多路径并行传输中吞吐量的建模与分析

宋 飞,苏 伟,张宏科,张思东
(北京交通大学电子信息工程学院下一代互联网设备国家工程实验室,北京 100044)

摘 要: 本文提出了一种多路径并行传输情况下吞吐量的建模方法,并在保证准确度的前提下给出了优化算法.模型结合不同传输阶段的特点对拥塞窗口的增长机制进行了分析和说明,通过往返传输时间和拥塞窗口的变化情况对吞吐量进行估计.在Matlab平台上实现该模型可以发现,优化算法能够大大降低复杂度,同时模型计算结果和仿真结果的拟合程度较好.

关键词: 多路径并行传输;吞吐量模型;拥塞窗口;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) 04-0887-07

Modeling and Analysis of Throughput in CMT

SONG Fei, SU Wei, ZHANG Hong-ke, ZHANG Si-dong
(*National Engineering Laboratory for Next Generation Internet Interconnection Devices, School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing* 100044, *China*)

Abstract: A modeling method of throughput in Concurrent Multipath Transfer (CMT) environment is proposed and an optimlzation algorithm is also given under the premise of accuracy. The increase mechanisms of congestion window are analyzed and interpreted based on the features of different transmission phases. The model estimates the throughput according to the changes of round trip time and congestion window. Optimlzation algorithm can decrease the complexity sharply when we implement our idea on Matlab platform. And the agreement of calculation results and simulation results is satisfactory.

Key words: concurrent multipath transfer; throughput model; congestion window; round trip time

1 引言

随着互联网接入方式的不断增加,终端设备大多配备了多种网络接口,用户对于网络的需求也越来越高[1].但由于目前的传输层协议只支持端到端的单路径传输,多种接入方式不能发挥最大的优势.特别是当路径发生故障时,单路径传输只能在多次重传仍未收到应答包之后终止发送.而多路径并行传输的出现使得最大限度的利用带宽资源成为了可能,也为传输层协议的设计提出了更高的要求.目前这一领域已经成为了学者们关注和研究的热点[4~6].

SCTP[2]和DCCP[3]协议是IETF提出的支持多家乡的两种传输层协议,但目前多路径并行传输的研究大多围绕着SCTP和TCP协议展开,和传统的TCP协议相比,SCTP协议使用关联来描述两个终端间的连接关系,并选择一条路径作为主路径,关联中的其他路径作为备用路径.发送端在主路径上发送数据包,并在备用路径上发送心跳包.当主路径发生故障时,协议将选择一条备用路径继续传输.若在多条路径(主路径和备用路径)上同时传输数据,整个关联的吞吐量将有可能得到大幅度的提高.

直接在多条路径上同时发送数据包将会引起不必要的快速重传和错误的拥塞窗口调整.针对这两个问题文献[4]设计了LS-SCTP.这种实现方式在发送端和接收端都进行了相应的修改,同时原始数据包格式也需要改变.为了避免对SCTP协议进行大规模的修改,Ye等研究人员提出了IPCC-SCTP协议[5].其设计思想和LS-SCTP相似,但IPCC-SCTP的实现只需要在发送端进行修改即可.文献[6]的作者也提出了一种只修改发送端的实现方法(CMT-SCTP)解决了上述两个问题.同时还提出了一种减少选择性确认(SACK)数量的算法,但该算法需要在发送端和接收端同时进行修改.

在建立数学模型方面,不少学者提出了一些针对SCTP传输特点的建模方法.这些模型对于建立多路径并行传输情况下的吞吐量模型具有一定的借鉴意义:

文献[7]参考了经典的TCP吞吐量模型[8]的分析方法.首先考虑在慢启动阶段、拥塞避免阶段和超时重传阶段传输数据包的数学期望,然后再分析上述三个阶

收稿日期:2008-08-27;修回日期:2009-12-20
基金项目:国家973重点基础研究发展规划(No.2007CB307100);国家自然科学基金重点项目(No.60833002);国家自然科学基金(No.60972010,60903150)

段所需时间的数学期望,最后给出 SCTP 吞吐量模型.该模型更为详细的介绍可见文献[9].文献[10]中利用 Markov 链的分析方法将整个模型分成两个部分:源模型和网络模型.目的是将 SCTP 的拥塞控制算法从动态变化的网络环境中分离出来,从而提高模型的准确度.在文献[11]中,作者结合了 TCP 时延模型[12]的思想,给出了基于两级门限故障切换机制的 SCTP 传输时延模型的建立方法.同时作者希望能够通过该模型实现动态调整关联内故障切换门限值的目标,从而提高传输性能.文献[13]给出了无线环境下 SCTP 吞吐量的模型,分析了传输过程中可能出现的几种类型的丢包(垂直切换丢包,拥塞丢包和错误丢包)对于吞吐量的影响.

上述模型虽然从不同的角度对 SCTP 的传输特点进行了描述,但都只考虑了单路径传输的情况.本文提出了一种多路径并行传输情况下吞吐量的建模方法和优化算法,并验证了模型的准确度.

文章整体结构如下:第二部分介绍模型的建立过程;第三部分描述使用 Matlab 实现该模型的方法,以及如何降低算法的复杂度;第四部分通过比较模型的计算结果和在 NS2 中的仿真结果来验证模型的准确度;第五部分对所做工作进行总结.

2 多路径并行传输吞吐量模型的建立

为了便于对模型的建立过程进行分析,以两条路径(路径 A 和路径 B)的 CMT-SCTP 实现方案为例进行描述.借鉴文献[13]的分析思路,首先做出如下假设:数据包丢失只发生在由发送端到接收端的方向上,既 SACK 在传输过程中不会发生丢失;不考虑接收缓存阻塞[14]的影响同时两条路径的网络性能参数相同.

在建立模型过程中需要用到的名词和变量主要包括:

Cwnd	发送端拥塞窗口(简称 CW)
SS	慢启动
CA	拥塞避免
MTU	最大传输单元
RTO	重传超时
RTT	传输层的往返传输时间
SST	慢启动门限值
q	每一个 Data Chunk[2]成功传输的概率
sk_size	SACK Chunk[2]的大小
ck_size	Data Chunk 的大小
R_{tx}	数据包传输速率
cw_i	在第 i 轮后两条路径上 CW 可能的大小
δ	物理层的往返传输时间
qq_i	第 i 轮后 Data Chunk 成功传输的概率
ck_i	前 i 轮中成功传输的数据包总和
RTT_i	前 i 轮中所有往返传输时间的总和

2.1 路径不存在 RTO 时

SCTP 关联建立之后,两条路径上总的 CW 为 4(每条路径的初始值为 2).即传输开始之前 $cw_0=[4]$, $qq_0=[1]$.传输开始后数组 cw_i 和 qq_i 的变化情况如图 1 所示.为了更好的说明拥塞窗口的变化过程,我们按照轮次进行描述.

2.1.1 在第一轮传输过程中

(1)传输数据包的总数为:

$$ck_1 = cw_0 \times q \times qq_0^T = 4q \tag{1}$$

(2)考虑单路径传输情况下,发送 cw 个数据包需

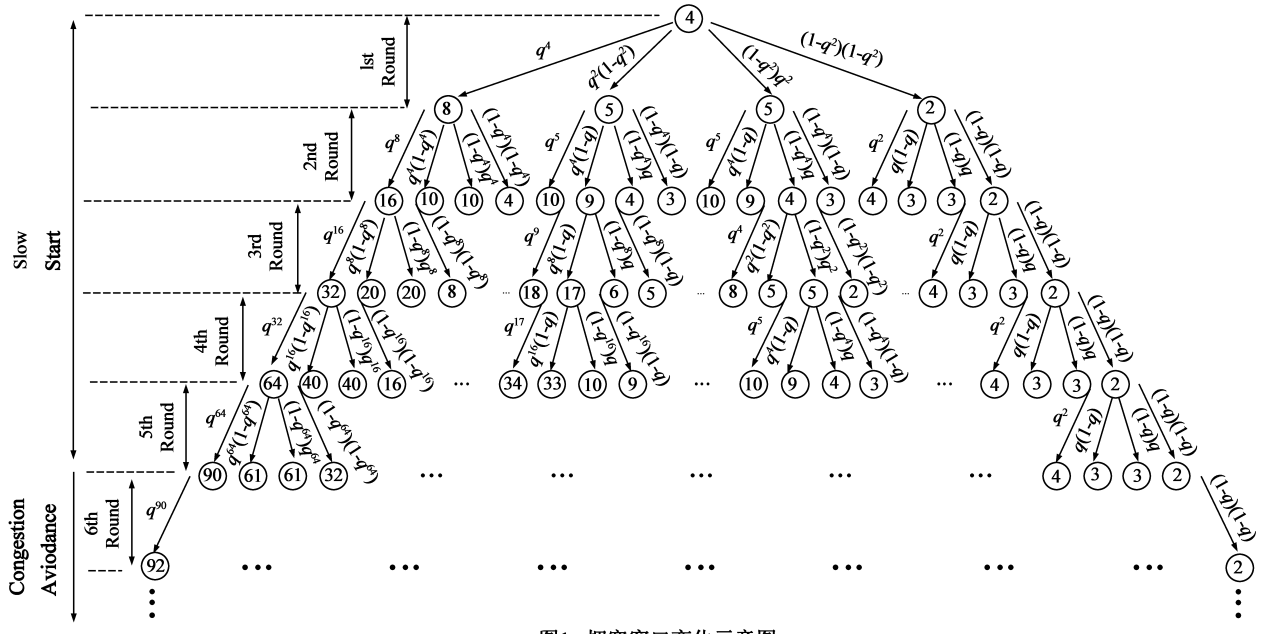

图1 拥塞窗口变化示意图

要的时间为:

$$rtt(cw)=\frac{cw\times ck\_size+(cw/b)\times sk\_size}{R_{tx}} \tag{2}$$

其中 b 为一个 SACK 中确认的 Data Chunk 的个数.则发送 cw_0 个数据包需要的时间为:

$$rtt(cw_0)\times qq_0^T+\delta=rtt(4)+\delta \tag{3}$$

因此根据假设条件可得多路径并行传输情况下传输时间为:

$$RTT_1=\frac{rtt(cw_0)\times qq_0}{2}+\delta=\frac{rtt(4)}{2}+\delta \tag{4}$$

(3)第一轮结束后,cw_0 和 qq_0 的值将发生变化,可能的情况包括:路径 A 和路径 B 的数据全部发送成功;路径 A 发送成功,路径 B 有丢包;路径 B 发送成功,路径 A 有丢包;两条路径都有丢包.

对于每条路径来说,若数据全部发送成功,则对应的 CW 的值将变为 2 * CW.若有丢包,发送端可以通过 SACK 中的 Gap 信息了解到,并将 CW 的值设置为 Max($CW/2$, MTU).因此两条路径上 CW 的值可能为(4,4),(4,1),(1,4),(1,1).对应的 cw_1 包含四个元素,即 $cw_1=[8\quad 5\quad 5\quad 2]$.

对于路径 A 和路径 B,拥塞窗口从 2 变到 4 的概率为 q^2,从 2 变到 1 的概率为 $1-q^2$.从整体来看,当两条路径的数据包全部传输成功时,两条路径 CW 之和将从 4 变到 8,其概率为 $q^2\times q^2$.当路径 A 成功传输而路径 B 发生丢包时,CW 之和为 5,到达该情况的概率为 $q^2\times(1-q^2)$.当路径 B 成功传输而路径 A 发生丢包时,CW 之和同样为 5,到达该情况的概率同样为 $q^2\times(1-q^2)$.当两条路径都发生丢包时,CW 之和为 2,其概率为 $(1-q^2)(1-q^2)$.根据以上描述不难求得 qq_1 的值,如式(5)所示(通过转秩的形式给出):

$$qq_1^T=\begin{bmatrix}qq_0(1)q^2q^2\\ qq_0(1)q^2(1-q^2)\\ qq_0(1)(1-q^2)q^2\\ qq_0(1)(1-q^2)(1-q^2)\end{bmatrix}=\begin{bmatrix}q^2q^2\\ q^2(1-q^2)\\ (1-q^2)q^2\\ (1-q^2)(1-q^2)\end{bmatrix} \tag{5}$$

其中 $qq_0(1)$表示 qq_0 中的第一个元素.

2.1.2　在第二轮传输过程中

(1)传输数据包总数为 $cw_1\times q\times qq_1^T$,前两轮传输数据包总数可以根据式(6)计算得到:

$$\begin{aligned}ck_2&=ck_1+cw_1\times q\times qq_1^T\\&=4q+[8q\quad 5q\quad 5q\quad 2q]\times\begin{bmatrix}q^2q^2\\ q^2(1-q^2)\\ (1-q^2)q^2\\ (1-q^2)(1-q^2)\end{bmatrix}\\&=4q+8q^5+5q^3(1-q^2)+5(1-q^2)q^3+2q(1-q^2)^2\\&=6q(q^2+1)\end{aligned} \tag{6}$$

(2)本轮 RTT 的值为 $(rtt(cw_1)\times qq_1^T)/2+\delta$.因此前两轮中总的 RTT 的值可以根据式(7)计算得到.

$$\begin{aligned}RTT_2&=RTT_1+\frac{rtt(cw_1)\times qq_1^T}{2}+\delta\\&=\frac{rtt(4)}{2}+\delta+\frac{rtt(1)\times cw_1\times qq_1^T}{2}+\delta\\&=\frac{4rtt(1)+(6q^2+2)rtt(1)}{2}+2\delta\\&=3(q^2+1)rtt(1)+2\delta\end{aligned} \tag{7}$$

(3)第二轮结束后,由于 cw_1 中包含 4 个元素,每个元素可能产生 4 个新元素,以 cw_1 第二个元素为例描述变化过程:根据拥塞窗口的变化规律,cw_2 中的第 5 个到第 8 个元素如式(8)所示:

$$cw_2(5:8)=[10\quad 9\quad 4\quad 3] \tag{8}$$

对应的 qq_2 中的值为(通过转秩的形式给出):

$$\begin{aligned}qq_2(5:8)^T&=\begin{bmatrix}qq_1(2)q^4q\\ qq_1(2)q^4(1-q)\\ qq_1(2)(1-q^4)q\\ qq_1(2)(1-q^4)(1-q)\end{bmatrix}\\&=\begin{bmatrix}q^2(1-q^2)q^4q\\ q^2(1-q^2)q^4(1-q)\\ q^2(1-q^2)(1-q^4)q\\ q^2(1-q^2)(1-q^4)(1-q)\end{bmatrix}\end{aligned} \tag{9}$$

其中 $qq_1(2)$表示 qq_1 中的第二个元素.

2.1.3　在第三轮传输过程中

(1)传输数据包总数为 $cw_2\times q\times qq_2^T$,前三轮传输数据包总数为 $ck_3=ck_2+cw_2\times q\times qq_2^T$.

(2)本轮中 RTT 的值为 $(rtt(cw_2)\times qq_2^T)/2+\delta$.则前三轮总的 RTT 值为:

$$RTT_3=RTT_2+\frac{rtt(cw_2)\times qq_2^T}{2}+\delta \tag{10}$$

(3)第三轮结束后,cw_2 中的每个元素可能产生 4 个新元素,根据上述规律可以计算出相应的 cw_3 和 qq_3.

2.1.4　在第 r 轮传输过程中

(1)传输数据包总数为 $cw_{r-1}\times q\times qq_{r-1}{}^T$.前 r 轮传输数据包总数为 $ck_r=ck_{r-1}+cw_{r-1}\times q\times qq_{r-1}{}^T$.

(2)本轮 RTT 值为 $(rtt(cw_{r-1})\times qq_{r-1}{}^T)/2+\delta$.

(3)前 r 轮中总的 RTT 的值为:

$$RTT_r=RTT_{r-1}+\frac{rtt(cw_{r-1})\times qq_{r-1}{}^T}{2}+\delta \tag{11}$$

则两条路径前 r 轮的吞吐量可根据式(12)计算得到:

$$Throughput_r=ck_r/RTT_r \tag{12}$$

2.2　路径存在 RTO 时

我们考虑只有一条路径发生故障时的吞吐量模型(当两条路径同时发生故障时吞吐量为 0).在实际的网络中,域间路由发生故障后往往需要数十分钟才能恢

复[15],在这段时间内端到端的传输将受到极大地影响.传统的 SCTP 协议中路径状态只有两种:Active 和 Failed.当某条路径出现 RTO 时,发送端将重传丢失的数据包.重传成功后(如在另一条路径上进行重传),仍会有数据包在发生故障的路径上传输,并导致再次的超时重传.只有当 RTO 次数达到路径失败检测阈值 Path.Max.Retrans (PMR)时.该路径的状态才会被设置为 Failed.PMR 的默认值为 5,即 SCTP 需要大约 63s 才能发现路径故障.

针对上述情况,Natarajan 等研究人员提出了一种 CMT-PF[16]的实现方法,通过引入 Potentially-Failed (PF)状态来解决由于路径故障造成的传输性能下降问题.具体过程为:当某一路径在传输过程中发生 RTO 时,将其设置为 PF 状态并且不再向该路径发送数据包(当所有的路径状态都变成 PF 时,数据包将在最后一条进入 PF 状态的路径上传输);在处于 PF 状态的路径上发送 Heartbeat(HB)包,并且在 HB 包丢失次数超过 PMR 之后,将路径状态设置为 Failed;当发送端收到 HB 应答包时将对应路径的状态设置为 Active,并将该路径的拥塞窗口设置为 1 个(或 2 个,发送端可以对该值进行设定) MTU,然后进入慢启动阶段.需要注意的是不能使用重传数据包的 SACK 作为将路径状态切换到 Active 的依据,因为不能确定该 SACK 是由重传数据包触发的还是由原始数据包触发的.

在启用该机制后,当路径 A 发生故障时,假定数据块 c 引发了 RTO.并且在重传定时器(T3 - rtx)超时之前,路径上又发送了 k 个 Data Chunk.丢失的数据块将在其他路径上(如路径 B)进行重传.根据多路径并行传输的发包机制,这些数据块将被标记为重传块,然后将它们填充进一个数据包中.若不能在一个数据包中发送全部的数据块,则发送第二个数据包.直到所有需要重传的数据块全部被正确接收,路径 B 才开始发送新的数据.

因此若在路径 B 上一共重传了 h 个数据包,则:

$$h = \left\lfloor \left(\frac{(k+1) * ck\_size}{MTU} \right) \right\rfloor \tag{13}$$

传输 h 次需要的时间为 $rtt(h) + \delta$,这个过程中发送数据包的总数为 $\mathrm{Min}(R_{tx} * (rtt(h) + \delta), k+1)$.当重传结束后,路径 B 开始传输新的数据.此时可以使用前文中介绍的算法进行计算.但需要分析路径 B 上 CW 的变化规律.我们分情况进行讨论.

2.2.1 路径 B 处于慢启动阶段时

当新确认的数据包大于 0 且"新收到数据包标志位"被置位时,CW 将被调整为:

$$Cwnd\ +\!= \mathrm{Min}(iNumNewlyAckedBytes, MTU) \tag{14}$$

其中 *iNumNewlyAckedBytes* 为新确认数据包的字节数,传输进行 h 次后,路径 B 的 CW 将增加 $(k+1) * ck\_size$.由于模型是以 MTU 作为 CW 的单位,可认为路径 B 的拥塞窗口增加了 h.

2.2.2 路径 B 处于拥塞避免阶段时

当新确认的数据包大于 0 且"新收到数据包标志位"被置位时:如果"部分确认字节数"小于当前的 CW,说明重传数据包的个数不足以使路径 B 的拥塞窗口增加;如果"部分确认字节数"大于当前的 CW,同时从路径 B 发送的"未被确认的数据包字节数"也大于 CW,路径 B 的拥塞窗口将增加 1.重传 h 次之后,路径 B 的拥塞窗口共增加

$$Cwnd\ +\!= \left\lceil \left(\frac{(k+1) * ck\_size}{Cwnd\_B} \right) \right\rceil \tag{15}$$

因此为了计算重传后的吞吐量,需要首先根据式(14)或(15)对拥塞窗口进行调整,然后使用发生超时后总的传输数据包个数除以总的发送时间.

3 模型的实现与优化

本文在 Matlab 平台上实现了上述算法.当模拟 n 轮发送过程时,使用数组 *cw* 和 *cw_old* 存放当前轮次和上一轮次中总的拥塞窗口大小,*qq* 和 *qq_old* 是它们对应的概率,数组 *cw_A* 和 *cw_B* 用于存放两条路径上拥塞窗口的值.数组 *cw_old_A* 和 *cw_old_B* 用于存放上一轮中两条路径上拥塞窗口的值.程序的伪代码如图 2 所示.

```
1) Initialize variables ;
2) for i from 1 to n do
    (i) for j from 1 to the length of cw do
          Save cw_A and cw_B to cw_old_A and cw_old_B ;
          Save cw and qq to cw_old and qq_old ;
          Calculate cw_A, cw_B, cw and qq ;
          Update the array of cw_A, cw_B, cw and qq;
    (ii) Calculate the value of ck and RTT ;
3) Calculate the throughput of CMT ;
```

图2 模型实现的伪代码

通过分析模型中拥塞窗口的变化规律不难发现,在第 i 轮次中,cw_i 的元素个数为 $4 * cw_{i-1}$,即 cw_i 中元素个数是以指数形式增长的($cw_i = 4^i$).因此直接使用上述算法进行计算效率非常低,需要找到一种优化算法来降低其复杂度.

在第二轮中,cw_2 由 16 个元素组成.当第一轮中两条路径的 CW 值相等时,对应在 cw_2 中生成的 4 个元素必然有两个是相等的.将 cw_1 中的 8 和 2 生成的元素进行合并,得到的结果为:[16 10 4]和[4 3 2].同时将对应的概率相加,所得结果如式(16)和(17)所示:

$$[q^8 \quad 2q^4(1-q^4) \quad (1-q^4)(1-q^4)] \tag{16}$$

$$[q^2 \quad 2q(1-q) \quad (1-q)(1-q)] \tag{17}$$

这一过程可以在生成 cw_2 时进行.完成第一次合并后 cw_2 的元素个数为 14 个.由于仍存在数值相同的元素,可以进一步合并.合并的原则是:cw_2 中数值相同的元素只保留一项,并将对应的概率相加作为该项的概率.

优化算法可归纳如下:假设当 $r-1$ 轮结束后 cw_{r-1} 里共有 m 个元素,则第 n 个元素 $cw_{r-1}(n)$ 表示整个关联拥塞窗口的第 n 种可能.令 $cw\_A_{r-1}(n)$ 和 $cw\_B_{r-1}(n)$ 分别为两条路径的拥塞窗口.考虑在第 r 轮中,若 $cw\_A_{r-1}(n)$ 和 $cw\_B_{r-1}(n)$ 不相等时.则可能出现如下四种情况:

$$\begin{bmatrix} 2(cw\_A_{r-1}(n)+cw\_B_{r-1}(n)) \\ 2cw\_A_{r-1}(n)+cw\_B_{r-1}(n)/2 \\ 2cw\_B_{r-1}(n)+cw\_A_{r-1}(n)/2 \\ cw\_B_{r-1}(n)/2+cw\_A_{r-1}(n)/2 \end{bmatrix} \tag{18}$$

$$\begin{bmatrix} qq_{r-1}(n)\,q^{cw\_A_{r-1}(n)}\,q^{cw\_B_{r-1}(n)} \\ qq_{r-1}(n)\,q^{cw\_A_{r-1}(n)}(1-q^{cw\_B_{r-1}(n)}) \\ qq_{r-1}(n)\,q^{cw\_B_{r-1}(n)}(1-q^{cw\_A_{r-1}(n)}) \\ qq_{r-1}(n)(1-q^{cw\_B_{r-1}(n)})(1-q^{cw\_A_{r-1}(n)}) \end{bmatrix} \tag{19}$$

当 $cw\_A_{r-1}(n)$ 和 $cw\_B_{r-1}(n)$ 相等时,式(18)和(19)可以化简为:

$$\begin{bmatrix} 2(cw\_A_{r-1}(n)+cw\_B_{r-1}(n)) \\ 2cw\_A_{r-1}(n)+cw\_B_{r-1}(n)/2 \\ cw\_B_{r-1}(n)/2+cw\_A_{r-1}(n)/2 \end{bmatrix} \tag{20}$$

$$\begin{bmatrix} qq_{r-1}(n)\,q^{cw\_A_{r-1}(n)}\,q^{cw\_B_{r-1}(n)} \\ 2*qq_{r-1}(n)\,q^{cw\_A_{r-1}(n)}(1-q^{cw\_B_{r-1}(n)}) \\ qq_{r-1}(n)(1-q^{cw\_B_{r-1}(n)})(1-q^{cw\_A_{r-1}(n)}) \end{bmatrix} \tag{21}$$

根据上述方法循环计算 m 次即可得到 cw_r 和 qq_r.此时 cw_r 中包含的元素个数$\in[3m,4m]$,这些元素能够进一步合并,从而保证 cw_r 中不存在数值相同的两个元素.合并过程中需要注意的是将 cw_r 中数值相同元素对应的 qq_r 中的值相加.使用优化算法之后,cw_i 的变化如图 3 所示.整个流程的伪代码如图 4 所示.

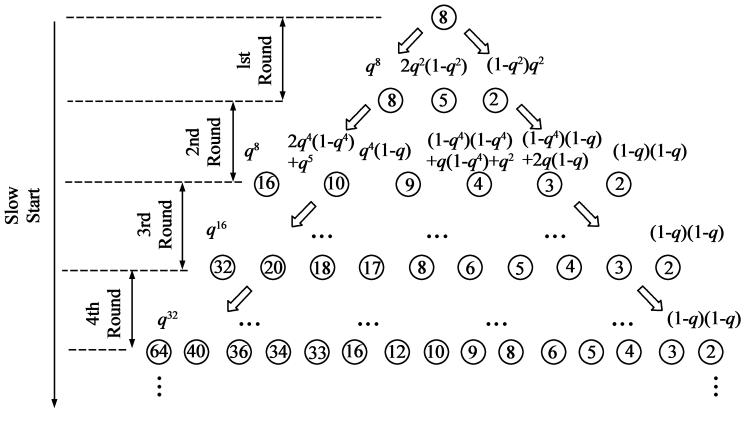

图3　优化后的拥塞窗口变化示意图

```
1) Initialize variables ;
2) for i from 1 to n do
    (i) for j from 1 to the length of cw do
        Save cw_A and cw_B to cw_old_A and cw_old_B ;
        Save cw and qq to cw_old and qq_old ;
        Calculate cw_A and cw_B ;
        if cw_old_A = cw_old_B then
            Calculate cw and qq according to (20) and (21) ;
        else
            Calculate cw and qq according to (18) and (19) ;
        Update the array of cw_A, cw_B, cw and qq;
    (ii) for k from 1 to the length of cw -1 do
        (a) for l from k+1 to the length of cw do
            if cw(k) = cw(l) then
                Delete cw_A(l) and cw_B(l) ;
                Delete cw(l) ;
                qq(k) += qq(l) ;
                Delete qq(l) ;
    (iii) Calculate the value of ck and RTT ;
3) Calculate the throughput of CMT ;
```

图4　简化后模型实现的伪代码

4　仿真结果及模型验证

为了验证模型的准确度,我们在 NS2-2.31[17] 上进行了仿真.图 5 为搭建的仿真环境拓扑结构图.

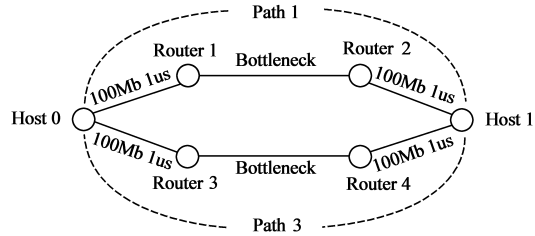

图5　仿真拓扑结构图

为了模拟不同的网络环境,在仿真过程中设置了不同的路径带宽、传输时延和丢包率.开启文献[6]中的 CUC、SFR 和 DAC 算法.Data Chunk 的大小设置为 1468 字节.使用 FTP 作为应用层协议.当没有 RTO 情况出现时,两条路径的丢包率设置为 0,0.015,0.03,0.045,0.06,0.075 和 0.09.比较结果如图 6 和图 7 所示.

图 6 是传输时延为 25ms、路径带宽为 1Mb 和 2Mb 情况下,仿真结果和模型计算结果的比较情况.从图中可以看出:在丢包率为 0 的情况下,模型能够很好的对传输吞吐量进行估计.随着丢包率的不断增加,模型的计算结果和仿真结果出现了偏差.同时路径带宽的不同也对模型的准确度有所影响.当带宽设置为 1Mb 时,仿真结果和模型计算结果相差较小.当带宽增加至 2Mb 时,两条曲线的间距变大,模型的准确度有所下降.

当传输时延为 50ms、路径带宽为 1Mb 和

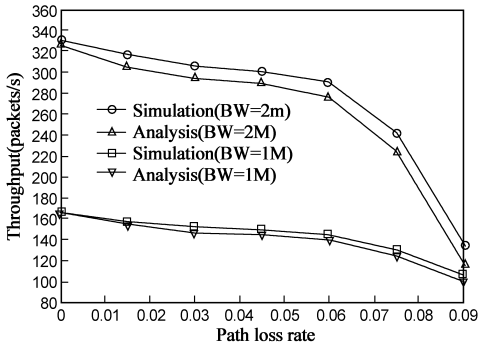

图6 仿真结果和模型计算结果的比较(传输时延25ms)

2Mb时,仿真结果和模型计算结果的比较情况如图7所示.两者的一致程度随着丢包率和路径带宽的增加而降低.和图6相比,当传输时延增加时,模型的准确度有所下降.但从整体来看,模型的计算结果仍能很好的反应吞吐量的变化趋势.

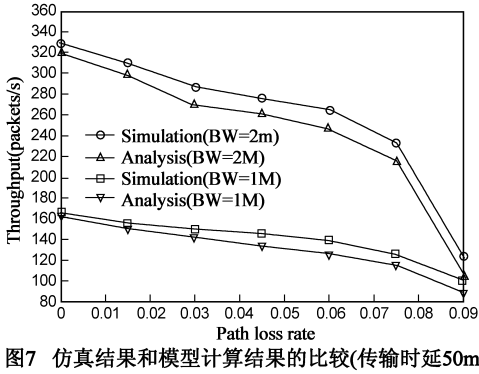

图7 仿真结果和模型计算结果的比较(传输时延50ms)

在考虑RTO对于吞吐量模型的影响时,我们设定路径丢包率为0,带宽为1Mb,开启CMT-PF算法.当路径A发生故障时,在RTO之后路径A将被置为PF状态.根据收到重传数据包的SACK时路径B所处的状态,我们分别进行比较(如图8和图9所示).

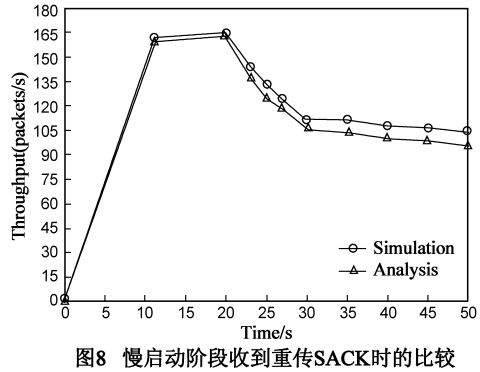

图8 慢启动阶段收到重传SACK时的比较

当设定路径A在20s发生故障时,丢失的数据包将在路径B上重新发送.当发送端收到重传数据包的SACK时,路径B处于慢启动阶段.此时仿真结果和模型计算结果的比较情况如图8所示.不难发现两者的一致性比较好,同时在路径A发生故障后,由于数据包只能在路径B上进行传输,因此总吞吐量有所下降.

若设定路径A在40s发生故障,当发送端收到重传数据包所对应的SACK时,路径B处于拥塞避免阶段.此时仿真结果和模型计算结果的比较情况如图9所示.我们能够得到和图8类似的结论.仿真结果表明,不论路径B处于何种状态,模型的准确度均比较理想.

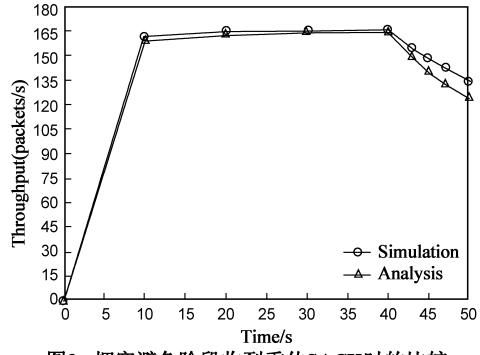

图9 拥塞避免阶段收到重传SACK时的比较

5 总结

本文在归纳单路径传输协议模型的基础上,提出了一种多路径并行传输情况下吞吐量的建模方法.根据路径上是否发生RTO将模型分为两种情况进行设计,并引入状态转换概率对拥塞窗口的增长方式进行分析.

通过观察模型中各种状态及其转换概率的特点,我们提出了一种优化算法:将拥塞窗口值相同的元素进行合并,然后将对应的概率相加,从而达到降低模型复杂度的目的.使用Matlab分别运行改进前后的程序可以发现,加入优化算法后的模型计算速度明显提高.

为了验证模型的准确度,我们选取了不同的路径带宽、传输时延和丢包率进行仿真.通过将模型的计算结果和仿真结果进行比较可以看出,该模型能够很好的预测传输吞吐量的变化趋势,具有较高的准确度.

参考文献:

[1] 张宏科,苏伟.新网络体系基础研究——一体化网络与普适服务[J].电子学报,2007,35(4):593-598.
Zhang Hongke, Su Wei. Fundamental Research on the Architecture of New Network-Universal Network and Pervasive Srevices [J]. Acta Electronica Sinica. 2007, 35(4): 593-598. (in Chinese)

[2] R Stewart, Q Xie, K Morneault, C Sharp, H Schwarzbauer, T Taylor, I Rytina, M Kalla, L Zhang, and V Paxson. Stream Control Transmission Protocol[S]. RFC 2960, 2000.

[3] E Kohler, M Handley, S Floyd. Datagram Congestion Control Protocol (DCCP)[S]. RFC 4340, 2006.

[4] A AI, T Saadawi, M Lee. LS-SCTP: a bandwidth aggregation technique for stream control transmission protocol[J]. Comput-

er Communications,2004,27(10):1012 – 1024.
[5] G Ye,T Saadawi,M Lee,IPCC-SCTP:an enhancement to the standard SCTP to support multi-homing efficiently[A]. In Proc.ICPCC[C].Phoenix,USA:2004.523 – 530.
[6] J Iyengar,P Amer,R Stewart.Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths[J]. Networking,IEEE/ACM Transactions on.2006,14(5):951 – 964.
[7] Z Yi,T Saadawi,M Lee,Analytic model of Stream Control Transmission Protocol[A].In Proc.PMCCS[C].Monticello,IL:2003.
[8] J Padhye,V Firoiu,D Towsley,J Kurose,Modeling TCP throughput:a simple model and its empirical validation[A].In Proc.ACM SIGCOMM[C].Canada:1998.303 – 314.
[9] Z Yi.Analytical Modeling of SCTP and Load Balancing SCTP [D].The City University of New York,2005.
[10] S Fu,M Atiquzzaman.Performance Modeling of SCTP Multihoming[A] In Proc.IEEE GLOBECOM[C].USA:2005.
[11] A Caro,J Iyengar,P Amer,G Heinz.Modeling SCTP Latency with Multihoming and Failovers[R].USA: University of Delaware,2002.
[12] N Cardwell,S Savage,T Anderson.Modeling TCP latency [A].In Proc.IEEE INFOCOM[C].Israel:2000.
[13] L Ma,F Yu,V Leung.Modeling SCTP throughput integrated wlan/cellular networks[A].In Proc.IEEE ICC[C].Seoul,Korea:2005.
[14] J Iyengar,P Amer,R Stewart.Receive buffer blocking in concurrent multipath transfer[A].In Proc.IEEE GLOBECOM [C].USA:2005.
[15] C Labovitz,A Abuja,A Bose,F Jahanian,Delayed Internet Routing Convergence[A].In Proc.ACM SIGCOMM[C]. Stockholm,Sweden:2000.
[16] P Natarajan,J Iyengar,P Amer.R Stewart.Concurrent multipath transfer using transport layer multihoming:Performance under network failures[A].In Proc.MILCOM[C].Washington,DC,USA:2006.
[17] NS-2 2.31.[Z].Univ.California,Berkeley,LBL,USC/ISI, and Xerox Parc,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

苏　伟　男,1978 年 10 月生于河北,博士,现为北京交通大学下一代互联网互联设备国家工程实验室讲师,主要研究方向为下一代互联网关键理论与技术.

(上接第 869 页)

[2] Younis M,Youssef M,Arisha K.Energy-aware routing in cluster-based sensor networks[A].Proceedings of the 10th IEEE International Symposium on Modeling,Analysis and Simulation of Computer and Telecommunications Systems[C].Fort Worth: IEEE Computer Society,2002.129 – 136.
[3] V Mhatre,C Rosenberg.Design guidelines for wreless sensor networks: communication.clustering and aggregation[J] Ad Hoc Network,2004,2(1):45 – 63.
[4] C Li,M Ye,G chen,J Wu.An energy-effieient unequal clustering mechanism for wireless sensor networks[A].Proeedings of the 2th IEEE International Conference on Mobile Ad-hoc and Sensor Systems[C].Washington,DC:IEEE,2005.597 – 604.
[5] Handy M J,Haase M,Timmermann D.Low energy adaptive clustering hierarchy with deterministic cluster-head selection [A].Proc of the 4th IEEE Conf on Mobile and Wireless Communications Networks[C].Stockholm: IEEE Communications Society:2002.368 – 372.
[6] Younis O,Fahmy S.HEED:a hybrid,energy-efficient,distributed clustering approach for ad hoc sensor networks[J].IEEE Transactions on Mobile Computing,2004,3(4):366 – 379.