计算机科学2002Vol. 29№. 3

面向电子白板的拥塞控制方案*)

An Electronic Whiteboard-Oriented Congestion Control Scheme

刘英智　吴文峻　金　天　盛向治

(北航国家软件重点开发实验室　北京100083)

Abstract　The Distributed Electronic Whiteboard is a multicast-based group-communicating tool. As there is no proper congestion control mechanism in the Whiteboard yet, this greatly hampers the more wide application of Whiteboard. The main purpose of this paper is to provide a rate-based multicast congestion control scheme--SRM-NEW for Whiteboard.

Keywords　Whiteboard, Reliable multicast congestion control, TFRC, ALF, SRM, SRM-NEW

1. 电子白板的技术背景

电子白板是一种多用途的实时交流工具,可以应用于网络视频会议、多媒体实时教学、科研讨论等范畴。电子白板的主要功能是使多人可以进行实时的文字、图形等信息的交流。电子白板是基于可扩展可靠组播协议SRM[1](Scalable Reliable Multicast Protocol)开发的,如图1所示为电子白板的设计框架图。其中最低层的就是SRM协议库。该协议库是以U. C. Berkeley的Mash小组开发的SRM lib2为原型,进一步改进开发而成的。

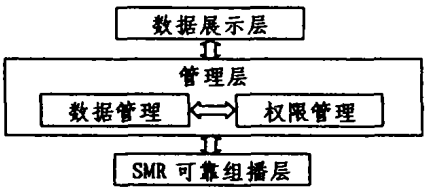

图1　电子白板结构框架图

SRM协议是由S. Floyd等人提出的,该协议的目的是实现组播中的可靠传输。SRM协议是一个面向轻型会晤应用的协议,也就是说SRM并不适合大规模的组播应用。其中的一个主要原因在于SRM中没有一个灵活有效的组播拥塞控制机制。SRM中的拥塞控制策略只是简单地限制了发送者的发送数率上限,这种方法虽简单可行并且也可以满足小型会议的要求,但是如果参与人数上升到一定的规模则必然会造成整个网络的拥塞崩溃。而且在参与人数很少而且网络状况良好的情况下,限制上限的方法则又不能有效地利用网络带宽。所以如不能改善SRM的拥塞控制策略,很难将电子白板用于大规模会议应用之中。

组播拥塞控制是当前的一个研究热点,人们提出了很多拥塞控制方法,但至今还没有形成一个统一的标准。本文的目的是针对电子白板的特点,提出一种适合电子白板的拥塞控制策略。

虽然本文提出的拥塞控制策略只是针对电子白板和SRM提出的,但是一样可以应用于一些和电子白板数据流相似、同样需要可靠传输的一些应用,如chat、网页共享、文件发布工具等。同时希望本文能对SRM协议的进一步发展起一定的作用。

2. 原SRM的拥塞控制策略

2.1　数据流特点分析

电子白板中共有四种数据信息流。分别是绘制信息流、会议信息流、请求报文流和修复报文流。下面将具体介绍几种数据流的特点。

1)绘制信息流,也就是由使用者的绘制而产生的报文。绘制信息流的产生取决于白板的使用者,由于绘制的不连续导致了产生的绘制信息流是没有规律、时断时续的,而且每次发出的报文量也大不相同。

2)会议信息(session报文)流,其作用是报告发送者目前的状态信息,以便于接收者获取其余参与者的状态,并检测数据报文的丢失情况,会议信息流占总数据流的比重很小,一般要小于5%。会议信息流是几种数据流中最有规律的一种,SRM层每隔几个RTT时间就会发出一次session报文,报文的大小取决于所带信息的多少,但是session报文大小的上限不能超过MTU(Max Transmittion Unit)的大小(在后面将具体讲解限制报文大小的原因)。

3)请求报文(request报文)流。请求报文的主要内容是请求数据的唯一标识。与会议信息一样,请求报文的大小不能超过MTU。请求报文流的发出是没有规律的,在网络状况良好并且没有过多新用户加入的情况下,极少会发生报文丢失,所以也就不会发出很多请求报文;相反,如果报文丢失严重或是由参与者过于频繁地加入时,请求报文就会变多。

4)修复报文(repair报文)流。修复数据流是和请求报文流相对应的一种数据流,当一个组播成员收到请求报文并且他有被请求的数据时,就会发出修复报文。修复报文的产生是没有规律的,其发送频率和网络的状况有很大的关系,而且当有新的组播成员加入的时候,会造成一个修复报文的高峰。

2.2　现有拥塞控制算法

SRM现有的拥塞控制策略比较简单,其采用了一种限制发送者发送速率上限的方式来限制发送的速率。这种拥塞控制策略与SRM中所采用的应用层成帧技术是密不可分的。

应用层成帧(ALF):应用层将一个数据块交给底层传送

*)本文得到了973海量信息系统项目的资助。刘英智　硕士生,研究领域为电子白板,拥塞控制算法。吴文峻　博士生,研究领域为可靠组播,拥塞控制算法。金　天、盛向治　博士生,研究领域为拥塞控制算法,组播多媒体技术。

之前，首先会将大的数据块拆分成相同大小的较小的报文（一个报文称为一帧），帧的大小小于 MTU（最大传输单元），这种方式就被称为应用层成帧。应用层成帧的优点在于避免了由于数据块过大在 IP 层被二次切分（被切分为小于 MTU 报文），而且每一帧数据都有唯一的标识，这样当应用层发现数据丢失时，可以具体地指定某一帧的重传而不需要原来的整个数据块的重传，这样就极大地减少了修复报文的数量；而且由于每一帧中都包含该帧在整个数据块中相对的位置，因此用户可以首先处理已经收到的数据，而不必等待整个数据块中全部帧的到达，这就减少了接收者的等待时间。这也就是上面提到的限制 session 以及 request 报文大小的原因。值得注意的是 repair 报文只是修复一帧的数据，而不是整个数据块。这是因为采用了 ALF 技术，请求报文不必请求整个绘制报文数据块的重传，而只需请求绘制报文数据块的某一帧的重传。

参考图2，我们可以进一步分析以下原有 SRM 的拥塞控制策略。

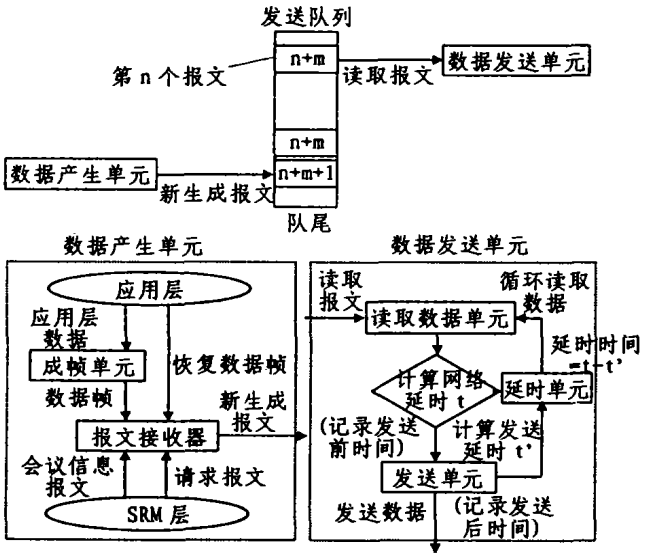

图2　原有 SRM 报文发送速率控制图

如图2所示，数据产生单元将新产生的报文不断地加入发送队列的队尾，而发送单元则不断地从队列头读取数据，并将数据按规定的速率发送出去。

在数据产生单元中，由应用层产生绘制报文和修复数据帧，其中大块的绘制报文会在应用层成帧单元中被拆分为小的数据帧，SRM 层则产生会议信息报文和请求报文，四种数据流通过报文接收器汇集起来，然后被加到发送队列的队尾。

在数据发送单元中，数据的发送是一个不断循环的过程：首先读取单元从发送队列中读出一个报文，根据当前的发送速率上限和报文的大小计算出该报文到达接收方应需的时间，即网络延时，再将数据提交给发送单元，由发送单元将数据发送出去。发送前后的时间值都会被记录下来，两者的差值被称为发送延时。之后会进入延时等待阶段，延时的时间为网络延时减去发送延时，其目的是保证报文的平均发送速率为发送速率上限，延时结束后进入下一次发送循环，读取单元又可以读取下一个数据报文。

3. 面向电子白板的拥塞控制机制原形 SRM-NEW 的实现

现有的组播拥塞控制策略主要可分为基于窗口控制和基于速率控制两种。其中基于速率的控制策略是采用一个 TCP 稳态速率公式来控制发送者的发送速率，这种方法和目前 SRM 的拥塞控制策略很相似，可以认为现有的 SRM 采用的是一种简单的基于速率的拥塞控制策略，只不过其发送报文的速率值是一个常值。而我们所要做的就是通过 TCP 稳态速率公式所反映的网络拥塞状况，动态地调整报文的发送速度，实现一种基于速率的可靠组播的拥塞控制。

基于速率的拥塞控制首先是在单播中实现的，S. Floyd 等人提出了一种 TFRC[2]（TCP Friendly Rate Control）算法，该算法以 Padhye 等人提出的 TCP 稳态速率公式[4]为基础，计算出一个 TCP 流在当前相同网络情况下（当前的 RTT 时间、报文丢失率、超时时间等情况下）的稳态的平均速率，发送者根据此速率值来调控自己的 TFRC 流发送速率。这样就可以使一个 TFRC 流的平均速率和 TCP 流在相同网络状况下的稳态速率相近，以达到和 TCP 流的友好性。TFRC 这种算法是成功的，而且已经提交为 IETF 的草案[7]。

S. Floyd 等人随即又提出了面向组播的基于速率的拥塞控制策略—TFMCC[3]。该策略仍采用 Padhye 的速率计算公式，但是 TFMCC 并没有提出一个明确的组播的拥塞控制的解决方案，而只是提出了一个大体的框架，其作者也不建议在 TFMCC 的基础之上开发任何应用。所以要根据 TFMCC 来设计面向 SRM 的拥塞控制策略是有一定的差距的，但可作为参考。

文[6]中提出了一种 SRM-TFRC 算法，其主要是针对 SRM 算法提出了一种基于公式的组播拥塞控制策略，该策略在模拟器中得到了很好的证实，对本文也有很大的借鉴作用。

我们将本文所提出的算法原形称为 SRM-NEW（SRM New Extension for Whiteboard），下面将具体地介绍 SRM-NEW 中的基于速率的拥塞控制的算法。

3.1　速率的计算

3.1.1　*速率计算公式*　基于速率的拥塞控制的基础是一个模拟 TCP 稳态速率的公式。目前为止较为准确的是 Padhye 等人所提出的公式，其公式如下：

$$B(l)=S/(t_{RTT}\sqrt{\frac{2bl}{3}}+t_0\min(1,3\sqrt{\frac{3bl}{8}})l(1+32l^2))$$

其中 B 是发送者所占用的带宽，亦即发送速率，s 是 TCP 的报文大小，l 是丢失率，t_0 是超时时间，t_{RTT} 是 RTT 时间，b 为一个应答报文所代表的接收到的报文数，b 一般为2。通过此公式就可以计算出一个 TCP 流的稳态的发送速率。

每个接收方就是根据此公式来计算应反馈的速率的值。由于 SRM 中采用了应用级成帧的技术，每个发出的报文大小是基本相同的，s 的取值为一个数据帧的大小，其他几个参数的获取将在下面的部分进行介绍。SRM-NEW 中将采用此公式。

3.1.2　*RTT 时间的估算*　现有的 SRM 中 RTT 是通过 session 报文所捎带时间信息计算得知的。

如图3所示，假设发送者 A 是一个新加入组播组的成员，他不知道到别人的 RTT 时间的值，而其他人（以接收者 B 为例）同样也不知道到 A 的 RTT 时间的值。在 SRM 中，发送方 A 会在会议信息报文中加入报文的发送时间。如图所示，发送者 A 在 t_0 发送了报文 p_0，p_0 中带有发送时间 t_0。接收者 B 在 t_1 时刻接收到了该报文，由于 SRM 中没有采用同步时钟，因此 t_1-t_0 并不是两者之间的真正网络延时（时间距离），并且注意 t_1-t_0 有可能为负值。接下来接收方 B 会在 t_2 时刻发送另一个

会议信息报文 p_1，p_1中会含有报文的发送时间 t_2以及上一步计算出来的 AB 之间的相对距离(t_1-t_0)。假设 A 在 t_3收到此报文，这时发送方 A 就可以按照公式 RTTab=(t_3-t_2)+(t_1-t_0)得到 A 到 B 的 RTT 时间。但是这个值的抖动可能会很剧烈，SRM 中采用了 EWMA 算法来平滑 RTT 的计算。算法如下：

$RTT=(t_3-t_2)+(t_1-t_0)$；
$RTT_{n+1}=(1-w)\times RTT_n+w\times RTT$。

同理，B 可以通过下一个 A 所发出的 session 报文得到 B 到 A 之间的 RTT 时间。

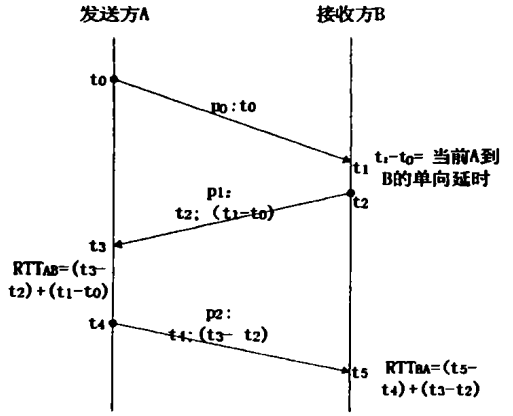

图3 RTT 计算过程图

但是这种计算 RTT 的方法不够精确，由于 session 信息的发送周期较长，会使接收方 B 的反馈等待间隔 t_2-t_1较大。当 RTT 抖动加大时，p_1所携带的单向距离时间(t_1-t_0)已经不能准确地反映出此时的 A 到 B 的单向时间距离，所以提高 RTT 计算准确度关键是通过增加携带反馈的报文数目来减少接收方的反馈等待间隔。

SRM-NEW 算法在利用会议信息报文捎带时间参数的基础之上，还利用部分绘制报文捎带时间参数来进行 RTT 时间的计算。这就要求在绘制报文中捎带报文的发送时间以及一个成员到所有其他成员的相对单向延时。由于绘制报文的发送是突发而集中的，因此就没有必要在每一个报文中加入计算 RTT 时间所需的信息，因为那样不仅增加接收方的处理时间，而且也增加了网络的负载。所以发送方应在每 n 个绘制报文加入一次计算 RTT 时间所需的参数，我们认为 n=3是一个比较理想的取值。

3.1.3 *超时时间的计算* TFRC 中给出了一种较为简单的方法，$t_0=4\times R$。其中 R 为当前的 RTT 时间，从单播的实验结果[2]证实了这种算法的可行性，所以这里将继续采用这种算法。

3.1.4 *丢失率的计算* 丢失率的计算是整个速率计算中最为重要的一个环节。但这里的丢失率并不是报文丢失率，而是丢失事件率。这是因为几种主要的 TCP 算法如 TCP-RENO 等对一个窗口内发生的多个报文丢失只做一次窗口削减，所以 TFRC 算法为了模拟 TCP 的这种行为，定义了丢失事件的概念。

丢失事件：一个丢失事件是指：当一个 RTT 时间段内至少存在一个报文丢失，我们就认为一个丢失事件发生了。也就是说丢失事件只关心一个 RTT 内是否存在报文丢失，而不关心到底有几个报文丢失。TFRC 中的丢失率 p 指的就是丢失事件率。下面将给出在 SRM 中具体计算丢失事件率的方法。

SRM 中是基于接收方进行丢失检测的。SRM 只对绘制数据的丢失进行检测，发送方将它所发出的绘制报文加上全局唯一的序号，接收方通过检查序号的不连续来检测报文的丢失。但实际上当使用者很少绘制时，白板所发送的相当大部分的报文都是非绘制报文。所以绘制报文的丢失率不能代表整个电子白板的报文丢失率，我们也就不能靠 SRM 中的编号方法(只对数据报文进行编号)来计算丢失率，而是应该对所有的报文进行编号以计算整体的丢失率。下面将逐步介绍 SRM-NEW 计算丢失事件率的方法。

报文的编号：如图2所示，现有的电子白板的四种数据流最终都是汇集到一个发送队列中再被发送出去的，所以我们可以在一个报文加入发送队列之前将其加上一个全局唯一的序号。当接收方收到这些带有全局唯一序号的报文就可以进行丢失事件率的计算。

计算丢失事件率的频率：我们认为没有必要每收到一个报文就进行一次丢失事件率的计算，因为接收方是通过 session 报文以及部分绘制报文来捎带反馈速率的，我们只需在发送反馈速率之前进行一次丢失事件率的计算即可。但是这就要求我们记录下每两个速率反馈之间的所有收到的报文的情况，每发送一次速率反馈也就开始了新的一个记录过程。由于丢失率的计算是需要历史数据的，因此我们在开始新的记录的同时要保留前一个记录的内容。

丢失事件的检测：

1)如要检测某个发送者 S 的丢失事件，应从当前记录中提取出 S 发出的全部报文信息，包括报文的全局唯一序号、报文的发送时间、收到报文时的 RTT 时间。

2)假设当前记录中收到的 S 的报文的序号是从 n 到 n+m。

3)检验上一个记录中 S 所发的最后一个报文的编号，如果编号为 n-1，则表明两个记录之间没有报文丢失；如果编号小于 n-1，则说明 n 之前存在报文丢失，如果发现丢失，则将上一个记录中 S 所发出的最后一个报文信息加入当前记录，使其成为当前记录的第一个报文。

4)假设已经发生了 l 个丢失事件，并假设最后一个丢失事件的终止报文的序号为 i(终止报文定义参见7)。

5)假设 j 是满足下列条件的最小的报文序列号

·$i<j<n+m$，

·我们没有收到序列号为 j 的报文。

6)假设 k 是满足下列条件的最小的报文号

·$j<k\leqslant n+m$，

·序列号为 k 的报文被收到，

·$T_k\geqslant T_{j-1}+RTT_{j-1}$。

其中 T_k 代表收到报文 k 的时间，RTT_{j-1}代表收到报文 j-1 时的 RTT 时间。这样我们就定义所有 j-1到 k 之间丢失的报文为一个丢失事件。

7)当 q 为满足如下条件的最大报文号，我们就认为该丢失事件的终止报文为 q，：

·$j\leqslant q<k$

·报文 q 没有被接收到。

8)检验所有当前记录，找出当前记录中的每一个丢失事件。

丢失事件率的计算：

1)事件率是通过丢失间隔求得的。丢失间隔是指两个丢

失事件终止报文之间的报文数目。假设两个相邻的丢失事件k和k+1的终止报文为i和q，则丢失间隔 $S_k=q-i+1$；

2)这里的丢失间隔的计算方法与TFRC中的算法相同，都是采用一种加权历史平均值的方法，其计算公式如下：

$$S_{new}=(\sum_{i=0}^{n-1}w_{i+1}S_i)/\sum_{i=1}^{n}w_i$$

S_{new}代表加权历史平均丢失间隔，S_i为最近的第i个丢失间隔，S_0代表最近丢失事件的终止报文到当前记录最后一个报文之间的距离。n的取值为8，$w_1,w_2,w_3,w_4=1$；$w_5=0.8$；$w_6=0.6$；$w_7=0.4$；$w_8=0.2$。

丢失事件率$=s_{new}$的倒数。

3.2 速率调控机制

3.2.1 *接收方的速率反馈机制* 与TFRC所面向的流媒体不同，电子白板并不苛求数据流的平滑性，只要能满足与TCP流的公平性即可。所以电子白板没有必要向TFRC那样频繁地进行速率的反馈，只需利用session报文以捎带的方式进行速率反馈即可。session报文一般以3－10个RTT为周期，从单个反馈的角度看可能稍显缓慢，但是组播中的每个接收者都是反馈者，从整体的角度看发送者所能得到的反馈的频率还是很高的。

3.2.2 *发送方的速率调控机制* 我们首先定义一个参与能力的概念。参与能力是指一个组播参与者所加入的网络的拥塞程度以及参与者对主机的处理能力。参与能力直接关系到一个组播参与者所能提供的发送速率以及能够忍受的接收速率。

在组播环境中，一个发送者会面临如何根据多方的速率反馈值来进行调整的问题。由于SRM-NEW主要是针对参与能力相近的群体提出的，因此我们将采用一种平均值的方法来将所有的反馈取平均值，希望这种方法可以达到一种整体的公平性。对于那些参与能力相差很大的群体就需要采用分组组播的方法，这在展望中会有较细的说明。下面是SRM-NEW中所采用的速率调节机制图。

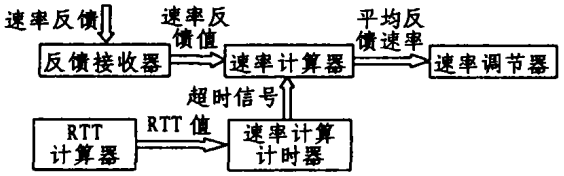

图5 SRM-NEW中的速率调节机制图

由于每个接收者都在向发送者反馈速率，这样发送者收到反馈的间隔是没有规律的。所以不能每收到一个速率反馈就进行一次速率调整，发送方应有一种比较固定的速率调控周期。在SRM-NEW中，发送者会每隔一个平均RTT时间进行一次速率的调整，平均RTT时间是发送者到所有接收者的RTT时间的平均值。RTT计算器使速率计算计时器得知当前的平均RTT时间，计时器循环不断地按平均RTT的值设置超时时钟，一旦超时，计时器就会向速率计算器发送一个超时信号。速率计算器每收到一个超时信号就会从反馈接收器那里读取一次所有接收者的最近一次的速率反馈，将这些反馈取平均值，再将此平均值传递给速率调节器，速率调节器会将当前的发送速率设为该平均值。

3.2.3 *慢启动* 为了模拟TCP的慢启动策略，TFRC采用了类似的算法：在一个RTT时间内，如果没有发生报文丢失则将速率调整为$R=2\times\min(R,Rrcvd)$，其中Rrcvd是接收者的接收速率。

组播中的慢启动过程所遇到的问题主要有两点，一是如何使慢启动者能够尽快地得到接收方的反馈，以避免自身发送速率的过分增长；二是如何在组播环境下尽量地减少反馈报文的个数，防止由于慢启动而引起的拥塞崩溃，尤其是在多个人同时加入的情况下。下面是我们在SRM-NEW中所提出的解决方案：

(1)为区分慢启动过程，发送者所发送的所有报文的"慢启动标志位"都会被设为1。

(2)首先发送一个"加入报文"，接收者一旦收到这个加入报文就会以最高的优先级对此报文进行反馈。为了防止多个接收者可能会同时发出反馈而造成反馈风暴，同样要对这个反馈加上一段延时。

(3)如果慢启动者在发出"加入报文"后的一段足够长的时间之内没有收到任何反馈信息，则可能有两种情况：一是"加入报文"或是反馈报文中的一方由于网络拥塞出现了丢失，二是此新加入者是该组播组中唯一的成员。为了区分这两种情况，我们会检验新加入者从加入组播时起到此后的一段足够长的时间内是否收到同组的报文，如果收到了同组的报文则表明此新加入者并不是该组播组的唯一的成员，所以是由于网络拥塞导致没有收到反馈报文，这时我们会中止慢启动过程，开始以最低的发送速率(每个RTT发送一个报文)发送报文；如果在一段足够长的时间内没有收到任何的同组成员的信息，则表明此参与者目前正处于一种孤立的状态，这时发送者会以一种较低的速率发送报文。一个孤立的成员发送任何绘制报文都是没有意义的，所以处于孤立状态的参与者将只发送会议信息报文，直到发现别的参与者才发送绘制报文。

如果慢启动者(简称S)在规定时间内收到了反馈报文，S会向收到的第一个反馈的发送者A单播一个特殊报文的应答，并和A之间以TCP的方式建立一种应答机制：以后A会每隔一个RTT时间向S报告一次收到的S的报文速度。S如果在2个RTT时间内收到此报文速度反馈就会将发送速度按$R=2\times\min(R,Rrcvd)$的方式增长，如果在2个RTT时间内收不到A的反馈则认为网络出现拥塞而结束慢启动过程。

任何一个接收者一旦发现S所发出的报文存在丢包，就会以最高的优先级组播出一个特殊的丢失警告报文，S一旦收到此报文就立即结束慢启动的过程。为防止警告报文所可能造成的反馈风暴，发出警告报文之前也会按SRM的方法加上随机延时以抑制反馈风暴的发生。

慢启动的结束：发送者一旦结束慢启动过程，则会以原有速率的一半发送报文，直到其收到接收方的速率反馈。

3.3 其他问题

3.3.1 *后加入者的问题* 后加入者的问题是组播所特有的问题，因为组播不像单播那样要事先建立收发双方间的连接，一个组播参与者可以随时地加入或是退出一个组播组，这也就带来了后加入者的问题。

一个后加入者的主要问题是历史数据的获取问题。在一个像SRM这样的可靠组播协议中，一个后加入者为了得到他所没有得到的历史数据(绘制报文)会发出请求报文，接收方收到请求报文后会发出相应的修复报文。在一个长期的会议中，如一个参与者较晚加入则会有很多的数据需要修复。虽然当前的SRM中已经对修复的范围做了限制，即只修复当

前页的数据，但当多个人同时加入或是新加入者不断地在不同页面之间切换时，同样会造成大量的请求报文的发出，随之而来的是更大量的修复报文的发出，这些都可能会带来网络的拥塞。下面将分别对两种现象提出解决的方案。

(1)多人同时加入。当多人同时加入时，每个人都会对默认当前页(第一页)的历史数据提出请求，也就是说他们请求的内容是相同的。由于修复报文的发出是以组播的方式发出的，一个修复报文会使所有的新加入者的空缺数据得到修复，因此就没有必要让每个人都发出请求。可以在所有新加入者中选出一个代表来发送请求报文，如果隔一段时间还没有得到修复报文，新加入者再发出自己的请求报文。第一个发出请求的人将被指定为代表，他所发出的请求报文会加入代表标志位，其他新加入者如果收到了代表的请求就不再对第一页的内容提出申请。但可能存在多个人争夺代表的情况，这时可以采用一种退避的方法，通过某个唯一标识(比如说IP)的比较来决定谁是代表。

2)后加入者的页间切换。对于一个后加入者，当他读完一页的内容会很自然地翻到下一页或是下几页继续阅读。由于SRM中只对当前页面进行修复，因此其翻到新页时可能要较长的时间才能得到新页的历史数据。为了提升新页历史数据修复速度，SRM-NEW中采用了如下的方法。

当对某一页进行修复的时候，同时以较低的速度发送相邻几页的请求报文，只要一个新参与者的翻动页的幅度不是很大，就能以较快的速度得到新页的历史内容。

3.3.2 *"空闲"用户的问题* 在使用电子白板的过程中，一个成员可能会长时间地转移他的注意力而不再关注电子白板，我们称参与者的这种状态为"空闲"状态。这时参与者虽不会发送任何的绘制数据的报文，但是其他的报文如session报文、请求以及修复报文的发送都是照常的。由于这时的成员已经不再关心会议，因此就没有必要给该成员以过多的带宽，完全可以将该成员的会议报文以及请求报文的发送数率降为平均反馈速率值的一半或是更低以减少带宽的占用，但是这里并不降低修复报文的发送速度，因为那样会降低其他用户的修复速度。

空闲状态的判定有以下几种方式。如果一个用户将电子白板的应用程序最小化的时间超过n秒，则认为该用户对电子白板的注意力很低，可将其状态设为"空闲"；如果电子白板的窗口没有被最小化，但是窗体不被激活的时间超过m秒，则认为该用户处于空闲状态(n与m的大小可定为30)。

一旦一个用户被判定处于空闲状态，他的部分数据的发送数率会被设置为应有速率(接收方的反馈速率的平均值)的一半。当用户激活了电子白板窗口，则结束用户的空闲状态，用户的发送速率值会以一种平缓的方式上调到应有的速率值。

总结与展望 SRM-NEW虽然改进了SRM中原有的速率控制机制，但是还有很多值得改进的地方。主要有以下3点：

1)对于大规模的组播应用，参与者的参与能力必然存在着极大的差异。发送者无论以任何速率组播数据都难以适应所有接收者的不同状况。文[9]中给出了一种分组组播——Scattercast的思想：将参与能力相近的用户分为一组，每组有一个代理，代理之间用TCP链接，组内采用SRM协议进行组播。用户可以根据自身的情况选择不同的代理，这样就将一个复杂的组播群体切分为很多参与能力相近的组，这也就避免了由于参与能力不同所带来的速率调控的难题。我们认为这也是SRM-NEW下一步的发展方向。

2)SRM-NEW中的不同数据流的发送还可以采用HF-SC[8]的调度算法，根据不同数据流的优先级采用不同的发送策略。

3)SRM-NEW目前还只是为电子白板这种数据流非连续而且不要求数据流平滑性的应用而设计的，下一步可以进一步改进其数据流的平滑性，使其更具通用性。

致谢 感谢李未老师与马世龙老师对本文的指导。同时要感谢卢剑、聂树瑄等人对本文的宝贵修改意见。

(下转第102页)

2002年全国理论计算机科学学术年会

会议征文通知(第二版)

由中国计算机学会理论计算机科学专业委员会主办、中南大学信息科学与工程学院承办、湖南省计算机学会和湘潭大学信息工程学院等协办的"2002年全国理论计算机科学学术年会"将于2002年10月在湖南长沙召开。会议录用论文将在《计算机科学》、《计算技术与自动化》正刊和《《计算机科学》2002年全国理论计算机科学学术年会论文专辑》于2002年9月前后正式出版，欢迎大家积极投稿。现将有关征文要求通知如下：

1. 应征论文应未在其他刊物或学术会议上正式发表过。特别欢迎有创见和有应用前景的论文。

2. 征文范围：程序理论(程序逻辑、程序正确性验证、形式开发方法等)；计算理论(算法设计与分析、复杂性理论、可计算性理论等)；语言理论(形式语言理论、自动机理论、形式语义学、计算语言学等)；人工智能(知识工程、机器学习、模式识别、机器人等)；逻辑基础(数理逻辑、多值逻辑、模糊逻辑、模态逻辑、直觉主义逻辑、组合逻辑等)；数据理论(演绎数据库、关系数据库、面向对象数据库等)；计算机数学(符号计算、数学定理证明、计算几何等)；并行算法(网络计算、分布式并行算法、大规模并行算法、演化算法等)

3. 投稿说明：1)提供中、英文论文名称、关键词、摘要、作者名(所有作者)、单位、联系地址、邮编、E-mail及电话，(详见《计算机科学》的格式要求)，投稿者提供电子稿、文稿各一份。2)一篇论文一般限3页面(大16开)，与会交流的论文方被采纳，未收录的论文不退稿。3)投寄地址：(410083)湖南长沙岳麓山中南大学信息科学与工程学院 刘 明 收

4. 重要日期：征文截止日期：2002年6月30日；论文录用通知：2002年7月30日；修改截止日期：2002年8月15日

5. 联系人：陈志刚，0731-8830797，czg@csu.edu.cn 刘明，0731-8876677(Tel./Fax)x-info@csu.edu.cn 周前，0731-8830700，infob@csu.edu.cn

识和信息交换，Java 和 CORBA，MAS 结合可以保证医学图像协调系统的实现，软件体系结构如图6所示。

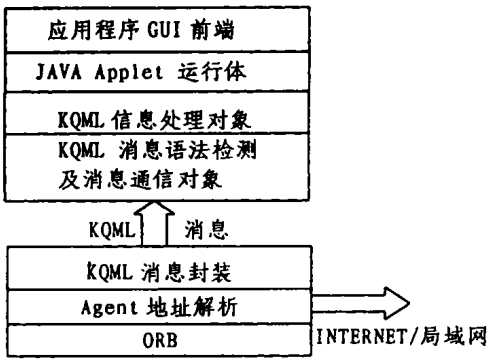

图6 协调 agent 软件体系逻辑结构

6. 医学图像分割协作的实现

医学图像在辅助医生对病症的诊断和临床实践以及医学研究方面上具有很重要的意义和广阔的应用前景，三维医学图像的发展已有20多年的历史，从三维图像数据的重建、显示到分析方面的研究，都做了大量工作。对于三维图像分析方法的研究，一个最基本且重要的问题就是图像分割方面，在单机上进行分割，已经产生了很多有意义的算法，如区域分割和表面边界分割，但是各个算法各有特色，针对的应用范围也各不相同，而且医学图像往往具有复杂的结构，包括不同的组织，如骨骼、软组织等，由于各种图像信息具有互补性质，通常通过对同一个部位的不同模式的医学图像信息进行集成来获得临床应用中所需要的全面信息，需要对不同的组织和图像模式采用不同的分割算法进行处理，这就需要网络协作处理技术。本文利用软件代理的分布式协作结构及 KQML，对三维医学图像的协作分割进行了算法设计，协作的流程如下：

1. 由用户界面输入欲分割图像或通过界面从网络浏览得到，并且输入分割目标及评价指标。向调节者(facilitator)注册登记。

```
Bind
: content tcp://medical: 8010
  /* 采用 TCP 协议，端口为8010，主机为 medical
: sender user
: receiver tcp://facilitator : 1000
: in-reply-to ID1 / * 信息标志
: language Prolog
: ontology Med / * Med 是有关医学处理术语的集合
```

2. 用户初始化，接收目标，向网络中的调节者(facilitator)发送请求 Query，获得能完成处理目标的 agent A (Query-Result)，agent A 评估自己的能力，决定是否接受处理目标，并将决定结果返回 facilitator，若接受任务，agent A 从自身知识库中得到图像划分算法，将图像进行划分，开辟共享空间，并定好任务处理表 agent A 向 facilitator 发送请求寻找协作伙伴。

```
Query
: content image segmentation
: sender user
: receiver tcp://facilitator:1000
: in-reply-to ID2
Query-Result
:content Agent A: image segmentation,
: sender server / * facilitator 名字
: receiver tcp://medical: 8010
: reply-with ID2
```

3. facilitator 通过注册表寻找到协作伙伴并将相应的 URL 反馈给 agent A(相应原语限于篇幅省略)，agent A 根据 URL 将任务进行分配，如 agent B 进行区域分割，agent C 进行表面分割等，各协作代理准备就绪则开始协作处理。相应参数和结果及任务流程可以在共享空间中进行交换和变更。

4. 各代理进行协作处理，不断查询分割进展，传递相关的信息，其间可终止某一代理的工作，再选择其他代理进行。等各个代理处理完毕后，agent A 将信息集成，将结果返回用户，并进行指标评定，退出。不满意则重新开始分割进程。

结论 医学图像协调处理是为了解决当前医学影像处理系统中计算量大，时间效率低，图像质量差等客观问题而提出的。本文将 KQML 语言和多 agent 协作机制引入到协调处理中，能较好地解决协调过程中的信息交换与知识处理等问题，并且在实际应用中，KQML 语言建立在 CORBA 底层通信机制上，有机地和 CORBA 结合，KQML 提供了标准的信息交换格式，为各 agent 之间信息传递提供了有力支持。采用 KQML 和 agent 机制的医学图像协调系统具有开放性并能适应新的应用环境，尤其是 Web 应用环境。

参 考 文 献

1 Huang J, Jennings N R, Fox J. Cooperaton Distributed Medical Care. In: Proc. Second Int. Conf. on Cooperative Information Systems (CoopIS-94), Toronto, Canada, 1994. 255～263
2 Lanzola G, et al. a framework for building cooperative software agents in medical applications. Artifical Intelligence in Medical, 1999, 16: 223～249
3 Horsch A, et al. Collaborative Work with Medical Images in a University Hospital Environment, Proceedings JENC7
4 Becker E, et al. a system for cooperative work in the medical domain. http://www.gris.uni-tuebingen.de/.
5 Finin T, Fritzson R. KQML as an agent communication language, the Proceedings of the third international conference on information and knowledge management (CIKM'94), ACM Press, Nov. 1994
6 孙宪鹏，张宇，等. 基于 KQML 语言的合同网协议模型及实现. 信息与控制，2000, 29(5): 454～460

(上接第36页)

参 考 文 献

1 Floyd S, et al. A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing. In: Proc. of ACM Sigcomm'95. 342～356
2 Floyd S, et al. Equation-Based Congestion Control for Unicast Applications : the Extended Version. Feb 2000. http://www.aciri.org/tfrc/
3 Handley M, Floyd S. Strawman Specification for Reliable Multicast Congestion Control. http://north.east.isi.edu/～mjh/rmcc.ps.gz
4 Padhye J, et al. Modeling TCP throughput: a simple model and its empirical validation. ACM SIGCOMM'98, Vancouver, CA, Sep. 1998
5 Padhye J. Towards a Comprehensive Congestion Control Framework for Continuous Media Flows in Best Effort Networks. A Dissertation of J. Padhye. March 2000
6 Wu Wenjun, Xu Ying, Lu Jian. SRM-TFRC: a TCP-friendly multicast congestion control scheme based on SRM². ICCN-MC2001
7 Handley M, Pahdye J, Floyd S, Widmer J. TCP Friendly Rate Control (TFRC): Protocol Specification. Internet draft draft-ietf-tsvwg-tfrc-02.txt, work in progress, May 2001
8 Stocia I, et al. A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and Priority Services. IEEE/ACM Transactions on Networking, 1997, 9
9 Chawathe Y, et al. RMX: Reliable Multicast for Heterogeneous Networks. INFORCOM 2000, Tel Aviv, Israel, March 2000