

SIP 和 RTSP 协议转换模块的设计与实现^①

白军伟, 孙建伟, 杨海波, 贾军营

(中国科学院 沈阳计算技术研究所, 沈阳 110168)

摘要: 随着流媒体技术与 IP 通信技术的快速发展, SIP 协议与 RTSP 协议的互联互通问题日益凸显. 论文从信令对应、地址映射、交互流程设计等几个方面对 SIP 与 RTSP 协议互通进行了论述, 并在 RTSP 代理服务器的基础上, 设计并完成了 SIP 与 RTSP 协议转换模块, 实现了 SIP 信令与 RTSP 信令的转换功能. 最后, 论文以 SIP Client 访问 RTSP 摄像头获取实时画面为例, 验证了该方法的可行性与正确性, 为基于 SIP 的多媒体调度系统与 RTSP 设备的集成问题提供了较好的解决方案.

关键词: SIP; RTSP; 协议转换; 协议互通

Design and Implementation of the SIP and RTSP Protocol Conversion Module

BAI Jun-Wei, SUN Jian-Wei, YANG Hai-Bo, JIA Jun-Ying

(Shenyang Institute of Computing Technology, Chinese Academy of Science, Shenyang 110168, China)

Abstract: As the rapid development of streaming media technology and IP communication technology, the SIP protocol and RTSP protocol interoperability issue becomes increasingly prominent. This paper discusses the protocol communication problem between SIP and RTSP protocol from the aspects of protocol method corresponding, address mapping, interactive process design and so on. And based on the RTSP proxy server, the SIP and RTSP protocol conversion module is designed and implemented, realized the protocol method conversion between SIP and RTSP. Finally, the paper verifies the feasibility and validity of the method by taking SIP Client access RTSP camera to obtain real-time image as an example, which provides a good solution for the integration between the multimedia scheduling system based on SIP protocol and RTSP devices.

Key words: SIP; RTSP; protocol conversion; protocol interoperability

近年来网络多媒体技术迅速发展, 网络通信协议日益多样化, 协议间的通信愈加困难, 多媒体通信协议的融合迅速成为目前的一个研究热点. 本文基于实验室现有科研成果—IP 调度系统, 提出了融合 RTSP 协议与 SIP 协议的方法, 致力于解决多媒体调度系统中普遍存在的兼容性问题.

1 概述

在目前多种多媒体通信协议混杂的情况下, SIP 协议凭借其强大的技术优势, 迅速脱颖而出. 特别是在多媒体调度应用中, 越来越多的厂商开始提供基于 SIP 的多媒体调度服务, 但缺少对 RTSP 终端的支持,

与现实中已部署的基于 RTSP 协议的设备出现不兼容问题, 难以与现有设施实现无缝集成, 引起资源浪费, 造成较大的经济损失.

因此, 本文在基于 SIP 的多媒体调度系统基础上, 设计并实现了支持 SIP 与 RTSP 协议信令转换的功能模块, 致力于解决 SIP 协议与 RTSP 协议的互通问题, 具有重要的现实意义. 首先, SIP 协议^[1]与 RTSP 协议^[2]均为应用层信令控制协议, 基于文本, 具有类似的会话控制方法(Method), 这为我们的协议互通与融合工作提供了依据; 其次, 在 2011.12.30 发布的 GB/T 28181-2011^[3]中指出, 视频监控互联互通推荐使用 SIP 协议, 而目前视频监控流媒体主要采用 RTSP 协议. 根

^① 收稿时间:2012-10-22;收到修改稿时间:2012-11-23

据目前的发展形势, 尽管 SIP 的优势日益凸显, 但在未来相当长的时间内, 在多媒体调度领域中 RTSP 与 SIP 共存的局面将会持续下去; 再次, 这也是投资保护的客观需要. 由于 RTSP 先于 SIP 出现, RTSP 协议已在实际中得到了广泛应用, 基于 RTSP 的产品设备已被广泛的投入到实际的应用当中. 因此, 新部署的多媒体调度系统应与现有的系统协同工作, 充分合理的利用现有设施提供服务, 以保护对原有系统的投资, 支持系统平滑演进.

2 SIP协议与RTSP协议

2.1 SIP 协议与 RTSP 协议简介

SIP 协议为应用层信令控制协议, 主要用于创建、修改和终止有一个或多个参与者参加的会话, 会话参与者可通过单播、组播进行通信. 它透明的支持名称映射与重定向服务, 支持个人移动业务. 但 SIP 并非是一个垂直集成的通信系统, 称其为一个与其他 IETF 协议一起构建一个完整多媒体框架的部件更为合适. 它的基本功能与操作并不依赖于任何其它协议, 即 SIP 本身不提供服务, 而是提供了一个实现不同服务的基础.

RTSP 协议与 SIP 协议同属应用层信令控制协议, 它提供了一个可扩展的框架, 用于建立并控制一个或多个时间同步的连续媒体流的连接与传输, 使实时数据(如音频与视频)的受控点播成为可能. 数据源可以是现场的实时数据, 也可以是存储的媒体文件. RTSP 对流媒体的播放提供了诸如快进、快退、暂停等播放控制, 但 RTSP 本身通常并不发送媒体数据, 而是充当多媒体服务器的网络远程控制器. 数据传输可以通过传输层的 TCP 协议, 也可通过 UDP 协议.

在多媒体调度领域, 人们早已不满足于传统的单一方式, 而是需要诸如文本、图像、音频、视频等多种媒体信息以超越时空限制的集中方式、作为一个整体呈现在人们眼前^[4]. 在此背景下, SIP 应运而生, 并凭借其更加简单、开放、灵活、可扩展等诸多优点, 迅速成为下一代网络中的核心协议之一^[5], 同时被公认是利用互联网基础设施进行全面集成通讯的绝好方式.

2.2 SIP 与 RTSP 信令方法比较

SIP 协议与 RTSP 协议有很多的共同点, 基于文本, 同属于应用层信令控制协议, 更重要是它们有类似的会话控制方法, 且协议实现通常使用相同的底层协议,

如用于传送实时数据流并提供质量反馈的 RTP/RTCP 协议^[6], 用于描述多媒体会话的 SDP 协议^[7]等.

SIP 信令控制方法主要有性能查询(OPTIONS)、注册(REGISTER)、确认(ACK)、请求(INVITE)、更新(UPDATE)、取消(CANCEL)、会话结束(BYE)等. 解释说明如表 1 所示.

表 1 SIP 基本信令说明

请求消息	消息含义
REGISTER	用于 UAC 在 UAS 上的注册, 完成地址绑定.
INVITE	发起会话请求, 邀请用户加入一个会话, 会话描述含于消息体中.
ACK	证实已收到对 INVITE 的最终响应.
UPDATE	属于 SIP 扩展方法之一, 允许 UAC 更新会话的参数而不改变对话状态.
CANCEL	取消尚未完成的请求, 对于已完成的请求则没有影响.
OPTIONS	查询对端支持的能力集.
BYE	结束会话.

类似的, RTSP 的信令方法有性能查询(OPTIONS)、媒体描述(DESCRIBE)、建立会话(SETUP)、播放(PLAY)、暂停播放(PAUSE)、重定向(REDIRECT)、会话结束(TEARDOWN)等. 解释说明如表 2 所示.

表 2 RTSP 基本信令说明

请求消息	消息含义
OPTIONS	查询对端支持的能力集.
DESCRIBE	获取 URI 指定媒体的描述信息.
SETUP	协商媒体传输方式, 请求服务器分配资源.
PLAY	请求服务器使用由 SETUP 建立的机制开始发送数据.
PAUSE	暂停流媒体的播放, 可携带 Range 头域, 但必须是一个时间点, 即一个精确的值.
REDIRECT	服务器通过该请求告诉客户端应该请求另外一个位置的媒体.
TEARDOWN	结束会话.

3 信令转换模块的设计与实现

3.1 系统概述

本系统主要包括三个模块, 如图 1 所示, SIP Server 作为 SIP 服务器, 主要负责完成与 SIP Client 的交互处理, 具有 SIP Server 的基本功能, 并完成 SIP Client 与 RTSP 终端的注册认证, 根据映射策略, 保存 SIP URI 与 RTSP URI 的映射关系; RTSP Proxy 作为代理, 兼具 Server 与 Client 的功能, 用于管理所代理的 RTSP 终端, 负责完成与 RTSP 终端的交互处理; 信令转换模块则实现了 SIP 信令与 RTSP 信令的转换功能.

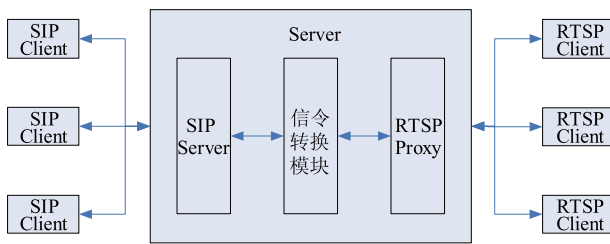


图 1 系统示意图

3.2 信令方法对应

SIP 协议与 RTSP 协议都基于文本, 有类似的一系列信令控制方法, 均依靠信令方法实现对会话的管理与控制。为此, 要实现 SIP 与 RTSP 协议间的通信, 首先应分析其信令方法的对应关系。参考 RFC3261 及 RFC2326, 总结如表 3:

表 3 信令方法对应关系表

SIP 信令方法	RTSP 信令方法	功能说明
OPTIONS	OPTIONS	获取对端支持的性能
OPTIONS	DESCRIBE	获取媒体描述信息
INVITE	SETUP	建立会话
INVITE 或 UPDATE	SETUP	添加一条媒体流到已存在的会话
INVITE 或 UPDATE	PLAY	播放 (开始传送媒体流)
INVITE 或 UPDATE	PAUSE	暂停媒体流发送
BYE	TEARDOWN	结束会话
BYE	REDIRECT	结束会话
REFER	REDIRECT	资源重定向

根据实际需要, RTSP OPTIONS 请求主要用来保持 RTSP Proxy 与 RTSP 媒体设备的活动状态, 以及用于 RTSP 终端与 RTSP 终端交互时获取对端的支持能力集。DESCRIBE 请求则用于与 SIP OPTIONS 信令的对应, 用来获取媒体描述信息。另外, 当 RTSP REDIRECT 的 Location 头部值为空时, 默认处理是终止会话, 此时与 TEARDOWN 具有相同的效果。

SIP 中 INVITE 和 UPDATE 都可用来修改变会话参数^[8]。不同之处在于 UPDATE 不会对 Dialog 的状态产生影响, 而 INVITE 则会改变对话状态。因此 UPDATE 可在第一个 INVITE 被应答之前发送(即收到 INVITE 200 OK 之前), 用来控制早期媒体会话。INVITE 则只能在第一个 INVITE 被应答之后发送(即在通话已经建立之后), 但在通话建立以后多用

INVITE 来改变会话参数, 因为大多数情况下修改会话参数需要用户参与, 比如征求用户是否允许, 而 UPDATE 要立即被响应。

3.3 SIP URI 与 RTSP URI 的映射关系

SIP URI 与 RTSP URI 的映射关系是为实现 SIP Client 与 RTSP 终端的信令转换服务的, 由数据库保存。其中 SIP 设备属性表与 RTSP 设备属性表分开存放, 而 SIP URI 和 RTSP URI 的映射关系则由单独的一张表存储, 以保持两者的相对独立性和较好的扩展性。当主叫方与被叫方属于同一设备类型时, 如均为 SIP Client, 则为正常的 SIP UA 间的通话; 若属于不同类型, 则会根据映射关系进行信令消息的转换。其中, 设备类型标示在终端设备向服务器注册时完成设置。以 SIP Client 作为主叫方为例, 在服务器收到请求消息后, 首先通过数据库中设备类型识别主叫方为 SIP UA, 然后通过消息请求行查询被叫方设备类型属性。若被叫方亦为 SIP UA 时, 则转交由 SIP Server 处理; 若被叫方为 RTSP 终端时, 则交由信令转换模块处理。

3.4 转换模块设计

信令转换模块应具有 RTSP 信令与 SIP 信令的转换功能, 以完成 RTSP 终端与 SIP Client 的双向交互。根据信令的转换方向, 该模块可分为两个部分:

- SIP_RTSP 方向信令转换模块: 完成从 SIP Client 到 RTSP 终端的控制信令转换过程。即接收 SIP Client 发送的 SIP 请求消息, 将其转换为相应的 RTSP 请求消息, 并发送给 RTSP 终端; 接收 RTSP 终端返回的响应消息, 将其转换为对应的 SIP 响应消息, 并发送给 SIP Client;

- RTSP_SIP 方向信令转换模块: 接收 RTSP 终端发送的 RTSP 请求消息, 将其转换为相应的 SIP 请求消息, 并发送给 SIP Client; 接收 SIP Client 返回的响应消息, 将其转换为对应的 RTSP 响应消息, 并发送给 RTSP 终端。

根据现实状况, 多媒体调度系统中要求 SIP Client 访问 RTSP 终端的需求更为普遍, 为此我们以 SIP_RTSP 方法信令转换模块为重点来进行论述, 据此来实现媒体流从 RTSP 终端到 SIP Client 方向的传输。这样不仅符合 RTSP Server 单向媒体流传输的特点, 也能更好的抓住论述的重点。

3.5 SIP Client 访问 RTSP 媒体设备交互流程设计

作为 RTSP Proxy, 支持动态加载被代理者对象, 并完成 RTSP 媒体设备在 SIP Server 上的注册过程。

RTSP Proxy 从 RTSP 媒体设备获取实时流, 由于 SIP Client 均通过 RTSP Proxy 媒体转发得到媒体数据, 因此即使有多个 SIP Client 同时访问同一媒体设备, RTSP Proxy 也只需从该媒体设备读取一遍媒体流, 形成一对多的映射关系, 从而极大地节省了带宽资源.

当第一个 SIP Client A 访问已注册的 RTSP 媒体设备 B 时, 如图 2 所示, 信令交互过程描述如下:

1) 系统上线后, RTSP Proxy 主动向 RTSP 媒体设备发送 DESCRIBE 请求, 收到响应消息后, 保存 SDP(记为 SDP1)信息并据此做基本初始化工作. 完成后定时向 RTSP 媒体设备发送 OPTIONS 消息, 保持 RTSP 连接处于活动状态;

2) 当 SIP Client A 访问 RTSP 媒体设备 B(第一次被访问)时, 向 SIP 服务器发送 INVITE(含 SDP, 记为 SDP2)请求, 建立会话, 并生成该会话的唯一标示 call-ID, 后面的信令交互都应携带该标示以说明其所属会话. 服务器收到请求消息后, 根据 SIP URI 查找对应的 RTSP URI, 查找成功后给 A 回应 100 Trying 响应(否则回送 Not Found). 请求消息经信令转换模块处理, 得到对应的 RTSP SETUP 请求. Proxy RTSP 保存 SDP2 后, 创建 RTP/RTCP 套接字, 建立 RTP 通道. 然后将转换得到的 RTSP SETUP 请求发送给 B;

3) Proxy RTSP 收到 B 的响应消息后, 分配好相关资源后, 随即构建 RTSP PLAY 请求, 发送给媒体设备 B. B 收到 PLAY 请求后, 发送 200 OK 响应, 开始向 RTSP Proxy 传送数据流;

4) Proxy RTSP 收到 PLAY 的响应消息后, 根据 SDP1 和 SDP2 进行媒体协商, 得到 SDP3, 确定 SIP Client A 可接收的媒体流. 再经过信令转换模块形成 SIP 200 OK 响应消息, 携带 SDP3 发送给 A;

5) A 收到 INVITE 响应消息后, 根据 SDP3 分配资源, 准备接收媒体流数据. 随即向服务器发送 SIP ACK 确认消息. Proxy RTSP 收到确认消息后, 将 SIP Client A 的数据接收地址加入到目的地址集中, 开始向 A 发送媒体数据. 之后, SIP Client 通过 RTSP Proxy 接收媒体流数据;

6) A 结束访问时, 主动发送 SIP BYE 消息给服务器, 经信令转换模块处理, 得到对应的 RTSP TEARDOWN 消息, 此时如果仅有 A 访问 B, 则由 RTSP Proxy 向 B 发送 TEARDOWN 消息, 否则将流引用次数减 1 后给 A 回送 SIP BYE 消息; 若 B 收到

TEARDOWN 消息, 则做相应处理后回送 RTSP 200 OK 响应, 再经信令转换模块, 向 A 发送 SIP 487(呼叫被终止)响应. 至此呼叫过程全部结束.

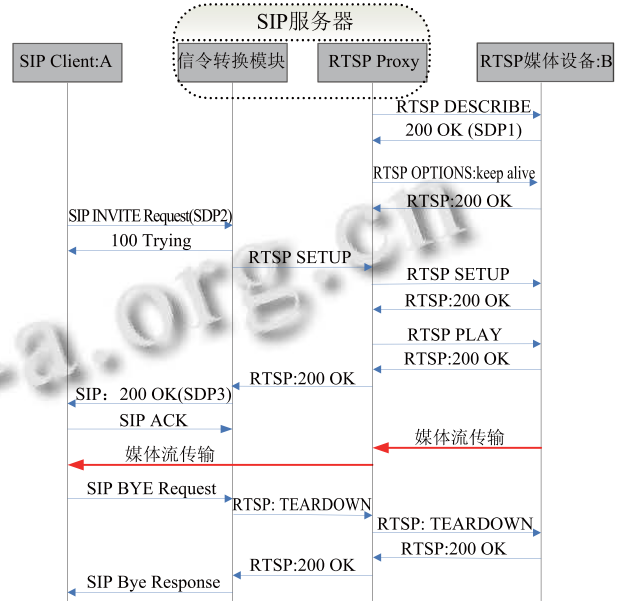


图 2 SIP Client 首次访问 RTSP 媒体设备信令交互示意图

当 A 和 B 已建立连接, 开始数据流传送之后, 此时如果有 SIP Client C 发送 INVITE 请求(含 SDP4)访问 B 时, RTSP Proxy 收到该请求后, 只需根据 SDP1 与 SDP4 得到媒体协商结果 SDP5, 并通过 SIP Server 向 C 发送 SIP 200 Ok 响应消息(含 SDP5). C 收到响应消息后, 根据 SDP5 分配资源、准备好接收媒体流数据后, 向服务器发送 SIP ACK 确认消息. Proxy RTSP 收到确认消息后, 将 B 相应媒体流引用数增 1 后, 只需将 C 的数据接收地址加入到目的地址集即可.

4 实验

本实验借助于实验室的 SIP 服务器, 集成以开源项目 Live555^[9]为基础、经二次开发得到的 RTSP Proxy, 添加信令转换模块后, 在局域网环境下, 通过 SIPHello Client 请求访问由 RTSP Proxy 所代理的 RTSP 摄像头, 成功获取到了 RTSP 摄像头所拍摄的实时画面. 并与通过 VLC 直接请求该 RTSP 摄像头拍摄的实时画面做比较, 通过人眼不能识别出存在画面延迟, 如图 3 所示.

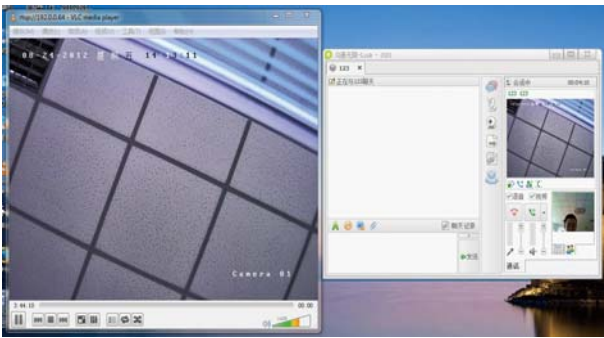


图 3 通过 VLC 及 SIPHello 访问 RTSP 摄像头

通过 WireShark 抓取 RTSP/SIP 协议数据包, 如图 4 所示, SIP 服务器的地址为 192.168.139.121, SIPHello 所在主机的 IP 地址为 192.168.139.76, RTSP 摄像头 IP 地址为 192.168.139.122.

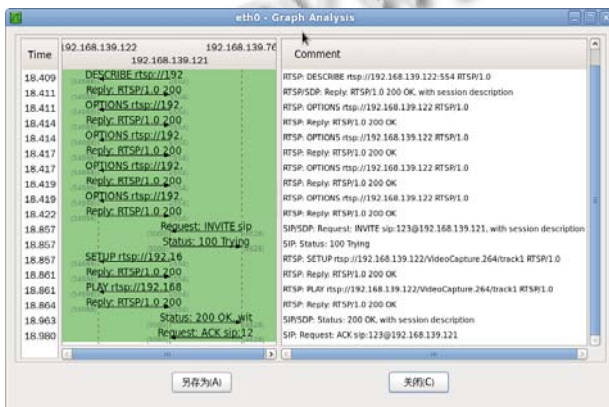


图 4 SIP 和 RTSP 信令交互过程

从实验结果可以看到, 协议信令转换过程按照预期的流程进行工作. 通过进一步分析协议中的各种参数转换细节, 该信令转换模块可以正确转换可识别的选项参数. 经过多次观察实验所获取的实时画面, 不存在显著的延迟现象, 具有较好的实时性. 加之 RTSP 代理服务器的存在, 该模块有较强的扩展性, 易于增加对协议扩展内容的支持, 同时, RTSP 终端无需做任

何改动, 易于集成与实施.

5 结语

本文模拟了多媒体调度系统中, 通过 SIP Client 访问 RTSP 媒体设备的过程, 从某一方面验证了 SIP Client 与 RTSP 终端实现连通的可能性, 实现了媒体流从 RTSP 终端到 SIP Client 方向的传输, 为基于 SIP 的多媒体调度系统与已部署的 RTSP 设备的集成问题提供了较好的解决方案. 实现媒体流的双向传输是我们下一步工作的研究重点, 这样我们不仅可以直接观察到监控现场的情景, 而且可以通过监控装置的音频采集设备实现与现场的实时通话.

参考文献

- 1 IETF RFC 3261.Session Initiation Protocol (SIP).2002,06.
- 2 IETF RFC 2326.Real Time Streaming Protocol (RTSP).1998, 04.
- 3 GB/T 28181-2011,安全防范视频监控联网系统信息传输、交换、控制技术要求[S].
- 4 凌庆华,石志强,程伟明.基于 SIP 的网络视频监控系统的设计与实现.计算机工程,2007,33(2):261-263.
- 5 司端锋,韩心慧,龙勤,潘爱民.SIP 标准中的核心技术与研究进展.软件学报,2005,16 (2):239-250.
- 6 IETF RFC 3550.A Transport Protocol for Real-Time Applications (RTP).2003,07.
- 7 IETF RFC 4566.Session Description Protocol(SDP).2006,07.
- 8 黄永峰等.下一代网络核心控制协议:SIP 及其应用.1st ed, 北京:人民邮电出版社.2009:17-40.
- 9 茅炎菲,黄忠东.基于 RTSP 协议网络监控系统的研究与实现.计算机工程与设计,2011,7(32):2523-2526.
- 10 IETF RFC 3984.RTP Payload Format for H.264 Video. 2005,02.
- 11 黄伦文,陈勇,李涵.基于 SIP 协议的网络视频监控系统的设计与实现.电脑知识与技术,2010,6(1):74-76.