基于分组网络的多点实时语音混合及调度算法*

杨树堂, 余胜生, 周敬利

(华中科技大学 计算机外存储系统国家专业实验室,湖北 武汉 430074)

E-mail: wtyst@hust.edu.cn

http://www.hust.edu.cn

摘要: 平滑、流畅的语音交流是当前高性能视频会议系统追求的主要目标之一. 为此,提出并实现了一种基于分组网络的多点实时语音混合及调度算法. 调度算法采用了排队模型分析、多点语音流的同步控制、数据预取以及缓冲区定时刷新等策略,在有限的复杂度内有效地保证了混合后语音的连续性. 同时,采用的混合方法保证了语音质量的自然度和可理解性. 算法的实现使得在多点情况下,此 H.323 视频会议系统比现有国外同类 H.323 产品(如 Netmeeting 等)具有更好的语音听觉效果.

关键词: 分组网络;多点通信;语音混合;调度算法

中图法分类号: TP391 **文献标识码**: A

近年来,分组网络上的实时多媒体通信业务得到了迅猛的发展,多点通信产品应运而生. 其典型的代表是视频会议系统. 在视频会议系统中,多点之间的语音交互最为频繁. 为了模拟真实会场的情况,增强会议的真实感及舒适感,系统应使参加会议的各方能够在需要时同时听到多个发言者的声音. 这就要求系统能将多个用户端传来的声音进行混合处理. 然而,在分组网络上,由于没有 QoS(quality of service)保证,网络拥塞产生了端到端通信的语音丢包、时延及抖动等问题[1],严重地影响了网上传输业务的服务质量. 而且,多个发送端是并发传送数据的,各方是否且何时发送数据以及它们发送语音包到达的相对次序等都具有很大的随机性和波动性. 这些问题使得多点实时语音的混合处理成为视频会议系统的技术难点.

本文的工作是基于 H.323 视频会议系统[2]的多点会议工作模式,对实时语音的混合算法及其实现进行研究.

1 多点语音混合的同步控制策略

本系统使用 UDP(user datagram protocol)作为实时音频和视频传输的传输层协议,并且使用 RTP(real-time transport protocol)[1]作为应用程序的一部分来完成实时媒体流的传输及其 QoS 监控的任务[3],以保证语音包在交给解码器时是有序的,并且使丢失率尽可能地小,这是本算法正确执行的前提.

此外,本算法还必须解决网络上语音流的同步控制问题. 由于分组网络没有统一的全局时钟,在实时视频会议中,发送端与接收端之间存在着潜在的时钟不匹配问题. 在这种情况下,通信的双方按照各自独立的时钟执行. 若不采取措施同步这些时钟,则由于视频/音频数据流基于发送端本

* 收稿日期: 1999-12-02; 修改日期: 2000-05-10

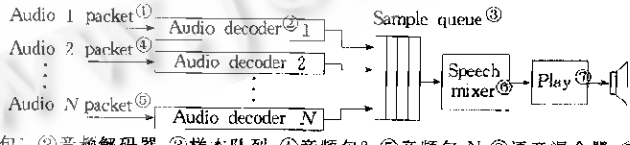
基金项目: 国家"九五"国防预研基金资助项目(15.8.4)

作者简介: 杨树堂(1968-),男,湖北新洲人,博士生,主要研究领域为语音压缩,自适应滤波,回声消除,多速率系统,QoS,网络流控;余胜生(1944-),男,江西南昌人,教授,博士生导师,主要研究领域为计算机系统结构,多媒体网络;周敬利(1947-),女,湖南长沙人,教授,博士生导师,主要研究领域为计算机系统结构,多媒体技术.

地时钟产生,使得接收端的本地时钟与 RTP 包、RTCP(RTP control protocol)包所携带的时间戳的包时钟不一致,不能保证播放端的实时性和声音的连续性.解决这个问题的办法是静音期间不传送包以及在接收端采用缓冲器吸收抖动、调节时钟的不匹配[4].

本文将这种方法在多点通信条件下进行了应用和扩展.多个发送端分别按自己的时钟采集语音数据,然后利用语音检测模块对其进行有声和无声的判别.若有声,则将其压缩、传送;无声,则不传送.相应地,接收端也按照自己的时钟播放语音数据.多点发送的音频流由混合模块(即执行本算法的模块)进行同步控制及混合,输出后得到单一的语音流.对于集中式系统,该语音流将由 MCU (multipoint control unit)控制编码后转发到组内各用户端,由用户端解码后播放;对于分散式系统,混合过程在用户终端进行,混合的结果直接在用户端播放.不论是在集中式还是分散式系统中,对混合模块而言,来自不同发送端的个体媒体流之间的时间关系完全依赖于独立媒体的到达时刻.多个音频流之间的相对时间关系由模块内部开辟的 Buffer 缓冲区来记录,各语音包在 Buffer 缓冲区中的驻留时间由算法控制.该缓冲区由几个固定长度的循环 Buffer 缓冲组构成.为了下文叙述方便,我们约定:本算法为了缓冲数据及实现同步控制,为参与会议的所有用户分配了一个大的缓冲区,称为 Buffer 区;Buffer 区包含多个 Buffer 组;一个 Buffer 组用于存放参与混合的某一个用户传来的解码语音数据,它具有循环结构,包含多个 Buffer 块;每个 Buffer 块用来存放一个解码后的语音包数据,其大小是固定的(SpeechPacketSize);Buffer 块的总数由参与会议的用户个数及 Buffer 组的大小决定.

Buffer 区是按组管理的,Buffer 组大小的选取与网络的性能有关,特别是与网络端到端的延时及抖动有关.瞬时的延时及抖动可以利用 RTCP 的头部信息中的有关项来计算,但是,这只是局部的信息.若要反应全局的延时信息,则需利用排队论方法分析分组网络上多点语音传输到混合端的模型(如图 1 所示).由于在 H.323 系统中,H.225.0 终端使用相互独立的 RTP 实例在不同的传输端口地址上独立地发送和接收音频与视频流[5],因此,可以单独分析分组语音流的网络传输模型.根据文献[6,7],当参与会议的用户个数在 8~15 之间时,用 Semi-Markov 模型和 Continuous-time Markov 模型均可较好地反映网络状况.本文采用的是 Continuous-time Markov 模型,限于篇幅,这里不再给出建立模型的细节.我们综合考虑由 RTCP 控制包所计算的时延、抖动以及排队模型的平均等待时延,依据这些值选择 Buffer 组大小的初始值,并根据测试情况加以调整,以获得连续的语音效果.



①音频包1,②音频解码器,③样本队列,④音频包2,⑤音频包 N,⑥语音混合器,⑦播放.

Fig. 1 Queue model of speech mixing in mulitipoint communication (on user end)

图1 多点通信的排队混合模型(基于用户端)

为了保证连续性,算法采用了自适应输出速率调节机制[8].当仅有一方发送数据,且其到达语音包的个数超过给定阈值时,出于同步考虑,直接将对于应该用户 Buffer 组中最旧的数据输出(不经混合),然后填入新的数据;当网络拥塞导致语音数据到达变慢(出现抖动现象)、Buffer 区中数据量偏少时,算法将只填充数据到某个 Buffer 组而不输出数据,从而降低播放速度(加大播放时间间隔).其核心思想就是控制每个语音包在缓存中的驻留时间,进而控制其同步播放时间[9].

2 调度策略及混合算法

为了对算法进行调度及对语音包在Buffer区中的驻留时间进行控制,本文引入了多个计数器:*InitBufCnt* 对预取的包数进行计数;*PresentBufNum* 对当前Buffer区中包含数据的Buffer块进行计数;*lpMixGlobal->RevBufCnt* 对接收到的语音包总数进行计数,估算工作时间,以便定时对缓冲区进行刷新;*lpMixBuffer->BufCnt* 对从某个用户接收到的语音包进行计数.经过测试确定了3个阈值:一个是 *BUFRESETTHRD*,用来控制模块的刷新,以消除累计误差的影响;一个是 *USERBUFNUM*,代表Buffer组的大小,用于音频流的同步控制;另一个是 *PREFETCHTHD*,代表预取语音包个数的阈值,用于调节缓冲数据及减少数据抖动现象.

(1) 预 取

考虑到输入数据到达时间的随机性及同步要求,本算法采用了预取机制,即首先将输入数据按照不同的用户号填入相应的Buffer组中.同时,利用 *InitBufCnt* 对预取数据的Buffer块的总数进行控制,当预取总数到达阈值 *PREFETCHTHD* 时,才进入下一个阶段.在此阶段中,采取一定的策略对各用户Buffer组加载数据的情况进行监控.若接收到的用户 $i(i=1,\ldots,N)$ 语音包的个数达到分配给它的总数 *USERBUFNUM*,但包含数据的Buffer块总数又小于 *PREFETCHTHD* 时,则将该用户Buffer组最旧Buffer块中的数据替换为新的语音包数据.

预取的数据量(时间)*PREFETCHTHD* 不宜太大也不宜太小.若预取量太大,虽然对缓冲、同步及防抖动有利,但引入的系统延时太大;如果延时太小,则所起到的缓冲作用不大.在本文中预取的包数初始值取为Buffer块总数的一半,并在实验过程中进行调整.

(2) 定时刷新

本文所采用的是循环更新数据的Buffer区存放参与混合的用户语音数据,而Buffer区是按照Buffer组来管理的,因此,对某个Buffer组而言,它每次接收到的总是最新的Buffer块数据,而输出送去播放的是最旧的Buffer块数据.由于无统一时钟以及网络传输存在时延、抖动等因素的存在,经过长时间的运行后,则算法模块的语音包到达时钟与由回放端同步的输出时钟之间的累积误差可能导致指针的操作错误,使得循环结构的Buffer组输入指针与输出指针之间的相对关系发生变化,出现输出的数据并非最旧的数据,或新旧数据交替出现,从而导致回放端播放的效果出现颤音现象.

为了避免这种情况的发生,本文设计了一个复位计数器(计数最大值为 *BUFRESETTHRD*),要求在算法模块运行一段时间后对其相关的计数器及指针进行复位,本文称这个过程为定时刷新.

(3) 调度策略

本调度算法获得理想效果的前提是各Buffer组中的数据没有发生失序和严重的丢包现象.这一点由第1节提到的利用RTP实现对QoS的监控可以得到保证.

当混合模块接收到一个语音包时,算法将按照如下策略进行调度:

步骤1.判断是否需要进行定时刷新.

步骤2.对参与会议的用户个数进行统计,若用户数小于2,则无须混合,直接将该语音包输出.否则,转入步骤3.

步骤3.若 *InitBufCnt*<*PREFETCHTHD*,则预取缓冲数据.

步骤4.否则进入正常的混合过程:

若 *PresentBufNum* 小于 *USERBUFNUM*,则算法控制仅接收数据而不输出数据,强制使回放

端播放速度变慢,然后转入步骤1;否则进入步骤5.

步骤5.若在参与会议的用户中,仅第i个用户在发送数据,并且该用户在混合端相应的Buffer组已满,则当新的数据仍来自该用户时,此时数据无须混合,直接将旧的数据输出后,填入新的语音包.然后,转入步骤1;否则表明当前参与会议系统内,多个用户的Buffer组中含有数据,转入步骤6.

步骤6.调用混合算法对这些用户的Buffer组中当前欲输出的Buffer块数据进行混合,将混合后的数据输出,并将新到数据填入相应用户的Buffer组中.然后转入步骤1.

在一般会议中,绝大多数时间都是一个人在发言,其他人在听,只有在自由讨论时才有多个人同时讲话的情况.本调度算法考虑了会议的这个特点,利用调度来尽量减少运算量:在算法中,只有执行混合算法处运算量最大,而按照本调度策略,只有在很少的情况下才调用它.

(4) 混合算法

混合算法的基本思想是,首先将多路语音进行线性叠加,然后对叠加语音数据进行溢出检查,并对含有溢出数据的混合语音包的语音样本进行滤波处理,采用平滑技术消除叠加引入的噪声.具体地说,混合算法主要完成以下功能:

• 线性叠加

将当前参与混合的各Buffer组中相应Buffer块数据转换到浮点域后,将对应的样本值按照式(1)进行线性叠加运算.同时,统计出参与混合的所有Buffer块中数据的最大值($Total$Max)以及叠加后数据的最大值($Mixed$Max).

$$mixedData[i]=\sum_{j=0}^{N-1}Inspeech[j,i],\ 0\leqslant i<SpeechPacketSize, \tag{1}$$

其中$mixedData[i]$为混合后语音包中的第i个样本,$Inspeech[j,i]$为第j个用户语音的第i个样本,N为参与会议的用户个数.

• 溢出判断及平滑处理

将混合后的浮点数据与所采用的PCM数据的最大与最小值(对于16 bit PCM数据,最大值为32 767,最小值为−32 768)相比较,若存在超出此范围的数据,则该混合语音包有溢出现象,需进行平滑处理:按照式(2)对混合后的数据幅度进行调整,然后将调整后的混合数据反向转换为PCM数据;否则,直接将混合后的数据反向转换为PCM数据,然后输出.

$$mixedData[i]=mixedData[i]*\frac{Total\text{Max}}{Mixed\text{Max}}*\mu, \tag{2}$$

其中$0\leqslant i<SpeechPacketSize$,$1\leqslant\mu<(Mixed\text{Max}/Total\text{Max})$.

本算法主要基于以下理由:

• 语音的有声段是一种连续、平滑的信号.两个连续、平滑波形叠加的结果仍然是连续、平滑的.溢出现象只是由于受到语音数据表示精度(16bit或8bit)的限制而产生的,而精度是由相应编码器的要求所决定的.

• 对语音波形进行平滑滤波处理不会改变语音的音质和内容,因为语音信号具有短时(10ms~30ms)相关性.在我们的H.323系统中,音频部分中所有语音包的大小均在此范围之内.通过平滑处理,即一个语音包的数据按照比例缩小,不会改变语音的特征参数(如共振峰及基音周期等)的大小,也不会改变语音信号的波形.

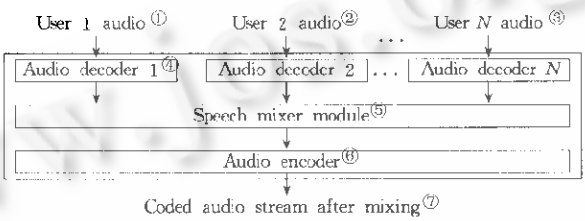
考虑到实时性的要求,本算法采用了上述简单而有效的平滑处理方法.算法的最大复杂度为

$(N-1)*SpeechPacketSize$ 次加法,$SpeechPacketSize$ 次乘法及 $N*SpeechPacketSize$ 次比较运算.

3 算法实现与仿真测试

3.1 算法实现概要

根据视频会议的要求,参与会议的各方要能够同时听到所有发言者的声音,而且还要能够动态地加入或退出会议.基于以上考虑,本文将所有相关模块(如视频和音频的编解码模块、RTP 模块等)做成独立的可重入的模块(可多个同时工作),同时将语音混合器也做成一个独立模块(Mixer 模块).Mixer 模块对原始语音(或解码后的语音)进行混合,即让它独立于编码器工作.它既能放到 MCU 上,也可以放在各用户终端完成多点语音的混合功能.



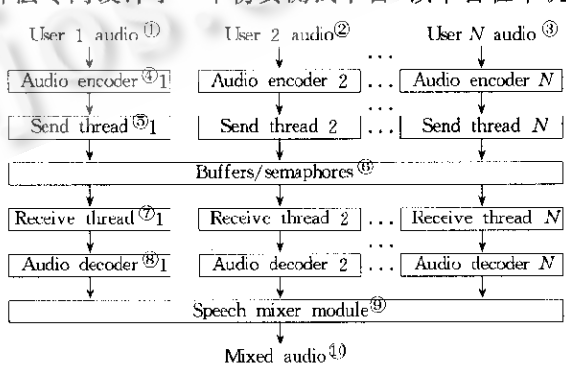
①用户音频1,②用户音频2,③用户音频 N,④音频解码器,⑤语音混合器模块,⑥音频编码器,⑦混合后的编码音频流.

Fig. 2 Mixing scheme on MCU
图2 基于MCU的混合方案

对于集中式会议系统,混合是在 MCU 上进行的.MCU 首先对各用户到达的语音进行解码,再将解码后的语音送到混合器进行混合.然后,将混合后的语音数据进行压缩,将压缩后的语音转发给参与会议的各用户端,其实现方案如图 2 所示;对于分散式会议,语音的混合是在用户端进行的.用户终端接收到参与会议的各方按照广播方式发来的音频流之后,将它们分别解码,然后送到混合模块进行混合,并将混合后的结果送扬声器播放,其实现过程可由图 1 看出.

3.2 仿真测试

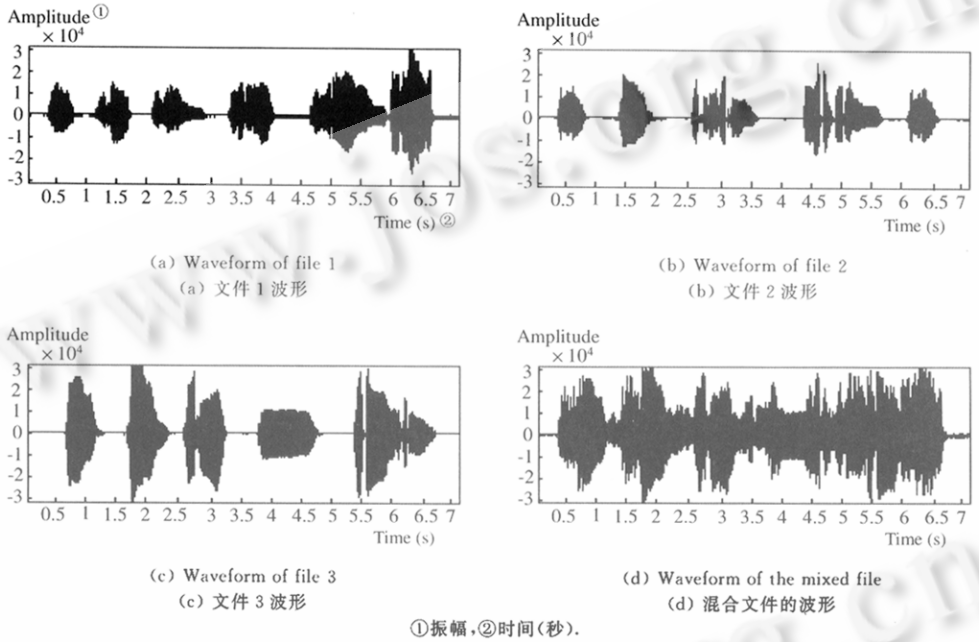
为了便于调试及控制,我们为混合算法专门设计了一个仿真测试平台.该平台在单机上采用多线程技术,利用其并发机制模拟网络上多点语音实时通信的情况.其基本思想是:采用文件加上线程的工作方式,以不同的波形文件代表实时会议中不同的用户语音,各波形文件的声音经压缩后,利用线程向测试平台传送数据;测试平台接收到来自用户的数据以后,调用相应的语音解压缩算法将数据解码后送入混合模块进行混合.混合后的结果仍然以波形的形式记录下来.图 3 给出了该方案的方框图.



①用户音频1,②用户音频2,③用户音频 N,④音频编码器,⑤发送线程,⑥缓冲区/信号灯,⑦接收线程,⑧音频解码器,⑨语音混合器模块,⑩混合后的音频.

Fig. 3 Realization scheme of the testing platform
图3 测试平台的实现方案

在测试平台上,我们采用了3个事先录制的长度为7.262秒的波形文件(8000Hz/16bit/Mono)作为参与混合的声音源进行测试,其波形分别如图4(a)~(c)所示.图4(d)为它们按照前述算法进行混合的结果.从图中可以看出,混合的语音波形反映出了三者叠加后连续的、除少量毛刺外大体上平滑的结果,而且没有数据溢出现象.混合后波形文件的播放效果基本上达到了相应编码器的音质:按MOS划分,G.711为4.4分,G.723.1,6.3kbit/s和5.3kbit/s算法分别为3.6和3.4分,G.729a为4.0分等.因此,本算法在测试平台上成功地实现了混合功能.



(a) Waveform of file 1
(a) 文件1波形

(b) Waveform of file 2
(b) 文件2波形

(c) Waveform of file 3
(c) 文件3波形

(d) Waveform of the mixed file
(d) 混合文件的波形

①振幅,②时间(秒).

Fig. 4 Waveforms of three original speeches and their mixed result on the testing platform
图4 3个原始语音的波形以及它们在测试平台上的混合波形

以此为基础,本算法被移植到我们用纯软件(C语言)实现的实时H.323会议系统中,并进行了测试.因为尚未实现网关功能,目前该系统的测试是在局域网上进行的.在10MB带宽的局域网环境下及Pentium 200MHz MMX CPU/32MB RAM以上配置的机器上,我们的H.323系统可以支持8个点的会议,在屏幕上可以同时看到8个QCIF大小的参与者视频图像,并且很好地完成了语音混合功能,音质连续、流畅、自然.

本文提出的H.323会议系统具有较好的兼容性,可与Microsoft Netmeeting2.1,Intel Proshare及CU-SeeMe等国外视频会议系统互连.通过多次测试、比较,结果表明,我们的H.323系统由于加入了本文描述的算法模块而真正实现了宽松、自然的会议环境,而像Netmeeting等,在多点情况下则达不到这个效果.它们采用的语音多路切换方式使得某个用户端虽然能听到多个会议参与者讲话的声音,但是会明显感觉到由于切换造成的语音断续现象.

4 结 论

本文提出并实现了一种基于分组网络的实时语音混合和调度算法,解决了在视频会议系统中实现多点语音混合的关键技术.仿真实验和实际的运行情况表明,本算法调度合理、结果正确.在多

点模式下,我们的 H.323 系统混合的语音音质优于国外已有的同类产品.

致谢 作者感谢同课题组的袁双庆博士和鲁宏伟副教授对本研究工作的支持和帮助.

References:

[1] Schulzrinne, H., Casner, S., Frederick, R., *et al*. RTP: a transport protocol for real-time applications. IETF RFC1889. IETF, 1996.

[2] ITU-T. Packet-Based multimedia communication systems. ITU-T Rec H.323V2, 1998.

[3] Zha, Hui. Dynamic QoS monitoring and control in real-time multimedia applications [MS. Thesis]. Huazhong University of Science and Technology, 1999 (in Chinese).

[4] Wilebeek-LeMair, M.H., Shae, Zon-Yin. Videoconferencing over packet-based networks. IEEE Journal on Selected Areas in Communications, 1997,15(6):1101~1114.

[5] ITU-T. Call signaling protocols and media stream packetization for packet based multimedia communications systems. ITU T Rec H.225.0V3, 1998.

[6] Yin, Nan-ying, Li, San-qi, Stern, T.E. Congestion control for packet voice by selective packet discarding. IEEE Transactions on Communications, 1990,38(5):574~683.

[7] Daigle, J.N., Langford, J.D. Models for analysis of packet voice communications systems. IEEE Journal on Selected Areas in Communications, 1986,4(6):847~855.

[8] Wang, Lei, Cai, An-ni, Sun, Jing-ao. Delay estimation and synchronization of real-time data. Journal of China Institute of Communications, 1999,20(2):46~52 (in Chinese).

[9] Wei, Tie-jun, Chen, Jun-liang. Real Time adaptive multimedia synchronization based on leaky window. Journal of China Institute of Communications, 1999,20(5):2~8 (in Chinese).

附中文参考文献:

[3] 查辉.连续媒体实时应用中服务质量的动态监测和控制[硕士学位论文].武汉:华中科技大学,1999.

[8] 王雷,蔡安妮,孙景鳌.延时估计和实时数据的同步控制.通信学报,1999,20(2):46~52.

[9] 魏铁军,陈俊亮.基于漏窗机制实时自适应多媒体同步.通信学报,1999,20(5):2~8.

A Multipoint Real-Time Speech Mixing and Scheduling Algorithm Based on Packet Networks*

YANG Shu-tang, YU Sheng-sheng, ZHOU Jing-li

(*National Storage System Laboratory, Huazhong University of Science and Technology, Wuhan* 430074, *China*)

E-mail: wtyst@hust.edu.cn

http://www.hust.edu.cn

Abstract: Smooth and fluent speech communication is one of the main aims of current high performance video conferencing system. For this purpose, a multipoint real time speech mixing and scheduling algorithm based on packet network is put forward and then realized. The scheduling algorithm, which contains several strategies, such as queue model analysis, synchronous control of multipoint speech streams, data pre-fetching, timely refreshing buffers and so on, effectively guarantees continuity of the mixed speech in a limited complexity, and its mixing method has proved to ensure naturalness and intelligibility of the mixed speech quality. The realization of this algorithm has led to a fact that the video conferencing system has better hearing perceptibility than those available oversea H.323 products, such as Netmeeting, based on the same H.323 specification.

Key words: packet network; multipoint communication; speech mixing; scheduling algorithm

* Received December 2, 1999; accepted May 10, 2000

Supported by the National Defence Pre-Research Project of the 'Ninth Five-Year-Plan' of China under Grant No. 15.8.4