

一种基于方程的多媒体实时流拥塞控制机制^{*}

An Equation-Based Congestion Control Mechanism for Realtime Stream

任立勇 卢显良

(电子科技大学计算机科学系 成都610054)

Abstract With the increasing of network multimedia, TCP congestion control mechanism, which is based on AIMD algorithm, cannot efficiently solve congestion problems resulting from multimedia streams. In this paper, we first analyze TCP throughput model, then an equation-based congestion control algorithm, EBCC, is presented. With this algorithm, the sender explicitly adjusts its sending rate as a function of the measured rate of loss events and RTT. Finally, theoretic analysis and experimental results show that EBCC can not only smoothly adjust sending rate to satisfy the requirement of lower delay-jitter of multimedia streams, but also assure TCP-friendliness.

Keywords Congestion control, Multimedia stream, Quality of Service, TCP-friendly

1 引言

端到端拥塞控制机制对 Internet 的鲁棒性和稳定性至关重要。目前 Internet 上许多应用使用基于窗口的拥塞控制算法对网络拥塞进行响应和避免网络崩溃。可以说,今天 Internet 的成功很大程度上得益于 TCP 拥塞控制算法的不断改进^[1]。

随着 WWW 应用和各种实时多媒体信息在 Internet 上的不断增加,仅仅依靠单一的 TCP 拥塞控制算法已经显得力不从心了。TCP 基于和式增加、积式减少(AIMD)的慢速启动、拥塞避免、快速重传、快速恢复机制对于那些时延要求不高但必须可靠传输的应用非常适合,但对于有各种各样服务质量(QoS)要求的实时多媒体应用它们则不能满足需要。这些实时应用对时延敏感,但不要求可靠传输,在连接期间,要求有一定的带宽下界,超时和失序的分组并不需要重传。为保证服务质量,许多多媒体应用以固定速率发送数据,而不考虑网络状况,因此常常造成网络拥塞甚至网络崩溃。另外,由于今天的 Internet 提供“尽力而为”服务和 FIFO 队列管理算法,因此当这些多媒体流与 TCP 流竞争网络资源时,往往造成 TCP 流饿死。为此,许多研究者提出了在单播和组播环境下各种适合于实时多媒体流的拥塞控制算法^[2~6],这些算法在一定程度上解决了实时多媒体流与 TCP 流竞争网络资源时的公平性问题,但在实际应用时都或多或少地存在一些问

题^[4]。

本文在分析已有的实时多媒体应用拥塞控制算法的基础上,提出了一种基于方程的拥塞控制算法 EBCC(Equation-Based Congestion Control),该算法不仅能满足实时多媒体应用相对稳定的发送速率要求,即在网络拥塞或网络空闲时,根据分组丢失率与往返时延 RTT 较为平滑地调节发送速率,更为重要的是,在同等条件的网络环境下,EBCC 流与 TCP 流能获得相当的吞吐量,较好地解决了协议间的公平性问题。

2 TCP 吞吐量模型

J. Mahdavi 等人第一次提出了基于方程的拥塞控制算法^[3],该方法主要针对在“尽力而为”的 Internet 环境下,为没有配置 TCP 传输控制的应用实现一种 TCP 友好(TCP-friendly)流控,与 TCP 的 AIMD 拥塞控制对每个分组丢失立即回退发送速率相比,基于方程的拥塞控制使用一个控制方程计算发送方的最大传输速率,发送方根据分组丢失率与 RTT 显式地调节发送速率。为保证与 TCP 流公平地竞争网络带宽,所选择的控制方程必须精确地描述出在稳定状态下 TCP 的速率响应函数。S. Floyd 等人提出了一种简单的 TCP 吞吐量模型^[6]:

$$T \leq \frac{1.5\sqrt{2/3} * M}{RTT * \sqrt{p}} \approx \frac{1.22 * M}{RTT * \sqrt{p}} \quad (1)$$

式中, T 为传输速率, M 为最大分组长度, RTT 为连接

^{*} 本文得到国家九五科技攻关项目基金资助,信息产业部生产发展基金资助。任立勇 博士研究生,主要研究方向为网络服务质量、新型网络协议、分布式操作系统等。卢显良 教授,博士生导师,主要研究方向为操作系统与网络应用技术。

往返时延, p 为最近分组丢失率。

(式(1)具体推导过程可参考文[6],另外,如果接收方发送 Delayed-ACK(即接收方每接收两个分组至多发送一个 ACK),此时可得到更严格的发送速率上界,即 $T \leq \frac{1.5\sqrt{1/3 * M}}{RTT * \sqrt{p}}$;

实际上,(1)式给出的仅仅是 TCP 流的发送速率上界,实际 TCP 流的发送速率远小于(1)式估算结果,这是由于 M 为连接的最大分组长度,而 RTT 为最小往返时延,因此,根据(1)式实现的基于方程的 TCP 友好流在与 TCP 流竞争时,往往占用更多的网络带宽,表现得极不友好,另外,上述模型没有考虑由于等待重传定时器超时(Timeout)而导致的 TCP 吞吐量下降(这在目前的 Internet 环境中相当普遍),因此在拥塞窗口小于4时,(1)式过高地估算了连接占用的带宽,此时分组丢失率大于16%[5]。

为了更精确地描述实际 Internet 网络环境下 TCP 吞吐量特性,J. Padhye 等人提出了有“无限”数据发送的 TCP 流在稳定状态下的吞吐量方程[2],该方程在推导过程中不仅考虑了快速重传对吞吐量的影响,同时还考虑了由于超时导致的吞吐量下降问题。该模型基于一种“轮转”的拥塞控制,轮转周期等于一个 RTT ,并假设每轮中的分组丢失与其它轮中的分组丢失无关,同时,若某一分组丢失,则本轮中以后的分组均丢失(这主要是因为目前大多数的网关中配置的 Drop-tail 算法所致)。在每一轮开始根据上一轮的分组丢失率与超时信息计算本轮的发送速率,吞吐量模型(复杂模型)为:

$$T = \frac{M}{RTT * \sqrt{2p/3 + T_{RTO}(3\sqrt{3p/8})p(1+32p^2)}} \quad (2)$$

(2)式中除 T_{RTO} 表示 TCP 重传超时值外,其余符号与(1)式中相应符号意义相同。

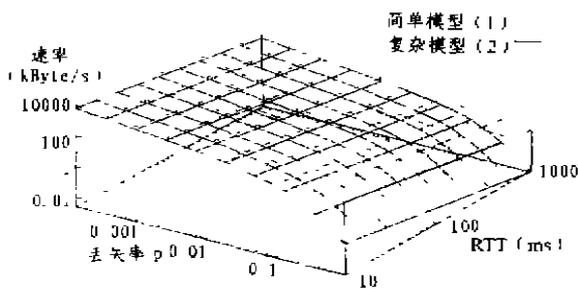


图1 两种 TCP 流量模型比较

图1所示为两种 TCP 吞吐量模型在不同网络环境下的速率比较。可以看出,当丢失率较小时,TCP 超时重发的情况很少,因此两种模型计算出的速率基本一致,但当丢失率较大时,TCP 的超时事件急剧增加,因

此复杂模型计算的速率下降很快,这主要是因为 TCP 拥塞窗口当出现分组超时将缩减为1所致,因此,由(2)式所描述的复杂模型更精确。

3 基于方程的拥塞控制 EBCC

实时多媒体流与 TCP 流的数据传输有明显的区别,首先,实时多媒体应用要求要相对稳定的发送速率;其次,实时多媒体应用对数据的可靠投递要求不是很高,过时、丢失或失序的分组不需要重发。而 TCP 协议拥塞控制与可靠传输耦合在一起,其基于窗口的 AIMD 算法造成连接的发送速率随网络拥塞状况波动较大。因此,传统的 TCP 拥塞控制显然不适合于实时多媒体应用。而基于方程的拥塞控制将拥塞控制与丢失重传分离开来,分组的丢失与错误由应用层来处理,因此能较好地满足实时多媒体应用的要求。

设计一个基于方程的拥塞控制算法最基本的问题是选择合适的控制方程,如果控制方程描述的吞吐量过于保守,则遵守该方程的实时多媒体流与 TCP 流竞争时,尽管它们是 TCP 友好的(TCP-friendly),但不能获得与 TCP 相同的带宽,反之,如果控制方程描述的吞吐量过高,则遵守该方程的实时多媒体流与 TCP 流竞争时,将造成 TCP 流饿死,因而是非 TCP 友好的(non TCP-friendly)。从第2节分析可知,(2)式所描述的吞吐量模型能较好地反映稳定状态下 TCP 流的吞吐量,因此,本文采用(2)式作为实时多媒体流的拥塞控制方程。

仔细分析(2)式控制方程,为计算稳定状态下的发送速率,需要确定四个变量 M, RTT, T_{RTO}, p 。其中 M 可通过路径 MTU 发现算法获得[1], T_{RTO} 可通过 RTT 估算,因此问题的关键是测量与估算往返时间 RTT 与分组丢失率 p ,下面我们分别描述。

3.1 往返时间 RTT 与超时值 T_{RTO} 的估计

EBCC 采用与 TCP 类似的方法,使用低通滤波器计算平均往返时间,该低通滤波器使用指数加权滑动平均算法(EWMA),即:

$$RTT \leftarrow (1-w) * RTT_{old} + w * RTT_{sample} \quad (3)$$

其中 $w \in [0, 1]$ 。当 w 较大时,平均 RTT 对短暂的时延变化较敏感,反之,当 w 太小时(如 $w=0.01$),平均 RTT 则不能及时反映网络的拥塞状态。由于 EBCC 要求传输速率“相对”稳定,平均 RTT 抖动幅度不宜太大,因此,本文选取较小的 w (如 $w=0.1$)。

对超时重传值 T_{RTO} 的估计,不同的 TCP 实现采取了不同的算法(如 Karn 算法与定时器补偿、大时延方差平均法等),但从方程(2)式可知,当丢失率 p 较小时, T_{RTO} 对发送速率的影响也较小,同时,与 TCP 不同的是,EBCC 并不需要用 T_{RTO} 来决定是否需要超时

重传,因此,为简单起见 EBCC 用下式估算超时值:

$$T_{RTO} = 4 * RTT \quad (4)$$

3.2 分组丢失率 p 的估计

分组丢失率作为评判网络拥塞状况的一个重要参数,其估计值的准确性对传输速率的调节相当关键。如 J. Padhye 等人提出的 TCP 友好速率控制协议(TFR-CP)^[3]在估计分组丢失率时,将时间分为固定的时间片,发送方计算前一个时间片中的丢失比率作为 p ,并据此计算发送速率。分析与实验证明,该方法由于计算时间尺度固定,因此,不能较好地反映网络当前状况,尤其是对于 RTT 变化较大的连接,其发送速率振荡非常剧烈,因此,本文采用随网络状态变化的动态时间尺度计算分组丢失率,同时,为保证分组丢失率估计的平滑性,可考虑对历史丢失率进行指数加权滑动平均(EWMA)或加权平均丢失间隔(WALI, Weighted Average Loss Interval)。

此处我们定义丢失间隔 s 为两个丢失事件间的分组数(包含第二个丢失分组),则分组丢失率 $p = 1/s$ 。对丢失间隔的估计进行指数加权滑动平均:

$$\bar{S} \leftarrow (1 - \eta) * \bar{S} + \eta * S_n \quad (5)$$

其中 $\eta \in [0, 1]$ 。式中 \bar{S} 为平均丢失间隔, S_n 为最近丢失间隔。由于 η 的取值大小影响着平均间隔对短暂拥塞的响应程度,因此指数加权滑动平均不能适应多种

网络条件。本文采用加权平均丢失间隔法(WALI)。

假设过去 n 个丢失间隔分别为 s_1, s_2, \dots, s_n , 其中 s_1 为当前丢失间隔, s_n 为当前第 n 个丢失间隔,平均丢失间隔定义为:

$$\bar{s} = \sum_{i=1}^n \omega_i s_i / \sum_{i=1}^n \omega_i \quad (6)$$

为保证平均丢失间隔既能及时反应最近丢失状况,又能一定程度上反应历史丢失状况,各丢失间隔权值可取值如下:

$$\omega_i = \begin{cases} 1 & 1 \leq i \leq \lfloor n/2 \rfloor \\ 1 - \frac{i - n/2}{n/2 + 1} & \lfloor n/2 \rfloor < i \leq n \end{cases} \quad (7)$$

本文取 $n=8$,此时各权值如表1所示。

表1 $n=8$ 时各丢失间隔权值

i	1	2	3	4	5	6	7	8
ω_i	1	1	1	1	0.8	0.6	0.4	0.2

如图2所示,我们对两种求平均丢失间隔方法做比较,实验中假定分组丢失为周期性的(为比较方便起见),可以看出,指数加权滑动平均(EWMA)当权值较大(取0.5)时,尽管能对网络拥塞状况作出快速反应,但由此计算出的分组丢失率将导致发送速率振荡,而加权平均间隔则能对分组的丢失间隔作出较为平滑的估计,满足 EBCC 的要求。

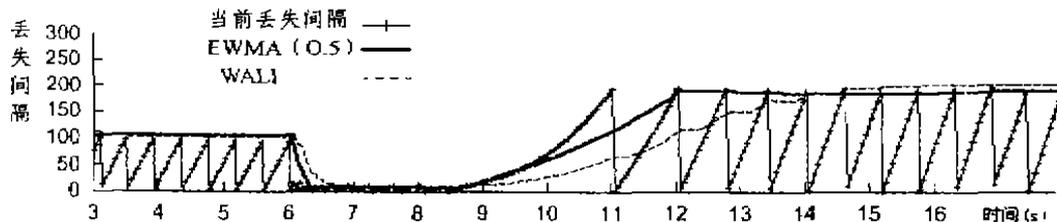


图2 两种平均丢失间隔估计

3.3 调节发送速率

EBCC 要求当估算出的发送速率 T 与实际发送速率 T_{est} 的差值超过一定的阈值时,需对发送速率进行调节,并且调节的幅度应受到限制。实际上,由于在估计往返时间 RTT 与分组丢失率 p 时,已经对它们进行了平滑处理,由此估算出来的发送速率不会出现大的振荡。因此,本文在调节发送速率时未再附加任何限制,仅简单地将发送速率调节至估算值。关于在稳定状态网络环境下对调节速率的幅度定量分析,我们拟另文分析。调节发送速率伪代码如图3所示。

```

Calculate send rate(T) using RTT and p
if |T - Test| > Δ
    then Test ← T
if Test < threshold
    then warning user or terminate connect waiting feedback from receiver
    
```

图3 调节发送速率

4 实验分析

为了进一步分析和验证 EBCC 的性能(即首先要

TCP 流共存时,能公平地共享瓶颈链路带宽),我们在 ns 网络仿真环境下实现了上述算法,采用的网络拓扑结构如图4所示。其中端主机1向3发送 EBCC 流,端主机2向4发送 TCP Reno 流,两种流均为“无限”数据流。为保证更公平竞争瓶颈链路 R_1, R_2 带宽,我们假定所有与路由器相连的链路带宽与传播时延完全一样。实验有关参数设置如下:分组长度 $M=500$ byte,计算 RTT 的权值 $w=0.1$,历史丢失间隔数 $n=8$ 。实验中共建立

了8个连接,两种类型流各4个,如图5所示为两种流的平均吞吐量,可以看出,EBCC流的速率变化较平滑,而TCP流速率震荡相当剧烈。另外,在开始大约30毫秒EBCC流也出现震荡现象,这主要是因为EBCC采用了与TCP相似的慢启动机制所致。

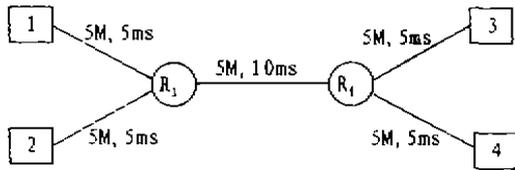


图4 网络拓扑结构

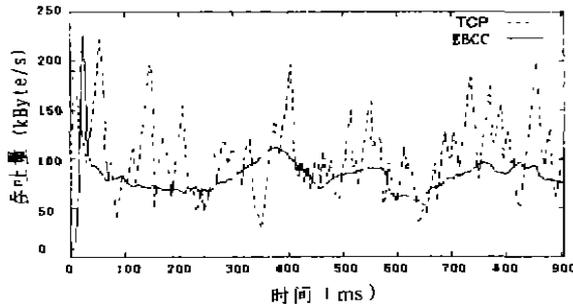


图5 TCP流与EBCC流吞吐量比较

为更加精确地描述EBCC流的性能,下面定义了4个度量单位。

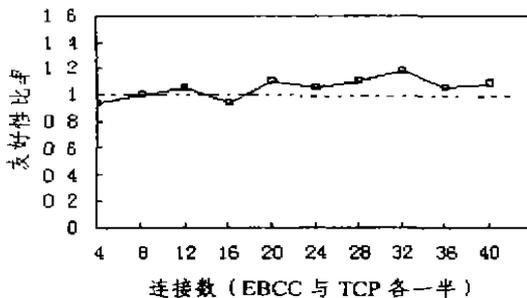


图6 EBCC友好性比率F

如图6所示为EBCC流的TCP友好性比率,图7为EBCC与TCP流在不同时间尺度下测量出的平滑系数(此处我们只选择了两个连接,即EBCC与TCP各一)。仿真实验表明,EBCC流的TCP友好性比率在1附近振荡,因此可以说EBCC流是友好的。同时EBCC流的平滑系数也比TCP流小得多,并在不同时间尺度下趋于稳定,由此可见EBCC流具有相对稳定的发送速率。

结束语 本文提出了一种基于方程的多媒体实时多媒体流拥塞控制机制EBCC,由于其控制方程基于

友好性比率 设 K_e, K_t 分别为EBCC连接与TCP连接数, K_e 个EBCC连接吞吐量分别为: $T_1^e, T_2^e, \dots, T_{K_e}^e$, K_t 个TCP连接吞吐量分别为: $T_1^t, T_2^t, \dots, T_{K_t}^t$, 则EBCC流与TCP流平均吞吐量为:

$$T_e = \frac{\sum_{i=1}^{K_e} T_i^e}{K_e} \text{ 和 } T_t = \frac{\sum_{i=1}^{K_t} T_i^t}{K_t} \quad (9)$$

于是,可定义EBCC友好性比率为:

$$F = T_e / T_t \quad (10)$$

平滑系数 我们采用了R. Jain定义的偏差系数(CoV)来描述连接发送吞吐量的变化,即平滑系数(SmCo)。设测量的时间尺度为 δ ,连接F从时间 t_0 到 $t_0 + \delta$ 的平均吞吐量定义为:

$$B_F, \delta(t_0) = (\text{bytes sent by } F \text{ between } t_0 \text{ and } t_0 + \delta) / \delta \quad (11)$$

则时间 $[t_0, T]$ 内的吞吐量序列为 $\{B_{F, \delta}(t_0 + i * \delta)\}_{i=1}^{(T-t_0)/\delta}$,由此可计算出当测量时间尺度为 δ 时,吞吐量的偏差系数为:

$$CoV_{F, \delta} = \frac{\sqrt{\frac{1}{(T-t_0)/\delta} \sum_{i=1}^{(T-t_0)/\delta} [B_{F, \delta}(t_0 + i * \delta) - B_F]^2}}{B_F} \quad (12)$$

式(12)中 B_F 为流F在时间 $[t_0, T]$ 内的平均吞吐量,即令(11)式中 $\delta = (T - t_0)$ 即可得 B_F ,偏差系数越小,连接速率变化就越小,即越平滑。

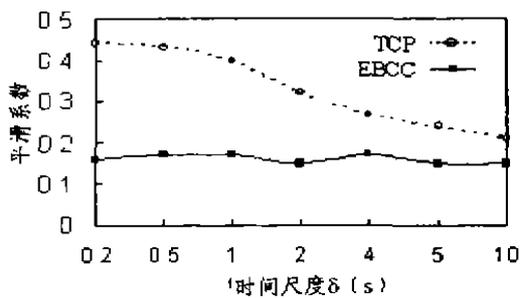


图7 两种流的平滑系数SmCo

TCP吞吐量响应函数,因此EBCC流能与TCP流公平共享链路带宽(即TCP友好的)。同时,在估计往返时延RTT与丢失率时,算法考虑了多媒体流的特点,对它们进行了特殊的平滑处理,因此能较为平滑地调节EBCC流发送速率。仿真实验表明EBCC较好地满足了实时多媒体流的要求。

需要指出的是,本文提出的EBCC仅适合单点投递应用,随着多点投递实时应用的不断增加,设计基于方程的多点投递实时应用拥塞控制是我们下一步的研究目标。

(下转第80页)

参考文献

- 1 石纯一,等. 基于 Agent 的计算. 知识科学与知识工程学术会议,1999.12
- 2 Hintikka J. Knowledge and belief. Cornell University Press, Ithaca, NY, 1962
- 3 Cohen P R, Levesque H J. Intention is Choice with Commitment. Artificial Intelligence, 1990, 42
- 4 Rao, Georgeff. Modeling rational Agents within a BDI-architecture. In: Proc of KR-91, San Mateo, CA, USA, 1991
- 5 Konolige K, Pollack M E. A Representationalist Theory of Intention. IJCAI-93, 390~395
- 6 Bell J. Changing Mental Attitudes, ECAI-94, Springer
- 7 Gaspar G, Coelho H. Where Do Intentions Come from? A Framework for Goals and Intentions Adoption, Derivation and Evolution. EPIA'95, Springer
- 8 d'Inverno M, Kinny D, Luck M, Wooldridge M. A Formal Specification of dMARS. In: M. P. Singh, A. S. Rao, and M. Wooldridge, eds. Intelligent Agents IV. Springer-Verlag Lecture Notes in AI Volume 1365, Feb. 1998
- 9 Wooldridge M. A Knowledge-Theoretic Semantics for Concurrent MetateM. In: J. Mueller, M. Wooldridge, and N. R. Jennings, eds. Intelligent Agents III, Springer-Verlag, 1997
- 10 Lesperance Y, Levesque H J, Reiter R. A Situation Calculus Approach to Modeling and Programming Agents. In: A. Rao and M. Wooldridge, eds. Foundations and Theories of Rational Agency. Kluwer, 1999
- 11 Shoham Y. Agent-oriented programming. Artificial Intelligence, 1993, 60, 51~92
- 12 Chanbr W. Proposition of formal semantics for multi-Agent systems. Computers and Industrial Engineering, 1999, 37, 453~456
- 13 Purvis M, Cranefield S. Agent modelling with Petri Net. In: Proc. Of CESA '96 IMACS Multiconference, IEEE-SMC, Lille, 1996
- 14 Cost R S, et al. Modelling Agent conversations with Colored Petri Nets. Third Conference on Autonomous Agents (Agents-99), Workshop on Agent Conversation Policies, Seattle, WA, May 1999
- 15 Jennings N R. Agent-Based Computing, Promise and perils. In: IJCAI-99, 1429~1436
- 16 徐晋晖. 多 Agent 模型与联盟机制研究. [清华大学博士论文]. 2000
- 17 Ferber J, Gutknecht O. Operational semantics of multi-Agent organization, in ATAL'99
- 18 Ferber J, Gutknecht O. A meta-model for the analysis and design of organizations in multi-agent systems. ICMA-98 Proceedings. IEEE 1998
- 19 Finn T, et al. KQML: An Information and Knowledge Exchange Protocol. In: Kazuhiro Fuchi and Toshio Yokoi, eds. Knowledge Building and Knowledge Sharing, Ohmsha and IOS Press, 1994
- 20 Labrou Y, Finn T. Semantics for an Agent Communication Language. In: Intelligent Agents IV: Agent Theories, Architectures and Languages. Michael Wooldridge, Munindar Singh and Anand Rao (editors.). Springer-Verlag, Lecture Notes in AI, Volume 1365, 1998
- 21 Van Eijk Roger M, de Boer Frank S, van der Hoek, Wiebe, Meyer John-Jules. Operational semantics for Agent communication languages [Report Utrecht University UU-CS-1999-08]
- 22 Wooldridge M. Verifiable semantics for Agent communication languages. In: Y. Demazeau, ed. Proc. of the Third Int Conf on Multi-Agent Systems (IC 多 Agent 系统98). IEEE Press, July 1998
- 23 Wooldridge M. Verifying that Agent implement a communication language. In: Proc. of the Sixteenth National Conf. on Artificial Intelligence (AAAI-99), Orlando, FL, July 1999
- 24 Lesperance Y, Levesque H J, Reiter R. A situation calculus approach to modeling and programming Agents. In: A. Rao and M. Wooldridge, eds. Foundations and Theories of Rational Agency. Kluwer, 1999
- 25 陆汝铃. 计算机语言的形式语义. 科学出版社, 1992
- 26 Zambonelli F, Jennings N R, Wooldridge M. Organisational Abstractions for the Analysis and Design of Multi-Agent Systems. In: Proc. 1st Int. Workshop on Agent-Oriented Software Engineering, Limerick, Ireland, 127~141

(上接第63页)

参考文献

- 1 Allman M, Paxson V, Stevens W. TCP Congestion Control. Request for Comments 2581, April 1999
- 2 Rejaie R, Handley M, Estrin D. An End-to-End Rate-based Congestion Control Mechanism for Realtime Streams in the Internet. In: Proc of INFOCOMM'99, 1999
- 3 Padhye J, Kurose J, Towsley D, Koodli R. A Model Based TCP-Friendly Rate Control Protocol. In: Proc. of NOSS-DAV'99, 1999
- 4 翟明玉, 吴国新, 顾冠群. Internet 可靠多点投递拥塞控制研究进展. 计算机研究与发展, 2000, 37(1): 9~16
- 5 Mahdavi J, Floyd S. TCP-Friendly Unicast Rate-based Flow Control. Note sent to end 2end-interest mailing list, Jan 1997
- 6 Floyd S, Fall K. Promoting the Use of End-to-End Congestion Control in the Internet. IEEE/ACM Transactions on Networks, 1999, 7(4): 458~472
- 7 Padhye J, Firoiu V, Towsley D. Modeling TCP Throughput: A Simple Model and its Empirical Validation. SIGCOMM Symposium on Communications Architectures and Protocols, 1998(Aug): 303~304
- 8 Mogul J, Deering S. Path MTU Discovery. Request for Comments 1191 Nov. 1991
- 9 Jain R. The Art of Computer Systems Performance Analysis. John Wiley and Sons, 1991