

异构网络中基于最小传输时间估计的MPQUIC 路径调度

姜慧良¹, 申佳胤¹, 王忠峰¹, 刘明鑫², 徐佳³

(1. 国铁吉讯科技有限公司; 2. 北京经纬信息技术有限公司, 北京 100081;
3. 南京邮电大学 计算机学院, 江苏 南京 210023)

摘要: 随着5G及未来6G网络的发展, 通信需求不断增加, 这对网络的可靠性、吞吐量和延迟提出了更高要求。异构网络环境中, 不同类型的网络(如地面5G网络、卫星网络、Wi-Fi等)能够联合提供更广泛的覆盖和灵活的资源调度。在此背景下, 提出一种基于最小传输时间估计的MPQUIC路径调度算法, 并将其应用于多路径传输协议, 以提高异构网络环境下的数据传输性能。针对动态且异构的网络链路, 现有的MPQUIC路径调度算法难以充分发挥各链路的带宽潜力, 导致吞吐量和整体性能表现不佳。为此, 所设计的算法在分配数据包时估计各路径所需传输时间, 并通过减少不同路径间的传输时间差异, 以缓解队头阻塞问题, 降低整体传输时间。与其他调度算法将所有数据包集中分配到表现最好的路径相比, 该算法能够充分利用异构路径的带宽资源, 提高整体网络传输性能。此外, 基于NS-3网络仿真器模拟了其在异构网络环境下的数据传输任务。实验结果表明, 所提算法相较于对比算法在平均吞吐量上提升约9.8%, 文件传输完成时间平均减少约11.3%。

关键词: 异构网络; 多路径传输; 路径调度; 移动通信

DOI: 10.11907/rjdk.242009

扫描二维码阅读全文:



中图分类号: TP393

文献标识码: A

文章编号: 1672-7800(2026)003-0122-08

MPQUIC Path Scheduling Based on Minimum Transmission Time Estimation in Heterogeneous Networks

JIANG Huiliang¹, SHEN Jiayin¹, WANG Zhongfeng¹, LIU Mingxin², XU Jia³

(1. China Railway Geceent Technology Co., Ltd.; 2. Beijing Jingwei Information Technology Co., Ltd., Beijing 100081, China; 3. Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: With the development of 5G and future 6G networks, communication demands are continuously increasing, driving higher expectations for network reliability, throughput, and latency. In heterogeneous network environments, different types of networks (such as ground 5G networks, satellite networks, Wi-Fi, etc.) can collaborate to provide broader coverage and flexible resource scheduling. In this background, this paper proposes a multi-path scheduling algorithm based on Minimum Transmission Time Estimation (MTTE), applied to the Multipath Quick UDP Internet Connections (MPQUIC) protocol, to enhance data transmission performance in heterogeneous network environments. Existing MPQUIC path scheduling algorithms struggle to fully exploit the bandwidth potential of dynamic and heterogeneous network links, leading to suboptimal throughput and overall performance. To address this issue, the proposed algorithm estimates the transmission time for each path during data packet allocation and reduces the differences of transmission time among the paths to alleviate Head-of-Line (HoL) blocking, ultimately minimizing the overall transmission time. Unlike other scheduling algorithms that allocate all data packets to the best-performing path, the proposed algorithm fully leverages the bandwidth resources of heterogeneous paths to enhance overall network transmission performance. We simulate the data transmission tasks under various network environments using NS-3 network simulator. The experimental re-

收稿日期: 2025-01-14

基金项目: 中国铁道科学研究院集团有限公司基金课题重点项目(2022YJ358); 国家自然科学基金面上项目(62372249)

作者简介: 姜慧良(1981-), 男, 国铁吉讯科技有限公司工程师, 研究方向为移动通信; 申佳胤(1986-), 男, 国铁吉讯科技有限公司高级工程师, 研究方向为移动通信; 王忠峰(1982-), 男, 国铁吉讯科技有限公司高级工程师, 研究方向为移动通信; 刘明鑫(1989-), 女, 北京经纬信息技术有限公司工程师, 研究方向为移动通信; 徐佳(1980-), 男, 博士, CCF高级会员, 南京邮电大学计算机学院教授、博士生导师, 研究方向为众包、边缘计算和无线传感器网络。本文通讯作者: 徐佳。

sults show that the proposed algorithm improves the throughput by approximately 9.8% and reduces the file transmission completion time by about 11.3% on average compared to baseline algorithms.

Key Words: heterogeneous networks; multipath transmission; path scheduling; mobile communication

0 引言

近年来,随着5G移动通信的快速发展,通信需求显著增加,这对通信质量的要求也大幅提升^[1-4]。与此同时,为偏远地区提供无缝、统一的通信覆盖成为亟待解决的重要问题。在异构网络环境中,不同类型的网络(如卫星网络^[5]、WiFi网络、5G网络等)结合使用,可以提供广泛的全球覆盖和灵活的资源管理。星地一体化网络(Satellite-Ground Integrated Network, SGIN)作为一种典型的异构网络方案,近年来受到了学术界和产业界的广泛关注^[6-7]。然而,在此类异构网络中,卫星链路不稳定性^[8]、用户移动性^[9]、网络拥塞、随机丢包等问题,往往会使得网络性能未能达预期。如何在这样的异构网络环境中实现高效、稳定的数据传输,已成为一个亟待解决的挑战。

图1展示了在异构网络环境中,多路径传输协议的应用。通过同时利用多个网络接口(如卫星链路、5G网络等)进行数据传输,可以有效提高吞吐量,并在某一网络路径出现故障时,避免连接中断,这种设计被称为多路径传输协议。目前,被广泛采用的多路径传输协议是多路径TCP(Multipath TCP, MPTCP),它是TCP的扩展版本,并已在iOS和Android等智能手机操作系统中得到支持^[10]。尽管MPTCP已被广泛研究,但其内核实现限制了灵活性和应用层的性能优化。

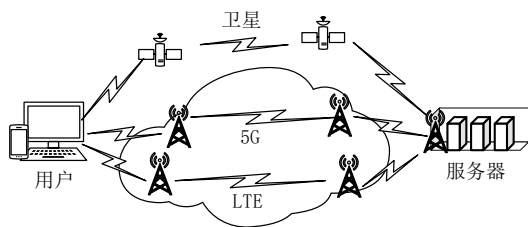


Fig. 1 Multipath transmission in heterogeneous network environment

图1 异构网络环境中的多路径传输

与此不同,多路径快速用户数据报互联网连接(Multipath Quick UDP Internet Connections, MPQUIC)实现在用户空间,这使得它具有更高的灵活性,具备广泛部署的巨大潜力^[11]。例如,阿里巴巴在其生产环境中进行的大规模MPQUIC视频传输A/B测试,验证了MPQUIC的可行性^[12]。由于HTTP/3.0采用QUIC作为其底层传输协议,MPQUIC为未来的Web的传输提供了广阔的前景^[18]。MPQUIC继承了MPTCP的多流传输设计,并有效解决了MPTCP的一些问题,如长时间的握手过程和不支持连接迁移等。

然而,在异构网络中应用MPQUIC时,如何高效地调度异构路径以应对不同网络路径之间的时延、带宽波动和拥塞等差异,仍然是一个挑战。若没有合适的调度机制,

MPQUIC可能无法充分发挥其潜力,甚至可能在某些情况下表现得与单路径UDP类似。特别是在面对不同网络路径的特性时,不合理的数据包分配可能会导致队头阻塞(Head-of-Line Blocking, HoL)问题,进一步影响整体性能。

当前的MPQUIC实现是使用最低往返时延(Minimum Round-Trip Time, MinRTT)优先调度^[12]。该调度程序将拥塞窗口已满的路径视为不可用,从而阻止它们被调度传输。在决策时,MinRTT优先考虑具有最低往返时延(Round-Trip Time, RTT)的可用路径进行数据传输。但是,在面对异构路径时,始终选择具有最低RTT的可用路径可能会导致性能下降,其原因主要包括:首先,MinRTT在作出决策时只考虑可用路径的RTT,而忽视了通过等待繁忙但快速的路径再次可用,从而缩短整体传输时间的可能性。如果即将完成数据传输的较快路径被视为不可用并被MinRTT排除,系统将不得不选择较慢的路径进行数据传输;其次,从长远来看,只选择具有最低RTT的路径会增加缓冲区膨胀的可能性;最后,MinRTT调度算法没有考虑丢包和拥塞窗口变化对路径性能的长期影响。在拥塞或丢包情况下,RTT最低的路径可能仍然被优先选择,但其实际可用带宽可能已减少。

为了解决MPQUIC由于路径异构导致的性能不佳问题,本文提出了一种基于最小传输时间估计(Minimum Transmission Time Estimation, MTTE)的路径调度算法。算法将综合考虑各传输路径的可用窗口大小和时延,估计每条路径的传输时间,并计算使得整体传输时间最小的数据包分配策略。本文提出的路径调度算法能充分使用所有的可用路径,避免了因为某条路径传输完成时间过长而导致整体的传输完成时间过长,从而减少数据传输时间。

本文主要贡献如下:①构建了异构网络中MPQUIC数据包调度模型,形式化了以最小化所有路径最大传输完成时间为目标的路径调度问题,并提出了一个基于最小传输时间估计的路径调度算法。该算法通过综合考虑各传输路径的特性,在为每条路径分配数据包时估计其所需要的传输时间,并通过减少不同路径间的传输时间差异,以缓解链路异构场景中的队头阻塞问题,降低整体传输时间;②与其他路径调度方法不同,本文提出的MTTE路径调度算法能动态响应网络变化,它不仅关注路径的当前RTT,还结合路径的拥塞窗口进行动态调整,从而在最大化吞吐量的同时,避免了因路径拥塞而引起的性能下降,显著降低了传输完成时间;③基于NS-3实验平台评估了所提出的MTTE路径调度算法性能。实验结果表明,与ECF相比,MTTE算法不仅可以充分利用异构路径,平均吞吐量提升了约9.8%,还有效缓解了队头阻塞(HoL)问题,显著优化了整体传输效率,其文件传输完成时间平均减少了约11.3%。

1 相关工作

1.1 QUIC 与 MPQUIC

最广为人知的多路径传输协议是 IETF 提出的 MPTCP^[10]。然而,基于 MPTCP 的方案在实际应用中暴露出一些基本缺陷^[14-15]:首先,MPTCP 是在操作系统的内核中实现,由于 TCP 几十年来的主导地位,使得大规模部署具有挑战性;其次,MPTCP 需要设计一种复杂的数据包调度算法,以解决异构非对称链路上乱序数据包引起的队头阻塞问题。

为了提高 HTTP 流量的性能,Google 开发了 QUIC 协议以克服 TLS/TCP 生态系统中的固有限制。Google 还将对 QUIC 的支持集成到了 Chrome 浏览器中,并在 2017 年验证了其大规模部署的可行性和性能^[16]。QUIC 采用基于 UDP 的轻量级数据结构概念,称为流,它可以在单个 QUIC 连接中进行多路复用。此设计可确保单个数据包的丢失仅影响包含该数据包中数据的特定流,而不是所有流都受到影响。与 TLS/TCP 相比,基于 QUIC 的 HTTP/2.0 具有多路复用功能,可以显著缓解 TCP 单字节节流阻塞引起的整个连接队头阻塞。它还提供了从内核空间过渡到用户空间的灵活性,以实现快速部署^[17]。认识到这些显著优势后,IETF 的 HTTP 和 QUIC 工作组合作,将 QUIC 上的 HTTP 映射命名为 HTTP/3.0,并将其确立为全球标准^[19]。

鉴于多路径功能的优势,在可预见的未来,多路径 QUIC 扩展可能会被广泛采用,包括 MPQUIC^[12]、多流 QUIC^[19]和插件化 QUIC^[20]。与此同时,Viernickel 等^[21]提出一种可部署的多路径 QUIC 协议,并证明了多路径 QUIC 可以实现比 MPTCP 更高的端到端吞吐量,特别是对于异构非对称路径,他们声称 MPQUIC 比 MPTCP 更适合当今的 Internet。在标准化工作方面,IETF 还致力于扩展 QUIC 以支持多路径,同时提出了相关工作草案^[22]。

1.2 MPQUIC 中的多路径调度算法

MPQUIC 的默认路径调度算法 MinRTT 选择具有最低 RTT 和 CWND 可用的路径以传输数据包^[12]。当此路径不可用,即拥塞窗口已满时,继续在剩余的可用路径中选择 RTT 最低的路径。因此,MinRTT 面临的一个挑战是忽略了通过等待最低 RTT 路径再次可用来缩短整体传输时间的可能性。

为了克服 MinRTT 的限制,BLEST^[23]和 ECF^[24]引入了一种等待机制,旨在平衡高吞吐量和低延迟。在正常情况下,当所有路径都有足够的拥塞窗口时,BLEST 的行为类似于 MinRTT。但是,如果最低 RTT 路径当前不可用时,BLEST 会评估是在较高 RTT 路径上传输数据,还是等到最低 RTT 路径再次可用。此外,BLEST 估计了在子流上发送一定数量的数据时 HoL 阻塞的风险,并决定数据包分配以最大限度地减少 HoL 阻塞。同样,ECF 也采用了类似于

BLEST 的等待机制。不同的是,ECF 将计算减少最低 RTT 路径空闲时间的收益。此外,ECF 在调度之前将检查每条路径的数据包到达时间,从而减少无序数据包的数量。然而,由于 BLEST 和 ECF 在调度过程中未能充分考虑路径的拥塞窗口特性,仅依赖时延和数据包到达时间进行调度,在实际异构网络中,面对显著的时延波动和数据包丢失等情况,其性能往往难以保持稳定。

为充分利用异构网络上的多路径资源,进一步提高 MPQUIC 的性能,本文提出了一种基于最小传输时间估计(MTTE)的 MPQUIC 路径调度算法。该算法将综合考虑各传输路径的特性,为每条路径估计传输分配到的数据包所需要的时间,并通过减少不同路径间的传输时间差异以最小化整体传输时间,进而缓解队头阻塞问题。此外,为了适应网络动态变化,MTTE 算法不仅关注路径时延,还结合路径的拥塞窗口进行动态调整,从而在最大化吞吐量的同时,避免了路径拥塞引起的性能下降。

2 基于最小传输时间估计的 MPQUIC 路径调度

本文首先构建了异构网络中 MPQUIC 数据包调度模型,将以最小化所有路径最大传输时间为目标的路径调度问题进行形式化。接着,提出了一种基于最小传输时间估计的路径调度算法以减少总体传输时间。本文常用符号如表 1 所示。

Table 1 Common symbols

表 1 常用符号

符号	涵义
P	路径集合
$\mathbb{D}^\tau$	时隙 τ 内需要传输的数据包集合
p_i	第 i 条路径
x_i^τ	时隙 τ 内分配给路径 p_i 的数据包数量
n	可用路径数量
r_i^τ	路径 p_i 在 τ 时隙开始时缓冲区剩余未发送数据包
$cwnd_i^\tau$	路径 p_i 在 τ 时隙的拥塞窗口大小
$Availcwnd_i^\tau$	路径 p_i 在 τ 时隙的可用窗口大小
T_i	路径 p_i 完成传输所需要的预计时间
$sRTT_i^\tau$	路径 p_i 在 τ 时隙的平滑往返时延
$unack_i^\tau$	路径 p_i 在 τ 时隙已发送但未收到 Ack 的包数量
$Round_i^\tau$	路径 p_i 在 τ 时隙传输 x_i^τ 个数据包所需的传输轮数

2.1 传输时间估计与问题形式化

本文为异构网络中的路径调度问题提供了一个形式化的模型。

假设 n 条可用路径集合为 $p = \{p_1, p_2, \dots, p_n\}$, 路径 p_i 在时隙 τ 时的平滑往返时延为 $sRTT_i^\tau$, 拥塞窗口为 $cwnd_i^\tau$, 路径 p_i 在 τ 时隙已发送但还未收到 Ack 的数据包数量为 $unack_i^\tau$, 则路径 p_i 在 τ 时隙的可用窗口大小 $Availcwnd_i^\tau$ 可以计算为如式(1)所示。

$$Availcwnd_i^\tau = cwnd_i^\tau - unack_i^\tau \quad (1)$$

假设在时隙 τ 内要发送的数据包集合为 $\mathcal{D}$ 。调度算法将数据 $\mathcal{D}$ 分配给各路径进行传输。令 r_i^τ 表示路径 p_i 在 τ 时隙开始时缓冲区剩余未发送数据包, $X^\tau = \{x_1^\tau, x_2^\tau, \dots, x_n^\tau\}$ 表示调度算法在时隙 τ 的分配策略集合,其中, x_i^τ 表示时隙 τ 内分配给路径 p_i 的数据包数量。因此,路径 p_i 传输分配到的 x_i^τ 个数据包所需的传输轮数 $Round_i^\tau$ 和传输时间 T_i 可以根据当前路径的窗口和时延信息分别估计为如式(2)、式(3)所示。

$$Round_i^\tau = \left\lceil \frac{x_i^\tau - Availcwnd_i^\tau + r_i^\tau}{cwnd_i^\tau} \right\rceil \quad (2)$$

$$T_i = Round_i^\tau \cdot sRTT_i^\tau + \frac{sRTT_i^\tau}{2} \quad (3)$$

本文路径调度问题的目标是为当前时隙 τ 找到一个合适的数据包分配策略集合 X^τ ,最小化所有路径的最大传输时间,如式(4)、式(5)所示。

$$\min \max \{T_1, T_1, \dots, T_n\} \quad (4)$$

$$s.t. \sum_{x_i^\tau \in X^\tau} x_i^\tau = |\mathcal{D}| \quad (5)$$

2.2 基于最小传输时间估计的路径调度算法

本文提出了一种基于最小传输时间估计的路径调度算法,该算法能够根据每条路径的当前状态(如RTT、拥塞窗口和未确认数据包数量等信息)进行动态调整,将数据包均匀分配到所有可用路径上,从而避免过度依赖单一路径的问题。具体而言,算法不仅关注路径的传输能力,还通过估计每条路径的传输时间,减少不同路径间的传输时间差异以优化数据包分配策略,缓解HoL阻塞问题,从而降低整体传输时间。与传统路径调度算法将所有数据包集中分配到表现最好的路径不同,MTTE这种均衡分配的策略能够充分利用异构路径的带宽资源,提高整体网络传输性能。此外,MTTE算法会优先考虑最小RTT的路径,并在路径不可用时及时调整分配策略。

为了实现MTTE调度算法,算法为每条路径 p_i 维护了以下关键变量:路径 p_i 的当前传输时间 T_i^{cur} ,所有路径的整体传输时间 T^{total} ,当前需要分配的数据包总数 M 。算法的工作流程分为两个主要部分:首先,通过遍历每条路径,更新路径的状态信息并计算每条路径的当前传输时间 T_i^{cur} ;其次,算法通过为整体传输时间 T^{total} 逐步增加时间步长 $span$,从而为每条路径估计在当前 T^{total} 下所能传输的数据包,其中 $span$ 为当前所有可用路径中的最小RTT。通过这种逐步增加时间步长的方式,算法能够有效控制各路径当前传输时间的差异,使其不超过最小RTT,从而减少整体传输时间 T^{total} 。在此过程中,算法还动态调整重新分配的数据包数量,以适应路径的可用性和网络状态变化。

算法1中描述了基于最小传输时间估计(MTTE)的路径调度详细设计。首先,算法遍历每条路径,更新其当前状态,包括获取每条路径的拥塞窗口 $cwnd_i^\tau$ 、剩余未发送数

据包数量 r_i^τ 、平滑往返时延 $sRTT_i^\tau$ 和可用拥塞窗口 $Availcwnd_i^\tau$ 。其中, $GetCWND(p_i)$ 函数获取路径 p_i 的拥塞窗口, $GetRemainPkt(p_i)$ 函数获取路径 p_i 的剩余未发送数据量, $GetSmoothedRTT(p_i)$ 函数计算路径 p_i 平滑往返时延, $GetUnackPkt(p_i)$ 函数获取路径 p_i 已发送但还未收到Ack的数据包数量(行2-6)。其次,算法判断路径是否可用(例如:路径发生故障断开),如果路径不可用,则会将该路径的剩余数据包重新分配,并从路径集合中移除该路径(行7-9);如果可用,算法通过式(2)和式(3)估计当前路径的传输时间 T_i^{cur} ,并更新当前可用的拥塞窗口 $Availcwnd_i^\tau$ (行10-13)。同时,算法在遍历每条路径的过程中记录并更新当前整体传输 T^{total} (行14)。在完成所有路径的初始传输时间计算后,算法进入数据包分配阶段。具体而言,算法通过为整体传输时间 T^{total} 逐步增加时间步长 $span$,在不超过 T^{total} 的约束下,算法将为所有路径尽可能多地分配数据包,直到继续增加数据包会超出当前整体传输 T^{total} ,或所有数据包都已分配完毕(行17-28)。同时,在每次分配后,需要分配的数据包数 M 、路径的当前可用窗口 $Availcwnd_i^\tau$ 和传输时间 T_i^{cur} 都会进行更新(行22-24)。最后,算法输出数据包分配策略集合 X^τ 和整体传输 T^{total} (行29)。

算法1 基于最小传输时间估计(MTTE)的路径调度算法

输入: 路径集合 p ,时隙 τ 内要发送的数据包集合 $\mathcal{D}$

输出: 分配策略 X^τ ,整体传输时间 T^{total}

```

1. function MTTESchedule
2.  $T_i^{cur} \leftarrow 0$ ;  $T^{total} \leftarrow 0$ ;  $M \leftarrow |\mathcal{D}|$ ;  $r_i^\tau \leftarrow 0$ ;  $x_i^\tau \leftarrow 0$ ;
    $Availcwnd_i^\tau \leftarrow 0$ ;  $sRTT_i^\tau \leftarrow 0$ ;
3. FOR  $i = 0$ ;  $i < n$ ;  $i++$  DO
4.  $cwnd_i^\tau \leftarrow GetCWND(p_i)$ ;
5.  $r_i^\tau \leftarrow GetRemainPkt(p_i)$ ;
6.  $sRTT_i^\tau \leftarrow GetSmoothedRTT(p_i)$ ;
7.  $Availcwnd_i^\tau \leftarrow cwnd_i^\tau - GetUnackPkt(p_i)$ ;
8. IF  $p_i$  is not available THEN
9.  $M \leftarrow M + r_i^\tau$ ;
10.  $p \leftarrow p \setminus \{p_i\}$ ;
11. ELSE
12.  $T_i^{cur} \leftarrow \left\lceil (r_i^\tau - Availcwnd_i^\tau) / cwnd_i^\tau \right\rceil \cdot sRTT_i^\tau$ ;
13.  $Availcwnd_i^\tau \leftarrow (r_i^\tau - Availcwnd_i^\tau) \bmod cwnd_i^\tau$ ;
14. END IF
15.  $T^{total} \leftarrow \max\{T^{total}, T_i^{cur}\}$ ;
16. END FOR
17.  $span \leftarrow GetMinRTT()$ ;
18. WHILE  $M > 0$  DO
19. FOR  $i = 0$ ;  $i < n$ ;  $i++$  DO
20. WHILE  $T_i^{cur} + sRTT_i^\tau \leq T^{total} \ \&\& \ M > 0$  DO
21.  $x_i^\tau \leftarrow x_i^\tau + \min(Availcwnd_i^\tau, M)$ ;
22.  $M \leftarrow M - \min(Availcwnd_i^\tau, M)$ ;
23.  $Availcwnd_i^\tau \leftarrow cwnd_i^\tau$ ;
24.  $T_i^{cur} \leftarrow T_i^{cur} + sRTT_i^\tau$ ;

```

```

25. END WHILE
26. END FOR
27. if  $M > 0$  THEN  $T^{total} \leftarrow T^{total} + sapn$ ;
28. END WHILE
29. RETURN  $X^*, T^{total}$ ;
30. END function

```

3 性能评估

3.1 实验环境

本文使用 NS-3 网络仿真器以评估所提出的基于最小传输时间估计 (MTTE) 的路径调度算法。NS-3 中的 MPQUIC 模块是基于 Padova 大学实现的 NS-3 QUIC 项目^[13]。在不同的网络状态下进行文件传输, 将其传输完成时间和吞吐量与 MinRTT^[12]、ECF^[23]、BLEST^[24]、PEEKABOO^[25] 4 个调度算法进行对比。

通过评估不同文件大小在异构路径状态下的传输完成时间以评价不同调度算法的性能。为了模拟不同的网络状态, 本文参考 MPQUIC 的实验加以实现^[13], 将默认网络拓扑设计为由 3 条独立的传输路径与基于这 3 条路径进行文件传输的客户端和服务端组成, 其网络拓扑如图 2 所示。并且, 在 NS-3 模拟平台上通过设置不同链路参数以模拟异构网络环境, 分别模拟异构网络中时延异构、带宽异构、时延与带宽异构对多路径数据传输的影响。在这些仿真网络环境上, 使用不同的路径调度算法进行数据传输, 并记录客户端从开始到完成所有数据传输所消耗的时间。此外, 在实验中使用了机会链路增加 (Opportunistic Linked Increased Algorithm, OLIA) 拥塞控制算法^[26]。

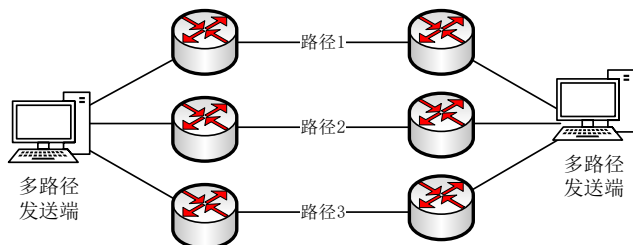


Fig. 2 MPQUIC network topology

图2 MPQUIC 网络拓扑

3.2 不同调度算法性能比较

本文首先将路径 1 的带宽、时延、路径损失率分别设置为 5~7.5 Mbps、100~120 ms、0.1%, 以模拟卫星链路, 将其余两条路径的带宽、时延、路径损失率都分别设置为 10~15 Mbps、20~30 ms、0.05%, 以模拟地面链路。其次, 通过传输不同大小文件的完成时间以比较每种算法的性能, 结果如图 3 所示。可以看出, 本文基于最小传输时间估计 (MTTE) 算法的传输完成时间始终保持最低, 并且相比性能次之的 PEEKABOO 算法, MTTE 的传输完成时间平均减少了约 11.3%。这种优越性来源于 MTTE 算法通过最大限度地减少不同路径间的传输时间差异以优化数据包分配

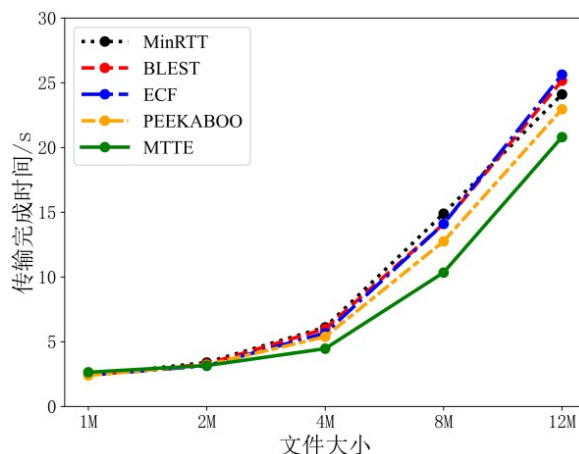


Fig. 3 Transfer completion time under different file sizes

图3 不同文件大小下的传输完成时间

策略, 从而使得每条路径得到均衡的负载, 避免了单一路径过载, 降低了传输完成时间的增长幅度; 而 PEEKABOO、BLEST 和 ECF 虽然考虑了不对称异构链路的影响, 但由于无法充分利用所有路径, 因此其表现没有达到理想效果。

本文提取了传输 12 M 文件时不同调度算法的瞬时吞吐量记录, 如图 4 所示。与 PEEKABOO 相比, 本文提出的 MTTE 调度算法平均吞吐量提升了约 9.8%。值得注意的是, 吞吐量在初期都呈指数式增长, 而随着时间推移吞吐量出现回落, 并最终趋于平稳, 这一现象主要是由于使用了 OLIA 拥塞控制算法。在网络初期, OLIA 能够充分利用带宽较高的路径, 迅速提高吞吐量。然而, 随着数据传输的持续进行, 网络中的拥塞逐渐积累, 导致了吞吐量的回落。最终, OLIA 通过自适应调整拥塞窗口, 使得吞吐量趋于稳定。

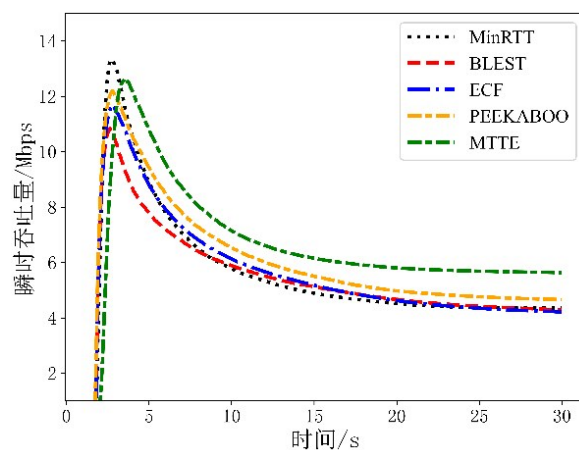


Fig. 4 Comparison of instantaneous throughput of different scheduling algorithms

图4 不同调度算法的瞬时吞吐量比较

为了模拟异构路径数量对不同调度算法性能的影响, 在实验场景中分别使用了 2~6 条异构路径以传输 8 M 大小的文件, 每条路径的损失率分别固定为 0.1%, 路径带宽分别从 5 M 增加到 30 M, 时延则分别从 70~50 ms 减少到 20~

40 ms。每种算法的结果如图5所示。可以看出,本文提出的MTTE算法在不同异构路径数量下的传输完成时间均显著优于其他4种调度算法。随着路径数量从2条增加到6条,MTTE的传输完成时间持续减少,这表明其能够充分利用新增路径的带宽资源。相比之下,PEEKABOO算法的性能虽然优于BLEST和ECF,但随着路径数量的增加,其传输完成时间减少的幅度较小。这是由于PEEKABOO在调度过程中虽然能够通过随机性和确定性策略适应异构路径的特性,但未能像MTTE一样实时估计各路径的传输时间并均衡分配负载,因此性能提升不如MTTE算法。而BLEST和ECF由于未能充分考虑路径的拥塞窗口特性,仅依赖时延和数据包到达时间进行调度,因此路径数量增加时性能变化不大,表现出较为稳定的趋势。此外,MinRTT由于仅关注最低时延路径,未能有效使用其他慢速路径的带宽资源,其传输完成时间始终保持在较高水平。

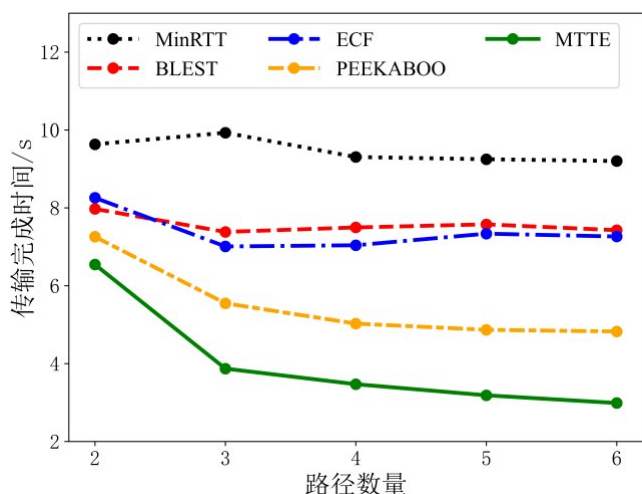


Fig. 5 Transmission completion time under different numbers of heterogeneous paths

图5 不同异构路径数量下的传输完成时间

为了模拟时延异构对不同调度算法性能的影响,在实验场景中同时使用路径1和路径2以传输2 M大小的文件,并将两条路径的带宽和路径损失率分别固定为10 Mbps和0.1%,将路径1的时延设置为20 ms,将路径2的时延从20 ms不断增加到200 ms,以增加两条路径之间的时延异构性。每种算法的结果如图6所示。可以看出,随着路径2的时延增加到路径1的2~6倍时,本文提出的MTTE在传输系统大小的文件时所需时间都要小于MinRTT、BLEST、ECF和PEEKABOO。这是由于MTTE可以根据链路特性预测传输时间以均衡分配数据包,这种策略不仅可以充分利用异构链路的带宽,而且还考虑了时延异构对HOL阻塞的影响。此外,由于MinRTT不考虑等待繁忙但时延较低的路径,无法充分利用路径1的带宽,因此其整体性能显著下降。随着路径2的往返延迟继续增加至路径1的8~10倍时,4种调度算法都趋于平缓,这是因为此时较低延迟的路径1已经成为数据传输的主要通道,而高延

迟的路径2对整体性能的影响逐渐被削弱。

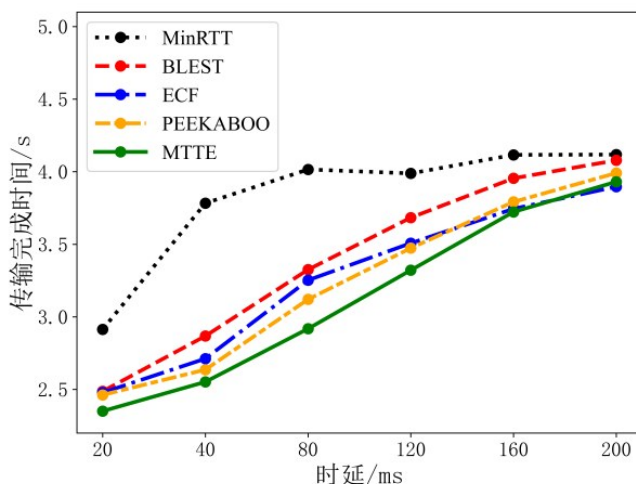


Fig. 6 Transmission completion time under different delays

图6 不同时延下的传输完成时间

为了模拟带宽异构对调度算法性能的影响,在实验场景中同时使用路径1和路径2传输2 M大小的文件,同时将两条路径的时延固定为20~30 ms,路径损失率固定为0.1%,将路径1的带宽设置为5 Mbps,将路径2的带宽从5 Mbps不断增加到30 Mbps,以增加两条路径之间的带宽异构性。实验结果如图7所示。可以看出,在两条路径带宽相等时,本文提出的MTTE算法优于其他对比算法,而且随着路径带宽异构性逐渐增大时,MTTE算法依旧优于其他算法。这归功于本文在数据包调度时会不增加整体传输时间的基础上将数据包均匀地分配给每条路径,从而提高了吞吐量。当路径2的带宽逐步增加至路径1带宽的5~6倍时,MTTE、PEEKABOO、BLEST和ECF传输完成时间都趋于一致,主要是因为此时高带宽路径2已经成为数据传输的主要通道,而低带宽路径对整体性能的影响逐渐被削弱。其中,MinRTT算法在决策过程中只考虑RTT最小的路径,而忽略了路径的带宽特性,导致大量数据被分配到带宽较低的路径,从而拖累了整体传输性能。因此,MinRTT算法在带宽异构环境下表现最差。

最后,为了同时模拟时延异构与带宽异构,在实验场景中同时使用路径1和路径2传输2 M大小的文件,并将两条路径的损失率固定为0.1%,其中路径1的带宽与时延固定为5 Mbps与20~30 ms,而路径2的带宽与时延分别设置为20 Mbps、10 Mbps与40 ms、60 ms、80 ms、100 ms,以同时增加路径的时延异构性与带宽异构性。实验结果如图8所示。可以看出,路径的时延异构性与带宽异构性同时增大时,MTTE始终保持较为稳定且优异的传输性能。相比于其他对比算法,MTTE的优势在于其调度决策是基于对传输时间的预测和调整。这使得MTTE能够有效应对路径特性的复杂变化,在充分利用高带宽路径的同时,平衡了时延较高路径的负面影响,从而提升整体数据传输效率。特别地,当路径2的带宽从20 Mbps降至10 Mbps时,

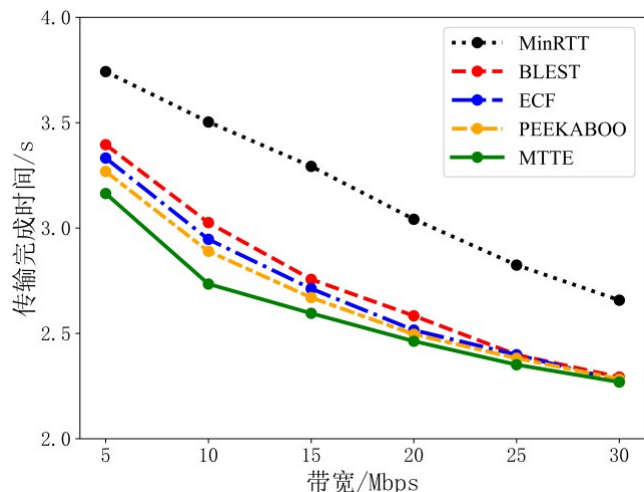


Fig. 7 Transmission completion time under different bandwidths

图7 不同带宽下的传输完成时间

在相同的时延条件下,传输完成时间普遍上升。这一结果符合带宽减小导致传输性能下降的预期,进一步验证了带宽对传输效率的重要影响。

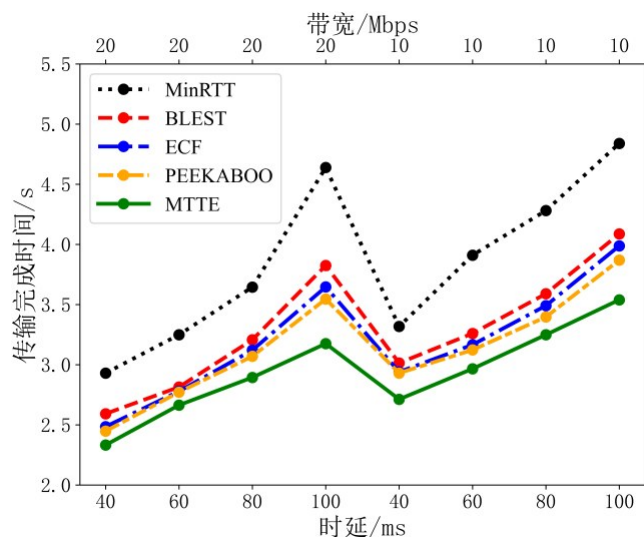


Fig. 8 Transmission completion time under different delays and bandwidths

图8 不同时延和带宽下的传输完成时间

4 结语

本文设计了一种新的适应异构网络场景的MPQUIC路径调度算法MTTE。该算法通过估计路径传输数据包所需时间,动态调整数据包的分配策略,使得数据包能够在多条异构路径之间均衡分配,最大化网络带宽的利用效率,并有效减少整体传输时间。同时,算法通过减少不同路径间的传输时间差异,缓解了队头阻塞问题,进一步优化了传输性能。为适应网络的动态变化,MTTE算法不仅关注路径时延,还结合路径的拥塞窗口进行动态调整,从

而在最大化吞吐量的同时,避免路径拥塞引起的性能下降。并且,本文基于NS-3仿真平台验证了算法的有效性。实验表明,相比其他对比算法,MTTE在异构网络中执行传输任务的效率显著提升,其平均吞吐量提升约9.8%,文件传输完成时间平均减少约11.3%。

尽管,本文的实验验证主要集中在文件传输场景,但计划在未来研究中将算法应用于更为复杂的实时应用场景,特别是视频传输等高实时性要求的应用。与文件传输相比,高实时性要求的应用对延迟和带宽要求更加严格,且网络环境动态变化对算法的适应性提出了更高挑战。因此,未来将重点优化算法在高实时性应用场景中的表现,尤其是在减少延迟、避免拥塞和确保传输质量方面。此外,还将探索算法在大规模网络环境中的应用,以评估其在复杂环境下的效率和可行性。通过这些后续研究,期望进一步拓展算法应用范围,并为实际网络中的多路径调度提供更具实用价值的解决方案。

参考文献:

- [1] LIANG Y C, ZHANG Q, LARSSON E G, et al. Symbiotic radio: cognitive backscattering communications for future wireless networks [J]. IEEE Transactions on Cognitive Communications and Networking, 2020, 6(4): 1242-1255.
- [2] SONG L Y, DI B, ZHANG H L, et al. Aerial access networks: integration of UAVs HAPs and satellites [M]. Cambridge: Cambridge University Press, 2023.
- [3] TATARIA H, SHAFI M, MOLISCH A F, et al. 6G wireless systems: vision requirements challenges insights and opportunities [J]. Proceedings of the IEEE, 2021, 109(7): 1166-1199.
- [4] YANG J Y, XIAO Z Y, CUI H X, et al. DQN-ALRM-based intelligent handover method for satellite-ground integrated network [J]. IEEE Transactions on Cognitive Communications and Networking, 2023, 9(4): 977-990.
- [5] WU M, LUO W, ZHANG S Y, et al. Research on space coverage performance evaluation of broadband satellite networks based on hybrid constellations [J]. Computational Linguistics Journal, 2016, 15(2): 111-114.
吴明, 罗威, 张思耀, 等. 基于混合星座的宽带卫星网络空间覆盖性能评估研究 [J]. 软件导刊, 2016, 15(2): 111-114.
- [6] CUI H X, ZHANG J, GENG Y H, et al. Space-air-ground integrated network (SAGIN) for 6G: requirements architecture and challenges [J]. China Communications, 2022, 9(2): 90-108.
- [7] ZHOU D, SHENG M, LI J D, et al. Aerospace integrated networks innovation for empowering 6G: a survey and future challenges [J]. IEEE Communications Surveys & Tutorials, 2023, 25(2): 975-1019.
- [8] LIU Z G, LIU Z X, WANG L, et al. Traffic-predictive routing strategy for satellite networks [J]. Electronics, 2024, 13(1): 6-12.
- [9] XING Y T, XUE K P, ZHANG Y, et al. An online learning assisted packet scheduler for MPTCP in mobile networks [J]. IEEE/ACM Transactions on Networking, 2023, 31(5): 2297-2312.
- [10] HANDLEY M, BONAVENTURE O, RAICIU C, et al. TCP extensions for multipath operation with multiple addresses [EB/OL]. <https://www.rfc->

- editor. org/rfc/rfc6824. html.
- [11] CONINCK Q D, BONAVENTURE O. Multipath QUIC: design and evaluation [C]//Proceedings of the 13th International Conference on Emerging Networking Experiments and Technologies, 2017: 160–166.
 - [12] ZHENG Z L, MA Y F, LIU Y M, et al. XLINK: QoE-driven multi-path QUIC transport in large-scale video services [C]//Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '21), 2021: 418–432.
 - [13] SHU S J, YANG W J, PAN J P, et al. A multipath extension to the QUIC module for ns-3 [C]//Proceedings of the 15th Workshop on Network Simulator-3, 2023: 86–93.
 - [14] QUENTIN D C, OLIVIER B. MultipathTester: comparing MPTCP and MPQUIC in mobile environments [C]//Proceedings of IEEE Conference Network Traffic Measurement and Analysis, 2019: 221–226.
 - [15] MORAWSKI M, IGNACIUK P. MPTCP or MPQUIC—which one is better for general purpose networking [C]//Romania: 25th International Conference on System Theory, Control and Computing, 2021: 600–605.
 - [16] ADAM L, ALISTAIR R, ALYSSA W, et al. The QUIC transport protocol: design and internet-scale deployment [C]//Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '17), 2017: 183–196.
 - [17] XING Y T, XUE K P, ZHANG Y, et al. A stream-aware MPQUIC scheduler for HTTP traffic in mobile networks [J]. IEEE Transactions on Wireless Communications, 2023, 22(4): 2775–2788.
 - [18] BISHOP M. Hypertext transfer protocol version 3 (HTTP/3) [EB/OL]. <https://www.rfc-editor.org/rfc/rfc9114.html>.
 - [19] QUENTIN D C, OLIVIER B. Multi-flow QUIC: a generic multipath transport protocol [J]. IEEE Communications Magazine, 2021, 59(5): 108–113.
 - [20] QUENTIN D C, FRANCOIS M, MAXIME P, et al. Pluginizing QUIC [C]//Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM'19), 2019: 59–74.
 - [21] VIERNICKEL T, FROEMMGEN A, RIZK A, et al. Multipath QUIC: a deployable multipath transport protocol [C]//Proceedings of IEEE International Conference on Communications, 2018: 1–7.
 - [22] LIU Y M, MA Y F, QUENTIN D C, et al. Multipath extension for QUIC [EB/OL]. <https://datatracker.ietf.org/doc/html/draft-ietf-QUIC1-multipath-05>.
 - [23] FERLIN S, ALAY O, OLIVIER M, et al. BLEST: blocking estimation-based MPTCP scheduler for heterogeneous networks [C]//Proceedings of 2016 IFIP Networking Conference and Workshops, 2016: 431–439.
 - [24] LIM Y, NAHUM E M, DON T, et al. ECF: an MPTCP path scheduler to manage heterogeneous paths [C]//Proceedings of 13th ACM International Conference on Emerging Networking Experiments and Technologies, 2016: 147–159.
 - [25] WU H J, ALAY O, ANNA B, et al. Peekaboo: learning-based multipath scheduling for dynamic heterogeneous environments [J]. IEEE Journal on Selected Areas in Communications, 2020, 38(10): 2295–2310.
 - [26] KHALILI R, GAST N, POPOVIC M, et al. MPTCP is not pareto-optimal: performance issues and a possible solution [J]. IEEE/ACM Transactions on Networking, 2013, 21(5): 1651–1665.

(责任编辑:孙娟)