

多路径并行传输中接收缓存阻塞问题的研究

宋 飞¹, 王 博², 张宏科¹, 张思东¹

(1. 北京交通大学电子信息工程学院下一代互联网互联设备国家工程实验室, 北京 100044;
2. 国家计算机网络应急技术处理协调中心, 北京 100029)

摘 要: 本文研究了多路径并行传输情况下接收缓存阻塞问题产生的原因及对传输性能的影响. 同时提出了一种适用于多路径并行传输吞吐量分析的建模方法. 该模型能够通过路径丢包率、往返传输时间等参数对吞吐量进行估计, 并综合考虑了快速重传、拥塞窗口与接收缓存容量的关系对结果的影响. 通过与仿真结果进行比较可以看出, 该模型能够在网络性能参数发生变化时很好的预测吞吐量的变化情况.

关键词: 多路径并行传输; 吞吐量模型; 接收缓存阻塞; 往返传输时间

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2010) 03-0552-04

Research on Receive Buffer Blocking in CMT

SONG Fei¹, WANG Bo², ZHANG Hong-ke¹, ZHANG Si-dong¹

(1. *National Engineering Laboratory for Next Generation Internet Interconnection Devices, School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing 100044, China;*
2. *National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China*)

Abstract: The reasons and influence of receive buffer blocking in Concurrent Multipath Transfer (CMT) environment are studied. A modeling method which is suitable for analyzing the throughput of CMT is also proposed. This model can be used to estimate the throughput according to the drop rate, round trip time and other parameters. It considers whether the fast retransmission is applied and the relationship between the size of congestion window and the volumes of receive buffer. Comparing with the simulation results, the model can predict the variation of throughput very well when the parameters of network are changed.

Key words: concurrent multipath transfer; throughput model; receive buffer blocking; round trip time

1 引言

随着互联网及相关技术的发展, 接入方式的种类越来越多(如 WiFi, WiMax, GPRS, 3G 等). 目前的笔记本电脑, 智能手机等移动终端设备大多具有两种或两种以上的接入方式. 与此同时, 由于实时多媒体业务的不断增加, 用户希望互联网服务提供商 (ISP) 能够提供更大的带宽. 如何合理的利用多种接入方式进行数据交互正在得到广泛的关注^[1~3]. 在这样的背景下, 多路径并行传输 (CMT) 已经逐渐成为研究的热点^[4~7].

目前, 多路径并行传输方面的研究工作大多围绕着 SCTP^[3]和 TCP 协议展开. 和传统的 TCP 协议相比, SCTP 协议使用关联来表征两个终端间的连接关系, 并选择一条路径作为数据传输的主路径, 关联中的其他路径作为备用路径. 发送端在备用路径上发送心跳包. 当主路径

发生故障时, 协议将选择备用路径继续进行传输. 而多路径并行传输的思路是在多条路径上同时传输数据, 从而提高整个关联的吞吐量. 然而, 无论通过何种方式实现多路径并行传输, 都存在传输过程中可能会出现接收缓存阻塞^[5]现象, 使得多路径并行传输的效率大大降低. 目前针对该问题的研究成果还不是很多^[6,7].

2 接收缓存阻塞的原因

在使用传统的 TCP 协议进行数据传输时, 数据包只在一条链路上进行发送. 由于存在路由器丢包和路由的变化, 数据包可能无法按序到达接收端, 即传输过程中的乱序问题. 当使用多条路径进行并行传输时, 由于每条路径的属性不同, 网络的性能参数 (如带宽、时延) 各异, 很难保证数据包能够同时到达接收端. 同时丢包也可能导致后发出的数据包先到达, 使乱序问题变得

更加严重.

对于面向可靠连接的 TCP、SCTP 等协议来说,数据包的发送速率主要和两个参数有关:当前的拥塞窗口和对端的接收缓存.第一个参数反映了网络当前的拥塞情况,第二个参数反映了对端的处理能力.在这类协议上实现多路径并行传输时,由于接收端只能按照数据包的编号顺序向应用层提交,当乱序问题发生时,接收端缓存将形成数据包堆积的情况.这将对传输性能造成严重影响.

3 接收缓存阻塞对吞吐量的影响

在上一节我们讨论了接收缓存阻塞产生的原因.在本节中我们结合文献[8]的分析方法,提出一种多路径并行传输情况下接收缓存阻塞对吞吐量影响的分析模型.假定在传输过程中各条链路上均未发生超时重传,并且忽略慢启动阶段对于传输的影响.

当关联中有 n 条路径时,令 W_n 为 n 条路径拥塞窗口之和,则每一条路径的拥塞窗口为 W_n/n .令 α 和 β 分别为每条路径上的增加和倍减因子,则当某一路径发生丢包时,相应路径上拥塞窗口的值将会减少 $\beta W_n/n$,因此整个关联拥塞窗口的值将变为 $(n-\beta)W_n/n$.

在拥塞避免阶段,发送端在一个往返传输时间(RTT)里收到某条路径上发送窗口数据包的所有确认

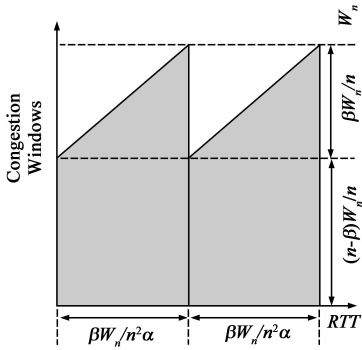


图1 无快速重传时的拥塞窗口变化情况

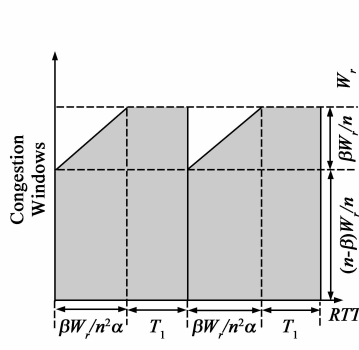


图2 无快速重传且受接收缓存影响时拥塞窗口的变化情况

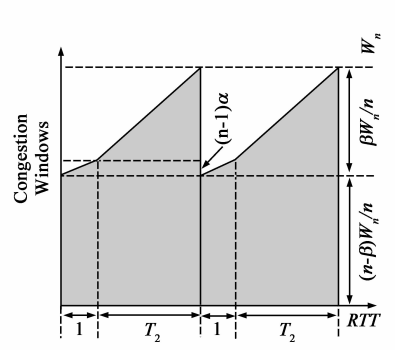


图3 考虑快速重传时的拥塞窗口变化情况

由于拥塞窗口增长的上限值为 W_r ,若求整个关联的吞吐量,需要首先求出图中 T_1 的值:

$$T_1 = \frac{2n^3\alpha - (2n-\beta)W_r^2\beta p}{2n^3\alpha p W_r} \quad (4)$$

则整个关联的吞吐量为:

$$\text{Throughput} = \frac{(1/p) \cdot \text{MSS}}{(\beta W_r / (n^2\alpha) + T_1 + 1) \cdot \text{RTT}} \quad (5)$$

3.3 考虑快速重传且拥塞窗口小于接收缓存时

在实际的传输过程中,发生丢包后发送端大多采用快速重传机制.在这种情况下,丢失的数据包将会被重新发送,并且发送端在收到重传数据包的选择性确认(SACK)之前,不增加相应的拥塞窗口.

信息后,会将该路径上的拥塞窗口增加 α ,因此整个关联的拥塞窗口将增加 $n\alpha$.为便于理解,首先分析简单情况下接收缓存阻塞对整个关联拥塞窗口变化的影响.

3.1 不考虑快速重传且拥塞窗口小于接收缓存时

在这种情况下,发生丢包的路径会直接进入拥塞避免阶段.由于某一条路径的丢包不会影响其他路径的正常传输,整个关联的拥塞窗口恢复到 W_n 需要 $\beta W_n/n^2\alpha$ 个 RTT.如图 1 所示,当链路丢包率为 p 时,可以得到如下等式:

$$\frac{\beta W_n}{n^2\alpha} \cdot \frac{(n-\beta)W_n}{n} + \frac{1}{2} \cdot \frac{\beta W_n}{n^2\alpha} \cdot \frac{\beta W_n}{n} = \frac{1}{p} \quad (1)$$

从而可以得到

$$W_n = n \sqrt{2n\alpha / (2n-\beta)p\beta} \quad (2)$$

令 W_r 为接收端缓存大小,当 $W_n < W_r$ 时,吞吐量可以表示为:

$$\text{Throughput} = \frac{(1/p) \cdot \text{MSS}}{(\beta W_n / (n^2\alpha) + 1) \cdot \text{RTT}} \quad (3)$$

其中 MSS 为最大分组数据的大小,RTT 为往返传输时间.

3.2 不考虑快速重传且拥塞窗口不小于接收缓存时

当考虑接收缓存对模型影响(即 $W_n \geq W_r$)时,整个关联的拥塞窗口变化情况如图 2 所示.

基于上述分析,在考虑快速重传时整个关联的拥塞窗口变化情况如图 3 所示.同时可以得到 T_2 的表达式为:

$$T_2 = (\beta W_n - n(n-1)\alpha) / (n^2\alpha) \quad (6)$$

考虑丢包率和吞吐量的关系可得到如下等式:

$$\frac{1}{2} \left(\frac{(2+T_2)(n-\beta)W_n}{n} + (1+T_2)(n-1)\alpha + W_n T_2 \right) = \frac{1}{p} \quad (7)$$

整理后即为:

$$W_n = \frac{2n - (1+T_2)(n-1)\alpha p n}{p((2+T_2)(n-\beta) + nT_2)} \quad (8)$$

通过上式可以求得 W_n 的值,当 $W_n < W_r$ 时,吞吐

量可表示为:

$$Throughput = \frac{(1/p) \cdot MSS}{\left(\frac{\beta W_n - n(n-1)\alpha}{n^2\alpha} + 2\right) \cdot RTT} \quad (9)$$

3.4 考虑快速重传且拥塞窗口不小于接收缓存时

当 $W_n \geq W_r$ 时,则整个关联的拥塞窗口变化情况如图 4 所示,拥塞窗口增长的上限值为 W_r ,若求整个关联的吞吐量,需要首先求出图中 T_3 和 T_4 的值.根据拥塞窗口的增长机制可以得到:

$$T_3 = (\beta W_r - n(n-1)\alpha) / (n^2\alpha) \quad (10)$$

根据丢包率的关系可以得到 T_4 的表达式为:

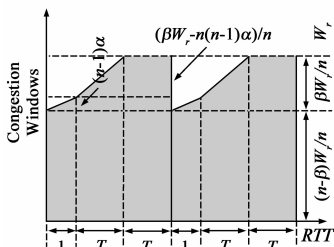


图4 考虑快速重传且受接收缓存影响时拥塞窗口的变化情况

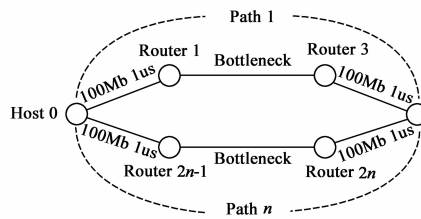


图5 仿真拓扑结构图

$$T_4 = \frac{2n - p((2 + T_3)(n - \beta) W_r + (1 + T_3)(n - 1) \alpha + n W_r T_3)}{2pnW_r} \quad (11)$$

此时整个关联的吞吐量可表示为:

$$Throughput = \frac{(1/p) \cdot MSS}{(T_3 + T_4 + 2) \cdot RTT} \quad (12)$$

4 仿真实验及结果分析

为了验证模型的准确性,我们在 NS2-2.31^[9] 平台上利用 SCTP-CMT 模块进行了仿真.图 5 为搭建的仿真环境拓扑结构图.

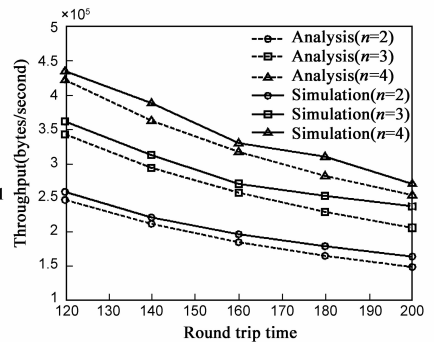


图6 不考虑接收缓存时RTT变化对于吞吐量的影响

当没有超时重传情况出现时,两条链路的瓶颈带宽都设定为 10Mb.通过设置不同的网络参数(往返传输时间、链路丢包率)进行仿真.开启文献[4]中的 CUC, SFR 和 DAC 算法.设置 Data Chunk 的大小为 1468 字节.使用 FTP 作为应用层协议.

对拥塞窗口不受接收缓存影响,并且考虑快速重传的情况进行仿真.设定接收端缓存大小为 2^{28} Bytes,以保证在仿真时间内吞吐量不受接收缓存的任何影响.

设定丢包率为 0.01,链路 RTT 在 120ms ~ 200ms 之间变化时的比较结果如图 6 所示.图中分别验证了当用于传输的链路数为 2,3 和 4 时模型的准确度.当传输时延比较小时,计算结果和仿真结果差距也比较小.由于模型中对于 RTT 的变化比较敏感,随着往返传输时间的不断增加,计算结果和仿真结果的一致程度有所下降.从图中可以看出,和使用两条链路进行传输相比,使用四条路径进行传输能够获得更大的吞吐量.

当设定链路的 RTT 为 100ms、丢包率从 0.01 增加到 0.05 时,模型计算结果和仿真吞吐量的比较情况如图 7 所示.随着链路数目的增加,丢包对于整体性能的影响也随之增加.这一点与我们的分析结果是一致的.

当丢包率增加时,使用两条链路传输的吞吐量受到的影响比使用四条链路传输的吞吐量受到的影响要小.由于随着链路数目的不断增加,发生丢包后可能产生的拥塞窗口变化情况也相应增加,因此模型的计算结果在丢包率较高时和仿真结果的差距较大.

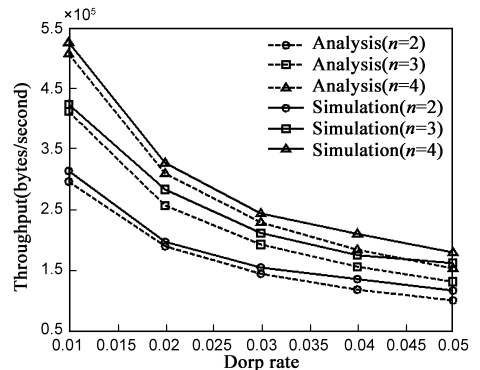


图7 不考虑接收缓存条件下丢包率变化对于吞吐量的影响

图 8 和图 9 是在考虑接收缓存的影响和快速重传的情况下模型的计算结果和仿真结果的比较情况.设定接收缓存为 16KB,丢包率为 0.01,链路 RTT 在 120ms ~ 200ms 之间变化.由于接收缓存阻塞问题的存在,使用四条路径并行传输的优势并不明显.随着 RTT 的不断增大,吞吐量也逐渐变小.在接收缓存受限的条件下,增加传输路径条数对于传输性能的改善并不明显.

当固定 RTT 的值为 100ms、变化链路丢包率时(从 0.01 到 0.05),吞吐量的变化情况如图 9 所示.在这种情况下,由于考虑了接收缓存和速重传的影响,丢包率的增加使得模型计算结果和仿真结果有些偏差(特别是当丢包率比较大时).但该模型仍能够较好的反应吞吐量的变化趋势.

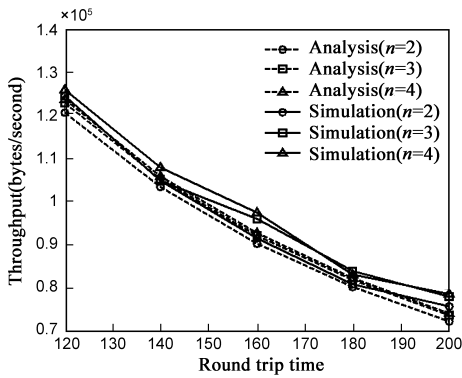


图8 考虑接收缓存条件下RTT变化对于吞吐量的影响

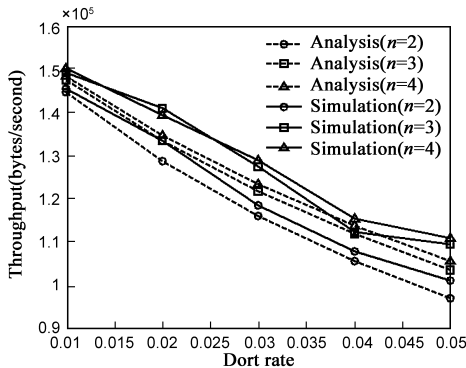


图9 考虑接收缓存条件下丢包率变化对于吞吐量的影响

5 总结

在使用单路径进行数据传输时,接收端缓存阻塞问题是造成传输吞吐量下降的重要原因之一.在多路径并行传输情况下,这种影响尤为突出.尽管每条路径使用独立的拥塞窗口及快速重传判别机制,但当数据包乱序情况发生时,传输层不能及时向上层提交已到达的数据包并清空接收缓存,导致传输性能的下降.因此在研究多路径并行传输问题时,需要考虑接收缓存阻塞所造成的影响.

本文在研究多路径并行传输特点的基础上,建立模型分析了接收缓存阻塞对传输吞吐量的影响.为使模型更加准确,我们根据发送端是否引入快速重传机制、接收缓存与拥塞窗口的大小关系,将模型分成四种情况进行讨论.通过和仿真结果进行比对可以得出,该模型能够在网络性能参数发生变化时,很好的预测多路径并行传输过程中吞吐量的变化情况.同时为分析接收缓存阻塞问题提供了一种新的思路.

参考文献:

[1] 杨冬,周华春,张宏科.基于一体化网络的普适服务研究[J],电子学报,2007,35(4):607-613.

Yang Dong, Zhou Huachun, Zhang Hongke. Research on pervasive services based on universal network[J]. Acta Electronica Sinica, 2007, 35(4): 607-613. (in Chinese)

- [2] RFC 4340, 2006. Datagram Congestion Control Protocol (DCCP)[S].
- [3] RFC 2960, 2000. Stream Control Transmission Protocol[S].
- [4] J Iyengar, P Amer, R Stewart. Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths[J]. IEEE/ACM Transactions on Networking, 2006, 14(5): 951-964.
- [5] J Iyengar, P Amer, R Stewart. Receive buffer blocking in concurrent multipath transfer[A]. In Proc IEEE GLOBECOM[C] st. Louis, USA; IEEE, 2005.
- [6] P Natarajan, N Ekiz, P Amer, et al. Concurrent multipath transfer using SCTP multihoming: Introducing the potentially-failed destination state[A]. In Proc IFIP Networking[C]. Singapore: IFIP, 2008.
- [7] J Iyengar, P Amer, R Stewart. Performance implications of a bounded receive buffer in concurrent multipath transfer[J]. Computer Communications. 2007, 30(4): 818-829.
- [8] Q Fu, J Indulska, S Perreau. et al. Exploring TCP parallelisation for performance improvement in heterogeneous networks[J]. Computer Communications. 2007, 30(17): 3321-3334.
- [9] Univ California, Berkeley, LBL, USC/ISI, Xerox Parc. NS-2. 31. [Z]. Available: <http://www.isi.edu/nsnam/ns>.

作者简介:



宋 飞 男, 1983年4月生于河北, 北京交通大学博士研究生, 主要研究方向为网络体系架构、网络协议分析与优化.

E-mail: song2000fei@gmail.com



王 博 男, 1982年2月生于甘肃. 2009年7月毕业于北京交通大学, 获工学博士学位. 现工作于国家计算机网络应急技术处理协调中心. 主要研究方向是网络负载均衡传输协议, 网络安全, 风险评估等.