文章编号：1672-2892(2011)01-0007-05

多路径传输控制协议技术综述

王 毅，廖晓菊，潘泽友

(西南计算中心，四川 绵阳 621900)

摘 要：随着互联网的应用发展，用户对带宽的需求日益增大。同时，伴随着宽带接入技术的发展，终端可以同时具有多条网络链接，然而传统传输控制协议(TCP)采取单路通信，因而造成资源浪费。为此，IETF 专门提出了多路径 TCP(MPTCP)来实现 TCP 的多路传输，从而提高链路利用率和协议鲁棒性。本文对国内外 MPTCP 的最新研究成果进行了总结，包括 MPTCP 的体系结构、路由和拥塞控制等内容，可为国内研究者进一步深入研究提供参考。

关键词：传输控制协议；多路径 TCP；体系结构；路由；拥塞控制

中图分类号：TN711.1；TP393 **文献标识码：**A

An overview of Multi-Path Transmission Control Protocol technology

WANG Yi，LIAO Xiao-ju，PAN Ze-you

(South-west Computing Center，Mianynag Sichuan 621900，China)

Abstract：With the development of Internet applications, users' demand for bandwidth is soaring sharply. Meanwhile, along with the development of broadband access technology, the endpoint can also adopt multiple network accesses. But due to one-way communication of traditional Transmission Control Protocol(TCP), the waste of resources will exist. To this end, IETF has specifically proposed Multi-Path TCP(MPTCP) to implement TCP multiplexing, thereby enhancing the efficiency and robustness. This paper gives a review of the IETF's research on MPTCP, including MPTCP architecture, routing and congestion control, aiming to provide a reference for deeply studying.

Key words：Transmission Control Protocol; Multi-Path TCP; architecture; routing; congestion control

伴随着互联网各种应用的兴起，特别是 P2P 的应用，如 BitTorrent,eMule,PPFilm,PPStream 等，用户对带宽的需求越来越大。同时，宽带接入技术也得到了前所未有的发展，尤其是无线宽带接入技术，如 WiFi,WiMAX,3G 等，使得一个用户终端同时具有到目标节点的多条链路。如果以资源共享的方式，把数据流分发到这多条链路上来提高网络带宽，这就是多路径 TCP(MPTCP)的思想，与传统 TCP 不同，MPTCP 可以提供端到端的多路通信。此外，MPTCP 还可以通过在多条链路上重传数据来提高鲁棒性。正是由于 MPTCP 的优越性，IETF 为此专门成立了 MPTCP 工作组[1-9]，致力解决 MPTCP 体系结构、拥塞控制、路由、API[6,9]、安全方面的问题[10]，并且保证 MPTCP 可向后兼容传统 TCP。MPTCP 预期目标包括[2]：a) 提高吞吐率，MPTCP 的吞吐率不低于通信节点间任意一条链路的吞吐率。这就要求 MPTCP 支持多路通信；b) 公平性，MPTCP 在任意一条链路上所占用的带宽不超过传统 TCP 模式下所占用的带宽，即具有 MPTCP 的通信双方在任意一条共享链路上的通信不会影响其他端节点的通信；c) 平衡拥塞，MPTCP 应该避免在拥塞的链路继续传输数据，这个目标是为了保证吞吐率和公平性。MPTCP 工作组成立于 2009 年，到目前还没有相关的 RFC 文档提出。本文就 2009 年~2010 年所发表的草案进行总结，旨在为国内学者对 MPTC 的进一步研究提供相关参考。

1 MPTCP 体系结构

在这个小节里将着重讨论 MPTCP 的体系结构，包括其分层的结构以及各个层次的功能，子流的建立和撤销，路径管理和 MPTCP 的按序传输。

收稿日期：2010-09-27；**修回日期：**2010-11-03

基金项目：中国工程物理研究院科学技术发展基金资助项目(2010B0403063)

1.1 MPTCP 层次结构

为了实现多路径的可靠传输，IETF 工作组扩展了传统 TCP 的功能，如图 1 所示。在应用层和传输层之间加入了支持多路径传输的 MPTCP 层，原有的 TCP 层只针对子流起作用，使得通信双方从应用层来看，传输层仍然是单路通信。

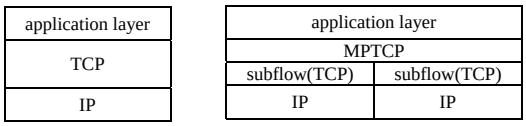

Fig.1 MPTCP layered representation
图 1 MPTCP 层次结构

新加入的 MPTCP 层功能主要包含[2]：a) 分流，把传统的 TCP 数据进行分流，分别在不同的子流上传输；b) 路径管理，用来探测和管理通信双方的可用路径。具体来说，MPTCP 按功能可以分为路径管理(PM)和包调度(PS)，如图 2 所示。路径管理负责通信双方的路径发现；包调度的功能包括：包的调度、子流接口和拥塞控制。当包调度接收到来自应用层的数据后，首先对数据进行处理，然后发送给子流；子流对数据添加序列号和确认号传给网络层。目的端子流收到数据后，将其交付给 MPTCP，MPTCP 对数据进行处理后上传给应用层。作为包调度的组成部分，拥塞控制用来控制各个子流的发送速率。

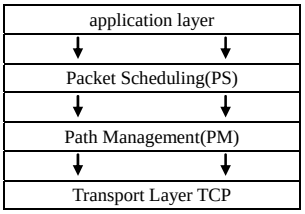

Fig.2 Function composition of MPTCP
图 2 MPTCP 的功能组成

1.2 初始化连接

与 TCP 一样，MPTCP 的初始化连接也需经历通信双方的 3 次握手。与 TCP 不一样的是，SYN,SYN/ACK 的交换过程中，增加了双方协商是否采用 MPTCP。为了兼容 TCP，MPTCP 的所有管理信息都通过 TCP 选项字段来传输，其选项编号由 INNA 来分配。在初始化连接过程中，如果通信一方支持 MPTCP，则在 SYN,SYN/ACK 携带一个 MP_CAP(Multipath Capable)选项。

图 3 中的 Sender Token 是由发送者产生的一条 MPTCP 连接的标志符，Token 只具有局部意义，并不要求通信双方 Token 值一致。MP_CAP 选项只出现在 SYN 和 SYN/ACK 的选项字段里。只有第一次和第二次握手都包含 MP_CAP 选项(表明通信双方都支持 MPTCP)，通信才采用 MPTCP；否则采用 TCP。

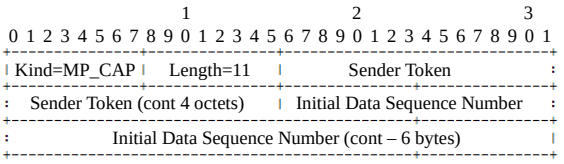

Fig.3 Multi-path capable option
图 3 支持多径选项

1.3 新建子流

支持 MPTCP 的通信双方 A,B 完成初始化连接后，在 A,B 之间便建立起一条通信链路。由于仅在这一条链路上通信，所以仍然是 TCP。A 和 B 可以通过新建子流来建立另一条通信链路，从而实现 MPTCP，MPTCP 的应用场景如图 4 所示。

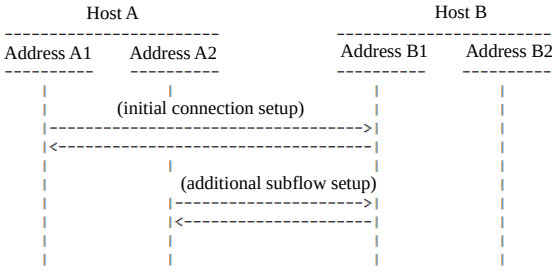

Fig.4 Example of MPTCP usage scenario
图 4 MPTCP 运用场景

图 4 中，主机 A,B 不但知道自己的 IP 地址，还可通过路径管理来获取对方的 IP 地址。A,B 任意一方可以采用一对当前没有使用的地址来新建一个子流。子流的建立通过传统的 SYN,SYN/ACK 交付完成。如图 5 所示，主机 A,B 新建了一条 Address A2<——>Address B2 的子流。

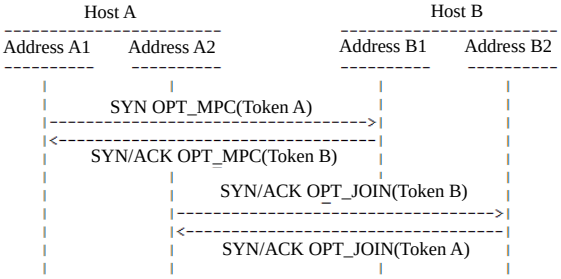

Fig.5 Starting a new subflow
图 5 新建子流

OPT_JOIN 选项用来新建一个子流，与初始化连接一样，该选项只包含在 SYN,SYN/ACK 数据段里，其格式如图 6 所示。

在 OPT_JOIN 选项里，Receiver Token 的值等同于收到 MP_CAP 里 Sender Token 的值，Address ID 只具有局部意义且具有唯一性，标识发送者在子流用到的 IP 地址。采用 Address ID 的好处在于如果一方发现自己当前 IP 地址不可用时，可以通过其他 IP 地址通告对方移除该不可用 IP 地址。为了路径管理的需要，通信双方需要在各自主机上保存一个 Address ID 到 IP 地址的映射——<Address ID,(Source IP Address,Token)>。

应该注意的是在 MPTCP 里，是通过 Token 将子流和 MPTCP 联系起来(通过 Token 实现复用分用)，这与 TCP 不同。在 TCP 里，则是通过端口来实现复用分用。MPTCP 规定目的端口可以任意设置，只要保证接收方 5 元组(协议，源 IP 地址，源端口，目的 IP 地址，目的端口)的唯一性。但是考虑到网络里中间件的存在，新建子流的目的端口号应和连接初始化中的目的端口号一致。这样做的考虑是以免网络监控软件混淆，以及最大化穿越网络中间件，如：防火墙、NAT 等。

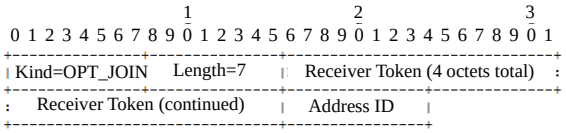

Fig.6 Join connection option
图 6 加入连接选项

1.4 关闭连接

在 TCP 里，FIN 数据段表示一方没有数据发送，当通信双方都确认了对方的 FIN 数据后，即关闭连接。MPTCP 为了保持各个子流的独立性，以及向后兼容 TCP，在一个子流上的 FIN 只是影响到该子流，只是关闭该子流而不是关闭整条 MPTCP 连接。

当应用层调用 close()函数，表明应用层已经没有数据需要发送。MPTCP 就通过所有的子流发送带有 OPT_DFIN 选项的 FIN 数据段给接收端，表示关闭 MPTCP 连接，如图 7 所示。

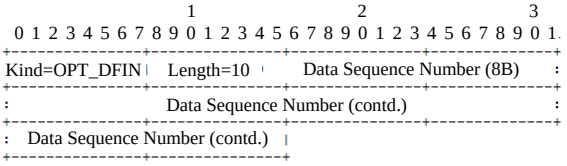

Fig.7 Close connection option
图 7 关闭连接选项

1.5 路径管理

路径管理负责交换通信双方的路径信息，本文中的路径用通信双方的 IP 地址来标识。那么路径管理包括：地址通告和地址撤销。

1.5.1地址通告

在 MPTCP 里，对于多宿主的通信方，可通过发送包含 OPT_ADDR 的数据段通告对方自己的其他 IP 地址。对方收到 OPT_ADDR 后就触发新建子流，过程如 1.3 节所示。采用这种向对方通告自己 IP 地址，然后由对方发起子流建立的被动过程，主要是考虑到对方在 NAT 后，多宿主通信方只能被动接收连接请求，如图 8 所示。

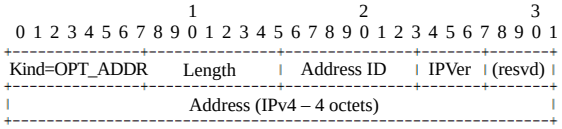

Fig.8 Add address option
图 8 添加地址选项

1.5.2地址撤销

当通信双方中的任意一方发现在一子流中自己的 IP 地址不可用时，就应该向对方发送包含 OPT_REMADR 的数据段来通知对方撤销该不可用地址。当收到 OPT_REMADR 后，端节点就应该从 Address ID 到 IP 地址映射表里移除 OPT_REMADR 包含的 IP 地址，并关闭使用这些 IP 地址的子流。发送和接收 OPT_REMADR 数据段都会触发关闭子流，如图 9 所示。

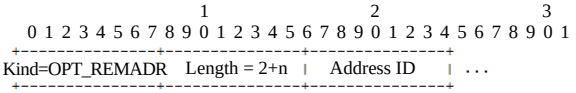

Fig.9 Remove address option
图 9 移除地址选项

1.6 MPTCP 按序传输

当子流接收到发送端发来的数据后，子流将数据交付给 MPTCP，MPTCP 对数据重组后再向上提交给应用层。为了实现数据的按序接收，在 TCP 里采用序列号的机制对数据进行编号。MPTCP 也采用类似对数据编号的机制，不过数据编号采用分层的两级模式——子流层序列号和数据层序列号。子流层序列号即是 TCP 中的序列号，而数据层序列号被封装在选项字段，如图 10 所示。OPT_DSN 包含了 Data Sequence Number(数据层序列号)和 Sub-flow Sequence Number(子流层序列号)，形成了数据序列号到子流序列号的映射。在 OPT_DSN 包含子流序列号，是考虑到网络中间件可能对子流的 TCP 协议包头的序列号进行修改。

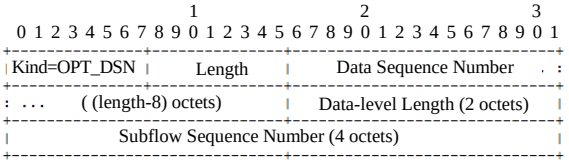

Fig.10 Data sequence number option
图 10 数据序列号选项

接收端拒绝接收不含 OPT_DSN 选项的数据段，并且不会对源端发送确认消息。在理想情况下，只需通过子流级的确认(即传统 TCP 的确认)来通知发送端数据已经顺利接收。但是考虑这样一个情况：目的端通过代理与发送端通信，当发送端发送给目的端的数据被代理接收后，代理代替目的端向发送端发送确认消息，但代理将数据发给目的端的过程中丢弃了该数据，在这种情况下，发送端仍然误认为目的端接收了数据。为了解决上述问题，MPTCP 提供一个面向 MPTCP 连接级的确认，其意义和 TCP 的 ACK 一样——OPT_DACK[3]，如图 11 所示。

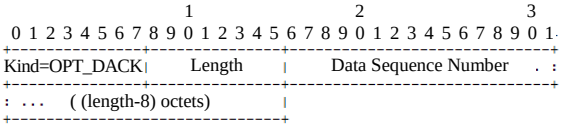

Fig.11 Connection-level acknowledge option
图 11 连接级确认选项

2 MPTCP 拥塞控制

拥塞控制是通过拥塞窗口来调节端节点的发送速率。在 MPTCP 拥塞控制里，不同的子流具有不同的拥塞窗口。为了实现资源共享，需要将各个子流的拥塞窗口耦合起来，以减少发送到拥塞链路的数据。文献[4]提出的拥塞控制算法基于 TCP New Reno，在慢开始、快重传、快恢复中，MPTCP 的拥塞控制和 TCP New Reno 一样。文献[4]只是对拥塞避免状态做了修改，提出了一个“连接增加”(Linked Increased)的算法，该算法并没有显著提高资源共享率，只能保证吞吐率达到多路径中在 TCP 下最佳链路状态下所能达到的吞吐率。但该算法能保证公平性，也具有良好的向后兼容性，易于部署。

针对吞吐率、公平性以及稳定性有各种不同的拥塞控制算法。在 MPTCP 的设计中，包含了一个子流接口，这个接口能够对数据转发进行检测，可检测丢包的时间和地点，那么在 MPTCP 拥塞控制设计里就可以综合这些信息来调节子流的数据发送速率。然而在文献[4]的设计里，并没有利用这些信息。

3 MPTCP 路由

本节主要讨论主机路由(可以通过 Route 命令来查看和修改)，如图 12 所示。

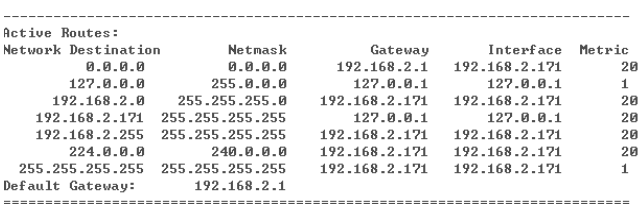

Fig.12 Host routing table
图 12 主机路由表

每条表项都包含了目的 IP 地址、掩码、网关(或者是下一条地址)，以及源 IP 地址和一个度量值，度量值越小，优先级就越高。

考虑如图 13 的网络模型，节点 A 具有 2 个网卡，其地址为 a1,a2，所对应的网关分别为 R1,R2，A 与 B 需建立通信。

假设通过 A 的主机路由得出 A 到 B 的最佳路由是 a2->R2->Internet->B。考虑在 TCP 情况下，如果 A 作为客户机主动与 B 建立连接，那么 A 选择路径 R2，并选择 a2 作为源地址(应用程序绑定源地址情况除外)。如果 B 作为客户机，A 被动建立，若 B 选择的目的地址为 a2，则符合 A 的主机路由配置。但是若 B 选择目标地址为 a1，那么 A 有 2 种选择，一是拒绝连接，二是采用源地址为 a1，通过 R2 发给 B，对于后者来说，a2 的 ISP 可能认为 a1 是伪装身份而将数据包过滤。

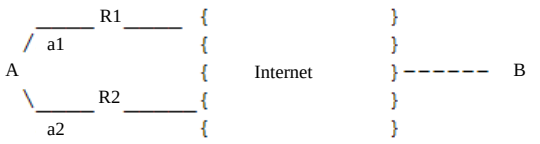

Fig.13 Communication between multi-homed host A and host B
图 13 多宿主 A 与 B 通信

在 MPTCP 的情况下，虽然 A 可以同时通过 a1,a2 和 B 通信，但是在 A 的主机路由下只能选择 a2。并不能实现 MPTCP 的多路机制。

为了解决上述 2 个问题，文献[5]提出对主机路由进行修改。对于同一个网络前缀，主机存储多个路由表项，路由表项包含了源地址、度量值和下一条地址。对于同一前缀的多个路由表项，其下一条地址各不相同，度量值越小，优先级越高。对于 MPTCP 主动连接情况下，初始化连接和 TCP 一样，接下来讨论主动连接的新建子流和被动连接的新建子流中的路由问题：

对于主动连接新建子流来说：

a) 筛选出与目的 IP 地址匹配的路由表项，从中保留最长前缀匹配的路由表项；

b) 如果没有匹配路由，则宣告目的地址不可达，子流建立失败；

c) 已经使用的 IP 地址作为一个集合，从剩下的路由表项里删除与集合里下一条地址相同的表项；

d) 如果没有剩下的路由表项，则宣告子流建立失败；

e) 从剩下的路由表项里选择度量值最小的路由表项，其下一跳作为路由，源地址为新建子流的源地址。

对于被动连接来说，如 B 向 A 发送连接请求，B 发往 A 的 SYN 数据段的源地址为 Add_S。若在主机路由表里，有 Add_S 到 B 的主动路由(即 Add_S 到 B 的路由符合最长前缀匹配)，那么 A 就将 Add_S 作为 SYN/ACK 的源地址来回复 B。否则如前面讨论的，A 可能拒绝连接，或者连接请求被 ISP 过滤。

4　结论

本文着重讨论了 MPTCP 的网络体系结构、拥塞控制和路由。MPTCP 目前的研究多数在于其体系结构部分，而没有针对安全方面的相关研究。例如在 MPTCP 里攻击者可以冒充一方向通信对方发送错误的地址移除信息，达到拒绝服务攻击的目的；也可以通过地址通告信息来实现中间人攻击[7-8]。从安全角度考虑，MPTCP 应该包括身份认证协议，这将是在未来研究中所需要解决的最重要的问题。本文总结并提炼了国内外 MPTCP 的最新研究成果，其目的是为国内研究者提供相关参考。

参考文献:

[1] Ford A,Raiciu C,Handley M,et al. Architectural Guidelines for Multipath TCP Development,draft-ietf-mptcp-architecture-00[S/OL]. [2010-09-27]. http://tools.ietf.org/html/draft-ietf-mptcp-architecture-00.

[2] Ford A,Raiciu C,Handley M,et al. Architectural Guidelines for Multipath TCP Development,draft-ford-mptcp-architecture-01[S/OL]. [2010-09-27]. http://tools.ietf.org/html/draft-ford-mptcp-architecture-01.

[3] Ford A,Raiciu C,Handley M. TCP Extensions for Multipath Operation with Multiple Addresses:draft-ford-mptcp-multiaddressed-03[S/OL]. [2010-09-27]. http://tools.ietf.org/html/draft-ford-mptcp-multiaddressed-03.

[4] Raiciu C,Handley M,Wischik D. Coupled Multipath-Aware Congestion Control:draft-raiciu-mptcp-congestion-01[S/OL]. [2010-09-27]. http://tools.ietf.org/html/draft-raiciu-mptcp-congestion-01.

[5] Handley M,Raiciu C,Bagnulo M. Outgoing Packet Routing with MP-TCP:draft-handley-mptcp-routing-00.txt[S/OL]. [2010-09-27]. http://tools.ietf.org/html/draft-handley-mptcp-routing-00.

[6] Sarolahti P. Multi-address Interface in the Socket API:draft-sarolahti-mptcp-af-multipath-01.txt[S/OL]. [2010-09-27]. http://tools.ietf.org/html/draft-sarolahti-mptcp-af-multipath-01.

[7] Bagnulo M. Threat Analysis for Multi-addressed/Multi-path TCP:draft-bagnulo-mptcp-threat-01[S/OL]. [2010-09-27]. http://tools.ietf.org/html/draft-bagnulo-mptcp-threat-01.

[8] Bagnulo M. Threat Analysis for Multi-addressed/Multi-path TCP, draft-ietf-mptcp-threat-02[S/OL]. [2010-09-27]. http://tools.ietf.org/html/draft-bagnulo-mptcp-threat-01.

[9] Scharf M,Ford A. MPTCP Application Interface Considerations, draft-scharf-mptcp-api-01[S/OL]. [2010-09-27]. http://tools.ietf.org/html/draft-scharf-mptcp-api-01.

[10] 路海. 网络访问控制技术及其应用分析[J]. 信息与电子工程, 2009,7(5):483-487. (LU Hai. Application and analysis of network access control[J]. Information and Electronic Engineering, 2009,7(5):483-487.)

作者简介:

王　毅(1986-)，男，四川省广元市人，硕士，主要研究方向为网络体系结构、网路安全.email:hellovitamine@163.com.

廖晓菊(1963-)，女，湖南澧县人，硕士，副研究员，主要研究方向为计算机网络与通信.

潘泽友(1955-)，男，武汉市人，学士，研究员，主要研究方向为计算机应用.