



## 基于灰色马尔科夫预测的移动多路传输调度算法

李 宁<sup>1</sup>, 衷璐洁<sup>1</sup>, 高 楷<sup>2</sup>

(1.首都师范大学 信息工程学院,北京 100048; 2.北京邮电大学 网络技术研究院,北京 100876)

**摘要:**多路径传输控制协议通过聚合多路径带宽提高资源利用率及网络吞吐量。在无线异构网络环境中,由于网络性能不稳定、传输路径性能差异等因素容易导致数据包乱序及缓冲区阻塞,对网络多路传输性能造成负面影响。为在有限的接收缓冲区内对数据包实施合理调度,基于灰色预测模型GM(1,N)与马尔科夫优化的前向传输时间(FTT)预测模型,提出一种自适应多路传输数据调度算法GMM-S。通过对未来时刻子流FTT的准确预测实现子流传输性能的有效评估,并以此作为数据分发依据进行传输数据的多子流动态调整。仿真实验结果表明,与RR和LowRTT算法相比,该算法可有效解决接收端数据包乱序问题,同时提升网络吞吐量。

**关键词:**多路径传输控制协议;灰色预测模型;马尔科夫算法;数据调度;丢包

开放科学(资源服务)标志码(OSID):



中文引用格式:李宁,衷璐洁,高楷.基于灰色马尔科夫预测的移动多路传输调度算法[J].计算机工程,2021,47(3):218-226.

英文引用格式:LI Ning, ZHONG Lujie, GAO Kai. Mobile multi-path transmission scheduling algorithm based on grey Markov prediction[J]. Computer Engineering, 2021, 47(3): 218-226.

## Mobile Multi-Path Transmission Scheduling Algorithm Based on Grey Markov Prediction

LI Ning<sup>1</sup>, ZHONG Lujie<sup>1</sup>, GAO Kai<sup>2</sup>

(1. Information Engineering College, Capital Normal University, Beijing 100048, China;

2. Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

**[Abstract]** Multi-Path Control Transmission Protocol (MPTCP) improves resource utilization and network throughput by aggregating multi-path bandwidth. However, in the wireless heterogeneous network environment, out-of-order packets and buffer block are serious due to many factors, such as unstable network performance and differences between the transmission paths, which reduces the performance of multi-path transmission. In order to schedule packet reasonably in the limited receiving buffer, this paper propose an adaptive multi-path data scheduling algorithm GMM-S based on the Grey prediction Model GM(1, N) and Markov optimized Forward Transmission Time (FTT) prediction mode. The algorithm evaluates the transmission performance of sub-flows through the accurate prediction of FTT of sub-flows in a future moment, and takes it as a basis for data distribution to realize dynamic sub-flows adjustment of transmitted data. The simulation results show that compared with the RR algorithm and LowRTT algorithm, the proposed algorithm can effectively solve the packet disorder problem at the receiving end, and improve the network throughput.

**[Key words]** Multi-Path Transmission Control Protocol(MPTCP); Grey prediction Model(GM); Markov algorithm; data scheduling; packet loss

DOI: 10.19678/j.issn.1000-3428.0056790

### 0 概述

随着移动互联网与通信技术的快速发展,5G网络、移动车载网络中的新型业务需求对网络传输速率及带宽提出了更高要求<sup>[1]</sup>。为保证移动设备的连接性能,

通常为其配置多个接口,以降低运行成本、简化网络管理并提供良好的用户体验。多路径传输控制协议(Multi-Path Transmission Control Protocol, MPTCP)使用多路径发送数据包,可满足实时性、交互性与个性化的业务需求。因此, MPTCP成为未来互联网数据传输的发展

基金项目:国家自然科学基金“程序分析及软件定义增强融合的移动多路传输关键技术研究”(61872253)。

作者简介:李 宁(1995—),女,硕士研究生,主研方向为网络传输协议;衷璐洁(通信作者),副教授、博士;高 楷,博士研究生。

收稿日期:2019-12-04 修回日期:2020-02-08 E-mail:1129190026@163.com

趋势之一<sup>[2]</sup>,但是通过多路径传输来提高网络性能仍存在诸多挑战。例如,在有限的接收缓冲区和异构网络环境中,由于网络性能不稳定以及传输路径性能差异较大等因素容易造成接收端乱序,从而引发缓冲区阻塞问题,降低网络多路径的传输性能<sup>[3]</sup>。其中,具有代表性的场景有车载移动用户与手机移动用户通过接入各自的异构网络访问互联网时,若不对存在路径差异和无线网络不可靠的情况进行管控,将会导致分组频繁乱序及数据丢失,造成数据包重传占用链路资源和接收端缓冲区阻塞等问题,严重影响服务质量与用户体验<sup>[4]</sup>。

本文综合考虑异构网络链路差异、子流性能评估以及多路径带宽负载均衡等问题,提出一种移动多路传输调度算法。该算法利用前向传输时间(Forward Transmission Time, FTT)实现子流传输性能的准确预测,并依据FTT对数据包进行分发,保障数据包到达缓冲区的时间趋近于最优子流FTT,以提高多路径带宽利用率与数据传输效率。

## 1 相关工作

### 1.1 MPTCP数据调度

MPTCP由IETF提出<sup>[5]</sup>,其作为TCP的扩展,目的是通过聚合多路径带宽来提高资源利用率及网络吞吐量。数据调度算法是MPTCP的重要组成部分,MPTCP数据调度对应用层数据流实施分段并在每个子流上进行数据传输,同时提供耦合拥塞控制以保障与正常TCP连接的公平性<sup>[6]</sup>。传统的MPTCP数据调度算法通常采用静态和动态两类方式。其中,静态方式主要针对同构网络环境,该类算法不考虑路径的状态特性,其是通过借助拥塞窗口对数据包进行分配。静态方式的代表性算法有随机调度和轮询调度,轮询调度(Round-Robin, RR)是一种广泛应用于数据包分配<sup>[7]</sup>的算法,该算法对将要发送的数据段,从当前所有可用路径中按子路径从1~N的顺序选取下一个有空余发送窗口的路径作为本次调度的路径选择。静态方式对于同构网络的性能表现较优,但在异构网络环境中其传输性能没有单路径传输更有优势。动态方式针对静态调度策略的不足,将路径特性纳入考虑,且每次对路径选择时根据各路径的传输特性优先选择传输质量较好的路径进行数据传输<sup>[8]</sup>。其中,基于最小往返时延(Round-Trip Time, RTT)的调度算法LowRTT<sup>[9]</sup>优先选择传输延迟最小的路径作为最佳路径,且当最佳路径的发送窗口为0时,再选择传输时延次小的路径作为最佳路径。与RR算法相比,LowRTT算法能够更好地缓解数据包乱序问题,但由于每次仅选用一条子流进行数据传输,其未能充分利用多路径并行传输的优势<sup>[10]</sup>。

### 1.2 灰色马尔科夫预测模型

灰色预测模型(Grey prediction Model, GM)是一种分析“部分信息已知,部分信息未知”不确定性系统问题的有效方法<sup>[11]</sup>。该模型通过累加生成方式

产生新的数据序列,使用新产生的数据序列代替原始数据序列,并进一步应用微分方程求解完成新数据序列的预测。与目前主流的指数平滑、回归分析、支持向量机及神经网络等预测方法相比,灰色预测模型具有无需统计大量样本数据即可解决小样本短序列建模问题的优势。灰色预测模型GM(1, N)是GM的一种改进模型,该模型在保留小样本数据预测特点的同时,增加了影响数据变化的其他因素数据,可进一步提升灰色预测模型的精度<sup>[12]</sup>。

马尔科夫模型对随机变化的动态系统进行预测时,它将时间序列看作一个随机过程,通过对事物不同状态的初始概率与状态之间的转移概率进行深入研究,确定事物未来状态的变化趋势<sup>[13]</sup>。这种模型的实质是一种建立在“状态”“状态转移”概念上的动态随机数学模型,即通过研究对象的当前时刻状态及状态转移概率来判断研究对象下一时刻的所处状态。

考虑到灰色模型的使用对象主要是预测时间短且波动性较小的系统对象,本文提出将灰色模型与马尔科夫模型相结合的多路传输数据调度机制——灰色马尔科夫多路传输调度(Grey-Markov Multi-path Transmission Scheduling, GMM-TS)。用灰色模型对数据进行拟合并找出变化趋势,然后在此基础上与马尔科夫模型相融合,利用其具有对随机过程提取概率分布规律的特点,进一步提升灰色预测模型对随机波动较大数据序列的预测准确度。

## 2 自适应多路传输数据调度机制

### 2.1 问题分析

在无线异构网络中采用MPTCP传输数据包时,不同类型的通信系统相融合需使用多条路径,且传输过程的链路性能存在较大差异。传输性能差异将造成传输序列号(Transmission Sequence Number, TSN)较大的数据包先到达缓冲区,从而引起数据包乱序问题<sup>[14]</sup>,而接收端由于无法及时将缓冲区中的数据向上层应用传输而造成数据流传输时间的延长。

图1为异构子流的多路传输示例。假设子流1的传输速率是子流2的3倍,TSN=1的数据包与TSN=2的数据包在同一时刻分别由子流1和子流2发送。当子流1发送的TSN=1的数据包到达接收端后提交至上层应用,继续发送TSN=3与TSN=4的数据包,当TSN=3与TSN=4的数据包到达接收端缓冲区后,暂存缓冲区并等待TSN=2到达后按序交付应用层。在时间段 $T$ 内提交到应用层的仅有TSN=1数据包,且此时的有效吞吐量 $Q=1/T$ ,而在顺序传输的情况下 $Q=3/T$ 。与此同时,由于接收端缓冲区的限制,乱序数据包会引发接收端缓冲区阻塞,致使拥塞控制机制减缓发送速率,甚至停止发送<sup>[15]</sup>。若MPTCP分发数据时仅考虑拥塞窗口和缓冲区大小,当发送窗口小于最大报文段MSS时,则会造成接收缓冲区中存在大量数据包碎片。由于拼接数据包会占用额外的资源,因此会加重数据包乱序。

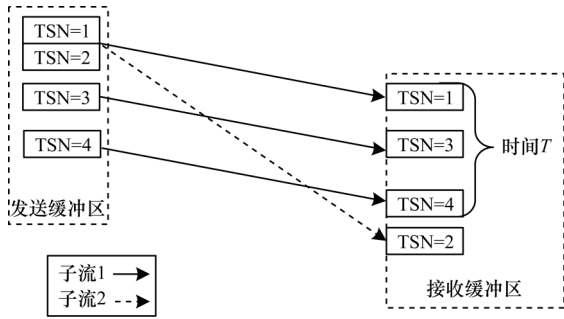


图1 异构子流的多路传输示例

Fig.1 Example of multi-path transmission of heterogeneous substreams

为缓解无线异构网络中因链路差异而造成的数据包乱序及缓冲区阻塞问题,本文提出一种基于灰色系统中离散数据预测的GM(1,N)预测模型与马尔科夫优化的自适应多路传输数据调度算法GMM-S。综合考虑子流的丢包率、发送窗口大小及拥塞状态对子流性能的影响,对未来时刻子流前向传输时延FTT进行预测并进行相关路径的性能评估,同时将其用作数据分发依据,实现传输数据的多子流自适应动态调整。

考虑到MPTCP的子流传输性能除了受到路径吞吐量Q、传输时延T、丢包率L与发送窗口S等多种因素影响外<sup>[16]</sup>,还会受到如终端与基站的距离等不确定因素的影响。为此,本文将MPTCP子流传输性能的变化过程R建模为一个受多种因素影响的灰过程,具体如式(1)所示:

$$\begin{aligned}
 R &= C_f + U_f \\
 C_f &= \alpha Q + \beta T + \gamma L + \delta S \\
 U_f &= \sum_{i=1}^k \varepsilon_i x_i, k=1, 2, \dots, n
 \end{aligned} \tag{1}$$

其中,  $C_f$  表示已知因素的影响部分,  $U_f$  表示未知因素的影响部分,  $Q, T, L, S$  分别表示吞吐量、传输时延、

丢包率及发送窗口,  $\alpha, \beta, \gamma, \delta$  表示各确定因素相应的权重,  $x_i$  表示影响R变化的未知因素,  $\varepsilon_i$  表示未知影响因素的权重,  $n$  为未知影响因素的个数。  $\alpha, \beta, \gamma, \delta$  值可通过建立GM(1,N)模型微分方程求得。

在无线异构网络多路传输过程中,慢启动阶段子流的FTT会呈现出几何级增长趋势,而在进入拥塞避免阶段时FTT则会大幅下降,当伴随丢包现象的发生时,FTT又将大幅上升<sup>[17]</sup>。一般而言,FTT的时间序列呈现近似几何级增长规律,同时在某些时刻存在较大波动。基于此,本文提出进一步引入马尔科夫模型以揭示系统在不同状态区间转移的内在规律,并对灰色预测模型的残差序列进行修正。

### 2.2 GMM-S自适应多路传输数据调度方案

基于GM(1,N)的FTT预测及马尔科夫优化多路数据传输调度方案主要由基于GM(1,N)预测与马尔科夫优化的子流性能评估模型以及数据分发调度2个部分组成,具体构成如图2所示。其中,实线箭头表示数据流,虚线箭头表示信息流。在STPEM中首先获取MPTCP子流中的发送窗口、丢包率、吞吐量和前向传输时延数据4组可直接获取的已知影响因素,即选取 $N=4$ ,建立GM(1,4)模型预测该子流下一时刻的FTT,之后在GM(1,4)预测基础上实施马尔科夫残差修正,即根据FTT预测值与实际值的相对误差建立马尔科夫残差修正模型对GM(1,4)预测值进行修正,以完成对子流FTT的预测,量化子流传输质量,且修正后的FTT预测值越小表示该子流的传输性能越好。随后,在DDS部分根据FTT预测值对子流进行分类,选取最短预测FTT的子流为优秀子流。在优秀子流上根据拥塞窗口大小发送足够多数量的数据包,而在普通子流上,则根据其与普通子流的预测FTT时延差,实施传输数据包发送数量的动态调整。

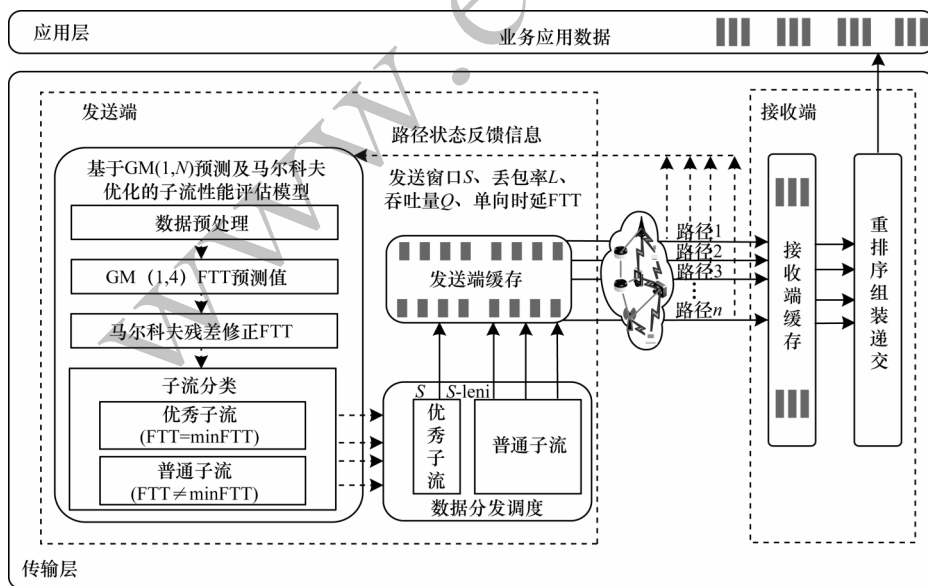


图2 GMM-S数据调度方案

Fig.2 GMM-S data scheduling scheme

2.2.1 GM(1,4)预测模型建立

本文选取子流发送窗口  $S$ 、丢包率  $L$ 、吞吐量  $Q$  和前向传输时延  $FTT$  作为原始预测样本数据:

1) 数据处理。令  $F = \{f_1, f_2, \dots, f_n\}$  表示 MPTCP 的子流集合,  $FTT_f$  为子流  $f$  的前向时延,  $S_f$  为子流  $f$  的发送窗口,  $L_f$  为子流  $f$  的丢包率,  $Q_f$  为子流  $f$  的吞吐量。假设当前为第  $t$  轮数据包发送, 以子流  $f_1$  为例, 子流数据包前向传输时间  $FTT_{f_1}$ 、发送窗口大小  $S_{f_1}$ 、子流丢包率  $L_{f_1}$  与子流吞吐量  $Q_{f_1}$  分别如式(2)~式(5)所示:

$$FTT_{f_1} = \{FTT_{f_1}(t-1), FTT_{f_1}(t-2), FTT_{f_1}(t-3), FTT_{f_1}(t-4)\} \quad (2)$$

$$S_{f_1} = \{S_{f_1}(t-1), S_{f_1}(t-2), S_{f_1}(t-3), S_{f_1}(t-4)\} \quad (3)$$

$$L_{f_1} = \{L_{f_1}(t-1), L_{f_1}(t-2), L_{f_1}(t-3), L_{f_1}(t-4)\} \quad (4)$$

$$Q_{f_1} = \{Q_{f_1}(t-1), Q_{f_1}(t-2), Q_{f_1}(t-3), Q_{f_1}(t-4)\} \quad (5)$$

将上述4组数据作为原始样本生成4个数列, 原始样本数据矩阵  $X^{(0)}$  为:

$$X^{(0)} = \begin{bmatrix} FTT^{(0)}(1) & S^{(0)}(1) & L^{(0)}(1) & Q^{(0)}(1) \\ FTT^{(0)}(2) & S^{(0)}(2) & L^{(0)}(2) & Q^{(0)}(2) \\ FTT^{(0)}(3) & S^{(0)}(3) & L^{(0)}(3) & Q^{(0)}(3) \\ FTT^{(0)}(4) & S^{(0)}(4) & L^{(0)}(4) & Q^{(0)}(4) \end{bmatrix} \quad (6)$$

将  $X^{(0)}$  通过一阶累加运算来增强原始数列的规律性, 弱化原始数据列的随机性, 得到更有规律的数据矩阵  $X^{(1)}$ :

$$X^{(1)} = \begin{bmatrix} FTT^{(1)}(1) & S^{(1)}(1) & L^{(1)}(1) & Q^{(1)}(1) \\ FTT^{(1)}(2) & S^{(1)}(2) & L^{(1)}(2) & Q^{(1)}(2) \\ FTT^{(1)}(3) & S^{(1)}(3) & L^{(1)}(3) & Q^{(1)}(3) \\ FTT^{(1)}(4) & S^{(1)}(4) & L^{(1)}(4) & Q^{(1)}(4) \end{bmatrix} \quad (7)$$

将  $FTT$  的一阶累加矩阵  $FTT^{(1)}$  进行紧邻均值计算, 生成紧邻均值矩阵  $Z_{FTT}^{(1)}$ :

$$Z_{FTT}^{(1)} = \{Z_{FTT}^{(1)}(2), Z_{FTT}^{(1)}(3), Z_{FTT}^{(1)}(4)\} \quad (8)$$

2) 模型建立。描述  $FTT$  多元一阶线性动态微分方程模型的灰微分方程如下所示:

$$FTT^{(0)}(k) + aZ_{FTT}^{(1)}(k) = b_2S^{(1)}(k) + b_3L^{(1)}(k) + b_4Q^{(1)}(k) \quad (k=2, 3, 4) \quad (9)$$

其中,  $FTT^{(0)}(k)$  表示关键影响因素  $FTT$  的原始数据, 称为灰导数,  $Z_{FTT}^{(1)}(k)$  为  $FTT$  一阶累加数据的紧邻均值, 称为背景值,  $a, b_i (i=2, 3, 4)$  表示参数, 且可采用最小二乘法求出待定系数  $a$  和  $b_i$  的值。

对于  $GM(1,4)$  的灰微分方程, 若将  $X^{(1)}(k)$  视为连续变量, 则  $X^{(1)}(k)$  为关于时间  $t$  的函数,  $GM(1,4)$

的白化微分方程定义如下:

$$\frac{dFTT^{(1)}}{dt} + aFTT^{(1)} = b_2S^{(1)}(t) + b_3L^{(1)}(t) + b_4Q^{(1)}(t) \quad (10)$$

对式(10)进行求解, 可得  $GM(1,4)$  的白化微分方程的时间响应函数如下所示  $\widehat{FTT}^{(1)}$ :

$$\widehat{FTT}^{(1)}(t+1) = \left[ FTT^{(0)}(1) - \frac{1}{a}(\widehat{b}_1\widehat{S}^{(1)}(t+1) + \widehat{b}_2\widehat{L}^{(1)}(t+1) + \widehat{b}_3\widehat{Q}^{(1)}(t+1)) \right] e^{-at} + \frac{1}{a}(\widehat{b}_1S^{(1)}(t+1) + \widehat{b}_2L^{(1)}(t+1) + \widehat{b}_3Q^{(1)}(t+1)) \quad (11)$$

累减还原得序列  $FTT^{(0)}$  的预测值:

$$FTT^{(0)}(t+1) = \widehat{FTT}^{(1)}(t+1) - \widehat{FTT}^{(1)}(t) \quad (12)$$

基于  $GM(1,4)$  的  $FTT$  预测算法描述如算法1所示。其中, 步骤1~步骤4是对数据的收集与预处理, 步骤5~步骤8对预处理后的数据建立  $GM(1,4)$  模型并进行  $FTT$  预测。

算法1 基于  $GM(1,4)$  的  $FTT$  预测

1. Data\_collection(  $FTT_f, S_f, L_f, Q_f$  ); //收集  $t$  时刻之前的 //4 组子流传输信息
2.  $X^{(0)} = FTT_f + S_f + L_f + Q_f$
3.  $X^{(1)} = \text{First\_Order\_Accumulative}(X^{(0)})$  //建立  $X^{(0)}$  的一 //阶累加数据矩阵  $X^{(1)}$
4.  $Z_{FTT}^{(1)}(t) = \frac{1}{2}(FTT^{(1)}(t) + FTT^{(1)}(t-1))$  //建立  $FTT$  的一 //阶累加数列的紧邻均值数列  $Z_{FTT}^{(1)}$
5. Build\_FTT\_GM(  $Z_{FTT}^{(1)}, X^{(1)}$  ) //根据式(9)建立  $FTT$  灰 //色微分方程
6.  $a, b_i = \text{Get\_FTT\_GM}(Z_{FTT}^{(1)}, X^{(1)})$  //求得  $FTT$  影响因素 //的权重  $a, b_i (i=2, 3, 4)$
7.  $\widehat{FTT}^{(1)}(t+1) = \text{Get\_FTT}(a, b_i, X^{(0)}, X^{(1)})$  //得到  $t+1$  时 //刻的  $FTT$  一阶累加数列
8.  $\widehat{FTT}^{(0)}(t+1) = \widehat{FTT}^{(1)}(t+1) - \widehat{FTT}^{(1)}(t)$

2.2.2 马尔科夫残差修正

$GM(1,4)$  模型预测所得  $\widehat{FTT}$  与实际值相对残差  $Diff$  的计算方法如式(13)所示, 根据残差  $Diff$  的大小进行如表1所示的划分。

$$Diff = \frac{FTT - \widehat{FTT}}{\widehat{FTT}} \quad (13)$$

表1 残差状态划分结果

Table 1 Residual state partition results

状态	含义	范围
E1	大	$(-\infty, -300\ 000]$
E2	偏大	$(-300\ 000, -150\ 000]$
E3	基本准确	$(-150\ 000, 150\ 000]$
E4	偏小	$(150\ 000, 300\ 000]$
E5	小	$(300\ 000, +\infty)$

1) 建立状态转移概率矩阵。由状态  $E_i$  转移到状态  $E_j$  的次数为  $n_{i \rightarrow j}$ , 以状态  $E_i$  为起点转向另一个状态的次数为  $N_i$ , 则状态  $E_i$  转移到状态  $E_j$  的状态转移概率为:

$$p_{i \rightarrow j} = \begin{cases} \frac{n_{i \rightarrow j}}{N_i}, N_i > 0 \\ 0, N_i = 0 \end{cases} \quad (14)$$

根据式(14)构建状态转移概率矩阵  $\mathbf{P}$  为:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{15} \\ p_{21} & p_{22} & \cdots & p_{25} \\ \vdots & \vdots & & \vdots \\ p_{51} & p_{52} & \cdots & p_{55} \end{bmatrix} \quad (15)$$

2) 计算预测值。建立状态转移概率矩阵  $\mathbf{P}$  后, 假设  $t$  时刻对象处于  $E_i$  状态, 若  $\mathbf{P}$  中的第  $i$  行满足  $\max P_i = p_{ij}$ , 则认为下一时刻  $E_i$  最有可能转移到  $E_j$  状态, 并取  $E_j$  状态区间  $[\theta_j, \theta_{j+1}]$  的中位数作为  $n+1$  时刻的预测值。

根据状态转移矩阵  $\mathbf{P}$  对  $\overline{\text{FTT}}$  进行修正, 并获得最终预测值:

$$\text{FTT}'(t+1) = \overline{\text{FTT}}(t+1) \times \left(1 + \frac{1}{2} E_{p_{\max}}\right) \quad (16)$$

其中,  $E_{p_{\max}}$  表示在状态转移矩阵中,  $t$  时刻状态  $E_i$  根据最大转移概率  $\max P_i$  转移到状态  $E_j$  的误差区间步长。

依次计算所有子流的前向时延预测值  $\text{FTT}'_F \{ \text{FTT}'_{f_1}, \text{FTT}'_{f_2}, \dots, \text{FTT}'_{f_n} \}$ , 即为 GM-Markov 模型预测所得的子流前向时延。

算法2给出了 GM(1,4) FTT 预测值马尔科夫修正算法的描述, 其中步骤1与步骤2负责建立状态转移矩阵, 步骤3~步骤7通过状态转移矩阵修正灰色模型 FTT 预测值, 步骤8对状态转移矩阵进行更新。

**算法2** GM(1,4) FTT 预测值马尔科夫修正

Update\_State\_Transition\_Matrix(FTT,  $\overline{\text{FTT}}$ ) //更新状态转移概率矩阵  $\mathbf{P}$  的函数

$$1. R = \frac{\text{FTT} - \overline{\text{FTT}}}{\overline{\text{FTT}}};$$

2.  $\mathbf{P} = \text{State\_Division}(R, E)$ ; //建立状态转移矩阵,  $E$  为残差  
//状态划分;  $\text{GMFTT\_Correction}(\mathbf{P}, \text{FTT}(t), \overline{\text{FTT}}(t+1))$  //修正 GM(1,4) FTT 预测值

3. while  $\text{FTT}(t) = E_i$ , do

4. find  $\max P_i$

5. if  $\max P_i = p_{i\max}$  then

$$6. \text{FTT}'(t+1) = \overline{\text{FTT}}(t+1) \times \left(1 + \frac{1}{2} E_{p_{\max}}\right)$$

7. end if

8. Update\_State\_Transition\_Matrix(FTT,  $\overline{\text{FTT}}$ )

2.2.3 基于  $\text{FTT}'_F$  的数据包分发算法

为减少缓存阻塞带来的负面影响, 本文在子流

上通过前向传输时延预测值评估该子流的性能, 并动态设定传输数据量。根据 GM-Markov 模型所求得的前向传输时延预测值  $\text{FTT}'_F$  对子流进行分类。将最小  $\text{FTT}'_{\min}$  的路径选作优秀子流, 记作  $f_i$ , 其他子流标记为普通子流。对于优秀子流  $f_i$ , 数据包分发方法<sup>[18]</sup>为:

$$S_i = \min(\text{cwnd}_i, C) \quad (17)$$

其中,  $\text{cwnd}_i$  表示子流  $f_i$  的拥塞窗口,  $C$  为发送缓冲区的待发送数据包。

假设子流  $f_j$  为普通子流, 其前向传输时延预测值为  $\text{FTT}'_j$ 。在  $\text{FTT}'_{\min}$  时间内普通子流较优秀子流少传输的数据量  $\text{LEN}_j$  为:

$$\text{LEN}_j = (\text{FTT}'_j - \text{FTT}'_{\min}) \times Q_j \quad (18)$$

为使数据包按序到达接收端, 减小传输时间差异, 则定义为普通子流并按下式发送数据:

$$S_j = \min(\text{cwnd}_j - \text{LEN}_j, C - \text{LEN}_j) \quad (19)$$

若  $S_j \leq 0$ , 则该子流不传输数据。这样的发送窗口大小动态调节方式, 可使子流前向传输时延趋近于最优子流前向时延, 从而减小路径间传输时延差异, 解决由传输时延差异带来的数据包乱序、缓存阻塞及传输性能下降等问题。

算法3给出了 GMM-S 数据包分发算法。其中, 步骤1先获取  $\text{FTT}'_F$  预测值, 步骤2~步骤5实现在优秀子流上发送数据包, 步骤6~步骤13实现在普通子流上数据包的发送。

**算法3** GMM-S 数据包分发

1. Get( $\text{FTT}'_F$ ) //获取  $\text{FTT}'_F$  预测值,  $\text{FTT}'_F$

2. if  $\text{FTT}'_i = \min \text{FTT}'_F$  then

3.  $\text{flow}_i = \text{best\_subflow}$

4.  $\text{data} = \min(\text{cwnd}_i, C)$

5. Send\_data(best\_subflow, data) //在优秀子流上发送数据

6. else

7.  $\text{flow}_i = \text{normal\_subflow}$

8.  $\text{data} = \min(\text{cwnd}_i - \text{LEN}_i, C - \text{LEN}_i)$

9. if  $\text{data} < \text{MSS}$  then

10.  $\text{data} = 0$

11. end if

12. Send\_data(normal\_subflow, data) //在普通子流上发送

//数据

13. end if

## 3 实验设置与分析

### 3.1 仿真环境设置

仿真实验环境基于 Ubuntu16.04, 64 位操作系统, 使用 NS 工具建立网络拓扑, NS 版本为 3.29 并添加了 MPTCP 模块<sup>[19]</sup>。网络测试拓扑如图3所示。

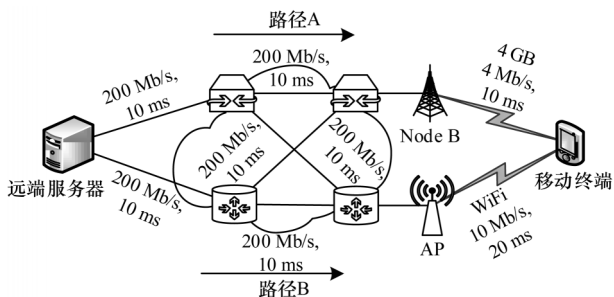


图 3 异构无线网络测试拓扑

Fig.3 Heterogeneous wireless network test topology

为模拟真实无线环境,在每条路径上设置分组丢失模型来模拟分布式分组丢失。路径 A 使用 LTE 蜂窝网络,设置 4 Mb/s 的接入带宽。考虑到 LTE 在大范围内相对稳定<sup>[20]</sup>,丢包率设置为固定值 0.01。路径 B 使用公共 WiFi 网络,设置 10 Mb/s 的接入带宽,路径丢包率设置为 0~1.0。无线异构网络拓扑的具体参数配置如表 2 所示。实验在 NS-3.29 仿真平台下,以传输数据量、重传次数、乱序块大小/个数与吞吐量等参数作为主要指标,同时选取 RR 算法及 LowRTT 算法作为传输性能的比较对象。

表 2 异构无线网络拓扑的参数配置

Table 2 Parameter configuration of heterogeneous wireless network topology

参数	路径 A	路径 B
无线接入技术	4G 网络	IEEE802.11b
接入带宽/(Mb·s <sup>-1</sup> )	4	10
接入链路时延/ms	10	20
默认丢包率	0.01	0.01
丢包率变化	固定	0~1.0
核心网带宽/(Mb·s <sup>-1</sup> )	200	200
核心网时延/ms	10	10

### 3.2 实验结果及分析

表 3 给出了 GMM-S、RR 及 LowRTT 三种调度算法的传输时间、传输数据总量、传输次数、重传次数、重传占比及最大重传数据段等数据。

表 3 3 种数据调度算法的数据传输结果

Table 3 Data transmission results of three data scheduling algorithms

数据调度算法	传输时间/s	传输数据总量	传输次数	重传次数	重传占比/%	最大重传数据段
MM-S	100	8 296 000	8 430	2 232	26.68	20 000
RR	100	8 254 000	8 302	2 840	34.21	28 000
LowRTT	100	8 217 000	11 245	4 369	38.85	30 169

相较于 RR 和 LowRTT 算法,本文提出的 GMM-S 算法在获得更高吞吐量的同时有效减少了重传次数。这主要是因为 GMM-S 基于 GM(1,N)和马尔科夫优化的子流性能评估模型更有效地完成路径质量的准确评价,实现子流数据包发送的自适应动态调整。

#### 3.2.1 GM(1,N)-Markov 预测误差分析

图 4 给出本文实验设置下从第 44 次数据包传输开始后采用 GM(1,4)-Markov 算法预测的最小 FTT 子流与实际最小 FTT 子流的对比数据。在第 1 次~第 43 次数据包传输过程中,Subflow\_3 是 WiFi 链路的主子流且有较高的带宽,在不产生丢包时该子流具有最小 FTT,在 GM(1,4)-Markov 预测算法状态转移概率矩阵中,Subflow\_3 表现出最大的矩阵元素值。此后,即从第 44 次开始,GM(1,4)-Markov 算法在综合分析拥塞窗口、吞吐量及丢包情况的基础上选出具有最小 FTT 的 Subflow\_1。在第 47 次数据包传输时,GM(1,4)-Markov 算法给出了最小 FTT 子流为子流 3 的预判,但由于此时实际传输子流仍为子流 1,因此出现预测值与实际值不一致的情形,即产生误差。从第 48 次数据包传输开始至第 59 次数据包传输,实际最小 FTT 子流为 Subflow\_3。类似的误差产生情形还发生在第 67 次数据包传输。在随后的数据包传输过程中,由于 Markov 残差修正算法的作用,误差得以不断修正,GM(1,4)-Markov 算法预测精度不断提高,且总误差控制在 1.8% 左右。

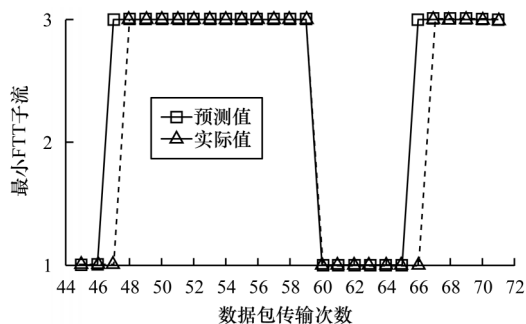


图 4 GM(1,N)-Markov 算法预测最小 FTT 子流效果

Fig.4 Effect of GM(1,N)-Markov algorithm on predicting the minimum FTT subflow

#### 3.2.2 数据包乱序分析

图 5 给出了应用 GMM-S、RR、LowRTT 三种数据调度算法在无线异构网络中传输过程中出现的数据块乱序情况。从图 5 可以看出:在 50 s~100 s 的时间间隔内,RR 算法共发生 199 次乱序,最大乱序块大

小为 27 000 bit; LowRTT 算法共发生 749 次乱序, 最大乱序块大小为 29 000 bit; 而 GMM-S 算法共发生 176 次乱序, 最大乱序块大小为 16 000 bit。总体而言, GMM-S 相比 RR 和 LowRTT 所产生的乱序块更少且乱序数据块也更小。这主要是由于 GMM-S 实现了路径质量的实时评估, 并在数据分发时根据子流预测 FTT 自适应地分发数据包, 尽可能地保障数据包按序到达所导致的, GMM-S 不仅明显减少了数据包乱序的发生次数, 同时也降低了乱序程度。

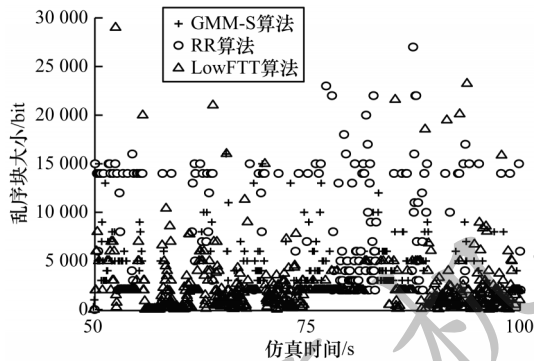
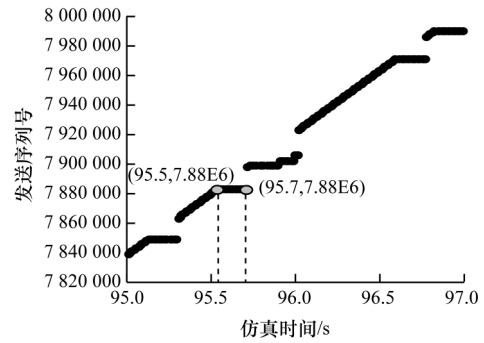


图5 3种数据调度算法的乱序到达结果对比

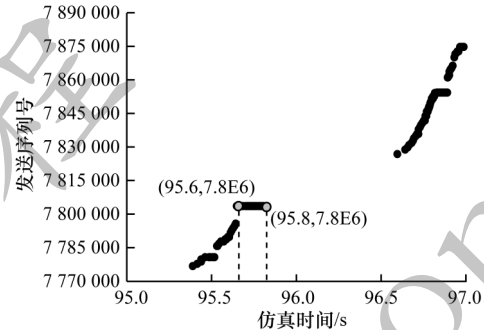
Fig.5 Comparison of out-of-order arrival results of three data scheduling algorithms

### 3.2.3 数据发送到达结果分析

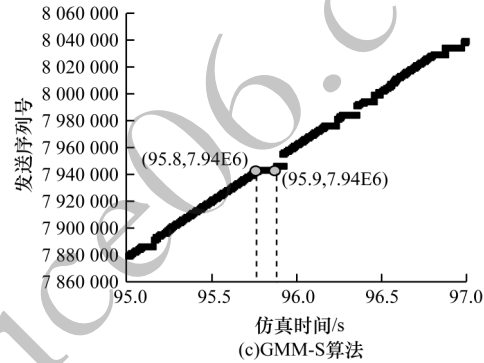
在实验中, 本文通过设置随机丢包率来模拟无线异构网络的不确定性, 在传输过程中丢包的发生会引发重传。图 6 给出了 GMM-S、RR、LowRTT 三种算法数据发送到达结果对比。在理想状态下, 按序传输数据包且接收端按序接收, TSN 随时间变化的图像应为单调平稳递增的图像, 但重传的发生会因为反复传输而使图像出现水平线段。从图 6 可以看出: RR 在 95.5 s 时发生了一次重传, 持续时间为 0.2 s; LowRTT 在 95.6 s 时发生了一次重传, 持续时间为 0.2 s; GMM-S 则是在 95.8 s 时发生了一次重传, 持续时间为 0.1 s。GMM-S 的重传持续时间最少。此外, 在数据传输过程中, 若发生乱序或发送数据包过大的情况会导致缓冲区阻塞, 致使发送端数据包停发, 在图像中表现为中断现象。在图 6 中, 95 s~97 s 的时间段内, RR 产生了 4 次明显中断, LowRTT 产生了 5 次明显中断, 而 GMM-S 仅产生 2 次明显中断, 且中断时间远小于 RR 和 LowRTT, 这表明 GMM-S 工作时接收端数据接收更为平稳。



(a)RR算法



(b)LowRTT算法



(c)GMM-S算法

图6 3种数据调度算法的数据发送到达结果对比

Fig.6 Comparison of data transmission and arrival results of three data scheduling algorithms

### 3.2.4 吞吐量

有效吞吐量 (Effective Throughput, ET)<sup>[21]</sup> 定义为单位时间  $t$  内由传输层送达应用层的数据量 (Application layer Data, AD), 即:

$$ET = \frac{AD}{t} \quad (20)$$

在图 7 中, RR 的有效吞吐量波动区间约为 0.1, LowRTT 有效吞吐量波动区间约为 0.15, 而 GMM-S 波动区间仅为 0.01, 且持续稳定在 10.06 Kb/s~10.07 Kb/s 之间。在 95 s~96 s 时间段内 RR、LowRTT 及 GMM-S 均发生了重传, 有效吞吐量发生了下降, 但由于 GMM-S 的自适应数据包分发机制会在产生丢包的子流上动态分配较少的数据包, 因此 GMM-S 的有效吞吐量在重传发生时未受到严重影响。

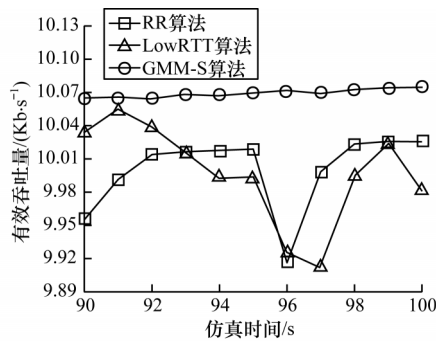


图 7 3 种数据调度算法的有效吞吐量对比

Fig.7 Comparison of effective throughput variation of three data scheduling algorithms

### 3.2.5 服务质量评估

在实际应用场景中,较大的端到端时延会严重影响实时传输的服务质量。比如在实时视频传输服务中,若因为发送率波动较大引起应用层数据接收时间间隔增大,用户会感受到明显的视频卡顿或中断。图 8 给出了 90 s~100 s 时间段内数据发送速率的变化情况。从图 8 可以看出,LowRTT 在 97 s 时发送速率存在大幅增加的现象,这是因为在 95 s~96 s 期间发生了数据包乱序现象,在 97 s 时缓冲区中的乱序数据包顺序交付给应用层,此时的拥塞窗口增大,发送窗口因而变大,LowRTT 的发送速率出现明显上升。而在 GMM-S 中,每条子流维护各自的拥塞窗口,数据分发调度综合考虑子流的传输性能为子流分发合理大小的数据包,因此不会出现 LowRTT 发送速率大幅波动的情形。

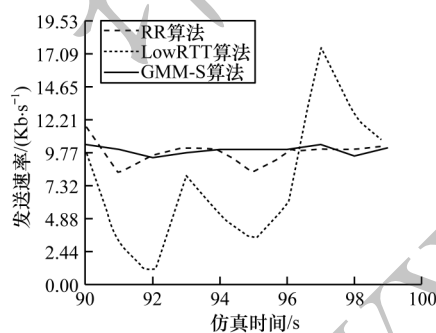


图 8 3 种数据调度算法的发送速率对比

Fig.8 Comparison of transmission rate of three data scheduling algorithms

图 9 给出了应用层数据接收时间间隔的分布数据,应用层接收时间间隔分布以平均值±95%的置信区间来反映。从图 9 可以看出:RR 的应用层数据接收时间间隔主要分布在 1.18E7 ns~1.2E7 ns 之间;LowRTT 的应用层数据接收时间间隔主要分布在 1.16E7 ns~1.18E7 ns 之间;而 GMM-S 的应用层数据接收时间间隔则主要分布在 1.11E7 ns~1.13E7 ns 之间,范围明显小于 RR 和 LowRTT。这表明 GMM-S 可提供更稳定的应用层数据传送,为网络传输服务质量及良好的用户

体验提供保障。

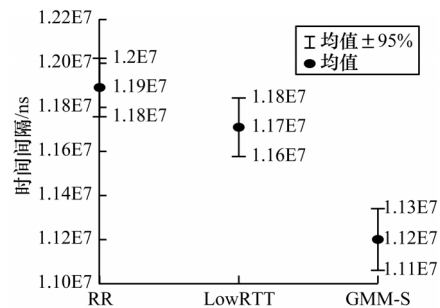


图 9 应用层数据接收时间间隔分布

Fig.9 Distribution of application layer data receiving interval

## 4 结束语

本文提出一种基于 GM(1, N) 前向传输时延预测与马尔科夫优化的自适应多路传输数据调度算法 GMM-S, 该算法综合考虑子流中吞吐量、丢包率及发送窗口等因素对传输时延的影响, 采用 FTT 预测对子流性能进行评估, 并通过调整相应发送窗口大小来最小化异构网络中的传输时延差异。实验结果表明, 该算法可有效解决数据包乱序问题并显著提升网络传输性能。下一步将采用个性化数据传输算法对多路传输数据调度进行优化, 进一步提高网络资源利用率与网络服务质量。

### 参考文献

- [ 1 ] ZHANG Tingguang, ZHAO Shuai, REN Bingfei, et al. Performance enhancement of multipath TCP in mobile Ad Hoc networks [C]//Proceedings of the 25th IEEE International Conference on Network Protocols. Washington D. C. , USA: IEEE Press, 2017: 1-2.
- [ 2 ] MENA J, BANKOLE P, GERLA M. Multipath TCP on a VANET: a performance study [C]//Proceedings of 2017 ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems. New York, USA: ACM Press, 2017: 39-40.
- [ 3 ] PER H, KARL-JOHAN G, ANNA B, et al. Low-latency scheduling in MPTCP [J]. IEEE/ACM Transactions on Networking, 2019, 27(1): 302-315
- [ 4 ] XUE Kaiping, HAN Jiangping, DAN Ni, et al. DPSAF: forward prediction based dynamic packet scheduling and adjusting with feedback for multipath TCP in lossy heterogeneous networks [J]. IEEE Transactions on Vehicular Technology, 2018, 67(2): 1521-1534.
- [ 5 ] FORD A C, RAICIU C, HANDLEY M, et al. Bonaventure, TCP extensions for multipath operation with multiple addresses[EB/OL]. [2019-11-02]. <https://www.ietf.org/rfc/rfc6824.txt.pdf>.
- [ 6 ] GAO Kai, XU Changqiao, QIN Jiuren, et al. Qos-driven path selection for MPTCP: a scalable SDN-assisted approach [C]// Proceedings of 2019 IEEE Wireless Communications and Networking Conference. Washington D. C. , USA: IEEE

- Press, 2019:1-8.
- [ 7 ] Multipath TCP-Linux kernel implementation[EB/OL]. [2019-11-02]. <http://multipath-tcp.org/>.
- [ 8 ] BONG-HWAN O, LEE J Y. Feedback-based path failure detection and buffer blocking protection for MPTCP[J]. IEEE/ACM Transactions on Networking, 2016,24(6):1-12.
- [ 9 ] PAASCH C, FERLIN S, ALAY O, et al. Experimental evaluation of multipath TCP schedulers[C]//Proceedings of 2014 ACM SIGCOMM Workshop on Capacity Sharing Workshop. New York, USA; ACM Press, 2014:27-32.
- [10] FROMMGEN A, ERBSHAUSSER T, BUCHANN A, et al. ReMP TCP; low latency multipath TCP[C]//Proceedings of 2016 IEEE International Conference on Communications. Washington D. C. , USA; IEEE Press, 2016:1-7.
- [11] DENG Julong. Basic method of grey system[M]. Wuhan; Huazhong Institute of Technology Press, 1987. (in Chinese) 邓聚龙. 灰色系统基本方法[M]. 武汉:华中工学院出版社, 1987.
- [12] LI Li. Prediction research based on AHP and grey GM (1, N) model[J]. Knowledge Economy, 2012(19): 13, 16. (in Chinese) 李礼. 基于 AHP 方法和灰色 GM(1, N)模型的预测研究[J]. 知识经济, 2012(19):13, 16.
- [13] ZHAO Dan, ZHANG Tianju, ZANG Hongfei. Prediction of groundwater depth the based on grey Markvo model[J]. Yellow River, 2015, 37(4):69-72. (in Chinese) 赵丹, 张天菊, 臧红飞. 基于灰色马尔可夫链的兰村泉域地下水位预测[J]. 人民黄河, 2015, 37(4):69-72.
- [14] YEDUGUNDLA K, SIMONE F, DREIBHOLZ T, et al. Is multi-path transport suitable for latency sensitive traffic[J]. Computer Networks, 2016, 105(4): 1-21.
- [15] JIANG Zhuo, WU Qian, LI Hewu. Review on cross-layer optimization of end-to-end multipath transmission over the Internet[J]. Journal of Software, 2019, 30(2): 302-322. (in Chinese) 江卓, 吴茜, 李贺武. 互联网端到端多路径传输跨层优化研究综述[J]. 软件学报, 2019, 30(2):302-322.
- [16] DONG P P, YANG W J, TANG W S, et al. Reducing transport latency for short flows with multipath TCP[J]. Journal of Network and Computer Applications, 2018, 108:20-36.
- [17] WANG Wei, WANG Xiaoqiang, WANG Dongyu. Energy efficient congestion control for multipath TCP in heterogeneous networks[J]. IEEE Access, 2018, 6:2889-2898.
- [18] PENG Q Y, WALID A, HWANG J, et al. Multipath TCP: analysis, design, and implementation [J]. IEEE/ACM Transactions on Networking, 2016, 24(1):596-609.
- [19] COUDRON M, SECCI S. An implementation of multipath TCP in ns3[J]. Computer Networks, 2017, 116:1-11.
- [20] PARTOV B, LEITH D J. Experimental evaluation of multipath schedulers for LTE/WIFI devices[C]//Proceedings of the 10th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization. Washington D. C. , USA; IEEE Press, 2016:41-48.
- [21] TAO Yang, HUANG Peng. Goodput improvement for MPTCP based on controlling difference of delay [J]. Application Research of Computers, 2015, 32(9):2753-2756. (in Chinese) 陶洋, 黄鹏. 基于时延差控制的MPTCP有效吞吐量提高方法[J]. 计算机应用研究, 2015, 32(9):2753-2756.

编辑 刘继娟