

DCCP 下的一种流媒体重传机制

郑超美, 邹群, 李荣鑫

ZHENG Chao-me, ZOU Qun, LI Rong-xin

南昌大学 计算中心, 南昌 330031

Computer Center, Nanchang University, Nanchang 330031, China

E-mail: chaomei_teacher@yeah.net

ZHENG Chao-me, ZOU Qun, LI Rong-xin. Retransmission method for multimedia streaming in DCCP/IP networks. *Computer Engineering and Applications*, 2009, 45(36): 123-125.

Abstract: Retransmission-based error recovery is inappropriate for multimedia applications, because of its latency. This solution can be attractive because it requires minimal network bandwidth, processing cost and efficiently improves the stream quality. A source controlled selective retransmission algorithm is presented with a decision algorithm based on the RTT and sending rate and the network congestion control determined by the TFRC. The emulating results show that significant quality improvement is possible with the proposed selective retransmission scheme.

Key words: retransmission; Datagram Congestion Control Protocol (DCCP); TCP-Friendly Rate Control (TFRC); Quality of Service (QoS)

摘要: 由于差错重传机制存在各种缺点, 并不适合流媒体传输, 若它的处理过程占用最小网络带宽和时间, 将大大提高流媒体的质量。提出了一种基于 DCCP 的源端控制的选择重传机制 (TBSR), 其重传由决策算法决定, 而发送速率和拥塞控制由 TFRC 控制。仿真结果表明, 机制能大大提高流媒体视频质量。

关键词: 重传; 数据拥塞控制协议 (DCCP); TCP 友好速率控制机制 (TFRC); 服务质量

DOI: 10.3778/j.issn.1002-8331.2009.36.036 **文章编号:** 1002-8331(2009)36-0123-03 **文献标识码:** A **中图分类号:** TP393

1 引言

随着网络技术的提高和网络的普及, 流媒体业已成为网络应用中不可或缺的一部分。无论对于有线网络还是无线网络, 数据包的丢失一般都会降低网络传输性能, 尤其是压缩数据, 如 MPEG 流。然而, 及时的重传却可以提高网络性能和流媒体质量。

因此, 重传机制成为流媒体研究的热点之一。目前最广泛流行的传输协议如 TCP、UDP 协议。TCP 是可靠传输协议, 但是它不适合流媒体的传输, 而且流媒体并不需要完全可靠的传输; UDP 则因为其缺乏拥塞控制, 不能与 TCP 流友好共存, 容易造成拥塞。在不可靠传输协议中只有 DCCP^[1]结合了 TCP 和 UDP 的优点, 支持 TCP 友好拥塞控制算法, 非常适合流媒体的传输。

目前关于 DCCP 的重传机制研究尚有不足。文献[2]提出了基于 DCCP 的可选择的 MPEG 流重传机制, 并分析和证明了 DCCP 协议可以提供 MPEG 流重传所需的必要信息, 但该机制必须在高带宽的状况下运行, 而且未考虑拥塞的状况; 文献[3]提出了带有决策算法 (Decision algorithm) 的选择重传机制, 其决策算法用来决定是否对一个丢失的数据包发出重传请求, 但

该算法没有考虑到缓冲区以及拥塞避免算法。

对此, 提出了一种基于 DCCP 的源端控制的重传机制 TBSR (TFRC-Based Selective Retransmission)。该机制分为两部分: 重传部分与发送速率和拥塞控制部分。重传部分对文献[3]的决策算法进行了改进, 提出了一种源端控制决策算法 (Source Controlled Decision Algorithm on RTT, SCDA), 该算法在原算法基础上在端缓冲区使用了 FLOOD 方法来充分考虑缓冲区的状况。发送速率和拥塞控制部分则由 TFRC^[4]决定。

2 源端控制的决策算法

基于 DCCP 的选择重传机制能否将丢失的数据包重传成功取决于当前网络状况和数据包的内容。DCCP 是一个拥有 TCP 友好拥塞控制、可靠地建立和拆除连接、确认以及适宜流媒体传输等特点的新协议。它拥有 TBSR 机制运行所需的所有信息。接收端可以通过 DCCP 数据包头部包含的序列号来了解哪些包丢失或者标记了。DCCP 拥有拥塞控制, 但是和 TCP 不同的是, 它提供可选的拥塞控制算法, 给应用程序一个选择怎样的拥塞控制机制的机会。目前 DCCP 有两种拥塞控制算法: TCP-Like 和 TFRC^[4]。这两种拥塞算法中 TFRC 更适合流媒体

基金项目: 江西省自然科学基金 (the Natural Science Foundation of Jiangxi Province of China under Grant No.2007GZS1905); 江西省教育厅基金 (the Fund of Department of Education of Jiangxi Province, China under Grant No.GJJ08062)。

作者简介: 郑超美 (1959-), 女, 副教授, 研究方向为计算机网络; 邹群 (1982-), 硕士研究生, 研究方向为计算机网络; 李荣鑫, 男, 讲师, 研究方向为计算机网络。

收稿日期: 2008-09-17 **修回日期:** 2008-11-19

传输。

为了能让重传成功,SCDA 算法的思路是在接收端应设置一个播放缓冲区。又因为 DCCP 没有数据包重传功能所以在 DCCP 源端也应设置一个发送缓冲区,让服务器缓冲区发送数据包到这个发送缓冲区内。然后提出了 Flood 方法来计算接收端缓冲区的延迟以确定发送端的发送缓冲区延迟来让重传达到较高的成功率。

2.1 Flood 方法

要想在源端控制重传,并让其处理过程尽量简化,就必须知道发送缓冲区的大小和它的时延。该文提出了一个方法——Flood 方法。

一个 Flood 过程是:首先,视频流发送器并不是立即将数据包传送到网络上,而是先将一部分数据包送到发送缓冲区,然后经过 T_{bd} 秒后再将这些数据包一起发送出去。这样接收端将接收到比它能处理的数据更多的数据包,这些数据包会放到接收端的播放缓冲区中。由于接收端处理速率得到的吞吐量应该等于视频流当前的比特率,发送端发送器就可以计算出接收端的处理速度。这样 Flood 的持续时间就可以被计算出来。

假设视频源的比特率是 $f(t)$,再经过一个 Flood 过程后发送器增加的发送速率为 $f_1(t)$ 。在理论上 $f_1(t)$ 在整个 $[T_{bd}, T_{bd}+\tau]$ 时间内不变。那么缓冲区内的缓冲的数据量就为:

$$f_1(t)\tau - f(t)\tau = X \quad (1)$$

其中, X 要在 τ 时间内发送出去, τ 可以由下式得出:

$$\tau = \frac{X}{f_1(t) - f(t)} \quad (2)$$

实际上 Flood 方法的持续时间取决于网络实际状况。TBSR 机制是在 DCCP 协议下使用 TFRC 拥塞算法来决定发送速率,则在整个 Flood 持续时间内, $f_1(t)$ 不是不变的。当缓冲区内的数据发送出去后,发送速率近似于视频流的比特率,并且接收端的播放缓冲区包含的 X 数据应为:

$$X = T_{bd}(t) \times f(t) \quad (3)$$

假设视频流比特率是不变的,那么 $T_{bd}(t)$ 就不会变化太大。在这个前提下,根据 Flood 方法,就可以知道缓冲区的发送延迟和大小。接下来的问题就是——需要多少时间才能让 X 数据包有足够的延迟重传成功?这就是 SCDA 算法要解决另外一个问题。

2.2 SCDA 重传成功所需的 RTT

第一次传送、重传和检测到丢包的有效时间和应该等于发送缓冲区延迟,因此约定成功地重传一个数据包的时间为:

$$\frac{3}{2}RTT + \delta < T_{bd} \quad (4)$$

其中, δ 为检测到丢包的时间。以后检测到丢包的时间就可以用 TFRC 算出,因为 DCCP 协议下的 TFRC 能充分利用到 DCCP 的确认报文——DCCP-ACK。在 DCCP 协议下接收端发送一个 DCCP-ACK 包在至少一个往返时间内确认数据包。如果协议给每个数据包都发送 ACKs,那么每次丢包都可以检测到。那么检测到丢包的时间为:

$$\frac{1}{f_p} < \delta < RTT \quad (5)$$

f_p 为数据包的发送频率,那么两个数据包之间的时间间隔为 $\frac{1}{f_p}$ 。

实际上,每个包的发送时间 t_1 和确认报文接收时间 t_0 都不同。接收到确认报文中包含的序列号,发送器就可以知道哪些

数据包需要重传。决策算法考虑出足够重传成功的时间应小于单向网络延迟:

$$(t_0 - t_1) - T_{bd} < RTT \quad (6)$$

播放缓冲区时延 T_{bd} 的限制由流媒体应用程序和网络延迟决定。单工流媒体应用程序可接受的 T_{bd} 最大值很小,但是对于双工应用程序 T_{bd} 最大值可达 150~200 ms。在重传时播放缓冲区的最低时延取决于网络延迟。该文的目标是找到能重传成功的最低时延。

根据公式(4)和(5),发送所需时间和检测到丢包的时间之和为:

$$\frac{3}{2}RTT + \delta \leq \frac{3}{2}RTT + RTT = 2.5RTT \quad (7)$$

目前最常用的计算往返时间的方法有 Normal distribution^[5] 的 $N(\mu, \sigma^2)$ 和文献[6]中的 heavy-tailed Gamma 方法。先使用 Normal distribution 中的模型 $N(\mu, \sigma^2)$ 来找出超时重传最小化时的时间,设 $\mu = RTT, \sigma^2 = (0.1RTT)^2$ 来决定发送缓冲区延迟。采用比这个模型里的 RTT 的平方差相对大一点的 RTT 来计算,因为该文目标是超时重传最小化,让重传成功率在 95%以上,则

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-x}^x \exp(-\frac{u^2}{2}) du = 0.95 \Rightarrow x = 1.65 \quad (8)$$

$$X = \mu + x\sigma = 2.91RTT \approx 3RTT \quad (9)$$

算出缓冲延迟 $T_{bd} = 3RTT_0$, RTT_0 为第一次 RTT 测量出的源端发送到目的端的时间。

根据以下方程算出 RTT 最大取值范围为 $(0.124RTT_0)^2$ 才可能使重传成功率达到 95%:

$$F(x; \mu; \sigma) = F(3; 2.5; 0.124) = 0.95 \quad (10)$$

再使用文献[6]中的 heavy-tailed Gamma 方法中模型计算:

$$\phi(x, \alpha, \beta) = \frac{x^{\alpha-1} \cdot e^{-\frac{x}{\beta}}}{\beta^\alpha \Gamma(\alpha)}, x > 0 \quad (11)$$

变化范围为:

$$Var(x) = \alpha \cdot \beta^2 \quad (12)$$

代入与 Normal distribution 中的模型 $N(\mu, \sigma^2)$ 相同的 RTT 值,变化范围为 $(0.1RTT)$ 来计算,得出 $\alpha = 100, \beta = 0.01$ 。同样重传成功率要达到 95%,再结合式(7)中的结果,则得出:

$$\phi(x; 2.5\alpha; \beta) = \frac{x^{2.5\alpha-1} \cdot e^{-\frac{x}{\beta}}}{\beta^{2.5\alpha} \Gamma(2.5\alpha)} = 0.95 \quad (13)$$

$$x = 2.76RTT \approx 3RTT \quad (14)$$

两种方法都算出 T_{bd} 为 $3RTT_0$ 时,重传成功率可达到 95%。由此可以得出,发送缓冲区取 T_{bd} 为 $3RTT_0$ 至 $(0.18RTT_0)^2$ 都可以让重传成功率达到 95%。这么大的变化范围足够让对延迟敏感的重传成功。

3 基于 TFRC 的选择重传机制

重传能否成功取决于视频流比特率和 TFRC 发送速率。当网络处于拥塞状态或接近拥塞状态时,重传很难成功。当网络路由器缓冲超载时,重传就可能加重网络负载甚至可能引起其他数据的丢失。这时就需要借助 TFRC 的拥塞避免机制来决定丢失的数据包是否要重传。

TFRC 是一种基于模型的拥塞算法,其基本思想是在接收端通过计算丢包率来检测网络的拥塞,并将信号反馈给发送端,而发送端利用以丢包率和回环时间为变量的吞吐率公式来

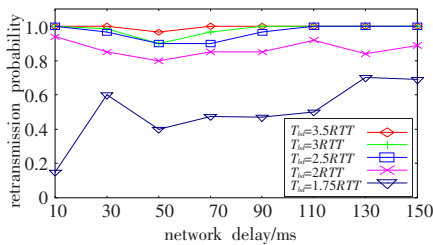


图2 重传概率

调整发送速率。因为需要对 TCP 保持友好性,所以 TFRC 使用的是在 TCP 协议中定义的吞吐率计算公式^[4]:

$$X = \frac{s}{t_{RTT} \sqrt{\frac{2p}{3}} + t_{no} 3 \sqrt{\frac{3p}{8}} p (1+32p^2)} \quad (15)$$

其中, X 为发送速率, s 为数据包大小, t_{no} 为 TCP 重传超时时间, p 为丢失事件率。

当网络没有拥塞时, MPEG 流比特率一般都低于 TFRC 发送速率;但是处于拥塞时, MPEG 流比特率则会高于 TFRC 流。因此在 TBSR 机制中,当 TFRC 发送速率低于视频流比特率时,重传不会成功。DCCP 协议提供了 TBSR 机制所需要的所有信息,不需要再发送数据包或其他手段来了解网络或机制运行状况,这样减少了大量的负担。发送的数据包应先存储在服务器的发送缓冲区然后再发送到网络上。同时使用 TFRC 算法来进行拥塞控制和 SCDA 决策算法协调工作。

4 仿真结果与分析

网络拓扑结构如图 1。整个环境是在 DCCP 协议下运行,带宽为 1 Mb/s, TBSR 机制则在节点 A 处运行。节点 B 维持使用 FIFO 的尾部丢弃算法控制的长度为 10 的一个队列。

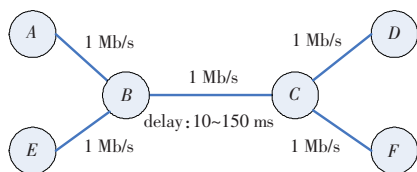


图1 网络拓扑结构

先验证 SCDA 算法中发送缓冲区要达到较高重传率所需的 RTT,结果如图 2。图 2 表明当 $T_{bd}=3RTT$ 时,数据包重传几乎都能成功。

TBSR 机制最大的目的是提高视频流的质量。为了能验证 TBSR,以 MPEG 流为例,测量它的平均最大信噪率(Peak

(上接 111 页)

5 结论

通过对移动 Ad Hoc 网络位置管理策略的研究,描述了 ALOM 这样一种新的位置管理策略。通过与原有 SLALoM 和 SLURP 策略比较,可以看出:此策略大大减少了服务区域的负载,从而也减少了位置更新的开销。特别是节点数目增大时,位置管理策略有了显著的提高。因此,对于大型移动 Ad Hoc 网络具有良好的可扩展性。

参考文献:

[1] Iwata A, Chiang C C, Pei G, et al. Scalable routing strategies for ad hoc wireless networks[J]. IEEE J Select Areas Commun, 1999, 17(8): 1369-1379.

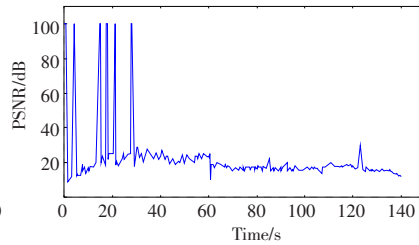


图3 未使用 TBSR 的 MPEG 流

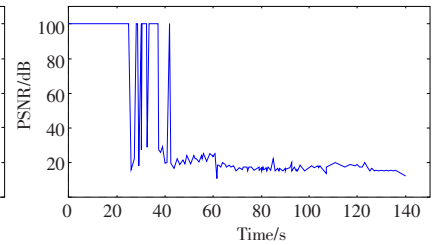


图4 使用了 TBSR 机制后的 MPEG 流

Signal-to-Noise Ratio, PSNR), 并将 T_{bd} 设为 $3RTT$, 结果如图 3 和图 4。

从实验结果来看, TBSR 有效地提高了 MPEG 流的质量。

5 结论

提出了一种新的源端控制的重传机制 TBSR。该机制能让所有需要重传的数据包在改进后的决策算法下达到较高重传成功率,同时不需要额外的管理信息,大大简化了重传机制。此外,计算出了能让重传成功的发送缓冲区的延迟以及其变化范围。又因为在 TBSR 融入了 TFRC,所以该机制对于拥塞有一定的应对能力。通过仿真实验,验证了计算出的发送缓冲区延迟能达到较高的重传成功率,并且 TBSR 能提高 MPEG 流的传输质量。

参考文献:

[1] Kohler, Handley, Floyd. RFC 4340 Datagram congestion control protocol[S]. Internet Engineering Task Force, 2006.

[2] Huszak A, Imre S. Selective retransmission of MPEG video streams over IP networks[C]// CSNDSP 2006, Patras, Greece, 2006.

[3] Picuch M, French K, Oprica G, et al. A selective retransmission protocol for multimedia on the internet[C]// Proceedings of SPIE International Symposium on Multimedia Systems and Applications, Boston, MA, USA, November 2000.

[4] Padhye J, Floyd S, Kohler E. Profile for DCCP congestion control ID 3: TFRC congestion control[S]. Internet-draft, 2002.

[5] Sato Y, Ata S, Oka I, et al. Using mixed distribution for modeling end-to-end delay characteristics[C]// Proceedings of the 8th Asia-Pacific Network and Management Symposium (APNOMS 2005), Japan, Okinawa, 2005.

[6] Kalman M, Girod B. Modeling the delays of successively-transmitted internet packets[C]// IEEE Conference on Multimedia and Expo, Taipei, Taiwan, 2004.

[2] Seung-Chul M Woo. Scalable routing protocol for Ad Hoc networks[J]. Wireless Networks, 2001(7): 513-529.

[3] Cheng C T, Lemberg H L, Philip S J, et al. SLALoM: A scalable location management scheme for large mobile Ad-Hoc networks[C]// Proceedings of IEEE Wireless Communications and Networking Conference, 2002: 574-578.

[4] Philip S, Ghosh J, Qiao Chun-ming. Performance analysis of a multilevel hierarchical location management protocol for ad hoc networks[J]. Elsevier Computer Communications, 2005(10): 1110-1122.

[5] Ho J S M, Akyildiz I F. Local anchor scheme for reducing signaling costs in personal communications networks[J]. IEEE/ACM Transactions on Networking, 1996, 4(5).

[6] Hou Ting-chao, Li V O K. Transmission range control in multihop packet radio networks[J]. IEEE Transactions on Communications, 1986, 34(1).