

# 基于队列的模糊拥塞控制算法\*

蔡朝晖<sup>+</sup>, 孙济洲, 李小图

(天津大学 计算机科学与技术系, 天津 300072)

## A Fuzzy Congestion Control Algorithm Based on Queue

QI Zhao-Hui<sup>+</sup>, SUN Ji-Zhou, LI Xiao-Tu

(Department of Computer Science and Technology, Tianjin University, Tianjin 300072, China)

+ Corresponding author: Phn: +86-22-27402621, Fax: +86-22-27402621, E-mail: qizhaohui@eyou.com, <http://www.tju.edu.cn>

Received 2003-06-05; Accepted 2003-12-31

Qi ZH, Sun JZ, Li XT. A fuzzy congestion control algorithm based queue. *Journal of Software*, 2005,16(2):286-294. <http://www.jos.org.cn/1000-9825/16/286.htm>

**Abstract:** Internet applications are developed rapidly. It is increasingly important for router itself to improve the ability to deal with networking congestion. Traditional Poisson model is unfit for Internet networks with burst flow. But self-similarity model suitable for Internet networks has not been used widely in practice because of its complex model and complicated calculation. By describing the practical buffer performance in routers, a new fuzzy congestion control model based on queues and a congestion control algorithm based on the model are presented. In the algorithm, all kinds of packets are firstly classified into queues according to their own priorities. Then the buffer state is divided into three phases, including normal, congestion avoidance, and congestion according to their buffer usage ratio. The three phases are crossover each other because of their fuzziness. Then by combining the whole congestion control, with the part congestion control, the fuzzy algorithm is carried out. Theoretical analysis and NS stimulation results show that the proposed algorithm has better networking performance in the fairness of all connections, compared with the traditional schemes, especially keeping from being affected by the connections with congestion. It really improves the routers' ability to deal with network congestion.

**Key words:** congestion control; buffer; queue; fuzzy theory; state matrix

**摘要:** 传统的 Poisson 统计流量模型对于以突发性流量为基本特征的 Internet 网络不再适应,而采用更加接近 Internet 网络流量特征的自相似模型,会具有复杂的建模过程和繁杂的计算.为此,从数据缓冲区占用情况的实时状态出发,运用模糊理论对缓冲区占用率状态这一模糊性问题进行描述,建立起模糊拥塞控制模型,并实现了对拥塞的模糊控制.它对所有到达的数据流按照一定的优先级进行分类,并把全局性缓冲区和各队列的局部性缓冲区按照正常、拥塞避免和拥塞的规则划分为 3 个具有交叉过渡域的几个阶段,然后采用整体和局部相结合的拥塞控制方法,

---

\* Supported by the National High-Tech Research and Development Plan of China under Grant No.2002AA142010 (国家高技术研究发展计划(863))

**作者简介:** 蔡朝晖(1976-),男,湖南衡阳人,博士,主要研究领域为计算机网络,分布式系统;孙济洲(1949-),男,教授,博士生导师,主要研究领域为并行计算,计算机网络与分布式系统,计算机图形图像;李小图(1977-),男,博士,主要研究领域为并行计算与分布式系统.

实现了队列调度过程中的模糊性处理.理论分析和 NS 实验仿真结果表明,该算法在保证各连接服务的公平性方面,特别是在保护和隔离非拥塞状态的连接上,取得了比传统方法更好的效果,从而更好地改进了路由器或者交换机的拥塞控制性能.

**关键词:** 拥塞控制;缓冲区;队列;模糊理论;状态矩阵

**中图分类号:** TP393      **文献标识码:** A

Internet 提供实时服务,其运行环境在不断地发生变化.基于 TCP/IP 协议栈的网络通信协议是 Internet 中的主流通信协议.传统的基于 TCP/IP 的 Best-effort 模型是网络中最为典型的服务模型,它采用端到端的拥塞控制机制<sup>[1-3]</sup>,其中路由器或者交换机采用先来先服务 FIFS(first in first serve)调度算法和尾部丢弃 TD(tail drop)策略来处理网络拥塞.然而,随着网络信息量的急剧增长,这种端到端的拥塞控制机制和数据包的调度模式不再有效,作为网络中的核心转发设备——路由器或者交换机必须要加强自身的拥塞处理能力.

从源端出发的数据包要经过一系列的路由器或者交换机才能够到达目的端.而路由器或者交换机的最为核心的功能就是完成数据包的转发,并将来不及转发的数据包,缓存于数据缓冲区,形成队列.因此,队列的调度与缓冲区的管理策略将直接关系到路由器转发调度质量的好坏.一个较为理想的队列调度算法将能够使得在共享同一端口的所有数据流达到最大的公平性.当前,关于这一领域的研究工作已经取得了一定的成果,但是,它们在实际的网络应用中关于公平性的实现一直得不到令人满意的结果.

尾部丢弃 TD-FIFO(tail drop first in first out)算法基于 FIFO 策略进行数据包调度<sup>[4]</sup>,以先来先服务的方式服务于所有到达的数据包流,当发生拥塞时,简单地按照同样的概率从队列尾部丢弃数据包,以减少队列的长度,这种算法尽管简单,但由于对所有类型的数据包都以同样的方式进行转发和丢包,使得在隔离和保护那些未发生拥塞的连接上,其性能不能令人满意.与尾部丢弃策略相比,随机提前检测 RED(random early detection)算法<sup>[5,6]</sup>,加入了两个新的机制进行队列的管理.它一方面经过早期检测预防拥塞的可能发生,另一方面通过平均队列的长度来调整数据包丢弃的概率,从而尽可能地平滑了短暂突发信息流所带来的不公平性.当一个数据包到达时,如果队列的加权平均长度  $L$  低于最小阈值  $\min th$ ,它就简单地进入队列.若  $L \in [\min th, \max th]$ ,则早期检测开始启动,并按照一定的概率对数据包进行丢弃.然而,某一高带宽流会由于比低带宽流有更快的数据包流而被丢弃更多的数据包,甚至一些不在拥塞状态的连接也因此而丢包,这显然是不公平的.此外,在 RED 算法运行的过程中,门限参数的配置也一直得不到好的解决.在公平性的实现上,基于轮询 RR(round robin)策略的加权公平队列 WFQ(weight fair queuing)算法<sup>[7-9]</sup>,有了一定程度的改进,它首先对所有到达的数据包进行分类,并使待发的数据包进入相应的队列缓冲区排队等待,各队列的数据包转发则以轮询的方式进行.这种算法在网络载荷较轻时,调度性能的公平性较好.但由于在算法当中没有针对在队列发生拥塞时的处理策略,其拥塞控制性能的公平性不能令人满意.

我们认为,运行于网络核心转发设备——路由器或者交换机中的队列调度算法,必须要能够区分不同类型的网络连接,如一些普通的 FTP 文件传输、HTTP 网页浏览文件等,用户对它们的传输性能要求不高;而其他一些基于 QoS<sup>[10,11]</sup>的高质量服务,如 IP 电话、视频会议、多媒体传输等,根本就不允许超时或者重发.因而在路由器中进行这些数据包的转发和调度时,要依据其服务质量要求有区别地进行处理.通常情况下,在网络正常时,任何连接的服务质量都是能够得到保证的.而当网络负荷繁重时,网络的整体服务性能就会变坏,或者是某些连接发生了局部拥塞,此时,路由器就必须采取有效的控制办法,对所有到达的数据包流进行公平性的调度.特别是在共享链路带宽、处理丢包时,必须要体现出公平性的原则,处理好真正处于拥塞状态的数据包流,有效地隔离和保护好未发生拥塞的连接.

本文所提出的基于队列的模糊拥塞控制 FCCA(fuzzy congestion control algorithm based queue)算法,从数据缓冲区占用情况的实时状态出发,运用模糊理论对缓冲区占用率状态这一模糊性问题进行描述,建立起模糊拥塞控制模型,并采用整体和局部相结合的方法,对单个发生拥塞的数据流以一定的概率进行惩罚性的丢包,并在全局发生拥塞时(网络负荷严重),进行全局性的随机丢包.从而使得路由器或者交换机在进行拥塞处理的过程中,智能地管理缓冲区,使真正处于拥塞状态的队列进行数据包丢弃,有效地隔离和保护了未发生拥塞的连接.

实验仿真结果表明,本文所提出的基于模糊理论的拥塞控制算法具有更好的拥塞控制性能.

### 1 基于队列的模糊拥塞控制模型

对于网络中到达路由器或者交换机同一转发端口的数据包流,通常来自于不同的源端,如果都对它们都采取同样的调度策略,那么,一些需要高速服务要求的数据包流会因此得不到应有的服务,而一些低速的数据包流会由于拥塞而占用更多宝贵的缓冲区资源.因此,在本文的拥塞控制算法中,假设路由器已经根据网络的 QoS 要求对所有的数据包进行了分类.这样,需要由路由器某接口转发的所有数据包流都已经按照一定的分类标准被分类.

但是,由于网络中多种业务的流量突发,TCP 流量控制机制以及 Internet 大量用户源节点的汇集效应等,都会使网络中的业务负荷强度发生动态变化.因此,我们首先让每一类型的数据包流经一个令牌桶,经过整形后再进入相应队列的缓冲区等待转发,以减少流量的突发性特征给我们的拥塞控制带来大的负面影响.图 1 是一个改进的流量整形的抽象模型.

我们让每一类型的数据包流经一个令牌桶,允许突发性通信量,但对于流量的突发性时间  $S$  的长度要进行限制.现在假设突发时间长度为  $S$  秒,令牌桶容量为  $C$  字节,令牌到达速率为  $\rho$  字节/秒,而最大输出数据流为  $M$  字节/秒.于是,可以得到突发通信量持续时间  $S = C / (M - \rho)$ .我们认为,如果某一类数据包流突发通信量持续时间超过  $S$ ,则认为可能发生拥塞,此时进入该队列缓冲区的输入数据包流的速度  $V = M + v$ ,其中  $v$  为令牌桶整形模型中数据包缓冲区溢出速率.

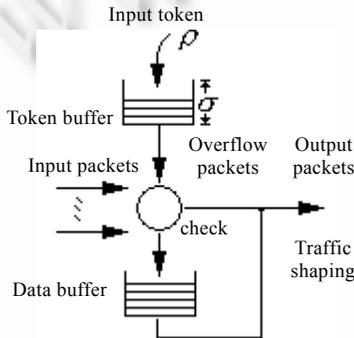


Fig.1 Abstract model of traffic shaping

图 1 流量整形的抽象模型

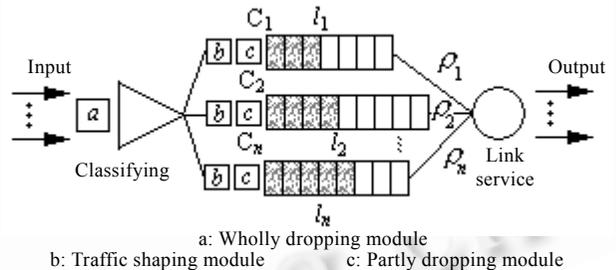


Fig.2 Fuzzy congestion control algorithm model

图 2 模糊拥塞控制算法模型

图 2 给出了结合令牌桶整形的拥塞控制算法的队列调度模型.

其中,全局丢包模块在全局缓冲区占用率达到全局阈值时,进行全局性的随机丢包;而每种类型队列之前,数据包流首先经过一个令牌桶整形模块,对突发性流量进行整形,然后是局部丢包模块,它在该队列缓冲区占用率达到相应的局部阈值,且发生全局性拥塞避免或全局性拥塞时,对该队列进行局部丢包,也就是在网络性能变坏时,对过多占用缓冲区的长队列实行惩罚性丢包.下标  $n$  为当前的队列总数. $C_j$  为  $j$  类队列缓冲区的大小. $l_j$  表示相应队列的实时长度,并定义  $u_j = l_j / C_j$  为该队列的缓冲区占用率.对于不同队列数据包的转发,则采用轮询<sup>[12]</sup>的方式进行,如果某一队列在轮到它时,没有待发的数据包,则轮到下一队列转发,并且在某一队列转发的过程中不允许中断,如果某一队列在属于它的服务时间段到达之前为空,同样也将轮到下一队列转发.而对于同一队列的数据包,则采取先入先出(FIFO)策略进行转发.

在图 2 所提出的模糊拥塞控制算法模型中,各类型队列的优先级权值  $\rho_j$  表示相应队列的优先级别.前面提到,各种不同类型的网络数据包的服务质量要求是不一样的,那么,我们则按照不同的服务质量要求对所有到达同一转发端口的数据包进行分类,如把属于普通的 HTTP 网页浏览类数据流归为最低的优先级,实时语音类数据流归为高一级的级别,实时图像类数据流归为更高的级别,实时图像与语音同步传输的数据流为再高一级的

级别,如此就可以把所有到达的数据包流进行优先级别的分类.然后,再赋予最低优先级队列的优先级权值  $\rho_1$  的值为 1,再高一级的优先级权值  $\rho_2$  为 2,依次直到最高级别  $\rho_n$  的值为  $n$ ,而对于相同级别的数据流则都要进入到同一类型队列进行转发.

如果共享出口的链路带宽为  $W$ ,那么,某一级别类型的队列  $j$  就可以获得  $\left(k_j \rho_j / \sum_{j=1}^n k_j \rho_j\right) W$  部分的带宽,其中  $k_j$  表示同一类型队列  $j$  中不同数据流的数量,从而保证相应级别的数据流获得了相应质量的服务.

## 2 基于队列的模糊拥塞控制算法

### 2.1 队列实时状态的描述

在网络中,由于多种业务的流量突发、TCP 流量控制机制以及 Internet 大量用户源节点的汇集效应等,都会使网络中的业务负荷强度发生动态变化.随着队列所对应的各个源端业务负荷强度的不断变化,相应队列的缓冲区实时占用率也在不断地发生动态的变化.图 3 是我们在实验过程中所测量到的某一队列缓冲区实时占用率的变化状态图,从中不难看出,随着网络负载的不断变化,占用率也在不断地发生动态的变化.

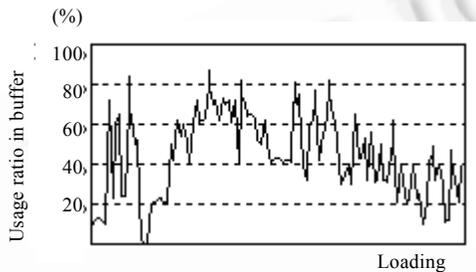


Fig.3 Realtime usage ratio in queue buffer

图 3 队列缓冲区实时占用率

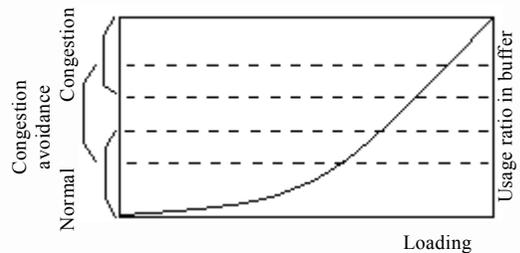


Fig.4 The dividing of realtime stages in queue buffer

图 4 队列实时阶段的划分

为了方便描述,我们把这一不断变化的过程抽象成一个网络载荷不断递增的过程,如图 4 所示,并把队列的实时状态划分为正常、拥塞避免和拥塞 3 个阶段.由于各个阶段的描述本身就是一个模糊性的概念,因此,在图 4 中,把缓冲区占用率按照不同的状态区间定义成相应的阶段.从图 4 中不难看出,随着载荷的不断增加,队列从正常状态,过渡到拥塞避免,再过渡到拥塞.

### 2.2 局部队列与全局性拥塞程度的评价

由于拥塞这一概念本身的模糊性,我们不可能用精确的数值来描述某一队列或者是全局的具体状态.只能说它们处于某一阶段的程度.比如说,某一队列当前的缓冲区占用率为 80%,那么我们会认为该队列当前发生拥塞的可能性极大,有可能还处于拥塞避免阶段,不可能正常.同样,对于全局缓冲区的占用情况也是一样的.因此,我们引入相关的模糊理论来描述这一模糊性的过程,充分利用模糊理论在处理不确定性问题上的优越性,以达到优化的拥塞控制性能.

我们根据 Zadeh 对模糊子集的定义,定义论域  $U$ :缓冲区占用率(可以是队列缓冲区或全局缓冲区占用率).取  $U$  到闭区间  $[0,1]$  上的 3 个映射  $\mu_{\tilde{L}}: U \rightarrow [0,1], u \rightarrow \mu_{\tilde{L}}(u); \mu_{\tilde{K}}: U \rightarrow [0,1], u \rightarrow \mu_{\tilde{K}}(u); \mu_{\tilde{C}}: U \rightarrow [0,1], u \rightarrow \mu_{\tilde{C}}(u)$  确定出  $U$  的 3 个模糊子集:  $\tilde{L}, \tilde{K}, \tilde{C}$ .  $\mu_{\tilde{L}}(u), \mu_{\tilde{K}}(u), \mu_{\tilde{C}}(u)$  分别称为  $u$  对于  $\tilde{L}, \tilde{K}, \tilde{C}$  的隶属函数.其中,模糊子集  $\tilde{L}, \tilde{K}, \tilde{C}$  分别表示图 4 中所描述的 3 个阶段(同样也适用于全局缓冲区占用率情况),即:正常、拥塞避免和拥塞 3 个阶段.

关于隶属函数的确定,带有一定的主观性,不同的人对于同一问题可能会建立起不同的函数形式,但只要能够反映客观事实,就认为是正确的.我们根据文献[14]中的模糊统计方法,确定出队列缓冲区占用率论域  $U$  上的 3 个隶属函数  $\mu_{\tilde{L}}(u), \mu_{\tilde{K}}(u), \mu_{\tilde{C}}(u)$ ,如图 5 所示.为了方面描述,我们将经过模糊统计方法得到的隶属函数进

行简化,如图 6 所示.

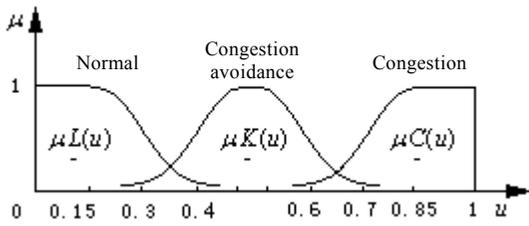


Fig.5 Subjection function

图 5 隶属函数

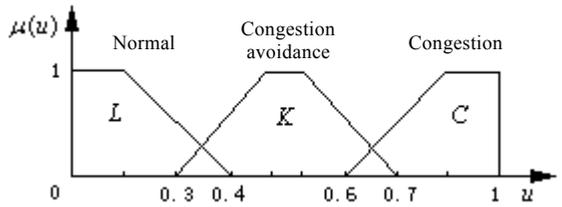


Fig.6 Simplifying subjection function

图 6 简化的隶属函数

队列的缓冲区占用率被划分为 3 个语言变量 {Less Congestion, Knee Congestion, Cliff Congestion}. 其划分的具体意义: Less Congestion (简称为  $L$ ), 表示该队列的缓冲区占用率处于正常状态; Knee Congestion (简称为  $K$ ), 表示该队列的缓冲区占用率处于拥塞避免阶段; Cliff Congestion (简称为  $C$ ), 表示该队列的缓冲区占用率就要达到饱和, 处于拥塞状态. 我们用决策集  $V$  来描述它们:  $V = \{L, K, C\}$ , 用来评价队列所处的实时状态.

上述是对拥塞程度的描述为单个队列的情形. 同样, 这一过程的描述也能适应于整个队列的全局性缓冲区的占用情况.

### 2.3 局部队列与全局性拥塞处理

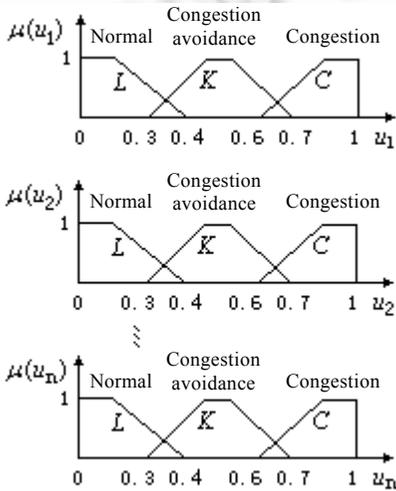


Fig.7 Subjection function chart

图 7 隶属函数分布图

首先, 我们来考察各类型队列的局部状态. 输入模糊变量: 队列 1 的缓冲区占用率  $u_1$ , 队列 2 的缓冲区占用率  $u_2, \dots$ , 队列  $n$  的缓冲区占用率  $u_n$ ; 由第 2.2 节, 不难得到它们的隶属函数, 如图 7 所示. 我们用因素集  $U$  表示:  $U = \{u_1, u_2, \dots, u_n\}$ . 那么, 从因素集  $U$  到决策集  $V$  存在着一个模糊关系, 我们用模糊矩阵  $R_{(3 \times n)}$  表示, 它的元素  $r_{ij}$  表示队列  $l_j$  属于状态  $L, K, C$  的程度 ( $i = 1, 2, 3; j = 1, 2, \dots, n$ ):

$$R_{(3 \times n)} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ r_{31} & r_{32} & \dots & r_{3n} \end{bmatrix}$$

关于模糊矩阵  $R_{(3 \times n)}$ , 对任意的  $\lambda \in [0, 1]$ , 记  $R_{\lambda(3 \times n)} = (\lambda r_{ij})$ . 其中,  $\lambda r_{ij} = \begin{cases} 1 & r_{ij} > \lambda \\ 0 & r_{ij} \leq \lambda \end{cases}$ , 则称  $R_{\lambda(3 \times n)}$  为模糊矩阵  $R_{(3 \times n)} = (r_{ij})_{3 \times n}$  的  $\lambda$  截矩阵. 本文使用  $R_{(3 \times n)}$  的 0 截矩阵  $R_{0(3 \times n)}$  来描述各类队列当前所处的实时状态, 为了方面描述, 我们称其为状态矩阵. 如果  $r_{ij} = 1$ , 则表示队列  $j$  处于状态  $i$  ( $i = 1, 2, 3; j = 1, 2, \dots, n$ ); 反之, 则不属于某一状态.

现在, 我们考察全局性的缓冲区占用率的情况. 输入模糊变量: 全局性的缓冲区占用率  $u$ ; 采用与单个类型队列类似的描述, 我们不难得到用来描述全局缓冲区占用情况的隶属函数  $\mu(u)$ 、模糊矩阵  $R_{(3 \times 1)}$  和  $R_{(3 \times 1)}$  的状态矩阵  $R_{0(3 \times 1)}$ . 限于篇幅, 不再一一列出.

为了得到各类型队列和全局性缓冲区占用的实时情况, 我们采用计数的办法, 随时跟踪缓冲区的情况, 一旦各占用率变量处于相应隶属函数的不同区间时, 就会触发相应动作, 从而智能性地控制拥塞, 并取得更好的效果. 表 1 描述了模糊拥塞处理的评判规则.

在所有的队列当中, 尽管各队列所对应的连接具有它们各自的权值和带宽, 但总会有一些连接在某一时间段内出现持续的高速现象, 这种高速到达是要超过该连接所拥有的缓冲区资源的. 于是, 相应的队列长度快速增长, 它的缓冲区占用率很快进入拥塞范围, 而此时, 全局有可能还处于正常状态. 但是, 我们不能完全排除该队列是由于较大流量的突发性 (尽管在图 2 中每类队列前加入了令牌桶整形) 而造成的, 因此, 如果此时全局缓冲区占

用率还处于正常状态,这表明缓冲区资源还很丰富,于是我们就扩充分配给该队列的缓冲区容量,使队列退出拥塞状态,以免在网络负载较轻时,长的队列流过早地受到惩罚。

**Table 1** Judging rules of fuzzy congestion control

表 1 模糊拥塞处理的评判规则

|                                   | Normal (the whole)  | Congestion avoidance (the whole)   | Congestion (the whole)   |
|-----------------------------------|---|--|--|
| Normal (queue $j$ )               | The whole state is normal;<br>Queue $j$ is normal   | The whole state is normal;<br>Queue $j$ is normal  | To drop packets according to the wholly dropping module in Fig.2   |
| Congestion avoidance (queue $j$ ) | The whole state is normal;<br>Queue $j$ is normal   | The whole state is normal;<br>Queue $j$ is normal  | To drop packets according to the wholly dropping module in Fig.2   |
| Congestion (queue $j$ )           | The whole state is normal;<br>To enlarge buffer in queue $j$ in order to make it drop out of congestion | The whole state is normal;<br>To drop packets according to the partly dropping module in Fig.2 | To drop packets according to the wholly dropping module in Fig.2<br>To drop packets according to the partly dropping module in Fig.2 |

这种局部性拥塞处理与全局性拥塞处理是相互独立的,只要条件得到满足,二者可以同时启动,也可以单独启动。

**2.4 算法处理步骤**

(1) 根据各类型队列缓冲区占用率的实时状态设置状态矩阵  $R_{0(3 \times n)}$ ,根据全局缓冲区占用率的实时状态设置状态矩阵  $R_{0(3 \times 1)}$ ;

(2) if 状态矩阵  $R_{0(3 \times 1)}$  中  $r_{11}=1$ ,则做

{ 搜索状态矩阵  $R_{0(3 \times n)}$  第 3 行中  $r_{3j}=1$  的列,扩充相应队列  $j$  的缓冲区容量  $C_j$ ,使该队列缓冲区占用率不在拥塞状态;

}

if 状态矩阵  $R_{0(3 \times 1)}$  中  $r_{21}=1$ ,则做

{ 搜索状态矩阵  $R_{0(3 \times n)}$  第 3 行中  $r_{3j}=1$  的列,启动相应队列  $j$  的局部丢包器进行惩罚性的丢包,使该队列退出拥塞状态;

}

if 状态矩阵  $R_{0(3 \times 1)}$  中  $r_{31}=1$ ,则做

{ 启动全局丢包器进行全局性随机丢包,直到全局退出拥塞状态;

搜索状态矩阵  $R_{0(3 \times n)}$  第 3 行中  $r_{3j}=1$  的列,启动相应队列  $j$  的局部丢包器进行惩罚性的丢包,使该队列退出拥塞状态;

}

(3) if 队列  $j$  的缓冲区容量  $C_j$  进行了扩展,则做

{ if 状态矩阵  $R_{0(3 \times 1)}$  中  $r_{31}=1$  或者  $r_{21}=1$  且  $r_{11} \neq 1$ ,则做

{ 恢复队列  $j$  的缓冲区容量  $C_j$  到原来的大小;

if 状态矩阵  $R_{0(3 \times n)}$  中队列  $j$  对应的元素  $r_{3j}=1$ ,则做

{ 启动相应队列  $j$  的局部丢包器进行惩罚性的丢包,使该队列退出拥塞状态;

}

}

}

(4) 如果全局缓冲区占用率由一种状态转换到另一种状态,或者是同时处于两种状态的过渡阶段,此时,描述全局缓冲区占用率状态的状态矩阵  $R_{0(3 \times 1)}$  中相应的元素要进行 0 或者 1 的转换;

(5) 如果某类型队列的缓冲区占用率由一种状态转换到另一种状态,或者是同时处于两种状态的过渡阶段,此时,描述各种类型队列缓冲区占用率状态的状态矩阵  $R_{0(3 \times n)}$  中相应列的元素要进行 0 或者 1 的转换;

(6) 如果有新的网络连接到达,则根据该连接特性,确定所属的队列类型,进入相应的队列流;

(7) 如果状态矩阵  $R_{0(3 \times 1)}$  或者状态矩阵  $R_{0(3 \times n)}$  中有元素进行了 0 或者 1 的转换,就触发第(2)步的运行;

(8) 结束.

### 3 性能评价

目前,NS 是一种比较成熟的网络仿真环境<sup>[14]</sup>,本文对 NS 进行了进一步的扩展.在原有的环境中已有关于尾部丢弃 TD-FIFO 算法、随机提前检测 RED 和加权公平队列 WFQ 等拥塞控制算法,但还没有关于对缓冲区使用率进行定量讨论的功能,为此,我们扩展了这一功能,实现了本文所提出的 FCCA 算法,并与 TD-FIFO,WFQ, RED 算法进行了性能比较.

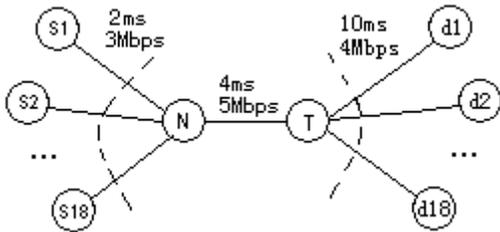


Fig.8 Network topology

图 8 网络拓扑结构

实验用的网络拓扑结构如图 8 所示.

为了得到接近真实网络流的运行情景,我们通过 18 个客户端经由瓶颈节点  $N$  中的同一转发端口向节点  $T$  发送数据流,其中,  $S_1$  和  $S_2$  为 CBR 传输模型,在其余每相邻的 2 个客户端中,一个为 FTP 传输模型,另一个为 CBR 传输模型,然后,再经由节点  $T$  向目的端发送.其中,  $d_1$  端接收  $s_1$  端的数据,  $d_2$  端接收  $s_2$  端的数据,其余依次类推,并定义经过节点  $T$  以后的数据传输与接收的成功率为 100%,以免给 FTP 流造成损失,影响节点  $N$  中运行各种算法时的性能评价.

节点  $N$  中的全局缓冲区大小为 420KB,当运行 TD-FIFO 算法时,按照 TD-FIFO 算法特性处理丢包.在运行其他 3 种算法时,则把 18 个客户端流分为 10 类,其中,  $S_1$  单独为第 1 类队列,  $S_2$  单独为第 2 类队列,其他每相邻 2 个客户端流归为一类,并赋予  $S_1, S_2$  为第 5 优先级,再从第 3 类队列流开始,一直到第 10 类,优先级分别为第 1~第 8 优先级,其中,第 1 优先级级别最高.当运行 RED 算法时,采用能够达到其高性能拥塞控制效果的门限值<sup>[6]</sup>,分别为 130KB 和 300KB;而 WFQ 算法则按照各优先级从  $N$  与  $T$  之间获得相应带宽.我们在运行本文的 FCCA 算法时,将全局缓冲区按照各优先级分配给各类型队列相应大小的缓冲区容量,其中,  $C_{S_1}$  和  $C_{S_2}$  都为 40KB,其他从第 8~第 1 优先级分别获得 25KB,30KB,依次递增直到 60KB,关于全局缓冲区以及各类型队列缓冲区实时状态的描述则按照它们的隶属函数设置.

在性能评价中,我们选择下面几个评价指标:

- ① 节点  $N$  所承受的网络负载率:所有的 18 个客户端的通信量的总和与节点  $N, T$  之间的转发带宽之比值;
- ② 全局性数据包的重复发送率:由于 TCP/IP 协议族的重发机制,网络中延迟或者被丢弃的数据包必须重传.因此,采用这一指标来考察各种算法下,节点  $N$  的转发效率,并定义重复发送率为单位时间内,所有进行过重传的客户端重复传送的数据包量与单位时间内共发送的数据包量之比值;
- ③ 相应客户端数据包流的成功转发率:单位时间内经  $N$  成功转发的该客户端所发送的数据包量与单位时间内该客户端共发送的数据包量之比值.

为了解各种算法对全局性数据包的重复发送率的影响,我们在 18 个客户端按照优先级随机发送相应强度的数据流,并保持增加,使节点  $N$  所承受的网络负载率逐渐递增.于是得到了在各种算法下全局性数据包的重复发送率情况,见表 2.由表 2 不难看出,随着网络负荷率的增大, FCCA 算法的重复发送率明显低于其他 3 种算法.其中, RED 算法随着网络负荷率的增大,由于采用了基于加权队列的拥塞控制策略,表现出了比 WFQ 算法更好的健壮性.而在运行 FCCA 算法时,由于它能够根据缓冲区实时状态,采取整体和局部相结合的策略,进行拥塞控制,保证了高优先级的快速流得到及时的转发,使总体的重传数据量大大减少,提高了节点  $N$  的转发效率,因此,在网络负荷率达到 1.8 时,还能够比尾部丢弃 TD-FIFO 算法的重传率低了约 40%.

图 9 是客户端  $S_1, S_2$  在网络负载率变化过程中,各种算法下的成功转发率曲线.由于客户端  $S_1, S_2$  都为第 5 优先级,那么在 FCCA 和 WFQ 算法中,它们都可以从节点  $N$  的转发出口获得约 0.54Mbps 的链路带宽.其中,图 9(a)是  $S_1$  的成功转发率曲线,在这一过程中,它的数据发送速率为 0.25Mbps;图 9(b)是  $S_2=1$ Mbps 时的成功转发率曲线.从结果来看,虽然网络负载率对所有的算法都有影响,但以 TD-FIFO 算法的影响最为明显,我们不难

理解,TD-FIFO算法由于对所有的数据包流不加任何区分,都是先来先服务,自然就得不到好的成功率.对于其他3种算法,本文提出的 FCCA 在图 9(a)中获得了最高的成功转发率,而在图 9(b)中大约在网络负载率为 1.5 左右时,开始下降,直到网络负载率为 2 时,成功率比 RED,WFQ 算法还要低.这与我们的实验预期结果是一致的,实现了在不同的网络负载率环境下,保护那些没有发生拥塞的数据流,并能够在网络负载率较重的情况下,惩罚那些大大超过自身应得资源的数据流.

Table 2 Repeat ratio of transmission

表 2 重复发送率

| Net load ratio | FCCA (%) | RED (%) | WFQ (%) | TD-FIFO (%) |
|----------------|----------|---------|---------|-------------|
| 1.1            | 0        | 0.21    | 0.24    | 9.09        |
| 1.2            | 0        | 0.62    | 0.64    | 16.6        |
| 1.3            | 0.12     | 1.50    | 1.52    | 23.0        |
| 1.4            | 0.23     | 2.12    | 2.82    | 28.5        |
| 1.5            | 0.48     | 4.86    | 5.26    | 33.3        |
| 1.6            | 0.98     | 7.21    | 8.42    | 37.5        |
| 1.7            | 1.82     | 10.2    | 12.3    | 41.1        |
| 1.8            | 4.12     | 13.1    | 16.5    | 44.4        |

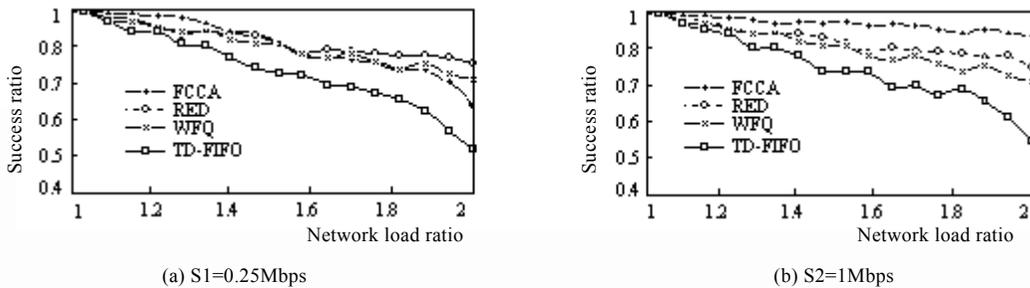


Fig.9 Packets success ratio with networking load

图 9 成功转发率随网络负载率的变化

图 10 是客户端 S1 分别在网络流量负载较轻和较重的情况下,运行各种算法时的成功转发率曲线.其中,横轴为 S1 的发送速率,它在从 0 递增至 1Mbps 的过程中,当超过 0.54Mbps 的界限,而相应的缓冲区占用率又进入拥塞区后,就认为该数据流发生局部拥塞(速度较长时间超过自己的优先级).我们看到,图 10(a)的网络负载率为 1,属于轻载,尽管 S1 的速率超过不少,但 FCCA 算法仍然具有最好的性能.这是由于网络整体负载较轻时,FCCA 算法能够利用丰富的缓冲区资源,临时扩展 S1 的缓冲区空间,使得在网络负载较轻时,不会由于较大的流量突发性而受到损失.而在图 10(b)中,由于网络整体负载严重,因而在 S1 发生大的突发性流时,不再临时扩展缓冲区,而认为该流发生局部拥塞,进行惩罚性的丢包.我们认为,FCCA 算法的这种特性,特别适合于网络中利用流量的恶意攻击者,对于那些在网络负载严重的情况下,还要长时间超速到达的数据流进行有针对性的丢包,而其他几种算法对此则无能为力.

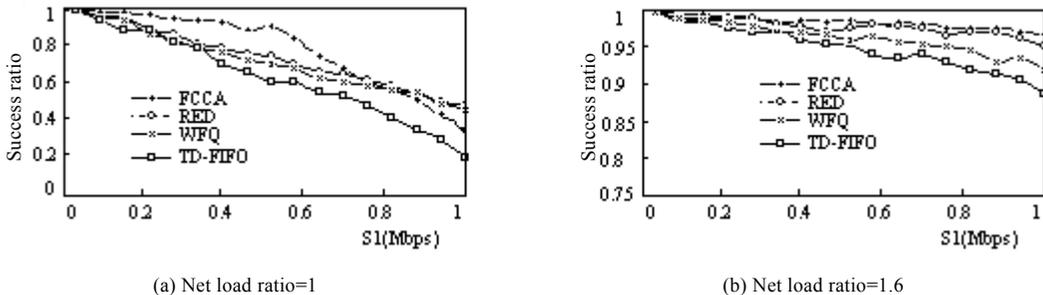


Fig.10 Packets success ratio under different networking load

图 10 网络负载率对成功转发率的影响

## 4 结束语

随着 Internet 的进一步升级,仅仅依靠端到端的拥塞控制已经无法满足要求,新的更为优秀的拥塞控制算法有待于进一步的研究与发展.提出更加接近网络实际流量特性的拥塞控制模型是关键.传统 Poisson<sup>[15]</sup>统计流量模型对于以突发性流量为基本特征的 Internet 网络不再适应,而采用更加贴近 Internet 网络流量特征的自相似模型<sup>[16,17]</sup>,由于其建模与计算过程复杂,实用性不强.为此,本文从转发缓冲区的实时状态出发,运用模糊理论对缓冲区占用率状态这一模糊性问题进行描述,建立起模糊拥塞控制模型,并采用整体和局部相结合的策略,充分利用模糊理论在处理不确定性问题上的优越性,获得了较好的拥塞控制性能.

不难看出,这种算法与 RED,WFQ,TD-FIFO 算法相比较,其实现相对复杂.如地址识别、数据包分类以及各缓冲区占用率的管理都将增加开销,但稍微增加的开销不会影响到路由器或者交换机的数据包转发性能.已提出的一些快速地址识别、数据包分类方法和快速测量技术的硬件实现方法等,使得 FCCA 算法能够满足高速路由器或者交换机在速度上的要求,因而具有实际意义.仿真实验结果表明,FCCA 算法采用局部和整体相结合的策略在保护和隔离非拥塞状态的连接上,具有比传统方法更好的公平性.

## References:

- [1] Jacobson V. Congestion avoidance and control. *ACM Computer Communication Review*, 1988,18(4):314-329.
- [2] Clark DD, Fang WJ. Explicit allocation of best-effort packet delivery service. *IEEE/ACM Trans. on Networking*, 1998,6(4):362-373.
- [3] Floyd S, Fall K. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Trans. on Networking*, 1999,7(4):458-472.
- [4] Braden B, Clark D, Crowcroft J, Dzvie B, Deering S, Estrin D, Floyd S, Jacobson V, Minshall G, Partridge C, Peterson L, Ramakrishnan K, Shenker S, Wroclawski J, Zhang L. Recommendations on queue management and congestion avoidance in the Internet. RFC 2309, 1998.
- [5] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. on Networking*, 1993,1(4):397-413.
- [6] Christiansen M, Jeffay K, Ott D, Smith FD. Tuning RED for Web traffic. *IEEE/ACM Trans. on Networking*, 2001,9(3):249-264.
- [7] Demers A, Keshav S, Shenker S. Analysis and simulation of a fair queueing algorithm. In: *ACM SIGCOMM. Communications Architectures & Protocols*. New York: ACM Press, 1989. 1-12.
- [8] Stiliadis D, Varma A. Efficient fair queueing algorithms for packet-switched networks. *IEEE/ACM Trans. on Networking*, 1998,6(2):175-185.
- [9] Stiliadis D, Varma A. Rate-Proportional servers: A design methodology for fair queueing algorithms. *IEEE/ACM Trans. on Networking*, 1998,6(2):164-174.
- [10] Reisslein M, Ross KW, Rajagopal S. A framework for guaranteeing statistical QoS. *IEEE/ACM Trans. on Networking*, 2002,10(1):27-42.
- [11] Marcus WS. An architecture for QoS analysis and experimentation. *IEEE/ACM Trans. on Networking*, 1996,4(4):597-603.
- [12] Chaskar HM, Madhow U. Fair scheduling with tunable latency: A round-robin approach. *IEEE/ACM Trans. on Networking*, 2003,11(4):592-601.
- [13] Li SY. Fuzzy Control. *Neurocontrol and Intelligent Cybernetics*. Harbin: Harbin Institute of Technology Press, 1996. 42-43 (in Chinese).
- [14] Ns. [Http://mash.cs.berkeley.edu/ns](http://mash.cs.berkeley.edu/ns)
- [15] Paxson V, Floyd S. Wide area traffic: The failure of Poisson modeling. *IEEE/ACM Trans. on Networking*, 1995,3(3):226-244.
- [16] Leland WE, Taqqu MS, Willinger W, Wilson DV. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. on Networking*, 1994,2(1):1-15.
- [17] Grossglauser M, Bolot J-C. On the relevance of long-range dependence in network traffic. *IEEE/ACM Trans. on Networking*, 1999,7(5):629-640.

## 附中文参考文献:

- [13] 李士勇.模糊控制.神经控制和智能控制论.哈尔滨:哈尔滨工业大学出版社,1996.42-43.