第三章 实时传输协议

本章目录

- RTP的基本设计理念
- RTP的标准元素
- 相关标准
- 未来标准的开发

本章描述了RTP框架的设计，从设计的理念和背景出发，概述了适用的标准，并解释了这些标准之间的相互关系。最后讨论了这些标准今后可能的发展方向。

RTP的基本设计理念

RTP的设计者面临的挑战是在一个不可靠的传输层之上构建一个健壮的、实时的媒体传输机制。他们实现了这个目标，设计遵循了应用级框架和端到端原则的双重哲学。

应用级框架

应用级框架背后的概念是由Clark和Tennenhouse在1990年首次阐明的。他们的中心论点是，只有应用程序对其数据有足够的了解，才能对如何传输数据做出明智的决定。这意味着对应用程序有意义的单元(应用程序数据单元、ADUs)中的数据，传输协议应该接受这些数据，并尽可能公开它们的传输细节，以便在发生错误时应用程序可以做出适当的响应。应用程序与传输程序一起，实现可靠的传输。

应用程序级框架源于这样一种认识，即应用程序可以通过多种方式从网络问题中恢复，正确的方法取决于应用程序和使用它的场景。在某些情况下，需要重新传输丢包数据的精确副本。在其他情况下，可能会使用保真度较低的副本，或者数据可能已经被替换，因此副本与原始数据不同。或者，如果数据只具有短暂的兴趣，则可以忽略丢包。这些选择只有在应用程序与传输紧密交互时才有可能。

应用程序级框架的目标与TCP的设计有些不一致，TCP的设计隐藏了底层IP网络的有损特性，以牺牲时效性来实现可靠的传输。但是，它非常适合基于UDP的传输和实时媒体的特性。如第二章所述，分组网络上的音视频通信，实时音视频媒体通常是容错的，但有严格的时间限制。使用基于UDP传输的应用程序级框架，我们可以在必要时接受丢包，但我们

也可以灵活地使用全范围的恢复技术，如在适当的情况下重传和错误恢复。

这些技术为应用程序提供了极大的灵活性，使其能够以适当的方式对网络问题作出反应，而不是受单个传输层的约束。

根据应用程序级框架的原则设计的网络不应该针对特定的应用程序。相反，它应该暴露通用传输层的限制，以便应用程序能够与网络合作，实现尽可能好的传输。应用层框架意味着OSI参考模型定义的严格层的弱化。这是一种实用的方法，它承认分层的重要性，但也承认需要公开较低层的一些细节。

应用程序级框架的哲学意味着能够对问题做出反应的智能网络感知应用程序。

端到端原则

RTP采用的另一种设计理念是端到端原则。它是设计必须通过网络进行可靠通信的系统的两种方法之一。在一种方法中，系统可以将正确传输数据的责任与数据一起往下传递，从而确保每一跳的可靠性。在另一种方法中，数据的责任由端点承担，即使单个跃点不可靠，也可以确保端到端的可靠性。第二种端到端方法渗透到互联网的设计中，TCP和RTP都遵循端到端原则。

端到端原则的主要结果是，智能倾向于向上冒泡到协议栈的顶部。如果组成网络路径的系统不负责数据，那么它们可以很简单，也不需要很健壮。它们可能会丢弃无法传输的数据，因为没有它们的帮助，端点也会恢复。端到端原则意味着智能是在端点上，而不是在网络中。

其结果是一个包含智能、能感知网络的端点和哑网络（不智能的网络）的设计。这种设计非常适合因特网—可能是最基本的哑网络—但是需要应用程序设计人员进行大量的工作。它的设计也不同于其他许多网络。以传统的电话网为例，它采用了智能网络和哑端点的模型，而MPEG传输模型允许哑接收器和智能发送者。这种设计上的差异改变了应用程序的风格，更加强调接收方的设计，使发送方和接收方在传输中更加平等。

实现灵活性

RTP框架被设计为可以满足许多场景，几乎不需要额外的协议支持。在很大程度上，这个设计是基于视频会议的轻量级会话模型。在这个场景中，RTP控制协议提供了所有必要的会话管理功能，连接会话所需的全部内容是IP地址和映射，其中映射是指从媒体定义到RTP有效负载类型标识符。这个模型也适用于一到多个场景——例如，网络电台，控制协议提供的反馈可以给信息源一个关于观众规模和接收质量的预估。

对于单播语音电话，一些人认为RTP提供了不必要的特性，并且对于高度压缩的语音帧来说是过重和低效的。在实践中，使用头压缩时，这不是一个强有力的反对论调，而且RTP所提供的特性可以轻松地扩展到多媒体和多方会话。还有一些人——例如数字电影社区——认为RTP没有充分满足他们的需求，应该包括更强的服务质量和安全支持，更详细的统计数据，等等。

RTP的优点是它提供了一个统一的实时音频/视频传输框架，可以直接满足大多数应用程序，但对于那些超出其限制的应用程序来说，它是可塑的。

RTP的标准元素

IP网络中音频/视频传输的主要标准是实时传输协议(RTP)，以及相关的配置文件和有效负载格式。RTP是由互联网工程任务组(IETF)的音频/视频传输工作组开发的，它已经被国际电信联盟(ITU)作为其H.323系列建议的一部分，并被其他几个标准组织采用。

RTP为实时媒体的传输提供了一个框架，在完成之前需要对其进行特殊用途的分析。对音频和视频会议的最低控制的RTP配置文件与RTP一起进行了标准化，还有几个配置文件正在开发中。每个配置文件都伴随着几个有效负载格式规范，每个规范都描述了特定媒体格式的传输。

RTP规范

RTP于1996年1月6日作为IETF拟议标准(RFC1889)发布，其标准草案的修订已基本完成。国际电联建议H.323的第一次修订包括RTP规范的逐字副本;后来的修订参考了当前的IETF标准。

在IETF标准过程中，规范经历了一个开发周期，在此期间，随着设计细节的确定，将生成多个互联网草案。当设计完成时，它作为一个建议的标准RFC发布。建议的标准通常被认为是稳定的，解决了所有已知的设计问题，适合实现。如果该标准被证明是有用的，并且该标准的每个特性都有独立的、可互操作的实现，那么它就可以被提升到标准草案的状态(可能包括修改以纠正在该标准中发现的任何问题)。最后，经过广泛的经验，它可以作为一个完整的标准RFC发布。超出所提议的标准状态的改进是造成许多协议从未实现的主要障碍。

RTP通常位于UDP / IP传输之上，通过丢包检测和接收质量报告、时序恢复和同步、有效负载和源标识以及媒体流中重要事件的标记来增强该传输。 RTP的大多数实现是应用程序或库的一部分，该应用程序或库位于操作系统提供的UDP / IP套接字接口之上。 但是，这不是唯一可能的设计，并且RTP协议中的任何内容都不需要UDP或IP。 例如，某些实现基于

TCP / IP之上的RTP，而其他实现甚至在非IP网络（例如异步传输模式（Asynchronous Transfer Mode，ATM）网络）上使用RTP。

RTP包括两部分:数据传输协议和相关的控制协议。RTP数据传输协议管理终端系统之间的实时数据传输，如音频和视频。它为媒体有效负载定义了额外的帧级别，包括用于丢包检测的序列号、用于恢复时序的时间戳、有效负载类型和源标识符，以及媒体流中重要事件的标记。还指定了使用时间戳和序列号的规则，尽管这些规则在某种程度上依赖于使用的配置文件和有效负载格式，以及在一个会话中多路复用多个流。第四章进一步讨论了RTP数据传输协议。

RTP控制协议(RTCP)提供接收质量反馈、参与者识别和媒体流之间的同步。RTCP与RTP一起运行，并定期报告这些信息。虽然数据包通常每隔几毫秒发送一次，但控制协议以秒为单位进行操作。RTCP中发送的信息对于媒体流之间的同步是必要的——例如，对于音频和视频之间的同步，并且可以根据接收质量反馈调整传输和识别参与者。第五章进一步讨论了RTP控制协议。

RTP支持混合器和翻译器的概念，即相当于中间盒，当它在媒体端点之间流动时可以进行操作。它们可以用于在不同的底层协议之间转换RTP会话—例如，在IPv4和IPv6网络上的参与者之间进行桥接，或者将单播参与者引入多播组。它们还可以以某种方式调整媒体流——例如，转换数据格式以减少带宽，或者将多个流混合在一起。

很难将RTP放在OSI的参考模型中。它执行许多通常分配给传输层协议的任务，但它本身并不构成一个完整的传输层。RTP还执行会话层(跨越不同的传输连接并以与传输无关的方式管理参与者标识)和表示层(为媒体数据定义标准表示)的一些任务。

RTP配置文件

了解RTP协议规范的限制非常重要，因为它在两方面刻意不完整。首先，该标准没有指定用于媒体播放和时序再生、媒体流之间的同步、错误隐藏和纠正或拥塞控制的算法。这完全是应用程序设计人员的职责范围，因为不同的应用程序有不同的需求，所以如果标准要求使用单一的行为，那就太愚蠢了。当然，它确实为这些算法提供了必要的信息，以供它们操作。后面的章节将讨论应用程序设计和提供这些特性必要的权衡。

其次，传输的一些细节可以通过配置文件和有效负载格式定义进行修改。这些特性包括时间戳的解析、媒体流中感兴趣事件的标记和有效负载类型字段的使用。可以通过RTP配置文件指定的功能包括:

- RTP报头中的有效负载类型标识符与有效负载格式规范之间的映射(有效负载格式规范描述了如何在RTP中使用单个媒体编解码器)。每个配置文件将引用多个有效负载格式并可

能指示如何使用特定的信令协议(例如SDP)来描述映射。
- RTP报头中的有效负载类型标识符字段的大小，以及用于标记媒体流中感兴趣的事件的位数。
- 固定的RTP数据传输协议报头的补充部分，如果该报头被证明不足以用于特定的应用程序类。
- RTP控制协议的报告间隔——例如，以牺牲额外的开销为代价使反馈更及时。
- 如果RTCP所提供的某些信息对应用程序没有用处，则对所使用的RTCP包类型进行限制。此外，配置文件可以定义RTCP的扩展以报告额外信息。
- 附加的安全机制——例如，新的加密和认证算法。
- 将RTP和RTCP映射到底层传输协议。

在撰写本文时，只有一个RTP配置文件:用于音频和视频会议的RTP配置文件，只有很少的控制。这个配置文件在1996年1月作为一个提议的标准(RFC 1890)和RTP规范一起发布，其标准状态草案的修订几乎已经完成。一些新的配置文件正在开发中。指定附加安全性的配置文件，以及反馈和修复机制，这些内容可能很快就会提供。

RTP有效负载格式

RTP框架的最后一部分是有效负载格式，它定义了在RTP中如何传输特定的媒体类型。有效负载格式由RTP配置文件引用，它们还可以定义RTP数据传输协议的某些属性。

尽管配置文件也可以为有效负载格式指定一些常规行为，但是RTP有效负载格式和配置文件之间的关系主要是一个名称空间。名称空间将RTP包中的有效负载类型标识符与有效负载格式规范联系起来，从而允许应用程序将数据与特定的媒体编解码器联系起来。在某些情况下，有效负载类型和有效负载格式之间的映射是静态的;在其他情况下，通过带外控制协议，映射是动态的。例如，具有最小控制的音频和视频会议的RTP配置文件定义了一组静态负载类型分配，以及一种机制，用于在标识负载格式的MIME类型和使用会话描述协议(SDP)的负载类型标识之间进行映射。

有效负载格式和RTP数据传输协议之间的关系是双重的:有效负载格式将指定某些RTP头字段的使用，并且可以定义附加的有效负载头。媒体编解码器产生的输出被转换成一系列的RTP数据包—一些部分映射到RTP报头，一些映射到有效负载报头，大部分映射到有效负载数据。这种映射过程的复杂性取决于编解码器的设计和所需的错误恢复程度。在某些情况下，映射很简单;而另一些情况则更为复杂。

最简单的情况下，有效负载格式只定义了媒体时钟和RTP时间戳之间的映射，并要求将每一帧的编解码器输出直接放到一个RTP包中进行传输。这方面的一个例子是G.722.1音频的有效负载格式。不幸的是，这在许多情况下是不够的，因为许多编解码器的开发没有参考包传输系统的需要，因此需要调整以适应这种环境。其他的是为分组网络设计的，但是需要额外的头信息。在这些情况下，有效负载格式规范定义了一个附加的有效负载包头以及该包头的生成规则，其中有效负载包头放置在主RTP包头之后。

许多有效负载格式已经被定义，与目前使用的多种编解码器相匹配，还有许多正在开发中。在撰写本文时，以下音频有效负载格式是常用的，尽管这不是一个详尽的列表:G.711, G.723.1, G.726, G.728, G.729, GSM, QCELP, MP3，和DTMF。常用的视频有效负载格式包括H.261、H.263和MPEG。

还有指定错误恢复方案的有效负载格式。例如，RFC 2198定义了一个音频冗余编码方案，RFC 2733定义了一种通用的基于奇偶校验编码的前向纠错方案。在这些有效负载格式中有一个额外的间接层，编解码器输出被映射到RTP包，这些包本身被映射来产生一个带容错的传输。误差校正将在第9章《错误恢复》中详细讨论。

可选的元素

RTP框架的两个可选部分在这个阶段值得一提:头压缩和多路复用。

报头压缩是一种方法，通过这种方法可以在每个链接的基础上减少RTP和UDP/IP报头的开销。它用于带宽受限的链路(例如蜂窝链路和拨号链路)，可以将RTP/UDP/IP报头的40字节组合减少到2字节，代价是压缩链路两端的系统需要额外处理。包头压缩将在第11章进一步讨论。

多路复用是将多个相关的RTP会话组合成一个的方法。同样，这样做的动机也是为了减少包头的开销，只不过这次是端到端的操作。多路复用将在第12章《多路复用和隧道》中讨论。

包头压缩和多路复用都可以被认为是RTP框架的一部分。与配置文件和有效负载格式不同，它们显然是系统的用于特殊用途的可选部分，而且许多实现都不使用这两个特性。

相关标准

除了RTP框架外，完整的系统通常还需要使用其他各种协议和标准来设置和控制呼叫、会话描述、多方通信和信令化服务质量要求。虽然本书没有详细介绍这些协议的使用，但在本节中，它提供了这些协议规范的说明和进一步的阅读建议。

完整的多媒体协议栈如图3.1所示，展示了RTP框架与支持的设置和控制协议之间的关系。本书中所讨论的协议和功能已被重点标记。

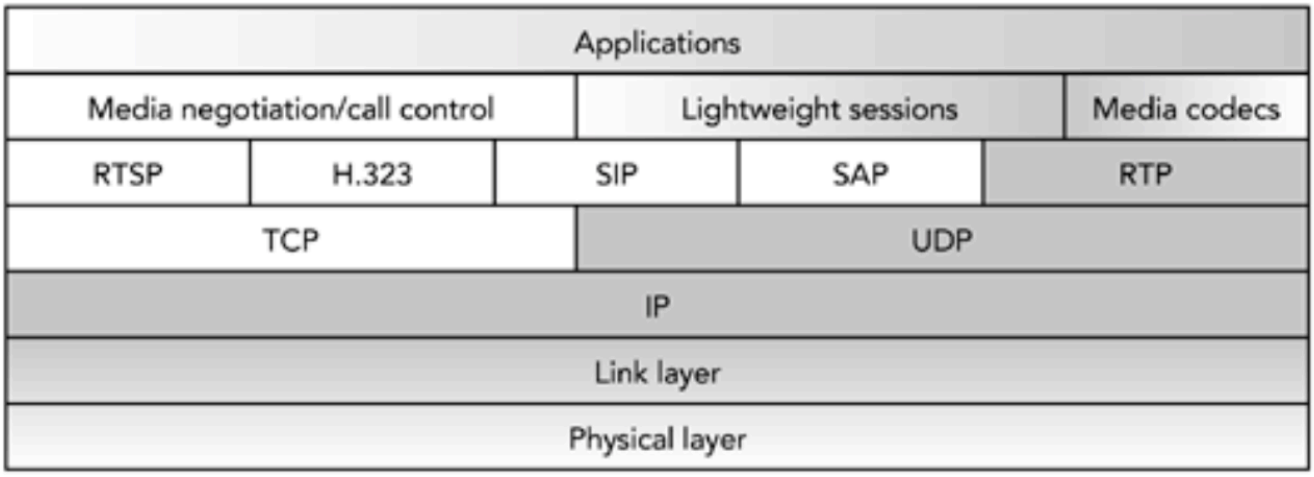

图3.1 多媒体协议堆栈

调用设置和控制

根据应用程序场景，可以使用各种呼叫设置、控制和广告协议来启动RTP会话:

- 为了启动交互式会话，无论是语音电话呼叫还是视频会议，有两个标准。这方面的最初标准是ITU建议H.323，最近IETF定义了会话发起协议(SIP)。
- 为了开始一个非交互式会话，例如视频点播，主要的标准是实时流协议(RTSP)。
- RTP最初的用途是与IP组播和轻量级会议模型一起使用。本设计采用了会话通知协议(SAP)和IP多播通知正在进行的会议，如向公众开放的研讨会和电视广播。

就会话中的参与者数量和这些参与者之间的耦合而言，这些协议的需求非常不同。有些会话是松耦合的，只有有限的成员控制和参与者的知识。其他的则是严格管理的，需要明确的许可才能加入、交谈、聆听和观看。

这些不同的需求导致为每个场景设计了非常不同的协议，并且在这个领域中正在进行大量的工作。RTP故意不包含会话发起和控制功能，这使得它适用于广泛的应用程序。

作为应用程序设计人员，除了RTP提供的媒体传输之外，还必须实现某种形式的会话发起、调用设置或调用控制。

会话描述

所有设置和通知协议都需要一种描述会话的方法。在这个领域中一个常用的协议是会话描述协议(SDP)，尽管也可能使用其他机制。

不管会话描述的格式如何，始终需要某些信息。媒体流所在的传输地址、媒体的格式、要使用的RTP有效负载格式和配置文件、会话活动的时间以及会话的目的等必须传递。

SDP将这些信息打包成一种文本文件格式，这种格式是人类可读的，并且易于解析。在某些情况下，该文件直接传递给RTP应用程序，从而提供足够的信息来直接加入会话。在另一些情况下，会话描述是协商的基础，是呼叫设置协议的一部分，然后参与者才能进入严格控制的电话会议。

第四章RTP《数据传输协议》对SDP进行了详细的讨论。

服务质量

虽然RTP的设计是通过IP提供的最佳服务进行操作，但有时能够保留网络资源是有用的，从而提高了RTP流的服务质量。再说一遍，这不是RTP提供的服务，需要另一个协议的帮助。在撰写本文时，还没有普遍接受的在互联网上保留资源的“最佳实践”。存在两个标准框架，集成服务和差异化服务，每个框架的部署都很有限。

通过使用资源保留协议(RSVP)，集成服务框架提供了严格的服务质量保证。路由器需要将可用容量划分为服务类，并考虑流量使用的容量。在开始传输之前，主机必须向路由器发出它的需求信号，只有在所需的服务类中有足够的容量时，路由器才允许保留成功。只要所有的路由器都尊重服务类，并且不过度承诺资源，这个需求就可以防止链接超载，从而提供有保证的服务质量。可用的服务类别包括有保证的服务(提供有保证的带宽水平、严格的端到端延迟限制、无阻塞性数据包丢失)和可控负载(提供与少量负载的最佳工作网络相当的服务)。

集成服务框架和RSVP需要对每个流程进行预订，并且难以聚合这些预订。因此，将RSVP扩展到大量的异类保留是有问题的，因为路由器必须保持大量的状态，而这个约束限制了它的部署。

差异化服务框架对服务质量采取了一种稍微不同的方法。它没有提供端到端资源保留和严格的性能保证，而是定义了几个每跳排队行为，通过在每个包的IP报头中设置服务类型字段来选择这些行为。这些单跳行为使路由器能够优先处理某些类型的流量，从而降低包或延迟的概率，但是由于路由器无法控制允许进入网络的流量数量，因此无法绝对保证满足

性能限制。差别化服务框架的优点是它不需要复杂的信令，并且状态需求比RSVP小得多。缺点是它只提供统计上的保证。

集成和差异化服务框架的组合非常强大，未来的网络可能会将它们组合在一起。RSVP可用于向边缘路由器发出应用程序需求的信号，然后这些路由器将这些需求映射到不同的服务流量类。这种组合允许边缘路由器拒绝过多的流量，提高了差异化服务网络所能提供的保障，同时将RSVP所需的状态保持在网络核心之外。

这两个框架都占据一席之地，但在撰写本文时，它们都没有达到临界量。未来的网络可能会采用某种形式的服务质量，但这一点也不确定。在此之前，我们的任务是使应用程序在当前部署的最佳网络中运行良好。

未来的标准开发

随着RTP标准草案状态的修订，协议规范中没有已知的未解决的问题，并且RTP本身在可预见的将来也不会改变。但是，这并不意味着标准工作已经完成。新的有效负载格式一直在开发中，新的配置文件将扩展RTP以包含新的功能(例如，用于安全RTP和增强反馈的配置文件)。

从长期来看，我们期望RTP框架与网络本身一起发展。网络中未来的变化也可能影响RTP，我们希望利用任何变化，来开发新的配置文件。我们还期待一系列新的有效负载格式规范，以跟上编解码器技术的变化，并提供新的错误恢复方案。

最后，我们可以预期在呼叫设置和控制、资源保留和服务质量方面的相关协议会有相当大的变化。这些协议比RTP更新，而且目前正在快速发展，这意味着这里的更改可能比RTP、其配置文件和有效负载格式的更改更重要。

总结

RTP提供了一个灵活的框架，用于通过IP网络传输实时媒体，如音频和视频。它的核心哲学——应用级框架和端到端原则——使其非常适合IP网络的独特环境。

本章概述了RTP规范、配置文件和有效负载格式。相关标准包括呼叫设置、控制和广告，以及资源保留。

本章介绍的RTP的两部分——数据传输协议和控制协议——将在接下来的两章中详细讨论。