TCP 拥塞控制算法建模分析

曹雪峰

(河北大学硕士研究生班,保定 071000)

摘 要:介绍 TCP 协议的拥塞控制算法:慢启动、拥塞避免、快重传和快恢复等。给出了使用 OPNET 仿真建模的方法,直观地演示了拥塞控制算法的运行过程,对结果进行分析。从方法学的角度看,这些对于研究、开发和教学均有一定的指导意义和参考价值。

关键词:TCP 协议;拥塞控制;拥塞窗口;仿真分析

0 引 言

在某段时间内,网络中要求传输过多的分组时,网络的性能开始下降,这种情况即称为拥塞。简单的说就是当用户对网络资源的需求超过了网络能提供的可用资源时的一种状态,即对资源的需求的总和在于系统可用资源。单纯地增加网络资源并不能解决拥塞问题,是这因为拥塞本身是一个动态问题,它不可能只靠静态的方案来解决,而需要协议能够在网络出现拥塞时保护网络的正常运行。目前对互联网进行的拥塞控制主要是依靠在源端执行的 TCP 拥塞控制机制。

1 TCP 拥塞控制算法

1.1 慢启动和拥塞避免

慢启动和拥塞避免算法被 TCP 发送端用来控制正在向网络输送的数据量。为了实现这些算法,必须向 TCP 每连接状态加入两个参量。拥塞窗口(cwnd)是对发送端收到确认(ACK)之前能向网络传送的最大数据量的一个发送端限制,接收端通知窗口(rwnd)是对未完成数据量的接收端限制。cwnd 和 rwnd 的最小值决定了数据传送。另一个状态参量,慢启动阈值(ssthresh),被用来确定是用慢启动还是用拥塞避免算法来控制数据传送。ssthresh 的初始值可以任意大(例如,一些实现使用通知窗口的尺寸),但是作为对拥塞的响应,其大小可能会被减小。慢启动算法在 cwnd<ssthresh 时使用,拥塞避免算法在 cwnd>ssthresh 时使用。当 cwnd=ssthresh 时,发送端既可以使用慢启动也可以使用拥塞避免。

当建立新的 TCP 连接时,设置 ssthresh=65535,执行慢启动算法,拥塞窗口初始化为一个数据包大小(缺省为 536bytes,即 1 个 MSS)。源端按 cwnd 大小发送数据,每收到一个 ACK 确认,cwnd 就增加一个数据包发送量。显然,cwnd 的增长将随 RTT 呈指数级增长:1、2、4、8……源端向网络中发送的数据量将急剧增加。

当 cwnd 超过 ssthresh 时慢启动结束,TCP 就执行拥塞避免算法,此时,cwnd 在每次收到一个 ACK 时只增加 1/cwnd 个数据包,这样,在一个 RTT 内,cwnd 将增加 1,所以在拥塞避免阶段,cwnd 不是呈指数增长,而是线性增长。当观察到拥塞时拥塞避免结束,将慢启动阈值(ssthresh)设置为当前拥塞窗口的一半,拥塞窗口设置为 1。如果 cwnd<ssthresh,TCP 重新进入慢启动阶段,如此重复。

1.2 快重传和快恢复

快重传算法以 3 个重复 ACK 的到达(收到 4 个一样的 ACK,其间没有任何其他包到达)为一个数据包已经丢失的标志。在收到 3 个重复 ACK 之后,TCP 不等超时重传时间到就重传已经丢失的数据包。

在快重传算法发送了看来已经丢失的数据包之后,快恢复算法支配了数据的传送,直到一个非重复 ACK 到达。具体实现如下:

①当第三个重复 ACK 收到时,设置 ssthresh 不大于等式 ssthresh=max(Flight size/2,2*MSS)给定的值。其中 Flight size 是已经被发送但还没有确认的数据的总量;

②重传丢失的包然后设置拥塞窗口 cwnd=

收稿日期:2008-09-08 修稿日期:2008-11-25

作者简介:曹雪峰(1967-),男,河北隆化人,副教授,承德民族师专数计系,研究方向为计算机网络应用技术

ssthresh+3*MSS，扩大拥塞窗口；

③对每个接收到的另一个重复 ACK，将 cwnd 增大 MSS 字节，这将人为地扩充拥塞窗口以反映已经离开网络的附加数据包；

④如果 cwnd 和接收端的通知窗口的值允许的话，发送一个新数据包；

⑤当下一个确认新数据的 ACK 到达时，设定 cwnd 值为 ssthresh（步骤 1 设置的值），按拥塞避免算法继续执行。

2 仿真分析

目前 TCP 协议主要包含有 4 个版本：TCP Tahoe、TCP Reno、TCP NewReno 和 TCP SACK。TCP Tahoe 包括“慢启动”、“拥塞避免”和“快速重传”3 个最基本的拥塞控制算法。TCP Reno 在 TCP Tahoe 基础上增加了“快恢复”算法。本文主要利用这两个版本对 TCP 拥塞控制的 4 个核心算法进行建模分析。

2.1 仿真建模

在 OPNET 中建立实验仿真模型如图 1，模型由 3 个节点组成：一个 Packet Discarder 节点，模拟数据传输错误，设置丢弃数据包的时间和数量；一个 FTP 服务器和一个客户机。仿真模型使用以下参数：链路为 1.544Mb/s 的 ppp_DS1 双向链路，客户机执行文件下载操作，下载文件大小为 11MB，仿真运行时间为 90s，仿真运行到 40~40.5s 时丢失 2 个数据包。在此模型上创建 3 个场景如下：

场 景 1：slow_start_and_congestion_avoidance with_ drop

在此场景中，客户机和服务器上只运行 TCP 的慢启动和拥塞避免算法。

场景 2：Tahoe_with_drop

在前一个场景的基础上，增加快重传算法。

场景 3：Reno_with_drop

在前一个场景的基础上，增加快恢复算法。

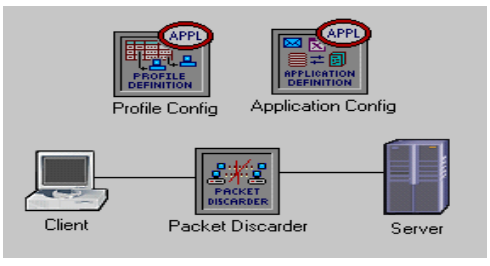

图 1 实验仿真模型

2.2 仿真结果和分析

（1）有数据包丢失时拥塞窗口变化

图 2 给出网络出现拥塞时不同算法下拥塞窗口变化曲线。从曲线可以看出，开始时各场景都执行慢开始算法，cwnd 值增长很快，当 cwnd=ssthresh 时（图 2 中①处），开始执行拥塞避免算法，cwnd 值线性增长。运行到 40s 时，有数据包丢失，网络出现拥塞。场景 1 不执行快重传算法，因此当最小超时重传时间（见图 3）到的时候才开始重传（图 2 中②处），并重新设置 cwnd 和 ssthresh 的值，再重复执行慢启动和拥塞避免算法；而在后两个场景中执行了快重传算法，因此不用等到超时重传时间到就开始重传。在场景 2 中没有执行快恢复算法，和场景 1 一样重新设置 cwnd 和 ssthresh 的值，再重复执行慢启动和拥塞避免算法；在场景 3 中执行快恢复算法，设置 cwnd 和 ssthresh 的值，当收到对新数据包的确认后直接执行拥塞避免算法。

（2）快恢复算法中 ssthresh 值的确定

在图 4 中①处为执行快恢复算法时设置的慢开始门限值，②处为与其对应的 Flight size 值。可以看出，慢开始门限值的设置符合公式 ssthresh=max (Flight size/2，2*MSS)的要求。

（3）不同算法下的吞吐量比较

从图 5 可以看出，使用了快重传和快恢复算法的网络，在网络出现拥塞时，网络吞吐量要远远高于只使用慢启动和拥塞避免算法的网络。

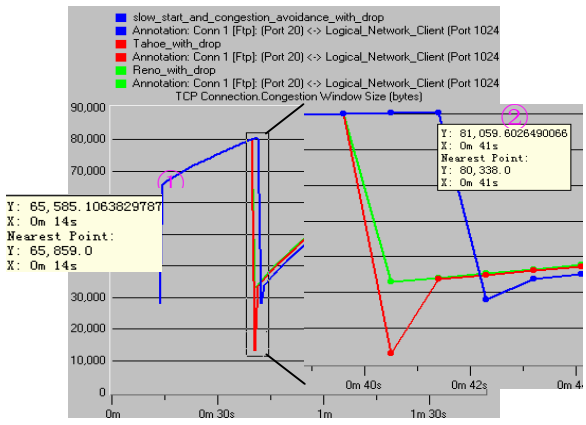

图 2 网络出现拥塞时拥塞窗口变化

	+ Retransmission Thresholds	Attempts Based
?	├Initial RTO (sec)	3.0
?	├Minimum RTO (sec)	1.0
?	├Maximum RTO (sec)	64
?	├RTT Gain	0.125

图 3 TCP Parameters 最小超时重传时间设置

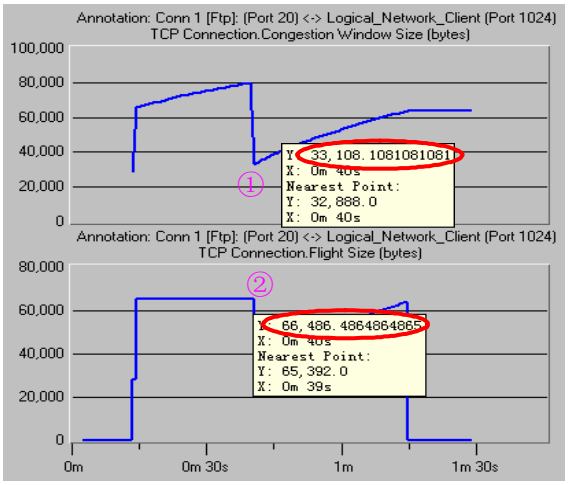

图 4 ssthresh 与 Flight size 的关系

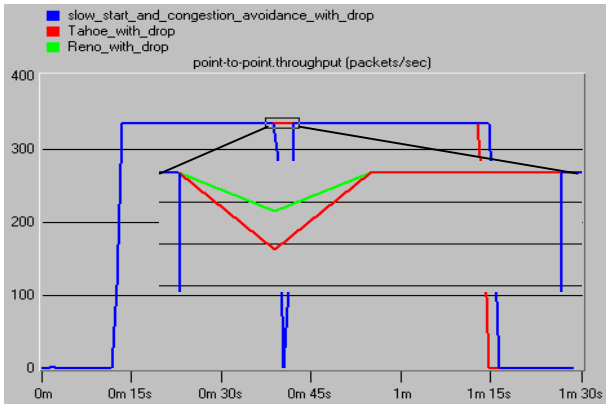

图 5 网络吞吐量比较

3 结 语

慢启动、拥塞避免、快速重传和快速恢复算法是 TCP 拥塞控制的 4 个核心算法，特别是快速重传和快速恢复算法提高了网络吞吐量。通过对算法进行建模分析，加深了对算法理解，同时也为进行网络仿真研究提供了系统科学的方法。

参考文献

[1]谢希仁. 计算机网络(第四版)[M]. 北京：电子工业出版社，2003：256~271

[2]James F.Kurose Keith W.Ross 著. 陈鸣译. 计算机网络自顶向下方法与 Internet 特色[M]. 北京：机械工业出版社，2005：175~188

[3]杨琳苹. 基于仿真的 TCP 流量控制机制的研究[J]. 四川理工学院学报(自然科学版)，2006，19(3)：75~80

[4]陈敏. OPNET 网络仿真[M]. 北京：清华大学出版社，2004：40~105

[5]王文博，张金文. OPNET Moduler 与网络仿真[M]. 北京：人民邮电出版社，2003：294~318

[6]黄镇建. 基于 NS2 的 DV 算法仿真及结果分析[J]. 现代计算机(专业版)，2008，03：28~30

[7]W. Richard Stevens. TCP Congestion Control[S]. IETF RFC 2581，1999

Modeling Analysis on TCP Congestion Control Algorithm

CAO Xue-feng

(Class of Graduate, Hebei University, Baoding 071000)

Abstract: Introduces the TCP protocol congestion control algorithm, such as Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery. Introduces the simulating process and direct-viewing demonstration congestion control algorithm running process, analyses the result. From the methodological point of view, it is significant for research, development and instruction.

Keywords: TCP Protocol; Congestion Control; Congestion Window; Simulation-Based Analysis