适用于无线网络流媒体传输的可靠多播协议设计

韩 莉[1,2] 钱焕延[1]

(南京理工大学计算机科学与技术学院 南京 210094)[1] (安徽大学计算机科学与技术学院 合肥 230039)[2]

摘 要 无线网络具有高误码率和高丢失率的特点,现有的组播协议无法为具有时延敏感特点的流媒体应用提供理想的传输质量。将FEC编码与基于NACK的反馈机制相结合,提出了一种以端到端方式运作的轻量级可靠传输协议。协议应用NACK抑制和NACK积累机制防止NACK风暴;引入了TCP友好的拥塞控制机制,与其他传输协议公平共享带宽;使用反馈轮(feedback round)机制,以"轮"为单位实现NACK控制,将接收端等待修复的时间平均到每一轮中,防止了抖动。实验表明,在最大发送速率为2Mbps的设定下,本协议在无线网络上的吞吐量是同类协议NORM的1.5倍,且消除了抖动,具有低延迟的特点,更适合于流媒体传输应用的需要。

关键词 NACK,FEC,反馈轮,无线网络,流媒体

NACK-oriented Reliable Multicast Transport Protocol for Live Streaming over Wireless Channel

HAN Li[1,2] QIAN Huan-yan[1]

(School of Computer Science & Technology, Nanjing University of Science & Technology, Nanjing 210094, China)[1]

(School of Computer Science and Technology, Anhui University, Hefei 230039, China)[2]

Abstract For wireless Channels are inherently lossy, it is challengeable to maintain the Qoality of Service(QoS) for live streaming. In this paper, we proposed a well-designed light-weighted Negative-ACKnowledgment(NACK) Oriented Reliable Multicast Protocol, which combined the FEC-based repair in its design. This protocol can provide end-to-end reliable transport of streams over generic IP multicast routing and forwarding services. For less complicated and more efficient implementation, feedback round mechanism is applied in NACK suppression. Experiment results show that our protocol gets much higher performance than NORM.

Keywords NACK, FEC, Feedback round, Wireless channel, Live streaming

随着无线网络的迅速发展,对基于语音和视频等流式媒体服务的需求量不断增大。视频以及音频的通信正逐步成为无线通信的主要业务之一。目前的流式媒体传输协议多是基于UDP协议的,而UDP协议对包的超时或丢弃不敏感。为了降低延迟,一般使用基于FEC的差错恢复算法,通过在传输中引入冗余信息来减少反馈,提高可靠性。在有线网络中,网络具有低误码率、高吞吐量,并在接收终端具有强计算能力和纠错能力,因而可以得到较好的效果。

相对于有线网络,无线网络具有高误码率和高丢失率的特点,且无线网络中存在大量计算能力受限的弱终端,它们的纠错能力和缓存能力都较弱,所以对于包的缺失非常敏感。因此,没有反馈的多播协议无法满足无线网络应用环境的需求。

本文将FEC编码与基于NACK的反馈机制相结合,提出了一种以端到端方式运作的轻量级可靠多播协议。协议应用NACK抑制和NACK积累机制防止NACK风暴;引入了TCP友好的拥塞控制机制,与其他传输协议公平共享带宽;使用反馈轮(feedback round)机制,以"轮"为单位实现NACK控制和对包丢失率、往返时间和TCP友好速率进行估算,将接收端等待修复的时间平均到每一轮中,防止了抖动。

本协议无需中间系统提供除基本的IGMP、路由转发服务外的任何辅助功能,充分利用了现有的网络资源,有很强的实用价值。

实验结果表明,在最大发送速率为2Mbps的设定下,本协议在无线网络上的吞吐量是同类协议NORM[1]的1.5倍,且消除了抖动,具有低延迟的特点,更适合于流媒体传输的需要。

1 差错控制机制

可靠多播旨在保证每个接收者都能正确地接收到所有的多播数据。流媒体应用具有时延敏感的特点,发送端到接收端超过几百毫秒的包基本就没有用处了。尤其在接收端数目较大的情况下,既要保证流媒体传输对时延的需求,又要完成较完整的接收,因此必须要有快速高效的差错检测和修复机制,且其处理方式应满足高效性和可扩展性要求。

1.1 基于否定确认的反馈机制

根据丢失检测是由发送者还是接收者进行,分为肯定(ACK, ACKnowledgement)确认和否定(NACK, Negative

到稿日期:2009-10-30 返修日期:2010-01-08 本文受安徽省教育厅重点科研项目(EJ2009A001Z)资助。

韩 莉(1975—),女,博士生,讲师,主要研究方向为计算机网络技术及应用,E-mail:hanli98@ahu.edu.cn;钱焕延(1950—),男,教授,博士生导师,主要研究方向为计算机网络技术及应用。

ACKnowledgement)确认两种。基于肯定确认 ACK 的模型是由发送者负责检测数据包是否被可靠传输。发送者保存每个接收者的数据包的接收状态信息并对每个数据包设置相应的定时器,接收者对每个收到的包进行确认,给发送者返回 ACK。这种方式应用到多播组中,就可能给发送者带来大量的 ACK 消息,即"ACK 风暴"。

基于否定确认 NACK 的模型是一种由接收者负责包丢失检测的模型,即只反馈未接收到的包信息,这样在丢失率不高的情况下就相对节约了网络带宽,减轻了发送者的负担以及"NACK 风暴"的程度。然而在丢失率频繁的情况下,仍存在大量反馈的问题,因此仍需要有相关机制来防止 NACK 风暴,特别是当丢失率较高、多播组很大时,同样可能带来 NACK 风暴。

1.2 基于 FEC 的差错恢复算法

无线网络中,由于网络丢失率较高,使用 FEC 前向纠错技术有利于减少反馈,提高多播的性能。

FEC 的主要思想是:k 个数据包在发送端经过冗余编码后生成 n 个数据包,将 n 个包发给接收端;接收端收到至少任意 k 个数据包就可以通过译码还原原始数据。

本文使用 Erasure Codes[2] 作为 FEC 编码算法。Erasure Codes 是基于范德蒙矩阵在有限域上运算的编码。它起源于处理链路层少量的不可纠正的错误或网络拥塞造成有效包的丢失;可以有效地在微处理器上实现,简单灵活。Erasure Codes 是一种线性码,可以通过线性代数理论进行分析。设 $x=x_0\cdots x_{k-1}$ 为原始数据,G 是 $n\times k$ 维矩阵。我们的协议使用的是系统码,在矩阵 G 中包括一个矩阵 I_k,其优点是简化了原始数据的重构过程。线性码 (n,k) 的变换可以表示为 $y=Gx=(I_k/P)x$,如图 1(a)所示。接收端收到 y 的 n 个分量中 k 个分量 y' 后,从 G 中提取出对应于分量 y' 的行,组成 $k\times k$ 矩阵 G'。G' 中各行线性无关;y' 和 G' 相当于图 1(b)中矩阵向量的灰色区域。原始数据包就可以通过对应于 k 个分量的 k 个等式还原出来,即 $y'=G'x$;其逆变换为 $x=G'^{-1}y'$。求解方程得到原始数据 x。

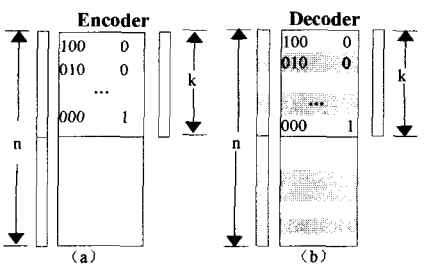

图 1 矩阵形式的编码/译码过程

2 协议描述

本协议基于 NACK 和 FEC 的差错恢复机制实现了流媒体在无线网络上的可靠组播。针对流媒体传输的要求和无线网络的特点,改进了现有的算法,简化了实现方法。实验证明,本协议在无线网络流媒体传输的应用性能要远远高于同类协议。

2.1 数据的表示方式

我们把流媒体数据分割成包,以包作为传输的基本单位。每个包的大小约为 1024 字节,在一个会话中,每个包有唯一的序列号。

FEC 编码算法以 FEC 组为单位,每个 FEC 组含有 n 个包,其中原始数据包的个数为 k,FEC 编码包的个数为 $n-k$ 个。当 $n=256$ 时,FEC 编码算法可以获得最高的编码效率[2]。我们定义每个 FEC 组中最多包含 128 个原始数据包和 128 个 FEC 包。实际上,k 要小于 128,这是由于 FEC 组中包的个数越多,修复时延越长,不适合流媒体实时性的要求。

发送端和接收端各自维护一个滑动窗口,滑动窗口是以 FEC 组为单位的。在一个会话中,每个 FEC 组有唯一的组号。

2.2 发送端描述

发送端的任务是从应用层接收数据,并按协议确定的数据发送速率 R 进行发送。当发送原始数据包的个数达到 k 时,发送端将根据当前的包丢失率 p,生成并发送 $k\times p$ 个 FEC 包。

为实现 NACK 抑制,发送端周期性地发送查询包,接收端仅在接收到查询包时才以 NACK 包响应。NACK 包有两个作用:通告包丢失的情况、辅助发送端估算 GRTT 值及冗余率。

发送端对接收端发回的 NACK 包进行累计,并在下一个发送周期生成并发送一定数量的修复包。

2.2.1 NACK 抑制和 NACK 积累

协议使用反馈轮(feedback round)机制。其简单原理为:在发送端将时间分为连续的时间段,每段称为一个轮,用唯一整数(称为轮号)标识,相邻轮的轮号是连续的;两个轮次之间的时间与当前的平均往返时间 GRTT 相关,GRTT 的选择应该有助于发送端在一轮的时间内接收到大部分接收端对本轮次的反馈。

发送端在一轮的开始发送一个查询包 REQUEST,此包通告了本轮次的 roundID、本轮次的查询时间戳 sendTime,及当前发送窗口中 FEC 组的最大组号。

由于流媒体传输的实时性要求,接收端接收到查询包时会立刻以 NACK 包应答;接收端统计当前各个 FEC 组丢失的包数,将其写入 NACK 应答,同时还回填了发起此次反馈的反馈轮次 roundID 及查询时间戳 sendTime。由于接收端不会在发现数据丢失时立刻发送 NACK 包,而是等待发送端的查询,该机制有效地实现了 NACK 抑制。

发送端维护两个数组 maxRequestList 和 pendingRequestList。

maxRequestList 中存放了一轮中发送端对从接收端接收到的 NACK 的累积结果 $\{x_i,x_{i+1},\cdots,x_{i+Gc-1}\}$;其中 x_{i+j} 表示所有接收端在组号为 $(i+j)$ 的 FEC 组丢包的最大个数,Gc 是发送端发送窗口中 FEC 组的个数。发送端一轮内发送修复包的总数为 $(1+p)\times\sum_{j=0}^{Gc-1}x_{i+j}$,其中 p 是包丢失率,其计算见 2.1.4 节;每一轮结束,maxRequestList 清零。

为减少延迟,发送端不会在一轮结束的时候才发送修复包,而是每经过一个发送周期就检查一次是否有补偿包需要发送,如果有,则发送补偿包。pendingRequestList 中存放的是在一个发送周期中累积的需要发送的补偿包的个数:$\{y_i,y_{i+1},\cdots,y_{i+Gc-1}\}$;其中 y_{i+j} 表示本发送周期在组号为 $(i+j)$ 的 FEC 组需要发送的补偿包的个数。发送端一个发送周期内发送修复包的总数为 $(1+p)\times\sum_{j=0}^{Gc-1}y_{i+j}$,其中 p 是包丢失

率，每完成一次发送，pendingRequestList 清零。

假定一个 NACK 包中的请求修复列表为$\{z_i, z_{i+1}, \cdots, z_{i+Gc-1}\}$，其中 z_{i+j}为$(i+j)$的 FEC 组需要发送的补偿包的个数。计算 maxRequestList 和 pendingRequestList 的算法描述为：

If($x_{i+j}=0$)$x_{i+j}=z_{i+j}$

else

if($z_{i+j}>x_{i+j}$)

$x_{i+j}=z_{i+j}$;

new$=z_{i+j}-x_{i+j}$;

if($y_{i+j}=0$)$y_{i+j}=$new;

else $y_{i+j}=y_{i+j}+$new

2.2.2 GRTT 测量

发送端需要计算接收端到发送端的平均往返时间 GRTT，其主要用于 NACK 抑制及拥塞控制的速率计算。

协议定义了两个常量 RTT_MIN 和 RTT_MAX。RTT_MIN 是协议设定的最小往返时间。由于无线网络中经常会出现突发性的包丢失现象，出现 1 秒甚至几秒的往返时间，因此需要设定最大往返时间 RTT_MAX 过滤网络中的坏数据。GRTT 的初值取为 RTT_MIN，具体算法描述如下。

1)在每一轮的周期内，接收端每接收到一个 NACK 包，就根据 NACK 包中的字段 sendTime 计算接收端到发送端的往返时间：

RTT=min(currentTime−sendTime,RTT_MAX);

sumRTT=sumRTT+RTT;

countRTT=countRTT+1;

2)当前轮结束时，计算 GRTT：

如果 countRTT 为 0，GRTT 不变；

否则

RTT_{inst}=sumRTT/countRTT;

GRTT$=(1-\gamma)\cdot$GRTT $+\gamma\cdot RTT_{inst}$;

其中，γ 为权值因子，取值为 0.1。

3)下一轮开始时，初始化 countRTT=0，sumRTT=0；设定该轮的持续时间为 $a\times$GRTT。a 的选择决定了接收端反馈的密度及流媒体的播放时延。本协议中 a 取值为 2。

2.2.3 TCP 友好特性

式(1)计算的是 TCP 发送速率的上限。如果发送端以低于或等于这个速率发送数据，则被认为具有 TCP 友好性[3]。

$$R'=S/(GRTT\times\sqrt{(2/3)\times p}+12\times\sqrt{(3/8)\times p}\times p\times(1+32\times p^2)) \quad (1)$$

式中，R'为计算出的发送速率上限；$GRTT$ 为数据包的往返时间；p 为接收端计算出的丢包事件率；S 为数据包的大小。

由于流媒体传输的实时性和传输质量的要求，需要确定一个最小的传输速率 R_MIN，它的选择与应用中媒体的类型有关。发送端实际的发送速率 R=MAX(R_MIN,R')。

2.2.4 包丢失率的估算

为了减少反馈，发送端在发送数据时应根据当时的包丢失率发送一定的冗余包，该冗余包的个数由包丢失率 p 决定。类似于 GRTT 的估算，包丢失率 p 的估算也以轮次为单位。

协议管理一个与发送窗口同步的丢失率统计窗口 cRepairList。cRepairList 存放到上一轮次结束时发送端发送的修复包的个数$\{r_i, r_{i+1}, \cdots, r_{i+Gc-1}\}$；其中 r_{i+j}表示发送端为组号为$(i+j)$的 FEC 组发送的修复包的总数，Gc 是发送端发送窗口中 FEC 组的个数。

在本轮次结束时，根据 maxRequestList 中的相应变量$\{x_i, x_{i+1}, \cdots, x_{i+Gc-1}\}$，更新 cRepairList 为$\{r_i+x_i\cdot(1+p_m), \cdots, r_{i+Gc-1}+x_{i+Gc-1}\cdot(1+p_m)\}$，其中 p_m 是第 m 轮的包丢失率。记 q_{i+j}是组号为$(i+j)$的 FEC 组在第 m 轮的包丢失率：

$$q_{i+j}=1-(k+x_{i+j}\cdot p_m)/(k+r_{i+j})$$

因此，下一轮次的丢包率 p_{m+1}取值为：

$$p_{m+1}=a\cdot\frac{1}{G_c}\sum_{j=0}^{Gc-1}q_{i+j}$$

2.3 接收方描述

接收端负责从网络接收数据，并将数据输出到应用层。同时，接收端还以 NACK 响应发送端的查询包，通告本节点的接收情况，辅助 GRTT 和包丢失率的统计。

接收端接收缓存的构成如图 2 所示，阴影部分为接收窗口，其余部分为接收缓存尚未使用的部分。接收窗口中 FEC 组的个数为 cPendingGroups，cPendingGroups 越大，表明接收延迟越大。

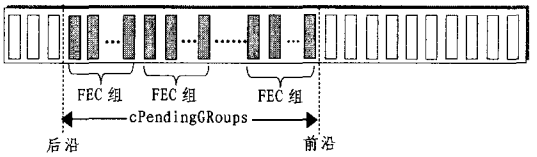

图 2 接收窗口的构成

接收端接收数据的算法描述如下。

1)接收端根据数据包包头中的组号，查询接收窗口。

2)如果包所在的 FEC 组不在接收窗口中，且组号大于接收端的当前组号，则接收端调整接收窗口的前沿到该组——这种机制可以有效地实现发送端和接收端的同步。

3)将接收到的数据写入相应的 FEC 组：

a)当 FEC 组中包的个数大于 k 时，接收端执行 FEC 解码算法，解码出该 FEC 组中的所有原始数据包。

b)如果该 FEC 组是接收窗口中最小的 FEC 组，则检查接收窗口，输出自最小接收窗口开始已完全解码的 FEC 组，并把接收窗口的后沿向前移动。

3 实验结果

实验条件：Bufflalo WHR-HPG54 无线 AP，及若干台 DELL d630 笔记本。终端操作系统为 windows XP。发送端最高发送速率和接收端最高接收速率均设定为 2Mbps。

图 3 显示了随着接收端数目的增加，本协议与 NORM[1] 协议在平均吞吐量上的对比结果。

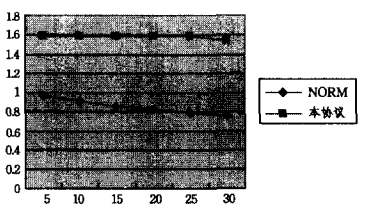

图 3 吞吐量比较

(下转第 172 页)

实验结果如图 2、图 3 所示。图中横轴为单表的数据量，以 100M 为单位；纵轴为查询响应时间，以毫秒为单位。

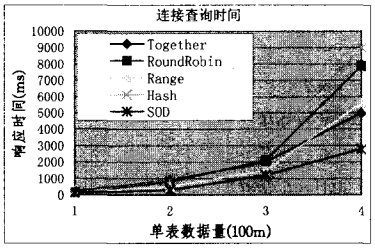

图 2　连接查询时间

从图 2 中可以看到，SOD 分布策略对于连接查询的反应明显优于其他分布策略，同时优于集中存储。原因在于需要进行连接的数据都分布在同一节点上，无需在系统内的节点之间传输数据，从而节省了大量的时间。

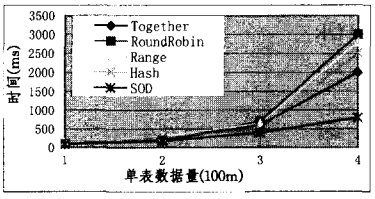

图 3　综合查询时间

从图 3 中可以看出，在进行综合查询时，各种分布策略在不同数据量下的反应时间差别较为明显，且在数据量较大时 SOD 策略明显优于其它分布策略。原因在于节省了大量的连接查询时间，从而提高了系统的查询性能。

结束语　根据云计算中数据分布存储的特点，本文提出了一种面向查询的数据库数据分布策略。根据数据库的实际查询模式进行数据的分布，将通常一起被查询的表分布到同一节点上，同时通过对各表的查询频率统计将经常被查询和很少被查询的表配对，以均衡各节点的负载，最终达到降低系统响应时间、提高系统性能的目的。通过实验证明，本文策略可以明显减少数据库查询响应时间，特别是连接查询的响应时间。

未来工作主要是实验确定对查询时间的统计，以研究对特定数据库的查询操作的响应时间情况，此种统计应该比仅对查询次数统计得到的结果更好；同时对于包含超大数据类型的表进行属性的查询分析，研究相关的垂直分布策略等。

参 考 文 献

[1] Codd E F. A relational model for large shared data banks[J]. Comm. ACM, 1970, 13(6): 377-387

[2] Ghemawat S, Gobioff H, Leung Shun-Tak. The Google File System[J]. SIGOPS Operating Systems Review, 2003, 37(5)

[3] Chang F, Dean J, Ghemawat S, et al. Bigtable: A Distributed Storage System for Structured Data[C]// 7th Symposium on Operating Systems Design and Implementation(OSDI 2006). Seattle, WA, USA, November 2006: 205-218

[4] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters[J]. Communications of the ACM, 2005, 51(1): 107-113

[5] 陈康，郑纬民. 云计算：系统实例与研究现状[J]. 软件学报，2009(5): 1337-1348

[6] Sylvain G, Le G. Using Cluster Computing to Support Automatic and Dynamic Database Clustering[C]// Third International Workshop on Automatic Performance Tuning(iWAPT). 2008: 394-401

[7] Guinepain S, Gruenwald L. Automatic Database Clustering Using Data Mining[C]// Database and Expert Systems Applications, 2006(DEXA '06). 17th International Conference. 2006: 124-128

[8] Zhong Ke, Dutt S. Effective partition-driven placement with simultaneous level processing and global net views[C]// IEEE/ACM International Conference on Computer Aided Design. Nov. 2000: 254-259

[9] Jean-Daniel C, Alain A, Alain A. Criteria to Compare Cloud Computing with Current Database Technology[C]// Dumke R, et al. , eds. IWSM / MetriKon/Mensura LNCS 5338. 2008: 114-126

[10] Fung Chi-Wai, Kamalakar K, Li Qing. Cost-driven vertical class partitioning for methods in object oriented databases[J]. The VLDB Journal, 2003, 12: 187-210

[11] Lee PeiZong. Efficient Algorithms for Data Distribution on Distributed Memory Parallel Computers[J]. IEEE Transactions on Parallel and Distributed Systems, 1997, 8(8): 825-839

[12] Li Wei, Chen C X. Efficient Data Modeling and Querying System for Multi-dimensional Spatial Data[C]// ACM GIS '08. Irvine, CA, USA, November 2008

(上接第 126 页)

实验结果表明，本协议的带宽利用率大于 75%，且随着接收端数目的增加，接收速率较稳定。NORM 协议的带宽利用率小于 50%，并随着接收端数目的增加呈下降趋势。

根据对实验结果的观察，本协议接收端接收数据的速率较稳定，基本消除了抖动。而使用 NORM 协议接收数据，接收端间或会出现 1～2 秒的停滞，这对于流媒体传输来说是不可接受的。

结束语　由于无线网络具有高误码率和高丢失率的特点，现有的可靠组播协议无法满足流媒体传输的实时性需求。

本文基于现有的可靠组播算法，设计了一种以端到端方式运作的轻量级的可靠组播协议。该协议使用 FEC 编码和 NACK 相结合的差错控制机制来实现可靠传输，并应用 TCP 友好的拥塞控制机制与其他传输协议公平共享带宽。该协议无需中间系统提供除基本的 IGMP、路由转发服务外的任何辅助功能，充分利用现有的网络资源，有很强的实用价值。

实验结果表明，在无线网络流媒体传输的应用环境中，相对于同类协议 NORM，本协议在吞吐量、带宽利用率及消除抖动方面上都有较大优势。

参 考 文 献

[1] Adamson B, Bormann C, Handley M, et al. NACK-Oriented Reliable Multicast Transport Protocol 〈draft-ietf-rmt-pi-norm-revised-13〉 Internet Draft, Intended status: Standards Track, Expires, December 2009

[2] Rizzo L. Effective erasure codes for reliable computer communication protocols[J]. ACM Sigcomm Computer Comm unication Review, 1997, 27(2): 24-36

[3] Sisalem D, Wolisz A. MLDA: a TCP_friendly congestion control framework for heterogeneous multicast environments[C]// Proc. of the 8th International Workshop on Quality of Service. Pittsburgh: [s. n.], 2000: 65-74