

---

# 사설망 인터페이스에서 토폴로지 요약 테스트를 위한 시뮬레이터 설계 구현 및 TA 알고리즘 성능분석

## Design and Implementation of Simulator for Topology Aggregation in Private Networks to Networks Interface and Performance Analysis of TA Algorithms

---

김남희\*, 김변곤\*\*, 서혜영\*, 박기홍\*  
군산대학교 컴퓨터정보과학과\*, 군산대학교 전자정보공학부\*\*

Nam-Hee Kim(nhkim@kunsan.ac.kr)\*, Byun-Gon Kim(bgkim@kunsan.ac.kr)\*\*,  
Hae-Young Seo(cute100@hotmail.com)\*, Ki-Hong Park(spacepark@kunsan.ac.kr)\*\*

---

### 요약

토폴로지 정보는 망에서 계층적으로 구성될 수 있으며, 토폴로지 정보를 요약하는 과정을 토폴로지 요약이라 한다. 이는 라우팅과 네트워크의 확장성에 매우 중요한 요소이다. 특히, 사설망 인터페이스에서 라우팅 알고리즘과 토폴로지 요약 알고리즘은 네트워크의 성능에 중요한 변수가 된다. 따라서 본 논문에서는 사설망 인터페이스에서 토폴로지 요약을 위한 라우팅 시뮬레이터를 설계 및 구현하였다. 그리고 구현된 시뮬레이터를 사용하여 기존 토폴로지 요약 알고리즘의 성능을 분석하였다. 구현된 사설망 인터페이스 시뮬레이터는 토폴로지 요약 알고리즘 개발에 유용하게 사용될 수 있다.

■ 중심어 : | 시뮬레이터 | 토폴로지 요약 |

### Abstract

Topology information can be constructed hierarchically in the networks and the process of aggregating topology information is known as TA(Topology Aggregation). It is very important for scalability in networks. In particular, routing and TA algorithm are very much elements on networks performance in private networks to networks interface. Therefore, in this paper, we designed and implemented routing simulator for TA in private networks to networks interface. And we evaluated and analyzed the existent TA algorithm using the implemented simulator. The implemented simulator can be used valuably to developing TA algorithm.

■ keyword : | Simulator | Topology Aggregation |

---

## 1. 서론

사설망 인터페이스 규격은 복잡한 구조를 가지고 있어 공중망에 적용하기는 현실적인 어려움이 있지만, 망의 동적인 변화에 적응성이 뛰어나고, 다양한

QoS(Quality of Service) 제공 능력을 가지고 있다[1]. 근래의 사설망 인터페이스 규격은 대부분의 상용 사설 ATM 스위치에서 구현되고 있으며, 공중망에의 적용 가능성과 함께 B-ISUP 공중망간의 효율적인 상호 연동을 위한 많은 연구가 진행되고 있다. 사설망 인터페

---

\* 본 논문은 2005년도 군산대학교 학술연구비의 지원에 의하여 연구되었습니다.

이스는 계층 구조를 이루고 각 레벨마다 여러 개의 PG(peer group)으로 구성된다. 각 그룹내의 PG leader가 상위나 하위 레벨과 정보를 주고받아, 각 레벨에서 사용 가능한 자원의 정보를 모든 레벨의 노드들이 알 수 있도록 한다[2].

노드들은 이들 정보를 이용하여 소스라우팅(source-routing)을 하게 된다. 소스 라우팅은 연결 설정 단계에서만 경로를 검색하므로, 라우팅 알고리즘의 속도보다는 얼마나 최적의 경로를 찾아내느냐가 더욱 중요하다. 이를 위해 ATM 포럼에서는 각 PG의 자원 정보를 얼마나 정확하고 효율적으로 요약할 것인가에 많은 노력을 기울이고 있다[1][3]. TA는 PG 내부 토폴로지 정보를 하나의 가상 노드(LGN : Logical Group Node)로 요약하는 것이다. 토폴로지 요약정보는 LGN 노드 도달 정보, 다른 LGN과의 가상 논리 링크 상태 정보, LGN 내부 구조 및 상태 정보가 포함된다. LGN 노드 도달 정보는 ATM 주소의 prefix를 이용하여 요약된다. 다른 LGN과의 가상 논리 링크 상태 정보는 사설망 인터페이스의 계층적인 구조에 따라 결정된다. 그러나, LGN 내부 구조 및 상태 정보는 토폴로지 요약 기법에 따라 정보량 및 정확성에 많은 차이가 나기 때문에 초기의 연구는 가상적인 내부 구조를 중심으로 이루어졌다[4][5]. 각 LGN은 포트를 통하여 다른 LGN과 연결되는데, 이러한 포트들의 내부 가상 연결을 위해 토폴로지 요약기법인 full mesh, star, spanning tree, star, 복합 구조 등이 이용되고 있다. Full mesh 토폴로지 요약 구조는 본래의 통신망 내부 토폴로지를 충분히 표현하고 있지만, 정보량이 많아 주로 다른 구조를 만들기 위한 전 단계로 이용된다. Full mesh 구조의 모든 가상 논리 링크에서 제공 가능한 QoS 파라미터를 결정하고 가능한 적은 수의 논리 링크를 가지고 full mesh를 대체하기 위해 spanning tree, star, simple node 등의 토폴로지 요약기법이 사용된다[6-9]. 이들 토폴로지 요약 기법은 망을 확장하는데 있어 매우 중요한 역할을 하고 또한 망에서의 효율적 라우팅을 위하여 필수적이다. 특히, 라우팅 알고리즘과 토폴로지 요약 알고리즘은 네트워크의 성능을 향상시키는데 있어 중요한 변수가 된다. 그러나 사설망 인터페이스를 위한 향상된 망요약 알고

리즘을 제안하였을 때 이를 쉽게 시뮬레이션 하기 위한 시뮬레이터의 개발은 매우 미비하다. 따라서 본 논문에서는 사설망 인터페이스에서의 TA 알고리즘을 쉽고 효율적으로 시뮬레이션 하기 위한 시뮬레이터를 설계하고 구현하였다. 그리고 구현된 시뮬레이터를 이용하여 기존 TA 기법인 full mesh, MST(minimum spanning tree), star, simple node 기법에 대해 구현한 시뮬레이터를 이용하여 성능을 분석하였다. 본 논문의 구성은 서론에 이어 제 2장에서는 구현하는 시뮬레이터의 이벤트 모델, 시뮬레이터의 모듈별 구성, 라우팅과 호처리 컴포넌트 등 구현하는 시뮬레이터 구성에 대해 기술하였고 3장에서는 구현된 시뮬레이터를 이용하여 기존 TA 알고리즘인 full mesh, MST(Minimum Spanning Tree), star, simple node 기법을 시뮬레이션 하여 성능을 분석하였고, 마지막으로 결론을 맺었다.

## II. Simulator 구현

본 논문에서 구현한 시뮬레이터는 토폴로지 요약과 라우팅 성능을 실험측정 할 수 있도록 하였으며 ANSI/C++로 작성되어 윈도우 NT나 유닉스 시스템에서 사용될 수 있다. 구현된 시뮬레이터는 ATM 포럼의 P-NNI 1.0 Spec에 근거하여, 2계층이상의 토폴로지 요약과 라우팅, crank-back을 지원하며, Generic-CAC, QoS 라우팅이 포함되어 있다.

### 1.1 이벤트 모델

본 논문에서 구현한 시뮬레이터의 discrete event 방식을 사용하였다. Discrete event simulation은 각 이벤트의 발생시점은 이산적으로 발생하지만 시뮬레이션의 시간추이는 연속적인 분포를 가지게 된다. 네트워크 시뮬레이션 방식은 크게 discrete time simulation과 discrete event simulation 방식으로 나눌 수 있다. discrete time 모델의 경우 같은 시간간격으로 시뮬레이션의 모든 컴포넌트(노드, 링크, 스위치)를 루프로 실행하면서 수행하는 방식으로 시간간격이 모든 프로세싱 중에서 작은 시간을 기준으로 실행하기 때문에 시뮬레

이전 실행 시간이 상당히 길어지고 불필요한 실행이 발생한다. 반면 discrete event 시뮬레이션 모델의 각 컴포넌트는 프로세스를 실행한 후 다음 발생할 프로세스를 결정한 후 이벤트에 프로세스가 실행할 시점(시간)을 기록하여 이벤트를 관리하는 이벤트 스케줄러에 이벤트를 삽입한다. 이벤트 관리 모듈은 이벤트가 삽입되면 이벤트의 실행 시간에 따라 내부 이벤트 리스트를 시간 순서로 정렬한다. 따라서 불필요한 검색과 실행이 제외되어 CPU 사이클을 줄일 수 있으므로 수행시간이 빠른 장점이 있다.

### 1.2 시뮬레이터 구성

본 시뮬레이터는 계층적 네트워크의 물리적인 요소를 컴포넌트로 모델링 하였다. 네트워크의 물리적인 요소는 크게 스위치(노드)와 링크, End system의로 구성되어 있으며, 이들을 객체로 모델링 하였다. 물리적인 링크는 링크 객체로 추상화되고, 스위치는 스위치객체로 추상되어, 모든 물리적인 네트워크 구성요소는 1:1 객체로 모델링 되었다.

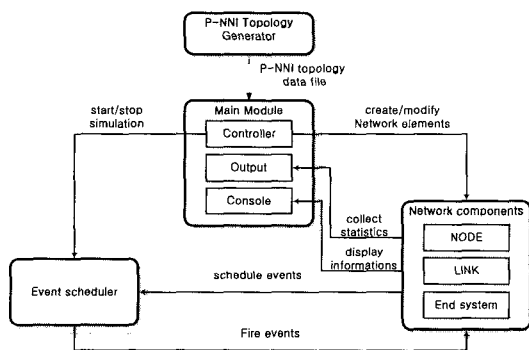


그림 1. 시뮬레이터 블록도

[그림 1]은 시뮬레이터의 블록도를 나타낸다. 각 블록에 대한 내용은 아래와 같다.

#### ■ 네트워크 컴포넌트

시뮬레이터는 링크, 노드, ES로 구성된 ATM 네트워크를 각각의 물리요소에 대응하는 객체로 구현하여 재

구성한다. 각 객체는 대응하는 물리요소의 동작을 메소드로 구현되어 있다.

#### ■ 링크

노드와 노드를 연결하는 물리링크를 추상화하며, 물리링크와 같이 대역폭, 지연과 AW(Administrative Weight)로 특성화된다.

#### ■ 스위치

ATM네트워크의 물리적인 스위치를 추상화하며, 스위치가 P-NNI 계층적 네트워크의 피어그룹에 속하여 있으며, 스위치가 PG(Peer Group) 리더이면, 스위치 객체는 LGN(Logical group node)를 포함한다. 또한 스위치객체는 ATM 스위치의 셀 스위칭 기능과 CAC, Source-routing 기능이 구현되어 있다. 스위치는 PG Leader 노드나 border 노드가 될 수 있으며, call processing, PTSP flooding, re-routing 기능을 구현한다.

#### ■ LGN(Logical Group Node)

LGN객체는 하위 PG 리더 노드에 포함되어 있으며, 하위 PG의 Link Aggregation과 Nodal aggregation, PTSP flooding을 수행하는 기능을 포함하였다.

#### ■ End-System

물리적인 ES를 추상화한 컴포넌트로서 소스 ES(SES)와 목적지 ES(DES: Destination ES)로 구성된다. 하나의 ES는 각각 스위치에 연결된다.

ES의 SES객체는 설정된 호 생성율에 따라 포아송 프로세스에 의해 호 생성을 하며, 콜발생에 따라 ATM 셀이 연결된 스위치로 전송된다. 발생한 호는 목적지 ES의 DES 객체로 연결되며, call processing은 스위치나 LGN 컴포넌트가 담당하도록 하였다.

#### ■ 메인 모듈

시뮬레이션 환경을 구성하고 제어하고, 결과를 수집하고, 분석, 출력하는 모듈로서, control module, output, console 컴포넌트가 있다.

▪ Control module

시뮬레이션 환경변수와 네트워크 토폴로지 파일을 읽어서, 네트워크 컴포넌트를 생성하고, 컴포넌트의 데이터를 설정하거나 갱신하며, 메인모듈의 Output 컴포넌트를 생성한다. 또한 시뮬레이션수행을 시행하고 종료하는 제어역할과 Output 컴포넌트에 결과를 수집하고, Console에 이벤트나 시뮬레이션 수행정보를 출력하도록 제어하는 기능을 수행한다.

▪ 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.

1.3 라우팅 프로토콜과 호 처리 컴포넌트

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. [그림 2]는 스위치와 LGN의 구성을 보여준다.

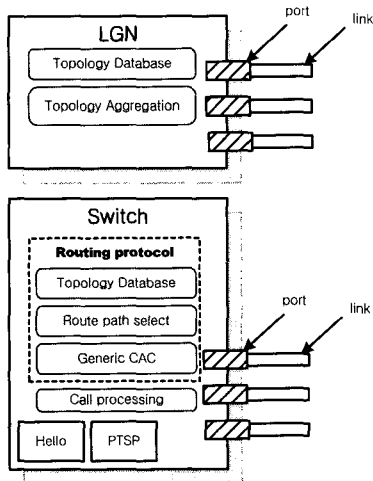


그림 2. LGN과 스위치컴포넌트 블록도

이렇게 라우팅 프로토콜과 호 처리를 독립적인 컴포넌트로 구현함으로써 스위치나 LGN 컴포넌트와 무관하게 프로토콜이나 토폴로지 요약 알고리즘을 적용할 수 있고, 새로운 프로토콜이 요구되면 각각의 프로토콜 컴포넌트를 약간의 수정으로 바로 적용할 수 있는 장점이 있다.

▪ 라우팅 프로토콜 컴포넌트

라우팅 프로토콜 컴포넌트는 크게 두가지 컴포넌트로 구성된다. 토폴로지 데이터베이스와 RoutePath-Select 알고리즘으로써 토폴로지 데이터베이스는 네트워크의 스위치와 링크의 상태정보가 저장되어 있고, 토폴로지 데이터베이스를 이용하여 RoutePathSelect는 요구 QoS를 만족하는 라우팅 경로를 계산한다. RoutePathSelect는 Dijkstra 알고리즘이나 제안된 라우팅 알고리즘이 구현되어 있다. 또한 PTSP 플러딩과 Hello 프로토콜이 포함한다. LGN 컴포넌트의 라우팅 프로토콜은 TA 기능을 수행한다.

▪ 호처리 컴포넌트

호의 생성, 연결, 해제를 수행하는 컴포넌트로 DTL 프로세싱기능을 포함한다. Source ES로부터 호 요청이 들어오면 source 스위치는 라우팅 경로를 계산하고, DTL을 생성하고 전달한다. 중간 스위치에서 라우팅을 수행하여 호가 reject될 경우 crank-back을 border 스위치에서 수행한다.

1.4 구현된 네트워크 토폴로지 생성기

[그림 3]은 구현된 시뮬레이터를 나타내며 본 시뮬레이터는 네트워크 토폴로지 생성을 위해 GUI방식의 토폴로지 생성기로 사용자가 마우스로 그래픽하게 네트워크를 구성하였다. PNNI 네트워크는 계층이 커질수록 네트워크의 노드와 링크가 많이 필요하므로, 거대한 네트워크를 구성하기 용이하게 자동생성 프로그램을 작성했다.

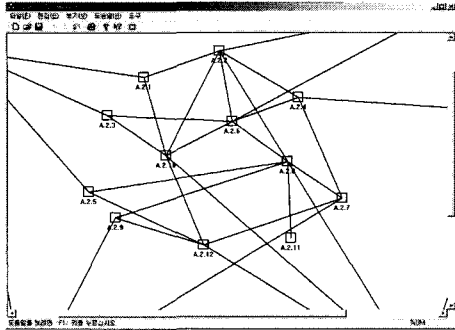


그림 3. 구현된 PNNI 네트워크 토폴로지 시뮬레이터 화면

### III. 시뮬레이터를 이용한 TA 알고리즘의 성능 분석

#### 1.1 시뮬레이션 환경

시뮬레이션은 기존 TA 알고리즘과 라우팅 알고리즘에 따른 성능비교를 본 논문에서 구현한 시뮬레이터를 이용하여 수행하였다. 시뮬레이션에 적용된 알고리즘은 full mesh, MST, star, simple node 방식을 각각 TA 방식에 따라 시뮬레이션 하였다. 시뮬레이션 망은 여러 개의 PG와 PG를 구성하는 노드들로 구성되었으며, TA는 simple node 방식과 복합구성을 적용할 때와 라우팅 시 경로 선택에서 링크의 지연시간 또는 대역폭을 기준으로 수행할 경우에 따라 시뮬레이션을 수행하여 call setup 성공률, 크랭크 백 횟수를 비교하였다.

그리고 시뮬레이션을 수행할 네트워크 모델은 3계층의 토폴로지 요약이 가능한 네트워크로서 최상위 PG가 3개이고, 각 PG의 노드 수는 50개이며, 각각의 LGN의 주소가 "A", "B", "C"고, 최하위 물리노드 주소는 "A.X.X", "B.X.X", "C.X.X"이며, 150개의 물리노드와 760의 링크로 이루어 졌다. 네트워크 토폴로지는 많은 노드와 그 이상의 링크로 구성되므로 토폴로지 자동 생성 프로그램을 이용하여 구성하였다.

#### 1.2 시뮬레이션 환경변수

[표 1]은 시뮬레이션 환경변수를 보여준다. 네트워크

부하는 각 ES에 평균 호 발생율을 설정하여, 평균 호 발생율에 따른 포이송 프로세스로 호가 생성된다. 호 설정이 성공하면 호는 지수분포에 따른 호 지속시간동안 호 연결이 지속된다.

PTSE 갱신시간에 따라 각 노드의 토폴로지 데이터베이스의 정확도가 결정된다고 볼 수 있다. PTSE 갱신시간이 짧을수록 데이터베이스정보의 정확도가 높으나 PTSP 교환 및 flooding 에 따른 노드와 망의 부하가 높아진다. 본 시뮬레이션에서는 PTSE 갱신시간을 300초에서 900초 사이로 설정하면서 시뮬레이션을 수행하였다.

각 노드는 패킷 처리 시 처리하기 위한 노드에서 프로세싱시간이 패킷이나 프로세싱 종류에 따라 다르다. Hello 패킷처리는 프로세싱시간이 짧으나 PTSP 패킷의 생성이나 라우팅과 같은 복잡한 종류의 프로세싱은 처리시간이 길다. 따라서 사설망 인터페이스 시뮬레이션시 각각 프로세싱에 따른 처리시간을 차등을 두어 결정하였다. 패킷은 Hello packet, PTSP, Database summary packet, PTSE REQUEST packet, DTL packet 등에 따라 프로세싱시간과 노드에서 CAC나 라우팅시 프로세싱 시간이 노드에 결정되어 있다. [표 1]에서 네트워크의 부하는 각 ES에 평균 호 발생율을 설정하여, 평균 호 발생율에 따른 포이송 프로세스로 호가 생성된다. 호 설정이 성공하면 호는 지수분포에 따른 호 지속 시간 동안 호 연결이 지속된다.

표 1. 시뮬레이션 파라미터

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 crankback 횟수	5회
Link bandwidth	155Mbps
Link delay	0.0001 ~ 0.0007 (sec)
Simulation time	5000 (sec)

표 2. Processing time

Processing Job	time (msec)
Hello packet	1
Database summary packet	3
PTSE Request packet	5
PTSP packet	5
CAC	3
Routing Path Select	5
DTL processing	3
ACK, NAK, Release	1

### 1.3 TA 알고리즘 및 성능분석

TA 알고리즘의 성능분석 전에 성능분석 하고자 하는 TA 알고리즘에 대해 간략히 기술하면 다음과 같다.

Full-mesh 기법은 논리 링크 상태 QoS를 결정하는 방법으로 LGN 내부 링크 상태 QoS 파라미터는 다른 LGN에서 TA LGN을 통과하여 다른 LGN으로의 연결 시 제공가능한 QoS 파라미터를 의미한다. 즉 full mesh 기법은 PG 내부 토폴로지를 기본으로 하여 임의의 경계노드에서 다른 경계노드로의 제공 가능한 QoS 파라미터를 구하는 것이다. Simple node 기법은 PNNI 라우팅 프로토콜의 오버헤드가 적고 구현의 복잡정도가 가장 간단하기 때문에 많이 이용되는 기법중의 하나로 이 방식에서는  $2 \times n \times (n-1)$ 개의 내부 링크 상태의 정보를 대표값으로 요약하므로 어떤 값을 사용하느냐에 대한 선택이 쉽지 않은 특징이 있다[3].

MST 방식은 spanning tree가 연결 그래프 G가 n개의 노드와 e개의 링크를 가진 경우 n개의 정점과 n-1개의 간선으로 구성되는 특징이 있다. 모든 노드들은 n-1개의 간선들을 결합하여 서로 연결될 수 있으며 각각의 간선 즉 링크에는 가중치가 할당될 수 있다. spanning tree를 구성하기 위해서는 full mesh의 구성이 필요하며 이러한 full mesh 대역폭을 최대로 할 것인지 지연을 최소로 할 것인지에 따라 그 구현 방법이 달라진다 [4][6].

Star 기법은 복합구조와 함께 제안된 것으로 구조가 간단하고 변하지 않는 장점이 있으며 논리 링크 상태 파라미터에 따라 차이는 있지만 full mesh 요약 기법에 비해 매우 큰 압축효과가 있다. Symetric 스타구조는 포트사이의 QoS 파라미터 값이 비슷한 대칭구조에 적

합한 구조이며 bidirectional asymmetric star 구조는 정보량은 다른 star 구조에 비해 많으나 full mesh 구조와 거의 같은 수준으로 요약할 수 있다는 특징이 있다 [3][7].

비교 대상 TA 알고리즘은 기존의 full mesh, MST, star, simple node 방식을 적용하여, 각 알고리즘과 라우팅 시 crank-back을 적용하거나 사용하지 않는 경우로 나누어 수행하였다. 또한 spanning tree나 star의 경우 full mesh로부터 변환시키기 때문에 full mesh가 어떻게 구성되는가에 따라 토폴로지 정보의 정확성이 다르게 나올 수 있다. Full mesh 경우 링크의 대역폭, 지연으로 계산하여 구현할 경우에 따라 수행하여 호 성공률, crank-back 비율의 항목에 대해 비교 평가하였다. Full mesh, MST, star, simple node 방식을 full mesh를 링크지연시간을 기준으로 구성 할 경우, 링크의 대역폭을 기준으로 구성할 경우에 따라 성능 평가를 수행하였다. 즉, full mesh 구조의 논리 링크 상태 정보 파라미터는 다중 경로들 중에서 가장 적은 지연 파라미터를 갖는 경로의 대역폭과 지연 파라미터 값이 된다. 이를 기존의 MST, star, 단순 노드 방식의 구조가 만들어지게 된다. 시뮬레이션 환경은 PTSE refresh time은 900sec로 설정하였고, crank-back을 적용하였다. 입력 트래픽은 균일하며, 상위계층 그룹 A의 호는 그룹 C, 그룹 B의 호는 그룹 A와 C, 그룹 C의 호는 그룹 A로 발생시켰다. PTSE refresh time은 발생 호의 평균 지속 시간의 약 3배에 해당한다. [그림 4][그림 5]는 호 성공률 및 성공한 호에서 크랭크 백 비율의 결과 그래프이다.

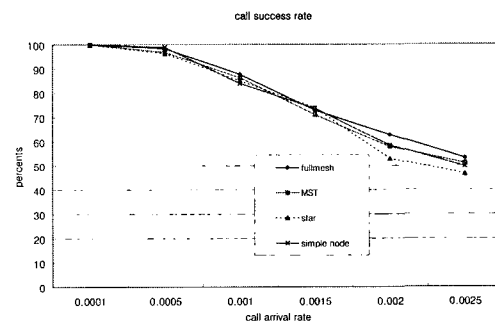


그림 4. 기존 알고리즘 별 호 성공률 (sort by delay 인 경우)

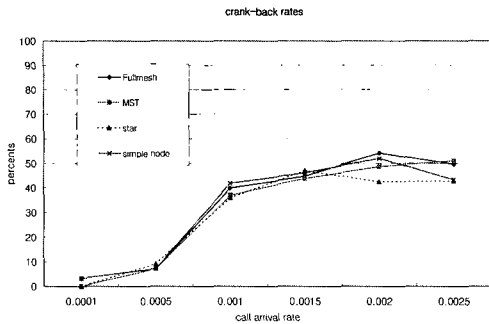


그림 5. 기존 알고리즘 별 crank-back rates (sort by delay 인 경우)

Full mesh가 가장 좋은 성공률을 보이고 있으며, 단순 노드 방식의 성공률도 다른 기법과 비해 낮은 성공률을 보이고 있지만, 가장 작은 정보량을 가지고 있으므로 라우팅 프로토콜의 오버헤드는 가장 적다. 입력 트래픽이 균일한 상황에서는 단순 노드 방식의 TA 기법도 좋은 결과를 얻을 수 있다고 볼 수 있다. Spanning tree나 star 기법은 full mesh에 비해 정보량은 적지만, full mesh로의 변환 결과는 full mesh와 큰 차이를 보이지 않는다.

물론 포트 수에 영향을 받기는 하지만 대체로 full mesh와 근사한 결과를 보이고 있다. 따라서 입력 트래픽이 불균형인 상황에서도 full mesh와 거의 같은 성능을 보일것으로 예측된다.

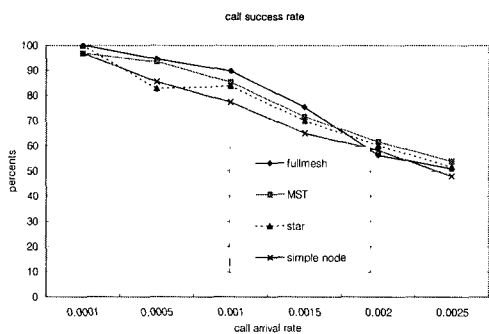


그림 6. 기존 알고리즘 별 호 성공률 (sort by bandwidth 인 경우)

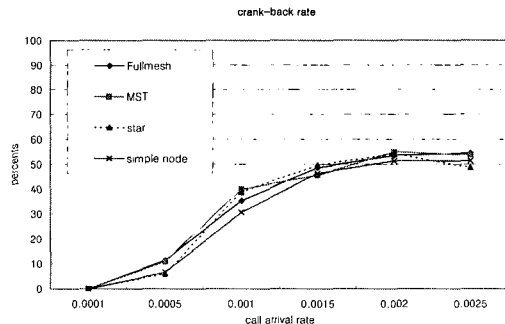


그림 7. 기존 알고리즘 별 crank-back rates (sort by bandwidth 인 경우)

[그림 6][그림 7]은 full mesh 구조의 논리 링크 상태 정보 파라미터는 다중 경로들 중에서 가장 큰 대역폭 파라미터를 갖는 경로의 대역폭과 지연 파라미터 값을 가지고 구성하고, MST, star, 단순 노드 방식의 구조를 완성한 sort by bandwidth 방식의 결과들이다. 호 성공률에 있어서는 sort by delay보다는 다소 적은 성공률을 보이고 있는데, 이는 그룹 A는 그룹 C로의 경로를 설정하기 때문에 많은 그룹을 지나게 되고, 각 그룹의 지연 파라미터가 대역폭이 가장 큰 경로를 기준으로 하기 때문에 sort by delay 보다는 큰 지연 파라미터를 가지고 있게 되고, 이로 인해서 요청 지연시간이 작은 호의 차단율이 증가해 sort by delay 방식보다 성공률이 낮게 나타난다.

[그림 8]의 그래프는 crank-back이 없는 경우와 있는 경우의 결과 그래프이다. 또한 PTSE refresh time이 작은 것과 큰 것의 결과 그래프들이다. Crank-back이 없는 경우 상당히 많은 차이를 보이고 있는데, 이는 crank-back이 사설망 인터페이스 라우팅 프로토콜에서는 아주 중요한 변수가 된다는 것을 알 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 있다.

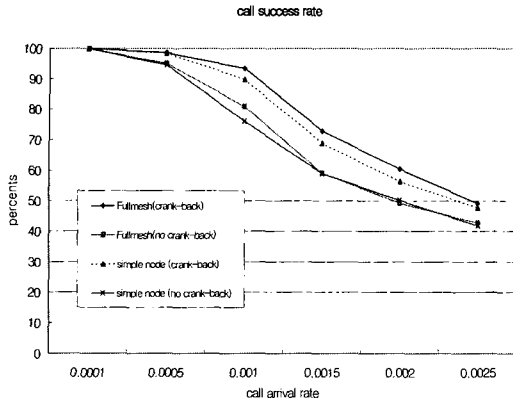


그림 8. Full mesh 와 simple node의 호 성공율

#### IV. 결론

본 논문에서는 사설망인터페이스에서의 TA 알고리즘을 쉽고 효율적으로 시뮬레이션 하기 위한 시뮬레이터를 설계하고 구현하였다. 그리고 구현된 시뮬레이터를 이용하여 기존 TA 기법인 full mesh, MST, star, simple node를 각 성능 요소별로 분석하였다. 구현된 사설망 인터페이스 라우팅 시뮬레이터는 토폴로지 요약 알고리즘 개발에 적용되어 유용하게 사용될 수 있을 것으로 사료된다. 앞으로의 연구과제는 대규모 망에서도 적용할 수 있는 가능한 한 정확하고 작은 정보량으로 보다 함축적인 논리 노드로 표현할 수 있는 TA 알고리즘의 연구와 라우팅 알고리즘의 연구가 필요하다.

#### 참고 문헌

- [1] Private Network-Network Interface Specification Version 1.0(PNNI 1.0), ATM Forum af-pnni-0055.000, Mar. 1996.
- [2] A. R. Ragozini, et al, "Analysis of the Performance of a Hierarchical PNNI Network," Proceedings of the 1999 2nd International Conference on ATM, 1999.
- [3] W. Lee, "Topology Aggregation for Hierarchical Routing in ATM Networks," Computer Communication Review, Apr. 1995.
- [4] E. Basturk and P. Stirpe, "A Hybrid Spanning Tree Algorithm for Efficient Topology Distribution in PNNI," Proceedings of the 1998 1st IEEE International Conference on ATM, 1998.
- [5] T. Korkmaz and M. Krunk, "Source-oriented topology aggregation with multiple QoS parameters in hierarchical ATM networks," Proceeding of the IEEE infocom'99, pp.137-146, 1999.
- [6] W. C. Lee, "Spanning Tree Method for Link State Aggregation in Large Communication Network," Proceedings of the IEEE infocom '95, Vol.1, 1995.
- [7] B. Awerbuch, Y. Du, and Y. Shavitt, "The effect of network hierarchy structure on performance of ATM PNNI hierarchical routing," Computer Communications, Vol.23, Issue10, pp.980-986, May 2000.
- [8] C. K. Tham, J. Mai, and L. W. C. Wong, "A QoS-based routing algorithm for PNNI ATM networks," Computer Communications, Vol.25, Issue7, pp.714-729, May 2002.
- [9] B. J. Chang and R. H. Hwang, "Distributed cost-based update policies for QoS routing on hierarchical networks," Information Sciences, Vol.159, Issues1-2, pp.87-108, Jan. 2004.



저 자 소 개

김 남 희(Nam-Hee Kim)

종신회원



- 1992년 2월 : 군산대학교 정보통신공학과 (공학사)
- 1994년 2월 : 전북대학교 전자공학과 (공학석사)
- 1997년 8월 : 전북대학교 전자공학과(공학박사)

• 2002년 ~ 현재 : 군산대학 컴퓨터정보과학과 조교수

<관심분야> : 컴퓨터네트워크, 센서망, 트래픽제어

김 변 곤(Byun-Gon Kim)

정회원



- 1990년 2월 : 한국항공대학교 항공전자공학과 (공학사)
- 1997년 2월 : 전북대학교 전자공학과 (공학석사)
- 2001년 8월 : 전북대학교 전자공학과(공학박사)

• 2005년 4월 ~ 현재 : 군산대 전자정보공학부 전임강사

<관심분야> : 초고속통신망, 광버스트, 트래픽제어

서 해 영(Hae-Young Seo)

정회원



- 1997년 2월 : 전북대학교 수학교육학과 (이학사)
- 2005년 2월 : 전북대학교 수학교육학과 (이학석사)
- 2005년 2월 ~ 현재 : 군산대학교 컴퓨터정보과학과 박사과정

<관심분야> : 컴퓨터네트워크, 큐잉이론

박 기 홍(Ki-Hong Park)

정회원



- 1982년 2월 : 숭실대학교 전자계산학과 (공학사)
- 1986년 2월 : 숭실대학교 전자계산학과 (공학석사)
- 1995년 3월 : 일본 도쿠시마 시스템공학과(공학박사)

• 1987년 ~ 현재 : 군산대학교 컴퓨터정보과학과 교수  
<관심분야> : 정보검색, 무선통신