# 数据报拥塞控制协议(DCCP)研究

陈尚兵[1] 何小燕[2] 钱积新[1]
(浙江大学信息学院 杭州310027)[1] (朗讯科技(中国)贝尔实验室 北京100044)[2]

## A Study of Datagram Congestion Control Protocol(DCCP)

CHEN Shang-Bing[1] HE Xiao-Yan[2] QIAN JI-Xin[1]
(Information School,Zhejiang University,Hangzhou 310027)[1] (Bell Lab of Lucent Technologies(China),Beijing 100044)[2]

**Abstract** Datagram Congestion Control Protocol(DCCP) implements a congestion-controlled, unreliable flow of datagrams suitable for use by applications such as streaming media. This paper specifies its background, main ideas, ongoing research and future implementation.

**Keywords** Stream-media, DCCP, Congestion control

## 1. 引言

众所周知,网络的视频/音频服务在飞速发展中,但是流媒体的质量却不高。应用的开发者们想尽办法来提供尽可能平稳的流媒体服务,他们的解决方案大部分基于 TCP 和 UDP 协议。TCP 协议可以提供可靠的端到端的传输控制,但是它的传输速率抖动大,而流媒体并不需要完全可靠的传输;UDP 协议简单实用,但是不提供拥塞控制,不能和 TCP 友好共存,容易引起网络的拥塞。研究者们在分析了问题和比较了各种可能的解决方案后[1],提出一种新的传输控制协议:数据报拥塞控制协议(Datagram Congestion Control Protocol,DCCP)[2]。DCCP 结合了 TCP 和 UDP 的优点,适合于那些不需要 TCP 那样的按序和完全可靠的传输,但是又希望采用某种 TCP 友好的拥塞控制算法的流媒体传输控制。它具有可靠的连接建立和拆除的管理,灵活的拥塞控制策略,是目前业界讨论的热点。

## 2. TCP 和 UDP 应用到流媒体时的不足

TCP 是一种可靠传输协议,它保证数据报被接收端按序接收。一个典型的 TCP 的连接有3个阶段:连接建立阶段、传输阶段和连接拆除阶段。在连接建立时,发送端发送 TCP 同步报文,接收端对这个同步报文发回确认,最后发送端再次向接收端发送一个应答报文,从而完成了 TCP 连接的三次握手。在数据传输阶段,TCP 采用一种"和式增加积式减少"(AIMD)的拥塞控制策略,发送端每收到一个确认,就对拥塞窗口加1。当源端检测到丢包事件,则将当前的拥塞窗口减半,降低发送速率,并重传丢失的数据报,保证了每一个数据报都能到达接收端。这个可靠性对一些应用来说是很好的,比如 WWW 和文件传输,但是对流媒体不一定是必要的。流媒体可以容忍少量的帧的丢失,却不能容忍 TCP 的重传带来的迟延,以及 TCP 的窗口缩放带来的速率抖动。

UDP 协议的优点是简单。没有连接建立过程和拆除过程,没有可靠性保证。大部分流媒体应用,比如微软的媒体播放器使用 UDP 来播放多媒体数据,它们的拥塞控制算法是在 UDP 之上的。这样产生的问题是:1)UDP 不是 TCP 友好的。UDP 没有提供拥塞控制机制,数据的发送速率由接收端的接收速率决定,当网络拥塞时,共享带宽的 TCP 流根据拥塞控制算法降低发送速率,而 UDP 流继续以固定速率发送数据,这样就会造成 TCP 流得不到公平的带宽、甚至"饿死",严重的话有可能造成网络拥塞崩溃。2)即使在 UDP 之上采用拥塞控制算法,由于这些算法是各自独立设计的,也不能保证 TCP 友好性和它们之间的公平性。3)很多防火墙产品禁止 UDP 连接通过,这样防火墙内部的众多用户难以得到流媒体服务。

早期的解决流媒体传输控制问题的方案有 RTP 协议[5]和 SCTP 协议[6]。RTP 也是运行在 UDP 上的传输层协议,不能提供任何保证及时提交的机制,也不防止错序提交。SCTP 增强了 UDP 业务并提供数据报的可靠传输,采用类似 TCP 的拥塞控制策略。但是,SCTP 开销很大,它从 TCP 继承来的某些可靠性对于流媒体来说也是不必要的。近年来出现的一些 TCP 友好的拥塞控制算法如 GAIMD、TFRC 等,可以用来进行流媒体传输控制,但是这些算法基于 UDP 之上,依赖应用开发者自己实现。

## 3. 数据报拥塞控制协议(DCCP)

DCCP 原名是 DCP(Datagram Control Protocol),为了在发音上能和 TCP 有明显的差异,改称 DCCP。DCCP 的基本思想是结合 UDP 的不可靠传输特性和 TCP 的拥塞控制能力,可靠地建立和拆除连接,并提供可选的拥塞控制算法,让应用开发者从自己的拥塞控制算法的实现工作中解脱,让目前那些使用 UDP 协议的各种流媒体应用也能够轻松地转到使用 DCCP 来。

### 3.1 DCCP 的连接模型

类似 TCP,DCCP 也是面向连接的。每一个 DCCP 连接运行在两点,比如 A 和 B 之间。但是 DCCP 的连接的含义更丰富。在 A 和 B 之间不同方向上有两个独立的半连接(half-connection)。按照对半连接的隶属情况,有4种类型的报文:A 到 B 的数据报文,B 到 A 的确认报文,它们属于 A→B 的半连接;B 到 A 的数据报文,A 到 B 的确认报文,它们属于 B→A 的半连接。另有一种 Data+Ack 的报文,同时属于两个半连

陈尚兵 博士研究生,主要研究方向是网络控制与优化。何小燕 博士,主要从事第三代移动通讯研究。

接，比如A→B的数据报文捎带了从A→B的确认报文。这种半连接的模型允许A和B各自选择合适的拥塞控制算法，每个半连接的特性参数在两端之间可进行充分的协商，以适应不同方向上数据流的特性。

### 3.2 DCCP报文头的格式和含义

所有的DCCP报文含有如下格式的头部：

| 源端口号 | | | 目的端口号 |
|---|---|---|---|
| 类型 | 保留 | 序列号 | |
| 数据偏移 | #NDP | 校验长度 | 校验和 |

图1 DCCP报文头的格式

表1 DCCP使用的包类型

| 类型值 | DCCP包类型 | 说明 |
|---|---|---|
| 0 | DCCP-Request | 建立连接请求 |
| 1 | DCCP-Response | 响应DCCP Request |
| 2 | DCCP-Data | 承载数据的报文 |
| 3 | DCCP-Ack | 确认报文 |
| 4 | DCCP-DataAck | 捎带Ack的数据报文 |
| 5 | DCCP-CloseReq | 连接拆除请求 |
| 6 | DCCP-Close | 连接拆除 |
| 7 | DCCP-Reset | 端点复原 |
| 8 | DCCP-Move | 支持多主机和移动 |

2个16位的源端和接收端端口号标识了当前的半连接；4位的类型字段标识了DCCP报文类型，目前DCCP使用的报文有：4位的保留字段留做将来使用，如可以用来增强后面的24位序列号，以提高DCCP支持的最大传输量。在一个方向上，数据报文和控制报文都使序列号逐一增大，不管这些报文属于哪个方向上的半连接。一个方向上的初始序列号随机选取，以防止外界对连接的截听。接收端通过序列号来了解哪些包被丢失或者标记了。8位的偏移值指明了从DCCP头到数据报的有效载荷的长度，以32位字长为度量。4位的#NDP位记录了已发送的不含数据的报文个数，以16为模，这帮助接收端来判断一个丢失的数据报是否包含了用户的数据，比如：报文10的#NDP为5，报文11丢失了，但接收端发现报文12的#NDP仍然为5，即可以判断丢失的报文包含了数据。校验长度指出报文的哪些部位是被校验和覆盖的，如果为0，则所有内容被校验。校验和的算法是和TCP/IP相同的。

### 3.3 DCCP连接的建立和拆除

如图2，典型的DCCP的通讯过程有类似于TCP的三个阶段。在连接初始化阶段，接收端A发出一个DCCP-Request报文，指明端口号和服务要求，以及其它需要协商的半连接特性参数，比如采用的拥塞控制算法(CCID)，是否支持显式拥塞指示ECN，是否支持确认向量(Ack-Vector)等等。发送端B发出DCCP-Response报文来响应请求，回答了对半连接特性参数的意见，以及其它建议的连接特性。A发出DCCP-Ack报文来响应DCCP-Response，确认了B端的第一个报文序号，并可能进一步进行半连接特性协商，直到两端的意见统一。在连接建立后，数据报按照不可靠的方式发送，即：数据报不一定被确认，没有被确认的数据报不需要像TCP那样重传。当连接结束，DCCP又采用了可靠方式来完成拆除，确保两端结束连接状态：发送端发出DCCP-CloseReq，请求结束会话，接收端发出DCCP-Close，确认会话的结束，发送端发出DCCP-Reset来清空这个会话的状态，收到DCCP-Reset接收端维持一段合理的时间的连接状态，以接收剩余的数据；或者直接由接收端发出DCCP-Close要求拆除连接(如本例子所示)。

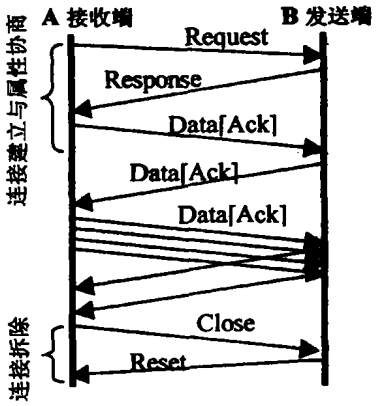


图2 DCCP的通讯流程图

在DCCP的连接初始化阶段和拆除，控制报文是可靠传输的，而传输阶段的数据传输是不可靠的。这种灵活性满足了流媒体应用的需要。

### 3.4 DCCP拥塞控制算法CCID

在考虑设计传输控制协议时的一个原则是：不应该由应用程序来设定发送数据的速率，否则的话，所有的应用程序会极力抢占带宽。因此，DCCP具有拥塞控制机制，并且不允许应用程序禁止拥塞控制。但是，DCCP给应用程序一个选择怎样的拥塞控制机制的机会。

管理DCCP的拥塞控制策略的模块为拥塞控制管理器，它提供的各种拥塞控制算法称为CCIDs(Congestion Control IDs)。目前有2种CCID可以选择：类TCP的拥塞控制(TCP-type congestion control)CCID2[3]和TCP友好的速率控制(TCP-Friendly Rate Control，TFRC)CCID3[4]。CCID2是基于窗口的，包含了类似于TCP的拥塞控制窗口，慢启动，拥塞避免和超时重传等；CCID3采用了基于方程的TFRC算法：发送端统计该方向上的半连接的丢包率p和往返传输时延$t_{RTT}$，根据下式调整其发送速率。TFRC算法能提供更平稳的数据流，并且数据流的发送速率能较快地收敛，吞吐率也较高。

$$T=s/(R\sqrt{\frac{2bp}{3}}+t_{RTO}(3\sqrt{\frac{3bp}{8}})p(1+32p^2))$$

在端到端之间，共存的以及不同方向上的半连接可以各自选择不同的CCID。另外，DCCP提供了添加新的拥塞控制算法的机制，这可以进一步扩展DCCP的性能。当然，所有新增的CCID应该是TCP友好的，并且得到IETF的承认。

### 3.5 DCCP半连接特性协商机制

DCCP的各半连接特性在连接建立过程中，以及在数据传输中途，根据应用程序的需要进行协商，协商过程中丢失的报文被强制重传，以保证可靠性。

所有的DCCP报文中，接着通用DCCP报文头部后面都有一个选项(option)字段，用来扩展DCCP报文的功能。选项字段的第一个八位字节标识了扩展功能的类型，其后的几个八位位组标识选项字段的长度和内容。主要的用于特性协商的功能扩展有：Change(类型值33)、Prefer(类型值34)和Confirm(类型值35)。内容字段中，第一个八位位组指示了协商的内容，比如是CCID，或者是关于是否支持确认向量、ECN、移动性，以及商量确认速率的大小等等，第二个八位位组指示了具体的特性值。Change被用来向特性拥有方(feature's location)请求改变特性值。特性拥有方可能会以Prefer通知对

方:将采用另一种值,或者可能以 Confirm 同意对方要求改变的值。未经协商,任何属性都不能改变。一个简单的 CCID 协商的过程如图3:B 请求 A 应用 ID 为3或者4的拥塞控制算法(注意按照 ID 排列顺序,B 更希望 A 采用 CCID3),A 则认为 CCID1,CCID2和 CCID5的算法比较好,B 只好重新请求 A 使用 CCID5,A 最终确认了 CCID5。

| | |
|---|---|
| B>A | Change(CC,3,4) |
| A>B | Prefer(CC,1,2,5) |
| B>A | Change(CC,5) |
| A>B | Confirm(CC,5) |

图3 一个 CCID 协商的例子

## 4. DCCP 的仿真和实现

DCCP 的基本设计已经结束,并已得到了 IETF 的承认。在刚刚结束的(2002年7月)日本横滨第54届 IETF 会议上 DCCP 工作组对协议的仿真和实现进行了深入的讨论,并总结了一些需要解决的问题。

目前有多个组织和个人在仿真研究或实现 DCCP 协议,其中 Patrick McManus 在 LINUX 内核中实现了基本的 DCP 协议[7];加利福尼亚大学伯克利分校的研究组织 CITRIS 在 NS 软件中对 CCID3进行了仿真,验证了 CCID3良好的 TCP 友好性[8]。他们正在进行其他 CCID 是实现,以及向 Linux、BSD 等系统中移植 DCCP。他们的实现允许新的拥塞控制算法能被容易地嵌入,留下了扩展空间。

已有的实现并没有提供良好的应用接口,因而不利于应用程序从其他协议转到 DCCP 来。在 DCCP 设计中的一些重要思想,比如支持确认向量等,是提高 DCCP 性能的一个关键,但是这些设计思想需要大量的仿真和实验来检验。另外,DCCP 协议的初衷是为用户提供一个轻型的传输控制协议,但是由于要实现众多的选项功能,现在的 DCCP 的包头是比较大的。如何改进设计,尽可能地减小包头的大小是一个开放的问题。

**结语** DCCP 的研究和实现还不充分,最终也许会有重要的变化。在最终的多媒体流传输控制方案确定之前,大部分应用还得使用 TCP 或者 UDP,或者其它基于它们之上的协议。但是研究结果表明 DCCP 是有效和 TCP 友好的,并且 DCCP 的研究表达了流媒体传输控制的清晰框架——不管最终的协议是如何的,一些基本的原则是:TCP 的拥塞控制机制决定了什么是公平的数据发送速率;UDP 的不可靠传输和简单性将是数据传输方式的目标。

### 参 考 文 献


1 Floyd S,Handley M,Kohler E. Problem Statement for DCP. Internet-draft, February 2002, Work in progress
2 Kohler E, Handley M, Floyd S, Padhye J. Datagram Congestion Control Protocol (DCCP), Internet-draft, June 2002, Work in progress
3 Floyd S, Kohler E. Profile for DCCP Congestion Control ID 2: TCP-like Congestion Control. Internet-draft, May 2002, Work in progress
4 Padhye J, Floyd S, Kohler E. Profile for DCCP Congestion Control ID 3: TFRC Congestion Control. Internet-draft, June 2002, Work in progress
5 Schulzrinne H, et al. RTP: A Transport Protocol for Real-Time Applications. RFC 1889
6 Stewart R, et al. Stream Control Transmission Protocol. RFC 2960
7 http://www.ducksong.com:81/dccp/
8 Sohn T, Zolfaghari E. Experimentations in Datagram Control Protocol: [Tech. Report]. Tech. Report, University of California at Berkeley, May 2002


(上接第114页)

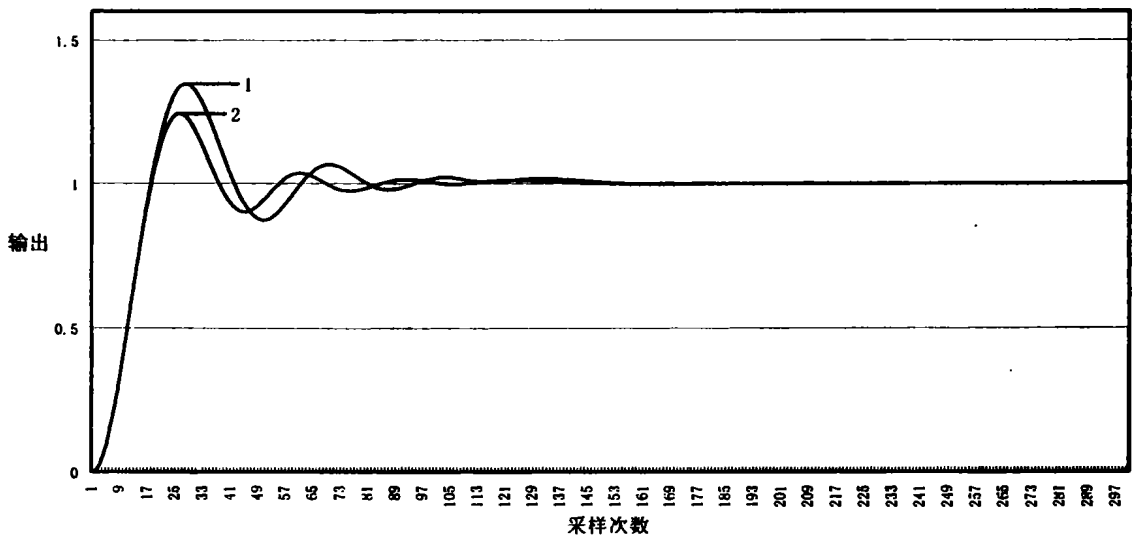


图4 二阶对象式(7)的系统响应:曲线1为只调整比例因子进行控制,曲线2为调整比例因子与修改规则相结合进行控制

图4显示了规则修改前和修改后对控制性能的影响。

### 参 考 文 献


1 Mandani E H. Applications of fuzzy algorithms for simple dynamic plant. Proc IEEE, 1974, 121(12): 1585~1588
2 Mudi Rajani K, Pal Nikhil R. A Robust Self-Tuning Scheme for PI- and PD-Type Fuzzy Controllers. IEEE Trans on Fuzzy Systems, 1999, 7(1): 2~15
3 章卫国,等. 模糊控制理论与应用. 西北工业大学出版社, 1999
4 龙升照,汪培庄. Fuzzy 控制规则的自调整问题. 模糊数学, 1982, 2(3): 105~111
5 He Shi-Zhong, Tan Shaohua, Hang Chang-Chieh, Wang Pei-Zhuang. Control of dynamical processes using an on-line rule-adaptive fuzzy control system. Fuzzy Sets and Systems, 1993, 54: 11~22
6 Nelder J A, Mead R. A simplex method for function minimization. The Computer Journal, 1965, 7: 308~313
7 李士勇. 模糊控制. 神经控制和智能控制论. 哈尔滨工业大学出版社, 1996
8 李东辉. Fuzzy 控制规则自调整和 Fuzzy 控制系统寻优及其仿真研究. 模糊数学, 1986, (3): 53~61