

RSVP 분산 시험 시스템 *

김병식[○] 전우직
 충남대학교 컴퓨터공학과
 {bskim, chun}@ce.cnu.ac.kr

Distributed Test System for RSVP

Byeongsik Kim[○] Woojik Chun
 Department of Computer Engineering, Chungnam National University

요약

라우팅 및 시그널링 프로토콜의 설계와 개발과정에서 가장 어려운 이슈들중의 하나가 개발된 프로토콜의 시험 및 성능을 측정하는 것이다. 이를 위해서는 시험 요소들의 구성 및 제어가 용이하면서 지리적으로 분산되어 있는 상태에서 시험을 가능하게 하는 방법과 도구가 필요하다. 본 논문에서는 인터넷 프로토콜 자체의 행동 및 성능 시험을 위한 WWW 기반의 분산 시험 시스템의 설계 및 개발에 관련된 내용을 다룬다. 특히 이 시험 시스템을 RSVP에 적용하는 방법과 RSVP에 관련된 시험 스위트들에 대해서 소개한다.

1. 서론

인터넷에서 부하제어형 서비스 [1]와 보장형 서비스 [2]와 같은 통합 서비스 [3]를 지원하기 위해서 IETF 및 여러 연구 기관에서는 전반적인 프레임워크, 프로토콜, 표준들을 개발해 오고 있다. 이런 연구들의 결과중의 하나가 RSVP (Resource Reservation Protocol) 이다 [4]. 이처럼 통합 서비스의 지원을 위한 새로운 프로토콜의 개발과 함께 프로토콜에 대한 기능 및 성능 시험을 위한 새로운 도구들의 개발도 진행되어야 한다. 본 논문에서는 구성 및 제어가 용이하도록 지리적으로 분산되어 있는 상태에서 프로토콜들을 시험할 수 있는 방법론과 도구의 개발에 대해서 언급한다. 또한 현재 여러가지 인터넷 관련 프로토콜들 중에서 RSVP에 대한 기능 및 성능 시험을 위한 시험 시스템에 초점을 둔다. 그러나 이 시험 시스템은 제어가 용이한 실험실 환경에서부터 실제 운용중인 망에 까지 쉽게 적용할 수 있는 여러가지 실험 구성을 허용하므로 상당한 융통성을 제공하며 다른 대상의 프로토콜에도 쉽게 적용이 가능하다는 장점을 가지고 있다.

이를 위해서 본 논문에서 제안하는 시험 시스템은 제 2.절에서 설명할 MASH Toolkit [5]을 기반으로 구현된다. 본 논문의 구성은 다음과 같다. 제 2.절에서 MASH Toolkit의 개요에 대해서 설명한다. 제 3.절에서는 MASH Toolkit에서 프로토콜 시험 시스템을 구현할 때의 일반적인 방법 및 필요한 각 요소와 기능들에 대해서 소개한다. 제 4.절에서는 여러가지 인터넷 관련 시그널링 프로토콜들 중에서 RSVP의 시험을 위한 시험 시스템의 구현과 시험 스위트의 개발 및 적용 결과에 대해서 소개하고, 제 5.절에서 결론 및 더 진행되어야 할 연구 과제에 대해서 기술한다.

2. MASH Toolkit 개요

버클리 대학의 MASH 연구팀에서는 DARPA, Intel, Microsoft, Xerox PARC의 지원하에 이질적인 환경에서 확장성이 좋은 멀티미디어 구조 (Multimedia Architecture that Scales across Heterogeneous environments)를 위한 연구를 진행 중이며 이를

*본 연구는 미국 표준 기술 연구소 (NIST)와의 공동연구 결과임.

MASH 프로젝트라 한다. MASH 프로젝트의 목표들 중의 하나는 인터넷상에서 멀티캐스트 기반의 공동연구를 위한 응용들을 설계, 개발, 배포하는데 있으며 이의 중간 연구 결과물로 MASH Toolkit을 개발해서 배포중에 있다.

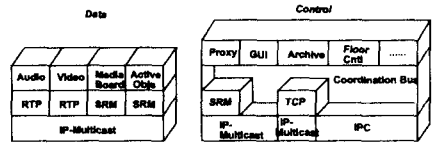


그림 1: MASH Toolkit의 구조

MASH 연구팀은 IP 멀티캐스트를 사용해서 멀티미디어 통신 및 공동 연구를 수행하기 위한 포괄적인 구조를 그림 1와 같이 정의하였으며 이 구조는 서로 밀접한 관련이 있는 여러가지 요소들로 구성되어 있다.

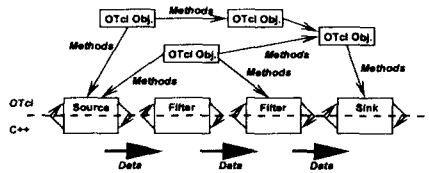


그림 2: MASH의 소프트웨어 모델

MASH의 소프트웨어 구조는 그림 2에서 보듯이 여러개의 멀티미디어 데이터가 여러개의 소스 객체들에 의해서 생성되고, 여러개의 필터 객체들을 통해서 전달되어서 최종적으로는 싱크 객체들에서 이용, 소모되는 모델로 되어있다. MASH Toolkit은 OTel(Object-oriented Tcl Extension) [6]을 기반으로 구현되어 있으며, 그 하부구조인 실시간 미디어 스트림들의 생성, 소비, 필터링은 C++로 구현된 클래스로 정의되어 있다. 또 필요한 경우에는 OTel 객체에서 C++로 구현된 메소드들의 호출도 가능하다.

3. MASH Toolkit에서의 시험 시스템

OSI 프로토콜의 적합성 시험 방법과 골격을 다루는 표준인 IS 9646 [7]은 OSI 프로토콜 이외에도 Q.2931 등과 같은 다른 통신 프로토콜로의 적용도 가능하다. 본 연구의 시험 시스템 설계에 있어서의 주요 원칙은 다양한 목적을 갖도록 시스템을 설계하는 데 있다. 또, 인터넷 프로토콜에 대한 시험 시스템의 재구성성이 용이하게 하기 위해서 그림 3과 같이 시험센터 (TC: Test Center), 주시험기 (MT: Master Tester), 부시험기 (ST: Slave Tester), 시험조정채널 (TCC: Test Coordination Channel), 프로토콜 객체 (PO: Protocol Object) 등과 같은 다섯가지의 주요 요소들을 정의하였다.

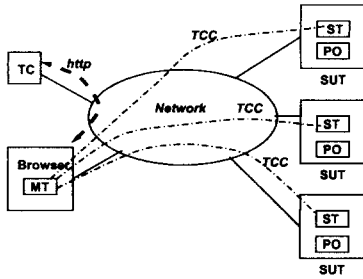


그림 3: 인터넷 프로토콜의 시험 구조 및 요소들간의 관계

TC는 WWW 브라우저를 통해서 요청된 MT와 ST의 코드와 시험 캠페인을 기술하는 스크립트 파일에 해당하는 시험 스위트를 저장하는 저장소의 역할을 한다. MT 코드와 ST 코드를 다운로드 받아서 실행을 시작한 시스템은 각각 MT와 ST로 동작한다. 또한 TC는 WWW 브라우저를 통해서 코드를 접근할 수 있도록 홈페이지를 제공하는 HTTP 서버의 역할을 한다. MT는 standalone 형태의 응용이나 WWW 브라우저에 embedded applet 형태로 수행된다. MT는 수행을 시작하면 GUI를 이용해서 시험 스위트를 선택해서 다운로드 받고 시험 스위트에 기술된 대로 파라미터들의 초기화, ST들과의 TCC 설정, ST로의 시험 스위트 다운로드, 시험 결과의 수집, 시험 결과 보고서의 생성 등과 같은 시험 캠페인을 수행한다. ST들은 MT에 의해서 요청되는 TCC를 기다리고 있는 디몬 프로세스 형태로 수행되며 일단 MT와 채널이 설정되면 수행 준비, 시험 스크립트의 수신 및 수행, MT와의 동기화, 관찰된 결과의 보고 등과 일을 MT에 의해서 지시받은 대로 수행한다. TCC는 MT와 하나 이상의 ST들 사이에 설정되는 채널로서 이 채널을 통해서 하나의 MT와 하나 이상의 ST들 사이의 시험 조정 작업이 이루어진다. PO는 RSVP, RTP, SRM 등과 같은 특정 프로토콜로 이 제어들을 위해서 API를 제공하고 C/C++ 혹은 OTcl로 구현된다.

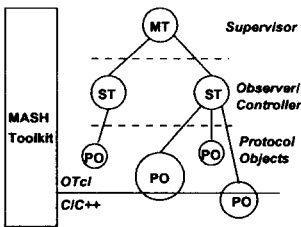


그림 4: MASH Toolkit로 새로운 객체의 추가

MASH Toolkit은 자체가 OTcl/C++ 객체들로 구현되어 있어서 이런 기본적인 객체들 이외에도 새로운 객체들의 추가가 아

주 쉽게 되어있다. 위에서 설명한 다섯가지 요소들을 객체로 구현해서 MASH Toolkit에 추가하면 그림 4과 같은 새로운 객체들의 계층구조를 형성하게 된다.

4. RSVP 시험 시스템

본 논문에서의 시험 시스템은 RSVP 구현물에 대한 기능상의 시험에 사용될 수 있고, 부하를 생성하는 기능과 성능을 측정할 수 있는 기능이 합해지는 경우에는 RSVP를 기반으로 하는 통합서비스의 지원 여부를 시험하는 도구로의 확장이 가능하다. 본 절에서는 이전에 설명한 MASH Toolkit에서 프로토콜 시험을 위해서 새로운 객체를 추가한 시험 시스템을 실제 RSVP에 적용하는 방법 및 시험 스위트에 대해서 설명한다.

4.1 RSVP 시험 시스템의 시험 구조

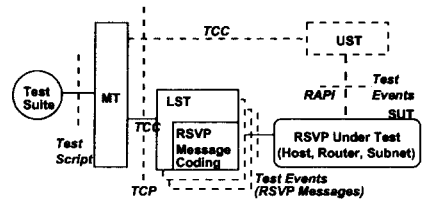


그림 5: RSVP를 위한 시험 구조

RSVP를 시험하기 위한 일반적인 시험 구조는 그림 5와 같다. 이때 시험 대상 시스템 (SUT: System Under Test)이 시험의 대상이 되는 RSVP 구현물에 해당하고 이 SUT는 호스트 측, 송신자나 수신자로 수행되는 단말에서 수행되는 경우일 수도 있고 라우터 혹은 라우터들의 모임으로 생각할 수도 있다. SUT가 여러 라우터들의 모임인 경우에 이 라우터들은 RSVP가 수행되는 라우터들과 수행되지 않는 라우터들이 혼재하는 경우에 시험상의 복잡한 문제를 신중하게 고려해야 할 필요가 있다.

ST의 역할은 시험 스위트에 기술된 내용을 SUT로 적용하고 SUT로부터 그 적용 결과를 받아서 MT로 보고하는 일이다. 상황에 따라서 프로토콜 시험을 위해서는 여러개의 ST들을 필요로 하는 경우가 있다. 예를 들면, OSI 7 계층 모델의 PCO에 따라서 ST는 LST(Lower ST)와 UST(Upper ST)로 나누어지기도 한다. UST는 특정 인터페이스를 통해서 SUT와 인터랙션을 할 수 있는 응용에 해당하고, 이 인터페이스는 SUT가 RSVP인 경우에 RAPI나 Winsock Generic QoS Mapping에 해당한다. 한편, LST는 RSVP 메시지의 교환을 통해서 SUT와 인터랙션하는 피어 RSVP 엔티티에 해당하며 RSVP의 참조 구현물 (reference implementation)로 생각할 수 있으며, SUT의 검증들을 위한 기본으로 사용된다.

라우터로 동작하는 RSVP나 라우터들의 모임에 있어서 RSVP를 시험하는 경우에는 UST가 직접 RAPI에 대한 제어권을 갖지 못하므로 UST의 사용은 필요하지 않다. 이런 경우에는 라우터가 가지고 있는 네트워크 인터페이스의 갯수에 해당하는 LST들과 MT만을 가지고 시험을 수행한다.

ST들은 쉽게 다운로드 되고 특정한 구성이 쉽게 될 수 있도록 가급적 경량화 및 특정 위치로의 다운로드가 가능한 모듈로 구현이 되어야 하므로 본 연구에서는 간단하고 이식성이 좋은 OTcl로 구현하였다.

4.2 RSVP 시험을 위한 시험 스위트

시험 스위트란 MT와 ST들에 의해서 수행되어야 할 시험 캠페

인을 기술한 스크립트 파일에 해당한다. 시험 스위트는 NAME, DESCRIPTION, INITIALIZATION, CONFIGURATION, CONTROL 부분으로 이루어져 있다.

```

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
}
    
```

위의 예는 현재까지 개발한 RSVP 기본 기능, Merging 규칙, Killer Reservation등을 시험 하는 시험 스위트들중에서 기본 기능을 시험하는 시험 스위트에 해당하며, CONTROL 부분에서 현재 정의되어 있는 프리미티브들의 예를 볼 수 있다.

4.3 Graphic User Interface

MASH Toolkit은 Netscape나 Microsoft Explorer를 위한 플러그인 모듈을 제공하며 이를 MASH Plugin 혹은 MPlug라 한다. MPlug는 WWW을 통해서 MASH 스크립트들을 다운로드 받을 때 자동으로 로딩되는 라이브러리에 해당한다. MT는 그림 6에서 보는 바와 같이 WWW 브라우저내에서 embedded 애플릿으로 실행된다. MT는 실행을 시작함과 동시에 다운로드 보고서 하는 시험 스위트를 선택할 수 있는 GUI를 보여주고 선택된 시험 스위트에 따라서 시험 캠페인을 실행하게 된다.

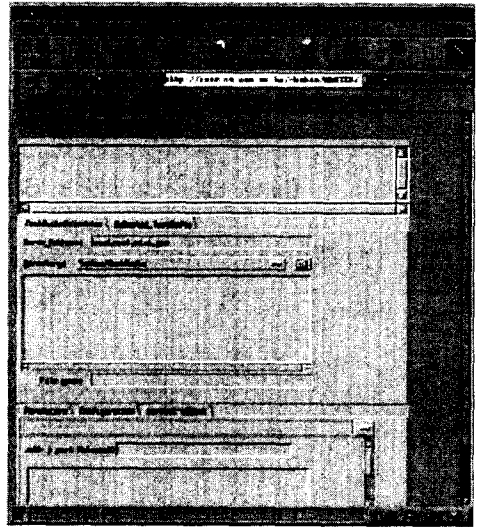


그림 6: MT의 GUI

5. 결론 및 향후연구

본 논문에서는 지역적으로 분산되어 있는 환경에서 시험 시스템의 재어가 용이한 시험 시스템에 대해 소개하였으며 이를 위해서 MASH Toolkit을 이용한 시험 시스템의 구현 방법과 RSVP를 대상으로 하는 시험 방법과 결과를 제시하였다.

향후 연구로는 RSVP 기반의 통합 서비스 제공 시스템으로의 적용을 위해서 트래픽을 생성하고 모니터링하는 툴과의 통합과 이와 관련된 시험 스위트들의 개발이 있으며, 더 나아가서는 다른 시그널링 프로토콜이나 라우팅 프로토콜로의 확장을 목표로 하고 있다.

참고 문헌

- [1] J. Wroclawski, "Specification of the Controlled-Load Network Element Service", RFC2211, September 1997.
- [2] S. Shenker, C. Patridge, and R. Guérin, "Specification of Guaranteed Quality of Service", RFC2212, September 1997.
- [3] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC1633, June 1994.
- [4] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol(RSVP) - Version1 Functional Specification", RFC2205, September 1997.
- [5] S. McCanne, E. Brewer, R. Katz, L. Rowe, E. Amir, Y. Chawathe, A. Coopersmith, K. Mayer-patel, S. Raman, A. Schuett, D. Simpson, A. Swan, T. Tung, D. Wu, and B. Smith, "Toward a Common Infrastructure for Multimedia-Networking Middleware", In Proc. 7th Intl. Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSS-DAV '97), St. Louis, Missouri, May 1997.
- [6] D. Wetherall and C. Lindblad, "Extending Tcl for Dynamic Object-Oriented Programming", In Proc. the Tcl/Tk Workshop, Ontario, Canada, July 1995.
- [7] "Information Processing Systems - Open System Interconnection", ISO International Standard 9646: OSI Conformance Testing Methodology and Framework.