는 내부 논리 링크가 하나로 집약되므로, 이와 같은 경우 TA 정보의 크기를 가장 작게 해준다. 그러나 비대칭 토폴로지 정보를 적절하게 반영하지 못하고, 본래의 PG의 다중 연결특성을 요약하지 못한다. 또한  $2 \times n \times (n - 1)$  개의 내부 링크 상태 정보를 하나의 내부 링크 상태 정보의 대표값으로 요약해야 하므로 어떤 값을 사용하는지의 선택이 쉽지 않다. 대부분 평균값을 쓰거나 가장 작은 값을 쓰게 되는데, 경계 노드 사이의 논리 링크 상태의 QoS 파라미터가 비교적 고르게 분포 되어있을 때에는 커다란 압축 효과를 얻을 수 있다. 그러나 링크의 QoS 파라미터가 동적으로 변하고, 방향에 따라서도 QoS 파라미터가 다르게 나타나므로 네트워크의 동적 변화에 대응하기 곤란한 방법이기도 하다. 또한 어떤 경계 노드에서 다른 경계 노드로의 전송 대역폭 고갈로 인한 영향이 논리 링크 상태 QoS 파라미터에 많은 영향을 주어 네트워크 자원을 효율적으로 사용하지 못하는 경우가 발생한다. 그럼에도 불구하고 crank-back을 이용한 경로 재할당 프로토콜을 이용하여 PNNI 망 전

체의 성능을 어느 정도 유지할 수 있다. 또한 simple node 방식은 PNNI 라우팅 프로토콜의 오버헤드가 가장 적고 구현의 복잡도가 가장 간단하기 때문에 많이 이용되고 있는 기법이다<sup>[1, 3, 5]</sup>.

#### 3.2 Full-mesh

Full-mesh 방식은 LGN 내부의 n개의 포트가 full-mesh로 형태로 연결된다. full-mesh 구조에서 논리 링크 상태 파라미터 값을 양방향 모두 같은 값으로 한다면  $n \times (n - 1)$ 개의 논리 링크가 필요하고, 방향성을 고려한다면  $2 \times n \times (n - 1)$ 개의 논리 링크가 필요하게 된다(그림 5). Full-mesh 방식은 PG내의 토폴로지를 보다 정확하게 요약할 수 있으나, 요약 정보의 양이 많아 라우팅 프로토콜 오버헤드가 증가하게 된다. 또한 Border 노드 수의 제곱에 비례하여 논리 링크 수가 증가하므로 경계 노드 수가 증가하면 거의 사용할 수 없는 기법이다. 그러나 다른 모든 구조를 구성하기 위한 기본 구조가 되므로 매우 중요한 구조이다.

Full-mesh TA 기법의 핵심은 논리 링크 상태 QoS 파라미터를 결정하는 방법이다. LGN 내부 링크 상태 QoS 파라미터는 다른 LGN에서 TA LGN을 통과하여 다른 LGN으로의 연결 설정 시 제공 가능한 QoS 파라미터를 의미한다. 즉, Full-mesh 기법은 PG의 내부 토폴로지를 기본으로 하여 임의의 경계 노드에서 다른 경계노드로의 제공 가능한 QoS 파라미터를 구하는 것이다. PG내에 임의의 경계 노드 사이에는 다중 경로가 존재하고, 각 경로마다 제공 가능한 QoS 파라미터가 다르므로, 다중 경로와 다중 파라미터를 요약하는 기법이 다른 구조의 TA 기법에 많은 영향을 미치는 중요한 연구 과제이다<sup>[5-8]</sup>.

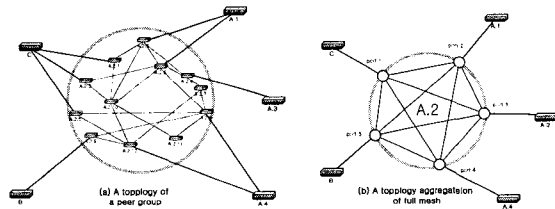


그림 5. full-mesh 구조  
Fig. 5. The structure of a full-mesh.

#### 3.3 Spanning tree

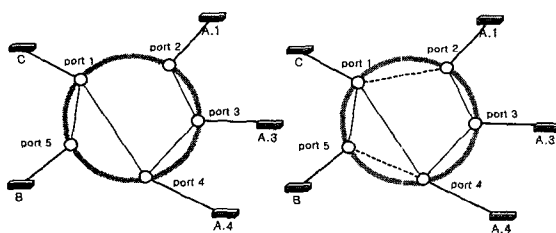
Spanning tree란 연결 그래프 G가 있을 때 그래프 G의 모든 정점을 포함하고 간선(두 정점간의 연결선)은

$G$ 의 부분 집합으로 구성된 tree를 말한다. Spanning tree는 연결 그래프  $G$ 가  $n$ 개의 정점(노드)과  $e$ 개의 링크를 가진 경우,  $n$ 개의 정점과  $n-1$ 개의 간선으로 구성된다. 모든 노드들은  $n-1$ 개의 간선들을 결합하여 서로 연결될 수 있다. 각각의 간선, 즉 링크에는 가중치(weight)가 할당되며, 이 가중치에 따라 spanning tree가 구성되어진다. 이런 가중치는 링크 용량, 지연 등의 요소이며, 결국 라우팅 경로를 선택하는데 필요한 비용이다. spanning tree 구성은 Prim 알고리즘을 이용하여 구할 수 있다<sup>[2, 4]</sup>. 그림 6은 spanning tree와 bypass를 추가한 T-spanning tree이다.

Spanning tree는 그 가중치를 구성하는 방법에 따라 크게 두 가지 방법으로 구현될 수 있다. minimum spanning tree(minST)는 tree위의 가중치의 총합이 최소가 되도록 모든 정점을 연결한 그래프이다. maximum spanning tree(maxST)는 tree위의 가중치의 총합이 최대가 되도록 모든 정점을 연결한 그래프이다<sup>[8]</sup>.

여기에선 링크 상태의 통합을 위해 spanning tree 방법을 이용한다. 링크의 상태벡터로서 대역폭과 지연을 고려하고 있다. 대역폭은 non-additive 링크 상태 파라미터로 보고 있으며, 지연은 additive 링크 상태 파라미터로 보고 있다. 그러므로 최적의 토폴로지 통합이 이루어지려면, 대역폭은 최대로 나타낼 수 있고, 지연은 최소로 나타낼 수 있는 방법이 필요하다.

Spanning tree를 구성하기 위해 full-mesh의 구성이 필요한데, 이러한 full-mesh는 대역폭을 최대로 할 것인지, 지연을 최소로 할 것인지에 따라 그 구현 방법이 달라진다<sup>[2, 4, 5]</sup>.



(d) Spanning tree

(e) T-spanning tree

그림 6. Spanning tree 구조

Fig. 6. The structure of a spanning tree.

### 3.4 Star

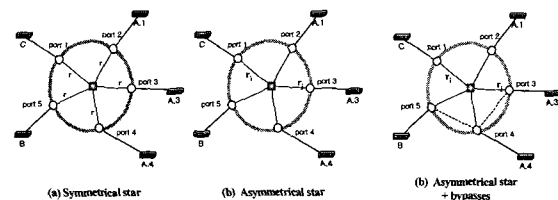
Star 구조는 복합 구조와 함께 PNNI 1.0 규격에서 제안한 것으로 구조가 간단하고 변하지 않는 장점이 있다

<sup>[1, 6, 7]</sup>. Star 구조의 논리 링크 상태 파라미터에 따라 차이가 있지만 full-mesh 요약 기법에 비해 매우 큰 압축 효과가 있다. Star 구조의 논리 링크 상태 파라미터 결정 방법에 따라 symmetric star, asymmetric star, bidirectional asymmetric star 구조가 있다

Star 구조는 LGN 내부 구조에 가상 노드를 추가하는데 이를 nucleus node라 한다. 또한 포트와 nucleus 노드 사이의 논리 링크를 spoke link라 한다. Spoke link의 모든 QoS 상태 파라미터가 모두 같은 구조를 symmetric star 구조라 한다. Symmetric star 구조는 QoS 파라미터가 모두 같으므로 Simple node TA 기법과 같은 구조라 볼 수 있다. Symmetric star 구조의 spoke link 상태 파라미터는 simple node 구조의 상태 파라미터와 같은 방법으로 구하여 반으로 나눈 값이 된다. Symmetric star 구조는 많은 압축 효과를 볼 수 있지만, full mesh 구조와 많은 차이가 난다. 이를 보완하기 위해 bypass link를 추가하면 복합 구조 TA 기법이 된다. Bypass link란 nucleus 노드를 거치지 않고 바로 포트 사이를 연결한 링크를 말한다. Bypass link의 결정은 모든 포트 사이의 논리 링크 상태 파라미터를 full mesh 구조와 star 구조에서 구하여 이들을 비교하여 결정한다<sup>[5-8]</sup>.

Asymmetric star 구조는 spoke link QoS 파라미터 값이 모두 다른 값을 가진다. Symmetric star 구조는 포트 사이의 QoS 파라미터 값이 비슷한 대칭 구조에 적합한 구조라 할 수 있는데, QoS 파라미터는 동적으로 변하므로 이에 적응하기 위해 asymmetric star 구조를 이용한다.

Bidirectional asymmetric star 구조는 Asymmetric star 구조가 spoke link의 방향성이 고려되지 않았던 것에 비해 spoke link의 양방향의 링크 상태 QoS 파라미터를 다르게 구한 것이다. Bidirectional asymmetric star 구조는 정보량은 다른 star 구조에 비해서 많으나 full mesh 구조와 거의 같은 수준으로 요약할 수 있다.



(a) Symmetrical star

(b) Asymmetrical star

(b) Asymmetrical star + bypasses

그림 7. Star 구조

Fig. 7. The structure of a star.

#### IV. Topology Aggregation의 분석

PNNI 시뮬레이션은 TA 알고리즘과 라우팅 알고리즘에 따른 성능비교를 목적으로 수행하였다. 시뮬레이션에 적용된 알고리즘은 full-mesh, spanning tree, star 그리고 simple node 방식을 각각 TA방식에 따라 시뮬레이션 하였다. 시뮬레이션을 위한 망은 네트워크 토폴로지 생성기를 사용하여 여러 개의 PG와 PG를 구성하는 노드들로 구성되었으며, TA는 simple node 방식과 complex node 방식을 적용할 때 라우팅시 경로 선택에서 링크의 지연 시간 또는 대역폭을 기준으로 수행할 경우에 따라 시뮬레이션을 수행하여 호 성공률, crank-back 발생횟수, 액세스 시간을 비교하였다.

시뮬레이션을 수행할 네트워크 모델은 3계층의 TA이 가능한 네트워크로서 최상위 PG가 3개이고, 각각의 LGN의 주소가 "A", "B", "C"고 최하위 물리 노드 주소는 "A.X.X", "B.X.X", "C.X.X"이며, 150개의 물리 노드와 760의 링크로 이루어져 있다. 네트워크 토폴로지는 많은 노드와 그 이상의 링크로 구성되므로 네트워크 토폴로지 생성기를 이용하여 구성하였다.

표 1은 시뮬레이션 환경 변수를 보여준다. 네트워크의 부하는 각 ES에 평균 호 발생률을 설정하여, 평균 호 발생률에 따른 포아송 프로세스로 호가 생성된다. 호 설정이 성공하면 호는 지수분포에 따른 호 지속 시간동안 호 연결이 지속된다.

표 1. 시뮬레이션 파라미터

Table 1. Simulation parameters.

시뮬레이션 환경 변수	값
PTSE refresh time	300 ~ 900 sec
Call arrival rate	0.009 ~ 0.016(call/sec/ node)
Mean call period	300 sec
Request bandwidth	1 ~ 3 Mbps
Request delay	10 ~ 100 msec
Max Call setup time	100 sec
Max crank-back 횟수	5회
Link bandwidth	155Mbps
Link delay	0.0001 ~ 0.0007 (sec)
Simulation time	5000 (sec)

그림 8, 9, 10는 TA알고리즘의 평균 호 설정시간, 호 성공률, crank-back 비율을 보여준다. 호 설정 시간에서는 simple node가 가장 좋은 성능을 보여주고 있으며, 호 성공률은 full-mesh 방식이 다른 방식에 비해 우수

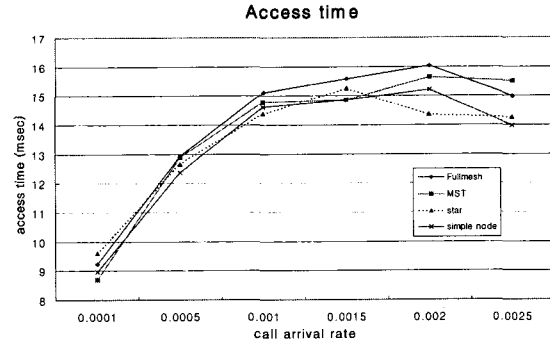


그림 8. 기존 알고리즘 별 access time(with crank-back)

Fig. 8. Access time of traditional TAs(with crank-back).

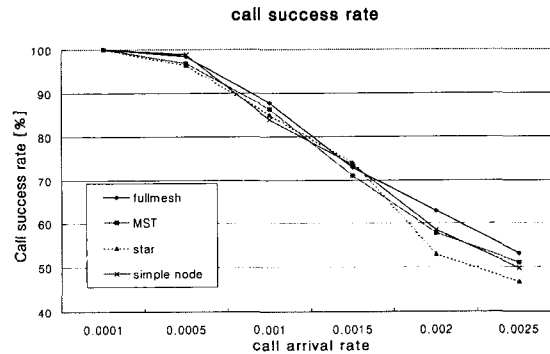


그림 9. 기존 알고리즘 별 호 성공률(with crank-back)

Fig. 9. Call success rates of traditional TAs.(with crank-back)

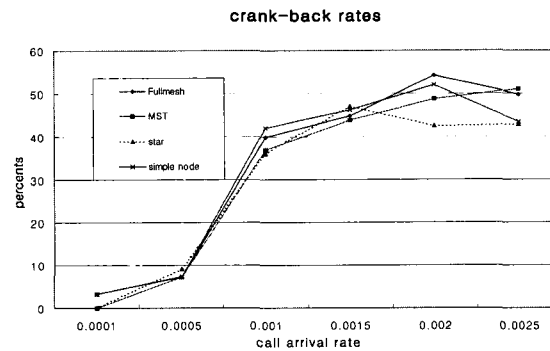


그림 10. 기존 알고리즘 별 crank-back rates

Fig. 10. Crank-back rates of traditional TAs.

하며, spanning tree, 와 star방식은 비슷한 성능을 가지며, simple node 방식이 가장 낮은 성공률을 보여준다. 이것은 full-mesh가 가장 정확한 토폴로지 정보를 제공하며 simple node는 PG을 하나의 점으로 나타내므로 정보량은 가장 작은 반면, 가장 부정확한 정보를 제공한다고 볼 수 있다.

그림 11은 full-mesh 와 simple node를 적용했을 때, crank-back이 없는 경우와 있는 경우의 호 성공율이다. crank-back이 없는 경우 crank-back을 적용했을 경우보다 호 성공률은 상당히 낮아짐을 알 수 있다. 따라서 crank-back은 토폴로지 정보의 부정확성으로 인해 경로 설정이 잘못될 경우 경로를 재설정하여 호 성공률을 높이는 장점이 있다. 이는 crank-back이 PNNI 라우팅 프로토콜에서는 아주 중요한 변수가 된다는 것을 알 수 있다.

그림 12는 crank-back을 사용한 경우와 crank-back 사용하지 않은 경우의 평균 호 설정 시간이다. crank-back을 사용한 경우 평균 호설정 시간이 증가하는데 이것은 경로를 따라서 호 설정 중에 호가 요구하는 QoS를 만족시키지 않을 경우, crank-back에 의한 경로 재설정으로 목적지까지 호 설정이 성공하는데 요구되는 시간이 증가했다고 볼 수 있다. 물론 crank-back이 많아지면, 네트워크 자원을 많이 소모하고 시그널링 오버헤드가 증가하는 단점이 있지만, 그룹내의 토폴로지 요약 정보를 이용하고, 실제의 링크 상태 정보를 이용한 경로 설정이 가능하지 않다면 crank-back은 꼭 필요한 과정이라 할 수 있겠다. 다만 crank-back 비율을 줄이기 위해 가능한 호의 경로 설정을 분산시킬 수 있는 방안의 연구도 병행되어야 하겠다.

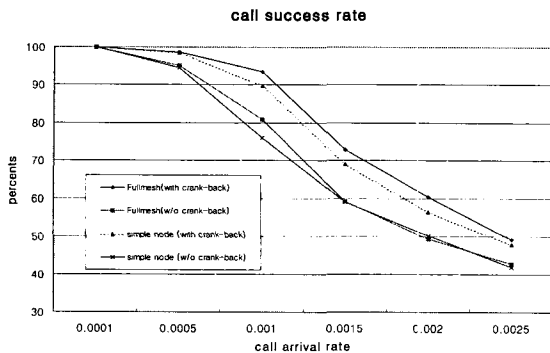


그림 11. full-mesh 와 simple node의 호 성공율  
Fig. 11. Call success rates of full-mesh vs simple node.

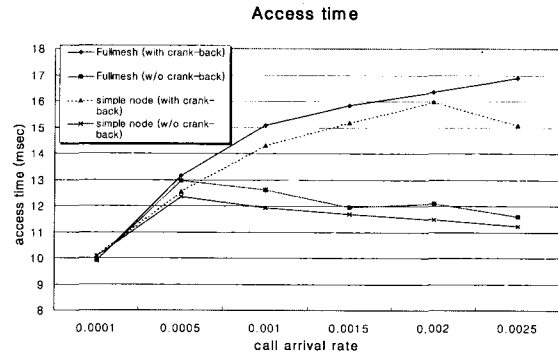


그림 12. full-mesh 와 simple node의 access time  
Fig. 12. Access time of full-mesh vs simple node.

## V. 결 론

본 논문에서는 TA 알고리즘의 기본이 되는 full-mesh, spanning tree, star, simple node 기법들의 성능 분석을 위해 PNNI 라우팅 시뮬레이터를 개발하고, 시뮬레이션을 통해 그 성능을 비교 분석하였다. PNNI 시뮬레이터는 네트워크 토폴로지 생성기, 3 계층 PNNI 망의 형성, PNNI 라우팅 및 시그널 프로토콜, 호 레벨 경로 설정이 가능하고, 다양한 TA 알고리즘이 구현되어 있다. 기존 알고리즘 분석 결과 정보량이 많은 full-mesh 기법이 다른 기법보다 다소 나은 성능을 보이고 있으나, spanning tree나 star 구조의 정보량에 비하면 그 효과는 그리 크지 않다고 볼 수 있다. 이것은 PNNI 망에서 경로를 찾기 위한 라우팅 테이블의 링크 상태 정보가 실제 링크 상태 정보와 다를 수 있기 때문이다. 또한 라우팅 테이블은 갱신주기 동안은 변하지 않는데, 실제 링크 상태 정보는 동적으로 변화하기 때문이다. 이는 spanning tree, star, simple node 방식이 full-mesh 보다 적은 정보량을 가지지만 성공률에 큰 차이를 보이지 않는 원인이 되고, TA 기법을 개발하는데 어려운 점이 되기도 한다.

시뮬레이션 결과 spanning tree, star 구조의 정보가 full-mesh 구조의 정보와 큰 차이가 나고 있지 않음을 알 수 있다. simple node 방식의 호 성공률도 다른 기법의 호 성공률에 비해서 크게 떨어지지 않는 것으로 봐서, 대규모 망의 TA 방식에 적용하기 위해서는 정보량을 줄일 수 있는 spanning tree, star 구조의 TA 방식이 타당할 것으로 보인다.

구현된 시뮬레이터는 점차 대형 망으로 발전해 가는 현재의 네트워크 발전 추세 및 새로운 네트워크 기기에



맞추어 라우팅 알고리즘 및 TA 알고리즘 개발에 적용되어 사용할 수 있으며, 빠른 기간 내에 보다 개선된 알고리즘 개발에 기여를 할 것으로 예상된다.

앞으로의 연구과제는 구현된 시뮬레이터를 이용하여 대규모 망에서도 적용할 수 있는 가능한 한 정확하고 적은 정보량으로 보다 함축적인 논리 노드로 표현할 수 있는 TA 알고리즘의 연구와 라우팅 알고리즘의 연구가 필요하다. 또한 PNNI 망에서 경로 설정을 분산할 수 있는 알고리즘에 관한 연구가 TA 알고리즘과 병행되어 수행되어야 하겠다.

참 고 문 헌

[1] Private Network-Network Interface Specification Version 1.0(PNNI 1.0), ATM Forum af-pnni-0055.000, Mar, 1996.

[2] Erol Basturk, Paul Stirpe, "A Hybrid Spanning Tree Algorithm for Efficient Topology Distribution in PNNI", Proceedings of the 1998 1st IEEE International Conference on ATM, 1998.

[3] T. Korkmaz, M. Krunk, "Source-oriented topology aggregation with multiple QoS parameters in hierarchical ATM networks", IEEE, p.137-146, 1999.

[4] Whay. C. Lee, "Spanning Tree Method for Link State Aggregation in Large Communication Network", Proceedings of the IEEE infocom '95, Vol 1, 1995,

[5] Baruch Awerbuch, Yuval Shavitt, "Topology Aggregation for Directed Graph" Proceedings of the Third IEEE Symposium on Computers and Communications, 1998.

[6] Whay C. Lee, "Topology Aggregation for Hierarchical Routing in ATM Networks", IEEE, 1998.

[7] King-Shan Lui, Klara Mahrstedt, "Topology Aggregation of Bandwidth-Delay sensitive Networks", IEEE, 1999.

[8] Whay C. Lee, "Topology Aggregation for Hierarchical Routing in ATM Networks", IEEE, 1998.

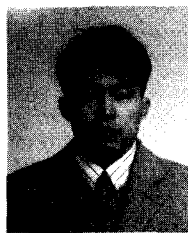
[9] Yunning Song, "The NIST ATM PNNI Routing Protocol Simulator(APRoPS), Operations and Programming Guide", Version 1.1, June, 1999.

[10] [Http://w3.antd.nist.gov/Hsntg/prd\\_aprops.html](http://w3.antd.nist.gov/Hsntg/prd_aprops.html), Home Page of AProPS

저 자 소 개

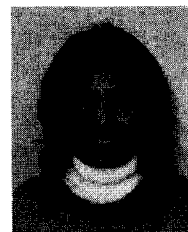


金 辯 坤(正會員)  
1990년 2월: 한국항공대학교 전자공학과(공학사). 1997년 2월: 전북대학교 대학원 전자공학과(공학석사). 2001년 8월: 전북대학교 대학원 전자공학과(공학박사). 2002년 3월~현재: 군산대학교 전자정보공학부 BK계약교수. <주관심분야: ATM 트래픽 흐름제어, PNNI, 라우팅알고리즘>



丁 光 日(正會員)  
1997년 2월: 전북대학교 전자공학과(공학사). 1999년 2월: 전북대학교 대학원 전자공학과(공학석사). 2001년 2월: 전북대학교 대학원 전자공학과 박사과정 수료 <주관심분야: ATM 트래픽 흐름제어, 라우팅알고리즘, 초고속 인터넷>

金 觀 雄(正會員) 第38卷 SD編 第4號 參照



申 鉉 順(正會員)  
1986년 2월: 한남대학교 전자계산학과(이학사). 1990년: 한남대학교 전자계산학과(이학석사). 2001년~현재: 충북대학교 대학원 컴퓨터과학과 박사과정. 1986년 3월~현재: 한국전자통신연구원 네트워크연구소

선임연구원 <주관심분야: ATM 스위칭, 주소분석 및 관리, 망관리, 네트워크 프로토콜(PNNI, OSPF, RSVP), 라우팅 알고리즘, QoS, 인터넷, 유무선 통합>

鄭 晁 澤(正會員) 第39卷 TC編 第4號 參照

田 炳 實(正會員) 第39卷 TC編 第4號 參照