

# 一种自适应调整抖动缓冲区大小的算法

胡 斌, 黎忠文

(厦门大学 信息科学与技术学院, 福建 厦门 361005)

**摘 要:** 在利用 IP 电话进行语音通话时, 语音的 QoS (Quality of service) 是衡量 IP 电话好坏的重要指标。为了提高 IP 电话的 QoS, 当前已经提出了很多提高语音 QoS 方面的策略, 其中比较著名的方案有: IntServ/ RSVP (集成服务/ 资源预留), DiffServ (区分服务)、MPLS (多协议标签交换协议) 等。中文主要是基于语音的接收端, 设计一种自适应调整抖动缓冲区的大小的一种算法, 从而消除语音时延抖动, 提高语音通话质量。  
**关键词:** 语音服务质量; 集成服务/ 资源预留; 区分服务; 多协议标签交换协议; 时延抖动  
**中图分类号:** TP393                      **文献标识码:** A                      **文章编号:** 1673- 629X( 2009) 02- 0035- 04

## One Kind of Auto- Adapted Adjustment Jitterbuffer Size Algorithm

HU Bin, LI Zhong-wen

(School of Information Science and Technology, Xiamen University, Xiamen 361005, China)

**Abstract:** When carries on conversation making use of the IP telephone, to judge the IP telephone, the QoS ( quality of service) nothing but is the important index. In order to improve the IP telephone' s QoS, it has already appeared a lot of strategies for enhancing voice. Among them, quite famous plan has: IntServ/ RSVP, DiffServ, MPLS. This paper is mainly based on the receiving terminal, design one kind of auto- adapted adjustment buffer' s algorithm which eliminates the delay jitter and improves the pronunciation connection quality.  
**Key words:** QoS; IntServ/ RSVP; DiffServ; MPLS; delay jitter

### 0 引 言

VoIP (voice over IP) 即 IP 电话——将人的语音进行压缩编码转换成数据包, 然后在因特网上进行传输, 从而实现语音通话的技术。众所周知, 在利用因特网发送语音包时, 由于通话对延时的敏感性, 基本上是采取 UDP 的传送方式, 即“尽力服务”, 所以就很难对语音包传送有很好的控制。在时变的信道中, 这种服务将会导致分组丢失和不可预测的延迟等, 从而语音的服务质量就得不到保证, 由于语音通信的实时性的特点, 所以, 语音通信对语音质量的要求比较严格。目前衡量语音质量的参数主要包括: 语音包传送的时延、时延抖动、语音包的丢包率和带宽等等。

①时延: 从语音发送端发送 A 语音包的时间为  $t_s$ , 在接收端收到语音包 A 的时间  $t_c$ , 则语音包传送的时延  $t = t_c - t_s$ ;

②抖动: 从语音发送端分别发送语音包 A、B、C, 在接收端收到语音包 A、B、C 的时间分别为:  $t_{sa}$ 、 $t_{sb}$ 、 $t_{sc}$ , 由于语音包到达终端的路径可能不一致, 导致语音包到达终端的时延就不一致, 即  $(t_{sb} - t_{sa})$  不等于  $(t_{sc} - t_{sb})$ , 于是就产生了抖动;

③丢包率: 从语音发送端所发送的语音包, 到达接收端后, 所丢失包的数量与所发送语音包数量的比值;

④带宽: 是指一个链路所能够容纳最大信息流量的标志, 是这个链路每秒可通过字节数的限制。

在语音网络传送数据包时, 有短包和长包, 其中短包占多数, 在文献[ 1] 对 PF 算法对短包传送不利的情況下进行了改进, 在不降低长包传送的前提下提出 3 种对短包时间调度的算法。在通信过程中, 影响语音 QoS 的因素有很多, 包括语音编码、当前网络情况、语音解码以及语音接收终端甚至包括语音流包的长度等等。

在语音传送过程中, 要进行语音的压缩和编码, 同时接收端则要进行解码恢复语音工作, 这不可避免地产 生一定的时延, 同样, 语音包在 Internet 进行传送时, 网络的具体情况往往很难预测, 比如一旦拥塞产生, 那么语音的丢包和时延就必然会产生, 对于语音质

收稿日期: 2008- 06- 03  
基金项目: 福建省自然科学基金项目 (A0410004); 厦门大学院士基金 (0630- E23011); 厦门大学新世纪优秀人才基金 (0000- X07116)  
作者简介: 胡 斌 (1983- ), 男, 江西南昌人, 硕士研究生, 主要研究领域为计算机网络与通信; 黎忠文, 博士后, 教授, 硕士生导师, 主要研究领域为主动网络、实时系统高可靠高安全技术等。

量的评测, NLR<sup>[2]</sup>作为一种测试工具,是集中于网络丢失包对网络所产生的影响。对于语音质量的改善,目前已经有很多文献和相关资料都提出了许多新思想和算法,对语音质量方面已经有了实质性的突破。比如在语音编码方面,已经出现了很多的编码算法,像 G. 711、G. 726、G. 729、G. 723. 1 等,它们的时延分别是: 0. 125ms、0. 125ms、15ms、37. 5ms。而比特率(kb/s)分别为: 64、32、8 和 6. 3。在网络方面,由于语音传送是采用 UDP 的传送方式,所以不同的语音包到达终端的路径往往不会一样,这种现象必然会对通话产生很大的影响,像丢包、时延、抖动等等。

为了解决上面的这些问题,目前已经有了很多很好的思路,文献[3]通过比较 Intserv 和 DiffServ 两种模型,提出了一种两者相结合新的 VoIP QoS 服务模型,文献[4]是基于抖动缓冲区,通过计算 1000 个语音包的网路时延,找出第四大及最小时延,分别记为 max 和 min,然后根据 max 和 min 来调整 jitterbuffer 的大小。文献[5]介绍了语音丢包的隐藏技术,主要包括:插入技术、插值技术、再生技术等,通过它们来提高语音的 QoS。文献[6]在调节 VOIP 网络系统中,如何估计语音包的最佳长度。文献[7]介绍了一种将基于播放时间的同步去抖算法和基于接收缓存数据量控制的再同步算法结合起来的提高语音 QoS 算法。文献[8]通过接收端所反馈回来的丢包率来自适应增大或减少语音包的冗余数据,但是在排除了因丢包风暴中而额外丢失包的数量情况下的。文献[9]是研究音频系统的 QoS,提出一种二次评估后采取的流量调节措施。文中的算法,就是在利用文献[8]所提出的思想,在文献[4]和文献[8]所提出概念模型的基础上,产生的一种新的思路和方法。

## 1 自适应调整抖动缓冲区的算法设计

由于网络上的一些不确定因素(比如:网络拥塞等等),造成语音包到达接收端的时间间隔与发送时间间隔不一致,从而导致语音播放时抖动产生,由于数据包传送时间不一致,必然也会产生一个问题,即包失序情况,接收端要进行包的重排工作。为了最大程度解决上述问题,在接收端设置抖动缓冲区的想法便应运而生,目前,有很多的文献和资料都介绍了如何设置抖动缓冲区,抖动缓冲区总结起来共有两种情况:其一是静态的,即接收端的缓冲区大小一旦设置,便不再改变,最佳缓冲区的大小为 20 个语音包大小<sup>[10]</sup>;其二是动态的,即在语音包的传送过程中,依据网络情况而进行自适应调整抖动缓冲区。文中就是基于第二种情况,提出的一种新的设计思想。

### 1.1 抖动缓冲区大小与丢包率及时延的关系

抖动缓冲区的大小与语音包的丢包率和时延存在着密切的关系,缓冲区  $B$  增大,能一定程度上减小丢包率,但是同时会增加语音包传送的时延,如果缓冲区  $B$  减小,传送的时延虽能减小,但其丢包率又将会增大。先从理论上来计算每个语音包在缓冲区中缓存的最佳时间,即只要缓存这个时间,就能把丢包降为 0,前提是时延要在合理范围之内。来看一组数据包  $i \in (1 \sim n)$ ,  $j \in (1 \sim n)$  且  $j = i + 1$ ,即  $j$  是  $i$  的后续一个语音包,则  $jns = \max\{|(R_j - S_j) - (R_i - S_i)|\}$ ,只要接收端的缓冲区大小超过  $jns$  则不存在丢包情况,数据包在接收端的缓存时间为  $jns$ ,反之,则会产生丢包,这种情况时,到达接收端的语音包,按照  $B$  的大小进行最大缓存。比如有 4 个语音包: A、B、C、D,发送的时间分别为 20、40、60、80(单位是 ms),到达接收端的时间分别为: 30、60、70、90,即  $jns = 10ms$ ,假设抖动缓冲区的大小  $B$  为 20ms,此时  $B > jns$ ,所以此时的丢包率为 0,播放时间为: 40、60、80、100;当  $B$  为 5ms 时,  $jns > B$ ,语音包按照  $B$  的大小进行缓存,此时这 4 个语音包的播放时间为: 35、55、75、95,现在来看看语音包  $B$ , $B$  包到达接收端的实际时间是 60ms,而经过抖动缓冲区的调整后,语音包  $B$  在理论上播放的时间应该是 55ms,所以结论是语音包  $B$  丢失了。为了验证上面的数据,笔者将在下节对抖动区的设置和丢包个数做一个大致的模拟。

### 1.2 抖动缓冲区与丢包率的关系

实验模拟环境采用 Microsoft Visual C++ ,对发送 1000 个语音包做了 10 次采样,模拟见图 3(位于上面带的连接图),从图中的实验数据可以得出,随着抖动缓冲区的不断减小,语音包丢包率呈上升的趋势。

### 1.3 算法的思路

通过上面一节的分析和实验模拟,已经很清楚地知道时延、丢包率与抖动缓冲区大小它们之间的密切关系,但本来又存在着一种矛盾,即“二者不可兼得”,所以要找出抖动缓冲区大小的一个平衡点,在保证消除抖动的情况下,尽可能减少时延的长度和丢包的个数,本节介绍的算法就是基于这点产生的,不过该算法是在保证时延在合理范围内最大程度上减少语音丢包的个数,在介绍之前,先来看一组数据<sup>[5]</sup>,根据美国的一项统计测试结果表明,对于语音品质,低于 200ms 的时延和 5% 的丢包率是可接受的范围,时延在 200ms ~ 400ms,而丢包率在 5% ~ 10% 之间时,是一个边缘范围,可满足部分用户的要求,当时延超过 400ms,丢包率超过 10% 时,则不能满足语音会话的要求。先来设置两个值:  $H = 10\%$ ,  $t_v \in (0, 400)$ ,  $H$  表示能接受

的最大丢包率,  $t_y$  表示能接受的语音延时的范围, 算法的流程图如图 1 所示, 从图中, 可以很明显地看出, 整个算法的前提就是一定要保证时延处于合理范围之内, 在这里为  $400\text{ms}$ 。也就是说, 在丢包率  $P$  能满足  $P > H$  而时延小于  $400\text{ms}$  时, 采取的措施是增大缓冲区  $B$  的大小, 使得  $P < H$  同时时延仍在合理范围之内。但当满足  $P < H$ , 时延超过  $400\text{ms}$  时, 则这时  $P$  可以适当超过  $H$  的值, 而把时延调整到上限  $400\text{ms}$ 。当时延超过  $400\text{ms}$  时, 则可以不断减小  $B$ , 直到时延调整到合理范围之内, 整个发送语音包的过程是一个循环, 从流程图中, 可以知道是设置每发送 20 个语音包而进行一次调整, 直到整个通话的结束。

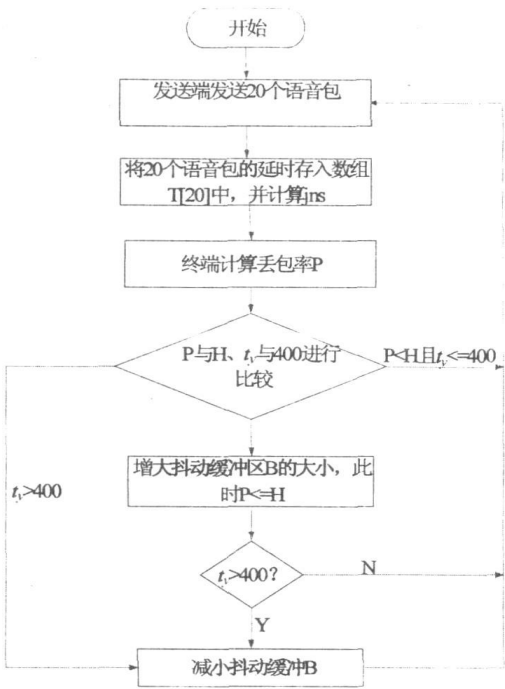


图 1 算法流程图

这里还有 3 个问题急需解决:

① 如何计算  $jns$  的值。

设置一个长度为 20 的队列, 用于存储一次语音通话过程中最新的 20 个语音分组的网络延时值, 通过该队列中存储的数据, 用关系式:  $jns = \max \{ | \sum_{i,j}^{20} (R_j - S_j) - (R_i - S_i) | \}$  (其中  $j = i + 1$ )。

② 如何增大抖动缓冲区。

我们知道, 当  $P > H$  时, 则要开始增大抖动缓冲区, 增大的方式如下: 比如语音包 A 理论上播放的值为  $t_{a1}$ , 而 A 实际播放的时间为  $t_{a2}$ , 则  $\tau_a = t_{a2} - t_{a1}$ , 用同样的方式计算其它所丢失语音包  $\tau$  的值, 假如丢包的个数为  $m$ , 则  $B' = B + \frac{1}{m} \sum_{i=1}^m \tau_i$ , 要是此时的丢包率的大小在合理范围之内, 则调整结束, 否则,  $B$  的大小就

调整为  $1.5 \times B$ , 记住, 不能在为保证丢包率降到合理范围内, 而无休止地增大缓冲区的大小, 当时延超过  $400\text{ms}$  时, 就以  $400\text{ms}$  为上限, 不再调整  $B$  的大小。

③ 如何减小抖动缓冲区。

当抖动缓冲区增大到一个值时, 丢包率会降低, 要是此时的时延在合理范围之内, 可以不用调整  $B$ , 而继续发送语音包。但要是此时的时延得不到满足, 则就减小  $B$  的大小, 减少的法则是: 比如  $\tau = t_y - 400\text{ms}$ , 则  $B' = B - \tau$ , 注意, 当  $B' > 0$  时, 此时的时延可以调整为  $400\text{ms}$  以内, 这里给出其严格的证明过程:

$B' = B - \tau$  推出  $B' = B + 400\text{ms} - t_y$  (备注:  $t_y$  表示时延)。

因为:  $t_y \leq t_c + B$  (备注:  $t_c$  表示每发送 20 个语音包时第一个语音包的传送时间)。

所以:  $B' \leq B + 400\text{ms} - t_c - B$ , 推出  $B' \leq 400\text{ms} - t_c$ , 调整后的时延  $t_y \leq B' + t_c \leq 400\text{ms}$ , 所以  $t_y \leq 400\text{ms}$  证毕。

当  $B' < 0$  时, 就将  $B'$  设为 0, 再继续发送语音包, 通过这种方式不断调整抖动缓冲区的大小。

2 实验仿真和比较

2.1 实验仿真

1) 对文献[4]的算法模拟来统计其丢包率及时延。实验仿真如图 2 所示, 从仿真数据可以得出, 其丢包率很小, 但时延却比较大。

统计图	
丢包率(%)	延时(ms)
0.3	773
0.2	786
0.1	555
0.3	757
0.2	495
0.3	551
0.1	749
0.2	586
0.1	881
0.2	831

图 2 丢包率和延时的统计图

2) 对文中的算法进行仿真模拟。如图 3 所示, 运用文中算法, 每次采样, 都计算发送 1000 个语音包的平均丢包率  $P1$  和平均缓冲区大小  $B1$  (带 \* 的连接图), 然后在相同缓冲区情况下计算其丢包率  $P2$  (带 • 的连接图)。

可以得出结论: 运用文中介绍的算法, 在相同缓冲区大小的情况下, 语音丢包率明显下降很多, 并且抖动缓冲区大小平均维持在  $150\text{ms}$  左右。

## 2.2 实验比较

现在来对图 2 和图 3 中的数据进行分析, 它们都是对 1000 个语音包进行采样。采用文献[4]算法得到的丢包率都很低, 在  $\leq 0.3\%$  浮动, 主要原因是此算法在发送的 1000 个语音包中选择了第 4 大延迟作为 max, 结果会把延迟最大的至多 3 个包丢弃(有可能第 2 大、第 3 大的延时与 max 相等, 所以至多有 3 个包丢失), 从丢包率看, 此算法明显比文中算法要好, 但在文献[5]中曾指出语音包的时延不能超过 400ms。文献[4]的最大缺点也正是在此。相比之下, 文中的算法是维持时延在合理范围之内对丢包率进行最大程度上的降低。通过 10 次采样, 缓冲区的大小平均维持在 150ms 左右, 丢包率在 4%~10% 左右浮动, 所以, 延时要明显好于上述算法。其实, 运用文中算法时, 要是不考虑 400ms 这个阈值的话, 在时延接近 700ms 时, 丢包率不一定会比文献[4]差。

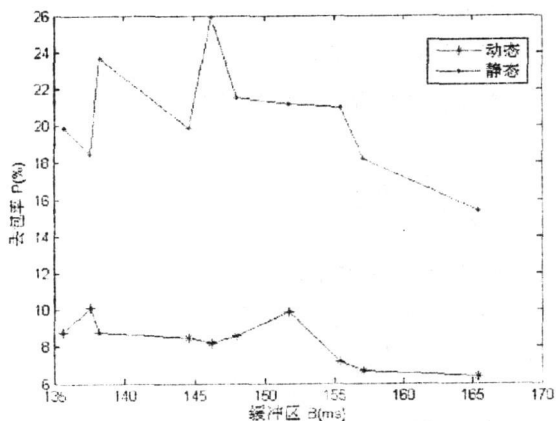


图 3 比较图

## 3 结束语

针对语音接收端如何动态设置抖动缓冲区大小而提出的一种思路, 在文中算法中, 是在保证时延的前提下, 而尽可能地减小丢包率, 但是, 算法还存在一些明显的不足: 首先, 设置发送完 20 个语音包时, 调整一次抖动缓冲区大小, 算法只能保证在每次发送的这 20 个语音包中最大可能消除抖动现象, 但前 20 个包中的最后一个语音包与后 20 个包中第一个语音包, 却不可避

免地产生了抖动; 其次是语音包到达缓冲区时, 由于网络情况的不稳定, 到达缓冲区的语音包会出现失序的情况, 这必然又会产生不必要的丢包情况。综上所述, 下一步的工作主要集中在上面几点, 要通过进一步的研究和学习, 最大限度地解决上述问题。

### 参考文献:

- [1] Khawam K, Kofman D. Flow Size-Aware Proportional Fair Scheduler[C]// Next Generation Internet Design and Engineering. 2006. NGI '06. 2006 2nd Conference. [s. l.]: [s. n.], 2006: 350 - 357. <http://ieeexplore.ieee.org/iel5/11058/34932/01678261.pdf?tp=&arnumber=1678261&isnumber=34932>.
- [2] Levy H, Zlatokrilov H. The Effect of Packet Dispersion on Voice Applications in IP Networks[J]. IEEE/ACM Transactions on Networking, 2006, 14(2): 277- 288.
- [3] 姚玉坤, 刘合武. IntServ 与 DiffServ 在 VoIP QoS 中的应用分析[J]. 计算机与数字工程学报, 2007, 35(4): 55- 58.
- [4] 周康, 裴楷, 吴宇红. VoIP 系统中消除时延抖动的研究与实现[J]. 电子科技, 2007(1): 51- 54.
- [5] 王培明, 施寅. VoIP 中丢包隐藏技术研究[J]. 计算机技术与发展, 2006, 16(7): 26- 28.
- [6] Hassan M, Alekseevich D F. Variable Packet Size of IP Packets for Voip Transmission[C]// Proceedings of the 24th Iasted International Multi-Conference Internet and Multimedia Systems and Applications. Innsbruck, Austria: [s. n.], 2006: 136- 141.
- [7] 陈明义, 钱敏, 谭勇. 提高 IP 电话 QoS 的综合去抖动同步策略研究[J]. 计算机工程与设计, 2006, 27(23): 4991 - 4993.
- [8] 祝宇成, 周应权. 一种实时语音传输中冗余控制算法的改进[J]. 科学技术与工程, 2007, 7(17): 4476- 4479.
- [9] 付强. RTP 协议及其 QoS 研究[D]. 南京: 南京邮电大学, 2006.
- [10] Augé J, Roberts J. Buffer Sizing for Elastic Traffic. Next Generation Internet Design and Engineering[C]// NGI '06. 2006 2nd Conference. [s. l.]: [s. n.], 2006: 33 - 40. <http://ieeexplore.ieee.org/iel5/11058/34932/01678220.pdf?tp=&arnumber=1678220&isnumber=34932>.

(上接第 34 页)

- 16th international conference on Supercomputing (ICS'02). New York: ACM Press, 2002: 84- 95.
- [4] Joseph S, Hoshiai T. Decentralized meta-data strategies: Effective peer-to-peer search[J]. IEICE Transactions on Communications, 2003: 86(6): 1740- 1753.
- [5] Sripanidkulchai K, Maggs B, Zhang H. Efficient content Location Using Interest-Based Locality in Peer-to-Peer Sys-

- tems[C]// In Proceedings of IEEE INFOCOM. San Francisco, CA: IEEE Infocom on Computer and Communications Societies, 2003: 2166- 2176.
- [6] Colomi A, Dorigo M, Maniezzo V. An investigation of some properties of an ant algorithm[C]// In: Proc. of the Parallel Problem Solving from Nature Conference (PPSN'92). Brussels, Belgium: Elsevier Publishing, 1992: 509- 520.