

# 多个 TCP 连接的拥塞丢包模型

陈相宁<sup>1</sup>,王 京<sup>1</sup>,程时昕<sup>2</sup>,陈 明<sup>2</sup>,蒋朱成<sup>2</sup>

(1. 清华大学微波与数字通信国家重点实验室,北京 100084;2. 东南大学移动通信国家重点实验室,江苏南京 210096)

**摘 要:** 本文提出了有关 TCP 连接的拥塞丢包分析模型.网络瓶颈一般承载许多 TCP 连接,瓶颈处不可避免的拥塞和缓存溢出,是导致网上丢包的主要原因.网络瓶颈处的行为很大程度上左右了网络性能.本文的模型估计了存在大量持续 TCP 连接时,网络瓶颈的丢包概率和网络传输中断概率,给出了对实际网络的良好近似.这对于研究 TCP 对网络性能的影响,提出改善网络性能的新算法,以及分析(从长远来看)TCP 还应做哪些改进,都是非常有用的.

**关键词:** 多 TCP 连接;拥塞控制;丢包模型

**中图分类号:** TN915 **文献标识码:** A **文章编号:** 0372-2112 (2002) 07-0990-05

## Packet-Dropping Model of Multiple TCP Connections

CHEN Xiang-ning<sup>1</sup>,WANG Jing<sup>1</sup>,CHENG Shi-xin<sup>2</sup>,CHEN Ming<sup>2</sup>,JIANG Zhu-cheng<sup>2</sup>

(1. State Key Lab on Microwave and Digital Communications, Tsinghua University, Beijing 100084, China;

2. National Mobile Communications Research Lab, Southeast University, Nanjing, Jiangsu 210016, China)

**Abstract:** In this paper, we propose an analytical model for the dropping behavior at the network bottleneck where large number of TCP (Transmission control protocol) connections exist. The inevitable congestion and buffer overflow at the bottleneck is the main reason of the packet loss in networks, which has great influence on the network performance. This model predicts the packet loss rate and transmission interruption rate of the sustained TCP connections sharing the bottleneck link. If most packet-losses are caused by network congestion, our models can give close approximations to the real networks that always contain many connections. The analytical model is useful in studying the effects of TCP, finding new effective algorithm of congestion control, and determining how TCP needs to be modified in the long term for efficient operation over emerging networks.

**Key words:** multiple TCP connections; congestion control; packet-dropping model

### 1 引言

互联网的流量特性很大程度上受到 TCP (Transmission Control Protocol) 协议的影响. TCP 协议被设计用于在不可靠的 IP (Internet Protocol) 传输上建立可靠的连接. 尽管光纤线路的采用和光纤技术的发展已使传输差错大大降低, 但是网络流量的增长经常超过网络处理能力的提高, 拥塞和缓存溢出<sup>[1]</sup> 仍然是 IP 传输需要考虑的因素. 在这种情况下, 研究拥塞和缓存溢出<sup>[2]</sup> 对 TCP 性能的影响是很有必要的.

文献主要通过仿真<sup>[3]</sup>、实时测量<sup>[4]</sup>和近似分析<sup>[5]</sup>来研究 TCP 传输性能. 参考文献[6]仿真研究了 TCP-Tahoe 和 TCP-Reno 单向文件传输的流量和公平性问题, 文献[7]通过对比仿真和实测的结果, 表明仿真研究的结果在大多数情况下是可信的. 然而受到仿真系统资源和仿真时间的限制, 仿真分析一般不适用于复杂多变的应用环境, 限制了研究结果的应用范围.

文献[8]分析了无线链路中的 TCP 性能, 文献[9]考虑了双向传输对拥塞控制的影响, 文献[10]研究了 ABR 业务承载

TCP 流量时端到端的特性. 这些简单分析主要针对标记的单个连接, 并假定该连接独享固定的缓存和链路带宽. 实际网络同时支持许多 TCP 连接, 并且为了提高传输效率, 这些连接共享链路带宽和节点缓存, 因而与理想的单个连接相比有较大的差别. 多个 TCP 连接的特性对网络规划具有更重要的意义.

本文针对 Reno 版 TCP 协议建立了丢包模型, 导出了多个 TCP 连接存在时的链路丢包数表达式, 并通过仿真进一步研究了缓存大小、到达流量特性和节点流控策略等对丢包性能的影响. 运用本文提出的建模方法, 可以建立其它版本 TCP 协议的分析模型; 本文的结果能够用于分析网络性能, 并为网络规划提供参考.

### 2 TCP 协议要点和网络瓶颈流量到达模型

TCP 协议对单一 TCP 连接行为进行了规范. 它规定了慢启动和拥塞避免两个阶段, 通过改变拥塞门限和发送窗口调节发送速率, 通过滑动窗口机制实现差错和拥塞控制<sup>[11]</sup>. Reno 由 BSD 发布于 1990 年, 它包括了慢启动、拥塞避免、快速重传和快速启动等特性, 是目前最主要也是最流行的 TCP

协议版本. 单条 TCP 连接发送窗口随时间的变化可以用图 1 表示.

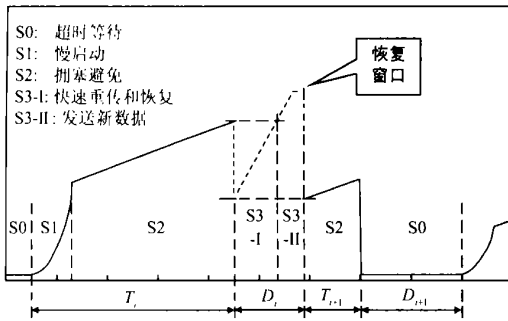


图 1 TCP 窗口随时间变化示意

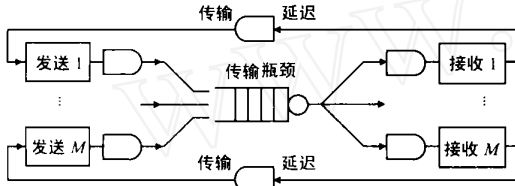


图 2 网络传输拓扑模型

本文考虑如图 2 所示的网络传输模型. 在这个模型中,  $M$  个 TCP 连接共享一条前向瓶颈链路, 反馈链路则仅有传输延迟而不会产生拥塞. 瓶颈链路被等效成一个缓存容量为  $B$ , 服务率为  $\mu$  的服务队列. 瓶颈链路前后的链路都有传输延迟.

网络瓶颈处总窗口长度具有图 3 所示的时间变化关系. 假设在  $t_1$  时刻发送窗口开始大于缓存容量, 此时发送的包在传输瓶颈处被丢弃. 经过一个往返传输延迟  $T$  以后, 发送方在  $t_3 = t_1 + T$  检测到这个丢包, 然后向下调整窗口长度. 随着作出窗口调整的连接逐渐增多, 发送窗口长度逐渐减小. 再设在  $t_2 = t_1 +$  时刻发生首个丢包的连接, 在  $t_4 = t_2 + T$  时刻检测到拥塞并作出窗口调整以后, 窗口尺寸降到  $B$  以下, 于是不再有新的丢包发生. 大约再经过  $T$  时间, 在  $t_5$  时刻所有发生丢包的连接都收到了首个丢包的重传应答, 被恢复的连接重新开始增大它们的发送窗口, 总窗口长度随之逐渐增大, 开始下一个传输周期.

通常认为在多个连接的链路中, 拥塞避免阶段在 TCP 稳态行为中占据主要地位<sup>[12]</sup>. 采用快启动 TCP 协议的连接只存在拥塞避免状态<sup>[13]</sup>. 因此假定当拥塞发生时, TCP 连接都已转入拥塞避免阶段, 所有连接的总窗口超过链路容量. 记:  $s$  是拥塞门限,  $T$  是往返时延,  $B$  是包括了链路容量的总缓存容量. 发生拥塞时, 链路上总共有  $N$  个连接处于拥塞避免状态,

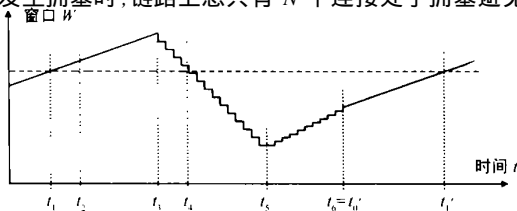


图 3 网络瓶颈总窗口长度变化示意图

若记  $W$  为总窗口大小,  $W_i$  为第  $i$  条连接的窗口大小,

$$W = \sum_{i=1}^N W_i \quad (1)$$

$$t = T / W_i \quad (2)$$

$$W_i = \begin{cases} 1, & W_i \leq s_i \\ 1/W_i + K, & s_i < W_i \leq r_i \\ 0, & W_i > r_i \end{cases} \quad (3)$$

式中:  $s_i$  是第  $i$  个连接的拥塞门限;  $r_i$  是接收方建议的窗口长度;  $K$  是对拥塞避免阶段窗口增量的修正 (V. Jacobson 建议该额外增量取 4 字节, TCP-Reno 取为  $1/8$  最大段长 = 64 字节). 当许多 TCP 连接竞争瓶颈链路带宽时, 发送窗口几乎不可能增大到接收方建议的窗口长度. 于是定义起始时间为进入拥塞的时间 ( $t_1 = 0$ ), 采用流体近似并假定这些包是均匀传递的, 有

$$\frac{dW}{dt} = \sum_{i=1}^N \frac{dW_i}{dt} = \frac{N + WK}{T} \quad (4)$$

$$W = (B + N/K) \exp(Kt/T) - N/K \quad (5)$$

### 3 交换结点的丢包模型

进入拥塞以后网络瓶颈处将发生丢包. 参考文献均假定常数丢包率, 并且认为发生丢包以后, 总能通过快速重传开始新的传输周期. 然而 Matthew Mathis 和 Vern Paxson 发现在前向信道上能观察到整个窗口的数据都被丢失的现象. 仿真表明, 由于网络的拥塞结点需要处理大量的 TCP 连接, 发生拥塞以后, 一些连接可能在一个窗口内有多个包被丢弃, 整个窗口数据被丢弃也是可能的. 下面就来讨论这个丢包问题.

在下面的分析中, 记窗口长度为  $W_i$  的第  $i$  个连接, 从  $t_1$  到  $t$  累计发送  $x_i(t)$  个包, 丢失  $y_i(t)$  个包;  $N$  个连接从  $t_1$  到  $t$  总共发送  $x(t)$  个包, 丢失  $y(t)$  个包, 并假设窗口随时间的变化  $W_i(t)$  是独立同分布的随机过程. 根据发送窗口与平均到达率的关系, 可以得到 TCP 窗口调整起作用以前的平均到达率、到达包数和丢包数

$$\lambda(t) = \frac{N + KB}{T(1 - e^{-K})} \exp \frac{Kt}{T} - \frac{N}{KT} \quad (6)$$

$$\bar{x}(t) = \frac{N + KB}{K(1 - e^{-K})} \left( \exp \frac{Kt}{T} - 1 \right) - \frac{Nt}{KT} \quad (7)$$

$$\bar{y}(t) = \frac{N + KB}{K(1 - e^{-K})} \left( \exp \frac{Kt}{T} - 1 \right) - \frac{N + KB}{KT} t \quad (8)$$

#### 3.1 分散丢包的策略

如果节点采取分散丢包的策略, 即在一个 RTT 时间中尽可能地丢包分散在不同的连接中 (这正是 TCP 协议所期望的). 于是当第  $i$  条连接检测到丢包时, 该连接的窗口长度将减为 1. 记

$$= B/N + 1/K \quad (9)$$

$$\bar{y}(t - T) = \frac{N e^{-K}}{1 - e^{-K}} \left( \exp \frac{Kt}{T} - 1 \right) - N \frac{t}{T} \quad (10)$$

$$W(t) = \frac{1}{C_N^{y(t-T)}} \left\{ \prod_{j=1}^{C_N^{y(t-T)}} \left( \prod_{i=1}^N \left[ 1 + \frac{N}{i - \bar{y}(t-T) + 1} W_{j,i}(t) \right] \right) \right\} \quad (11)$$

$$= - N^2 \exp \frac{2Kt}{T} + N \left[ \frac{t}{T} + \frac{1}{K} + \dots \right]$$

$$\exp \frac{Kt}{T} - N \frac{K+1}{K} \left( \frac{t}{T} + \right) - \frac{N}{K} \quad (12)$$

$$= \frac{e^{-K}}{1 - e^{-K}}, = \frac{1}{1 - e^{-K}} \quad (13)$$

式中  $C_N^{y(t-T)}$  是  $N$  条连接中有  $y(t-T)$  条检测到丢包时排列组合数. 窗口回落到  $B$  的时间可由  $W(t_b) = B$  得到,

$$\exp \frac{2Kt_b}{T} - \left( \frac{t_b}{T} + \dots + \frac{1}{K} \right) \exp \frac{Kt_b}{T} + \frac{K+1}{K} \frac{t_b}{T} + \dots + \frac{1}{K} = 0 \quad (14)$$

此时到达率可分为两部分, 一部分是检测到丢包时连接产生的流量,

$$p_1(t) = N/T \quad (15)$$

$$p_1(t) = \left[ \exp \frac{Kt}{T} - 1 \right] - \frac{t}{T} \quad (16)$$

另一部分是未检测到丢包时连接产生的流量,

$$p_2(t) = N \frac{K}{T} \exp \frac{Kt}{T} - \frac{N}{KT} \quad (17)$$

$$p_2(t) = 1 - p_1(t) \quad (18)$$

$$p(t) = p_1(t) + p_2(t) \quad (19)$$

到  $t_b$  时刻为止, 节点总共发送的包数  $x(t_b)$  为

$$x(t_b) = \int_0^{t_b} p(t) dt = A_1 \exp \frac{2Kt_b}{T} + A_2 t \exp \frac{Kt_b}{T} + A_3 \exp \frac{Kt_b}{T} + A_4 t_b^2 + A_5 t_b + B t_b / T + A_6 \quad (20)$$

总共丢失的包数为

$$y(t_b) = x(t_b) - \mu t_b = A_1 \exp \frac{2Kt_b}{T} + A_2 t \exp \frac{Kt_b}{T} + A_3 \exp \frac{Kt_b}{T} + A_4 t_b^2 + A_5 t_b + A_6 \quad (21)$$

其中:  $A_1 = -N^2 \frac{1}{2}, A_2 = N^2 \frac{1}{T},$   
 $A_3 = N \left[ \left(1 + \frac{1}{K}\right) - \frac{1}{K} + \frac{K+1}{K^2} \right], A_4 = -N \frac{K+1}{2KT^2},$   
 $A_5 = -\frac{N}{T} \left[ 1 + \frac{K+1}{K} \right], A_6 = N \left\{ e^{2K} \frac{1}{2} + e^K \left[ \left( \frac{1}{K} - \right) \right. \right.$   
 $\left. \left. - \left[ 1 + \frac{K+1}{K^2} \right] + \frac{K+1}{2K} + \frac{K+1}{K} \right] \right\}.$

丢两个以上包的连接数为

$$n_{l2} = \begin{cases} y(t_b) - N, & \forall N < y(t_b) < 2N \\ N, & \forall y(t_b) > 2N \end{cases} \quad (22)$$

### 3.2 随机丢包策略

如果采用随机丢包策略, 即对到达包不加区分, 一旦缓冲区满立即丢弃新到达的包. 这是交换机所期望的一种丢包方式. 在这种丢包策略下, 即使丢弃相同数量的包, 也可能对 TCP 连接产生不同的影响. 例如当所有丢包恰好发生在一个连接上时, 到达率只会很小的下降, 而该连接将重传超时; 若丢包分散在很多不同的连接上, 很多连接进入窗口调整使窗口迅速减小, 可能大部分连接都不会重传超时. 对此需要采用统计平均的方法进行分析.

假设每一条连接的流量均服从泊松到达, 第  $i$  条连接在  $t$  时刻到达包数为  $x_i(t)$ , 平均到达包数为  $\bar{x}_i(t)$ ,

$$p(x_i(t) = x_i) = \frac{1}{x_i!} (\bar{x}_i(t))^{x_i} \exp(-\bar{x}_i(t)) \quad (23)$$

$$x(t) = \prod_{i=1}^N x_i(t) \quad (24)$$

$$p(x(t) = x) = \frac{1}{x!} (\bar{x}(t))^x \exp(-\bar{x}(t)) \quad (25)$$

$$y(t) = \begin{cases} x(t) - [\mu t], & x(t) > [\mu t] \\ 0, & x(t) \leq [\mu t] \end{cases} \quad (26)$$

$$p(y_i | x_i, x) = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \begin{bmatrix} x - x_i \\ y - y_i \end{bmatrix} / \begin{bmatrix} x \\ y \end{bmatrix} \quad (27)$$

$$p(y_i) = \sum_{x=0}^x \frac{x_i! (x-x_i)! y! (x-y)!}{x! (x-y-y_i)! (y-y_i)! (x-y-x_i+y_i)! x!} \cdot \frac{\bar{x}_i^x}{x_i! x!} \exp(-\bar{x}_i - \bar{x}) \quad (28)$$

$$p(y_i = 0 | \bar{x}_i) = \sum_{x=0}^x \frac{(x-x_i)! (x-y)! \bar{x}_i^x}{x! (x-y-x_i)! x_i! x!} \exp(-\bar{x}_i - \bar{x}) \quad (29)$$

$$\sum_{x=0}^x \sum_{y_i=0}^x \left( \frac{x-y}{x} \right)^{x_i} \frac{\bar{x}_i^{x_i}}{x_i!} \exp(-\bar{x}_i) \frac{\bar{x}^x}{x!} \exp(-\bar{x}) \quad (30)$$

$$= \exp \left( -\frac{y}{x} \bar{x}_i \right) \frac{\bar{x}^x}{x!} \exp(-\bar{x}) \quad (31)$$

$$1 + \sum_{n=1}^n \frac{\bar{x}_i^n}{n!} \sum_{j=0}^n \binom{n}{j} (-1)^{n-j} \left( \frac{[\mu t]}{x} \right)^j \sum_{x=[\mu t]+j+1}^x \frac{\bar{x}^x}{x!} \exp(-\bar{x}) \quad (32)$$

$$1 + (1 - C_0) \sum_{n=1}^n \frac{\bar{x}_i^n}{n!} \left( \frac{[\mu t]}{x} - 1 \right)^n - \sum_{n=1}^n \frac{\bar{x}_i^n}{n!} \sum_{j=0}^n \binom{n}{j} \left( \frac{[\mu t]}{x} \right)^j \cdot (-1)^{n-j} \frac{\bar{x}_i^{[\mu t]}}{[\mu t]!} \exp(-\bar{x}) \frac{\bar{x}^x}{[\mu t+1]!} \sum_{k=0}^x \left( \frac{\bar{x}}{x+d} \right)^k \quad (33)$$

$$= C_0 + \exp \left( -\frac{y}{x} \bar{x}_i \right) \{ 1 - C_0 - C_1 C_2 \} \quad (34)$$

其中参数  $d$  决定了式(33)中用于近似原数列和的等比数列的

收敛速度,  $C_0 = \sum_{x=0}^{[\mu t]} \frac{\bar{x}_i^x}{x!} \exp(-\bar{x}), C_1 = \frac{\bar{x}_i^{[\mu t+1]}}{[\mu t+1]!} \exp(-\bar{x}),$

$$C_2 = \frac{\bar{x}+d}{d} \left( 1 - \exp \left( -\frac{\bar{x}_i d}{x+d} \right) \right).$$

$$p(y_i = 1 | \bar{x}_i) = \sum_{x=0}^x \sum_{y_i=0}^x \frac{x_i y (x-x_i)! (x-y)! \bar{x}_i^{x_i} \bar{x}^x}{x! (x-y-x_i+1)! x_i! x!} \exp(-\bar{x}_i - \bar{x}) \quad (35)$$

$$\sum_{x=0}^x \sum_{y_i=0}^x \frac{y}{x-y-y_i} \left( \frac{x-y}{x} \right)^{x_i} \frac{\bar{x}_i^{x_i} \bar{x}^x}{x_i! x!} \exp(-\bar{x}_i - \bar{x}) \quad (36)$$

$$= \sum_{x=0}^x \frac{y}{x} \bar{x}_i \exp \left( -\frac{y}{x} \bar{x}_i \right) \frac{\bar{x}^x}{x!} \exp(-\bar{x}) \quad (37)$$

$$- \sum_{n=1}^n \frac{\bar{x}_i^n}{n!} \sum_{j=0}^n \binom{n}{j} (-1)^{n-j} \left( \frac{[\mu t]}{x} \right)^j \left\{ 1 - \sum_{x=0}^{[\mu t]+j} \frac{\bar{x}_i^x}{x!} \exp(-\bar{x}) \right\} \quad (38)$$

$$= \bar{x}_i \exp \left( -\frac{y}{x} \bar{x}_i \right) \left\{ (1 - C_0) \frac{\bar{y}}{x} + C_1 \exp \left( -\frac{\bar{x}_i d}{x+d} \right) - C_1 C_2 \frac{\bar{y}}{x} \right\} \quad (39)$$

$$p(y_i > 1) = 1 - p(y_i = 0) - p(y_i = 1) \quad (40)$$

平均没有发生丢包的连接数  $n_0$  和丢 1 个包的连接数  $n_1$  分别为



$$n_{i0} = \sum_{k=0}^N \sum_{l=1}^{C_N^k} k \binom{k}{i=1} p^{(l)}(y_i=0) \binom{N}{i=k+1} (1 - p^{(l)}(y_i=0)) \quad (41)$$

$$= \frac{\partial}{\partial x} \left\{ \sum_{i=1}^N (p(y_i=0) + 1 - p(y_i=0)) \right\} \Big|_{x=1} \quad (42)$$

$$= \sum_{i=1}^N p(y_i=0) \quad (43)$$

$$= \sum_{i=1}^N \sum_{x=0}^{\infty} \exp\left[-\frac{y_i}{x}\right] \frac{\bar{x}^x}{x!} \exp(-\bar{x}) \quad (44)$$

$$n_{i1} = \sum_{k=0}^N \sum_{l=1}^{C_N^k} k \binom{k}{i=1} p^{(l)}(y_i=1) \binom{N}{i=k+1} (1 - p^{(l)}(y_i=1)) \quad (45)$$

$$= \sum_{i=1}^N p(y_i=1) \quad (46)$$

$$= \sum_{i=1}^N \sum_{x=0}^{\infty} \frac{y_i}{x} \exp\left[-\frac{y_i}{x}\right] \frac{\bar{x}^x}{x!} \exp(-\bar{x}) \quad (47)$$

丢包发生  $T$  时间以后, TCP 连接检测到丢包而对窗口做出调整. 由于采用随机丢包, 显然检测到丢包的连接数少于分散丢包的情况, 发送窗口下降的速度也慢于分散丢包. 于是在发送窗口尺寸下降到缓存容限以内之前, 到达包数不少于采用分散丢包策略时的到达包数, 链路丢包数也不少于采用分散丢包策略时的丢包数. 然而另一方面, 通常从窗口开始下降到停止丢包的时间远小于往返传输时延, 这段时间到达的包数和丢包数相对较少. 这里用分散丢包策略的到达包数和丢包数来近似, 得到随机丢包策略的性能上限.

记进入拥塞时单条 TCP 连接的初始到达率为  $\rho_i$ , 如果忽略  $t_3 - t < t_4$  时流量特性的变化, 平均到达率和平均到达包数可写为:

$$\rho_i = \left( \rho_0 + \frac{1}{KT} \right) \exp\left(\frac{Kt}{T} - \frac{1}{KT}\right) \quad (48)$$

$$\bar{x}_i(t) = \frac{\rho_0 T}{K} \left( \exp\left(\frac{Kt}{T} - 1\right) \right) + \frac{1}{K_2} \left( \exp\left(\frac{Kt}{T} - 1\right) \right) - \frac{t}{KT} \quad (49)$$

考虑两种典型分布下的丢包情况, 并用分散丢包策略下的链路到达包数和丢包数, 来作为随机丢包策略下链路到达包数和丢包数的近似. 如果进入拥塞时的初始到达率  $\rho_i$  都相同, 这时

$$\bar{x}_i = \bar{x} / N \quad (50)$$

如果进入拥塞时的初始到达率  $\rho_0$  服从指数分布,

$$f(\rho_0) = \frac{1}{ie} \exp\left(-\frac{\rho_0}{ie}\right) \quad (51)$$

$$p(y_i=0) = \sum_{x=0}^{\infty} \exp\left[-\frac{y_i}{x}\right] \frac{\bar{x}^x}{x!} \exp(-\bar{x}) f(\rho_0) d\rho_0 \left\{ \frac{1-C_0}{1+C_5} - \frac{\bar{x}+d}{d} \left[ \frac{C_1}{1+C_5} - \frac{C_1}{1+C_5+C_6} \exp\left(\frac{-C_3 d}{x+d}\right) \right] \right\} \exp\left[-C_3 \frac{y_i}{x}\right] + C_0 \quad (53)$$

$$p(y_i=1) = \sum_{x=0}^{\infty} \frac{y_i}{x} \exp\left[-\frac{y_i}{x}\right] \frac{\bar{x}^x}{x!} \exp(-\bar{x}) f(\rho_0) d\rho_0 \quad (54)$$

$$= \exp\left[-C_3 \frac{y_i}{x}\right] \left\{ \frac{y_i}{x} \left[ 1 - C_0 - C_1 \frac{\bar{x}+d}{d} \right] \left\{ \frac{C_3}{1+C_5} + \frac{C_4}{(1+C_5)^2} \right\} + C_1 \exp\left[\frac{-C_3 d}{x+d}\right] \left[ 1 + \frac{\bar{x}+d}{x} \right] \right\} \left\{ \frac{C_3}{1+C_5+C_6} + \frac{C_4}{(1+C_5+C_6)^2} \right\} \quad (55)$$

式中:  $C_3 = \frac{1}{K^2} \left( \exp\left(\frac{Kt}{T} - 1\right) - \frac{t}{KT} \right)$ ,  $C_4 = \frac{ieT}{K} \left( \exp\left(\frac{Kt}{T} - 1\right) \right)$ ,  $C_5 = \frac{y_i}{x} - \frac{ieT}{K} \left( \exp\left(\frac{Kt}{T} - 1\right) \right)$ ,  $C_6 = \frac{d}{x+d} - \frac{ieT}{K} \left( \exp\left(\frac{Kt}{T} - 1\right) \right)$ .

#### 4 仿真结果和讨论

理论计算与仿真结果示于图 4~6. 这里仿真工具采用伯克利大学洛伦兹国家实验室研制的网络仿真工具“ns<sup>[14]</sup>”. 仿真中的 TCP 连接为持续的 FTP 文件传输. 图 4 和图 6 中的“x”点代表仿真结果, 它们都几乎落在相应条件的理论分析曲线上, 表明了理论分析能与仿真结果能够吻合. 从图 4 可以看出, 采用 TCP 进行流量控制, 链路总缓存越大, 活动连接数越多, 拥塞期间总共到达的 TCP 包就越多; 受到传输容量的限制, 总共丢弃的 TCP 包也越多. 这表明多 TCP 连接的特性与单条 TCP 连接相比会有很大的差别.

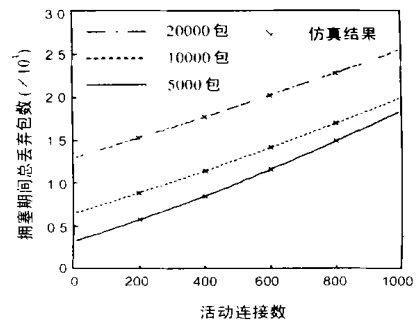


图 4 分散丢包时总丢包数与活动连接数的关系

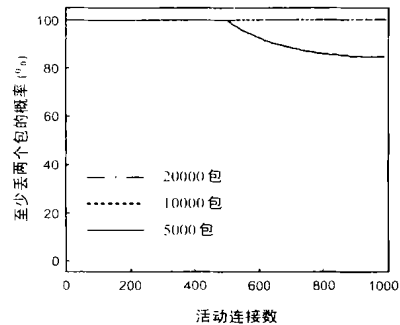


图 5 丢两个包的概率与活动连接数的关系

从图 5 可以看出, 采用分散丢包策略, 几乎每条连接都要丢弃 2 个或 2 个以上的包, 表明分散丢包实际上不能减少重传超时. 图 6 则显示, 采用随机丢包策略, 随着链路上活动连接数的增多, 丢 2 个或 2 个以上包的概率略有下降. 这是因为当活动连接数增加时, 拥塞节点的总丢包数比活动连接数增长得慢, 使平均每条连接的丢包数有所降低. 当各个连接的到达率服从指数分布时, 丢 2 个或 2 个以上包的概率最小. 这可

能是因为当到达率不同时,各条连接到到达的总包数会不同,一些到达包较少的连接丢 2 个或 2 个以上包的可能性很小,其总的效果是使丢 2 个或 2 个以上包的连接数有所下降.图 6 中的“×”点代表采取随机丢包(路由器的通常设置)时的仿真结果.这个仿真结果与到达率服从指数分布、在网络瓶颈处采取随机丢包策略时的理论分析结果比较吻合,表明本文的理论能够在一定程度上对实际网络进行近似.

运用本文的理论,我们来分析从互联网上下载较大文件时的情况.假设被下载的文件位于较远的服务器,或者从服务器到本地主机的传输链路上存在传输瓶颈.由于下载的文件较大,下载过程必然持续较长的时间,并且通常都会遭遇到网络拥塞.从图 4 可以看出,采用 TCP 协议进行拥塞和流量控制时,一旦发生拥塞,由于往返传输时间的原因,当窗口调整机制起作用时,已经有大量的包被丢弃了.从图 5 和图 6 可以看出,发生拥塞以后,不仅很多连接会发生丢包,而且单条连接丢 2 个或 2 个以上包的概率也很大.因为 TCP-Reno 只能恢复一个窗口内的单个丢包,2 个或 2 个以上的丢包将导致 TCP 重传超时.实际上 IP 网络的往返传输时间通常可达数百毫秒,重传超时时间可能超过数秒甚至数十秒.在这种情况下,虽然重传定时器尚未超时,但是会话层认为会话连接已经丢失,于是中断了会话过程.这就解释了互联网中较大的文件难以一次下载成功的原因.

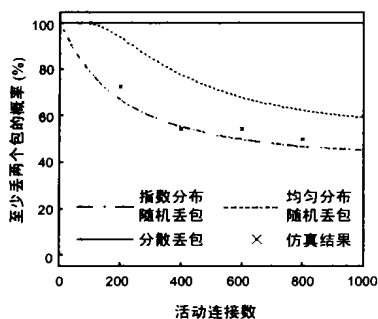


图 6 丢两个包的概率与活动连接数的关系

## 5 结论

目前的 TCP 事实标准使用拥塞丢包来检测链路容量.只要不发生丢包,发送窗口就会一直增加,最终导致拥塞丢包发生.因此拥塞是不可避免的.在数字光纤传输网络中,网络瓶颈处的拥塞是网络丢包的主要原因.

本文提出了 TCP 连接的丢包分析模型,估计了存在大量持续 TCP 连接时,网络瓶颈的丢包概率,给出对实际网络的良好近似. TCP 连接在拥塞发生时的丢包性能表明, TCP-Reno 在进行流量控制时具有很大的局限性.本文的研究还表明,当流量控制的反应时间较长时,为了保证网络的稳定工作,有必要引入资源预留等其它流量控制机制.

## 参考文献:

[1] 陈相宁,程时昕.一种自适应早期包丢弃方案[J].电子与信息学报,2001,23(8):764-771.

- [2] Kenji Kawahara, Koichiro Kitajima, Tetsuya Takine, Yuji Oie. Packet loss performance of selective cell discard schemes in ATM switches [J]. IEEE Journal, 1997, SAC-15(5):903-913.
- [3] K Fall, S Floyd. Simulation-based comparisons of Tahoe, Reno and SACK TCP [R]. Berkeley: Lawrence Berkeley Laboratory, 1996.
- [4] Boning Feng, Dipak Ghosal, Narana Kannappan. Impact of ATM ABR control on the performance of TCP-Tahoe and TCP-Reno [A]. Proc. of Globecom97 [C]. Phoenix: Globecom, 1997. III-1832-1837.
- [5] Omar Eloumi, Hossam Afifi. Improving RED algorithm performance in ATM networks [A]. Proc. of Globecom97 [C]. Phoenix: Globecom1997. III-1062-1067.
- [6] R Jain, R Goyal, S Kalyanaraman, F S. Performance of TCP over UBR and buffer requirements [S].
- [7] Georgios Y Lazarou, Victor S Frost, Joseph B Evans, Douglas Niehaus. Using measurements to validate simulation models of TCP/IP over high speed ATM wide area networks [R/OL]. 1996, <http://citeseer.nj.nec.com/lazarou96using.html>.
- [8] T V Lakshman, U Madhow. The performance of TCP/IP for networks with high bandwidth-delay products and random loss [J]. IEEE/ACM Trans., 1997, Networking-5(3):336-350.
- [9] L Zhang, S Shenker, D D Clark. Observations on the dynamics of a congestion control algorithm: The effects of two-way traffic [J]. ACM Trans., 1991 CCR-21(4):133-147.
- [10] Sanjay Shakkottai, Anurag Kumar. TCP over end-to-end ABR: A study of TCP performance with end-to-end rate control and stochastic available capacity [A]. Proc. of Globecom98 [C]. Sydney: Globecom, 1998. III-2027-2034.
- [11] V Jacobson. Congestion avoidance and control [A]. Proc. Of ACM SIGCOMM88 [C]. Stanford, CA: SIGCOMM, 1988. F-314-329.
- [12] Matthew Mathis, Jeffrey Semke, Jamshid Mahdavi. The macroscopic behavior of the TCP congestion avoidance algorithm [J]. ACM Trans., 1997, CCR-27(3):1-16.
- [13] Venkata, N Padmanabhan, Randy H Katz. TCP fast start: A technique for speeding up web transfers [A]. Proc. of Globecom98 [C]. Sydney: Globecom, 1998. F-41-46.
- [14] S McCanne, S Floyd, K Fall. NS-LBNL network simulator [CP/OL]. Lawrence Berkeley Laboratory, 2001, <http://www.nrg-ee-lbl.gov/ns>.

## 作者简介:



陈相宁 男,1967 年生于江苏南京,1989 年在南京大学获理学本科学士,1992 年在电子科技大学获工学硕士,同年进入南京熊猫从事移动通信研发工作,1996 年进入东南大学移动通信国家重点实验室从事通信网络的理论研究,2000 年获博士学位,现为清华大学微波与数字通信国家重点实验室博士后.

王 京 (见本期第 940 页)

信号打包传送到 Internet 上,处理节点和网络节点分布在 Internet 的不同位置,并对 RF 信号进行处理.然后,经过低层处理后的信息传送到另外的网络节点,进行网络层及高层的处理以及数据访问等.因此 Internet 相当于 DWCS 的物理传输网络,而分布式处理节点和网络节点成为一些专用的 Internet 节点,从而形成全分布式无线通信系统.当然这一阶段能否实现主要取决于 Internet 技术的发展.

DWCS 采用多层分布式结构,可以灵活地实现一些无线通信系统.如果将第三代移动通信系统中的 UTRAN 部分采用软件无线电实现,每根天线收发某一个小区用户信息,那么 DWCS 就可以完全实现第三代移动通信系统的功能.另外,分布式处理部分可以控制发给每根天线的信号,例如给多根天线发送相同的信号,即实现直放站的功能.

## 5 结论

基于以上分析,可以得到 DWCS 具有以下特点:

⑧ 分布性 DWCS 在逻辑上采用四层分布式结构,各个功能节点按需要分布在不同的地理位置,简化工程实现难度.

⑧ 开放性 在 DWCS 的基本结构上可以实现现有的和可预见未来的协议和技术,并充分利用了有线网络技术,如高速 Internet 网络.

⑧ 无缝覆盖 天线与处理单元的分离可以灵活地设置天线和控制不同天线上的收发信号,实现空间的无缝覆盖.

⑧ 灵活性 采用软件无线电技术实现分布式处理系统,可以灵活地实现不同的无线传输体制,而不需要重新设计硬件平台,有效地解决了无线系统之间互不兼容的问题.

⑧ 可扩展性 DWCS 是一个开放的无线通信平台,基于这个平台可以进行再开发、扩展,实现不同传输体制和业务,如全分布式无线信息系统.

⑧ 可伸缩性 DWCS 可以根据需要在规模和大小方面自行配置,减小设备成本.

## 参考文献:

- [ 1 ] 尤肖虎,曹淑敏,李建东.第三代移动通信系统发展与展望[J].电子学报,1999,27(11A):3-8.
- [ 2 ] Wang J,Zhao M,Xu X,Yao Y. Research on hardware platform of the software radio [J]. IEICE Trans Commun,2000,E83-B(6):1210-1216.
- [ 3 ] Kitayama K,Kuri T,Ogawa T. Error-free optical 156-Mbit/s millimeter-wave wireless transport through 60GHz external modulation [C]. OFC 98 technical digest Japan:OFC,1998.
- [ 4 ] Chu T,Gans MJ. Fiber optic microcellular radio [J]. IEEE Trans. Vehicular Technology,1991,40(3):599-606.
- [ 5 ] Web site:http://www.forxcomwireless.com [Z].
- [ 6 ] Yanikomeroglu H,Sousa E S. Antenna interconnection strategies for personal communication systems [J]. IEEE Journal on selected areas in Commun.,1997,15(7):1327-1336.

## 作者简介:

王京男,1958年1月生于北京市,1986年在清华大学无线电电子学系获通信与电子系统专业硕士学位,现为中国第三代移动通信研究开发项目总体组成员、第三代移动通信技术试验专家组专家、国家“十五”863计划通信技术主题专家、中国电子学会高级会员、中国通信学会无线及移动通信委员会委员、清华大学信息科学技术学院院长助理、微波与数字通信国家重点实验室副主任和无线传输与个人通信研究室主任,研究兴趣包括分布式无线通信系统与网络、虚拟无线移动通信系统与网络、第三代移动通信系统与技术、移动通信系统信道编译码技术和扩频码分多址技术中的多用户检测、时空分集等.



姚彦男,1938年生于浙江省兰溪,1962年毕业于清华大学无线电电子学系,长期在清华大学供职,现为清华大学电子工程系教授,博士生导师,微波与数字通信国家重点实验室主任,无线电电子学研究所副所长,全国高技术重点图书通信技术领域编审委员会委员,中国通信学会微波与卫星通信专业委员会委员,中国通信学会国际学术工作委员会委员,中国通信学会会士,中国电子学会高级会员,IEEE高级会员.

赵明男,1971年10月生于甘肃兰州,1993年在清华大学获得学士学位,1998年在清华大学电子工程系获得博士学位,现为清华大学电子工程系讲师,专业特长为无线个人通信系统的理论与技术,已经发表论文20多篇,目前的主要研究方向为:新一代无线通信系统、软件无线电、无线及个人通信技术.

周世东男,1969年12月生于江苏南京,清华大学电子工程系副教授,1991年毕业于东南大学,1994年在东南大学和硕士学位,1997年在清华大学获博士学位,在国内外发表多篇论文,目前的主要研究领域为移动通信、扩频通信、编码与调制及接收算法.