

网络多流控制器

赵永祥,陈常嘉

(北方交通大学通信系,北京 100044)

摘 要: 本文提出了一种实现简单的网络元件,我们称之为多流控制器。多流控制器能够识别汇聚流中的最大速率流并把它从汇聚流中分离出来。多流控制器可以用于汇聚流特征的测量、非拥塞响应流的惩罚等多个领域。本文分析了静态条件下的多流控制器数学模型,本文通过大量 ns 仿真,证明了本文给出的数学描述与多流控制器实际工作特征的一致性。

关键词: 流控器; 汇聚流特征; 网络测量

中图分类号: TP393

文献标识码: A

文章编号: 0372-2112 (2003) 02-0259-05

Networks Multiple Flow Regulator

ZHAO Yong-xiang, CHEN Chang-jia

(Dept. of Communications, Northern Jiaotong Univ., Beijing 100044, China)

Abstract: Multiple Flow Regulator (MFR), a novel network component that can be implemented simply, is proposed in this paper. MFR can identify the flow with the biggest rate inside an aggregated flow and separate it from this aggregated flow. MFR can also be used in measurements of characters of an aggregated flow, and to penalize non-responsive flows and so on. Static mathematical model of MFR is analyzed in this paper. Extensive simulations are conducted based on ns-generated flows. Simulation results show that, the working character of a MFR coincides with the mathematical descriptions given in this paper.

Key words: flow regulator; character of aggravated flow; network measurement

1 引言

Turner 在 1986 年提出的关于漏桶的技术构想^[1]和 Cruz 在 1991 年发表的关于限制流的理论框架^[2,3],形成了对单个流进行控制的基本体系结构。单流控制在技术和理论上的成功,在于抓住了应用的主要矛盾——允许并限制突发,并以简洁、有效和相对独立的方式成功地解决了这个问题。因此基于漏桶技术的单流控制已成为现代网络中的一个重要控制技术。

随着网络应用的大规模展开和网络技术的飞速发展,从九十年代开始出现了大量对网络进行控制的新方案和新需求,它们的共同特征是要在汇聚流中,对每个用户流进行控制。使用多个单流控制器来实现控制目标不失是一种解决办法,但为每个可能的用户流设置一个单流控制器会使得系统过于复杂,而且对于大多数网络实际情况而言,一个汇聚流中到底会有多少个用户流是一个很难确定的问题,这进一步增加了使用单流控制器来对汇聚的多个流进行控制的难度。

当前研究工作的共同观点是针对一个特定的需求,基于一个特定的网络配置,提出一个解决该问题的具体技术方案,这些解决方案有的可能比较实用,但它们均不是从多流控制的角度提出问题。类似于单流控制,多流控制尽可能地脱开网

络的具体资源和具体配置,将网络的控制目标描述为所希望得到的输出流间的某种特定关系,并寻找能满足这种特定关系的技术实现方案。这样做的好处是,在研究方面可以摆脱过多的具体网络和具体应用方面的细节,而把注意力集中在多流控制的主要问题方面;在实践上可以将一个有效的多流控制技术应用于各种具体问题中,而无须为每个应用分别设计控制方案。

国际上目前尚未见到关于多流控制的研究构想,最接近的可能是近年来提出的每跳行为 PHB (Per Hop Behavior) 概念,它的目标是将网络的整个控制问题用单个路由器的转发行为来进行描述和固化,以达到网络可扩展的目的。PHB 概念所包括的内容过于庞杂,只有在对这个概念进一步细化才能有效地展开有实质意义的研究,而多流控制技术和多流控制理论正是 PHB 概念的一个细化。我们关于研究多流控制器的想法是在我们对惩罚不负责任流的研究工作中形成的^[4,5],进一步我们发现,将多流控制概念与主动队列管理 (Active Queue Management) 适当分离,更有利于根据不同的控制目标设计不同的网络部件以及从不同角度分析各自的控制性能,多流控制的主要任务是根据不同控制目标调整其输出流间的相互关系,而队列管理的目的是有效调整缓存空间与输出通道间的

收稿日期:2001-06-07;修回日期:2002-09-16

基金项目:国家自然科学基金(No. 60132030, No. 60202001)

关系,尽量减小总体时延和总体丢失率.进一步我们发现,可以将同一种设计的多流控制器运用在多种不同类型的网络应用中,而对于全光网络,目前要找寻的正是一种不使用或几乎不使用数据缓存的多流控制技术,这进一步证实了从多流控制的角度研究网络控制问题的重要意义.

本文提出一种多流控制器的实现方法.这个多流控制器可以在不记忆各个流的状态下实现,可以孤立的考核单个多流控制器的控制行为和控制性能,每一个流不须附带任何专门为多流控制器使用的附加控制信息.它的最大特征是它可以看成为一个网络元件,可以在不改变网络的任何设置下插入网络的任何一点,对输入的多个流进行调控,使输出的汇聚流具有所希望的特征.

本文第二节给出关于多流控制器的一个基于随机切换的实现框架,第三节对这个实现框架在恒定流(CBR和Poisson流)、无限记忆盒长度下的理论模型和解析分析,第四给出大量基于ns的仿真工作.第五节给出有限记忆盒长度多流控制器的解析模型.最后,第六节给出本文的总结.

2 多流控制器的实现

2.1 多流控制器基本构成

多流控制器的基本构成示于图1,有三个外部端口:输入端口,同标识输出端口和异标识输出端口.多流控制器由五个主要内部单元组成:切换单元、比较单元、记忆盒、随机地址产生器和控制时序产生器,它们的功能分述如下:

切换单元 有3个数据端口:输入端口 M_I ,同标识输出端口 S_O 和异标识输出端口 D_O ,以及一个控制输入端口 C_I .当 C_I 端口激活时输入的是一个0/1信号,当有新分组到达输入端口后,切换单元检查 C_I 端口是否激活,当激活后,若 C_I 端口的输入值为1则将新到达分组切换到同标识输出端口,否则切换到异标识输出端口.

比较单元 有2个流标识输入端口 M_I 和 R_I ,以及控制输入输出端口 C_I 和 C_O .当控制输入端口 C_I 上的值为1时,比较单元激活控制输出端口 C_O ,并比较 M_I 和 R_I 端口上的流标识,若它们相同则 C_O 端口输出1信号,否则 C_O 端口输出0信号.

记忆盒 是一个存储器,由若干可地址的流标识记忆单元组成.它的数据输入端口 I 接收流标识,并在读写控制端口 WR 的控制下将输入的流标识写入地址端口 AD 所表明的流标识记忆单元内.同样的,在读写控制端口 WR 的控制下,记忆盒将地址端口 AD 指定的流标识记忆单元内的内容输出到数据输出端口 O .

随机地址产生器 在输入到 C_I 端口的控制信号的作用下,随机地产生记忆盒所需的地址.

控制时序产生器 接收新分组到达信号 A_p 和流比较结果信号 A_C ,产生多流控制器所需的各种控制时序,主要的输出有记忆盒读写控制信号 WR_C ,随机地址产生器控制信号 AD_C 和比较单元输出激活信号 A_a .

2.2 多流控制器的工作流程

多流控制器的工作流程如下:

(1)随时监视输入端口是否有新分组到达,新到达分组的流标识被检出并存入新流标识寄存器后, A_p 信号激活向控制时序产生器发出新分组到达信号.

(2)一旦接收到新分组到达信号 A_p ,控制时序产生器依概率 p 对记忆盒进行刷新,即控制时序产生器独立地产生一个0到1之间的随机数 N ,若 $N \leq p$ 则对记忆盒进行刷新,否则跳过这次刷新过程,转到下面的步骤3.若控制时序产生器确定要对记忆盒刷新,则通过 AD_C 信号触发随机地址产生器为记忆盒产生一个随机地址,并通过 WR_C 信号控制记忆盒将存放在新流标识寄存器的内容写入记忆盒中由随机地址指定的流标识记忆单元内.

(3)控制时序产生器通过 AD_C 信号触发随机地址产生器为记忆盒产生一个随机地址,并通过 WR_C 信号控制记忆盒将随机地址指定的流标识记忆单元存放内容输出到参考流标识寄存器,并通过 A_a 信号激活比较单元.

(4)比较单元在控制输入输出端口 C_I 收到 A_a 激活信号后,比较输入端口 M_I 和 R_I 的流标识,并将比较结果输出到控制输出端口 C_O ,端口 C_O 的输出一方面输入到切换单元作为控制信号,也作为流比较结果信号 A_C 提供给控制时序产生器.

(5)控制时序产生器在收到流比较结果信号 A_C 后,启动新一轮的对记忆盒刷新,若 A_C 信号为1,即比较结果为标识相同,则控制时序产生器依概率 p_s 决定是否对记忆盒进行刷新,否则 A_C 信号为0,即比较结果为标识不相同,则控制时序产生器依概率 p_d 决定是否对记忆盒进行刷新.若确定不对记忆盒刷新,则转到步骤1,准备接收下一个新分组,否则在对记忆盒刷新后转到步骤1.

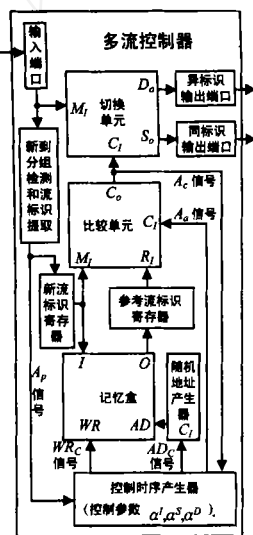


图1 多流控制器的构成

3 多流控制器的极限描述和可能应用

本节将推导多流控制器的静态数学描述,并根据这个静态描述给出多流控制器的一些可能应用.首先假定输入到多流控制器的是短记忆长寿命流的汇聚流,记汇聚流中流的个数为 N ,每个到达分组依概率 u_j 属于流 j .对于CBR流或长寿命的TCP流来说,概率 u_j 可以物理解释为汇聚流中流 j 的归一化输入速率,若流 j 的输入速率记为 $r_j^{(I)}$,则 $u_j^{(I)} = r_j^{(I)} / \sum_k r_k^{(I)}$.描述多流控制器的关键参数是记忆盒中的流占用比例FOR (Flow Occupancy Ratio),对于流 j 的FOR记为 p_j ,并定义为记忆盒中存储流 j 的单元个数与总的记忆盒存储长度之比值.记流 j 在同标识端口和异标识端口的输出速率分别为 $r_j^{(So)}$ 和 $r_j^{(Do)}$.知道了流占用比例 p_j 后有:

$$\begin{cases} r_j^{(Do)} = (1 - p_j) r_j^{(I)} \\ r_j^{(So)} = p_j r_j^{(I)} \end{cases} \quad (1)$$

假定输入到多流控制器的流是长寿命的且概率特征不变,则经过初始阶段后,多流控制器的行为将达到平衡状态,平衡状态下的流占用比例 FOR 可基于下面两种物理假设求出,第一个物理假设是,当系统处于平衡状态后,记忆盒中流 j 的比例与试图增加流 j 存贮单元的比例相等. 由于试图增加流 j 的努力为 $i(1 - p_j)u_j + d(1 - p_j)^2u_j + s(1 - p_j)p_ju_j$, 所以有:

$$p_j = \frac{i u_j + d(1 - p_j)u_j + s p_j u_j}{i + d + (s - d)k p_k u_k} \quad (2)$$

第二个物理假设是,当系统处于平衡状态后,记忆盒中试图增加流 j 存贮单元的努力应该等于试图减少流 j 存贮单元的努力. 由于试图减少流 j 的努力为 $i p_j(1 - u_j) + d p_j k j(1 - p_k)u_k + s p_j k j p_k u_k$ 所以有:

$$p_j = \frac{(i + d)u_j}{i + d + (d - s)u_j + (s - d)k p_k u_k} \quad (3)$$

可以证明式(2)和(3)是等同的.

选择不同的刷新参数 i, s, d , 多流控制器将表现出不同的控制特征,主要的控制特征可分为三类,分别对应于 $(s = 0, d = 0)$ 、 $(i = 0, s = 0)$ 和 $(s = 0)$. 第一种情况较为类似于文献[6]已进行的讨论,所以我们跟随文献[6]的命名称其为 SRED 控制器;第二种情况有些类似于文献[7]的方案,我们称其为 CHOKe 控制器;第三种情况较为类似于文献[4,5],所以我们称其为 Trash 控制器. 对于 SRED 控制器和 CHOKe 控制器可以容易得到:

$$p_j^{(SRED)} = u_j \quad (4)$$

$$p_j^{(CHOKe)} = \frac{u_j}{1 + u_j - k p_k u_k} \quad (5)$$

而对于 Trash 控制器,最有兴趣的是 $(i = 0, d = 0)$ 的情况,从(3)有:

$$(i + d)(p_j - u_j) + (s - d)p_j(k p_k u_k - u_j) = 0$$

当 $(i = 0, d = 0, s > 0)$ 时,或者 $p_j = 0$ 或者 $k p_k u_k - u_j = 0$. 令 u_{\max} 是汇聚流中速率最大的流的归一化速率,则只有当 $u_j = u_{\max}$ 时才可能有 $p_j > 0$. 如果记汇聚流中取得最大归一化速率的流的数目为 N_{\max} ,则对于 Trash 控制器有

$$p_j^{(Trash)}(i = 0, d = 0) = \begin{cases} 0, & u_j < u_{\max} \\ 1/N_{\max}, & u_j = u_{\max} \end{cases} \quad (6)$$

对于 Trash 控制器,我们又可以分为输入辅助刷新 $Trash-I$ ($d = 0$) 和异标识输入辅助刷新 $Trash-D$ ($i = 0$) 两类,分别令 $i = i'/s$ 和 $d = d'/s$, 则对应的流占用比例分别为:

$$\begin{cases} p_j^{(Trash-I)} = \frac{u_j}{i + k p_k u_k - u_j} \\ p_j^{(Trash-D)} = \frac{u_j}{(1 - i')(k p_k u_k - u_j)} \end{cases} \quad (7)$$

图 2 给出了不同类型多流控制器在两个输入流时的控制特征,它们具有的共同特征是 $p_j(u_j)$ 均为 u_j 的增函数,这一点不难从下式看出

$$p_j - p_i =$$

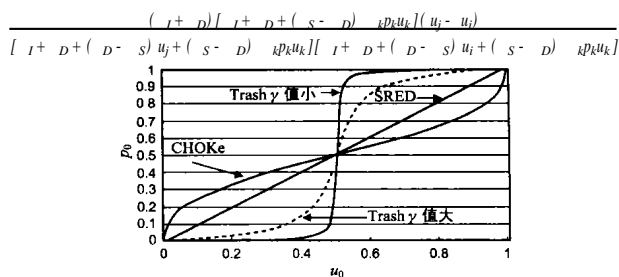


图 2 各种多流控制器的 FOR

把多流控制器同标识端口的输出丢弃就可以实现对最大速率流的处罚. 从图 2 可以看出,不同类型的多流控制器的惩罚能力是互不相同的,惩罚能力最弱的是 CHOKe 类型的多流控制器,而惩罚能力最强的是 Trash 类型的多流控制器. 特别当 γ 的值很小时,Trash 类型的多流控制器非常类似于一个开关门,只对那些具有最大输入速率的流进行惩罚,而几乎不影响输入速率小的流.

当利用 Trash 类型多流控制器的门特征进行惩罚时,尽管汇聚流中速率最大流的速率非常小,也会受到非常严重的损伤,可以通过图 3 所示的人为提供导流的办法克服这一现象. 这时只有当最大速率流的速率大于导流速率之后,才会受到严重的惩罚.



图 3 利用导流限制对最大速率流的惩罚

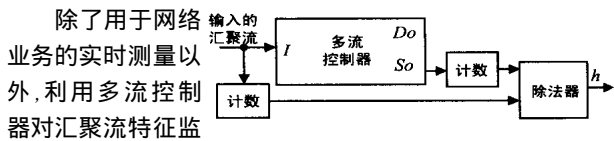
采用图 4 的配置可以对汇聚流的特征进行无流状态的测量,我们称图 4 中除法器的输出 h 为碰撞率,因为它恰好是输入分组与随机抽取的记忆盒中单元有相同流标识的概率,即:

$$h = p_j u_j$$

不难看出当采用 Trash 类型多流控制器的门特征进行测量时,碰撞率 $h^{(Trash)}$ 就是汇聚流中的最大单流速率,即 $u_{\max} \geq h^{(Trash)}$ 而当采用 SRED 或 CHOKe 类型多流控制器时,碰撞率 h 的倒数是汇聚流中流的个数,即 $N \geq 1/h$. 由于在汇聚流特征的测量中会碰到许多细节问题,很难在一篇文章中细致讨论,所以我们将用另一篇论文中展开对这一问题的讨论.

除了用于网络业务的实时测量以外,利用多流控制器对汇聚流特征监测的一个可能应用

图 4 利用多流控制器对汇聚流特征测量是监视服务器的负荷平衡情况,以及对于拒绝业务攻击进行识别和告警. 例如利用 Trash 类型多流控制器的门特征对输入到服务器的汇聚流进行实时监测,并预置一个门限,当汇聚流中存在单流的最大速率超过预置门限时,则启动告警指示,并可从同标识端口的输出分组中识别速率超过预置门限的流.



4 多流控制器仿真

输入的流为 CBR 流, 记输入流的个数为 N , 输入流 $j(j=0, \dots, N-1)$ 的输入速率记为 $r_j^{(I)}$, 归一化输入速率记为 $u_j^{(I)} = r_j^{(I)} / r_k^{(I)}$. 多流控制器的设定参数包括记忆盒长度 L 和刷新参数 α, s, D . 在仿真中对每个流 j 的 FOR p_j 进行在线测量, 并分别通过异同标识端口和异标识端口的输出测量流 j 的 $r_j^{(So)}$ 和 $r_j^{(Do)}$. 所有的仿真用软件 ns2^[8] 实现.

4.1 多个输入流的仿真

4.1.1 不同输入速率下的多个流的仿真 本节进行多个输入流的仿真. 仍然固定总输入速率 $r^{(I)}$ 为 1Mbps, 流的总数为 10, 令流 0 的速率为最大, 流 1 的速率为次大, 其它流的速率均相等. 令流 0 的速率 $r_0^{(I)}$ 为 300Kbps, 流 1 的速率 $r_1^{(I)}$ 从 100Kbps 到 300Kbps 选 10 个点, 流 $j(j=2, \dots, N-2)$ 的速率为 $r_j^{(I)} = \frac{1}{N} (r^{(I)} - r_1^{(I)} - r_0^{(I)})$. 记忆盒长度为 10, 保持 $D=0, s=1$, 变化 α 为 0.01 和 0.1.

图 5 给出了同标识端口的输出速率, 由图可见, 当 α 为 0.01 时, 当 r_1 的速率低于 300Kbit/s (既 r_0 的速率) 时, 同标识端口几乎没有流 1 的分组出现, 但是当 r_1 的速率超过

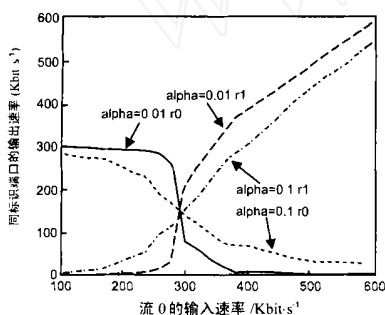


图 5 多个流时的同标识端口的输出速率

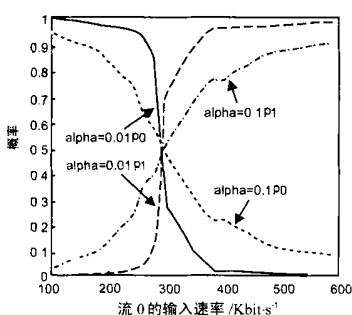


图 6 多个流时的记忆盒的占用概率

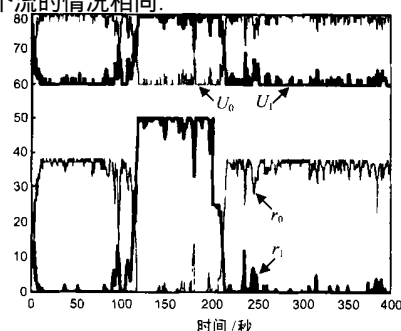


图 7 多个流时的时间动态行为

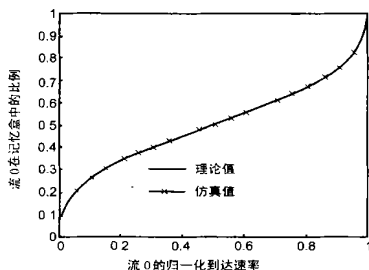


图 8 CHOKe 的仿真值与理论值的比较

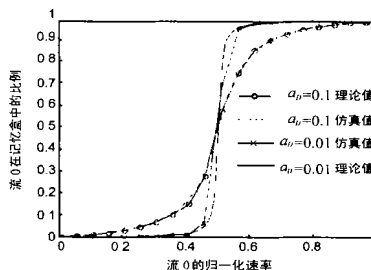


图 9 Trash.pass 的仿真值与理论值的比较

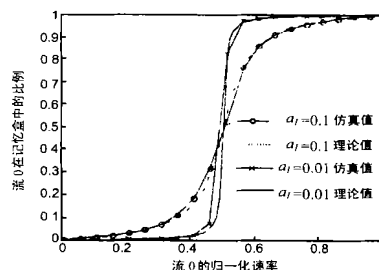


图 10 Trash.inp 的仿真值与理论值的比较

4.2 仿真数据与理论结果的比较

4.2.1 CHOKe 的理论值与仿真值的比较 设 $\alpha=0, D=1, s=0$, 记忆盒长度为 20, 理论值用第 3 节中的公式计算, 结果如下图 8 所示. 由图可见, 理论值与仿真值几乎重合.

4.2.2 Trash.D 的理论值与仿真值的比较 设 $\alpha=0, s=1$, 变化 D 为 0.01、0.1, 记忆盒长为 20, 理论值用第 3 节中的公式计算, 结果如图 9 所示. 由图可见, 当流 0 的归一化速率在 0.5 附近时, 理论值与仿真值略有不同, 其他地方理论值与

300Kbit/s 时, 也就是流 1 成为所有流中速率最大者, 则流 1 在同标识端口的速率迅速增加, 图 6 绘出了记忆盒中缓存占用概率, 显然, 当 r_1 的速率超过 300Kbit/s 时, 流 1 在缓存中占用的概率以非常陡峭的斜率增加, 这说明 Trash 类型的多流控制器在多个流的情况下依然保持了将最大速率流导向同标识端口的工作特征. 当 α 为 0.1 时, 控制器的性能下降. 保持上面仿真设置不变, 把流的数目增加到 50, 仿真结果变化不大, 就不再重复了.

4.1.2 多个流时的动态行为 下面探讨在多流条件下的动态行为. 流的总数为 9. 令流 0 的输入速率 $r_0^{(I)}$ 为 300Kbps. 流 1 的速率 $r_1^{(I)}$ 头 100 秒内为 200Kbps, 100 秒到 200 秒内为 400Kbps, 200 秒到仿真结束为 400Kbps. 其它流的速率均相等, 均为 50Kbps. 选择 α 为 0.01, 记忆盒长度为 20.

图 7 绘出了 r_0, r_1 , 流 0 在记忆盒数目 U_1 , 流 1 在记忆盒数目 U_2 . 为了在一个图内显示出这四个量, U_1, U_2 分别在仿真数据的基础上加了 60, r_0, r_1 的单位为分组/秒, 分组长度为 1000 字节. 由图可见, 在多个流的条件下, 控制器能够动态地从输入流中挑出速率最大的流并把它送到同标识端口输出. 把 α 为 0.001 也有相同的结论但是响应速度大为减缓, 这个结果与两个流的情况相同.

4.2.3 Trash.I 的理论值与仿真值的比较

设 $D=0, s=1$, 变化 α 为 0.01 和 0.1. 记忆盒长度为 20, 理论值用第 3 节中的公式计算, 结果如图 10 所示. 由图可见, 与 Trash.D 类似, 当流 0 的归一化速率在 0.5 附近时, 理论值与仿真值略有不同, 其他地方理论值与仿真值几乎重合.

5 有限记忆盒长度多流控制器的解析模型

第 3 节给出了多流控制器在缓存长度无限大时的记忆盒

占用率概率分布的极限形式,实际使用的多流控制器的记忆盒长度都只可能是有限的,因此有必要分析多流控制器在有限记忆盒长度下的工作特征,以及讨论极限形式对有限记忆盒长度的多流控制器工作特征的逼近程度,有兴趣了解推导过程的读者可参阅文[9].

6 总结

这些工作表明多流控制器是一个有重要意义的,有挖掘潜力的和有广泛应用前景的概念,多流控制器在网络控制方面可以有多种实际应用,可能的应用包括:(1)维护网络资源的公平共享;(2)惩罚高输入速率或不实施端到端拥塞控制的用戶;(3)网络测量,其中包括测量汇聚流中的单流特征,如最大单流速率、流速率的偏差,流的个数等,这些测量可以应用于各种网络或服务器负荷平衡性检查和监测;(4)按速率进行流分离:可用于速率敏感负荷分配和工作等等.多流控制器的应用范围有可能还要广泛,需要进一步探讨.

参考文献:

- [1] J Turner. New directions in communications (or which way to the information age?) [J]. IEEE Commun. Mag. 1986, 24(10): 8 - 15.
- [2] R L Cruz. A calculus for network delay, part I: network elements in isolation [J]. IEEE Trans. On Information Theory, 1991, 37(1): 114 - 131.
- [3] R L Cruz. A calculus for network delay, part II: network analysis [J]. IEEE Trans. On Information Theory, 1991, 37(1): 132 - 141.
- [4] 赵永祥,陈常嘉.一种对非 TCP 友好流的实用无状态惩罚算法[J].通信学报,2001,23(8):100 - 107.
- [5] 赵永祥,陈常嘉.一种对非 TCP 友好流的无状态惩罚算法[J].铁道学报,2001,23(1):66 - 71.
- [6] Ott, T, Lakshman, T, Wong, L, SRED: Stabilized RED [A]. IEEE Proc. INFOCOM [C]. 1999. 1346 - 1355.
- [7] Rong Pan, Balaji Prabhakar, Konstantinos Psounis. CHOKe: A stateless active queue management scheme for approximating fair bandwidth allocation [A]. IEEE Proc. INFOCOM [C]. 2000. 942 - 951.
- [8] ns Network Simulator (version 2.1b6) [EB/OL]. <http://www.mash.cs.berkeley.edu/ns/>, 1999.
- [9] 赵永祥.端到端拥塞控制的强制实施及多流控制器[D].北京:北方交通大学研究生院,2001.91 - 111.

作者简介:



赵永祥 男,1970 年出生于云南昆明,2001 年于北方交通大学通信系获博士学位,现为北方交通大学通信系讲师,主要研究方向:调度及缓存管理,拥塞控制. email: yongxiang_zh@sina.com.

陈常嘉 男,北方交通大学教授,博士生导师,主要研究领域:调度及缓存管理,拥塞控制,网络路由算法等.