
RTP 协议分析

第 1 章. RTP 概述

1.1. RTP 是什么

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。

1.2. RTP 的应用环境

RTP 用于在单播或多播网络中传送实时数据。它们典型的应用场合有如下几个。

简单的多播音频会议。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)。

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。

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（CSRC 表，见 RTP 的封装）可以确认谈话者。

1.3. 相关概念

1.3.1. 流媒体

流媒体是指 Internet 上使用流式传输技术的连续时基媒体。当前在 Internet 上传输音频和视频等信息主要有两种方式：下载和流式传输两种方式。

下载情况下，用户需要先下载整个媒体文件到本地，然后才能播放媒体文件。在视频直播等应用场合，由于生成整个媒体文件要等直播结束，也就是用户至少要在直播结束后才能看到直播节目，所以用下载方式不能实现直播。

流式传输是实现流媒体的关键技术。使用流式传输可以边下载边观看流媒体节目。由于 Internet 是基于分组传输的，所以接收端收到的数据包往往有延迟和乱序（流式传输构建在 UDP 上）。要实现流式传输，就是要从降低延迟和恢复数据包时序入手。在发送端，为降低延迟，往往对传输数据进行预处理（降低质量和高效压缩）。在接收端为了恢复时序，采用了接收缓冲；而为了实现媒体的流畅播放，则采用了播放缓冲。

使用接收缓冲,可以将接收到的数据包缓存起来,然后根据数据包的封装信息(如包序号和时戳等),将乱序的包重新排序,最后将重新排序了的数据包放入播放缓冲播放。

为什么需要播放缓冲呢?容易想到,由于网络不可能很理想,并且对数据包排序需要处理时耗,我们得到排序好的数据包的时间间隔是不等的。如果不用播放缓冲,那么播放节目会很卡,这叫时延抖动。相反,使用播放缓冲,在开始播放时,花费几十秒钟先将播放缓冲填满(例如 PPLIVE),可以有效地消除时延抖动,从而在不太损失实时性的前提下实现流媒体的顺畅播放。

到目前为止,Internet 上使用较多的流式视频格式主要有以下三种:RealNetworks 公司的 RealMedia ,Apple 公司的 QuickTime 以及 Microsoft 公司的 Advanced Streaming Format (ASF)。

上面在谈接收缓冲时,说到了流媒体数据包的封装信息(包序号和时戳等),这在后面的 RTP 封装中会有体现。另外,RealMedia 这些流式媒体格式只是编解码有不同,但对于 RTP 来说,它们都是待封装传输的流媒体数据而没有什么不同。

第 2 章. RTP 详解

2.1. RTP 的协议层次

2.1.1. 传输层的子层

RTP(实时传输协议),顾名思义它是用来提供实时传输的,因而可以看成是传输层的一个子层。图 1 给出了流媒体应用中的一个典型的协议体系结构。

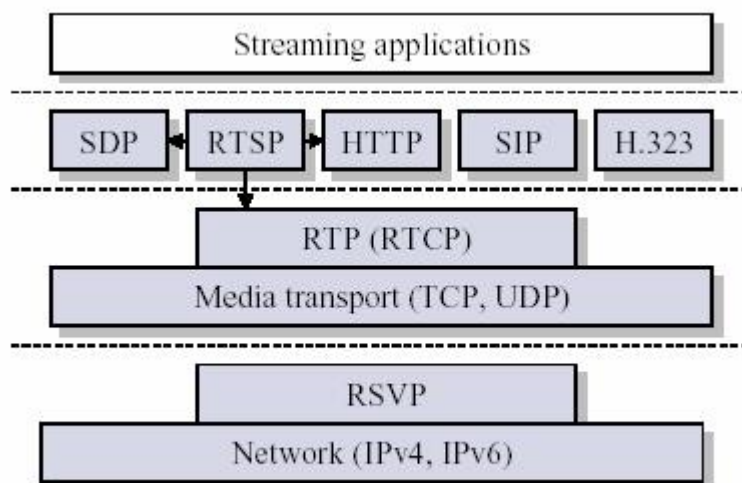


图 1 流媒体体系结构

从图中可以看出，RTP 被划分在传输层，它建立在 UDP 上。同 UDP 协议一样，为了实现其实时传输功能，RTP 也有固定的封装形式。RTP 用来为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。

2.1.2. 应用层的一部分

不少人也把 RTP 归为应用层的一部分，这是从应用开发者的角度来说的。操作系统中的 TCP/IP 等协议栈所提供的是我们最常用的服务，而 RTP 的实现还是要靠开发者自己。因此从开发的角度来说，RTP 的实现和应用层协议的实现没不同，所以可将 RTP 看成应用层协议。

RTP 实现者在发送 RTP 数据时，需先将数据封装成 RTP 包，而在接收到 RTP 数据包，需要将数据从 RTP 包中提取出来。

2.2. RTP 的封装

一个协议的封装是为了满足协议的功能需求的。从前面提出的功能需求，可以推测出 RTP 封装中应该有同步源和时戳等字段，但更为完整的封装是什么样子呢？请看图 2。

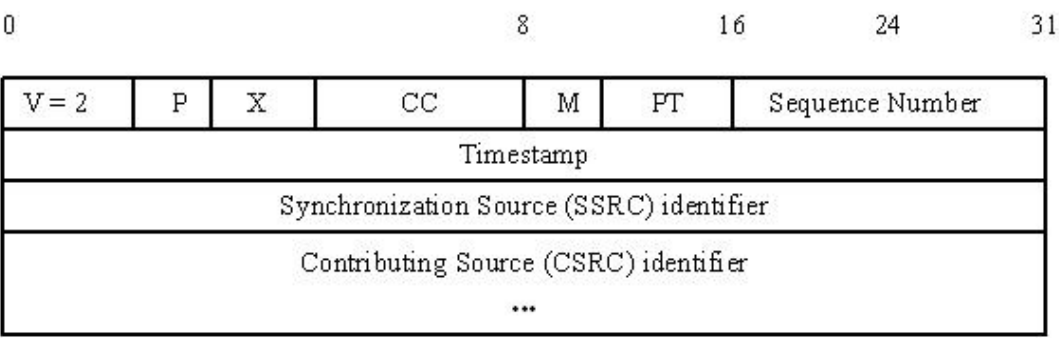


图 2 RTP 的头部格式

- 版本号（V）：2 比特，用来标志使用的 RTP 版本。
- 填充位（P）：1 比特，如果该位置位，则该 RTP 包的尾部就包含附加的填充字节。
- 扩展位（X）：1 比特，如果该位置位的话，RTP 固定头部后面就跟有一个扩展头部。
- CSRC 计数器（CC）：4 比特，含有固定头部后面跟着的 CSRC 的数目。
- 标记位（M）：1 比特,该位的解释由配置文档（Profile）来承担。
- 载荷类型（PT）：7 比特，标识了 RTP 载荷的类型。
- 序列号（SN）：16 比特，发送方在每发送完一个 RTP 包后就将该域的值增加 1，接收方可以由该域检测包的丢失及恢复包序列。序列号的初始值是随机的。
- 时间戳：32 比特，记录了该包中数据的第一个字节的采样时刻。在一次会话开始时，时间戳初始化成一个初始值。即使在没有信号发送时，时间戳的数值

也要随时间而不断地增加（时间在流逝嘛）。时间戳是去除抖动和实现同步不可缺少的。

同步源标识符(SSRC)：32 比特，同步源就是指 RTP 包流的来源。在同一个 RTP 会话中不能有两个相同的 SSRC 值。该标识符是随机选取的 RFC1889 推荐了 MD5 随机算法。

贡献源列表（CSRC List）：0~15 项，每项 32 比特，用来标志对一个 RTP 混合器产生的新包有贡献的所有 RTP 包的源。由混合器将这些有贡献的 SSRC 标识符插入表中。SSRC 标识符都被列出来，以便接收端能正确指出交谈双方的身份。

2.3. RTCP 的封装

RTP 需要 RTCP 为其服务质量提供保证，因此下面介绍一下 RTCP 的相关知识。

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。

从图 1 可以看到，RTCP 也是用 UDP 来传送的，但 RTCP 封装的仅仅是一些控制信息，因而分组很短，所以可以将多个 RTCP 分组封装在一个 UDP 包中。

RTCP 有如下五种分组类型。

类型	缩写表示	用途
200	SR (Sender Report)	发送端报告
201	RR (Receiver Report)	接收端报告
202	SDES (Source Description Items)	源点描述
203	BYE	结束传输
204	APP	特定应用

表 1 RTCP 的 5 种分组类型

上述五种分组的封装大同小异，下面只讲述 SR 类型，而其它类型请参考 RFC3550。

发送端报告分组 SR (Sender Report) 用来使发送端以多播方式向所有接收端报告发送情况。SR 分组的主要内容有：相应的 RTP 流的 SSRC，RTP 流中新产生的 RTP 分组的时间戳和 NTP，RTP 流包含的分组数，RTP 流包含的字节数。

RTCP 头部的格式

版本 (V)：同 RTP 包头域。

填充 (P)：同 RTP 包头域。

接收报告计数器 (RC)：5 比特，该 SR 包中的接收报告块的数目，可以为零。

包类型 (PT)：8 比特，SR 包是 200。

长度域 (Length)：16 比特，其中存放的是该 SR 包以 32 比特为单位的总长度减一。

同步源 (SSRC) : SR 包发送者的同步源标识符。与对应 RTP 包中的 SSRC 一样。

NTP Timestamp (Network time protocol) SR 包发送时的绝对时间值。NTP 的作用是同步不同的 RTP 媒体流。

RTP Timestamp : 与 NTP 时间戳对应, 与 RTP 数据包中的 RTP 时间戳具有相同的单位和随机初始值。

Sender' s packet count : 从开始发送包到产生这个 SR 包这段时间里, 发送者发送的 RTP 数据包的总数. SSRC 改变时, 这个域清零。

Sender' s octet count : 从开始发送包到产生这个 SR 包这段时间里, 发送者发送的净荷数据的总字节数 (不包括头部和填充)。发送者改变其 SSRC 时, 这个域要清零。

同步源 n 的 SSRC 标识符 : 该报告块中包含的是从该源接收到的包的统计信息。

丢失率 (Fraction Lost) : 表明从上一个 SR 或 RR 包发出以来从同步源 n(SSRC_n)来的 RTP 数据包的丢失率。

累计的包丢失数目 : 从开始接收到 SSRC_n 的包到发送 SR,从 SSRC_n 传过来的 RTP 数据包的丢失总数。

收到的扩展最大序列号 : 从 SSRC_n 收到的 RTP 数据包中最大的序列号 ,

接收抖动 (Interarrival jitter) : RTP 数据包接受时间的统计方差估计

上次 SR 时间戳 (Last SR,LSR) : 取最近从 SSRC_n 收到的 SR 包中的 NTP 时间戳的中间 32 比特。如果目前还没收到 SR 包, 则该域清零。

上次 SR 以来的延时 (Delay since last SR,DLSR) : 上次从 SSRC_n 收到 SR 包到发送本报告的延时。

2.4. RTP 的会话过程

当应用程序建立一个 RTP 会话时, 应用程序将确定一对目的传输地址。目的传输地址由一个网络地址和一对端口组成, 有两个端口: 一个给 RTP 包, 一个给 RTCP 包, 使得 RTP/RTCP 数据能够正确发送。RTP 数据发向偶数的 UDP 端口, 而对应的控制信号 RTCP 数据发向相邻的奇数 UDP 端口 (偶数的 UDP 端口 + 1), 这样就构成一个 UDP 端口对。RTP 的发送过程如下, 接收过程则相反。

1) RTP 协议从上层接收流媒体信息码流 (如 H.263), 封装成 RTP 数据包; RTCP 从上层接收控制信息, 封装成 RTCP 控制包。

2) RTP 将 RTP 数据包发往 UDP 端口对中偶数端口; RTCP 将 RTCP 控制包发往 UDP 端口对中的接收端口。

第 3 章. 相关的协议

3.1. 实时流协议 RTSP

实时流协议 RTSP (Real-Time Streaming Protocol) 是 IETF 提出的协议, 对应的 RFC 文档为 RFC2362。

从图 1 可以看出, RTSP 是一个应用层协议 (TCP/IP 网络体系中)。它以 C/S 模式工作, 它是一个多媒体播放控制协议, 主要用来使用户在播放流媒体时可以像操作本地的影碟机一样进行控制, 即可以对流媒体进行暂停/继续、后退和前进等控制。

3.2. 资源预定协议 RSVP

资源预定协议 RSVP(Resource Reservation Protocol)是 IETF 提出的协议，对应的 RFC 文档为 RFC2208。

从图 1 可以看出，RSVP 工作在 IP 层之上传输层之下，是一个网络控制协议。RSVP 通过在路由器上预留一定的带宽，能在一定程度上为流媒体的传输提供服务质量。在某些试验性的系统如网络视频会议工具 vic 中就集成了 RSVP。