

스패닝 트리 프로토콜을 이용한 브릿지 시뮬레이션*

A Simulation of Bridge using the Spanning Tree Protocol

이숙영** · 이은화** · 이미정** · 채기준** · 최길영*** · 강 훈***

Sook-Young Lee · Eun-Wha Lee · Mee-Jeong Lee · Ki-Joon Chae · Kil-Young Choi · Hun Kang

Abstract

MAC (media access control) bridge is used to interconnect separate LANs and to relay frames between the BLANs (bridged LANs). Bridge architecture consists of MAC entity, MAC relay entity and bridge protocol entity and performs learning, filtering and forwarding functions using filtering database. In this paper, we simulate these functions of bridge and the STP (spanning tree protocol). The STP derives an active topology from an arbitrarily connected BLAN. Our simulation model assumes a BLAN consisted of three bridges forming a closed loop. In order to remove the loop, each bridge process exchanges configuration BPDU (bridge protocol data unit) with other bridge processes connected to the bridge itself. To simulate the communication between bridges, we implement the IPC (inter-process communication) server using message queues. Our simulation results show that the assumed BLAN contains no closed loop and then there is no alternative route and no unnecessary traffic.

1. 서론

대부분의 망들은 여러 개의 LAN들로 구성되며, LAN들을 하나의 망으로 연결하기 위하여 2 계층에서 지원되는 브릿지가 사용된다. 브릿지의 종류에는 소스 라우팅 브릿지 (Source Routing Bridge)와 투명 브릿지 (Transparent Bridge)가 있다. 본 논문에서는 Ethernet을 대상으로하여 IEEE 802.1D MAC (Media Access Control) 브릿지 표준안에 근거한 투명 브릿지를 시뮬레이

션한다.

브릿지의 구조는 브릿지 프로토콜 엔티티와 MAC 릴레이 엔티티, 그리고 각 포트마다 연결된 MAC 엔티티의 세 가지 요소로 구성되며 브릿지마다 필터링 데이터베이스를 관리한다. MAC 릴레이 엔티티는 MAC 엔티티를 통하여 브릿지로 들어온 프레임에 대하여 다음의 세가지 기능을 수행한다. 우선 프레임의 근원지 주소를 이용하여 BLAN (Bridged LAN) 상에 존재하는 호스트들의 위치를 학습한다. 그리고 난 다음 필터링

* 본 연구는 한국전자통신연구원 광대역 통신망 연구부 위탁 연구 과제에 의한 것임.

** 이화여자대학교 컴퓨터학과

*** 한국전자통신연구원 광대역 통신망 연구부 차세대 인터넷 연구실

데이터베이스를 이용하여 프레임의 필터링과 포워딩을 결정한다. 만일 BLAN 상에 루프가 존재한다면 목적지 호스트는 같은 프레임을 중복하여 받는 혼란을 겪게 되며 또한, 망 자원의 낭비를 초래하게 된다. IEEE 802.1D 작업 그룹에서는 이러한 문제를 해결하기 위하여 BLAN을 대상으로한 스페닝 트리 프로토콜 (Spanning Tree Protocol)을 제안하였으며 이는 브릿지의 상위 계층에 위치하는 브릿지 프로토콜 엔티티에 의하여 수행된다[1][2].

본 논문은 3개의 브릿지가 루프를 포함하는 토폴로지로 구성된 BLAN을 모델로 설정하고, 브릿지의 기능과 함께 스페닝 트리 프로토콜을 이용하여 종단점들 간의 선택적인 라우팅을 제거하고 망을 트리 구조로 재구성하는 과정을 시뮬레이션하였다.

각 브릿지 프로세스들은 필터링 데이터 베이스를 구축하여 사용자의 데이터 프레임을 포워딩하면서 동시에 루프를 제거하기 위한 스페닝 트리 알고리즘을 수행한다. 스페닝 트리를 구성하기 위하여 각 브릿지 프로세스들은 자신의 지역 정보를 구성 BPDU (Bridge Protocol Data Unit)로 구성하여 자신과 연결된 모든 다른 브릿지 프로세스들과 주고 받게 된다. 이러한 브릿지 프로세스들간의 통신을 시뮬레이션하기 위하여 메시지 큐를 이용한 IPC (InterProcess Communication) 서버를 이용한다. IPC 서버는 각 브릿지 프로세스들의 식별자를 테이블로 유지하여 한 브릿지에서 전송된 BPDU를 그와 연결된 모든 브릿지 프로세스로 브로드캐스트하는 것을 시뮬레이션한다. 시뮬레이션 결과 가정한 BLAN 상에 존재했던 루프가 제거되고 토폴로지가 트리구조로 재구성되어 프레임의 포워딩시 불필요한 트래픽이 발생하지 않게됨을 알 수 있다.

본 논문의 구성은 다음과 같다. 2절에서는 브릿지의 구조와 기능 그리고 스페닝 트리 프로토콜에 대하여 설명하고, 3절에서는 스페닝 트리 프로토콜을 구현하여 브릿지 기능을 테스트하는 시뮬레이션 모델을 설명하고 시뮬레이션 결과를 분석한다. 마지막으로 4절에서는 결론 및 앞으로의 연구과제를 제시한다.

2. 브릿지의 개요

하나 이상의 포트가 구성된 브릿지는 각 포트마다

상태 정보를 유지하며 이를 이용하여 BLAN 상에 존재하는 호스트들의 위치를 학습하고 LAN들 사이에서 프레임의 포워딩과 필터링을 수행한다. IEEE 802 작업 그룹들 중 데이터 링크 계층에서 브릿지의 표준화 작업은 802.1D 작업 그룹에 의하여 이루어지며 이에 근거하여 본 장에서는 브릿지의 구조와 그 기능에 관하여 설명한다.

2.1 브릿지 구조

브릿지는 MAC 엔티티, MAC 릴레이 엔티티 그리고 브릿지 프로토콜 엔티티의 세 가지 요소로 구성된다. 이 중 MAC 엔티티는 IEEE 802에서 제시하는 매체 접근 방법에 의존한 기능 (Media Access Method Dependent Function)을 수행한다. MAC 릴레이 엔티티는 MAC 엔티티가 제공하는 서비스를 사용하여 호스트들의 위치 학습과 함께 프레임 릴레이와 필터링을 수행한다. 보다 상위 계층에 존재하는 브릿지 프로토콜 엔티티는 스페닝 트리 프로토콜에 의하여 BLAN의 토폴로지를 재구성한다. 이 엔티티는 각 포트마다 제공되는 논리 연결 제어 (Logical Link Control) 프로시저를 사용한다. 그림 1은 이러한 세 가지 엔티티로 구성된 브릿지의 구조를 보여주고 있다[3].

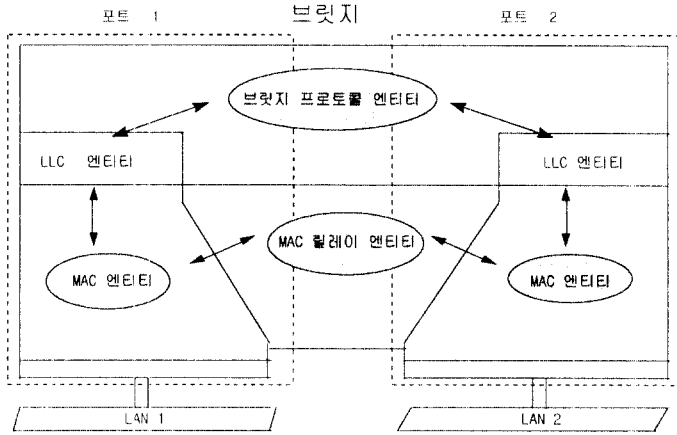
2.2 브릿지의 기능

브릿지는 종단 스테이션의 위치 정보를 학습하여 망상의 프레임을 필터링하거나 포워딩 시키는 기능을 수행한다. 이것은 브릿지를 구성하고 있는 세 가지 엔티티들 중 MAC 릴레이 엔티티에 의하여 수행된다. 이를 위하여 각 포트들의 상태 정보와 함께 브릿지 차원에서 관리되는 필터링 데이터베이스가 필요하다.

2.2.1 포트 상태 정보

각 브릿지 포트의 상태 정보는 그 포트와 연결된 MAC 엔티티로부터 받은 프레임의 처리 방법을 결정하는 중요한 요소이다. 포트가 취할 수 있는 상태로는 다음의 다섯 종류가 있으며 막힘 상태에서 시작하여 전송 상태로 갈수록 더 활성화된다.

- ① 불능 상태 (Disabled State)



〈그림 1〉 두 개의 포트로 구성된 브릿지 구조

이 상태의 포트는 스페닝 트리 프로토콜의 연산에서 제외되며 모든 브릿지 프로세스들은 이 상태의 포트를 통하여 전송된 프레임들을 무시한다. 이러한 불능 상태는 다른 모든 상태로부터 전이될 수 있으며 관리자에 의하여 작동 가능한 상태로 전이될 때 비로소 막힘 상태가 된다.

② 막힘 상태 (Blocking State)

이 상태의 포트는 프레임의 중복된 전송을 막기 위하여 위치 학습과 필터링 그리고 포워딩등의 모든 브릿지의 기능이 수행되지 않는다.

③ 듣기 상태 (Listening State)

듣기 상태는 막힘 상태에서부터 전이되며 마찬가지로 모든 브릿지 기능이 수행되지 않는다. 이 상태는 막힘 상태에서 학습 상태로 넘어가는 준비 단계이다.

④ 학습 상태 (Learning State)

포트가 학습 상태가 되면 프레임 필터링과 포워딩을 준비하기 위하여 망 내 종단 스테이션의 위치 학습을 시작한다. 즉, 이 상태의 포트를 통하여 전송된 프레임을 이용하여 그 근원지 주소에 해당하는 스테이션의 위치 정보를 필터링 데이터베이스에 기록한다.

⑤ 전송 상태 (Forwarding State)

전송 상태의 포트를 통하여 전송된 프레임들은 근원지 종단 스테이션의 위치 학습과 함께 프레임의 필터링과 포워딩에 모두 참여하게 되는 가장 활성화된 상태이다.

2.2.2 필터링 데이터베이스

필터링 데이터베이스는 MAC 주소와 그에 해당하는 포트 번호를 하나의 엔트리로 하여 구성되며 정적 엔트리와 동적 엔트리로 구성된다. 정적 엔트리에 의하여 명시되는 MAC 주소는 그룹 주소와 브로드캐스트 주소를 포함하며 관리자에 의하여 생성된다. 반면, 동적 엔트리는 그룹 주소가 아닌 특정한 종단 스테이션만을 나타내며 학습 프로세스에 의하여 생성되고 수정된다. 또한 그 생성 시점이나 가장 최근에 수정된 시점으로부터 일정 시간이 경과하면 자동적으로 제거된다. 이렇게 구성된 필터링 데이터베이스는 프레임의 필터링과 포워딩에 사용된다.

2.2.3 학습 프로세스

학습 상태나 전송 상태인 포트를 통하여 전송된 MAC 프레임들은 브릿지의 세 가지 프로세스들 중에서 우선 학습 프로세스로 전달된다. 학습 프로세스는 프레임의 근원지 주소를 조사하고 그것이 전송된 포트의 상태 정보를 참조하여 근원지 종단 스테이션에 대한 지역 정보를 필터링 데이터베이스에 추가한다. 이러한 학습 프로세스에 의하여 BLAN 상에서 특정 스테이션으로의 경로를 추론할 수 있게 된다. 학습 프로세스가 필터링 데이터베이스의 동적 엔트리를 생성하거나 수정하기 위해서는 근원지 주소가 특정한 하나의 단말만을 명시해야 한다.

2.2.4 필터링 프로세스

MAC 릴레이 엔티티는 프레임의 필터링을 통하여 망 내의 불필요한 트래픽을 제거한다. 우선 전송 상태인 포트를 통하여 프레임이 도착하면 필터링 데이터베이스내에 그 프레임의 목적지 주소에 해당하는 엔트리가 있는지를 조사하여 포워딩 시킬 포트를 결정한다. 만일 결정된 포트가 그 프레임이 들어온 포트와 일치하는 경우에는 프레임을 필터링 시킨다.

2.2.5 포워딩 프로세스

MAC 릴레이 엔티티는 필터링 데이터 베이스를 조사하여 전송된 프레임의 목적지 주소에 해당하는 LAN 과 연결된 포트를 결정한다. 그리고 나서 만일 프레임 이 필터링되지 않은 경우에는 그 포트의 상태 정보에 따라서 포워딩이 수행된다. 프레임의 포워딩은 그 LAN 과 연결된 포트의 상태가 전송 상태인 경우에만 수행되며 만일 해당하는 포트가 존재하지 않는 경우에는 프레임이 도착한 포트를 제외한 다른 모든 포트로 브로드캐스트시킨다.

2.3 스페닝 트리 프로토콜

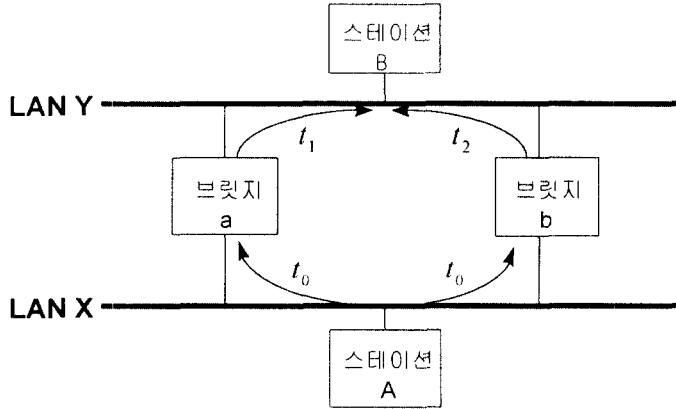
BLAN에서는 그림 2와 같이 토폴로지 상에 루프가 발생할 수 있다. 예를 들어 시간 t_0 에 스테이션 A가 스테이션 B를 목적지로서의 프레임을 전송하면 두 브릿지 a와 b가 모두 그 프레임을 받게 된다. 프레임을 받은 이 브릿지들은 스테이션 A가 LAN X 방향으로 위치한다는 사실을 필터링 데이터베이스에 기록하고 그 프레임 을 LAN Y로 포워딩 시킨다. 이때 브릿지 a는 시간 t_1 에 포워딩을 시작하고 브릿지 b는 그 보다 조금 뒤인 t_2 에 포워딩을 한다면 목적지 스테이션 B는 결국 같은 프레임을 두 번 받게된다[1][2].

이렇게 망 내에 존재하는 루프는 목적지 종단 스테이션에 중복된 프레임을 도착하게 하여 혼란을 초래할 뿐만 아니라 망 내에 불필요한 트래픽을 증가시킨다. 이를 제거하기 위하여 망 내에 선택적인 라우팅이 존재하지 않도록 토폴로지를 트리 형태로 구성하는 작업이 필요하다. IEEE 802.1D에서는 이러한 작업을 스페닝 트리 프로토콜로 정의하고 있다. 이 스페닝 트리 프로토콜은 물리적으로 연결되어 있는 BLAN을 선택적

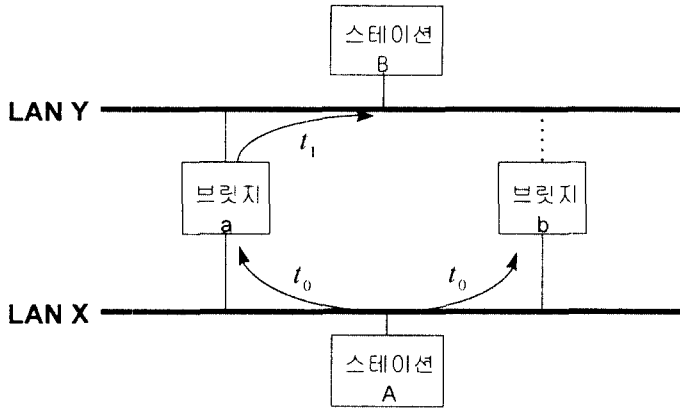
인 라우팅이 없는 간단한 액티브 토폴로지 (Active Topology) 즉, 논리적인 연결로 재구성하여 BLAN 상의 비효율적인 루프를 제거하는 방법이다[1][3][4]. 이를 위하여 여러 브릿지 프로토콜 엔티티들은 BPDU를 이용하여 필요한 정보를 주고 받으며 각 포트의 상태 정보를 변경시켜 프레임의 포워딩을 제어한다. 본 절에서는 이러한 스페닝 트리 프로토콜의 작동 과정을 액티브 토폴로지의 구성 과정을 중심으로 살펴본다.

2.3.1 액티브 토폴로지의 구성

BLAN을 구성하는 브릿지들 중 우선 순위가 가장 높은 임의의 브릿지를 루트 브릿지로 정한다. 루트 브릿지를 제외한 다른 브릿지들은 루트 브릿지로 가기 위한 포트가 한 개씩 존재한다. 이 포트를 루트 포트라고 한다. 그리고 각 LAN은 하나 이상의 브릿지에 직접 연결되어 있으며 LAN은 이 브릿지들 중 루트 브릿지로 가기 위하여 거쳐야 하는 브릿지를 한 개 선택하게 된다. 이 때 선택된 브릿지를 지정 브릿지라고 하며 지정 브릿지의 포트들 중에서 이 LAN과 연결되어 있는 포트를 LAN의 지정 포트라고 한다. 액티브 토폴로지는 각 브릿지에 할당된 유일한 브릿지 식별자 (Bridge Identifier)와 각 포트마다 주어진 경로 비용 (Path Cost) 그리고 포트 식별자 (Port Identifier)의 세 가지 요소에 의하여 결정된다. 각 브릿지는 고정된 우선 순위를 가지며 가장 높은 우선 순위를 갖는 브릿지가 루트 브릿지로 선정되고 그 루트 브릿지와 다른 모든 브릿지 포트들간의 경로 비용이 결정된다. 이 비용을 루트 경로 비용 (Root Path Cost)이라고 한다. 각 LAN의 지정 포트는 그 LAN과 연결된 포트들 중 가장 적은 값의 루트 경로 비용을 갖는 포트로 결정되며 만일 둘 이상의 포트가 같은 루트 경로 비용을 갖는 경우에는 그 포트가 속한 브릿지의 브릿지 식별자와 그들의 포트 식별자의 우선 순위에 따라서 결정된다. 이와 유사한 방법으로 각 브릿지는 루트 브릿지로 가는 가장 적은 비용의 포트를 루트 포트라고 선정한다. 위와 같은 과정을 거쳐서 액티브 토폴로지가 완성된다. 그림 3은 그림 2의 물리적인 토폴로지를 액티브 토폴로지로 재구성한 것이다. 실선은 물리적인 토폴로지와 액티브 토폴로지에 모두 참여하는 링크이며 점선은 액티브 토폴로지에서 제외된 링크이다. 실선으로만 구성된 토폴로



<그림 2> 루프가 있는 BLAN 토폴로지



<그림 3> 그림 2의 액티브 토폴로지

지에서는 루프가 발생하지 않는다.

2.3.2 프로토콜 정보의 전송

브릿지 프로토콜 엔티티들은 구성 BPDU (Configuration BPDU)와 토폴로지 변경 알림 BPDU (Topology Change Notification BPDU)를 이용하여 필요한 정보들을 상호 교환하므로써 액티브 토폴로지를 구성한다. BPDU를 전송하는 MAC 프레임의 목적지 주소는 브릿지 그룹 주소로 명시되어 프레임이 전송되는 LAN 상의 모든 브릿지들에게 전송된다. 액티브 토폴로지가 형성되는 시작 단계에서는 모든 브릿지들이 자신을 루트 브릿지로 가정하여 자신과 연결된 모든 LAN에 정

기적으로 구성 BPDU를 전송한다. 구성 BPDU는 이 프레임을 생성한 브릿지의 식별자, 그것을 전송하는 포트에 의하여 루트라고 여겨지는 브릿지의 식별자, 두 브릿지 사이의 경로 비용 그리고 이 구성 BPDU가 전송되는 포트의 식별자로 구성된다. 구성 BPDU를 받은 브릿지는 이러한 정보들과 현재 자신이 루트 브릿지라고 믿고 있는 브릿지의 정보(브릿지 식별자의 우선 순위나 루트 경로 비용 같은 정보)를 비교한다. 이 비교 작업을 통하여 그 구성 BPDU가 전달된 LAN에 대한 지정 포트와 지정 브릿지를 다시 결정하고 보다 나은 정보를 그 브릿지의 모든 포트를 통하여 계속 전송한다. 토폴로지 변경 알림 BPDU는 망 내의 토폴로지가

변경된 경우 이를 루트 브릿지에게 알리기 위하여 루트 포트에 전송된다.

2.3.3 포트 상태의 전이

구성 BPDU와 토폴로지 변경 알림 BPDU를 주고받으므로써 망 내의 각 포트들은 2.2.1에서 설명한 여러 상태들 사이에서 변경되어 스페닝 트리를 구성한다. 이러한 포트의 상태 변이는 BLAN내에서 토폴로지 정보가 전송될때의 전파 지연 시간을 고려한 것이다. 즉, 액티브 토폴로지가 변경될때 그 전파 지연 시간 동안 일시적으로 루프가 발생하여 프레임이 중복되거나 순서가 어긋나는 오류가 발생할 수 있다. 이러한 문제를 해결하기 위하여 각 브릿지들은 새로운 토폴로지 정보가 전송됐을 때 이전의 액티브 토폴로지에 근거하여 전송된 BPDU들이 타임 아웃되도록 기다려야만 한다. 이 시간 동안에는 필터링 데이터베이스내의 정보들도 타임 아웃시키고 새로운 지역 정보를 얻는 작업이 필요하다. 그렇게 하므로써 포트가 처음 전송 상태로 들어갔을 때 초기 프레임의 범람을 최소화시킬 수 있다. 따라서 각 포트들은 프로토콜에 의하여 막힘 상태에서 전송 상태로 변경되기 전에 듣기 상태를 거치게 된다. 이 듣기 상태에서 일정 기간의 타이머가 종결되면 학습 상태로 전이되고 최종적으로 전송 상태가 되나 프로토콜 정보에 의하여 막힘 상태로 되돌아 갈 수도 있다.

2.3.4 프로토콜 타이머

BLAN의 구성요소가 제거되거나 혹은 관리상의 이유로 토폴로지가 변경되는 경우에 액티브 토폴로지를 재구성하기 위하여 타이머가 필요하다. 망 내의 모든 브릿지들은 같은 값을 사용하여 이전 정보의 타임 아웃을 인식하며 이 값은 망 내의 각 LAN 상에 BPDU가 전송되는 전파 지연 시간을 의미한다. 액티브 토폴로지가 안정된 경우에는 하나의 루트 브릿지에 의하여 정기적으로 구성 BPDU가 전송되며 이 경우는 토폴로지 정보가 타임 아웃되지 않았음을 의미한다. 모든 브릿지들은 루트 포트를 통하여 전송된 정보를 저장하고 그 타임 아웃 값을 조사한다. 만일 루트 브릿지에 문제가 발생한 경우에는 다른 브릿지들의 프로토콜 정보 역시 타임 아웃된다. 즉, 브릿지의 루트 포트가 타임

아웃되고 다른 포트가 루트 포트에 선택되며 이러한 새 루트 포트에 기반한 정보가 그 브릿지가 지정 브릿지의 역할을 하고 있는 LAN 상으로 계속 전달된다.

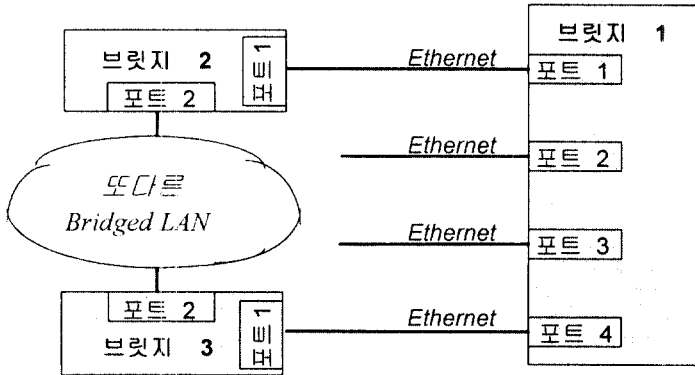
3. 브릿지 시뮬레이션

이 장에서는 앞서 설명한 스페닝 트리 프로토콜을 구현하여 브릿지 기능을 테스트하는 시뮬레이션 과정을 설명한다. 본 논문의 시뮬레이션은 크게 두 부분으로 구성된다. 우선 스페닝 트리 프로토콜을 이용하여 액티브 토폴로지를 형성하는 과정을 시뮬레이션하고 또한 각 종단 스테이션들의 위치 학습과 함께 프레임의 필터링과 포워딩 기능을 시뮬레이션한다.

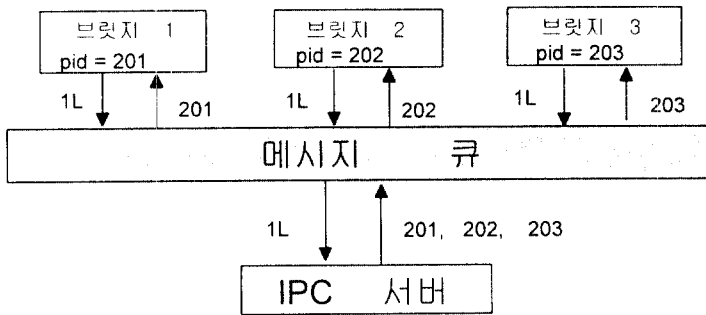
3.1 시뮬레이션 모델과 프로세스간 통신

스패닝 트리 형성과정과 브릿지 기능을 시뮬레이션하기 위하여 그림 4와 같이 루프를 포함한 모델을 가정한다. 브릿지 1의 두 포트는 각각 브릿지 2와 브릿지 3에 연결되며 다시 브릿지 2와 브릿지 3은 또 다른 BLAN에 의하여 연결되어 있다. 브릿지들간의 모든 링크는 Ethernet으로 가정하며 세 개의 브릿지 1,2,3 사이에는 2.3절에서 설명한 비효율적인 루프가 존재한다. 이러한 모델을 기반으로 스페닝 트리 알고리즘을 시뮬레이션하여 세 개의 브릿지 사이의 루프가 제어됨을 보인다.

각 브릿지 프로토콜 엔티티들은 스페닝 트리를 형성하기 위하여 자신의 파라미터 정보를 이용하여 구성 BPDU를 작성하고 이를 상호 교환하는 동작을 반복한다. 이러한 각 브릿지 프로세스들간의 통신을 시뮬레이션하기 위하여 IPC 서버가 필요하며 본 논문에서는 메시지 큐를 이용한다. 그림 5와 같이 메시지 큐를 사이에 두고 IPC 서버와 브릿지 프로세스 사이에 클라이언트-서버 모델로써 브릿지 프로세스들간의 통신이 이루어진다. 각 브릿지들은 IPC 서버에게 자신의 프로세스 식별자 (PID)를 등록하며 IPC 서버는 이를 이용하여 브릿지 프로세스들을 구별하게 된다. IPC 서버는 BLAN 상의 모든 브릿지들의 프로세스 식별자를 테이블로 관리하여 임의의 브릿지 포트를 출발한 구성 BPDU가 그 포트와 연결된 LAN 상의 모든 브릿지로



<그림 4> 시뮬레이션 모델



1L, 201, 202, 203 : message type

<그림 5> 메시지 큐의 모델

브로드캐스트되는 과정을 시뮬레이트한다.

3.2 자료 구조

각 브릿지와 그에 속한 포트들은 스패닝 트리를 구성하기 위한 파라미터 정보를 브릿지 프레임과 포트 프레임으로써 관리하게 된다. 브릿지 프레임은 브릿지 식별자, 루트 포트 식별자 그리고 그 루트 포트를 통한 루트 경로 비용을 포함한다. 또한 각 포트 프레임은 포트 식별자와 포트 상태 정보 그리고 그 포트와 연결된 LAN에 대한 정보인 지정 브릿지와 지정 포트에 대한 정보를 포함하고 있다. 브릿지는 이러한 정보들을 이용하여 구성 BPDU와 토폴로지 변경 알림 BPDU 프레임을 생성하고 다른 브릿지 프로토콜 엔티

티들과 정보를 교환한다. 이러한 모든 프레임들의 포맷은 IEEE의 802.1D MAC 브릿지에서 제공하는 표준 프레임 포맷에 근거한다[3].

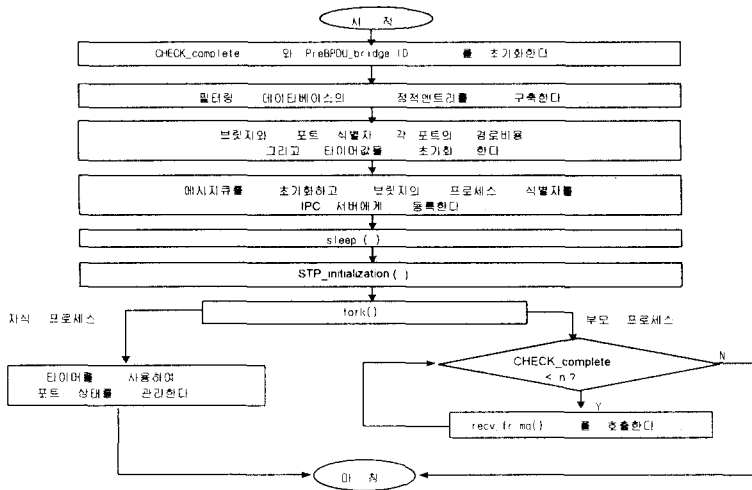
3.3 스패닝 트리 프로토콜 시뮬레이션

본 시뮬레이션의 전체 흐름도는 그림 6과 같다. 먼저, 시뮬레이션의 종결을 판별하기 위한 두 변수, CHECK_complete와 preBPDU_bridgeID를 초기화한다. CHECK_complete 변수는 같은 브릿지 식별자를 갖는 연속된 구성 BPDU의 개수를 나타내며 preBPDU_bridgeID는 가장 최근에 받은 구성 BPDU의 브릿지 식별자를 기록한다. 이 두 변수는 다음과 같은 사실에 근거하여 시뮬레이션의 종결을 결정한다. 구성 BPDU는

진정한 루트 브릿지나 혹은 루트가 되고자 시도하는 브릿지에 의하여 생성되어 다른 브릿지들로 전송된다. 만일 토폴로지내의 루프가 제거되어 액티브 토폴로지가 완성되면 단 하나의 루트 브릿지만이 구성 BPDU를 생성하게 되고 망 내에 전송되는 모든 구성 BPDU들의 루트 브릿지 식별자는 같은 값을 나타내게 된다. 이렇게 스페닝 트리의 완성 여부는 각 브릿지가 같은 루트 브릿지 식별자를 갖는 구성 BPDU를 n번 연속하여 받은 경우로 판별되며 이를 기준으로 시물레이션을 종결한다.

두 등록할 때까지 스페닝 트리 알고리즘의 시작을 지연시킨다.

STP_initialization()에서는 스페닝 트리 알고리즘의 수행 과정에서 변경되는 파라미터들을 초기화시킨다. 즉, 모든 포트를 지정 포트로 선언하고 그 상태를 듣기 상태로 초기화 한다. 또한 루트 경로 비용과 루트 포트를 초기화하고 토폴로지 변경을 감지하는 파라미터들을 거짓으로 초기화 한다. 모든 초기화 과정을 마친 브릿지 프로세스는 자신을 루트 브릿지로 가정하고 모든 포트를 통하여 구성 BPDU를 전송하므로써 스페닝 트



〈그림 6〉 시물레이션의 전체 흐름도

시물레이션의 종결에 대한 초기화가 끝나고나면 각 브릿지 프로세스는 필터링 데이터베이스의 정적 엔트리를 파일로 구축한다. 그리고 나서 시물레이션 수행 동안 그 값이 변경되지 않는 브릿지와 각 포트의 식별자 그리고 개별적인 포트의 경로 비용을 할당한다. 또한 각 포트의 상태 전이에 사용되는 타이머들은 IEEE 802.1D에 의하여 정의된 값으로 초기화시킨다[1]. 브릿지 프로토콜 자체에 대한 초기화가 끝나면 다른 브릿지 프로세스와의 통신을 위하여 IPC 서버에게 자신의 프로세스 식별자를 등록한다. IPC 서버는 BLAN 상의 모든 브릿지들의 프로세스 식별자를 알아야 하므로 각 브릿지 프로세스는 IPC 서버와 초기화를 마치고 나서 sleep() 시스템 함수를 이용하여 다른 프로세스들이 모

리 프로토콜을 개시한다. 스페닝 트리 프로토콜이 구동하고나면 자식 프로세스를 생성하여 포트의 상태 전이를 담당하는 타이머를 구동시킨다. 부모 프로세스는 CHECK_complete 변수가 미리 명시된 n개를 초과하지 않는 범위내에서 계속적으로 recv_fr_mq()를 호출하여 IPC 서버로부터 전송되는 프레임들을 처리한다.

recv_fr_mq()에서는 전송된 프레임들 디캡슐레이션하여 그 프레임이 스페닝 트리 프로토콜을 위한 제어 프레임인지 사용자를 위한 데이터 프레임인지를 구별한다. 망 내의 데이터 프레임을 가정하기 위하여 사용자 데이터 MAC 프레임을 생성하는 프로세스를 시물레이션에 추가하였다. 이 프로세스는 일정 간격으로 데이터 프레임들을 생성하여 IPC 서버에게 전달한다. 프레임

의 판별은 목적지 주소를 이용하며 그것이 IEEE 802.1D에서 명시한 브릿지 그룹 주소인 '0x0180c2000000'와 동일하면 제어 프레임으로 간주한다. 제어 프레임으로 판정된 프레임은 LLC 계층을 거쳐 상위에서 존재하는 브릿지 프로토콜 엔티티로 전달되어 2.3에서 설명한 스페닝 트리 프로토콜을 수행한다. 그리고 데이터 프레임인 경우에는 그 프레임이 도착한 포트의 상태를 조사한다. 만일 그 포트가 불능이나 막힘 또는 들기 상태일 경우에는 그 프레임을 버리고 수행을 종결하며 학습이나 전송 상태인 경우에만 MAC 릴레이 엔티티로 전달하여 브릿지의 세 가지 기능을 수행한다.

mac_relay() 함수가 이에 해당하며 다음 절에서는 이 함수에서 수행되는 위치 학습을 비롯한 프레임 필터링과 포워딩 등의 시뮬레이션 과정을 설명한다.

3.4 브릿지 기능의 시뮬레이션

mac_relay()에서는 MAC 엔티티로부터 받은 데이터 프레임의 근원지 주소를 먼저 조사한다. 이것이 만일 그룹 주소가 아닌 특정한 종단 스테이션을 나타내면 계속하여 필터링 데이터베이스에 그 근원지 주소에 해당하는 엔트리가 존재하는지를 검사한다. 만일 필터링 데이터베이스에 없다면 [프레임의 근원지 주소, 프레임이 도착한 포트 번호]의 형태로 새로운 종단 스테이션에 대한 위치 정보를 추가한다. 이 때 토폴로지의 변경으로 인하여 프레임의 근원지 주소는 같으나 그것이 도착한 포트가 다른 경우에는 해당하는 엔트리의 포트 번호를 갱신한다.

한 프레임에 대하여 이와 같은 위치 학습 프로세스가 수행되고 나면 계속하여 필터링과 포워딩 프로세스가 수행된다. 포워딩 프로세스는 필터링 데이터베이스를 검색하여 그 프레임의 목적지 주소에 해당하는 엔트리가 있는지를 조사한다. 만일 해당하는 주소의 엔트리가 없는 경우에는 그 프레임이 도착한 포트를 제외한 모든 포트에 브로드캐스트시킨다. 이때 브로드캐스트되는 포트들의 상태는 전송 상태에 있어야만 한다. 그렇지 않고 프레임의 목적지 주소에 대한 정보가 필터링 데이터베이스에 존재하는 경우에는 해당하는 포트가 그 프레임이 도착한 포트와 일치하는지의 여부를 검사한다. 만일 두 포트가 일치하면 이 프레임은 필

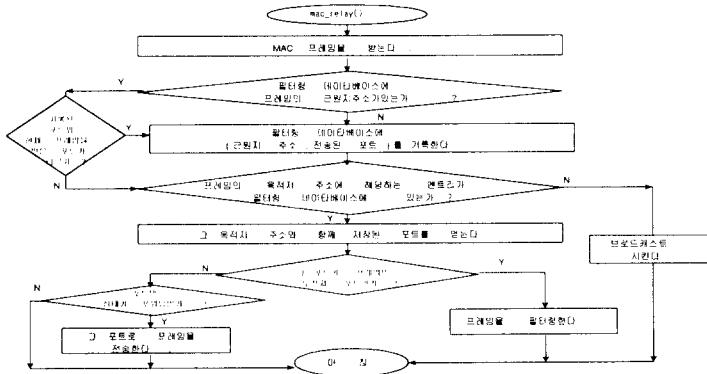
터링 프로세스에 의하여 더 이상 포워딩되지 않도록 버린다. 그러나 도착한 포트와 포워딩시킬 포트가 일치하지 않으면서 포워딩할 포트가 전송 상태이면 그 포트에 프레임을 포워딩한다. 이러한 브릿지의 위치 학습, 필터링 그리고 포워딩 프로세스의 흐름도는 그림 7과 같다.

3.5 시뮬레이션 결과

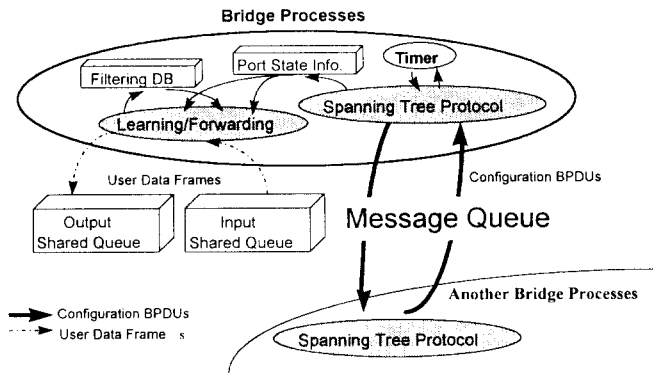
본 논문에서 시뮬레이션한 브릿지 프로세스의 전체 구성도는 그림 8과 같다. 하나의 브릿지 프로세스는 크게 스페닝 트리 프로토콜을 수행하는 부분과 브릿지의 위치 학습과 함께 프레임 필터링과 포워딩 등의 브릿지 기능을 수행하는 두 부분으로 구성하였다. 스페닝 트리 프로토콜에 의하여 브릿지의 각 포트 상태가 결정되고 이 상태 정보와 필터링 데이터베이스를 함께 이용하여 브릿지의 기능을 수행하였다. 한 브릿지내의 모든 포트들은 하나의 큐를 공유하며 다른 브릿지 프로세스와의 통신은 메시지 큐를 이용하여 시뮬레이션하였다.

〈표 1〉은 브릿지와 각 포트들의 파라미터를 중심으로 망 내의 루프가 제거되어 액티브 토폴로지가 형성되는 과정을 나타내고 있다. 브릿지의 파라미터에는 루트 브릿지로 가기 위한 루트 포트와 그때의 경로 비용을 나타내는 루트 경로 비용 파라미터가 있다. 브릿지내의 각 포트들의 파라미터로는 '포트 상태'와 그 포트가 루트라고 생각하는 브릿지의 식별자인 '루트 브릿지' 그리고 그 포트와 연결된 LAN 상에서 루트 브릿지로 가기 위한 경로를 제공하는 브릿지인 '지정 브릿지'와 그 때 경과하게 되는 포트를 나타내는 '지정 포트'가 있다. 브릿지와 포트의 식별자는 브릿지ID와 포트ID가 각각 가장 높은 우선 순위를 나타내는 것으로 가정한다.

스패닝 트리 프로토콜의 시작 단계에서는 모든 브릿지가 자신을 루트 브릿지로 가정하므로 각 포트의 지정 브릿지와 지정 포트가 모두 자신을 가리킨다. 그리고 그 정보를 기반으로 구성 BPDU를 생성하여 모든 포트에 전송하며 다른 브릿지로부터 구성 BPDU를 받은 브릿지는 자신의 정보와 그 BPDU 내의 정보를 비교하여 더 우선 순위가 높은 정보를 택한다. 두번째 단



<그림 7> 브릿지 기능의 흐름도



<그림 8> 브릿지 프로세스의 전체 구성도

계는 각 브릿지들 사이에서 구성 BPDUs를 한 번 주고 받았을 때 파라미터 값의 변화를 보여주고 있다. 이 단계에서는 각 브릿지의 포트들이 자신의 정보보다 더 우선 순위가 높은 구성 BPDUs를 받았을 경우 자신의 정보를 바꾼다. 브릿지2의 포트1은 브릿지1의 포트1로부터 받은 구성 BPDUs의 루트 브릿지 식별자가 자신이 가지고 있던 정보인 브릿지2 보다 더 높은 우선 순위를 가지므로 브릿지2의 포트1은 루트 브릿지와 지정 브릿지 그리고 지정 포트 정보를 브릿지1과 브릿지1의 포트1로 수정한다. 또한 브릿지3의 포트1도 같은 과정을 거쳐서 브릿지1이 루트 브릿지임을 인식하게 된다. 그리고 브릿지3의 포트2는 브릿지2로부터 브릿지2가 루트 브릿지인 구성 BPDUs를 받게되어 자신의 정보인 브릿지3 보다 우선하므로 루트 브릿지를 브릿지2로 수

정한다. 세 번째 단계에서 브릿지2와 브릿지3의 각 포트들은 자신이 가진 정보보다 더 높은 우선 순위를 가진 구성 BPDUs를 받아서 포트 정보가 바뀐 것을 브릿지에게 알린다. 그러면 브릿지는 루트 포트를 선정하고, 루트 경로 비용을 계산한 다음 지정 포트를 선택한다. 그리고 지정 포트로 선택된 포트는 자신의 정보를 갱신하게 된다. 브릿지2의 경우 포트1이 더 높은 우선 순위를 가진 구성 BPDUs를 받아 포트 자신의 정보가 바뀌었으므로 브릿지2의 루트 포트를 포트1로 결정하고, 이 포트의 경로 비용이 브릿지2의 루트 경로 비용이 된다. 그리고 브릿지2는 포트2를 지정 포트로 선정하여 포트2의 루트 브릿지를 브릿지1로 하고 지정 브릿지와 지정 포트를 자신의 브릿지 식별자2와 포트 식별자2로 설정한다. 브릿지3의 경우도 두 번째 단계

〈표 1〉 스페닝 트리 프로토콜의 시뮬레이션 수행 과정

시작 단계							
브릿지 식별자	루트 포트	루트 경로 비용	포트 식별자 (경로 비용)	포트 상태	루트 브릿지	지정 브릿지	지정 포트
브릿지 1	.	0	포트1 (5) 들기	브릿지1	브릿지1	포트1	
			포트4 (10) 들기	브릿지1	브릿지1	포트4	
브릿지 2	.	0	포트1 (15) 들기	브릿지2	브릿지2	포트1	
			포트2 (5) 들기	브릿지2	브릿지2	포트2	
브릿지 3	.	0	포트1 (20) 들기	브릿지3	브릿지3	포트1	
			포트2 (10) 들기	브릿지3	브릿지3	포트2	
두번째 단계							
브릿지 식별자	루트 포트	루트 경로 비용	포트 식별자 (경로 비용)	포트 상태	루트 브릿지	지정 브릿지	지정 포트
브릿지 1	.	0	포트6 (5) 들기	브릿지1	브릿지1	포트1	
			포트4 (10) 들기	브릿지1	브릿지1	포트4	
브릿지 2	.	0	포트1 (15) 들기	브릿지1	브릿지1	포트1	
			포트2 (5) 들기	브릿지2	브릿지2	포트2	
브릿지 3	.	0	포트1 (20) 들기	브릿지1	브릿지1	포트4	
			포트2 (10) 들기	브릿지2	브릿지2	포트2	
세번째 단계							
브릿지 식별자	루트 포트	루트 경로 비용	포트 식별자 (경로 비용)	포트 상태	루트 브릿지	지정 브릿지	지정 포트
브릿지 1	.	0	포트6 (5) 들기	브릿지1	브릿지1	포트1	
			포트4 (10) 들기	브릿지1	브릿지1	포트4	
브릿지 2	포트1	15	포트1 (15) 들기	브릿지1	브릿지1	포트1	
			포트2 (5) 들기	브릿지1	브릿지2	포트2	
브릿지 3	포트1	20	포트1 (20) 들기	브릿지1	브릿지1	포트4	
			포트2 (10) 들기	브릿지1	브릿지3	포트2	
최종 결과							
브릿지 식별자	루트 포트	루트 경로 비용	포트 식별자 (경로 비용)	포트 상태	루트 브릿지	지정 브릿지	지정 포트
브릿지 1	.	0	포트1 (5) 들기	브릿지1	브릿지1	포트1	
			포트4 (10) 들기	브릿지1	브릿지1	포트4	
브릿지 2	포트1	15	포트1 (15) 들기	브릿지1	브릿지1	포트1	
			포트2 (5) 들기	브릿지1	브릿지2	포트2	
브릿지 3	포트1	20	포트1 (20) 들기	브릿지1	브릿지1	포트4	
			포트2 (10) 막힘	.	.	.	

에서 포트1과 포트2 모두 구성 BPDU를 받아 포트 자신의 정보가 바뀌었다. 브릿지3은 두 포트 정보 중 루트 브릿지 식별자의 우선 순위가 더 높은 포트1을 루트 포트라고 정하고 루트 경로 비용은 이 포트의 경로 비용으로 설정된다. 그리고 브릿지3의 포트2는 지정 포트가 되어 브릿지1을 루트 브릿지로 하고 자신이 속한 브릿지3을 지정 브릿지로 수정한다. 마지막 단계에서는 브릿지2의 포트2와 브릿지3의 포트2 중에서 루트 경로 비용이 더 적은 브릿지2의 포트2가 두 브릿지 사이에 놓인 LAN의 지정 포트라고 선택되고 따라서 브릿지3의 포트2는 막힘 상태로 되돌아 간다. 이러한 과정을 통하여 그림 4의 BLAN은 브릿지1을 루트로하는 스페닝 트리로 재구성되어 루프가 제거된다.

4. 결론 및 향후 연구

본 논문에서는 스페닝 트리 프로토콜을 근간으로 하여 브릿지의 기능을 시뮬레이션하였다. 각 브릿지의 포트들은 다양한 상태 정보를 유지하고 있으며 그 중 학습 상태와 전송 상태인 포트를 통해서만 호스트의 위치 학습과 프레임의 포워딩이 수행되도록 하였다. 메시지 큐를 통한 IPC 서버에 의하여 각 브릿지 프로세스들은 자신의 지역 정보를 주고 받으며 스페닝 트리의 루트에 해당하는 브릿지를 결정하였다. 하나의 루트 브릿지가 선정되고 나면 프레임 릴레이 도중 BLAN 상에서 발생하게 되는 루프가 제거되고 토폴로지내의 두 호스트 사이에는 단 하나의 라우팅 경로만 존재하게 된다. 그러므로 망 내의 불필요한 트래픽이 발생하지 않고 LAN들을 보다 효율적으로 연결할 수 있었다.

향후의 연구 과제로는 스페닝 트리 프로토콜을 여러 다른 토폴로지를 대상으로 시뮬레이션을 수행하고 그 결과를 바탕으로 성능을 측정하여 필요한 개선점을 찾는 것이 필요하다.

참고문헌

- [1] W. Stallings, *Data and Computer Communications*, 5th Edition, Macmillan Publishing, New York, Prentice Hall, 1997.
- [2] A. S. Tranenbaum, *Computer Networks*, Prentice Hall,

- 3th Edition, New Jersey, 1996.
- [3] IEEE 802.1D, *IEEE standard for Local and Metropolitan Area Networks: Media Access Control Bridges*, Mar. 1991.
- [4] R. Perlman, *Interconnections Bridges and Routers*. Addison-Wesley, Massachusetts, 1992.
- [5] W. Richard Stevens, *Unix Network Programming*, Prentice Hall, New Jersey, 1994.
- [6] A. Alles, *Routing and Internetworking in ATM Networks*, Engineering InterOp, Las Vegas, Nevada, Mar. 1995.

● 저자소개 ●



이숙영

1991~1995년 이화여자대학교 컴퓨터학과 학사.
 1995~1997년 이화여자대학교 컴퓨터학과 석사.
 1998년~현재 LG 정보통신 연구원.



이은화

1992~1996년 국립 경상대학교 컴퓨터과학과 학사.
 1996~1998년 이화여자대학교 컴퓨터학과 석사.
 1998년~현재 LG 전자 연구원.



이미정

1983~1987년 이화여자대학교 전자계산학과 학사.
 1987~1989년 University of North Carolina at Chapel Hill 컴퓨터학과 석사.
 1990~1994년 North Carolina State University 컴퓨터공학과 박사.
 1994년~현재 이화여자대학교 컴퓨터학과 조교수.
 주 관심분야 고속 통신 프로토콜 설계 및 성능 분석, 고속통신망을 통한 멀티미디어 트래픽 전송, ATM 연동 기술.

**채기준**

1982년 연세대학교 수학과 학사.
 1984년 미국 Syracuse Univ. 전자계산학과 석사.
 1990년 미국 North Carolina State Univ. 컴퓨터공학과 박사.
 1990~1992년 미국 해군사관학교 전자계산학과 조교수.
 1992년~현재 이화여자대학교 컴퓨터학과 부교수.
 1996년~현재 이화여자대학교 정보전산원 원장.
 주 관심분야 고속통신망, LAN, 망관리, 성능평가.

**최길영**

1985년 경북대학교 전자공학과 학사.
 1987년 경북대학교 전자공학과 석사.
 1987년~현재 한국전자통신연구원 차세대인터넷연구실 선임연구원
 주 관심분야 ATM-LAN, B-ISDN, 고속라우팅.

**강 훈**

1980년 연세대학교 전자공학과 학사.
 1987년 Iowa State University 컴퓨터공학과 석사.
 1992년 Iowa State University 컴퓨터공학과 박사.
 1993년~현재 한국전자통신연구원 차세대인터넷연구실장
 주 관심분야 ATM-LAN, 고속근거리 통신망, 고속라우팅, 액세스망구조.