

주문형 서비스 시스템을 위한 RTSP 프로토콜의 구현

배수영^o, 조창식, 마평수*, 김상욱**
한국전자통신연구원*, 경북대학교 컴퓨터과학과**

Implementation of RTSP Protocol for a On-Demand Service System

SuYoung Bae, ChangSik Cho, PyeongSoo Mah*, SangWook Kim**
ETRI*, Kyungpook National Univ.**

요 약

RTSP(Real Time Streaming Protocol) 프로토콜은 어플리케이션 레벨에서 실시간 데이터 전송의 세션과 제어를 표현한다. 하지만, RTSP는 모든 스트리밍 솔루션에 대한 내용을 포함하고 있어 그 내용이 포괄적이고, 회사별 자신의 응용과 환경에 맞게 RTSP 표준을 구현하고 있어 상호호환이 어렵다. 본 논문에서는 주문형 미디어 시스템에 필요한 기본 기능을 RTSP 표준을 이용하여 어떻게 구현해야 하는지에 대한 방법을 제시함으로써, RTSP 지원 업체간의 호환성을 유도한다.

1. 서론

HTTP를 기반한 RTSP(Real Time Streaming Protocol) 프로토콜[1]은 어플리케이션 레벨에서 실시간 데이터 전송의 세션과 전송 제어를 표현한다. 세션의 생성 및 디스크립션 정보 교환을 비롯하여, 스트리밍의 멈춤, 진행, 종료 등의 전송 제어까지 스트리밍을 위해 요구되는 서버와 클라이언트 사이의 모든 통신을 기술한다.

스트리밍 제어 프로토콜은 RTSP와 MMS(Microsoft Multimedia Server)[2]가 큰 주류를 이루고 있다. MMS의 경우, 서버 개발자는 솔루션 개발에 대한 사용료[3]를 부담해야 하고, MS가 제공하

는 강력한 MS 서버가 있으므로 다수의 일반 서버 개발자들은 RFC 표준으로 지정된 RTSP를 지원하고자 한다. 하지만, RTSP는 모든 스트리밍 솔루션에 대한 내용을 포함하고 있어 그 내용이 포괄적이고, 회사별 자신의 응용과 환경에 맞게 구현하고 있어 상호호환이 어렵다.

본 논문에서는 주문형 미디어 시스템에 필요한 기본 기능에 대해 RTSP 표준을 이용하여 어떻게 구현해야 하는지에 대한 방법을 제시함으로써, RTSP 지원 업체간의 상호 호환을 유도한다. 본 논문의 2장과 3장에서 RTSP 프로토콜에 대한 소개와 주문형 미디어 서비스를 위한 RTSP 프로토콜의 구현을 다루고, 4장에서 결론을 맺는다.

2. RTSP : Real Time Streaming Protocol

RTSP는 스트리밍의 세션 형성부터 종료까지, 클라이언트와 서버 사이에 전송되는 하나 이상 스트리밍에 대한 세션 관리 및 데이터 전송을 제어하는 프로토콜이다. RTSP는 HTTP에서 출발하였으나, 그와 구별되는 자신만의 메소드와 프로토콜 식별자를 가진 독립된 프로토콜로 개발되었다.

RTSP는 미디어 전송과 분리된 자신만의 채널을 사용하여, 자신과 연결된 데이터 전송 채널들을 관리한다. 하위 전송 프로토콜로 TCP, UDP를 모두 지원하지만 메시지의 전송 보장을 위해 TCP 사용이 권고된다.

RTSP 메시지는 요구(Request) 메시지와 응답(Response) 메시지로 구분된다. 일반적으로 클라이언트에서 서버로 스트리밍 제어 요구가 요구(Request) 메시지가 되고, 그에 대한 서버의 응답이 응답(Response) 메시지에 해당하지만, 영구적인 TCP 연결 경우에 한해, 서버가 클라이언트에게 요구(Request) 메시지를 보내는 것을 허용한다. 확장된 BNF(Augmented Extended Backus-Naur Form)로 RTSP 메시지의 기본 구조를 표현하면 다음과 같다.

```
Generic-message = start-line
                  *message-header
                  CRLF
                  [ message-body ]
```

요구(Request)/응답(Response) 메시지는 start-line과 하나 이상의 헤더 필드, 헤더 필드의 끝을 알리기 위해 삽입되는 CRLF(Carriage Return / Line Feed), 선택적인 메시지 내용(message-body)으로 구성된다.

start-line은 메시지의 핵심 부분으로, 두 메시지 타입에 따라 다른 구조를 가진다. 요구 메시지의 start-line은 Request-Line로, 응답 메시지는 Status-Line로 표현되고, 그 내용은 다음과 같다.

Request-Line = Method SP Request-URI SP RTSP-

Version

Status-Line = RTSP-Version SP Status-Code SP
Reason-Phrase CRLF

Request-Line은 메소드(Method), Request-URI, 프로토콜 버전(RTSP-Version), CRLF로 구성된다. SP(Space character)는 빈공간을 의미한다. Request-URI는 RTSP의 하부 연결 프로토콜과 서버 어드레스, 포트 번호, 프리젠테이션 또는 컨텐츠 파일 위치에 대한 추상 경로를 나타낸다. 메소드(Method)는 토큰(token)으로 표현되며, Request-URI가 나타내는 컨텐츠에 대한 메시지 송신자의 요구 내용을 담고 있다. RTSP에 의해 지원되는 메소드는 크게 송신자가 수행할 세션 및 스트리밍 제어 정보를 담고 있다. RTSP에 의해 지원되는 메소드는 11가지로서, DESCRIBE, ANNOUNCE, OPTIONS, GET_PARAMETER, PAUSE, PLAY, RECORD, REDIRECT, SETUP, SET_PARAMETER, TEARDOWN 이다.

요구(Request) 메시지에 대해 응답으로 전달되는 응답(Response) 메시지의 start-line은 프로토콜 버전(RTSP-Version), 상태코드(Status-Code), 상태 코드와 연결된 문자열(Reason-Phrase)로 구성된다. 상태코드(Status-Code)는 요구 메시지에 대한 처리 결과를 수로 표시하고 있으며, 1로 시작하는 세자리 상태코드 1XX는 처리 진행을, 2XX는 처리 성공, 3XX와 4XX, 5XX는 각각 스트리밍 서버 변환, 클라이언트, 서버 에러를 나타낸다. 상태 코드와 연결된 문자열(Reason-Phrase)은 상태 코드 대한 설명이다.

RTSP 메시지의 기본 구조에서 두번째 요소인 메시지 헤더(message-header) 필드는 공용 헤더(general-header), 요구 메시지 헤더(request-header) 또는 응답 메시지 헤더(response-header), 엔터티 헤더(entity-header)로 구성된다. 헤더는 메시지의 종류 및 내용에 따라서 메시지를 구성하는 필드가 다르다. 공용 헤더는 요구, 응답 메시지 둘 다에 공통으로 사용되는 내용으로서 메시지의 시퀀스 번호[CSeq]와 메시지가 만들어진 날짜[Date] 등이 여기에 해당된다. 요구 메시지 헤더(request-header)

는 요구 메시지에만 삽입되는 헤더로서, 라이브 스트리밍 참여 여부 필드[Conference], 세션 번호[Session], 스트리밍 전송에 대한 네트워크 설정 정보[Transport], 재생 구간[Range] 필드 등이 각 메소드에 맞게 배치된다. 응답 메시지 헤더(response-header)는 응답 메시지에 삽입되며, 요구 메시지 헤더(request-header)와 마찬가지로 세션 번호[Session], 재생 구간[Range] 필드 뿐만 아니라, 서버의 지원 메소드 공개[Public] 필드와 RTP 패킷 정보[RTP-Info] 필드 등이 각 메시지에 맞게 포함된다. 엔터티 헤더(entity-header)는 따라오는 메시지 내용(message-body)과 연결되어, 메시지를 기술하기 위해 사용된 언어 및 인코딩 정보 등을 담고 있다.

RTSP의 메시지의 기본 구조에서 마지막 요소인 메시지 내용(message-body) 필드는 DESCRIBE 메소드의 응답으로 제공되는 상세 정보 등을 저장하며, 일부 요구(Request), 응답(Response) 메시지에만 존재한다.

3. 주문형 서비스 시스템을 위한 RTSP 프로토콜 구현

RTSP는 여러 스트리밍 응용에 적용되기 위해 스트리밍 제어 메소드는 간단하게 하게 하고, 그 세부적인 필드를 자세히 두고 있다. 따라서, 실제 주문형 서비스를 제공하는 시스템의 상태보다 상대적으로 작은 수의 메소드가 지원되므로, 재생기에서의 상태 변화는 여러 메소드 및 필드를 조합하여 사용해야 한다. 다음 표 1은 주문형 서비스 시스템의 상태 변화를 위해 조합되는 RTSP 메소드이다.

표 1. 주문형 서비스 시스템의 상태 변화와 적용될 RTSP 메소드 조합

재생 상태	방향	사용 메소드
연결 설정	C→S	SETUP
재생	버퍼링	C→S PLAY
	정상재생, 역재생	C→S (PAUSE), (PLAY), PLAY
	빠른재생, 빠른되감기	C→S PAUSE, PLAY, PLAY

	점프	C→S	PAUSE, PLAY, PLAY
	잠시 멈춤	C→S	PAUSE
	재생 정지	C→S	TEAR-DOWN

◎ 연결 설정

클라이언트에서 서버로 연결 설정은 SETUP 메소드를 담은 요구 메시지와 그에 대한 서버의 응답으로 해결된다.

◎ 재생

버퍼링과 정상 재생, 역 재생, 빠른 재생과 빠른 되감기는 모두 PLAY 메소드를 가진 요구 메시지로 처리된다. 이를 위해 헤더는 다양한 필드를 가지고 있고, 주문형 시스템은 이를 적절하게 활용한다.

RTSP 서버는 클라이언트의 PLAY 요구 메시지에 대해 큐(queue)로 구현해 처리하도록 요구하고 있다. 이는 앞선 PLAY 메시지가 완전히 수행된 후, 다음 PLAY 메시지가 처리됨을 의미한다. 따라서, PLAY 메시지의 기능을 정확하게 파악해서 메시지의 요구 범위를 명확하게 설정해야 하며, 그렇지 않을 경우 원치 않은 결과를 초래할 수 있다.

□ 버퍼링

버퍼링은 PLAY 요구 메시지의 요구 메시지 헤더(request message header)에 Speed 필드와 Range 필드를 첨가함으로써 수행된다. Speed 필드는 일반 재생의 전송 속도를 기준으로 데이터의 전송 속도를 상대적으로 조절할 수 있게 하고, Range는 자신의 버퍼 사이즈와 컨텐츠의 비트율을 고려해서 해당 속도로 전송 받아야 하는 구간을 설정할 수 있다. Range 필드의 값은 NPT(normal play time)으로 표현되며, 재생 시작 시간을 0.0으로 하는 초 단위 값이다.

□ 정상 재생과 역 재생

(역)재생은 세션 시작 후 처음일 경우와 다른 재생 모드에서 변화된 경우로 나뉘어 메시지를 전송한다. 예를 들어, 빠른 (역)재생 모드 등의 다른 재생 모드에서 (역)재생 모드로의 변환은 이전에 있던 두 가지를 고려 해야 한다. 첫번째는 서버의 PLAY 요구 메시지 처리 방식에 의해서 새로운 PLAY 요구 메시

지를 처리하기 위해서 이미 진행중인 PLAY 요구메시지를 종료시켜야 한다. 이를 위해 PAUSE 요구메시지를 전달한다. 두 번째는 주문형 시스템 내의 버퍼에 든 데이터를 사용할 수 있는지를 확인한다. 현재 상태가 잠시 멈춤 상태이고 사용자로부터 새로 들어온 명령이 잠시 멈춤 상태가 되기 이전의 명령과 같다면, 버퍼는 재사용 할 수 있을 것이다. 이 경우, 버퍼링을 위한 PLAY 요구 메시지를 발생시킬 필요가 없다. 클라이언트는 서버로 최종 PLAY 요구메시지를 보내기 전에 이들 경우를 고려해서 필요한 선행 작업을 수행한다.

PLAY 요구 메시지에서 정상 재생과 역 재생을 구별은 요구 메시지 헤더(request message header)에 Scale 필드를 첨부함으로 해결된다. Scale 필드는 1.0은 정상 재생, -1.0은 역재생을 의미한다.

□ 빠른 재생, 빠른 되감기

(역)재생과 동작 과정은 같다. 다만 클라이언트에서 서버로 보내는 두개의 PLAY 요구 메시지에서 Scale 필드를 이용하여 재생의 속도를 조절한다. 기준 값은 1.0이며, 이 값의 정수배로 재생 속도를 조절한다.

□ 점프

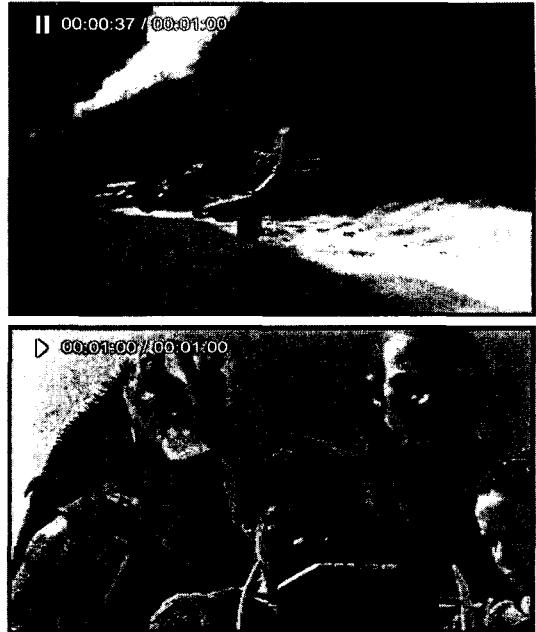
점프도 빠른 재생과 마찬가지로 정상 재생 모드와 동일하게 동작한다. 정상 재생 모드와 다른 점은 두개의 PLAY 요구 메시지, 즉 버퍼링을 위한 PLAY 요구 메시지와 점프 후 일반 재생을 위한 PLAY 요구 메시지에서 재생 구간을 나타내는 Range 필드를 사용자가 입력한 값으로 적절하게 설정하는 것이다.

● 잠시 멈춤(Pause)

잠시 멈춤은 세션 유지 상태에서 서버에서 스트리밍을 잠시 멈추도록 요구하는 것으로 PAUSE 메소드를 요구 메시지에 실어 서버에 전송한다.

● 재생 정지

클라이언트의 TEARDOWN 요구 메시지는 서버에게 해당 세션에 할당한 모든 리소스를 해제하고 세션 종료를 유도한다. 클라이언트는 TEARDOWN 요구 메시지를 서버에 보낸 후, 자신도 종료한다.



<그림1> VOD 재생 시스템 적용

4. 구현 및 결론

본 논문에서는 RTSP를 지원하는 주문형 서비스 시스템 간의 상호 호환을 유도하기 위해, 주문형 시스템의 여러 재생 모드에 대해 RTSP 표준을 어떻게 적용해야 하는지에 대한 방법을 소개하였다. 소개된 재생 모드는 세션 설정에서 버퍼링, 재생, 잠시멈춤, 되감기, 빨리감기, 점프, 세션 종료 등이며, 이들은 표 1과 같이 여러 개의 RTSP 메소드를 조합하여 완전하게 구현될 수 있었다. 그림 1은 본 논문에서 소개한 내용을 구현하여 VOD 재생 시스템을 구현한 예이다. 입력은 리모콘 입력이며, RTSP 서버는 자체 개발하였다.

[참고문헌]

- [1] RFC 2306, Real Time Streaming Protocol (RTSP), 1998.04.
- [2] MMS, Microsoft Media Server,
<http://sdp.ppona.com/>
- [3] <http://members.microsoft.com/consent/Info/TaskInfo.aspx?pkid=914>