

# 네트워크 기반 원격 협업 지원을 위한 End-to-End IP 멀티캐스트 모니터링 프레임워크 연구

조 진 용<sup>†</sup> · 공 정 욱<sup>†</sup>

## 요 약

IP 멀티캐스트는 네트워크 자원을 효율적으로 사용할 수 있는 장점에도 불구하고, 관리도메인 간 정책적 이견 및 멀티캐스트 가용성과 도달성 등 기술적 문제로 인해 활용 확장에 한계를 가지고 있다. 멀티캐스트 오류 등 기술적 문제에 대한 탐지 및 차단은 IP 멀티캐스트의 안정화를 위해 우선적으로 해결되어야 할 과제이다. 본 논문은 네트워크와 네트워킹 환경에 대한 오류의 탐지 및 원인 추정이 가능하며, 효율적 활용·유지·확장을 위해 경편성(portability)과 관리성(manageability)을 갖춘 멀티캐스트 모니터링 프레임워크를 제안한다. 제안된 프레임워크는 IP 멀티캐스트의 가용성, 도달성 및 대화성의 탐지가 가능하며 백그라운드(background) 시험 및 요구형(on-demand) 오류 검증을 통해 네트워킹 환경의 성능 장애 요소들을 추정할 수 있다. 또한, 네트워크 자원을 효율적으로 활용함으로써 프레임워크의 확장성(scalability)을 제공한다. 네트워크 테스트베드에서 행해진 실험을 통해 제안된 프레임워크 확장성과 효율성을 입증한다.

키워드 : IP 멀티캐스트, 멀티캐스트 모니터링, 원격 협업

## Realizing an End-to-End IP Multicast Monitoring Framework for the Support of Networked Collaboration

Jinyong Jo<sup>†</sup> · JongUk Kong<sup>†</sup>

## ABSTRACT

Despite its graceful bandwidth saving nature, IP multicast has not been successful in widely spreading and using. Challenging includes policy and technological issues such as inter-domain routing, multicast availability and reachability. Properly detecting and isolating the faults would be the first step to stabilize IP multicast. In this paper, we introduce a scalable multicast monitoring framework. To efficiently cope with multicast network and networking problems, it enlarges its monitoring scopes from collecting delivery statistics to verifying end-to-end multicast availability, reachability and interactivity. We carry out various networking experiments to verify the scalability and feasibility of the proposed framework.

Keywords : IP Multicast, Multicast Monitoring, Networked Collaboration

### 1. 서 론

고성능 컴퓨팅 자원과 첨단 네트워크를 활용하는 원격 협업은 효율적인 그룹 통신을 필요로 한다. 협업의 영역이 의학 분야 [1]까지 확대됨에 따라 멀티캐스트 환경 하에서 고품질 미디어 활용에 대한 관심이 높아지고 있다. 액세스 그리드 [2], ConferenceXP 및 DVTS(digital video over IP) 등은 이러한 사용자 요구에 부합하는 기술로써 다자간 그룹 통신과 데이터 공유 등의 기능을 IP 멀티캐스트를 통해 사용자에게 제공하는 서비스이다.

다양한 협업 응용이 활성화되고 IPTV와 같은 수익모델

이 창출되기 위해서는 멀티캐스트 인프라의 안정화가 필수적이다. 하지만, 관리 도메인 간 정책적 이견 및 전송망 관리의 어려움 [26] 등으로 인해 오버레이 네트워크나 P2P 기술들을 통해 IP 멀티캐스트의 문제점들을 보완하려는 노력이 계속되고 있는 실정이다. 멀티캐스트 인프라의 안정화와 협업 응용의 활용 촉진을 위해 효과적인 IP 멀티캐스트 모니터링 프레임워크가 요구된다.

IP 멀티캐스트의 장애 요소들을 탐지하기 위해서는 가용성(availability), 도달성(reachability) 및 대화성(interactivity) 등이 파악되어야 한다. 멀티캐스트 장애 탐지를 위해 다양한 연구들이 진행되어 왔지만, 패킷 전달 통계의 수집 및 제공에 초점을 둬서 네트워크 접속단과 타 관리도메인(administrative domain)에서 발생하는 오류 탐지에 제한이 있다. 또한, 협업 응용이 여러 이질적 사용자 그룹에 의해

<sup>†</sup> 정 회 원 : 한국과학기술정보연구원 슈퍼컴퓨팅본부 선임연구원  
논문접수 : 2009년 6월 25일  
수정일 : 1차 2009년 9월 1일  
심사완료 : 2009년 9월 14일

활용된다는 점을 감안할 때, 모니터링 프레임워크는 시스템 측면, 유니캐스트 브리징과 멀티캐스트 가용성 등 중단간 네트워킹(networking) 문제를 수용해야 한다.

전송 과정에서 수집한 통계는 네트워크 혼잡(congestion) 등의 증상을 의미하기 때문에 멀티캐스트 오류를 차단하는데 충분한 정보로써 활용되기 힘들다. mtrace [6] 등을 활용한 오류 탐지는 관리도메인 내·외부(intra-/inter-domain)의 멀티캐스트 도달성 문제를 검증하는데 있어 효과적이다. 일반적으로 단대단(end-to-end) 통신 범위가 여러 네트워크 관리도메인을 포함하며, 다수의 서비스 제공자들이 각각의 도메인을 관리하기 때문에, 멀티캐스트 모니터링에 있어 대화성(또는, 양방향성)의 확인은 매우 중요하다.

액세스 그리드와 같은 원격 협업 응용의 운영자들이 멀티캐스트 문제해결(trouble-shooting)을 위해 취하는 일반적인 절차는 현상인지, 자가측정, 상황보고, 오류수정 등의 단계로 요약된다. 현상인지의 단계는 음성과 영상 등 멀티미디어의 인식 품질(perception quality)과 특이 현상(예, 특정 미디어의 수신 불가능, 단방향 수신에 국한해 가능 등)에 의해 파악된다. 자가측정의 단계에서는 멀티캐스트 문제탐지를 위해 개발된 다양한 독립형 소프트웨어 도구(stand-alone software tool)들이 이용되며 정보 수집을 위해 문제가 인지된 응용 사용자들 간에 상호 측정된다. 수집된 정보들은 NOC(network operation center)에 보고된 후 최종적으로 문제가 수정된다.

자가측정 및 상황보고의 단계는 절차의 순서변화와 관계 없이 중단 사용자 간 상호 측정이 요구된다. 즉, 응용 지원의 측면에서 볼 때 멀티캐스트 프레임워크는 중단 호스트를 포함하는 단대단 간 자가측정이 가능한 구조를 가져야 한다. 또한, 멀티캐스트의 특성 상 그룹 커뮤니케이션이 가능하고 모니터링 세션에 대해서 효과적으로 제어할 수 있는 구조가 필요하다.

자가측정의 단계는 그룹 참가자들 간에 e-mail 등을 이용해 시간 순차적으로 수행되기 때문에 측정 작업이 시간 소모적(time-consuming)이다. 특히, 참가자들의 지리적 위치에 시차가 존재할 경우, 메시지 교환에 장시간이 요구되므로 신속한 오류수정이 어렵다. 따라서, 멀티캐스트 모니터링 프레임워크는 응용 이용자의 요구가 있는 즉시(on-demand) 실시간으로 자가측정할 수 있는 구조를 가지고 다양한 독립형 소프트웨어 도구를 수용 가능해야 한다. 즉, 프레임워크를 구성하는 컴포넌트가 중단 호스트에 쉽게 이식될 수 있는 경편성(portability)과 지속적인 도구의 제공 및 프레임워크의 갱신 등이 용이하도록 전체 프레임워크에 대한 관리성을 높여야 한다.

원격 협업 응용이 다양한 멀티미디어를 활용한다는 점을 고려하면 모니터링 프레임워크는 IP 멀티캐스트 프로토콜 또는 라우팅 등에 의해 발생하는 문제를 포함해 멀티미디어의 트래픽 특성이나 네트워킹 환경 등이 야기하는 문제를 탐지할 수 있는 방안을 제공해야 한다. 라우팅 루프(loop)나 NAT(network address translation)의 사용 등은 일반적으로

멀티미디어 트래픽의 양방향 전송을 불가능(또는, 단방향 전송만 가능)하게 만들 수 있다.

HD(high-definition) 영상이 원격 협업에 적용되면서 수십 Mbps의 대역폭을 필요(예, MPEG2 MP@HL)로 하는 응용들이 활용되고 있다. 고화질 영상을 활용하는 응용이 발생시키는 트래픽은 폭주(burst)하는 특성이 있어 저사양의 네트워크 허브(hub) 등을 이용해 중단 호스트에 연동할 경우 성능 문제로 인해 전송율(throughput)이 크게 낮아질 수 있다. 방화벽의 사용 역시 전송 성능을 저하시키는 경향이 있다. 낮은 전송율은 악화된 인식 품질을 의미한다. 원활한 원격 협업 등 응용 지원의 입장에서, 멀티캐스트 모니터링 프레임워크는 멀티미디어의 트래픽 특성을 적용한 자가측정 방법을 제공해야 한다.

결론적으로, 응용 지원을 위한 모니터링 프레임워크의 설계는 네트워크와 네트워킹 환경에 대해서 종합적으로 고려되어야 한다. 본 연구는 효과적인 멀티캐스트 네트워크 및 네트워킹 환경의 오류탐지를 위해 중앙 관리 서버와 모니터링 클라이언트들로 구성된 IP 멀티캐스트 모니터링 프레임워크를 제안한다. 제안된 프레임워크를 통해 단대단(end-to-network), 또는 단대단 간 오류의 원인 분석, 패킷 전달 경로 상의 통계 수집 등을 통해 IP 멀티캐스트 네트워킹의 가용성, 도달성 및 대화성을 확인할 수 있다.

본 논문의 2장에서는 관련 연구들을 조사하고, 3장에서 멀티캐스트 모니터링을 위해 제안된 프레임워크의 세부구조를 살펴본다. 4장에서 프레임워크 확장성 등 성능 측정을 수행하고 5장에서 결론을 맺는다.

## 2. 관련 연구

본 절에서는 기존에 제안된 멀티캐스트 모니터링 프레임워크에 대해서 살펴본다. mtrace [6], mcping 및 mctrace [8, 9], mHealth [10], MMag [11], RTPmon [12], RMON [13], mantra [14], MRMON [15, 16] 등 다양한 연구 결과들이 존재하지만 본 논문과 많은 유사성을 갖는 Multicast beacon, SMRM, MRM 및 MQM에 초점을 둔다.

Multicast beacon [17]은 분산 클라이언트들과 중앙 서버로 구성된다. 클라이언트들은 모니터링 세션에 참가한 동료 클라이언트들과 RTP(real-time transport protocol [18]) 패킷을 교환함으로써 전달 통계를 수집한다. 수집된 패킷 전달 통계는 손실, 지연, 지연 변이(jitter), 비순차 도착(out-of-ordered arrival) 및 중복된 패킷의 수 등을 포함한다. 웹(Web)을 통한 사용자 인터페이스가 제공된다. 단점으로는 멀티캐스트 가용성 및 연결 제약 등의 문제를 탐지할 수 없다는 점이다. 또한, 능동 탐지(active probing)를 위해 사용하는 RTP 트래픽이 제어 불가능하므로 네트워크에 부정적인 영향을 줄 수 있다.

SMRM [19]은 기존 SNMP [17]를 이용한 모니터링 환경 하에서도 능동 탐지가 가능한 구조이다. SMRM도 multicast beacon과 동일하게 중앙 서버와 다수의 분산 클라이언트로

구성되지만, 상대적으로 다양한 모니터링 기능들을 제공한다. SMRM이 제공하는 사후 분석(postmortem analysis)은 능동 탐지를 위한 트래픽 특성을 임의 설정할 수 있다. 하지만 제어 평면(control plane)이 멀티캐스트 세션으로부터 충분히 분리되지 못했기 때문에 멀티캐스트 가용성이 없는 클라이언트들이 고립될 수 있다. 또한, 의사(pseudo) SNMP가 네트워크 스위치나 라우터에 직접 적용되면 해당 전송 장비들에 처리 부하를 가중시킬 가능성이 높고 관리도메인의 확장에 한계가 있다.

사용자 측면에서 멀티캐스트 도달성을 확인하기 위해 MRM(multicast reachability monitor [20])이 개발되었다. MRM은 몇몇 Cisco<sup>TM</sup> 라우터들에게 적용되었으며, 테스트 송신자, 테스트 수신자 및 중앙 관리자로 구성된다. MRM은 테스트 수신자에게 통계 보고 및 통계 수집 방법 등을 설정할 수 있고, 테스트 송신자가 능동 탐지를 위한 패킷 스트리밍 스케줄을 설정할 수 있다. MQM (multicast quality monitor [21])은 MRM과 내부 구조가 유사하며 대화성 모니터링을 위해 개발되었다. 클라이언트들 간 상호 탐지(inter-probing)를 위한 절차(handshaking), 도달성 및 품질 모니터링 방법 등 세부적인 프레임워크가 MRM과 구별된다.

본 논문에서 제안한 프레임워크는 내부 구조, 다양한 탐지 방법의 제공, 탐지 영역의 단대망 확장, 향상된 접근성 제공 등에서 기존 연구와 차이점을 갖는다. 또한, 백그라운드 시험(background test)을 위한 트래픽 양을 최소화하고 요구형(on-demand) 시험을 강화함으로써 네트워크 대역폭을 효율적으로 사용한다. 프레임워크를 통해 사용자는 접속망에서 멀티캐스트 지원여부, 멀티캐스트 응답요구(ping-pong), 라우팅 패스 추적, 멀티캐스트 제약사항 탐지 및 중단 간 패킷 전달 통계를 수집할 수 있으며 모든 사용자 명령의 실행과 탐지 결과가 웹 인터페이스를 통해 확인된다.

### 3. 멀티캐스트 모니터링 프레임워크

제안된 프레임워크를 살펴보기에 앞서 본 논문에서 사용될 가용성(availability), 도달성(reachability) 및 대화성(interactivity)에 관한 용어를 정의한다. IP 멀티캐스트의 연결 관계는 방향성 그래프  $G=(N,L,g)$ 로 표시될 수 있다.  $N$ 은 중단 호스트  $e$  및 라우터  $r$ 의 집합이며,  $L$ 은  $e$ 와  $r$  또는  $r$ 과  $r$ 을 연결하는 링크  $l$ 의 집합이다.  $g$ 는  $\exists x,y,z \in N$ 인 두  $x$ 와  $y$ 에 대해서 링크  $l$  또는  $l$ 과  $z$ 의 집합을 사상하는 함수로써 방향성을 갖는 순서쌍  $(x,y)$ 로 정의한다. 즉, 경로  $e_0,l_0,r_1,l_1,\dots,l_{k-1},r_k$ 에 대해  $e_0$ 에서  $r_k$ 로 연결성을 가질 때(패킷이 전달 될 때) 순서쌍  $(e_0,r_k)$ 로 나타낸다.

• 가용성

단대망(end-to-network)의 관점에서 멀티캐스트 패킷의 송·수신 가능성을 의미한다. 즉, 임의의  $e_i \in N$ 와 인접한

$r_{i+1} \in N$ 에 대해서  $(e_i,r_{i+1}) \wedge (r_{i+1},e_i)$ 의 특성을 가질 때  $e_i$ 은  $r_{i+1}$ 에 대해 가용성을 갖는다.

• 도달성

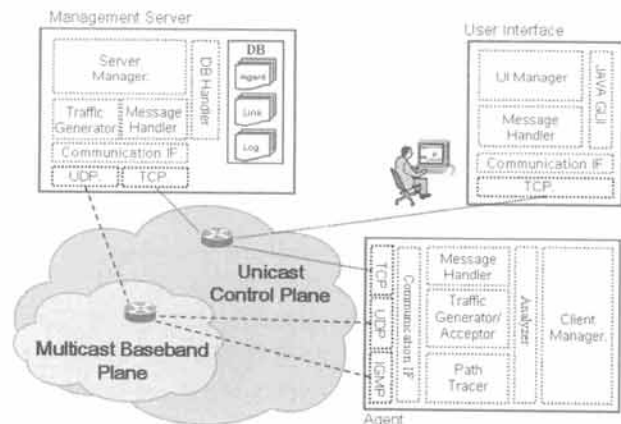
단대단(end-to-end)의 관점에서 중단 간 멀티캐스트 패킷의 단방향 전송 가능성을 의미한다. 즉, 멀티캐스트 그룹에 속한  $\exists e_i, e_j \in N$ 에 대해서  $(e_i, e_j)$ 의 특성을 가질 때  $e_i$ 에서  $e_j$ 로의 도달성을 갖는다.

• 대화성

단대단의 관점에서 중단 간 멀티캐스트 패킷의 양방향 도달성을 의미한다. 즉, 멀티캐스트 그룹에 속한  $\exists e_i, e_j \in N$ 에 대해서  $(e_i, e_j) \wedge (e_j, e_i)$ 의 특성을 가질 때  $e_i$ 와  $e_j$ 는 대화성을 갖는다.

#### 3.1 프레임워크 개요

프레임워크는 (그림 1)과 같이 통신 영역과 관리 서버(management server), 모니터링 대리자(agent) 및 사용자 인터페이스(user interface)로 구성된다. 통신 영역은 시그널링 메시지들을 교환하기 위한 제어 평면(unicast control plane)과 멀티캐스트 환경에서 오류 탐지 및 통계 수집을 위해 이용되는 기저 평면(multicast baseband plane)으로 구분된다. 제어 평면에서 TCP를 이용해 연결설정을 하며 기저 평면에서는 UDP 및 IGMP(Internet group management protocol [22])를 이용해 모니터링을 수행한다. 사용자 인터페이스는 웹 브라우저(Web browser)를 통해 제공 된다<sup>1)</sup>. 관리 서버는 IP 멀티캐스트를 이용 가능한 네트워크에 연결되며 관리 소프트웨어를 통해 기능이 활성화된다. 또한, 인터넷에 연결된 임의의 컴퓨터에서 대리자 소프트웨어를 구동시킴으로써 모니터링 대리자의 기능을 활성화시킬 수 있다. 관리 서버, 모니터링 대리자 및 사용자 인터페이스는 Java를 이용해 구현되었으며 서버에서 데이터베이스 관리를 위해 JDBC(Java database connectivity)가 활용됐다.



(그림 1) 멀티캐스트 모니터링 프레임워크

1) 사용자 인터페이스는 자바애플릿의 형태로 실행되며 임의의 서버 또는 관리 서버에서 제공하는 HTTP 데몬에 의해 서비스 된다.

통신 영역을 제어 평면과 기저 평면으로 분리함으로써 멀티캐스트 서비스로부터 고립된 모니터링 대리자들을 쉽게 탐지하고 SSM(source-specific multicast) 환경 하에서도 패킷 전달 통계를 수집할 수 있다. SSM은 일대다(one-to-many)방식의 멀티캐스트 규약으로 단방향 전송만 허락된다. 멀티캐스트 독립성은 NAT(network address translator)이나 방화벽 등으로 인해 연결 제한 문제를 가지고 있는 모니터링 대리자의 탐색을 용이하게 한다. 독립성은 제어메시지의 교환을 위해 멀티캐스트에 의존하지 않는 것을 의미한다 [16].

모니터링 대리자(agent)는 지리적으로 분산된 시스템으로 사용자 명령에 따라 오류 탐지 및 통계 수집 등의 작업을 수행한다. 예를 들어, 패킷 전달 통계의 수집 명령을 전달 받은 각 대리자는 메시지의 유효성을 검사한 후, 멀티캐스트 트래픽 발생기(traffic generator) 또는 수집기(acceptor)의 역할을 한다. 대리자들이 멀티캐스트 패킷을 지정된 멀티캐스트 세션에 송신하거나 해당 세션으로부터 패킷 전달 통계를 수집한다. 또한, 사용자는 적어도 2개 이상의 모니터링 대리자를 선택해 각 대리자에게 동료 대리자들의 IP 주소를 전달함으로써 라우팅 경로 등을 탐지하게 한다. 모니터링 대리자의 메시지 조정기(message handler)는 수행 결과를 관리 서버에게 피드백 한다. 사용자에게 의한 실시간 명령과 결과 응답은 관리 서버를 통해 이루어진다.

제어 평면에서 관리 서버의 주 역할은 명령 처리와 내부 데이터베이스의 관리이다. 각 멀티캐스트 대리자들에 의해 생성되는 응답과 웹 인터페이스를 통한 사용자 명령은 관리 서버를 통해 상호 전달된다. 사용자는 웹 인터페이스를 통해 패킷 전달 통계를 수집하거나 오류 탐지를 수행할 하나 이상의 멀티캐스트 대리자들을 선택할 수 있다. 탐지 및 통계 수집에 참여할 멀티캐스트 대리자의 수를 제한시킴으로써 네트워크 대역폭 소모를 줄이고 피드백 폭주(feed implosion)를 방지한다.

관리 서버의 DB 조정기(database handler)는 새로운 모니터링 대리자가 연결 설정을 시도할 때, 대리자의 상세 정보(IP 주소 및 시스템 이름 등)를 내부 DB에 저장한다. 저장된 대리자 정보와 단대망 간 상태 정보는 웹 인터페이스를 통해 사용자에게 제공된다. 망대망 간 멀티캐스트 오류 탐지 및 통계 수집은 사용자의 명시적 요청에 의해 수행되며, 각 대리자들로부터 수집된 통계 정보는 관리 서버를 통해 사용자에게 전달된다. 관리 서버는 대리자가 수집한 정보를 내부 DB에 보관한다. 대리자에 의해 수집되어 관리 서버에 저장되는 정보는 왕복지연시간(round-trip time), 손실율(loss rate), 지연 변이 및 라우팅 경로 등이다.

관리 서버는 그룹 내 모니터링 대리자들의 멀티캐스트 가용성을 확인하기 위해 주기적으로 멀티캐스트 패킷을 송신하며 모니터링 대리자들도 상호 간 도달성 및 대화성의 확인을 위해 탐지 목적의 패킷을 발생시킨다.

### 3.2 메시지 전송규약

제안된 프레임워크는 상호 배타적인 2개의 통신 평면을

제공한다. 이를 위해 본 논문에서는 제어메시지 전송규약(CMTP, control message transport protocol)과 멀티캐스트 기저메시지 전송규약(BMTP, baseband message transport protocol)을 정의한다. 각 규약은 독립된 패킷 헤더를 가지며 기반 통신 평면에 종속적으로 사용된다.

#### 3.2.1 제어메시지 전송규약(CMTP)

제어메시지 전송규약(CMTP)은 제어 평면에서 메시지를 교환할 때 이용된다. 예를 들어, 사용자 명령이 관리 서버를 거쳐 대리자에게 전달되는 과정 또는 대리자의 응답이 관리 서버를 거쳐 사용자에게 피드백 되는 전 과정에서 CMTP 메시지가 사용된다. 또한, 모니터링 대리자들 간 소프트웨어 일관성(consistency)을 유지하기 위해 버전 관리 및 업데이트를 수행할 때 CMTP 패킷을 이용해 메시지를 교환한다.

멀티캐스트 기저메시지 전송규약(BMTP)은 모니터링 대리자 간 또는 관리 서버와 대리자 간에 오류 탐지 및 통계 수집을 위해 멀티캐스트 패킷을 송·수신할 때 사용되며 활용 형태는 제어메시지 전송규약과 유사하다. 예를 들어, 관리 서버나 모니터링 대리자가 그룹 내 대리자들의 멀티캐스트 가용성 검사를 위해 주기적으로 발생시키는 멀티캐스트 트래픽의 헤더는 기저메시지 전송규약에 따른다.

CMTP 패킷은 (그림 2)와 <표 1>과 같이 각각 패킷 헤더와 MDM(multicast description message)으로 구성된다. CMTP패킷 헤더는 총 16바이트의 고정 필드와  $4 \times n$  ( $n \leq 16$ ) 바이트의 유동 필드로 구분된다. 그룹 내에서 탐지 및 통계 수집에 참가할 모니터링 대리자들의 수( $n$ )가 유동 부분의 크기를 결정한다. 사용자는 웹 인터페이스를 통해 해당 모니터링 대리자들을 선택한다.

CMTP 패킷 헤더의 각 필드는 다음과 같은 역할을 수행한다.

- 프로토콜 버전(V, 4 비트)
- eXtension(X, 1 비트): CMTP 패킷의 MDM 포함 여부를 지시한다.
- Count(C, 4 비트): 오류 탐지 및 통계 수집을 위해 멀티캐스트 그룹에 참여해야 되는 모니터링 대리자들의 수 ( $C=n$ )를 의미한다.
- Mode(M, 1 비트): 관리서버가 CMTP 패킷을 수신했을 때 MDM을 해석하고 DB 저장 여부( $M=0$ )를 판단한다. M이 0일 경우 관리서버는 CMTP 패킷을 최종 목적지로 전달하는 역할을 한다.
- Active(A, 1 비트): 통계 수집을 위해 모니터링 대리자의 역할을 송신자 또는 수신자로 설정한다.

		8				16				24				32			
V	C	X	M	A	UNUSE				Primitive				Secondary				
IP address of the packet originator																	
Multicast group address																	
Port number of the originator								Port number of the group									
IP address of agent																	
.....																	

(그림 2) CMTP(control message transport protocol) 패킷 헤더

<표 1> Measurement description message

	내용
v=0	모니터링 대리자 버전
o=me@localhost	모니터링 대리자 이름
p=true or false	사실 IP 사용 여부
e=true or false	멀티캐스트 지원 여부
t=2873493469	측정 시간
r=<peer, results>	결과

- Primitive(1 바이트)<sup>2)</sup>: 명령의 종류를 결정한다. CMTP 패킷이 사용하는 주(primitive) 명령으로는 *ASK*, *RELAY*, *ORDER*, *RETURN* 등이 있다. *ASK*는 사용자 또는 모니터링 대리자와 관리서버 간 또는 모니터링 대리자와 동료 대리자 간 일반 정보를 교환할 때 사용되며, *RELAY*는 사용자가 대리자에게 (또는 역의 방향으로) 명령을 전달하기 위해 관리서버에게 메시지 중계 요청을 할 때 사용된다. *ORDER* 명령은 대리자 간 모니터링을 목적으로 사용되거나 사용자로부터 *RELAY* 요청을 받은 명령이 관리서버로부터 대리자에게 중계되어야 할 때 관리서버가 설정한다. 반대로 *RETURN* 명령은 대리자가 사용자에게 모니터링 결과의 중계를 요청할 때 관리서버에서 해당 명령을 이용해 사용자에게 보고한다. 부(secondary) 명령은 중단간 연결성 검사 등을 위해 주 명령의 세부 명령을 제시한다. 예를 들어, 주 명령 *ASK*는 *beatAlive*, *getAllAgentInfo*, *getSessionInfo*, *getFiles*, and *getReachabilityMatrix* 등의 부 명령을 가질 수 있다. 또한, 주 명령 *ORDER*나 *RELAY*는 부 명령 *normalProbes*, *twowayPing*, *onewayPing*, *twowayBurst*, *onewayBurst*, *twowayTrace* 및 *onewayTrace*를 가질 수 있다.
- IP address of packet originator(4 바이트) 및 port number of the originator: 특정 명령에 대해서 CMTP 패킷을 최초 생성한 시스템의 IP 주소이다. 프레임워크 내 모든 제어 메시지들은 반드시 관리 서버를 거쳐야 하기 때문에 CMTP 명령의 발생원이 명시적으로 포함되어야 한다. 모니터링 대리자는 응답 목적의 CMTP 패킷의 생성원이 될 수 있다. 예외로, 모니터링 대리자가 생성할 수 있는 유일한 요청 목적의 CMTP 패킷은 *beatAlive* 메시지이다.
- Multicast group address(4 바이트) 및 port number of the group: 실시간 멀티캐스트 세션 모니터링을 목적으로 추후 프레임워크 확장을 위해 확보되어 있는 필드이다.
- IP address of agent: 오류 탐지 및 통계 수집을 위해 그룹에 참여해야 하는 대리자들의 IP 주소이다. 모니터

링 대리자가 요청 목적의 CMTP 패킷(*beatAlive*)을 송신할 때는, 해당 대리자와 멀티캐스트 도달성이 있는 세션 내 동료 대리자들의 IP 주소를 기록한다. 최대 16 개까지 수용 가능하며 *beatAlive* 패킷을 송신할 때마다, 도달성이 있는 동료 대리자의 IP 주소를 임의 선택한다.

제어 평면에서 관리 서버는 중단 사용자와 모니터링 대리자를 연결시켜주는 인터페이스의 역할을 수행한다. 관리 서버는 메시지 개조 인터페이스(message adaptation interface)를 제공함으로써 CMTP 패킷을 수정할 수 있는 권한을 갖는다. 예를 들어, 관리 서버가 사용자로부터 *<RELAY, twowayPing>* 메시지를 각각 주 명령과 부 명령으로 수신하고 CMTP 패킷의 Mode가 1로 설정되어 있을 경우, *<RELAY, twowayPing>* 메시지를 *<ORDER, twowayPing>*으로 수정해 모니터링 대리자들에게 전달한다.

CMTP 패킷은 바이트 정렬된 헤더를 사용하기 때문에 라우팅 패스 정보 등 가변 길이의 측정 결과를 포함하는데 어려움이 있다. MDM(measurement description message)은 모니터링 대리자 또는 사용자가 관리 서버와 보다 유연하게 정보를 교환할 수 있는 방안을 제공한다. MDM은 CMTP 패킷에 캡슐화되어 제공되며 eXtension 비트가 1로 설정되어 있으면 CMTP 패킷이 MDM을 포함하고 있다. MDM은 <표 1>과 같이 대리자, 시간 및 결과 정보 등을 기록한다. 결과 (r=<peer, results>)는 측정에 참가한 상대 대리자의 주소와 측정 결과들을 갖는다.

3.2.2 기저메시지 전송규약(BMTP)

멀티캐스트 기저 평면에서 탐지 및 통계 수집을 위해 BMTP(baseband message transport protocol) 패킷이 정의된다. BMTP는 기존 RTP/RTCP(RTP control protocol) 규약을 활용한 왕복지연시간의 계산을 보다 정교하게 한다. RTCP SR(sender report) 및 RR(receiver report) 패킷을 이용해 왕복지연시간을 추정할 수 있지만, RTCP 패킷의 송신 간격이 크기 때문에 측정된 지연시간이 부정확할 수 있다. BMTP는 응용계층 프로토콜로 IP/UDP/RTP에 캡슐화되어 사용되므로 패킷 전달 통계는 RTP와 BMTP 헤더를 통해 확보한다. (그림 3)은 BMTP 패킷의 헤더 형태를 보여준다.

BMTP 헤더 형태는 하부 전송계층에 독립적으로 구성되며 관리 서버 및 모니터링 대리자들 사이에서 탐지용 멀티캐스트 패킷을 송·수신 할 때 사용된다. BMTP 헤더는 *normalProbes*, *onewayPing*, *twowayPing*, *onewayBurst* 및

8		16		24		32	
V	C	Task		Sequence number			
IP address of a multicast packet originator							
Timestamp (1 <sup>st</sup> half)							
Timestamp (2 <sup>nd</sup> half)							
IP address of a multicast packet replier							
Elapsed time since a replier receives this packet							

(그림 3) BMTP(baseband message transport protocol) 패킷 헤더

2) 각 부 명령은 이름이 의미를 내포하고 있기 때문에 구체적인 설명은 생략한다.

*twowayBurst* 등의 명령으로 구분된다. 각 대리자들은 BMTP 헤더의 Task 필드를 CMTP 헤더의 부 명령 필드 값과 일치되도록 설정한다. Count 필드는 멀티캐스트 탐지 패킷을 수신 받을 대리자들의 총 수이다. 타임스탬프(timestamp)는 BMTP 패킷이 생성된 시간을 마이크로 초 단위로 나타낸다. 마지막으로 경과시간(elapsed time)은 BMTP 응답요구 패킷을 수신 받은 동료 대리자가 BMTP 응답 패킷을 보낼 때까지 해당 대리자에서 경과한 시간을 의미한다. 단방향 트래픽 집중 시험(burst test)을 위해 두 모니터링 대리자를 이용할 경우 경과시간은 설정되지 않는다. 트래픽 집중 시험은 단위 시간 당 가능한 많은 수의 패킷들을 송·수신함으로써 라우터나 스위치의 멀티캐스트 패킷 포워딩(forwarding) 성능과 방화벽 등의 존재 여부를 추정하는데 활용 된다. 일반적으로 방화벽에서 패킷이 필터링 될 때 전송 성능은 크게 낮아진다.

3.3 멀티캐스트 가용성 및 일관성

접속망(access network)에서 IGMP 질의(query) 패킷의 수신은 첫 번째 홉 라우터(first-hop router)와 연동된 중단에서 멀티캐스트 서비스의 가용성(availability)을 결정짓는 중요한 판단 기준이다. 또한, 가용성은 중단이 멀티캐스트 그룹에 참여(join)하고 해당 그룹으로부터 멀티캐스트 패킷을 수신 가능한지의 여부를 나타낸다.

제안된 프레임워크에서는 IGMP 질의 패킷과 멀티캐스트 시험 패킷의 수신 여부를 기준으로 모니터링 대리자에서의 멀티캐스트 가용성을 확인한다. 멀티캐스트 시험 패킷의 수신 여부는 관리 서버가 주기적으로 보내는 시험 패킷을 이용한다. 시험 패킷은 *normalProbes* 명령을 갖는 BMTP 규약을 준수한다. 현재 시험 패킷의 패킷 간 전송 간격은 5초로 설정되어 있다. 멀티캐스트 가용성은 있지만, 관리 서버와 특정 모니터링 대리자 간의 경로(path) 상에 오류가 존재하는 경우를 고려해 그룹 내 모든 대리자들은 시험 패킷을 주기적으로 발생시킨다. 모니터링 대리자의 시험 패킷 발생 주기는 30초 이다.

모니터링 대리자는 가용성 확인을 위해 시험 패킷의 수신 여부와 해당 패킷의 발생원(시험 패킷을 생성한 모니터링 대리자) 등의 탐지 결과를 수집한다. 수집된 결과는 CMTP *beatAlive* 메시지를 통해 주기적으로 관리 서버에게 전달된다. 관리 서버와 모니터링 대리자 간에 주기적으로 교환되는 *beatAlive* 패킷을 통해 결점이 있는 대리자들이 쉽게 탐지될 수 있으므로 제어 평면의 관리성(manageability)이 높아진다. 모니터링 대리자들이 시험 패킷을 주기적으로 발생 시킴으로써 그룹 내의 모든 대리자들 간에 도달성 및 대화성을 추가적으로 확인할 수 있다. 멀티캐스트 가용성의 확인을 위해서 약  $8s_p \times (1/i_s + s_g/i_a)$  bps의 네트워크 대역폭이 필요하다.  $s_p$ 는 시험 패킷의 크기(64 바이트)이며  $s_g$ 는 멀티캐스트 그룹 크기이다. 또한,  $i_s$ 는 관리서버가 생성하는 시험 패킷의 송신 주기(5초)이며  $i_a$ 는 대리자가 발생시키는 시험 패킷의 송신 간격(30초)이다<sup>3)</sup>. 결론적으로, 제안된 프

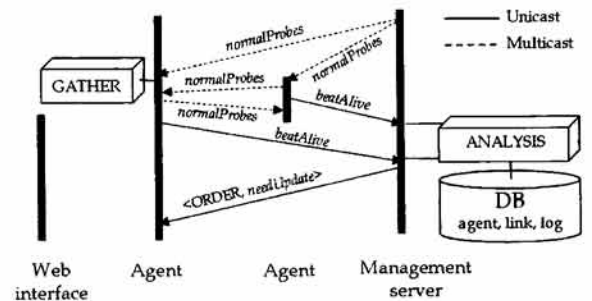
레이워크에서 멀티캐스트 그룹 크기가 1,000일 때 백그라운드 시험을 위해 약 102.415 Kbps의 네트워크 대역폭이 필요하다.

제안된 프레임워크는 단대망 간의 멀티캐스트 가용성을 포함해 동일 구간의 연결성 장애를 탐지한다. 협업 응용의 활용에 있어 IP 주소 변환기(network address translator)의 사용은 멀티캐스트 연결성 장애를 초래하는 대표적인 예이다.

소프트웨어 일관성은 모든 모니터링 대리자들이 동일한 버전의 소프트웨어를 활용하는 것을 의미한다. 일관성을 유지하는 것은 비표준(proprietary) 통신 규약을 이용하는 프레임워크의 유지 관리에 있어 중요한 요소이다. 예를 들어, 지리적으로 분산된 모니터링 대리자에게 새로운 탐지 방법이 추가된 대리자 소프트웨어를 배포할 때 대리자 간 일관성이 없으면 탐지 오류를 발생시킬 수 있다. 각 모니터링 대리자는 CMTP *beatAlive* 패킷을 이용해 60초 주기로 관리 서버와 통신한다. 관리 서버는 *beatAlive* 패킷에 포함된 MDM 메시지의 모니터링 대리자 버전(v) 정보를 이용해 일관성을 검사한다.

(그림 4)는 가용성 및 일관성 확인을 위해 관리 서버와 모니터링 대리자 간에 필요한 통신 절차를 보여준다. 가용성의 측정은 BMTP *normalProbes* 패킷을 이용해 멀티캐스트 기저평면에서 수행되며 각 모니터링 대리자는 CMTP *beatAlive* 패킷에 포함된 MDM 메시지에 수집된 정보를 기록해 관리 서버에 주기적으로 전달한다.

관리 서버는 *beatAlive* 패킷을 수신하면 모니터링 대리자가 주소 변환 여부를 확인한다. CMTP *beatAlive* 패킷 헤더는 패킷 생성자의 원 IP 주소와 포트정보를 포함하고 있다. 주소 변환기의 존재 유무를 파악하기 위해서 관리 서버는 CMTP 헤더의 패킷 발생원(originator) 주소와 소켓(socket)에서 전달 받은 패킷의 IP 주소를 비교해 판단 한다. 소프트웨어 버전이 서로 불일치 할 때는 CMTP <ORDER, needUpdate> 명령을 해당 대리자에게 송신하며 대리자는 관리 서버와 추가적인 메시지 교환을 통해 갱신된 소프트웨어를 다운 받아 일관성을 유지한다. 명령 전달의 순서와 내용을 정리하면 다음 <표 2>와 같다.



(그림 4) 가용성 및 일관성 확인 절차

3) 멀티캐스트 지원 라우터의 prune 간격이 일반적으로 60초 내외이므로 트리(tree)를 유지할 수 있도록 송신 간격이 설정되어야 한다.

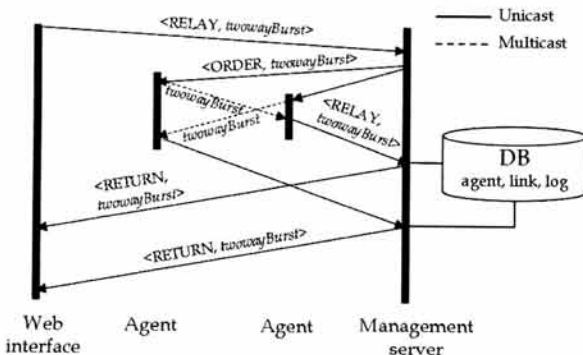
〈표 2〉 가용성 및 일관성 확인 절차의 요약

순번	송신	수신	규약	명령
	내용			
-	Agent	Agent	BMTP	<i>normalProbes</i>
	가용성 확인을 위한 백그라운드 시험			
-	Server	Agent	BMTP	<i>normalProbes</i>
	가용성 확인을 위한 백그라운드 시험			
1	Agent	Server	CMTP	<i>beatAlive</i>
	상태 정보에 대한 주기적인 보고			
2	Server	Agent	CMTP	<ORDER, needUpdate>
	소프트웨어 갱신 명령			

3.4 종단간 패킷 전달 통계

네트워크 혼잡 등으로 인한 전달 지연, 지연 변이 및 패킷 손실은 멀티미디어의 재생 품질을 악화시키는 주요한 원인이다. 제안된 프레임워크는 재생 품질에 영향을 주는 네트워킹 요소들의 상태 파악을 위해 단방향 집중 시험(uni-directional burst test) 및 양방향(bi-directional) 집중 시험 등 요구형 통계 수집을 수행한다. 집중 시험은 네트워크의 패킷 포워딩 성능과 혼잡 상태 등의 파악이 목적이다. 두 집중 시험을 위한 CMTP 패킷의 부 명령은 각각 *onewayBurst*와 *twowayBurst*이다.

(그림 5)는 모니터링 대리자(agent)들이 관리 서버로부터 CMTP <ORDER, twowayBurst> 명령을 받아 양방향 집중 시험을 수행하고 결과를 피드백하는 과정을 보여준다. 사용자는 웹 인터페이스를 통해 CMTP <RELAY, twowayBurst> 명령을 관리 서버에게 알린다. 관리 서버는 RELAY 주 명령을 ORDER로 변환한 후, 모니터링 대리자에게 양방향 집중 시험을 명령한다. CMTP 패킷을 전달받은 모니터링 대리자는 BMTP *twowayBurst* 패킷을 발생시키는 동시에 동료 대리자로 부터의 패킷 수신을 위해 대기 상태에 들어간다. 시험에 참여하지 않은 모니터링 대리자들은 도착한 BMTP 패킷을 처리하지 않는다. 패킷 전달 통계를 수집한 모니터링 대리자는 응답 목적의 CMTP



(그림 5) 양방향 집중 시험 절차

<RELAY, twowayBurst> 명령을 통해 관리 서버에게 결과를 전달한다. 관리 서버는 수집 통계를 내부 DB에 저장하고 CMTP <RETURN, twowayBurst> 명령을 통해 사용자에게 전달한다. 명령 전달의 순서와 내용을 정리하면 다음 <표 3>과 같다.

시험에 참여한 모니터링 대리자는 최적 일차 추정자(optimal first-order estimator)를 이용해 지연 변이를 추측한다. 대리자 ( $e_m, e_n$ ) 간에 응답요구 시험을 수행하고  $e_m$ 에서 BMTP 패킷  $i$ 의 송신 시간이  $t_{<s,i>}$ 이고  $e_n$ 에서 해당 패킷  $i$ 의 수신 시간이  $t_{<r,i>}$ 라고 가정하면 패킷  $i$ 와  $i+n$ 번째 패킷의 송신 시간차( $d_{<i,i+n>}$ )를 계산할 수 있다.

$$d_{<i,i+n>} = (t_{<r,i+n>} - t_{<r,i>}) - (t_{<s,i+n>} - t_{<s,i>})$$

최적 일차 추정자에 의한  $i$ 번째 패킷의 지연 변이( $j_i$ )는 다음과 같다.

$$j_i = j_{i-1} + (|d_{<i-1,i>}| - j_{i-1}) / 16$$

BMTP 패킷의 크기는 64바이트이며, 패킷 간 송신 간격은 10ms로 설정 된다.

〈표 3〉 양방향 집중 시험 절차의 요약

순번	송신	수신	규약	명령
	내용			
1	Web	Server	CMTP	<RELAY, twowayBurst>
	양방향 집중 시험 요구 명령의 전달 요청			
2	Server	Agent	CMTP	<ORDER, twowayBurst>
	양방향 집중 시험 명령			
3	Agent	Agent	BMTP	<i>twowayBurst</i>
	대리자 간 양방향 집중 시험 수행			
4	Agent	Server	CMTP	<RELAY, twowayBurst>
	집중 시험 결과 보고 명령의 전달 요청			
5	Server	Web	CMTP	<RETURN, twowayBurst>
	집중 시험 결과의 보고			

3.5 멀티캐스트 도달성 및 대화성

멀티캐스트 도달성(reachability)과 대화성(interactivity)은 가용성(availability)을 확인하기 위해 모니터링 대리자들이 주기적으로 발생시키는 시험 패킷을 통해서도 판단 가능하다. 하지만, 대역폭 소모량을 줄이기 위해 시험 패킷의 발생 주기가 30초로 설정되어 있으므로, 가용성 상태의 갱신 주기가 길어진다. 또한, 네트워크 상황 파악을 위한 부가적인 정보 획득이 어렵다는 문제가 있다. 단점을 보완하기 위해서, 멀티캐스트 응답요구(multicast ping)와 라우팅 경로 추적(multicast trace) 방법을 프레임워크에 수용한다.

멀티캐스트 응답요구는 응용계층에서 구현되었으며 모니

터링 대리자들 간에 BMTP 패킷을 교환하는 방식이다. 이를 통해 단대단 간 멀티캐스트 대화성을 파악할 수 있다. 즉, 양방향 도달성을 실시간으로 확인할 수 있으며 추가적으로 왕복지연시간의 측정이 가능하다. 왕복지연시간은 모니터링 대리자들 간에 클럭 동기화된 타임스탬프를 갖지 않은 환경하에서 전달 지연을 효과적으로 추정할 수 있다.

대화성을 실시간 판별하기 위해 모니터링 대리자는 점대다(point-to-group) 방식으로 멀티캐스트 응답요구 패킷을 발생시킨다. 점대다 방식의 멀티캐스트 응답요구는 잠재적으로 피드백 폭주를 일으킬 수 있다. 또한, 도달성 문제로 인해 응답을 보내지 않은 모니터링 대리자를 응답요구 패킷을 발생시킨 대리자가 파악할 수 없는 문제가 있다. 이는 특정 대리자가 동료 대리자들에 대한 정보를 가지고 있지 않고, 그룹 주소를 이용해 멀티캐스트 응답을 요구하기 때문이다. 멀티캐스트 응답요구의 이러한 문제점을 해결하기 위해, 제안된 프레임워크는 응답요구에 응해야 하는 대리자의 수를 한 쌍으로 제한한다. 또한, 동료 대리자에 대한 정보를 제공한다.

멀티캐스트 응답요구를 통한 도달성 검증을 위해서, 지정된 모니터링 대리자는 동료 대리자의 IP 주소를 BMTP 패킷 헤더에 등록한 후, 그룹 주소로 응답요구 패킷을 송신한다. BMTP 응답요구 패킷을 수신 받은 동료 대리자들은 헤더를 확인 해 자신이 동료 대리자로 등록되어 있으면 응답 패킷을 발생시킨다. BMTP 헤더에 등록된 동료 대리자의 응답 패킷은 응답요구를 발생시킨 대리자 주소, 응답요구 패킷의 송신 시간( $t_s$ ) 및 동료 대리자에서 응답요구 패킷의 수신 시점부터 응답 패킷의 송신 시점까지 지체된 시간( $t_h$ ) 등의 정보를 포함한다. 응답요구를 발생시킨 대리자는 응답 패킷을 수신했을 때( $t_c$ ),  $t_{rtt} = t_c - t_s - t_h$  와 같이 왕복지연시간( $t_{rtt}$ )을 계산한다.

멀티캐스트 라우팅 경로 추적은 단대망 또는 망대망의 입장에서 도달성과 대화성을 확인하는데 효과적이고 간소화된 방법을 제공한다. 경로 추적을 위해 *mtrace*를 프레임워크에 수용한다. *mtrace*는 사용자에게 은닉되어 있는 네트워크 정보를 보여줌으로써 멀티캐스트 라우팅 루프와 같은 경로 설정 오류를 탐지해 낼 수 있다. 네트워크 운영자들은 관리도메인 간 멀티캐스트 역 경로(reverse-routing path) 및 손실율(per-hop loss rate) 등을 파악하기 위해 *mtrace*를 활용한다. 멀티캐스트 라우팅 루프와 같은 네트워크 오류는 도달성 및 대화성에 치명적인 장애를 초래하기 때문에 원격 협업 응용의 활용에 있어 반드시 차단되어야 할 문제이다. *mtrace*를 효과적으로 사용하기 위해서는 양방향 탐지가 필수적이다. 예를 들어, 관리도메인 외부에서 내부의 방향으로 라우팅 루프가 존재하면 동일한 방향으로 경로 추적이 이루어져야 한다.

제안된 프레임워크는 양방향 탐지가 가능한 구조로, *onewayTrace* 또는 *twowayTrace* CMTP 부 명령을 이용해 단방향 또는 양방향 경로 추적을 한다. 단방향 경로 추적의 경우, 그룹 내 모니터링 대리자 또는 그룹 외부의 시스템들

이 경로 추적의 최종 목적으로 사용될 수 있다. 경로 추적 결과는 관리 서버에 저장된 후 사용자에게 전달되며, 사용자는 전달된 결과를 통해 라우팅 루프 등 네트워크 오류를 탐지하고 혼잡 구간을 추정할 수 있다.

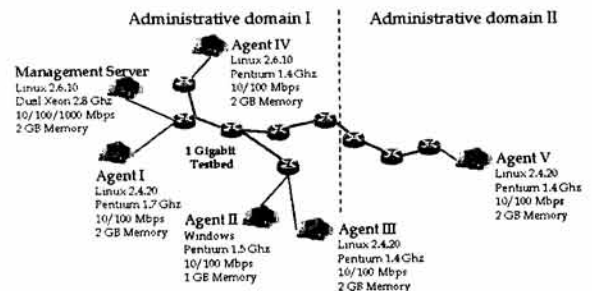
#### 4. 성능 측정 및 결과

본 절에서는 제안된 프레임워크의 성능을 살펴보기 위해 테스트베드를 구성하고 실험을 수행한다. 성능 측정을 통해 관리 서버가 기저 평면에서 피드백 폭주를 효과적으로 처리하고, 제어 평면에서 다수의 사용자 요구를 동시처리 가능함을 보인다.

실험을 위해 (그림 6)과 같이 하나의 관리 서버와 5개의 모니터링 대리자들을 지리적으로 분산시켜 구축했다. 도메인(administrative domain) I은 국가 과학기술연구망인 KREONET(Korea Research Environment Open NETWORK)의 관리 영역이며 도메인 II의 대리자는 광대역통합연구개발망인 KOREN(Korea Advanced Research Network)과 연결되어 있다. 모니터링 대리자들은 Windows™ 또는 Linux 플랫폼에서 동작한다. 관리 서버는 1Gbps 백본 네트워크에 연결되어 있으며 각 모니터링 대리자는 10~100Mbps LAN에 연결되어 있다. 임의의 두 모니터링 대리자들은 최대 8홉(hop) 거리에 있으며 왕복지연시간은 최대 9.8ms이다.

(그림 7)은 개발된 프레임워크의 웹 인터페이스와 도달성 측정 결과를 보여준다. 웹 인터페이스는 사용자에게 측정 방법(debugging type), 그룹에 참여할 모니터링 대리자(target), 탐지 방향(probing direction) 등의 선택권을 준다. 웹 인터페이스는 현재 활성화 되어 있는 모든 모니터링 대리자들에 대한 세부 정보와 각 대리자의 멀티캐스트 가용성 및 연결 제한 여부 등을 보여 준다.

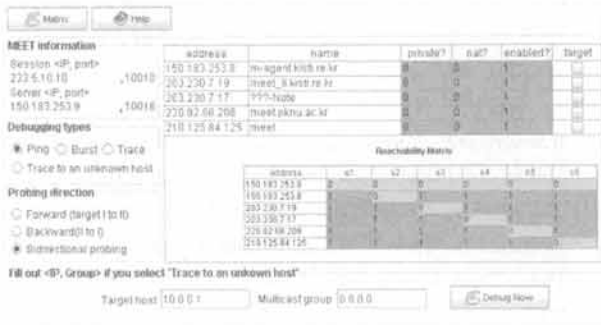
멀티캐스트 기저 평면에서 피드백 억제가 관리 서버의 시스템 부하에 미치는 영향을 조사하기 위해서 실험을 수행한다. 효과적인 피드백 억제는 제안된 프레임워크의 대화성 측정이 갖는 확장성을 보여준다<sup>4)</sup>. 모니터링 대리자가



(그림 6) 테스트베드 설정

4) 제안된 프레임워크의 확장성은 다양한 모니터링 도구의 프레임워크 내 수용 가능 여부 등 시스템 측면과 패킷 데이터의 범람 억제 등 네트워크 측면을 동시에 고려한다. 시스템 측면의 확장성은 소프트웨어 일관성을 통해 확보하며, 최상의 경우 약 48,000 모니터링 대리자 사이의 가용성(100 bps/대리자)을 실시간으로 확인 가능하므로 네트워크 측면의 확장성도 가지고 있다. 대화성 측정은 요구형 오류 검증 방식이다.



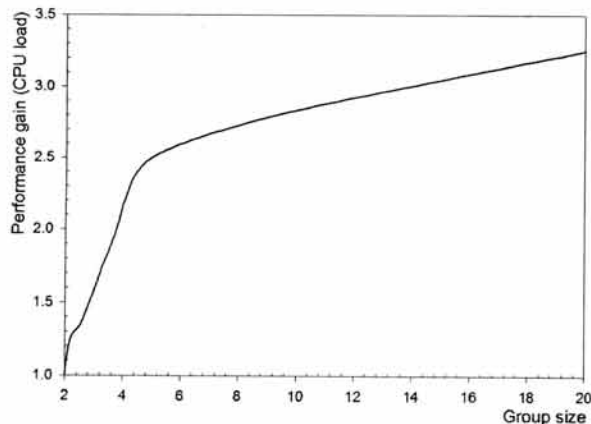


(그림 7) 웹 인터페이스

BMTP 응답요구 패킷을 동료 대리자에게 송신하는 시간이  $t_1$  이고  $t_2$  는 응답요구에 대한 응답 패킷의 도착시간이라고 했을 때 피드백 억제가 없다면, 응답을 요구한 모니터링 대리자는  $t_2 - t_1$  시간 동안 최대  $n + 1$  개의 패킷을 처리해야 한다.  $n$  은 그룹 내 모니터링 대리자들의 총 수이다. 실험을 위해 모니터링 대리자 VI 와 V를 지정한 후, BMTP 패킷을 발생시켜 모니터링 대리자 II에 송신하도록 했다.  $t_2 - t_1$  은 250ms로 설정했다.

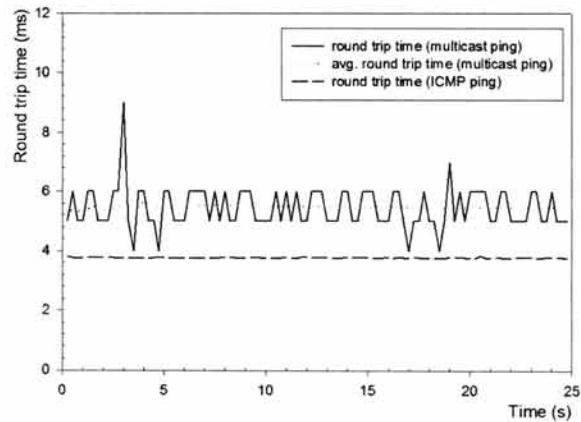
(그림 8)은 CPU 부하의 성능 이득(performance gain)이며 모니터링 대리자 II에서 Linux top 명령을 이용해 해당 프로세스의 CPU 부하를 측정했다. BMTP 패킷의 On/Off 주기는 각각 250ms이며 첫 번째 On 구간으로부터 1초 간격으로 총 10초간 측정한 평균값을 CPU 부하로 나타낸다. 성능 이득은  $\lambda_{ns}/\lambda_{fs}$  로 정의한다. 피드백 억제를 했을 때 측정된 CPU 부하가  $\lambda_{fs}$  이며 피드백 억제를 하지 않았을 경우는  $\lambda_{ns}$  이다. (그림 8)에서 보는 바와 같이 제안된 프레임워크는 멀티캐스트 그룹 크기가 커질수록 성능 이득도 증가한다. 멀티캐스트 그룹 크기가 20이고 피드백 억제를 하지 않았을 경우 모니터링 대리자 II에서 측정된 CPU 부하는 약 34%이다.

프레임워크 내에서 응답요구 시험(multicast ping)에 이용



(그림 8) 성능 이득

되는 BMTP 패킷은 응용 계층에서 처리되므로 측정 결과의 실효성이 떨어질 수 있다. 측정 결과의 실효성을 살펴보기 위해서 모니터링 대리자 I 과 II를 지정해 멀티캐스트 응답요구 시험을 수행했다. (그림 9)는 응답요구 시험을 통해 측정된 왕복지연시간의 변화를 ICMP(Internet control message protocol [24]) ping과 비교한 결과이다. 구현된 응답요구 시험의 왕복지연시간이 ICMP ping에 비해 실시간 변화폭이 상대적으로 크지만, 평균치는 약 1.5ms 정도의 미세한 차이를 보임을 알 수 있다. 변화폭이 큰 이유는, Java 언어의 특성 상 작업 스케줄링(task scheduling) 및 쓰레기 수집(garbage collection) 등이 실시간으로 이루어지지 않아 시간 측정에 어려움이 있기 때문이다 [25].



(그림 9) 왕복지연시간 비교

### 5. 결론

본 논문은 중단간 멀티캐스트 오류 탐지를 위한 모니터링 프레임워크를 제안했다. 제안된 프레임워크는 제어평면과 기저평면을 분리하고 관리 서버를 통해 작업 제어를 수행함으로써 모니터링 그룹에 대한 관리성을 높인다. 또한, 백그라운드 시험을 최소화하고 요구형 시험을 확대함으로써 네트워크 자원을 효과적으로 사용한다. 소프트웨어 일관성의 지원은 추후 모니터링 방법의 확충 등 프레임워크를 확장하는데 유효하다. 제안된 프레임워크를 통한 멀티캐스트 가용성, 도달성 및 대화성 등의 확인은 IP 멀티캐스트를 활용하는 협업 응용의 안정적 활용에 기여한다. 추후 연구 과제는 멀티캐스트 오류 탐지 방법의 추가 연구와 실시간 세션 모니터링 기능의 구현이다.

### 참고 문헌

[1] R. Smith, "Grid computing: A brief technology analysis," CTOnet Technical Report, 2004.  
 [2] L. Childers, T. Disz, R. Olson, M. Papka, R. Stevens, and

T. Udeshi, "Access Grid: Immersive group-to-group collaborative visualization," in Proc. 4th International Immersive Projection Technology Workshop, 2000.

[3] K. Egevang and P. Francis, "The IP network address translator (NAT)," IETF RFC 1631, May 1994.

[4] A. Ganjam and H. Zhang, "Connectivity restrictions in overlay multicast," in Proc. 14th International Workshop on Network and Operating Systems Support for Digital Audio and Video, 2004, pp.54-59.

[5] J. Weinberger, C. Huitema, and R. Mahy, "STUN-simple traversal of User Datagram Protocol (UDP) through Network Address Translators (NATs)," IETF RFC 3489, 2003.

[6] W. Fenner and S. Casner, "A "traceroute" facility for IP multicast," IETF Internet Draft, 2000.

[7] M. Fedor, M. Schoffstall, and J. Davin, "Simple Network Management Protocol (SNMP)," IETF RFC 1157, May 1990.

[8] P. Namburi, K. Sarac, and K. Almeroth, "Practical utilities for monitoring multicast service availability," Elsevier Computer Communications, 2005.

[9] K. Sarac and K. Almeroth, "Monitoring IP multicast in the internet: Recent advances and ongoing challenges," IEEE Communications Magazine, Vol.43, No.10, pp.85-91, Oct 2005.

[10] D. Makofske and K. Almeroth, "MHealth: A real-time graphical monitoring tool for Mbone," in Workshop. Network and Operating System support for Digital Audio and Video (NOSSDAV), 1999.

[11] J. Jo, W. Seok, J. Kwak, and O. Byeon, "Design and implementation of QoS measurement and network diagnosing framework for IP multicast in advanced collaborative environment," in Proc. International Conference on Computer and Information Science (ICIS), 2005, pp.545-550.

[12] D. Bacher, A. Swan, and L. Rowe, "A third-party RTCP monitor," ACM Multimedia, pp.437-483, Nov., 1996.

[13] S. Baldbusser, "Remote network monitoring management information base," IETF RFC 2819, May 2000.

[14] K. Sarac and K. Almeroth, "Supporting multicast deployment efforts: A survey of tools for multicast monitoring," Journal of High Speed Networking, Special Issue on QoS for Multimedia on the Internet, Vol.9, No.3, 2000.

[15] E. Al-Shaer and Y. Tang, "MRMON: remote multicast monitoring," in Network Operations and Management Symposium (NOMS), 2004, Vol.1, pp.585-598.

[16] K. Sarac and K. Almeroth, "Supporting multicast deployment efforts: A survey of tools for multicast monitoring," Journal of High Speed Networking, March, 2001.

[17] M. Kutzko and T. Rimovsky, "Multicast beacon v0.9," IETF Internet Draft, Nov., 2003.

[18] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," IETF RFC 3550, 2003.

[19] E. Al-Shaer and Y. Tang, "QoS path monitoring for multicast network," Journal of Network and System Management (JNSM), 2002.

[20] K. Almeroth, K. Sarac, and L. Wei, "Supporting multicast management using the multicast reachability monitor (MRM) protocol," Tech. rep., UCSB, May, 2000.

[21] F. Dressler, "An approach for QoS measurements in IP multicast networks - MQM multicast quality monitor," in Proc. Third International Network Conference (INC2002), 2002.

[22] W. Fenner, "Internet group management protocol, version 2," IETF RFC 2236, Nov. 1997.

[23] D.L. Mills, "Network Time Protocol (NTP)," IETF RFC 958, 1985.

[24] J. Postel, "Internet control message protocol," IETF RFC 792, 1981.

[25] S. Cavalieri, "Exploring real-time features of Java VM," in Proc. IEEE 28th Annual Conference of the Industrial Electronics Society (IECON), Nov., 2002, Vol.3, pp.2538-2543.

[26] C. Diot, B. Levine, J. Lyles, H. Kasseem and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," IEEE Network, vol. 15, no. 1, Jan/Feb. 2000.



### 조진용

e-mail : jiny92@kisti.re.kr

1999년 전남대학교 컴퓨터공학과(학사)  
 2002년 광주과학기술원 정보통신공학과(공학석사)  
 2007년~현 재 광주과학기술원 정보기전공학부 박사과정

2003년~현 재 한국과학기술정보연구원 슈퍼컴퓨팅본부 선임연구원

관심분야: 멀티미디어 시스템 및 서비스, 오버레이 네트워크, 망가상화



### 공정욱

e-mail : kju@kisti.re.kr

1993년 한국과학기술원 전기 및 전자공학과(학사)  
 1998년 포항공과대학교 정보통신학과(공학석사)

2008년 충남대학교 정보통신공학과 박사수료  
 1993년~2001년 (주)데이콤 중앙연구소 선임연구원

2001년~2002년 (주)맥스웨이브 책임연구원 및 한국항공우주연구원 초빙연구원

2002~현 재 한국과학기술정보연구원 슈퍼컴퓨팅본부 선임연구원

관심분야: 광통신망, 망 자원 관리 및 제어