

多路并行传输中数据调度算法的优化

余东平*, 张剑峰, 王 聪, 李 宁

(解放军理工大学 通信工程学院, 南京 210007)

(*通信作者电子邮箱 yudongpingxm@126.com)

摘 要: 针对异构无线网络环境中, 基于流控制传输协议(SCTP)的多路并行传输协议(CMT-SCTP)存在接收缓存阻塞和路径负载失衡等问题, 提出一种改进的轮询数据调度算法。该算法根据每条路径上的发送队列信息和拥塞状况对网络状况进行估计, 并按照各路径上的网络状况分配相应的传输任务量, 缩短数据包在接收端缓冲区的平均排队时延, 减少接收端乱序数据包的数量。仿真结果表明, 改进的轮询数据调度算法能有效提升 CMT-SCTP 在异构无线网络环境中的传输效率, 有效缓解接收缓存的阻塞, 且对不同的网络场景具有很好的适应性。

关键词: 多路并行传输; 数据调度; 负载均衡; 流控制传输协议; 往返时延

中图分类号: TN915.04 **文献标志码:** A

Optimization of data scheduling algorithm in concurrent multipath data transfer

YU Dongping*, ZHANG Jianfeng, WANG Cong, LI Ning

(College of Communication Engineering, PLA University of Science and Technology, Nanjing Jiangsu 210007, China)

Abstract: To solve the problem of receiver buffer blocking and load unbalance of Concurrent Multipath data Transfer using Stream Control Transmission Protocol (CMT-SCTP) in heterogeneous network environments, an improved round-robin data scheduling algorithm was proposed. The network condition of each path was estimated by the proposed algorithm according to the sender queue information and the congestion status of links. Then the proposed data scheduling algorithm distributed the transmission task to each path based on its network condition, curtailed the queuing time of data chunks in sender buffer and reduced the number of out-of-order data chunks in receiver buffer. Simulation results show that the improved round-robin data scheduling algorithm can effectively enhance the transmission efficiency of CMT-SCTP in a heterogeneous wireless network environment and mitigate the receiver buffer blocking problem. It can also adapt to different network conditions.

Key words: Concurrent Multipath data Transfer (CMT); data scheduling; load balance; Stream Control Transmission Protocol (SCTP); Round-Trip Time (RTT)

0 引言

随着无线通信技术的快速发展, 多种无线网络, 如无线局域网(Wireless Local Area Network, WLAN)、通用移动通信系统(Universal Mobile Telecommunications System, UMTS)等日益成熟, 为用户提供了越来越快速和便捷的接入方式。终端设备也为多种接入方式配备了相应的网络接口, 用户能自由选择终端设备上某个可用的网络接口进行数据传输。传输控制协议(Transmission Control Protocol, TCP)和用户数据报协议(User Datagram Protocol, UDP)这类传统的传输协议, 在数据传输时只能绑定一个网络接口, 数据传输速率很大程度上受到接入网网络状况的影响。如果接入网出现故障, 则用户上网可能会随之断开, 对用户体验造成严重影响。多路并行传输(Concurrent Multipath data Transfer, CMT)^[1-3]是指用户利用多穴终端的多个网络接口同时接入到网络中, 在通信双方之间建立多条传输路径并行传输数据。采用多路并行传输

可充分利用多种网络资源, 避免单点效应, 增强网络可靠性, 还能提高端到端的数据传输速率和网络带宽资源利用率^[4]。

实现多路并行传输, 目前研究较多的是基于流控制传输协议(Stream Control Transmission Protocol, SCTP)^[5]的多路并行传输协议(Concurrent Multipath data Transfer using Stream Control Transmission Protocol, CMT-SCTP)。SCTP的出现最初是为了解决公共电话交换网(Public Switched Telephone Network, PSTN)的信令消息在IP(Internet Protocol)网上的可靠传输问题, SCTP的多穴特性使得传输端点能绑定多个IP地址并建立多条端到端的传输路径。在建立关联后, SCTP利用主路径传输数据, 其他路径则作为备用, 当主路径失效时, 自动切换到备用路径进行数据传输。SCTP能有效提高信令消息在IP网上的传输可靠性, 但SCTP本身并不支持多路并行。CMT-SCTP首先由特拉华州大学的Iyengar等^[1]提出, 通过对SCTP的合理扩展实现了多路并行传输, 并解决了由于

收稿日期: 2013-11-08; 修回日期: 2013-12-19。

基金项目: 国家自然科学基金资助项目(61371123, 6130116); 江苏基金重点专项(BK2011002); 江苏青年基金资助项目(BK2012055)。

作者简介: 余东平(1989-), 男, 湖南澧县人, 硕士研究生, 主要研究方向: 通信信号系统; 张剑峰(1964-), 男, 江苏丰县人, 教授, 博士, 主要研究方向: 通信信号系统; 王聪(1975-), 男, 安徽池州人, 副教授, 博士, 主要研究方向: 通信信号系统; 李宁(1967-), 上海人, 副教授, 博士, 主要研究方向: 认知无线电、信号与信息处理。

接收端乱序引起的不必要的重传问题、拥塞窗口增长问题以及 ACK(Acknowledgement) 流量问题。

在 CMT-SCTP 中,传输层为每个数据块都分配一个传输顺序号(Transmission Sequence Number, TSN),用来保证数据的可靠性。数据调度算法负责将待传输数据块调度到各条路径的发送队列中。由于每条路径的可用带宽、队列长度等参数各不相同且时刻变化,因此各条路径上数据块的传输时延也各不相同。这就很难保证从发送端发出的数据块能按照 TSN 的顺序到达接收端,产生大量的乱序数据块。接收端收到乱序的数据块后不能立即提交给应用层,需要存储在接收缓存中。如果各条路径上的数据块到达接收端的时延相差较大,则乱序数据块滞留在接收缓存中的时间会相应增加,使得有限的接收缓存空间被大量的乱序数据块阻塞^[6]。面向可靠传输的传输层协议,其数据发送速率主要与拥塞窗口以及接收缓存空闲区的大小有关。第二个参数反映了接收端的处理能力,当接收缓存被阻塞时,不仅会限制网络状况较差路径上的数据发送速率,也会降低网络状况较好路径上的数据发送速率,对整体的传输性能造成影响。

目前,该问题已得到了广泛的关注和研究,文献[7]中提出通过合理地管理接收端和发送端缓冲区的方式减少乱序。这种方式能有效地减少每条路径上未被确认数据的总量,不足是没有考虑各路径的带宽、时延上的差异,不能充分地利用网络状况较好的路径。文献[8]中提出了一种新的能够感知链路质量的数据调度机制——CMT-QA(Quality-aware Adaptive Concurrent Multipath data Transfer),发送端周期性估计各路径的网络状况,并估算出当前数据包通过各条链路到达接收端所需的时间,然后合理地安排每个待传输数据包的传输路径和发送时间,以保证所有数据包正序到达接收端。本文设计了一种改进的轮询(Round-Robin)^[9]数据调度算法,能根据各条路径的传输能力动态调整各自的传输任务量,让带宽较大的路径承担较多的任务,而带宽较小的路径承担相对较少的任务。通过合理地分配传输任务量,使得各条路径上数据块的平均时延相等,从而减少接收端乱序数据块的数量。

1 轮询的数据调度算法

CMT-SCTP 的每条路径都有独立的拥塞控制机制、独立的发送队列和拥塞窗口(Congestion Window, $cwnd$)。当应用层有数据要传输时,发送端轮询每条路径的拥塞窗口 $cwnd$,若路径 i 的拥塞窗口 $cwnd_i$ 与此路径上的未被确认数据总量 $outstanding_i$ 之差大于当前数据块的大小,则将此数据块加入到路径 i 的发送队列中,其中 $i \in \{1, 2, \dots, n\}$ 且 n 为路径总数。如果当前路径不满足上述条件,则询问下一条,直到找到满足条件的路径为止。接收窗口(Receiver Window, $rwnd$)代表接收端缓冲区空闲区的大小,所有路径上能发送数据的总量不得超过 $rwnd$ 与未被确认数据总量之差。式(1)和式(2)为各路径上的数据发送量 $senddata_i$ 需要满足的条件:

$$senddata_i \leq cwnd_i - outstanding_i \quad (1)$$

$$\sum_{j=1}^n senddata_j \leq rwnd - \sum_{j=1}^n outstanding_j \quad (2)$$

在轮询的数据调度算法中,当发送端找到一条满足发送条件的路径时,就尽其所能地在此路径上发送数据包,直到此路径上的相关参数不满足式(1)和式(2),且不考虑各条路径发送能力之间的差别。在异构的网络环境中,各条路径的带宽、时延都相差较大,这种轮询的数据调度算法会导致接收端产生严重的乱序^[10-12],降低整体传输效率。带宽非对称的传输场景如图1所示,发送端利用两条独立的路径与接收端通信,已知路径1与路径2的带宽之比为1:2,发送一个数据包分别需要2ms和1ms,各路径上的传播时延忽略不计。具体的传输过程如表1所示。在总共16ms的传输过程中,由于接收缓存被阻塞,导致路径2在总共5ms的时间内处于空闲状态。从时间上看,带宽较小的路径1的利用率为100%,而带宽较大的路径2的利用率为68.7%,整体带宽利用率仅为79.2%,带宽利用率较低。

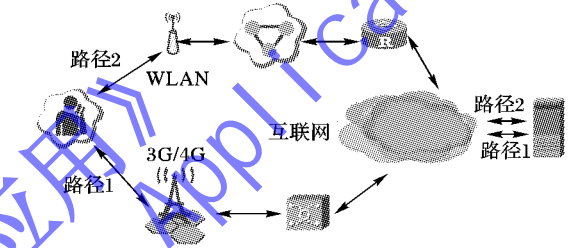


图1 多路并行传输场景

表1 采用轮询算法的数据调度过程

时间	路径1			路径2			$rwnd$	接收缓存中乱序的数据
	$cwnd$	未被确认 num	TSN	$cwnd$	未被确认 num	TSN		
0	5	0	空闲	5	0	空闲	10	无
1	5	5	1~5	5	5	6~10	10	无
2	5	5	1~5	5	4	7~10	9	6
3	5	4	2~5	5	3	8~10	8	6~7
4	5	4	2~5	5	2	9~10	7	6~8
5	5	3	3~5	5	1	10	6	6~9
6	5	3	3~5	10	2	11~12	5	6~10
7	5	2	4~5	10	1	12	4	6~11
8	5	2	4~5	10	1	13	3	6~12
9	5	1	5	10	1	14	2	6~13
10	5	1	5	10	0	空闲	1	6~14
11	10	10	15~24	10	0	空闲	10	无
12	10	10	15~24	10	0	空闲	10	无
13	10	9	16~24	10	1	25	10	无
14	10	9	16~24	10	0	空闲	9	25
15	10	8	17~24	10	1	26	9	25
16	10	8	17~24	10	0	空闲	8	25~26

2 改进的轮询数据调度算法

在多条端到端的传输路径中,往返时延(Round-Trip Time, RTT)越小的路径网络状况越好。如果某条路径上的RTT增长过快,则该路径很可能发生了拥塞,需要相应地减小发送队列的长度。在改进的轮询数据调度算法中,发送端通过测量数据块的RTT估计路径 i 上的网络状况,用 Q_i 表示,定义如式(3)所示。式(4)表示两条不同路径网络状况的相互关系,其中, $i \in \{1, 2, \dots, n\}$ 且 n 为路径总数。

$$Q_i = 1/RTT_i \quad (3)$$

$$\frac{Q_i}{Q_j} = \frac{1/RTT_i}{1/RTT_j} = \frac{RTT_j}{RTT_i} \quad (4)$$

在数据调度过程中,路径*i*上能发送数据的总量 $senddata_i$ 需满足式(5),其中, $Q_{\max} = \max\{Q_1, Q_2, \dots, Q_n\}$ 。在式(5)中, Q_i/Q_{\max} 代表当前路径与网络状况最好路径的路径质量之比。通过这个比值,在一定程度上限制网络状况较差路径上的数据发送量,缩小各路径上 RTT 之间的差距,使得不同路径上的数据包到达接收端的平均时延相等,减少接收缓存中被阻塞数据包的数量。各路径之间 RTT 的差距越大,式(5)对数据发送量的限制作用就越大。为了更有效地均衡各条路径上的负载,在改进的轮询数据调度算法中路径*i*上能发送数据的总量 $senddata_i$ 需满足式(6),其中, $cwnd_{\max} = \max\{cwnd_1, cwnd_2, \dots, cwnd_n\}$ 。

$$senddata_i \leq [\alpha \times (Q_i/Q_{\max}) \times cwnd_{\max} + (1 - \alpha) \times cwnd_i - outstanding_i] \quad (5)$$

$$senddata_i \leq \left[\beta \times cwnd_i / \sum_{j=1}^n cwnd_j + (1 - \beta) \times Q_i / \sum_{j=1}^n Q_j \right] \times \left(rwnd - \sum_{j=1}^n outstanding_j \right) \quad (6)$$

式(6)根据当前 $rwnd$,对各路径上所能发送的数据量进行分配,其中 $rwnd - \sum_{j=1}^n outstanding_j$ 为所有路径上能发送数据的总量。式(6)在分配传输任务时,综合考虑了当前路径上的拥塞窗口和路径质量在所有路径中所占的比例。在式(5)和式(6)中, α, β 为权重因子,通过对多次仿真的结果进行分析,发现当 $\alpha = 0.73, \beta = 0.8$ 时传输效率最高。

改进的轮询机制继承了轮询数据调度算法的优点,在原有协议的基础上无需对接收端和包格式进行改动。表2对改进后的数据传输过程进行了分析,传输场景与图1中相同。分析表明,在总共 16 ms 的时间内接收端总共收到并确认了 21 个数据包。与改进前相比,整体的传输效率提高了,且接收端在传输过程中没有被阻塞,所有路径的利用率达到 100%。改进的轮询数据调度算法的设计架构如图2所示,数据调度模块根据反馈的队列信息和路径信息将应用层的数据流分配到各路径的发送队列。

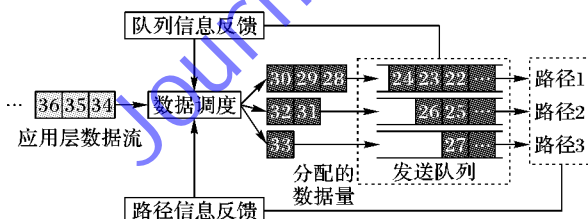


图2 数据调度架构

改进的轮询数据调度算法描述如下:

```
while(应用层有数据需要发送)
for(对于每条路径  $i \in \{1, 2, \dots, n\}$ ,  $n$  为路径总数)
if(路径i处于非活动状态)
将此路径发送队列中的数据复制至待调度
队列中,并清空此路径的发送队列;
continue;
end if
```

if ($\exists senddata_i > 0$, 且满足式(4) 和式(5))

取 $senddata_i$ 的最大值 $maxsend_i$;

取 $m = 0, amount_i = 0$;

while($amount_i \leq maxsend_i$)

$m = m + 1$;

将待调度队列中的一个数据块分配给路

径*i*, 此数据块大小为 $chunk_m$;

$amount_i = amount_i + chunk_m$;

end while

end if

end for

end while

表2 采用改进轮询算法的数据调度过程

时间	路径1			路径2			rwnd	接收缓存中 乱序的数据
	cwnd	未被确认 num	TSN	cwnd	未被确认 num	TSN		
0	5	0	空闲	5	0	空闲	10	无
1	5	3	1~3	5	3	4~6	10	无
2	5	3	1~3	5	2	5~6	9	4
3	5	2	2~3	5	1	6	8	4~5
4	5	2	2~3	5	2	7~8	7	4~6
5	5	1	3	5	1	8	6	4~7
6	5	1	3	5	2	9~10	5	4~8
7	5	3	11~13	5	1	10	10	无
8	5	3	11~13	5	3	14~16	10	无
9	5	2	12~13	5	2	15~16	9	14
10	5	2	12~13	5	1	16	8	14~15
11	5	1	13	5	3	17~19	7	14~16
12	5	1	13	5	2	18~19	6	14~17
13	5	3	20~22	5	1	19	10	无
14	5	3	20~22	5	3	23~25	10	无
15	5	2	21~22	5	2	24~25	9	23
16	5	2	21~22	5	1	25	8	23~24

3 仿真分析

使用 NS-2.34^[13] 网络仿真工具作为本文的仿真工具。网络拓扑如图3所示,发送端和接收端分别具有三个网络接口,路径1、路径2和路径3为三条独立的端到端双向路径。各条路径的可用带宽是非对称的,设定路径1的带宽为 0.2 Mbps,路径2的带宽为 0.4 Mbps,路径3的带宽为 0.6 Mbps,发送端应用层的数据发送速率恒定为 1.2 Mbps。

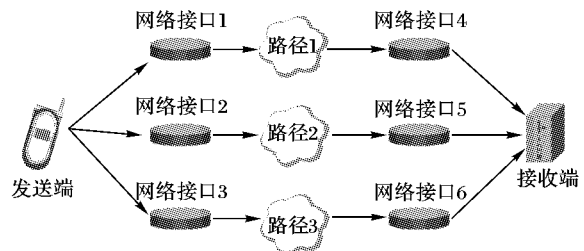


图3 仿真结构

3.1 往返时延的仿真分析

改进后的轮询数据调度算法根据各路径质量之间的关系,限制了路径质量较差路径上的发送队列长度,使其负载减小,缩小各条路径上往返时延 RTT 之间的差距。因此,为了验证改进算法中式(4)的作用,首先对改进前后各路径的

RTT 进行仿真分析。得到结果如图 4~5 所示。其中,图 4 为 CMT-SCTP 在利用改进前的轮询算法进行多路并行传输时,各路径上的 RTT 随时间变化的曲线。当达到稳定状态后,三条路径的 RTT 相差较大,说明三条路径上数据包到达接收端的总时延相差较大,易导致数据包乱序到达接收端。图 5 为 CMT-SCTP 在利用改进后的轮询数据调度算法进行多路并行传输时,各路径上的 RTT 随时间变化的曲线。三条路径上的 RTT 都处于 0.38 s 和 0.45 s 之间,各路径的 RTT 值相差较小,通过不同路径传输的数据包到达接收端的时延差较小。图 6 为数据调度算法改进前后,各路径上的接收窗口 $rwnd$ 随时间变化的曲线。在仿真中接收缓存的大小为 64 KB。从图 6 可以看出,CMT-SCTP 在利用改进后的轮询算法进行传输时 $rwnd$ 值高于改进前的 $rwnd$ 值。这说明由于各路径上的往返时延 RTT 的差距缩小,使得乱序数据包在接收缓存中的平均等待时间相应缩短,接收缓存中滞留的数据块能更快让出缓存空间。

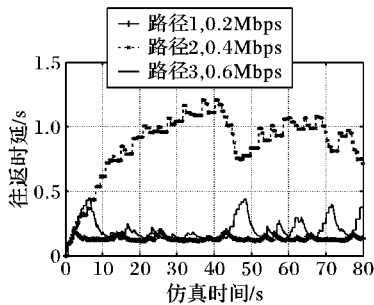


图4 改进前各路径的往返时延

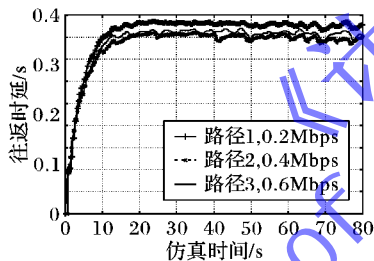


图5 改进后各路径的往返时延

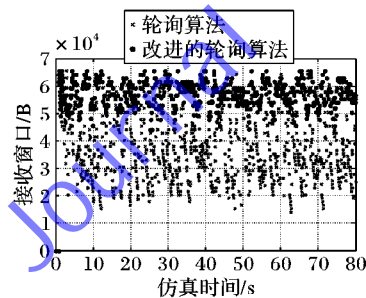


图6 接收缓存空闲区的大小

3.2 数据吞吐量的仿真分析

在改进的轮询数据调度算法中,为了让各条路径上的负载与其网络状况相适应,可根据式(6)对各路径上所能发送数据的总量进行分配。即让网络状况较好的路径承担较多的传输任务,而网络状况较差的路径承担较少的传输任务,最终使得多条路径上的带宽利用率相同。为了验证改进算法中式(6)的作用,对各路径上的数据吞吐量进行仿真分析。图 7 为 CMT-SCTP 在利用改进前的轮询算法进行多路并行传输

时,三条路径上数据吞吐量的变化曲线。在初始时刻,三条路径的数据吞吐量迅速增长,路径 2 和路径 3 上的吞吐量增长一段时间后,开始急剧下降。通过分析可知,此时接收缓存中阻塞了较多的乱序数据包,接收缓存空闲区变小,根据式(2),使得路径 2 和路径 3 上所能发送的数据量变小。图 8 为 CMT-SCTP 在利用改进后的轮询算法进行数据传输时,三条路径的数据吞吐量随时间的变化曲线。当达到稳定状态后,三条路径的数据吞吐量分别与各路径的可用带宽相等。三条路径上的带宽利用率都达到最大。图 9 为 CMT-SCTP 在利用改进前和改进后的轮询算法进行数据传输时,三条路径上的数据吞吐量之和随时间变化的曲线。在初始时刻,吞吐量迅速增长。此时的接收端缓冲空闲区较大,两条路径能尽其所能地发送数据。当达到稳定状态时,使用改进后的轮询算法能最大化利用三条路径的带宽,数据吞吐量达到 1.2 Mbps。而使用改进前的轮询算法的数据吞吐量接近 0.6 Mbps,显然,其带宽利用率较低。

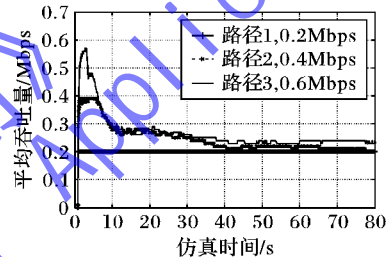


图7 改进前各路径上的平均吞吐量

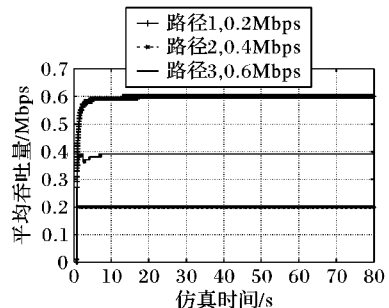


图8 改进后各路径上的平均吞吐量

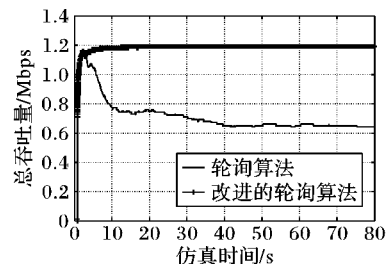


图9 改进前后总吞吐量对比

3.3 异构网络环境下传输效率的仿真分析

在异构无线网络环境中,各路径的可用带宽和时延经常变化,路径上可用带宽的大小易受到接入网用户数量、无线信号强度等因素的影响。因此,一个好的传输层协议要能适应各种不同的网络环境,在可用带宽不断变化的情况下都能够获得较高的传输效率。下面就以带宽变化为例,分析轮询数据调度算法改进前后 CMT-SCTP 的传输效率。表 3 记录了仿真数据,其中路径 1 和路径 2 的总带宽为 1.2 Mbps,表中数据是在带宽之差不同的条件下三种不同的传输方式完成传输

11.4 Mb 的数据所需的时间;第一种方式应用改进前的轮询数据调度算法进行多路并行传输;第二种方式采用改进后的轮询数据调度算法进行多路并行传输;第三种方式采用标准的 SCTP 在路径 2 上进行单路传输。图 10 为三种传输方式完成数据传输所需时间随着路径 2 与路径 1 的带宽之差逐渐增大的变化曲线。当带宽之差大于 4.4 Mbps 时, CMT-SCTP 采用改进前的轮询数据调度算法进行多路并行传输的效率比 SCTP 的单路传输效率更低。从图 10 中可知,采用改进后的轮询算法进行多路并行传输的效率始终高于单路传输,并且随着各路径带宽之差的变化,传输效率基本能保持稳定,这说明改进的算法对带宽的变化具有良好的适应性。

表 3 带宽非对称时的传输时间

带宽/Mbps		传输时间/s				
路径 1	路径 2	带宽之差	带宽之比	轮询算法	改进的轮询算法	路径 2 单路传输
0.0	1.2	1.2	0.00	9.69	9.69	9.60
0.1	1.1	1.0	0.09	33.62	9.95	10.38
0.2	1.0	0.8	0.20	21.53	9.94	11.30
0.3	0.9	0.6	0.33	16.22	9.91	12.44
0.4	0.8	0.4	0.50	12.66	9.63	13.87
0.5	0.7	0.2	0.71	11.15	9.64	15.71
0.6	0.6	0.0	1.00	9.73	9.78	18.17

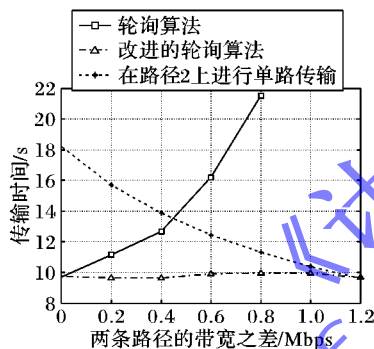


图 10 三种传输方式的传输效率对比

4 结语

随着无线通信技术的快速发展以及硬件成本的降低,终端设备具备了多路并行传输的基本条件。CMT-SCTP 是实现多路并行传输的代表性方案。但在异构无线网络环境中, CMT-SCTP 存在一些亟须解决的问题,如接收缓存阻塞以及路径负载分配不均等。这些问题可能导致多路并行传输中带宽利用率的不均衡,进而影响整体的传输效率。

本文详细分析并找出造成上述问题的原因,对轮询的数据调度算法进行改进,提出了一种优化的基于轮询机制的数据调度算法。该方法一方面能限制各条路径上往返时延 RTT 的差异;另一方面对各条路径所能发送的数据总量进行合理分配。仿真结果显示,改进后的轮询数据调度算法能有效地分配各条路径数据发送量的比例,使得各路径上的 RTT 值更加接近,并能在一定程度上缓解接收缓存阻塞问题。改进后的轮询调度算法,在不同的带宽比例下,都能取得很高的传输效率。不足是不能完全消除接收端数据包的乱序问题,并且,在数据传输初始阶段数据吞吐量增长较慢,所以还需要对此

进行更加深入的研究和讨论。

参考文献:

- [1] IYENGAR J R, AMER P D, STEWART R. Concurrent multi-path transfer using SCTP multihoming over independent end-to-end paths [J]. IEEE/ACM Transactions on Networking, 2006, 14(5): 951 - 964.
- [2] ZHUANG W H, MOHAMMADIZADEH N, SHEN X M. Multipath transmission for wireless Internet access-from an end-to-end transport layer perspective [J]. Journal of Internet Technology, 2012, 13(1): 1 - 17.
- [3] BECKE M, DREIBHOLZ T, ADHARI H, et al. A future Internet architecture supporting multipath communication networks [C]// Proceedings of the 2012 IEEE Network Operations and Management Symposium. Piscataway: IEEE Press, 2012: 639 - 642.
- [4] ALPCAN T, SINGH J P, BASAR T. Robust rate control for heterogeneous network access in multihomed environments [J]. IEEE Transactions on Mobile Computing, 2009, 8(1): 41 - 51.
- [5] STEWART R R, XIE Q, MORNEAULT K, et al. Stream Control Transmission Protocol (SCTP) [S/OL]. [2013-05-06]. <http://tools.ietf.org/html/rfc4960>.
- [6] SONG F, WANG B, ZHANG H, et al. Research on receive buffer blocking in CMT [J]. Acta Electronica Sinica, 2010, 38(3): 552 - 555. (宋飞, 王博, 张宏科, 等. 多路径并行传输中接收缓存阻塞问题的研究[J]. 电子学报, 2010, 38(3): 552 - 555.)
- [7] ADHARI H, DREIBHOLZ T, BECKE M, et al. Evaluation of concurrent multipath transfer over dissimilar paths [C]// Proceedings of the 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications. Piscataway: IEEE Press, 2011: 708 - 714.
- [8] XU C, LIU T, GUAN J, et al. CMT-QA: quality-aware adaptive concurrent multipath data transfer in heterogeneous wireless networks [J]. IEEE Transactions on Mobile Computing, 2013, 12(11): 2193 - 2205.
- [9] CAO Y, XU W. A demand based packet scheduling algorithm for multipath transfer [J]. Journal of Software, 2012, 23(7): 1924 - 1934. (曹宇, 徐伟明. 一种按需分配的多路径传输分组调度算法[J]. 软件学报, 2012, 23(7): 1924 - 1934.)
- [10] DREIBHOLZ T, RATHGEB E P, RÜNGELER I, et al. Stream control transmission protocol: past, current, and future standardization activities [J]. IEEE Communications Magazine, 2011, 49(4): 82 - 88.
- [11] YU B, YU D, SUN J. Application of Markov decision process in scheduling algorithm on concurrent multipath redundancy transmission [J]. Journal of Chinese Computer Systems, 2012, 33(4): 847 - 851. (于波, 于东, 孙建伟. 马尔可夫决策过程在多路径冗余传输调度算法中的应用[J]. 小型微型计算机系统, 2012, 33(4): 847 - 851.)
- [12] CAO Y, XU M, FU X. Delay-based congestion control for multipath TCP [C]// Proceedings of the 2012 20th IEEE International Conference on Network Protocol. Washington, DC: IEEE Computer Society, 2012: 1 - 10.
- [13] UC Berkeley, LBL, USC/ISI, et al. NS-2 documentation and software, version 2.34 [EB/OL]. [2013-06-03]. <http://www.isi.edu/nsnam/ns/>.