交换式以太网中连续实时流媒体的可靠组播*

王 军+, 吴志美

(中国科学院 软件研究所 多媒体通信与网络研究开发中心,北京 100080)

Reliable Multicast for Real-Time Continuous-Media Streams over Switch Ethernet

WANG Jun+, WU Zhi-Mei

(Multimedia Communication and Network Engineering Research Center, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: +86-10-62645407, Fax: +86-10-62645410, E-mail: wyj@iscas.ac.cn, http://www.iscas.ac.cn

Received 2002-11-26; Accepted 2003-01-28

Wang J, Wu ZM. Reliable multicast for real-time continuous-media streams over switch Ethernet. *Journal of Software*, 2004,15(2):229~237.

http://www.jos.org.cn/1000-9825/15/229.htm

Abstract: A reliable multicast protocol for real-time continuous-media streams on switch Ethernet after research of the LANs' special features for reliable multicast is proposed. The recovery of this protocol begins with the receiver's detection of packet loss by packet's number, and the second step of it is the NAK (negative acknowledgment) suppression procedure, and then the sender will retransmit the lost packet by the received NAK, and finally the receiver will discard or accept the recovered packet based on the limit of the real-time playback. The experimental results and the analysis of performance demonstrate the effectiveness and reliability of the approach.

Key words: reliable multicast; NAK (negative acknowledgment) suppression; real-time continuous-media stream; roundtrip delay

摘 要: 对目前局域网中连续实时流媒体的可靠组播进行了研究.首先分析了局域网影响可靠组播的几个特性,然后设计了局域网内针对连续实时流媒体的基于重传的可靠组播传输协议.协议中的错误恢复首先利用接收者检测包丢失,同时采用 NAK(negative acknowledgment)抑制减少产生的反馈包;然后发送者根据 NAK 请求重传已经丢失的数据包;接收者根据本地播放时间限制决定接收或者丢弃迟到的重传包.实验结果和性能分析证明了该可靠组播协议的可靠性和有效性.

关键词: 可靠组播;NAK(negative acknowledgment)抑制;连续实时流媒体;往返延迟

中图法分类号: TP393 **文献标识码**: A

* Supported by the National Grand Fundamental Research 973 Program of China under Grant No.G1998030405 (国家重点基础研究发展规划(973)); the Project of the Beijing Committee of Science and Technology of China under Grand No.H011710010123 (北京市科学技术委员会资助项目)

作者简介: **王军**(1976－),男,河南汤阴人,博士,主要研究领域为多媒体数据压缩,网络通信;**吴志美**(1942－),男,研究员,博士生导师,主要研究领域为多媒体通信,宽带网络,IPv6.

采用组播技术的连续流媒体应用可以在计算机网络上以较小的网络开销提供与传统广播电视和电台相同的业务.这些组播应用有严格的时间要求和可靠的传输要求,要求网络同时考虑实时传输同步和传输可靠性的服务质量要求.有限的处理能力、传统的包交换网络机制和尽力而为服务模型是网络丢包的根源,组播应用中错误恢复和控制过程对数据传输可靠性性能的影响主要取决于两个方面[1]:终端处理能力和网络拥塞状况.

近年来,针对可靠组播的错误恢复机制大致有基于重传的错误恢复、基于前向纠错编码的错误恢复、基于数据重构的错误隐藏技术、基于交织编码的错误恢复技术[2]等几种.文献[3,4]对目前的可靠组播协议进行了分类:基于 ACK(acknowledge)重传的可靠组播协议[5,6]、基于 NAK(negative acknowledgment)重传的可靠组播协议[7~9]、基于令牌环的可靠组播协议(TRP/RMP)[10]和基于树结构的可靠组播协议[11~13].这些协议多数应用于 Bulk-Data 传输,主要应用于白板共享、工作组共享、文件传输和分布式系统中的消息传递等环境中,针对时间受限连续流媒体的可靠传输研究比较少.在有限的几个应用于连续流媒体的可靠传输协议中,多数集中在广域网和 Internet 环境中的可靠组播传输.例如,文献[14,15]中研究的是 Internet 异构环境中接收者如何根据自身条件自适应地选择 FEC(forward error correction)或者 ARQ(automatic repeat request)进行错误恢复,目前的可靠组播协议还没有考虑 LAN(特别是支持二层组播的交换式以太网)特性对可靠组播应用造成的影响.

因此,我们在本文中将重点结合局域网特性和连续流媒体应用的时间特性,研究在局域网环境中,在严格时间约束的条件下进行连续流媒体可靠传输的方法和机制,并且设计相应的实现方法.本文描述的针对连续实时数据流的可靠组播协议具有如下特点:IP 组播和二层组播作为高速交换式以太网上的组播支持;NAK-based 反馈和 NAK 抑制;错误恢复分级重传机制;时间受限条件下的可靠传输.

本文第 1 节分析在局域网中进行连续实时流组播需要考虑的几个因素,包括分析组播传输中数据丢失产生的原因、连续流媒体端到端的传输模型和数据生存期等.第 2 节描述本文提出的可靠组播协议的主要内容:NAK 抑制过程、分级的可靠传输服务、实时性限制和缓冲方案设计等.第 3 节给出该协议的性能分析.最后得出结论.

1 可靠组播中的关键技术

1.1 连续流媒体基于重传的端到端传输模型

连续流媒体的播放质量受编码算法、往返延迟、延迟抖动和差错控制这几个因素的影响.交互式实时流媒体应用对往返延迟和延迟抖动比较敏感,实时交互的重要性高于可靠性,所以一般采用容错系数较高的编码算法,例如 G.723 和 H.263+等,很少采用基于重传的错误恢复方法.本文所研究的是回放型的连续流媒体应用,其特点是允许客户端有很长的初始连接时间,因此允许采用基于重传的错误恢复方法,但是播放过程中客户端对数据丢失和实时同步播放比较敏感,因此又需要有一定的时间限制.这种应用的客户端通常采用预先缓冲数据的方式,在丢失数据对应的播放时刻到来之前进行重传恢复.

连续流媒体基于重传的端到端传输模型如图 1 所示.

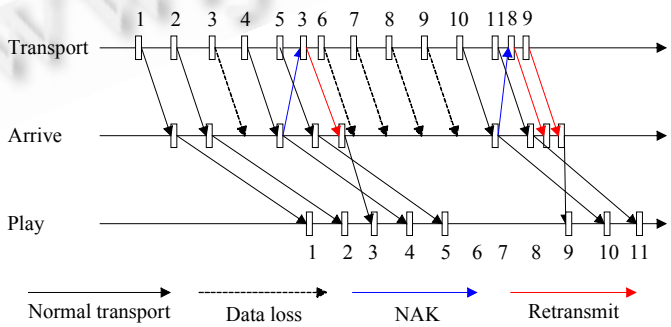

Fig.1 Retransmit-Based terminal-to-terminal transport model of continuous-media streams

图 1 连续流媒体基于重传的端到端传输模型

接收者接收到数据包 4 时检测到数据包 3 的丢失,产生反馈 NAK_3,发送者重传第 3 个包,接收端在播放时刻之前收到数据包 3,插入到播放缓冲中进行正常解码播放.由于接收者检测到数据包 6 和 7 丢失的时刻(接收到数据包 10 的时刻)已经超过了它们的播放时刻,因此接收者不对包 6 和 7 产生 NAK 要求.数据包 8 的重传到达时刻大于它的播放时刻,因此重传包 8 被接收者丢弃.

从图 1 中可以直观地看出,预先缓冲的时间(包 1 播放时刻和接收时刻之差)与可以被正确恢复的最大连续丢包数有关,同时往返时间决定了恢复单个包丢失所需的最小缓冲时间.

1.2 可靠组播中的数据丢失分析

1.2.1 数据丢失的特性

许多研究[16~17]表明,由于采用包交换网络和尽力而为的服务模型,网络传输中的数据丢失是无法避免的,特别是在一个规模较大的会话过程中,数据报的丢失更加普遍.文献[17]显示,在 Mbone 的组播会话过程中,大部分接收者都会经历 2%~5%的数据丢失.文献[2]阐述了网络中数据丢失的特性:大部分数据丢失是来源于网络瞬时拥塞所引起的“单个包”的丢失,两个和两个以上连续数据报丢失的情况比较少见,因此,可靠传输研究的重点应该是针对“网络瞬时拥塞”所引起的数据丢失.本文的协议设计过程主要考虑的就是连续包丢失个数 $n<5$ 的丢失情况.对于“网络严重拥塞”情况下的数据丢失,进行 FEC 和重传等错误恢复反而会加重拥塞状况,导致更多的数据丢失,设计者应该考虑的是对拥塞制造者进行流量控制.

1.2.2 数据丢失检测的方法

根据检测者的身份不同,可以将数据丢失的检查分为 Sender-Initiated 和 Receiver-Initiated 两种方法.Sender-Initiated 方法也称为基于 ACK 的方法:接收者为每个包产生惟一正确反馈 ACK,发送者匹配反馈和已发送的数据包,没有收到反馈的数据包被认为是被丢失的报文.Receiver-Initiated 方法又称为基于 NAK 的方法:接收者检查正常接收的数据包的序号,不连续的序号意味着数据包的丢失,接收者向发送者报告标识数据丢失的 NAK 反馈.由于不同传输路径导致的接收乱序所引起的包序号不连续情况在局域网环境中极为少见,而且局域网中的数据丢失率非常小,接收者产生 NAK 的数目远小于 ACK 的情况,所以,在本文中进行错误检测时,采用检测不连续的包序号的数据丢失检测方法,接收者为丢失的包产生 NAK.

1.2.3 数据丢失产生的原因和处理方法

在可靠组播过程中,有效数据的丢失发生在 4 个地方:发送端、发送端网络、接收端网络和接收端(以太网是平坦型的拓扑结构,子节点数目很多,中间节点和链路较少,其传输特性类似于发送端网络,因此可以将中间节点和链路归结到发送者网络).

发送端数据丢失的主要原因是发送者的处理资源不足.发送者的处理资源花费在发送原始组播数据,处理反馈信息和进行重传处理.在服务器(发送者)设计时一般都会考虑负载问题,在普通传输情况下,发送者的原始组播数据造成的负载不会超过设计上限.发送者处理反馈的开销与组规模和采用的反馈机制有关,为了便于分析,将接收端所经历的数据丢失率分为两部分,至少被两个接收者所经历的数据丢失记为公共的丢失率 P_0,只被单个接收端所经历的丢失记为独立丢失率 $P_i(i=1,2,...,n)$,n 是整个会话中的成员数量,P_0 可以看作是发送端和发送端链路造成的公共丢失,P_i 可以看作是接收端和接收端链路造成的丢失.由于接收端之间相关性很小,P_i 可以认为是独立分布过程,每个接收者所经历的丢失率为 P_0+P_i,整个组播会话的丢失率为 $P_0+\sum P_i$.再假设每个反馈占用处理器开销为 1 个单位,每个重传占用开销为 1 个单位,则对于任一个数据包,几种反馈机制对发送者所产生的处理器开销见表 1.

Table 1 Comparison of additional feedback cost

表 1 额外反馈开销的比较

	ACK	NAK	Retransmission
Unicast	$N(1-P_0)+\sum(1-P_i)$	$nP_0+\sum P_i$	$nP_0+\sum P_i$
Multicast	Token-Ring: $(1-P_0)+\sum(1-P_i)$	$P_0+\sum P_i$	$P_0+\sum P_i$

因为在局域网环境中,$P_0<<1$ 且 $P_i<<1$,则有 $n(1-P_0)+\sum(1-P_i)>>\max(nP_0+\sum P_i,(1-P_0)+\sum(1-P_i))>>P_0+\sum P_i$,可知,采用组播 NAK 抑制的方法所占用的发送者开销最小,$nP_0+\sum P_i>>P_0+\sum P_i$ 的结果表明,采用组播进行数据重传

对降低发送端数据丢失概率有非常积极的意义,特别是在规模较大的组播会话中,这种作用更加明显.

在发送端网络引起的数据丢失通常是由当时的网络拥塞状况引起的,包括瞬时和长期的拥塞.网络拥塞的直接原因是网络流量的输出速率超出了网络所能接纳的上限,与可靠组播相关联的 3 种流量是原始发送数据、反馈等控制信息和重传数据,其中反馈等控制信息是由接收端发往发送端,在以太网的全双工工作模式下对有效原始数据的传输不会有影响,即使在半双工模式下,反馈信号在整个流量中所占比例也非常小,几乎可以忽略不计.例如,假设 MPEG-2 流组播会话中丢包率 P<5%,原始数据流占用 5.6Mbps 带宽,每个包 4KByte 左右,一个 NAK 包按 100Byte 计算,这时可以计算出所需的 NAK 带宽约 7Kbps<<5.6Mbps,占整个会话带宽的极小部分.错误恢复中重传数据的流量被数据丢失率决定,在网络发生拥塞时,会产生更多的重传数据,从而导致网络状况的进一步恶化,因此需要对重传数据加以限制,不能影响原有的数据传输.因此,在我们的设计方案中增加了对重传数据的速率控制,重传的数据流带宽不超过原始数据流的 10%(认为数据丢失率>10%时,网络上出现了严重阻塞,这时重传数据流的带宽不应再增加).

如果采用组播重传和组播 NAK 方式,接收端网络与发送端网络的流量占用和数据丢失状况保持一致.单播情况下,接收端网络的负担较轻,只与本地的接收状况有关.从整个系统性能考虑,发送端网络影响所有接收者,其带宽的价值是接收端网络的 n 倍(n 是组成员的个数),其重要性要高于接收端网络,只有在不影响发送端网络性能的情况下,才考虑接收端网络的流量占用情况,因此,采用组播重传和组播 NAK 成为首选的方案.对于由于接收端处理能力不足导致的丢包,也必须在不加重发送端负载的基础上进行考虑.在设计过程中,我们对重传数据流进行了流量控制,实际上也就控制了会话中规定的丢失率上限,对于数据丢失率最大的接收者(常称为 crying baby),协议只保证在给定范围内的数据恢复,高于这个范围,发送者不再进行重传服务,从而防止单个接收者的过多错误恢复请求影响整个会话中其他成员的正常接收.

1.3 交换式以太网特性对可靠组播的影响

1.3.1 组播支持

与广域网结构相比,以太网结构的局域网本身就支持组播.最早以共享介质为基础的以太网采用广播方式进行“全网组播”,目前应用广泛的交换式以太网采用 IGMP Snooping, CGMP 和 GMRP 等方式增强了对组播范围限制的支持,实现针对组成员的“组内组播”.无论采用何种方式,在以太网中通信设备组播发送一个包给所有组成员接收者与发送一个包给一个接收者所承受的开销几乎一样,相比而言,广域网中基于软件选路的路由器需要为每个端口复制组播包,开销要大得多.因此,在以太网中权衡单播和组播策略时,应该优先考虑组播策略,例如数据重传和反馈采用组播实现.

1.3.2 平坦型拓扑和低延迟

在局域网的一个组播会话过程中,参与者的地域位置非常接近,系统结构和性能比较接近,传播延迟和往返时间非常小而且比较平均,这种情况属于“平坦型”的网络拓扑,因此,在广域网中比较突出的“异构性”问题在局域网中并不明显,基于反馈和重传汇聚的算法对系统性能的提升作用不大,在广域网中效果很好的树形结构可靠组播反而不适合在局域网环境中应用.低延迟的另一个影响是增大了采用重传作为错误恢复手段的可能性.重传与 FEC、错误隐藏和交织编码相比,主要的缺点是重传引入了额外传输延迟,在广域网环境中的这个额外延迟会非常大,分布不均匀,降低了整个会话的实时性,而在局域网中,平均往返时间一般小于 5ms,对组播会话的影响微乎其微.

1.3.3 传输高可靠性

在局域网中进行网络传输的另一个优点是高带宽、低传输误码率.局域网中的数据丢失大多数是由于网络拥塞或者是通信终端处理能力不足,传输过程中出现错误的比例极小,在这种环境中考虑可靠传输的性能,更多的是考虑如何降低错误恢复过程占用的终端处理能力.

1.3.4 优先级策略

广域网中因为要适应不同的网络结构和设备,所以对不同组播组的数据没有进行优先级控制等智能策略,交换式以太网中的智能化网络设备(通常是智能交换机)可以利用自身的带宽分配和 CBQ 等优先级控制策略,

为可靠组播协议中的不同数据提供不同的传输等级.例如,原始数据包采用最高优先级的组播组,反馈等控制报文采用低一级优先级的组播组,重传报文采用最低优先级的组播组,同时可以对控制报文和重传报文实行带宽限制.采用这种分级传输和流量控制的 QoS 方法,可以使错误恢复过程避免影响原有的组播发送过程,避免在网络负担较重的情况下引起整个会话质量的下降.

2 可靠组播协议描述

本节将详细描述针对交换式以太网上连续流媒体的传输特点设计的可靠组播协议的执行过程.

2.1 基本流程

可靠组播的基本过程包括 4 个阶段:第 1 次发送;错误检测,包括 NAK 产生和 NAK 抑制;发送者重传;接收者播放.第 1 次发送过程与普通 IP 组播过程没有任何区别.在连续流媒体可靠组播协议中,接收者的播放过程主要是对数据的生存期进行检查,在播放时刻之前到达的数据包被插入到播放缓冲当中,其他迟到的数据包被简单丢弃或者进行解码过程的错误恢复.

在初始加入时,接收者根据从其他方式得到的发送者信息,加入两个组播组,其中一个发送/接收原始数据,另一个发送/接收 NAK 和重传数据.没有部署可靠组播协议的接收端只需加入第 1 个组播组,这种方式使可靠传输协议能够兼容以往的组播系统,方便原有系统的升级.资源有限的接收端在加入两个组以后,如果经历的数据丢失不能全部恢复,在错误积累到一定程度时(经过恢复后的丢失率仍>5%,或者解码器无法达到所要求的解码质量),该接收者会主动离开第 2 个组播组.这样,既可以减少它(crying baby)对原有可靠组播会话的影响,又可以使它增加自己的处理能力和本地可用带宽,减小数据丢失的可能性.

2.2 可靠组播中的实时性限制

在连续实时流媒体的可靠组播协议中,实时性的因素影响了发送者和接收者的重传和反馈策略,并且影响了系统的缓冲设计和质量保证参数.下面我们将分析在给定往返时间以后,为了达到最大可恢复的连续丢包数,发送端数据最小缓冲时间、接收端预先缓冲时间以及发送数据包时间间隔之间的联系和设计要求:

假设数据包调度发送时间用 S 表示,到达时间为 A,播放时间为 P,重传时间为 R,数据包之间的传输间隔为 T_{interval},往返时间为 $2T_r$(T_r 可以认为是端到端的平均延迟),接收端检测到数据丢失后的随机等待时间用 $T_{\text{wait}}=T_r$ 表示(已知 $2T_r>T_{\text{wait}}>0$,取平均 T_{wait} 作为分析时的取值),接收端收到第 1 个包时的预先缓冲时间用 $T_{\text{buffer}}=P_1-A_1$ 表示.为了便于分析,采用全局统一时钟为基准.根据定义可知,$P_i=A_i+T_{\text{buffer}}$,$A_i=S_i+T_r$,$A_{i+n}=A_i+nT_{\text{interval}}$.几种典型的传输过程描述如下:

无丢失传输:数据包 i 从 S_i 开始,于 A_i 到达接收端,并在 P_i 时刻播放.

丢失与恢复:数据包 i 在 S_i 发送,在 A_{i+n} 被检测到丢失(从 i 开始连续丢失 n 个包,n=1 时只丢失第 i 个包),随机等待时间 T_{wait} 以后,接收者发出 NAK 信号,发送端在 $R_i=A_{i+n}+T_{\text{wait}}+T_r$ 时刻收到 NAK 以后开始重传,重传数据在 R_i+T_r 时间后到达接收端,随后在 P_i 时刻被播放.

由于实时播放的限制,丢失的包 i 被正常恢复的必要条件是 $P_i>R_i+T_r$.代入已知条件,可以得到:

$$T_{\text{buffer}}+A_i>R_i+T_r=A_{i+n}+T_{\text{wait}}+T_r+T_r=A_i+nT_{\text{interval}}+3T_r,$$

即当 $T_{\text{buffer}}>nT_{\text{interval}}+3T_r$ 时,连续 n 个丢包可以被正常恢复.为了满足最小的恢复要求,即 n=1 的情况,接收端的缓冲时间必须满足 $T_{\text{buffer}}>T_{\text{interval}}+3T_r$ 的要求.

综上所述,我们有以下结论和处理过程:

(1) 当接收端的预先缓冲时间满足 $T_{\text{buffer}}>nT_{\text{interval}}+3T_r$ 时,就可以保证连续丢包数最大为 n(一般<5)情况下的可靠组播传输,同时至少要满足 $T_{\text{buffer}}>T_{\text{interval}}+3T_r$ 才能实现最基本的错误恢复(n=1 的丢失情况).

(2) 发送时间间隔 T_{interval} 越大,接收端检查到数据丢失的时间就越迟,因而所要求的预先缓冲也就越大,所以在传输能力和处理性能允许的范围内,应尽可能地减小传输间隔.

(3) 在指定服务质量 n(用最大连续丢包数表示,也可称为最长连续丢失时间)的前提下,发送端需要为每个数据包保留 $\max(R_i-S_i)$时间,即 $A_{i+n}+T_{\text{wait}}+T_r-S_i=S_i+nT_{\text{interval}}+T_r+T_r+T_r-S_i=nT_{\text{interval}}+3T_r$.所以,发送端为了维护重传

数据,本地的数据缓冲最小应为$(nT_{\text{interval}}+3T_r)$*$Rate$,其中 $Rate$ 是发送数据的平均传输率.发送者收到 NAK 之后,对于在本地缓冲中保留时间小于 $nT_{\text{interval}}+3T_r$ 的重传数据,立即进行调度传输,超过这一限制的重传数据不被重传(或者被放在较为低级的发送队列中,这样可以在保证服务质量的前提下,尽可能地提高传输的可靠性).这个时间限制描述为 MAX_$RETRANSMIT_TIME_i$=$nT_{\text{interval}}+3T_r$.

(4) 接收端为数据包 i 产生NAK之前,判断当前时间 T_{current} 是否有可能产生正常恢复,即判断 $T_{\text{current}}+2T_r\le P_i$.当不满足这一不等式时,重传数据的到达时间必然大于 P_i,不能被正常播放,所以不产生 NAK 反馈.这个限定时间描述为 MAX_NAK_TIME_i=$P-2T_r$,即 $T_{\text{current}}\le$MAX_NAK_TIME_i 时允许产生 NAK.

2.3 NAK过程

接收者从原始数据组播组收到数据包 j 以后,与当前期待的包序号 i 进行比较:如果 $i=j$,则将该数据包链接到本地缓冲中,更新期待的包序号为 i+1,继续进行接收;如果 $i>j$,则因为重传数据和原始发送数据采用两个组播组,这个数据包不可能是重传包,只可能是乱序或循环的包,这种情况在局域网中出现的概率极小,简单丢弃即可;如果 $i<j$,则发现了不连续的包序号,接收者进入 NAK 过程.

接收者根据不连续的包序号间隔将$[i,j-1]$组成一个丢失列表,然后随机等待 $T_{\text{wait}}(0<T_{\text{wait}}<2T_r)$时间.在这段时间内,接收者有以下几种处理方式:当时钟超时事件发生时,接收者通过另一个组播组发送该丢失列表(NAK)给发送者;当收到序号为 $m(i\le m\le j-1)$的 NAK 报文时,接收者从当前丢失列表中删去 m;在当前时间超过了第 $m(i\le m)$包的 NAK 实时性限制,即 $T_{\text{current}}>$MAX_NAK_TIME_m 时,从丢失列表中删除所有$\le m$ 的序号;如果所有$[i,j-1]$内序号都被删除(特别是 $i=j$ 的情况下),取消此次 NAK 过程.

结论:通过接收者在发送 NAK 前的随机等待和发送抑制,保证了每个丢失的数据包最多只产生一个 NAK 信号,实现了 NAK 抑制;对连续丢失的数据包产生一个 NAK,使得总的 NAK 数目大大减少,节省了网络带宽和通信双方的处理器资源;对 NAK 的实时性检查防止了在时间限制下无效 NAK 的产生.

2.4 重传过程

重传和 NAK 都使用同一个组播组,发送者在收到序号从 i 到 $j-1$ 的 NAK 以后,检查本地缓冲是否存在要求的重传数据(本地缓冲受第 2.2 节中描述的实时性限制),如果存在,则进行调度发送.

重传数据和原始发送数据采用两个优先级队列,在进行网络输出时,原始数据优先发送到第 1 个组播组中,在原始数据发送空闲时,重传数据才被发送到第 2 个组播组中.原始数据(实时产生的数据和预先存储在磁盘上的数据)根据数据包中的时间戳信息或预先设定的输出速率进行发送,我们使用基于双漏桶的速率控制算法[18]来描述它的输出,假设为(r,B,P),则加入重传数据以后,控制参数调整为$(110\%r,(n+1)B,125\%P)$,即使重传数据平均速率不超过原始数据流的 10%,重传数据的突发度与最大可恢复的连续丢包数 n 相关,是原始参数的 n 倍,二者瞬时速率之和不超过原始数据最大瞬时速率的 125%.

结论:对重传数据采用实时性检查,可以防止无效重传数据产生;基于优先级的发送策略使得用于错误恢复的重传数据不会影响原有的组播会话;基于双漏桶的速率控制策略限制了重传数据的带宽和规模,防止网络拥塞状况的出现和加重.

3 性能分析

第 2 节分析了为了保证可靠传输和实时传输条件,接收者的预先缓冲时间、原始数据的发送间隔、最大可连续恢复的丢包数以及往返时间之间的联系和制约关系,即 $T_{\text{buffer}}>nT_{\text{interval}}+3Tr$.为了验证这一关系,我们设计了以下实验来检验协议的性能,并给出了在指定服务质量要求下协议参数的设计过程.此实验建立在高速交换以太网环境中,交换机采用基于 Galileo 芯片设计的智能交换机,其中采用 IGMP Snooping 功能支持二层组播功能,网络拓扑采用三级平坦型拓扑结构.

为了测试可靠组播协议对发送数据丢失的恢复的情况,我们使用一个随机数发生器产生一个 0~7 的随机序列 $random_i$.在组播传输过程中,每传输 10 个数据包,丢失 $random_i$ 个数据包.实验环境在一个封闭的高速网络环境中进行,采用高性能计算机作为发送端和接收端,因此,除了主动随机丢弃以外,可以排除网络传输丢失和

处理能力等其他情况导致的数据丢失,这样便于分析和比较几次实验的结果,而且可以避免考虑丢失NAK包等复杂情况.组播会话由一个发送者和4个接收者参与,接收者A不加入用于可靠传输的组播组,接收者B,C,D采用不同的预留缓冲时间.网络中的往返时间$2T_r$由一个GRTT(global roundtrip time)过程进行测量,平均值为2ms,即T_r=1ms.发送数据包大小为12KBytes,输出流是符合MPEG-1格式的视频节目流,平均速率为1440Kbps,即发送间隔$T_{interval}$=(12K*8/1440K)*1000=65ms.

第1个实验是测量接收者的预留缓冲时间对时间受限连续流的可靠传输的影响.发送者服务质量设计为6,即保证至少6个连续丢失数据的错误恢复,则本地缓冲保留$6T_{interval}+3T_r$=393ms的已发送数据,数据包在缓冲区中的时间超过393ms以后,被发送者从本地缓冲中删除.接收者B的预先缓冲时间设为$T_{interval}+3T_r$=68ms,即在播放前,B预先缓冲68ms的数据,再开始播放,所以数据包的到达时刻和播放时钟相差至多68ms.同理,可以设C为$3T_{interval}+3T_r$=198ms,D为$5T_{interval}+3T_r$=393ms.实验持续5分钟,我们选取第260数据包~第490数据包之间的传输情况进行分析和比较.

图2给出了4个接收者在最后进行播放时所遭遇的数据丢失以及发送者重传的数据包.图中数据采用直方图显示,纵坐标表明连续丢失的数据包个数或者连续重传的数据包个数,横坐标表明丢失的包序号或者重传的包序号,例如,图2中的一个直方图横坐标为300,纵坐标为4,表明连续丢失了300,301,302和303这4个数据包.

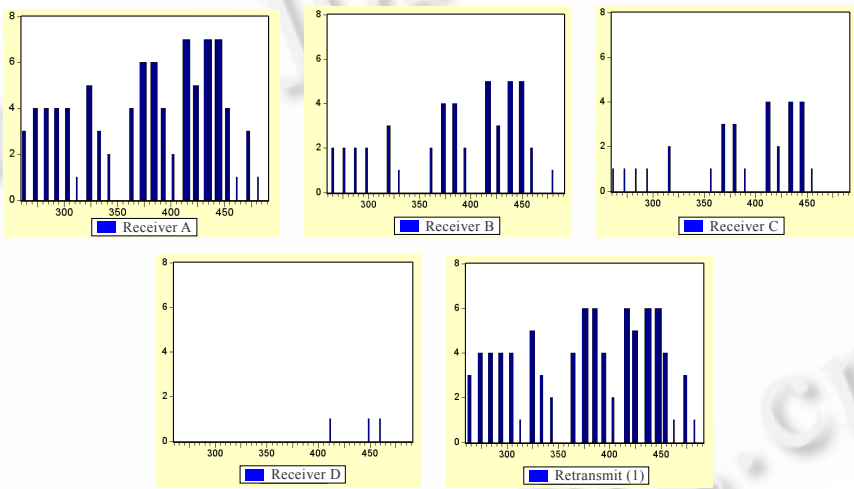

Fig.2 Experiment 1: Data loss of receiver A, B, C and D, retransmit of sender

图2 实验1:接收者A,B,C,D的数据丢失,发送者的数据重传

从图2中可以看出接收者A所经历的数据丢失.因为数据丢失是在发送端主动丢弃的,所以接收者B,C和D接收的原始数据与A相同,都经历了相同时间、相同序号的数据丢失,所不同的是,B,C和D部署了可靠组播,可以通过发送者的重传进行错误恢复.从图2中发送者重传报文的图表可以看出,对于最多连续丢失6个包的情况,发送者都可以进行正常的重传,而在序号400~450之间出现的3次连续7个数据丢失,发送者只重传了7个丢失包中的6个.根据我们前面的分析可知,由于这次实验中发送者的缓冲只维护最多6个数据包,因此发送者的本地缓冲中找不到请求的重传数据.比较图2中B,C,D的数据丢失图表可以看出,随着接收端预先缓冲的增大,错误恢复的比例上升,接收者D完全正确接收了发送者的重传数据,所经历的丢失只是由于发送者的缓冲限制所致,而接收者C和B中的大部分丢失都是因为本地播放时钟的实时性限制.例如,接收者B的预先缓冲小于两个数据包的传输间隔,同时大于1个间隔,所以只接纳了连续重传数据中最新的一个数据包,其他数据包由于时间限制被B丢弃了,同理,C的缓冲时间大于3个间隔又小于4个间隔,因此,对于小于或等于3的连续丢失都可以正常恢复,大于3的连续丢失中最多只恢复了最新的3个重传数据.由此可以验证以下结论:接收端预先缓冲数据的时间T_{buffer}与可连续恢复的数据丢失n成正比,保证n个连续丢失被恢复的必要条件是$T_{buffer}>nT_{interval}+3T_r$成立.

将发送者本地缓冲时间修改为 $3T_{interval}+3T_r$=198ms 或者 $T_{interval}+3T_r$=68ms 以后,重复这次实验,接收者 A,B,C 和 D 的设置不变,图 3 和图 4 是两次实验的最终结果.为了便于比较,两次实验采用同一个节目文件和同一个已生成的随机序列,所以发送者输出的原始数据流保持不变,接收者 A 的数据丢失情况与实验 1 完全相同,可以参见图 2 中的数据.

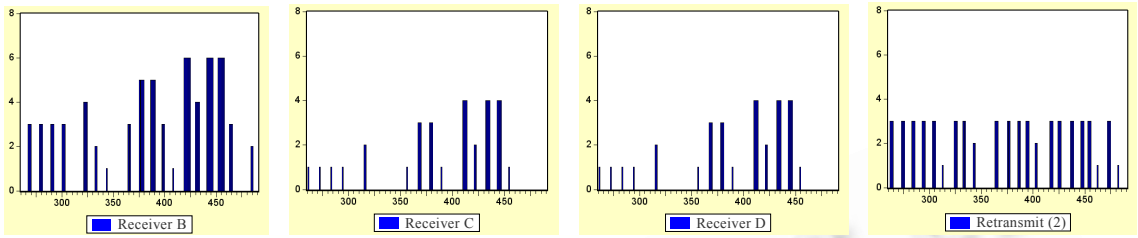

Fig.3 Experiment 2: Data loss of receiver B, C and D, retransmit of sender

图 3 实验 2:接收者 B,C,D 的数据丢失,发送者的数据重传

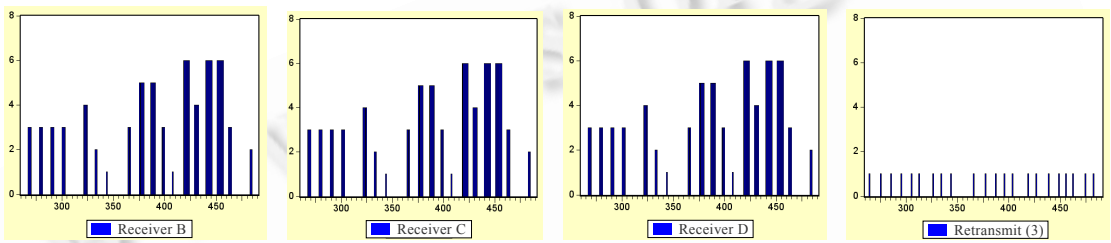

Fig.4 Experiment 3: Data loss of receiver B, C and D, retransmit of sender

图 4 实验 3:接收者 B,C,D 的数据丢失,发送者的数据重传

比较实验 2、实验 3 与实验 1 的区别,首先是发送者的最大连续重传数据包依次减少:实验 1 中最多可到 6 个,实验 2 中最多可到 3 个,实验 3 中只能到 1 个.因为接收者经历的丢失相同,所以最大连续重传减少的原因在于发送者缓冲的减小,而且数值与缓冲时间 $nT_{interval}+3T_r$ 中的系数 n 相等.比较图 4 中 3 个接收者的连续丢失,情况完全相同,图 3 中接收者 C 和 D 接收情况相同,而且都等于当前原始数据丢失个数减去当前重传个数,这说明接收者的最大连续丢失恢复只能在发送者最大连续重传的限制之内.

通过这 3 次实验,我们可以验证在上一节分析的实时性限制结论:当接收端的预先缓冲时间和发送者重传缓冲都满足 $T_{buffer}>nT_{interval}+3T_r$ 时,就可以实现连续丢包数小于或等于 n 情况下的可靠组播传输.在不满足这些约束的情况下,发送端缓冲的减小会减少可靠组播会话中所有接收者的最大可恢复数据,接收端缓冲的减小只会影响它自身的错误恢复.另外,从实验中可以看出,发送者只需要维护很小的缓冲($T_{interval}+3T_r$),接收者只要经历很小的延迟($T_{interval}+3T_r$),就可以恢复单个数据包丢失的情况(多数情况下的数据丢失),系统花费极小的代价就可以保证很高的可靠传输性,更进一步说明了本文提出的协议是一个轻负载、高效率的可靠组播方案.

4 结 论

本文分析了连续数据流的播放模型对可靠传输的实时性限制,讨论了可靠组播过程中数据丢失的性质和处理办法,然后描述了以太网特性对可靠组播的影响,最终给出了针对以太网中连续实时数据流进行可靠组播传输的实现方法,并对其中的 NAK 抑制和实时性限制作了详细的讨论和分析.第 3 节通过实验验证了我们的分析结果.在未来的工作中,我们将首先解决协议中未完成的工作,例如对可变 $T_{interval}$ 的支持,然后将致力于提供一个综合的可靠组播框架,在这个框架中,通过设定不同的应用参数,可以解决所有的可靠组播传输问题,包括在局域网、广域网和 Internet 各种网络应用环境中,Bulk-Data 传输、连续实时流媒体等各种数据传输方式.

References:

[1] Perkins C, Hodson O. Options for repair of streaming media. IETF RFC 2354, University College London, 1998.

[2] Papadopoulis C. Error control for continuous media and large scale multicast applications [Ph.D. Thesis]. Washington University, 1999.

[3] Lane R, Daniels S, Yuan X. An empirical study of reliable multicast protocols over Ethernet-connected networks. In: Proc. of the ICPP 2001. Valencia: Florida State University, 2001. 553~560.

[4] Levine BN, Garcia-Luna-Aceves JJ. A comparison of reliable multicast protocols. Multimedia Systems, 1998,6(5):334~348.

[5] Talpade R, Ammar MH. Single connection emulation (SCE): An architecture for providing a reliable multicast transport service. In: Proc. of the IEEE International Conference on Distributed Computing Systems. Vancouver, 1995.

[6] Floyd S, Jacobson V, Liu C-G, McCanne S, Zhang L. A reliable multicast framework for light-weight sessions and application level framing. IEEE/ACM Trans. on Networking, 1997,5(6):784~803.

[7] Holbrook HW, Singhal SK, Cheriton DR. Log-Based receiver-reliable multicast for distributed interactive simulation. In: Proc. of the SIGCOMM'95. 1995.

[8] Ramakrishnan S, Jain BN. A negative acknowledgement with periodic polling protocol for multicast over LAN. In: IEEE INFOCOM'97. 1997.

[9] Koifman A, Zabele s. RAMP: A reliable adaptive multicast protocol. In: Proc. of the IEEE INFOCOM. 1996. 1442~1451.

[10] Whetten B, Kaplan S, Montgomery T. A high performance totally ordered multicast protocol. In: Proc. of the INFOCOM. 1995.

[11] Yavatkar R, Griffioen J, Sudan M. A reliable dissemination protocol for interactive collaborative applications. In: Proc. of the ACM Multimedia'95. 1995.

[12] Paul S, Sabnani KK, Lin JC, Bhattacharyya S. Reliable multicast transport protocol RMTP. IEEE Journal on Selected Areas in Communications, 1997,15(3):407~421.

[13] Chiu DM, Kadansky M, Provino J. A congestion control algorithm for tree-based reliable multicast protocols. In: Proc. of the IEEE INFOCOM'02. 2002.

[14] Rubenstein D, Kurose J, Towsley D. Real-Time reliable multicast using proactive forward error correction. In: Proc. of the 8th Int'l Workshop NOSSDAV. 1998.

[15] Rubenstein D, Kurose J, Towsley D. A study of proactive hybrid fec/arq and scalable feedback techniques for reliable real-time multicast. Computer Communications, 2001,24(5-6):563~574.

[16] Handley M. An examination of Mbone performance. Research Report, ISI/RR-97-450, University of Southern California Information Sciences Institute, 1997.

[17] Yajnik M, Kurose J, Towsley D. Packet loss correlation in the Mbone multicast network. In: Proc. of the IEEE Global Internet Conference. 1996.

[18] Elwalid A, Mitra D. Traffic shaping at a network node: Theory, optimum design, admission control. In: Proc. of the IEEE INFOCOM. 1997. 445~455.