

	文件名称： RTSP 协议总体设计说明 版 本：A. 2016. 0901	页 码：1 0F 26 机密等级： 保密
发行日期：2016. 09. 01	生效日期：2016. 09. 01	文件编号：0001

RTSP 协议

总体设计说明

文档作者	刘才红	创建日期	2016/08/25
项目经理		批准日期	
会签审核记录表			
会 签 审 核 单 位	签 核	会 签 审 核 单 位	签 核

注释：

	文件名称： RTSP 协议总体设计说明 版 本：A. 2016. 0901	页 码：2 OF 26 机密等级： 保密
发行日期：2016. 09. 01	生效日期：2016. 09. 01	文件编号：0001

目录

1. 引言.....	3
1.1 编写目的.....	3
1.2 定义.....	3
1.3 参考资料.....	3
2. 系统概述.....	3
2.1 RTSP 概述.....	3
2.2 RTSP 特性.....	3
2.3 RTSP 状态转换图解.....	4
2.4 传输过程图解.....	5
2.5 RTSP 消息格式.....	5
2.6 RTSP 中的 C（CLIENT）与 S（SERVER）交互流程图解.....	6
2.7 RTSP 关键字段说明.....	6
2.9 RTP 协议概述.....	9
2.10 RTP 应用环境.....	9
2.11 RTP 报头格式.....	9
2.12RTCP 协议概述.....	10
2.13 RTCP 功能介绍.....	10
2.14 RTCP 报文格式.....	10
3. 软件体系总体结构设计.....	13
3.1 设计目标.....	13
3.2 软件系统状态转换图示.....	14
3.3 功能模块设计.....	14
4. 数据结构设计.....	14
4.1 数据结构设计.....	14
4.2 SPS PPS 解释（具体参考 H264 标准）.....	21
4.3 接口函数设计.....	22
5. 软件运行环境与开发工具.....	24
6. 软件体系的关键技术和难点.....	24
7. 附录（可选）.....	24
8. 相关文件.....	25
9. 相关记录.....	25
附录 A:.....	25

注释：

	文件名称： RTSP 协议总体设计说明 版 本：A. 2016. 0901	页 码： 3 OF 26 机密等级： 保密
发行日期：2016. 09. 01	生效日期：2016. 09. 01	文件编号：0001

1. 引言

1.1 编写目的

编写此文档的目的是为了开发人员对该协议有个更好的了解以及开发参考。

1.2 定义

列出本文中用到的专门术语的定义和外文首字组词的原词组。

1.3 参考资料

RTSP 参考《RFC2326》

RTP 参考《RFC3550》《RFC 3605》

RTCP 参考《RFC3550》

SDP 参考《RFC 4566》

2. 系统概述

2.1 RTSP 概述

RTSP（Real Time Streaming Protocol），参考标准为 RFC2326，实时流传输协议，是 TCP/IP 协议体系中的一个应用层协议。RTSP 在体系结构上位于 RTP 和 RTCP 之上，其使用 TCP 或 UDP 完成数据的传输；HTTP 与 RTSP 相比，HTTP 请求由客户机发出，服务器作出响应，使用 RTSP 时，客户机和服务器都可以发出请求，即 RTSP 可以是双向的；RTSP 是用来控制声音或影像多媒体串流协议，并允许同时多个串流需求控制，传输时所用的网络通信协定并不在其定义范围内。RTSP 协议默认端口：554，默认承载协议为 TCP。

2.2 RTSP 特性

2.2.1 流控分离

从控制逻辑上来说 RTSP 和 FTP 相似，流控和数据流是分开的。

2.2.2 可扩展性

RTSP 协议是基于文本的协议，所以具有较强的可扩展性。

2.2.3 安全机制

RTSP 使用网页安全机制。

2.2.4 记录设备控制

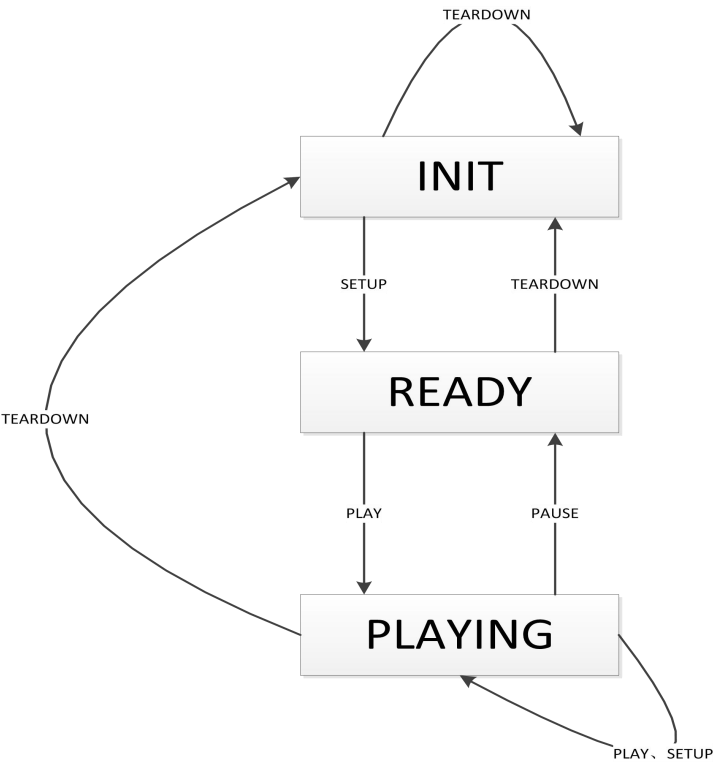
协议可控制记录和回放设备。

2.2.5 代理和防火墙友好

协议可由应用和传输防火墙代理，防火墙需要理解 SETUP 方法，为 UDP 流打开一个“缺口”。

	文件名称： RTSP 协议总体设计说明 版 本：A. 2016. 0901	页 码：4 OF 26 机密等级： 保密
发行日期：2016. 09. 01	生效日期：2016. 09. 01	文件编号：0001

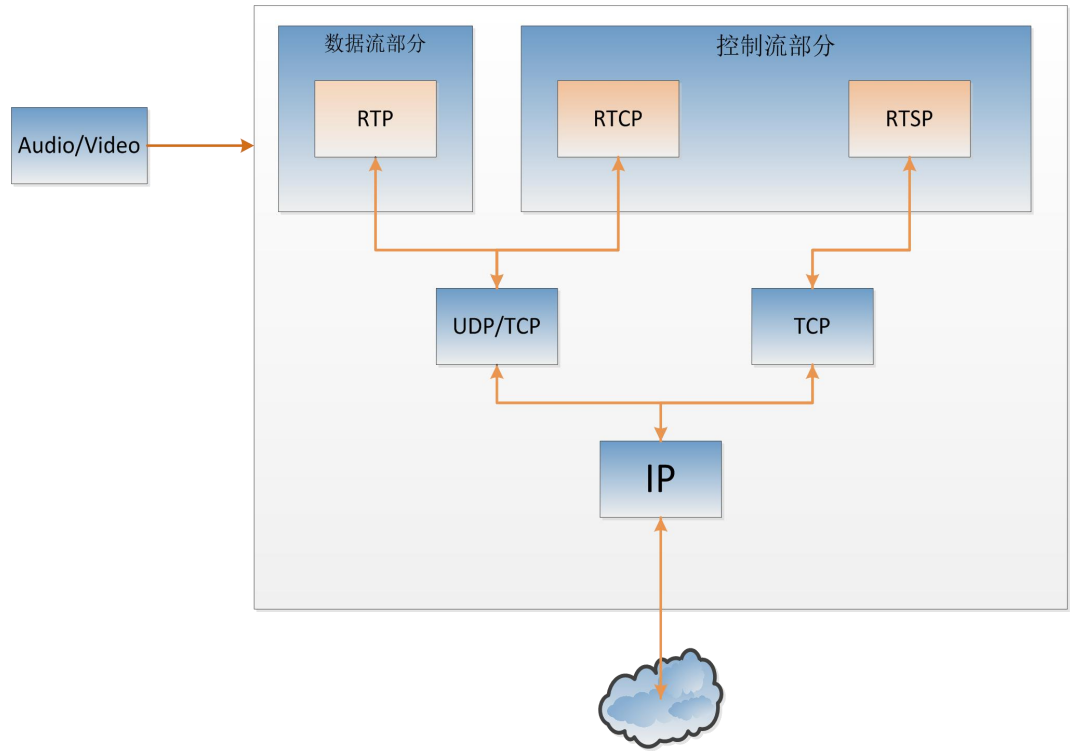
2. 3 RTSP 状态转换图解



注释：

	文件名称： RTSP 协议总体设计说明 版 本：A. 2016. 0901	页 码：5 0F 26 机密等级： 保密
发行日期：2016. 09. 01	生效日期：2016. 09. 01	文件编号：0001

2.4 传输过程图解



2.5 RTSP 消息格式

2.5.1 请求格式 (Request)

备注：CR：回车 LF：换行

方法 (Method)	URI	RTSP版本	CR LF
消息头	CR	LF	
CR	LF		
消息体	CR	LF	

注释：

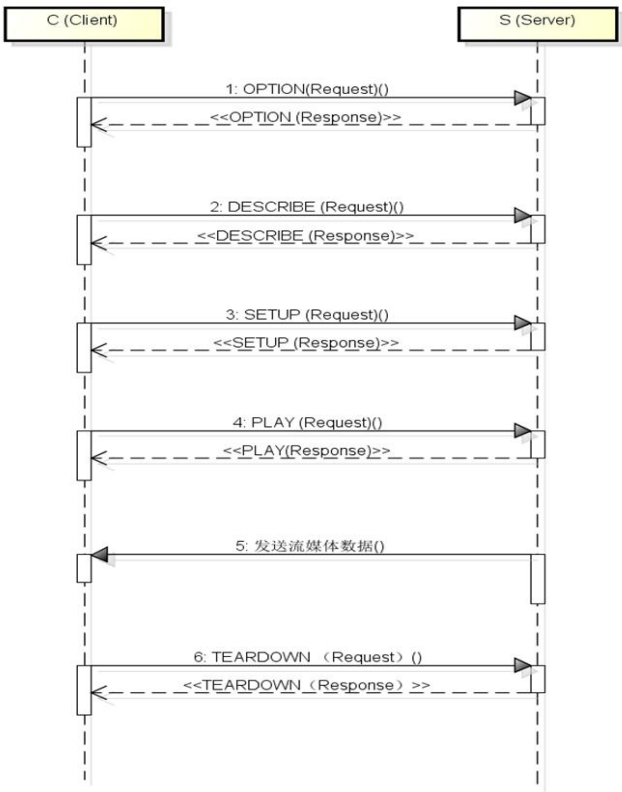
	文件名称： RTSP 协议总体设计说明 版 本：A. 2016. 0901	页 码： 6 OF 26 机密等级： 保密
发行日期：2016. 09. 01	生效日期：2016. 09. 01	文件编号：0001

2. 5. 2 回应消息格式（Response）

备注： CR： 回车 LF： 换行



2. 6 RTSP 中的 C（Client）与 S（Server）交互流程图解



2. 7 RTSP 关键字段说明

2. 7. 1 关键字：OPTIONS

得到服务器提供的可用方法（OPTION、DESCRIBE、SETUP、TEARDOWN、PLAY、PAUSE、SCALE、GET_PARAMETER、SET_PARAMETER）。

注释：

	文件名称： RTSP 协议总体设计说明 版 本：A. 2016. 0901	页 码：7 OF 26 机密等级： 保密
发行日期：2016. 09. 01	生效日期：2016. 09. 01	文件编号：0001

2.7.2 关键字：DESCRIBE

请求流的 SDP 信息。

注解：此处需要了解 H264 Law Data 如何生成 SPS PPS 信息。

2.7.3 关键字：SETUP

客户端提醒服务器建立会话，并建立传输模式。

注解：此处确定了 RTP 传输交互式采用 TCP（面向连接）还是 UDP（无连接）模式。

2.7.4 关键字：PLAY

客户端发送播放请求。

注解：此处引入 RTP 协议及 RTCP 协议。

2.7.8 关键字：PAUSE

播放暂停请求。

注解：此关键字经常用在录像回放当中，实时视频流几乎用不到。

2.7.9 关键字：TEARDOWN

客户端发送关闭请求

2.7.10 关键字：GET_PARAMETER

从服务器获取参数，目前主要获取时间参数（可扩展）

2.7.11 关键字：SET_PARAMETER

给指定的 URL 或者流设置参数（可扩展）

2.8 SDP 信息解释

```
v=<version>
o=<username> <session id> <version> <network type> <address type><address>
s=<session name>
i=<session description>
u=<URI>
e=<email address>
p=<phone number>
c=<network type> <address type> <connection address>
b=<modifier>:<bandwidth-value>
t=<start time> <stop time>
r=<repeat interval> <active duration> <list of offsets from
start-time>
z=<adjustment time> <offset> <adjustment time> <offset> ....
k=<method>
k=<method>:<encryption key>
a=<attribute>
a=<attribute>:<value>
m=<media> <port> <transport> <fmt list>
v = （协议版本）
o = （所有者/创建者和会话标识符）
```

注释：

	文件名称: RTSP 协议总体设计说明 版 本: A.2016.0901	页 码: 8 OF 26 机密等级: 保密
发行日期: 2016.09.01	生效日期: 2016.09.01	文件编号: 0001

s = (会话名称)
 i = * (会话信息)
 u = * (URI 描述)
 e = * (Email 地址)
 p = * (电话号码)
 c = * (连接信息)
 b = * (带宽信息)
 z = * (时间区域调整)
 k = * (加密密钥)
 a = * (0 个或多个会话属性行)

时间描述:

t = (会话活动时间)
 r = * (0 或多次重复次数)

媒体描述:

m = (媒体名称和传输地址)
 i = * (媒体标题)
 c = * (连接信息 — 如果包含在会话层则该字段可选)
 b = * (带宽信息)
 k = * (加密密钥)
 a = * (0 个或多个媒体属性行)

注解: 如果该协议仅仅用在视频流媒体当中, 不考虑带宽及其他会话信息, 常用到的 SDP 描述关键字段分别为:

1) m 字段: “m=<media> <port> <transport> <fmt list>”:

样例: m=video 0 RTP/AVP 96

解释: 作为媒体描述的重要组成部分描述了媒体信息的详细内容: 表示一个 Session 的 video 是通过 RTP 格式来传送的, 其中 Payload 值是 96, 传输端口开没有确定; 其中 Payload 值请参考 RFC 文档。

2) b 字段: “b=<modifier>:<bandwidth-value>”:

样例: b=AS:70

解释: 70 是 VIDEO 的 bitrate

3) a 字段: “a=<attribute>:<value>”:

样例: a=control:trackID=0

解释: 表示通过流媒体 0 来传输 VIDEO, a 的属性有很多种, 常用的是 control, 比如“a=range:npt=0-72.080000”

表示流媒体的长度, 再比如: “a=rtptime:96 MPEG4-GENERIC/32000/2” 表示音频为 AAC 的其 sample 为 32000 等等, 根据具体的要求来使用。

4) v 字段: 表示 SDP 的版本, 一般填 0。

5) o 字段: 定义了 SDP 一些源信息, “o=<username> <session id> <version> <network type> <address type><address>”

username: 该服务器的名称; session id: 会话 ID, 版本信息, 网络类型: IPV4 或 IPV6 传输地址;

样例: “o=StreamingServer 3331435948 1116907222000 IN IP4 192.168.1.123”

6) c 字段: “c=IN IP4 0.0.0.0 //connect 的信息, 分别描述了: 网络协议, 地址的类型, 连接地址”

7) t 字段: “t=0 0 //时间信息, 分别表示开始的时间和结束的时间, 一般在流媒体的直播的时移中见的比较多”。

注释:

	文件名称： RTSP 协议总体设计说明 版 本：A.2016.0901	页 码： 9 OF 26 机密等级： 保密
发行日期：2016.09.01	生效日期：2016.09.01	文件编号：0001

2.9 RTP 协议概述

RTP 全名是 Real-time Transport Protocol（实时传输协议）。它是 IETF 提出的一个标准，对应的 RFC 文档为 RFC3550（RFC1889 为其过期版本）。RFC3550 不仅定义了 RTP，而且定义了配套的相关协议 RTCP（Real-time Transport Control Protocol，即实时传输控制协议）。RTP 用来为 IP 网上的语音、图像、传真等多种需要实时传输的多媒体数据提供端到端的实时传输服务。RTP 为 Internet 上端到端的实时传输提供时间信息和流同步，但并不保证服务质量，服务质量由 RTCP 来提供。

2.10 RTP 应用环境

RTP 用于在单播或多播网络中传送实时数据。

2.10.1 简单的音频会议：语音通信通过一个多播地址和一对端口来实现，一个用于音频数据（RTP），另一个用于控制包（RTCP）。

2.10.2 音频和视频会议：音频和视频会议。如果在一次会议中同时使用了音频和视频会议，这两种媒体将分别在不同的 RTP 会话中传送，每一个会话使用不同的传输地址（IP 地址+端口）。如果一个用户同时使用了两个会话，则每个会话对应的 RTCP 包都使用规范化名字 CNAME（Canonical Name）。与会者可以根据 RTCP 包中的 CNAME 来获取相关联的音频和视频，然后根据 RTCP 包中的计时信息(Network time protocol)来实现音频和视频的同步。

2.10.3 翻译器和混合器。翻译器和混合器都是 RTP 级的中继系统。翻译器用在通过 IP 多播不能直接到达的用户区，例如发送者和接收者之间存在防火墙。当与会者能接收的音频编码格式不一样，比如有一个与会者通过一条低速链路接入到高速会议，这时就要使用混合器。在进入音频数据格式需要变化的网络前，混合器将来自一个源或多个源的音频包进行重构，并把重构后的多个音频合并，采用另一种音频编码进行编码后，再转发这个新的 RTP 包。从一个混合器出来的所有数据包要用混合器作为它们的同步源（SSRC，见 RTP 的封装）来识别，可以通过贡献源列表可以确认谈话者。

2.11 RTP 报头格式

V	P	X	CC	M	PT	序列号
时间戳						
同步信源（SSRC）标识符						
特约信源（CSRC）标识符						
payLoad(audio,video...)						

说明：

- 1) V: RTP 协议的版本号，占 2 位，当前版本号为 2。
- 2) P: 填充标志，占 1 位，如果 P=1，该报文在尾部填充一个或多个额外的 8 位组，他们不是有效载荷的一部分。
- 3) X: 扩展标志，占 1 位，如果 X=1，则在 RTP 报头后面跟一个扩展报头。
- 4) CC: CSRC 技术器，占 4 位，指示 CSRC 标识的个数。

注释：

	文件名称: RTSP 协议总体设计说明 版 本: A. 2016. 0901	页 码: 10 OF 26 机密等级: 保密
发行日期: 2016. 09. 01	生效日期: 2016. 09. 01	文件编号: 0001

5) M: 标记, 占 1 位, 不同的有效载荷有不同的含义, 对于视频标记一帧的结束; 对于音频, 标记会话的开始。

6) 同步信源 (SSRC) 标识符: 占 32 位, 用于标识同步信源。该标识符是随机选择的, 参加同一视频会议的两个同步信源不能有相同的 SSRC。

7) 特约信源 (CSRC) 标识符: 占 32 位, 可以有 0~15 个。每个 CSRC 标识了包含在该 RTP 报文有效载荷中的所有特约信源。

8) PT: 有效载荷, 占 7 位, 用于说明 RTP 报文中有效载荷的类型, 如 AAC, H264 等。

9) 序列号: 占 16 位, 用于标识发送者所发送的 RTP 报文的序列号, 每发送一个报文, 序号增加 1, 接收者通过序列号来检测报文丢失情况, 重新排列报文, 恢复数据。

10) 时间戳 (TimeStamp): 占 32 位, 时间戳反映了该 RTP 报文的第一个八位组的采样时刻。接受者使用时间戳来计算延迟和抖动, 并进行同步控制。

2. 12RTCP 协议概述

RTCP 的主要功能是: 服务质量的监视与反馈、媒体间的同步, 以及多播组中成员的标识。在 RTP 会话期 间, 各参与者周期性地传送 RTCP 包。RTCP 包中含有已发送的数据包的数量、丢失的数据包的数量等统计资料, 因此, 各参与者可以利用这些信息动态地改变传输速率, 甚至改变有效载荷类型。RTP 和 RTCP 配合使用, 它们能以有效的反馈和最小的开销使传输效率最佳化, 因而特别适合传送网上的实时数据。

2. 13 RTCP 功能介绍

2.13.1 提供数据发布的质量反馈

RTCP 是作为 RTP 传输协议的一部分, 与其他传输协议的流和阻塞控制有关。反馈对自适应编码控制直接起作用, 但 IP 多播经验表明, 从发送者收到反馈对诊断发送错误是至关重要的。给所有参加者发送接收反馈报告允许问题观察者估计那些问题是局部的, 还是全局的。诸如 IP 多播等发布机制使网络服务提供商之类的团体可能接收反馈信息, 充当第三方监控者来诊断网络问题。反馈功能由 RTCP 发送者和接收者报告执行。

2.13.2 RTCP 带有称作规范名字(CNAME)的 RTP 源持久传输层标识

如发现冲突, 或程序重新启动, 既然 SSRC 标识可改变, 接收者需要 CNAME 跟踪参加者。接收者也需要 CNAME 与相关 RTP 连接中给定的几个数据流联系。

2.13.3 可选功能是传送最小连接控制信息, 如辨识参加者。

2. 14 RTCP 报文格式

RTCP 包括五种类型的报文:

SR (Sender Report) 报文: 发送者报文。

RR (Receiver Report) 报文: 接收者报文。

SDES (Source Description) 报文: 源描述报文。

BYE (GoodBye) 报文: 退出报文。

APP (Application-Defined) 报文: 自定义报文

下面分别介绍一下各自的报文格式:

1) SR 报文:

	文件名称： RTSP 协议总体设计说明 版本：A. 2016. 0901	页 码： 11 0F 26 机密等级： 保密
发行日期：2016. 09. 01	生效日期：2016. 09. 01	文件编号：0001

V	P	RC	PT=SR=200	Length
发送者的SSRC				
NTP 时间戳 高字节			NTP时间戳 低字节	
RTP 时间戳				
发送的报文数				
发送的字节数				
SSRC_1 第一个源的SSRC				
丢包率		累计包丢失数		
接收到的扩展的最高序列号				
到达间隔抖动				
上一SR报文（LSR）				
自上一SR的时间（DLSR）				
SSRC_2（第二个源的SSRC）				
...				
特定协议扩展				

解释：

SR 报文由三部分组成：第一部分 8 字节长度，分别为：V 版本（2 比特，当前为 2）；填料 P（1 比特），如设置填料比特，该 RTCP 包在末端包含一些附加填料比特，并不是控制信息的基本部分，填料的最后一个比特统计了多少字节必须忽略，在复合的 RTCP 包中，复合包作为一个整体加密，填料比特只能加在最后一个单包的后面；接收报告块计数 RC（5 比特），该包中所含接收报告块的数目，零值有效；包类型 PT（8 比特），包含常数 200，用于识别 RTCP 的 SR 包；长度 Length（16 比特），以 32 比特字为单位，该 RTCP 包的长度减一，包括头和任何填料；SSRC（32 比特），SR 包发起的同步源标识符；

第二部分：发射机信息 20 比特，此域包含以下信息，NTP 时间标识（64 比特）；RTP 时间标识（32 比特）；发送报文数（32 比特）；发送字节数（32 比特）；

第三部分：零到多个接收报告块，SSRC_n（源识别符）32 比特；累计包丢失数 24 比特；接收到的扩展的最高序列号 32 比特；到达间隔抖动 32 比特；上一报文（LSR）32 比特；DLSR32 比特。

注：具体细节参考 RFC 文档

2) RR 报文格式

注释：

	文件名称： RTSP 协议总体设计说明 版 本：A. 2016. 0901	页 码：12 0F 26 机密等级： 保密
发行日期：2016. 09. 01	生效日期：2016. 09. 01	文件编号：0001

V	P	RC	PT=SR=201	Length
发送者的SSRC				
SSRC_1 第一个源的SSRC				
丢包率		累计包丢失数		
接收到的扩展的最高序列号				
到达间隔抖动				
上一SR报文（LSR）				
自上一SR的时间（DLSR）				
SSRC_2（第二个源的SSRC）				
...				
特定协议扩展				

解释：
接收者报告 RR 与 SR 包基本相同，除了包类型包含常数 201 和没有 SR 中 5 个字（NTP，RTP 和发射机包及字节计数），余下的与 SR 包的相同。

3) SDES 报文

V	P	SC	PT=SDES=202	Length
SSRC/CSRC_1				
SDES Items				
...				

SDES 报文分为三层结构，由头和数据块组成，数据块可以没有也可以有多个，具体描述如下：

(1) 项描述如下：

- ①版本 V，填充（P）、长度：如 SR 包中所描述。
- ②包类型（PT）：8 位，包含常数 202，标识 RTCP SDES 包。
- ③源计数（X）：1 位，包含在 SDES 包中的 SSRC/CSRC 块中，0 值有效，但没有意义

(2) 源描述项

CNAME=1	Length	User and Domain name	...
---------	--------	----------------------	-----

- ①CNAME：规范终端标识 SDES 项；CNAME 标识在 RTP 连接的所有参加者中应是唯一的。
- ②NAME：用户名称 SDES 项，描述源的真正名称。
- ③E-mail：电子邮件地址 SDES 项，邮件格式参考 RFC822 规定；连接期间电子邮件保持常数 3。
- ④PHONE：电话号码 SDES 项，PHONE=4 常量。
- ⑤LOC：用户地理位置 SDES 项，LOC=5 常量。

注释：

	文件名称： RTSP 协议总体设计说明 版本：A. 2016. 0901	页 码： 13 0F 26 机密等级： 保密
发行日期：2016. 09. 01	生效日期：2016. 09. 01	文件编号：0001

- ⑥TOOL：应用或工具名称 SDES 项，NOTE=7 常量。
- ⑦NOTE：通知/状态 SDES 项，NOTE=7 常量。
- ⑧PRIV：专用扩展 SDES 项。

4) BYE 报文

V	P	SC	PT=BYE=203	Length
SSRC/CSRC				
...				
Length		Length		

解释：

如混合器接收到一个 BYE 包，混合器转发 BYE 包，而不改变 SSRC / CSRC 标识。如混合器关闭，它也应该发出一个 BYE 包，列出它所处理的所有源，而不只是自己的 SSRC 标识。作为可选项，BYE 包可包括一个八位组计数，后跟表示离开原因的很多八位组文本，如：“cameramalfunction"或"RTP loopdetected"。字符串具有同样的编码，如在 SDES 中所描述的。如字符串填充包至下 32 位边界，字符中就不以空结尾；否则，BYE 包以空八位组填充。

5) APP 报文

V	P	SC	PT=APP=204	Length
SSRC/CSRC				
Name(ASCII)				
Application-dependent data				

解释：

APP 包用于开发新应用和新特征的实验，不要求注册包类型值。带有不可识别名称的 APP 包应被忽略掉。测试后，如确定应用广泛，推荐重新定义每个 APP 包，而不用向 IANA 注册子类型和名称段。

3. 软件体系总体结构设计

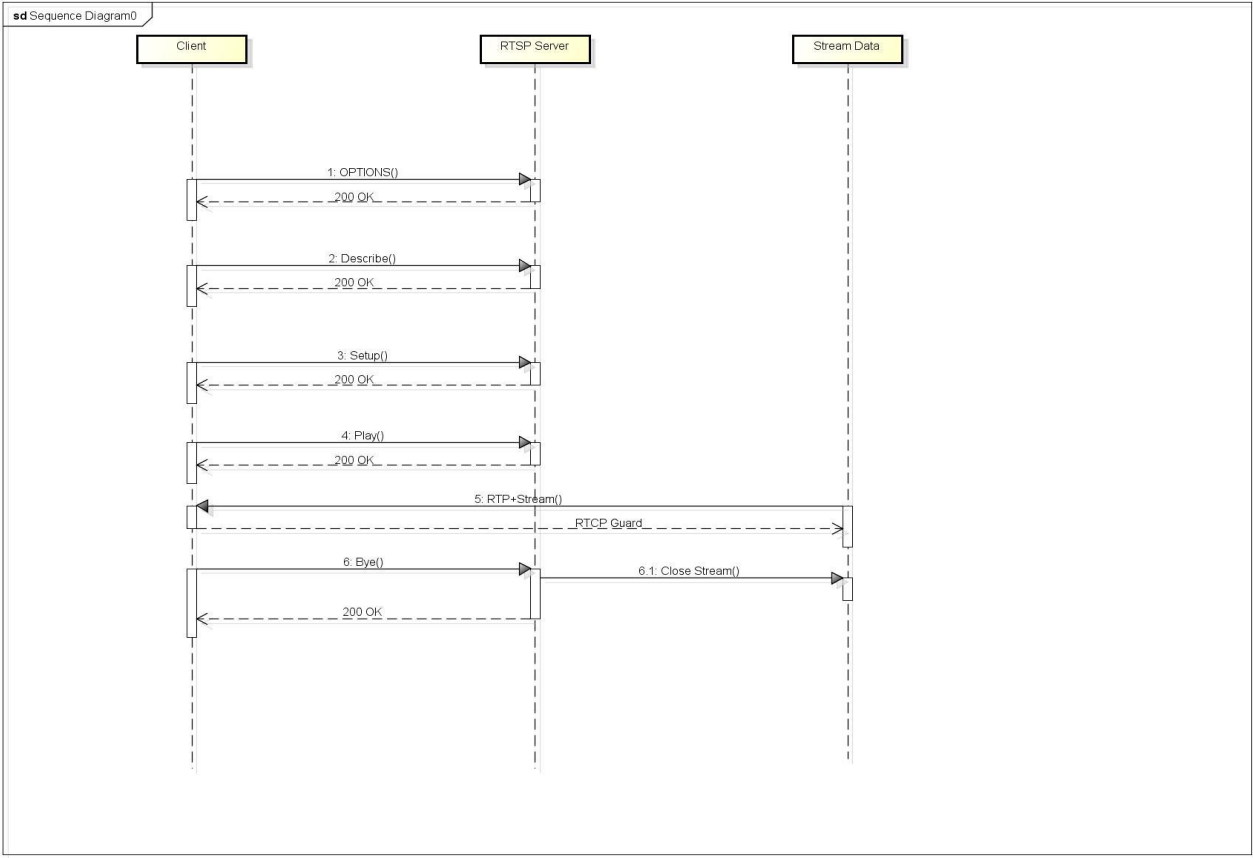
3.1 设计目标

针对系统方案中提出的相关功能、性能要求，提出解决方案、达到的目标。

注释：

	文件名称： RTSP 协议总体设计说明 版 本：A. 2016. 0901	页 码：14 0F 26 机密等级： 保密
发行日期：2016. 09. 01	生效日期：2016. 09. 01	文件编号：0001

3.2 软件系统状态转换图示



3.3 功能模块设计

3.3.1 功能模块的划分

- 整个 RTSP 流媒体主要有四个组成部分：
- 第一个部分：网络层，该层基于 TCP/IP 四层结构。
 - 第二个部分：应用层服务器构建，该层级取决于构建该服务器的系统（Linux，Windows）以及所涉及的并发访问数量。
 - 第三个部分：RTP/RTCP 传输。
 - 第四个部分：H264 Stream 及 Audio Stream。

4. 数据结构设计

4.1 数据结构设计

注释：

	文件名称: RTSP 协议总体设计说明 版 本: A.2016.0901	页 码: 15 OF 26 机密等级: 保密
发行日期: 2016.09.01	生效日期: 2016.09.01	文件编号: 0001

4.1.1 基本宏定义

```
#define RTSP_MAX_STREAM_TYPE 2
#define RTSP_MAX_CHAR_LEN 16
#define RTSP_MAX_STRING_LEN 128
#define RTSP_MAX_URI_LEN 256
#define RTSP_MAX_PROTOCOL_BUFFER 4096
#define RTSP_MAX_STREAM_TYPE 2//0-video 1-audio
```

4.1.2 RTSP 状态结构体

```
/*
*****
*****
*****RTSP Session Type Enum*****
*****Author:mark.liu*****
*****Date:2014/05/08*****
typedef enum
{
    RTSP_STATE_INIT            = 0,
    RTSP_STATE_READY           = 1,
    RTSP_STATE_PLAY            = 2,
    RTSP_STATE_STOP            = 3,
    RTSP_STATE_BUTT
}RTSP_Session_State;
```

4.1.3 RTSP Session Info 结构体

```
/*
*****
*****
*****RTSP Session Info Struct*****
*****Author:mark.liu*****
*****Date:2014/05/08*****
typedef struct
{
    struct list_head    list;//会话链表
    int                 nChId;//通道ID
    RTP_Transport_Type  eTransType;//RTP 传输协议(TCP、UDP、Mutil)
    RTP_Tcp-Token       stInterleaved[RTSP_MAX_STREAM_TYPE];//TCP传输模式下流的分隔符
    char                szId[RTSP_MAX_CHAR_LEN];// 会话SessionID
    char                szSSRC[RTSP_MAX_CHAR_LEN];//同步源ID
    char                szUserAgent[RTSP_MAX_STRING_LEN];//用户代理
    char                szUri[RTSP_MAX_URI_LEN];//Uri 连接
    int                 nRtspSocket;//RTSP socket
    char                szRemoteIpAddr[32];//此处仅限于IPV4
    char                szHostIpAddr[32];
    char                szRecvBuffer[RTSP_MAX_PROTOCOL_BUFFER];
    char                szSendBuffer[RTSP_MAX_PROTOCOL_BUFFER];
    int                 nMsgSize;
    unsigned int        u16RemotePort;//远程端口
    unsigned char       u8RequestStreamFlag[RTSP_MAX_STREAM_TYPE];//流类型标记
    unsigned short      u16RemoteRTPPort[RTSP_MAX_STREAM_TYPE];//UDP or Mutil RTP port and RTCP port
    unsigned short      u16RemoteRTCPPort[RTSP_MAX_STREAM_TYPE];//TCP RTP port and RTCP port
    unsigned long        u32LastSndReq;//上次发送Request 的Seq序列
    unsigned long        u32LastSndSeq;//上次回复Seq序列
    unsigned long        u32LastRecvSeq;//接收Seq序列
    int                 nLan;//判断内外网标记
    RTSP_Session_State  eState;//会话状态
    RTP_Sender_S        stRtpSender;//RTP发送结构
    pthread_t           sessThd;//一个任务
    Stream_Handle       Video_Handle;//Video Stream 句柄
    Stream_Handle       Audio_Handle;//Audio Stream 句柄
    char                nonce[17];//RTSP 认证nonce序列 (包括普通的认证和digest认证模式等)
    unsigned int         audio_timestamp;//音频时间戳
    unsigned int         video_timestamp;//视频时间戳
    unsigned short      fFrameType;//帧类型
    unsigned short      is_sync;//同步标记
    unsigned int         status_bit;//*0位-第一个视频帧类型(0-I帧 1-P帧) 1位-音视频是否同步 2位-是否有音频数据*/
}RTSP_Session_Info,*LPRTSP_Session_Info;
```

注释:

	文件名称: RTSP 协议总体设计说明 版 本: A.2016.0901	页 码: 16 OF 26 机密等级: 保密
发行日期: 2016.09.01	生效日期: 2016.09.01	文件编号: 0001

4.1.4 RTP TCP Token 结构体

```

/*****
/*****
/*****RTSP TCP Token Struct*****/
/*****Author:mark.liu*****/
/*****Date:2014/05/08*****/
typedef struct
{
    char *token;
    int opcode;
} RTP_Tcp-Token,*LPRTSP_Tcp-Token;

```

4.1.5 RTP Over Tcp Header 结构体

```

/*****
/*****
/*****RTP TCP Trans Head Struct*****/
/*****Author:mark.liu*****/
/*****Date:2014/05/08*****/
typedef struct
{
    unsigned char u8Dollar;
    unsigned char u8ChnId;
    unsigned short u16PacketSize;
}RTP_Over_Tcp_Header,*LPRTSP_Over_Tcp_Header;

```

4.1.6 RTP Sender Struct 结构体

```

/*****
/*****
/*****RTP Sender Struct*****/
/*****Author:mark.liu*****/
/*****Date:2014/05/08*****/
typedef struct
{
    RTP_Tcp-Token          stInterleaved[2];    /*interleaved for TCP video and audio*/
    unsigned int           nAudioSsrc;
    unsigned int           nVideoSsrc;
    unsigned int           nPayload[2];
    unsigned short         u16LastSn[2];        /*last rcv sn*/
    unsigned short         u16LastTs[2];        /*last rcv sn*/
    unsigned short         u16LastSndSeq[2];
    unsigned short         u16ChnId;            /*该sender所对应的通道号*/
    int                    nRtpTcpSock;
    int                    nRtpSock[RTSP_MAX_STREAM_TYPE];    /*send/rcv socket*/
    int                    nRtcpSock[RTSP_MAX_STREAM_TYPE];
    int                    nRtpPort[RTSP_MAX_STREAM_TYPE];
    int                    nRtcpPort[RTSP_MAX_STREAM_TYPE];
    struct sockaddr_in     addrRemote[2];
    unsigned long          u32LastSndPts[2];    /*上一帧数据的pts*/
    unsigned long          u32LastSndTs[2];    /*上一帧数据的timestamp*/
    char                   *pFrameBuf;
    unsigned char          szBuff[RTSP_MAX_PACKET_BUFF];    /*保存待发送数据的rtp buffer, 存放已经过rtp打包的数据*/
    unsigned int           nBuffLen;            /*buff中的实际需要发送数据的长度*/
}RTP_Sender_S,*LPRTSP_Sender_S;

```

注释:

	文件名称: RTSP 协议总体设计说明 版 本: A.2016.0901	页 码: 17 OF 26 机密等级: 保密
发行日期: 2016.09.01	生效日期: 2016.09.01	文件编号: 0001

4.1.7 RTSP Server Info 结构体

```

/*****
/*****
/*****RTSP Server Info Struct*****/
/*****Author:mark.liu*****/
/*****Date:2014/05/08*****/
typedef struct
{
    pthread_t thd;
    int nInit;
    int nSrvSocket;
    int nSrvPort;
    int nMaxCh;
    int nStreamNum;
    int nUseAuth;
    int nSrvExit;
    RTSP_SRV_STATE_E eState;
    RTSP_Session_Info sessList;
    unsigned long clientCnt[RTSP_MAX_CHN];
}RTSP_Server_Info, *LPRTSP_Server_Info;

```

4.1.8 RTSP Stream Callback 结构体

```

/*****
/*****
/*****RTSP Stream Callback Struct*****/
/*****Author:mark.liu*****/
/*****Date:2014/05/08*****/
typedef int RTSP_AV_HDL;

typedef RTSP_AV_HDL (openStream) (RTSP_Session_Info *pSess, int type);

typedef int (getVideoSdpInfo) (RTSP_Session_Info *pSess, char *profileLevelId, char *sps_pps);

typedef int (getVideoStream) (RTSP_Session_Info *pSess, char *pBuf, int *frameType, unsigned int *time);

typedef int (getAudioStream) (RTSP_Session_Info* pSess, char *pBuf, int *frameType, unsigned int *time);

typedef int (getAudioSdpInfo) (RTSP_Session_Info* pSess, unsigned int *pSample, int *pCh);

typedef int (setPlayType) (RTSP_AV_HDL pHdl, int nType);

typedef int (closeStream) (RTSP_Session_Info *pSess, int streamType);

typedef int (checkUserCb) (char *szUser, char *szPsw, int nGetPsw);

typedef int (checkAudioEnable) (int channel);
typedef struct
{
    checkUserCb      *funcCheckUser;
    openStream       *funcOpenStream;
    closeStream      *funcCloseStream;
    getVideoSdpInfo  *funcGetVideoSdp;
    getVideoStream   *funcGetVideoStream;
    getAudioSdpInfo  *funcGetAudioSdp;
    getAudioStream   *funcGetAudioStream;
    checkAudioEnable *funcCheckAudioEnable;
}RTSP_PLAY_CB, *LPRTSP_PLAY_CB;

```

注释:

	文件名称: RTSP 协议总体设计说明 版 本: A.2016.0901	页 码: 18 OF 26 机密等级: 保密
发行日期: 2016.09.01	生效日期: 2016.09.01	文件编号:0001

4.1.9 RTSP Server 配置结构体

```

/*****
/*****
/*****RTSP Server Config Struct*****/
/*****Author:mark.liu*****/
/*****Date:2014/05/08*****/
typedef struct
{
    int nMaxCh;
    int nStreamNum;
    int nRtspPort;
    int nUseAuth;
    int bRtspMode;//0被动上传视频,1主动上传视频
    int nRes[4];
}RTSP_Config,*LPRTSP_Config;
```

4.1.10 RTP PayLoad Struct 结构体

```

/*****
/*****
/*****RTP PayLoad Struct*****/
/*****Author:mark.liu*****/
/*****Date:2014/05/08*****/
typedef enum
{
    RTP_PT_ULAW          = 0,          /* mu-law PCMU*/
    RTP_PT_GSM           = 3,          /* GSM */
    RTP_PT_G723          = 4,          /* G.723 */
    RTP_PT_ALAW          = 8,          /* a-law PCMA*/
    RTP_PT_G722          = 9,          /* G.722 */
    RTP_PT_S16BE_STEREO  = 10,         /* linear 16, 44.1khz, 2 channel */
    RTP_PT_S16BE_MONO    = 11,         /* linear 16, 44.1khz, 1 channel */
    RTP_PT_MPEGAUDIO     = 14,         /* mpeg audio */
    RTP_PT_JPEG          = 26,         /* jpeg */
    RTP_PT_H261          = 31,         /* h.261 */
    RTP_PT_MPEGVIDEO     = 32,         /* mpeg video */
    RTP_PT_MPEG2TS       = 33,         /* mpeg2 TS stream */
    RTP_PT_H263          = 34,         /* old H263 encapsulation */
    RTP_PT_H264          = 96,         /* */
    RTP_PT_G726          = 97,         /* */
    RTP_PT_AAC           = 96,
    RTP_PT_ADPCM         = 99,
    RTP_PT_INVALID       = 127
}RTP_PT_E;
```

注释:

	文件名称: RTSP 协议总体设计说明 版 本: A.2016.0901	页 码: 19 OF 26 机密等级: 保密
发行日期: 2016.09.01	生效日期: 2016.09.01	文件编号: 0001

4.1.11 RTSP CODEC 类型结构体

```
typedef enum
{
    RTSP_CODEC_UNKNOWN = 0,

    RTSP_VIDEO_CODEC_START = 10,
    RTSP_VIDEO_H264,
    RTSP_VIDEO_MPEG4,
    RTSP_VIDEO_MJPEG,
    HI_RTSP_VIDEO_END,

    RTSP_AUDIO_CODEC_START = 20,
    RTSP_AUDIO_CODEC_AAC,
    RTSP_AUDIO_G711A,
    RTSP_AUDIO_G711U,
    RTSP_AUDIO_CODEC_END
} RTSP_CODEC_TYPE;
```

4.1.12 RTSP Trans Method 结构体

```
typedef enum
{
    /* method codes */
    RTSP_SETUP_METHOD      = 0 ,
    RTSP_DESCRIBE_METHOD   = 1 ,
    RTSP_REDIRECT_METHOD   = 2 ,
    RTSP_PLAY_METHOD       = 3 ,
    RTSP_PAUSE_METHOD      = 4 ,
    VOD_SESSION_METHOD     = 5 ,
    RTSP_OPTIONS_METHOD    = 6 ,
    RTSP_RECORD_METHOD     = 7 ,
    RTSP_TEARDOWN_METHOD   = 8 ,
    RTSP_GET_PARAM_METHOD  = 9 ,
    RTSP_SET_PARAM_METHOD  = 10,
    RTSP_EXTENSION_METHOD  = 11,
    RTSP_REQ_METHOD_BUTT
}RTSP_REQ_METHOD_E;
```

4.1.13 RTP 传输模式结构体

```
typedef enum
{
    RTP_TRANSPORT_UDP = 0,
    RTP_TRANSPORT_TCP,
    RTP_TRANSPORT_UDP_MUL,
    RTP_TRANSPORT_BUTT
}RTP_TRANSPORT_TYPE_E;
```

注释:

	文件名称: RTSP 协议总体设计说明 版 本: A.2016.0901	页 码: 20 OF 26 机密等级: 保密
发行日期: 2016.09.01	生效日期: 2016.09.01	文件编号:0001

4.1.14 RTP Header 结构体

```
typedef struct _RTP_HDR_S
{
    #if (BYTE_ORDER == LITTLE_ENDIAN)
        /* byte 0 */
        unsigned short cc      :4;    /* CSRC count */
        unsigned short x       :1;    /* header extension flag */
        unsigned short p       :1;    /* padding flag */
        unsigned short version :2;    /* protocol version */

        /* byte 1 */
        unsigned short pt      :7;    /* payload type */
        unsigned short marker  :1;    /* marker bit */
    #elif (BYTE_ORDER == BIG_ENDIAN)
        /* byte 0 */
        unsigned short version :2;    /* protocol version */
        unsigned short p       :1;    /* padding flag */
        unsigned short x       :1;    /* header extension flag */
        unsigned short cc      :4;    /* CSRC count */
        /*byte 1*/
        unsigned short marker  :1;    /* marker bit */
        unsigned short pt      :7;    /* payload type */
    #else
        #error YOU MUST DEFINE BYTE_ORDER == LITTLE_ENDIAN OR BIG_ENDIAN !
    #endif

    /* bytes 2, 3 */
    unsigned short seqno  :16;    /* sequence number */

    /* bytes 4-7 */
    unsigned long ts;           /* timestamp in ms */

    /* bytes 8-11 */
    unsigned long ssrc;        /* synchronization source */
}RTP_HDR_S;
```

注释:

	文件名称： RTSP 协议总体设计说明 版 本：A.2016.0901	页 码： 21 0F 26 机密等级： 保密
发行日期：2016.09.01	生效日期：2016.09.01	文件编号：0001

4.1.15 RTP 之 H264 NUAL 结构体

```
typedef struct tagFUIndicator
{
    char TYPE : 5;      /* set to 28 or 29 */
    char NRI : 2;       /* the same to nal */
    char F : 1;         /* set to 0 */
}PACKED FUIndicatorDef;

typedef struct tagFUHeader
{
    char Type : 5;      /* set to nal type */
    char R : 1;         /* reserve */
    char E : 1;         /* end flag */
    char S : 1;         /* start flag */
}PACKED FUHeaderDef;

typedef struct tagFU_A
{
    /*根据RFC3984 FU-A的RTP荷载格式*/
    /*+-----+
    |0|1|2|3|4|5|6|7|
    +-+-+-+-+-+-+-+-+
    |F|NRI|Type = 1c|
    +-----+*/
    FUIndicatorDef stFUIndicator;
    /*+-----+
    |0|1|2|3|4|5|6|7|
    +-+-+-+-+-+-+-+-+
    |S|E|R| NalType |
    +-----+*/
    FUHeaderDef stFUHeader;
}PACKED FU_ADef;
```

4.2 SPS PPS 解释（具体参考 H264 标准）

4.2.1 H264 帧

对于 H.264 而言，每帧的界定符为 00 00 00 01 或者 00 00 01。
如下一个 H264 帧的片段：

00 00 00 01 67 42 C0 28 DA 01 E0 08 9F 96 10 00
00 03 00 10 00 00 03 01 48 F1 83 2A 00 00 00 01
68 CE 3C 80 00 00 01 06 05 FF FF 5D DC 45 E9 BD
E6 D9 48 B7 96 2C D8 20 D9 23 EE EF ...

第一帧是：00 00 00 01 67 42 C0 28 DA 01 E0 08 9F 96 10 00 00 03 00 10 00 00 03 01 48 F1 83 2A。
第二帧是：00 00 00 01 68 CE 3C 80。
第三帧是：00 00 01 06 05 FF FF 5D DC 45 E9 BD E6 D9 48 B7 96 2C D8 20 D9 23 EE EF ..

帧类型有：

- NAL_SLICE = 1 非关键帧
- NAL_SLICE_DPA = 2
- NAL_SLICE_DPB = 3

注释：

	文件名称: RTSP 协议总体设计说明 版 本: A. 2016. 0901	页 码: 22 OF 26 机密等级: 保密
发行日期: 2016. 09. 01	生效日期: 2016. 09. 01	文件编号: 0001

NAL_SLICE_DPC =4
NAL_SLICE_IDR =5 关键帧
NAL_SEI = 6
NAL_SPS = 7 SPS 帧
NAL_PPS = 8 PPS 帧
NAL_AUD = 9
NAL_FILLER = 12

4.2.2 SPS（序列参数集 Sequence Parameter Set）

SPS 对于 H264 而言，就是编码后的第一帧，如果是读取的 H264 文件，就是第一个帧界定符和第二个帧界定符之间的数据的长度是 4

4.2.3 PPS（图像参数集 Picture Parameter Set）

PPS 就是编码后的第二帧，如果是读取的 H264 文件，就是第二帧界定符和第三帧界定符中间的数据长度不固定。

4.2.4 IDR（及时解码刷新 Instantaneous Decoding Refresh）

I 和 IDR 帧都是使用帧内预测的。它们都是同一个东西而已,在编码和解码中为了方便，要首个 I 帧和其他 I 帧区别开，所以才把第一个首个 I 帧叫 IDR，这样就方便控制编码和解码流程。IDR 帧的作用是立刻刷新,使错误不致传播,从 IDR 帧开始,重新算一个新的序列开始编码。而 I 帧不具有随机访问的能力，这个功能是由 IDR 承担。IDR 会导致 DPB（DecodedPictureBuffer 参考帧列表——这是关键所在）清空，而 I 不会。IDR 图像一定是 I 图像，但 I 图像不一定是 IDR 图像。一个序列中可以有很多的 I 图像，I 图像之后的图像可以引用 I 图像之间的图像做运动参考。一个序列中可以有很多的 I 图像，I 图像之后的图索可以引用 I 图像之间的图像做运动参考。

对于 IDR 帧来说，在 IDR 帧之后的所有帧都不能引用任何 IDR 帧之前的帧的内容，与此相反，对于普通的 I-帧来说，位于其之后的 B-和 P-帧可以引用位于普通 I-帧之前的 I-帧。从随机存取的视频流中，播放器永远可以从一个 IDR 帧播放，因为在它之后没有任何帧引用之前的帧。但是，不能在一个没有 IDR 帧的视频中从任意点开始播放，因为后面的帧总是会引用前面的帧。

4.3 接口函数设计

4.3.1 Server 启动

```

/*****
/*函数名称: rtsp_server_start                               **/
/*参数:                                           **/
/*    (IN) : pCfg对应RTSP_CONFIG结构体, 启动RTSP服务配置信息**/
/*    (IN) : pPlayCb对应RTSP_PLAY_CB结构体, 注册stream回调 **/
/*返回值: <0 失败 >=0 成功                               **/
/*****/
int rtsp_server_start(RTSP_CONFIG * pCfg, RTSP_PLAY_CB * pPlayCb)
```

4.3.2 Server 停止

```

/*****
/*函数名称: rtsp_srv_stop                               **/
/*参数: 无                                           **/
/*返回值: <0 失败 >=0 成功                               **/
/*****/
int rtsp_srv_stop();
```

注释:

	文件名称: RTSP 协议总体设计说明 版 本: A.2016.0901	页 码: 23 OF 26 机密等级: 保密
发行日期: 2016.09.01	生效日期: 2016.09.01	文件编号:0001

4.3.3 RTSP 会话处理函数接口

```

/*****
/*函数名称: rtsp_session_handle
/*参数:
/* (IN):pSess对应RTSP_SESSION_INFO结构体, 一个会话处理函数**/
/*返回值: <0 失败 >=0 成功
/*****
int rtsp_session_handle(RTSP_SESSION_INFO *pSess);

```

4.3.4 RTSP AV 传输函数接口

```

/*****
/*函数名称: create_rtsp_av_transmit
/*参数:
/* (IN):pSess对应RTSP_SESSION_INFO结构体, 一个会话处理函数**/
/*返回值: <0 失败 >=0 成功
/*****
int create_rtsp_av_transmit(RTSP_SESSION_INFO *pSess);

```

4.3.5 RTSP AV 传输销毁函数接口

```

/*****
/*函数名称: destroy_rtsp_av_transmit
/*参数:
/* (IN):pSess对应RTSP_SESSION_INFO结构体, 一个会话处理函数**/
/*返回值: <0 失败 >=0 成功
/*****
int destroy_rtsp_av_transmit(RTSP_SESSION_INFO *pSess);

```

4.3.6 RTP Packet 函数接口

```

/*****
/*函数名称: rtsp_stream_packet_rtp_header
/*参数:
/* (OUT):pHdr Rtp 报文头部
/* (IN): cc,x,p,version,pt,mark,seqno,pts,ssrc参考RTP协议
/*返回值: <0 失败 >=0 成功
/*****
int rtsp_stream_packet_rtp_header(RTP_HDR_S * pHdr, unsigned short cc, unsigned short x, unsigned short p,
                                unsigned short version, unsigned short pt, unsigned short mark,
                                unsigned short seqno, unsigned long pts, unsigned long ssrc);

```

4.3.7 RTP 发送数据函数接口

```

/*****
/*函数名称: rtsp_send_stream_data
/*参数:
/* (IN):pSess
/* (IN):pBuf数据
/* (IN):len数据长度
/* (IN):time 时间戳
/*返回值: <0 失败 >=0 成功
/*****
int rtsp_send_stream_data(RTSP_SESSION_INFO * pSess, char * pBuf, int len, unsigned int time);

```

注释:

	文件名称： RTSP 协议总体设计说明 版 本：A.2016.0901	页 码：24 OF 26 机密等级： 保密
发行日期：2016.09.01	生效日期：2016.09.01	文件编号：0001

4.3.8 RTCP 守护函数接口

```

/*****
/*函数名称: do_rtcp_handler                               */
/*参数:                                           */
/*  (IN) :pSess                                       */
/*  (IN) :index                                       */
/*返回值: <0 失败 >=0 成功                               */
/*****
int do_rtcp_handler(RTSP_SESSION_INFO * pSess, int index);

```

5. 软件运行环境与开发工具

开发平台 Linux+VLC
工具 source insight
参考源码：Live555 or StreamServer
解码参考：ffmpeg

6. 软件体系的关键技术和难点

- 本套系统开发难点：
- 1.服务器稳定及并发访问量的控制
 - 2.RTP Over Tcp 传输模式下“\$”符号应用
 - 3.RTP Over Multicast 下三层交换机的穿透
 - 4.SDP 描述信息中的 SPS PPS 获取（要进过 base64 编码）
 - 5.RTCP 网络守护（SR 报告，RR 报告）

7. 附录（可选）

对系统的关键技术和难点、特点等需要的一些附加说明资料。

注释：

	文件名称： RTSP 协议总体设计说明 版 本：A. 2016. 0901	页 码： 25 OF 26 机密等级： 保密
发行日期：2016. 09. 01	生效日期：2016. 09. 01	文件编号：0001

8. 相关文件

序号	文件名称	文件编号	编写单位

9. 相关记录

序号	文件名称	文件编号	归档单位
----	------	------	------

附录 A：

修改记录表 1

修改序号		版本	
修改人		修改日期	
修改内容			
修改原因			

修改记录表 2

修改序号		版本	
修改人		修改日期	
修改内容			

注释：

	文件名称： RTSP 协议总体设计说明 版本：A. 2016. 0901	页 码： 26 0F 26 机密等级： 保密
发行日期：2016. 09. 01	生效日期：2016. 09. 01	文件编号：0001

修改 原因	
----------	--

注释：