RTP ： 应用于实时应用的传输协议

摘要:

本文档阐述了实时传输协议 (RTP), RTP 提供了端到端的网络传输功能，适合于通过多播或单播网络服务的实时数据传输应用，例如：音频，视频等，RTP 不强调资源保留而且并不保证实时服务 QOS(quality-of-service)，实时控制协议(RTCP)对数据进行监控和提供最小的控制和鉴别功能。RTP 和 RTCP 是网络层和传输层上面的独立协议。本协议支持 RTP 级别的 translators and mixers。 本文档大部分和 RFC 1889(已废)在封包格式上并没有改动，而仅仅是在怎样用本协议的规定和算法管理进行了改动，最大的改动是对定时算法的增强，当试图发送 RTCP 报文时, 发现有很多个参与者同时加入一个会话时, 提供最小化传输控制。

1.介绍

本文档详细描述了实时传输协议(RTP)，RTP 为实时数据传输例如交互的音频和视频提供了端到端传输服务。服务包括有效载荷的类型确认，序列编码，时间戳和传呼监控。典型应用是利用 UDP 的多路技术校验和服务而在之上运行 RTP，两者都提供一定的传输控制功能。然而，RTP 还可以与其它适合的协议并用，如果底层网络支持多路分发，RTP 还可以提供数据给多路终点。需要注意的是 RTP 不提供任何的机制以保证数据的实时性并且也不保证它的 QOS，而是依赖底层的服务来提供这些功能，RTP 既不保证传输或者是阻止无序传输，也不假定底层网络是可信任的和传输报文是有序的。RTP 中的序列号允许接收器重建发送器发来的报文，但是序列号还可以用来确定报文的合适位置，例如：在视频解码中，序列中没有了需要解码的报文。虽然 RTP 协议最初是为了满足多媒体会议的需求的而制定的，然而 RTP 协议也适用于连续数据的存储，交互式的分布仿真，active badge，管理和测量应用。本文档定义了 RTP，包含两个紧密联系的部分：

1. 实时传输协议(RTP)，传输具有实时特性的数据。

2. RTP 控制协议(RTCP)，监控 QOS 和传递会话中参与者的信息。而后者对于“宽松会话”(loosely controlled) 来说是足够了，即没有明确的成员控制和 建立, 但是协议没有必要支持一个应用的控制需求的全部。在一个单独的会话控 制协议中可能全部或部分包含了这项功能，这不在本文档讨论之内。

RTP 代表了一种新类型协议, 遵循应用级框架和(integrated layer processing) ，由 Clark 和 Tennenhouse [10] 提出的。即 RTP 可以很容易的扩展[P5]以提供某种特定应用的信息，并且可以经常集成进某 种应用处理而不是作为补充而成立单独层。RTP 是有意的做成不完整的协议框 架。本文档详细描述了那些所有应用都适用的常用功能。不像常规协议那样为了使协议更加一般化而提供附加的功能或是为了增加剖析功能而提供一个选项机 制。RTP 可以根据需要而修改或者增加头字段。例子见于 5.3 和 6.4.3。

因此，除了本文档外，特定应用还需要其它的文档(见第 13 章)

3. 一个剖面详述文档，定义了一系列有效载荷码字和它们映射格式(例，媒体编码)，也定义了为了某一 类特定应用而需要扩展和修改的 RTP。典型的就是仅仅基于一个 profile 而运行的应用。在 RFC 3551 [1] 可以找到用于音频和视频的 profile。

4. 有效载荷格式说明文档，定义了怎样在 RTP 中携带特定的有效载荷，例如音频和视频编码。

2.RTP 使用场景

以下章节描述了使用 RTP 的部分特性。选择的例子是用来阐明基于 RTP 的应 用的基本操作，而不是限制 RTP 仅能用于此类应用。在这些例子中，RTP 是在 IP 和 UDP 之上 carried，遵循在 RFC 3551 中为音频和视频而建立的惯例。

2.1 简单的多播音频会议

IETF 工作组利用 Internet 上的为了语音通信而提供的 IP 多播服务讨论了最新的协议文档，基于某种分配机制，工作组主席得到了多播组的地址和一对端口。其中的一个端口是为了音频数据而准备的，另一个是用来控制(RTCP)报文的。将地址和端口分发给与会者。基于安全考虑，可以将数据报文和控制报文加密，在这种情况中，会生成密钥并将它分发给与会者。详 细的分配和分发机制不在本文档讨论之内。

与会者例如在 20 毫秒的持续期内发送一段音频数据。将 RTP 头加在每一段音频数据的前面，然后插入在 UDP 报文中。RTP 头标明何种类型的数据封装(例如 PCM，ADPCM 或 LPC)以利于发送器拆分报文。例如，给低带宽的参与者进行调节或者对网络拥塞进行重新操作。

像其它网络一样，Internet 偶尔会丢失报文或对报文重排序，或延迟不定长时间。为了防止这些意外，RTP 头字段中含有时间信息和序列号允许接收器按照源报文重建时序。在本例中，讲话者每 20 毫秒持续的发送音频报文段。在会议中的每个源的 RTP 报文的时序是独立重建的。接收者也可以应用序列号来确定丢失了多少报文。

在会议期间，工作组的成员会有人离开或加入进来，所以了解在当前时刻谁在收听以及他们的收听质量如何是有必要的。出于这个目的，每一个音频应用的实体周期的在 RTCP 端口多点传送接收报告和它的使用者的名字。接受报告标明接收者接收到的当前讲话者的通信质量，也可以用来控制自适应编码。除了使用者名字外，还包含其它的鉴别信息。当有人离开时，site 会发送 RTCP BYE 报文

2.2 音频和视频会议

如果会议期间同时应用了音频和视频媒体，它们会作为独立的 RTP 会话 来传输。即，将会为了每一个媒介开一对 UDP 端口来独立传输 RTPRTCP 报文和多播地

址。在 RTP 级，音频和视频会话并没有直接的联系，除非在两个会话中使用者在 RTCP 报文中使用同样的显示名称，这样两个会话就可以联系起来了。

这样做的一个动机就是允许某些与会者可以选择仅仅接收某一媒介的数据。尽管这样，利用会话中的 RTCP 报文携带的时序信息可以重新同步播放音频和视频。

2.3 Mixer 和 Translator

到目前为止，我们已经假设了所有的 sites 希望接收同样格式的媒体数据，然而这经常是不适当的。考虑这样一种情况，与会者中的大部分人在高速网络链路中而某个地方的小部分与会者却只能低速率连接。可以在低带宽区域放置一个 RTP 级的中继(relay)，称作 Mixer，而不是强迫播所有人都用低带宽。Mixer 将接收的音频报文重新同步以重建发送者的恒定的 20 毫秒间隙，将这些重建音频流混合成单一流，并且将音频编码翻译给一个低带宽然后通过低速率链路形成低带宽的报文流。这些报文可以单播也可以多播给更多的接收者。RTP 头字段包含了用于 Mixer 识别混合报文中的源的方法，这样就收者就可以识别出正确的讲话者。

一些音频会议的与会者可能是由高速链路连接的但却不是直接通过多播直达的。例如，他们可能在一个应用级的防火墙的后面，而防火墙不准 IP 报文同过。对于这些 sites，混合就不是必需的了，而另一种 RTP 级的中继称作 Translator 就派上用场了。安装两个 translators，防火墙一面一个，外面的 translator 所有的接收到的多播报文经过安全连接直达到内部的 translator，内部的 translator 再一次作为多播报文传输给限制在 site 内部网络的多播组。

可以基于很多种目的而设计 Mixers 和 Translators 。一个例子就是在独立的视频流中，视频 Mixer scales 个人的图像，然后将它组合进一个视频流来模拟组场景。其它的关于转换包括一组主机仅仅发送基于 IP/UDP 的报文给仅仅基于 ST-II，或者是从没有经过重新同步或混合而单独的源来的 packet-by-packet 编码流。

2.4 层编码

多媒体应用可以根据就收者的能力或者是网络拥塞来调节传输速率。许多的补充协议将速率适应能力放在源端。由于不同种类的接收者需求不一样的带宽，这种方法并不能很好的用于多播传输。结果导致一个最小公分母场景，其中网格中最小的管道指示直播的质量和忠诚度。

因此，速率适应的职责可以放在接收端，合并一个通过层传输的层编码。在通过 IP 多播的 RTP 中接收者可以适应网络的不同并且通过加入适合的多播子组而控制就收带宽。

3.定义

RTP payload：在 RTP 报文中的有效载荷，例如音频采样或者压缩的视频数据。payload 格式和解释不在本文档范围之内。
RTP packet：包含定长的头字段的数据报文，源端或者 payload 数据可能是空。一些低层的协议可能需要定义一种 RTP 报文封装协议。代表性的例子就 是协议下的一个报文包含一个单独的 RTP 报文，但是也可以包含几个 RTP 报文，这取决于所用的封装方法。
RTCP packet：一种包含与 RTP 数据报文很相似的定长头字段的控制报文，紧随头字段的是结构元素，因不同 RTCP 报文而具有不同结构。通常情况下几个 RTCP 报文合在一起作为一个混合的 RTCP 报文在协议下传送；这项功能由每个 RTCP 报文中头字段中的长度域来指定。
Port：“传输协议用来区分主机下的不同应用，它是抽象出来的，TCP/IP 协议可以识别用正整数的端口” [12] OSI 模型中的传输层所用的传输选择器等 同于端口。RTP 依赖更低层来提供例如端口机制来在会话中提供多播的 RTP 和 RTCP 报文。
Transport address ： 网络地址与端口的组合 ，用来识别传输级的终点，例如一个 IP 地址和一个 UDP 端口。报文是从源端传送到目的端的。
RTP media type ： 一个 RTP media type 是可以在单个的 RTP 会话中携带的 payload 类型的合集。由 RTP Profile 根据 RTP payload 类型指定 RTP media types。
Multimedia session：普通组参与者的一系列并发的 RTP 会话。例如视频会议中可能包含一个音频 RTP 会话和一个视频 RTP 会话。
RTP session ： 一组参与者利用 RTP 来通信的组合。一个参与者可 以同时加入几个 RTP 会话中。在一个 多媒体会议中，除非编码将多个 媒体编入单数据流中，否则每个媒介都会独立的在自己的 RTP 会话中传送自己的 RTCP 报文。参与者利用不同的目的传输地址对来区分就收到的不同的 RTP 会话，其中传输地址对包括一个网络地址和一对 RTP 和 RTCP 端口。在一个 RTP 会话中的所有参与者分享同一个目的传输地址对，如同 IP 多播，而不像单波网络中每个参与者的地址和端口对各不 相同。在单播的情况，参与者可能与其他者共享同一对端口或者每两个人用一对 端口。识别每一个 RTP 会话是利用两个 RTP 中的 SSRC identifier 空隙。在一个 RTP 会话中的一组参与者都有具有接收任何参与者的 RTP 中的 SSRC identifier 或是 RTCP 中的 CSRC。例如，考虑三方会议的情况， 每两个人都用一对不同的单播 UDP 端口。如果其中的一位与会者给仅另一个与会者发送 RTCP 反馈，这样会议就由三的独立的点到点的 RTP 会话组成。如果其中的一个与会者将他接收到的另一个与会者的数据所做的 RTCP 反馈给另外两个人，这时会议就是由一个多方 RTP 会话组成。后面的例子模拟了 在 IP 多播通信时三方会议会发生的情况。

RTP 协议框架在这里定义一些改变，但是某种特定协议的或是应用设计会限制这 些定义的变化。

Synchronization source (SSRC)：RTP 报文流的一个源，由 RTP 头中定义 的 32-bit 的 SSRC identifier 来标识，这样做是为了不依赖网络地址。所有从 同一个同步源出来的报文都具有相同的时序和序列号间隔，这就使得一个接收组可以

依靠这些同步源来进行重放。同步源可以是报文流的发送者，这些报文是从 麦克风或是相机得到的信号源或是一个 RTP mixer。同步源可能会改变数据的格式，例如，音频编码。SSRC identifier 在特定的 RTP 会话中必须是全局的随机值。参议这不必在多媒体会话中为了所 有 RTP 会话使用同一个 SSRC identifier；SSRC identifiers 是由 RTCP 分配的。在一个会话中，如果参与者中从不同的视频源端生成多个流， 那么每个必须标识为不同的 SSRC。

Contributing source (CSRC)：RTP 报文的一个 source，对由 RTP Mixer 所输出的混合信号有作用。Mixer 在 RTP 头中插入一系列的 SSRC identifier，用来生成某种特定的报文。这一系列的就叫 CSRC list。如音频会议中，Mixer 标识出输出的报文由哪个讲话者的构成，这就使得就收这可以知道现在讲话者是谁，即使所有的数据包都包含相同的 SSRC identifier。

End system：一种应用，产生或是接收在 RTP 报文中传送的目录。在 特定的 RTP 会话中，一个终端系统可以充当一个或者几个同步源，但是一般情况下是一个。

Mixer：一个中介系统，从一个或几个源端接收报文，用某种方式合成然后输出 新的 RTP 报文，这中间可能会改变原来报文的格式。不同输入源端来的数据时序可能不同，Mixer 会做一些调整产生自己的时序。所有从 Mixer 输出的数据包将会标识 Mixer 作为它们的同步源。

Translator：一个中介系统，前向输出 RTP 报文。translator 包括没有经过混合的转换编码设备，从多播到单播的复制品，和应用级的防火墙滤波器。

Monitor：在 RTP 会话中接收由发送者发送的 RTCP 报文的应用，特别是接收报告， 为分发监测估计当前的 QOS，错误诊断和长期的统计。监视器功能可以集成进 参加会话的应用中，但是也可以是独立的不另外参加或发送接收 RTP 数据报文的应用。这叫做第三方的监视器。也有这样的情况，在会话中第三方的监视器接收 RTP 数据报文但是不发送 RTCP 报文或是其它的被希望的。

Non-RTP means：为了提供一个适用的服务而额外附加的协议或机制。特别是在多媒体会议中，需要一种控制协议来分发多播地址和密钥来加密，协调所用的加密算法，在 RTP payload type 值与 payload 格式之间定义动态的映射，这样的 例子包括会话初始协议，ITU 建议的 H.323 [14] 和应用 SDP 的应用，例如 RTSP。对于一个简单的应用，可以应用电子邮件或是会议数据数据库。更详细的这样的协议或机制不在本文档之内。

4.字节序，校正，时间格式

网络字节序携带所有的整数域，即，有用的字节为首。在有更详细的传 输的顺序描述。不像其它的常数是十进制的。

所有的头数据都被指派它的自然长度，即，16-bit 域指派两个偏移，32-bit 域所指派的必须能被 4 整除等。用来填充的字节取值为零。

绝对日期和时间是用网络时间协议(NTP)的时间戳格式来 标识的，秒级的与 0h UTC on 1 January 1900 [4] 有关。NTP 的时间戳使用 64-bit 无符号的定点表示，整数部分用前 32-bit，小数部分用后 32-bit 来表示。在一些情况下，可以用一种更简洁的表示法，即用中间的 32-bit，也就是说，低 16 位表示整数，高 16 位表示小数，高 16 位的整数位必须独立确定。

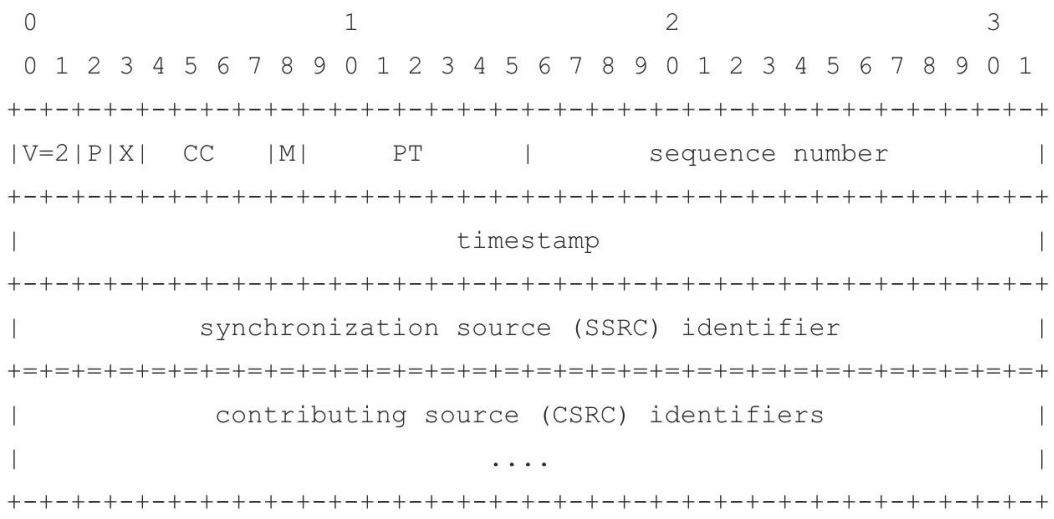
运行网络时间协议不需要其他的协议。也可以用其它的时间源或者根本不需要。运行 NTP 对来自不同的主机的同步流有益。

NTP 时间戳到 2036 年就会循环回零的状态，但是对于 RTP 来说，仅仅是利 用了 NTP 时间戳的不同取值。只要时间戳对之间在 68 年内，用模块化的构架来 减或比较就不用考虑 NTP 循环的情况。

5. RTP 数据传输协议

5.1 RTP 定长的头字段

RTP 头格式如下：



每一个 RTP 包中都有前 12 个字节，而 CSRC identifier 列表仅在 Mixer 插入时才有。域解释如下：

version (V)：2 bits

RTP 版本标识。说明中定义了两个版本

padding (P)：1 bit

如果设定 payload，在报文的末端就会包含一个或者多个填充字节，这不属于有效载荷。最后一个字节的填充有一个计数器，标识需要忽略多少个填充字节。一些加密算法可能需要固定块长度的填充，或者是为了在更低层数据单元中携带一些 RTP 报文。

extension (X)：1 bit

如果设定了拓展位，定长头字段后面会有一个头扩展

CSRC count (CC)：4 bits

CSRC count 标识了定长头字段中包含的 CSRC identifier 的数量。

marker (M)：1 bit

marker 是由一个轮廓定义的。用来允许标识在像报文流中界定帧界等的事件。一个轮廓可能定义了附加的标识位或者通过修改有效载荷类型域中的位数量来指定没有标识位。

payload type (PT)：7 bits

这个字段定一个有效载荷的格式和在应用中定义解释。轮廓可能指定一个从有效载荷格式码字到有效载荷格式的默认静态映射。也可以通过 non-RTP 方法来定义附加的有效载荷的格式码字。在 RFC 3551[1]中定义了一系列的默认音视频映射。一个 RTP 源有可能在会话中改变有效载荷的格式，但是这个域在复用独立的媒体时是不同的。接收者必须忽略它不识别的有效载荷的格式。

sequence number：16 bits

每发送一个 RTP 数据报文序列号值加一，接收者也可用来检测丢失的包或者重建报文序列。初始的值是随机的，这样就使得已知明文攻击更加困难，即使源并没有加密，因为要通过的 translator 会做这些事情。。

timestamp：32 bits

timestamp 反映的是 RTP 数据报文中的第一个字段的采样时刻的时间瞬时值。采样时间值必须是从恒定的和线性的时间中得到以便于同步和抖动计算必须保证同步和测量保温抖动到来所需要的时间精度。时钟频率是与 payload 所携带的数据格式有关的，在轮廓中静态的定义或是在定义格式的有效载荷格式中，或通过 non-RTP 方法所定义的有效载荷格式中动态的定义。如果 RTP 报文周期的生成，就采用虚拟的采样时钟而不是从系统时钟读数。例如，在固定比特率的音频中，时钟会在每个采样周期时加一。如果音频应用中从输入设备中读入 160 个采样周期的块，时钟就会每一块增加 160，而不管块是否传输了或是丢弃了。

SSRC：32 bits

SSRC 域识别同步源。为了防止在一个会话中有相同的同步源有相同的 SSRC identifier， 这个 identifier 必须随机选取。生成随机 identifier 的算法。虽然选择相同的 identifier 概率很小 ，但是所有的 RTP 实施必须检测和解决冲突。描述了冲突的概率和解决机制和 RTP 级的检测机制，根据唯一的 SSRC identifier 前向循环。如果有源改变了它的源传输地址，就必须为它选择一个新的 SSRC identifier 来避免被识别为循环过的源。

CSRC list：0 to 15 items，32 bits each

CSRC list 表示那些在本报文中对 payload 作了贡献的源。号数是由 CC 域定的。如果有多于 15 个贡献源，只有 15 个源可以被标识。CSRC identifier 是由 Mixer 利用贡献源的 SSRC identifier 插入的。例如，对于音频报文，所有混合在一起的源的 SSRC identifier 被例出来，以便就收者识别出正确的讲话者。

5.2 RTP 会话的多路复用

为了协议处理更有效率，复用点的数量应最小化，在 RTP 中，复用技术 是由目的传输地址(网络地址和端口号)提供的。例如，在音频与视频独立编码的 远程电信会议中，每一个媒介都是由单独的带有自己的目的传 输地址的 RTP 会话来携带的。

分开的音频和视频流不应该在一个 RTP 会话中携带也不应该基于有效载荷类型或 SSRC 域来解复用。不同的 RTP 媒体但是用同样的 SSRC 较差的报文会产生几种问题：

1。 假设，两个音频流公用相同的 RTP 会话，具有相同的 SSRC 值，其中的一 个改变了编码因此需要一个不同的 RTP 有效载荷类型，就是不能识别出到底是 哪个流改变了编码。

2。 SSRC 是用来识别一个单时序和序列值间隙的。如果媒体时钟速率不同，交叉的多有效载荷类型需要不同的时隙和不同的序列间隙来通知哪个有效载荷类型承受了损失。

3。 RTCP 发送者和接受者报告在每个 SSRC 仅能描述单时序和序列号间隙， 没有携带有效载荷类型域。

4。 RTP Mixer 不能将不兼容的交叉 媒体流合并成单一流。

5。 在一个 RTP 会话中携带多个媒体流就不会出现下面几种情况：利用不同 的网络路径或者分配的网络资源；希望接收子媒体，例如视频超出了带宽的承受 能力而至接收音频，接受方对不同的媒体进行不同的处理，而用独立的 RTP 会话 就可以执行单或多媒体处理。

每一个媒体用一个 SSRC 但是在一个 RTP 会话中发送它们可以避免上述的前 三种情况，但是不能解决 后两种情况。

另一方面，在一个 RTP 会话中具有不同 SSRC 值的同一媒体的相关源的多路 复用对多播会话来说是很标准的。上述问题就不会出现：例如，RTP 混合可以 合并多个媒体音频源，对它们进行同样的处理。这对具有不同 SSRC 值得相同媒 体的复用也是很合适的，这样就不存在后两个问题了。

6. RTP 控制协议--RTCP

RTP 控制协议是基于在会话中的对所有参与者周期传输的控制报文 的，与数据报文使用相同的分发机制。协议必须提供对数据报文和控制报文的复用，例如用 UDP 的不同端口号。RTCP 有下面四个功能：

1。 最基本的功能是对提供分发数据的质量反馈。这是 RTP 作为一个完整的 传输协议所必须的部分，与其它的提供流或拥塞控制的传输协议有关。反馈对控制自适应编码有直接的影响。进一步的 IP 多播实验表明它对从接收者的反馈的作出分发错误诊断也很重要。将就受报告反馈给所有人就使得关注问题的人知道问题时本地的或是全局的。利用像 IP 多播的分发机制，可以使像服务提供者这样的实体收到反馈和 作为第三方监测诊断网络问题的所在。这种反馈的功能有 RTCP 发送方和接收者报告实现。

2。 RTCP 为 RTP 源携带一个稳定的传输级的标识符，叫规范名称或 CNAME，。因为 SSRC identifier 可能因为发现了冲突或是程序重起而改变，接受者需要 CNAME 追踪每一个参与者。接受者也可以用 CNAME 来协调从一个参与者而来的一系列相关 RTP 会话的数据流，例如同步音频和视频。跨媒体同步需要 NTP 和 RTP 时间戳包含在发送者发送的 RTCP 报文中。

3。前两个功能需要所有参与者都发送 RTCP 报文，因此必须控制速率使得 RTP 可以让各夺得参与者加入。每个人都对所有人发送他的控制报文，这就使得 人人都可以独立的观察参与者的数量。这个数量用来计算发包的速率。

4。 这个功能是可选的，传达最小的会话控制信息，例如在用户界面显示参与者的身份。这最有可能用在 "松散控制" 会话中，因为参与者不需要身份控制或是参数传递就可以进入或退出。RTCP 扮演一个直达接受者的便利通道的角

色，但是不需要支持所有应用需要的控制通信的全部要求，需要一种更高一级的会话控制协议，这不在本文当讨论范围之内。