

TINA 連結管理를 위한 federation의 모델링 및 性能分析

正會員 朴 明 桓*, 許 琪 中**, 曹 圭 燮***

The Modeling and Performance Analysis of Federation for TINA Connection Management

Myeong Hwan Park*, Ki Jung Huh**, Kyu Seob Cho*** *Regular Members*

요 약

TINA의 연결관리는 분산된 컴퓨팅 환경에서 단대단 연결의 설정, 해제와 변경을 위한 기본적인 관리 기능이다. 그리고 TINA의 federation은 연결 관리 수행 시 이종의 서브네트워크 연결을 위한 개념을 제시하고 있으며 이를 위해 TINA는 federation 상호 동작 모델로서 joint 모델과 cascaded 모델을 제시하고 있다. 본 연구는 federation의 상호 동작을 모델링하고 이를 기반으로 federation의 수행 절차를 정의한다. 정의된 federation 수행 시 발생하는 순차성 에러를 제거하기 위한 running-wait 알고리즘과 federation 수행의 시간 지연 특성과 전체 연결의 신뢰성을 향상시키기 위한 asynchronous-with-broadcasting 방안을 제안한다. 제안한 알고리즘 및 방안이 적용된 federation 모델들을 OPNET으로 시뮬레이션을 수행하고 실험한 결과를 토대로 이들의 성능을 비교 분석한다.

ABSTRACT

The TINA's connection management that supervises the establishment, release and modification of end-to-end connection is essential management function to manage and control of network at distributed computing environment. And the TINAs federation that was defined conception for connection of two kinds' subnetwork. TINA presented two basic federation models that are joint model and cascaded model. In this paper interaction of federation is modeled and at the base of its modeling procedure of federation is defined. These papers propose the running-wait algorithm in order to remove sequence error and asynchronous-with-broadcasting method in order to rise belief of overall connection and delay. The performance of the algorithm and method is analyzed by simulation with OPNET.

*데이콤종합연구소 망관리연구팀
**인덕전문대학교 정보통신과
***성균관대학교 전기전자컴퓨터공학부
論文番號:97272-0806
接受日字:1997年 8月 6日

I. 서 론

통신망의 진화와 통신 서비스의 고도화로 인한 통신망의 품질 향상과 안정된 망 운용이 정보화 사회의 필수 요소로 인식되고 있는 추세이다. 따라서 통신망 장비의 운용 및 관리에 있어서 많은 변화, 즉 통신망 관리가 통신망 구축 이후의 사후 개념이 아니라 통신망 구축 및 서비스 계획과 더불어 망 구축 초기 단계로부터 고려되어야 할 요소가 된다. 이러한 개념의 일환으로 IEEE(Institution of Electrical and Electronic Engineers)에서 CNOM(Committee on Network Operation and Management)을 결성하여 이에 대한 활발한 연구를 진행하고 있다. ITU(International Telecommunication Union)와 ISO(International Standards Organization)와 같은 국제 표준화 기구와 ATM(Asynchronous Transfer Mode) Forum이나 NM(Network Management) Forum과 같은 산업 표준화 기구에서 망 관리 표준화를 적극 전개하고 있다[1]. 이 결과로 ITU-T에서 전기 통신망의 운용 관리를 위한 기본적인 구조로서 TMN(Telecommunication Management Network) 개념을 정립하여 국제 표준으로 제시하고 있으며 현재 이에 대한 연구가 지속적으로 진행되고 있다[2][3][4].

TMN은 통신망 운용 관리의 효율성 향상과 통신망의 경제적 운용을 위하여 통신망 장비의 운용 및 관리 집중화, 자동화를 목표로 하고 있다. 이를 위해 통신망 장비의 상태를 원격으로 감시 및 제어를 수행할 수 있는 운용 시스템(OS: Operations System) 혹은 망 관리 시스템(NMS: Network Management System)을 네트워크에 적용한 구조를 제시하고 있다. 그리고 운용 시스템과 통신망 장비인 망 요소(NE: Network Element)간의 상호 작용을 규정한다. 이들간의 동작은 관리자(manager), 관리 대행자(agent)와 통신망 구성 요소의 객체 지향적 파러다임인 관리 객체(MO: Managed Object)로 모델링 된다.

한편, ATM과 SDH(Synchronous Digital Hierarchy)에 의한 통신망의 광대역화에 부응하여 세계 각국의 통신망 운용자와 장비 제조사 및 소프트웨어 관련 단체가 TINA-C(Telecommunication and Information Network Architecture-Consortium)를 결성하여 미래 통신망 구조의 표준화를 진행하고 있다. TINA의 기본 개념은 통신 서비스의 광대역화와 다양화에 따른

유연성을 제공하기 위하여 통신망 구조를 객체 지향적 구조로 정립하는 것이다[5][6][7]. 그리고 네트워크를 관리하는 소프트웨어적 측면의 기능들을 객체 지향 구조를 모델링하기 위해 TINA의 실현 개념으로 ODP(Open Distributed Processing)개념을 적용하고 있다[8][9][10]. 앞으로의 모든 통신망 운용 관리는 TMN 개념을 기본으로 하며 거시적으로는 TINA 구조를 지향하는 망 관리 구조로 발전될 것이다.

TINA의 관리 영역은 성능 관리, 장애 관리, 보안 관리, 계정 관리, 연결 관리, 자원 관리로 구분된다. 특히 연결 관리는 단대단 연결의 설정, 해제, 변경을 위한 작업을 주관하기 때문에 분산 컴퓨팅 환경의 망의 관리 및 제어를 위한 필수적인 관리 기능이다. 그리고 TINA의 federation은 연결 관리 수행 시 이종의 서브네트워크 연결을 위한 개념을 제시하고 있으며 이를 위해 TINA는 federation 상호 동작 모델로서 joint 모델과 cascaded 모델을 제시하고 있다[11]. 그러나 제시된 cascaded 모델의 경우 인접한 노드에 의한 경로 설정에 있어 전체 연결에 대한 신뢰성 문제가 발생되며, joint 모델의 경우 보다 구체적인 연구가 현재 진행 중에 있다.

본 논문은 이를 해결하기 위해 우선적으로 federation의 상호 동작을 모델링하고 이를 기반으로 federation의 수행 절차를 정의한다. 그리고 정의된 federation 수행 시 발생하는 순차성 에러를 제거하기 위한 running-wait 알고리즘과 federation 수행의 시간 지연 특성과 전체 연결의 신뢰성을 향상시키기 위한 asynchronous-with-broadcasting 방안을 제안한다. 제안한 알고리즘 및 방안이 적용된 federation 모델들을 OPNET으로 시뮬레이션을 수행하고 실험한 결과를 토대로 이들의 성능을 비교 분석한다.

본 논문의 구성은 II장에서 TINA의 연결 관리를 위한 연산 관점 그리고 federation 상호 동작 모델의 특성을 기술하고, III장에서 federation의 개념적 모델 및 수행 절차와 문제점을 정의한다. IV장에서 제시한 문제를 해결하기 위한 새로운 방안 및 알고리즘을 제안하며, IV장에서 제안한 방안 및 알고리즘의 성능 분석을 위한 시뮬레이션 모델을 설정하고 이를 바탕으로 수행된 시뮬레이션 결과를 분석하고, V장에서 결론과 향후 연구로 끝맺는다.

II. TINA의 연결 관리

1992년에 각국의 통신망 운용자와 장비 제조자 및 소프트웨어 관련 단체들이 결성한 TINA의 정보 네트워크(information network)는 시간, 장소, 양, 형태와 관계없이 망 사용자에게 요구되는 접근과 관리 능력을 보다 효율적으로 제공하기 위한 것이다. 따라서 TINA는 이 정보 네트워크를 기반으로 통신망 구조의 변화 없이 다양한 형태의 유연성있는 신규 서비스 도입을 부과하고 정보와 통신을 융합하기 위한 통합 구조 모델로서 객체 지향적 통신망 구조 개념을 제시하고 있다[5][6]. 이를 위해 TINA는 ATM과 SDH 기반의 통신망 구조에 대한 제어, 관리, 서비스 기능들을 객체 지향적 모델의 구조로 구성하기 위한 연구를 진행하고 있다. TINA의 망 자원 정보 모델은 그림 1과 같이 기존 권고 사항인 기존의 OSI(Open System Interconnection) 시스템 관리의 기능, SDH 전송망의 기능적 개념 구조, M.3100에 기술된 자원 관리 측면과, ATM과 SDH와 같은 현존하는 망 요소 정보 모델을 적용하였다[12][13][14].

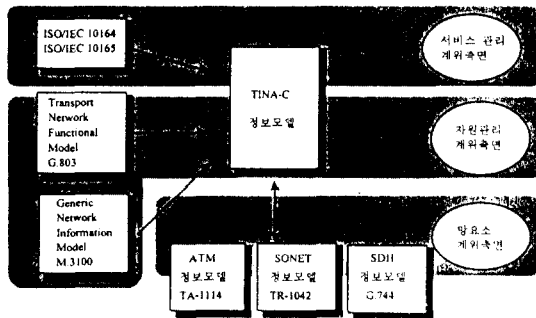


그림 1. TINA의 구성 형태

TINA의 정보 모델과 TMN 기능적 계층을 비교하면 TINA의 서비스 관리 계위 및 자원 관리 계위는 TMN의 서비스 관리 계층 및 네트워크 관리 계층(NML: Network Management Layer)과 거의 유사하다. 특히 TMN의 네트워크 관리 계층과 망 요소 관리 계층(NEL: Network Element Layer)은 TINA의 자원 관리 계위와 긴밀한 관계가 있다. 이는 TINA의 관리

구조는 망 요소 관리 계층 측면의 관리뿐만 아니라 네트워크 관리 계층 측면의 관리를 제공함으로써 분산 환경의 관리 기능을 수행함을 의미한다.

TINA의 개념적 구조는 컴퓨팅 구조, 관리 구조, 서비스 구조, 네트워크 구조로 분류되며 특히 컴퓨팅 구조는 분산 환경에서 통신망 제어를 위해서 분산 처리 환경(DPE: Distributed Processing Environment)상의 연산 객체(CO: Computational Object) 상호 동작의 규칙을 제공한다. 이를 위해 적용된 ITU-T RM-ODP (Reference Model for Open Distributed Processing)는 사업자 측면의 서비스를 정의한 사업자(Enterprise) 관점, 서비스 또는 서비스 요소가 필요로 하는 정보를 정의한 정보(Information) 관점, 서비스 제공을 위한 연산 객체간의 상호 동작을 정의한 연산(Computational) 관점, 분산 처리 환경의 모델을 정의한 엔지니어링(Engineering) 관점과, 기술적인 선택에 관한 사항을 정의한 기술(Technology) 관점으로 분류된다[10]. 특히 연산 관점은 분산 처리 환경에서의 분산 어플리케이션으로 구성된 연산 객체의 상호 동작을 객체 통신(object communication)으로 정의하고, 엔지니어링 관점은 분산 처리 환경의 하부 구조를 정의함으로써 연산 객체의 위치 정보와 상호 동작 메커니즘을 제공한다.

분산 컴퓨팅 환경의 망 관리와 제어에 필수적인 관리 기능을 제공하는 TINA의 연결 관리 기능은 연결의 종단 지점의 위치와 망 자원의 제어를 포함한 망 연결의 생성, 변경 및 해제 기능과, 연결에 사용되는 자원의 식별과 망 자원과 경로의 선택에 요구되는 정보 관리 기능 그리고 망 운용자와 사용자가 이용하는 연결 관리의 제어와 모니터 기능을 제공한다[11].

1. 연결 관리를 위한 정보(Information) 관점

연결 관리를 위한 정보 관점은 ODP 정보 관점을 바탕으로 연결 관리에 필요한 망 자원의 추상적 개념을 기술한다. 이는 다음과 같은 두 가지 형태의 관리 객체로 분류된다. 하나는 연결이 수행되기 전 또는 연결 후 남아 있는 정적 관리 객체이고 다른 하나는 연결이 설정된 후 생성되는 연결성 관리 객체이다.[15]

ITU-T G.803의 계층화(layering)개념에 따라 통신망은 전달되는 특성 정보에 의해 상호 연결되고 호환성 있는 입출력의 모임인 계층 네트워크(layer network)

의 구성된다[16]. 그리고 분할화(partitioning) 개념에 따라 각 서브네트워크는 보다 작은 네트워크로 분해되고 링크로서 상호 연결된다. 이를 기반으로 한 TINA의 연결 관리에서 사용되는 정적 관리 객체들은 다음과 같다.

- LNW(Layer Network): 특성 정보 전달을 지원하는 기능들의 집합을 나타내는 관리 객체
- SNW(Subnetwork): 계층 네트워크 내부 구조의 서브네트워크를 정의하는 관리 객체
- TL(Topological Link): 다른 서브네트워크간의 연결을 포함하는 관리 객체
- LTP(Link Termination Point): Topological Link를 위한 종단점과 NWCTP를 포함하는 관리 객체
- CON(Connection): 다른 서브네트워크간의 연결을 나타내는 관리 객체
- NWTTP(Network Trail Termination Point): Trail의 종단점을 나타내는 관리 객체
- NWCTP(Network Connection Termination Point): CON과 SNC의 종단점을 나타내는 관리 객체
- TC(Tandem Connection): CON과 SNC의 포함하는 관리 객체

TINA의 연결 관리에서 정의하는 연결성 관리 객체는 다음과 같다.

- Trail: 계층 네트워크의 종단점간의 특성 정보 전달을 책임지는 관리 객체
- SNC (Subnetwork Connection): 서브네트워크 종단점간의 단방향 또는 양방향의 연결을 정의하는 관리 객체
- Edge: SNC와 NWTPs(network Termination Points)의 연결성을 정의하는 관리 객체

2. 연결 관리를 위한 연산(Computational) 관점

TMN은 실질적으로 망 요소에 관한 관리 정보를 망 요소 관리 계층 측면에서 정의하고 있는데 반해, TINA는 ODP의 연산 관점을 기반으로 분산 컴퓨팅 환경의 연결 제어를 위한 연산 객체를 네트워크 관리 계층 측면에서 정의한다. TINA의 연결 관리를 위해 정의된 연산 객체들의 종류 및 특징을 살펴보면 다음

과 같다[11].

- CC(Connection Coordinator)
 - LNC에게 trail 생성 요구
- LNC(Layer Network Coordinator)
 - 계층 네트워크의 trail을 제공
 - 다른 영역과의 federation을 제공
 - 자기 영역 내의 CP에게 서브네트워크 연결을 요구
- CP(Connection Performer: EML, NML, Federation)
 - 자기 영역 내의 서브네트워크 연결을 제공
 - 하부의 CP에게 보다 작은 서브네트워크 연결을 요구
- RCM(Resource Configuration Manager)
 - 연결 관리 패키지의 구성
 - CP와 LNC에게 네트워크 위상 정보를 제공
 - CP가 NWTP를 선택할 때 자료 제공

이와 같은 연산 객체들을 이용한 연결 관리의 수행은 다음과 같다. 우선 계층화된 서브네트워크의 연결 요구를 수행하기 위하여 각각의 서브네트워크를 관장하는 연결 관리 패키지가 구성된다. 연결 관리 패키지는 이종의 서브네트워크 연결을 책임지는 LNC와 서브네트워크 연결을 책임지는 CP로 구성된다. 하나의 서브네트워크는 분할화 개념에 의해 보다 작은 서브네트워크로 나누어지며 분할된 서브네트워크의 연결 관리를 관장하는 여러 CP들이 구성된다.

관리 영역별로 수행되는 연산 객체로 연결 관리 패키지를 구성하면 그림 2와 같다. 그림 2에서 서브네트워크 SNW11은 보다 작은 서브네트워크 SNW12와 SNW13으로 구성되고 서브네트워크 SNW21은 보다 작은 서브네트워크 SNW22와 SNW23으로 구성된다. 이와 같이 구성된 두개의 서브네트워크(SNW11과 SNW21)를 관리하기 위해서 연결 관리 패키지 1과 연결 관리 패키지 2가 구성된다. 서브네트워크 SNW 11과 서브네트워크 SNW 21간의 탠덤 연결(tandem connection)의 수행을 위한 연결 관리 패키지의 LNC 1과 LNC 2가 구성된다. 그리고 분할된 보다 작은 서브네트워크의 연결 관리를 수행하기 위한 CP 11과 CP 21이 구성되며 이들은 각각 SNW 12와 SNW 13, SNW 22와 SNW 23의 연결 관리를 관장한다. 가장 최하위의 서브네트워크 연결은 CP 12, CP 13, CP 22, CP 23

에 의해 자신이 관리 영역인 SNW 12, SNW 13과 SNW 22, SNW 23 내에서 수행된다.

CP에서 수행되는 서브네트워크 연결의 설정 단계는 3단계로 수행된다. 1단계는 서브네트워크 연결에 해당하는 객체를 생성한 후, 소스 네트워크 주소(source network address)에 해당하는 edge 관리 객체를 생성하여 서브네트워크 연결 관리 객체에 첨부한다. 그리고 생성된 edge에 소스 주소(source address)에 해당하는 NWCTP(Network Connection Termination Point) 중 사용하지 않는 하나의 NWCTP를 대응시킨다. 2단계는 목적지 네트워크 주소(destination network address)에 해당하는 edge 관리 객체를 생성하고 서브네트워크 연결 관리 객체에 첨부한다. 3단계는 생성된 edge 관리 객체에 목적지 주소(destination address)에 해당하는 NWCTP 중 사용하지 않는 하나의 NWCTP를 대응시킨다.

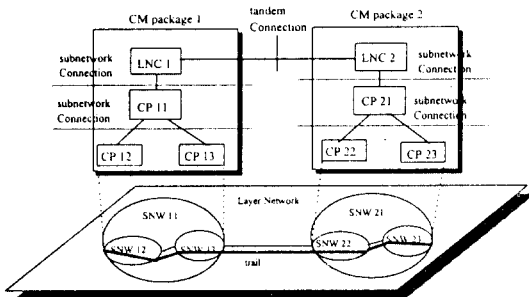


그림 2. 연결 관리를 위한 연산 객체 구조

RCM 연산 객체는 연결 관리 수행 시 네트워크의 구성을 유지하면서 연결 관리 패키지의 요구에 대한 적절한 라우팅 경로 결정을 통해 관리 객체의 균형을 제공한다. RCM 연산 객체의 전체 기능은 아직 완전히 정의되어 있지 않지만 현재까지 정의된 주된 기능은 NWCTP(Network Termination Point Pool)안에 있는 NWCTP의 요구 수신 기능, CP에게 NWCTP로부터 선택한 특정 NWCTP의 송신 기능, 선택된 NWCTP의 상태 전환 기능, CP의 요구 시 NWCTP의 해제 기능 등이다. NWCTPPool은 네트워크 종단점의 모음으로서 라우팅 기능에 영향을 주게 된다.

RCM 연산 객체는 이러한 기능을 연결 관리 패키

지에 제공하기 위하여 다음과 같이 동작된다. LNC에게 getLNWInfo() 동작으로 LNW 식별자와 trail을 묶는데 사용되는 source와 sink NWCTP를 제공하고, getLNWTopology() 동작으로 SNW 식별자, SNW 형태(composite, component)와 LTP를 제공한다. 네트워크 계층의 CP(NML_CP)에게 getSNWInfo() 동작으로 SNW 식별자와 SNC를 묶는데 사용되는 source와 sink NWCTP를 제공하고 getSNWTopology() 동작으로 SNW 식별자, SNW 형태(composite, component)와 LTP를 제공한다. 망 요소 관리 계층의 CP(EML_CP)는 서브네트워크 정보를 알 필요가 없기 때문에 getSNWInfo() 동작만을 사용한다[15].

3. Federation 상호 동작 모델

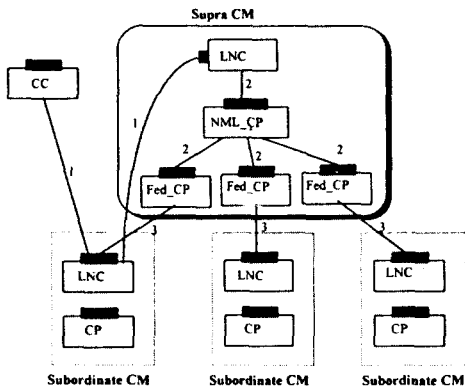
Federation은 둘 이상의 독립적인 조직적 구조간에서 한 구성원이 다른 구성원의 자원을 공유하기 위한 상호 동작 방법의 확장 개념이다. 이는 TINA의 관리 기능에 모두 적용되며 특히 연결 관리에서 서로 다른 관리 영역간의 연결을 위한 관리 정보 교환 및 상호 동작을 위해 사용된다. 이를 위해 TINA는 federation 상호 동작 모델로서 joint 모델과 cascaded 모델을 제시하고 있다.

Joint 모델은 전체 서브네트워크 정보를 처리하는 별도의 supra 연결 관리 패키지에 의해 단대단 연결을 설정한다. Cascaded 모델은 전체 연결을 조정하는 별도의 노드없이 오직 이웃한 연결 관리 패키지의 네트워크 정보만으로 단대단 연결을 설정한다. 그림 3은 TINA의 joint 모델과 cascaded 모델의 상호 동작을 나타낸다.

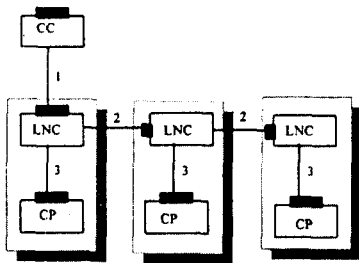
Joint 모델의 상호 동작은 1 단계로서 CC가 하부 연결 관리 패키지의 LNC에게 trail의 생성을 요구하면 LNC는 자신의 trail 생성 요구를 supra 연결 관리 패키지의 LNC에게 전달한다. 2 단계로서 supra 연결 관리 패키지의 LNC는 전체 관리 영역을 자신의 서브네트워크로 간주하기 때문에 하부 연결 관리 패키지에서 요구한 trail 생성에 해당되는 서브네트워크 연결을 자신의 NML-CP에게 전달한다. 그리고 자신의 NML-CP는 생성된 서브네트워크 연결을 탠덤 연결로 전환시키기 위하여 Fed-CP에게 서브네트워크 연결을 전달한다. 3 단계로서 Fed-CP는 NML-CP로부터 수신한 서브네트워크 연결을 탠덤 연결로 전환하

여 해당 하부 연결 관리 패키지에 전달한다.

Cascaded 모델의 상호 동작은 1 단계로서 CC로부터 trail 생성 요구를 수신한 LNC는 자신이 관리하는 인접한 서브네트워크 정보만을 바탕으로 서브네트워크 연결과 탠덤 연결을 결정한다. 2, 3단계로서 만약에 탠덤 연결이 필요한 경우에는 인접한 연결 관리 패키지에 탠덤 연결을 요구하고 서브네트워크 연결이 필요할 경우에는 자신의 CP에게 서브네트워크 연결을 요구한다.



(a) Joint 모델



(b) Cascaded 모델

그림 3. Federation의 상호 동작 모델

III. Federation의 모델링

1. Pseudo LNC

본 연구에서는 cascaded 모델에서 2회 이상의 federation 수행을 위해 초기의 initiateTC를 생성한 LNC

를 master로 정의하며 master LNC로부터 initiateTC를 중계 받은 NLC는 pseudo LNC로 정의한다. Pseudo LNC의 기능은 master LNC로부터 initiateTC를 수신한 후 자신과 인접한 다른 LNC의 탠덤 연결이 요구되면 master LNC를 대신하여 자신의 인접한 서브네트워크 정보에 의해 결정된 인접한 LNC에게 initiateTC를 전달하는 것이다. 이를 위해서 자신에게 initiateTC를 요구한 연산 객체의 식별자인 borrower와 initiateTC 전달하고자 하는 연산 객체의 식별자인 lender를 정의한다.

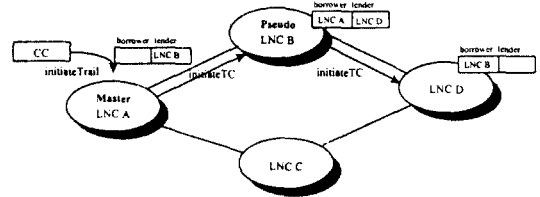


그림 4. Pseudo LNC를 적용한 federation 수행 형태

그림 4는 pseudo LNC를 적용한 federation의 수행 형태를 나타낸다. Master LNC A가 CC로부터 initiateTrail을 수신하면 pseudo LNC B로 initiateTC를 송신한다. 이때 master LNC A는 borrower를 비워두고 최초의 pseudo LNC B를 lender로 설정한다. Master LNC A로부터 initiateTC를 수신한 pseudo LNC B는 borrower를 LNC A로 설정한 후 연결 관리를 수행한다. 만약에 pseudo LNC B에서 인접한 LNC D의 탠덤 연결이 요구되면 lender를 LNC D로 설정한 후 master LNC A를 대신하여 LNC D로 initiateTC를 송신한다. Pseudo LNC B로부터 initiateTC를 수신한 LNC D는 borrower를 LNC B로 설정한 후 자신의 서브네트워크 연결을 수행한다. LNC D에서 서브네트워크 연결이 완료되면 응답 메시지를 자신의 borrower인 pseudo LNC B로 전달하고 이를 수신한 pseudo LNC B는 자신의 borrower인 master LNC A로 서브네트워크 연결의 응답 메시지를 중계한다.

2. Federation의 개념적 모델 및 수행 절차

Federation 수행의 개념적 모델은 그림 5와 같다.

그림 5의 Tr(F1), Tr(F2)는 federation 요청 메시지의 트랜잭션을 나타낸다. Tr(R1), Tr(R2), Tr(R3)는 서브네트워크 연결 요구 메시지의 트랜잭션을 나타낸다. Tr(a1), Tr(a2), Tr(a3)와 서브네트워크 연결 요구에 대한 응답 메시지의 트랜잭션을 나타낸다. 이와 같은 federation의 트랜잭션을 시간 영역으로 표시하면 그림 6과 같다.

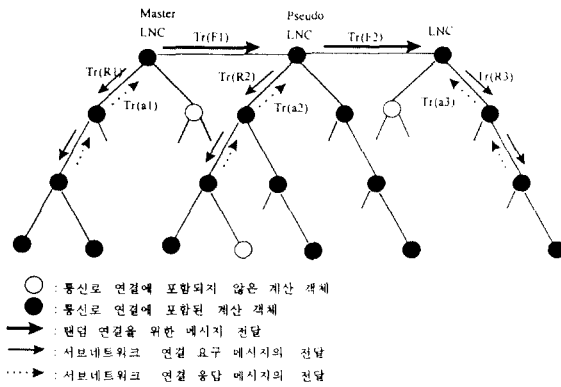
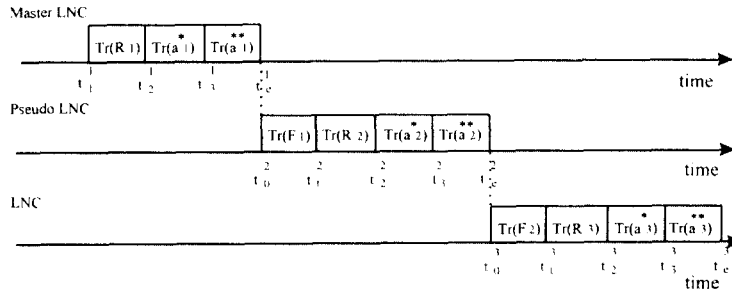


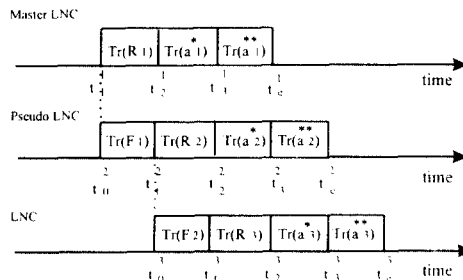
그림 5. Federation 수행의 개념적 모델

그림 6의 Tr(a*)는 서브네트워크 연결 요구에 대한 서브네트워크의 최초 발신단 종단점의 응답 메시지의 트랜잭션을 나타낸다. Tr(a**)는 서브네트워크 연결 요구에 대한 서브네트워크의 최종 착신단 종단점의 응답 메시지의 트랜잭션을 나타낸다. 그리고 t_0 는 master LNC를 제외한 LNC의 federation 요청 수신 시간을, t_1 는 모든 LNC의 자기 서브네트워크에 대한 서브네트워크 연결 요청 생성 시간을, t_2 는 모든 LNC에서 자기 서브네트워크의 최초 발신단 종단점에 대한 서브네트워크 연결 수행 결과의 수신 시간을, t_3 는 모든 LNC에서 자기 서브네트워크의 최종 착신단 종단점에 대한 서브네트워크 연결 수행 결과의 수신 시간을, t_c 는 모든 LNC에서 자기 서브네트워크의 연결 관리의 종료 시간을 나타낸다. 본 연구에서는 이상의 트랜잭션을 적용한 federation 모델의 수행 절차를 그림 6(a)와 같은 동기적 형태와 그림 6(b)와 같은 비동기적 형태로 정의한다.

동기적 형태와 비동기적 형태의 수행 절차를 비교하면 다음과 같다. 동기적 형태의 federation은 RCM



(a) 동기적 형태



(b) 비동기적 형태

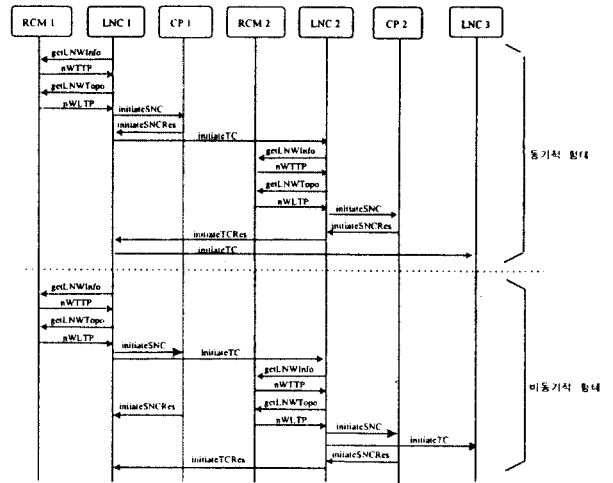
그림 6. 시간 영역의 federation 수행 절차

연산 객체를 통하여 탠덤 연결과 서브네트워크 연결에 필요한 관리 정보를 별도로 확보하기 때문에 federation 발생 시점(t'_0)이 자기 서브네트워크의 연결 관리의 종료 시간(t')과 같은 시점에서 수행된다($t'_0=t'$). 비동기적 형태의 federation은 RCM 연산 객체를 통하여 탠덤 연결과 서브네트워크 연결에 필요한 관리 정보를 동시에 확보하기 때문에 federation의 발생 시점(t'_0)이 자기 서브네트워크에 대한 서브네트워크 연결 요청의 생성 시간(t'_i)에서 수행된다($t'_0=t'^{-1}$). 이와 같은 동기적, 비동기적 형태의 수행 절차를 적용한 cascaded 모델과 joint 모델의 세부적인 수행 절차를 그림 7과 같다.

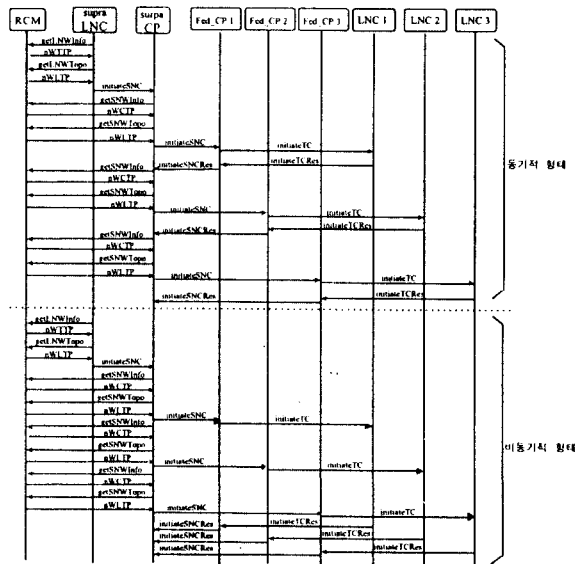
그림 7에서 동기적 형태의 joint 모델과 cascaded 모델은 자신의 RCM을 통하여 trail에 대한 서브네트워크 정보만을 획득한 후 자신의 서브네트워크 연결을 수행한다. 그리고 인접한 서브네트워크의 federation이 요구되면 자신의 RCM을 통하여 trail에 대한 탠덤 정보만을 획득한 후 인접한 pseudo LNC에게 탠덤 연결을 수행한다. 반면에 비동기적 형태의 cascaded 모델은 자신의 RCM을 통하여 trail에 대한 서브네트워크 정보와 탠덤 정보를 동시에 획득한 후 자신의 서브네트워크 연결의 요구와 인접한 pseudo LNC에게 탠덤 연결을 요구한다. 그리고 비동기적 형태의 joint 모델도 supra LNC가 자신의 RCM을 통한 전체 trail에 대한 서브네트워크 정보와 탠덤 정보를 동시에 획득한 후 각 서브네트워크 연결을 자신의 CP를 통하여 하부 연결 관리 패키지로 송신된다.

전체 연결 관리에 대한 응답 메시지의 수집 단계를 살펴보면, 동기적 형태의 joint 모델과 cascaded 모델의 경우 서브네트워크 연결이 완료된 후 탠덤 연결을 수행하기 때문에 서브네트워크 연결의 응답 메시지는 서브네트워크 연결의 요구 순서로 수집된다. 그리고 비동기적 형태의 joint 모델의 경우 각 서브네트워크 연결의 응답 메시지는 하나의 supra LNC에 직접적으로 수집되기 때문에 supra LNC 자체의 처리 능력으로 서브네트워크 연결에 대한 응답 메시지의 수집 순서를 보장한다. 그러나 비동기적 형태의 cascaded 모델의 경우에는 각 연결 관리 패키지의 서브네트워크 연결의 수행 시간이 불일치로 인하여 pseudo LNC를 경유하여 master LNC에서 수집되는 서브네트워크 연결의 응답 메시지의 순서가 서브네트워크 연결

의 요구 순서와 어긋나는 순차성 에러 문제가 발생한다. 따라서 본 연구는 이러한 순차성 에러를 해결하기 위하여 running-wait 알고리즘을 제안한다. 또한 cascaded 모델에서 발생하는 2회 이상의 federation이 발생할 경우 이웃하지 않는 연결 관리 패키지에 대한 관리 정보의 부재로 인하여 발생하는 전체 연결 관리



(a) cascaded 모델



(b) Joint 모델

그림 7. Federation 모델의 수행 절차

의 신뢰성 문제를 해결하기 위해서 asynchronous-with-broadcasting 방안을 제안한다.

IV. 새로운 방안 및 알고리즘의 제안

1. Running-wait 알고리즘

비동기적 형태의 cascaded 모델에서 서브네트워크 연결의 수행 시간 차이로 발생하는 순차성 문제를 해결하기 위한 pseudo LNC의 running-wait 알고리즘은 그림 8과 같이 서브네트워크 연결 결과의 수신 과정과 랜덤 연결 결과의 수신 과정으로 분류된다. Running-wait 알고리즘의 동작 원리는 pseudo LNC에서 lender로부터 수신한 initiateTCRes를 자신의 서브네트워크의 연결 관리 수행 여부에 의해 borrower에게 전달할 것인지를 결정하는 것이다. 즉 lender의 initiateTCRes를 필터링함으로써 master LNC에서 발생하는 순차성 에러를 해결하는 것이다.

Running-wait 알고리즘의 수행을 시간 영역으로 표시하면 그림 9와 같다. 그림 9에서 D_i 는 i 번째 서브네트워크의 총 수행 시간을, D_R 는 running-wait 알고리즘에 의한 지연 시간을 나타낸다. t_i^1 는 i 번째 서브네트워크의 서브네트워크 연결 시작 시간을, t_i^2 는 i 번째 서브네트워크의 서브네트워크 연결 종료 시간을, t_i^* 는 i 번째 서브네트워크의 running-wait 알고리즘에 의해 지연된 서브네트워크 연결 종료 시간을 나타낸다.

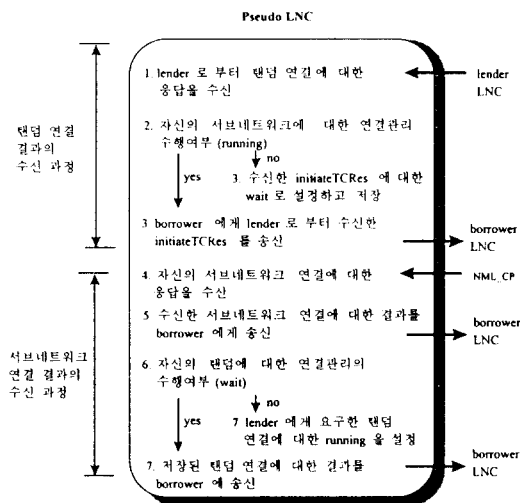


그림 8. Running-wait 알고리즘

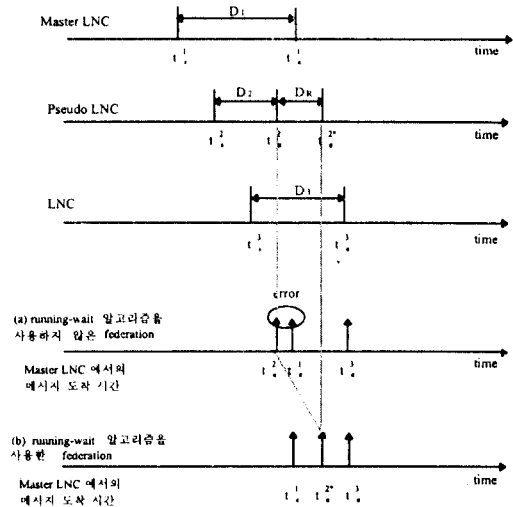


그림 9. 시간 영역의 Running-wait 알고리즘

그림 9 (a)와 같이 running-wait 알고리즘을 사용하지 않을 경우 pseudo LNC의 서브네트워크 연결 종료 시간인 t_i^2 가 master LNC의 서브네트워크 연결 종료 시간인 t_i^1 보다 앞서 발생함으로써 master LNC에서 연결 관리 수집 결과의 순차성 에러가 발생한다. 이에 반해 그림 9 (b)와 같이 running-wait 알고리즘을 사용할 경우 pseudo LNC의 서브네트워크 연결 종료 시간인 t_i^2 가 master LNC의 서브네트워크 연결 종료 시간인 t_i^1 보다 running-wait 알고리즘에 의한 D_R 만큼 지연된 t_i^* 가 됨으로써 연결 관리의 수행 결과가 연결 관리의 요구 순서와 일치되어 master LNC에 수집된다.

2. Asynchronous-with-Broadcasting 방안

그림 4에서 처럼 CC로부터 trail 연결 요구를 받은 master LNC A는 분산된 RCM에 의해 인접한 서브네트워크(LNC B, LNC C)의 정보만으로 LNC B, C의 연결 관리를 수행한다. 그러나 master LNC A에서 pseudo LNC B 또는 LNC C를 경유하여 LNC D에 대한 federation을 수행하고자 할 경우 pseudo LNC들(LNC B, LNC C)을 통한 LNC D의 정보를 master LNC가 직접 알 수 없기 때문에 LNC D에 대한 효율적이고 신뢰성 있는 pseudo LNC의 선택이 어렵게 된다.

만약에 LNC B와 LNC D 사이의 연결 상에 장애가 발생한 경우 이에 대한 정보 없이 master LNC A가 LNC B를 pseudo LNC로 선택하여 federation을 시도하게 되면 pseudo LNC B와 LNC D간의 장애로 인하여 federation의 수행이 실패한다. 그리고 master LNC A가 LNC D의 federation 수행함에 있어 pseudo LNC C를 경유하는 경로가 있음에 불구하고 연결 관리를 수행하지 못하게 되는 결과를 초래하여 전체 연결의 신뢰성을 저하시킨다.

따라서 본 연구는 둘 이상의 선택 가능한 pseudo LNC를 경유하는 federation의 신뢰성을 향상시키기 위한 asynchronous-with-broadcasting 방안을 제안한다. Asynchronous-with-broadcasting 방안은 전체 전송로에 정보를 broadcasting을 통하여 획득한 후 running-wait 알고리즘을 적용된 비동기적 형태의 federation을 수행하는 방안이다. 그리고 이 방안은 초기에 전체 연결로 설정의 성공 여부 및 인접한 서브네트워크와 탠덤 정보를 획득하는데 소요되는 시간을 비동기적 형태의 federation 수행 시간으로 최소화한 방안이다.

이 방안의 수행 방법은 다음과 같다. 우선 master LNC가 전체 연결의 정보 요구를 인접한 모든 pseudo LNC로 broadcasting한다. 이때 전체 연결의 성공 여부 및 서브네트워크와 탠덤 연결의 정보를 요구하는 메시지를 TCInfoReq 메시지라 정의하고 TCInfoReq 메시지의 응답 메시지를 TCInfoRes 메시지라 정의한다. TCInfoReq 메시지를 수신한 pseudo LNC는 자신의 RCM을 통하여 요구된 전체 연결의 성공 여부 및 서브네트워크와 탠덤 연결의 정보를 결정한다. 그리고 이를 TCInfoRes에 첨부하여 master LNC로 전달한다. Pseudo LNC로부터 TCInfoRes를 수신한 master LNC는 수집한 정보를 기반으로 QoS(Quality of Service)가 가장 큰 경로를 선택하고 이 경로에 대한 연결 관리를 수행한다. 이때 master LNC는 비동기적 형태의 federation 수행 절차에 따라 선택된 pseudo LNC에게 탠덤 연결을 위한 initiateTC를 송신하고 동시에 하부의 CP에게 자신의 서브네트워크 연결을 위한 initiateSNC를 송신한다.

Asynchronous-with-broadcasting 방안과 running-wait 알고리즘이 적용된 비동기적 형태의 cascaded 모델의 수행 절차는 그림 10과 같고 이의 세부적인 내용은 다음과 같다.

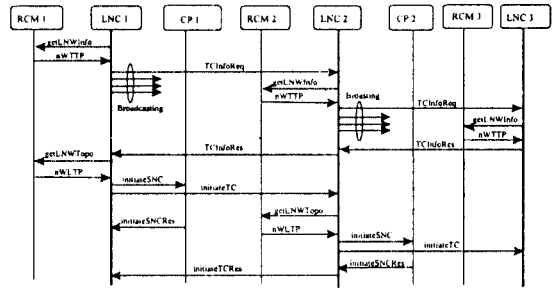


그림 10. 제안한 알고리즘 및 방안이 적용된 비동기적 형태의 cascaded 모델의 수행 절차

Master LNC 1이 CC로부터 initiateTrail을 수신하면 이에 대한 NWTTTP 정보를 자신의 RCM 1에게 getLNWInfo 동작으로 요구한다. RCM 1으로 부터 NWTTTP 정보를 수신한 master LNC 1은 이를 TCInfoReq에 첨부하여 자신의 인접한 모든 LNC들로 broadcasting 한다. 이때 master LNC 1은 이 연결 관리의 시작 지점이기 때문에 borrower를 비워두고 lender를 broadcast로 설정한다.

TCInfoReq를 수신한 Pseudo LNC들은 borrower를 master LNC 1으로 설정한 후 자신의 RCM들에게 getLNWInfo 동작으로 NWTTTP 정보를 요구한다. RCM들로부터 NWTTTP 정보를 수신한 pseudo LNC들은 수신한 TCInfoReq의 최종 착신단 종단점과 비교하여 전체 연결의 성공 여부를 결정한다. 만약에 최종 착신단 종단점과의 연결을 위한 또 다른 federation이 요구되면 자신의 RCM들로부터 수신한 NWTTTP 정보를 수신한 TCInfoReq에 첨부하여 자신의 인접한 모든 LNC들로 broadcasting 한다. 이때 pseudo LNC는 lender를 broadcast로 설정한다.

TCInfoReq를 수신한 두 번째 pseudo LNC들은 lender를 첫번째 pseudo LNC로 설정한 후 자신의 RCM들에게 getLNWInfo 동작으로 NWTTTP 정보를 요구한다. RCM들로부터 NWTTTP 정보를 수신한 두 번째 pseudo LNC들은 수신한 TCInfoReq의 최종 착신단 종단점과 비교하여 전체 연결의 성공 여부를 결정한다. 전체 연결의 성공 여부가 결정되면 자신의 borrower인 첫번째 pseudo LNC로 TCInfoRes를 송신한다. TCInfoRes를 수신한 첫번째 pseudo LNC는 자신의 borrower인 master LNC로 TCInfoRes를 송신

한다.

Master LNC A는 Pseudo LNC들로부터 수신한 전체 네트워크 정보와 사용자가 요구한 QoS를 기초로 하여 initiateTC를 송신할 최적의 통신로를 설정한다. 이후의 과정은 running-wait 알고리즘이 적용된 비동기적 형태의 cascaded 모델의 수행 절차와 동일하다.

V. 실험 및 고찰

1. 시뮬레이션 환경 설정

본 연구는 MIL3사의 OPNET(Optimized Network Engineering Tools)을 이용하여 제안한 알고리즘 및 방안이 적용된 federation 모델의 성능을 분석하였다. 시뮬레이션 모델은 크게 cascaded 모델과 joint 모델로 구분하고 이 모델에 동기적 형태의 수행 절차와 비동기적 형태의 수행 절차를 적용한다. 또한 비동기적 형태의 cascaded 모델에 제안한 running-wait 알고리즘과 asynchronous-with-broadcasting 방안을 적용하고 이를 편의상 표 1과 같이 모델 1~모델 6으로 칭한다.

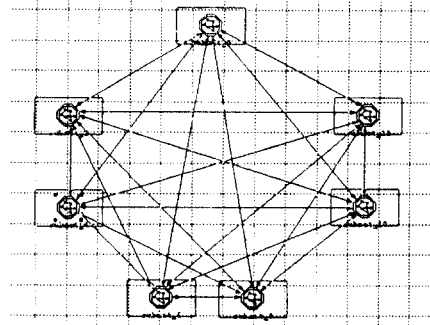
표 1. 시뮬레이션 수행 모델

cascade 모델		joint 모델	
모델 1	동기적 형태의 수행 절차	모델 2	동기적 형태의 수행 절차
모델 3	비동기적 형태의 수행 절차	모델 4	비동기적 형태의 수행 절차
모델 5	모델 3 + running-wait 알고리즘		
모델 6	모델 5 + asynchronous-with-broadcasting 방안		

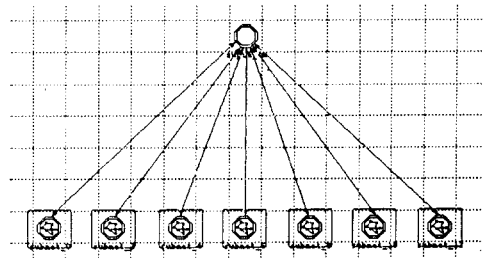
Cascaded 모델과 joint 모델의 시뮬레이션 네트워크 모델은 그림 11(a), (b)와 같다. 7개의 서브네트워크가 mesh 형태로 구성된 그림 11 (a) 모델은 cascaded 모델이 적용된 모델 1, 모델 3, 모델 5, 모델 6에 공통으로 사용된다. 하나의 별도의 노드와 7개의 서브네트워크로 구성된 그림 11 (b) 모델은 joint 모델이 적용된 모델 2, 모델 4에 공통으로 사용된다. 하나의 서브네트워크 구성은 최상의 연결 관리 패키지에 의해 직접 제어되는 첫번째 서브네트워크와 하위의 연결 관리 패키지에 의해 제어되는 두 번째 서브네트워크

로 분할되고 하나의 연결 관리 패키지에 의해 제어되는 EML_CP는 3개로 구성한다. 따라서 시뮬레이션 모델은 1개의 supra 연결 관리 패키지, 7개의 하부 연결 관리 패키지(7개의 LNC, 14개의 NML_CP, 42개의 EML_CP)로 구성된다.

전국적인 규모의 망 관리 시스템이 설치된 경우를 가정하여 네트워크 관리 계위간의 거리는 100km로 설정하고 네트워크 관리 계층과 망 요소 관리 계층간에는 동일 구역에 설치된다고 가정하여 20km로 설정한다. Federation이 발생하는 탠덤의 거리는 네트워크간에 설치된다는 점을 감안하여 200km를 설정한다. 각 전송로의 전송 속도는 DS1급의 1544Mbps라고 가정하였다. 연결 관리 패키지간은 광섬유가 적용되는 것을 가정하여 각 연결 관리 패키지간의 전파 시간 지연(propagation delay)을 정하였다. 하나의 CC에서만 연결 요구가 발생되며 포아송 분포를 적용한 연결 요구 발생의 변화에 따른 각 모델의 성능을 비교 분석하였다. 각 연산 객체의 메시지 처리율은 연



(a) cascaded 모델



(b) Joint 모델

그림 11. 시뮬레이션을 위한 네트워크 모델

결 요구 변화에 대한 상대적인 변화 추이를 분석하기 위하여 모든 연산 객체의 메시지 처리율을 100 message/sec로 가정한다.

2. 성능 분석

그림 12는 제안한 running-wait 알고리즘의 성능을 분석하기 위하여 모델 1, 3, 5에 연결 요구의 변화율 (arrival rate)을 5.2~7.6 message/sec로 변화시킨 성능 분석 결과이다. 여기서 에러 발생률은 전체 연결 요구의 발생 회수에 대한 수집 결과의 순차성 오류 발생 횟수를 백분율로 환산한 값이다. 그림 12에서 동기적 형태의 cascaded 모델(모델 1)은 순차성 에러를 발생시키지 않으나 비동기적 형태의 cascaded 모델(모델 3)은 연결 요구의 발생률이 증가할수록 에러 발생률은 증가된다. 이에 반해 제안한 running-wait 알고리즘이 적용된 비동기적 형태의 cascaded 모델(모델 5)은 동기적 형태의 cascaded 모델과 같이 순차성 오류가 발생되지 않음을 나타낸다. 따라서 비동기적 형태의 cascaded 모델은 자신이 갖고 있는 근본적인 순차성 오류 문제를 running-wait 알고리즘으로 해결함으로써 동기적 형태의 cascaded 모델과 같은 순차적 특성을 갖게 된다.

그림 13은 제안한 알고리즘 및 방안의 시간 지연 특성을 분석하기 위하여 federation 횟수를 2회로 고정하고 연결 요구의 변화율을 5.2~8.0 message/sec로 변화시킨 성능 분석 결과이다. 여기서 시간 지연은 연결 요구가 발생한 시점부터 연결 요구가 완료된 시점까지의 시간이다. 그림 13에서 cascaded 모델이 joint 모델보다 전반적으로 연결 요구 변화율에 따른 시간 지연 특성이 우수하다. joint 모델의 경우 연결 요구 변화율에 대하여 상당히 불안정적인 특성을 나타내는 반면에 cascaded 모델의 경우 연결 요구 변화율에 대하여 안정적인 특성을 나타낸다. 동기적 형태의 federation 모델보다 비동기적 형태의 federation 모델이 시간 지연 특성이 우수하다. 제안한 알고리즘 및 방안이 적용된 모델 5와 모델 6의 결과는 순차성 에러를 제어하기 위해 발생하는 시간 지연과 초기의 통신로 설정을 위한 시간 지연으로 인하여 순수한 비동기적 형태의 cascaded 모델보다 다소 시간 지연 특성이 저하되나 동기적 형태의 cascaded 모델보다 우수한 시간 지연 특성을 나타낸다. 이는 제안한 알고

리즘 및 방안의 수행 시 발생하는 부가적인 시간 지연 특성은 비동기적 형태의 우수한 시간 지연 특성으로 인하여 충분히 극복됨을 나타낸다.

그림 14는 federation 횟수의 증가에 따른 시간 지연 특성을 분석하기 위해서 연결 요구 발생률을 5.2 message/sec로 고정하고 federation을 발생 횟수를 0~6회로 증가시킨 성능 분석 결과이다. 그림 14에서 joint 모델의 경우 federation 횟수가 4회를 지나면서 급격한 시간 지연 특성의 증가를 나타내지만 cascaded 모델의 경우 federation 횟수의 증가에 대하여 시간 지연의 변화가 거의 없다. 비동기식 형태의 federation 모델의 경우 동기식 형태의 federation 모델과 달리 federation 횟수의 증가와 관계없이 안정적인 시간 지연 특성을 나타낸다. 또한 제안한 알고리즘 및 방안이 적용된 모델 5와 모델 6의 경우 순차성 에러를 처리하기 위한 시간 지연과 초기의 연결 관리 설정을 위한 시간 지연 때문에 federation 횟수가 증감함에 따라 모델 3보다 시간 지연이 증가하나 동기적 형태의 federation 모델들보다 상당히 우수한 시간 지연 특성을 나타낸다.

결론적으로 시간 지연 특성은 모델 3이 가장 우수하나 이 모델은 순차성 에러와 전체 연결의 신뢰성 문제가 남아 있다. 따라서 이를 해결하기 위해 제안한 알고리즘 및 방안이 적용된 모델은 동기적 형태의 순차성을 유지하면서 동기적 형태의 federation 모델

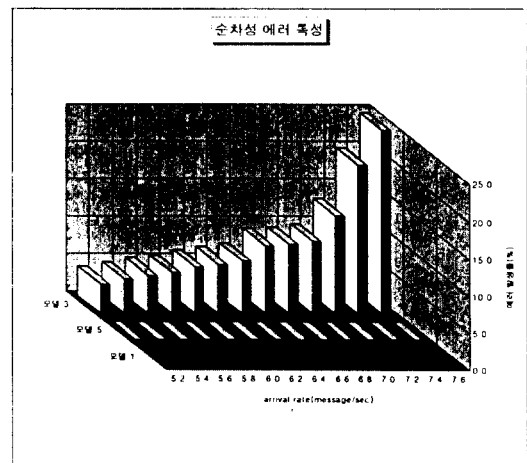


그림 12. 순차성 에러 특성

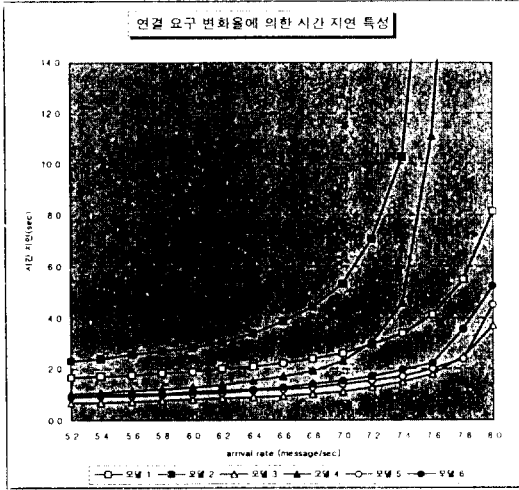


그림 13. 연결 요구 변화율에 의한 시간 지연

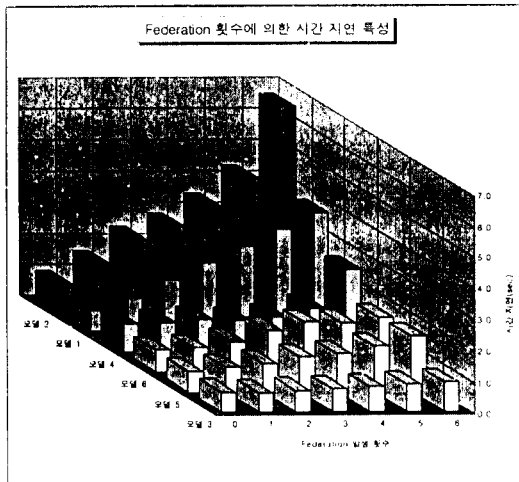


그림 14. Federation 횟수에 의한 시간 지연 특성

보다 우수한 시간 지연 특성을 갖게 되며 또한 비동기적 형태의 joint 모델과 거의 유사한 시간 지연 특성을 갖게 된다.

VI. 결 론

본 논문은 TINA에서 제시된 cascaded 모델의 전체 연결에 대한 신뢰성 문제를 해결하기 위해 우선적으로

로 TINA의 federation 상호 동작 모델에서 연속적인 2회 이상의 federation 발생에 대한 구체적인 방안으로 pseudo LNC를 제시한다. 그리고 이를 기반으로 동기적 형태의 federation 수행 절차와 비동기적 형태의 federation 수행 절차를 모델링 하였다. 그리고 비동기적 형태의 federation 수행 절차의 경우에 발생하는 순차성 문제를 해결하기 위한 running-wait 알고리즘을 제안하며 또한 cascaded 모델의 전체 연결의 신뢰성 문제를 해결하기 위한 asynchronous-with-broadcasting 알고리즘을 제안한다. 이들의 성능 분석을 위하여 시뮬레이션을 수행하였다. 시뮬레이션 결과로서 비동기적 형태의 cascaded 모델의 순차성 오류 특성과 제안한 알고리즘 및 방안의 시간 지연 특성을 분석하였다. 분석 결과를 통해 제안한 방안 및 알고리즘이 적용된 federation 모델의 시간 지연 특성이 순차성을 보장하면서 동기적 형태의 federation 모델보다 우수하고 비동기적 형태의 joint 모델과 거의 유사함을 입증하였다.

본 논문은 7개의 서브네트워크에 국한하여 단대단 통신로 연결 관리의 일부분을 수행한 것으로 향후에는 보다 확장된 개념의 단대단 통신로 연결 관리 모델로 발전하여야 할 것이다. 즉 ATM망에 대하여 광범위하게 적용될 수 있는 시나리오를 설정한 후 이를 적용한 단대단 통신로 연결 관리의 연구를 수행함으로써 통신망 관리의 기본이 되는 단대단 통신로 연결 관리 체계를 보다 세밀하게 정립할 수 있을 것이다.

참 고 문 헌

1. NM Forum, omnipoint version 1.0, 1992.
2. ITU-T Rec. M.3010, Principle for a Telecommunication Management Network(TMN),1992
3. ITU-T Rec. M.3100, Generic Network Information Model, 1992.
4. ITU-T REC. X.701, System Management Overview, 1992.
5. The Third TINA workshop, 1992.
6. TINA-C, Overall Concepts and Principles of TINA version 1.0, 1994.
7. TINA-C, Management Architecture, 1994.
8. ITU-T Draft Rec. G.tna, Transport Network Ar-

- chitecture, 1994.
9. ITU-T Draft Rec. I.311, B-ISDN General Network Aspects, 1992.
 10. ITU-T Rec. X.901~X.903, Open Distributed Processing(ODP), 1993.
 11. TINA-C, Connection Management Architecture, 1994.
 12. ISO 10164, Information Technology-OSI System Management, 1991.
 13. ISO 10165, Information Technology-OSI Structure of Management Information, 1991.
 14. Masaki Wakano, Motoharu Kawanishi, Lars Richter, "Information Model to Support TINA Service and Management Applications", Globecom '94, 1994.
 15. Won-Kyu Hong, Eun-Ho Choi, "Multimedia Communication Management Architecture for Information Highways", ICC '95, 1995.
 16. ITU-T Rec. G.803, Architecture of Transport Network based on SDH, 1993.



朴 明 桓(Myeong Hwan Park) 정희원

1965년 9월 20일생

1991년 2월:성균관대학교 전자공학과 공학사

1994년 2월:성균관대학교 대학원 전자공학과 공학석사

1997년 2월:성균관대학교 대학원 전자공학과 공학박사

1997년 4월~현재:데이콤종합연구소 IT기술팀 주임연구원

관심분야:IT, THN, CORBA, TINA

許 琪 中(Ki Jung Huh)

정희원

1955년 8월 31일생

1982년 2월:성균관대학교 전자공학과 공학사

1991년 2월:성균관대학교 대학원 전자공학과 공학석사

1997년 2월:성균관대학교 대학원 전자공학과 공학박사

1981년 12월~1992년 8월:대우 통신(주) 근무

1992년 8월~현재:인덕전문대학 정보통신과 조교수

曹 圭 鏞(Kyu Seob Cho)

정희원

1951년 5월 3일생

1974년 2월:성균관대학교 전자공학과 공학사

1976년 2월:성균관대학교 대학원 전기공학과 공학석사

1989년 2월:성균관대학교 대학원 전자공학과 공학박사

1977년 3월~1992년 2월:ETRI 책임연구원

1993년 3월~현재:성균관대학교 전자공학과 정교수