

## 传输延迟感知的多路径并发差异化路径数据分配算法

杜文峰, 吴真, 赖力潜

(深圳大学 计算机与软件学院, 广东 深圳 518060)

**摘要:** 针对多路径并发传输模型的整体性能在部分通信路径性能突变时会急剧下降的多种原因进行分析, 选取对性能影响最大的因素(传输延迟)进行了优化, 提出了一种针对多路径并发传输模型的传输延迟感知的路径数据分配算法。通过结合各条通信路径的实际传输延迟来分配多条通信链路的数据发送量, 以及各条路径上发送的数据分组序号, 减少路径性能恶化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, 所提算法能够相对默认的轮询数据分配算法取得较好的运行性能。

**关键词:** 多路径并发通信; 数据分配; 性能优化

中图分类号: TP393

文献标识码: B

文章编号: 1000-436X(2013)04-0149-09

## Delay-sensitive data allocation scheme for CMT over diversity paths

DU Wen-feng, WU Zhen, LAI Li-qian

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

**Abstract:** The performance of CMT association degrades remarkably when the performances of parts of paths deteriorate. Based on the analysis of different network configurations, a delay sensitive data allocation scheme was proposed to distribute data to different paths over multi-diversity network with reference of their transmission delay, which is a key factor to the whole performance. Meanwhile, the transmission sequence number of each chunk will also be considered. The result of analysis and simulation reveal the performance of our scheme can achieve much better performance than the original round-robin scheme.

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

### 1 引言

流传输控制协议(SCTP, stream control transmission protocol)是由 IETF 在 2000 年提出的一种面向连接的传输协议<sup>[1]</sup>。通过引入多路径、多数据流的方式, SCTP 协议允许终端利用其多个网络接入设备与通信对端组成耦联(association), 实现 IP 网络中的可靠数据传输。

SCTP 协议在建立耦联时将协商和指定特定的数据流为主通信路径。当且仅当主通信路径失效时, SCTP 耦联才会利用其他备用通信路径传输数据, 即多条路径中仅有一条处于工作状态, 而其他路径处于备份状态。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(CMT, concurrent multipath transfer)<sup>[2]</sup>, 利用 SCTP 耦联提供的多个数据流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部分, 即通过轮询(RR, round robin)方式在所有路径上发送数据。然而, 由于各条路径在实际运行过程中的吞吐量、传输延迟、路径出错率等性能参数各不相同, 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。本文正是针对该问题进行分析, 提出

收稿日期: 2012-01-31; 修回日期: 2012-10-06

基金项目: 国家自然科学基金资助项目(61003271); 深圳市基础研究基金资助项目(JC201005280427A)

**Foundation Items:** The National Natural Science Foundation of China (61003271); The Strategy Grant of Shenzhen Science and Technology Department (JC201005280427A)

了一种传输延迟感知的多路径并发差异化路径数据分配(DSDAS, delay-sensitive data allocation scheme for concurrent multipath transmission over diversity paths)算法。该算法根据路径的实际传输性能来分配和设置各条路径的传输数据量,以及在各条路径中传输的数据分组序号,降低到达接收端数据分组之间的间隙,避免传输性能较好的通信路径等待性能较差的通信路径而出现的队头阻塞(HOL, head of line)问题,减少不必要的系统重传。

## 2 相关研究

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连网络,或者经过不同状态的同构网络,在运行过程中的传输性能不一定相同<sup>[3]</sup>。在 CMT 模型中,如果路径之间的传输性能差异较大,到达接收端的数据分组将出现严重的乱序,接收方必须等待接收序列前面的数据分组补齐后才能将数据传给上层应用。根据 SCTP 协议的设计,当接收端出现传输序列号(TSN, transmission sequence numbers)不连续的数据块时,接收方将向发送端发送 SACK 控制块,通知对方当前的分组间隔(gap reports)。当某个已经发送的 TSN 号在间隔报告中连续出现 4 次后,发送方将启动快速重传机制来重新发送该数据分组。可以发现,不必要的快速重传将导致不必要的拥塞控制窗口变化,严重影响了 CMT 模型的数据传输效率,对系统的整体性能影响较大。

为了分析路径传输性能差异对 CMT 模型的影响,本文使用 NS2 对具有 6 条通信路径的 CMT 传输模型进行了仿真。通过对具有 6 条 1Mbit/s 带宽、45ms 延迟的 CMT 传输模型,5 条 1Mbit/s 带宽、45ms 延迟的通信路径,以及 1 条 0.5Mbit/s 带宽、45ms 延迟的 CMT 传输模型和 5 条 1Mbit/s 带宽、45ms 延迟的通信路径,以及 1 条 1Mbit/s 带宽、300ms 延迟的 CMT 传输模型进行仿真,可以看到在相同网络配置情况下,路径传输延迟差异对 CMT 传输模型的影响较大,如图 1 所示。因此,如何分析和利用各条通信路径的传输延迟差异,如何优化 CMT 传输模型的数据发送过程将成为影响整个耦联性能的关键因素。

当前针对 CMT 传输模型的研究主要集中于协议工作原理<sup>[4-6]</sup>、传输策略<sup>[7]</sup>和传输性能<sup>[8]</sup>的讨论和介绍。

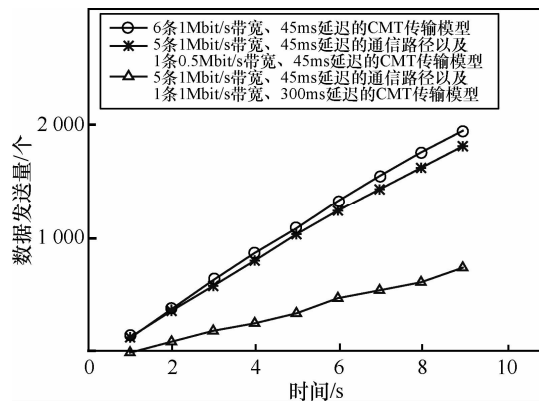


图 1 差异化通信路径对 CMT 传输模型的影响

文献[9]讨论了利用传输延迟作为路径衡量标准的可行性,并介绍了 SCTP 耦联主路径选择和切换方面的研究,其路径性能评估标准与本文相似。文献[10]对 SCTP、TCP、UDP 3 种协议在不同网络配置情况下的运行性能进行了仿真比较;文献[11]对 CMT 传输模型的路径错误检测和重传策略进行修改,改进了 CMT 传输模型在传输过程中的运行性能;文献[8]将 CMT 引入到服务切换过程中,通过在切换双方并发传输数据,保证切换的连续性;文献[12,13]讨论了将 SCTP 耦联的多个数据流进行带宽汇聚的可能性,并对可能出现的问题进行了分析;文献[14]在 Linux 平台上实现了 CMT 传输模型,并比较了 CMT 传输模型和 TCP 传输模型的性能;在文献[15]中对整个 SCTP 耦联的带宽进行预测分析;文献[16]建议为每个 SCTP 耦联的数据流单独进行拥塞控制,确保传输的公平性;文献[17]中对 CMT 传输模型的多种重传策略进行了分析;文献[18]中设计了一种快速重传方式来改进 SCTP 协议在切换过程中的运行性能;文献[19]将负载均衡原理应用于 SCTP 多数据流传输过程中,在多条路径中进行选择性地发送数据;文献[20]中针对 SCTP 耦联的各个数据流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 cwnd 和慢启动阈值(slow-start threshold)来调整各个数据流的发送速度;文献[21]中提出了使用资源池来统计和分析 CMT 各条通信路径的运行性能,并根据 Min-Max 原理对各条路径进行数据分发。

可以发现,目前国内外针对 CMT 传输模型的研究主要在于工作机制,以及失败重传等方面,针对如何在差异化环境中进行多路径并行传输的研究还比较少。在 CMT 传输模型中,传输耦联由多条 QoS 各不相同的通信路径组成,耦联的服务质量

将由传输数据的所有路径的服务质量决定。因此，如何选择通信路径，以及如何协调多个路径的传输数据量成为影响整个 CMT 传输模型服务质量的重要因素，路径的选择与协调将直接决定整个 CMT 传输模型的服务质量。尽管在 SCTP 协议中可以使用心跳包来收集各路径的状态信息，例如路径的工作状态，但是在实际网络环境中，路径性能可能包括可用带宽、链路延迟、分组丢失率、链路类型、链路冗余等，路径的工作状态并不能为路径选择提供较好的参考信息。

根据图 1 所示的仿真结果，网络传输延迟相对于其他网络参数而言对 CMT 传输模型的整体性能影响较大。本文正是在分析了多种网络配置的情况下，针对多条路径的传输延迟性能差异进行分析，提出了结合各条路径的传输延迟来动态调整各条路径的数据发送量以及发送序号，确保整个 CMT 传输模型在多路径差异化网络环境中的传输性能。

### 3 传输延迟感知的多路径并发差异化路径数据分配算法

为了避免不同路径的传输性能差异对 CMT 传输模型整体运行性能造成的影响，本算法为各条路径设置了独立的发送缓冲和接收缓冲，同时根据各条路径的实际传输延迟来协调数据分组在各条路径中的传输。即在进行数据发送之前，根据各条路径的传输性能来分配各个路径的传输数据量，以及在各条路径中的传输数据分组序号。

假设在通信终端 A 和通信终端 B 之间存在一个 SCTP 耦联，使用 CMT 传输模型进行并发通信。SCTP 耦联由  $M$  条独立的通信路径组成，且各条路径的传输性能各不相同，即各路径所提供的带宽、分组丢失率、回路延迟(RTT, round trip time)等性能不一定相同，如图 2 所示。

令 SCTP 耦联的路径  $m$  使用符号  $P_m$  表示， $1 \leq m \leq M$ 。

在进行初次数据发送之前，发送端将在每条路径上发送心跳分组来获取各条路径的回路延迟。接收方将根据路径的回路延迟计算各条路径的传输延迟。

令路径  $P_m$  在当前时刻的回路延迟为  $RTT_m$ ，且在  $\Delta t$  时间范围内，CMT 通信对端各条路径的回路延迟变动均低于一定范围  $\Delta b$ 。将 SCTP 耦联的所有路径按照  $RTT$  值从小到大排序，得到集合  $R=\{R_1, R_2, \dots, R_m, \dots, R_M\}$ ，其中， $R_1$  和  $R_M$  分别对应传输延迟值最小和最大的通信路径。

若当前时刻需要发送的数据量为  $T$ ，发送方将遍历集合  $R$ ，根据各条路径的传输延迟比将待发送的数据分配给各条指定路径进行传输。即按照各条路径的传输延迟大小，将数据按路径传输延迟与最小传输延迟之间的比例逐步分配给各条通信路径。

可以发现，当且仅当以下条件满足时，发送方才为路径  $R_m$  分配发送数据。

$$T > \left\lfloor \frac{R_m}{R_1} \right\rfloor \times MTU_1 + \left\lfloor \frac{R_m}{R_2} \right\rfloor \times MTU_2 + \left\lfloor \frac{R_m}{R_3} \right\rfloor \times MTU_3 + \dots + \left\lfloor \frac{R_m}{R_{m-1}} \right\rfloor \times MTU_{m-1}, 1 < m \leq M \quad (1)$$

发送方在遍历集合  $R$  的过程中依次对各条路径执行式(1)的判断过程，并为相应路径分配传输数据。

假设当前时刻发送方需要使用  $r$  条路径才能完成数据分配，则在各条路径中传输的数据分组传输序列号设置如下。

- 1) 在  $R_2$  传输时间内，发送方将依次为路径  $R_1$  分配  $\lfloor R_2/R_1 \rfloor$  次传输数据单元。
- 2) 在  $R_3$  传输时间内，路径  $R_1$  经过  $\lfloor R_2/R_1 \rfloor$  次分配后，发送方仍然有数据未被发送，则发送方向路径  $R_2$  分配一次传输数据；此时，路径  $R_1$  继续按

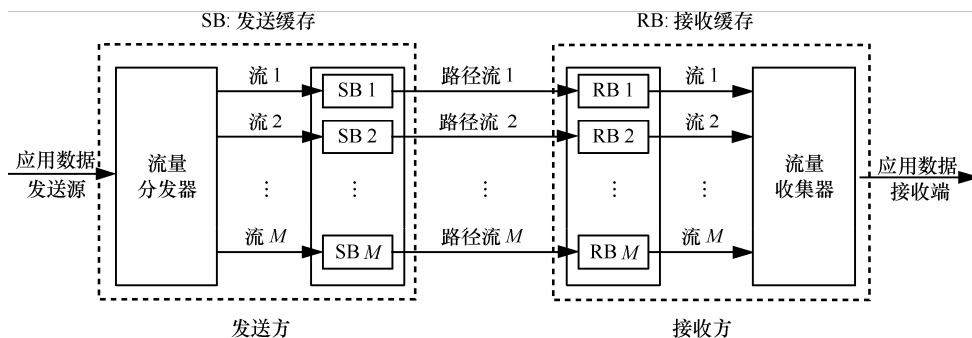


图 2 多路径并发通信模型

序进行数据发送。

3) 在  $R_4$  传输时间内, 路径  $R_1$  经过  $\lfloor R_3/R_1 \rfloor$  次分配, 路径  $R_2$  经过  $\lfloor R_3/R_2 \rfloor$  次分配后, 发送方仍然有数据未被发送, 则发送方将向路径  $R_3$  分配一次传输数据; 此时, 路径  $R_1$  和  $R_2$  继续 1) 和 2) 的传输过程, 按序进行数据发送。

4) 一般地, 路径  $R_1$  经过  $\lfloor R_m/R_1 \rfloor$  次分配, 路径  $R_2$  经过  $\lfloor R_m/R_2 \rfloor$  次分配,  $\dots$ , 路径  $R_{m-1}$  经过  $\lfloor R_m/R_{m-1} \rfloor$  次分配后, 发送方仍然有数据未被发送, 则向路径  $R_m$  分配一次传输数据; 此后, 路径  $R_1, R_2, \dots, R_{m-1}$  继续按照以上步骤进行传输, 按序进行数据发送。

当待发送的数据量超过整个 CMT 模型一轮遍历所能承载的数据量时, 即

$$T > \frac{R_M}{R_1} \times MTU_1 + \frac{R_M}{R_2} \times MTU_2 + \frac{R_M}{R_3} \times MTU_3 + \dots + MTU_M \quad (2)$$

发送方将通过多轮传输将数据分配给 CMT 耦联的相应路径。每一轮的分配过程均按照上述方式进行。

接收方根据收到数据分组的情况向发送方反馈 SACK 信息。发送方结合通信路径的当前 RTT 值与 ACK 信息来获取各条路径的最新工作状态。当某条路径的传输延迟变化超过一定阈值时, 发送方利用各条路径的实时传输延迟对集合  $R$  进行更新。

与此同时, 当 CMT 耦联的各条通信路径性能相近, 或者性能较差的通信路径恢复正常时, 各条路径在传输过程中引入的延迟相差较小。此时, 式 (1) 的判断过程等同于 CMT 耦联中默认的轮询数据发送算法。

### 4 性能分析

在分析算法性能之前, 先做以下假设: 整个 CMT 耦联由  $M$  条独立的通信路径组成, 且每条路径的业务到达满足速率为  $\lambda_i$  的泊松分布; 同时, 各条路径的传输时延均满足负指数分布, 且路径  $P_i$  的传输延迟为  $\mu_i$ 。由于本算法为每条路径设置独立的发送缓存, 令路径  $P_i$  允许的发送缓冲区为  $A_i$ , 即能够容纳  $A_i$  个数据分组。

可以得到, 每条路径的传输过程为  $M/M/1/A_i$  的排队过程, 其业务到达满足速率为  $\lambda_i$  的泊松分布, 业务服务速率满足速率为  $\mu_i$  的负指数分布。

假设在时刻  $t$ , 路径  $P_i$  中已经有了  $k$  个数据分组,  $0 \leq k \leq A_i$ 。令  $\Delta t$  为一个足够小的时间段, 以至于在该时间段内路径  $P_i$  中仅有一个数据分组需要发送, 或者仅能进行一次数据发送操作。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分组为  $\lambda_i \Delta t + o(\Delta t)$ ; 同时, 路径  $P_i$  中被发送完的数据分组为  $k\mu_i \Delta t + O(\Delta t)$ 。

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  $k$  个待发送数据分组的概率  $p_{i,k}$  为

$$p_{i,k} = \begin{cases} \left( \frac{\lambda_i}{\mu_i} \right)^n \times p_{i,0} = \rho_i^n \times p_{i,0}, & n = 0, 1, 2, \dots, A_i \\ 0, & n \geq A_i \end{cases} \quad (3)$$

其中,  $\rho_i = \lambda_i / \mu_i$ , 当  $\rho_i < 1$  时, 有

$$p_{i,0} = \left[ \sum_{n=0}^{A_i} \left( \frac{\lambda_i}{\mu_i} \right)^n \right]^{-1} = \left[ \frac{1 - \left( \frac{\lambda_i}{\mu_i} \right)^{A_i+1}}{1 - \frac{\lambda_i}{\mu_i}} \right]^{-1} = \frac{1 - \rho_i}{1 - \rho_i^{A_i+1}} \quad (4)$$

可以得到, 在路径  $P_i$  中的传输的平均数据分组数量  $L_i$  为

$$\begin{aligned} L_i &= \sum_{k=0}^{A_i} k \times p_{i,k} = \frac{1 - \rho_i}{1 - \rho_i^{A_i+1}} \times \rho_i \times \sum_{n=0}^{A_i} \frac{d}{d\rho_i} (\rho_i^n) \\ &= \frac{1 - \rho_i}{1 - \rho_i^{A_i+1}} \times \rho_i \times \sum_{n=0}^{A_i} \frac{d}{d\rho_i} \left( \frac{1 - \rho_i^{A_i+1}}{1 - \rho_i} \right) \\ &= \frac{\rho_i}{1 - \rho_i} - \frac{(A_i + 1) \times \rho_i^{A_i+1}}{1 - \rho_i^{A_i+1}} \end{aligned} \quad (5)$$

同时, 路径  $P_i$  中的平均等待数据分组数量  $L_{qi}$  为

$$L_{qi} = \sum_{k=0}^{A_i} (k-1) \times p_{i,k} = L - (1 - p_{i,0}) \quad (6)$$

由于路径  $P_i$  的发送缓冲有限, 只能接收  $A_i - 1$  个数据分组。当路径  $P_i$  处于状态  $A_i$  时, 该路径将不接收新的数据分组, 即此路径接收新数据分组的概率为  $1 - p_{i,A_i}$ 。因此, 可以得到单位时间内实际进入路径的数据分组数量的平均值为

$$T_i = \sum_{k=0}^{\infty} \lambda_i \times p_{i,k} = \sum_{k=0}^{A_i-1} \lambda_i \times p_{i,k} = \lambda_i \times (1 - p_{i,A_i}) = \mu_i \times (1 - p_{i,0}) \quad (7)$$

同样, 可求出数据分组的平均逗留时间  $W_i$  和平均等待时间  $W_{qi}$  为

$$W_i = \frac{L}{T_i} = \frac{L}{\lambda_i \times (1 - p_{i,A_i})} \quad (8)$$

$$W_{qi} = \frac{L_{qi}}{T_i} = \frac{L_{qi}}{\lambda_i \times (1 - p_{i,A})} \quad (9)$$

且  $W_i = W_{qi} + 1/\mu_i$ 。

根据 SCTP 协议的拥塞控制原理，发送方的数据到达接收方后，接收方将缓存所有已收到的数据分组，并将低于当前 SACK 对应 TSN 的所有数据分组提交给上层应用，并从接收缓冲区中释放数据空间。当接收方向上层应用提交数据后，接收方将通过 SACK 信息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 耦联的传输性能将由接收缓冲区的释放速度和执行条件决定。

#### 4.1 轮询数据分配

在当前 CMT 传输模型的数据传输过程中，发送方将在多条路径中轮询发送数据，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  $1/M$ 。

为了简化分析过程，本文假设接收方的接收缓存容量为各个路径的一次发送数据量之和，且每条路径在每次传输过程中仅发送 1 个数据分组。由于轮询数据分配算法的数据发送过程是循环执行的，本文将针对数据首次轮询和过程中轮询 2 种情况进行讨论。

##### 1) 首轮数据传输

在首轮数据传输中，所有的通信路径均未发送过数据。假设当前时刻路径  $P_i$  的回路延迟在所有通信路径中处于最大。此时，在接收方的缓冲区中，所有轮询次序在路径  $P_i$  后的路径将受到路径  $P_i$  的传输性能影响而降低性能，即这些路径中传输的数据将会因为路径  $P_i$  的数据未到达而阻塞在接收缓冲区。而所有轮询次序在路径  $P_i$  前的路径中传输的数据分组将由于数据连续而被提交给上层应用。轮询数据分配算法将为这些路径继续分配数据，并将开始新一轮数据传输过程，如图 3 所示。

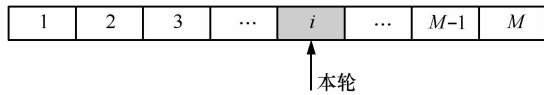


图 3 在首轮传输过程中被阻塞的路径

可以得到，由于路径  $P_i$  传输性能影响而被阻塞的路径数量  $K_i$  为

$$K_i = (M - i) + \sum_{t=i}^M f(t) \quad (10)$$

其中， $f(t)$  为判断所有轮询次序在路径  $P_i$  前的路径是否完成传输的标识函数，即

$$f(t) = 1, \text{ 若 } RTT_t \geq 2 \times RTT_i \quad (11)$$

假设在首轮数据传输中路径  $P_i$  的回路延迟在  $M$  条通信路径中处于最大的概率为  $p_i$ ，可以得到轮询数据分配算法在首轮数据传输过程中可能被阻塞的路径数量  $K$  为

$$K = \sum_{i=1}^M \left( (M - i) + \sum_{t=i}^M f(t) \right) \times p_i \quad (12)$$

##### 2) 非首次进行数据传输

在前一轮数据传输过程中，所有轮询次序在路径  $P_i$  前的路径将由于传输数据连续而被提交给上层应用，提前开始了新一轮数据传输过程。而所有轮询次序在路径  $P_i$  后的路径将由于接收缓冲区中的数据被阻塞而继续等待。当在路径  $P_i$  中传输的数据到达后，所有在接收方缓存中被阻塞的连续数据分组将被提交给上层应用。

假设在本轮传输中，路径  $P_j$  的回路延迟在所有未完成数据传输的路径中处于最大。此时，所有轮询次序在路径  $P_j$  后的路径将由于接收缓冲区中的数据被提交而开始数据传输。同时，所有提前发送的数据有可能已经被接收，或者仍在传输过程中。轮询数据分配算法在非首轮传送过程中的路径阻塞情况示意如图 4 所示。

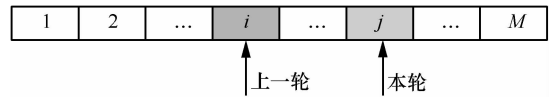


图 4 在非首轮传输过程中被阻塞的路径

可以得到，由于路径  $P_j$  的传输性能影响而被阻塞的路径数量  $K_j$  为

$$K_j = (M - j) + \sum_{t=j}^M g(t) \quad (13)$$

其中， $g(t)$  为判断所有轮询次序在路径  $P_j$  前的路径是否完成传输的标识函数。标识函数  $g(t)$  的取值情况如下。

- ① 路径  $P_t$  在路径  $P_j$  之后，路径  $P_t$  不能提前发送数据。即当  $RTT_t \geq 2 \times RTT_j$ ，且  $t \geq j$  时， $g(t) = 1$ 。
- ② 路径  $P_t$  在路径  $P_j$  之前，路径  $P_t$  在本轮数据传输过程中提前发送数据，且提前发送的数据仍在传输过程中，即当  $RTT_t \geq 3 \times RTT_j - RTT_t$ ， $RTT_t < 2 \times RTT_j$ ，且  $t < j$  时， $g(t) = 1$ 。

③ 路径  $P_i$  在路径  $P_i$  之前, 路径  $P_i$  在本轮数据传输过程中提前发送数据, 且提前发送的数据已经被接收, 即当  $RTT_j \geq RTT_t$ ,  $RTT_i \geq 2 \times RTT_t$ , 且  $t < i$  时,  $g(t)=1$ 。

同样, 假设在当前数据传输过程中路径  $P_j$  的回路延迟在  $M$  条通信路径中处于最大的概率为  $p_j$ , 可以得到轮询数据分配算法在多次数据传输过程中可能被阻塞的路径数量  $K$ 。

$$K = \sum_{j=1}^M \left( (M-j) + \sum_{t=1}^j g(t) \right) \times p_j \quad (14)$$

可以发现, 当某条路径的传输性能变差时, 使用轮询数据分配算法的 CMT 耦联可能出现类似 TCP 协议中的 HOL 问题。即如果在使用轮询数据分配算法的 CMT 耦联中存在部分传输性能较差的路径, 其他传输性能较好的通信路径必须等待性能较差的通信路径完成数据传输后, 即等接收缓存中被拥塞的数据提交后, 才能继续发送数据。同时, 传输性能较差的路径所处的轮询位置越靠前, 整个耦联的传输性能下降越明显。

由于轮询数据分配算法在每一轮传输过程中都必须等待传输往返延迟最大的路径完成数据传输后才能将接收到的数据提交给上层应用, 因此, 在每个  $RTT_{\max}$  时间段内, 使用轮询数据分配算法的 CMT 传输模型能够成功传输的数据量为当前所有被阻塞的通信路径的传输数据量之和。

同时, 可以得到数据分组在整个 CMT 耦联中的平均逗留时间  $W$  为

$$W = \max(W_1, W_2, \dots, W_M) \quad (15)$$

且数据分组在传输过程中的平均等待时间  $W_q$  为

$$W_q = \max(W_{1q}, W_{2q}, \dots, W_{Mq}) \quad (16)$$

#### 4.2 路径传输延迟感知的数据分配

在本算法中, 发送端将充分利用各条路径的传输性能, 允许传输往返延迟较小的通信路径进行多次传输, 让所有通信路径在允许的时间范围内尽可能地传输数据。由于各条路径在传输过程中是相互

独立的, 整个 CMT 耦联的吞吐量并不会因为某一条通信路径的传输性能变坏而急剧降低。在  $RTT_{\max}$  时间范围内, 各条路径的传输次数如图 5 所示。

同时, 定义路径  $R_i$  的回路延迟  $RTT_i$  在分配中出现的次数占总分配次数的比例为路径  $R_i$  在传输过程中获得数据分组的概率  $R_i$ 。假设当前时刻, 发送端在集合  $R$  中遍历到第  $r$  条路径时即可完成数据传输, 则可以得到路径  $R_i$  被选择发送数据的概率为

$$p_i = \frac{\frac{RTT_r}{RTT_i}}{\sum_{j=1}^r \frac{RTT_r}{RTT_j}} = \frac{1}{RTT_i} \quad (17)$$

可以看到, 传输往返延迟较大的路径在数据分配过程中获得数据分组的概率相对较小, 能够降低延迟较高的路径对 CMT 传输模型带来的影响。

根据第 3 节给出的数据分配算法, 以及数据分组传输序号设置原理, 只有当数据量超过式(1)中的阈值时, 发送端才给路径  $R_m$  分配数据。因此, 由于路径  $R_j$  传输性能影响而被阻塞的路径数量  $K_j$  为

$$K_j = \sum_{t=1, t \neq j}^M d(t) \times \frac{RTT_t}{RTT_j} \quad (18)$$

其中,  $d(t)$  为判断所有回路延迟比路径  $R_j$  的回路延迟小的路径是否完成传输的标识函数。标识函数  $d(t)$  的取值情况如下

$$d(t) = \begin{cases} 1, & RTT_j < \left\lceil \frac{RTT_j}{RTT_t} \right\rceil \times RTT_t \\ 0, & \text{其他} \end{cases} \quad (19)$$

因此, 使用本算法的 CMT 耦联在每一个  $RTT_{\max}$  时间内能够传输的数据量  $K$  为

$$K = \sum_{i=1}^M \frac{RTT_{\max}}{RTT_i} \quad (20)$$

此时, 数据分组在整个 CMT 耦联中的平均逗留时间  $W$  为

$$W = \sum_{i=1}^M p_i \times W_i \quad (21)$$

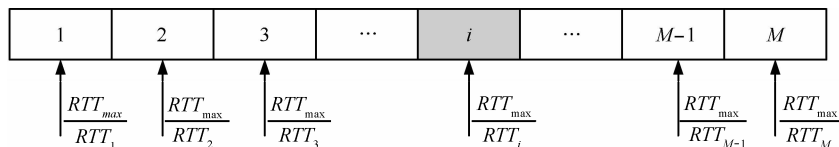


图 5 路径传输次数

且整个数据分组在传输过程中的平均等待时间  $W_q$  为

$$W_q = \sum_{i=1}^M p_i \times W_{iq} \quad (22)$$

与此同时，由于本算法将结合每条路径的传输延迟情况来为各条路径分配数据，其算法复杂度为  $O(kn)$ ，相对于 CMT 耦联默认轮询分配算法的复杂度  $O(n)$  并未有较大提高。

### 5 仿真结果

通过对 NS2 中的 sctp.cc 和 sctp-cmt.cc 等文件进行修改，本文在原有 CMT 轮询数据分配的基础上分别对各种网络性能参数进行了分析，给出了轮询数据分配算法以及本文提出的路径传输延迟感知的数据分配算法在并发差异化通信路径中的传输运行性能。仿真参数设置如图 6 所示。

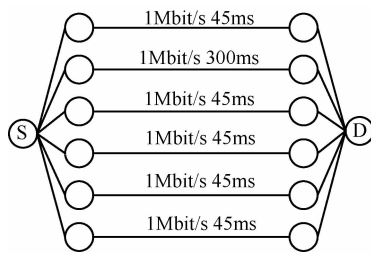


图 6 仿真环境

图 7 统计了 DSDAS 算法与 RR 算法在 10s 内发送数据分组的数量信息。可以看到，在整个仿真过程中本算法发送数据分组的数量都远高于默认的 RR 算法。由于本算法为所有通信路径设置了独立的发送缓存和接收缓存，并且在数据分配过程中减少对性能恶化通信路径所分配的数据量，根据多条路径的整体运行性能来预估计发送数据分组序号，可以减少由于部分通信路径性能恶化所带来的性能影响。RR 算法在传输过程中受到性能较差路径的影响，发送数据分组的数量相对较低。

图 8 记录了 DSDAS 算法与 RR 算法在运行过程中的拥塞窗口 cwnd 变化。由于该算法采用了独立的发送缓存和接收缓存，因此该算法将多条路径的 cwnd 进行了平均。从图中可以看出，由于 CMT 耦联进行首次数据发送时并不能获得各条通信路径的网络状况，在发送初期将根据网络默认配置进行数据发送，即以较大的 cwnd 值发送数据。然而，由于 CMT 耦联中存在着性能较差的通信路径，可

能让数据在传输过程中丢失或者引入延迟较大，因此数据发送方将根据拥塞控制原理来调整 cwnd 的值。DSDAS 算法在一轮数据传输后，获取了多条通信路径的网络状况，并根据各条路径的传输延迟来调整各条路径的数据发送量，能够在较短时间内调整 cwnd 的值，并尽早进入慢启动阶段。相反，默认的 RR 算法将一直等待每一条通信路径完成数据传输后，才能进入下一轮循环，因此其 cwnd 值调整时间滞后较多。从仿真开始 0.81s 后，DSDAS 算法所对应的拥塞窗口增长稳定，且不断递增。相反，RR 算法在仿真过程中采用轮询的方式在多个路径上发送数据，拥塞窗口在整个仿真过程中周期性波动较大。与此同时，RR 算法由于部分通信路径的往返延迟较大，首次 cwnd 的更新周期相对较长，且更新时间也相对靠后。

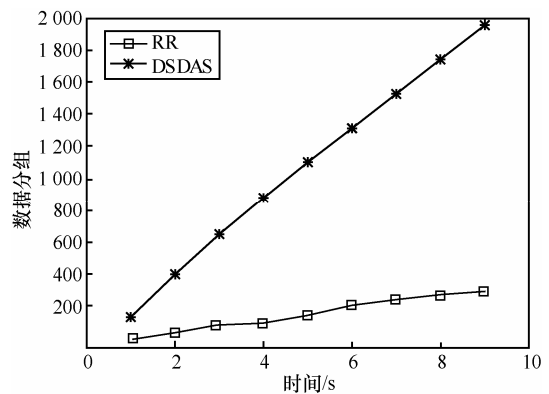


图 7 发送数据量

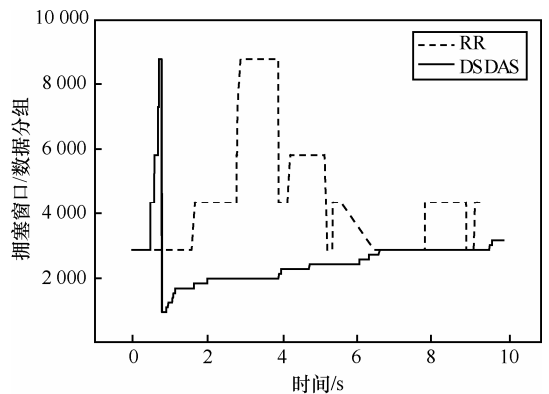


图 8 拥塞窗口

图 9 中记录了 DSDAS 算法和 RR 算法在实时运行过程中的吞吐量。RR 算法由于周期性将数据分配给性能较差的通信路径，网络性能周期性地波动。而 DSDAS 算法为各条路径设置独立的发送、接收缓存，且能够针对各条路径的传输延迟进行优化，整体传输性能相对较优。

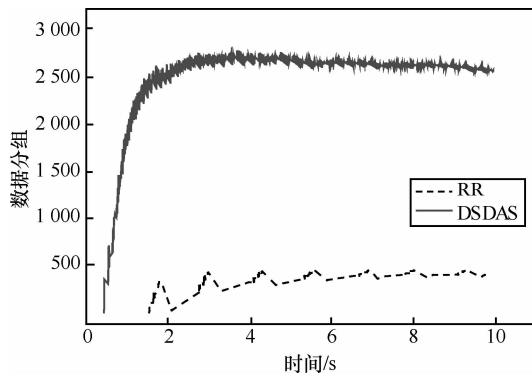


图 9 实时吞吐量

图 10 中统计了 2 种算法从 1.54s 开始传输数据至仿真结束的实时延迟。可以看到 DSDAS 算法在整个仿真过程中引入的传输延迟相对较小，平均传输延迟为 0.101s。RR 算法采用轮询方式，周期性地为多条路径分配数据，受到性能较差路径的影响，RR 算法在传输过程中引入的延迟较大，且周期性波动。

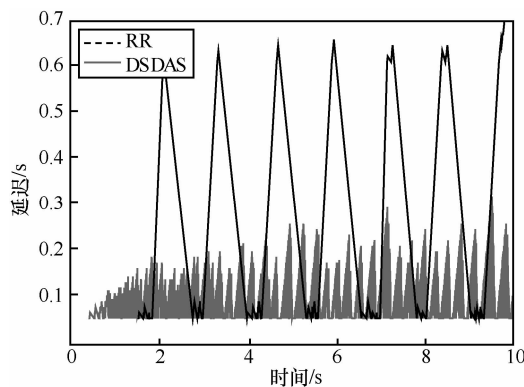


图 10 实时延迟

图 11 统计了 2 种数据分配算法在仿真过程中的实时延迟抖动。从图中可以看出在整个仿真过程中 RR 算法的延迟抖动相对较大，且周期性出现较

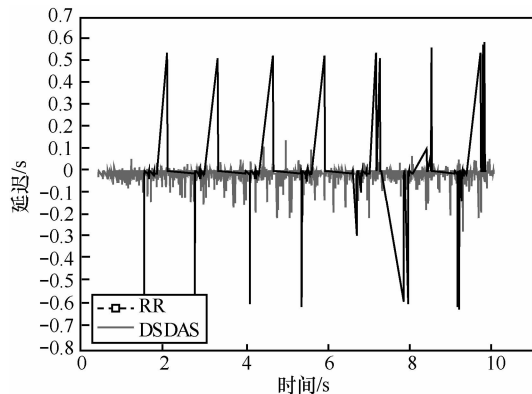


图 11 延迟抖动

大波动。相反，本文提出的 DSDAS 数据分配算法在传输过程中引入的延迟抖动相对比较稳定，该结果与图 10 中的实时延迟结果相一致。

### 6 结束语

由于 CMT 并发通信模型的多条通信路径采用同一个发送缓冲和接收缓冲，性能较差的通信路径将影响其他路径的数据传输。本文在分析了 CMT 并发传输模型的基础上，对原有的轮询数据分配方式进行了改进。通过分析各种网络参数对并发传输模型带来的影响，本文针对最关键的影响参数进行了优化，提出了利用路径传输延迟来优化各条路径的数据分配量，以及发送数据分组序号。

分析和实验结果表明，利用传输延迟来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。在传输过程中结合各条路径的运行性能来设定各个流中数据分组的传输序列号，能够避免性能较差的通信路径所导致的 HOL 问题。同时，为各条路径设置独立的发送缓冲和接收缓冲可以提高路径的并发传输效率。

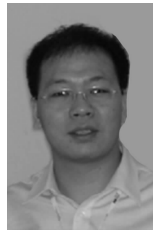
### 参考文献:

- [1] IETF RFC 4960. Stream Control Transmission Protocol[S]. 2007.
- [2] IYENGAR J R, AMER P D, STEWART R. Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths[J]. IEEE/ACM Trans on Networking, 2006, 14(5):951-964.
- [3] 林闯,王元卓,任丰原.新一代网络 QoS 研究[J].计算机学报, 2008, 31(9):1525-1535.
- LIN C, WANG Y Z, REN F Y. Research on QoS in next generation network[J]. Chinese Journal of Computers, 2008, 31(9):1525-1535.
- [4] BUDZISZ L, FERRUS R, CASADEVALL F, et al. On concurrent multipath transfer in SCTP-based handover scenarios[A]. IEEE International Conference on Communications[C]. Germany, 2009. 1-6.
- [5] PEROTTO F, CASETTI C, GALANTE G. SCTP-based transport protocols for concurrent multipath transfer[A]. IEEE Wireless Communications and Networking Conference[C]. Hong Kong, China, 2007. 2969-2974.
- [6] YUAN Y, ZHANG Z D, LI J, et al. Extension of SCTP for concurrent multi-path transfer with parallel subflows[A]. IEEE Wireless Communications and Networking Conference[C]. Sydney, Australia, 2010. 1-6.
- [7] BUDZISZ L, FERRUS R, CASADEVALL F. Design principles and performance evaluation of msctp-cmt for transport-layer based handover[A]. IEEE 69th Vehicular Technology Conference[C]. Barcelona, Spain, 2009. 1-5.
- [8] AYDIN I, SHEN C C. Performance evaluation of concurrent multipath transfer using sctp multihoming in multihop wireless networks[A]. Eighth IEEE International Symposium on Network Computing and Applications[C]. Cambridge, USA, 2009. 234-241.



- [9] LEU F, JENQ F, JIANG F. A path switching scheme for SCTP based on round trip delays[J]. *Computers and Mathematics with Applications*, 2011, 62:3504-3523.
- [10] DAINOTTI A, LORETO S, PESCAP A, *et al.* SCTP performance evaluation over heterogeneous networks[J]. *Concurrency and Computation: Practice and Experience*, 2007, 19:1207-1218.
- [11] NATARAJAN P, EKIZ N, AMER P D, *et al.* Concurrent multipath transfer during path failure[J]. *Computer Communication*, 2009, 32: 1577-1587.
- [12] ARGYRIOU A, MADISETTI V. Bandwidth aggregation with SCTP[A]. *IEEE Global Telecommunications Conference*[C]. San Francisco, USA, 2003. 3716-3721.
- [13] HUANG C, ZENG Z. SCTP-based bandwidth aggregation across heterogeneous networks[A]. *Pacific-Asia Workshop on Computational Intelligence and Industrial Application*[C]. Wuhan, China, 2008. 650-654.
- [14] WANG Y, RHEE I, HA S. Augment SCTP multi-streaming with plug-gable scheduling[A]. *IEEE Conference on Computer Communications Workshops*[C]. Shanghai, China, 2011. 810-815.
- [15] CHENG R, LAI C, HUANG Y, *et al.* Multi-stream bandwidth estimation for SCTP in high-speed networks[A]. *Fourth International Conference on Communications and Networking in China*[C]. Xi'an, China, 2009. 1-4.
- [16] SEO D, KIM H, KIM J G. Analysis of dynamic congestion control of SCTP[A]. *International Conference on Convergence and Hybrid Information Technology*[C]. Daejeon, Korea, 2008. 687-693.
- [17] IYENGAR J R, AMER P D, STEWART R. Retransmission policies for concurrent multipath transfer using SCTP multihoming[A]. *Proc 12th IEEE International Conference on Networks*[C]. Berlin, Germany, 2004. 713-719.
- [18] MA L, YU F R, LEUNG V C M. Performance improvements of mobile SCTP in integrated heterogeneous wireless networks[J]. *IEEE Transactions on Wireless Communications*, 2007, 6(10):3567-3577.
- [19] TSAI C, HSU S. Study of SCTP over mixed-traffic communication networks[A]. *International Symposium on Communications and Information Technologies*[C]. Bangkok, Thailand, 2006. 222-226.
- [20] CHENG R, DENG D, CHAO H, *et al.* An adaptive bandwidth estimation mechanism for SCTP over wireless networks[A]. *5th International Conference on Future Information Technology*[C]. Busan, Korea, 2010. 1-6.
- [21] SHAILENDRA S, BHATTACHARJEE R, BOSE S K. An implementation of min-max optimization for multipath SCTP through bandwidth estimation based resource pooling technique[J]. *International Journal of Electronics and Communications*, 2013, 67(13):246-249.

#### 作者简介:



杜文峰(1977-), 男, 云南曲靖人, 博士, 深圳大学副教授、硕士生导师, 主要研究方向为无线网络、带宽分配、多路径并发通信等。



吴真(1987-), 男, 江西上饶人, 深圳大学硕士生, 主要研究方向为 SCTP 协议、多路径并发通信等。



赖力潜(1989-), 男, 广东梅州人, 深圳大学硕士生, 主要研究方向为多路径并发通信、容错通信。