doi:10.3969/j.issn.1000-436x.2012.z2.033

多路径 TCP 拥塞控制算法研究

刘佩[1,2]，任勇毛[1]，李俊[1]
（1.中国科学院 计算机网络信息中心，北京 100190; 2.中国科学院大学，北京 100190）

摘 要：首先介绍了多路径传输协议的几种典型的拥塞控制算法，然后对 MPTCP 协议进行了理论分析，包括 MPTCP 拥塞控制算法在瓶颈链路 TCP 公平性、平衡拥塞能力以及 flap 现象，实验分析表明 LINKED INCREASES 算法效果最好。最后，指出其存在的问题，并指出了进一步的研究方向。

关键词：拥塞控制；多路径 TCP；综述；NS2
中图分类号：TP393.01 **文献标识码**：A **文章编号**：1000-436X(2012)Z2-0233-06

Survey on multipath TCP congestion control

LIU Pei[1,2], REN Yong-mao[1], LI Jun[1]
(1. Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China;
2. University of Chinese Academy of Sciences, Beijing 100190, China)

Abstract: First， several congestion control algorithms of multipath TCP recently had been introduced and discussed. Next, some simulation experiments in NS2 was did, the performance of MPTCP congestion control algorithm in TCP fairness of the bottleneck link, resource pooling and flapness was analyzed. At last, some problems existed and possible research directions in the future was pointed out.

Key words: congestion control; multipath TCP; survey; NS2

1 引言

移动终端多信号（3G, Wi-Fi 等）与各种新的云计算数据中心体系结构(如 FatTree，Bcube)的提出，使得一个用户终端可以通过多条路径到达一个目标节点。当前互联网体系结构仍然沿用二十多年前传统的 TCP/IP 协议架构，端到端的 TCP 连接只能选择其中一条“最佳路径”，一旦发生链路故障，TCP 连接就可能中断[1,2]，发送端与目的端的多路径并没有充分利用。多路径传输协议[3]的提出是为了克服传统 TCP 在新的体系结构中传输性能低下的弱点。而 MPTCP 在 TCP 的基础上旨在通过资源共享的方式，将数据流分发到多条链路上，从而提高链路利用率。同时，通过多个端到端的连接来提高其顽健性。

1987 年，Van Jacobson 在客户端依据探测到的分组丢失率来创造了加法增长乘法减少的拥塞控制算法，依据此算法调整拥塞窗口。这种算法在单路径 TCP 中性能表现很好，但是这种适用于单路径 TCP 的拥塞控制算法并不适用于多路径 TCP。

早在 1995 年，Christian Huitema 就提出了多路径传输的概念。随后研究人员陆续设计出了一些多路径 TCP 的方案，如 Parallel TCP (pTCP, 2002)[3]，mTCP (2004)等，试图将多路径并行传输能力引入

收稿日期：2012-10-23
基金项目：国家重点基础研究发展计划（“973”计划）基金资助项目（2012CB315803）；中国科学院知识创新工程青年人才领域基金资助项目（CNIC_QN_1203）
Foundation Items: The National Basic Research Program of China (973 Program)(2012CB315803); The Knowledge Innovation Program of the Chinese Academy of Sciences (CNIC_QN_1203)

到 TCP 协议当中。然而这些方案提出各子流之间进行独立拥塞控制，导致对传统 TCP 的不公平性，致使其发展缓慢，因此没有得到大范围的推广与应用。近年，多路径 TCP 成为一个新的研究热点。现在的多路径 TCP 逐渐采用了由 Kelly、Voice[5]等相继提出的联合拥塞控制。联合拥塞控制采用了多种实现机制，文献[4]对其中拥塞控制算法理论做了详细的分析，文献[5]针对其中的算法也做出了相应的分析。本文在其基础上，对多路径 TCP 在瓶颈链路模型下的 TCP 公平性进行分析，不仅考虑了增大 TCP 数目，还考虑了增大多路径 TCP 子流的影响。同时，对多路径 TCP 拥塞控制算法的平衡拥塞能力进行评估。实验结果表明，多路径 TCP 相对于独立控制的拥塞控制算法有很大的提高，但仍然存在着一些问题。

2 多路径 TCP 拥塞控制协议

2.1 多路径拥塞控制协议的目标

拥塞控制算法是传输层架构的核心，良好的拥塞控制算法才能保证网络资源的利用率最大化。良好的拥塞控制算法更是多路径传输协议成功的关键，故多路径拥塞控制协议设定了以下目标[6~8]。

1) 一个多路径流获得的吞吐量必须不小于一个单路径 TCP 流在最好路径上传输的吞吐量，来保证实施多路径 TCP 的必要性。

2) 一个多路径 TCP 流必须保证在任何一条路径或所有路径的的链路容量占有率高于单路径 TCP 流在最好的路径上的占有率，来保证多路径协议多传统 TCP 的公平性。

3) 多路径协议子流应当选择轻度拥塞的路径进行传输，来保证多路径 TCP 的均衡负载，即保证 resource pooling。

传统的 TCP 基于窗口的拥塞算法是：当没有丢失的时候加法增长，当检测到丢失时乘法减少。具体表现为：对于路径上的 ACK，以 $1/w$ 来增加 w 的窗口；对于路径上的分组丢失，以 $w/2$ 来减少窗口。

2.2 早期多路径 TCP 拥塞算法

MPTCP 不是最早提出的多路径 TCP，之前提出的 pTCP、cTCP 等旨在解决端到端 TCP 只进行一条最优链路传输效率低下的问题。

pTCP(parallel TCP) 是由 Hung-Yun Hsieh 和 Raghupathy Sivakumar 在文献[9]中率先提出了一种端到端的传输层多路传输控制协议。它主要由 2 个部分组成:一个是负责给各子流分配数据和聚合带宽的功能模块 SM（striped connection manager）；另一个是彼此负责独立传输数据的子流模块 TCP-v。TCP-v 在拥塞控制上面各子流采用传统的 TCP 拥塞控制算法。这就导致在多个流并发的时候，TCP-v 具有很强的侵占性，对单路径 TCP 的公平性很差。

mTCP 旨在解决如何更有效地利用多条路径，所以每条子流采取独立的拥塞控制算法，原则上互不影响，但是当有子流与单路径 TCP 共享链路时，将会获得比单路径 TCP 更多的带宽。为了解决此问题，mTCP 对子连接进行检测，一旦发现共享链路时，将会抑制分组丢失率高和网络性能差的路径，以求达到对 TCP 的公平性。

cTCP（concurrent TCP）[10]是一种在传输层实现多路径负载均衡(MPLB, multi-path load balancing)的方案。cTCP 采用了单窗口机制，各子连接采用联合式增加、独立式减少，但是缺乏理论依据。

综上分析，早期提出的多路径 TCP 在拥塞控制算法方面都是采取了子连接独立控制的方式，虽然采取一定的措施去抑制瓶颈链路对 TCP 的公平性，但是效果不明显，所以实施意义不大，近年来一直发展缓慢。

2.3 MPTCP 拥塞控制算法

早期提出的多路径 TCP 算法一般采取独立的拥塞控制机制，这导致其在 TCP 公平性上表现很差。IETF 从 2009 年开始推动一个关于多路径 TCP 的规范，目前通过了一些草案，在拥塞控制算法方面已经有了很大的改进。

多路径 TCP 每个子流通过设置权重来控制公平性，对此提出了 EWTCP(equal weighted TCP) 算法，EWTCP 克服各子流独立增长带来的侵占性，对每个子流的拥塞窗口增长提供了参数限制，保证其不过多地抢占其他单路径 TCP。其采用算法：

对于 path r 上的 ACK，以 α/W_r 增加 W_r 的窗口；

对于 path r 上的分组丢失，以 $W_r/2$ 来减少窗口。

其中，W_r 指的是窗口大小，$\alpha=1/\sqrt{n}$，n 是路径的数目。

该算法有效地实现了 TCP 公平性，保证了目标 2），但是它给每条链路分配相同的权重，不能有效地将流量转移到轻度拥塞的路径之上。此外，该算

法是基于路径上的 RTT 相等，当 RTT 不相等时，该算法没有提供相应调整，使得性能表现很差，在第 3 节将有实验验证。

为了更有效地选择路径，满足多路径 TCP 提出的目标。产生了 COUPLED 算法：

对于 path r 上的 ACK，以 $1/W_{\text{total}}$ 增加 W_r 的窗口；

对于 path r 上的分组丢失，以 $W_{\text{total}}/2$ 来减少窗口。

其中，W_{total} 是所有子流的窗口大小之和。

COUPLED 算法采用了从总体链路去控制整个拥塞算法，各个子流的拥塞控制耦合度很高，且当链路 RTT 不相等时，该算法总是选择分组丢失率最小的路径，当分组丢失率小的路径不是拥塞程度最低的路径时，这与目标 3）相背离。

MPTCP 的后 2 个目标是要保证在传输过程中对单路径 TCP 的公平性，同时保证各个子流之间拥塞窗口的增大更倾向于轻度拥塞的子流。

MPTCP 为了满足设定目标，充分考虑了 RTT 不相等的情况而提出了 LINKED INCREASES 算法。

对于 path r 上的每次 ACK，以 $\min(a/W_{\text{total}}, 1/W_r)$ 来增加 W_r 的窗口。

对于 path r 的每次分组丢失，以 $W_r/2$ 来减少窗口。

其中，$a = \bar{W}_{\text{total}} * \dfrac{\max_r W_r / RTT_r^2}{(\sum W_r / RTT_r)^2}$，$a$ 用来确定不同子流根据链路状况获得的窗口增长大小，$1/W$ 是为了保证拥塞窗口增长的上限[6]，用来保证 TCP 公平性。

综上分析，多路径传输协议在拥塞控制算法采用独立控制时，具有很强的侵占性，导致对传统 TCP 的不公平，早期的多路径控制算法出现了这些问题；MPTCP 采用联合拥塞控制算法，并充分考虑每个路径的 RTT 不相等的情况，使得拥塞控制有了很大的提升。

3　多路径 TCP 仿真实验分析

本文实验环境采用 Ubuntu 11.10，NS2 3.14，以及 google 提供的 MPTCP 扩展分组。在实验分析中，首先模拟了多路径 TCP 与单路径 TCP 共享瓶颈链路的情景，验证 MPTCP 拥塞控制算法对 TCP 公平性；其次分析了其在各子流之间平衡负载的性能，最后，分析了 MPTCP 拥塞窗口在多条路径拥塞程度相同或相近时 flap 表现（拥塞窗口会在 2 条路径上来回跳动）。

3.1　对于共享链路 TCP 公平性分析

EWTCP 算法假设所有路径的 RTT 是相等的，在这种情况下，每个子流获得的权重的是相等的，所以每个子流的拥塞窗口理论上为 W_{total}/N，每个 EWTCP 连接与 TCP 连接获得的吞吐量是相等的，但在实际网络中，每条路径上的 RTT 是不一致的，这就造成该算法出现瓶颈。当通过同一瓶颈的 TCP 的流数目增多时，EWTCP 算法并不能保证 TCP 的公平性，会大量抢占 TCP 的瓶颈链路占有率。

本文首先模拟测试了 EWTCP 与 LINKED INCREASES 算法在瓶颈路径（图 1 所示）对 TCP 的公平性，实验结果如图 2 所示。

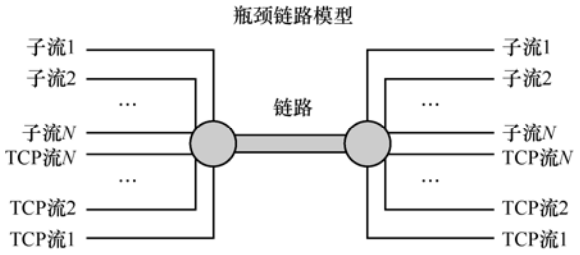


图 1　瓶颈路径 TCP 公平性测试模型

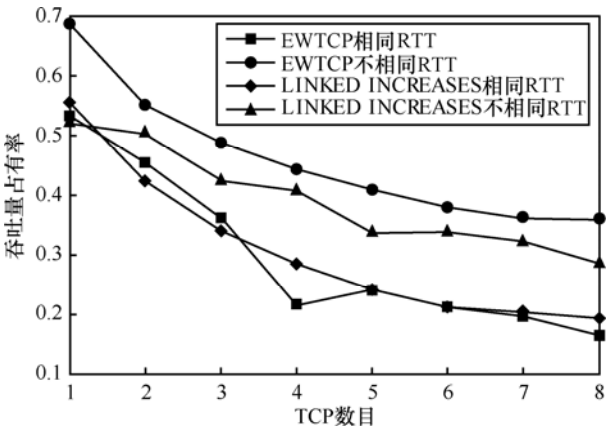


图 2　瓶颈链路的吞吐量占有率比较

由图 2 可以看出，在 RTT 相等时，EWTCP 与 LINKED INCREASES 算法一样保证了对单路径 TCP 的公平性，随着单路径 TCP 数目的增多，吞吐量占有率下降；当 RTT 不相等时，EWTCP 性能很差，明显抢占了其他单路径 TCP 的流量，表现出极大的不公平性。LINKED INCREASES 由于考虑了 RTT 对于算法的影响，所以性能要优于 EWTCP。

同时考察了 MPTCP 算法子流数目对瓶颈链

路 TCP 公平性的影响。发现当子流数目加倍时，LINKED INCREASES 吞吐量占有率并没有加倍，保持在 50%左右，这在一定程度上保持 TCP 的公平性，而 EWTCP 则表现了较差的性能，如图 3 所示。

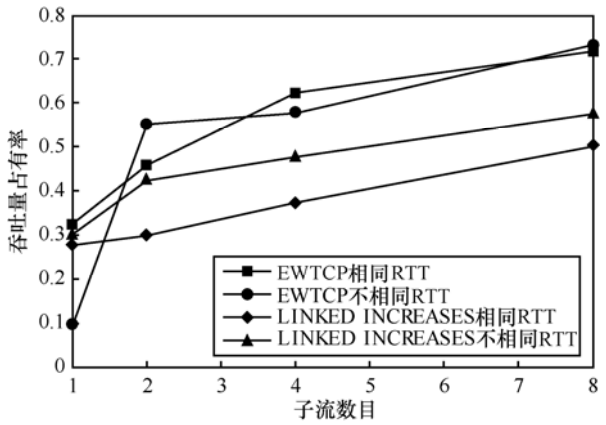


图 3　MPTCP 瓶颈链路吞吐量占有率

3.2　平衡拥塞

平衡拥塞是多路径 TCP 另一个目标，也是多路径 TCP 比较关注一个方面。本文建立了 resource pooling 模型，该模型是用来验证多路径 TCP 对资源的分配情况。模型如图 4 所示。

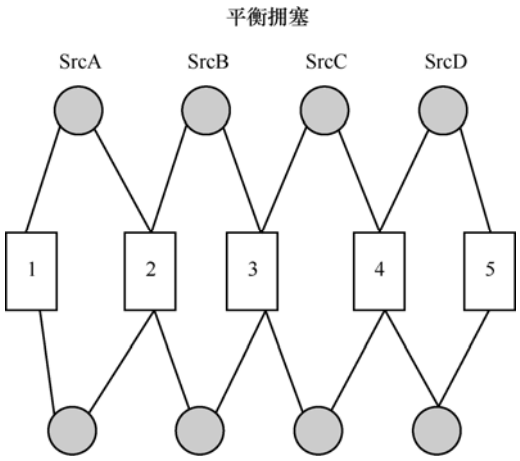


图 4　resource pooling 拓扑模型

EWTCP 算法主要保证了对各个子流的公平性，COUPLED 算法选择分组丢失率最低的链路，而 LINKED INCREASES 算法选择拥塞程度小的路径，图 5 描述的是 LINKED INCREASES 算法在拥塞程度不同情况下拥塞窗口大小的比较，开始的时候 2 条路径拥塞程度基本相同，所以拥塞窗口变化基本一致，100s 之后，增大了其中一条传输路径的拥塞程度，其拥塞窗口随即下降，将流量都转移到另一条路径上。由此可以看出 LINKED INCREASES 算法具有良好的平衡拥塞能力。

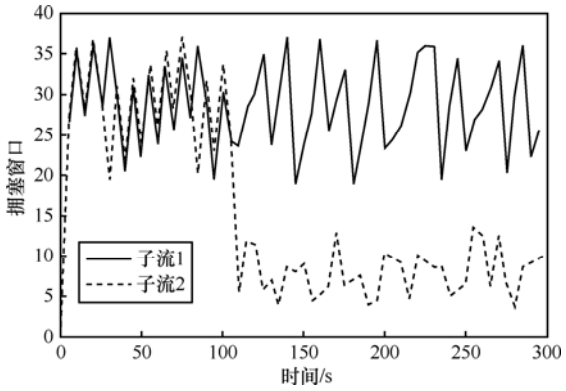


图 5　LINKED INCREASES 拥塞选择

同时，采用图 4 所描述的 resource pooling 拓扑结构来测试算法平衡拥塞资源分配的能力。每一条路径设置为 10Mbit/s，每个路径的平均 RTT 为 80ms，通过计算得出每个流的平均吞吐量如表 1 所示。

表 1　　**吞吐量表示**

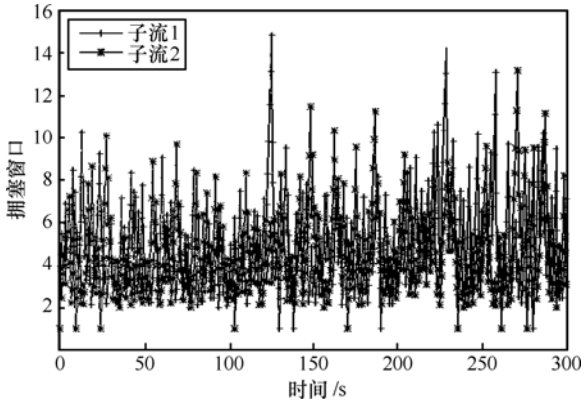
子流	算法		
	EWTCP	Coupled	LINKED INCREASES
Flow A	13.950 12	7.332 67	13.551 21
Flow B	4.417 01	0.565 04	5.013 89
Flow C	4.648 31	0.565 25	5.248 89
Flow D	9.057 78	0.567 51	9.471 58

在 resource pooling 拓扑模型中，50Mbit/s 的链路应该由所有链路平均分配。由表 1 可以看出，在 resource pooling 方面，EWTCP 算法表现仅次于 LINKED INCREASES 算法。LINKED INCREASES 算法在平衡拥塞方面性能表现良好，得到了所有链路的最高的吞吐量，但是中间 2 条链路获得的吞吐量远低于平均吞吐量 12.5Mbit/s，依然存在着资源最优分配的问题。Coupled 算法表现出了最差的性能，每一条路径获得很小的吞吐量并且平衡拥塞的能力很差。

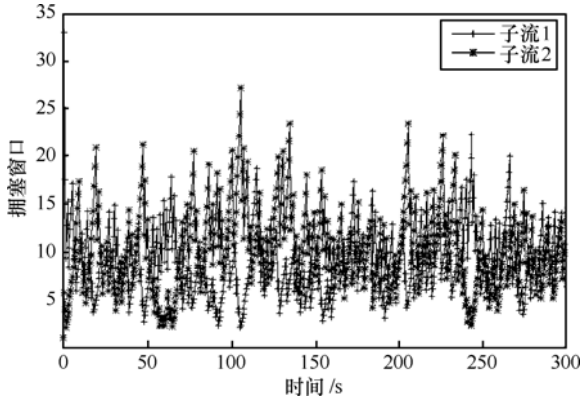
3.3　flapness

当多路径 TCP 试图平衡拥塞时，会出现 flapness 问题。当传输路径的分组丢失率 P 相近或者相等时，子路径的窗口会出现跳跃性，当它选择

一条路径进行传输时，会发现另一条路径拥塞程度很低，于是转向另一条路径增大该路径的拥塞窗口，同时发现刚刚的路径拥塞程度降低，于是又重新增大原来路径的窗口，于是会出现路径之间的来回跳转，本文称这种现象为窗口的 flapness。本文模拟实验了 LINKED INCREASES 算法在 2 个拓扑模型的性能，如图 6 所示。



(a) 1 条 TCP



(b) 5 条 TCP

图 6　LINKED INCREASES 算法在瓶颈链路下的 flap 表现

由图 6 可以看出，LINKED INCREASES 算法在瓶颈链路的 flap 表现。当传统的 TCP 流为 1 条时，LINKED INCREASES 算法几乎不存在窗口的跳跃，当传统 TCP 流增大到 5 条时，LINKED INCREASES 算法窗口跳跃依旧不明显，说明 LINKED INCREASES 算法在控制路径间的 flap 现象方面表现很好。

由图 7 可以看出，MPTCP 无论在哪种模型下，都没有出现强烈的 flap。主要是因为 MPTCP 在牺牲一定程度平衡拥塞的基础上，加入参数来限制每条子流的抢占型，使其表现出较好的性能。

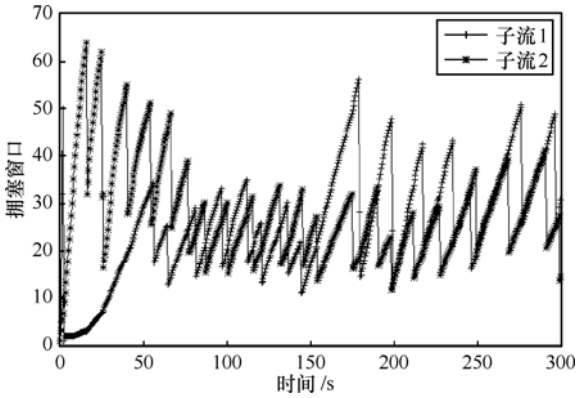


图 7　LINKED INCREASES 算法在 resource pooling 模型下的 flap 表现

3.4　算法比较

以上采用理论分析及实验，从 TCP 公平性、平衡拥塞及 flap 3 个方面比较了各种算法，结果如表 2 所示。

表 2　　多路径 TCP 算法性能比较

算法	TCP 公平性	平衡拥塞	flap
EWTCP	RTT 相等时表现好；RTT 不相等时，表现差	各子流获得相同流量	无
COUPLED	表现差	表现差	有
LINKED INCREASES	保持一定的 TCP 公平性	将流量转移到拥塞程度低的路径上	无

4　多路径 TCP 存在的问题

早期论文提出的多路径传输协议中，普遍存在的问题是对 TCP 的不公平性。主要表现在多路径 TCP 子流与传统 TCP 经过同一瓶颈链路时，子流享有与传统 TCP 相同的链路占有率，抢占了 TCP 的流量。而造成此现象的主要原因是各个子流进行独立控制拥塞，采用与传统 TCP 相同的拥塞控制算法。

IETF 在进行新的多路径 TCP 草案时，采用联合控制拥塞算法，即各子流的拥塞窗口要联合控制。这在一定程度上保证了对 TCP 的公平性。但是也产生了一些新的问题，比如它不能很有效地探测分组丢失率高的链路，导致不能充分利用分组丢失率高链路的容量；其次，当链路拥塞程度相近时，MPTCP 会随机选择一条链路分配大量的拥塞窗口，而对另一条链路则分配几乎 0 拥塞窗口，而且会在 2 条链路之间出现跳跃。

路径 r 的传输速率与 $P_r^{-1/\varepsilon}$ 成比例，其中，$\varepsilon \in [0,2]$，MPTCP 中的 LINKED INCREASES 算法虽然实现了平衡拥塞的能力，但是在资源的最大化

利用上仍存在提升空间，为此应该：1）应用 Kelly 与 Voice 的增长机制，保证资源最优利用；2）充分考虑 TCP 不同路径的 RTT、路径拥塞程度以及分组丢失率的关系。

MPTCP 的实现是在现有 TCP 基础之上，而对不同的版本将表现出不同程度的公平性，MPTCP 在实现上应该选择的 TCP 版本将是一个值得考虑的问题。应该选择的拥塞控制算法是一个值得深入研究的问题。

5 结束语

多路径 TCP 是近年来的一个热点研究课题。拥塞控制算法在其性能方面表现非常重要。MPTCP 是目前提出来的端到端的多路传输协议，已经日渐成熟。在拥塞控制方面，MPTCP 采用了各子流进行联合拥塞控制，试图在保证对其他单路径 TCP 公平的情况下，将流量转移到轻度拥塞的链路，以达到较高性能的传输。但该协议的拥塞控制算法在链路之间 RTT 相等的时候存在性能问题以及资源的最大化利用问题。

参考文献：

[1] HEDRICK C. Routing Information Protocol[R]. 1988.

[2] MOY J. Ospf Version 2[R]. 1998.

[3] HSIEH H Y, SIVAKUMAR R. pTCP: an end-to-end transport layer protocol for striped connections[A]. IEEE International Conference on Network Protocols (ICNP)[C]. Paris, France, 2002. 24-33.

[4] KEY P, MASSOULI L, TOWSLEY D. Combining multipath routing and congestion control for robustness[A]. Proc of IEEE 40th Conference on Information Sciences and Systems(CISS)[C]. Princeton, USA, 2006. 345-350.

[5] WISCHIK D, RAICIU C, GREENHALGH A, *et al*. Design, implementation and evaluation of congestion control for multipath TCP[A]. Proc Usenix NSDI[C]. Boston, USA, 2011.

[6] RAICIU C, WISCHIK D, HANDLEY M. Practical Congestion Control for Multipath Transport Protocols[R]. 2010.

[7] RAICIU C, BARR S, PLUNTKE C, *et al*. Improving datacenter performance and robustness with multipath TCP[A]. Proc of the ACM SIGCOMM 2011 conference on SIGCOMM[C]. New York, USA, 2011. 266-277.

[8] RAICIU C, HANDLEY M, FORD A. Multipath TCP design decisions. work in progress[EB/OL]. www.cs.ucl.ac.uk/staff/C.Raiciu/files/mtcp-design.pdf, 2009.

[9] PADHYE J, FIROIU V, TOWSLEY D, *et al*. Modeling TCP throughput: a simple model and its empirical validation[A]. Proc of the ACM SIGCOMM '98 Conference on Applications[C]. Palo Alto, USA, 1998. 303-314.

[10] RAICIU C, HANDLY M, WISCHIK D. Coupled Congestion Control for Multipath Transport Protocols[R]. 2011.

作者简介：

刘佩（1987-），女，河北石家庄人，中国科学院硕士生，主要研究方向为网络传输协议。

任勇毛（1981-），男，湖南邵阳人，博士，中国科学院计算机网络信息中心副研究员，主要研究方向为网络协议及体系结构。

李俊（1968-），男，安徽桐城人，中国科学院计算机网络信息中心研究员、博士生导师，主要研究方向为网络体系结构及网络安全。