

组织：中国互动出版网 (<http://www.china-pub.com/>)

RFC 文档中文翻译计划 (<http://www.china-pub.com/compters/emook/aboutemook.htm>)

E-mail: ouyang@china-pub.com

译者：李超 (licc_li licc_li@sina.com)

译文发布时间：2001-5-23

版权：本中文翻译文档版权归中国互动出版网所有。可以用于非商业用途自由转载，但必须保留本文档的翻译及版权信息。

Network Working Group

Request for Comments: 2198

Category: Standards Track

C. Perkins

I. Kouvelas

O. Hodson

V. Hardman

University College London

M. Handley

ISI

J.C. Bolot

A. Vega-Garcia

S. Fosse-Parisis

INRIA Sophia Antipolis

September 1997

用于冗余音频数据的RTP负载格式

(RFC 2198 RTP Payload for Redundant Audio Data)

本备忘录的状态

本档讲述了一种Internet社区的Internet标准跟踪协议，它需要进一步进行讨论和建议以得到改进。请参考最新版的“Internet正式协议标准”(STD1)来获得本协议的标准化程度和状态。本备忘录的发布不受任何限制。

摘要

本文描述了一种在使用实时传输协议(RTP版本2)时对冗余音频数据进行编码的负载格式。在此提出这套机制的主要目的是为了开发针对包易丢失网络(如Internet Mbone)的音频会议工具。尽管如此，该机制并不局限于此类应用。

目录

本备忘录的状态.....	1
摘要.....	1
1. 介绍.....	2
2. 需求与动机.....	2
3. 负载格式说明.....	3
4. 局限性.....	4
5. 同 SDP 的关系.....	5
6. 安全性考虑.....	5
7. 示例.....	6
8. 作者地址.....	7
9. 参考文献.....	7

1. 介绍

随着Internet Mbong团体间多媒体会议得到更广泛的应用，用户必定会进一步认识到，大多数应用都要求服务提供相当好的质量。我们知道有很多因素都会影响到会议的质量，其中最明显的就是包丢失问题。这个问题已经持续多年，并随着Internet的普及以及由此带来的负载增加而变得更加尖锐。即便是丢包率很低的情况下对语音理解性的破坏也会导致人们对Internet多媒体会议的可行性产生怀疑。数据流冗余就是作为该问题的解决方案之一而提出的[1]。在平均连续丢包率很低的情况下，如果一个包丢失了，则接收方还可通过后续包中的冗余数据对失去的信息进行重组和恢复[2]。最近的工作[4][5]显示，针对当前Internet上的若干种包丢失模型，该机制都可以很好地工作。

本文描述了用于对冗余编码的音频数据进行传输的RTP负载格式。第二节说明了定义这种负载格式的需求和动机，并未定义其具体形式。第三节定义了冗余音频数据的RTP负载格式。

2. 需求与动机

RTP应用中对冗余编码机制有如下需求：

- 每个包必须携带一个主编码和一个或多个冗余编码。
- 由于对冗余信息可以采用多种编码形式，每个冗余编码块都必须有一个编码类型标识符。
- 由于可能采用变长编码，每个编码后的块都必须有长度指示符。
- RTP头提供时间戳字段表示编码数据的创建时间。当使用冗余编码时该字段可以参考主编码数据的创建时间。冗余数据块与主数据可能在时间上会有一些间隔，因此每个冗余编码块都要有自己的时间戳。为了减少时间戳字段占用的字节数，可用冗余编码和主编码时间戳的差值来进行编码。

为标准RTP规范增加冗余音频扩展有两个基本的方法：一个包含有冗余的扩展头，或者定义一个或多个额外的负载类型。

通过将所有的冗余信息放在扩展头中，那些不需要实现冗余的应用程序就可以轻松地丢弃该头而专注于处理主编码数据。

不过，这套机制也有一些弊端：

- 大量的额外负担，扩展头占用的4个字节和可能多达3个字节的扩展尾填充（为满足

4字节边界的要求)。对很多应用都无法接受这么大的负担。

- 使用扩展头限制应用程序只能使用一种冗余编码，除非引入更多的结构。这同样也会造成更多的负担。

基于上述原因，我们放弃了使用RTP扩展头的方式来实现音频冗余编码的方法。

RTP音视频会议框架列出了一系列的负载类型并为会议控制协议定义新的编码类型提供了一个可容纳32种编码的动态范围。因此，冗余音频应用可以采用下面两种方法来分配额外的RTP负载类型：

1. 定义一个动态的编码机制，对每个主/冗余组合的负载类型均应用RTP动态负载类型范围。
2. 定义一个固定的负载类型来表示有冗余的包。它既可以分配给一个静态RTP负载类型也可以进行动态分配。

可以为所提供的32个动态负载类型中的每种类型都定义一个可标识特定编码组合的负载类型集合。这样做可能造成一些限制，对那些只有一个冗余块的包是可行的，因为这样的包的组合数并不多。而多冗余块的需求使得编码组合数大大增加，这种方法就无法适用了。

对上面方法的一个改进版本就是，在会议开始前从32种编码组合的集合中选择决定会议期间使用那种方法。会议中的所有工具都可以用这套编码组合工作集来进行初始化。工作集之间的通信通过使用外部的，带外机制来实现。由于用同样的参数来启动工具时要格外小心，所以安装设置的过程十分复杂。这个方案只用一个字节来鉴别编码组合，因此比前者更有效。

然而，在将负载类型映射到冗余数据组合的分配过程中所固有的复杂性可能会导致人们放弃使用这种机制。

此外还有一种更灵活的方法，就是以负载类型来表示有冗余的包。于是该包就成为一个容器，在其中可封装多个负载。这样的方法可以把任意数量的冗余数据封装到一个包中，因此使用十分灵活。当然，每个封装后的负载都要有一个头来表示所包含的数据类型，这也会带来一点小小的负担。但总之这还是一个比较好的方案，它兼具灵活性和扩展性，同时负担也相对较小。本文的剩余部分就将着重描述这种方法。

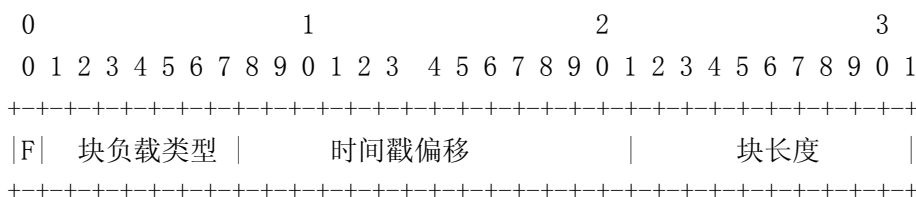
3. 负载格式说明

本文中的关键字“必须”，“必须不”，“要求的”，“应该”，“不应该”，“会”，“不会”，“建议”，“或许”，“可选的”在 RFC 2119 中解释。

为新的包格式分配 RTP 负载类型已超出本文范畴，不在此赘述。特定类型应用程序的 RTP 框架应该负责为编码分配负载类型，如若不能则应在动态范围内选择一个负载类型。

一个承载了冗余数据的RTP包应该有一个标准RTP头，同时要在负载类型中表示其中含有冗余信息。头中其它字段与主数据块相关。

紧接着RTP头是一些附加头，定义于下图中，它们规定了包所携带的每个编码的内容。此后是数据块，其中包括了这些编码的标准RTP负载数据。注意到所有的头都要同32位边界对齐，但负载数据却往往不能对齐。如果一个包中携带了多个冗余编码，则它们应对应不同的时间段：没必要为包的一个时间段制作多个数据拷贝。



头中的各位定义如下：

标志位 (F) : 1位, 头中的第一位, 表示后面是否还有另一个头块。如果该位为1表示后面还有头块, 如果该位为0表示这是最后一个头块。

块负载类型 (block PT) : 7位, 表示该块的RTP负载类型。

时间戳偏移 (timestamp offset) : 14位, 本块相对于RTP头时间戳的无符号时间戳偏移量。使用无符号偏移则说明冗余数据的发送必须在主数据已经发送之后, 因此要从当前时间中减去主数据的发送时间来决定冗余数据所在块的时间戳。

块长度 (block length) : 10位, 表示对应数据块的字节长度, 其中不包括头的长度。

我们注意到采用无符号偏移对使用冗余数据起了一定的限制: 即不可能在发送主编码前发送冗余信息。这将对某些方法产生影响, 因为一些适于冗余的低带宽编码可在编码过程的早期产生, 从而可以提前发送。但是, 额外的符号位会造成时间戳偏移范围减少, 这点是不可接受的。同时增加字段长度超过14位也限制了块长度字段。由此看来, 限制冗余数据必须在主数据之后发送所带来的问题比起限制其它字段的长度而言要少一些。

冗余块的时间戳偏移是由同一单元中的主数据时间戳来度量的(如: 音频采样, 与主数据保持同样的时钟频率)。这点说明冗余编码和主编码的采样频率必须相同。

我们还要注意到, 块长度和时间戳偏移分别为10位和14位, 而不是常见的8位和16位。这样的编码有时使对包头的解析变得比较复杂, 也造成了一些额外的处理负担, 但使用通常的方式会造成很多问题: 一个8位块长度字段对大多数情况下的编码都是足够的, 但并非全部, 例如: 一段80ms的PCM和DVI音频包长度超过256字节, 不能编码到单字节长度字段。当然也可以在块长度字段中增加额外的结构(例如可以用高位为1表示低7位为按字长度编码而非字节), 但这样处理方式上十分复杂。而使用10位块长度字段在牺牲了时间戳值范围的情况下既保留了处理简单性, 也提供了更大的长度范围。

主编码块头位于包的最后。我们可以忽略本块头中的时间戳和块长度字段, 因为他们可以通过RTP头和整个包的长度来判断。主数据块的头由一个零F位和一个块负载类型组成, 总共8位。如下图所示:

```

0 1 2 3 4 5 6 7
+-----+
|0|   Block PT   |
+-----+
```

最后一个头之后就是数据块, 存储顺序和头的排列顺序相同。数据块之间不需要填充或者使用其它分隔, 一般不需要32位对齐。如此选择仍是为了在损失一定额外解码时间的情况下降低带宽负担。

编码的选择应该反映其对带宽的需求。冗余编码所占带宽应远远小于主编码所占带宽: 然而该原则也有些例外, 即如果主编码本身带宽就很小, 且需要很高的处理能力, 则往往使用主编码的拷贝来作为冗余。即便如此, 冗余编码绝不能比主编码的所占带宽高。

一般情况下没必要使用多级冗余。在某些需要多级冗余情况下, 每层的带宽需求都要明显低于前一级。

4. 局限性

RTP标志位并非是为冗余数据块而保留的。因此, 如果主数据丢失(其中包括标志位), 则标志位也会同时丢失。但我们认为它并不会造成什么大麻烦: 因为即使标志位同冗余信息一起传输也有丢失的可能, 所以在编写应用程序时应该充分考虑到这些。

另外, CSRC信息也不是为冗余数据保留的。冗余音频包RTP头中CSRC数据只同主数据有关。由于CSRC数据相对较少发生变化, 因此我们建议需要此信息的应用程序可假定RTP头中的

CSRC数据能够用于重建冗余数据。

5. 同 SDP 的关系

使用冗余负载时必须将其绑定到一个动态负载类型。这一过程可以通过带外机制来实现，不过更通用一些的办法就是利用SDP[6]协议来进行关于绑定的通信。SDP定义了一套机制用于将动态负载类型绑定到特定的编解码器、采样率、和多个使用”rtpmap”属性的通道。下面就是一个使用RTP视听框架[3]的例子：

```
m=audio 12345 RTP/AVP 121 0 5
a=rtpmap:121 red/8000/1
```

上例说明一个RTP音频流正在使用负载类型121（一个动态负载类型），0（PCM u-law）和5（DVI）。“rtpmap”属性用于将负载类型121绑定到编解码器”red”，表示该编解码器是一个冗余帧，采样率8KHz，且是单声道的。当与SDP一同使用时，术语”red”表示本文中所讨论的冗余格式。

为此我们规定了PCM和DVI的附加格式。因此接收端必须为使用这些格式做好准备。这一规定意味着在缺省情况下发送方可以发送冗余，也可以发送PCM或DVI。但对于冗余负载，我们顺带提出这点意味着只能使用PCM或DVI作为冗余编解码。注意到”m=”字段中的定义的附加负载格式本身也可能是动态负载类型，而一旦如此，就需要一些附加属性”a=”来描述这些动态负载类型。

接收一个冗余流所需的一切就是如此。不过要发送一个冗余流，发送方得知道建议使用的主编码和第二编码（如tertiary）。该信息对于冗余格式是明确的，并通过使用附加属性”fmp”来指定，该属性传达了格式特定的信息。一个会话目录并不解析fmp属性的值，而仅仅是将它转交给媒体工具。为了冗余性，我们定义了RTP负载格式的格式参数列表，通过斜线”/”分隔开。

完整的例子如下：

```
m=audio 12345 RTP/AVP 121 0 5
a=rtpmap:121 red/8000/1
a=fmp:121 0/5
```

上例说明发送方缺省模式为冗余，其中PCM作为主编码，而DVI作为第二编码。除非该编码已在媒体行（“m=”）中指定为有效，否则不能在fmp属性中指定编码。

6. 安全性考虑

包含冗余数据的RTP包从属于RTP规范[2]以及任何适用的RTP框架（如[3]）所讨论的安全性考虑。这意味着媒体流的安全性要通过加密来达到。对冗余数据进行加密可通过下面两种方法实现：

1. 对整个流进行加密，所有的参与者都必须拥有密钥才可进行解密。在这种情况下，加密采用通常的方式来进行，无须什么特别的操作。
2. 流的一部分和其余部分以不同的密钥进行加密。采用这种方式，最后一个包的冗余拷贝就无法进行发送，因为已经没有后续包能采用正确的密钥进行加密。类似的限制也出现于使能和禁止加密的过程中。

从两者中具体选择哪种方式进行加密是编码者的问题，而解码者应可以无须修改而对任何一种形式进行解密。

音频流的低带宽冗余对于解决包丢失的保护问题是一种很有效的方法，与此同时，应用设计者也应该注意到，大量冗余数据会造成网络拥塞的增加和加剧包丢失现象，这将使采用冗余数据试图解决的包丢失问题变得更糟。在最极端的情况下，还会导致网络拥塞过度，甚至形成拒绝服务攻击。

7. 示例

一个RTP音频数据包，包括一个DVI4(8KHz)主编码块和一个单独的8KHz LPC编码的冗余块，两者长度均为20ms。参照RTP视听框架[3]所定义，如下所示：



8. 作者地址

Colin Perkins/Isidor Kouvelas/Orion Hodson/Vicky Hardman

Department of Computer Science

University College London

London WC1E 6BT

United Kingdom

EMail: {c.perkins|i.kouvelas|o.hodson|v.hardman}@cs.ucl.ac.uk

Mark Handley

USC Information Sciences Institute

c/o MIT Laboratory for Computer Science

545 Technology Square

Cambridge, MA 02139, USA

EMail: mjh@isi.edu

Jean-Chrysostome Bolot/Andres Vega-Garcia/Sacha Fosse-Parisis

INRIA Sophia Antipolis

2004 Route des Lucioles, BP 93

06902 Sophia Antipolis

France

EMail: {bolot|avega|sfosse}@sophia.inria.fr

9. 参考文献

- [1] V. J. Hardman, M. A. Sasse, M. Handley and A. Watson; Reliable Audio for Use over the Internet; Proceedings INET'95, Honolulu, Oahu, Hawaii, September 1995. <http://www.isoc.org/in95prc/>
- [2] Schulzrinne, H., Casner, S., Frederick R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.
- [3] Schulzrinne, H., "RTP Profile for Audio and Video Conferences with Minimal Control", RFC 1890, January 1996.
- [4] M. Yajnik, J. Kurose and D. Towsley; Packet loss correlation in the Mbone multicast network; IEEE Globecom Internet workshop, London, November 1996
- [5] J. -C. Bolot and A. Vega-Garcia; The case for FEC-based error control for packet audio in the Internet; ACM Multimedia Systems, 1997
- [6] Handley, M., and V. Jacobson, "SDP: Session Description Protocol (draft 03.2)", Work in Progress.
- [7] Bradner, S., "Key words for use in RFCs to indicate requirement levels", RFC 2119, March 1997.