

基于 MPTCP 的前向预测数据调度方法*

胡庆, 邹然[†], 周莉

(重庆邮电大学软件技术中心, 重庆 400065)

摘要: 现有 MPTCP 的数据调度方案无法满足多路径的动态变化, 出现链路传输性能降低的问题。通过对每条路径上的 ACK 数、往返时延进行预测, 提出一种前向预测速度调度方法。利用 NS-3 对该算法仿真, 表明该算法能提高吞吐量, 更加合理, 并使负载更加均衡。

关键词: 多路径传输协议; 数据调度; 前向预测; 吞吐量

中图分类号: TP393 **文献标志码:** A **文章编号:** 1001-3695(2013)02-0560-02

doi:10.3969/j.issn.1001-3695.2013.02.067

Forward prediction data scheduling mechanism for MPTCP

HU Qing, ZOU Ran[†], ZHOU Li

(Center of Software Technology, Chongqing University of Posts & Telecommunications, Chongqing 400065, China)

Abstract: As the existing scheduling method of MPTCP could not satisfy the dynamic changes of the multi-path, the performance of transmission was reduced. This paper gave a new data scheduling algorithm of MPTCP, named FPSA which used the knowledge of time and the number of ACK estimation on each path. The simulation results of NS-3 show that, this algorithm improves throughput, and is more reasonable and more balanced load.

Key words: MPTCP; data scheduling; forward prediction; throughput

MPTCP(multi-path TCP, 多路径传输)协议支持多条路径同时传输数据,因而具有高容错性、高可靠性、高吞吐量、高安全性的特点。MPTCP 协议的思想最早是由 Huitema^[1]于 1995 年提出。2009 年, IETF 为此专门成立了 MPTCP 工作组, 主要致力于解决 MPTCP 体系结构、拥塞控制、路由、API、安全方面^[2-5]的问题。在 2011 年 3 月形成 rfc 文案^[6]。

2009 年, 研究中对 MPTCP 提出的数据调度方案有两种: 一是数据从应用层到达后马上分配给各个路径传输; 二是利用各个路径的拥塞窗口 CWND, 分配数据给可用路径。这两种方法都非常简单, 对于解决链路拥塞和缓冲拥塞几乎没有太大作用。2011 年, rfc 文案^[6]提出利用数据序号映射结合段在不同子流中的链接级序号编号, 从而使得不同子流上的段在接收端接收到后可以准确地重新排序, 但是这种方案还只是一种概念, 并没有提出具体算法, 更没有说明如何进行排序以及怎么来决定哪个包应该以什么速率在哪条子流中传输, 故十分不完善。

笔者提出一种基于 MPTCP 的前向预测调度算法(forward prediction scheduling algorithm, FPSA), 旨在减少接收端的数据重排。利用对接收到的包数量、时间等进行预测估计, 在已有 MPTCP 基础上, 提出一种更合理的数据调度算法。到目前为止, 本文已经实现了最大稳定性能的多径传输层通信, 并在接收缓存区的数据重组速度保持最低。

1 FPSA 设计

在多路径传输中建立一种有效调度机制是十分必要的, 这

一部分中笔者提出一种前向预测数据调度算法(FPSA), 专注于控制传输的数据, 使得这些数据抵达接收端时发生重传的几率最小, 并与现有 MPTCP 调度算法以及 Round Robin Scheduling 进行对比分析, 结果得出笔者提出的 FPSA 更能满足 MPTCP 对吞吐量及整体链路传输性能等的要求。

1.1 参数说明

考虑到多路径传输通过发送端可以同时使用 l 个路径传输数据给接收端。每条路径 i 都有其特点: a) RTT_i , 即包在路径 $path_i$ 上的轮时间, 其值为包从离开发送端缓存到发送端接收到反馈的确认信息 ACK 的整个时间; b) FT_i , 即包从离开发送端缓存到接收端接收到该包的整个时间, 即 RTT_i 的一半; c) S_i , 即路径 i 的拥塞窗口大小。同时, 为了避免复杂化, 包大小与最大段大小(MSS)保持一致。因此, 每条路径的包的排队延迟时间是固定的。笔者标示每条路径的变量 ε_i 为

$$\varepsilon_i = \frac{\text{packetSize}}{\text{throughput}}$$

因此, 如果在路径 $path_i$ t 时间中, 传输拥塞窗口值为 S_i 的包, 这些包的接收时间分别为 $t + FT_i + \varepsilon_i, t + FT_i + 2\varepsilon_i, \dots, t + FT_i + S_i\varepsilon_i$; 相对应地, 返回确认信息 ACK 的时间分别为 $t + RTT_i + \varepsilon_i, t + RTT_i + 2\varepsilon_i, \dots, t + RTT_i + S_i\varepsilon_i$ 。

拥塞窗口值 S_i 会随着成功接收的确认信息 ACK 的增加而增加。因此, 新的传输将在时间 $t + RTT_i + S_i\varepsilon_i$ (即在最后一个确认信息 ACK 被成功接收时) 开始。

1.2 前向预测数据调度算法设计

假设两条路径, 即 $path_i$ 和 $path_j$, 其 RTT 分别为 RTT_i 和

收稿日期: 2012-05-09; 修回日期: 2012-07-16 基金项目: 多维应急通信系统关键技术研究(CSTC2009AB2245); 重庆市教委科技计划基金资助项目(KJ090516)

作者简介: 胡庆(1958-), 女, 重庆人, 教授, 双硕士, 主要研究方向为通信与传输技术、宽带网络技术; 邹然(1988-), 女(通信作者), 硕士研究生, 主要研究方向为通信网络、通信网体系结构(summerkidram@gmail.com); 周莉(1987-), 女, 硕士研究生, 主要研究方向为网络管理及应用。

RTT_j , 且 $RTT_j > RTT_i$ 。Path_j 准备在 t 时刻开始传输新的包, 然而, 由于 path_i 传输包的速度比 path_j 快, 所以可对 path_j 上的包进行合理调度, 将 path_j 上需要传输包的数量可在其接收包之前预先调度到 path_i 上进行传输。

为实现目标, 指定以下几个观点: a) 时刻 t' 为 path_j 的接收包时间; b) 对 path_i 上的每一次传输, 计算在时刻 t' 之前其成功传包的包数量; c) 取得整个传输在时刻 t' 之前成功传包的包数量。如果 $t_i + FT_i + \varepsilon_i > t'$, 则所有 t' 时刻到达的包数量为 0, 即 $n_i = 0$ 。最后,

$$N = \sum_{t_i \leq t'} n_i$$

其中, N 为 t' 时刻之前估算出的所有成功接收包的数量。

FPSA 步骤如下: 当需要为各子流分配数据包时, 首先会按照各子流的各参数排序, 分别计算出各路径发送窗口的剩余空间, 并预测各路径能够成功传包的数量; 然后, 选取所有路径中, 需要向发送窗口添加数据包, 并且成功传包数量最小的路径作为基准路径, 为其他路径依次分配预留空间; 数据的分配按照成功传包数量由大到小的顺序分配给各路径。

以下为调度算法的伪代码:

```

while( ! pktSent)
{
  if( congestion window space)
  {
    N = estimateNBPkSDlvrldB4PktInHand()
    tempSN = SN
    SN = SN + N
    if( ! markedAsSent( SN + 1 ))
    {
      send( + + SN)
      pktSent = true
      markAssent( SN)
      if( N! = 0)
      SN = tempSN
    }
  }
  else
  SN ++
}
else
{
  if( path - i == path - i ->last path)
  {
    path - i = path - i ->first path
  }
  else
  {
    path - i = path - i ->next path
  }
}
}

```

1.3 FPSA 与现有 MPTCP 算法对比分析

2011 年, rfc 文案^[6]中提出利用路径管理原件中提供的可用路径, 数据包按照轮询的方式, 即 round robin scheduling (RRS), 以轮叫的方式依次将请求调度不同的服务器, 即每次调度执行 $i = (i + 1) \bmod n$, 并选出第 i 台服务器。算法的优点是其简洁性。但在 MPTCP 中, 会出现各路径性能的差异引起整体吞吐量下降, 不能达到各路径最佳性能, 由于该算法不记录当前所有连接的状态, 是一种无状态调度, 它无法满足多路径传输的适应性, 限制了传输的吞吐量, 加重接收端缓存压力, 降低了整体链路的性能。

笔者提出的 FPSA, 对在接收端接收到包的时间和成功传包数进行预测, 然后决定用哪一个特定的路径来传输数据包, 是一种动态预测算法, 它满足了 MPTCP 的适应性, 避免接收缓存区由于重组产生的阻塞, 提高了整体吞吐量及整体链路的性能。

2 仿真与性能分析

笔者使用网络模拟器 (NS-3)^[7] 对算法机制进行性能评估。假定两个主机间已建立多路径传输通道, 主机之间同时使用 RRS 方案和 FPSA 方案进行仿真对比, 持续传输同样的数据

流。比较这两种算法表明, 在本文机制中重传数明显减少, 特别是路径以不同延迟通信时。仿真中每个字段大小固定为 1 500 Byte, 拓扑采样时间设为 30 s。

如图 1、2 所示, 在拥塞窗口数因子上, RRS 方法显示拥塞窗口增长稳定性并不理想。虽然包独自传输, 但这些包的序列号仍然是紧密联系在一起的。FPSA 使得包在路径上有效传输, 在路径达到确定的传输层机制后增长稳定。

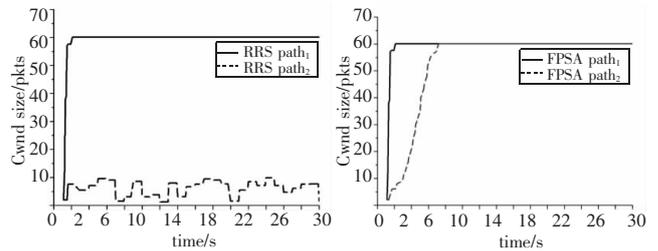


图 1 RRS 中 Cwnd 增长情况

图 2 FPSA 中 Cwnd 增长情况

在图 3 和 4 中, 对比了 FPSA 和现有 RRS 中两种路径的吞吐量。在 FPSA 中, 吞吐量在路径达到确定的传输层机制后保持稳定。

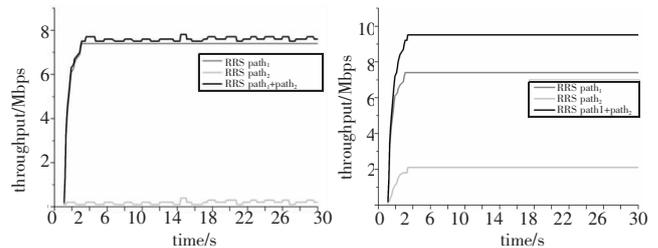


图 3 RRS 中吞吐量情况

图 4 FPSA 中吞吐量情况

另一方面, RRS 方法中由于失序数据从而发生快重传, 使得拥塞窗口一直很小, 导致堵塞, 降低了吞吐量。在图 5 中, 分别显示了 FPSA 和 RRS 中两条路径中的总成功传包数量。

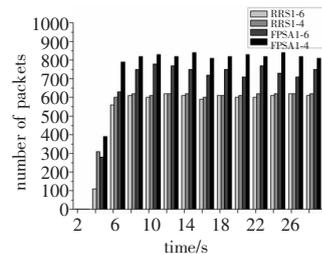


图 5 成功传输的包的总数量对比

RRS1-6 和 RRS1-4 表示 RRS 中路径延迟的比率分别设置为 1 : 6 和 1 : 4 时的情况, FPSA 中同理。FPSA 显然胜过 RRS 机制, FPSA 中路径的总成功传包数量高于 RRS 机制中路径的总成功传包数量。

3 结束语

在多路径传输中减少重传递是十分重要的。笔者提出一种预测包接收时间的调度机制, 包按序在多条路径中传输。虽然本文的调度机制是基于现有 MPTCP 的研究, 但其也可以进一步应用到 TCP 或 SCTP 中。通过使用 NS-3 进行验证, 对比结果表明本文的调度机制减少了多路径传输中的重传数, 提高了吞吐量。拉下来笔者将进一步深入对 MPTCP 数据调度中关于动态预留的研究。

参考文献:

[1] HUITEMA C. Multi-homed TCP draft-huitema-multi-homed-0 [S]. Valencia, IETF, 1995. (下转第 565 页)

没有产生拥塞,但是由于阈值 T 设置较小,使得车辆在路段上的停留时间容易超过 T ,将发出拥塞通告,其余车辆将重新规划路线绕行该路段,这样反而使得到达目的地的行车时间增大。而当 ε 设置较大时, T 值过大,即使路段真正发生了拥塞也可能导致车辆并不发出拥塞警告,使得其余车辆继续经由该路段去往目的地,从而增加了在该路段的通行时间,导致平均行车时间增大。从图 4 中可以看出, ε 的值在 2~2.5 时平均行车时间较小。 $\varepsilon < 2$ 时,由于错误的拥塞警告导致车辆绕行而使得行车时间增加;而 $\varepsilon > 3$ 时,平均行车时间显著提高,且高于某值时基本不怎么变化。因此,在后面的仿真实验中取 $\varepsilon = 2.2$ 。

表 1 仿真参数设置

参数	值
仿真区域大小	4500 m × 3500 m
路口数量	36
路段数量	62
车辆数	100, 200, 400, 600, 800, 1000
车速范围/km/h	40~80
通信半径/m	200
MAC 协议	802.11
ε	1~5
路边设施的数量	30(仅对 FBTV)

4.2 平均行车时间的比较

跟踪 5 辆汽车,统计它们在六种车辆密度情形下的平均行车时间,结果如图 5 所示。

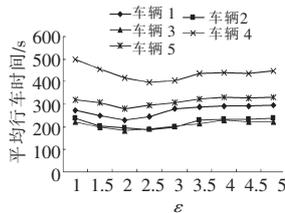


图 4 ε 对 CATP 的影响

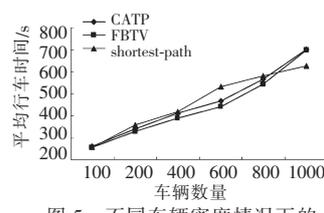


图 5 不同车辆密度情况下的平均行车时间

由图 5 中可发现,在车辆密度较小的情况下,如车辆数量为 100 或 200 时,三种出行方案的平均行车时间不是很明显。这是由于车辆的数量较少时,车流量在路段的承载范围内,不会产生路段拥塞的情况,因此无须作出行车路线的修正,CATP 和 FBTV 实际上就是运行 Dijkstra 算法,因此三者的平均行车时间差不多。

然而,当车辆的数量逐渐增大时,如在 400 或 600 时,某些路段将产生拥塞,对于 shortest-path 方案,由于一旦计算好路径就不再更改,所以当拥塞时将增大路段的通行时间,导致平均行车时间增大。而对于 CATP 和 FBTV,则可以根据拥塞情况作出行车路线的调整,因此平均行车时间比 shortest-path 方案要少。此外,FBTV 由于有路边设施的存在,相比于 CATP 更能

明确掌握对路段的拥塞情况,而 CATP 则有可能产生误判,因此 FBTV 比 CATP 的行车时间更少些。

但是当车辆密度很高时,使道路上的交通状况达到一个非正常的拥挤状态,此时 FBTV 和 CATP 方案相比于 shortest-path 的方案来说优势已经不明显。这是因为车辆密度很大时,多个路段将产生拥塞,FBTV 和 CATP 将不断重新计算路线绕行,但是并不能取得较好的效果,反而会由于绕行带来更长的平均行车时间。

5 结束语

出行计划是车载自组织网络中的一个重要应用。本文提出了一种拥塞自适应的动态出行计划方案 CATP,每辆汽车能够获取各路段的平均通行时间或平均停留时间,根据这些信息判断是否要经过这些路段,并重新计算新的行车路径,从而帮助用户优化其出行计划。通过仿真实验可以看出,在车辆密度中等的情形下,CATP 能够获得较短的平均行车时间,优化了驾驶者的行车路线,从而提高道路交通效率。

参考文献:

- [1] U S Department of Transportation. Ten-year national program plan and research agenda for intelligent transportation systems in the United States [R]. Washington DC: The Intelligent Transportation Society of America and the United States Department of Transportation, 2001.
- [2] HOH B, GRUTESER M, HERRING R, et al. Virtual trip lines for distributed privacy-preserving traffic monitoring [C]//Proc of the 6th International Conference on Mobile Systems. 2008:15-28.
- [3] FERNANDEZ-CABALLERO A, GOMEZ F J, LOPEZ J. Road-traffic monitoring by knowledge-driven static and dynamic image analysis [J]. *Expert Systems with Applications*, 2008, 35(3):701-719.
- [4] YU Yi-min, LIU Bin, WU Ting, et al. Flow-based travel plan via VANET [J]. *International Journal of Digital Content Technology and Its Applications*, 2011, 5(6):16-17.
- [5] ZHAO Jing, CAO Guo-hong. VADD: vehicle-assisted data delivery in vehicular Ad hoc networks [C]//Proc of Conference on Information Communications. 2006:1-12.
- [6] ROHLING H, BUSCHE H. SOTIS: a self-organizing traffic information system based on car-2-car communications [C]//Proc of Workshop on Self-Organising Wireless Sensor and Communication Networks. 2009:329-339.
- [7] YANG Yi, BAGRODIA R. Evaluation of VANET-based advanced intelligent transportation systems [C]//Proc of the 6th ACM International Workshop on Vehicular Internet Working. 2009:3-12.
- [8] SONG Chao, LIU Ming, GONG Hai-gang. Distributed real-time scheme based algorithm for travel plan in vehicular Ad hoc networks [J]. *Journal of Application Research of Computers*, 2009, 26(9):3440-3441.

(上接第 561 页)

- [2] FORD A, RAICIU C, HANDLE M. Draft-IETF-MPTCP-multiaddressed-02, TCP extensions for multipath operation with multiple address [S]. Valencia: IETF, 2010.
- [3] FORD A, RAICIU C, HANDLEY M, et al. Draft-IETF-MPTCP-multiaddressed-03 [S]. Valencia: IETF, 2010.
- [4] RAICIU C, HANDLEY M, WISCHIK D, et al. Draft-IETF-MPTCP-congestion-00 [S]. Valencia: IETF, 2010.
- [5] BAGNULO M. Draft-IETF-MPTCP-threat-06 [S]. Valencia: IETF, 2010.

- [6] FORD A, RAICIU C, HANDLEY M, et al. RFC 6182, Architectural guidelines for multi-path TCP development [S]. 2011.
- [7] NS-3 project. NS-3 manual and model library and tutorial [EB/OL]. (2011-12). <http://www.nsnam.org/docs/release/3.15/manual/html/index.html>.
- [8] BERNARDI G. MPTCP: a delay-tolerant transport protocol for Ad hoc networks [D]. Dublin: University of Dublin, 2006.
- [9] BARRE S, PAASCH C, BONAVENTURE O. Multi-path TCP: from theory to practice [S]. Valencia: IETF Networking, 2011.