基于实时传输协议的丢包实时修复*

张 钶, 谢忠诚, 鞠九滨

(吉林大学 计算机科学系,吉林 长春 130023)

E-mail: jjb@mail.jlu.edu.cn

http://www.jlu.edu.cn

摘要: 在诸如IP电话和远程会议系统等实时应用中,使用实时传输协议RTP(real-time transport protocol)在因特网上传输的数据包不可避免地会丢失,极大地影响了传输服务质量.在RTP上增加丢包修复功能可以解决这个问题.介绍了在RTP上采用FEC(forward error correction)修复丢包的方法,基于这个方法设计并实现了一个支持丢包修复功能的RTP库,说明了丢包修复功能的测试方法和结果.

关键词: RTP(real-time transport protocol);丢包;修复;FEC(forward error correction)

中图法分类号: TP393 **文献标识码**: A

话音包与数据包在因特网上传输时可能有较大的延迟和丢包概率.丢包将严重地影响IP电话的话音质量[1],因此采取丢包修复方法是很有必要的.造成媒体包丢失的原因有这样几个:路由器和网关发生拥塞,某些包可能被丢弃;包传输过程中的延迟过大,因到达接收端太晚而无法回放;在多任务操作系统中,工作站超载运行时可能因调度困难而不能及时处理传输.

目前有4种常见的丢包修复方法[2].重发法是修复TCP丢包常用的方法,但语音数据对时延敏感,只有当重发延迟满足实时应用的要求时,重发法才是适用的[3].冗余法(媒体相关的FEC(forward error correction)法)[1]是把第n个数据包的副本重新用更低位码率的压缩算法,如GSM(global system mobile)语音压缩,附加到第$n+1$个数据包中一起发送,如果第n个包丢失了,可以从第$n+1$个包中所附带的副本来恢复;如果附带多个副本,还可以修复连续丢包.这种方法需要额外的CPU及带宽开销,并且只有大致的修复能力.媒体无关的FEC[2]法有精确的修复能力,发送端对连续多个数据包作异或运算求出FEC数据包,并以奇偶包的形式发送,丢失的数据包可以用接收到的数据包和FEC包作异或运算再生出来.此方法的缺点是与冗余法相比有较大的延迟(缓存多个包以作FEC运算),增加了带宽需求.交叠法[2]和差错分散法(error spread)[4]降低或分散了丢包的影响,它们的优点是不增加带宽需求,但在传输两端中增加缓冲延迟开销.几类修复方法的性能对比见表1.由此可见,媒体无关FEC法和冗余法适合实时传输应用.

对丢包现象的大量研究发现,丢失单包的概率最大,因此丢包修复主要解决丢失单包的修复问题.

IP电话和远程会议系统,从早期的Vat,NeVoT,到后来基于H.323[5]协议的NetMeeting和IP电话,都把实时传输协议RTP(real-time transport protocol)[6]作为传送实时媒体的工具.RTP

* 收稿日期: 1999-12-23; 修改日期: 2000-03-23

基金项目: 高等学校博士学科点专项科研基金资助项目(1999018319)

作者简介: 张钶(1971-),男,山东泰安人,博士,主要研究领域为计算机网络,分布式系统;谢忠诚(1972-),男,辽宁阜新人,硕士,主要研究领域为计算机网络,分布式系统;鞠九滨(1935-),男,黑龙江哈尔滨人,教授,博士生导师,主要研究领域为计算机网络,分布式系统.

已经成为实时媒体传输的通用标准,被 H.323 系列标准定义在 H.225.0[7]的附录中.RTP 还包括一个实时控制协议 RTCP(RTP control protocol),用于进行松散的对话控制,提供服务质量 QoS (quality of service)报告和支持媒体同步等.

Table 1 Overheads of different lost repair schemes

表 1 不同修复策略开销的对比

	Delay①	Bandwidth overhead②	Processing overhead③
Retransmission④	Medium⑤	Variable⑥	High⑦
Interleaving⑧	High	None⑨	Low⑩
Media-Independent FEC⑪	High	High	Low
Redundancy⑫	Small⑬	Variable	Variable, may be large⑭

①延迟,②带宽开销,③处理开销,④重发法,⑤中,⑥变化,⑦高,⑧交叠法,
⑨不变,⑩低,⑪媒体无关 FEC 法,⑫冗余法,⑬小,⑭变化,可能很大.

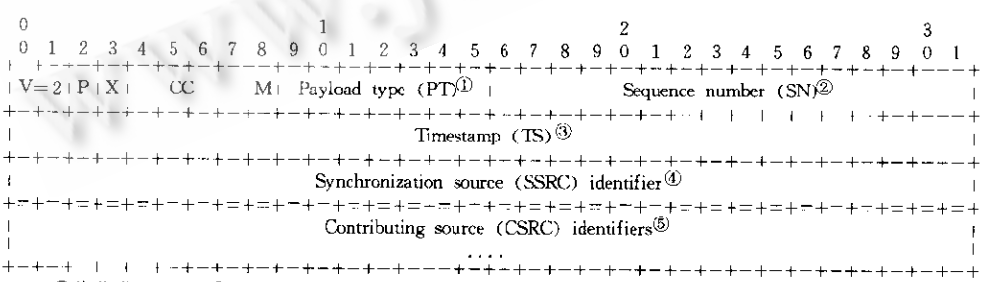
实时媒体传输中的丢包检测和修复需要 RTP 的支持,例如通过包序列号监测丢包,把 RTP 的传输实时媒体的功能和丢包修复能力结合在一起,将是实时媒体传输中解决丢包问题的一个较好的方法.实现 RTP 协议以及基于该协议的工具最早有 RTP Tools[8]和 RTPmon[9],用于处理 RTP 数据包和监控 RTCP 报告.RTP Library[10]提供了基本的 RTP 协议功能.但支持丢包修复的 RTP 工具还处于研究阶段,目前还未见报道.

本文介绍了我们设计和实现的一个具有丢包修复功能的 RTP 库.它采用与媒体无关的 FEC 法实时修复所有丢失的单包和部分丢失的多包.同时,并在我们设计和实现的测试平台上验证了这项功能.

1 实时传输协议(RTP)

为了说明在 RTP 上利用 FEC 法对丢失的包进行修复的原理,有必要先来介绍 RTP 的有关细节.

RTP 传送音频和视频等连续的媒体数据.RTP 数据包由一个 12 字节的包头(如图 1 所示)和传送的净载数据组成.在包头中:净载类型域标识出 RTP 净载数据格式,以便接收端确定相应的处理程序;时间戳反映出在 RTP 数据包的净载中第 1 个字节的采样时刻,该域使媒体数据在接收方能够按时间序列重新组装;序列号代表发送方发出的包序列,它可以使接收者监测丢包和重建包序列;指示器位使一些重大的事件(如对音频数据标识出一个"交谈进发"的开始)能在数据包流中被突出地标记出来;同步源标识符用于唯一地标识出 RTP 对话中的一个同步源.



①将载类型(PT),②序列号(SN),③时间戳(TS),④同步源(SSRC)描述符,⑤贡献源(SCRC)描述符.

Fig. 1 RTP header format

图1 RTP包头的格式

RTCP 包定期地在与数据包相同的多点传送组中传送,即使在没有媒体数据传送的情况下,它

们也可以作为对话成员的活跃指示器.RTCP 的下述功能与本文的工作直接有关:

(1) QoS 监控和拥挤控制.RTP 对话参与者使用发送报告 SR(sender report)和接收报告 RR (receiver report)提供接收质量反馈,使所有的对话成员都能够大致了解其他参与者的进展情况.

(2) 媒体同步.RTCP 发送报告中包含了实际时间和相应的 RTP 时间戳的指示器.这两个值允许不同媒体,如音频和视频之间实现"唇同步".

(3) 标识.RTCP 的 SDES 包中为每一个对话成员提供一个全局唯一的标识符,称为教名或 CNAME.SDES 的其他项提供发送者的其他描述信息,如姓名、电子邮件地址、地理位置等.

(4) 会话大小的估计和规划.每个对话成员定期发送 RTCP 包,当对话包含数百个与会者时,必须限制控制流量只占对话带宽的一小部分,通常建议分配给 RTCP 的带宽占总对话带宽的 5%.

发送者可以基于该反馈修改它的传输策略,接收者可以确定出现的问题是局部的还是全局的,网络管理员也可以使用只接收 RTCP 包的监控器来评价它的网络性能.SR 包含媒体同步所需的信息以及累计发送包数和字节数,这样,接收者可以估计出实际的数据流量.RR 包括接收到的最大序列号、丢包数、平均延迟抖动和用于计算往复延迟的时间戳.通常情况下,RTP 在 UDP 协议上运行,但这是以不能保证数据完整性为代价的.

2 用媒体无关前向误差校正法(FEC)修复丢包

2.1 校验编码

定义 $f(x,y,\ldots)$是应用在数据块 $x,y,\ldots$的按位异或操作.假定每个数据块是长度为 L 的一组简单比特流,则函数对所有数据块做异或运算后输出长度为 L 的校验块,特别地,$f(x)=x$.校验块一般由一组线性无关的数据块的组合来产生,这种组合称为校验编码.修复数据的方法一般是对 k 个媒体数据块产生 1 个或多个校验块,共有 n 个数据块和 $n-k$ 个校验块,接收方只要接收到其中任意 k 个块就可以再生出原来的 k 个媒体数据包.对于给定的 n 和 k,有以下多种检验编码(左面的字母代表输入的数据块流,右面的字母代表数据块和校验块流):

方式 0. 不能修复丢包,定义为 $a,b,c,d,\ldots\rightarrow a,b,c,d,\ldots$

方式 1. 能修复单个丢包,如果 b 和 $f(a,b)$交换位置,允许连续丢失两包(媒体+FEC 包)的修复.定义为 $a,b,c,d,e,f\rightarrow a,f(a,b),b,f(b,c),c,f(c,d),d,\ldots$

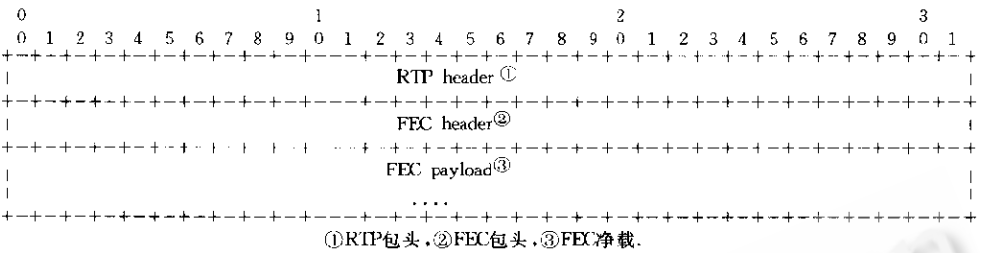
方式 2. 修复所有单包丢失和若干连续丢失的包,但它比方式 1 的代价要小:$a,b,c,d,e,f,g\rightarrow f(a,b),f(a,c),f(a,b,c),f(c,d),f(c,e),f(c,d,e),\ldots$

方式 3. 能修复连续丢失的 1、2、3 个包,但需要 4 个包的延迟:$a,b,c,d,e,f\rightarrow f(a),f(b),f(a,b,c),f(c),f(a,c,d),f(a,b,d),f(d),\ldots$

2.2 实现媒体无关 FEC 的 RTP 净载格式

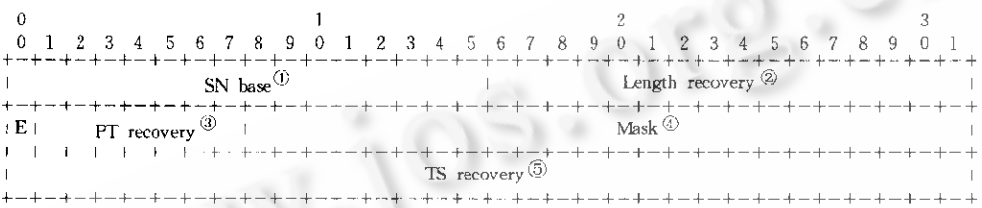
校验块要在单独的 FEC 包中传送.图 2 是一种封装在 RTP 数据包中实现媒体无关 FEC 的净载格式[11].FEC 包由在 RTP 净载部分的一个 FEC 包头和 FEC 净载组成.该格式可以传送任意长度和校验方式的数据块,可以对 RTP 净载和重要的 RTP 包头域如长度、净载类型等进行修复.校验包头(如图 3 所示)有 96 比特,其中定义了一个 24 比特的偏移屏蔽码域.如果第 i 个比特被置位 1,则表明序列号为 $N+i$ 的媒体包和本 FEC 包相关联,其中 N 是 FEC 包头中的序列号基域.FEC 包头中的其他域有:

· 长度修复域:决定修复后的媒体包的长度.例如,当两个长度为 3(0b011)和 5(0b101)个字的



①RTP包头,②FEC包头,③FEC净载.

Fig. 2 FEC packet structure
图2 FEC包结构



①序列号基,②长度修复域,③净载类型修复域,④偏移屏蔽码,⑤时间修复域.

Fig. 3 Parity header format
图3 校验包头格式

媒体包进行异或操作形成FEC包,那么长度修复域应当置为0b011 xor 0b101=0b110.

- 净载类型修复域:同本FEC包相关联的媒体包的净载类型值应用保护操作过程得到.
- 序列号基域:本FEC包保护的那些媒体包中最小的序列号值.
- 时间戳修复域:同本FEC包相关联的媒体包的时间戳值应用保护操作过程得到.
- E位:用来指示FEC头部扩展.

FEC包的净载部分是对本FEC包相关联的媒体包的CSRC列表、净载、填充部分和扩展部分应用保护操作得到的.所谓保护操作,是为了修复丢包发送端对媒体包执行的一种操作,其过程如下:第1步,从不同的媒体包头中取得各种域值、净载和填充部分(如果存在)附加在一起形成一个二进制数据块,如果这些数据块的长度不等,用0填充至与最长的数据块等长;第2步,对这些数据块做校验运算得出FEC数据块;最后把FEC结果数据放入FEC包中.其具体操作是把FEC数据块的第1个比特写入FEC包的填充标志位,把第2个比特写入FEC包的扩展标志位,将下面4个比特写入FEC包的CC域,再将下面1个比特写进FEC包的指示器位,将下面7个比特写入FEC包头的净载类型修复域,然后将下面32个比特写入FEC包头的时间戳修复域,将下面16个比特写进FEC包的长度修复域,剩下的比特都作为FEC包的净载.

2.3 丢包修复

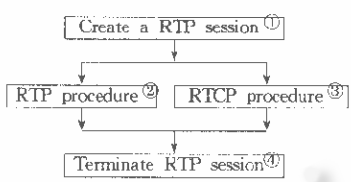
接收端检测到丢包后要启动修复过程,重建丢失的媒体包.修复过程分两部分:第1步根据实现者选择的算法,确定为修复丢失的媒体包必须具有哪些包(媒体包和FEC包);第2步重建丢失数据.假设T是修复媒体包X_i所需的媒体包和FEC包列表.丢包重建过程如下:(1) 对T中媒体包和FEC包执行保护操作的步骤1;(2) 对所有的二进制数组执行异或操作,产生修复数组;(3) 产生一个新的具有12字节长标准RTP包头的无净载RTP包;(4) 修复数组中的比特值依次填入新包中的相应域(第1个比特填充到标志位,第2个比特填充到扩展标志位,接下来的4个比特填充到CC域,...);(5) 新包序列号设置为X_i的序列号;(6) 把新包的SSRC值设置为其保护的媒体流的SSRC.以上过程将完全修复一个RTP数据包的包头和净载.

3 支持丢包修复功能的 RTP 库

H.225.0 中对 RTP 的原始定义做了少量修改,使用中一般遵从 H.225.0 中的规定而不是它的原始定义.参考上述二者的定义,我们开发了一个通用的 RTP 库.这个库在提供 RTP 基本功能的基础上支持基于媒体无关 FEC 功能的丢包修复机制.

3.1 RTP 的实现

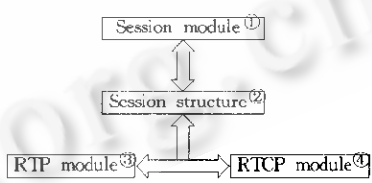
在一个 RTP 对话中,以相应的对话结构为中心实现对 RTP 数据流和 RTCP 控制流的产生和解析."对话结构(见下文中 RTPSessionInfo 的结构)"记录了对话中的重要信息,在对话进行过程中,结构中的许多信息不断被动态地更新,程序也根据各种信息的变化而采取相应的措施.对话的处理流程如图 4 所示.在新建对话过程中,调用者指定对话所需的基本元素(如净载类型、采样信息、CNAME 等),并得到一个对应本对话结构的对话句柄.此后,调用者可以根据应用的需要使用 RTP/RTCP 过程生成和解析媒体及控制包.在执行过程中,程序内部会通过调用者提供的对话句柄自动更新相应对话结构中的时间戳、序列号等信息,并且整理发送总包数、总净载字节数等统计数字.对话终止后,调用者必须关闭对话过程,释放对话所占用的句柄、内存等资源,保证程序的正常运行.RTP 协议库程序由 Session 模块、RTP 模块和 RTCP 模块这 3 部分组成(如图 5 所示).Session 模块的功能是建立、管理、更新 RTP 对话、信息获取和关闭 RTP 对话.RTP 模块执行 RTP 数据包和 FEC 包的打包、拆包、丢包检测与修复等工作.RTCP 模块的功能是 RTCP 包的打包和解析、获取各种统计信息.协议库运行在 UDP 协议上,数据包和 RTCP 包使用两个连续的端口,数据端口使用较低的偶数端口.在协议实现中遇到的一些问题,如 SSRC 冲突处理、随机数产生方法、指示器位定义、CNAME 确定等,均可参考协议的规定来解决.RTP 数据包中不包含长度域.由 UDP 包提供包长度标识.



①新建一个RTP会话,②RTP过程,③RTCP过程,④关闭该RTP会话.

Fig. 4 Flowchart of RTP session processing

图4 对话的处理流程



①Session模块,②对话结构,③RTP模块,④RTCP模块.

Fig. 5 Relationship of modules in RTP32 library

图5 RTP32协议库中各模块之间的关系

```
struct RTPSessionInfo
{
    unsigned int SSRC;                          //本地参与者的 SSRC
    unsigned char CName[STRINGLENGTH];          //CNAME
    unsigned int Timestamp;                     //时间戳
    unsigned short SequenceNumber;              //序列号
    unsigned short FECSequenceNumber;           //FEC 序列号
    unsigned char PayloadType;                  //净载类型
    short SampleRate;                           //采样率
    ...
    unsigned char FECPayloadType;               //FEC 净载类型
    unsigned short RTPVersion;                  //RTP 版本号
    unsigned char RealName[STRINGLENGTH];       //真实姓名
    unsigned char EMail[STRINGLENGTH];          //电子邮件地址
```

```
    ...
    unsigned short ParticipantNum;                  //与会者数
    struct RTPEndpointStatsEx * ParticipantList;    //与会者信息列表
    unsigned int PacketsSent;                       //本用户发送的总包数
    unsigned int OctetsSent;                        //本用户曾发送的总净载字节数
    ...
    BUFFER RTPPacketBuffer;                         //将要发送的RTP媒体包缓冲区
    BUFFER RTPFECBuffer;                            //将要发送的RTP FEC包缓冲区
    BUFFER RTCPPacketBuffer;                        //将要发送的RTCP包缓冲区
    struct ParsedRTPPacket * ParsedRTP;             //解析后的RTP包
    ...
}
```

3.2 丢包修复机制的实现

协议库中选择第2节中的方式1作为丢包修复的校验编码,使用图2中的净载格式传输FEC校验数据.方式1对于所有单包丢失及某些连续两包丢失的情况均可修复,而且计算复杂度低、延迟小.它唯一的缺点是需要增加1倍的带宽,但与其他方式相比,其性能价格比是比较好的.在RTP数据包的打包过程中采用上述校验编码和保护操作过程生成FEC包,每两个序列号连续的媒体包就有一个与之相关联的FEC包.接收方把所有接收到的FEC包都放到相应参与者的一个FEC队列中.由RTP媒体包的拆包过程维护该队列.当拆包过程发现当前收到的数据包的序列号同上一次收到的数据包的序列号之间的差值大于1时,则可断定发生了丢包现象.然后,启动丢包修复操作:

(1) 如果本参与者保存了前一个到来的媒体包,则转步骤(4);

(2) 新到的或新修复的包与上一次收到的包的序列号之差如果不大于1,则转步骤(8);

(3) 利用FEC队列和该媒体包修复丢包.如果成功修复,则转步骤(2),否则转步骤(8);

(4) 新到的或新修复的包与上一次缓存的包的序列号之差如果不大于1,则转步骤(6);

(5) 利用FEC队列和该媒体包修复丢包.如果修复成功,则转步骤(4),否则转步骤(6);

(6) 缓存的或新修复的包与上面刚修复的最小序列号之差如果不大于1,则转步骤(8);

(7) 利用FEC队列和该媒体包修复丢包.如果修复成功,则转步骤(6),否则转步骤(8);

(8) 将新到的媒体包缓存,并从FEC队列中将无用的FEC包清理掉;

(9) 结束.

经修复操作后,已丢失的媒体包被尽可能多地恢复出来.在有些情况下,这种方法还能修复多个连续丢失的媒体包.FEC包在紧接RTCP的下一个端口中作为独立的媒体流传输,它们具有自己的序列号空间,但时间戳仍来源于媒体包.传输FEC格式的RTP数据包采用动态净载类型标识,没有实现FEC方式的接收者可以忽略接收到的FEC包,从而实现后兼容性.

4 功能测试

为了测试协议库的功能,我们设计了一个协议测试系统.通过界面可以设置本地和远端的主机名和RTP/RTCP口号,从而建立一个RTP对话;指定丢包率,使得测试平台按给定的比率丢弃数据包;手动使能或禁用FEC丢包修复机制;关闭指定的RTP对话.协议库测试的主要内容是RTP数据包和RTCP控制包的收/发功能、FEC法丢包修复功能.

采用一种定量的测试策略.发送端发送的数据文件格式如下:

```
1       1       1       1
...
4000    4000    4000    4000
```

每一行的内容由发送端的一个媒体包发送,接收端将每个 RTP 对话接收到的 RTP/RTCP 数据分别存储到两个文件中,从收到的数据序列来观察丢包和丢包修复的效果.

在测试 RTP/RTCP 包的收发时,先在局部网络连接的两台 PC 机 A 和 B 上分别运行测试系统.建立连接后,定时发送 RTP 数据包和 RTCP 控制包.对话关闭后,观察该对话接收到的 RTP 数据序列和 RTCP 报告信息.可以发现,接收到的 RTP 数据序列段是连续的,没有丢包;从 RTCP 报告包也可以看到积聚丢包数为 0,这就证明实现的 RTP 的正确性.

在测试 FEC 丢包修复功能时先按无丢包修复方式,按一定的丢包(包括 RTP 媒体包和 FEC 冗余包)率发送数据;之后运行一段时间,并启动丢包修复方式;在对话结束后查看收到的 RTP/RTCP 数据.从相应的数据文件可见,在丢包比率是 30%的情况下,RTP 数据文件中的前半部分数据序列有许多跳跃间隔,表示发生了丢包现象,对应的 RTCP 报告也显示积聚丢包比率大约是 30%.在 RTP 数据文件的后半部分基本没有跳跃间隔,证明 FEC 方式的修复能力.多次测试选择不同的丢包率(见表 2)均可得到同样的结论.

Table 2 Results of lost repair testing (%)

表 2 修复功能测试结果 (%)

Test sequence①	Lost rate selected②	Lost rate after repairing③
1	30	<3
2	20	<1
3	10	<0

①测试序列,②选择丢包率,③修复后的丢包率.

协议库目前只实现了一种校验编码方式,还没有使用一些适合在较低带宽情况下采用的校验编码.今后工作中要进一步拓展协议库的灵活性,使它能根据网络当前的性能动态地改变校验编码方式,以适应应用环境的变化.

References:

[1] Hardman, V., Sasse, M. A., Handley, M., *et al*. Reliable audio for use over the Internet. 1995. http://www-mice.cs.ucl.ac.uk/multimedia/publications/RAT/inet95-rat.ps.

[2] Perkins, C., Hodson, O. Options for repair of streaming media. draft-ietf-avt-info-repair-03. Work in Progress. 1998. http://www.ietf.org/internet-crafts/draft-ietf-avt-info-repair-03.txt.

[3] Dempsey, B. J., Weaver, A. C. On retransmission-based error control for continuous media traffic in packet-switching network. Computer Networks and ISDN Systems, 1996,28(5):719~736.

[4] Hung Q. Ngo, Srivatsan, Varadarajan, Jaideep, Srivastavas. On achieving lower consecutive losses for continuous media streams. http://cs.mnu.edu/tr1999/99-005.ps.

[5] ITU-T Recommendation H.323. Packet based multimedia communications systems. 1998. http://www.databeam.com/Standards/H.323/H323 Primer.html.

[6] Shulzrinne, H., Casner, S., Frederick, R., *et al*. RTP: a transport protocol for real-time applications. Technical Report, RFC1889. 1996. http://www.ietf.org/rfc/rfc1889.txt.

[7] ITU-T Recommendation H.225.0. Media stream packetization and synchronization for visual telephone systems on non-guaranteed quality of service LANs. 1998. http://www.databeam.com/Standards/H.323/H323-Primer.html.

[8] Schulzrinne, H. GMD fokus, RTP tools. 1998. ftp://ftp.cs.columbia.edu/pub/schulzrinne/rtptools/.

[9] Swan, A., Berkeley, B. D. RTPmon. 1998. ftp://mm-ftp.cs.berkeley.edu/pub/rtpmon/rtpmon.tar.gz.

[10] Mastromartino, E. A. RTP library. 1998. ftp://ftp.iasi.rm.cnr.it/pub/RTP/.

[11] Rosenberg, J., Schulzrinne, H. An RTP payload format for generic forward error correction. Internet Draft, Work in Progress. 1998. http://www.ietf.org/internet drafts/draft-ietf-avt-fec-04.txt.

Real-Time Repairing Lost Packets Based on Real-Time Transport Protocol *

ZHANG Ke, XIE Zhong-cheng, JU Jiu-bin

(*Department of Computer Science, Jilin University, Changchun* 130023, *China*)

E-mail: jjb@mail.jlu.edu.cn

http://www.jlu.edu.cn

Abstract: In real-time applications, such as IP phones and remote conference systems utilizing RTP (real-time transport protocol) for transmitting continuous media data over Internet, packet loss that impairs quality of transmitting severely is inevitable. It is a better way to solve this problem by adding a function of lost repairing to RTP. In this paper, a method of lost repairing with FEC (forward error correction) based on RTP and an RTP library, which was designed and implemented based on this method are introduced. Then a testing method of lost repairing function and the results are illustrated.

Key words: RTP (real-time transport protocol); packet loss; lost repairing; FEC (forward error correction)

* Received December 23, 1999; accepted March 23, 2000

Supported by the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 1999018319