# 具有显式反馈的拥塞控制系统研究进展

张慧翔 戴冠中 姚 磊 潘文平
(西北工业大自动化学院 西安 710072)

**摘 要** 近年来越来越多的高带宽光纤网络和长时延的卫星网络融入到 Internet 中来。研究发现,传统 TCP 协议的拥塞控制机制随着网络带宽和延迟的增大而趋于不稳定。研究这种高带宽时延乘积网络环境下的拥塞控制机制成为热点研究课题,其中一个重要趋势就是采用显式反馈方法。本文讨论了基于 IP 网络中的显式反馈机制,包括传统 ECN、各种类 ECN 机制以及新的拥塞控制协议;分析了各种机制的特点,总结了现有研究的问题并指出了几个值得继续研究的要点。

**关键词** 拥塞控制,显式反馈,高带宽时延乘积网络

## Survey of Congestion Control Mechanism with Explicit Feedback

ZHANG Hui-Xiang DAI Guan-Zhong YAO Lei PAN Wen-Ping
(College of Automation, Northwestern Polytechnical University, Xi'an 710072)

**Abstract** In recent years, as the Internet evolves to incorporate very high-bandwidth optical links and more large-delay satellite links, the congestion control mechanism in traditional TCP become inefficient and prone to instability. Research on the congestion control issue in the high bandwidth-delay product networks is very active. A important trend is to improve the congestion control with explicit feedback from routers in the transmit path. The explicit feedback mechanism based on IP networks, including ECN, ECN-like and some typical new congestion control protocols, are studied in this paper. Their strengths and weaknesses are also analyzed. At last, the further research aspects are pointed out.

**Keywords** Congestion control, Explicit feedback, High bandwidth-delay product networks

## 1 引言

随着计算机网络的迅速发展,拥塞控制机制的目标从单独地避免拥塞趋向于怎样有效地利用网络资源。TCP 协议的拥塞控制机制[1]已显示出很多不足,它是基于隐式反馈的,不依赖于网络的中间节点,将丢包和传输延迟的增长作为网络出现拥塞的指示。在实际网络中,引起丢包和延迟变化的因素是很多的:可能由于中间节点队列溢出引起丢包,也可能由于设备故障或者链路干扰导致节点校验错误引起丢包;可能由于中间节点队列长度增长引起延迟增加,也可能由于传输路径很长(如卫星链路)或者链路层数据包重传引起延迟增加。因此,TCP 协议拥塞控制机制采用的拥塞指示是不确定的。

TCP 协议采用了保守的拥塞窗口增长方式,不能有效地利用网络带宽。TCP 拥塞避免算法使得在一个 RTT 内拥塞窗口至多增长一个 MSS(Max Segment Size),也就是所谓的加性增长 AI。在 RFC3649[2] 中提到,一个 RTT 为 100ms、MSS 为 1500bytes 的 TCP 连接,为了达到约 10Gbps 的传输速率,要求其拥塞窗口必须增长到 83333 个 MSS,并要求每 $5\times10^9$ 个数据包产生一次丢包,这对当前网络而言是不切实际的。

由于 TCP 的传输速率跟 RTT 成反比例关系[24],随着无线广域网和卫星链路的迅速发展,经这些链路传输的数据流将被分配较少的带宽,导致了严重的不公平性问题。

分析表明,现有 TCP 配合主动队列管理 AQM(Active Queue Management)[25] 的拥塞控制机制随着链路带宽以及传输延迟的增长会趋于震荡,变得不稳定,不适合将来高带宽时延乘积网络的要求。文[3]基于控制理论的框架给出了分析。

为了弥补 TCP 现有拥塞控制机制的不足,近年来涌现了大量的改进措施。HighSpeed TCP[2]、Scalable TCP[4]、BIC[5]、CUBIC[6]、TCP Westwood+[7] 等通过修改现有 TCP 响应函数,快速地收敛到平衡状态,在拥塞事件发生时放慢退避速率,以有效地利用网络带宽。TCP Vegas[8]、Fast TCP[9] 则结合数据包传输延迟来探测网络负载状态。这些改进都是端节点收集信息来猜测网络负载状态,是一种隐式反馈。这种猜测的不确定性和滞后性使其存在很多局限性。

新的拥塞控制机制的一个重要趋势就是采用显式反馈机制。显式反馈需要中间节点(如路由器)的参与,它明确地告知数据发送端关于网络中的负载状况,去除了 TCP 拥塞控制机制中反馈不确定性问题,同时数据发送端也能及时地根据显式反馈的网络负载信息,进行更积极的响应,更有效地利用网络带宽。

显式反馈的实现有很多种方式。如 Source Quench[26] 方式中,路由节点直接通知数据发送端调节发送速率,不需要数据接收节点的参与。ATM ABR[27] 服务通过由发送端发送资源管理数据包获取传输路径上允许的最大发送速率,最后通过接收端反馈回发送端。ECN[10] (Explicit Congestion Noti-

张慧翔 博士研究生,研究方向为 Internet 网的拥塞控制和 QoS 控制;戴冠中 教授,博士生导师,研究方向为网络化控制、控制理论;姚 磊 博士研究生,主要研究方向为点对点分布式系统;潘文平 博士研究生,主要研究方向为 Web QoS 控制。

fication)方式中，路由节点根据网络负载标记 IP 包头中的比特位，标记信息由接收端反馈回发送端，发送端根据标记信息调节发送速率。

本文主要介绍并分析在 IP 网络中的显式反馈机制，包括 ECN、类 ECN 和一些新的显式拥塞控制机制。文章组织如下，第 2 节分析了 ECN 机制和它的几个变种。第 3 节分析了几种新的具有代表性的显式反馈机制。最后总结了现有的显式反馈机制，并指出了几个研究要点。

## 2 ECN 与类 ECN 机制

ECN 方式由于其实现简单，路由节点不需维护数据流状态，不需额外产生数据包，且能很好地嵌入到现有互联网络中，因而得到了广泛的关注，已有很多应用。在 ECN 的基础上，又提出了 Anti-ECN[11] 和 Multilevel ECN[12]。下面分别介绍这三种 ECN 机制。

### 2.1 ECN[10]

ECN 特点是在网络出现拥塞时标记 IP 数据包中的一个比特位。它的实现需要 AQM 的参与。原有 AQM 机制(如 RED[13])根据网络状况提前丢弃数据包，使数据发送端提前做出拥塞避免反应；当结合 ECN 时则不再单纯丢弃数据包，而是配合标记数据包头中的一个比特位，该位置 1 表示中间节点存在拥塞。

数据接收端将标记信息反馈到数据发送端，发送端对置位的数据包按照丢包事件进行响应。

AQM 标记的只有 IP 包头中的一个比特位，而在实际部署时，ECN 机制采用四个比特位和 TCP 协议结合起来。IP 头中有两个比特位：ECT(ECN Capable Transport)标志位和 CE(Congestion Experienced)标志位。ECT 标志位置 1 表示数据发送端支持 ECN 机制，中间节点可以标记 CE 位。CE 位就是用来指示网络中是否存在拥塞。对于不支持 ECN 的数据发送端，中间节点的 AQM 机制依然采用丢包的方式。TCP 头中有两位：ECE(ECN-Echo)标志位和 CWR (Congestion Window Reduced)标志位。数据接收端利用 ECE 位通过 ACK 包将 CE 位信息反馈到发送端。发送端采用 CWR 位告知接收端，它已做出拥塞避免反应。

一些恶意的用户为了获得更大的发送速率，可能在接收端忽略 CE 位，而不反馈网络拥塞信息。为了防止接收端欺骗的问题，发送端设置一个随机数跟随整个反馈过程。首先在发送端，数据包设置 ECT = 1、CE = 0 和一个随机数。当中间节点检测到拥塞标记 CE 位为 1，同时随机数被清 0。数据接收端通过 ACK 包反馈 ECE 位和随机数(已经清 0)。发送端发现随机数被清 0，则知道网络存在拥塞，无论 ECE 位是否置 1 都将做出拥塞避免反应。发送端为了欺骗发送端必须猜测这个随机数。

### 2.2 Anti-ECN(AECN)[11]

Anti-ECN 是一种类似于 ECN 的方式，它同样利用了 ECN 中 IP 头的两个比特位，一位表示发送端支持 Anti-ECN 机制，一位为中间节点标记位(AECN)。Anti-ECN 最大的不同是在中间节点低负载时标记 AECN 位为 1，通过数据发送端采用比 TCP 更积极的响应获取更高的网络利用率。在数据包发送时，AECN 位初始置为 1。

在中间节点，Anti-ECN 结合 AVQ[14] 的主动队列管理算法来进行标记。中间节点维护一个大小为 $B$ 的虚拟队列，相应处理能力为 $\eta C$，$0<\eta<1$，C 是节点的带宽。当每一个数据包到达中间节点，计算虚拟队列长度，如果此时虚拟队列长度为 0，那么表示节点负载较轻，对 AECN 位跟“1”进行与操作；如果虚拟队列长度大于 0，对 AECN 位跟 0 进行“与”操作。然后更新虚拟队列长度，加上数据包大小，等待下一个数据包到达。如果虚拟队列溢出，则丢弃相应数据包。标记的信息经接收端由 ACK 反馈到发送端。

数据发送端根据 AECN 标志位是否置 1 采用不同的算法。拥塞窗口 $W$ 的增长方式为

$$W=W+\frac{\Delta}{W}$$

$\Delta$ 为窗口增长大小。如果 AECN 位置 1 则 $\Delta=W$；如果置 0 则 $\Delta=1$。对丢包事件的响应依然是拥塞窗口减半。发送端根据标志位来选择窗口增长算法，而不是像 TCP 一样根据一个固定的阈值[20]来转换算法，从而获得了更大的发送速率。

文[11]通过仿真实验详细分析了 $B$ 和 $\eta$ 的值对网络性能的影响，但并没有给出一个最优值，所以 $B$ 和 $\eta$ 的选择是一个难点。

### 2.3 Multilevel ECN(MECN)[12]

传统 ECN 和 AECN 都是一种二元指示，指明传输路径上存在或不存在拥塞，并没有指明拥塞的程度。为此 MECN 尝试将网络负载分等级反馈回接收端。MECN 在传统 ECN 基础上把存在拥塞的情况扩展为三个等级：轻度、中度和重度拥塞。MECN 利用 IP 头 ECN 使用的两个标志位(CE，ECT)，表 1 说明了两个位的不同组合的含义。丢包事件表示传输路径经历重度拥塞。

表 1 MECN(CE,ECT)组合

| (CE,ECT) | 含义 |
|---|---|
| 00 | 发送端不支持 MECN |
| 01 | 传输路径没有经历拥塞 |
| 10 | 传输路径经历轻度拥塞 |
| 11 | 传输路径经历中度拥塞 |

MECN 在中间节点采用 RED 主动队列管理算法来进行标记。为了配合这种分等级的定义，RED 算法的标记机制做了相应修改，新定义一个位于 min_th 和 max_th 之间的阈值 mid_th。TCP 数据发送端初始标记 CE 位和 ECT 位为“01”或者“00”。RED 算法计算平均队列长度，当平均队列长度小于 min_th 则什么都不做；当平均队列长度位于 min_th 和 mid_th 之间，标记为“10”，标记概率为 0～P1max；当平均队列长度位于 mid_th 到 max_th 之间，标记为“11”，标记概率为 0～P2max；当平均队列长度超过 max_th 则丢弃数据包。如图 1 所示。

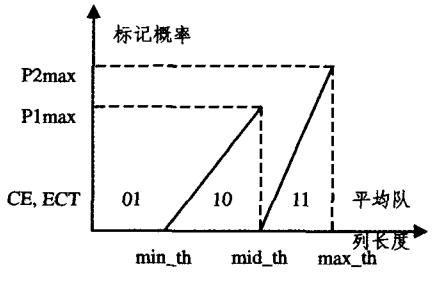


图 1 MECN 标记示意图

数据接收端根据 IP 包头中两个标志位(CE，ECT)，设置 TCP 头中的两个标志位(CWR，ECE)，并通过 ACK 向发送端

反馈。数据发送端收到应答 ACK，根据 TCP 头两个标志位(CWR，ECE)做出不同的响应。响应函数如表 2 所示。

表 2　MECN 端系统响应函数

| 负载指示 | 端系统响应函数 |
|---|---|
| 没有拥塞 | $cwnd=cwnd+1/cwnd$ |
| 轻度拥塞 | $cwnd=cwnd*80\%$ |
| 中度拥塞 | $cwnd=cwnd*60\%$ |
| 重度拥塞 | $cwnd=cwnd*50\%$ |

MECN 采用了 IP 包头中的两个标志位(CE，ECT)，由于此后提出的 RFC3168[10]把该两位的组合用于防止接收端的欺骗而传递一个随机数的比特位。两者发生冲突，导致 MECN 在实际中不可行。MECN 给我们的启示是一种更精确的网络信息反馈会带来更大的性能改善。文[12]通过仿真实验表明 MECN 与传统 ECN 相比，具有较小的平均队列长度和队列收敛时间，具有较小的丢包率和较高的链路利用率。

## 3　几种典型的显式反馈机制

ECN 的提出将 TCP 协议原有的隐式反馈变革为显式反馈。MECN 表明，细粒度的反馈带来更好的效益。近年来提出的新的拥塞控制机制普遍采用了细粒度反馈的方式，与现有 ECN 方式相比，路由节点和端系统都有很大改变。我们选择了几种有代表性的机制进行了分析，包括 Quick Start[15]、XCP[16]、VCP[17]、RCP[18]和 CADPC[19]。我们主要关注中间节点的反馈方式和端系统对显式信息的响应。

### 3.1　Quick Start(QS)[15]

现有 TCP 设置初始拥塞窗口为 1 到 4 个 TCP 段(MSS)大小[20,21]。较小的初始拥塞窗口，使 TCP 在高带宽时延乘积网络中需要较长的慢启动时间。对于数据传输量较小的 TCP 连接，慢启动阶段占用了过多的时间，效率低下。因此，RFC3649[2]建议，慢启动阶段拥塞窗口在中间节点的帮助下设置较大的初始值。该问题被 IETF 单独提出来，即 Quick Start，同时给出了 QS 在 TCP 中应用的实现。

QS 定义了 IP 的可选字段 Quick Start Request(QSR)，包含请求的发送速率、QS TTL 值和一个 30bits 的随机数。TCP 发送端在建立连接时设置 SYN 包或者 SYN/ACK 包的 IP 字段 QSR，这种数据包称为 QS 请求数据包。QS TTL 被设置为随机值，并保存 IP TTL 和 QS TTL 的差值 TTL Diff。

中间路由节点根据网络负载状态可以允许、拒绝或者降低请求速率，同时对 IP TTL 和 QS TTL 减 1。路由节点不允许提高请求速率。如果降低请求速率则修改随机数中对应比特位为新的随机数位。如果拒绝该 QSR，则删除 IP 包头中的 QSR 字段或者对 QSR 字段的请求速率、QS TTL、随机数清 0。不支持 QS 的路由节点不更改 QSR 字段的值。

QS 定义了 TCP 包头的可选字段 Quick Start Response，包含反馈的请求速率、TTL Diff 和接收到的随机数。TCP 接收端接收到 QSR 数据包，计算 TTL Diff，复制 QSR 包中的请求速率和随机数到 TCP 包头，反馈回 TCP 发送端，这种数据包称为 QS 响应数据包。如果 QSR 的请求速率为 0，接收端不发送 QS 响应数据包。发送端检查 QS 响应数据包 TTL Diff 值，如果与发送时一致，表示路径上所有路由都支持 QS。发送端检查随机数是否符合修改规则，保证接收端没有欺骗行为。发送端如果没有收到 QS 响应数据包，则采用 TCP 拥塞控制算法。

QSR 请求成功，发送端设置初始拥塞窗口大小为 cwnd $=(R\times T)/(MSS+H)$，R 为请求速率，T 为当前估计 RTT 值，H 为 IP 和 TCP 包头大小，cwnd 单位为 MSS。

除了在 TCP 连接建立阶段使用 QS，文[15]还建议在 TCP 发送端经历一段空闲时间后无法有效估计可用带宽时使用，比如一个 HTTP 连接，在用户思考时间内没有数据传输，TCP 的发送速率无法反应当前网络状态，即可采用 QS，获取合适的窗口大小。

QS 通过路由节点的支持，增大了 TCP 慢启动算法的拥塞窗口初始值，缩短了协议的收敛时间。IETF 对 QS 机制的实现正在不断的完善。

### 3.2　XCP(eXplicit Control Protocol)[16]

XCP 是基于窗口的拥塞控制协议，它不再简单地对网络负载划分等级，而是将路由节点的空余带宽根据每个数据流的 *rtt* 值和拥塞窗口 cwnd 值分配到所有经过该节点的数据流中去，显式地告诉发送端拥塞窗口的大小。

#### 3.2.1　拥塞头部(Congestion header)

路由节点并不需要维护每个数据流的状态信息，而是通过 XCP 数据包的拥塞头部携带状态信息。拥塞头部有三个字段：当前拥塞窗口大小 H_cwnd，当前估计 RTT 值 H_rtt，发送端期望拥塞窗口增量 H_feedback。前两个字段对路由节点是只读的，最后一个字段由路由节点更新以直接控制发送端的拥塞窗口大小。H_feedback 可以是正或为负的增量。如果应用程序希望达到速率 r，则发送端设置

$$H\_feedback=(r*H\_rtt-H\_cwnd)/n$$

其中 $n$ 为当前拥塞窗口中的数据包数。路由节点可以根据网络负载状态修改 H_feedback，经接收端反馈回发送端。发送端根据反馈重新设置拥塞窗口：

$$cwnd=\max(cwnd+H\_feedback,s)$$

s 为数据包的大小。如果网络带宽允许，那么在一个 RTT 后发送端将达到 r 的发送速率。对于丢包事件，XCP 跟 TCP 一样减半 cwnd。

XCP 接收端类似于 TCP 接收端，只是在应答 ACK 中复制了接收到的数据包中的拥塞头部信息，以反馈到发送端。

#### 3.2.2　XCP 路由节点控制器

XCP 路由器设计了两个控制器：效率控制器(efficiency controller)和公平控制器(fairness controller)。前者计算路由节点空余带宽，后者根据每个数据流的 *rtt* 值和拥塞窗口大小分配空余带宽。在控制周期的选择上，XCP 取一段时间内处理的所有数据包携带的 H_rtt 的平均值，取平均值的好处在于可以平滑基于窗口协议的突发性问题[28]。

效率控制器最大化利用网络带宽，它计算在一个控制周期中系统能提供多大的空余带宽，该值用可增加的字节数 $\Phi$ 来表示，可能为正也可能为负值。$\Phi=\alpha dS-\beta Q$，$\alpha$，$\beta$ 为常数，推荐取值 0.4 和 0.226，$d$ 为控制周期大小，$S$ 为空余带宽(也就是路由节点带宽与数据到达速率的差值)，$Q$ 就是持久队列长度。$\Phi$ 的计算是基于乘性增加乘性减少原则(MIMD)[28]，使 XCP 能有效地利用网络带宽。

公平控制器负责将 $\Phi$ 公平地分配到一个周期内经过路由节点的每一个数据包中去。如果 $\Phi>0$，所有数据流平均分配空余带宽，也就是线性增长；如果 $\Phi<0$，根据各个数据流占用的带宽来分配，也就是乘性减少。当 $\Phi=0$，为了让公平性持续收敛，提出了带宽漂移(bandwidth shuffling)的概念，总有一定量的带宽同时参与 AIMD。漂移带宽 $h$ 的计算如下：$h$

$=\max(0,\gamma y-|\Phi|)$，$y$ 是在一个控制周期内数据到达速率，$\gamma$ 推荐值为 0.1。

这样 XCP 分配给一个控制周期内第 $i$ 个数据包的增量为：$H\_feedback_i=p_i-n_i$。$p_i$ 是通过 AI 的反馈量。在一个控制周期内，空余带宽被平均分配，拥塞窗口增量 $p_i$ 为空余带宽和 $rtt$ 的乘积，即 $p_i$ 与 $rtt_i$ 成正比；$p_i$ 被分配到数据流的每个数据包中去，也就是说 $p_i$ 跟其所在数据流数据包数目成反比；而数据包数目与 $cwnd_i/s_i$ 成正比，与 $rtt_i$ 成反比。定义 $p_i=\varepsilon_p\frac{rtt_i^2 s_i}{cwnd_i}$，$\varepsilon_p$ 为常量，$s_i$ 为第 $i$ 个数据包大小。$n_i$ 是通过 MD 的反馈量，在一个控制周期内，$n_i$ 按照各个数据流当前占用带宽比例来分配，即与当前拥塞窗口成正比，与数据包数目成反比，与 $rtt$ 成正比，定义 $n_i=\varepsilon_n rtt_i s_i$，$\varepsilon_n$ 为常量。

仿真实验表明 XCP 可以达到很高的链路利用率、很小的排队队长和几乎接近于零的丢包率，能适应高带宽时延乘积网络以及传统网络环境，能显著改善网络的性能，具有较强的公平性。

3.2.3 XCP 的启示

XCP 协议是一种显式准确的反馈，类似于 ATM 网络中 ABR 机制[27]，但是 XCP 不需要在路由节点维护数据流状态，而是转移到数据包拥塞控制头中，XCP 路由节点对每个数据包需要三次乘法和一些加法操作。

XCP 协议提出了效率控制和公平控制分离的概念，提高了协议设计的灵活性和扩展性，如可以修改公平控制器提供区分服务。

XCP 协议路由参数 $\alpha,\beta$ 是常数，与数据发送节点数目，端节点 $rtt$ 以及链路带宽无关，而我们知道 TCP/AQM 机制会随节点数目、节点 $rtt$ 以及链路带宽的变化性能变得不稳定。

XCP 协议的不足在于提供了拥塞控制头，需要额外的 IP 头空间；XCP 路由节点需要一定量的计算，这对于 XCP 的应用也有一定阻碍。

### 3.3 VCP (Variable-structure congestion control protocol)[17]

VCP 协议采用了 MECN 方式的标记模式，使用 IP 包头中 ECN 的两个比特位(CE,ECT)，两个位的不同组合指明不同的情况："00"表示发送端不支持 VCP，"01"表示网络为低负载，"10"表示网络高负载，"11"表示网络过载。VCP 根据负载因子(Load Factor)来标记数据包。VCP 发送端根据不同的网络负载采用不同的窗口增长策略。

VCP 路由器在一个固定的周期 $t_p$ 内计算负载因子 $\rho_l$。研究表明 Internet 上数据流的 RTT 值都小于 200ms[29]，因此 $t_p$ 取 200ms。$\rho_l$ 的计算公式如下：

$$\rho_l=\frac{\lambda_l+\kappa_q\tilde{q}_l}{\gamma_l C_l t_p}$$

$\lambda_l$ 示在周期内的数据流量，$\tilde{q}_l$ 表示周期内的持久队列长度，$\kappa_q$ 表示队列收敛速度(取 0.5)，$\gamma_l$ 表示处理效率(取 0.98)，$C_l$ 表示链路带宽。定义 $\hat{\rho}_l$ 为量化的负载因子值。表 3 说明了路由节点的标记策略。

表 3 VCP 路由节点标记策略

| $\rho_l$ | $\hat{\rho}_l$ | 负载 | (CE,ECT) |
|---|---|---|---|
| $0\leqslant\rho_l<80\%$ | $\hat{\rho}_l=80\%$ | 低负载 | 01 |
| $80\leqslant\rho_l<100\%$ | $\hat{\rho}_l=100\%$ | 高负载 | 10 |
| $\rho_l\geqslant 10\%$ | $\hat{\rho}_l>100\%$ | 过载 | 11 |

VCP 数据接收端通过 ACK 将(CE,ECT)反馈回数据发送端。发送端响应如表 4 所示。

表 4 VCP 端系统响应函数

| (CE,ECT) | 端系统响应函数 |
|---|---|
| 01 | MI: $cwnd(t+rtt)=cwnd(t)*(1+\varepsilon)$ |
| 10 | AI: $cwnd(t+rtt)=cwnd(t)+\alpha$ |
| 11 | MD: $cwnd(t+\delta t)=cwnd(t)*\beta$ |

其中 $rtt=t_p$，$\delta t\to 0+$，$\varepsilon>0$，$\alpha>0$，$0<\beta<1$，推荐取 $\alpha=1$，$\beta=0.875$。定义 $\varepsilon=\kappa\frac{1-\hat{\rho}_l}{\hat{\rho}_l}$，$\kappa$ 为常数(取 0.25)。在低负载阶段，$\hat{\rho}=80\%$，那么 $\varepsilon=0.0625$。

为了处理 $rtt$ 不同的情况，VCP 对 MI，AI 阶段的参数根据当前 $rtt$ 值进行了修正：

$$\varepsilon_s=(1+\varepsilon)^{rtt/t_p}-1,\alpha_s=\alpha\frac{rtt}{t_p}$$

MD 阶段不受 RTT 影响。这种修正使具有不同 $rtt$ 的 VCP 数据流的拥塞窗口能收敛到相同的数值。为了保证发送速率($rate=cwnd/rtt$)收敛的公平性，在 AI 阶段，取 $\alpha_{rate}=\alpha_s\frac{rtt}{t_p}=\alpha(\frac{rtt}{t_p})^2$。

仿真实验表明 VCP 能够达到 XCP 同样的性能，即具有高链路利用率，低持久队列长度和接近于 0 的丢包率，但是公平性收敛较慢。VCP 只利用 IP 头中的两个标志位达到了很好的网络性能，比较容易推广。同时，VCP 能够独立于端节点 rtt 和链路带宽。不足之处在于控制周期取为常量 200ms，完全是实验选择的结果。

### 3.4 RCP(Rate Control Protocol)[18]

RCP 的设计目的是使各个数据流平等地分享瓶颈链路的带宽。传输路径上的路由节点负责分配链路带宽。RCP 采用基于速率的拥塞控制机制。RCP 类似于 XCP 扩展了数据包头。扩展的包头中包含当前 RTT 的估计值和期望发送速率 $R_p$ 两个字段。

发送端在发送每个数据包时设置当前 RTT 估计值和期望的发送速率(也可以设置为无穷大)。路由节点维护了一个平均 RTT 值 $d_0$ 和一个允许发送速率值 $R(t)$，要求所有通过该路由的 $R_p$ 数据流的最大不能超过 $R(t)$。路由节点每收到一个数据包，检查包头中的 $R_p$，如果 $R_p>R(t)$，那么设置 $R_p=R(t)$；如果，$R_p\leqslant R(t)$那么维持 $R_p$ 不变。同时，跟 XCP 类似，路由节点利用数据包中的 RTT 值更新 $d_0$。数据包沿传输路径到达接收端，$R_p$ 值被设置为不超过该路径上允许的最大传送速率。接收端将 $R_p$ 反馈回发送端。发送端以 $R_p$ 值发送数据。当出现丢包情况，RCP 重传数据包。

对每一个数据包计算 $d_0$：

$$d_0=gain\times RTT_{packet}+(1-gain)\times d_0^{last}$$

权值 $gain=0.02$。这种意味着数据传送时间较长的数据流对 $d_0$ 的影响较大，也正是这种数据流的发送行为决定着闭环控制的稳定性。在每个平均 RTT 时间 $d_0$ 内计算 $R(t)$：

$$R(t)=R(t-d_0)+\frac{[\alpha(C-y(t))-\beta\frac{q(t)}{d_0}]}{\hat{N}(t)}$$

其中 $R(t-d_0)$为上个 $d_0$ 时刻的速率值，$C$ 为链路带宽，$y(t)$为 $d_0$ 时间内数据包的到达速率，$q(t)$为即时队列长度，$\alpha,\beta$ 为常量。$\hat{N}(t)=C/R(t-d_0)$表示路由节点估算的当前数据流数目。表 $C-y(t)$示空余的带宽，$\frac{q(t)}{d_0}$表示队列清空所需的

带宽，表达式 $ABW_{d_0}=\alpha(C-y(t))-\beta\frac{q(t)}{d_0}$ 表示在下个 $d_0$ 时间内变化的带宽，最后除以 $\hat{N}(t)$，平均分配给所有数据流。

在 RCP 的实现中 $R(t)$ 的更新周期并不是 $d_0$，而是 $T=\min(\tau,d_0)$，$\tau$ 为用户定义的常数值，仿真试验设置为 10ms。$T$ 保证 $R(t)$ 在一个平均 RTT 时间内至少更新一次。通过设置较小 $\tau$ 的值可以加快 $R(t)$ 的更新频率，这样每次变化的带宽就为 $ABW_{d_0}\times\frac{T}{d_0}$。在每个 $T$ 计算 $R(t)=R(t-T)[1+\frac{ABW_{d_0}\times\frac{T}{d_0}}{C}]$。

RCP 新定义数据流完成时间(Flow Completion Time)[30]来描述其优势。数据流完成时间定义为连接建立时间(一个 RTT)加上数据传输时间。采用 RCP 协议，每个数据流在开始的第一个 RTT 时间内就能获得一个公平的发送速率，而对于 TCP 数据流，由于初始拥塞窗口很小，可能需要多个 RTT 才能收敛到一个公平的发送速率。对于 XCP，它对新数据流带宽分配的速度较慢，特别是在空余网络带宽很小的情况下，需要多个 RTT 才能收敛到公平的发送速率。现有 Internet 中数据流的大小呈重尾分布[31]，这就意味着利用 RCP 大部分的数据流在几个 RTT 时间内就能完成数据传输。RCP 这种在第一个 RTT 收敛的方式可能会导致短时间内的网络过载，引起路由队列的突发性增长，有丢包的可能性。过低估计 $\hat{N}(t)$ 也会导致网络过载。RCP 在复杂多变的网络环境下合理估计 $\hat{N}(t)$ 需要深入的研究。

仿真试验表明，RCP 能较好地适应网络环境动态变化，性能不受链路带宽、传送延迟和数据流大小变化的影响，能保持较高的网络利用率和基本为零的丢包率，能较好保证数据流公平的享有瓶颈链路，较 TCP，XCP 有更快的数据流完成时间。

### 3.5 CADPC/PTP [19]

XCP、VCP 和 RCP 等协议都是利用传输的数据包来获取反馈信息，CADPC/PTP 则利用额外的信令数据包来获取反馈。

PTP(Performance Transparency Protocol)[22]协议用于获取各种网络性能参数，如平均瓶颈队列长度、链路最大传输单元 MTU 等。CADPC 通过 PTP 获取路由节点的状态信息。

发送端发送 PTP 数据包到接收端，传输路径上的每个路由节点附加相应信息到 PTP 数据包，包括路由接口地址、路由带宽、业务流量和当前时间标签。路由带宽取路由 MIB[23]对象的 ifSpeed 值，单位为 bps；业务流量取 MIB 对象的 ifOutOctets 或 ifInOctets 值，单位为 byte。

接收端维护了传输路径路由信息表，记录收到的 PTP 数据包中由各个路由节点附加的信息。接收端每收到两个 PTP 数据包，计算每个路由的时间差值 $\delta$、链路带宽 $B$ 和业务流量差值 $\lambda$。接收端计算可用带宽 $B-\frac{\lambda}{\delta}$。接收端将可用带宽最小的路由信息(即 $\delta,B,\lambda$)反馈回接收端。

CADPC(Congestion Avoidance with Distributed Proportional Control)是 ATM ABR 服务交换机 CAPC 机制的分布式版本。发送端接收到 PTP 数据包的反馈，计算下一时刻的发送速率：

$$x(t+1)=x(t)(2-\frac{x(t)}{B(t)}-\frac{\lambda(t)}{B(t)})$$

在有 $n$ 个用户的网络中，如果瓶颈带宽为 1，每个用户的吞吐量收敛于 $1/(n+1)$，整个网络吞吐量收敛于 $n/(n+1)$。当网络用户较少时，CADPC 效率较低。CADPC 是针对高带宽时延乘积网络设计的拥塞控制机制，仿真实验表明该机制能保持较高的网络利用率和基本为零的丢包率，具有较快的收敛速度，性能与 RTT 无关，具有较好的公平性。由于 PTP 属于信令协议，CADPC 发送 PTP 获取反馈的速度不是很快(一般是每 4 个 RTT 发送一个 PTP 包)，那么 CADPC 会忽略那些传输数据量小的连接(如 web 数据流)，使其不适应于该类数据流的传输。

**总结** 表 5 比较了各种显式反馈机制所需字节大小、路由和端节点实现的复杂度以及实际部署的难易度。其中需要额外字节最多的是 XCP 和 RCP 机制，CADPC 采用 PTP 信令协议，其发送频率不大，所需的额外字节数比较小。QS、XCP、VCP 和 RCP 将可用带宽测量和分配工作转移到了路由节点，增大了路由节点的复杂性，同时使端系统实现变得简单。ECN 和 AECN 还是一种二元的拥塞指示，在性能上不及 MECN 等细粒度的显式反馈机制，但是部署很简单。

表 5 显式反馈机制比较

| 显式反馈机制 | 路由标记大小 | 路由复杂度 | 端系统实现 | 部署难易度 |
|---|---|---|---|---|
| ECN | 4bit | 简单 | TCP | 易 |
| AECN | 4bit | 简单 | TCP | 易 |
| MECN | 4bit | 较简单 | 简单修改 TCP | 较容易 |
| QS | 8bytes | 复杂 | TCP | 较容易 |
| XCP | 多字节 | 复杂 | 简单 | 较难 |
| VCP | 4bit | 复杂 | 修改 TCP | 较难 |
| RCP | 多字节 | 复杂 | 简单 | 较难 |
| CADPC | 多字节 | 较简单 | 简单 | 较容易 |

现有 TCP 协议已经不适应高速发展的 Internet，显式反馈为 TCP 拥塞控制机制改进和未来新协议设计提供了一个解决方案，但是在实现上还存在很多需要克服的困难。首先，需要权衡反馈信息的粒度问题。粗粒度使信息反馈不足或者不精确，细粒度则需要更多的字节数。其次，需要保证协议有较快的收敛速度，能高效地利用链路带宽，这也是现在研究中的热点，QS、XCP、RCP 和 CADPC 都具有较快的收敛速度。其次，网络是一个复杂的动态系统，需要保证协议的性能不受变化的网络参数(如数据流数目、传输延迟、链路带宽等)的影响，XCP、VCP 和 RCP 在这方面做了初步的探索。其次，在保证同类型协议数据流公平性的同时，还需保证不同类型协议数据流之间的公平性，XCP 等机制能较好的保证同协议数据流的公平性，对于不同协议的数据流的公平性，需要路由节点额外的工作(多队列)。再次，显式反馈面临恶意用户欺骗等安全问题，现有 ECN 已经提出了一些解决办法(如随机数)，对于 XCP 和 RCP 等新协议需要进一步的研究。

高带宽延迟乘积网络的出现使拥塞控制再次成为研究的热点，本文总结了近年来一些典型的研究成果，指出了几个具有价值的研究要点。

## 参 考 文 献


1 Jacobson V. Congestion avoidance and control. In: ACM SIGCOMM '88, 1988

2 Floyd S. HighSpeed TCP for Large Congestion Windows. RFC3649, Dec 2002

3 Low S H, Paganini F, Wang J, et al. Dynamics of tcp/aqm and a scalable control. In: Proc. of IEEE INFOCOM 2002, June 2002


(下转第 36 页)

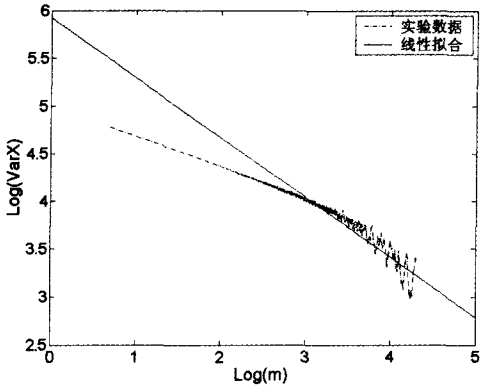


图 11 $H=0.85$ 时，原始方法的估计结果

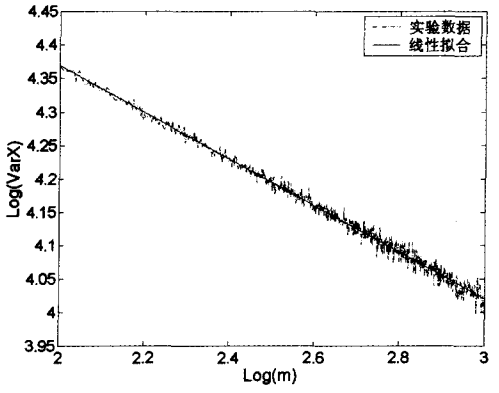


图 12 $H=0.85$ 时，优化方法的估计结果

实验结果总结在表 2。

由表 2 可见，使用经验公式来选择数据块大小的变化范围，可以明显地提高 Hurst 参数估计的准确度，并且也能极大地降低计算复杂度，减少运算时间。

**结论** 本文研究了自相似数据流 Hurst 参数估计的方法，并深入研究了方差时间图法，发现数据块大小的选择范围对估计结果有很大影响。通过深入分析影响原因，本文提出了一个基于数据块选择的估计方法，并给出了选择范围的经验公式。实验证明，该方法可以明显地提高 Hurst 参数估计的准确度，减少运算时间。

## 参 考 文 献


1 Leland W E, Taqqu M S, Willinger W, et al. On the self-similar nature of Ethernet traffic (extended version). IEEE/ACM Transactions on Networking, 1994, 2:1～15

2 Beran J, Sherman R, Taqqu M S, et al. Long-range dependence in variable bit rate video traffic. IEEE Trans. on Communication, 1995, 43(2/3/4), 1566～1579

3 Mandelbrot B, Van Ness J. Fractional Brownian motions, fractional noises and applications. SIAM Rev, 1968,10: 422～437

4 Beran J. Statistics for Long-Memory Processes. New York: Chapman & Hall, 1994

5 Montanari A, Taqqu M S, Teverovsky V. Estmating long-range dependence in the presence of periodicity: an empirical study [J]. Mathematical and Computer Modeling, 1999, 29: 217～228

6 Zhang H F, Shu Y T, Yang O. Estimation of Hurst Parameter by Variance-time Plots. In: Communications, Computers and Signal Processing, 1997. '10 Years PACRIM 1987-1997 Networking the Pacific Rim'. 1997 IEEE Pacific Rim Conference on Volume 2, 20-22 Aug. 1997, 883～886


(上接第 22 页)


4 Kelly T. Scalable TCP: Improving Performance in HighSpeed Wide Area Networks. In: ACM Computer Communications Review, April 2003

5 Xu L, Harfoush K, Rhee I. Binary Increase Congestion Control for Fast Long-Distance Networks. In: Proc. of IEEE INFOCOM 2004, March 2004

6 Rhee I, Xu L. Cubic: a new tcp-friendly high-speed tcp variant. Protocols for Fast Long-distance Networks Workshop, Lyon, France 2005

7 Grieco L A, Mascolo S. Performance evaluation and comparison of westwood+, new reno, and vegas tcp congestion control. SIGCOMM Computer Communication Review, 34(2): 25～38

8 Brakmo L S, O'Malley S W, Peterson L L. TCP Vegas: New techniques for congestion detection and avoidance. ACM SIGCOMM Conference, May 1994. 24～35

9 Jin C, Wei D X, Low S H. FAST TCP: Motivation, Architecture, Algorithms, Performance. In: Proc. of IEEE INFOCOM, Mar 2004

10 Ramakrishnan K, Floyd S, Black D. The Addition of Explicit Congestion Notification to IP RFC 3168. In: Proposed Standard, September 2001

11 Kunniyur S. AntiECN Marking: A Marking Scheme for High Bandwidth Delay Connections. In: Proc. of ICC, May 2003

12 Durresi A, Sridharan M, Liu C, et al. Multilevel Explicit Congestion Notification. presented at SCI2001

13 Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking, August 1993, 1(4): 397～413

14 Kunniyur S, Srikant R. Analysis and design of an adaptive virtual queue algorithm for active queue management. In: Proc. ACM SIGCOMM, 2001

15 Floyd S, Allman M, Jain A, et al. Quick-Start for TCP and IP. Internet-draft draft-ietf-tsvwg-quickstart-07. txt, work in progress, October 2006

16 Katabi D, Handley M, Rohrs C. Congestion control for high bandwidth-delay product networks. In: SIGCOMM'02: Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM Press, 2002. 89～102

17 Xia Y, Subramanian L, Stoica I, et al. One more bit is enough. ACM SIGCOMM Computer Communication Review, October 2005, 35(4): 37～48

18 Dukkipati N, Kobayashi M, Zhang-Shen R, et al. Processor Sharing Flows in the Internet. In: Thirteenth International Workshop on Quality of Service

19 Welzl M. Scalable router aided congestion avoidance for bulk data transfer in high speed networks. PFLDNet 2005 Workshop, Lyon, France

20 Allman M, Paxson V, Stevens W. TCP Congestion Control. RFC 2581, April 1999

21 Allman M, Floyd S, Partridge C. Increasing TCP's Initial Window. RFC 3390, October 2002

22 Welzl M. The performance transparency protocol (ptp). internet-draft draft-welzl-ptp-05, http://www.welzl.at/ptp, June 2001

23 Katz D. IP router alert option. Internet Engineering Task Force, RFC 2113, Feb 1997

24 Hassan M, Jain R. High Performance TCP/IP Networking. Pearson Education International, 2004

25 Braden B. Recommendations on Queue Management and Congestion Avoidance in the Internet. RFC2309, 1998

26 Nagle J. Congestion Control in IP/TCP Internetworks. RFC 896, 1984

27 ATM Forum T. Traffic Management Specification, Version 4.1: [Technical Report AF-TM-0121.000]. The ATM Forum, 1999

28 Welzl M. Network Congestion Control. John Wiley & Sons Ltd, 2005

29 Jiang H, Dovrolis C. Passive Estimation of TCP Round-Trip Times. ACM Computer Communications Review, July 2002, 32(3): 75～88

30 Dukkipati N, McKeown N. Why Flow-Completion Time is the Right Metric for Congestion Control. ACM SIGCOMM Computer Communication Review, January 2006, 36(1)

31 Paxson V, Floyd S. Wide Area Traffic: The Failure of Poisson-Modeling. IEEE/ACM Transactions on Networking, June 1995, 3(3): 226～44