网络拥塞控制算法综述

武航星[1] 慕德俊[1] 潘文平[1] 乔梅梅[2]
(西北工业大学自动化学院 西安 710072)[1] (北京交科公路勘察设计研究院有限公司)[2]

摘 要 随着计算机网络的持续快速发展,各种网络应用需求不断涌现,造成网络数据流量的激增。网络拥塞问题变得越来越严重,网络拥塞控制也一直是网络研究的最关键热点问题之一。在本文中,作者着重阐述了 TCP 拥塞控制和 IP 拥塞控制中的典型算法以及目前一些较有影响的拥塞控制算法,并指出了这些算法的优缺点。最后分析了当前拥塞控制算法设计过程中存在的不足,并给出了一个有意义的研究方向。

关键词 TCP 拥塞控制,IP 拥塞控制,控制理论

The Summary of Congestion Control Algorithm in Network

WU Hang-Xing[1] MU De-Jun[1] PAN Wen-Ping[1] QIAO Mei-Mei[2]
(College of Automation, Northwestern Polytechnical University, Xi'an 710072)[1]
(Beijing Jiaoke Highway Surwaysing, Design and Research Institute)[2]

Abstract With the rapid development of Internet, various applications based on Internet have emerged and the data flows on Internet are increased abruptly. Network congestion has become a serious problem and congestion control has been the focus of attention in the field of network. This paper emphatically describes typical algorithms in TCP and IP congestion control and some current important congestion control algorithms and points out advantages and disadvantages of these algorithms. Subsequently the drawback in designing congestion control algorithm at present is analyzed, and an interesting direction is provided in the end.

Keywords TCP congestion control, IP congestion control, Cybernetics

1 引言

早在 Internet 发展的初期,已经有人认识到了网络中存在拥塞的可能性。在文[1]中,Nagle 提出了纯数据报协议 IP 和传输层协议一起运作时会产生一些不同寻常的拥塞问题。但正如文中所说,由于这两种协议在当时的网络中的普遍应用,这一问题总体上没有被人们所承认。在 1986 年 10 月,计算机网络经历了第一次由于拥塞引起的网络崩溃,在此期间,仅距离 400 码,中间有两个 IMP 跳的 LBL(Lawrence Berkeley Laboratory)到 UC Berkeley 之间的数据吞吐量从 32kbps 降到了 40bps[2]。从此,网络拥塞控制问题引起了研究者们的广泛关注,成为了网络研究中的最关键问题之一。

实质上,网络产生拥塞的最根本原因是端系统提供给网络的负载超出了网络资源的存储和处理能力,文[3]使用图 1 清楚地表述了网络负载与吞吐量和延迟之间的关系。当网络轻载时,随着负载的增加,吞吐量随之迅速增加,延迟增长缓慢;当过了 Knee 点之后,负载增加时吞吐量增加趋于平缓,延迟增长较快;当超过 Cliff 点后负载增加时吞吐量急剧下降,延迟急剧上升。拥塞时网络的具体表现为数据包经受的时延增大,丢弃概率增高。而发送放在遇到大时延时进行的不必要重传会引起路由器利用其链路带宽来转发不必要的分组拷贝。数据包丢弃概率的增加也会引起发送方执行重传因缓存溢出而丢弃的数据包。这些都导致链路传输容量的浪费,有效利用率降低。

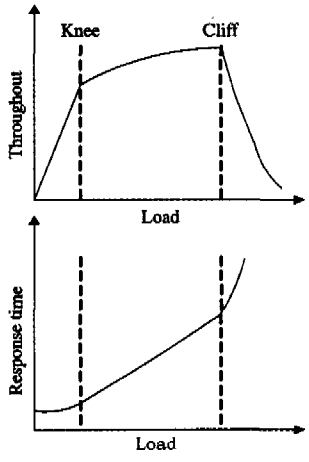

图 1 负载与吞吐量、响应时间关系曲线

网络发展到今天,其应用领域不断拓宽,各种应用模式不断涌现,像音频和视频这样的对网络资源要求较高的多媒体应用更是呈现出爆炸性的增长趋势。而目前的网络资源相对于快速增长的网络应用模式是远远不够的。因此如何使相对有限的网络资源更加高效的利用,尽最大可能满足这些应用需求,避免拥塞崩溃的发生。这正是拥塞控制研究的目的和意义。

在下面的章节中,我们将按网络层次和拥塞控制算法实现的位置,将网络拥塞控制算法大致分为两类:即 TCP 拥塞

武航星 博士生,主要研究方向为网络拥塞控制和流量控制;慕德俊 教授,博士生导师,主要研究领域为并行计算、信息安全;潘文平 博士生,研究方向为网络服务质量;乔梅梅 工程师。

控制和IP拥塞控制，进行详尽的分析。图2显示了TCP拥塞控制和IP拥塞控制的作用范围。

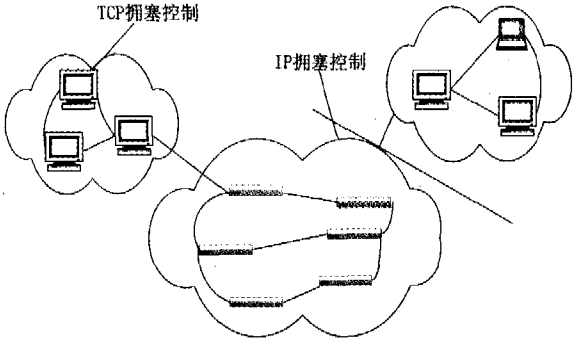

图2 internet中的TCP/IP拥塞控制

本文第2部和第3部详细介绍了TCP和IP拥塞控制中典型的算法思想，并分析和讨论它们的优缺点。第4部分对TCP与IP拥塞控制进行了比较。第5部分介绍了目前几个具有影响的算法。最后对本文进行总结，指出目前拥塞控制算法的设计过程中存在的问题，并据此提出了一个有意义的研究方向。

2 TCP拥塞控制典型算法分析

目前在Internet上实际使用的拥塞控制基本上是建立在TCP窗口控制基础之上的，据统计，Internet上的95%的数据流使用的是TCP协议[4]，因此TCP拥塞控制一直是网络拥塞控制研究的重点。TCP拥塞控制的基本框架是在文[2]中提出的，文中Van Jacobson指出用显式的方法来实现基于窗口的运输层协议会导致端系统对网络拥塞的错误反应，并提出了一系列算法来解决这一问题，这些算法包括：rtt的估计、重传计数器的指数回退、慢启动、拥塞窗口的动态调整等。正是这些算法奠定了TCP拥塞控制的基础。此后的TCP拥塞控制方法基本上都是在此基础上的一些改进。以下是对这些改进中的典型算法的分析：

2.1 TCP Tahoe[2,5,6]

Tahoe算法由3个主要部分组成，加性增与乘性减(AIMD)、慢启动、快速重传。“加性增”用来谨慎地探测端到端路径上的可用带宽，具体表现为TCP发送方在无丢包事件发生时，每收到一个ACK，就将拥塞窗口增加一点，直到丢包事件发生，这个线性增长阶段也称为拥塞避免。“乘性减”方法在丢包事件发生时将当前的拥塞窗口值减半，这样就降低了发送方的发送速率，从而减轻拥塞程度。慢启动是数据发送的初始化阶段，发送方以很慢的速率开始发送，但以指数的速度快速增加其发送速率，从而减小初始阶段因发送方发送窗口过小而造成的带宽浪费。快速重传是当发送方收到3个冗余的ACK而检测到丢包时不必等到重传超时而立即重传丢失的包。这些方法正是TCP拥塞控制的基础，以后的各种改进都依赖于此基础。

2.2 TCP Reno[7,8]

TCP Reno是在Tahoe的基础上增加了“快速恢复”阶段。和Tahoe相比较，Reno在快速重传后并不将拥塞窗口减至1MSS，进入慢启动阶段。而是将拥塞窗口减半，进入拥塞避免阶段。这是因为与发生超时事件不同，收到冗余的ACK表明发送的其它包已经被接收方收到，两个端系统间仍有数据的流动，网络处于轻度拥塞。因此，发送端直接将拥塞窗口减至1MSS是不合适的。但Reno算法也存在缺点，当发送端检测到拥塞后，要重传数据包丢失到检测到丢失时发送的全部数据包，这其中包括已正确传输到接收端，不必重传的包。下面的NewReno和Sack算法正是针对Reno算法的这一缺点的改进版本。

2.3 TCP NewReno[9~11]

NewReno对Reno的改进是通过尽量避免Reno在快速恢复阶段的许多重传超时，利用一个ACK来确定部分发送窗口，立即重传余下的数据包，从而提高网络性能。目前，在Internet中最广泛使用的是NewReno算法。然而NewReno算法也存在着不足，文[12]分析并指出了NewReno在高速远距离网络中不能有效利用带宽的不足，并给出了解决方法。

2.4 TCP Sack[13]

如前所述，Sack算法也是对Reno的改进，当检测到拥塞后，不用重传数据包丢失到检测到丢失时发送的全部数据包，而是对这些数据包进行有选择的确认和重传，从而避免不必要的重传，减少时延，提高网络吞吐量。由于使用选择重传，所以在一个窗口中数据包多包丢失的情况下，Sack性能优于NewReno。但是Sack的主要缺点是要修改接收端TCP。

2.5 TCP Vegas[14,15]

Vegas算法和上面的算法相比有很大的不同，它是通过观察以前的TCP连接中的RTT值的改变情况来控制拥塞窗口的。如果RTT变大，Vegas就认为网络拥塞发生，就减小拥塞窗口；反之，则增大拥塞窗口。这样做的好处是拥塞控制机制的触发只与RTT的改变有关，而与包的具体时延无关。在拥塞避免阶段，拥塞窗口值由下式决定：

$$cwnd(t+\Delta t)=\begin{cases} cwnd(t)+1, diff<\alpha/base\_rtt \\ cwnd(t), \alpha/base\_rtt\leqslant diff\leqslant\beta/base\_rtt \\ cwnd(t)-1, diff>\beta/base\_rtt \end{cases}$$

式中，$diff=\frac{cwnd(t)}{base\_rtt}-\frac{cwnd(t)}{rtt}$；$rtt$为观察到的回路响应时间；$base_rtt$为所观察到的所有$rtt$的最小值；$\alpha$和$\beta$为两个常数。上式表明如果所有数据包的RTT稳定不变，则$cwnd$将不变。然而Vegas不能保证在具有不同RTT的连接之间公平分配带宽，文[16]针对这一问题进行了改进。并且文[17,18]通过仿真试验和分析证实，虽然Vegas在一定程度上提高了吞吐量，降低了丢包率，但是Vegas存在错误估算往返传播时延的可能，从而导致算法性能下降的问题。文[19,20]针对这些问题，对算法提出了一些改进。

表1为这些TCP典型拥塞算法的一个简单比较。

表1 TCP典型拥塞算法的比较

	优点	缺点
TCP Tahoe	建立了TCP拥塞控制的基础，避免了拥塞崩溃的发生	没有快速恢复，轻度拥塞时，拥塞窗口减小过大(减至1)，降低了吞吐量
TCP Reno	增加了快速恢复，轻度拥塞时保持较高的拥塞窗口。	检测到丢包，重传所有丢失与检测到丢事件所有的包。
TCP NewReno	利用一个ACK确认部分发送窗口，避免了过多的重传	在高速网络中不能有效利用带宽。(其实这也是所有现行TCP共同的缺点)
TCP Sack	检测到拥塞时，选择性的重传包，避免了不必要的重传。	要修改TCP接收端，实现复杂。

3 IP 拥塞控制典型算法分析

TCP 基于窗口的拥塞控制机制对于 Internet 的鲁棒性起到了关键性的作用。然而，随着 Internet 本身的迅猛发展，其规模越来越庞大，结构也日趋复杂，研究者们也认识到仅仅依靠 TCP 端到端的拥塞控制是不够的，网络也应该参加资源的控制工作。网络本身要参与到拥塞控制中，已经成为了一个不可回避的问题。现在，IP 层拥塞控制的研究也越来越多，已经形成了一个新的热点研究方向。IP 层拥塞控制就其本质来说是通过对路由器缓冲区队列中的分组实施调度和管理来影响 TCP 拥塞控制的动态性能以达到目的的。已经出现了一系列的队列调度和管理的算法来实现拥塞控制。下面，我们会对其中的一些典型算法进行详细描述。

3.1 先进先出(FIFO)

FIFO 又叫先到先服务(FCFS)，即第一个到达的包将被首先服务。由于路由器的缓存总是有限的，当缓冲区满后，随后到达的包将被丢弃。由于 FIFO 总是丢弃到达队尾的包，所以经常和去尾(Drop-Tail)算法在概念上被混淆。FIFO 是一种包的调度策略，Drop-Tail 是一种包的丢弃策略。由于 FIFO 和 Drop-Tail 实施起来比较简单，因而目前去尾的先入先出是 Internet 上最广泛使用的队列调度管理方式。然而，去尾的 FIFO 自身存在一些问题，例如：它不能区分不同的数据流，也不提供强制数据流遵守拥塞控制的机制，这就有可能让某些数据流强占大量带宽，引发公平性问题。例如 UDP 数据流可以向网络任意发送数据包，从而引起其它数据流的包的丢弃；去尾的 FIFO 也会导致不合理的丢包模式，在文[21]中对此有详细的讨论。在小规模的统计多路复用的情况下，它还存在全局同步问题，引发多个数据发送方同时减小发送速率。

3.2 公平排队(FQ)和加权公平排队(WFQ)[22,23]

针对上述算法存在的第一个问题，FQ 和 WFQ 相继产生。WFQ 是对 FQ 的改进，FQ 可以看作 WFQ 的特例，因为在 WFQ 中，如果取所有权值相同，则 WFQ 又变为 FQ。因此，在本部分，我们只对 WFQ 进行分析。图 3 对 WFQ 算法进行了描述。

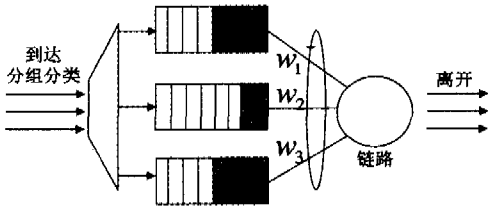

图 3 加权公平排队(WFQ)

到达的分组被进行分类并在合适的等待区域排队，WFQ 调度器以循环方式为各个类提供服务，即首先服务 1 类，接着服务 2 类，再服务 3 类。然后重复这种模式。WFQ 的优点是每个类在固定的时间间隔内都可能收到一定数量的服务。具体而言，在 WFQ 下，对于分配到权值 w_i 的类 i，在有分组要传送的一定时间间隔中，可保证类 i 得到等于 $w_i/(\sum w_j)$ 的服务，式中 $\sum w_j$ 为所有类别权值之和。对于传输速率为 R 的链路，第 i 类可获得 $R\times w_i/(\sum w_i)$ 的吞吐量。WFQ 在目前的一些路由器产品中已经得到了应用[24]。虽然 FQ 和 WFQ 可实现公平性，但是，其缺点是需要在路由器中维护每个数据流的状态，造成路由器的负担，而且也不易实现。

3.3 随机检测算法(RED)[25]

RED(Random Early Detection)算法在 1993 年由 Floyd 和 Jacobson 提出，该算法通过监控路由器中的数据包排队长度，在缓存占满之前，一旦发现拥塞迫近就按一定的概率丢弃进入路由器的数据包，这样就可以及早地通知源端减小拥塞窗口，从而减少进入网络的数据量。这意味着路由器以后不必丢弃更多的数据包，从而提高网络的吞吐量。RED 算法主要包含两部分：如何监控平均队列长度和如何丢弃数据包。平均队列长度是由指数加权滑动平均(EWMA)来计算的：

$$Q_{avg}\leftarrow(1-w_q)\times Q_{avg}+w_q\times q$$

其中，w_q 为权值。q 为采样测量时的队列长度。丢包概率由下式计算：

$$P=\begin{cases}0, & Q_c<Q_{\min}\\ \dfrac{(Q_{avg}-Q_{\min})}{(Q_{\max}-Q_{\min})}\times P_{\max}, & Q_{\min}\leqslant Q_c<Q_{\max}\\ 1, & Q_{\max}\leqslant Q_c\end{cases}$$

其中 $Q_{\min}$ 为最小阈值，$Q_{\max}$ 为最大阈值，$P_{\max}$ 为最大丢弃概率。上式表明当一个数据包到达队列时，如果平均队列长度小于最小阈值，则此包进入队列排队；如果平均队列长度介于最大阈值和最小阈值之间，则根据计算得到的丢弃概率 P 丢弃此包；如果平均队列长度大于最大阈值，则直接丢弃此包。图 4 清晰地显示了 RED 算法的思想。RED 算法通过早期随即丢弃数据包，使得平均队列长度始终处于较低的水平，有利于对短期突发数据流的吸收。而且由于随即丢弃的使用，避免了去尾算法的全局同步问题。然而，RED 算法也存在着缺点，如，算法是参数敏感的，而且其参数配置问题一直没有得到很好的解决。而且其稳定性也存在着问题[26,27]。针对这些问题，此后又出现了一大批的 RED 改进算法。比较著名的如文[28~31]。目前，对 RED 算法的研究和改进依然是一个研究的热点问题。在文[32,33]中分别提出了两种可用于分析不同的 RED 性能的模型。

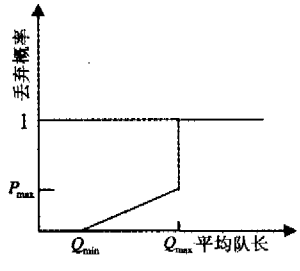

图 4 RED 中丢弃概率与平均队长的关系

3.4 显示拥塞指示算法(ECN)[34]

上面介绍的算法都是用包丢失作为拥塞指示信息，通知端系统网络发生了拥塞，而 ECN(Explicit Congestion Notification)则采用了一种完全不同于它们的方法。ECN 从以前的 DECbit[rj90]算法得到启示，在源端数据包中嵌入 ECN 使能发送比特位，由路由器根据网络的具体情况设置 CE(Congestion Experienced)比特位。源端接收到这种置位的数据包后就认为网络发生拥塞，从而减小发送速率。使用 ECN 的好处是避免了不必要的丢包发生，而且和使用重传超时计数器和三个冗余的 ACK 相比，发送端可以更快速的得到拥塞通告。但是 ECN 也存在两个潜在的问题[35]：第一，对于不兼容 ECN 的连接，会忽略 ECN 通告信息。第二，ECN 信息本身可能被丢弃，使得拥塞通告信息不能到达发送方。目前，关于 ECN 的研究也一直比较多，大多都集中在如何使 TCP 和

ECN更加有效的协作来提高网络性能这一方面。文[36～40]中，可以看到这一方面比较新的一些方法和思路。文[36]分析了不同的ECN机制对TCP性能的影响，并提出了一种新的可提供一定公平性的包标记策略。文[37]提出了一种新的基于ECN的拥塞控制算法，通过测量包的延时来预测网络的拥塞状况，使用ECN对预测的准确性进行确认。文[38]提出了一种基于主动队列管理的ECN机制，通过标记接收方发出的ACK而不是标记发送方发出的数据包来提供拥塞通告信息。文[39]提出了一种基于ECN的结构可变的方法，在RTT和TCP连接数量不定的情况下也性能良好。文[40]对ECN进行了少量的改动，提出了一种预先控制的方法，增强了主动队列管理中的P控制器和PI控制器的稳定性。

表2为这些IP典型拥塞控制算法的一个简单比较。

表2 IP典型拥塞控制算法的比较

	优点	缺点
FIFO	简单易于实现	会导致全局同步和公平性问题
FQ和WFQ	可实现链路的公平分配	需要维护每个数据流的状态，实现较复杂。
RED	早期预测拥塞，平均队列长度小，有利于吸收短期突发数据流	参数确定较难，而且稳定性也存在问题
ECN	通过标记而不是丢包通知拥塞，避免了不必要的丢包	不兼容ECN的连接会忽略通告信息。通告信息本身可能被丢弃。

4 TCP与IP拥塞控制的比较

从以上的分析讨论中可以看出，TCP拥塞控制本质上是一种基于信源的拥塞控制方法，实现在端系统中。显然这种拥塞控制方法在网络拥塞发生到感知到拥塞后采取控制行动之间存在着比较大的延迟，在传输数据不大的情况下，很可能传递拥塞信息的反馈在数据传输完后才到达发送源端。IP拥塞控制由于在网络中实现，因而可以及时感知到网络拥塞的发生，采取控制行为。TCP拥塞控制还存在的一个问题是公平性问题，这一问题出现在TCP数据流和UDP数据流共存时，当拥塞发生时，TCP执行拥塞控制策略减小发送速率，而UDP不会减小发送速率，从而得到更多的带宽。在不同的TCP数据流之间也存在着公平性问题，这是由于不同的TCP连接具有不同的发送窗口和RTT，具有较大发送窗口和较小RTT的连接在拥塞发生时将占有更多的带宽资源。然而在IP拥塞控制中，可以区分不同的发送源端产生的数据流，可以在路由器中通过队列调度方案，接受或丢弃不同发送源端产生的数据，从而实现带宽的公平使用。对于短期的拥塞，在IP拥塞控制来处理比较好，但是对于长期的拥塞，IP拥塞控制就显得无能为力，拥塞将通过临近的路由器和链路在网络中扩展。而TCP拥塞控制却可以经受长期的拥塞。IP拥塞控制还存在的一个问题是增加了路由器的复杂性。表3是TCP与IP拥塞控制的一个简单比较。

表3 TCP与IP拥塞控制的比较

	TCP拥塞控制	IP拥塞控制
实现位置	端系统中	网络内部
延迟	较大	无
不同数据流间的公平性	难于实现	可以实现
长期拥塞	可以处理	无法处理
短期拥塞	可以处理	较好处理

5 其他一些有影响的拥塞控制算法

基于方程的拥塞控制(equation-based congestion control)[41～43]，这一算法首次出现在文[44]，它的主要目的是使可用带宽不被侵略性的发现和使用，在响应拥塞的同时保持相对稳定的发送速率。它把发送速率作为丢失事件率的函数，并使用控制方程显示得给出可接收的最大发送速率。因此丢失事件率的计算是该算法的一个关键部分。但是在文[45]中，指出了基于方程的拥塞控制算法中，TFRC流和TCP流会具有不同的平均发送速率，而这些流具有的不同发送速率会导致各个流丢包率的不同。同时文[45]中也分析了产生这一问题的原因。

适应性虚拟队列(AVQ)[46,47]：AVQ(Adaptive Virtual Queue)的目的是设计一个低时延、低丢包率、高连接利用率的AQM(Active Queue Management)机制。AVQ算法维护一个容量小于实际队列容量的虚拟队列，当实际队列中有数据包到达时，虚拟队列也做出相应更新以反映数据包的到达。当虚拟缓冲队列溢出时实际队列表现为标记或丢弃数据包。通过虚拟队列容量的调整来保证连接的预期利用率。AVQ的一个很好的特性是在没有反馈延迟时，在使连接利用最大化的同时能够保证所有端系统间的公平性。然而AVQ算法的不足是实现起来比较复杂，参数选取比较困难。在文[48]中，给出了较为系统的规则来调整AVQ算法的控制参数。

快速TCP系统(FastTCP)[49]：Fast Tcp(Fast Active Queue Management Scalable TCP)是加州理工大学正在开发的一种新的协议。Fast Tcp对发送端的软硬件进行修改，可连续监测发送分组往返时延(RTT)，可预测不丢失数据的情况下连接所能支持的最大数据传输速率。Fast TCP的拥塞控制算法分为两部分，这两部分分别在源端和路由器中设计和执行。路由器不断将检测到的拥塞信息反馈给发送端，发送端根据得到的拥塞信息调整发送率。Fast TCP可在目前的互联网基础设施上运行，可使带宽利用率达到90%。然而在文[50]中指出，FastTCP由于不能精确估计分组RTT可能会致不公平性问题和路由器队列的振荡。在文中同时也给出了一种改进的方法

TCP Westwood[51]：TCP Westwood是一种由发送方采用新的方法来调整拥塞窗口的TCP协议。其主要目的是在少量丢包时对拥塞窗口进行少量的减少而不是拥塞窗口减半来提高连接吞吐量。它通过估算的端到端可用带宽来调整拥塞窗口的大小，而且对端到端带宽的估计无需接收方和中间路由器的参与。TCP Westwood算法主要有两个部分：即可用带宽的估算和拥塞窗口、慢启动阈值的调整。TCP Westwood的主要创新点在于通过检测接收到的ACK，在发送方估算该连接的可用带宽，并在拥塞事件发生时据此来调节发送方的拥塞窗口和慢启动阈值的大小。其调整拥塞窗口的伪代码如下，当收到3个冗余的ACK时：

```
if (3 DUPACKs are recieved)
    ssthresh=BWE×RTT_min/seg_size;
    if(cwin>ssthresh)/*拥塞避免*/
      cwin=ssthtresh;
    endif
endif
```

当发生超时事件时：

```
if( timeout expires )
    cwin=1;
    ssthresh=BWE×RTT_min/seg_size;
    if(ssthresh<2)
        ssthres=2;
    endif
endif
```

其中 $ssthresh$ 为慢启动阈值，$cwind$ 为拥塞窗口大小，BWE 为估算的连接带宽，RTT_{min} 为测到的 RTT 最小值，seg_size 为数据包的大小。仿真结果显示，TCP WestWood 性能优于 TCPReno，可获得较高的吞吐量。但是，该算法也存在一定的问题，在文[52]中，指出了 WestTCP 存在的可能过高估计带宽而导致某些连接饿死的问题，并提出了一种解决方法。

小结 从以上的 TCP 和 IP 拥塞控制中典型算法的分析介绍中，不难看出，这些算法虽然在以往算法的基础上进行了改进，系统性能有所改善，但其自身几乎都存在着这样或那样的问题。其直接原因就是这些算法的设计都是针对局部的某一具体问题进行，依靠直觉的推断，根据经验改进算法，缺乏一套有效的系统的理论分析工具对算法的设计进行指导。控制理论作为一门相当成熟的系统理论，有相当多的方法可以借鉴到拥塞控制中来。近来，国内外的很多专家学者都认识到了可以应用控制理论的方法来解决 Internet 中的拥塞控制问题，并进行了一些尝试工作[53,54]。然而，由于 Internet 本身是一个复杂的巨系统，而且其中的各种不同控制策略之间也会互相影响，使得对网络稳定性和动态性能的分析更加困难，因而这方面的研究还不成熟。所以，如何有效的将控制理论的思想运用于日趋复杂的 Internet 中，来指导目前单纯根据经验来改进算法的不足，将是一个未来研究的热点问题，也是一个难点问题。

参 考 文 献

1 Nagle J. Congestion Control in IP/TCP Internetworks. RFC896, 1984

2 Jacobson V. Congestion Avoidance and Control. In: Proceeding of SIGCOMM '88, August 1988. 314～329

3 Jain R, Ramakrishnan K K, Chiu Dah-Ming. Congestion Avoidance in Computer Networks with a Connectionless Network Layer: [Technical Report]. Digital Equipment Corporation DEC-TR-506, 1988. http://www.cis.ohio-state.edu/～jain

4 Caserri C, Meo M. A new approach to model the stationary behavior of TCP connections. In: Proceedings of IEEE INFOCOM 2000, Tel-Aviv, Israel, CA, 2000. 367～375

5 Hoe J. Improving the start-up behavior of a congestion control scheme for TCP. In: Proc. SIGCOM Symposium on Communications Architectures and Protocols, 1996

6 Fall K, Floyd S. Simulation-based comparisons of Tahoe, Reno, and SACK TCP. In: Proc. IEEE INFOCOM 2000, Tel aviv, Israel, CA: IEEE Computer Society, 2000

7 Stevens W. TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms. RFC 2001, 1997

8 Jacobson V. Modified TCP Congestion Avoidance Algorithm: [Technical Report]. http://ftp.ee.lbl.gov/

9 Floyd S, Henderson T. The NewReno modification to TCP's Fast recovery algorithm. RFC 2582. 1999

10 Clark D, Hoe J. Start-up Dynamics of TCP's congestion control and avoidance schemes: [MS dissertation]. Presentation to Internet End-to-End research Group: [Technical Report]. 1995

11 Grieco L A, Mascolo S. Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control. Computer Communication Review, 2004, 34 (2): 25～38

12 Song B, Chung K S, Rhee S H. A new TCP congestion control for high-speed long-distance networks. Information networking lecture notes in Computer Science, 2004, 3090: 606～615

13 Matthew W, Jamshid W, Sally F, et al. TCP selective acknowledgement options. RFC 2081, 1996

14 Brakmo L S, Peterson L. TCP vegas: End-to-End congestion avoidance on a global Internet. IEEE Journal on Selected Areas in Communications, 1995, 13(8): 1465～1480

15 Brakino L S. TCP Vegas: New techniques for congestion detection&avoidance. In: Proc. SIGCOM'99, 1999

16 Byun H J, Lim J T. On a fair congestion control scheme for TCP vegas. IEEE Communtcations Letters FEB 2005, 9 (2): 190～192

17 Low S, Peterson L, Wang L. Understanding TCP Vegas: A Duality Model. In: Proc SIGMETRICS'01, June 2001

18 Mo J, La R, Anantharam V, et al. Analysis and Comparison of TCP Reno and Vegas. In: Proc. INFOCOM'99, Mar. 1999

19 Hengartner U, Bolloger J, Gross T. TCP Vegas Revised. In: Proc. INFOCOM'00, March 2000

20 Srijith K N, Jacob L, Ananda A L. TCP Vegas-A: Improving the performance of TCP Vegas, Computer Cmmunications, 2005, 28(4): 429～440

21 Floyd S, Jacobson V. On Traffic Phase Effects in Packet-Swithched. Gateways. 1992, 3(3): 115～156

22 Demers A, Keshav S, Shenker S. Analysis and Simulation of a Fair Queue Algorithm. Internetworking: Research and Experience, 1990, 1(1): s-26

23 Parekh A, Gallager R. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. IEEE/ACM Transactions on Networking, 1993, 1(3)

24 Cisco Systems Inc. Advanced QoS services for the Intelligent Internet. http://www.cisco.com/warp/public/cc/pd/iosw/ioft/ioqo/tech/qos_wp.htm

25 Floyd S, Jacobson V. Random Early Detection Gateways for Congestion Avoidance. ACM/IEEE Transactions on Networking, 1993, 1(4): 397～413

26 Christiansen M, Jeffay K, Ott D, et al. Tuning RED for Web traffic. In: Proceedings of ACM SIGCOM 2000, Stockholm, Sweden, 2000. 139～150

27 Firoiu V, Borden M. A study of active queue management for congestion control. In: Proceedings of INFOCOM 2000, Tel Aviv, Isarel, 2000. 1435～1444

28 Cisco Company. Distributed weighted random early detection. http://cco.cisco.com

29 Ott T J, Lakshman T V, Wong L H. SRED: stabilized RED. In: of INFOCOM 1999, New York, USA, 1999. 1346～1355

30 Floyd S, GummadiR, Shenker S. Adaptive RED: An algorithm for increasing the robustness of RED's active queue management. http://www.cs.berkeley.edu/

31 Anjum F, Tassiulas L. Balanced-RED: An algorithm to achieve fairness in Internet. In: Proceedings of IEEE INFOCOM 1999, New York, USA, 1999

32 Guan L, Awan I U, Woodward M E. Stochastic modelling of

random early detection based congestion control mechanism for bursty and correlated traffic. IEEE Proceedings-Soetware, 2004, 151 (5): 240~247

33 Asfand-E-Yar, Awan I, Woodward M E. RED based congestion control mechanism for Internet traffic at routers. Information networking: Convergence in Brodband and Mobile Networking Lecture Notes in Computer Science, 2005, 3391: 142~151

34 Ramakrishnan K K, Floyd S. A Proposal to Add Explicit Congestion Notification (ECN)to IP, RFC 2481, Jan. 1999

35 Pentikousis K, Badr H. An evaluation of TCP with explicit congestion notification. Annales des Telecommunications-Annals oetelecommuntcations, 2004, 59 (1-2): 170~198

36 Zabir S M S, Ashir A, Shiratori N. An efficient approach to performance improvement of different TCP enhancements using ECN. IEICE Transactions on Informaytion and Systems, 2002, E85D (8): 1250~1257

37 Liu H S, Xu K, Xu M W. A novel ECN-based congestion control and avoidance algorithm with forecasting and verifying. Telecommunications and Ndetworking-ICT 2004 Lecture Notes in Computer Science, 2004, 3124: 199~206

38 Matsuda T, Nagata A, Yamamoto M. Active ECN mechanism for fairness among TCP sessions with different round trip times. IEICE Transactions on Communications, 2004, E87B (10): 2931~2938

39 Yan P, Gao Y, Ozbay H. A variable structure control approach to active queue management for TCP with ECN. IEEE Transactions on Control Systems Technology, 2005, 13 (2): 203~215

40 Zhu R J, Teng H T, Hu W L. A predictive controller for AQM router supporting TCP with ECN. Content Computing Proceedings Lecture Notes in Computer Science, 2004, 3309: 131~136

41 Floyd S, Handley M, Padhye J, et al. Equation-based Congestion Control for Unicast Applications. February 2000. http://www.aciri.org/tfrc/

42 Floyd S, Handley M, Padhye J. A Comparison of Equation-Based and Aimd Congestion Control. February 2000. http://www.aciri.org/tfrc/

43 Ng S W, Chan E. Equation-based TCP-friendly congestion control under lossy environment. Journal of Systems Archirecture, 2005, 51 (9): 542~569

44 Mahdavi J, Floyd S. TCP-friendly Unicast Rate-based Flow Control. Note sent to end2end-interest mailing list, Jan. 1997

45 Rhee I, Xu L S. Limitations of equation-based congestion control. Cmputer Communicatton Review, 2005, 35 (4): 49~60

46 Kunniyur S, Srikant R. Analysis and design of an adaptive virtual queue algorithm for active queue management. In: Proceedings of ACM SIGCOM 2001, San Diego, CA, USA, 2001

47 Kunniyur S S, Srikant R. An Adaptive Virtual Queue (AVQ)algorithm for Active Queue Management. IEEE-ACM Transactions on Networking, 2004, 12 (2): 286~299

48 Tan L S, Yang Y, Lin C. Scalable parameter tuning for AVQ. IEEE Communications Letters, 2005, 9 (1): 90~92

49 Jin C, Wei D, Low S H. FAST TCP: From theory to experiments. IEEE Network, 2005, 19 (1): 4~11

50 Tan L S, Yuan C, Zukerman M. FAST TCP: Fairness and queuing issues. IEEE Commjinications Letters, 2005, 9 (8): 762~764

51 Casetti C, Gerla M, Mascolo S. TCP westwood: End-to-end congestion control for wired/wireless networks. Wirelfss Networks, 2002, 8 (5): 467~479

52 Grieco L A, Mascolo S. End-to-end bandwidth estimation for congestion control in packet networks. Quality of Service in Multiservice IP Networks. Proceedings Lecture Nots in Computer Science, 2003, 2601: 645~658

53 Shor M H, Lik W J. Application of control theory of modeling and analysis computer system. http://www.cse.ogi.edu/~kangli/

54 Habibipou F, Khajepour M, Galily M. Application of control engineering methods to congestion control in differentiated service networks. Control Engineering Practice, 2006, 14 (4): 425~435

2007 年中国计算机大会(CNCC 2007)
征文通知

2007 中国计算机大会(2007 China National Computer Conference, CNCC 2007)由中国计算机学会、苏州市人民政府主办，苏州市科学技术协会承办，将于 2007 年 10 月 18 日至 20 日在苏州举行。它将为我国计算机界提供一个交流最新研究成果的舞台。CNCC 2007 是继 CNCC2003, CNCC2005 和 CNCC2006 之后的中国计算机界又一次盛会。

CNCC 2007 的议题涉及计算机领域各个方面。本次大会将安排大会特邀报告、专题报告、企业专题论坛和热点问题讨论，同时将举办有关 IT 技术的展览。

本届大会将举办一系列展览，欢迎海内外企业、出版社、高校和研究所来参展。参展主题不限，可以是企业产品、出版物、高校和研究所研究成果以及组织形象等。

CNCC 2007 诚请广大计算机界研究人员、技术人员以及其他相关人士投稿。会议的议题主要包括(但不限于此)：

高性能计算机；高性能计算机评测；传感器网络；嵌入式系统；对等计算；生物信息学；网格计算；网络存储系统；编译系统；虚拟现实；多核处理器；人工智能；理论计算机科学；软件工程；多媒体技术；信息安全技术；普适计算；数据库技术；搜索引擎技术；图形学与人机交互；中文处理；互联网络；模式识别；计算机应用技术。

投稿须知

作者投往本届大会的稿件必须是原始的、未发表的研究成果、研究经验或工作突破性进展报告。稿件须以中文撰写，以 word 文件格式提交。所有稿件将依据统一的原则进行审理，然后大会根据稿件的审理结果决定稿件是否录用。所有录用稿件将收录在本届大会论文集中。此外，本届大会的优秀稿件将推荐在《计算机学报》、《软件学报》、《计算机研究与发展》上发表。

重要日期：征稿截止 2007 年 7 月 30 日　论文处理结果通知 2007 年 8 月 30 日

详细情况见 http: ccf. org. cn/cncc2007　或者　www. jsjkx. com