

HEVC(H.264)를 지원하는

# Media Streaming Server 개발

공과대학 컴퓨터공학부

김 원 용 (2007-11663)

박 종 찬 (2008-11613)

정 창 주 (2008-11718)

# Overview

## Media Streaming Server

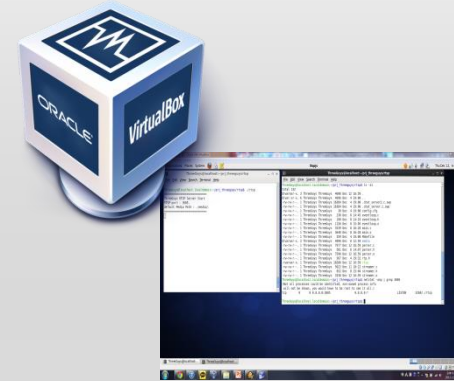


Client

### VLC Media Player

---

- 오픈소스 기반의 무료 미디어 플레이어
- 별도의 코덱팩 없이 대부분의 코덱 재생 지원
- 다양한 OS 지원(Windows, Linux, Mac OS X ...)
- 스트리밍 환경 지원



Server

### ThreeGuys RTSP Server

---

- C 언어로 구현된 RTSP 스트리밍 서버
- MPEG-4, H.264 인코딩 파일 재생 가능
- 다중 클라이언트 접속 지원
- 포트 설정 및 재생 파일 선택 가능



# Environment

## Linux Server



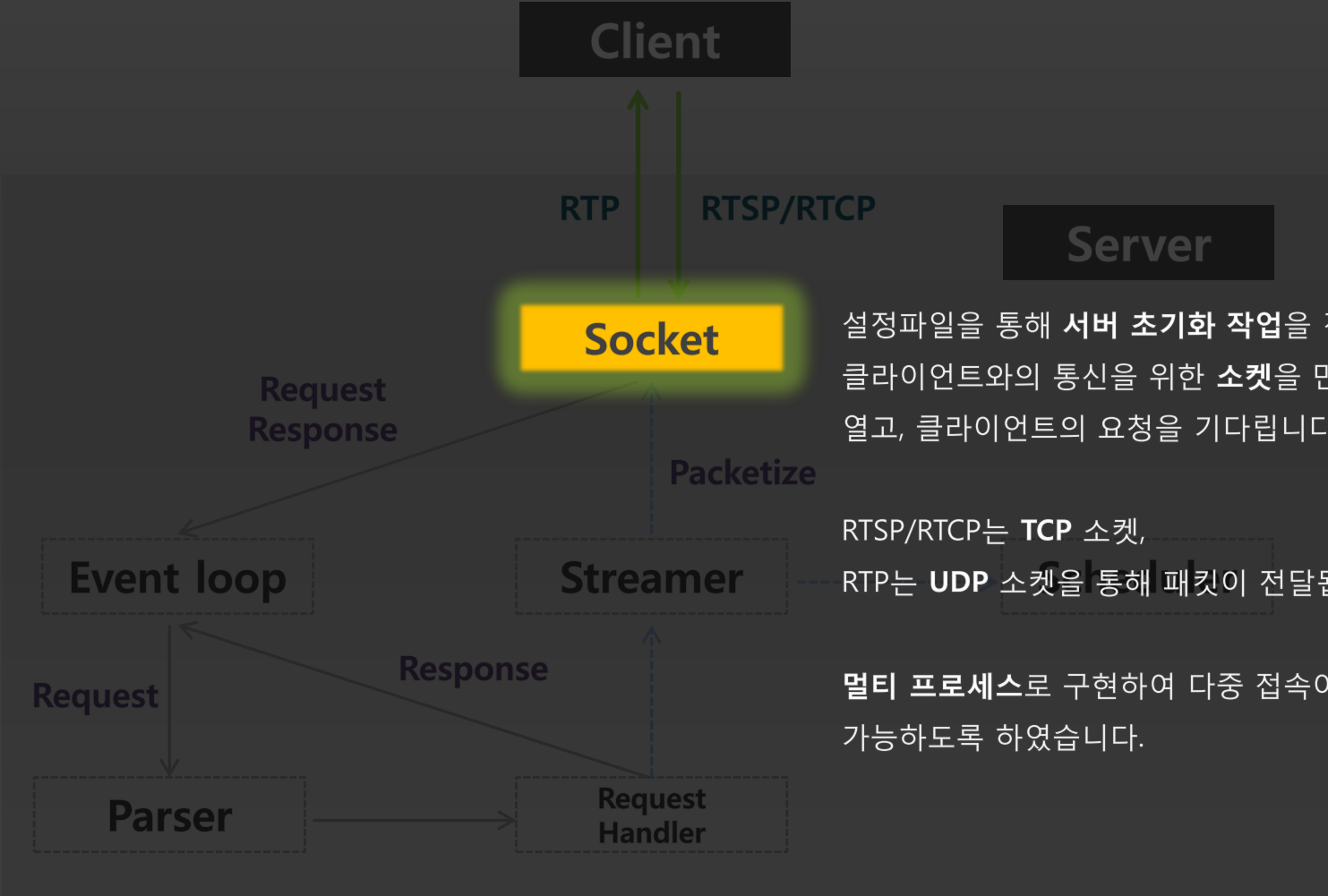
### Not Allowed

Red5, Darwin 등의 **오픈소스 스트리밍 서버**를 그대로 이용하는 것을 불허합니다.  
또한 라이브러리 및 각종 오픈소스 기반의 **프레임워크, 웹서버**를 사용하지 않습니다.



# Diagram

## 서버 주요 모듈 - Socket



설정파일을 통해 서버 초기화 작업을 진행한 후, 클라이언트와의 통신을 위한 소켓을 만들어서 열고, 클라이언트의 요청을 기다립니다.

RTSP/RTCP는 TCP 소켓, RTP는 UDP 소켓을 통해 패킷이 전달됩니다.

멀티 프로세스로 구현하여 다중 접속이 가능하도록 하였습니다.









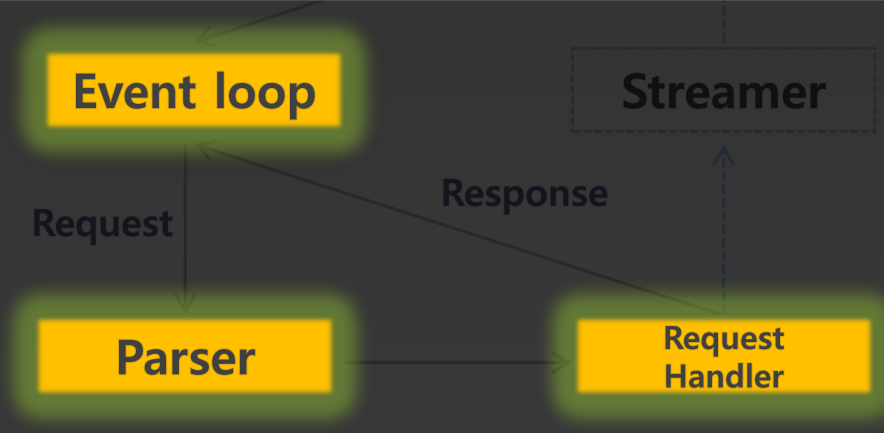
# Diagram

## 서버 주요 모듈 - Event loop + Parser + Request Handler

```

SETUP
Client
0000 08 00 27 da 27 10 08 00 27 00 50 bd 08 00 45 00 ..'.'...'.P...E.
0010 00 de 17 be 40 00 80 06 f0 a3 c0 a8 38 01 c0 a8 .....@... 8...
0020 38 66 23 7b 0b bd 5c 84 42 fd 2b e9 62 73 50 18 8f#{...}.B.+bsP.
0030 3f d1 7e e3 00 00 53 45 54 55 50 20 72 74 73 70 ?.~...SE TUP rtsp
0040 3a 2f 2f 31 39 32 2e 31 36 38 2e 35 36 2e 31 30 ://192.1 68.56.10
0050 32 3a 33 30 30 35 2f 73 61 6d 70 6c 65 32 2e 74 2:3005/s ample2.t
0060 73 2f 20 52 54 53 50 2f 31 2e 30 0d 0a 43 53 65 s/ RTSP/ 1.0..Cse
0070 71 3a 20 34 0d 0a 55 73 65 72 2d 41 67 65 6e 74 q: 4..Us er-Agent
0080 3a 20 4c 69 62 56 4c 43 2f 32 2e 30 2e 30 20 28 : LibVLC /2.0.0 (
0090 4c 49 56 45 35 35 35 20 53 74 72 65 61 6d 69 6e LIVE555 Streamin
00a0 67 20 4d 65 64 69 61 20 76 32 30 31 31 2e 31 32 g Media v2011.12
00b0 2e 32 33 29 0d 0a 54 72 61 6e 73 70 6f 72 74 3a .23)..Tr ansport:
00c0 20 52 54 50 2f 41 56 50 3b 75 6e 69 63 61 73 74 RTP/AVP ;unicast
00d0 3b 63 6c 69 65 6e 74 5f 70 6f 72 74 3d 35 33 39 ;client_ port=539
00e0 31 34 2d 35 33 39 31 35 0d 0a 0d 0a 14-53915 ....

```



클라이언트의 요청이 RTSP 프로토콜로 들어오면 RTSP 패킷을 파싱하여, 요청에 따른 필요 작업을 분기하도록 합니다.

클라이언트에서 처음 서버 연결을 요청하면 **OPTION** → **DESCRIBE** → **SETUP** → **PLAY** 순으로 요청하며, **PAUSE** 로 일시정지, **TEARDOWN** 으로 연결 종료를 요청할 수 있습니다.

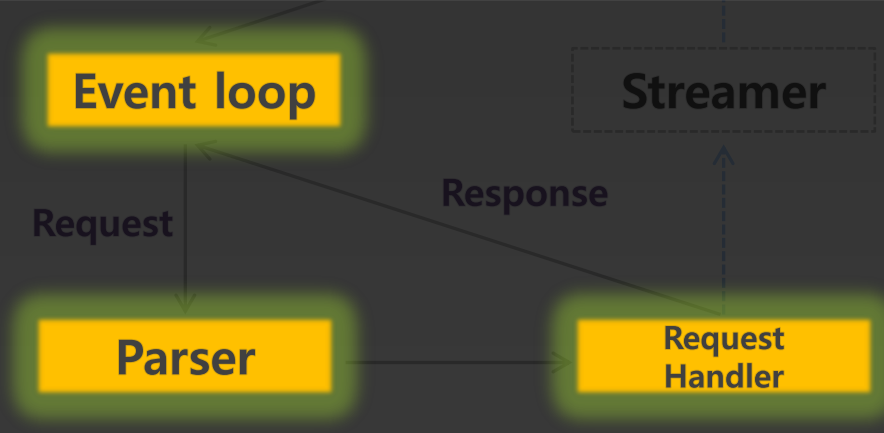
# Diagram

## 서버 주요 모듈 - Event loop + Parser + Request Handler

```

PLAY
0000 08 00 27 da 27 10 08 00 27 00 50 bd 08 00 45 00 ..'.P...E.
0010 00 d1 17 c3 40 00 80 06 f0 ab c0 a8 38 01 c0 a8 .....@...8...
0020 38 66 23 7b 0b bd 5c 84 43 b3 2b e9 63 3e 50 18 8f#{...C.+c>P.
0030 3f 9f 25 be 00 00 50 4c 41 59 20 72 74 73 70 3a ?.%...PL AY rtsp:
0040 2f 2f 31 39 32 2e 31 36 38 2e 35 36 2e 31 30 32 //192.168.56.102
0050 3a 33 30 30 35 2f 73 61 6d 70 6c 65 32 2e 74 73 :3005/sample2.ts
0060 2f 20 52 54 53 50 2f 31 2e 30 0d 0a 43 53 65 71 / RTSP/1.0..Cseq
0070 3a 20 35 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a : 5..User-Agent:
0080 20 4c 69 62 56 4c 43 2f 32 2e 30 2e 30 20 28 4c LibVLC/ 2.0.0 (L
0090 49 56 45 35 35 35 20 53 74 72 65 61 6d 69 6e 67 IVE555 streaming
00a0 20 4d 65 64 69 61 20 76 32 30 31 31 2e 31 32 2e Media v 2011.12.
00b0 32 33 29 0d 0a 53 65 73 73 69 6f 6e 3a 20 31 33 23)..session: 13
00c0 38 36 38 36 31 36 37 32 0d 0a 52 61 6e 67 65 3a 86861672 ..Range:
00d0 20 6e 70 74 3d 30 2e 30 30 30 2d 0d 0a 0d 0a npt=0.0 00-....

```

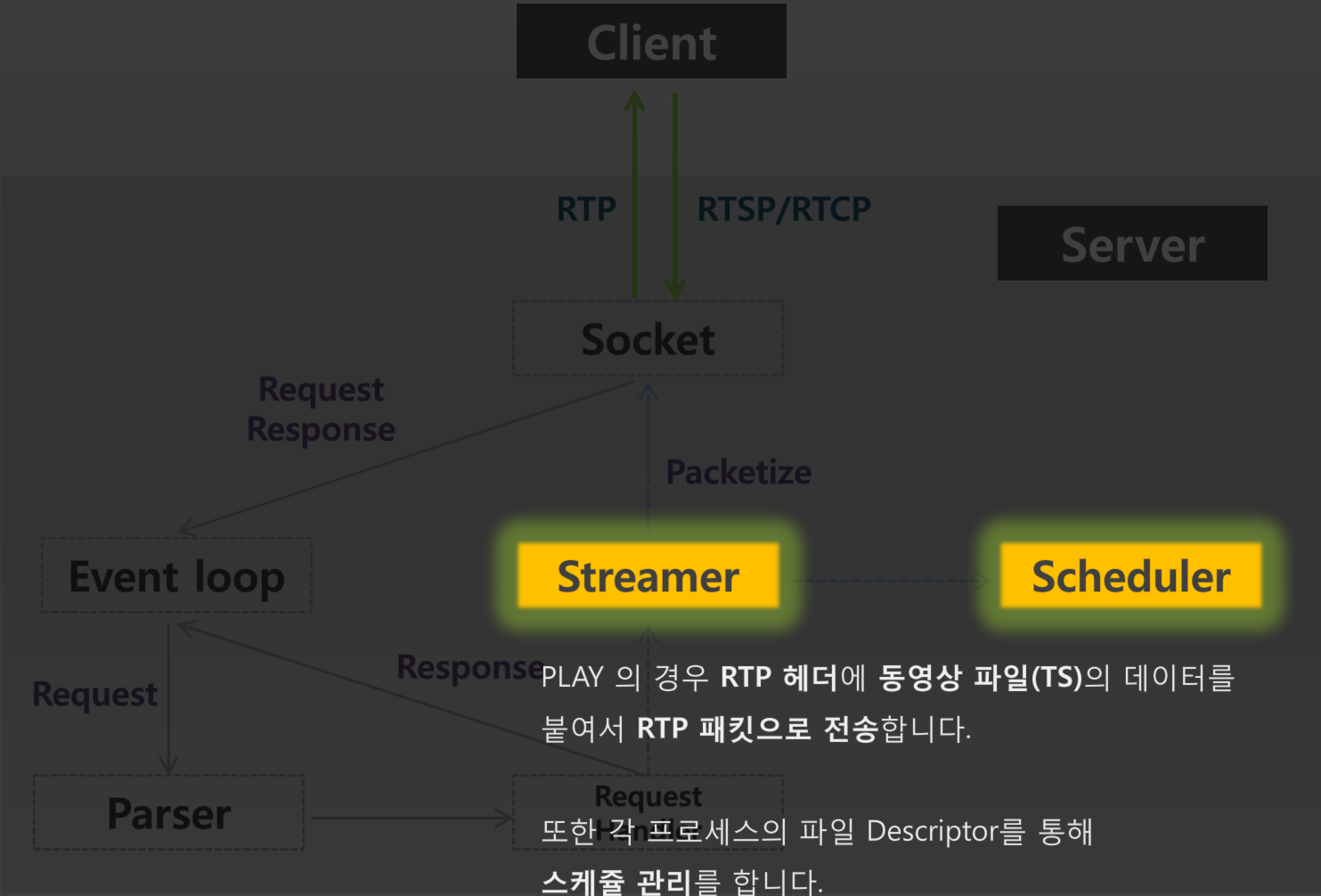


클라이언트의 요청이 RTSP 프로토콜로 들어오면 RTSP 패킷을 파싱하여, 요청에 따른 필요 작업을 분기하도록 합니다.

클라이언트에서 처음 서버 연결을 요청하면 **OPTION** → **DESCRIBE** → **SETUP** → **PLAY** 순으로 요청하며, **PAUSE** 로 일시정지, **TEARDOWN** 으로 연결 종료를 요청할 수 있습니다.

# Diagram

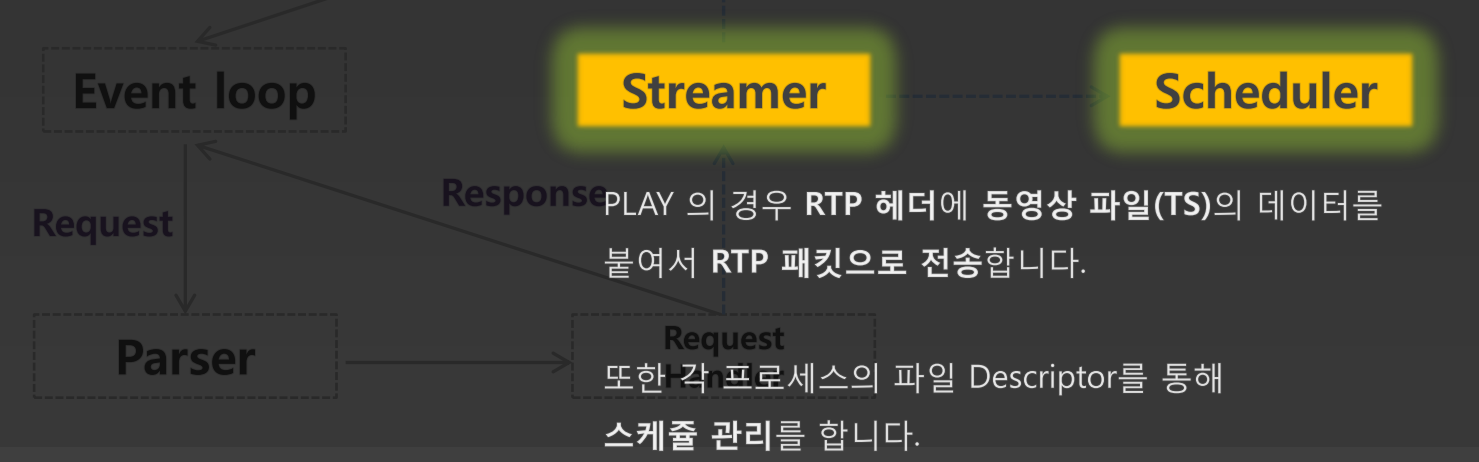
## 서버 주요 모듈 - Streamer + Scheduler



# Diagram

## 서버 주요 모듈 - Streamer + Scheduler

DATA										Client														
0000	08	00	27	00	50	bd	08	00	27	da	27	10	08	00	45	00	..	.	P	..	.	..	..	E
0010	02	5c	00	00	40	00	40	11	46	d9	c0	a8	38	66	c0	a8	..	..	@	@	F	..	..	8f
0020	38	01	1b	3a	ce	6a	02	48	4c	13	80	a1	d3	5b	00	01	8	..	..	j	H	L	..	[
0030	28	67	32	d5	a9	52	47	01	00	10	a0	df	4c	ad	0e	eb	(g2	..	..	RG	..	..	..	[
0040	2f	5a	36	0a	e0	f9	28	6e	09	bf	4d	3b	22	50	b5	de	/z6	..	..	(n	..	..	..	L
0050	b2	2f	6f	16	d8	23	95	5c	00	dc	eb	27	56	29	99	e9	/o	..	..	#	..	..	..	P
0060	de	da	79	22	fd	a8	a0	56	58	e6	c4	6b	85	a5	26	ba	..	..	y	..	..	v	v	&
0070	05	33	40	c5	3a	15	30	2b	d9	0d	9c	0a	c7	7c	6b	9c	..	..	@	..	..	..	..	k
0080	7b	64	f1	56	16	50	d8	89	10	d2	d1	97	67	fa	da	a4	{d	..	..	V	..	..	..	g
0090	c7	6e	5f	f0	ed	db	db	90	73	d9	a7	23	b4	14	90	ba	..	..	n	..	..	..	..	s
00a0	7b	36	db	fd	b1	3e	d8	6d	f3	ef	ad	3d	c5	bb	00	3b	{6	..	..	>	..	..	..	#
00b0	4e	f7	e8	69	40	ab	a8	91	3b	f0	06	6e	43	fc	90	32	..	..	n	..	..	..	..	..
00c0	fa	6e	8f	89	90	db	cf	0c	61	ac	64	2f	a5	12	00	44	..	..	n	..	..	..	..	..
00d0	01	27	22	73	09	08	42	10	bf	5d	8c	e7	0d	6c	74	0a	..	..	n	..	..	..	..	..
00e0	92	42	1c	e1	2f	0d	c2	bf	40	69	1a	e9	83	f8	20	0a	..	..	B	..	..	..	..	..
00f0	2a	f3	47	01	00	11	3b	2a	92	15	e1	44	e9	e6	d3	ab	..	..	B	..	..	..	..	..
0100	a9	d3	a4	7b	e9	07	f0	0f	97	e1	89	b7	09	65	6b	83	*G	..	..	..	..	..	..	..
0110	b3	0f	28	15	1a	0a	4c	96	73	e8	0d	2b	39	f6	1c	a3	..	..	{	..	..	..	..	..
0120	43	4d	76	53	f0	c6	9e	60	76	c6	e4	51	e9	04	d0	54	..	..	(	..	..	..	..	..
0130	da	41	af	61	90	48	3f	59	a5	47	e2	c8	ab	b5	2f	4a	CMVS	..	..	L	..	..	..	..
0140	43	b2	da	87	8b	af	e6	df	e4	5a	e6	ed	df	d7	5e	fe	..	..	A	..	..	..	..	..
0150	03	d9	4f	12	2e	d1	08	3d	fa	f0	94	91	ab	7c	9d	47	..	..	A	..	..	..	..	..
0160	34	b5	fd	4e	1d	0c	24	92	06	cf	bd	49	11	8f	a0	72	..	..	C	..	..	..	..	..
0170	4c	4a	e6	f9	df	81	dc	06	24	9e	94	77	72	b8	80	68	..	..	O	..	..	..	..	..
0180	9c	f0	72	20	c3	7a	8d	0f	40	52	ed	c6	f2	2c	09	9f	..	..	4	..	..	..	..	..
0190	d4	0e	e8	52	86	3a	b0	c7	7b	2f	e3	bd	8c	30	e5	ec	LJ	..	..	N	..	..	..	..

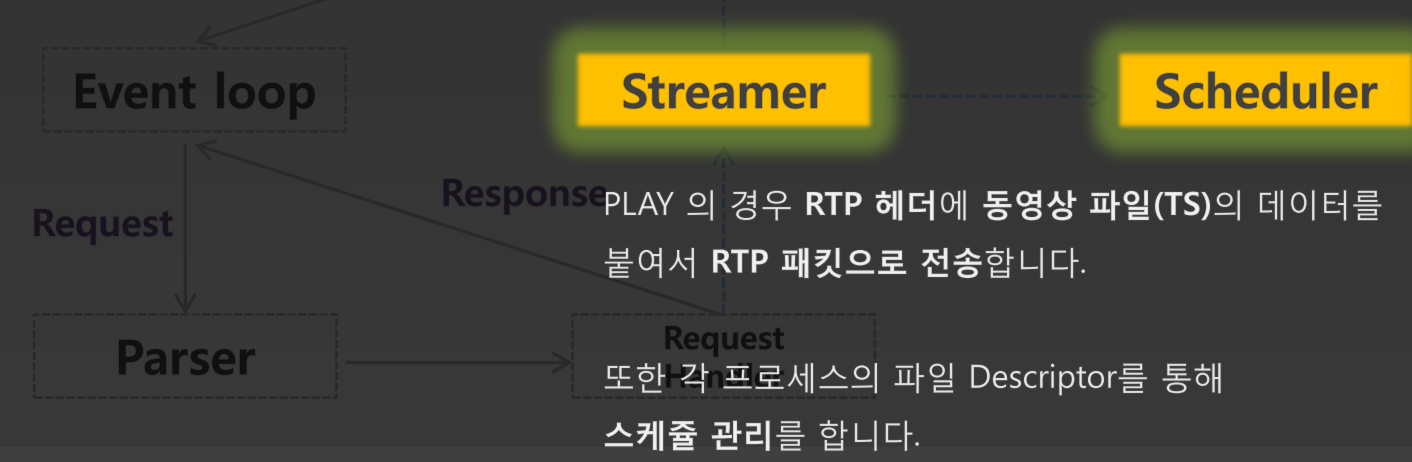
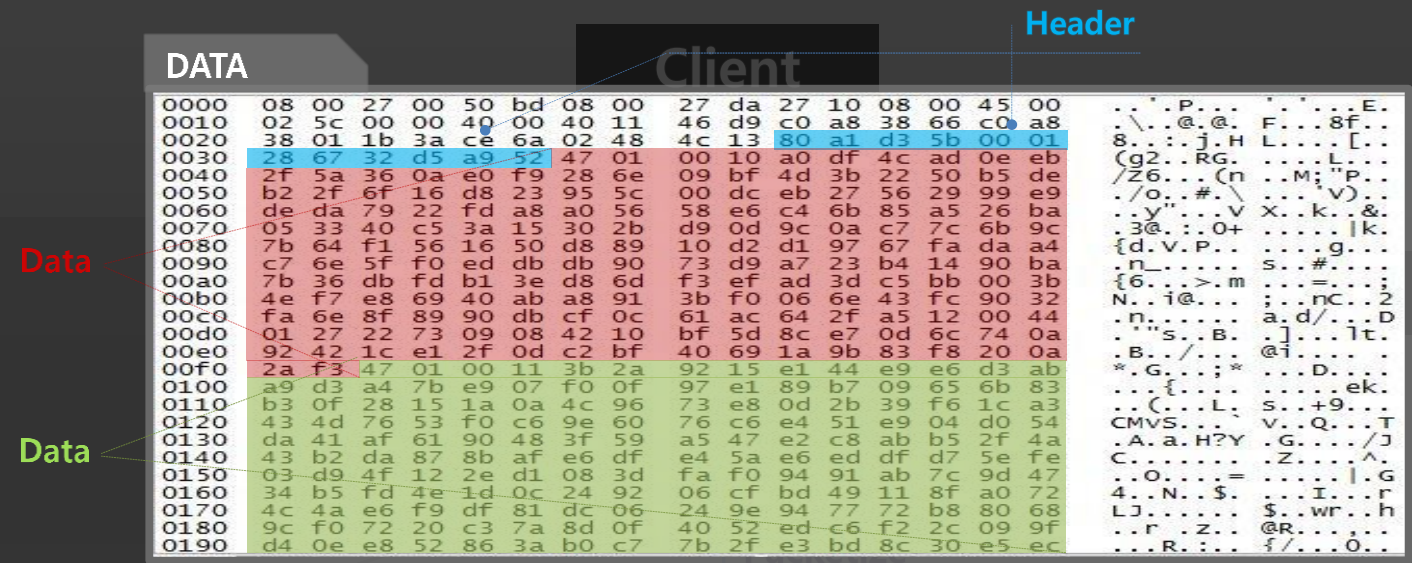


PLAY의 경우 RTP 헤더에 동영상 파일(TS)의 데이터를 붙여서 RTP 패킷으로 전송합니다.

또한 각 프로세스의 파일 Descriptor를 통해 스케줄 관리를 합니다.

# Diagram

## 서버 주요 모듈 - Streamer + Scheduler



# ISSUES

## Bit Rate에 따른 패킷전송 문제

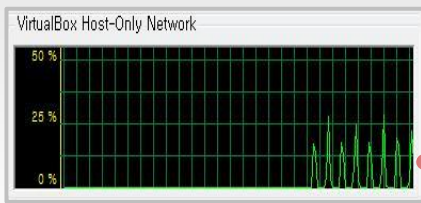
### 문제점

동영상의 Bit Rate를 고려하지 않을 경우, 너무 빠르거나 늦게 재생되는 문제 발생

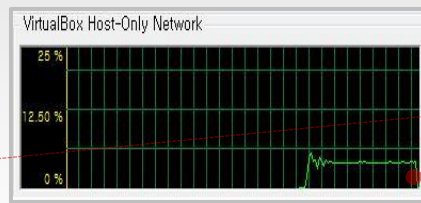
### 해결방법

1. **CBR (Constant Bit Rate)** 방식으로 동영상 인코딩 ※ Bit Rate = 4,000K Byte with FFmpeg
2. 매번 재생 시간을 확인하며 전송량 제어

- \* 현재까지의 전송량 : **Bit Count**
- \*  $\text{Bit Count} = (\text{Current Time} - \text{Start Time}) \times \text{Bit Rate}$
- \* 이 값보다 클 경우 **sleep** 하도록 함



적용 전(前)



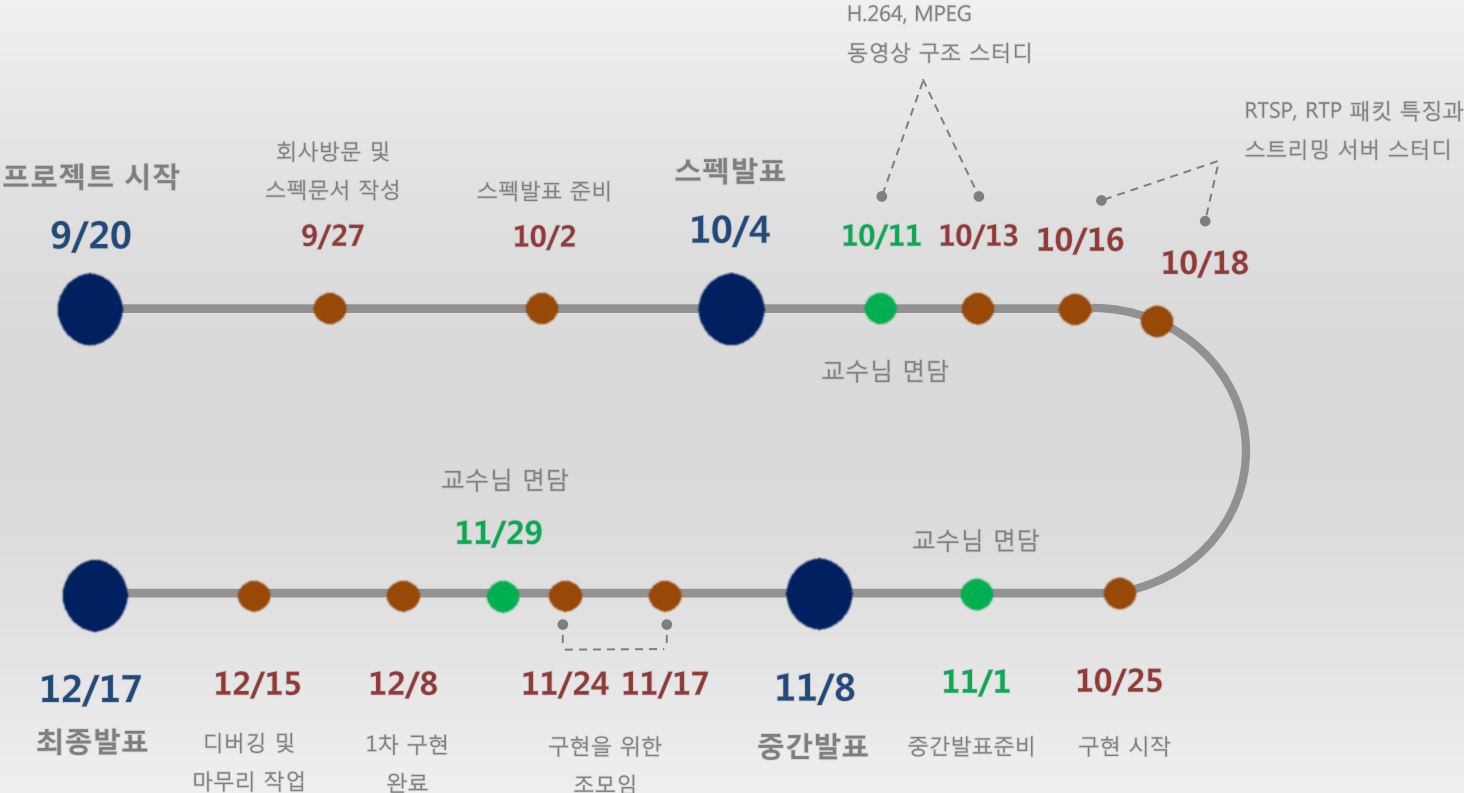
적용 후(後)

시간에 따른 네트워크 이용률이  
일정한 수치로 변화



# TASK

## Schedule





# TASK

## Task Allocation

### ● 김 원 용 (조장)

- 전반적인 프로젝트 관리 및 일정 조정
- 설계 및 디자인 총괄
- Parser 및 Request Handler 구현
- 소켓 프로그래밍 관련 주요 이슈 담당

### ● 박 종 찬

- 개발 환경 구축 및 GitHub 초기 연동 작업
- 테스트 셋 총괄
- Event Handler 구현
- Linux 프로그래밍 관련 주요 이슈 담당

### ● 정 창 주

- RTP 패킷화 구현(Streamer, Scheduler)
- RTSP, RTCP 주요 기능 총괄
- HEVC 타이밍 문제 해결
- MPEG, HEVC 등 주요 이슈 담당





**DEMO**

# Q & A