



术到极致，几近于道！

博客园

首页

新随笔

联系

订阅

管理

从零开始写一个发送h264的rtsp服务器(上)

转自：<http://blog.csdn.net/jychen105/article/details/47006345>

一、什么是RTSP

通常所说的rtsp协议其实包含三个协议：rtsp协议, rtp协议, rtcp协议

各协议运作流程概要：

第一阶段：rtsp协议负责沟通传输什么数据，传的是图像还是声音，还是两者混合？图像的话传是h264流，还是h265流，还是jpeg流？后续的rtp, rtcp协议是采用tcp还是udp，端口号是多少都是通过第一阶段的rtsp协议确定的。

第二阶段：通过rtp协议传输数据，rtcp进行网络传输质量的监控

第三阶段：通过rtsp协议中断整个协议的传输

二、RTSP协议详解

rtsp协议是个文本协议，运行于tcp协议之上，服务器默认监听端口554，当然也可以修改成其他端口，一般不改。

协议包格式

- 客户端协议格式

第一行：方法名：XXX\r\n

第二行：CSeq:XXX\r\n

第N行:XXX\r\n

\r\n

- 服务器端协议格式

第一行：RTSP/1.0 XXX\r\n

第二行：CSeq：XXX\r\n

第N行：XXX\r\n

\r\n

公告

昵称：明明是悟空

园龄：9年7个月

粉丝：203

关注：4

+加关注



<	2021年12月						>
日	一	二	三	四	五	六	
28	29	30	1	2	3	4	
5	6	7	8	9	10	11	
12	13	14	15	16	17	18	
19	20	21	22	23	24	25	
26	27	28	29	30	31	1	
2	3	4	5	6	7	8	

搜索

找找看

谷歌搜索

协议最终都是以两个\r\n分隔

rtsp通信精简版

步骤	客户端	服务端	含义
1	请求OPTIONS	回应OPTIONS	查询服务器支持哪些命令或者方法
2	请求DESCRIBE	回应DESCRIBE	查询服务器发送流的描述信息
3	请求SETUP	回应SETUP	告诉服务器以TCP or UDP建立RTP，RTCP连接，并告诉端口号
4	请求PLAY	回应PLAY	告诉服务器可以用RTP协议发送数据过来了
最后一步	请求TEARDOWN	关闭连接	告诉服务器关闭连接

rtsp通信详细版本

客户端：192.168.1.109

服务端：192.168.1.188

1. OPTIONS

客户端请求：

```
OPTIONS rtsp://192.168.1.188/h264 RTSP/1.0
CSeq: 2
User-Agent: LibVLC/2.1.2 (LIVE555 Streaming Media v2013.12.05)
```

服务端回应：

```
RTSP/1.0 200 OK
CSeq: 2
Date: Mon, Jul 21 2014 09:07:56 GMT
Public: OPTIONS, DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE, GET_PARAMETER, SET_PARAMETER
```

需注意的点：

- 回应的CSeq与请求的CSeq必须一致，后续rtsp命令下同。
- PAUSE, GET_PARAMETER, SET_PARAMETER这几个命令可无，也就是一些命令服务器可以不实现。OPTIONS，SETUP，PLAY是一定要有的，其他的未认真调查。

2. DESCRIBE

客户端请求：

```
DESCRIBE rtsp://192.168.1.188/h264 RTSP/1.0
CSeq: 3
User-Agent: LibVLC/2.1.2 (LIVE555 Streaming Media v2013.12.05)
Accept: application/sdp
```

服务端回应：

```
RTSP/1.0 200 OK
CSeq: 3
Date: Mon, Jul 21 2014 09:07:56 GMT
Content-Base: rtsp://192.168.1.188/h264/
Content-Type: application/sdp
Content-Length: 547

v=0
o=- 1405932398518315 1 IN IP4 192.168.1.188
s=Session streamed by "hua_an"
i=h264
t=0 0
```

我的标签

Linux(135)

web开发(84)

java(63)

C/C++(59)

android(46)

Linux内核(38)

chromium(29)

webrtc(26)

H264(17)

数据库(17)

更多

随笔档案

2020年5月(2)

2020年4月(4)

2019年12月(1)

2019年6月(1)

2019年5月(3)

2019年4月(1)

2019年1月(1)

2018年12月(4)

2018年11月(2)

2018年10月(3)

```
a=tool:LIVE555 Streaming Media v2012.10.24
a=type:broadcast
a=control:*
a=range:npt=0-
a=x-qt-text-nam:Session streamed by "hua_an"
a=x-qt-text-inf:h264
m=video 0 RTP/AVP 96
c=IN IP4 0.0.0.0
b=AS:500
a=rtptime:96 H264/90000
a=fmtp:96 packetization-mode=1;profile-level-id=64002A;sprop-parameter-sets=J2QAKq2EBUViuKxUcQgKisVxWKjiECSFITk8nyfk/k/J8nm5s00IEkKQnJ5Pk/J/J+T5PNzZpsqAUAW6bIAAAfQAAGGocAAABbjYAAD0JBe91lA=,KP4Briw=
a=control:track1
```

注意的点：

- 服务端回应了两个协议，一个是rtsp的回应，还回应了一个sdp协议(Session Description Protocol)，sdp协议最后一个只有一个\r\n。rtsp协议是两个\r\n结尾。
- sdp协议中描述了发送的流为h264，采样率为90000Hz (h264的采样率为90000HZ,因此时间戳的单位为1(秒)/90000,因此如果当前视频帧率为25fps,那时间戳间隔或者说增量应该为3600,即每个nalu单元的时间间隔为3600。此句话可以放在rtsp协议封装h264时理解。)
- sprop-parameter-sets为sps跟pps的base64编码组合。（sps，pps为h264 nalu单元的Sequence Parameter Sets (SPS) 和 Picture Parameter Set (PPS)，后续会介绍）

3. SETUP

客户端请求：

```
SETUP rtsp://192.168.1.188/h264/track1 RTSP/1.0
CSeq: 4
User-Agent: LibVLC/2.1.2 (LIVE555 Streaming Media v2013.12.05)
Transport: RTP/AVP/TCP;unicast;interleaved=0-1
```

服务端回应：

```
RTSP/1.0 200 OK
CSeq: 4
Date: Mon, Jul 21 2014 09:07:56 GMT
Transport: RTP/AVP/TCP;unicast;destination=192.168.1.109;source=192.168.1.188;interleaved=0-1
Session: 5C01EACE
```

此处为TCP建立RTP跟RTCP。RTP，RTCP都是利用此路连接，不再建立新连接，只是数据包格式不一样而以。

注意interleaved=0-1, 0为传送RTP数据，1为RTCP数据

4. PLAY

客户端请求

```
PLAY rtsp://192.168.1.188/h264/ RTSP/1.0
CSeq: 5
User-Agent: LibVLC/2.1.2 (LIVE555 Streaming Media v2013.12.05)
Session: 5C01EACE
Range: npt=0.000-
```

服务端回应

```
RTSP/1.0 200 OK
CSeq: 5
Date: Mon, Jul 21 2014 09:07:56 GMT
Range: npt=0.000-
Session: 5C01EACE
RTP-Info: url=rtsp://192.168.1.188/h264/track1;seq=28626;rtptime=98136845
```

播放，开始传输rtsp数据。同时告知开始序号为28626, 开始时间戳为98136845

5. 最后一步

2018年9月(1)
2018年8月(3)
2018年7月(9)
2018年6月(2)
2018年5月(8)
更多

阅读排行榜

1. java写入文件的几种方法分享(94317)

2. 如何在java程序中调用linux命令或者shell脚本(68602)

3. 1080P、720P、4CIF、CIF所需要的理论带宽(68040)

4. linux mysql 操作命令(66225)

5. Connection reset by peer的常见原因及解决办法(55079)

评论排行榜

1. 总结一下数据库的 一对多、多对一、一对一、多对多 关系(5)

2. 比较StringBuffer字符串内容是否相等?(3)

3. 二进制的计算（计算机为什么采用补码存储数据）(3)

4. 传指针和传指针引用的区别/指针和引用的区别（本质）(2)

5. C++中引用（&）的用法和应用实例(2)

推荐排行榜

客户端请求：

```
TEARDOWN rtsp://192.168.1.188/h264/ RTSP/1.0
CSeq: 7
User-Agent: LibVLC/2.1.2 (LIVE555 Streaming Media v2013.12.05)
Session: 5C01EACE
```

服务端回应：

关闭RTSP连接

三、RTP协议

根据之前rtsp的协议，rtp即可以通过tcp，也可以通过udp传输。但是数据包协议格式是一样的。

- tcp传输

协议格式	‘\$’	0	长度（=RTP协议头长度+RTP数据长度）	RTP协议头	RTP数据
长度信息	1字节	1字节	2字节	sizeof(RTP协议头)	N

- udp传输

协议格式	RTP协议头	RTP数据
长度信息	sizeof(RTP协议头)	N

– 数据传输采用网络字节序

– RTP数据长度小于MTU长度，一般是1500，目的是为了网络传输避免分片，未深究如何避免分片的。

– 传输的流媒体数据就是封装在RTP数据部分

RTP协议头

```
0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC  |M|    PT    |      sequence number      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                       timestamp          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier      |
+=====+
|         contributing source (CSRC) identifiers          |
|                   ....                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

//set CC = 0, so there is no CSRC
```

转成c语言结构体（小端字节序）：

```
typedef struct
{
    /* byte 0 */
    unsigned char csrc_len:4; /* CC expect 0 */
    unsigned char extension:1; /* X  expect 1, see RTP_OP below */
    unsigned char padding:1; /* P  expect 0 */
    unsigned char version:2; /* V  expect 2 */

    /* byte 1 */
    unsigned char payload:7; /* PT  RTP_PAYLOAD_RTSP */
    unsigned char marker:1; /* M   expect 1 */

    /* byte 2,3 */
    unsigned short seq_no; /*sequence number*/
```

1. 线程安全的单例模式(6)

2. java写入文件的几种方法分享(5)

3. JAVA 的wait(), notify()与synchronized同步机制(5)

4. jsp放在web-inf下的注意事项(5)

5. 表现层(jsp)、持久层(类似dao)、业务层（逻辑层、service层）、模型（javabean）、控制层（action）(5)

最新评论

1. Re:关于socket——SO_SNDBUF and SO_RECVBUF

“SO_”前缀是指“套接字选项”，所以是的，这些是每个套接字缓冲区的设置。通常有系统范围内的默认值和最大值。SO_RECVBUF更容易理解：它是内核分配的缓冲区的大小，用来保存从网络上到达的数据和拥...

--成熟里的秋天

2. Re:sk_buff封装和解封装网络数据包的过程详解

学习了

--咖啡猫二世

3. Re:netfilter的钩子——数据包在内核态得捕获、修改和转发

打扰了，请问您的代码是基于哪一版本的linux内核呀

--RiverGone

4. Re:总结一下数据库的 一对多、多对一、一对一、多对多 关系

通俗易懂，学习了，谢谢大佬们啊

--别说我太单纯

5. Re:总结一下数据库的 一对多、多对一、一对一、多对多 关系

```
/* byte 4-7 */
unsigned long timestamp; /*time stamp*/

/* byte 8-11 */
unsigned long ssrc;      /* stream number is used here. */
} RTPHeader; /*12 bytes*/
```

四、RTCP

rtcp主要是用来解决网络流控的，如果是自实现rtsp服务器传输实时流，可以不实现此部分。

如果是实现rtsp客户端，建立实现一个Recvive Report包就行了，此包主要是报告丢包率，网络延迟是多少，方便服务器端调节。
因为发的都是实时流，简单起见，recvive report封包时，丢包率为0，延迟也为0

五、h264是如何通过rtsp发送的

下篇

标签: rtsp

好文要顶

关注我

收藏该文



明明是悟空

关注 - 4

粉丝 - 203

+加关注

0
推荐

0
反对

« 上一篇： malloc基本实现

» 下一篇： 从零开始写一个发送h264的rtsp服务器(下)

posted @ 2018-03-19 17:25 明明是悟空 阅读(6937) 评论(0) 编辑 收藏 举报

刷新评论 刷新页面 返回顶部

登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) 博客园首页

编辑推荐：

- 理解ASP.NET Core - 模型绑定&验证
- [翻译].NET 6 中的 dotnet monitor
- .NET Core 如何配置 TLS Cipher（套件）？
- 记一次 .NET 某智能服装智造系统 内存泄漏分析
- 大学毕业三年的一些经历与思考

Mark.2021.5.17

--别说我太单纯



最新新闻：

- 泡泡玛特越来越“重” 潮玩需要新故事 (2021-12-09 12:11)
 - “1.5个” 村里人眼里的“张同学” (2021-12-09 12:00)
 - “塌房” or 过气，都是玲娜贝儿的结局 (2021-12-09 11:50)
 - 苹果出事了，十年来最严重的一次！ (2021-12-09 11:35)
 - B面宁德时代：超级工程背后的造富运动 (2021-12-09 11:20)
- » 更多新闻...