1. 概念：

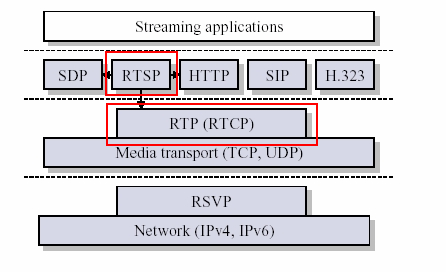
.RTSP：应用层协议，RTSP提供了一个可供扩展的框架，它的意义在于使得实时流媒体数据的受控和点播变得可能。RTSP是一个流媒体表示协议，主要用来控制具有实时特性的数据发送，但它本身并不传输数据，而是依赖下层协议提供的服务。

.RTP：是在Internet上处理多媒体数据流的一种网络协议，利用它可以实现多媒体数据的实时传输。RTP协议负责对流媒体数据进行封包并实现媒体流的实时传输，分为数据包头部和负载。RTP协议的目的在于提供实时数据的传输，一般是在传输协议之上作为应用程序的一部分加以实现。

.RTCP：RTCP协议需要和RTP协议一起配合使用，RTP本身并不能为按序传输数据包提供可靠的保证，也不提供流量控制和拥塞控制，这些都由RTCP负责完成。

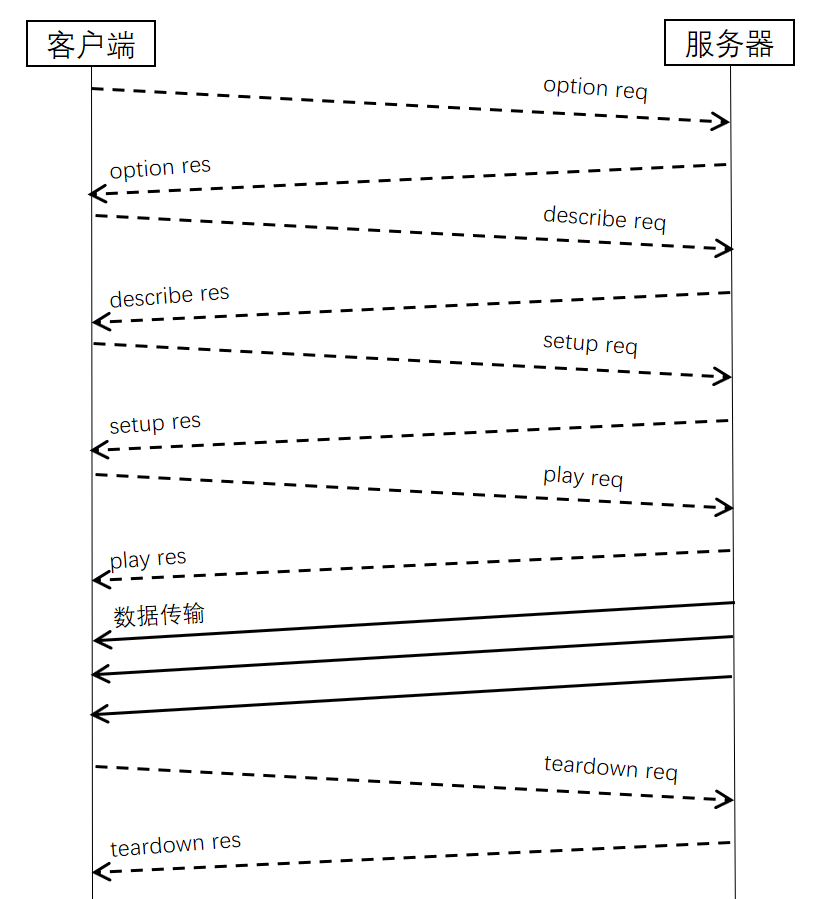
.SDP：会话描述协议，基于文本的协议，在流媒体中只用于描述会话信息，媒体协商由RTSP完成。

网络协议分层情况：



小结：通常情况下，底层具体传输协议采用UDP；为实现多媒体数据的实时传输，RTP协议对需要传输的数据进行封装，加入时间戳等标记，然后作为UDP数据包的数据部分；链路传输质量的反馈由RTCP保证，RTCP负责记录传输的情况，包括丢失的包数量、网络状况等信息；而用户与服务器间的媒体协商则由应用层的RTSP协议保证，即，RTSP被用于控制流媒体的传输，并不用于转发数据，多媒体数据的传输通过RTP/RTCP协议完成；在RTSP建立连接时，需要使用SDP描述媒体信息。

1. 简单的RTSP消息交互流程：
   1. 查询服务器端可用方法
   2. 得到媒体描述信息
   3. 建立RTSP会话
   4. 请求开始传输数据
   5. 关闭会话，退出
2. 流程图：

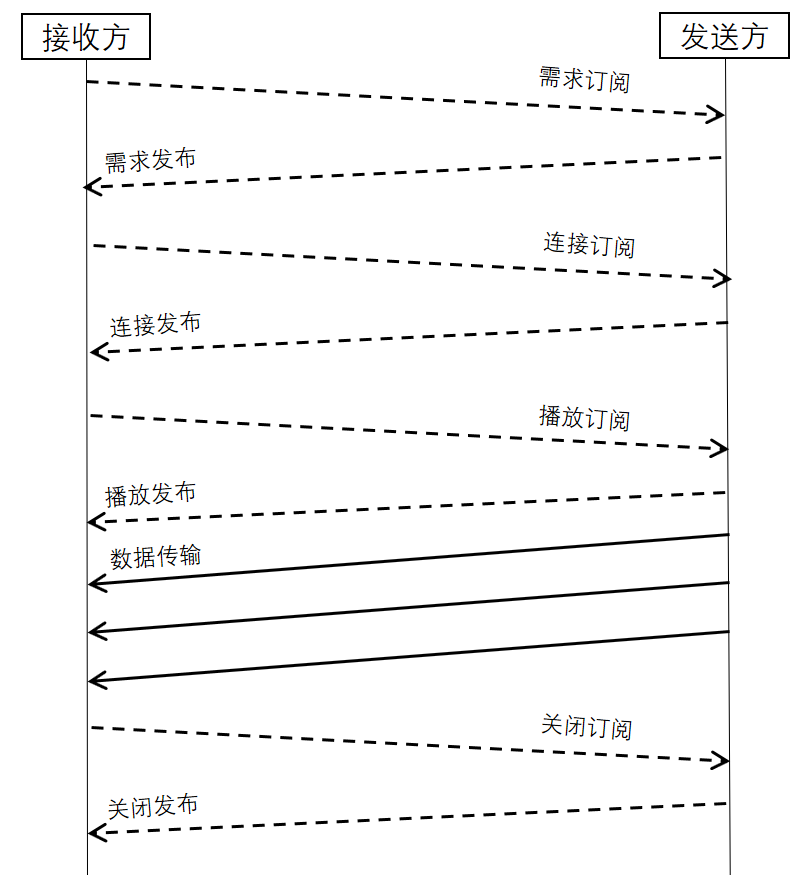


虚线代表RTSP协商 流程，实线表示数据传输，具体每步含义为：

* 1. option req：客户端询问服务器有哪些方法可用
  2. option res：服务器回复，包含所有可用方法
  3. describe req：客户端要求得到服务器提供的媒体描述信息
  4. describe res：服务器回应媒体描述信息，为SDP格式
  5. setup req：请求建立会话，此过程建立RTP通道
  6. setup res：建立会话，并返回建立的session id
  7. play req：请求传输，主要功能是指定传输起止时间、播放速度
  8. play res：回复
  9. 具体数据传输
  10. teardown req：请求关闭会话
  11. teardown res：关闭会话

其中，必不可少的是describe和setup两步，其他协商过程可以通过事先约定等方式省去。

1. 在发布订阅中流程图：



具体流程为：

* 1. 需求订阅：接收方提出订阅需求，希望获得某个视频的实时传输服务
  2. 需求发布：发送方接收订阅并回复，包含视频内容、格式、大小等描述信息
  3. 连接订阅：接收方希望建立连接，提出订阅，包含
  4. 连接发布：发送方回复自身地址
  5. 播放订阅：接收方希望播放视频，包含视频播放的起止时间、速度、码率等信息
  6. 播放发布：发送方回复并准备发送RTP封装的视频信息
  7. 数据传输：发送方开始传输数据
  8. 关闭订阅：接收方希望结束视频传输
  9. 关闭发布：接收方结束传输并回复