基于流体流的 TCP Vegas/RED 分析研究*)

邓晓衡　陈志刚　张连明

(中南大学信息科学与工程学院　长沙410083)

摘　要　本文采用基于流体流的方法对TCP Vegas协议进行研究分析，同时结合网络结构和主动队列管理机制RED算法进行建模分析，通过网络仿真软件NS2和数学软件matlab分别对TCP Vegas/RED运行情况进行模拟，其结果分别作为协议实验实际值和理论估计值进行比较，实验结果表明模型与实验结果能匹配一致。最后指出本文方法的局限性和进一步研究的方向。

关键词　拥塞控制，TCP Vegas，RED

Analysis on TCP Vegas/RED Based on Fluid-flow and Packets

DENG Xiao-Heng　CHEN Zhi-Gang　ZHANG Lian-Ming

(College of Information Science and Technology, Central South University, Changsha 410083)

Abstract　This paper analyzes fluid-based TCP Vegas algorithm with very low probability of fragment marking, models the network and active queue management mechanism of RED, and designs various scenarios to simulate through matlab and network simulater software NS2. Simulation results show that model and simulation match very well. Lastly, the limitation of the approach and further research issues are pointed out.

Keywords　Congestion control, TCP Vegas, RED

1　引言

传输控制协议(TCP)提供了端到端的可靠的带拥塞控制的互联网服务。自从TCP产生以来，研究者对于拥塞控制进行了不懈的研究并取得了大量的成果[1,2]。最初的TCP Tahoe采用慢启动和拥塞避免机制，TCP Reno实现了丢包快速重传和从拥塞状态中快速恢复，TCP SACK通过应答消息返回发送端更完全的信息，可确定哪一个包丢失，解决了在一个窗口中出现多个包丢失的情况。以上方法都是通过往返时间(Round Trip Time，RTT)与重传超时时间(Retransmission Timout，RTO)比较判断网络是否拥塞，其中TCP Reno成为目前互联网中广泛使用的一种协议。TCP vegas[1]是一种TCP Reno的改进算法，采用一些新技术，降低了分组的丢弃率，显著提高了网络的吞吐量达37～71%[1]。其基本思想是通过发送端测量到的包的RTT变化情况推断出网络的拥塞程度从而调整拥塞窗口(cwnd)的大小，如果RTT增大，源端减少cwnd，从而减少传输速率。

网络中间节点(如路由器)分组队列缓存管理一般采用主动式缓存管理(Active Queue Management，AQM)，主要有FIFO(First In First Out)、RED(Random Early Detection)、DDR(Deficit Round Robin)算法。其中，Floyd提出的随机早期检测RED算法[3]是网络队列管理的主要算法。RED算法的思想是：网络转发结点如路由器缓存区通过一个低通滤波函数计算缓存区的平均队列长度作为网络拥塞程度的度量，缓存的平均队列长度达到一个门限值时就对到达的分组按照一定的概率进行标记或丢弃，随着队列长度增加丢弃概率线性增长，长度到达另一个门限值时，丢弃所有到达的分组，避免网络的长时间拥塞。RED算法中缓存区是一个FIFO的分组队列，队列长度一般以分组的个数为单位。

TCP Vegas协议作为一种改进了源端窗口机制TCP协议，能自适应地调节源节点的发送速率，达到适合的网络带宽，具有诸多优点，极有可能作为互联网广泛使用的端到端的传输控制机制。对于目前互联网广泛配置的协议TCP Reno的建模分析已经有大量的研究结果[4～6]，对于理解其工作的理论机制，为进一步改进其局限性提供了依据。而对于TCP Vegas的建模一般都集中在较粗糙的基于数据包水平[7]，文[8]利用控制理论基于流体流对TCP Vegas进行了建模分析；H. L. Steven通过对偶理论对TCP Vegas进行建模[9]，虽对性能、公平性和稳定性进行了研究，并结合主动队列管理REM机制进行了分析，但其主要基于定性的研究，没有考虑分组丢弃情况。

本文通过对TCP Vegas和RED算法建立数学模型，基于流体流分析和比较TCP Vegas处于平衡状态属性和动态特性，对网络的性能、公平性、稳定性进行分析；并通过仿真实验验证相关理论分析的结论。本文第2节对网络和相关协议建模，第3节通过仿真实验对模型进行验证，第4节对模型相关属性和特点进行分析，最后总结全文。

2　基本模型与分析

本节考虑在由多个端节点和多个中间节点(路由器)构成多链路的复杂网络环境中，对TCP Vegas协议和主动队列管理机制RED建模，描述基于流体流的TCP Vegas的行为。本文的研究采用了文[4]，对TCP Reno基于流体流的分析有些类似，主要以TCP Vegas的拥塞避免阶段的行为为研究对

*)基金项目：国家自然科学基金(10375024)；湖南省自然科学基金(02JJY2097)。邓晓衡　博士研究生，主要研究方向为Web缓存技术，流量管理，网络拥塞控制，网络优化。陈志刚　博士，教授，博士生导师，主要研究方向为网络计算与分布式处理。张连明　博士研究生，主要研究方向为网络行为学，网络性能优化。

象。

2.1 网络建模

我们考虑一种通常情况，网络由多个终端节点和中间节点连接构成，分别用终端节点和连接的集合表示 L 和 N。因此，集合 L 中各连接具有有限带宽容量 $c=(c_l, l\in L)$，源端节点集合 N 各节点以序号 i 标记，网络连接 L_i 可以定义为一个 $L\times N$ 的矩阵 R。

$$R_{li}=\begin{cases}1 & \text{if } l\in L_i\\ 0 & \text{if } l\notin L_i\end{cases}$$

源节点 i 发送数据的往返时间为 $RTT_i(t)$，$RTT_i(t)$由传输过程的传输延迟和连接 L_i 上所有中间节点的排队延迟之和共同组成，故有，

$$RTT_i(t)=propi(t)+\sum_l R_{li}\frac{q_{l\_cur}(t)}{c_l} \tag{1}$$

其中，$prop_i(t)$为传输延迟，$q_{l_cur}(t)$为连接 l 的瞬时队列长度。

对网络的基本参数建模以后，我们考虑每个连接 l 在 t 时刻的分组标记概率为 $p_l(t)$(中间节点对缓存区的分组按一定概率进行标记(ECN 位置1)，或丢弃，本文采用分组标记策略)，中间节点分组队列管理可以采用不同的方式，本文采用 RED 算法，2.2节将对其建模。

分组通过连接 l 的标记概率 $p_l(t)$可以反映网络拥塞程度，反馈给源节点；源节点根据相应的窗口调整机制，改变发送窗口的大小，本文中为 TCP Vegas。定义源节点 i 在 t 时刻接总的分组标记概率 $P_i(t)$，有

$$P_i(t)=\sum_l R_{li}p_l(t-T_{li}^{b}(t)) \tag{2}$$

其中 $T_{li}^{b}(t)$为从连接 l 将应答反馈到 i 的延迟。

一般源节点 i 在 t 的发送速率定义为发送窗口 $w_i(t)$与往返时间 RTT_i 的比值，即

$$x_i(t)=\frac{w_i(t)}{RTT_i(t)} \tag{3}$$

那么，连接 l 总的分组到达速率 $X_l(t)$为

$$X_l(t)=\sum_i R_{li}\frac{w_i(t-T_{li}^{f}(t))}{RTT_i(t-T_{li}^{f}(t))} \tag{4}$$

其中，$w_i(t-T_{li}^{f}(t))$为 $T_{li}^{f}(t)$时间前源节点的发送窗口的大小，$T_{li}^{f}(t)$为分组从源节点 i 转发到连接 l 的延迟。

对于所有的连接 $l\in L_i$，由总的分组到达速率 $X_l(t)$和各连接的带宽容量 c_l，结合式(1)，我们可以计算出连接的瞬时队列长度 $q_l(t)(b_l(t)>0)$由下面微分方程式决定

$$\begin{aligned}\frac{dq_{l\_cur}(t)}{dt}&=X_l(t)-c_l=\sum_i R_{li}\frac{w_i(t-T_{li}^{f}(t))}{RTT_i(t-T_{li}^{f}(t))}-c_l\\&=\sum_i R_{li}\frac{w_i(t-T_{li}^{f}(t))}{proc_i+\sum_k R_{ki}\times q_{l\_cur_k}(t-T_{li}^{f}(t))/c_k}-c_l\end{aligned} \tag{5}$$

其中，k 属于连接 i 所经过的链路的集合，分组排队延迟为各链路延迟的累积。微分方程式表明队列在总的分组到达速率超过连接带宽容量时成立，否则中间节点处理速率大于分组到达速率，队列将为空。

2.2 AQM 建模

主动队列管理是中间节点的一种主要的缓存区管理机制，而 IETF(互联网工程任务组)把 RED[3]作为唯一的分组标记/丢弃候选策略。因此，本文只对 RED 建模分析。

RED 算法中的基本原理如图1所示，在链路 l 上的路由器缓存区平均队列长度小于一门限值分组全部通过，大于另一门限值则分组全部标记/丢弃，在两门限值之间分组丢弃概率由0线性增加到一预设最大分组标记/丢弃概率，因此，分组标记概率与平均队列长度的相互关系为

$$p_l(t)=\begin{cases}0, & 0<q_{l\_ave}(t)\leqslant q_{l\_min}\\ p_{l-max}\dfrac{q_{l-ave}(t)-q_{l\_min}}{q_{l\_max}-q_{l-min}}, & q_{l\_min}<q_{ave-l}(t)<q_{l-max}\\ 1, & q_{l\_ave}(t)\geqslant q_{l\_max}\end{cases} \tag{6}$$

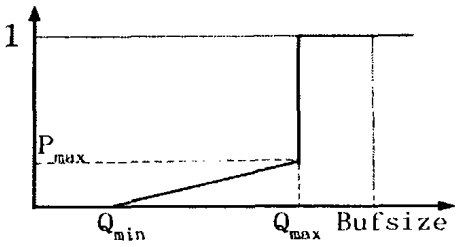

图1 RED 算法

RED 算法的各主要参数参见表1。

表1 RED 算法的主要参数

参数	功能
q_{l_ave}	平均队列长度
q_{l_min}	下限阈值，超过该值分组全部通过转为按一定概率丢弃/标记
q_{l_max}	上限阈值，超过该值分组将被被全部丢弃/标记
P_{l_max}	最大丢弃/标记分组概率，对应 ql-max 处的概率
q_{l_cur}	瞬时队列长度
P_l	分组丢弃/标记概率
w_{l_q}	q_{l_ave}计算权值，反映 q_{l_cur}对 q_{l_ave}的影响程度，$0<w_{l\_q}<1$

根据网络已经计算出瞬时队列长度 q_{l_cur}，在 RED 算法中平均队列长度 q_{l_ave}为基于对队列长度每隔时间 δ 的取样值的指数加权移动平均值[4]，即

$$q_{l\_ave}((k+1)\delta)=(1-w_{l-q})q_{l-ave}(k\delta)+w_{l-q}\times q_{l-cur}(k\delta) \tag{7}$$

进一步转化为微分方程式：

$$\frac{dq_{l-ave}}{dt}=Aq_{l-ave}(t)+Bq_{l-cur}(t) \tag{8}$$

对于一个数据采样系统[11]$q_{l_ave}(t_{k+1})$为：

$$q_{l\_ave}(t_{k+1})=e^{A(t_{k+1})}x(t_k)+\int_k^{k+1}e^{A(t_{k+1})-T}BdTq_{l-cur}(t_k) \tag{9}$$

微分方程式与离散时间系统在采样点可精确匹配，因此，比较(6)，(9)式，有

$$1-w_{l-q}=e^{A\delta}$$

即，

$$A=\frac{\log_e(1-w_{l-q})}{\delta}=-B$$

故有

$$\frac{dq_{l-ave}}{dt}=\frac{\log_e(1-w_{l-q})}{\delta}(q_{l\_ave}(t)-q_{l-cur}(t)) \tag{10}$$

值得说明的一点是 δ 为采样时间间隔，则 $1/\delta$ 为采样频率，可以看作分组传输速率，δ 的取值非常重要，我们取链路的带宽容量 $c_l=1/\delta$，则有：

$$\frac{dq_{l\_ave}}{dt}=c_l\log_e(1-w_{l-q})(q_{l-ave}(t)-q_{l\_cur}(t)) \tag{11}$$

2.3 TCP Vegas 建模

TCP 协议实现可靠的端到端数据传输，经过不断改进形

成了许多的变体，一般采用 AIMD(Additive Increase Multiplicative Decrease)机制保证网络的稳定可靠地运行，其中 TCP Reno 目前在互联网使用比较广泛，TCP Vegas[1]在拥塞避免阶段的实现机理较其他算法有较大区别，引言部分进行了简单描述。本节将基于前述的网络模型对 TCP Vegas 协议组合 RED 算法基于流体流建立数学模型，建模的主要对象为 TCP 协议处于稳态状态，即拥塞避免阶段。

先简要讨论 TCP Reno 协议[10]，在 t 时刻，源节点 i 数据传送速率为 $x_i(t)$ *packets/sec*，T_i 为往返时间，假定全部分组成功发送，全部返回应答，接收到应答的速率为 $x_i(t-T_i(t))$。如果网络出现一定程度的拥塞，分组中有概率为 $P_i(t)$ 部分被丢弃/标记，那么源节点分组被丢弃/标记的速率应为 $x_i(t-T_i(t))P_i(t)$，而每接收一个带有分组被标记信息的应答，发送窗口 $w_i(t)$ 减少为 $w_i(t)/2$，因此，窗口减少的速率为 $x_i(t-T_i(t))p_i(t)w_i(t)/2$。也可推断出源节点成功传送分组的概率为 $(1-P_i(t))$，速率为 $x_i(t-T_i(t))(1-p_i(t))$，源节点每接收一个分组成功传送的应答，发送窗口增加 $1/w_i(t)$，因此，窗口增长的平均速率为 $x_i(t-T_i(t))(1-p_i(t))\frac{1}{w_i(t)}$；总体平均起来，TCP Reno 协议发送窗口的变化规律可由下面微分方程式表示：

$$\frac{dw_i}{dt}=x_i(t-T_i(t))\left((1-P_i(t))\frac{1}{w_i(t)}-P_i(t)\frac{w_i(t)}{2}\right) \quad (12)$$

对于 TCP Vegas 协议分析方法与 TCP Reno 类似，不过结合了流体流和窗口进行分析。该模型是假定分组标记的概率非常小，使得相邻两个分组标记的间隔，TCP Vegas 处于一种稳定运行状态，当分组丢弃概率高时，TCP Vegas 演变为 TCP Reno 协议。

首先，我们讨论一个分组被标记/丢弃的传输过程，如图2所示。

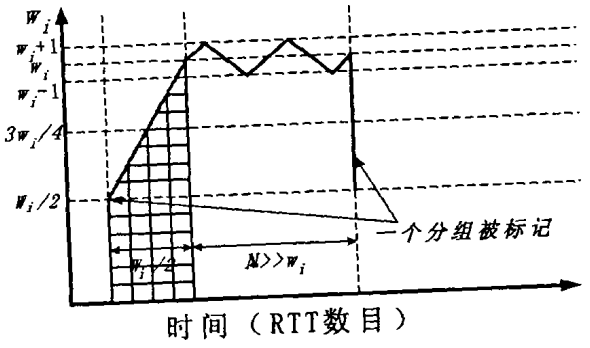

图2 一个分组被标记的窗口变化

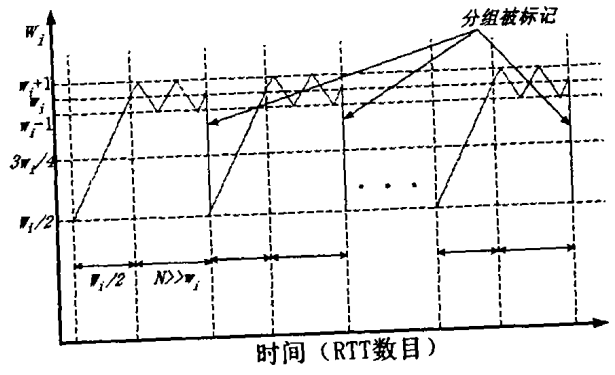

图3 多个窗口分组被标记窗口变化

当网络出现拥塞，中间节点的采用 RED 算法标记一个分组，当源节点 i 接收到该分组被标记的应答，将发送窗口 w_i 减少为 $w_i/2$(假定此时，拥塞已经解除，若没有解除此过程将继续下去)，接下来，源节点 i 每接收一个分组成功发送的应答，发送窗口 w_i 增加 $1/w_i$，窗口增加过程经历约 $W_i/2$ 个 RTT 时间间隔，$W_i/2$ 为稳定运行时的拥塞窗口，发送窗口由 $W_i/2$ 增加到 W_i，传送的分组数为梯形的面积，发送窗口的大小表示为分组的个数。然后，发送窗口维持在 W_i 附近振荡，直到网络再次发生拥塞，重新进入拥塞避免阶段，当有多个分组被标记情形如图3所示，源节点的发送窗口将随着接收到分组被标记的应答而不断重复图2所示的过程，其中 W_i 大小并不一定如图3所示在同一水平上，为便于描绘而设定大小相同。

我们考虑前述的网络环境，假定 t 时刻源节点 i 的发送速率为 $x_i(t)$，网络存在一定程度的拥塞以概率 $P_i(t)$ 标记分组，分组标记的速率为 $x_i(t-T_i)\ P_i(t)$，因此，窗口减少的速率为 $x_i(t-T_i)\ P_i(t)w_i(t)/2$，接着，源节点 i 每接收一个成功发送的应答，发送窗口增加 $1/w_i$，根据前面分析，一个分组标记将导致 $W_i(t)/2$ 个窗口递增过程，因此，有速率增加的概率为 $\frac{3p_i(t)W_i(t-T_i)^2}{8}$，发送窗口增加的速率可以表示为 $x_i(t-T_i)\frac{3p_i(t)W_i(t-T_i)^2}{8w_i(t)}$，恰好为图2中所示的梯形的面积；此后的分组传送过程为速率保持适合的网络传输带宽附近变化，窗口变化可表示为：

$$x_i(t-T_i(t))\left(1-\left(1+\frac{3W_i(t-T_i)^2}{8}\right)\times p_i(t)\right)\times \mathrm{sgn}(z(t))/w_i(t)$$

其中 $sgn(z)$ 为一符号函数，z 为正数结果为1，z 为负数结果为 -1，z 为0结果为0。因此，全过程窗口变化情况可由式(13)决定，且必须满足条件 $\frac{3p_i(t)W_i(t-T_i)^2}{8}+p_i\leqslant 1$。

$$\begin{aligned}\frac{dw_i}{dt}=&x_i(t-T_i(t))\left(1-\frac{3W_i(t-T_i)^2}{8}p_i(t)-p_i(t)\right)\\&\times sgn\left(z(t)+\frac{3W_i(t-T_i)^2}{8}p_i(t)\right)/w_i(t)\\&-x_i(t-T_i)\frac{w_i(t)}{2}p_i(t)\end{aligned} \quad (13)$$

根据文[1]，TCP Vegas 具体算法如下，设定 $base\_rtt$ 为所有测得 RTT 中最小的，那么在网络没有发生拥塞的情况下，有期望得到的带宽：$B_{expected}=cwnd/base\_rtt$，实际带宽：$B_{actual}=cwnd/rtt$，两者之差：$diff=B_{expected}-B_{actual}$，同时定义两个门限阈值 $\alpha,\beta(\alpha<\beta)$。若 $diff<\alpha/base\_rtt$，表明发送速率小于可用带宽，$cwnd$ 增加1，若 $diff>\beta/base\_rtt$，表明发送速率大于可用带宽，$cwnd$ 减少1，若 $\alpha/base\_rtt<diff<\beta/base\_rtt$，表明发送速率与可用带宽趋于一致，$cwnd$ 保持不变，为了研究方便进行适当简化，令 $\zeta=\alpha=\beta$。有拥塞窗口变化关系如式(14)。

$$cwnd(t)=\begin{cases}cwnd(t-T)+1, & \text{if } diff<\zeta/base\_rtt\\ cwnd(t-T), & \text{if } diff=\zeta/base\_rtt\\ cwnd(t-T)-1, & \text{if } diff>\zeta/base\_rtt\end{cases} \quad (14)$$

$$diff=\frac{cwnd}{base\_rtt}-\frac{cwnd}{rtt}\ (base\_rtt \text{ 为测得的最小 } rtt)$$

当一个连接启动后，随着时间的推移拥塞窗口将趋向一个稳定值，此时有

$$diff=\frac{cwnd}{base\_rtt}-\frac{cwnd}{rtt}=\zeta/base\_rtt \quad (15)$$

故有 TCP Vegas 协议运行于稳态时，速率为

$$X=cwnd/rtt=\zeta/(rtt-base-rtt) \tag{16}$$

在前述分析中的图2，图3以及式(13)中的 W_i 即为式(14)中的拥塞窗口 $cwnd$，只不过式(13)，(14)，(15)中没有考虑实际网络多个链路和多个会话的情况。结合前述的网络模型以及拥塞窗口的变化，式(13)中的符号函数 $sgn(z(t))$ 中的 $z(t)$ 值可以由下式决定

$$z(t)=\frac{\zeta}{base\_rtt}-diff=\frac{\zeta}{base\_rtt}-\frac{cwnd(t)}{base\_rtt}+\frac{cwnd(t)}{rtt_i(t)} \tag{17}$$

我们再回顾 TCP Vegas 协议的拥塞避免机制，当拥塞发生之后，拥塞窗口减半后的变化过程遵循式(14)，因此式(13)可转化为：

$$\frac{dw_i}{dt}=\frac{x_i(t-T_i(t))(1-p_i(t))\times sgn(z(t))}{w_i(t)}-x_i(t-T_i)\times p_i(t)\frac{w_i(t)}{2} \tag{18}$$

其中，$z(t)$ 由(17)式决定。

3 模型验证

基于上述分析，对 TCP Vegas/RED 数学模型利用数学软件 matlab 加以实现，获得理论估计值；通过网络仿真软件 NS2[13]进行模拟实验，作为协议运行的实际值，对实际值和理论值进行比较。我们设计了多个实验方案，在不同网络环境下模拟，验证我们的数学分析模型。实验结果表明数学分析模型能较精确反映 TCP Vegas 的行为。为正确有效反映数学模型的情况，用 matlab 建模计算理论值的输入初值采用 NS2仿真实验时收集的数据，包括 $x_i(t)$，$p_i(t)$，$rtt_i(t)$，但由于模型是关于 TCP Vegas 处于稳定运行(拥塞避免)阶段，因此去除掉从 NS2收集到的慢启动阶段数据(前7秒)。

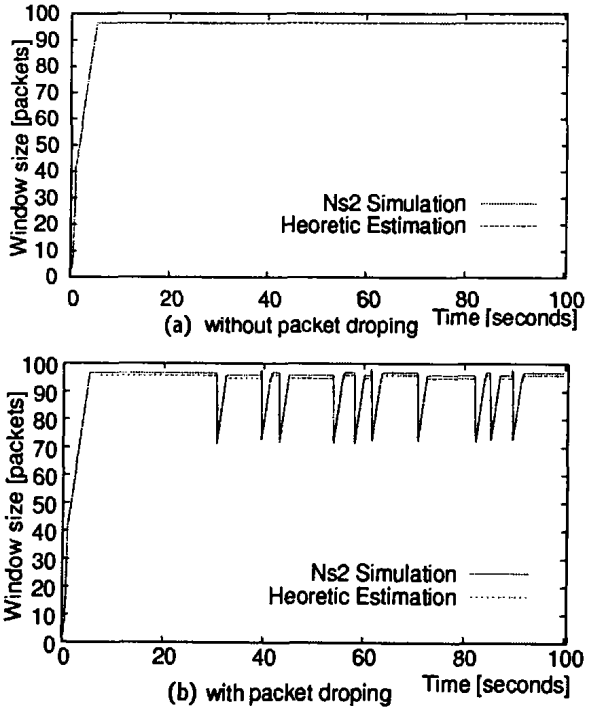

图4 单连接单链路情况

在实验1中只有一个连接独享一个瓶颈链路，分别设计存在丢包和不丢包两种情况进行实验，瓶颈链路容量为5Mb，分组大小为500bytes。队列管理机制为 RED，RED 算法中的各参数分别为 $P_{max}=0.1$，$q_{min}=20$pks，$q_{max}=200$pks，$w=0.0001$，RTT=35ms，会话为传输大量数据持久性的 ftp 业务，其中 NS2仿真的拥塞窗口如图4(a)所示，实际值与理论值维持在96pkts 附近振荡，此时队列几乎为空，长度小于2，反映了 TCP Vegas 能调整窗口大小适合网络带宽，维持高吞吐量，丢包率几乎为0。当网络以 rate=0.00001的概率丢包时，如图4(b)。

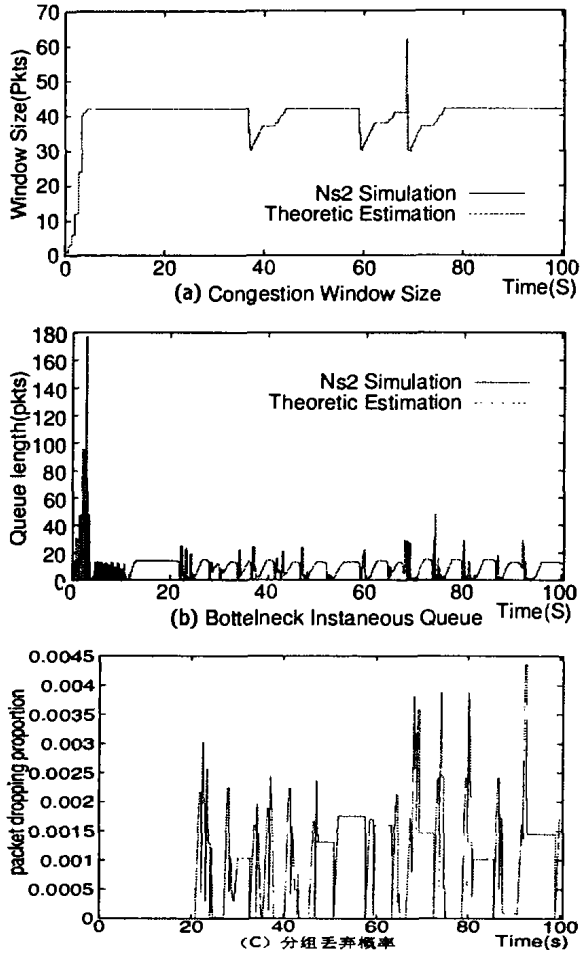

图5 多个连接共享一个瓶颈链路

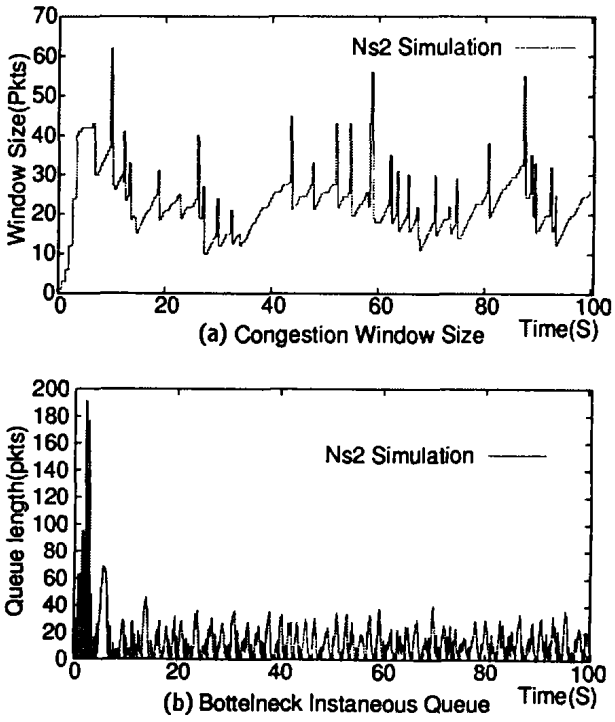

图6 网络分组丢弃率很高情况

实验2中设计了多个会话共享单个瓶颈链路的环境。这个

实验中我们采用了网络模拟中广泛使用的哑铃结构，一个瓶颈链路由16个会话共享其带宽。瓶颈链路容量为15Mb，其他链路均为100Mb 和去尾(DropTail)的丢包机制，往返延迟 RTT＝160ms。实验中对瞬时队列长度 qcur，拥塞窗口 cwnd 的理论值和实际值进行了比较。如图5(c)显示了分组的丢弃概率，图5(a)显示了在单个链路情况 TCP Vegas 能调节源节点的发送窗口到一个适当的值，确保发送速率与链路的可用带宽匹配，而不是像 TCP Reno 那样不断循环地线性增加窗口，然后将发送窗口减半。

实验3中采用实验完全相同的网络环境，连接数增加到32，系统处于严重过载状态，分组丢弃率较高，TCP Vegas 协议演变为 TCP Reno，前面模型不能适用，由图6(a)可以看出，拥塞窗口处于频繁的抖动之中，无法维持在一稳定值。瓶颈链路的队列长度，如图6(b)也由于窗口的抖动而振荡，导致利用率下降。

4 稳定状态分析

实验仿真表明流体流模型能有效反映 TCP Vegas 协议的行为，现在我们再利用模型分析其平衡状态的行为。根据式(18)，设定微分等式值为0，基于前面分析我们已经有，

$$0=x_i(t-T_i(t))\left(\frac{(1-p_i(t))\times sgn(z(t))}{w_i(t)}-p_i(t)\frac{w_i(t)}{2}\right) \tag{19}$$

此时发送速率显然不为0，只能是第二项为0，而要维持窗口变化为0，则有 $sgn(z(t))$ 为0，那么 $p_i(t)w_i(t)2$ 也必须为0，而 $w_i(t)$ 不为0只能是分组丢弃概率为0。因此，在稳态下，TCP Vegas 的分组丢弃概率为0，根据(17)式有稳态窗口大小为

$$w^*=\frac{\zeta\times RTT^*}{RTT^*-Base\_rtt} \tag{20}$$

结论 本文对网络和 RED 算法分别建模，同时基于流体流对 TCP Vegas 算法进行分析研究，获得了在分组标记概率较小的情况下网络行为的模型，对基于分组水平的流量模型的改进。该模型仍还有局限性，其适用条件受到一定约束，要求分组标记概率很小，因此，在网络拥塞较严重，即分组标记概率较大的条件下，其对网络行为建模是进一步研究的方向。

参 考 文 献

1 Brakmo L S, Perterson L L. TCP Vegas: End-to-End Congestion Avoidance on a Global Internet. IEEE Journal on Selected Areas in Communication, 1995, 13(8)

2 Jacobson V. Congestion Avoidance and Control, ACM Computer Communications Review, 1988, 18(4): 314～329

3 Floyd S, Jacobson V. Random Early Detection Gateways for Congestion Avoidance [J]. IEEE/ACM Transactions on Networking, 1993, 1(4): 397

4 Misra V, Gong Weibo G, Towsley D. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED. In: Proc. of ACM/SIGCOMM, 2000

5 Cardwe N, Savage S, Anderson T. Modeling TCP latency. In: Proc. IEEE INFOCOM, Tel-Aviv, Israel, March 2000. 1742～1751

6 Misra V, Gong W, Towsley D. Stochastic differential equation modeling and analysis of tcp-windowsize behavior. In: Proc. PERFORMANCE 99, Istanbul, Turkey, Oct. 1999

7 Babis S C, Vernon M K. Modeling the Throughput of TCP Vegas. In: Proc. ACM SIGMETRICS 2003 Int'l. Conf. on Measurement and Modeling of Computer Systems (Sigmetrics 2003) San Diego, June 2003

8 Takagaki K, Ohsaki H, Murata M. Analysis of a window-based flow control mechanism based on TCP Vegas in heterogeneous network environment. IEICE Transactions on Communications, 2002. 89～97

9 Low S H, Peterson L, Wang L. Understanding Vegas: A Duality Model. Journal of ACM, March 2002, 49(2): 207～235

10 Low S H, Paganini F, Doyle J C. Internet congestion control: an analytic perspective. IEEE Control System Magazine, Feb. 2002

11 奥斯特隆姆 K，威顿马克 B J. 计算机控制系统理论与设计. 王晓陵，等译. 北京：北京科技出版社，1987

12 Sastry N, Lam S S. A Theory of Window-Based Unicast Congestion Control. In: Proc. IEEE ICNP 2002, Paris, November 2002

13 NS project. the network simulator-ns-2 (Web site). http://www.isi.edu/nsnam/ns/

(上接第58页)

的要求，此时 TCP/IP 的弊端也逐渐暴露出来，层次结构的特点使得把新的技术和标准引入现有网络中出现了困难，只能在原有层次结构中进行修补始终很受限制，不能很好地满足新服务的需求，另外由于几个网络协议层的冗余操作也使网络性能下降。Braden 等人提出的无层次的基于角色的网络体系结构允许将现有的较大的协议如 IP、TCP 等进行模块化重组，使它们变成许多小的单元从而与各种特定的功能对应起来，具有简洁、高效、可扩展和能够更好地实现 QoS 和安全保障机制等优点。

参 考 文 献

1 Clark D, Tennenhouse D. Architectural Considerations for a New Generation of Protocols [C]. In: Proc. of Sigcomm-90, 1990. 200～208

2 Tennenhouse D, Wetherall D. Towards an Active Network Architecture [J]. Computer Communication Review, 1996, 26(2)

3 Lazar A. Programming Telecommunication Networks [C]. USA: In: Proc. 5 th Intl. Workshop on Quality of Service 1997, IWQOS'97: 3～24

4 Paxson V, Allman M, Dawson S, et al. Known TCP Implementation Problems [S]. RFC2525, 1999, 3

5 Mustafa A, Hassan M, Jha S. Design and Performance of a rate control feedback architecture for TCP/IP Network [C]. Phoenix, Arizona: In: 21st IEEE Intl. Performance Computing (ISBN 0-7803-7371-5/02) and Communications Conf. (IPCCC 2002), 2002-04-03 to 2002-04-05

6 Stewart R, Metz C. SCTP: New Transport Protocol for TCP/IP [J]. IEEE Internet Computing, 2001(6): 64～69

7 Engel R, Kandlur D, Mehra A, Saha D. Exploring the Performance Impact of QoS Support in TCP/IP Protocol Stacks [C]. In: Proc. of IEEE INFOCOM98, 1998. 883～892

8 Boecking S. 面向对象的网络协议[M]. 北京：机械工业出版社，2000

9 Boecking S, et al. A Run-Time System for Multimedia Protocols [C]. Fourth Intl. Conf. on Computer Communications and Networks (ICCCN'1995), 1995, 9: 178～185

10 Siemens A G. Performance and Software Evaluation of the Modular TIP Communication System [C]. USA: 5th Intl. Conf. on Computer Communications and Networks, 1996, 10

11 Burkhard S. Configuration of Protocols in TIP [R]: [University of Cambridge Computer Laboratory's Technical Report]. 1995, 6: 368～373

12 金诚，杜勇，曾家智. 基于对象的动态协议配置[J]. 计算机应用，2001

13 Braden B, Faber T, Handley M. From Protocol Stack to Protocol Heap - Role-Based Architecture [R]. First Workshop on Hot Topics in Networking, 2002, 10