

웹 기반 멀티캐스트 멀티미디어 응용

신 명 기†

요 약

본 논문은 웹 브라우저만을 사용하여 Mbone 세션에 참가하여 실시간 멀티미디어 데이터를 이음매 없이 수신할 수 있는 멀티캐스트 멀티미디어 웹 응용에 관해 기술한다. 본 응용은 웹 상에서 세션 발견을 위해 SDP, SAP 프로토콜을 이용하여 HTML 세션 페이지를 생성시키고, 실시간 스트림의 수신을 위해 RTP 프로토콜을 웹 구조에 통합함으로써 구현 가능하다. 웹 상의 RTP 재생은 넷스케이프 플러그인 방식과 자바 애플릿 방식으로 구현되었다. 넷스케이프 플러그인은 수신측의 지연 버퍼를 조정하는 방식으로 최소한의 품질을 보장하도록 구현하였으며, 애플릿 방식은 플랫폼-독립적이며, 새로운 미디어의 확장이나 업그레이드를 위해 추가적인 프로그램을 인스톨 해야 하는 불필요함을 없앨 수 있다. 특히 후자의 접근 방식은 핫 자바 브라우저 상에서 프로토타입핑 되어, 앞으로의 차세대 인터넷 멀티캐스트 환경에 적용될 새로운 응용이나 미디어를 위한 이상적이고 확장성을 지닌 솔루션을 제공한다. 추가적으로 본 구조에서 멀티캐스트 미디어의 확장성을 보이기 위해 멀티캐스트를 통한 HTML 문서의 분배라는 새로운 응용 구조를 제안한다.

A Web Based Multicast Multimedia Application

Myung-Ki Shin†

ABSTRACT

This paper presents a multicast multimedia Web application that allows Web users to join an Mbone session and receive real-time audio/video seamlessly. We attempt to identify a solution that can be deployed immediately for applying multicast media to the Web. This is accomplished by generating an HTML session page using SDP/SAP for session discovery and integrating RTP/RTCP into the Web for transmission of real-time stream. Our RTP player implementation is built on Netscape plug-in, which provides minimal QoS control as adapting an appropriate buffer size, and Java Applet, which makes it platform independent, scalable and easy to deploy. Thus, there is no longer need for new program installation to multicast various types of real-time media. Especially, RTP Applet approach provides an ideal and scalable solution for the new media or application "x" proposed on the Internet Multicast Infrastructure in near future. In addition, we describe on architecture for distribution of HTML pages via multicast.

1. 서 론

현재의 인터넷은 전세계 100여국에서 1600만명 이상이 연결되는 세계 최대의 통신망으로 발전하고 있다. 오늘날, 실시간 오디오, 비디오 데이터를 인터넷을

통해 전송하는 것은 Mbone(Multicast Backbone) 상에서 국제회의 등을 중계하는 것을 볼 수 있듯이 그리 놀라운 일은 아니다. Mbone은 인터넷상에서 멀티캐스트를 지원하는 호스트와 라이터로 구성된 가상 실험망을 가리키며, 다자간 회의나 중계를 위한 핵심 기술로써 인정받고 있다[1].

한편, 월드 와이드 웹은 인터넷상의 산재해 있는

† 정 회 위 : 한국전자통신연구원 표준연구센터 연구원
논문접수 : 1998년 7월 15일, 심사완료 : 1998년 9월 29일

여러 가지 정보들을 공유하기 위한 편리하고 단순한 방법을 제공함으로써 인터넷의 대부분의 응용과 미디어를 통합하여 사용자 인터페이스를 대변하는 수단으로 자리잡아 가고 있다[2]. 그러나 웹은 정보를 검색하기 위해 TCP 기반의 HTTP 프로토콜을 사용하게 되는데 이는 실시간 멀티미디어 데이터를 지원하지 못한다는 단점을 가지고 있다. 예를 들어 기존의 웹을 사용하여, Mbone 세션에 참가를 통한 멀티캐스팅 되는 오디오, 비디오를 이음매 없이 수신할 수는 없다.

물론 최선형 서비스만을 지원하는 현재의 인터넷에서 실시간 멀티미디어 서비스를 제공하는 것은 많은 문제점을 드러내고 있다. 전송 패킷들이 망 사정으로 인해 지연되고 손실될 수 있기 때문이다. 따라서 기존 인터넷의 TCP/IP 구조를 수정하지 않는 범위에서 이를 해결하기 위해 IETF에서는 RTP(Real-time Transport Protocol)/RTCP(RTP Control Protocol)를 개발하여 표준화 하였다. 또한 현재 멀티캐스트 세션은 세션 발견을 위해 IETF MMUSIC WG에서 개발하여 표준화중인 SAP(Session Announcement Protocol)와 SDP (Session Description Protocol)를 사용한다.

본 연구의 목적은 웹 기반 멀티캐스트 오디오, 비디오 응용을 구현하는 것으로, 이는 웹에 RTP와 SDP, SAP 프로토콜을 이음매 없이 통합하여 구축하여야 한다. 본 응용은 오디오, 비디오를 이용하는 사용자에게 웹 브라우저라는 재생 도구의 편리함을 제공하게 되며, 인터넷 Mbone 관련 표준에 따라 개발하여 웹 사용자는 기존 Mbone 응용의 실차 없이 멀티캐스트 스트림을 이음매 없이 수신할 수 있다는 장점을 가지게 된다.

2. 멀티캐스트 멀티미디어 웹 응용을 위한 요구사항

본 장에서는 웹에서 멀티캐스트 멀티미디어 데이터를 적용시키기 위한 몇가지 요구사항을 정의한다. 본 논문은 현재 제안된 여러 가지 방식들을 분석하여, 현재 구조의 커다란 수정 없이 쉽게 적용 가능한 해결책들을 정하고자 한다.

2.1 웹 상의 RTP 프로토콜 통합

실시간 응용들은 정해진 시간 내에 데이터를 전송해야 하는 제약이 있다. 이러한 이유로 오류로 인한 재전송과 같은 TCP 기반의 기법을 패킷 도착 시간을 지연 시킴으로써 실시간 응용을 위해서는 사용될 수 없다. 따라서 IETF에서는 이를 해결하기 위해 UDP 기반의 RTP를 실시간 응용을 위한 표준 전송 프로토콜로 제정했다. RTP는 페이로드 유형 식별, 시퀀스 넘버링, 타임 스탬프를 지원하며, 이러한 실시간 데이터는 RTCP라는 프로토콜에 의해 모니터 되도록 설계되었다. RTP/RTCP는 IETF RFC 1889에 정의되어 있으며, 오디오, 비디오를 위한 프로파일은 RFC1990에 정의되어 있다[3, 4]. 현재 Mbone상의 vic, vat, rat와 같은 대부분의 오디오, 비디오 응용들은 RTP를 기반으로 구현되어 있으며, 넷스케이프의 LiveMedia나 마이크로소프트의 Netshow 같은 제품들도 RTP를 지원함으로써, RTP는 명실상부한 인터넷상의 실시간 정보를 전송하기 위한 표준 프로토콜로써 자리잡은 상태이다.

최근 웹 상에 RTP를 통합하기 위한 몇몇 연구들이 있었다. 웹 프로토콜인 HTTP는 TCP 기반의 프로토콜로써 실시간 스트림을 전송을 위해서는 적합하지 않다[5]. 예를 들어 실시간 스트림의 수신자가 TCP 재전송을 기다리게 된다면, 실시간 데이터의 재생을 위해 많은 지연이 발생할 수 있다. 이를 해결하기 위해 HTTP를 UDP 상에서 사용하는 새로운 HTTP 프로토콜을 제안하는 연구들이 있었으나, 이러한 방식은 브라우저/서버 구현을 더욱 복잡하게 만들 수 있으며, 브라우저가 두개의 서로 다른 처리 루틴, 즉 TCP상의 HTTP와 UDP상의 HTTP 처리 루틴을 가져야 한다는 단점을 가지고 있다. 한편 다른 연구에서는 RTP를 단순히 웹의 보조 응용에 의해 재생만을 처리하게 하는 방식을 제안하여 개발한 바 있으며[6], 이러한 접근 방식의 단점은 실시간 스트림 정보(메타 정보)를 얻기 위해 추가적인 라운드-트립 지연을 가져온다는 점에 있으나, 기존 웹 구조와 HTTP 프로토콜을 손상시키지 않고, 쉽게 적용 가능하다는 잇점도 포함하고 있다.

2.2 멀티미디어 URL과 세션 기술 프로토콜(SDPI)의 사용 비교

웹상에서 URL은 인터넷상에 산재한 자원을 이드레싱하기 위한 방식이다. 최근 실시간 멀티미디어를 위

한 URL을 정의하기 위한 여러 연구들이 있었다. 예를 들어 일부 시스템에서는 실시간 RTP 스트림을 위해 다음과 같은 URL을 정의하여 사용한다.

```
rtp://224.0.12.16:5555:127:h.261 ...
```

이러한 방식은 별도의 HTML 페이지를 사용하지 않고 원하는 스트림을 웹 서버 연결없이 송신자로 부터 직접 받아볼 수 있다는 장점을 가지고 있다. 즉, 웹 클라이언트가 세션 시작시에 HTTP 서버와 통신없이 스트림을 받아들 수 있다. 이러한 멀티미디어 URL을 정의하는 방식은 실시간 스트림이 웹 서버와 별도로 존재한다는 점에서 착안한 것이다. 이 접근 방식은 장점도 있지만, URL를 더욱 복잡하게 한다는 단점을 가지고 있다. IETF MMUSIC WG 의장인 M. Handley 는 실시간 스트림을 위한 URL 구문을 정의하는 것은 잘못된 생각이라고 주장한다. 예를 들어 멀티캐스트 세션과 같은 세션 정보들은 멀티캐스트 주소, 포트, 미디어 유형 등의 정보들을 포함하게 되어 URL 자체가 길어질 수 있으며, 특히 한 세션에 여러 개의 미디어 정보(오디오, 비디오 등)가 포함되게 되면, URL로는 이를 표현하기에는 불충분하기 때문이다. 따라서 IETF 에서는 MMUSIC WG에서 개발하여 표준화된 SDP를 이용하도록 권고하고 있다[7, 8]. 즉, 웹 서버에서 application/x-sdp 유형의 MIME 정보에 따른 정보를 제공하도록 하는 방식이다. 예를 들어 Precept사의 IP/TV 프로그램 가이드는 HTML상의 멀티미디어 세션을 어드레싱하기 위해 application/x-sdp MIME 정보를 이용하도록 구현한 제품이다[9]. 결국 표준화의 방향은 IETF와 W3C에서 제안한 SDP MIME 유형과 HTML 4.0 OBJECT 태그 방식[10]에 따라 정해질 것으로 기대된다. 예를 들어 실시간 멀티미디어 정보를 포함하는 미래의 HTML 문서는 다음과 같이 쓰여질 수도 있을 것이다.

```
<OBJECT DATA="mbone.sdp" TYPE="application /x-sdp"
  HEIGHT="400" WIDTH="500">
.....
</OBJECT>
```

2.3 웹 상의 멀티미디어 데이터의 아음매 없는 재생

웹 상에서 실시간 멀티미디어를 재생하기 위해서는 멀티미디어 보조 응용을 사용하는 것이 일반적이나, 최근에는 넷스케이프나 익스플로러와 같은 브라우저에

서 플러그인을 지원함으로써 각 데이터에 맞는 플러그인을 미리 인스톨 하기만 하면 재생시에 별도의 보조 응용의 도움 없이도 바로 브라우징 할 수 있어 많이 사용되고 있다. 이러한 방식은 기존 브라우저의 수정 없이 플러그인 프로그램을 각 브라우저에 맞게 구현만 하면 된다는 장점도 있으나, 각각의 웹 브라우저 사용자가 일일이 플러그인 프로그램을 인스톨 해야 하고, 또 업그레이드 시에도 똑 같은 작업을 반복해야 한다는 번거로움도 가지고 있다. 따라서 최근에는 이에 대한 단점을 자바 애플릿을 이용하여 해결하려는 연구가 진행 중이다.

최근 선 마이크로시스템즈에서는 자바 미디어 프레임워크를 개발하여 제안 중이며, 이 API를 사용하여 애플릿을 개발하면, 플러그인과 동일한 효과를 가지면서, 자바 애플릿의 특징인 프로그램 다운로드 사용으로 인해, 플랫폼 독립적이며, 자신의 브라우저에 프로그램을 인스톨 해야 하는 필요가 없게되어 보다 사용자에게 편리함을 제공하게 된다. 단 넷스케이프와 익스플로러와 같은 일반적인 브라우저들이 자바 미디어 API를 지원하지 않아 아직까지는 일반적으로 사용할 수 없어 자바 브라우저인 핫자바에서만 가능하다는 단점을 가지고 있다.

2.4 인터넷상의 서비스 품질 제어(QoS) 방법

실시간 멀티미디어 응용들은 지연, 지연 지터, 손실률, 대역폭에 의한 서비스 품질제어(Quality of Service: QoS)를 요구한다. 그러나 최선형 서비스에 따른 현재의 인터넷 환경하에서 QoS 요구사항을 만족시킨다는 것은 어렵다. 이에 따라 기존 인터넷 환경하에서 최소한의 QoS를 제공하려는 연구들이 활발히 진행중이다. 먼저, Bolot와 Turtetti[11]는 송신자가 수신자에 의해 제공되는 피드백 정보에 따라 전송율을 제어하는 방식인 송신자-기반 적응(sender-initiated adaptation) 알고리즘을 제안하였다. 이 방식은 각각의 수신자가 송신자에게 자신의 패킷 손실률을 보고하여, 수신자는 이 보고에 따라 망 상태를 결정하게 되고, 따라서 엔코더의 전송율을 제어함으로써 전송율을 증가하거나 감소시키는 것이다. 이 방식은 송신자가 하나의 전송률에 따라 패킷을 전송하게 됨으로써 이기종간의 멀티캐스트 환경하에서는 적합하지 않다. 즉, 모든 수신자의 대역폭 요구사항을 동시에 단일 전송률로 만족시킬 수는 없다.

또 다른 연구방식으로는 McCann과 Jacobson[12]이 제안한 RLM(Receiver driven Layered Multicast)라고 불리는 수신자 기반 적응(receiver-oriented adaptation) 기법을 들 수 있다. RLM은 계층화된 멀티캐스트 전송에 따른 계층화된 소스 코딩 알고리즘을 이용하여 동작한다. 송신자는 이종의 멀티캐스트 그룹에 따른 각 계층의 소스 데이터를 전송한다. 각 수신자는 자신의 망 상태에 따른 멀티캐스트 그룹에 참가, 혹은 탈퇴함으로써 서비스 품질을 보장받게 된다. 즉, 수신자의 이종간 환경은 RLM에 의해 해결된다.

물론 위의 두 기법은 실시간 멀티미디어 응용이 완벽한 QoS를 보장받기 위한 완전한 해결 방법은 아니다. 실시간 멀티미디어 응용이 완벽한 QoS를 제공받기 위해서는 대역폭 예약과 트래픽 제어 방식이 요구된다. 따라서 기존 인터넷 구조를 수정하여 QoS를 보장하기 위한 방식으로 RSVP(Resource Reservation Protocol)와 Integrated Service, 등이 개발되었고 최근 Differentiated Service 등이 IETF를 중심으로 활발히 연구중이다. 이 방식의 문제점은 기존 인터넷에 곧바로 적용하기란 쉽지 않을 것이라는 데 있다.

3. 멀티캐스트 멀티미디어 웹 응용 구조

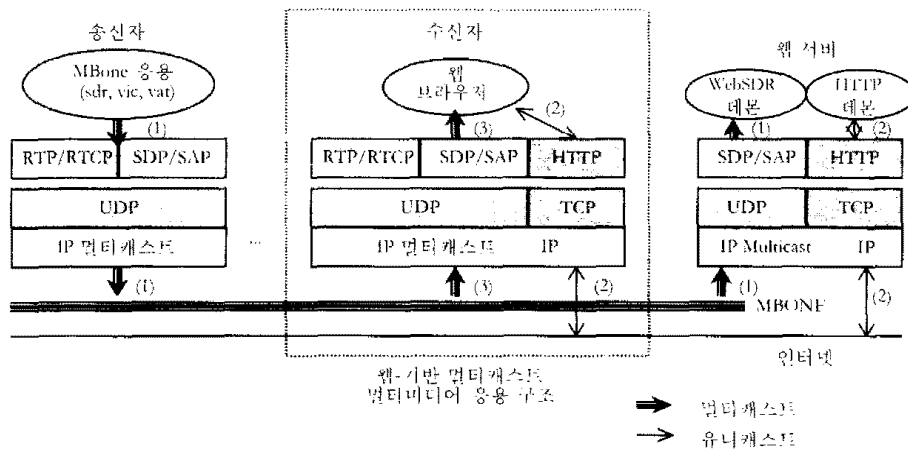
본 논문은 실시간 멀티미디어 전송을 위한 인터넷 표준 프로토콜인 RTP/RTCP와 세션 발견을 위한 SDP, SAP 표준 프로토콜을 포함하는 웹 기반 멀티캐스트 오디오, 비디오 응용 구조를 제안한다. (그림 1)

본 웹 브라우저 만들 가진 수신자 즉의 구조를 포함하여, MBone 오디오, 비디오 응용인 sdr, vic, vat을 가진 송신자, 그리고 웹SDR(Session DiRectory) 데몬을 포함하는 웹 서버 구조를 나타낸 것이다. 본 논문은 멀티캐스트 되는 실시간 오디오, 비디오를 웹 브라우저 하나만으로 수신하기 위해 관련된 기술들을 하나하나 다룰 것이다.

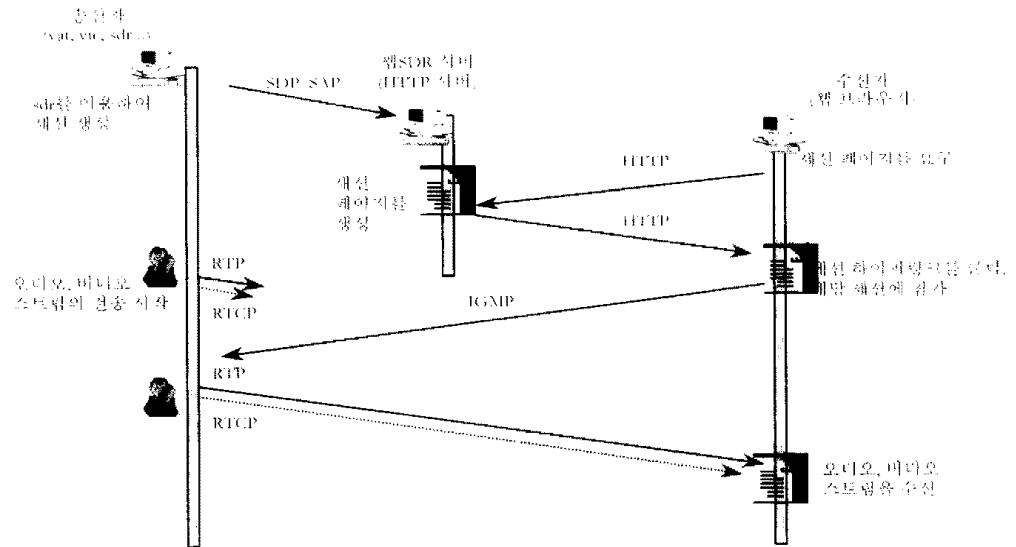
일반적인 MBone 사용자인 송신자가 MBone 응용을 이용하여 오디오, 비디오를 전송하게 되면, 웹 브라우저만을 가지고 있는 수신자는 먼저 HTTP를 통해 세션에 대한 정보를 웹 서버로부터 얻어온다. 이 정보는 SDP, SAP 프로토콜에 따라 멀티캐스트 주소, 포트, 미디어 유형 등으로 파싱되며, 수신자는 IGMP를 통해 그 세션에 참가하게 되고, RTP를 이용하여 실제 실시간 스트림을 수신하게 된다.

송신자와 수신자, 그리고 웹 서버 측에서의 동작 과정은 (그림 2)와 같으며, 이를 간략히 나타내면 다음과 같다.

- 송신자(일반적인 MBone 응용 사용자)
 - sdr를 이용하여 세션 생성
 - vat와 vic를 이용하여 오디오, 비디오 전송을 시작
- 웹 서버(웹SDR 데몬이 동작중인 일반적인 웹서버)
 - 세션 페이지와 파일을 생성
 - 수신자가 해당 세션 페이지를 요구하면 이를 반환



(그림 1) 웹-기반 멀티캐스트 멀티미디어 응용 구조
(Fig. 1) Web Based Multicast Multimedia Application Architecture



(그림 2) 응용 시나리오
(Fig. 2) Application Scenario

- 수신자(웹 브라우저)
 - 세션 페이지를 요구
 - 세션 파일을 얻기 위해 특정 하이퍼링크를 클릭
 - 세션에 참가하여, 오디오, 비디오 수신

3.1 웹SDR 서버

웹 브라우저는 오디오, 비디오 수신을 위한 해당 세션의 정보들을 알아야 한다. 본 연구에서는 웹 상에서 멀티캐스트 어드레싱을 하기 위해 <application/x-sdp> MIME 유형 방식을 채택하였다. SDP는 세션을 기술하기 위해 사용되며, 이 패킷은 SAP 프로토콜에 의해 수행된다. 이를 위해 일반적인 웹 서버에서 동작 가능한 웹SDR 데몬 프로그램을 설계하였다.

웹SDR 데몬은 웹 서버 내에서 데몬으로 동작하여 SAP에 의해 수신되는 패킷을 받아 세션 페이지(예를 들어 websdr.html)의 생성시키고, 이때 SDP MIME 정보를 세션 페이지에 포함시킨다. 플러그인 경우, 오디오, 비디오 스트림의 재생을 위해 <embed> 태그가 사용되며, 예를 들면 HTML 세션 페이지는 다음과 같다.

```
<H1> Mbone Audio/Video </H1>
....
<EMBED SRC=3567890/mbone.sdp TYPE=application/x-sdp
WIDTH=500 HEIGHT=400>
```

자바 애플릿의 경우는 RTP 오디오/비디오 재생기의 자동 다운로드를 위해 <APPLET> 태그가 내장된다. 예를 들면 HTML 세션 페이지는 다음과 같다.

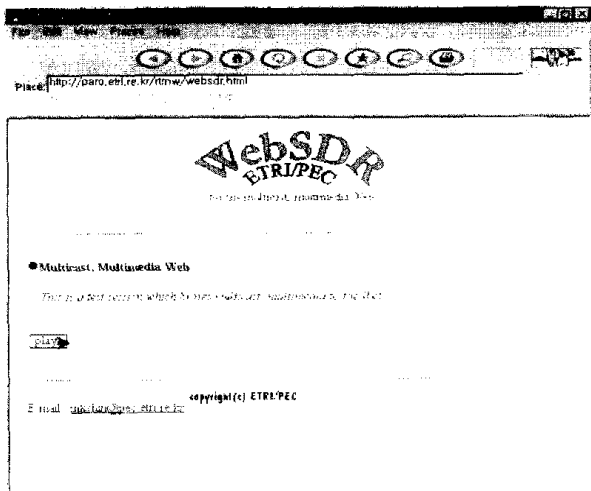
```
<H1> Mbone Audio/Video </H1>
....
<APPLET CODE=RTPPlay.class WIDTH=500
HEIGHT=400>
<PARAM TYPE=file VALUE=3567890/mbone.sdp>
</APPLET>
```

sdp 세션 파일은 SDP 표준에 따라 멀티캐스트 주소, 포트, 미디어 유형과 같은 정보를 포함하게 된다. 예를 들면 오디오, 비디오를 포함하는 세션인 mbone.sdp의 내용은 다음과 같다.

```
m - audio 21690 RTP/AVP 0
c = IN IP4 224.2.143.56/127
m = video 51990 RTP/AVP 31
c = IN IP4 224.2.200.96/127
....
```

(그림 3)은 웹SDR를 이용하여 생성된 세션 페이지의 예를 나타낸 것이다. 웹 사용자는 세션 페이지(예를 들면, <http://www.mbone.or.kr/rtmw/websdr.html>)를

웹 브라우저를 통해 요구한 후, 세션 페이지를 보고, 원하는 세션의 하이퍼링크(PLAY 버튼)을 클릭함으로써 원하는 세션에 참가할 수 있다. 이때 "리로드" 버튼을 누르지 않고, 실시간으로 변하는 세션 정보를 주기적으로 반영하기 위해 HTML 세션 문서내에 <meta http-equiv="refresh" content="60" url ... > 메타 정보를 삽입하였다. 웹SDR 프로그램은 C로 구현되었으며, 모든 웹 서버에 인스톨 가능하다.

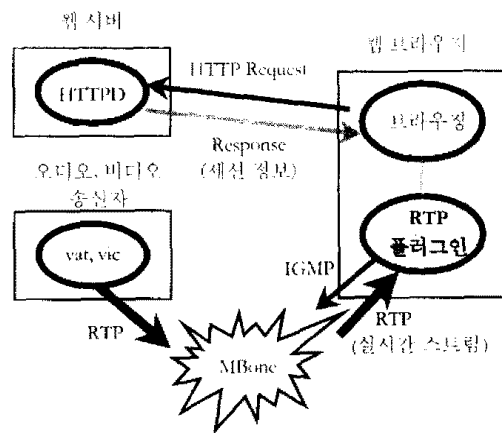


(그림 3) 웹SDR에 의해 자동 생성된 세션 정보 페이지
(Fig. 3) An Example of Session Page Generated Using WebSDR

3.2 RTP 플러그인

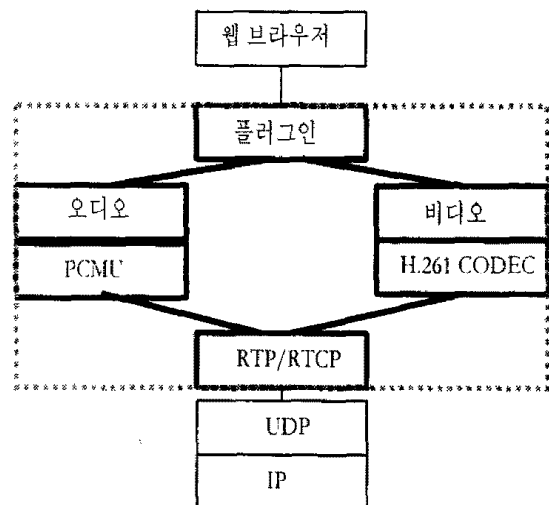
본 논문은 실시간 미디어의 재생을 이음매 없이 하기 위한 플러그인 방식과 차바 애플릿 방식을 이용하여 웹과 RTP 프로토콜간의 통합 구조를 구현하였다. 이 방식의 단점은 세션 정보를 얻기 위해 웹 서버(웹SDR 서버)에 추가적인 접근을 필요로 한다는 점이지만, 현재의 HTTP 프로토콜의 수정 없이 RTP를 웹에 통합할 수 있고 쉽게 적용 가능하다는 장점을 가지고 있다. 먼저 RTP 플러그인과 웹과의 통합 구조는 (그림 4)에 나타내었다. 플러그인은 넷스케이프나 익스플로러에서 웹 브라우저에 이음매 없이 새로운 미디어 유형을 통합하기 위해 대표적인 기술로 인정 받고 있다[13].

RTP/RTCP 플러그인 모듈은 sdp 파일로부터 멀티캐스트 주소, 포트, 미디어 유형과 같은 세션 정보를 얻게 되며, 이를 통해 IGMP를 이용하여 원하는 세션



(그림 4) 웹과 RTP 플러그인과의 통합 구조
(Fig. 4) Structure of Integrating RTP Plug-in into the Web

에 참가하게 되고 오디오, 비디오 스트림 수신을 위해 RTP와 RTCP 소켓을 여는 일을 한다. 현재 인코딩 유형은 비디오를 위한 H.261과 오디오를 위한 PCM 방식만을 지원한다. (그림 5)는 PCM 오디오와 H.261 비디오를 위한 플러그인 구조를 나타낸 것이다. 비디오 스트림의 경우, RTP/RTCP 플러그인 모듈은 H.261 코덱 'p64' 를 이용하여 패킷을 디코딩하여, 웹 브라우저내에 재생하기 위한 RGB 유형 데이터로 변환하며, 오디오 스트림의 경우는 이송 버퍼링을 이용하여 패킷을 수집한 후 사운드 카드를 통해 PCM 데이터를 재생한다.



(그림 5) RTP 플러그인 모듈 구조
(Fig. 5) Structure of RTP/RTCP Plug-in Modules

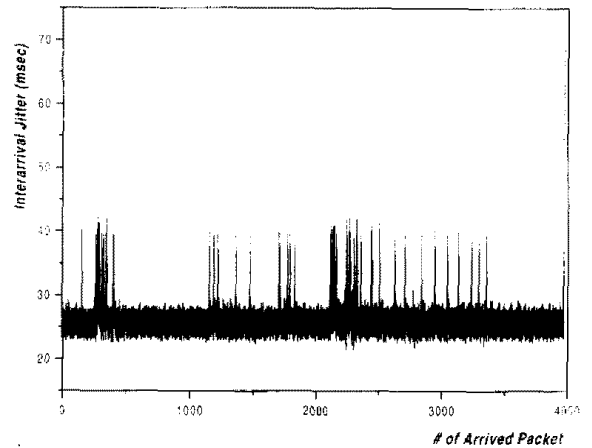
RTP/RTCP 플러그인은 C++로 구현되었으며 (그림 6)은 동작 예를 나타낸 것이다. (그림 6)에서 비디오 화면 아래쪽에는 RTCP를 이용하여 생성된 QoS 정보를 보여주도록 설계하였다. QoS 정보로는 SSID, RTP 패킷 손실률, 지연 지터 등을 포함한다.



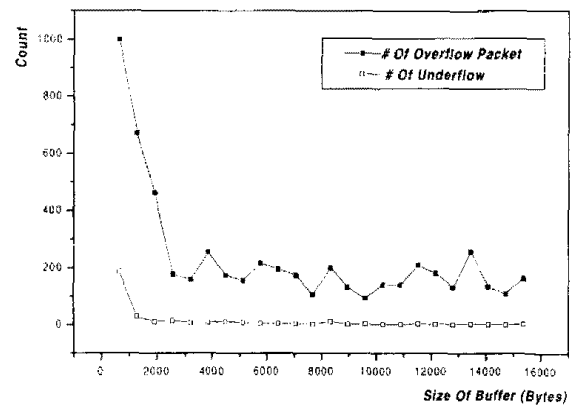
(그림 6) RTP 플러그인 동작 예
(Fig. 6) Operation Example of RTP/RTCP Plug-in

최선형 서비스를 지원하는 현재의 인터넷 환경하에서 최소한의 QoS를 보장하기 위해 본 연구에서는 수신자 기반 적응 알고리즘에 기반한 지연 버퍼를 조정하는 새로운 방식을 설계하여 이를 적용하였다. 이 방식의 기본 아이디어는 먼저 세션 수신의 시작 시에 몇 개의 패킷의 지연 지터를 계산한 후, 적당한 버퍼 크기를 찾는 데 있다. (그림 7)은 LAN상에서 방송되는 Mbone 오디오 세션의 패킷 도착간 지터를 측정함으로써 약 26ms 정도의 평균 지터값을 보여주고 있다. 이와 같이 측정된 오디오 세션의 평균 26ms 정도의 패킷 도착간 지터를 보상하는 적절한 최소 버퍼 크기를 찾기 위하여 RTP로 전송되는 오디오 패킷이 125s 마다 발생하는 오디오 샘플을 160개씩 전송하는 특징을 반영하여 다음 (그림 8)과 같은 결과를 측정하였다. 이는 수신단에서 두개의 재생버퍼를 가지게 하고 각 버퍼를 160의 4배수로 크기를 증가 시켜 160의 100배수인 경우까지 즉, 100개의 RTP 오디오 패킷을 저장할 수 있는 경우까지 고려하여 해당 버퍼크기에서 재생되지 못하고 버려지는 오버플로우 패킷의 수와 사운드 카드로 재생을 위해 버퍼가 보내지는 경우에 버퍼

가 재생할 패킷을 가지지 못하는 경우인 언더플로우의 횟수를 측정하였다.



(그림 7) Mbone 오디오 세션의 도착간 지터 측정
(Fig. 7) Delay Jitter of an Mbone Audio Session



(그림 8) 재생 버퍼 크기 변화에 따른 오버플로우 패킷수와 언더플로우 횟수

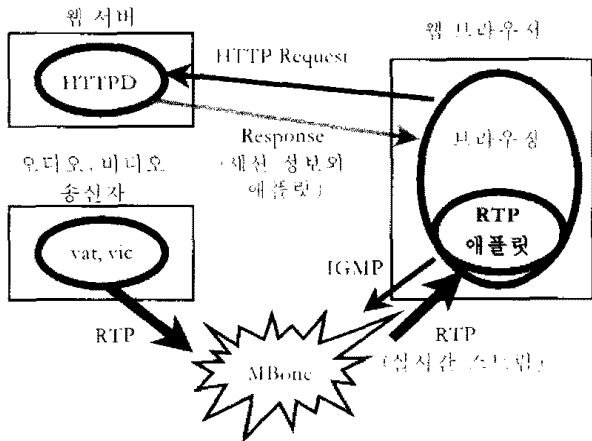
(Fig. 8) Service Quality versus Buffer Size in an Audio Session

(그림 8)에서 알 수 있듯이 재생버퍼의 크기가 증가할수록 오버플로우 패킷의 수와 언더플로우 횟수가 감소함을 알 수 있다. 그러나, 버퍼 크기가 3200 바이트 일 때까지는 연속된 감소추세를 보이나 이후로는 소폭의 오르내림이 있으므로 3200바이트 크기의 버퍼로 26ms의 오디오 세션의 지터를 보상함으로써 세션을 재생품질을 보장하면서 지나친 재생지연을 피할 수 있게 되었다. 다시 말하면, 전송지연 시간의 약 15배 정도의 버퍼를 사용하여 좋은 미디어 재생 품질을 얻을 수 있다. 이러한 방법으로 일정시간 세션의 지터를 관

장한 후, 이를 기반으로 변화량에 따라 버퍼의 크기를 조절하는 방법을 취함으로써 수신난이직 서비스 품질을 개선할 수 있었다. 기존의 Mbone 오디오 응용인 vat의 성능과 비교하여 비슷한 수준의 지연이 발생하였으나 깊이가 적은 재생을 얻어낼 수 있어서 재생 서비스 품질의 개선이 확인되었다.

3.3 RTP 자바 애플릿

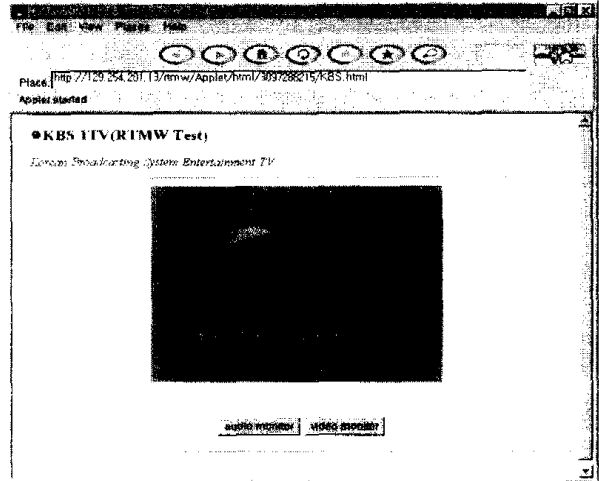
본 연구에서는 플랫폼 독립적인 환경을 지원하기 위해 자바 애플릿 방식을 적용하였다. 이 기간의 환경에서 RTP 오디오, 비디오를 수신하고자 하는 웹 사용자에게 자바 애플릿 방식은 프로그램을 자동 다운로드를 가능하게 하며, 이를 통해 여러 플랫폼에서 프로그램을 개발해야 하는 번거로움과 프로그램의 업그레이드의 필요성을 없앴다는 여러가지 장점을 가지고 있다. (그림 9)는 RTP 애플릿을 이용한 웹과의 통합 과정을 나타낸 것이다.



(그림 9) RTP 애플릿과 웹과의 통합 구조
(Fig. 9) Structure of Integrating RTP Applet into the Web

RTP 애플릿은 JDK(Java Developers Kit) 1.1과 자바 미디어 API 1.0을 이용하여 구현되었다. 현재 자바 미디어 API는 H.261, JPEG 비디오와 G711(PCM), GSM, DVI 오디오 스트림을 위한 재생을 지원하며, 이는 Mbone 응용과 호환 가능하다[14]. 현재 버전에서는 오직 수신만을 지원하나, 선 마이크로소프트에서는 곧 송신도 지원할 예정이어서, 본 응용을 이용하면 웹을 이용한 영상회의도 가능한 것으로 보인다. 또한 JDK1.1과 미디어 API의 해석을 위해 자바 애플릿 방식은 핫 자바 브라우저를 사용한다. 이때 핫자바의 Security Manager는 멀티캐스트 소켓의 사용에 제약을 가지고

있어서, 본 연구에서는 java.lang.SecurityManager 클래스를 수정하여 사용하였다. (그림 10)은 RTP 애플릿을 이용하여 오디오, 비디오를 수신한 예를 나타낸 것이다.



(그림 10) RTP 애플릿을 이용한 오디오, 비디오의 수신 예
(Fig. 10) Operation Example of RTP Applet

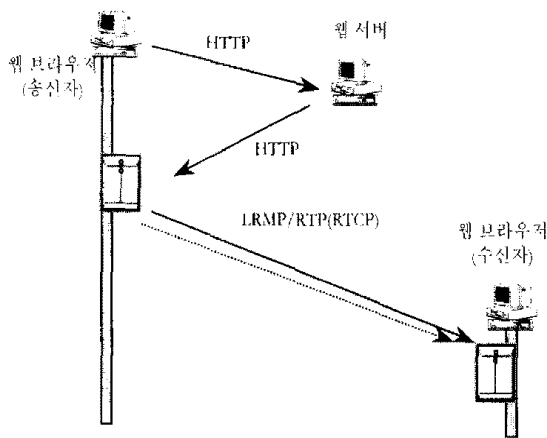
4. 멀티캐스트 미디어 확장 : 멀티캐스트를 이용한 HTML 문서의 분배

IP 멀티캐스트는 푸시 응용, 소프트웨어 분배, 멀티캐스트 프락시 캐쉬 등 새로운 응용을 가능하게 한다. 본 연구의 결과물인 웹 기반 멀티캐스트 멀티미디어 응용은 새로운 멀티캐스트 미디어나 응용의 확장을 용이하게 한다. 본 장에서는 오디오, 비디오 외에도 자바 환경하에서의 새로운 미디어의 확장성을 증명하기 위해, 멀티캐스트를 이용한 HTML 문서의 분배라는 새로운 응용을 설계하였다. 이 응용은 기존 웹 구조에 멀티캐스트 관련 기능을 확장함으로써 구현 가능하다.

4.1 멀티캐스트+HTTP

웹 문서의 멀티캐스트 전송은 오디오/비디오와 같은 실시간 데이터의 비신뢰적인 전송과는 달리 어느 정도의 신뢰성(reliability)이 요구되며, 이때 범위성(scalability)을 보장해야 한다. 그러나 IETF에서는 새로운 신뢰성을 지닌 멀티캐스트 전송 프로토콜 접근 방식은 전체 인터넷상에서는 적합하지 않으며 표준 프로토콜의 작업도 현재로서는 시기상조로 보고있다. 왜냐하면 잘못된 연산 하나가 전체 인터넷 망을 포화시킬 우려가

있기 때문이다[15]. 이를 위해 본 연구에서는 새로운 신뢰성 멀티캐스트의 개발을 고려하지 않고, 기존의 이미 개발된 신뢰성 프로토콜 중 구현이 용이하며, 경량(light-weight)의 단순한 프로토콜인 LRMP [16]를 선택하였다. 또한 기존 HTTP의 사용과 멀티캐스트의 통합을 위해 HTTP+멀티캐스트 접근 방식을 사용했다. 여기에서 '+'는 프로토콜 확장의 의미가 아닌 단순히 연속적인 흐름을 의미한다. (그림 11)은 이 프로토콜들의 흐름도를 나타낸 것이다. (그림 11)은 하나의 웹 문서를 송신자가 수신자에게 전송하기 위한 단계를 HTTP.request, HTTP.response, Multicast.send, Multicast.join 메시지로 흐름으로 보여준다. 즉, 웹 문서를 멀티캐스팅 하기 위해서는 먼저 웹 서버로부터 기존의 HTTP 프로토콜을 통하여 가져온 후, 이 문서를 멀티캐스트 프로토콜에 의해 지정된 그룹 수신자에게 전송하게 된다. 이 접근방식은 단순하며, 기존의 멀티캐스트 관련 프로토콜을 그대로 적용할 수 있다는 장점이 있다.



(그림 11) HTTP + 멀티캐스트 흐름도
(Fig. 11) A Flow of HTTP+Multicast

4.2 멀티캐스트 URL과 하이퍼링크

본 논문에서는 웹 문서 자체를 멀티캐스팅 하기 위해 문서 전송과 수신을 위해 필수적으로 요구되는 부분을 정의하여 URL 스킴을 확장하였다. "multicast" 프로토콜 식별자로 구분되는 이 스킴은 HTML 웹 문서를 전송하고 수신하는데 사용될 수 있도록 설계되었다. 제안한 멀티캐스트 URL 스킴은 다음과 같다.

```
multicast://multicast_address:port:tll/host/file_path/file_name
multicast://multicast_address:port
```

첫번째 스킴은 host/file_path/file_name의 웹 문서를 multicast_address, port, tll 값을 가지고 멀티캐스트 전송하는데 사용되며, 두번째 스킴은 이러한 전송된 문서를 수신하는데 사용되는 방식이다. 예를 들어 224.4.12.12 멀티캐스트 어드레스와 포트 4444로 구분되는 세션에 tll 16의 값으로 www.mbone.or.kr/index.html 문서를 전송/수신 하기 위한 URL은 각각 다음과 같이 사용할 수 있다.

```
multicast://224.4.12.12:4444:16/www.mbone.or.kr/index.html
multicast://224.4.12.12:4444
```

또한 사용자는 HTML 문서 내에 정의된 하이퍼링크에 의해 웹 문서를 네비게이션 하게 된다. 따라서 HTML 문서의 멀티캐스팅을 위한 하이퍼링크 앵커는 멀티캐스트 프로토콜 식별자를 포함하는 멀티캐스트 어드레싱을 위한 URL을 표현할 수 있어야 한다. 멀티캐스트 전송을 위한 HTML 하이퍼링크 형식은 다음과 같이 정의하였다.

```
<A HREF = multicast://multicast_address:port:tll/host/
file_path/file_name> description
</A>
```

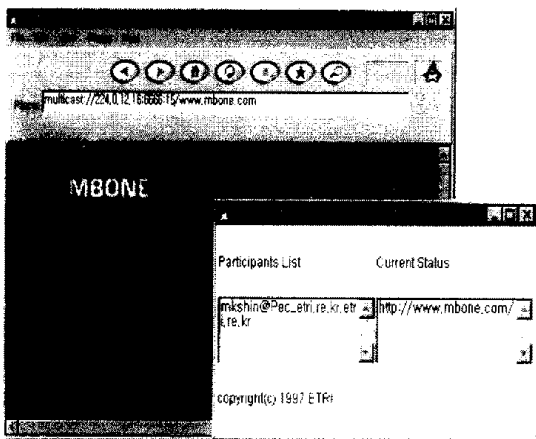
예를들어 HTML 문서내에 다음과 같이 사용할 수 있다.

```
<A HREF = multicast://224.4.12.12:4444:127/www.
mbone.or.kr/index.html> 멀티캐스트 하이퍼링크
</A>
```

4.3 핫자바 멀티캐스트 프로토콜 핸들러

앞서 설계한 웹 문서의 멀티캐스트 분배 응용을 플랫폼 독립적이며 기존 웹 브라우저의 수정 없이 3장에서 구현한 응용구조에 구현하기 위해 핫자바에서 지원되는 프로토콜 핸들러를 이용하여 구현하였다. 핫자바는 새로운 프로토콜을 추가하기 위해 브라우저 자체를 재컴파일없이 단순한 추가 형태로 확장할 수 있다. 즉, 핫자바에서 지원되는 프로토콜 핸들러 개념을 이용하면, 웹 브라우저내에서 기본적으로 지원되는 HTTP 프로토콜 외에 새로운 기능인 멀티캐스트 프로토콜 처리 부분을 브라우저에 확장하기 위해서는 기존의 브라우

적 소스의 수정없이 "add on" 개념으로 자연스럽게 추가가 가능하다. 현재 신 마이크로시스템즈에서 제공되는 핫자바 정식 버전은 1.1로 프로토콜 핸들러를 로컬에 인스톨하는 형태로 구축하여야 하지만, 이후 제공될 버전에서는 애플릿처럼 프로토콜 핸들러의 자동 다운로드가 가능함으로써 필요한 프로그램 자체를 다운받아서 사용하는 네트워크 컴퓨터의 환경에 적용할 수 있어 진정한 클라이언트 독립적인 환경으로 원격 회의를 할 수 있다. 현재 구현은 JDK 1.1을 이용하고 웹 브라우저로는 핫자바 1.1 버전을 사용하여 멀티캐스트 기능을 프로토콜 핸들러로 방식으로 구축하였다. 따라서 본 논문에서 확장한 멀티캐스트 URL을 웹 브라우저 자체에 아무런 수정없이 사용할 수 있다. (그림 12)는 구현된 환경에 따라 핫자바 브라우저에서 URL 장을 통해 멀티캐스트 하고자 하는 문서와 주소, 포트, TTL 등을 입력한 화면을 나타낸 것이다. 그림에서 멀티캐스트 우측 아래편 윈도우는 LRMP를 통해 주고받은 RTCP 패킷을 통해 현재 상태를 보여주도록 구현한 화면이다.



(그림 12) 핫자바 멀티캐스트 프로토콜 핸들러를 이용한 웹 문서의 분배 예
 (Fig. 12) An Example on Distribution of HTML Pages Using HotJava Multicast Protocol Handler

5. 관련 연구 비교

최근 웹에 실시간 멀티캐스트 멀티미디어 기능을 통합하려는 연구가 활발히 진행 중이다[17, 18]. 1996년 1월 W3C에서는 프랑스 INRIA 연구소를 중심으로 Real-time audio/video WG을 만들어 활동을 시작하였

다. 이 작업 그룹의 목적은 실시간 멀티미디어 웹 표준화 작업을 위해 방향을 제시하는 것으로써, 1996년 RTMW(Real Time Multimedia Web)라는 워크샵을 개최하여, 이 분야에 대한 연구를 활성화 하는 계기를 주었고, 최근 Synchronized Multimedia란 이름으로 변경되어 SMIL이라는 언어를 표준화 중에 있다.

Vosaic 시스템은 새로운 실시간 프로토콜인 VDP의 개발을 통해 웹에 실시간 멀티미디어를 수용한 최초의 시스템이며[19], RTMW'96 워크샵 중 Randa El Marakby의 연구는 RTP 미디어 전송, RTCP RR과 SDPS를 이용한 QoS 모니터링을 제공하는 응용을 플러그인으로 프로토타입핑 한 것이다[7]. 본 연구에서 개발한 멀티캐스트 멀티미디어 응용과 비교해 볼 때, [20][21]의 연구는 유니캐스트 환경만을 고려한 것으로, 웹 상에 실시간 오디오, 비디오 스트림의 이음매 없는 수신면에 초점을 둔 반면, 세션의 참가에 대한 구체적인 방법 제시는 미비했다. 미국 Precept사가 개발한 IP/TV는 MBone을 이용한 상업용 응용으로 멀티캐스트 된 오디오, 비디오를 PC에서 수신 가능하도록 한다. IP/TV의 프로그램 가이드는 세션 정보를 얻기 위해 본 응용과 마찬가지로 SDP를 사용한다. 본 연구의 결과 중 하나인 웹SDR은 IETF MMUSIC의 권고안과 IP/TV의 프로그램을 적용하여 개발한 것이다.

멀티캐스트를 이용한 웹 문서의 분배 기술에 있어서는 앞선 다섯 응용을 비교해 볼 수 있다. 구현된 최초의 멀티캐스트 웹 브라우저로는 1995년 EIT의 Vinary Kumar가 개발한 Shared Mosaic[20]을 들 수 있다. Shared Mosaic은 NCSA XMosaic 브라우저를 URL 멀티캐스팅이 가능하도록 수정한 것으로 웹 문서 자체를 멀티캐스팅 하지는 않는다. 같은 해 NCSA에서 개발된 Webcast[21]는 MBONE과 웹 브라우저를 연결하여 웹 문서 자체를 IP 멀티캐스트를 통해 전송하도록 지원하는 제3의 응용 프로그램으로 Mosaic CC(Common Client Interface)로 구현되었으며, RMP (Reliable Multicast Protocol)을 이용하여 웹 문서를 신뢰성있게 전송하려고 노력하였다. 1996년에 소개된 Gilles Dauphin의 mMosaic[22] 역시 NCSA Mosaic의 멀티캐스트 기능이 확장된 프로토타입 시스템이다. 위의 시스템들은 모두 Unix 환경의 Mosaic 브라우저를 기반으로 구현되었으며, 대부분 멀티캐스트의 결과를 웹 브

라우팅 입장에서 브라우징 하는데 초점을 맞춘 초기 프로토타입들이다.

최근에는 플랫폼-독립적인 환경을 지원하기 위해 자바를 이용한 연구 결과들이 선보이고 있다. 대표적으로 mWeb[23]과 WebCanal[24] 같은 시스템들이 개발되었다. 이들 프로그램들은 Webcast와 같이 Mbone과 웹 브라우저를 연결해주는 중간 연결 연결의 자바 응용으로써 신뢰성있는 멀티캐스팅을 위해 SRM(Scalable Reliable Multicast) 프로토콜을 확장한 SRFPD(Scalable Reliable File Distribution Protocol) /SR RTP(Scalable Reliable Real-time Transport Protocol)과 LRMP(Light Reliable Multicast Protocol)을 각각 사용한다. 위의 두 시스템들은 새로운 멀티캐스트 수송 프로토콜을 적용하여 신뢰성 있는 웹 문서 전송을 보장하고 자바 언어로 구현하여 플랫폼 독립적인 환경을 지원한다는 장점을 가지고 있다. 그러나, 이 프로그램들은 웹 브라우저 자체가 아닌 제3의 자바 응용이 웹 문서 멀티캐스트 기능을 처리함으로써 웹 구조면에서 멀티캐스트 지원을 위한 확장된 프로토콜, 어드레싱 기법, 문서 형식 등을 정의하여 확장된 웹 구조를 설계하지는 못했다. 웹 면에서 멀티캐스트 기능을 이음매 없이 확장하려는 노력은 표준화된 멀티캐스트 웹을 구축한다는 점에서 중요하다. 본 저자는 1997년 논문에서 멀티캐스트 기능을 WWW 구조면에서 이음매 없이 확장할 수 있도록 설계한 내용을 제안한 바 있다[25]. 본 논문은 이를 기반으로 웹-기반 멀티캐스트 멀티미디어 응용에 확장하여 설계한 것이다.

6. 결 론

본 연구는 다음과 같은 세가지 결과를 포함한다.

- RTP 플러그인, 자바 애플릿을 적용하여 RTP를 웹 응용에 이음매 없이 통합시킴.
- 웹 브라우저는 웹SDR에 의한 세션 페이지를 이용하여 Mbone 세션에 참가 가능.
- 핫자바 멀티캐스트 핸들러를 통해 멀티캐스트를 이용한 웹 문서의 분배 응용 개발.

본 연구는 실시간 멀티캐스트 멀티미디어 웹 응용을 구현한 것으로, 웹 상에 RTP/RTCP와 SDP, SAP

프로토콜을 통합함으로써 가능하다. 이를 위해 여러 가지 관련 요구사항들을 정리하고, 현재의 인터넷에서 즉시 적용 가능한 기술들을 선별하여 본 응용을 설계하고 구현하였다. 본 응용은 현재 IETF에서 진행중인 표준에 따라 적용함으로써, 기존 Mbone 응용과 호환 가능하며, 기존 Mbone 응용을 일일이 인스톨 할 필요 없이 웹 브라우저 하나만으로 Mbone 세션에 참가하여 멀티캐스트 되는 오디오, 비디오를 수신할 수 있도록 한 것이 특징이다. 특히 RTP 자바 애플릿 방식은 Mbone 사용자가 다양한 미디어 유형의 멀티캐스트를 수신받기 위해 프로그램들을 다운로드 받고, 업그레이드 하는 번거로움을 없앴으로써, 추후 차세대 멀티캐스트 인터넷 환경에 적용 가능하며, NC(Network Computer)와 같은 환경하에서 새로운 응용이나 미디어를 위한 이상적이고 확장성을 지닌 솔루션을 제공한다. 추가적으로 본 구조에서 미디어의 확장성을 보이기 위해 멀티캐스트를 통한 HTML 문서의 분배라는 새로운 응용 구조를 제안하였다.

참 고 문 헌

- [1] Kevin Savetz, Neil Randall, Yves Lepage, "MBONE: Multicasting Tomorrow's Internet," <http://www.northeast.com/savetz/mbone/toc.html>, 1996.
- [2] Tim Berners-Lee, Robert Cailliau, Jean-Francois Groff, Bernd Pollermann, "World-Wide Web: The Information Universe," 1992.
- [3] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," <ftp://ds.internic.net/rfc/rfc1889.txt>, 1996.
- [4] H. Schulzrinne, "RTP Profile for Audio and Video Conferences with Minimal Control," <ftp://ds.internic.net/rfc/rfc1890.txt>, 1996.
- [5] Philipp Hoschka, "Towards a Real-Time Multimedia Web," WWW 4th Conference BOF Presentation Material, <http://www.inria.fr/rodeo/personnel/hoschka/bof/bof-talk.ps>, 1996.
- [6] Randa El-Marakby, David Hutchison, "Integrating RTP into the World Wide Web," RTMW'96 Workshop, <http://www.w3.org/pub/WWW/Audio>

Video/RTMW96.html, 1996.

[17] V. Jacobson, Mark Handley, "SDP: Session Description Protocol." Internet Draft, <ftp://ietf.org/internet-drafts/draft-ietf-mmusic-sdp-04.txt>, 1997.

[8] Mark Handley, "Applying Real-time Multimedia Conferencing Techniques to the Web." RTMW'96 Workshop, <http://www.w3.org/pub/WWW/AudioVideo/RTMW96.html>, 1996.

[9] Precept, IP/TV, <http://www.precept.com/>, 1997.

[10] W3C Recommendation, HTML 4.0 Specification, <http://www.w3.org/TR/REC-html40/>, 1997.

[11] J. Bolot and T. Turletti, "A rate control mechanism for packet video in the Internet," Proc. IEEE Infocom'94, June 1994.

[12] Steven McCanne, Van Jacobson, Martin Vetterli, "Receiver-driven Layered Multicast," ACM SIGCOMM'96, August 1996.

[13] Netscape Communications Corporation., "Plug-in Guide," <http://developer.netscape.com/library/documentation/communicator/plugin/index.htm>, 1997.

[14] Sun Microsystems, "Java Media & Communications," <http://java.sun.com/products/java-media/>, 1998.

[15] Allison Mankin, Allyn Romanow, "IETF Criteria For Evaluating Reliable Multicast Transport and Application Protocols," Internet Draft.

[16] Tie Liao, "Light-weight Reliable Multicast Protocol as an Extension to RTP," http://webcanal.inria.fr/lrmp/lrmp_rtp.html, 1997.

[17] W3C, "Audio, Video, and Synchronized Multimedia." <http://www.w3.org/AudioVideo/>, 1997.

[18] Philipp Hoschka, "Sound and Video on the Web," 5th WWW Conference Tutorial, <http://www.inria.fr/rodeo/personnel/hoschka/WWW5-tutorial.ps.gz>, 1996.

[19] Zhigang Chen, See Mong Tan, Roy H. Campbell, Yongcheng Li, "Real Time Video and Audio in the World Wide Web," <http://choices.cs.uiuc.edu/Papers/New/vosaic/vosaic.html>, 1996.

[20] EIT, Shared NCSA Mosaic, http://www.eit.com/software/share_mosaic/, 1995.

[21] NCSA, "Collabarative document sharing via the MBONE," <http://www.ncsa.uiuc.edu/SDG/Software/XmosaicCCI/webcast.html>, 1995.

[22] Gilles dauphin, mMosaic, <http://sig.enst.fr/~dauphin/mMosaic/index.html>, 1996.

[23] Peter Parnes, Mattias Mattsson, Kare Synnes, Dick Schefstroem, "mWeb: a framework for distributed presentations using the WWW and the MBone," RTMW'96 Workshop, <http://www.w3.org/pub/WWW/AudioVideo/RTMW96.html>, 1996.

[24] Tie Lia, WebCanal: a Multicast Web Application, 6th WWW Conference, 1997.

[25] Myung Ki Shin, "Extending the WWW for Multicasting an HTML Document," INET'97, <http://pec.etri.re.kr/~mkshin/publication/inet/index.htm>, 1997.



신 명 기

e-mail : mkshin@pec.etri.re.kr

1992년 홍익대학교 전자계산학과 졸업(학사)

1994년 홍익대학교 대학원 전자계산학과 졸업(석사)

1994년~ 현재 한국전자통신연구원 표준연구센터 연구원

관심분야 : 인터넷 멀티캐스트 기술, 실시간 멀티미디어 전송 기술, 사바