

实时流媒体协议 (RTSP)

状态说明

此文档指定了一个用于互联网标准跟踪的协议。
互联网社区，并讨论和提出建议
改进。请参阅“互联网”当前版本。
官方协议标准 (STD 1) 的标准化状态
此协议的状态。本备忘录的分发不受限制。

版权声明

版权 (C) 互联网协会 (1998)。版权所有。

摘要

实时流媒体协议，或称RTSP，是一种应用层协议
协议对实时数据传输的控制
属性。RTSP提供了一个可扩展的框架，以启用
受控、按需交付实时数据，例如音频和
视频。数据来源可以包括实时数据流和存储
clips。此协议旨在控制多个数据传输
会话，提供选择交付通道（如UDP）的途径，
组播UDP和TCP，并提供选择交付的方式
基于RTP (RFC 1889) 的机制。

目录

* 1 简介	5
+ 1.1 目的	5
+ 1.2 需求	6
+ 1.3 术语	6
+ 1.4 协议属性	9
+ 1.5 扩展 RTSP	11
+ 1.6 总体操作	11
+ 1.7 RTSP 状态	12
+ 1.8 与其他协议的关系	13
* 2 符号约定	14
* 3 协议参数	14
+ 3.1 RTSP版本	14

+3.2 RTSP URL	14
+3.3 会议标识	16
+3.4 会话标识	16
+3.5 SMPTE 相对时间戳	16
+3.6 正常播放时间	17
+3.7 绝对时间	18
+3.8 选项标签	18
o 3.8.1 使用 IANA 注册新的选项标签	18
* 4 RTSP 消息	19
+ 4.1 消息类型	19
+ 4.2 消息头	19
+ 4.3 消息体	19
+ 4.4 消息长度	20
* 5 通用报头字段	20
* 6 请求	20
+6.1 请求行	21
+6.2 请求头字段	21
* 7 响应	22
+ 7.1 状态行	22
o 7.1.1 状态码和原因短语	22
o 7.1.2 响应头字段	26
* 8 实体	27
+ 8.1 实体头部字段	27
+ 8.2 实体体	28
* 9 连接	28
+ 9.1 管道化	28
+ 9.2 可靠性和致谢	28
* 10 方法定义	29
+ 10.1 选项	30
+ 10.2 描述	31
+ 10.3 公告	32
+ 10.4 设置	33
+ 10.5 玩	34
+ 10.6 暂停	36
+ 10.7 拆解	37
+ 10.8 获取参数	37
+ 10.9 设置参数	38
+ 10.10 重定向	39
+ 10.11 记录	39
+ 10.12 嵌入式（交织）二进制数据	40
* 11 状态码定义	41
+11.1 成功 2xx	41
o 11.1.1 250 存储空间不足	41
+11.2 重定向 3xx	41
+11.3 客户端错误 4xx	42
o 11.3.1 405 方法不允许	42
o 11.3.2 451 参数未理解	42
o 11.3.3 452 会议未找到	42

o 11.3.4 453 网络带宽不足	42
o 11.3.5 454 会话未找到	42
o 11.3.6 455 在此状态下方法无效	42
o 11.3.7 456 标题字段对资源无效	42
o 11.3.8 457 无效范围	43
o 11.3.9 458 参数为只读	43
o 11.3.10 459 聚合操作不允许	43
o 11.3.11 460 仅允许聚合操作	43
o 11.3.12 461 不支持的传输	43
o 11.3.13 462 目的地不可达	43
o 11.3.14 551 选项不支持	43
* 12 12.1 报告头字段定义	44
+ 12.1 接受	46
+ 12.2 接受编码	46
+ 12.3 接受语言	46
+ 12.4 允许	46
+ 12.5 授权	46
+ 12.6 带宽	46
+ 12.7 块大小	47
+ 12.8 缓存控制	47
+ 12.9 会议	49
+ 12.10 连接	49
+ 12.11 内容基础	49
+ 12.12 内容编码	49
+ 12.13 内容语言	50
+ 12.14 内容长度	50
+ 12.15 内容位置	50
+ 12.16 内容类型	50
+ 12.17 CSeq	50
+ 12.18 日期	50
+ 12.19 过期	50
+ 12.20 从	51
+ 12.21 主机	51
+ 12.22 If-Match	51
+ 12.23 如果修改后	52
+ 12.24 最后修改日期	52
+ 12.25 位置	52
+ 12.26 代理认证	52
+ 12.27 代理-要求	52
+ 12.28 公共	53
+ 12.29 范围	53
+ 12.30 引用者	54
+ 12.31 重试-后	54
+ 12.32 需要项	54
+ 12.33 RTP-Info	55
+ 12.34 比例	56
+ 12.35 速度	57
+ 12.36 服务器	57

+ 12.37 会话	57
+ 12.38 时间戳	58
+ 12.39 运输	58
+ 12.40 不支持	62
+ 12.41 用户代理	62
+ 12.42 变化	62
+ 12.43 通过	62
+ 12.44 WWW-Authenticate	62
* 13 缓存	62
* 14 个示例	63
+14.1 媒体点播（单播）	63
+14.2 容器文件流	65
+14.3 单一流容器文件	67
+14.4 使用多播的实时媒体演示	69
+14.5 将媒体播放到现有会话中	70
+14.6 记录	71
* 15 语法	72
+ 15.1 基础语法	72
* 16 安全注意事项	73
* RTSP协议状态机	76
+ A.1 客户端状态机	76
+ A.2 服务器状态机	77
* B 与 RTP 的交互	79
* C 使用SDP进行RTSP会话描述	80
+ C.1 定义	80
C.1.1 控制URL	80
o C.1.2 媒体流	81
C.1.3 负载类型(们)	81
C.1.4 格式特定参数	81
C.1.5 展示范围	82
C.1.6 可用时间	82
o C.1.7 连接信息	82
o C.1.8 实体标签	82
+ C.2 聚合控制不可用	83
+ C.3 可用聚合控制	83
* D 最小化 RTSP 实现	85
+ D.1 客户	85
o D.1.1 基本播放	86
o D.1.2 认证启用	86
+ D.2 服务器	86
o D.2.1 基本播放	87
o D.2.2 认证启用	87
* 作者地址	88
* F 致谢	89
* 参考文献	90
* 全版权声明	92

1 简介

1.1 目的

实时流媒体协议 (RTSP) 建立和控制
单个或多个时间同步的连续流
媒体, 如音频和视频。它通常不提供
连续流本身, 尽管是连续的交织
媒体流与控制流相结合是可能的 (见第10.12节)。
换句话说, RTSP充当一个 “网络远程控制”
多媒体服务器。

要控制的流集合由一个表示定义
描述。本备忘录未定义一个格式用于一个
演示描述。

没有RTSP连接的概念; 相反, 服务器维护
一个由标识符标记的会话。RTSP会话与任何事物都没有关联
到一个传输层连接, 例如TCP连接。在
RTSP会话, 一个RTSP客户端可能打开和关闭许多可靠的
服务器发出RTSP请求的传输连接。
或者, 它可能使用无连接传输协议, 例如
UDP。

RTSP控制的流可能使用RTP [1], 但其操作
RTSP 不依赖于用于承载的传输机制
连续介质。协议在语法上故意相似
并且操作到HTTP/1.1 [2], 以便扩展机制到HTTP
在大多数情况下也可以添加到RTSP。然而, RTSP在以下方面有所不同:
HTTP的重要方面数量:

- * RTSP引入了许多新方法, 并且有所不同
协议标识符。
- * 默认情况下, 一个RTSP服务器需要维护状态, 几乎在所有情况下都需要
案例, 与HTTP的无状态特性相对。
- * RTSP服务器和客户端都可以发出请求。
数据通过不同的协议进行带外传输。
异常情况。
RTSP被定义为使用ISO 10646 (UTF-8) 而不是ISO 8859-1 ,
与当前HTML国际化努力保持一致[3]。
- * 请求URI始终包含绝对URI。因为
向后兼容一个历史错误, HTTP/1.1 [2]
仅携带请求中的绝对路径并放置主机
名称在单独的标题字段中。

这使得 “虚拟主机” 更容易, 其中单个主机有一个
IP地址托管多个文档树。

协议支持以下操作：

媒体从媒体服务器检索：

客户端可以通过HTTP请求演示描述。
一些其他方法。如果演示正在多播，则
演示描述包含多播地址和
端口号用于连续媒体。如果演示
仅通过单播发送到客户端，客户端
提供安全原因的目的地。

媒体服务器会议邀请：

一个媒体服务器可以被“邀请”加入现有的
会议，无论是将媒体回放演示中还是
记录演示文稿中的所有或部分媒体。此
模式对于分布式教学应用很有用。几个
会议中的各方可以轮流“推动遥控器”
控制按钮。”

向现有演示文稿添加媒体：

特别对于现场演示，如果{v*}有用，那就很有帮助。
服务器可以告诉客户端关于额外媒体开始
可用。

实时流传输请求可能由代理、隧道和缓存处理，如下所示
HTTP/1.1 [2].

1.2 需求

"必须"、"禁止必须"、"必需"、"应当"、"禁止应当"

"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", 和 "OPTIONAL" 在此 {v*}
文档应按照RFC 2119 [4]中所述进行解释。

1.3 术语

一些术语已从HTTP/1.1 [2] 中采用。未翻译的术语
此处定义与HTTP/1.1中相同。

聚合控制：

The control of the multiple streams using a single timeline by
服务器。对于音频/视频流，这意味着客户端
可能发送单个播放或暂停消息来控制两者
音频和视频流。

会议：

一个多方、多媒体演示，其中“多”意味着
大于或等于一。

客户端：

客户端请求从媒体连续获取媒体数据服务器。

连接：

在两个之间建立的传输层虚拟电路程序用于通信目的。

容器文件：

一个可能包含多个媒体流的文件，这些流通常包含一个在同时播放时的演示。RTSP服务器可能提供对这些文件的聚合控制，尽管这个概念容器文件未嵌入到协议中。

连续介质：

数据中存在源和之间的时序关系
汇入点；也就是说，汇入点必须再现时序关系在源处存在的。最常见的例子包括连续媒体是音频和动态视频。连续媒体可以是实时（交互式）的，其中存在一个“紧密”源与汇之间的时序关系，或流（回放），其中关系较为宽松。

实体：

请求的有效负载中传输的信息或响应。一个实体由以下形式的元信息组成
实体头部字段和以实体形式的内容正文，如第8节所述。

媒体初始化：

数据类型/编解码器特定初始化。这包括如下事物如时钟速率、颜色表等。任何传输-独立信息，客户所需
媒体流的播放发生在媒体初始化过程中流设置阶段。

媒体参数：

参数特定于媒体类型，可能在之前被更改或在进行流媒体播放时。

媒体服务器：

服务器为一个或多个{v*}提供播放或录制服务更多媒体流。一个媒体流内的不同媒体流演示可能来自不同的媒体服务器。A
媒体服务器可能位于与以下相同或不同的主机上：网页服务器从其调用的表示。

媒体服务器间接引用：

媒体客户端重定向到不同的媒体服务器。

(媒体) 流：

单个媒体实例，例如，音频流或视频流以及单个白板或共享应用分组。当使用RTP时，一个流由所有RTP和RTCP数据包，由RTP会话中的源创建。这是等同于DSM-CC流的定义([5])。

消息：

RTSP通信的基本单元，由一个结构化字节序列，符合在以下定义的语法中匹配的语法第15节并通过连接或传输无连接协议。

参与者：

会议成员。参与者可能是一个机器，例如，媒体记录或播放服务器。

演示： {v*}

一组一个或多个流，作为流给客户端展示完整媒体流，使用一个演示描述作为定义如下。在大多数RTSP上下文中，这通常意味着对那些流进行汇总控制，但不必。

演示描述：

一个演示描述包含有关一个或多个{v*}的信息。更多演示文稿中的媒体流，例如集合编码，网络地址以及关于{v*}的信息内容。其他IETF协议，如SDP（RFC 2327 [6]）使用“现场演示的“会话”术语。演示描述可能采用多种不同的格式，包括但不限于不限于会话描述格式SDP。

响应：

一个RTSP响应。如果是指HTTP响应，则是明确指出。

请求：

一个RTSP请求。如果是指HTTP请求，则是明确指出。

实时流媒体协议会话：

一个完整的RTSP“事务”，例如，观看一部电影。一次会话通常包括一个客户端设置一个运输机制用于连续媒体流（SETUP），开始流使用PLAY或RECORD，并关闭

流式传输带有TEARDOWN。

运输初始化：

运输信息谈判（例如，港口编号，{v*}）

传输协议（客户端与服务器之间）。

1.4 协议属性

RTSP具有以下属性：

可扩展：

新方法和参数可以轻松添加到RTSP中。

易于解析：

RTSP 可以被标准的 HTTP 或 MIME 解析器解析。

安全：

RTSP 重新使用网络安全机制。所有 HTTP 认证机制，如基本（RFC 2068 [2, 第11.1节]）和摘要认证（RFC 2069 [8]）直接适用。一个人也可以重用传输层或网络层安全机制。

与运输无关：

RTSP可能使用不可靠的数据报协议（UDP）（RFC 768 [9]，一个可靠的报文协议（RDP，RFC 1151，不广泛使用[10]或可靠的流协议，如TCP（RFC 793 [11]）如其实现应用层可靠性。

多服务器支持：

每个演示文稿中的媒体流都可以驻留在不同的服务器。客户端自动建立几个并发控制会话与不同的媒体服务器。媒体同步在传输层进行。

控制录音设备：

协议可以控制录音和播放设备，以及可以在两种模式之间切换的设备（"VCR"）。

流控制与会议发起的分离：

流控制与邀请媒体服务器到

会议。唯一的要求是会议

启动协议提供或可用于创建一个

独特的会议标识符。特别是，SIP [12] 或 H.323

[13] 可用于邀请服务器加入会议。

适用于专业应用：

RTSP通过SMPTE时间戳支持帧级精度
允许远程数字编辑。

演示描述中性：

协议不强制特定的表示
描述或元文件格式，并能传达类型
格式应使用。然而，展示描述必须
包含至少一个RTSP URI。

代理和防火墙友好：

协议应易于被应用程序和
传输层（SOCKS [14]）防火墙。防火墙可能需要
理解SETUP方法以为UDP媒体打开一个“洞”
流

HTTP友好型：

在合理的情况下，RTSP重用HTTP概念，以便
现有基础设施可以重用。此基础设施
包含PICS（互联网内容选择平台）
[15,16]用于将标签与内容关联。然而，RTSP
不只是在HTTP中添加方法，还控制
大多数情况下，连续媒体需要服务器状态。

适当的服务器控制：

如果客户端可以启动流，则它必须能够停止流
流。服务器不应在这种情况下向客户端开始流式传输
一种客户无法停止流的方式。

运输谈判：

客户端可以在之前协商运输方式
实际上需要处理一个连续的媒体流。

能力协商：

如果基本功能被禁用，则必须有一些干净的
机制以确定哪些方法客户端不使用
将要实现。这允许客户端展示
合适的用户界面。例如，如果正在寻找不是
允许的，用户界面必须能够禁止移动一个
滑动位置指示器。

早期RTSP的要求是多客户端功能。

然而，确定更好的方法是确保

该协议易于扩展到多客户端

场景。流标识符可以被多个控制使用

流，以便“传递远程”成为可能。

协议不会解决多个客户端如何协商访问的问题；

这是留给“社交礼仪”或其他某种地面的

控制机制。

1.5 扩展 RTSP

由于并非所有媒体服务器都具有相同的功能，媒体服务器不可避免地需要支持不同的请求集。对于示例： Translated Text： 示例：

- * 服务器可能只能播放，因此没有必要支持 RECORD 请求。
- * 服务器可能无法进行查找（绝对定位），如果仅支持现场活动。
- * 一些服务器可能不支持设置流参数，因此不支持 GET_PARAMETER 和 SET_PARAMETER。

服务器应实现第12节中描述的所有报头字段。

它取决于演示描述的创作者不要询问服务器不可能。这种情况在HTTP/1.1 [2]中类似。在[H19.6]中描述的方法不太可能得到支持跨所有服务器。

RTSP可以扩展为三种方式，按顺序列于此。变化支持的范围：

- * 现有方法可以通过添加新参数进行扩展，只要这些参数可以安全地被接收者忽略。（这是相当于向HTML标签添加新参数。）如果客户端需要在方法扩展时需要负确认不支持，可能添加一个对应扩展名的标签在“要求”字段中（见第12.32节）。
- * 新方法可以被添加。如果消息的接收者执行不理解请求，以错误代码501（未实现）响应已实现）并且发送者不应尝试使用此方法再次。客户端还可以使用 OPTIONS 方法来查询服务器支持的方法。服务器应列出支持使用公共响应头的方法。
- * 可以定义协议的新版本，允许几乎所有方面（除了协议版本号的位路）到更改。

1.6 总体操作

每个演示和媒体流都可以通过RTSP URL进行标识。整体展示和媒体的属性展示由展示描述定义文件，其格式超出了本规范的范畴。

The presentation description file may be obtained by the client using 演示描述文件可以通过客户端获取

HTTP 或其他方式，如电子邮件，并且不一定被存储在媒体服务器上。

关于本规范的用途，一个演示描述是假定描述一个或多个演示，每个演示维护一个共同的时间轴。为了阐述的简便，无损失一般性，假设该表示描述恰好包含一个这样的展示。展示可能包含多个媒体流。

演示描述文件包含对媒体的描述流组成演示，包括它们的编码，语言和其他参数，使客户端能够选择最合适的媒体组合。在本演示中描述，每个可独立控制的媒体流RTSP通过RTSP URL进行识别，该URL指向媒体服务器处理该特定媒体流并将其存储的流命名那个服务器。多个媒体流可以位于不同的服务器；例如，音频和视频流可以跨服务器用于负载均衡。描述还列举了哪些传输方法 服务器能够使用的。

除了媒体参数外，网络目标地址和端口需要确定。可以有多种工作模式。杰出的：

单播：

媒体传输到RTSP请求的源地址，使用客户端选择的端口号。或者，使用媒体与RTSP在同一条可靠的流上传输。

多播，服务器选择地址：

媒体服务器选择多播地址和端口。这是典型情况为实时或接近按需媒体传输

多播，客户端选择地址：

如果服务器要参与现有的多播会议，多播地址、端口和加密密钥是由会议描述给出，通过以下方式建立此规范范围之外。

1.7 RTSP 状态

RTSP控制一个可能通过单独协议发送的流，与控制信道无关。例如，RTSP控制可能在TCP连接上发生，同时数据通过UDP流动。因此，数据交付继续，即使媒体未收到任何RTSP请求

服务器。此外，在其生命周期内，单个媒体流可能由不同TCP上依次发出的RTSP请求控制连接。因此，服务器需要维护“会话状态”能够将RTSP请求与流相关联。状态转换在A节中描述。

许多RTSP中的方法不贡献于状态。然而，以下在定义分配和使用中扮演着核心角色服务器上的流资源：SETUP、PLAY、RECORD、PAUSE以及拆解。

设置：

导致服务器为流分配资源并启动一个RTSP会话。

播放和录制：

在通过SETUP分配的流上开始数据传输。

PAUSE:

暂时停止流而不释放服务器资源。

拆解：

释放与流相关的资源。RTSP会话服务器上不再存在。

RTSP方法中，有助于状态使用的采用Session头部字段（第12.37节）以识别RTSP会话的状态正在被操纵。服务器生成会话标识符针对SETUP请求（第10.4节）。

1.8 与其他协议的关系

RTSP与HTTP在功能上存在一些重叠。它也可能与HTTP交互，即与流媒体内容的初始接触通常通过网页进行。当前协议规范旨在允许在网页之间不同的交接点服务器和实现RTSP的媒体服务器。例如，演示描述可以通过HTTP或RTSP获取，其中减少基于网页浏览器的场景中的往返次数，同时允许对于独立RTSP服务器和客户端，它们不依赖于HTTP所有。

然而，RTSP与HTTP在数据传输方面存在根本区别在另一个协议中离线发生。HTTP是非对称协议，其中客户端发起请求，服务器响应。在RTSP中，媒体客户端和媒体服务器都可以发出请求。RTSP请求也不是无状态的；它们可能设置参数并继续控制媒体流，即使在该之后很长时间

请求已被确认。

在至少两个领域重用HTTP功能具有优势，
即安全性和代理。要求非常相似，所以
具有在缓存、代理上采用HTTP工作的能力，
身份验证很有价值。

虽然大多数实时媒体将使用RTP作为传输协议，RTSP
未绑定到RTP。

RTSP 假设存在一种表示描述格式，
可以表示演示的静态和时态属性
包含多个媒体流。

2 符号约定

由于许多定义和语法与HTTP/1.1相同，
本规范仅指向它们定义的章节
而不是复制它。为了简洁，[HX.Y]应被视为指代
到当前HTTP/1.1规范（RFC 2068 [2]）的第X.Y节。

所有本文件中指定的机制均在其中以中英文描述。
散文和类似于在中所使用的增强型巴科斯-诺尔范式（BNF）
[H2.1]。它在RFC 2234 [17]中详细描述，其中包含公式 {v*}。
差异在于本RTSP规范维持了“1#”符号
对于逗号分隔的列表。

在这个备忘录中，我们使用缩进和较小字号的段落来提供
背景和动机。这是为了给那些
未参与制定规范的过程
对RTSP中事物为何如此存在的理解。

3 协议参数

3.1 RTSP版本

[H3.1] 适用，将 HTTP 替换为 RTSP。

3.2 RTSP URL

"rtsp" 和 "rtspu" 方案用于引用网络资源
通过RTSP协议。本节定义了特定方案的
RTSP URL 的语法和语义。

```
rtsp_URL = ( "rtsp:" | "rtspu:" )  
// 主机 [ ":" 端口 ] [ 绝对路径 ]  
主机      = <IP地址的合法互联网主机域名  
(以点分十进制形式表示)，如第2.1节所述
```

关于RFC 1123 \cite{rfc1123}>
端口 = *数字

绝对路径在[H3.2.1]中定义。

请注意，片段标识符和查询标识符没有明确定义
当前的意义，解释留给RTSP
服务器。

该方案rtsp要求通过可靠的途径发布命令
协议（在互联网中，TCP），而rtspu方案标识
一个不可靠的协议（在互联网中，UDP）。

如果端口为空或未指定，则假定端口为554。语义
这些资源可以通过RTSP进行控制
服务器监听TCP（方案 "rtsp"）连接或UDP（方案
"rtspu"）数据包在该主机端口的，以及请求URI为
资源是 rtsp_URL。

IP地址在URL中的使用应尽可能避免
(参见RFC 1924 [19]。)

一个演示或流由一个文本媒体标识
标识符，使用字符集和转义约定 [H3.2]
URLs (RFC 1738 [20])。URLs可能指代一个流或一个聚合
流，即，一个演示。相应地，描述在
第10节可以应用于整个演示文稿或单个
流在演示中。注意，某些请求方法可以
仅适用于流，不适用于演示文稿，反之亦然。

例如，RTSP URL：
rtsp://media.example.com:554/twister/audiotrack

识别演示文稿 "twister" 中的音频流，其中
可以通过通过TCP连接发出的RTSP请求进行控制
端口554的主机media.example.com。

同样，RTSP URL：
rtsp://media.example.com:554/twister

识别 "twister" 这种表示形式，它可能由以下组成
音频和视频流。

这并不暗示在URL中引用流的标准化方式。
演示描述定义了层次关系
在演示文稿和各个流的URL中。A
演示描述可能将流命名为 "a.mov" 以及整个
演示 "b.mov"。

RTSP URL 的路径组件对客户端来说是透明的，并且不向客户端公开。不表示服务器具有任何特定的文件系统结构。

此解耦还允许使用展示描述
通过简单地替换非RTSP媒体控制协议即可
URL中的方案。

3.3 会议标识符

会议标识对RTSP是透明的，并使用{v*}进行编码。
标准URI编码方法（即，空白字符LWS用%转义）。它们可以
包含任何八位字节值。会议标识符必须全局
独特。对于H.323，应使用conferenceID值。

会议ID = 1*xchar

会议标识用于允许RTSP会话获取
多媒体会议的参数，媒体服务器是
参与。这些会议由协议创建
此规范范围之外，例如，H.323 [13] 或 SIP
[12]。而不是RTSP客户端明确提供传输
信息，例如，它要求媒体服务器使用
值在会议描述中。

3.4 会话标识符

会话标识符是不透明的任意长度字符串。线性
空格必须进行URL编码。会话标识符必须选择
随机且**必须**至少八字节长，以防止猜测
更难。（见第16节。）

会话ID = 1*(字母 | 数字 | 安全)

3.5 SMPTE 相对时间戳

SMPTE 相对时间戳表示相对于开始时间的时序
the clip. 相对时间戳以SMPTE时间码表示
帧级访问精度。时间码的格式为
小时:分钟:秒:帧.subframes，以起始点为原点
的片段。默认SMPTE格式是“ SMPTE 30 drop ”格式，其中
帧率为每秒29.97帧。其他SMPTE代码可能为
支持（例如“ SMPTE 25 ”）通过使用替代用途
"smpte时间"。对于时间值中的"帧"字段可以假设
0到29的值。30与29.97帧之间的差值
每秒通过丢弃前两个帧索引（值）处理
每分钟的00和01，除了每第十分钟。如果帧
值为零时，它可以省略。子帧以{v*}为单位进行测量。
一帧的百分之一。


```

smpte-range = smpte-type "=" smpte-time "-" [ smpte-time ]
smpte类型 = "smpte" | "smpte-30-drop" | "smpte-25"
; 其他时间码可能被添加
smpte-time = 1*2位数字 ":" 1*2位数字 ":" 1*2位数字 [ ":" 1*2位数字 ]
[ "." 1*2位数字 ]

```

示例：

```

smpte=10:12:33:20-
smpte=10:07:33-
smpte=10:07:00-10:07:33:05.01
smpte-25=10:07:00-10:07:33:05.01

```

3.6 正常播放时间

正常播放时间（NPT）表示流的绝对位置
of the presentation. 时间戳由
一个十进制分数。小数点左边的部分可以表示为
以秒或小时、分钟和秒为单位。右侧的部分
小数点衡量秒的分数。

演示的开始对应于0.0秒。负
值未定义。特殊常量现在定义为
当前直播事件的瞬间。仅适用于直播事件。

NPT在DSM-CC中被定义为：“直观上，NPT是时钟，{v*}保持不变。”
观看者与一个程序关联。它通常以数字方式显示在
一个VCR。NPT在正常播放模式下正常前进（比例 = 1），
当进行快速正向扫描时（高正）速度更快
缩放比例），在扫描反向时递减（高负缩放
比例）并且在暂停模式下是固定的。NPT（逻辑上）等同于
SMPTE时间码。[5]

```

npt-range = ( npt-time "-" [ npt-time ] ) | ( "-" npt-time )
npt-time = "now" | npt-sec | npt-hhmmss
npt-sec = 1*数字 [ "." *数字 ]
npt-hhmmss = npt-hh ":" npt-mm ":" npt-ss [ "." *数字 ]
npt-hh = 1*数字 ; 任何正数
npt-mm = 1*2位数字 ; 0-59
npt-ss = 1*2位数字 ; 0-59

```

示例：

```

npt=123.45-125
npt=12:05:35.3-
npt=现在-

```

语法符合ISO 8601。npt-sec表示法已优化
用于自动生成，用于消费的 ntp-hhmmss 表示法
由人类读者。常量 "now" 允许客户端请求

接收实时流而不是存储的或延时流版本。这是必需的，因为既不是绝对时间也不是零时间适用于此情况。

3.7 绝对时间

绝对时间以ISO 8601时间戳表示，使用UTC（GMT）。秒分之一可能表示。

```
utc-range  = "时钟" "=" utc-time "-" [ utc-time ]
utc-time   = utc-date "T" utc-time "Z"
utc-date   = 8位数字      ; < 年年年年月月日日 >
utc-time   = 6位数字 [ "." 小数部分 ] ; < HHMMSS.小数部分 >
```

示例：1996年11月8日14时37分20又四分之一秒
UTC: 翻译文本：

1996年11月08日T14时37分20.25秒Z

3.8 选项标签

选项标签是用于指定新选项的唯一标识符，在 {v*} 中保持不变。RTSP. 这些标签用于 Require（第 12.32 节）和 Proxy-需要（第12.27节）标题字段。

语法：

```
选项标签 = 1*xchar
```

新RTSP选项的创建者应将选项前缀为反向域名（例如，“com.foo.mynewfeature”是一个恰当的名称）对于可以联系到发明者的功能（发明者可通过“foo.com”联系），或注册新选项至互联网数字分配机构（IANA）。

3.8.1 使用 IAN 注册新的选项标签

A 当注册新的RTSP选项时，以下信息应提供如下：

- * 选项的名称和描述。名称可以是任意长度，但是长度不应超过二十个字符。名称必须不包含任何空格、控制字符或句点。
- * 选项变更控制权的指示者（对于示例，IETF，ISO，ITU-T，其他国际标准化实体，一个财团或特定的公司或团体公司）；

- * 如有，进一步描述的引用，例如
（按优先级排序）一个RFC、一篇已发表的论文、一项专利
提交，一份技术报告，文档化的源代码或计算机
手动；
- * 对于专有选项，请联系信息（邮政和电子邮件）
地址）；

4 RTSP 消息

RTSP 是一种基于文本的协议，并使用 ISO 10646 字符集。
UTF-8 编码（RFC 2279 [21]）。行以 CRLF 结尾，但
接收器应准备通过 {v*} 解释 CR 和 LF。
它们作为行终止符。

基于文本的协议使得在 {v*} 中添加可选参数变得更加容易。
自我描述的方式。由于参数的数量和
命令频率低，处理效率不是
关注。基于文本的协议，如果操作得当，也允许轻松
实现研究原型在脚本语言中的实现
Tcl、Visual Basic 和 Perl。

10646字符集避免了棘手的字符集切换，但
只要使用US-ASCII，对应用程序就是不可见的。
这是RTCP使用的编码。ISO 8859-1将 {v*}翻译为
直接转换为Unicode，高八位字节为零。ISO 8859-1
具有最高有效位设置的字符表示为
1100001x 10xxxxxx. (参见RFC 2279 [21])

RTSP消息可以通过任何底层传输协议进行传输
这是8位纯净。

请求包含方法、方法操作的对象以及
参数以进一步描述该方法。方法是幂等的，
除非另有说明。方法也设计为需要很少
或媒体服务器上没有状态维护。

4.1 消息类型 见 [H4.1] 4.

2 消息头 见 [H4.2] 4.3 消

息体 见 [H4.3]

4.4 消息长度

当一条消息包含消息体时，该消息体的长度为
身体由以下之一确定（按优先级顺序）：

1. 任何响应消息不得包含消息体
(例如1xx、204和304响应)始终终止
由标题字段之后的第一个空行分隔，无论其内容如何
消息中存在的实体头部字段。（注意：An
空行仅由CRLF组成。）

2. 如果存在内容长度标头字段（第12.14节），
其字节数表示消息体的长度。
如果此报头字段不存在，则其值为零
假设。

3. 通过服务器关闭连接。（关闭连接）
无法用于表示请求体的结束，因为
那将不会给服务器发送回一个的可能性
响应。）

请注意，RTSP目前不支持HTTP/1.1的“分块”
传输编码（见[H3.6]）并需要存在{v*}。
内容长度报头字段。

考虑到返回的演示描述的适度长度，
服务器应该始终能够确定其长度，即使
它动态生成，使得分块传输编码
不必要的。即使有内容长度必须存在
任何实体体，规则确保即使在没有的情况下也有合理的行为
长度未明确给出。

5 通用报头字段

查看[H4.5]，除了Pragma、Transfer-Encoding和Upgrade头信息
未定义：

通用报头	=	Cache-Control	；第12.8节		Connection	；第12.10节
	Date	；第12.18节		Via	；第12.43节	

6 请求

客户端向服务器或反之亦然发送的请求消息包括，
在第一条消息中，要应用的方法
资源、资源标识符和协议
正在使用的版本。

```
Request      =      Request-Line          ; Section 6.1
                *(
                |      general-header       ; Section 5
                |      request-header       ; Section 6.2
                |      entity-header )      ; Section 8.1
                CRLF
                [ message-body ]          ; Section 4.3
```

6.1 请求行

请求行 = 方法 SP 请求URI SP RTSP版本 CRLF

```
方法      =      "描述"          ;第10.2节      |      "公告"          ;第10.3节
|      "获取参数"      ;第10.8节      |      "选项"          ;第10.1节      |
"暂停"      ;第10.6节      |      "播放"          ;第10.5节      |      "录
制"      ;第10.11节      |      "重定向"      ;第10.10节      |      "设置"
;第10.4节      |      "设置参数"      ;第10.9节      |      "拆卸"      ;第10.
7节      |      扩展方法
```

扩展方法 = 令牌

请求URI = "*" | 绝对URI

RTSP版本 = "RTSP" "/" 1*数字 "." 1*数字

6.2 请求头字段

```
request-header  =      Accept              ; Section 12.1
                    |      Accept-Encoding   ; Section 12.2
                    |      Accept-Language   ; Section 12.3
                    |      Authorization    ; Section 12.5
                    |      From              ; Section 12.20
                    |      If-Modified-Since ; Section 12.23
                    |      Range            ; Section 12.29
                    |      Referer          ; Section 12.30
                    |      User-Agent        ; Section 12.41
```

请注意，与HTTP/1.1 [2] 相比，RTSP请求始终包含
绝对URL（即包括方案、主机和端口）
而不是仅仅绝对路径。

HTTP/1.1 要求服务器理解绝对URL，但客户端应使用Host请求头。这完全是需要与 HTTP/1.0 服务器向后兼容，一个考虑不适用于RTSP的情况。

请求-URI中的星号“*”表示该请求不应用于特定资源，但针对服务器本身，并且仅允许当使用的方法不一定适用于一个资源。一个例子将是：

选项 * RTSP/1.0

7 响应

[H6]适用，但HTTP-Version被RTSP-Version替换。同样，RTSP定义了额外的状态码，并未定义一些HTTP 状态码。有效的响应码及其可用方法在表1中定义了{v*}。

在接收并解释请求消息后，接收者响应一个RTSP响应消息。

```
Response      =      Status-Line          ; Section 7.1
                  *(
                    |      general-header    ; Section 5
                    |      response-header   ; Section 7.1.2
                    |      entity-header     ; Section 8.1
                    |      CRLF
                    [ message-body ]        ; Section 4.3
```

7.1 状态行

响应消息的第一行是状态行，由状态行组成协议版本后跟一个数字状态码，和与状态码相关的文本短语，每个元素由SP字符分隔。不允许在以下情况下使用CR或LF：最终CRLF序列。

状态行 = RTSP版本 SP 状态码 SP 原因短语 CRLF

7.1.1 状态码和原因短语

状态码元素是3位整数结果码。尝试理解和满足请求。这些代码是完全定义在第11节中。Reason-Phrase的目的是提供一个简短的文本描述状态码。状态码旨在用于自动机，Reason-Phrase 旨在供人类使用用户。客户端无需检查或显示原因-短语。

状态码的第一个数字定义了响应的类别。
最后两位数字没有任何分类作用。有5个
值对于第一个数字：

- * 1xx：信息性 - 请求已接收，继续处理
- * 2xx：成功 - 动作已成功接收、理解，
并且接受
- * 3xx：重定向 - 为了继续，必须采取进一步的操作
完成请求
- * 4xx：客户端错误 - 请求包含语法错误或无法
已履行
- * 5xx：服务器错误 - 服务器未能完成显然
有效请求

The individual values of the numeric status codes defined for 个体数值的数字状态代码定义如下
RTSP/1.0，以及相应的Reason-Phrase的示例集，是
以下列出。此处列出的原因短语仅作参考
- 它们可以被本地等效物替换，而不会影响
协议。请注意，RTSP采用了大多数HTTP/1.1 [2] 状态码和
添加从x50开始的RTSP特定状态码以避免冲突
使用新定义的HTTP状态码。

状态码 =

"100"	; Continue
"200"	; OK
"201"	; Created
"250"	; Low on Storage Space
"300"	; Multiple Choices
"301"	; Moved Permanently
"302"	; Moved Temporarily
"303"	; See Other
"304"	; Not Modified
"305"	; Use Proxy
"400"	; Bad Request
"401"	; Unauthorized
"402"	; Payment Required
"403"	; Forbidden
"404"	; Not Found
"405"	; Method Not Allowed
"406"	; Not Acceptable
"407"	; Proxy Authentication Required
"408"	; Request Time-out
"410"	; Gone
"411"	; Length Required
"412"	; Precondition Failed
"413"	; Request Entity Too Large
"414"	; Request-URI Too Large
"415"	; Unsupported Media Type
"451"	; Parameter Not Understood
"452"	; Conference Not Found
"453"	; Not Enough Bandwidth
"454"	; Session Not Found
"455"	; Method Not Valid in This State
"456"	; Header Field Not Valid for Resource
"457"	; Invalid Range
"458"	; Parameter Is Read-Only
"459"	; Aggregate operation not allowed
"460"	; Only aggregate operation allowed
"461"	; Unsupported transport
"462"	; Destination unreachable
"500"	; Internal Server Error
"501"	; Not Implemented
"502"	; Bad Gateway
"503"	; Service Unavailable
"504"	; Gateway Time-out
"505"	; RTSP Version not supported
"551"	; Option not supported

扩展代码

extension-code = 3DIGIT

Reason-Phrase = *<TEXT, excluding CR, LF>

实时流媒体协议 (RTSP) 状态码是可扩展的。RTSP应用程序不要求了解所有已注册状态码的含义, 尽管如此理解显然是可取的。然而, 应用必须理解任何状态码的类别, 如第一个指示的数字, 并将任何未识别的响应视为等同于x00 该类状态码, 除外的{v*} 未识别的响应不得缓存。例如, 如果一个客户端收到无法识别的状态码431 安全地假设其请求中存在某些问题, 并且将响应视为已收到400状态码。在这种情况下案例, 用户代理应向用户展示返回的实体与响应一起, 因为该实体可能包含人类- 可读信息, 将解释不寻常的状态。

Code	reason	
100	Continue	all
200	OK	all
201	Created	RECORD
250	Low on Storage Space	RECORD
300	Multiple Choices	all
301	Moved Permanently	all
302	Moved Temporarily	all
303	See Other	all
305	Use Proxy	all

400	Bad Request	all
401	Unauthorized	all
402	Payment Required	all
403	Forbidden	all
404	Not Found	all
405	Method Not Allowed	all
406	Not Acceptable	all
407	Proxy Authentication Required	all
408	Request Timeout	all
410	Gone	all
411	Length Required	all
412	Precondition Failed	DESCRIBE, SETUP
413	Request Entity Too Large	all
414	Request-URI Too Long	all
415	Unsupported Media Type	all
451	Invalid parameter	SETUP
452	Illegal Conference Identifier	SETUP
453	Not Enough Bandwidth	SETUP
454	Session Not Found	all
455	Method Not Valid In This State	all
456	Header Field Not Valid	all
457	Invalid Range	PLAY
458	Parameter Is Read-Only	SET_PARAMETER
459	Aggregate Operation Not Allowed	all
460	Only Aggregate Operation Allowed	all
461	Unsupported Transport	all
462	Destination Unreachable	all
500	Internal Server Error	all
501	Not Implemented	all
502	Bad Gateway	all
503	Service Unavailable	all
504	Gateway Timeout	all
505	RTSP Version Not Supported	all
551	Option not support	all

表 1 : RTSP 方法中的状态码及其使用

7.1.2 响应头字段

响应头字段允许请求接收者传递关于响应的附加信息，无法放入状态行。这些头部字段提供了关于的信息。服务器以及关于通过以下标识的资源进一步访问的信息请求URI。

```
response-header =      Location           ; Section 12.25
                    |      Proxy-Authenticate ; Section 12.26
                    |      Public           ; Section 12.28
                    |      Retry-After      ; Section 12.31
                    |      Server           ; Section 12.36
                    |      Vary             ; Section 12.42
                    |      WWW-Authenticate ; Section 12.44
```

响应头字段名只能在以下情况下可靠地扩展：
组合协议版本变更。然而，新的或
实验性头部字段可以赋予响应-语义
表头字段 如果通信中的所有各方都认可它们
响应头字段。无法识别的头字段被视为
实体头部字段。

8 实体

请求和响应消息可能在不另外说明的情况下传输实体
受请求方法或响应状态码限制。一个实体
由实体头部字段和实体体组成，尽管某些
仅包括实体头部的响应。

在这个部分，发送者和接收者均指代客户端
或服务器，取决于谁发送和谁接收实体。

8.1 实体头部字段

实体头部字段定义了关于{v*}的可选元信息。
实体主体或，如果没有主体存在，关于标识的资源
根据要求。

```
entity-header =      Allow                ; Section 12.4
                    |      Content-Base    ; Section 12.11
                    |      Content-Encoding ; Section 12.12
                    |      Content-Language ; Section 12.13
                    |      Content-Length   ; Section 12.14
                    |      Content-Location ; Section 12.15
                    |      Content-Type     ; Section 12.16
                    |      Expires          ; Section 12.19
                    |      Last-Modified   ; Section 12.24
                    |      extension-header
extension-header = message-header
```

扩展头机制允许额外的实体头字段
待定义，但不得更改协议，但这些字段不能
被假定可被收件人识别。无法识别的标题
字段应由接收者忽略并由代理转发。

8.2 实体主体

参见 [H7.2]

9 连接

RTSP请求可以通过几种不同的方式传输：

- * 持久性传输连接用于多个请求-响应事务；
- 每个请求/响应事务一个连接；
- 无连接模式。

运输连接的类型由RTSP URI定义（第{v*}节）
3.2). 对于 "rtsp" 方案，假设存在持久连接，
当方案 "rtspu" 要求发送 RTSP 请求时无需
设置连接。

与HTTP不同，RTSP允许媒体服务器向{v*}发送请求。
媒体客户端。然而，这仅支持持久
连接，因为媒体服务器否则没有可靠的方式来
达到客户端。此外，这也是请求的唯一方式
媒体服务器到客户端可能需要穿越防火墙。

9.1 管道化

支持持久连接或无连接模式的客户端
MAY "管道" 其请求（即，发送多个请求而不
等待每个响应）。服务器必须将其响应发送给那些
请求的顺序与接收到的请求顺序相同。

9.2 可靠性与致谢

请求除非发送到{v*}，否则会被接收者确认。
多播组。如果没有确认，发送者可能
重发相同消息，在往返时间（RTT）超时后。
往返时间估计为TCP（RFC 1123）[18]中所述，其中{v*}保持不变。
初始往返值500毫秒。实现可以缓存
上一次RTT测量作为未来连接的初始值。

如果使用可靠的传输协议来承载RTSP，请求必须
不得重新传输；RTSP应用程序必须依赖
底层传输以提供可靠性。

如果底层可靠传输（如TCP）和RTSP
应用重传请求，每个数据包可能
损失导致两次重传。接收器通常无法
利用应用层重传功能，因为 {v*}

传输栈不会交付应用层
在第一次尝试到达接收者之前进行重传。
如果数据包丢失是由拥塞引起的，多个
重传不同层将加剧拥塞。

如果在小RTT局域网中使用RTSP，则使用标准程序
优化初始TCP往返估计，例如在中使用
T/TCP (RFC 1644) [22]，可能有益。

时间戳报头（第12.38节）用于避免
重传模糊问题 [23, 第301页] 并消除了这种需求
对于Karn算法。

每个请求在CSeq头部（第{v*}节）中携带一个序列号。
12.17），对于每个不同的请求增加一个
传输。如果由于缺乏而重复请求
确认，请求必须携带原始序列号
（即，序列号不会递增）。

系统实现RTSP必须支持通过TCP承载RTSP，并且可以
支持UDP。RTSP服务器的默认端口为554，适用于UDP
并且TCP。

一些目的地为同一控制端点的RTSP数据包可能
打包成一个单独的低层PDU或封装到TCP中
流。RTSP数据可能与RTP和RTCP数据包交织。
与HTTP不同，RTSP消息必须包含Content-Length头
每当该消息包含有效负载时。否则，一个RTSP数据包
在最后一个空行之后立即终止
消息头。

10 方法定义

方法令牌表示要对资源执行的方法
通过请求URI识别。方法是区分大小写的。新
方法可能在将来被定义。方法名称可能不能以
一个\$字符（十进制24）必须是标记。方法有
总结于表2中。

method	direction	object	requirement
DESCRIBE	C->S	P,S	recommended
ANNOUNCE	C->S, S->C	P,S	optional
GET_PARAMETER	C->S, S->C	P,S	optional
OPTIONS	C->S, S->C	P,S	required (S->C: optional)
PAUSE	C->S	P,S	recommended
PLAY	C->S	P,S	required
RECORD	C->S	P,S	optional
REDIRECT	S->C	P,S	optional
SETUP	C->S	S	required
SET_PARAMETER	C->S, S->C	P,S	optional
TEARDOWN	C->S	P,S	required

表 2：RTSP 方法概述、其方向以及 {v*}
对象（P：展示，S：流）它们所操作的对象

表2的说明：建议使用PAUSE，但不是必需的，因为一个完全功能的服务器可以被构建，但不支持此功能方法，例如，用于实时流。如果服务器不支持一个特定方法，它必须返回“501 未实现”和一个客户端不应再次尝试此服务器的方法。

10.1 选项

该行为与[H9.2]中描述的等效。一个OPTIONS请求可以随时发出，例如，如果客户端即将尝试一个非标准请求。它不影响服务器状态。

示例： Translated Text：示例：

C->S: 选项 * RTSP/1.0

CSeq: 1

需要: 隐式播放

代理要求：gzip压缩的消息

S->C: RTSP/1.0 200 正确

CSeq: 1

公共：DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE

请注意，这些必然是虚构的特征（人们希望我们不会故意忽略一个真正有用的功能因此我们可以在本节中有一个强有力的例子）。

10.2 描述

The DESCRIBE method retrieves the description of a presentation or 媒体对象由请求URL从服务器识别。它可能使用 客户端指定的描述格式 Accept 标头 理解。服务器响应以描述请求的内容 资源。DESCRIBE 响应对构成媒体 RTSP初始化阶段。

示例： Translated Text： 示例：

```
C->S: 描述 rtsp://server.example.com/fizzle/foo RTSP/1.0
CSeq: 312
接受: application/sdp, application/rtsl, application/mheg
```

```
S->C: RTSP/1.0 200 OK   S->C: RTSP/1.0 200 正常
CSeq: 312
日期：1997年1月23日 15:35:06 GMT
内容类型: application/sdp
内容长度：376
```

```
v=0
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP研讨会
i=关于会话描述协议的研讨会
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=只接收
m=音频 3456 RTP/AVP 0
m=视频 2232 RTP/AVP 31
m=白板 32416 UDP WB
a=方向:横排
```

响应DESCRIBE必须包含所有媒体初始化 信息关于它所描述的资源。如果是一个媒体客户端 从除DESCRIBE之外的其他来源获取表示描述 并且该描述包含一套完整的媒体初始化 参数，客户端应使用这些参数，而不是其他 请求通过RTSP获取相同媒体的描述。

此外，服务器不应将DESCRIBE响应作为手段 媒体间接引用。

明确的游戏规则需要建立，以便客户有 无歧义确定何时请求媒体初始化的方法 信息通过DESCRIBE获取，以及何时不使用。通过强制使用DESCRIBE

对包含所有流集的媒体初始化的响应
描述的内容，并劝阻使用DESCRIBE来描述媒体
间接引用，我们避免了可能由其他引起的循环问题
方法。

媒体初始化是任何基于RTSP的系统所需的要求，
但是RTSP规范并未规定必须这样做
通过DESCRIBE方法。RTSP客户端有三种方式
可能接收初始化信息：

- * 通过RTSP的DESCRIBE方法；
- * 通过其他协议（HTTP、电子邮件附件等）；
- * 通过命令行或标准输入（因此作为浏览器工作）
辅助应用程序通过SDP文件或其他媒体启动
初始化格式）。

在实用互操作性的利益下，它非常
建议最小服务器支持DESCRIBE方法，并且
强烈推荐最小客户端支持执行能力
作为一个接受媒体初始化文件的“辅助应用程序”
从标准输入、命令行和/或其他方式
适用于客户端的运行环境。

10.3 公告

公告方法有两个用途：

当从客户端发送到服务器时，ANNOUNCE帖子发布{v*}的描述
演示或由请求URL标识的媒体对象
服务器。当从服务器发送到客户端时，ANNOUNCE更新会话
实时描述。

如果向演示文稿中添加新的媒体流（例如，在直播期间）
演示（），整个演示描述应发送
再次，与其只是额外的组件，不如组件
可被删除。

示例： Translated Text：示例：

```
C->S: 公告 rtsp://server.example.com/fizzle/foo RTSP/1.0
CSeq: 312
日期：1997年1月23日 15:35:06 GMT
会话：47112344
内容类型: application/sdp
内容长度：332
```

```
v=0
o=mhandley 2890844526 2890845468 IN IP4 126.16.64.4
```


s=SDP研讨会
i=关于会话描述协议的研讨会
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=只接收
m=音频 3456 RTP/AVP 0
m=视频 2232 RTP/AVP 31

S->C: RTSP/1.0 200 正确
CSeq: 312

10.4 设置

SETUP 请求为一个 URI 指定要使用的传输机制用于流媒体。客户端可以发出一个SETUP请求来流已经播放以更改传输参数的服务器可以允许。如果它不允许这样做，它必须响应错误 "455 方法在此状态下无效"。为了任何中间防火墙，客户端必须指示传输参数即使它对这些参数没有影响，对于示例，其中服务器广播一个固定的组播地址。

由于SETUP包含所有传输初始化信息，防火墙和其他中间网络设备（需要此信息（）免了解析的繁重任务。描述已保留给媒体的响应，其中包含 {v*} 初始化。

传输头指定了可接受的传输参数。
数据传输客户端；响应将包含
服务器选择的传输参数。

C->S: 设置 rtsp://example.com/foo/bar/baz.rm RTSP/1.0
CSeq: 302
传输：RTP/AVP；单播；客户端端口=4588-4589

S->C: RTSP/1.0 200 正确
CSeq: 302
日期：1997年1月23日 15:35:06 GMT
会话：47112344
运输: RTP/AVP;单播;
客户端端口=4588-4589；服务器端口=6256-6257

服务器在响应SETUP时生成会话标识符请求。如果一个对服务器的SETUP请求包含一个会话标识符，服务器必须将此设置请求捆绑到

现有会话或返回错误 "459 聚合操作失败"
允许" (见第11.3.10节)。

10.5 玩

PLAY方法告诉服务器开始通过{v*}发送数据。
机制在SETUP中指定。客户端不得发出PLAY请求
直到任何未完成的SETUP请求被确认为止
成功。

The PLAY request positions the normal play time to the beginning of 播放请求将正常播放时间定位到开始处
指定的范围并交付流数据，直到结束
范围已达到。PLAY请求可能被流水线（排队）；服务器
必须按顺序排队执行PLAY请求。也就是说，一个PLAY
请求在先前的PLAY请求仍然活跃时到达
延迟至第一个完成。

这允许精确编辑。

例如，无论两个PLAY请求间隔多远
在下面的示例中，服务器将首先播放第10秒
通过15，然后，紧接着，20至25秒，并且
最后30秒至结束。

```
C->S: 播放 rtsp://audio.example.com/audio RTSP/1.0
CSeq: 835
会话：12345678
范围：npt=10-15
```

```
C->S: 播放 rtsp://audio.example.com/audio RTSP/1.0
CSeq: 836
会话：12345678
范围：npt=20-25
```

```
C->S: 播放 rtsp://audio.example.com/audio RTSP/1.0
CSeq: 837
会话：12345678
范围：npt=30-
```

查看PAUSE请求的描述以获取更多示例。

一个没有Range头的PLAY请求是合法的。它开始播放
从开始流除非流已被暂停。如果{v*}
流已被通过PAUSE暂停，流传输将在以下位置恢复。
暂停点。如果正在播放流，此类PLAY请求不会
进一步操作，并可由客户端用于测试服务器活动性。

The Range header may also contain a time parameter. This parameter 指定了一个UTC时间，播放应在该时间开始。如果消息在指定时间后收到，开始播放立即。时间参数可用于帮助同步从不同来源获得的流。

对于按需流，服务器回复实际的范围，其中 {v*} 保持不变。将被播放。这可能不同于请求的范围，如果对请求的范围与有效帧边界对齐是 the media source. 如果在源中没有指定范围，请求，当前位置在回复中返回。单位的回复中的范围与请求中的范围相同。

在播放所需范围之后，演示文稿会自动暂停，仿佛已发出一个PAUSE请求。

以下示例从SMPTE开始播放整个演示文稿
时间码 0:10:20 至剪辑结束。播放将从此处开始
1997年1月23日15:36。

```
C->S: 播放 rtsp://audio.example.com/twister.en RTSP/1.0
CSeq: 833
会话: 12345678
范围: smpte=0:10:20- ; 时间=19970123T153600Z
```

```
S->C: RTSP/1.0 200 正确
CSeq: 833
日期: 1997年1月23日 15:35:06 GMT
范围: smpte=0:10:22- ; 时间=19970123T153600Z
```

为了播放现场演示的录音，可能需要期望使用时钟单位：

```
C->S: 播放 rtsp://audio.example.com/meeting.en RTSP/1.0
CSeq: 835
会话: 12345678
范围: 时钟=19961108T142300Z-19961108T143520Z
```

```
S->C: RTSP/1.0 200 正确
CSeq: 835
日期: 1997年1月23日 15:35:06 GMT
```

一个仅支持播放的媒体服务器必须支持npt格式并且可能支持时钟和SMPTE格式。

10.6 暂停

PAUSE 请求导致流传输中断 (暂停)暂时。如果请求URL指定了一个流,则仅播放和录制该流已停止。例如,对于音频,这相当于静音。如果请求的URL指定了{v*}演示或流组,所有当前活动流的交付流在演示或组内的播放被暂停。在恢复后回放或录音,轨道的同步必须维护。任何服务器资源都保留,尽管服务器可能会关闭会话暂停后的持续时间以及免费资源指定于Session头中的超时参数设置消息。

示例: Translated Text: 示例:

```
C->S: 暂停 rtsp://example.com/fizzle/foo RTSP/1.0
CSeq: 834
会话: 12345678
```

```
S->C: RTSP/1.0 200 OK   S->C: RTSP/1.0 200 正常
CSeq: 834
日期: 1997年1月23日 15:35:06 GMT
```

The PAUSE request may contain a Range header specifying when the
流或演示应停止。我们将这一点称为

"暂停点"。标题必须恰好包含一个值,而不是一个时间范围。流的正常播放时间设置为暂停点。暂停请求在服务器第一次正在遇到当前指定的任何时间点待处理 PLAY 请求。如果 Range 头指定的时间在任何当前挂起的PLAY请求,错误 "457 无效范围" 是返回。如果一个媒体单元(如音频或视频帧)开始在恰好暂停点进行演示,它不会被播放或已记录。如果缺少Range头,则流式传输无效。立即在收到消息和暂停点中断设置为当前正常播放时间。

Translated Text: PAUSE 请求可能包含一个指定何时

一个PAUSE请求将丢弃所有排队的PLAY请求。然而,暂停媒体流中的点必须保持。后续的播放请求没有Range头从暂停点继续。

例如,如果服务器有对范围10到15的播放请求,20 到 29 个待处理,然后收到 NPT 21 的暂停请求,它将开始播放第二个范围,并在NPT 21处停止。如果暂停请求为NPT 12,而服务器正在NPT 13处播放,提供服务第一次播放请求,服务器立即停止。如果暂停请求为NPT 16,服务器在完成第一个后停止

播放请求并丢弃第二个播放请求。

作为另一个例子，如果一个服务器已收到播放范围请求
10 至 15 然后是 13 至 20（即重叠范围），PAUSE
请求NPT=14将在服务器播放第一个时生效
范围，第二个PLAY请求实际上被忽略，
假设PAUSE请求在服务器启动之前到达
播放第二个，重叠的范围。无论何时暂停
请求到达时，它将NPT设置为14。

如果服务器已经发送了超过指定时间的数据
范围标题，一个PLAY仍然会在那个时间点继续播放
假设客户端在此之后已丢弃数据。这
确保连续的暂停/播放循环，无间隔。

10.7 拆解

TEARDOWN请求停止给定URI的流传输，
释放与其相关的资源。如果URI是
演示 URI，此演示的任何 RTSP 会话标识符
与该会话相关的不再有效。除非所有传输
参数由会话描述和SETUP请求定义
必须在再次播放会话之前发出。

示例： Translated Text：示例：

```
C->S: 拆卸 rtsp://example.com/fizzle/foo RTSP/1.0
CSeq: 892
会话：12345678
S->C: RTSP/1.0 200 OK   S->C: RTSP/1.0 200 正常
CSeq: 892
```

10.8 获取参数

The GET_PARAMETER request retrieves the value of a parameter of a
在URI中指定的演示文稿或流。回复的内容
响应留给实现。无参数的 GET_PARAMETER
实体主体可用于测试客户端或服务存活状态（“ping”）。

示例： Translated Text：示例：

```
S->C: GET_PARAMETER rtsp://example.com/fizzle/foo RTSP/1.0
CSeq: 431
内容类型: text/parameters
会话：12345678
内容长度：15
```

收到的数据包
抖动

C->S: RTSP/1.0 200 OK C->S : RTSP/1.0 200 OK

CSeq: 431

内容长度 : 46

内容类型: text/parameters

数据包接收数 : 10

抖动: 0.3838

"文本/参数"部分仅是一个示例类型。
参数。这种方法故意定义得比较宽松，与
意图是回复内容和响应内容将是
定义在进一步实验之后。

10.9 设置参数

此方法请求设置一个参数的值，对于
由URI指定的演示文稿或流。

一个请求应只包含一个参数，以允许客户端
确定为什么某个特定请求失败。如果请求包含
多个参数，服务器必须仅在所有以下条件满足时才对请求进行操作
参数可以成功设置。服务器必须允许
参数需要反复设置为相同的值，但它可能不允许
更改参数值。

注意：媒体流的传输参数必须仅使用 {v*} 设置
SETUP命令。

限制设置传输参数为SETUP
防火墙的好处。

参数以细粒度方式分割，以便有{v*}
可以提供更有意义的错误指示。然而，它可能会
感觉允许设置多个参数，如果是一个原子
设置是可取的。想象一下客户端控制设备的情况，其中客户端执行
不想相机平移除除非它也能向右倾斜
同时的角度。

示例： Translated Text : 示例：

C->S: 设置参数 rtsp://example.com/fizzle/foo RTSP/1.0

CSeq: 421

内容长度 : 20

内容类型: text/parameters

barparam: barstuff

S->C: RTSP/1.0 451 无效参数

CSeq: 421 翻译文本：CSeq: 421
内容长度：10
内容类型：text/parameters

barparam

"文本/参数"部分仅是一个示例类型。
参数。这种方法故意定义得比较宽松，与
意图是回复内容和响应内容将是
定义在进一步实验之后。

10.10 重定向

一个重定向请求告知客户端它必须连接到另一个
服务器位置。它包含必填的头部 Location，其中
指示客户端应针对该URL发起请求。它可能
包含参数范围，表示重定向
生效。如果客户端想要继续发送或接收
媒体为此URI，客户端必须发出一个{TEARDOWN}请求。
当前会话以及为指定会话设置的SETUP
主机。

此示例请求将此URI的流量重定向到新服务器
在给定的播放时间：

S->C: 重定向 rtsp://example.com/fizzle/foo RTSP/1.0
CSeq: 732
位置：rtsp://bigserver.com:8001
范围：时钟=19960213T143205Z-

10.11 记录

此方法根据初始化记录一系列媒体数据
演示描述。时间戳反映开始和结束
时间（UTC）。如果没有给出时间范围，则使用开始或结束时间
提供在演示描述中。如果会议已经
开始，立即开始录制。

服务器决定是否将记录的数据存储在
请求URI或另一个URI。如果服务器不使用请求-
URI，响应应当是201（已创建）并包含一个实体
描述请求的状态并引用新资源，
并且一个位置头。

一个支持录制现场演示的多媒体服务器必须
支持时钟范围格式；SMPTE格式没有意义。

在这个例子中，媒体服务器之前被邀请到会议指出。

```
C->S: 记录 rtsp://example.com/meeting/audio.en RTSP/1.0
CSeq: 954
会话：12345678
会议：128.16.64.19/32492374
```

10.12 嵌入（交织）二进制数据

某些防火墙设计和其他情况可能迫使服务器将RTSP方法和流数据交织。这种交织应通常应避免，除非必要，因为它会复杂化客户端和服务器操作并施加额外的开销。交错二进制数据应仅在使用TCP承载RTSP时使用。

流数据，如RTP数据包，被ASCII美元符号封装签名（24十六进制），随后是一个一字节通道标识符，随后是封装的二进制数据的长度，以二进制形式表示，{v*}网络字节序的双字节整数。流数据随后立即之后，没有CRLF，但包括上层协议头。每个\$块恰好包含一个上层协议数据单元，例如，一个RTP数据包。

通道标识符在传输头中定义，其格式为 {v*}。
交错参数（第12.39节）。

当传输选择为RTP时，RTCP消息也会交织通过服务器在TCP连接上。默认情况下，RTCP数据包是发送到高于RTP通道的第一个可用通道。客户端可以显式请求在另一个通道上的RTCP数据包。通过指定两个通道在交织参数中完成运输头（第12.39节）。

RTCP 在两个或更多流进行同步时是必需的交错地以这种方式。此外，这也提供了一种方便的方法当通过TCP控制连接隧道RTP/RTCP数据包时由网络配置所必需的并将它们传输到UDP当可能时。

```
C->S: 设置 rtsp://foo.com/bar.file RTSP/1.0
CSeq: 2  翻译文本：CSeq: 2
运输：RTP/AVP/TCP；交错=0-1
```

```
S->C: RTSP/1.0 200 正确
CSeq: 2  翻译文本：CSeq: 2
日期：1997年6月5日 18:57:18 GMT
运输：RTP/AVP/TCP；交错=0-1
```


会话：12345678

C->S: 播放 rtsp://foo.com/bar.file RTSP/1.0

CSeq: 3

会话：12345678

S->C: RTSP/1.0 200 OK S->C: RTSP/1.0 200 正常

CSeq: 3 翻译文本：CSeq: 3

会话：12345678

日期：1997年6月5日 18:59:15 GMT

RTP-Info: url=rtsp://foo.com/bar.file;

seq=232433;rtptime=972948234

S->C: \$000{2 字节长度}{"长度"字节数据，带RTP头部

}` 请注意，源文本中只有一个

S->C: \$000{2 字节长度}{"长度"字节数据，带RTP头部

}` 请注意，源文本中只有一个

S->C: \$001{2 字节长度}{"长度"字节 RTCP 数据包}

11 状态码定义

适用于的情况，HTTP状态[H10]代码被重用。状态代码在此不重复具有相同意义的项。参见表1。
列出哪些请求可能返回哪些状态码。

11.1 成功 2xx

11.1.1 250 存储空间不足

服务器在接收到一个RECORD请求后返回此警告，该请求可能无法完全满足，因为存储不足
空格。如果可能的话，服务器应使用Range头
指示可能还能记录的时间段。由于其他
服务器上的进程可能正在消耗存储空间
同时，客户应仅将其视为一个估计值。

11.2 重定向 3xx

参见 [H10.3]。

在RTSP中，重定向可用于负载均衡或
将流请求重定向到拓扑上更接近的服务器
客户端。确定拓扑邻近性的机制超出了
本规范的范围。

11.3 客户端错误 4xx

11.3.1 405 方法不允许

请求中指定的方法不允许用于该资源
由请求URI识别。响应必须包含一个Allow
标题包含请求资源的有效方法列表。
此状态码也用于请求尝试使用
方法在SETUP期间未指定，例如，如果是一个RECORD请求
即使传输头中的模式参数只有
指定的播放。

11.3.2 451 参数未理解

请求的接收者不支持一个或多个参数
包含在请求中。

11.3.3 452 会议未找到

会议标题字段指示的会议是未知的
媒体服务器。

11.3.4 453 网络带宽不足

请求被拒绝，因为带宽不足。
这可能是由资源预留引起的，例如
失败。

11.3.5 454 会话未找到

RTSP会话标识符在会话头中缺失，
无效，或已超时。

11.3.6 455 此状态下方法无效

客户端或服务器无法在当前状态下处理此请求
状态。响应应包含一个Allow头以使错误
恢复更容易。

11.3.7 456 资源标题字段无效

服务器无法对所需的请求头进行操作。例如，
如果PLAY包含Range头部字段但流不允许
寻找。

11.3.8 457 无效范围

给定的范围值超出范围，例如，超出末尾演示。

11.3.9 458 参数为只读

The parameter to be set by SET_PARAMETER can be read but not 参数由SET_PARAMETER设置的参数可以读取但不能修改后。

11.3.10 459 聚合操作不允许

请求的方法可能不适用于所讨论的URL，因为这是一个聚合（展示）URL。该方法可以应用于流URL。

11.3.11 460 仅允许聚合操作

请求的方法可能不适用于所讨论的URL，因为它不是一个聚合（展示）URL。该方法可以应用在演示URL上。

11.3.12 461 不支持的传输

传输字段不包含支持的传输规范。

11.3.13 462 目的地不可达

数据传输通道无法建立，因为客户端地址无法访问。此错误最有可能客户端尝试放置无效目的地的结果参数在传输字段中。

11.3.14 551 选项不受支持

An option given in the Require or the Proxy-Require fields was not 受支持。应返回“不受支持”的标题，表明选项不支持的情况。

12 报头字段定义

HTTP/1.1 [2] 或其他，此处未列出的非标准头部字段
当前没有明确的含义，应当被忽略
接收者。

表3总结了RTSP使用的标题字段。输入“g”
指代在请求和响应中都可以找到的通用请求头
响应，类型“R”表示请求头，类型“r”表示
响应头，并且类型“e”表示实体头字段。
字段在“支持”列中标记为“req.”的必须
由接收者针对特定方法实现，而字段
标记为“opt.”的为可选。请注意，并非所有标记为“req.”的字段都是必需的。
将被发送在每个此类请求中。“req.”仅表示
该客户端（用于响应头）和服务端（用于请求头）
必须实现字段。最后一列列出了对应的方法。
此报头字段具有意义；“实体”这个名称指的是
所有返回消息体的方法。在本规范中，
DESCRIBE 和 GET_PARAMETER 属于此类。

Header	type	support	methods
Accept	R	opt.	entity
Accept-Encoding	R	opt.	entity
Accept-Language	R	opt.	all
Allow	r	opt.	all
Authorization	R	opt.	all
Bandwidth	R	opt.	all
Blocksize	R	opt.	all but OPTIONS, TEARDOWN
Cache-Control	g	opt.	SETUP
Conference	R	opt.	SETUP
Connection	g	req.	all
Content-Base	e	opt.	entity
Content-Encoding	e	req.	SET_PARAMETER
Content-Encoding	e	req.	DESCRIBE, ANNOUNCE
Content-Language	e	req.	DESCRIBE, ANNOUNCE
Content-Length	e	req.	SET_PARAMETER, ANNOUNCE
Content-Length	e	req.	entity
Content-Location	e	opt.	entity
Content-Type	e	req.	SET_PARAMETER, ANNOUNCE
Content-Type	r	req.	entity
CSeq	g	req.	all
Date	g	opt.	all
Expires	e	opt.	DESCRIBE, ANNOUNCE
From	R	opt.	all
If-Modified-Since	R	opt.	DESCRIBE, SETUP
Last-Modified	e	opt.	entity
Proxy-Authenticate			
Proxy-Require	R	req.	all
Public	r	opt.	all
Range	R	opt.	PLAY, PAUSE, RECORD
Range	r	opt.	PLAY, PAUSE, RECORD
Referer	R	opt.	all
Require	R	req.	all
Retry-After	r	opt.	all
RTP-Info	r	req.	PLAY
Scale	Rr	opt.	PLAY, RECORD
Session	Rr	req.	all but SETUP, OPTIONS
Server	r	opt.	all
Speed	Rr	opt.	PLAY
Transport	Rr	req.	SETUP
Unsupported	r	req.	all
User-Agent	R	opt.	all
Via	g	opt.	all
WWW-Authenticate	r	opt.	all

RTSP头部字段概述

12.1 接受

The Accept request-header field can be used to specify certain
演示描述内容类型，这些类型是可接受的
响应。

The "level" parameter for presentation descriptions is properly The "级别"参数对于演示描述是正确的
定义为 MIME 类型注册的一部分，而不是在这里。

查看[H14.1]以了解语法。

使用示例：

Accept: 应用/rtsp, 应用/sdp;level

level=2 翻译文本：level=2

12.2 接受编码

参见 [H14.3]

12.3 接受语言

参见[H14.4]。注意，指定的语言适用于
演示描述和任何原因短语，不是媒体
内容。

12.4 允许

The Allow response header field lists the methods supported by the 允许响应头字段列出了支持的方法
资源由请求URI标识。此字段的目的是
严格告知收件人有关 {v*} 相关的有效方法
资源。必须在405（方法不允许）响应中存在一个Allow头部字段。
允许的)响应。

使用示例：

允许：SETUP, PLAY, RECORD, SET_PARAMETER

12.5 授权

参见 [H14.8]

12.6 带宽

带宽请求头字段描述了估计的带宽
对客户端可用，表示为正整数并测量
每秒比特。客户端可用的带宽可能会改变
在RTSP会话期间，例如，由于调制解调器再训练。

带宽 = "带宽" ":" 1*数字

示例： Translated Text： 示例：
带宽：4000

12.7 块大小

此请求头字段是从客户端发送到媒体服务器的
请求服务器获取特定媒体数据包大小。此数据包
大小不包括IP、UDP或RTP等底层头信息。
服务器可以自由使用低于一个块大小的块大小
请求的。服务器可以将此数据包大小截断到最近的
最小值的倍数，媒体特定的块大小，或覆盖它
如有必要，使用媒体特定的尺寸。块大小**必须**是
正数十进制数，以字节为单位。服务器仅返回
如果值在语法上无效，则出现错误（416）。

12.8 缓存控制

缓存控制通用报头字段用于指定指令
该内容必须被所有缓存机制遵守
请求/响应链。

缓存指令必须通过代理或网关传递
应用，无论其对该应用的重要性如何，
自指令可能适用于所有收件人沿
请求/响应链。无法指定缓存-
特定缓存的指令。

缓存控制应仅在SETUP请求中指定，其
响应。注意：Cache-Control 不控制 {v*} 的缓存
响应类似于HTTP，而是由{v*}标识的流
SETUP 请求。对 RTSP 请求的响应不可缓存，除非
对于DESCRIBE的响应。

```
Cache-Control           = "Cache-Control" ":" 1#cache-directive
cache-directive         = cache-request-directive
                        | cache-response-directive
cache-request-directive = "no-cache"
                        | "max-stale"
                        | "min-fresh"
                        | "only-if-cached"
                        | cache-extension
cache-response-directive = "public"
                        | "private"
                        | "no-cache"
                        | "no-transform"
                        | "must-revalidate"
```

缓存扩展 | "代理重新验证"
 | "max-age" "=" delta-seconds
 | 缓存扩展
 =令牌 ["=" (令牌 | 引号字符串)]

不缓存：

指示媒体流不得在任何地方缓存。
 这允许源服务器防止缓存，即使是通过缓存
 已配置为向客户端返回过时响应的
 请求。

公开：

指示媒体流可以被任何缓存缓存。

私有：

指示该媒体流是为单个用户设计的
 并且**不得**由共享缓存进行缓存。私有（非-
 共享缓存可能会缓存媒体流。

不转换：

一个中间缓存（代理）可能发现将其转换为 有用。
 媒体类型的一个特定流。代理可能会，对于
 示例，在视频格式之间转换以节省缓存空间或
 为了减少慢链路上的流量。严重
 操作问题可能发生，然而，当这些
 变换已应用于旨在的流
 某些类型的应用程序。例如，用于
 医学影像，科学数据分析以及那些使用
 端到端认证全部依赖于接收一个流
 与原始实体完全位对位相同。
 因此，如果一个响应包含no-transform指令，
 一个中间缓存或代理不得更改{v*}的编码
 流。与HTTP不同，RTSP不提供部分
 此点处的转换，例如，允许翻译为
 一种不同的语言。

仅当缓存中存在时：

在某些情况下，例如网络极其糟糕的时候
 连接性，客户端可能希望缓存只返回那些
 当前存储的媒体流，且不接受
 这些来自源服务器。为此，客户端可能
 在请求中包含 only-if-cached 指令。如果它
 接收此指令后，缓存应使用以下指令响应
 缓存的与其它一致的媒体流
 请求的约束，或响应以504（网关
 超时）状态。然而，如果一组缓存正在
 作为具有良好内部连接的统一系统运行，
 此类请求可以在该组缓存内转发。

最大过时时间：

指示客户端愿意接受一个媒体流
该已超过其过期时间。如果max-stale是
分配一个值，然后客户端愿意接受一个
响应未超过其过期时间
指定的秒数。如果没有分配值给
max-stale，然后客户端愿意接受一个过时的
任何年龄的响应。

min-fresh:

指示客户端愿意接受一个媒体流
其新鲜度寿命不少于其当前年龄加上
指定的秒数。即，客户端想要
响应将至少保持指定的新鲜
秒数。

必须重新验证：

当必须在SETUP中存在must-revalidate指令时
响应由缓存接收，该缓存必须不使用
条目在其过时后响应后续
请求不先与源服务器重新验证它。
这意味着缓存必须进行端到端的重新验证，每
时间，如果，仅根据原始服务器的Expires
缓存响应已过时。）

12.9 会议

此请求头字段在一条{v*}-之间建立逻辑连接。
预先建立的会议和RTSP流。会议ID必须
不对同一RTSP会话进行更改。

会议 = "会议" ":" 会议-id 示例：

会议：199702170042.SAA08642@obiwan.arl.wustl.edu%20Starr

返回452（452会议未找到）响应代码，如果{v*}
会议ID无效。

12.10 连接 查看 [H14.10] 12.

11 内容基础 查看 [H14.11] 1

2.12 内容编码 查看 [H14.12]

12.13 内容语言

参见 [H14.13]

12.14 内容长度

此字段包含方法内容长度（即在最后一个标题之后的双CRLF处）。与HTTP不同，它必须在所有包含除标题之外内容的消息中包含{v*}消息的一部分。如果它缺失，则默认值为零。假定。它根据[H14.14]进行解释。

12.15 内容位置

参见 [H14.15]

12.16 内容类型

参见[H14.18]。请注意，适用于RTSP的内容类型是可能在实际中仅限于表示描述和参数值类型。

12.17 CSeq

CSeq字段指定RTSP请求的序列号-响应对。此字段必须在所有请求中存在并响应。对于包含给定序列的每个RTSP请求数字，将会有有一个相应的响应具有相同的数字。任何重传的请求必须包含相同的序列数字作为原始值（即序列号不会增加）对于相同请求的重传）。

12.18 日期

参见 [H14.19]。

12.19 过期

The Expires entity-header field gives a date and time after which the 过期实体头部字段给出了一个日期和时间，之后描述或媒体流应被视为过时。
解释取决于方法：

描述响应：

The Expires header indicates a date and time after which the 过期头指示一个日期和时间，在此之后描述应被视为过时。

一个过期的缓存条目通常不会被缓存（无论是缓存还是缓存）返回。
代理缓存或用户代理缓存）除非它首先经过验证
原始服务器（或带有新鲜缓存的中间缓存）
实体副本）。参见第13节以获取更多讨论。
到期模型。

存在一个过期字段并不意味着原始
资源将在、之前或之后发生变化或不再存在
时间

The format is an absolute date and time as defined by HTTP-date in 格式是由HTTP-date定义的绝对日期和时间
[H3.3]; 它**必须**采用RFC1123日期格式：

过期 = "过期" ":" HTTP日期

一个其使用的例子是

到期时间：星期四，01 12月 1994 16:00:00 GMT

RTSP/1.0客户端和缓存必须处理其他无效日期格式，
特别包括值"0"，如过去发生过的那样
(即，“已过期”）。

为了标记一个响应为“已过期”，源服务器应使用
一个与Date报头值相等的过期日期。为了标记
响应为“永不失效”，源服务器应使用Expires
大约在回复发送后一年左右的时间。
RTSP/1.0 服务器不应发送超过一年的过期日期
未来。

存在一个带有日期值的Expires头字段
未来在媒体流中的时间，否则默认情况下
非缓存表示媒体流是可缓存的，除非
指示由 Cache-Control 头字段（第 12.8 节）另行说明。

12.20 从

参见 [H14.22]。

12.21 主机

这个HTTP请求头字段对于RTSP不是必需的。它应该是
静默忽略，如果发送。

12.22 If-Match

参见 [H14.25]。

此字段特别有用，以确保{v*}的完整性
 演示描述，在通过 {v*} 获取的情况下
 表示外部于RTSP（如HTTP），或在以下情况下：
 服务器实现保证了{v*}的完整性
 描述在DESCRIBE消息和SETUP之间的时间
 消息。

标识符是一个不透明的标识符，因此它不是特定的
 任何特定的会话描述语言。

12.23 If-Modified-Since

The If-Modified-Since request-header field is used with the DESCRIBE If-Modified-Since 请求头字段与 DESCRIBE 一起使
 并且设置方法以使它们具有条件性。如果请求的变体
 自本字段指定的时间以来尚未修改，a
 描述将不会从服务器返回（DESCRIBE）或
 流将不会被设置（设置）。取而代之的是，一个304（未修改）
 响应将返回，不包含任何消息体。

如果修改后 = "如果修改后" ":" HTTP日期

一个域的例子是：

如果 -Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT -自修改以来: Sat, 29 Oct 1994 19:43:31 GMT

12.24 最后修改时间

最后修改实体头部字段指示日期和时间，
 哪个原始服务器认为的表示描述或
 媒体流最后被修改。参见[H14.29]。有关方法
 描述或公告，标题字段指示最后
 修改描述的日期和时间，对于SETUP则是其日期和时间
 媒体流。

12.25 位置

参见 [H14.30]。

12.26 代理身份验证

参见 [H14.33]。

12.27 代理-要求

代理-要求头用于指示代理敏感功能
 该代理必须支持。任何Proxy-Require头
 特性不支持代理的必须被负向
 被代理确认，如果不受支持。服务器

应将此字段与“要求”字段同等对待。

查看第12.32节以获取有关此消息机制的更多详细信息
并且一个使用示例。

12.28 公共

参见 [H14.35]。

12.29 范围

此请求和响应头字段指定了一个时间范围。
范围可以用多个单位指定。此规范
定义了smpte（第3.5节）、npt（第3.6节）和时钟
（第3.7节）范围单位。在RTSP中，字节范围 [H14.36.1] 是
无意义且**必须**不使用。标题也可能包含一个
时间参数为UTC，指定操作的时间
生效。支持Range头的服务器必须
理解NPT范围格式，并且应该理解SMPTE范围
格式。Range 响应头指示了哪个时间段
实际上正在播放或录制。如果Range头给出在
时间格式无法识别，收件人应返回“501”
未实现”。

范围是半开区间，包括下限点，但
排除上一点。换句话说，一个范围从a到b开始
正好在时间a，但停止在b之前。只有a的开始时间
媒体单元，如视频或音频帧，是相关的。例如，
假设视频帧每40毫秒生成一次。范围是10.0-
10.1 将包括从 10.0 或更晚时间开始的视频帧
将包括一个从10.08开始的视频帧，即使它持续了
超出区间。另一方面，一个范围是10.0-10.08
排除10.08时的帧

```
Range           = "Range" ":" 1\#ranges-specifier  
                  [ ";" "time" "=" utc-time ]  
ranges-specifier = npt-range | utc-range | smpte-range
```

示例： Translated Text：示例：

范围：时钟=19960213T143205Z-；时间=19970123T143720Z

该表示法与用于HTTP/1.1 [2] 字节的表示法类似。
范围标题。它允许客户端从媒体中选择摘录
对象，以及从给定点播放到结束，以及从
当前定位到指定点。播放开始可以
计划在未来任何时间进行，尽管服务器可能会拒绝
保留服务器资源以延长空闲期。

12.30 引用者

参见[H14.37]。该URL指的是演示描述的URL，通常通过HTTP获取。

12.31 Retry-After

参见 [H14.38]。

12.32 需求

The Require header is used by clients to query the server about 需要头由客户端用于查询服务器关于选项它可能支持也可能不支持。服务器必须响应使用不支持的标题来负确认此标题那些不受支持的选项。

这是为了确保客户端-服务器交互将立即进行，当双方都理解所有选项时边，并且只有在选项不被理解时才会减速（如在{v*}中）。案例上述）。对于匹配良好的客户端-服务器对，交互收益迅速，节省了通常由谈判所需的往返行程机制。此外，它还在以下情况下消除了状态歧义：客户端需要服务器不理解的功能。

需要 = "需要" ":" 1#选项标签

示例： Translated Text：示例：

C->S: 设置 UP rtsp://server.com/foo/bar/baz.rm RTSP/1.0

CSeq: 302

需要: funky-feature

Funky-Parameter: funkystuff

S->C: RTSP/1.0 551 选项不受支持

CSeq: 302

不支持: funky-feature

C->S: 设置 UP rtsp://server.com/foo/bar/baz.rm RTSP/1.0

CSeq: 303

S->C: RTSP/1.0 200 正常

CSeq: 303

在这个例子中，"funky-feature" 是特征标签，它表示客户端需要虚构的Funky-Parameter字段。

"Funky-Feature"和Funky-Parameter之间的关系不是通过RTSP交换进行通信，因为该关系是一个不可变属性 "funky-feature" 因此不应每次交换时都传输。

代理和其他中介设备应忽略以下功能：
在此领域不被理解。如果特定的扩展需要
中间设备支持它时，扩展应该被标记
在代理要求字段中而不是（见第12.27节）。

12.33 RTP-Info

此字段用于在PLAY中设置RTP特定的参数
响应。

url: 表示以下RTP参数对应的流URL。 seq:

指示第一个数据包的序列号
流。这允许客户端优雅地处理数据包
在寻找时。客户端使用此值来区分
数据包来自在查找之前产生的数据包
在寻求之后产生。

实时时间：

指示与时间值对应的RTP时间戳
范围响应头。（注意：对于聚合控制，一个
特定流可能实际上不会生成一个数据包。
范围时间值返回或暗示。因此，没有
保证指示序列号的包
通过seq实际上具有由rtptime指示的时间戳。）
客户端使用此值来计算RTP时间映射
NPT.

从RTP时间戳到NTP时间戳（系统时间）的映射是
通过RTCP可用。然而，此信息不足以
生成从RTP时间戳到NPT的映射。此外，在
为确保此信息在必要时可用
时间（立即在启动或寻道后），并且它是
可靠交付，此映射放置在RTSP控制中
通道

为了补偿长时间的连续漂移
演示文稿，RTSP客户端还应将NPT映射到NTP，
使用初始RTCP发送者报告进行映射，之后
报告以检查与映射的漂移。

语法：

```
RTP-Info      = "RTP-Info" ":" 1#stream-url 1*参数
流式URL       = "url" "=" url
参数          = ";" "seq" "=" 1*数字
              | ";" "rtptime" "=" 1*数字
```

示例： Translated Text： 示例：

```
RTP-Info: url=rtsp://foo.com/bar.avi/streamid=0;seq=45102 ,
url=rtsp://foo.com/bar.avi/streamid=1;seq=30211
```

12.34 比例

1 的刻度值表示正常播放或记录在正常情况下
前瞻视率。如果不为1，则该值对应于比率
关于正常观看速率。例如，比例为2
指示为正常观看速率的两倍（“快进”）和比率
0.5表示正常观看速度的一半。换句话说，一个
2的比率具有正常游戏时间以两倍于wallclock rate的速度增加。
对于每秒已过（墙钟）时间，内容持续2秒
将送达。负值表示反向方向。

除非Speed参数要求否则，数据速率
应当保持不变。缩放变化的实现取决于
服务器和媒体类型。对于视频，例如，服务器可能会提供
仅关键帧或选定的关键帧。对于音频，它可能进行时间缩放
音频同时保留音高或，不那么理想地，传递
音频片段。

服务器应尝试近似观看速率，但可能
限制它支持的缩放值范围。响应
必须包含服务器选择的实际缩放值。

如果请求包含一个Range参数，新的缩放值将
当时生效。

缩放 = "缩放" ":" ["-"] 1*数字 ["." *数字]

示例以3.5倍正常速度反向播放：

比例：-3.5

12.35 速度

此请求头字段参数请求服务器交付数据以特定速度发送给客户端，取决于服务器能力与愿望以给定速度服务媒体流。服务器实现是可选的。默认为比特率的流。

参数值以十进制比例表示，例如，一个值为2.0表示数据将以正常速度的两倍快速交付。A零速无效。如果请求包含一个Range参数，新速度值将在那时生效。

速度 = "速度" ":" 1*数字 ["." *数字]

示例： Translated Text： 示例：
速度：2.5

使用此字段会改变用于数据传输的带宽。它是适用于特定情况下预览的。在更高或更低的速率下进行演示是必要的。实施者应记住会话的带宽可能已协商。事先（通过除RTSP之外的其他方式），因此重新协商可能有必要。当数据通过UDP传输时，它非常建议使用如RTCP之类的机制来跟踪数据包丢失费率。

12.36 服务器

参见 [H14.39]

12.37 会话

此请求和响应头字段标识一个RTSP会话。由媒体服务器在SETUP响应中启动并结束。在演示URL上执行拆解。会话标识符由媒体服务器（见第3.4节）。一旦客户端收到一个会话标识符，它必须对任何与该标识符相关的请求返回它。会话。服务器不必设置会话标识符，如果它具有其他识别会话的方法，例如动态生成的URL。

会话 = "会话" ":" 会话-id [";" "超时" "=" 时间增量秒]

超时参数仅允许出现在响应头中。服务器用它来指示客户端服务器持续了多长时间。准备在关闭会话前等待 RTSP 命令之间活动不足（见A节）。超时以{v*}为单位测量。

秒，默认为60秒（1分钟）。

请注意，会话标识符用于标识 RTSP 会话跨传输会话或连接。控制消息超过一个 RTSP URL 可能在一个 RTSP 会话中发送。因此，它是可能客户端会使用相同的会话来控制许多流组成一个演示，只要所有流都来自同一服务器。（参见第14节中的示例）。然而，多个“用户”会话对于同一URL来自同一客户端必须使用不同的会话标识符。

会话标识符用于区分多个交付
来自同一客户端的相同URL的请求。

如果会话未找到，则返回响应 454（会话未找到）。
标识符无效。

12.38 时间戳

客户端发送时间戳通用头描述了何时请求服务器。时间戳的值具有意义仅对客户端有效，可以使用任何时间尺度。服务器必须回显与完全相同的值或可能，如果它有准确的信息关于这，添加一个表示秒数的浮点数自它收到请求以来已过去的时间。时间戳是由客户端用于计算到服务器的往返时间，因此它可以调整重传的超时值。

时间戳 = "时间戳" ":" *(数字) ["." *(数字)] [延迟]
延迟 = *(数字) ["." *(数字)]

12.39 交通运输

此请求头指示要使用哪种传输协议并且配置其参数，如目标地址，压缩、单次多播的生存时间和目标端口流。它设置那些未被演示所确定的值描述。

运输项以逗号分隔，按优先顺序列出。
参数可以添加到每个传输中，用分号分隔。

The Transport header MAY also be used to change certain transport parameters. 服务器可以拒绝更改现有参数。
流 传输头也可能被用来更改某些传输

服务器可能在响应中返回一个传输响应头，以响应指示实际选择的值。

一个传输请求头字段可能包含一个传输列表
可接受客户的选项。在这种情况下，服务器必须
返回实际选择的单个选项。

The syntax for the transport specifier is The syntax for the 传输指定符是
传输/配置文件/下传输。

默认值对于“lower-transport”参数是特定的。
配置文件。对于RTP/AVP，默认为UDP。

以下是一些 e 配置参数相关

运输：

通用参数：

单播 | 多播：

互斥指示是否为单播或多播
交付将尝试。默认值是多播。
客户端能够处理单播和多播
多播传输必须通过以下方式指示这种能力：
包括两个完整的运输规范，具有单独的参数
对于每个。

目标：

目标地址，流将被发送到。客户端可能
指定多播地址的destination参数。
为了避免成为无意识的远程-的肇事者
受控拒绝服务攻击，服务器应
验证客户端并应在之前记录此类尝试
允许客户端将媒体流直接导向一个地址，而不是
由服务器选择。这在RTSP特别重要。
命令通过UDP发出，但实现不能依赖于
作为自身客户端识别的可靠手段的TCP。
服务器不应允许客户端直接将媒体流定向到
地址与地址命令不同的地址
从。

源： {v*}

如果流源地址与{v*}不同
从RTSP端点地址（播放服务器）派生而来
或指定客户端在记录中的源，源可能被指定。

此信息也可能通过SDP获取。然而，由于
这是运输的特性，而不是媒体初始化的特性，{v*}
权威信息来源应在SETUP中
响应。

层级：

该媒体要使用的组播层数
流。层被发送到连续的地址，起始
在目标地址处。

模式：

模式参数指示要支持的方法。
本次会话。有效值是PLAY和RECORD。如果不
提供，默认为PLAY。

追加：

如果模式参数包括RECORD，则追加参数
指示媒体数据应附加到现有
资源而不是覆盖它。如果请求追加
并且服务器不支持此操作，它**必须**拒绝
请求而不是覆盖由以下标识的资源
URI。附加参数在模式参数被忽略时无效。
不包含RECORD。

交错：

The interleaved parameter implies mixing the media stream with
控制流在使用的任何协议中
控制流，使用第10.12节中定义的机制。
The argument provides the channel number to be used in the \$
声明。此参数可以指定为范围，例如，
交错=4-5 在运输选择的情况
媒体流需要它。

交错参数表示将媒体流与

翻译文本：参数提供了要使用的通道号在 \$

这允许RTP/RTCP的处理方式与它本身的方式相似
完成UDP，即一个通道用于RTP，另一个通道用于RTCP。

多播特定：**ttl:**

多播生存时间

RTP 特定：**端口：**

此参数提供多播的RTP/RTCP端口对
会话。它被指定为一个范围，例如，端口=3456-3457。

客户端端口：

此参数提供了单播RTP/RTCP端口号对
客户端已选择接收媒体数据 and 控制的哪个部分
信息。它被指定为一个范围，例如，
客户端端口=3456-3457。

server_port:

This parameter provides the unicast RTP/RTCP port pair on which the server has chosen to receive media data and control information. It is specified as a range, e.g., server_port=3456-3457.

ssrc:

The ssrc parameter indicates the RTP SSRC [24, Sec. 3] value that should be (request) or will be (response) used by the media server. This parameter is only valid for unicast transmission. It identifies the synchronization source to be associated with the media stream.

```

Transport          =  "Transport" ":"
                    1\#transport-spec
transport-spec     =  transport-protocol/profile[/lower-transport]
                    *parameter
transport-protocol =  "RTP"
profile            =  "AVP"
lower-transport    =  "TCP" | "UDP"
parameter          =  ( "unicast" | "multicast" )
                    ";" "destination" [ "=" address ]
                    ";" "interleaved" "=" channel [ "-" channel ]
                    ";" "append"
                    ";" "ttl" "=" ttl
                    ";" "layers" "=" 1*DIGIT
                    ";" "port" "=" port [ "-" port ]
                    ";" "client_port" "=" port [ "-" port ]
                    ";" "server_port" "=" port [ "-" port ]
                    ";" "ssrc" "=" ssrc
                    ";" "mode" = "<"> 1\#mode "<">
ttl                =  1*3(DIGIT)
port               =  1*5(DIGIT)
ssrc               =  8*8(HEX)
channel            =  1*3(DIGIT)
address            =  host
mode               =  "<"> *Method "<"> | Method

```

示例： Translated Text： 示例：

交通: RTP/AVP;多播;ttl=127;模式="播放",
RTP/AVP;单播;客户端端口=3456-3457;模式="播放"

传输头仅限于描述单个RTP
流。(RTSP也可以将多个流作为单个
实体。)使其成为RTSP的一部分，而不是依赖于众多
会话描述格式极大地简化了设计
防火墙

12.40 不支持

The Unsupported response header lists the features not supported by 不支持响应头列出了不支持的功能服务器。在通过特征指定的情况下
代理需求字段（第12.32节），如果路径上有代理
客户端与服务器之间，代理必须插入一条消息
回复错误信息“551 选项不受支持”。

查看第12.32节以获取使用示例。

12.41 用户代理

参见 [H14.42]

12.42 变化

参见 [H14.43]

12.43 通过

参见 [H14.44]。

12.44 WWW-Authenticat

参见 [H14.46]。

13 缓存

在HTTP中，响应-请求对被缓存。RTSP不同
显著在这方面。响应不可缓存，与{v*}
异常情况，由DESCRIBE返回的表示描述中的
包含在ANNOUNCE中。（除非是关于{v*}的回复，否则没有回应）
DESCRIBE 和 GET_PARAMETER 不返回任何数据，缓存不被
实际上对这些请求来说是个问题。）然而，对于{v*}来说，这是所希望的。
连续介质数据，通常以带外方式提供
到RTSP，需要缓存，以及会话描述。

在收到一个SETUP或PLAY请求后，代理确定是否
具有最新版本的连续媒体内容及其
描述。它可以确定副本是否是最新的。
发出SETUP或DESCRIBE请求，分别，并比较
最后修改头与缓存副本的相同。如果副本不是
最新版本中，它根据需要修改SETUP传输参数
将请求转发到原始服务器。后续控制
指令如PLAY或PAUSE随后将代理未经修改地传递。
代理将连续媒体数据发送到客户端，同时
可能为以后重用创建本地副本。确切的行为
允许缓存的由缓存响应指令给出

在第12.8节中描述。缓存必须回答任何DESCRIBE请求如果它目前正在向请求者提供流，则它是可能流描述的低级细节可能已经在原始服务器上更改。

请注意，与HTTP缓存不同，RTSP缓存是“切割-通过”多样性。而不是从整个资源中检索原始服务器，缓存简单地复制流数据，就像它经过其前往客户端的途中。因此，它不会引入额外延迟。

客户端看来，RTSP代理缓存就像一个常规媒体服务器，对媒体源服务器像客户端一样。就像一个HTTP缓存必须存储内容类型、内容语言等。它缓存的对象，媒体缓存必须存储演示描述。通常，缓存消除所有传输引用（即，多播信息）从演示描述中，自这些与从缓存到数据传输的数据无关客户端。编码信息保持不变。如果缓存能够翻译缓存的媒体数据，它将创建一个新的演示描述，包含所有编码可能性 {v*} 可以提供。

14 示例

以下示例涉及流描述格式，这些格式是非标准，例如RTSL。以下示例不应被翻译。用作这些格式的参考。

14.1 媒体点播（单播）

客户端C从媒体服务器A（audio.example.com）请求电影并且V（video.example.com）。媒体描述存储在网页上服务器W。媒体描述包含对{v*}的描述演示及其所有流，包括解码器可用、动态RTP有效载荷类型、协议栈和内容信息，例如语言或版权限制。它也可能提供关于电影时间线的指示。

在这个例子中，客户端只对{v*}的最后部分感兴趣电影。

C->W: GET /twister.sdp HTTP/1.1
主机: www.example.com
接受: application/sdp

W->C: HTTP/1.0 200 正常
内容类型: application/sdp

v=0
o=- 2890844526 2890842807 IN IP4 192.16.24.202
s=RTSP 会话
m=音频 0 RTP/AVP 0
a=控制：rtsp://audio.example.com/twister/audio.en
m=视频 0 RTP/AVP 31
a=控制：rtsp://video.example.com/twister/video

C->A: 设置 rtsp://audio.example.com/twister/audio.en RTSP/1.0
CSeq: 1
传输：RTP/AVP/UDP；单播；客户端端口=3056-3057

A->C: RTSP/1.0 200 OK A->C：RTSP/1.0 200 OK
CSeq: 1
会话：12345678
传输：RTP/AVP/UDP；单播；客户端端口=3056-3057；
服务器端口号=5000-5001

C->V: 设置 rtsp://video.example.com/twister/video RTSP/1.0
CSeq: 1
传输：RTP/AVP/UDP；单播；客户端端口=3058-3059

V->C: RTSP/1.0 200 OK V->C: RTSP/1.0 200 正常
CSeq: 1
会话：23456789
传输：RTP/AVP/UDP；单播；客户端端口=3058-3059；
服务器端口号=5002-5003

C->V: 播放 rtsp://video.example.com/twister/video RTSP/1.0
CSeq: 2 翻译文本：CSeq: 2
会话：23456789
范围：smpte=0:10:00-

V->C: RTSP/1.0 200 OK
CSeq: 2 翻译文本：CSeq: 2
会话：23456789
范围：smpte=0:10:00-0:20:00
RTP-Info: url=rtsp://video.example.com/twister/video;
seq=12312232;rtptime=78712811

C->A: 播放 rtsp://audio.example.com/twister/audio.en RTSP/1.0
CSeq: 2 翻译文本：CSeq: 2
会话：12345678
范围：smpte=0:10:00-

A->C: RTSP/1.0 200 OK A->C：RTSP/1.0 200 OK
CSeq: 2 翻译文本：CSeq: 2
会话：12345678

范围：smpte=0:10:00-0:20:00

RTP-Info: url=rtsp://audio.example.com/twister/audio.en;
seq=876655;rtptime=1032181

C->A: 拆卸 rtsp://audio.example.com/twister/audio.en RTSP/1.0
CSeq: 3
会话：12345678

A->C: RTSP/1.0 200 OK
CSeq: 3

C->V: 拆卸 rtsp://video.example.com/twister/video RTSP/1.0
CSeq: 3
会话：23456789

V->C: RTSP/1.0 200 OK
CSeq: 3

尽管音频和视频轨道位于两个不同的服务器上，并且可能开始时间略有不同，并且可能随时间漂移彼此之间，客户端可以使用标准RTP同步这两个{v*}方法，特别是包含在RTCP发送器中的时间尺度报告。

14.2 容器文件的流式传输

为了本例的目的，容器文件是一个存储实体，在哪些属于同一最终用户的多个连续媒体类型展示存在。实际上，容器文件代表了一个RTSP 演示，其中每个组件都是 RTSP 流。容器文件是广泛用于存储此类演示的常用手段。虽然组件作为独立流运输，但是它希望为这些流保持一个共同上下文服务器端。

这使服务器能够保持单个存储句柄打开轻松。它还允许在所有流的情况下平等对待所有流。服务器对流的任何优先级排序。

也可能演示文稿作者希望防止选择性检索流以保留艺术效果的综合媒体展示。同样，在如此紧密的结合呈现，希望能够通过单个控制消息控制所有流聚合URL。

以下是一个使用单个RTSP会话进行控制的示例多流。它还说明了使用聚合URL的应用。

客户端C从媒体服务器M请求一个演示文稿。电影是存储在容器文件中。客户端已获取一个RTSP URL以容器文件。

C->M: 描述 rtsp://foo/twister RTSP/1.0
CSeq: 1

M->C: RTSP/1.0 200 OK M->C: RTSP/1.0 200 正确
CSeq: 1
内容类型: application/sdp
内容长度: 164

v=0
o=- 2890844256 2890842807 IN IP4 172.16.2.93 Translated Text: o=- 2890844256 2890842807 IN IP4 172.16.2.
s=RTSP 会话
i=RTSP会话使用示例
a=控制: rtsp://foo/twister
t=0 0
m=音频 0 RTP/AVP 0
a=控制: rtsp://foo/twister/audio
m=视频 0 RTP/AVP 26
a=控制: rtsp://foo/twister/video

C->M: 设置 rtsp://foo/twister/audio RTSP/1.0
CSeq: 2 翻译文本: CSeq: 2
传输: RTP/AVP; 单播; 客户端端口=8000-8001

M->C: RTSP/1.0 200 OK M->C: RTSP/1.0 200 正常
CSeq: 2 翻译文本: CSeq: 2
传输: RTP/AVP; 单播; 客户端端口=8000-8001;
服务器端口号=9000-9001
会话: 12345678

C->M: 设置 rtsp://foo/twister/video RTSP/1.0
CSeq: 3 翻译文本: CSeq: 3
传输: RTP/AVP; 单播; 客户端端口=8002-8003
会话: 12345678

M->C: RTSP/1.0 200 OK M->C: RTSP/1.0 200 正确
CSeq: 3 翻译文本: CSeq: 3
传输: RTP/AVP; 单播; 客户端端口=8002-8003;
服务器端口号=9004-9005
会话: 12345678

C->M: 播放 rtsp://foo/twister RTSP/1.0
CSeq: 4 翻译文本: CSeq: 4
范围: npt=0-
会话: 12345678

M->C: RTSP/1.0 200 OK
CSeq: 4 翻译文本 : CSeq: 4
会话 : 12345678
RTP-Info: url=rtsp://foo/twister/video;
seq=9810092;rtptime=3450012

C->M: 暂停 rtsp://foo/twister/video RTSP/1.0
CSeq: 5
会话 : 12345678

M->C: RTSP/1.0 460 仅允许聚合操作
CSeq: 5

C->M: 暂停 rtsp://foo/twister RTSP/1.0
CSeq: 6
会话 : 12345678

M->C: RTSP/1.0 200 OK
CSeq: 6
会话 : 12345678

C->M: 设置 rtsp://foo/twister RTSP/1.0
CSeq: 7
传输 : RTP/AVP ; 单播 ; 客户端端口=10000

M->C: RTSP/1.0 459 聚合操作不允许
CSeq: 7

在第一次失败实例中，客户端尝试暂停一个流（在这种情况下为视频）的演示。这是不允许的，对于服务器提供的那个演示。在第二个实例中，聚合URL可能不能用于SETUP，并且有一条控制消息所需每个流以设置传输参数。

这保持了传输头部的语法简单，并允许简单解析防火墙中的传输信息。

14.3 单流容器文件

某些RTSP服务器可能将所有文件都视为“容器文件”，但其他服务器可能不支持这样的概念。因此，客户端应使用会话中规定的规则描述用于请求URL，而不是假设一致性URL始终可以使用。以下是一个多-的示例。流服务器可能期望提供单个流文件：

Ac cept: 应用/x-rtsp-mh, 应用/s dp

CSeq: 1

S->C RTSP/1.0 200 OK S->C RTSP/1.0 200 正好

CSeq: 1

基于内容：rtsp://foo.com/test.wav/

内容类型：application/sdp

内容长度：48

v=0

o=- 872653257 872653257 IN IP4 172.16.2.187

s=mu-law 波形文件

i=音频测试

t=0 0

m=音频 0 RTP/AVP 0

a=控制:流ID=0

C->S 设置 up rtsp://foo.com/test.wav/streamid=0 RTSP/1.0

运输: RTP/AVP/UDP;单播;

客户端端口=6970-6971 ; 模式=播放

CSeq: 2

S->C RTSP/1.0 200 OK S->C RTSP/1.0 200 正常

传输：RTP/AVP/UDP；单播；客户端端口{v*}6970-6971；

服务器端口号{v*}6970-6971；模式{v*}play

CSeq: 2

会话：2034820394

C->S PLAY rtsp://foo.com/test.wav RTSP/1.0 Translated Text: C->S 播放 rtsp://foo.com/test.wav RTSP/1.0

CSeq: 3 翻译文本：CSeq: 3

会话：2034820394

S->C RTSP/1.0 200 OK S->C RTSP/1.0 200 正常

CSeq: 3

会话：2034820394

RTP-Info: url=rtsp://foo.com/test.wav/streamid=0;

seq=981888;rtptime=3781123

注意SETUP命令中的不同URL，然后切换回
到PLAY命令中的聚合URL。这完全合理
当存在多个具有聚合控制的流时，但较少
比在流数量为特殊情况的直观性更强
一个。

在这个特殊情况下，建议服务器对{v*}宽容。
实现发送：

C->S PLAY rtsp://foo.com/test.wav/streamid=0 RTSP/1.0 Translated Text: C->S 播放 rtsp://foo.com/test.wav/streamid=0

CSeq: 3 翻译文本：CSeq: 3

在最坏的情况下，服务器应发送回：

S->C RTSP/1.0 460 仅允许聚合操作
CSeq: 3 翻译文本：CSeq: 3

一个人也希望服务器实现也能对{v*}宽容
以下：

C->S 设置 rtsp://foo.com/test.wav RTSP/1.0
运输: rtp/avp/udp;客户端端口=6970-6971;模式=播放
CSeq: 2 翻译文本：CSeq: 2

由于此文件中只有一个流，因此不会产生歧义
这意味着什么。

14.4 使用多播的实时媒体演示

媒体服务器M选择多播地址和端口。这里，我们
假设该Web服务器仅包含指向完整的指针
描述，而媒体服务器M维护完整的描述。

C->W: GET /concert.sdp HTTP/1.1
主机: www.example.com

W->C: HTTP/1.1 200 正常
内容类型: application/x-rtsp

<会话>
<track src="rtsp://live.example.com/concert/audio">
</会话>

C->M: 描述 rtsp://live.example.com/concert/audio RTSP/1.0
CSeq: 1

M->C: RTSP/1.0 200 OK
CSeq: 1
内容类型: application/sdp
内容长度：44

v=0
o=- 2890844526 2890842807 IN IP4 192.16.24.202
s=RTSP 会话
m=音频 3456 RTP/AVP 0
a=控制：rtsp://live.example.com/concert/audio
c=IN IP4 224.2.0.1/16

C->M: 设置 rtsp://live.example.com/concert/audio RTSP/1.0
CSeq: 2 翻译文本：CSeq: 2

传输：RTP/AVP；多播

M->C: RTSP/1.0 200 OK

CSeq: 2

传输：RTP/AVP；多播；目标=224.2.0.1；

端口=3456-3457；TTL=16

会话：0456804596

C->M: 播放 rtsp://live.example.com/concert/audio RTSP/1.0

CSeq: 3

会话：0456804596

M->C: RTSP/1.0 200 OK

CSeq: 3

会话：0456804596

14.5 将媒体播放到现有会话中

一个会议参与者C希望媒体服务器M播放
将一个演示磁带放入现有会议中。C表示媒体
服务器上网络地址和加密密钥已经
由会议提供，因此不应由服务器选择。
示例省略了简单的ACK响应。

C->M: 描述 rtsp://server.example.com/demo/548/sound RTSP/1.0

CSeq: 1

接受: application/sdp

M->C: RTSP/1.0 200 1 好的

内容类型: application/sdp

内容长度：44

v=0

o=- 2890844526 2890842807 IN IP4 192.16.24.202

s=RTSP 会话

i=参见上文

t=0 0

m=音频 0 RTP/AVP 0

C->M: 设置 up rtsp://server.example.com/demo/548/sound RTSP/1.0

CSeq: 2

传输：RTP/AVP；多播；目标=225.219.201.15；

端口=7000-7001；TTL=127

会议：199702170042.SAA08642@obiwan.arl.wustl.edu%20Starr

M->C: RTSP/1.0 200 OK

CSeq: 2

传输：RTP/AVP；多播；目标=225.219.201.15；

端口=7000-7001 ; TTL=127
会话 : 91389234234
会议 : 199702170042.SAA08642@obiwan.arl.wustl.edu%20Starr

C->M: 播放 rtsp://server.example.com/demo/548/sound RTSP/1.0
CSeq: 3 翻译文本 : CSeq: 3
会话 : 91389234234

M->C: RTSP/1.0 200 OK
CSeq: 3 翻译文本 : CSeq: 3

14.6 记录

会议参与者客户端C请求媒体服务器M进行录制
会议的音频和视频部分。客户使用
公告方法提供关于记录的元信息
会话到服务器。

C->M: 公告 rtsp://server.example.com/meeting RTSP/1.0
CSeq: 90
内容类型: application/sdp
内容长度 : 121

v=0
o=相机1 3080117314 3080118787 IN IP4 195.27.192.36
s=IETF 会议, 慕尼黑 - 1
i=第三十九次IETF会议将在德国慕尼黑举行
u=http://www.ietf.org/meetings/Munich.html
e=IETF信道1 <ietf39-mbone@uni-koeln.de>
p=IETF信道1 +49-172-2312 451
c=IN IP4 224.0.1.11/127
t=3080271600 3080703600
a=工具 : sdr v2.4a6
a=类型 : 测试
m=音频 21010 RTP/AVP 5
c=IN IP4 224.0.1.11/127
a=时间 : 40
m=视频 61010 RTP/AVP 31
c=IN IP4 224.0.1.12/127

M->C: RTSP/1.0 200 OK M->C: RTSP/1.0 200 正常
CSeq: 90

C->M: 设置 rtsp://server.example.com/meeting/audiotrack RTSP/1.0
CSeq: 91
传输 : RTP/AVP ; 多播 ; 目标=224.0.1.11 ;
端口=21010-21011 ; 模式=记录 ; TTL=127

M->C: RTSP/1.0 200 OK

CSeq: 91

会话: 50887676

传输: RTP/AVP; 多播; 目标=224.0.1.11;

端口=21010-21011; 模式=记录; TTL=127

C->M: 设置 rtsp://server.example.com/meeting/videotrack RTSP/1.0

CSeq: 92

会话: 50887676

传输: RTP/AVP; 多播; 目标=224.0.1.12;

端口=61010-61011; 模式=记录; TTL=127

M->C: RTSP/1.0 200 OK M->C: RTSP/1.0 200 正常

CSeq: 92

传输: RTP/AVP; 多播; 目标=224.0.1.12;

端口=61010-61011; 模式=记录; TTL=127

C->M: 记录 rtsp://server.example.com/meeting RTSP/1.0

CSeq: 93

会话: 50887676

范围: 时钟=19961110T1925-19961110T2015

M->C: RTSP/1.0 200 OK

CSeq: 93

15 语法

RTSP语法用增强的巴科斯-诺尔范式 (BNF) 描述在RFC 2068 [2]中所述。

15.1 基础语法

OCTET	=	<any 8-bit sequence of data>
CHAR	=	<any US-ASCII character (octets 0 - 127)>
UPALPHA	=	<any US-ASCII uppercase letter "A".."Z">
LOALPHA	=	<any US-ASCII lowercase letter "a".."z">
ALPHA	=	UPALPHA LOALPHA
DIGIT	=	<any US-ASCII digit "0".."9">
CTL	=	<any US-ASCII control character (octets 0 - 31) and DEL (127)>
CR	=	<US-ASCII CR, carriage return (13)>
LF	=	<US-ASCII LF, linefeed (10)>
SP	=	<US-ASCII SP, space (32)>
HT	=	<US-ASCII HT, horizontal-tab (9)>
<">	=	<US-ASCII double-quote mark (34)>
CRLF	=	CR LF


```

LWS          =      [CRLF] 1*( SP | HT )
TEXT         =      <any OCTET except CTLs>
tspecials    =      "( \" | \" ) \" | \"<\" | \">\" | \"@\"
                  |      \",\" | \";\" | \":\" | \"\\\" | \"<\">
                  |      \"/\", \"[\" | \"]\" | \"?\" | \"=\"
                  |      \"{\" | \"}\" | SP | HT

token        =      1*<any CHAR except CTLs or tspecials>
quoted-string =      ( <\"> *(qdtext) <\"> )
qdtext       =      <any TEXT except <\">>
quoted-pair   =      \"\\\" CHAR

message-header =      field-name ":" [ field-value ] CRLF
field-name    =      token
field-value   =      *( field-content | LWS )
field-content =      <the OCTETs making up the field-value and
                    consisting of either *TEXT or
                    combinations of token, tspecials, and
                    quoted-string>

safe          =      "\\$" | "-" | "_" | "." | "+"
extra         =      "!" | "*" | "$' $" | "(" | ")" | ", "

hex           =      DIGIT | "A" | "B" | "C" | "D" | "E" | "F" |
                    "a" | "b" | "c" | "d" | "e" | "f"
escape        =      "\\%" hex hex
reserved      =      ";" | "/" | "?" | ":" | "@" | "&" | "="

unreserved    =      alpha | digit | safe | extra
xchar         =      unreserved | reserved | escape

```

16 安全注意事项

由于RTSP服务器在语法和用法上的相似性
并且 HTTP 服务器，[H15] 中概述的安全考虑
应用。具体来说，请注意以下内容：

身份验证机制：

RTSP和HTTP共享相同的认证方案，因此
应遵循相同的处方
身份验证。参见[H15.1]关于客户端身份验证问题，
关于支持多重的[问题15.2]
身份验证机制。

滥用服务器日志信息：

RTSP和HTTP服务器可能具有类似的日志记录
机制，因此应该同样受到保护
这些日志的内容，从而保护了隐私

服务器用户。参见[H15.3]关于HTTP服务器关于服务器日志的建议。

敏感信息传输：

没有理由相信通过 {v*} 传输的信息
RTSP可能比通常传输的灵敏度低
通过HTTP。因此，所有关于{v*}的预防措施
数据隐私保护和用户隐私适用
实现RTSP客户端、服务器和代理的开发者。查看
[H15.4] 详细信息请参阅。

基于文件和路径名称的攻击：

尽管RTSP URL是不透明的句柄，它们不一定
具有文件系统语义，预计会有很多
实现将翻译请求URL的部分
直接到文件系统调用。在这种情况下，文件系统
应遵循[H15.5]中概述的预防措施，例如
检查路径组件中的 ".."。

个人信息：

实时流媒体协议（RTSP）客户端通常能够获取与HTTP相同的信息
客户端是（用户名、位置等），因此应该是
同等。参见[H15.6]获取更多建议。

隐私问题与Accept头部相关：

自同年5月起，RTSP中存在的“Accept”报头与之前相同
HTTP，与[H15.7]中概述的相同注意事项相关
它们的使用应遵循。

DNS欺骗：

大概，考虑到通常的连接时间较长
与HTTP会话相关的RTSP会话，RTSP
客户端DNS优化应较少出现。
尽管如此，[H15.8]中提供的建议仍然
与任何试图依赖 {v*} 的实现相关
DNS到IP映射在单次使用映射之后仍保持有效。

位置头和欺骗：

如果单个服务器支持多个不
相互信任，然后必须检查位置值
响应中生成的 Content-Location 标头
在所述组织的控制下确保它们执行
不尝试使它们无权无效化资源
权限。 ([H15.9])

除了当前HTTP规范中的建议外
(RFC 2068 [2]，截至本文撰写时)，未来的HTTP规范可能
提供关于安全问题的额外指导。

以下是对RTSP实现增加的考虑因素。

集中拒绝服务攻击：

协议提供了远程控制的机会
拒绝服务攻击。攻击者可能发起流量
通过指定它们作为以下内容流向一个或多个IP地址：
目标在SETUP请求中。当攻击者的IP地址
在这种情况下可能已知，但这并不总是有用的
防止更多攻击或确定攻击者
身份。因此，一个RTSP服务器应该只允许客户端-
指定的RTSP初始化流量流的目标地址
服务器已验证客户端的身份，无论是与以下之一进行比对
数据库中已知使用RTSP认证机制的用户
(最好使用摘要认证或更强的认证)，或其它
安全手段。

会话劫持：

由于传输层之间没有关系
连接和一个RTSP会话，恶意用户可以进行
客户端以随机会话标识符发出请求，其中
会影响毫无戒备的客户。服务器应使用
大、随机且非顺序的会话标识符
最小化这种攻击的可能性。

身份验证：

服务器应实现基本和摘要[8]
身份验证。在需要更严格安全性的环境中
控制消息，RTSP控制流可能为
加密。

流问题：

RTSP仅提供流控制。流传输问题
本节未涉及，本备忘录的其余部分也未涉及。
实时流传输协议实现很可能依赖于其他协议
例如 RTP、IP 多播、RSVP 和 IGMP，并应解决
安全考虑在那些和其他中提出
适用规格。

持续可疑行为：

RTSP服务器应在{v*}返回错误代码403（禁止）
接收一个被认为的单一实例的行为
安全风险。RTSP服务器也应意识到尝试
探测服务器以寻找弱点和入口点，并且可能
任意断开连接并忽略后续请求的客户端
被认为违反当地安全策略的。

附录A：RTSP协议状态机

RTSP客户端和服务端状态机描述了{v*}的行为
RTSP会话初始化至RTSP会话的协议
终止。

状态是按对象定义的。一个对象是唯一的
通过流URL和RTSP会话标识符识别。任何
请求/回复使用表示RTSP演示的聚合URL
由多个流组成将对个体产生影响
所有流的状态。例如，如果演示/电影
包含两个流，/movie/audio 和 /movie/video，然后
以下命令：

```
播放 rtsp://foo.com/movie RTSP/1.0
CSeq: 559
会话：12345678
```

将对电影/音频和电影/视频的状态产生影响。

这个例子并不表示表示流的标准方式
URLs 或文件系统中的关系。参见第3.2节。

请求 OPTIONS、ANNOUNCE、DESCRIBE、GET_PARAMETER、
SET_PARAMETER 对客户端或服务端状态没有影响，并且
因此未列在状态表中。

A.1 客户端状态机

客户端可以假定以下状态

S:

中文翻译：

初始化：

SETUP 已发送，等待回复。

准备：

SETUP 响应已接收或 PAUSE 响应已接收，此时处于播放状态。

播放：

播放回复已接收

录制：

记录已接收回复

通常，客户端在收到回复时改变状态
请提供需要翻译的英文文本，以便我进行翻译。
位置（例如，PAUSE），状态也会相应改变。如果没有
显式设置对象（例如，它是必需的）

通过多播组可用)，状态从Ready开始。在此案例中，只有两种状态，准备状态和播放状态。客户端也有当{v*}结束时，状态从播放/录制变为就绪请求的范围已达到。

"下一个状态"列表示接收后假定的状态成功响应(2xx)。如果一个请求返回3xx状态码，则状态变为Init，状态码为4xx不会产生变化状态。对于每个状态未列出的消息不得发出客户端处于该状态，除了不影响的消息之外状态，如上所述。从服务器接收到REDIRECT重定向相当于从服务器接收到3xx重定向状态。

state	message sent	next state after response
Init	SETUP	Ready
	TEARDOWN	Init
Ready	PLAY	Playing
	RECORD	Recording
	TEARDOWN	Init
Playing	SETUP	Ready
	PAUSE	Ready
	TEARDOWN	Init
	PLAY	Playing
Recording	SETUP	Playing (changed transport)
	PAUSE	Ready
	TEARDOWN	Init
	RECORD	Recording
	SETUP	Recording (changed transport)

A.2 服务器状态机

服务器可以处于以下状态：

初始化：

初始状态，尚未收到有效的SETUP。

准备：

最后接收到的SETUP设置成功，已发送回复或之后播放，最后收到的PAUSE成功，回复已发送。

播放：

最后收到的PLAY成功，已发送回复。数据正在处理中。已发送。

录制：

服务器正在记录媒体数据。

通常，服务器在接收到请求时改变状态。如果服务器处于播放或录制状态，且在单播模式下，它可能恢复到Init并拆除RTSP会话，如果它尚未收到健康信息，例如RTCP报告或RTSP命令，来自客户端用于定义的间隔，默认为1分钟。服务器可以在会话响应中声明另一个超时值标题（第12.37节）。如果服务器处于就绪状态，它可以在此间隔内未收到RTSP请求，则恢复到Init状态超过一分钟。注意，某些请求（如PAUSE）可能在未来的时间或位置上有效，以及服务器状态变化在适当的时候。服务器从播放状态回退到{v*}。录制到请求范围的末尾状态为“Ready”客户端

REDIRECT消息发送后立即生效，除非它具有指定重定向何时生效的Range头。在此一个案例，服务器状态也会在适当的时候改变。

如果对象不需要显式的SETUP，状态从{v*}开始准备就绪，并且只有两种状态，准备就绪和正在播放。

"下一个状态"列表示发送后所假设的状态。成功响应（2xx）。如果一个请求导致状态码为3xx，状态变为Init。状态码为4xx不会引起任何变化。

state	message received	next state
Init	SETUP	Ready
	TEARDOWN	Init
Ready	PLAY	Playing
	SETUP	Ready
	TEARDOWN	Init
	RECORD	Recording
Playing	PLAY	Playing
	PAUSE	Ready
	TEARDOWN	Init
	SETUP	Playing
Recording	RECORD	Recording
	PAUSE	Ready
	TEARDOWN	Init
	SETUP	Recording

附录B：与RTP的交互

RTSP允许媒体客户端控制选定的、非连续的媒体展示的部分，使用RTP渲染这些流媒体层[24]。渲染RTP流的媒体层不应受NPT跳跃的影响。因此，RTP序列号和RTP时间戳必须在NPT跳跃中连续且单调。

例如，假设时钟频率为8000 Hz，数据包化100 毫秒的间隔以及一个初始序列号和时间戳零。首先我们玩NPT 10到15，然后跳过并玩NPT 18至20。第一段以RTP数据包的形式呈现。序列号0至49和时间戳0至39,200。第二段由序列号为50的RTP数据包组成通过69，时间戳为40,000至55,200。

我们不能假设RTSP客户端可以与RTP通信媒体代理，因为这两个可能是不相关的进程。如果RTP时间戳显示与NPT相同的间隔，媒体代理将假设演示中存在停顿。如果跳跃在NPT足够大，RTP时间戳可能会溢出，媒体代理可能后来认为后续的包是先前包的重复演绎完毕。

对于某些数据类型，RTSP层与{v*}之间的紧密集成RTP层将是必要的。这绝不排除以上限制。结合RTSP/RTP媒体客户端应使用RTP-Info字段用于确定传入的RTP数据包是否被发送在查找之前或之后。

对于连续音频，服务器应将RTP标记位设置为开始处理新的PLAY请求。这允许客户端执行播放延迟自适应。

对于缩放（见第12.34节），RTP时间戳应与{v*}相对应播放时间。例如，当播放30fps录制的视频时帧/秒，缩放比例为二，速度（第12.35节）为一，的服务器会丢弃每秒第二帧以维护和传输视频数据包具有每帧3,000个正常时间戳间隔，但NPT每帧视频会增加1/15秒。

客户可以通过注意RTP来维持NPT的正确显示时间戳值，为重新定位后到达的第一个数据包。序列参数的RTP-Info（第12.33节）报头提供下一个段落的第一个序列号。

附录C：使用SDP进行RTSP会话描述

会话描述协议（SDP，RFC 2327 [6]）可用于描述RTSP中的流或演示。此类用法仅限于指定访问方式和编码（s）用于：

聚合控制：

一个由一个或多个服务器流组成的演示文稿
这些不可用于总体控制。此类
描述通常通过HTTP或其他非RTSP协议检索
意味着。然而，它们可能通过公告方法接收。

非聚合控制：

一个由单个来源的多个流组成的演示文稿
服务器可用于聚合控制。此类
描述通常是在响应一个DESCRIBE时返回的
在URL上的请求，或通过ANNOUNCE方法接收。

此附录描述了如何检索SDP文件，例如，
通过HTTP确定RTSP会话的操作。它还
描述客户端应如何解释回复中返回的SDP内容
对一个DESCRIBE请求。SDP不提供任何机制，以便客户端
可以区分，无需人工指导，几种媒体
同时渲染的流和一组替代方案
(例如，两种不同语言的音频流)。

C.1 定义

"会话级"、"媒体级"和其他关键/属性术语
本附录中使用的名称和值应按定义使用
SDP (RFC 2327 [6]):

C.1.1 控制URL

"控制URL的“a=控制:”属性用于传达控制URL。”
属性用于会话和媒体描述。如果
用于单个媒体，它表示要使用的URL
控制该特定媒体流。如果在会话中找到
层级，该属性表示聚合控制的URL。

示例： Translated Text： 示例：
a=控制： rtsp://example.com/foo

此属性可能包含相对和绝对URL。
遵循RFC 1808 [25]中规定的规则和惯例。
实现应按以下顺序查找基本URL：

1. RTSP内容基础字段
2. RTSP内容位置字段
3. RTSP请求URL

如果此属性仅包含一个星号(*)，则URL是视为一个空的嵌入式URL，因此继承整个基本URL。

C.1.2 媒体流

"m="字段用于枚举流。预期所有指定的流将以适当的方式渲染同步。如果会话是单播，端口号作为服务器向客户端的推荐；客户端仍然将其包含在其SETUP请求中，可能忽略此内容推荐。如果服务器没有偏好，它应该设置端口号值设为零。

示例： Translated Text： 示例：
m=音频 0 RTP/AVP 31

C.1.3 载荷类型 (s)

负载类型在“m=”字段中指定。如果有效载荷类型是来自RFC 1890 [1]的静态有效载荷类型，没有其他信息是必需的。如果它是一种动态有效载荷类型，则媒体属性“rtpmap”用于指定媒体类型。“编码名称”在“rtpmap”属性中可能是以下之一在RFC 1890（第5节和第6节）中指定，或一种实验性编码使用SDP（RFC 2327 [6]）中指定的“X-”前缀。编解码器-特定参数未在此字段中指定，而是在以下描述的“fmp”属性。寻求实现的实现者。注册新的编码应遵循RFC 1890 [1]中的程序。如果媒体类型不适合RTP AV配置文件，那么它就是建议创建一个新的配置文件并设置适当的配置文件名称可用于“m=”字段中代替“RTP/AVP”。

C.1.4 格式特定参数

格式特定的参数使用“fmp”媒体传递属性。“fmp”属性的语法是特定的编码(s)该属性所引用的。注意，数据包化间隔使用“ptime”属性进行传达。

C.1.5 展示范围

"a=range" 属性定义了存储的总时间范围
会话。（直播会话的长度可以从 "t" 中推断出来）

"r" 参数。）除非演示包含 {v*} 媒体流。
不同持续时间，范围属性是会话级别的
属性。单位首先指定，然后是值范围。

The units and their values are as defined in Section 3.5, 3.6 and 3.7. 单位及其值如第3.5节、3.6节所述

示例：

a=范围:npt=0-34.4368

a=范围：时钟=19971113T2115-19971113T2203

C.1.6 可用时间

"t=" 字段必须包含起始和停止的合适值
同时处理聚合和非聚合流控制的计时。与
聚合控制，服务器应指示一个停止时间值
它保证描述有效，并有一个开始时间
等于或早于DESCRIBE请求的时间
已接收。它也可能表示开始和结束时间为0，意味着
会话始终可用。使用非聚合控制，
值应反映会话所对应的确切期间
根据SDP语义可用，且不依赖于其他
表示（例如包含描述的网页的生命）
为此目的。

C.1.7 连接信息

在SDP中，"c=" 字段包含媒体的目标地址
流。然而，对于按需单播流和一些多播
流，目标地址由客户端通过
SETUP 请求。除非媒体内容有固定目的地
地址，"c=" 字段应设置为合适的空值。
地址类型 "IP4"，此值为 "0.0.0.0"。

C.1.8 实体标签

可选的 "a=etag" 属性标识会话的版本
描述。对于客户端来说是透明的。SETUP请求可能包括
此标识符在If-Match字段中（见第12.22节）仅
允许会话建立，如果此属性值仍然对应
与当前描述的相对应。属性值是不透明的
可能包含SDP属性值内允许的任何字符。

示例： Translated Text： 示例：

a=标签：158bb3e7c7fd62ce67f12b533f06b83a

可以认为“o=”字段提供相同的功能。然而，它是以一种可能会使 {v*} 服务器需要支持多个会话的约束描述同一媒体的其他类型，除了SDP内容。

C.2 聚合控制不可用

如果演示文稿不支持聚合控制和多个媒体部分被指定，每个部分必须具有控制URL通过“a=控制：”属性指定。

示例： Translated Text： 示例：

v=0

o=- 2890844256 2890842807 IN IP4 204.34.34.32 Translated Text: o=- 2890844256 2890842807 IN IP4 204.34.34.32

s=我从网页上来的

t=0 0

c=IN IP4 0.0.0.0

m=视频 8002 RTP/AVP 31

a=控制：rtsp://audio.com/movie.aud

m=音频 8004 RTP/AVP 3

a=控制：rtsp://video.com/movie.vid

请注意，描述中控制URL的位置暗示了该客户端建立到各自的RTSP控制会话服务器 audio.com 和 video.com。

建议SDP文件包含完整的媒体初始化信息即使它被发送到媒体客户端通过非RTSP方式。这是必要的，因为没有机制以指示客户端应请求更详细的信息通过DESCRIBE获取媒体流信息。

C.3 可用聚合控制

在这个场景中，服务器有多个流可以被整体进行控制。在这种情况下，既有媒体级别的a=控制：属性，用于指定流URL，并且一个会话级别的“a=控制：”属性，该属性用作请求聚合控制的URL。如果媒体级别的URL是相对的，根据C.1.1节解析为绝对URL以上。

如果演示只包含一个流，媒体级别的a=控制：属性可以完全省略。然而，如果演示文稿包含多个流，每个媒体流部分必须包含其自身的“a=控制”属性。

示例： Translated Text： 示例：

v=0

o=- 2890844256 2890842807 IN IP4 204.34.34.32 Translated Text: o=- 2890844256 2890842807 IN IP4 204.34.34.32

s=包含

i=<更多信息>

t=0 0

c=IN IP4 0.0.0.0

a=控制：rtsp://example.com/movie/

m=视频 8002 RTP/AVP 31

a=控制：trackID=1

m=音频 8004 RTP/AVP 3

a=控制：trackID=2

在这个例子中，客户端需要建立一个单独的RTSP

会话到服务器，并使用URLs

rtsp://example.com/movie/trackID=1 和

rtsp://example.com/movie/trackID=2 设置视频和音频流，分别。URL rtsp://example.com/movie/ 控制着整部电影。

附录D：最小RTSP实现

D.1 客户

客户端实现必须能够执行以下操作：

- * 生成以下请求：SETUP、TEARDOWN以及PLAY之一 (i.e., 一个最小的回放客户端)或RECORD (i.e., 一个最小的录制客户端)。如果实现了RECORD, 则ANNOUNCE必须已实现。

- * 请在请求中包含以下头信息：CSeq、Connection、会话, 传输。如果实现了ANNOUNCE, 则具有{v*}的能力包含头部 Content-Language、Content-Encoding、Content-长度和内容类型也应如此。

解析和理解以下响应中的以下标题：CSeq, 连接、会话、传输、内容语言、内容-编码, 内容长度, 内容类型。如果实现了RECORD, 位置头也必须理解。RTP兼容实现也应实现 RTP-Info。

了解接收到的每个错误码的类别并通知最终用户, 如果存在, 则为4xx类错误代码的5xx. 通知要求可能可以放宽, 如果最终用户明确地不希望它用于一个或所有状态码。

- * 预期并响应来自服务器的异步请求, 例如作为公告。这并不一定意味着它应该实现 ANNOUNCE 方法, 仅此而已, 它必须响应对服务器收到的任何请求进行肯定或否定。

尽管不是必需的, 但在当前时间以下内容强烈推荐关于实际互操作性初始的出版实现和/或成为“良好公民”。

- * 实现RTP/AVP/UDP作为有效的传输方式。

- * 包含用户代理头。

理解附录C中定义的SDP会话描述

- * 接受标准媒体初始化格式 (如SDP) 输入、命令行或其他适当的操作系统方式环境作为其他应用的“辅助应用程序”应用 (例如网页浏览器)。

可能存在与最初不同的RTSP应用程序

由RTSP规范贡献者所构想

上述要求没有意义。因此,

建议以上仅作为指南, 而非严格需求。

D.1.1 基本播放

为了支持媒体流的按需播放，客户端必须另外能够执行以下操作：
生成PAUSE请求；
实现 REDIRECT 方法，以及 Location 头。

D.1.2 允许身份验证

为了访问需要 {v*} 的 RTSP 服务器上的媒体演示身份验证，客户端还必须能够执行以下：
识别401状态码；
解析并包含WWW-Authenticate头；
实现基本身份验证和摘要身份验证。

D.2 服务器

一个最小化的服务器实现必须能够执行以下操作：

- * 实现以下方法：SETUP、TEARDOWN、OPTIONS 和或者 PLAY（用于最小播放服务器）或 RECORD（用于最小记录服务器）。如果实现了RECORD，则ANNOUNCE也应予以实施。
- * 包含以下头信息在响应中：连接，内容长度，内容类型，内容语言，内容编码，交通，公共。包含位置头部的功能应实现如果RECORD方法符合RTP规范的话实现也应实现 RTP-Info 字段。
- * 解析并适当地响应以下标题在请求：连接、会话、传输、需求。

尽管不是必需的，但在当前时间以下内容强烈推荐关于实际互操作性初始的出版实现和/或成为“良好公民”。

- * 实现RTP/AVP/UDP作为有效的传输方式。
- * 服务器头部的包含。
- 实现DESCRIBE方法。
- * 生成附录C中定义的SDP会话描述

可能存在与最初不同的RTSP应用程序由RTSP规范贡献者所构想
上述要求没有意义。因此，
建议以上仅作为指南，而非严格需求。

D.2.1 基本播放

为了支持媒体流的按需播放，服务器必须另外能够执行以下操作：

识别范围标题，如果未进行搜索则返回错误支持。
实现PAUSE方法。

此外，为了支持普遍接受的用户界面特性，以下是在线点播媒体中高度推荐的服务器：

- * 包含并解析范围头，使用NPT单位。
SMPTE 单位的实现建议。
- * 请在媒体中包含媒体演示的长度
初始化信息。
- * 包含从数据特定时间戳到NPT的映射。当RTP
当使用时，RTP-Info字段的rtptime部分可能被用来
将 RTP 时间戳映射到 NPT。

客户端实现可以使用长度信息的存在
确定剪辑是否可寻址，并可见地禁用寻址
功能，对于长度信息不可用的片段。

A common use of the presentation length is to implement a "slider
bar" 既是进度指示器也是时间线
system定位工具。

请注意，由于源文本中包含公式符号 $\{v^*\}$ ，翻译

映射从RTP时间戳到NPT是必要的，以确保正确
滑块位置

D.2.2 允许身份验证

为了正确处理客户端认证，服务器必须另外能够执行以下操作：

- 生成所需的401状态码
资源。
解析并包含WWW-Authenticate头
- * 实现基本身份验证和摘要身份验证

附录E：作者地址

施洛兹林内亨宁
计算机科学系
哥伦比亚大学
1214 阿姆斯特丹大道
纽约, NY 10027
美国

邮箱：schulzrinne@cs.columbia.edu

Anup Rao
Netscape 通信公司
501 E. 中间菲尔德路
山景城, 加利福尼亚州 94043
美国

电子邮件：anup@netscape.com

罗伯特·兰菲尔
RealNetworks
1111 Third Avenue Suite 2900
西雅图, 华盛顿州 98101
美国

robla@real.com

附录 F：致谢

此备忘录基于原始RTSP文档的功能
提交于96年10月。它还借鉴了格式和描述
HTTP/1.1.

此文档极大地受益于所有那些人的评论
参与MMUSIC-WG。除了那些已经
提到，以下个人为此做出了贡献
规范：

拉胡尔·阿加瓦尔，托尔斯坦·布朗，布伦特·布朗宁，布鲁斯·巴特菲尔德，
Steve Casner, Francisco Cortes, Kelly Djahandari, Martin Dunsmuir，
埃里克·弗莱施曼，杰伊·盖根，安迪·格里农，V.古鲁普拉斯德，彼得
海特，马克·汉德利，布拉德·海夫塔-高布，约翰·K·霍，菲利普·霍斯卡，
安妮·琼斯，安德斯·克莱默茨，露丝·兰，斯蒂芬妮·莱夫，乔纳森
Lennox, Eduardo F. Llach, Rob McCool, David Oran, Maria Papadopouli，
Sujal Patel, Ema Patki, Alagu Periyannan, Igor Plotnikov, Pinaki
沙赫，大卫·辛格，杰夫·史密斯，亚历山大·索科爾斯基，戴尔·斯塔门，以及
约翰·弗朗西斯·斯特拉克

参考文献

- 1 Schulzrinne, H., "音频和视频会议的RTP配置文件"
具有最小控制", RFC 1890, 1996年1月。
- 2 Fielding, R. , Gettys, J. , Mogul, J. , Nielsen, H. , 和 T.
伯纳斯-李, 《超文本传输协议 - HTTP/1.1》, RFC
2068年1月。
- 3 Yergeau, F. , Nicol, G. , Adams, G. , 和M. Duerst ,
"超文本标记语言的国际标准化", RFC
2070年1月。
- 4 Bradner, S., "在RFC中使用的关键词以指示 {v*}
需求级别", BCP 14 , RFC 2119 , 1997年3月。
- 5 ISO/IEC, "信息技术 - 通用运动图像符号编码"
图片和相关的音频信息 - 第6部分: 扩展
关于数字存储媒体和控制, " 国际草案 "
标准 ISO 13818-6, 国际标准化组织
标准化 ISO/IEC JTC1/SC29/WG11 , 日内瓦, 瑞士 ,
1995年11月。
- 6 Handley, M. , 和 V. Jacobson , "SDP: Session Description Protocol"
协议", RFC 2327, 1998年4月。
- 7 弗兰克斯, J. , 霍尔姆-贝克, P. , 和J. 霍斯特勒 , "对{v*}的扩展"
HTTP: 摘要访问认证", RFC 2069, 1997年1月。
- 8 Postel, J., "用户数据报协议", STD 6 , RFC 768 , 八月
1980.
- 9 Hinden, B. 和 C. Partridge, "可靠数据版本 2"
协议 (RDP) ", RFC 1151 , 1990年4月。
- 10 Postel, J., "传输控制协议", STD 7, RFC 793 ,
九月1981年。
- 11 H. Schulzrinne, "全面多媒体控制
架构用于互联网, " 在Proc. 国际
数字网络和操作系统支持的研讨会
音频和视频 (NOSSDAV) , (圣路易斯, 密苏里州) , 1997年5月。
- 12 国际电信联盟, "可视电话
本地局域网系统和设备, 提供以下内容:
非保证服务质量,"H.323 建议书,
电信标准化部门, 国际电信联盟, 日内瓦 ,
瑞士, 1996年5月。

- 13 麦克Mahon, P., "SOCKS版本GSS-API认证方法"
5", RFC 1961, 1996年6月。
- 14 J. Miller, P. Resnick 和 D. Singer, "评级服务与"
评分系统 (及其机器可读描述), "
推荐 REC-PICS-services-961031, 万维网联盟 (World Wide Web)
联盟), 马萨诸塞州波士顿, 1996年10月。
- 15 J. Miller, T. Krauskopf, P. Resnick, 和 W. Treese, "PICS
标签分布、标签语法和通信协议,"
推荐 REC-PICS-labels-961031, 万维网联盟 (World Wide Web)
联盟), 马萨诸塞州波士顿, 1996年10月。
- 16 克罗克, D. 和 P. Overell, "增强BNF用于语法"
规格: ABNF", RFC 2234, 1997年11月。
- 17 Braden, B., "互联网主机需求 - 应用和 {v*}
支持", STD 3, RFC 1123, 1989年10月。
- 18 Elz, R., "IPv6地址的紧凑表示", RFC
1924年, 1996年4月。
- 19 Berners-Lee, T., Masinter, L. 和 M. McCahill, "Uniform
资源定位符 (URL), RFC 1738, 1994年12月。
- 20 Yergeau, F., "UTF-8, ISO 10646的转换格式",
RFC 2279, 1998年1月。
- 22 Braden, B., "T/TCP - TCP的会话扩展 {v*}
功能规范", RFC 1644, 1994年7月。
- 22 W. R. Stevens, 《TCP/IP 详解: 实现》, 第2卷。
阅读, 马萨诸塞州: Addison-Wesley, 1994。
- 23 Schulzrinne, H., Casner, S., Frederick, R. 和 V. Jacobson,
"RTP: 实时应用的传输协议", RFC
1889年, 1996年1月。
- 24 Fielding, R., "相对统一资源定位符", RFC 1808,
六月 1995。

完整版权声明

版权 (C) 互联网社会 (1998年)。版权所有 保留。

此文档及其翻译可以复制并提供给
其他人, 以及对其做出评论或以其他方式解释的衍生作品
或协助其实施者可能准备、复制、发布
并且分发, 全部或部分, 不受任何限制
善良, 前提是上述版权声明和本段保持不变
包含在所有此类副本及其衍生作品中。然而, 这
文档本身不得以任何方式修改, 例如通过删除
版权声明或对互联网协会或其他
互联网组织, 除需用于目的之外
在开发互联网标准的情况下, 其程序为
在互联网标准流程中定义的版权必须
随后, 或根据需要将其翻译成除{v*}以外的其他语言
英文。

以上授予的限制权限是永久的, 不会失效。
被互联网协会或其继任者或受让人撤销。

此文档及其包含的信息提供基于
"现状"基础和互联网社会与互联网工程
任务组放弃所有明示或暗示的保证, 包括
但不限于任何关于使用信息的保证
此处不会侵犯任何权利或任何隐含的保证
商誉或特定用途适用性。