软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(8):2041−2055 [doi: 10.13328/j.cnki.jos.004691]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于传输时延预测的多路径并发传输数据分配算法*

杜文峰, 赖力潜, 吴 真

(深圳大学 计算机与软件学院,广东 深圳 518060)

通讯作者: 杜文峰, E-mail: duwf@szu.edu.cn, http://www.szu.edu.cn

摘 要: 针对多路径并发传输模型的整体性能在路径性能存在差异时会急剧下降的原因进行分析,给出了获取通信路径传输时延的有效评估方案,并在此基础上提出了一种基于传输时延预测的多路径并发传输数据分配算法.该算法通过获取和预测数据块在各条路径上引入的传输时延,以按序到达为目标对多路径并发传输模型发送回合内和发送回合间的数据分配过程进行优化,能够有效地减少路径传输性能差异对多路径并发传输模型整体性能带来的影响.分析和实验结果表明,该算法相对于默认的轮询数据分配算法能够取得较好的运行性能.

关键词: 多路径并发传输;数据分配;性能优化

中图法分类号: TP393

中文引用格式: 杜文峰,赖力潜,吴真.基于传输时延预测的多路径并发传输数据分配算法.软件学报,2015,26(8):2041−2055. http://www.jos.org.cn/1000-9825/4691.htm

英文引用格式: Du WF, Lai LQ, Wu Z. Transmission delay prediction based data allocation scheme for concurrent multipath transfer. Ruan Jian Xue Bao/Journal of Software, 2015,26(8):2041−2055 (in Chinese). http://www.jos.org.cn/1000-9825/4691.htm

Transmission Delay Prediction Based Data Allocation Scheme for Concurrent Multipath Transfer

DU Wen-Feng, LAI Li-Qian, WU Zhen

(College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China)

Abstract: The performance of a CMT association degrades remarkably when the transmission capabilities of its paths are diverted. Based on the analysis of different network configurations, a transmission delay prediction based data allocation scheme is proposed to distribute data to different paths with a feasible delay measurement mechanism. To reduce the impact brought by out of order packet, the proposed scheme improves the data distribute process of inter and intra transmission round by accessing and predicting the arriving time of each packet in each path. The result of analysis and simulation reveal the performance of the presented scheme can achieve much better performance than the original round-robin scheme.

Key words: concurrent multipath transfer; data allocation; performance optimization

随着网络技术的快速发展,用户终端可以具备多种网络访问方式,能够同时接入并使用多种网络.然而,现有的TCP或UDP协议仅允许多宿主机通过某一种网络接入技术来访问网络提供的各种资源[1,2].为了充分利用网络终端的多种网络接入方式,Delaware 大学的协议工程实验室提出了多路径并发传输模型(concurrent multipath transfer,简称 CMT)[3],利用流传输控制协议(stream control transmission protocol,简称 SCTP)[4]偶联提供的多个数据通路来传输待发送数据的不同部分,实现数据各个部分的并发传输.

在现有的CMT传输模型中,当通信终端之间需要传输数据时,发送方将依次在各条路径上传输数据的不同

* 基金项目: 国家自然科学基金(61003271, 61170283); 国家高技术研究发展计划(863)(2013AA01A212); 深圳市技术研究开发计划(CXZZ20120820155332951)

收稿时间: 2013-12-10; 修改时间: 2014-05-09; 定稿时间: 2014-07-01

部分,即,通过轮询方式(round robin,简称 RR)在所有路径上依次发送数据.如果 CMT 传输模型的路径之间存在较大的传输性能差异,到达接收端的数据块将出现严重的乱序现象,频繁出现队头阻塞(head of line,简称 HOL),严重地影响了 CMT 传输模型的整体性能.

然而在实际网络环境中,通信路径可能由于经过多个同构或者异构的互联网络,或者经过不同状态的同构网络,通信路径在运行过程中的传输性能可能存在差异.如果将不同通信路径上的数据流进行同等对待,将对 CMT 传输模型的运行性能造成较大影响.本文针对 CMT 传输模型在差异化网络中的性能下降问题进行讨论,提出了一种基于传输时延预测的多路径并发传输数据分配算法(transmission delay prediction based data allocation scheme for CMT,简称 TDPDA).该算法通过评估通信路径的传输性能对数据块到达接收端的时间进行预测,并决定数据块在各种接入网络中的分配策略以改进 CMT 传输模型在差异化网络环境中的传输性能.

1 相关研究

利用多路径并发传输模型,多宿主机可以更好地利用多种网络接入进行数据传输,如图 1 所示.

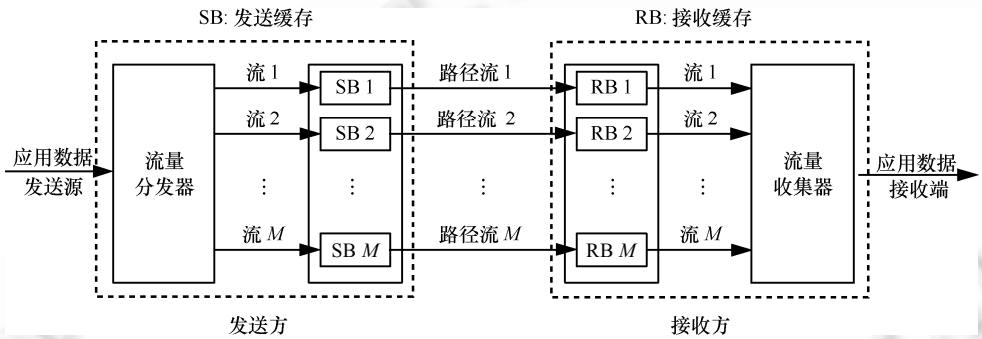

Fig.1 Concurrent multipath transfer

图 1 多路径并发传输

根据 SCTP 协议的设计,当接收端出现传输序列号(transmission sequence number,简称 TSN)不连续的数据块时,接收方将缓存数据块,并向发送端发送选择确认(selective acknowledgement,简称 SACK)控制块,通知对方当前的包间隔(gap report)、接收缓存的大小以及空缺的数据块[5].当某个已经发送的数据块的 TSN 在间隔报告中连续出现 4 次,发送方将启动快速重传机制来重新发送该数据块.

然而在 CMT 传输模型中,如果路径之间的传输性能差异较大,采用轮询算法按序发送的数据块可能由于经过不同的路径而引入不同的传输时延,部分 TSN 较大的数据块可能先于 TSN 较小的数据块到达[6],出现严重的乱序现象.数据块的乱序到达将触发快速重传过程的误启动,导致不必要的拥塞控制窗口变化,严重影响了 CMT 传输模型的数据传输效率.同时,由于接收方的缓存容量有限,当接收端的缓存中塞满了乱序到达的数据块时,将导致接收缓存溢出.此时,接收端只能接收空缺的数据块,丢弃所有到达的后续数据块.

目前,针对 CMT 传输模型的研究主要集中在传输性能分析、路径选择、流量分割等方面.文献[7]中提出了建立路径和子流的概念,能够在限制流速情况下对缓冲区负载有较好的改善;文献[8]采用包分割方式,按照路径的权重去分配数据,尽可能地实现路径间的负载均衡;文献[9]通过流分割方式,按照特定哈希函数来选择路径,减少了接收端的乱序;文献[10]提出了按最大最小路径传输延时之差进行流量分割的方法,将单个流切分成多个流分片,每个分片按流量均衡算法分配到相应路径,较好地保证了路径间的负载均衡与包的有序性;文献[11]基于真实网络环境中的数据块统计分析,以时间为界限划分流片,更好地保证了路径负载均衡,提高了系统性能;文献[12]提出了最少负载路径优先的调度原则,避免了路径负载过重问题;文献[13]中提出了一种基于分组到达时间的负载均衡算法 ATLB(arrival-time matching load-balancing),通过估算每条路径的发送队列中最后一个分组到达接收端的时延,把将要发送的分组分配给时延最小的路径,确保在多条路径上并发传输的分组尽可

能地按序到达接收端;文献[14]中提出以最后离开数据发送缓冲区的时间和最早进入数据发送缓冲区的时间之差与数据发送缓冲区大小的比值作为评价路径性能的标准,并且数据调度采用的是路径性能和拥塞窗口之积来计算路径的数据发送能力;文献[15]中建立路径的端到端延时的函数,通过最小最大值方法求出最适合的分割流向量,并对延时的变化和乱序进行优化;文献[16]中针对 SCTP 各条路径的可用带宽进行估计,通过设置各条通信路径的初始拥塞窗口和慢启动阈值来调整各个数据流的发送速度;文献[17]中提出了一种数据块乱序调度方法,根据不同路径的延时不同,使乱序发送的数据块能按序到达;文献[18]中提出了最大最小路径延时算法,并获取最佳的流量分割;文献[19]计算出传输过程中各个环节产生的延时,通过选择延时最小的路径来发送数据,同时考虑路径重传问题.

目前国内外针对 CMT 传输模型如何在差异化网络环境中进行多路径并发传输的研究还比较少.如何评估通信路径的传输性能以及如何协调多条路径的数据传输,成为影响整个 CMT 传输模型服务质量的重要因素[20,21].本文针对差异化网络对 CMT 传输模型的性能影响进行分析,提出了一种基于传输时延预测的多路径并发传输数据分配算法,通过对数据块到达接收端的时间进行分析、预测,降低接收方数据块到达的乱序度,确保整个 CMT 传输模型在差异化网络环境中的传输性能.

2 基于传输时延预测的多路径并发传输数据分配算法

为了减少路径传输性能差异对多路径并发传输模型带来的影响,本文首先给出了一种路径传输时延评估方案,并在此基础上讨论数据块在各路径上的分配过程.

2.1 通信路径传输时延评估方案

本文在数据块的头部加入时间戳,记录数据块的发送时刻.令时延 d 是发送端开始传输数据块的第 1 个字节到接收端接收到该数据块的最后一个字节所需要的时间.根据文献[22]所述,时延 d 主要包含节点处理时延 d_{proc}、排队时延 d_{queue}、传输时延 d_{trans} 和传播时延 d_{prop}.

为了评估路径的实时传输性能,发送端将两个大小一致的数据块放入某一通信路径的发送队列.可以得到第 1 个数据块在传输过程中引入的时延 d_1 为

$$d_1=d_{proc}+d_{queue}+d_{trans}+d_{prop}\approx d_{queue}+d_{trans}+d_{prop} \tag{1}$$

第 2 个数据块必须等待第 1 个数据块完全发送完毕后才能开始发送,因此,第 2 个数据块在传输过程中引入的时延 d_2 为

$$d_2=d_{proc}+d_{queue}+2\times d_{trans}+d_{prop}\approx d_{queue}+2\times d_{trans}+d_{prop} \tag{2}$$

可以看出,路径的传输时延 d_{trans} 为数据块 1 和数据块 2 在传输过程中引入的时延差:

$$d_{trans}=d_2-d_1 \tag{3}$$

若传输数据块的大小固定为 $dataSize$,则路径的吞吐量 R 可以表示为

$$R=dataSize/d_{trans} \tag{4}$$

在网络传输性能最优的情况下,通信链路的排队时延将会变得很小.此时,数据块在传输过程中引入的时延 $d_{\min}$ 约为

$$d_{\min}=d_{trans}+d_{prop} \tag{5}$$

而路径的传播时延 d_{prop} 为

$$d_{prop}=d_{\min}-d_{trans} \tag{6}$$

在传输过程中,接收端将记录所有通信路径最近一次传输的数据块的发送时间和传输时延.若当前接收的数据块的发送时间与上次接收的数据块的发送时间相同,则认为这两个数据块为同时进入该路径发送队列的数据块.根据上述原理,可以通过当前数据块与上次接收的数据块之间的时延差来获得路径传输时延:$d_{trans\_new}=d_2-d_1$.若当前数据块引入的传输时延低于记录的最小时延 $d_{\min}$,则更新路径的最小时延.

为了减小路径传输性能突发变化对路径性能评价造成的影响,本文采用了类似于 TCP 协议的 RTT 更新方法来管理路径传输时延.可以得到,路径传输时延为

$$d_{trans}=ALPHA\times d_{trans\_old}+(1-ALPHA)\times d_{trans\_new} \quad (7)$$

其中,*ALPHA* 为平滑系数,取值 0.75.

与此同时,接收端将在 SACK 数据块中加入各条通信路径的传输性能信息,通过 SACK 数据块,将路径的实际传输性能以及可用通信路径数量告知发送终端.发送端分析 SACK 数据块的头部来定位各路径性能参数的位置,并利用该信息来协助数据分发.

2.2 最早到达优先路径选择方案

根据 CMT 传输模型的运行原理,发送端将在接收窗口允许的前提下向接收端的各个目的 IP 发送不超过其路径拥塞窗口 *cwnd* 个数据块.本文定义 CMT 传输模型发送端每次调用数据块的发送过程为一个发送回合.根据数据发送量与接收窗口之间的关系,发送方必须保证发送回合内和发送回合之间的数据块有序到达.

2.2.1 发送回合内的数据块有序到达

如果待发送的数据量低于接收端的接收窗口,那么发送端将在一个发送回合内完成数据分配.

发送端将在各通信路径空闲时间 t_{idle} 的基础上预测下一个数据块的到达时间 t_{arrive},并选择在预测到达时间最小的通信路径上分配一个大小为最大传送单元(maximum transmission unit,简称 MTU)的数据块.

$$t_{arrive}=t_{idle}+d_{trans}+d_{prop} \quad (8)$$

分配数据块后,发送端将更新通信路径的重新进入空闲时间 t_{idle_new}、数据发送量以及 CMT 接收窗口:

$$t_{idle\_new}=t_{idle\_old}+d_{trans} \quad (9)$$

此过程循环进行,直到接收窗口或者所有通信路径的 *cwnd* 全部用完为止.

图 2 给出了一个发送回合内数据块有序到达的例子,其中,宽度代表时间跨度;分块代表标准的数据块,大小为 1MTU.为了简化说明,发送终端只接入到两种网络中,且分别用路径 *A* 和路径 *B* 表示.路径 *A* 的传输时延为 d_{tran_A},传播时延为 d_{prop_A};路径 *B* 的传输时延为 d_{tran_B},传播时延为 d_{prop_B}.在发送回合开始之前,路径 *A* 和路径 *B* 的拥塞窗口空闲值分别为 4MTU 和 3MTU.

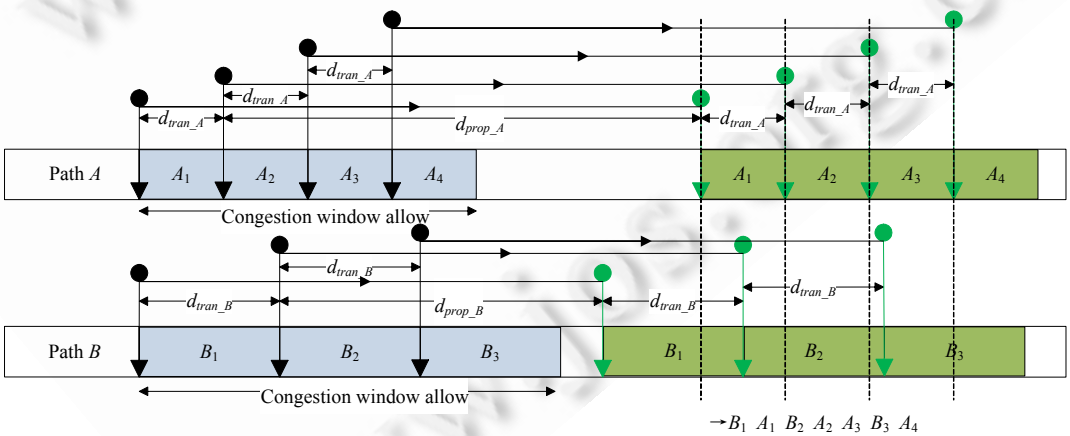

Fig.2 A sample of orderly data distribution in one round transmission

图 2 发送回合内的有序数据分发案例

当发送回合开始后,发送端将分析和比较数据块 A_1 的到达时间 $d_{tran\_A}+d_{prop\_A}$ 和数据块 B_1 的到达时间 $d_{tran\_B}+d_{prop\_B}$.因为数据块 B_1 的到达时间小于数据块 A_1 的到达时间,发送端优先在路径 *B* 中分配一个数据块.同时,发送端更新路径 *B* 重新进入空闲的时间 $t_{idle\_B}=t_{idle\_B}+d_{trans\_B}$,路径 *B* 的拥塞窗口空闲值变为 2MTU.

然后,发送端比较数据块 A_1 与数据块 B_2 的到达时间.数据块 A_1 的到达时间为 $d_{tran\_A}+d_{prop\_A}$,数据块 B_2 的到达时间为 $t_{idle\_B}+d_{tran\_B}+d_{prop\_B}=2\times d_{tran\_B}+d_{prop\_B}$.从图 2 中可以看出,数据块 A_1 的到达时间早于数据块 B_2 的到达时间,发送端在路径 *A* 中分配数据块 A_1,并更新路径 *A* 的重新进入空闲时间 $t_{idle\_A}=t_{idle\_A}+d_{trans\_A}$.此时,路径 *A* 的

拥塞窗口空闲值变为 3MTU.

接着,发送端将比较数据块 A_2 与数据块 B_2 的到达时间,其中,数据块 A_2 的到达时间为 $2\times d_{tran\_A}+d_{prop\_A}$,数据块 B_2 的到达时间为 $2\times d_{tran\_B}+d_{prop\_B}$.可以得出,数据块 B_2 的到达时间早于数据块 A_2 的到达时间.此时,发送端将在路径 B 上分配数据块 B_2.

以此类推,可以得到发送端最终的数据分配方案为 $B_1,A_1,B_2,A_2,A_3,B_3,A_4$.此时,接收端在本回合内的数据块接收序列如图 3 所示.

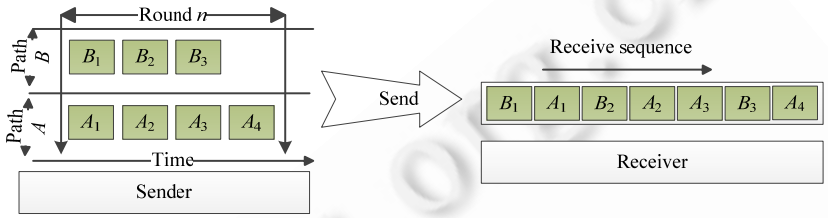

Fig.3 Receive sequence of data packet in one round transmission

图 3 发送回合内的数据块接收序列

CMT 默认的轮询数据块分配算法由于未考虑路径的传输性能差异,其数据块分配的 TSN 为 A_1=1,A_2=2,A_3=3,A_4=4,B_1=5,B_2=6,B_3=7.接收端的接收序列为 5 1 6 2 3 7 4,序列的逆序数为 8;若采用本文提出的基于传输时延预测的数据块分配策略,且路径性能预估准确,数据块分配的 TSN 为 A_1=2,A_2=4,A_3=5,A_4=7,B_1=1,B_2=3,B_3=6.其接收序列为 1 2 3 4 5 6 7,序列的逆序数为 0.

2.2.2 发送回合间的数据块有序到达

尽管发送回合内的数据块有序到达方案可以保证一次传输过程中的数据块按序到达,然而当连续进行多次数据传输时,发送回合之间依然存在着数据块乱序到达的可能性.为此,本文在发送回合内数据块有序到达方案的基础上,对下一发送回合的数据块分配过程进行优化.

由于待发送的数据量超过了一个发送回合内允许发送的数据,发送端将通过多个发送回合来传输数据.发送端首先采用发送回合内的数据块有序到达方案来分配数据,直到任一通信路径的拥塞窗口空闲值为 0.

此时,发送端将预估所有路径在下一发送回合的数据块最早到达时间 $t_{pathmin}$,其值为路径的重新进入空闲时间 t_{idle}、等待下回合触发发送时间 t_{wait} 和下回合数据块时延 $d_{trans}+d_{prop}$ 之和:

$$t_{pathmin}=t_{idle}+t_{wait}+d_{trans}+d_{prop} \tag{10}$$

若 $t_{pathmin}$ 小于下一发送回合的数据块可能最早到达时间 $t_{\min}$,则令 $t_{\min}=t_{pathmin}$.由于 t_{wait} 在具体通信协议中是一个动态的值,与协议的具体设置(延迟 SACK)有关,本文设置 t_{wait} 为 $d_{trans}+d_{prop}$ 作为缓冲.

在分配数据块前,发送端将预测和比较数据块在各路径上的到达时间,得到数据块的最小到达时间 t_{round}:

(1) 如果 $t_{round}<t_{\min}$,那么数据块在指定路径上的分配不会影响发送回合之间的数据块按序到达,此时,发送端将向该路径分配数据块,并在当前发送回合内完成数据传输.

(2) 如果 $t_{round}>t_{\min}$,那么本回合分配的数据块将在下一发送回合的最早到达时间之后被收到,数据块的分配将造成发送回合间的数据乱序.此时,本算法停止为该路径分配数据,直接开始下一回合的数据块分配.

图 4 是一个多回合的数据块分配例子.路径 A 和路径 B 的拥塞窗口空闲值分别为 2MTU 和 3MTU,且路径 A 的性能远远优于路径 B.

如果采用 CMT 默认的轮询数据分配机制,各个数据块分配的 TSN 为 A_1=1,A_2=2,B_1=3,B_2=4,B_3=5,A_3=6,…虽然在第 n 个发送回合内的数据块接收序列为 1 2 3 4 5,但是在第 n+1 个发送回合内的数据块 A_3 却早于第 n 个发送回合内的数据块 B_2,B_3 到达,造成接收端的数据块乱序现象.当路径 A 与路径 B 的传输性能差异较大时,数据块乱序情况更加严重.

若采用本文提出的基于传输时延预测的数据块分配算法,发送端首先根据发送回合内的数据块有序到达方案在路径 A 上分配数据块 A_1 和 A_2,然后再利用回合间的数据块有序到达方案来计算下一发送回合的最早到达时间 $t_{pathmin}=t_{idle}+t_{wait}+d_{trans}+d_{prop}=t_{idle}+2\times d_{trans}+2\times d_{prop}$(图中从左往右第 2 条虚线指示所在).此时,发送端可以预估数据块 A_3 在第 n+1 个发送回合的到达时间,并禁止超过数据块 A_3 到达时间的数据块 B_2,B_3 发送.此时,数据块的发送序列号为 A_1=1,A_2=2,B_1=3,A_3=4,…,如图 5 所示.

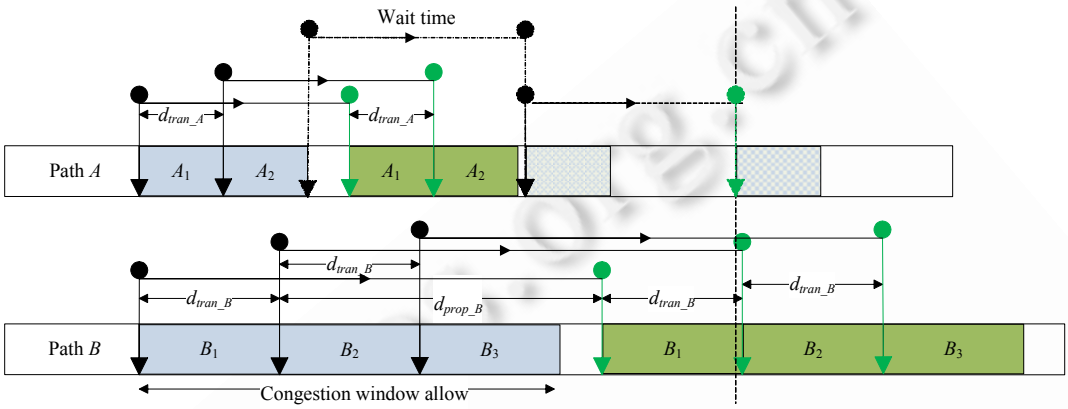

Fig.4 A sample of orderly data distribution in multi rounds transmissions

图 4 发送回合间的有序数据分发案例

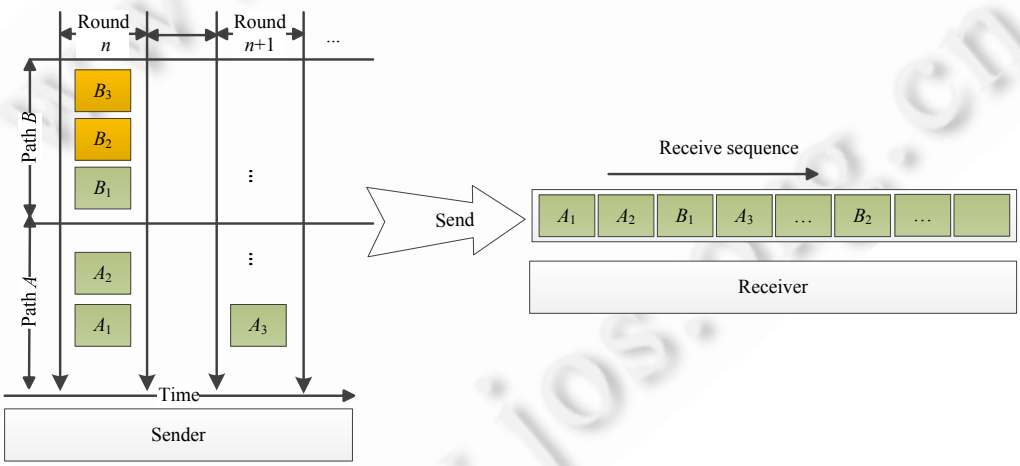

Fig.5 Receive sequence of data packet in multi rounds transmissions

图 5 发送回合间的数据块接收序列

3 性能分析

在分析算法的性能之前,先做以下假设:整个 CMT 传输模型由 M 条独立的通信路径组成,且第 i 条路径 P_i 的业务到达满足速率为 λ_i 的泊松分布,传输时延满足均值为 μ_i 的负指数分布,即 d_{trans_i}.由于本算法为各条路径设置独立的接收缓存 A_i,即,路径 P_i 的接收缓存空闲值为 A_i,路径的传输过程可以建模为 $M/M/1/A_i$ 的排队过程.

假设在时刻 t,路径 P_i 中已经有了 k 个数据块,$0\leqslant k\leqslant A_i$.令 Δt 为一个足够小的时间段,以至于在该时间段内路径 P_i 中仅有 1 个数据块需要发送,或者仅能进行 1 次数据块发送操作.即,路径 P_i 在时间段 $(t,t+\Delta t)$ 内被分配到的需要发送的数据块为 $\lambda_i\times\Delta t+o(\Delta t)$;同时,路径 P_i 中被发送完的数据块为 $k\times\mu_i\times\Delta t+O(\Delta t)$.

根据路径使用状态转移的马尔可夫性,可以得到路径 P_i 在任意时刻有 k 个待发送数据块的概率 $p_{i,k}$ 为

$$p_{i,k}=\begin{cases}\left(\dfrac{\lambda_i}{\mu_i}\right)^n\times p_{i,0}=\rho_i^n\times p_{i,0}, & n=0,1,2,...,A_i\\ 0, & n\geqslant A_i\end{cases} \tag{11}$$

其中,$\rho_i=\lambda_i/\mu_i$.

当$\rho_i<1$ 时,有:

$$p_{i,0}=\left[\sum_{n=0}^{A_i}\left(\frac{\lambda_i}{\mu_i}\right)^n\right]^{-1}=\left[\frac{1-\left(\dfrac{\lambda_i}{\mu_i}\right)^{A_i-1}}{1-\dfrac{\lambda_i}{\mu_i}}\right]^{-1}=\frac{1-\rho_i}{1-\rho_i^{A_i+1}} \tag{12}$$

可以得到,在路径 P_i 中传输的平均数据块数量 L_i 为

$$L_i=\sum_{k=0}^{A_i}k\times p_{i,k}=\frac{1-\rho_i}{1-\rho_i^{A_i+1}}\times\rho_i\times\sum_{n=0}^{A_i}\frac{d}{d\rho_i}(\rho_i^n)=\frac{1-\rho_i}{1-\rho_i^{A_i+1}}\times\rho_i\times\sum_{n=0}^{A_i}\frac{d}{d\rho_i}\left(\frac{1-\rho_i^{A_i+1}}{1-\rho_i}\right)=\frac{\rho_i}{1-\rho_i}-\frac{(A_i+1)\times\rho_i^{A_i+1}}{1-\rho_i^{A_i+1}} \tag{13}$$

同时,路径 P_i 中的平均等待数据块数量 L_{qi} 为

$$L_{qi}=\sum_{k=0}^{A_i}(k-1)\times p_{i,k}=L_i-(1-p_{i,0}) \tag{14}$$

由于路径 P_i 的接收缓存空闲值有限,只能接收 A_i 个数据块.当路径 P_i 处于状态 A_i 时,该路径将不接收新的数据块,即,此路径接收新数据块的概率为 $1-p_{i,A_i}$.因此,可以得到单位时间内实际进入路径的数据块数量的平均值为

$$T_i=\sum_{k=0}^{\infty}\lambda_i\times p_{i,k}=\sum_{k=0}^{A_i-1}\lambda_i\times p_{i,k}=\lambda_i\times(1-p_{i,A_i})=\mu_i\times(1-p_{i,0}) \tag{15}$$

同样,可以求出数据块的平均逗留时间 W_i 和平均等待时间 W_{qi} 为

$$W_i=\frac{L_i}{T_i}=\frac{L_i}{\lambda_i\times(1-p_{i,A_i})} \tag{16}$$

$$W_{qi}=\frac{L_{qi}}{T_i}=\frac{L_{qi}}{\lambda_i\times(1-p_{i,A_i})} \tag{17}$$

且 $W_i=W_{qi}+1/\mu_i$.

根据 CMT 的拥塞控制原理,数据块到达接收方后,接收方将缓存所有已收到的数据块,并将低于当前 SACK 对应 TSN 的所有数据块提交给上层应用,并从接收缓冲区中释放存储空间.当接收方向上层应用提交数据块后,接收方将通过 SACK 数据块通知发送方当前接收缓存的大小,允许发送方根据情况发送数据.因此,使用不同数据分配算法的 CMT 传输模型的传输性能将由接收缓冲区的释放速度和执行条件决定.

3.1 轮询数据分配算法

在 CMT 传输模型默认的数据分配过程中,发送方将在多条路径中轮询分配数据,即,按一定顺序将数据分配给各条通信路径.当有数据需要发送时,各条路径获得数据的概率是相同的,即,各条路径获得数据的概率 p 为 $1/M$.

类似于文献[23],可以得到数据块在整个 CMT 传输模型中的平均逗留时间 W 为

$$W=\max(W_1,W_2,\ldots,W_M) \tag{18}$$

且数据块在传输过程中的平均等待时间 W_q 为

$$W_q=\max(W_{1q},W_{2q},\ldots,W_{Mq}) \tag{19}$$

3.2 基于传输时延预测的多路径并发传输数据分配算法

在本算法中,发送端将充分利用各条路径的传输性能,允许传输性能较好的通信路径进行多次传输.由于各

条路径在传输过程中是相互独立的,整个 CMT 传输模型的吞吐量并不会受到性能较差的通信路径的影响.在 D 时间范围内,各条路径的传输次数如图 6 所示.

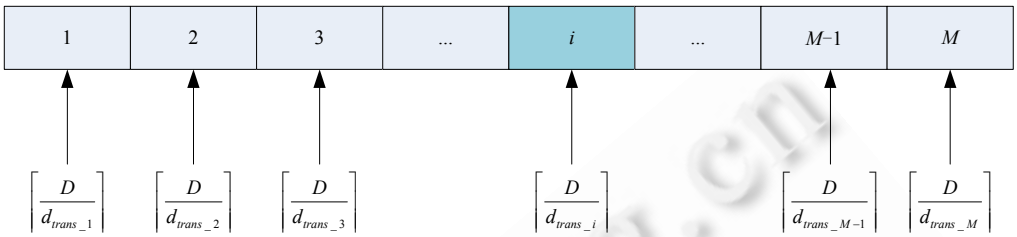

Fig.6 Principle of transmission in each path

图 6 路径传输原理

同时,定义路径 P_i 的传输次数占总分配次数的比例为路径 P_i 在传输过程中获得数据块的概率 p_i,则

$$p_i=\frac{\frac{D}{d_{trans\_i}}}{\sum_{j=1}^{M}\frac{D}{d_{trans\_j}}}=\frac{\frac{1}{d_{trans\_i}}}{\sum_{j=1}^{M}\frac{1}{d_{trans\_j}}} \tag{20}$$

可以看到,传输时延较大的路径在数据分配过程中获得数据块的概率相对较小.因此,本算法能够降低传输性能较差的路径对 CMT 传输模型带来的影响.

根据上一节给出的数据分配算法以及数据块传输序号设置原理,可以得到受路径 P_j 传输性能影响而被阻塞的路径数量 K_j 为

$$K_j=\sum_{t=1,t\neq j}^{M}d(t)\times\frac{d_{trans\_t}}{d_{trans\_j}} \tag{21}$$

其中,$d(t)$为判断传输时延比路径 P_j 的传输时延小的路径是否完成传输的标识函数.$d(t)$的取值情况如下:

$$d(t)=\begin{cases}1, & d_{trans\_j}<\left\lceil\frac{d_{trans\_j}}{d_{trans\_t}}\right\rceil\times d_{trans\_t}\\ 0, & \text{other}\end{cases} \tag{22}$$

因此,使用本算法的 CMT 传输模型在每一个 D 时间内能够传输的数据量 K 为

$$K=\sum_{i=1}^{M}\frac{D}{d_{trans\_i}} \tag{23}$$

此时,数据块在 CMT 传输模型中的平均逗留时间 W 为

$$W=\sum_{i=1}^{M}p_i\times W_i \tag{24}$$

且数据块在传输过程中的平均等待时间 W_q 为

$$W_q=\sum_{i=1}^{M}p_i\times W_{iq} \tag{25}$$

由于本算法结合路径的传输时延来为各条路径分配数据,其算法复杂度为 $O(kn)$,相对于当前 CMT 传输模型默认的轮询数据分配算法的复杂度 $O(n)$并未有较大的提高.

4 仿真实验结果及分析

通过对 NS2 中的 sctp.cc 和 sctp-cmt.cc 等文件进行修改,本文在 CMT 轮询数据分配算法的基础上实现了基于传输时延预测的多路径并发传输数据分配算法和 ATLB 算法,并对 3 种数据分配算法在路径传输性能差异较小和较大情况下的数据分配过程进行了仿真.

为了统计数据块到达接收端的乱序程度,本文在仿真过程中引入了逆序数.如果数据块到达接收端时,接收端已经收到了 TSN 大于该数据块 TSN 的其他数据块,则该数据块与之前到达的数据块为乱序到达.此时,逆序数为接收端已经接收到的 TSN 大于当前数据块 TSN 的数据块数量.

仿真过程中的参数设置见表 1.

Table 1 Simulation parameters

表 1 实验参数

参数	初始值	参数	初始值
MTU	1 500bytes	IP header size	20bytes
数据块大小	1 468bytes	Path.Max.Retrans	5
初始接收窗口	65 536bytes	初始慢启动阈值	65 536bytes
初始拥塞窗口	2MTU	是否可靠	True
是否按序	false	带宽	10Mbps
延迟	50ms	–	–

本文在实验过程中通过控制 MTU 和数据块的大小,确保一个 MTU 中最多包含 1 个数据块.

4.1 路径传输性能差异较小情况下的仿真结果

路径传输性能差异较小情况下的仿真拓扑设置如图 7 所示.

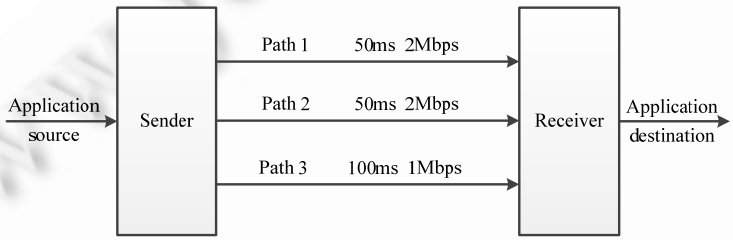

Fig.7 Simulation topology of CMT with low performance difference

图 7 低差异多路径并发传输仿真拓扑

图 8 展示了使用 TDPDA 算法、ATLB 算法与轮询数据分配算法(standard)的 CMT 传输模型在低差异网络情况下的主要性能对比.

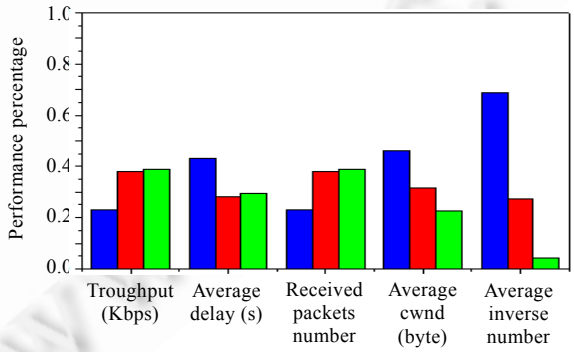

Fig.8 Simulation result of CMT with low performance difference

图 8 低差异多路径并发传输中的仿真结果

由图 8 可以看出,TDPDA 算法与 ATLB 算法的所有性能指标均优于轮询数据分配算法.TDPDA 算法的整体吞吐量相对于轮询数据分配算法提高了 68%.虽然 TDPDA 算法无法达到其最大性能(2Mbps+2Mbps+1Mbps),但是也从另一个侧面证明了 TDPDA 算法不会超过其链路容量进行超额的数据发送,避免导致数据拥

塞,且数据块的平均逆序数比轮询数据分配算法减少 94%.TDPDA 算法在路径的整体吞吐量、平均时延、接收数据包数量等性能指标上与 ATLB 算法表现一致,但平均拥塞窗口比 ATLB 算法有所改进.与此同时,由于采用了基于传输时延预测的按序到达机制,TDPDA 算法能够有效地避免数据块乱序到达,其数据块平均逆序数与轮询数据分配算法和 ATLB 算法相比均有大幅度的改进.

图 9 给出了低差异网络环境中 TDPDA 算法、ATLB 算法与轮询数据分配算法的逆序数对比图.由图中可以看出,相同时间内,TDPDA 算法可以接收更多的数据块,且逆序数处于 0~1.2 的区域内,相对集中;而标准 CMT 传输模型接收的数据块较少,且逆序数波动较大;尽管 ATLB 算法的逆序数比标准 CMT 传输模型小,然而其逆序数相对 TDPDA 算法有所增加.由于乱序到达的数据块无法移交给应用层,只能阻塞于接收端的接收缓存中,逆序数的大小直接体现了接收缓存的拥塞程度.图 10 展示了 3 种算法的接收缓存实时变化结果,可以看出:使用 TDPDA 机制的 CMT 传输模型的接收缓存中拥有更多的空闲空间;CMT 标准数据分配算法的接收缓存空间在 3 种算法中的表现最差;ATLB 算法相对于 CMT 标准算法在接收缓存空间上有较大改进,但其算法性能的稳定性不如 TDPDA 算法.

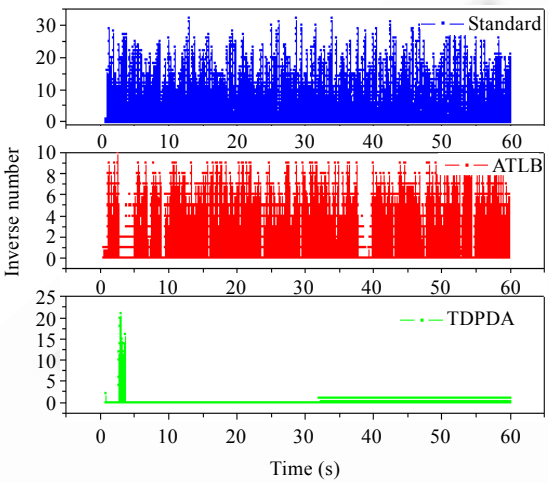

Fig.9 Inverse number of CMT with low performance difference

图 9 低差异多路径并发传输中的逆序数

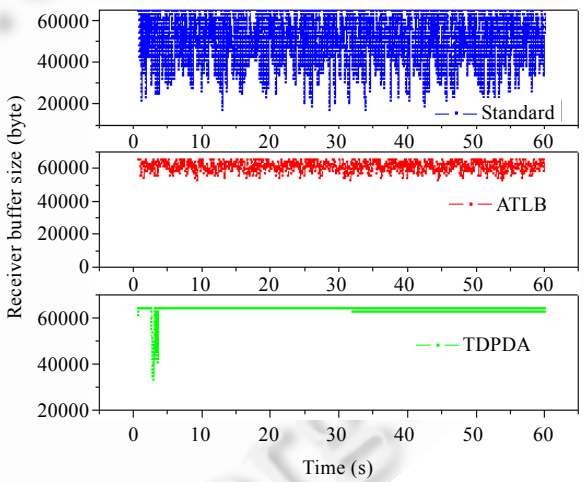

Fig.10 Receiver buffer of CMT with low performance difference

图 10 低差异多路径并发传输中的接收缓存

图 11 给出了使用 3 种数据分配算法的 CMT 传输模型拥塞窗口对比结果.为简单起见,本算法将各条路径的拥塞窗口进行求和.可以看出,由于 TDPDA 数据分配算法能够减少或者避免传输性能较差的通信路径,性能较差的通信路径由于未分配到数据块而保持较小的拥塞窗口.因此,使用 TDPDA 算法的 CMT 传输模型所对应的拥塞窗口比标准 CMT 传输模型要少一些.尽管 ATLB 算法选择了时延较小的通信路径传输数据,但其算法稳定性不如 TDPDA 算法,对应的拥塞窗口比 TDPDA 算法有小幅的增加.

图 12 给出了低差异网络环境中使用 3 种数据分配算法的 CMT 传输模型在传输时延方面的对比.可以得出:使用 TDPDA 算法的 CMT 传输模型在传输过程中引入的时延比标准 CMT 和使用 ATLB 算法的偶联要少,主要集中在 0~0.09 范围内,且时延抖动较小;标准 CMT 传输模型的传输时延较大,且传输时延抖动范围较广,多次出现数据块接收空闲时间;ATLB 算法相对于 CMT 标准算法,在传输时延和时延抖动等方面均有改进.

在低差异网络环境中,使用 3 种数据分配算法的 CMT 传输模型的吞吐量对比如图 13 所示.可以得出:由于 TDPDA 算法在开始阶段需要对通信路径的传输性能进行预测评估,使用 TDPDA 数据分配算法的 CMT 传输模型在仿真初期的吞吐量低于标准 CMT 传输模型和使用 ATLB 算法偶联的吞吐量;随着路径传输性能预测过程的结束,使用 TDPDA 算法的 CMT 传输模型所获得的吞吐量会逐渐超过标准 CMT 传输模型以及使用 ATLB 算

法的 CMT 传输模型的吞吐量.

图 14 给出了使用 3 种数据分配算法的 CMT 传输模型的各路径吞吐量变化图.可以得出:TDPDA 算法经过初期的路径传输性能预测后,能够利用预测结果来协助数据块分配,可以有效地避免数据块乱序到达造成的接收缓存溢出,其路径吞吐量相对稳定且优于标准 CMT 的各路径吞吐量;ATLB 算法由于选择传输时延小的路径传输数据,能够有效减少数据乱序现象.尽管 ATLB 算法相对于 CMT 默认数据分配算法能够取得较好的吞吐量,但其吞吐量仍略低于使用 TDPDA 算法的 CMT 传输模型的各路径吞吐量.

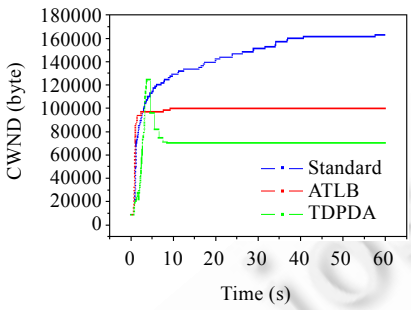

Fig.11 Congestion window of CMT with low performance difference

图 11 低差异多路径并发传输中的拥塞窗口

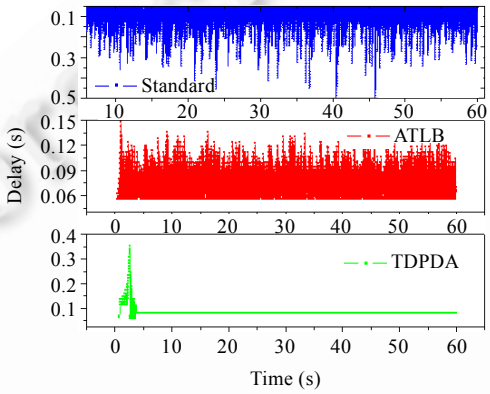

Fig.12 Delay of CMT with low performance difference

图 12 低差异多路径并发传输中的时延

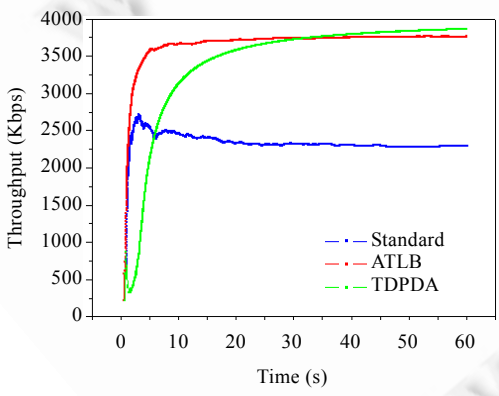

Fig.13 Throughput of CMT with low performance difference

图 13 低差异多路径并发传输中的吞吐量

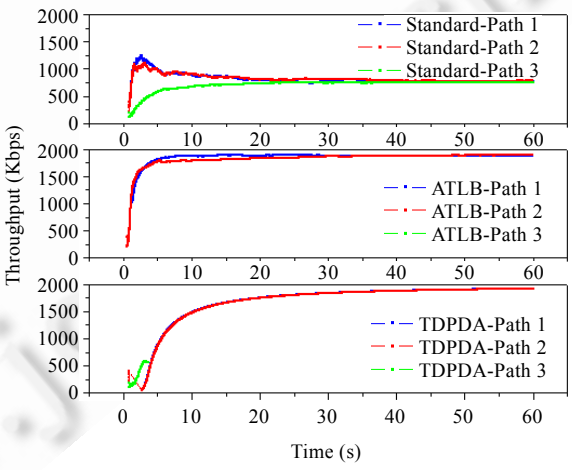

Fig.14 Throughput of paths in low performance difference CMT

图 14 低差异多路径并发传输中的路径吞吐量

4.2 路径传输性能差异较大情况下的仿真结果

为了提高路径之间的传输性能差异,本次仿真将路径 3 的传输时延由 100ms 增加到 500ms.仿真拓扑设置如图 15 所示.

图 16 给出了使用 TDPDA 算法、ATLB 算法与轮询数据分配算法的 CMT 传输模型在高差异网络环境下的主要性能对比.可以得出:

(1) 由于标准 CMT 传输模型受到传输性能较差的链路影响,整体性能有较大的降低,其吞吐量由、2 291Kbps 降低到了 552Kbps,减少了 76%;且其他性能指标也有不同程度的降低.

(2) TDPDA 算法能够很好地避免性能较差的链路对 CMT 传输模型整体性能的影响,吞吐量变化较小(由 3 863.06Kbps 降低为 3 856.72Kbps);且使用 TDPDA 算法的 CMT 传输模型能够有效地防止数据块乱序到达,平均逆序数比标准 CMT 传输模型减少 95%,平均传输时延为 0.081 4s;相对于标准 CMT 传输模型的平均时延 0.340 2s 降低了 76%.

(3) ATLB 算法选择传输时延最小的路径传输数据,在吞吐量、平均时延、接收数据包上与 TDPDA 算法性能一致,逆序数比 TDPDA 算法有所增加,与 CMT 默认的数据分配算法相比有大幅度的改进.

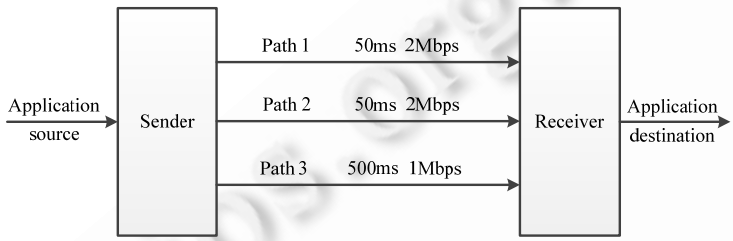

Fig.15 Simulation topology of CMT with high performance difference

图 15 高差异多路径并发传输仿真拓扑

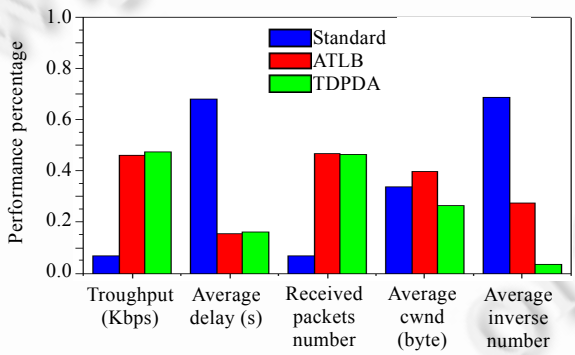

Fig.16 Simulation result of CMT with high performance difference

图 16 高差异多路径并发传输中的仿真结果

图 17 给出了高差异网络环境中使用 3 种数据分配算法的 CMT 传输模型的逆序数对比.由于 TDPDA 算法能够有效地抑制数据块乱序到达,其逆序数稳定且集中在 0~1 的范围内,在相同时间范围内能够发送更多的数据块;标准 CMT 由于受到传输性能较差的路径影响,逆序数相对较大,且发送的数据块数量较少;ATLB 算法的逆序数介于 TDPDA 算法和标准 CMT 数据分配算法之间,相对于 CMT 标准数据分配算法有较大幅度的改进,但比 TDPDA 算法的逆序数有所增加.图 18 给出了高差异网络环境中 3 种数据分配算法所对应的接收缓存对比.可以得出:使用 TDPDA 算法的 CMT 传输模型的接收缓存明显优于标准 CMT 传输模型的接收缓存,且出现接收缓存溢出的次数明显少于标准 CMT 传输模型.ATLB 算法的接收缓存比 CMT 标准数据分配算法有明显的改进,但接收缓存波动性比 TDPDA 算法要大.

图 19 给出了高差异网络环境中使用 3 种数据分配算法的 CMT 传输模型的拥塞窗口对比.可以得出,TDPDA 算法的总体拥塞窗口相对标准 CMT 要小.由于 TDPDA 算法能够减少或者避免在传输性能较差的通信路径上发送数据,性能较差的通信路径的拥塞窗口将保持较小值.同时,TDPDA 算法可以有效地避免拥塞,其拥塞窗口波动较小.ATLB 算法的拥塞窗口比 TDPDA 算法小幅增加,算法波动性较小.CMT 默认的数据分配算法的拥塞窗口最大,且波动性较大.

图 20 给出了使用 3 种数据分配算法的 CMT 传输模型在高差异网络环境中的传输时延对比.可以得出:使用 TDPDA 算法的 CMT 传输模型在经历了连接建立初期、完成了路径传输性能预测后,能够有效地避免在传输性能较差的路径上发送数据,传输时延相对于标准 CMT 有较大程度的提升,保持在 0.1s 内,且传输时延抖动较小;ATLB 算法的引入的传输时延比 CMT 标准算法有较大改进,时延抖动比 TDPDA 算法大.

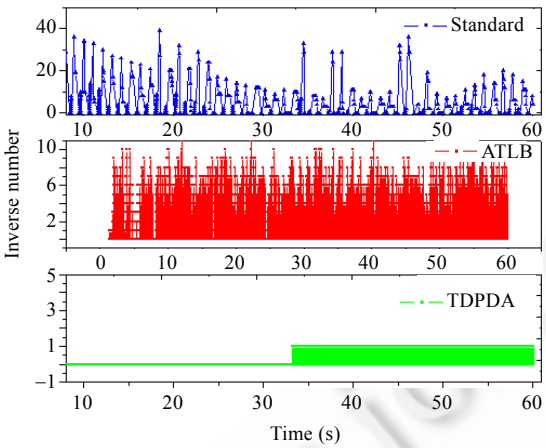

Fig.17 Inverse number of CMT with high performance difference

图 17 高差异多路径并发传输中的逆序数

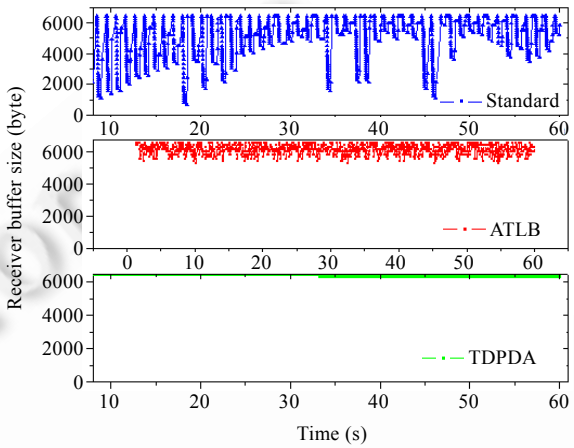

Fig.18 Receiver buffer of CMT with high performance difference

图 18 高差异多路径并发传输中的接收缓存

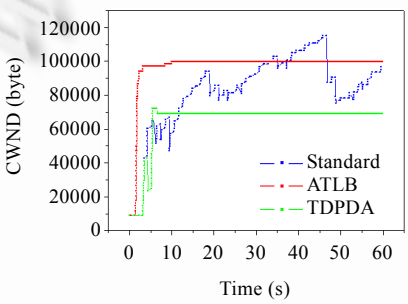

Fig.19 Congestion window of CMT with high performance difference

图 19 高差异多路径并发传输中的拥塞窗口

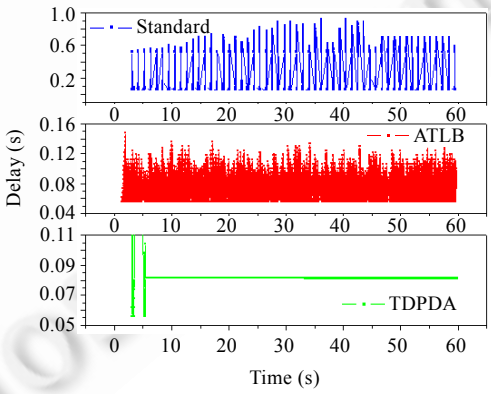

Fig.20 Delay of CMT with high performance difference

图 20 高差异多路径并发传输中的时延

高差异网络环境中使用 3 种数据分配算法的 CMT 传输模型的吞吐量对比如图 21 所示.可以得出:使用 TDPDA 数据分配算法的 CMT 传输模型在路径传输性能预测完成后,能够有效地运用预测结果来协助数据分配,避免或者减少在传输性能较差的通信路径上发送数据,能够在传输性能差异较大的网络环境中获得较好的整体吞吐量;然而,标准 CMT 传输模型由于在传输过程中出现了严重的 HOL 问题,其吞吐量降低明显;ATLB 算法偏向选择路径传输时延较小的路径发送数据,获得的吞吐量和 TDPDA 算法性能比较一致.但 ATLB 算法的吞吐量在仿真后半段稍低于 TDPDA 算法的吞吐量.

图 22 给出了高差异网络环境中使用 3 种数据分配算法的 CMT 传输模型的各路径吞吐量对比.可以得出:标准 CMT 传输模型由于受到传输时延较大的路径 3 影响,路径 1 和路径 2 的吞吐量均有较大程度的降低;

TDPDA 算法能够有效地评估各条通信路径的传输性能,以数据块按序到达为目标,停止在路径 3 中发送数据,避免了性能较差的通信路径对 CMT 传输模型整体性能的影响.可以看出:使用 TDPDA 数据分配算法的 CMT 传输模型的整体吞吐量以及路径 1 和路径 2 的吞吐量相对于标准 CMT 传输模型均有很大的提升;ATLB 算法的路径传输性能与 TDPDA 算法在各条路径上的吞吐量相近,与标准 CMT 传输模型相比有较大的改进.

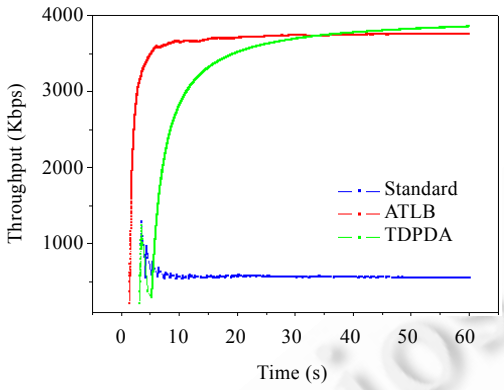

Fig.21 Throughput of CMT with high performance difference

图 21 高差异多路径并发传输中的吞吐量

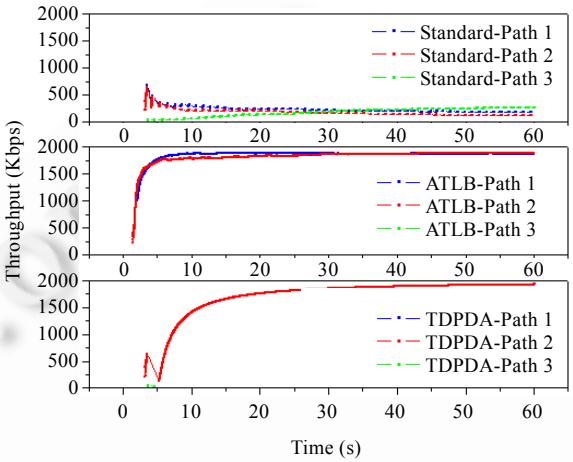

Fig.22 Throughput of paths in high performance difference CMT

图 22 高差异多路径并发传输中的路径吞吐量

5 结束语

由于 CMT 并发传输模型的多条通信路径采用同一个接收缓存,路径之间的传输性能差异将严重影响模型的整体传输性能.本文在分析了 CMT 并发传输模型工作原理的基础上,对多路径并发传输模型在差异化网络环境中的数据分配问题进行优化.通过预测数据块在各条路径上传输所引入的时延,以按序到达为目标对多路径并发传输模型发送回合内和发送回合间的数据分配过程进行优化.

分析和实验结果表明:利用传输时延来预测和优化数据分配过程可以在一定程度上优化 CMT 传输模型的运行性能,能够避免性能较差的通信路径所导致的 HOL 问题,获得较小的通信延迟、较好的传输吞吐量.

致谢 在此,我们向对本文的工作给予支持和建议的《软件学报》编辑部表示感谢,感谢他们认真细致的工作.

References:

[1] Alpcan T, Singh JP, Basar T. Robust rate control for heterogeneous network access in multihomed environments. IEEE Trans. on Mobile Computing, 2009,8(1):41−51. [doi: 10.1109/TMC.2008.85]

[2] Meyer D, Zhang L, Fall K. Report from the IAB workshop on routing and addressing. IETF RFC 4984, 2007. http://www.ietf.org/rfc/rfc4984.txt

[3] Iyengar JR, Amer PD, Stewart R. Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths. IEEE/ACM Trans. on Networking, 2006,14(5):951−964. [doi: 10.1109/TNET.2006.882843]

[4] Stewart R. Stream control transmission protocol. IETF RFC 4960, 2007. http://www.ietf.org/rfc/rfc4960.txt

[5] Cui L, Cui X, Jin JJ, Koh SJ, Lee WJ. Countermeasures to impacts of bandwidth and receiving buffer on CMT schemes. Procedia Engineering, 2011,15:3723−3727. [doi: 10.1016/j.proeng.2011.08.697]

[6] Yang W, Li HW, Wu JP. PAM: Precise receive buffer assignment method in transport protocol for concurrent multipath transfer. In: Proc. of the 2010 Int'l Conf. on Communications and Mobile Computing. IEEE, 2010. 413−417. [doi: 10.1109/CMC.2010.325]

[7] Cao Y, Xu MW. A demand based packet scheduling algorithm for multipath transfer. Ruan Jian Xue Bao/Journal of Software, 2012, 23(7):1924−1934 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4130.htm [doi: 10.3724/SP.J.1001.2012.04130]

[8] Leung KC, Li VO. Generalized load sharing for packet-switching networks I: Theory and packet-based algorithm. IEEE Trans. on Parallel and Distributed Systems, 2006,17(7):694−702. [doi: 10.1109/TPDS.2006.90]

[9] Shi W, Macgregor MH, Gburzynski P. Load balancing for parallel forwarding. IEEE/ACM Trans. on Networking, 2005,13(4): 790−801. [doi: 10.1109/TNET.2005.852881]

[10] Kandula S, Katabi D, Sinha S, Berger A. Dynamic load balancing without packet reordering. ACM SIGCOMM Computer Communication Review, 2007,37(2):51−62. [doi: 10.1145/1232919.1232925]

[11] Shi L, Liu B, Sun CH, Yin ZY, Bhuyan LN, Chao HJ. Load-Balancing multipath switching system with flow slice. IEEE Trans. on Computers, 2012,61(3): 350−365. [doi: 10.1109/TC.2010.279]

[12] Tari Z, Broberg J, Zomaya A, Baldoni R. A least flow-time first load sharing approach for distributed server farm. Journal of Parallel and Distributed Computing, 2005,65(7):832−842. [doi: 10.1016/j.jpdc.2005.02.007]

[13] Hasegawa Y, Yamaguchi I, Hama T, Shimonishi H, Murase T. Improved data distribution for multipath TCP communication. In: Proc. of the 2005 IEEE Global Telecommunications Conf. IEEE, 2005. 271−275. [doi: 10.1109/GLOCOM.2005.1577632]

[14] Xu CQ, Liu TJ, Guan JF, Zhang HK, Muntean GM. CMT-QA: Quality-Aware adaptive concurrent multipath data transfer in heterogeneous wireless networks. IEEE Trans. on Mobile Computing, 2013,12(11):2193−2205. [doi: 10.1109/TMC.2012.189]

[15] Prabhavat S, Nishiyama H, Ansari N, Kato N. Effective delay-controlled load distribution over multipath networks. IEEE Trans. on Parallel and Distributed Systems, 2011,22(10):1730−1741. [doi: 10.1109/TPDS.2011.43]

[16] Cheng RS, Deng DJ, Chao HC, Chen WE. An adaptive bandwidth estimation mechanism for SCTP over wireless networks. In: Proc. of the 5th Int'l Conf. on Future Information Technology. IEEE, 2010. 1−6. [doi: 10.1109/FUTURETECH.2010.5482732]

[17] Wang JY, Liao JX, Li TH. OSIA: Out-of-Order scheduling for in-order arriving in concurrent multi-path transfer. Journal of Network and Computer Applications, 2012,35(2):633−643. [doi: 10.1016/j.jnca.2011.09.004]

[18] Shailendra S, Bhattacharjee R, Bose SK. Optimized flow division modeling for multi-path transport. In: Proc. of the 2010 Annual IEEE India Conf. (INDICON). IEEE, 2010. 1−4. [doi: 10.1109/INDCON.2010.5712713]

[19] Leu FY, Jenq FL, Jiang FC. A path switching scheme for SCTP based on round trip delays. Computers and Mathematics with Applications, 2011,62(9):3504−3523. [doi: 10.1016/j.camwa.2011.08.066]

[20] Piratla NM, Jayasumana AP, Bare AA, Banka T. Reorder buffer-occupancy density and its application for measurement and evaluation of packet reordering. Computer Communications, 2007,30(9):1980−1993. [doi: 10.1016/j.comcom.2007.03.001]

[21] Adhari H, Dreibholz T, Becke M, Rathgeb EP, Tüxen M. Evaluation of concurrent multipath transfer over dissimilar paths. In: Proc. of the 2011 IEEE Workshops of Int'l Conf. on Advanced Information Networking and Applications (WAINA). IEEE, 2011. 708−714. [doi: 10.1109/WAINA.2011.92]

[22] Kurose JF, Ross KW. Computer Networking: A Top-Down Approach. 6th ed., Addison-Wesley, 2013.

[23] Du WF, Wu Z, Lai LQ. Delay-Sensitive data allocation scheme for CMT over diversity paths. Journal of China Institute of Communication, 2013,34(4):149−157 (in Chinese with English abstract).

附中文参考文献:

[7] 曹宇,徐明伟.一种按需分配的多路径传输分组调度算法.软件学报,2012,23(7):1924−1934. http://www.jos.org.cn/1000-9825/4130.htm [doi: 10.3724/SP.J.1001.2012.04130]

[23] 杜文峰,吴真,赖力潜.传输延迟感知的多路径并发差异化路径数据分配算法.通信学报,2013,34(4):149−157.

杜文峰(1977－),男,云南曲靖人,博士,副教授,主要研究领域为无线网络,多点并发传输,冲突解析,绿色网络.

吴真(1987－),男,硕士,主要研究领域为多点并发传输.

赖力潜(1989－),男,硕士,主要研究领域为多路径并发传输,容错通信.