

MASH Toolkit을 이용한 인터넷 프로토콜 분산 시험 시스템

정희원 김병식*, 전우직*

Distributed Test System for Internet Protocols in MASH Toolkit

Byeongsik Kim*, Woojik Chun* *Regular Members*

요약

인터넷 관련 라우팅 프로토콜이나 시그널링 프로토콜들을 개발한 후에는 구현된 프로토콜의 기능 시험 및 성능 평가가 이루어져야 한다. 이를 위해서 기능 시험 및 성능 시험을 자동으로 행할 수 있는 시험 시스템의 필요성이 널리 인식되고 있다. 또한 이런 시스템은 시험 환경의 구성 및 각 시험 요소의 제어가 용이해야 함과 동시에 시험 요소 및 시험 대상 프로토콜이 지리적으로 분산되어 있는 상태에서 시험이 가능해야 한다. 본 논문에서는 인터넷 프로토콜의 기능 및 성능을 시험하는 시험 시스템의 설계 및 구현방법에 초점을 맞추고 있으며, 프로토콜 시험을 위해서 분산된 시험 스크립트를 기술하는 환경, 한 지점에서 지리적으로 분산되어 있는 여러 개의 시험 시스템의 구성 요소들을 제어하는 방법, 시험 시스템의 각 요소들의 구성을 쉽게 할 수 있는 시험 시스템 구조, 웹 브라우저 상에서 다운로드 및 실행하도록 함으로써 시험 시스템의 각 요소들의 이식성을 달성할 수 있는 방법에 대해서 소개한다. 또한 이 시험 시스템의 적용을 위해서 RSVP를 대상 프로토콜로 하여 시험 시스템의 구조 및 필요한 시험 스크립트에 대해서 기술한다.

ABSTRACT

One of the most difficult issues in the design and development of adaptive routing and signaling protocols is testing and measurement. To adequately test such technology requires methodologies and tools that permit topologically distributed experiments to be easily configured and controlled. Our research is focusing on the design and development of a distributed Internet protocol and performance test system. This paper discusses (1) provision of a distributed scripting environment for protocol testing, (2) the ability to control multiple, topologically distributed test system components from a single point, (3) a test system architecture that allows easy configuration and customization of test system components, (4) portability of test system components by allowing downloading and execution in WWW browsers, and (5) RSVP test system and RSVP-related test suites.

I. 서론

최근 특정 대역폭이나 실시간 전송을 필요로 하는 멀티미디어 통신 응용들의 사용이 날로 증가하고 있다. 이로 인해 기존의 최선형(Best-Effort) 서비스와 보다 향상된 QoS(Quality of Service) 지원이 가능한 인터넷의 필요성을 대두시키게 되었다.

그 결과 IETF(Internet Engineering Task Force)에서는 통합 서비스(Integrated Services)^[1]와 차별 서비스(Differentiated Services)^[2] 등과 같은 새로운 서비스 모델을 제시하고 표준화 하였다.

IETF에서는 통합 서비스와 차별 서비스의 제공을 위한 시그널링 프로토콜로 Resource Reservation Protocol(RSVP)^[3,4]를 제안하였다. RSVP 메커니즘

* 충남대학교 컴퓨터공학과

논문번호: 00336-0828, 접수일자: 2000년 8월 28일

* 본 연구는 미국 표준 기술 연구소(NIST)와의 공동 연구로 수행된 결과임.

에서 송신자는 자신이 생성하고자 하는 트래픽의 특징을 명세화 하기 위해서 Path 메시지를 수신자들로 전송한다. 이 때, 망의 중간 라우터들은 라우팅 프로토콜에 의해서 결정된 다음 라우터로 Path 메시지들을 전송하며 최종적으로 수신자는 해당 플로우에 대한 망 자원의 예약 요청을 위해서 Resv 메시지를 Path 메시지가 전달되어 온 역 방향으로 전달한다. 중간 라우터들은 Resv 메시지에 의한 자원 예약 요청을 허용 혹은 거절할 수 있다. 만일 요청을 거절하는 경우에는 수신자에게 ResvErr 메시지를 전달하고 시그널링을 종료한다. 만일 요청을 허용하는 경우에는 해당 플로우에 대해서 링크의 대역폭이나 버퍼 용량 등과 같은 자원을 예약하고 플로우의 상태 정보를 각 라우터에 유지한다.

통합 서비스는 기존의 최선형 서비스 외에 엄격한 지연 한계를 필요로 하는 응용들을 위한 보장형 서비스(Guaranteed Service)^[5]와 망의 부하가 많이 걸려 있을 때에도 부하가 많이 걸려 있지 않은 상태에서 기존의 최선형 서비스가 제공하는 정도의 QoS를 제공할 수 있는 서비스로 정의되는 부하제어형 서비스(Controlled-Load Service)^[6]로 구성된다. 보장형 서비스는 특정 패킷들에게 일정 수준의 대역폭 보장, 엄격한 종단간(End-to-End) 지연 한계, 큐잉으로 인한 패킷의 손실이 없도록 제공되는 서비스를 의미한다. 이 서비스는 엄격한 실시간 전송을 필요로 하는 응용들을 위해서 제공되는 서비스이며 이 서비스의 목적은 패킷 스위칭 망에서 전송선에 의해서 제공되는 서비스의 보장성을 달성하고자 하는데 목적이 있다. 부하제어형 서비스란 부하가 많이 걸려있지 않은 상황에서 기존의 최선형 서비스가 제공하는 정도의 서비스를 망의 어떠한 상황에서도 제공할 수 있는 서비스를 의미한다. 일반적으로 기존의 최선형 서비스를 기반으로 하는 인터넷은 망에 부하가 많이 걸려 있지 않을 때에는 성능이 어느 정도 유지되지만 부하가 많이 걸리는 경우에는 성능이 급격히 저하되는 성질을 가지고 있다. 이런 망에 부하제어형 서비스를 도입하면 항상 일정하게 어느 정도의 망 성능을 필요로 하는 응용들에게 적합한 서비스를 제공할 수 있다. 부하제어형 서비스의 정의는 이 서비스에 관련된 명확한 지연 한계에 대한 정의가 없으며 심지어는 패킷 손실률에 대한 제한 사항도 포함하고 있지 않기 때문에 승인 제어, 스케줄링, 버퍼 관리에 있어서 상당한 융통성을 허용한다.

인터넷상에서 통합 서비스를 개발하고 확장시켜

나가기 위해서는 관련 프로토콜들의 성능을 측정하고 분석할 수 있는 도구도 동시에 개발되어야 한다. 그러나, 프로토콜 자체의 개발 및 시장 진출을 위한 관심과 활동에 비해서 성능의 측정이나 프로토콜 자체의 시험을 위해서는 연구 및 투자가 미흡한 실정이다. 이를 위해서 본 논문에서는 인터넷 프로토콜들에 대해서 시험 방법과 지리적으로 분산되어 있는 상황에서 시험 방법의 재구성 및 제어가 용이한 시험 시스템의 구현 방법 및 결과에 대해서 기술한다. 또한 구현한 시스템을 RSVP에 적용한 결과에 대해서 소개한다.

본 논문의 구성은 다음과 같다. 제 2 절에서는 본 논문에서 제안하는 시험 시스템이 기반으로 하고 있는 MASH Toolkit에 대해서 소개한다. 제 3 절에서는 MASH Toolkit을 이용해서 일반적인 프로토콜 시험 시스템을 설계하고 구현하는 방법에 대해서 소개한다. 제 4 절에서는 본 논문에서 제안하는 시스템을 RSVP를 대상 프로토콜로 적용하는 방법 및 적용 결과에 대해서 기술한다. 마지막으로 제 5 절에서는 결론 및 향후 연구 과제에 대해서 기술한다.

II. MASH Toolkit

DARPA, Intel, Microsoft, Xerox PARC의 지원 하에 버클리 대학의 MASH 연구팀에서는 이질적이고 분산된 환경에서 멀티미디어 공동 연구를 위한 확장성이 좋은 멀티미디어 구조(Multimedia Architecture that Scales across Heterogeneous environments)를 개발하는 MASH 프로젝트를 진행 중이다. MASH 프로젝트는 인터넷상에서 멀티캐스트 기반의 공동 연구를 위한 응용들의 설계, 개발, 배포에 있으며 이 프로젝트의 중간 연구 결과물로 "MASH Toolkit" 이라는 멀티미디어 네트워킹 Toolkit을 개발하여 배포하고 있다. MASH Toolkit은 오디오/비디오 등과 같은 실시간 멀티미디어 스트림들을 사용하기 위한 미들웨어로서 버클리 대학의 Continuous Media Toolkit(CMT)^[7], MIT의 VuSystem^[8], LBL/UCB의 Mbone tools^[9,10] 등과 같은 여러 가지 멀티미디어 Toolkit들의 특징들을 결합시킨 확장성이 뛰어난 고성능 Toolkit이다^[11].

1. MASH Toolkit 구조

MASH 연구팀은 그림 1에서와 같이 IP 멀티캐스트를 기반으로 하는 인터넷 멀티미디어 통신 및 공동연구를 위한 포괄적인 구조를 정의하였으며 이

구조는 서로 밀접한 관련이 있는 다음과 같은 여러 요소들로 구성되어 있다.

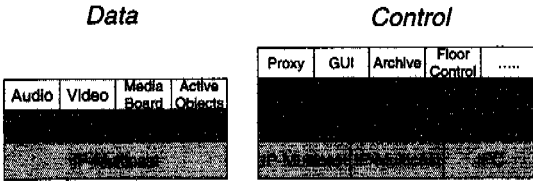


그림 1. MASH Toolkit의 구조

Scalable Reliable Multicast(SRM) : SRM은 응용들에게 신뢰성 있는 멀티캐스트를 지원하기 위한 프로토콜이다^[2]. MASH 연구팀은 SRM 프로토콜 메커니즘을 재사용이 가능하도록 구현하기 위해서 *libsrn* 이라는 C++클래스 라이브러리로 구현하였다.

Real-Time Transport Protocol(RTP) : RTP는 오디오나 비디오와 같은 실시간 데이터를 전달하는 응용들을 위해서 종단간(end-to-end) 전달 기능을 제공하는 인터넷 표준 프로토콜이다^[3]. RTP 자체만으로 QoS나 신뢰성 있는 데이터 전달을 보장하는 것은 아니지만 RTP는 거의 모든 실시간 응용들에게 공통적으로 포함되어야 할 기본적인 기능만을 제공한다. 그 외의 각 응용들에 대한 특정 요구사항들은 프로파일 형태로 RTP위에 추가된다. RTP는 RTP 세션에 참여하고 있는 응용들에게 QoS 피드백 제공을 담당하는 RTP Control Protocol(RTCP)와 같이 동작하는 형태를 취한다.

MediaBoard : 일반적으로 미팅에 참여한 사람들이 자신의 아이디어를 다른 참여자들에게 설명하기 위해서 화이트 보드나 프로젝터를 도구로 사용하는 것과 마찬가지로 인터넷 컨퍼런싱에서 사용할 수 있는 도구로 MediaBoard를 개발하였다. IP 멀티캐스트 그룹의 참여자들은 이 도구를 이용하여 자신의 아이디어를 그래픽으로 표현하여 설명하는데 이용한다. MediaBoard는 신뢰성 있는 데이터 전달을 위해서 SRM을 기반으로 동작한다.

Coordination Bus : Coordination Bus는 새로운 응용들을 생성하기 위해서 필요한 MASH 시스템내의 여러 구성 요소들을 조합하는 역할을 담당한다. 응용 개발자들은 새로운 응용을 개발하기 위해서 필요한 MASH Toolkit내의 여러 요소들을 Coordination Bus를 통해 조합하거나 매핑시킬 수 있다.

Proxy : Proxy는 사용자의 대역폭 요구 사항과 현재 이용 가능한 망의 대역폭을 기반으로 오디오, 비디오, 이미지 등의 질을 조정하는 역할을 한다.

Archive : 상호 동작하는 여러 세션들에 대해서 캡처링, 인덱싱, 저장, 검색 등을 Archive 모듈에서 담당한다.

Floor Control : MBone 비디오 컨퍼런싱 도구들의 개발로 인해 인터넷상에서 여러 명이 동시에 참여가 가능한 세미나 회의가 가능하다. 일반적으로 이런 MBone 세미나에서는 근거리의 참여자들의 질문 횟수가 원거리에 있는 참여자들의 질문 횟수보다 많다. 이런 차이를 위해서 Floor Control 메커니즘이 필요하며 MASH 연구팀에서는 대규모의 MBone 세미나에서 질문을 편리하게 하기 위한 Floor Control 메커니즘으로 QuestionBoard라는 도구를 개발하였다^[4].

2. MASH Toolkit의 소프트웨어 모델

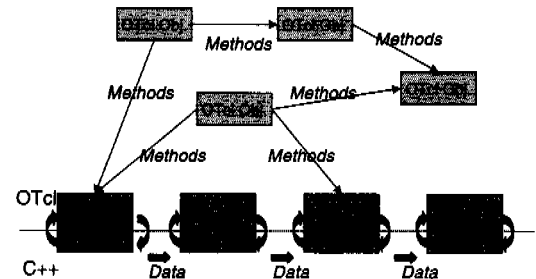


그림 2. MASH Toolkit의 소프트웨어 모델

멀티미디어 네트워킹 Toolkit들의 구현은 객체 지향 개념을 이용해서 구현된다. 예를 들어, 프로토콜 모듈들이나 미디어 코덱(media codec)들은 계층적인 클래스 구조나 상속 개념을 이용하고 여러 단계의 처리 과정을 형성하는 객체들 사이에 메시지를 전달하는 방법을 이용해서 편리하게 구현할 수 있다. MASH Toolkit은 그림 2에서와 같이 멀티미디어 데이터가 여러 개의 소스 객체들에 의해서 생성되고 여러 개의 필터 객체들을 통해서 전달되고 최종적으로는 싱크 객체들에 의해서 이용되고 소모되는 VuSystem의 객체 지향 모델을 확장해서 이용하고 있다. MASH Toolkit은 각 모듈들이 C++와 같은 시스템 프로그래밍 언어를 사용하여 구현되고 이 구현된 모듈들은 OTcl-(Object Tcl)과 같

은 스크립트 언어를 이용해서 구성, 연결, 제어되는 모델로 되어있다.

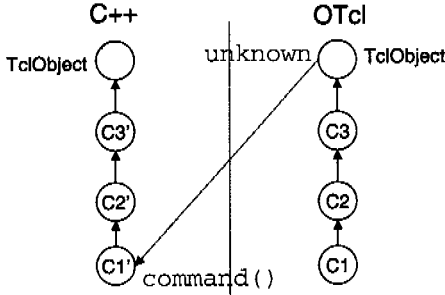


그림 3. OTcl 과 C++ 사이의 메소드 호출

그림 3은 OTcl로 구현된 메소드가 C++로 구현된 메소드를 호출하는 방법을 나타내고 있다. 예를 들어, 호출한 메소드가 OTcl로 구현된 클래스들(C1, C2, C3)에서 발견되지 못하면 OTcl로 구현된 unknown() 메소드가 호출되고, 이 메소드에 의해서 C++로 구현된 객체의 command() 메소드가 자동으로 제어를 넘겨받는다. Tcl이나 OTcl에서 명령어를 확장하는 방법과 마찬가지로 command() 메소드는 C 형태의 스트링 인자들의 벡터를 취해서 해당 메소드를 번역해 내고 적절한 수행을 거친 후 결과를 되돌려 준다. 따라서 하나의 객체는 메소드들이 OTcl과 C++ 두 영역에 걸쳐서 구현될 수 있는 추상적인 엔티티로 볼 수 있다. 즉, C++ 코드가 OTcl로 정의된 메소드들을 호출할 수 있으며 역으로 OTcl 코드는 C++로 정의된 메소드들의 호출이 가능하다.

III. MASH Toolkit을 이용한 시험 시스템

ISO(International Standards Organization)에서는 OSI(Open Systems Interconnection) 프로토콜들의 적합성 시험을 위한 방법론과 골격을 ISO9646 표준으로 정하였다^{15,16)}. 이 표준에 의하면 추상적인 시험 방법으로 지역적 시험(Local Test), 분산 시험(Distributed Test), 상호 조정 시험(Coordinated Test), 원거리 시험(Remote Test) 방법들을 제시하고 있다. 본 연구에서는 ISO9646에서 제시하는 적합성 시험 방법 중 분산 시험 방법을 이용해서 인터넷 프로토콜들의 성능을 시험하는 시스템을 설계하고 구현하였다.

시험 시스템을 설계할 때 고려해야 할 가장 중요

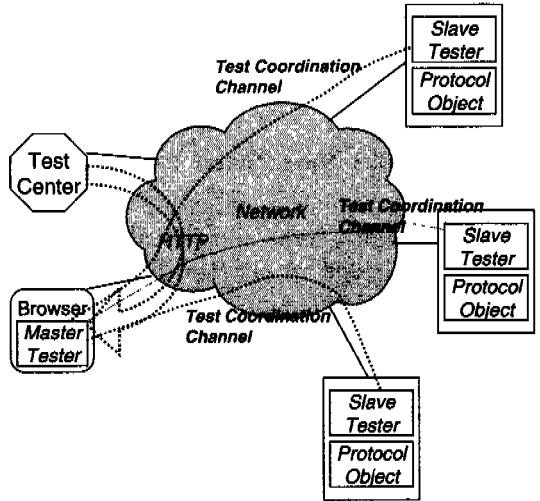


그림 4. 시험 시스템 구조 및 주요 요소들

한 사항중의 하나는 이 시스템이 다양한 목적을 위해서 다양하게 재구성 가능해야 한다는 점이다. 본 연구에서는 이를 위해서 그림 4에서와 같이 다섯 가지의 주요 요소들을 정의하였다.

시험 센터(TC : Test Center) : 웹 브라우저를 통해서 요청될 때 다운로드가 가능한 주 시험기와 부 시험기의 코드를 저장하고 있는 시스템이며 시험 센터로부터 다운로드된 주 시험기 코드와 부 시험기 코드를 수행하는 시스템은 각각 주 시험기 및 부 시험기가 된다. 시험 센터는 주 시험기와 부 시험기에 의해서 액세스가 가능한 시험 스위트(시험 캠페인을 기술한 스크립트 파일)도 유지한다. 시험 센터는 웹 브라우저를 통해서 코드를 액세스할 수 있도록 홈페이지를 제공한다. 즉, 시험 센터는 HTTP 서버로 동작하게 된다. 주 시험기가 동작을 시작하면 시험 센터가 유지하고 있는 시험 스위트들의 리스트를 요청하고 이 리스트에서 선택한 시험 스위트에 대한 코드를 시험 센터에게 요청한다. 이런 요청 및 이에 대한 응답은 HTTP를 통해서 수행된다.

주 시험기(MT : Master Tester) : 독립적으로 수행이 가능한 응용이나 웹 브라우저에 삽입(embedded) 애플릿으로 수행되며 주 시험기가 수행을 시작할 때, 필요한 시험 스위트를 선택해서 다운로드 받을 수 있도록 GUI를 제공한다. 시험 스위트를 다운로드 받은 주 시험기는 시험 스위트에 기술된 대로 파라미터들의 초기화, 부 시험기들과의 시

험 조정 채널들의 설정, 시험 스크립트의 다운로드, 부 시험기들의 수행 제어, 시험 결과의 수집 등과 같은 일을 수행한다.

부 시험기(ST : Slave Tester) : 주 시험기로부터 요청 받은 시험 조정 채널을 기다리는 데몬(Daemon) 프로세스 형태로 수행되며 일단 주 시험기와 채널이 설정되면 부 시험기는 수행 준비, 시험 스크립트의 수신, 시험 스크립트의 수행, 관찰된 결과의 보고등과 같이 주 시험기의 요구사항을 수행한다. 또한 부 시험기는 다운로드된 시험 스크립트의 지시사항에 따라 특정 이벤트를 적용하거나 그 결과를 관찰할 수 있는 능력을 가지고 있다.

시험 조정 채널(TCC : Test Coordination Channel) : 하나의 주 시험기와 하나 이상의 부 시험기들 사이에 설정되는 채널로 주 시험기와 부 시험기들의 시험 캠페인 수행을 조정하는 역할을 담당한다. 필요한 시기에 부 시험기들에게 시험 스크립트의 다운로드, 다운로드 받은 시험 스크립트의 수행, 수행 결과의 수집 등과 같은 조정이 시험 조정 채널을 통해서 이루어진다.

프로토콜 객체(PO : Protocol Object) : RSVP, RTP, SRM 등과 같이 C, C++, OTcl로 구현된 특정 프로토콜에 해당하며 제어를 위해서 API를 제공한다.

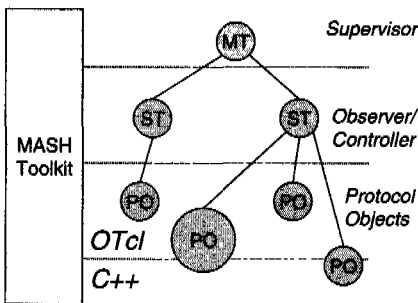


그림 5. MASH Toolkit의 새로운 객체들

MASH Toolkit은 C++나 OTcl 각각으로 구현된 객체들과 C++와 OTcl로 혼합된 형태로 구현된 여러 개의 객체들로 구성되어 있으며 새로운 객체들을 쉽게 삽입할 수 있도록 되어 있다. 이런 MASH Toolkit의 성질을 이용하여 본 연구에서는 위에서 설명한 시험 시스템의 주요 요소들을 MT, ST, PO

등과 같은 객체들로 구현하여 기존의 MASH Toolkit내로 삽입하였다. 그림 5에서는 새로운 객체들이 삽입된 상태에서 각 객체들의 계층 구조를 보여 준다. ST 객체들은 기존 MASH Toolkit의 "Observer" 클래스로부터 상속되고 "Supervisor" 클래스로부터 상속된 MT 객체에 의해서 제어되는 구조를 갖는다. MT나 ST와 같은 각각의 객체들은 하나 이상의 PO들에 대해서 제어가 가능하고 성능 시험을 위한 특정한 성격을 갖는 트래픽의 생성도 용이하다. 또한 MT 객체는 GUI를 포함하며 시험을 수행하는데 필요한 시험 스위트들을 유지하고 시험 결과를 분석하는 기능도 가지고 있어야 한다.

IV. RSVP 시험 시스템

본 연구에서 구현한 시험 시스템은 일반적인 인터넷 프로토콜들을 시험하기에 적절한 시스템이다. 그러나 본 논문에서는 여러 가지 인터넷 프로토콜들 중에서 RSVP를 시험 대상 프로토콜로 택하여 시험 시스템을 적용하여 보았다. 특히, 이 시험 시스템은 RSVP의 기본 시그널링 메커니즘의 시험 이외에도 트래픽을 생성하고 측정하는 기능과 조합되면 종단간(End-to-End) QoS를 측정하는 도구로의 확장도 용이하다.

1. RSVP 시험을 위한 시험 구조

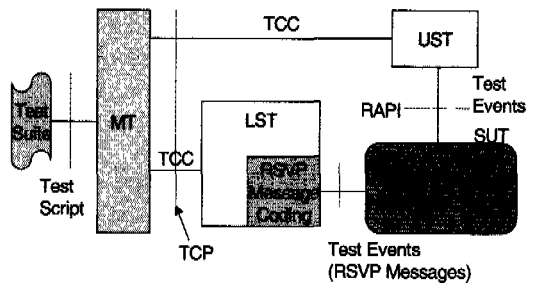


그림 6. RSVP 시험을 위한 일반적인 시험 구조

RSVP 시험 시스템의 일반적인 시험 구조는 그림 6과 같다. 이 구조는 RSVP 시험 시스템을 위한 구조이지만 다른 프로토콜들의 적합성 시험이나 QoS 기반의 성능 시험을 위해서 확장이 용이한 매우 융통성 있는 구조이다. 또한 이 구조는 MT, ST, 시험 대상 시스템(SUT : System Under Test) 등과 같은 세 가지 형태의 모듈로 구성된다. MT는 시험의 전반적인 책임을 담당하는 모듈로써 시험 스위트에

기술되어 있는 시험 행위를 조정하는 역할 및 통신 매커니즘을 통해서 ST들로 시험 명령어(시험 스크립트)를 전달하는 역할을 담당한다. MT와 ST 사이의 조정 및 통신은 그림 6에서 보는 바와 같이 TCC를 통한 시험 조정 절차(TCP : Test Coordination Procedure)에 의해서 이루어진다. MT가 수행해야 하는 기본적인 기능은 다음과 같다.

- 시험 스위트를 읽고 해석
- ST들의 활성화 및 제어
- ST들로부터 수집된 결과들의 수집
- 시험 결과의 분석 및 보고서 생성

SUT는 본 시험 시스템의 대상이 되는 제3자가 구현한 RSVP에 해당한다. RSVP는 호스트나 라우터에서 수행될 수 있는 프로토콜이므로 시험 대상 시스템은 호스트, 라우터 그리고 라우터 모듈들(라우터 클라우드)에서 수행되는 경우를 모두 고려해야 한다. 이 세 가지 경우에서 라우터 클라우드에 대한 시험이 가장 복잡한 형태를 띄게되며 그림 6에서는 이 세 가지 경우를 모두 고려한 일반적인 시험 구조를 보여 준다.

ST의 역할은 SUT에게 MT로부터 받은 시험 명령어들에 의해서 시험 이벤트를 적용하고 이 이벤트에 대한 SUT의 응답을 관찰해서 그 결과를 MT로 전달해 주는 역할을 담당한다. 일반적인 시험 시스템에서는 여러 개의 ST를 필요로 한다. 본 논문에서는 OSI 7계층 모델의 제어 및 관찰점(PCO : Point of Control and Observation)에 따라서 적용할 수 있는 하위 부 시험기(LST : Lower Slave Tester)와 상위 부 시험기(UST : Upper Slave Tester)로 나누어서 시험 시스템을 구성한다. UST는 정의된 인터페이스를 통해서 SUT와 인터랙션이 가능한 일반적인 어플리케이션처럼 동작한다. 여기서의 정의된 인터페이스에는 RAPI(RSVP Application Programming Interface)^[17]나 Winsock Generic QoS Mapping^[18]이 해당된다. 한편 LST는 RSVP 메시지들의 교환을 통해서 SUT와 인터랙션하는 피어 RSVP 엔터티에 해당하므로 RSVP 메시지를 코딩하고 디코딩하는 기능을 가지고 있어야 한다. 따라서 LST는 RSVP의 참조 구현물(Reference Implementation) 형태로 제공되어야 하고 SUT를 검증하기 위한 기본 도구로 사용된다.

ST들의 구현은 특정 시험 구성에서 요구한대로 특정 위치로 다운로드가 용이하도록 가능하면 작은 모듈로 구현되어야 한다. 따라서 본 연구에서는 ST

들을 단순하고 쉽게 다운로드가 가능한 OTcl로 구현하였다. 일단 ST들은 특정 위치로 다운로드된 이후에는 실제로 시험을 담당하는 역할을 하며 TCP를 사용해서 MT와 통신한다.

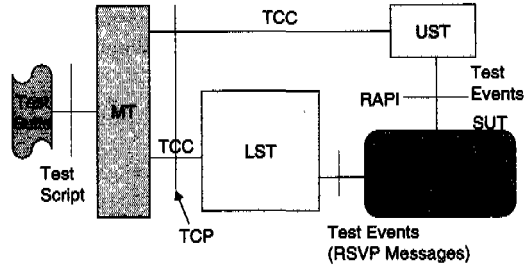


그림 7. 호스트의 RSVP 시험을 위한 구조

그림 7에서는 호스트에서 수행되는 RSVP를 시험하기 위한 구조이다. MT는 시험 경우에 기술된 명령어들을 LST와 UST에게 전달한다. LST는 MT에 의해서 전달받은 명령어에 의해서 RAPI를 통해 세션 설정과 자원 예약 설정을 위해서 시험 대상 시스템인 RSVP와 인터랙션을 하고 LST는 SUT에 의해서 생성되는 메시지들을 받게 된다. RSVP의 자원 예약 기능을 시험하기 위해서는 LST에 트래픽을 생성하는 기능도 포함될 수 있다.

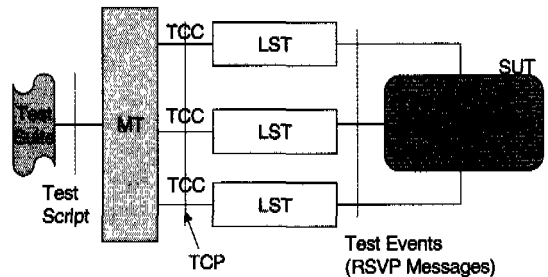


그림 8. 라우터의 RSVP 시험을 위한 구조

그림 8은 RSVP 라우터(RSVP-enabled Router)에서 수행되는 RSVP를 시험하기 위한 구조이다. 라우터의 RSVP를 시험하기 위해서는 라우터의 네트워크 인터페이스 갯수 만큼 ST들을 필요로 한다. 또한 라우터의 경우에는 RAPI를 통해서 RSVP를 제어하기가 불가능하므로 LST들만을 사용한다. MT는 LST들에게 시험 명령어를 전달하고 하나의 LST를 선정해서 시험을 초기화하게 하고 나머지 LST들은 SUT로부터의 반응을 관찰하고 그 결과를 MT에게 전달한다.

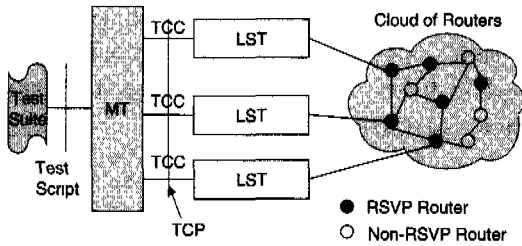


그림 9. 라우터 클라우드에서의 전개 시험

라우터 클라우드의 RSVP를 시험하는 경우는 좀 더 복잡한 형태를 띄게 된다. 시험 환경에 따라서 클라우드 내의 모든 라우터들에 대한 제어가 가능할 수도 있고 단지 몇 개의 라우터에 대해서만 제어가 가능할 수도 있다. 실험실 환경에서의 시험(Lab Testing)은 시험기와 SUT들이 지리적으로 근접해 있고 시험기가 전체 시험에 대한 제어가 쉬운 반면 RSVP 라우터들이 전개가 진행 중이며 운용되고 있는 환경에서의 전개 시험(Deployment Testing)은 다음과 같은 이유들로 인해 부가적인 어려움이 따른다

- 시험하고자 하는 클라우드에 대한 제어 및 관찰 능력이 제한적
- 한 클라우드내에 여러 개의 라우팅 경로 존재 가능
- 시험 결과가 MT와 LST들 사이의 지연에 민감한 반응 가능성
- 시험 결과가 네트워크의 트래픽에 영향 받을 가능성 존재

그림 9는 라우터 클라우드 안의 RSVP를 전개 시험을 위한 시험 구조이며 라우터 클라우드에 대한 액세스는 MT와 미리 설정된 몇 개의 LST들로 제한된다.

2. RSVP 시험 스위트

시험 스위트는 MT와 ST들에 의해서 수행되어야 할 시험 캠페인을 기술한 스크립트 파일에 해당하며 시험 스위트들은 시험 센터에 유지되어 있다가 MT의 요청이 있을 때 MT로 다운로드된다. 각 시험 스위트는 이 시험 스위트의 이름 부분(NAME), 실제 시험 스위트를 기술하는 부분(DESCRIPTION), 파라미터들의 디폴트값을 기술하는 부분(INITIALIZATION), MT와 연결될 ST들을 구성하는 부분(CONFIGURATION), MT에서 수행되어야 할 시험 캠페인을 제어하는 부분(CONTROL)으로

구성된다. 이때 CONTROL 부분은 start, stop, abort등과 같은 시험 조절을 위한 프리미티브들을 갖는 OTcl 스크립트 언어로 쓰여진다. RSVP의 시험을 위해서 정의하고 구현한 프리미티브들은 다음과 같다.

start : 실행하고자 하는 ST들의 리스트 생성, 시험 조절 채널 설정, 새로운 시험 스크립트의 인스턴스 생성, 파라미터들의 세팅등을 수행

stop : ST의 현재 시험 수행을 정상적으로 종료하고 시험 스크립트 큐와 이벤트 큐의 내용을 제거하고 RSVP 세션과 같이 PO에 의해서 만들어진 결과들은 유지

abort : 모든 ST들의 현재 수행을 정상적으로 종료하고 다음 시험 스위트들의 적용은 init 프리미티브의 수행후에 가능

download : ST들에서 수행될 시험 스크립트들을 전송하며 전송전에 각 ST들은 준비 상태(ready state)에 있어야 함

pause/resume : 특정 ST나 모든 ST들의 수행을 일시 중지/재실행

init : 모든 ST들의 실행을 종료시킴과 동시에 모든 큐들을 클리어시키고 MT가 새로 수행을 시작하는 경우에 수행되는 프리미티브로 MT는 기존의 채널들을 모두 제거하고 새로운 채널들을 설정

sync : MT와 특정 ST사이에서 동기화를 위한 마크(임의의 스트링)를 전송

wait : 다른 시험기가 특정 동기화 마크를 전송할 때까지 시험기의 수행을 일시 중지

check : RSVP에 의해서 생성되는 특정 upcall 이벤트를 검사하며 이 이벤트는 Path_Event, Resv_Event, Path_Error, Resv_Error, Resv_Confirm중의 하나가 됨

위와 같은 프리미티브들을 이용해서 RSVP의 기본 동작,merging rule^[4], killer reservation^[19], RSVP 기반의 통합 서비스등을 시험할 수 있는 시험 스위트들을 개발하였다. 이 중에서 RSVP의 기본 기능을 시험하는 시험 스위트는 다음과 같다.

```

NAME RSVPBasicTestSuite
DESCRIPTION {
    This is the testsuite for testing RSVP.
    This checks basic connectivity.
}
INITIALIZATION {
    protocol udp
    
```

```

r 100
b 200
p 429
m 100
M 150
multiaddr 224.224.30.30
multiport 3232
addr 129.6.55.161
port 5555
ttl 10
}
CONFIGURATION {
  ST1 168.188.46.137 3131 sender
  ST2 168.188.46.134 3131 receiver
}
CONTROL {
  start ST1
  start ST2
  download ST1 {
    set s [new RSVP]
    set sid [$s session $protocol $multiaddr
$multiport]
    wait send-ready
    $s sender $sid $port $ttl "t $r $b $p $m
$M"
    check Resv_Event
    $s close
    sync send-done
  }
  download ST2 {
    set r [new RSVP]
    set sid [$r session $protocol $multiaddr
$multiport]
    wait rcv-ready
    check Path_Event
    $r close
    sync rcv-done
  }
  wait ST2 rcv-ready
  sync ST1 send-ready
  wait ST1 send-done
  stop ST1
  wait ST2 rcv-done
  stop ST2
}

```

이 시험 스위트는 RSVP가 두 개의 네트워크 인터페이스를 가지고 있는 라우터에서 수행될 때 기본적인 RSVP의 동작을 시험하는 시험 스위트이다. 라우터가 두 개의 인터페이스를 가지고 있으므로 두 개의 ST(ST1 과 ST2)를 필요로 하며 ST1 과 ST2는 RSVP 세션에 대해서 각각 송신자와 수신자 역할을 담당한다. MT는 ST1, ST2와 각각 시험 조정 채널을 설정하고 download 프리미티브를 사용해서 각 ST들에게 시험 스크립트를 다운로드한다. 일단 시험 스크립트가 다운로드 되면 ST들은 RSVP 프로토콜 객체를 생성하고 세션을 설정한다. 세션

설정후에는 wait, check, sync 등과 같은 프리미티브가 지시하는 내용을 수행한 결과에 의해서 발생하는 특정 이벤트를 관찰한다.

이 시험 스위트는 RSVP의 기본적인 행동만을 시험 하기 위한 것이므로 Tspec 파라미터들인 r, b, p, m, M^[20]가 관찰되는 결과에 영향을 미치지 않지만 RSVP 기반의 통합 서비스 시스템을 시험하는 경우에는 이 파라미터들에 의해서 관찰되는 결과가 중요한 시험 결과로 이용될 수 있다.

3. 그래픽 사용자 인터페이스

MASH Toolkit은 Netscape Communicator와 Microsoft Internet Explorer에서 사용할 수 있는 플러그인 모듈인 MASH Plugin(MPlug) 모듈의 생성이 가능하다. MPlug는 WWW으로부터 MASH 스크립트를 다운로드할 때마다 호출되는 동적으로 로딩되는 라이브러리에 해당한다.

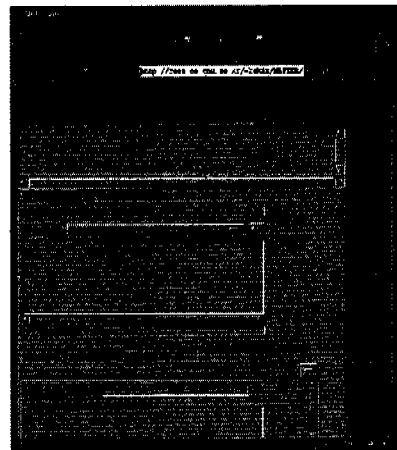


그림 10. MT의 사용자 인터페이스

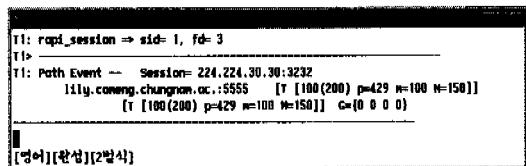


그림 11. UST의 Upcall

현재 구현한 시험 시스템에서 MT는 그림 10에서와 같은 형태로 WWW 브라우저내에서 임베디드(embedded) 애플릿으로 수행된다. MT가 수행을 시작하면 MT는 필요한 시험 스위트를 선택해서 다운로드 받을 수 있도록 사용자 인터페이스를 보여준

다. 그림~ref[fig:ust_gui]은 제 4.2 절에서 기술한 시험 스위트를 적용한 경우에 RSVP 세션의 수신자 역할을 담당하는 UST에 나타나는 upcall의 예를 보여준다.

V. 결론 및 향후 연구

본 논문에서는 지리적으로 분산되어 있는 환경에서, 시험 시스템의 각 구성 요소의 설치 및 제어에 용이한 인터넷 프로토콜의 시험 방법 및 시험 시스템에 대해서 소개하였다. 본 논문에서 언급한 시험 시스템은 분산 시험 스크립팅 환경, 특정한 한 위치에서 지리적으로 분산되어 있는 여러 개의 시험 시스템의 구성 요소들을 제어할 수 있는 능력, 시험 시스템 요소들의 쉬운 재구성, WWW 브라우저를 통해서 시험 시스템 요소들의 다운로드 및 실행을 통한 쉬운 이식성등을 제공한다. 또한 본 논문에서는 구현한 시험 시스템을 RSVP에 적용해 봄으로써 그 가능성을 보였으며 실제 시험 스위트 및 이 시험 스위트의 적용 결과를 제시하였다.

현재 이 시험 시스템은 RSVP를 시험 대상 프로토콜로 하고 있고 RSVP와 관련된 시험 스위트에 초점을 맞추고 있지만 MASH Toolkit이 가지고 있는 확장성을 그대로 가지고 있기 때문에 다른 여러 프로토콜의 시험을 위한 시스템으로의 확장이 용이하다. 현재 이 시험 시스템이 실제 프로토콜의 성능을 측정할 수 있는 도구로의 확장을 위해서는 시험 시스템내에 임의의 트래픽을 생성하고 측정할 수 있는 기능이 포함되어야 한다. 또한 이 시험 시스템의 개발 경험을 통해 현재 인터넷 분야에서 큰 이슈가 되고 있는 MPLS(MultiProtocol Label Switching)^[21,22] 관련 프로토콜에 대한 시험 시스템으로의 확장 및 관련 시험 스위트 개발을 향후 연구 목표로 하고 있다.

참고 문헌

[1] R. Braden, D. Clark, and S. Shenker. Ingetrated Services in the Internet Architecture: an Overview. RFC1633, June 1994.
 [2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. RFC2475, December 1998.

[3] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A New Resource ReSerVation Protocol. IEEE Network, September 1993.
 [4] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification. RFC2205, September 1997.
 [5] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service. RFC2212, September 1997.
 [6] J. Wroclawski. Specification of the Controlled-Load Network Element Service, RFC2211, September 1997.
 [7] Univ. of California, Berkeley. <http://bmr.berkeley.edu/frame/research/cmt>. Continuous Media Toolkit (CMT)
 [8] C. Lindbald and D. Tennenhaus. The VuSystem: A Programming System for Compute-intensive Multimedia. IEEE Journal of Selected Areas in Communications, 14(7), September 1996.
 [9] Lawrence Berkeley Laboratory. Visual Audio Tool.
 [10] S. McCanne and V. Jacobson. vic: A Flexible Framework for Packet Video. In Proc. of ACM Multimedia, pages 511-522, San Francisco, CA, November 1995.
 [11] S. McCanne, E. Brewer, R. Katz, L. Rowe, E. Amir, Y. Chawathe, A. Coopersmith, K. Mayerpatel, S. Raman, A. Schuette, D. Simpson, A. Swan, T. Tung, D. Wu, and B. Smith. Toward a Common Infrastructure for Multimedia-networking Middleware. In Proc. of 7th Intl. Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '97), St. Louis, Missouri, May 1997.
 [12] S. Floyd, V. Jacobson, S. McCanne, C. Liu, and L. Zhang. A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing. In Proc. SIGCOMM'95, September 1995.
 [13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for

Real-Time Applications, RFC1889, January 1996.

[14] R. Malpani and L. Rowe. Floor Control for Large-Scale Mbone Seminars. In Proc. of The Fifth Annual ACM Intl. Multimedia Conf., November 1997.

[15] Information Processing Systems - Open System Interconnection. ISO International Standard 9646: OSI Conformance Testing Methodology and Framework.

[16] K. Knightson. OSI Protocol Conformance Testing - IS 9646 Explained. McGraw-Hill, Inc., 1993.

[17] R. Bradner and D. Hoffman. RAPI - An RSVP Application Programming Interface : Version 5. draft-ietf-rsvp-rapi-01.txt, 1998.

[18] Y. Bernet. Winsock Generic QoS Mapping (DRAFT). Microsoft, 1997.

[19] M. Talwar. RSVP Killer Reservations. draft-talwar-rsvp-kr-01.txt, January 1999.

[20] J. Wroclawski. The Use of RSVP with IETF Integrated Services. RFC2210, September 1997.

[21] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. draft-ietf-mpls-arch-06.txt, August 1999.

[22] R. Callon, P. Doolan, N. Feldman, A. Fredette, and A. Viswanathan. A Framework for Multiprotocol Label Switching. draft-ietf-mpls-framework-05.txt, September 1999.

전 우 직(Woojik Chun)
 1982년 2월 : 서울대학교 컴퓨터공학과 졸업(공학사)
 1984년 2월 : 서울대학교 대학원 컴퓨터공학과 졸업
 (공학석사)
 1984년~1987년 : 한국전자통신연구원 데이터통신 연구실
 1992년 5월 : Univ. of Delaware 전산학과(공학박사)
 1992년~1993년 : 한국전자통신연구원 정보통신표준 센터
 1993년~현재 : 충남대학교 컴퓨터공학과 부교수
 1997년 8월~1998년 7월 : 미국 NIST Guest
 Researcher
 2000년~현재 : (주) 라오넷 대표

김 별 식(Byeongsik Kim)
 1993년 2월 : 충남대학교 컴퓨터공학과 졸업 (공학사)
 1995년 2월 : 충남대학교 대학원 컴퓨터공학과 졸업
 (공학석사)
 1997년 8월~1998년 7월 : 미국 NIST Guest
 Researcher
 1999년 2월 : 충남대학교 대학원 컴퓨터공학과 박사
 과정 수료
 <주관심 분야> 프로토콜 시험, 인터넷 QoS, MPLS망