多源协作的传输控制机制

权 伟,崔恩放,张宏科
(北京交通大学电子信息工程学院,北京 100044)

摘 要: 多路径传输控制协议(Multipath Transmission Control Protocol,MPTCP)是传输控制协议的一种扩展,实现端到端动态地利用多个地址建立多条传输路径,从而提高网络传输质量和可靠性. 但是,MPTCP 仍存在局限,能够解决端到端已确定的多路径调度问题,但难以实现不同端以及变化端间的路径协作. 智慧协同网络是一种新型的未来网络体系,其核心思想是通过网络组件的智慧协作,最大限度优化利用网络资源,提高网络工作效率. 本论文提出了基于智慧协同网络的多源协作传输控制机制,实现了传输控制协议从单源多路径向多源多路径的突破. 具体来说,首先刻画了一种智慧协同网络的多源协作传输架构,引入子源协作传输方法,并详细介绍了多源协作机制的理论模型、报文格式以及子源协作管理的核心工作流程. 通过仿真实验验证,多源协作传输控制机制能够将拥塞窗口平均利用率从 51% 提升至 96%,并提高网络利用率和吞吐量.

关键词: 智慧协同网络; 多源协作传输; TCP 僵化; MPTCP;
中图分类号: TN911 **文献标识码:** A **文章编号:** 0372-2112 (2018)10-2527-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2018.10.029

Multi-source Collaborative Transmission Control Mechanism

QUAN Wei, CUI En-fang, ZHANG Hong-ke
(*School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing* 100044, *China*)

Abstract: MPTCP, as an extension of the TCP, is proposed to support using multiple addresses for multipath transmission. However, focusing on the collaboration of multiple interfaces inner one terminal, the original MPTCP protocol is difficult to realize collaboration among multiple sources. Smart Identifier Network (SINET) is a new architecture for the future Internet, the core idea of which is to maximize network utilization through a large scale of component collaborations. In this paper, we extend the current uni-source MPTCP to the multi-sources MPTCP, and propose an SINET based multi-source collaborative transmission solution. Specially, we introduce the multiple sub-source collaboration methods and details the models and the core workflow. Simulation results show that the proposed mechanism can improve the utilization ratio of congestion windows from 51% to 96%, hence improves network utilization and throughput greatly.

Key words: SINET; Multi-source collaborative transmission; TCP inflexibility; MPTCP

1 引言

IETF 工作组早在 RFC793[1] 中定义了一种面向连接的、可靠的、基于字节流的传输层通信协议,即 TCP 传输控制协议(Transmission Control Protocol,TCP). 随着 TCP 协议内部滑动窗口和拥塞控制机制的不断完善,TCP 协议有效解决了互联网早期诸多问题,从而在现有互联网中得到广泛应用. 然而,在 TCP 协议设计之初,网络情况并不复杂,网络功能需求也相对简单. 随着互联网规模不断扩大,TCP 协议僵化不灵活的问题日益突显[2].

TCP 协议是基于单一 IP 地址的套接字设计,即当设备拥有多个 IP 地址时,在一次 TCP 连接中只能使用一个 IP 地址. 然而,互联网体系架构中 IP 地址代表着网络接口,一台设备可以同时拥有多个网络接口. 需要绑定单一 IP 地址的 TCP 协议造成设备不能充分利用已有的网络资源. 例如,移动终端同时拥有 4G 和 WiFi 接口时,TCP 连接仅能使用其中一个接口进行通信. 同时,当设备高速移动时由于 IP 地址的频繁切换也会造成数据中断[3]. 基于此问题,D. Towsley 等人提

收稿日期:2017-05-17;修回日期:2018-01-10;责任编辑:郭游
基金项目:国家自然科学基金(No. 61602030);国家重点研发计划(No. 2016YFE0122900);中央高校基本科研业务费(No. 2016RC036)

出了一种使用多条路径同时传输的方法,并进行了理论验证[4]. O. Bonaventure 等人进一步提出了 MPTCP 协议[5],并在 RFC6824[6] 中给出 MPTCP 的定义,将 TCP 报文的选项字段进行扩展,使 TCP 可以支持多路径传输. C. Xu 等人对 MPTCP 协议的不同设计方案进行了系统调研[7].

尽管 MPTCP 协议实现了利用端到端多个 IP 的多路径传输,降低了单链路拥塞造成的影响,但在实际操作中,终端 IP 地址数目是有限的,其网络接口也是有限的,接入网资源仍难以得到充分的利用. 另外,MPTCP 本身缺乏公平性,其网络资源分配通常需要优化[2,8]. 除此之外,MPTCP 存在乱序问题,会出现吞吐量大大低于带宽聚合的情况[9,10].

为此,经过一段时间的研究,我们认为产生上述问题的根本原因在于 TCP 端到端思想. 具体来说,TCP 连接中参与传输控制的只有目的终端和源终端,难以利用中间缓存发挥作用. 理想的网络服务是当用户设备告知网络传输对象和传输内容时,网络可以充分调度自己的可用资源,提供一个低时延、稳定的数据传输. 然而,当前已有的传输机制均无法灵活利用网络资源,没有可行机制去抵消由于链路不可预知拥塞造成的网络时延.

本文的核心目标是设计一种多源协作的传输机制来适应未来互联网新型业务的发展[11]. 智慧协同网络是一种新型的未来网络体系,其核心思想是通过网络组件的智慧协作,最大限度优化利用网络资源,提高网络工作效率和服务质量[12-16]. 基于智慧协同网络架构,本文提出多源协作的传输控制机制,通过将 TCP 协议中的 Buffer 空间分割到子源设备,网络子源设备可以相互协作、多径传输,为终端设备更好的提供数据传输,并且根据不同的网络情况灵活的改变子源数目和策略,从而充分利用网络资源. 多源协作的传输控制机制可以利用带宽弥补拥塞造成的时延,并提高网络利用率和吞吐量.

2 多源协作机制结构模型

为了将用户的拥塞窗口交由网络更多地参与控制,多源协作机制在原始 TCP 端对端思想中引入中间子源. 例如,用户上传文件时,原始 TCP 需要等待对端确认,才可以继续发送下一段数据,而由于拥塞,用户不能以完全带宽上传,要低于带宽. 采用多源协作机制,可以先发送给子源,由于距离子源比较近,所以拥塞丢包较低,用户将以接近线速的速度迅速发送给子源,然后由子源跟对端一一确认,使得用户终端可以提前解放. 用户下载文件时,子源可以通过多路径,利用剩余带宽获取数据,用户从子源处就近下载,从而利用冗余带宽来弥补时延.

另外,由于子源所掌握的多个拥塞窗口属于不同终端,这样网络可以根据自己的带宽子源以及用户的服务类型,合理的分配带宽,尽量避免接入网的拥塞,充分利用接入网. 各传输机制窗口模型对比如图 1 所示.

图1 各传输机制窗口模型对比

3 多源协作机制理论模型

当前研究学者已经提出一些多路径 TCP 的流模型[17,18]. 在此基础上,我们对所提出的多源协作机制的理论模型进行分析描述.

3.1 数学模型

我们将网络节点之间的连接用集合 $L=\{1,\cdots,|L|\}$ 表示. 在多源协作机制中,网络由终端和源共享,终端用集合 $T=\{1,\cdots,|T|\}$ 表示,源用集合 $S=\{1,\cdots,|S$

|}表示,子源用集合 $SS=\{1,\cdots,|SS|\}$ 表示. 源与子源之间有多条路径 r,路径 r 包含特定的连接. 如图 2 所示,$S1$ 与 $SS1$ 之间有多条路径,连接($L1$,$L6$,$L11$,$L13$)组成一条路径,($L3$,$L10$,$L13$)组成另外一条路径. 子源可以将路径临时分配给某个终端使用,如图子源 $SS1$ 将路径($L1$,$L6$,$L11$,$L13$)和路径($L3$,$L10$,$L13$)分配给终端 $T1$ 使用.

对于每条路径 r,用τ_r表示其 RTT,ω_r表示其拥塞窗口大小. 这里需要注意的是,该拥塞窗口由源和子源共同拥有,对应的子源跟终端连接侧窗口为ω_T,则$x_r=\omega_r/\tau_r$为路径 r 的发送速率,则子源 $SS1$ 的实时带宽为 $\sum_r x_r=\sum_r \omega_r/\tau_r$,此带宽由接入终端共享. 同样的,终端 T 的发送速率为$x_T=\omega_T/\tau_T$. ω_T与ω_r没有具体的等式关系. 以上传文件为例,我们描述一下它们的关系.

我们引入数据量W_T为子源拥有终端 T 的数据量,采用双窗口模型,如图 3 所示. W_T在这里可以看作是终端上传时子源缓存的数据量,终端上传速率$x_T=\omega_T/\tau_T$可以超过多路径带宽,即 $\sum_r x_r=\sum_r \omega_r/\tau_r$,这样$W_T$会变大;如果终端上传速率小于多路径带宽,$W_T$会变小,缓存的数据会被消耗.

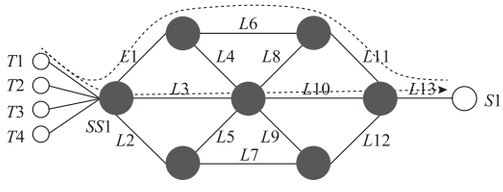

图2　多源协作机制数学模型

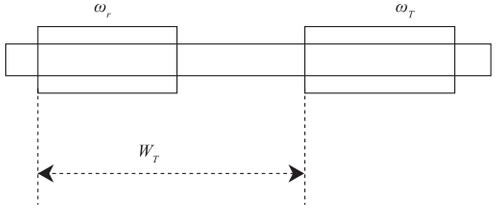

图3　双窗口模型

3.2　带宽弥补时延等式

我们设传输数据量为 Q,用 T 表示传递完 Q 的总时延,总时延又分为发送时延、传播时延、处理时延和排队时延,根据之前的数学模型,一般使用 MPTCP 传输时,总时延为:

$$T=Q/\sum_r x_r \tag{1}$$

在增加子源后,源可代替缓存,并且其带宽可超过连接它的单一终端,终端再次接收数据时可从附近源接收,理想情况下网络时延几乎为 0. 我们看作以设备最大带宽获取数据,设为 M,则总传输时间为:

$$T=Q/M \tag{2}$$

这样我们得到简单带宽弥补时延等式:

$$Q/\sum_r x_r=Q/M \tag{3}$$

达到此平衡时,用户逼近自己设备最大带宽接收数据,相当于以高于设备的带宽弥补路径拥塞等造成的时延. 使用户逼近设备最大带宽接收数据,这在没有中间缓存的传统网络中是很难做到的.

在实际情况中,我们发现 MPTCP 协议在路径增加以后,对于终端的 CPU 内存消耗也增加较快,这是由于到达的数据包发生了乱序,而且其乱序比单一路径 TCP 要更加严重. 随着路径数目的增加,对于 MPTCP 速率进一步扩大的瓶颈则在于处理时延.

多源协作机制中,子源与终端距离较近,而且连接路径要少于子源与源之间的路径,子源会先排好序再发送给终端,这相当于将终端 CPU 内存消耗转移到了子源上.

4　多源协作信令与协议设计

这一小节我们简单介绍一种实现多源协作机制的方式,将多源协作机制在传输层实现,用 MS-MPTCP 表示.

4.1　信令类型

首先,MPTCP 通用报文格式如图 4 所示:

MS-MPTCP 的子类型在 MPTCP 原有基础上进行了扩充,利用了 0x8 到 0xc 共五个标识,具体含义如下表 1:

表 1　MS-MPTCP 扩充操作

类型值	符号	名称
0x8	MS_SHARE	子源请求加入信令
0x9	MS_JOIN	子源加入连接信令
0xa	MS_DSS	子源数据序列信令
0xb	ADD_SOURCE	添加子源信令
0xc	REMOVE_SOURCE	删除子源信令

4.2　基本流程

(1) 建立 MS-MPTCP 连接

MS-MPTCP 在 MPTCP 的基础上增加了 MS_SHARE(子源请求加入连接信令),其报文格式如图 5 所示:

其中,选项类型、选项长度、子类型、版本以及 A-H 含义均与 MPTCP 中的 MP_CAPABLE 信令相同;子源的 key 为子源产生的 key;端在 MPTCP 连接中使用的 key 为端在 MPTCP 中与对方第一次 MPTCP 握手提供的 key.

MS-MPTCP 连接建立在 MPTCP 的基础上,可先建立 MPTCP 连接、增加地址,然后通过子源请求加入连接

信令(MS_SHARE)由子源向一端请求加入连接,由该端提供相关密钥.当该端不支持 MS-MPTCP 功能时,则不会响应,子源加入失败回到 MPTCP 连接.

另外,当双方均不支持 MS-MPTCP 时,无论子源向任何一方请求加入连接均不会被响应;当一方支持、另一方不支持时,支持方可由第一次握手的版本号得知对方版本不支持,当子源向支持方请求加入连接时,支持方会不响应,即有任何一端不支持时子源均不会被响应,是否支持由版本号判断.

(2) 增加新子源

MS-MPTCP 在 MPTCP 的基础上增加了子源加入连接信令(MS_JOIN),其报文格式如图 6 所示:

在建立连接的基础上,增加子源的方式类似于 MPTCP 的新地址加入连接的方式.当子源获得相关密钥后可通过 MS_JOIN 信令加入连接,具体握手流程继承之前所述的 MPTCP,不同之处在于终端记录下该次握手为子源加入连接.

(3) 常规 MS-MPTCP 操作

a. 数据序列映射

MS_DSS 选项报文格式如图 7 所示:

MS-MPTCP 相对于 MPTCP 增加了 MS_DSS 信令,该信令有两个作用:其一,在传输数据给子源时告知其传输的数据集;其二,用于告知对端该子源拥有的数据集.

b. 数据确认

数据确认继承 MPTCP 方式,报文不做改动.

c. 关闭连接

关闭子源时,可由任一端主机发送 MP_FAST-CLOSE(MPTCP 协议中的快速关闭信令)信令告知子源 C 不再发送数据给子源 C 将关闭该连接,子源 C 告知另一端主机关闭连接,停止向子源 C 继续传输数据,并在该子源拥有数据传输完成后关闭连接.

d. 可靠性和重传

重传方式包括两种,第一种为当从源或者子源请求数据需要重传时,继续从该源或子源请求重传;第二种当子源断开连接无法重传时则从源获取.

e. 拥塞控制

拥塞控制方法可灵活设定.

f. 子源策略

子源策略包括源传输给子源数据的起始和结束位置、是否使用子源、子源协作策略等等,可灵活设定.该策略的实施可由其它扩展信令完成,无需设计新信令.

(4) 子源地址信息交换

a. 告知子源地址

ADD_SOURCE 报文格式如图 8 所示:

通过 ADD_SOURCE 添加子源地址,由一端主机向另一端主机发送 ADD_SOURCE 信令,该信令包含子源地址,另一端主机响应该信令.不响应或者丢包则添加失败.

b. 移除子源地址

REMOVE_SOURCE 报文格式如图 9 所示:

通过 REMOVE_SOURCE 删除子源地址.请求删除子源时,可由任一端主机发送 REMOVE_SOURCE 信令告知子源 C 请求删除子源,子源 C 响应并告知另一端主机请求删除子源,另一端主机响应.

图4 MPTCP通用报文格式

图5 MS_SHARE报文格式

图6　MS_JOIN报文格式

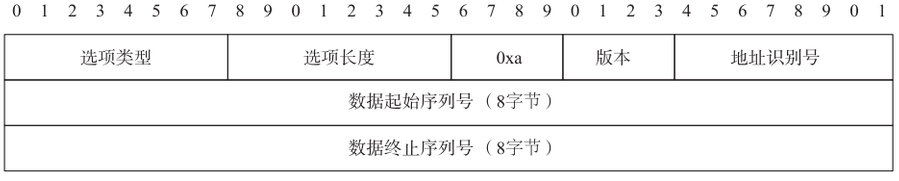

图7　MS_DSS报文格式

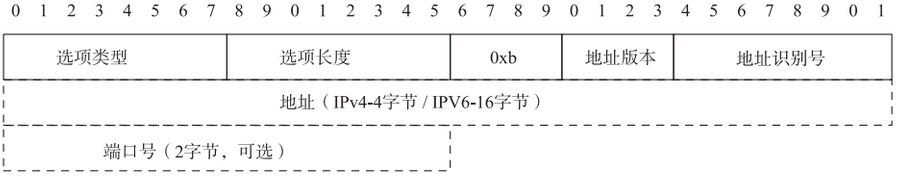

图8　ADD_SOURCE报文格式

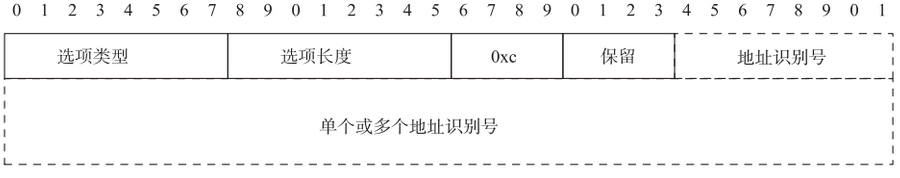

图9　REMOVE_SOURCE报文格式

5　实验仿真

我们进行了多源协作机制的仿真实验,仿真工具采用 NS3,版本为 3. 13,实验拓扑如图 10.

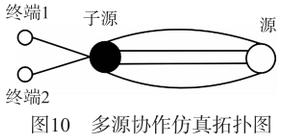

图10　多源协作仿真拓扑图

我们假设子源拥有两个终端,子源与源之间路径数为 4,每条路径速率相等. 子源分配给终端各两条路径. TCP 与源之间的拥塞控制算法为 Reno. 仿真结果如图 11 和 12.

图 11 分别为子源拥有终端 1 和终端 2 的拥塞窗口,此窗口是对源侧而言的,主要用于获取和缓存数据. 终端 1 通过子源对外获取数据通过路径 1 和路径 2,终端 2 通过子源对外获取数据通过路径 3 和路径 4. 设定最大接收窗口为 10Mbytes,则每条路径的窗口变

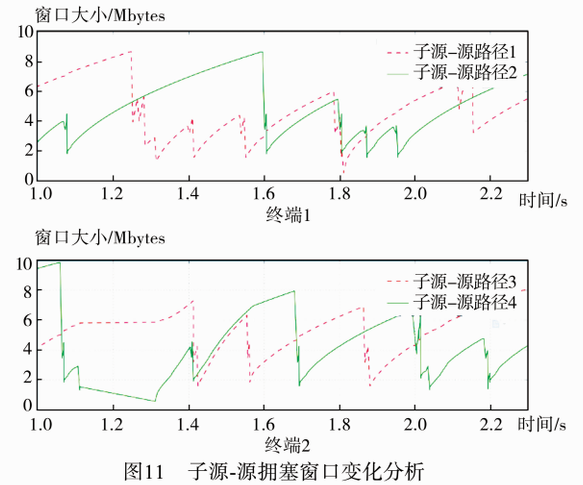

图11　子源-源拥塞窗口变化分析

化如图 11 所示. 经过计算得到四条路径的平均利用率为 51%,即如果终端只利用一条路径传输时,利用率仅为 51%,当利用两条路径时,由于多条路径聚合带宽窗口可能会超过 10Mbytes,因此,接收端会限制单条路径的速度,所以 MPTCP 两条路径的利用率会低于 51%.

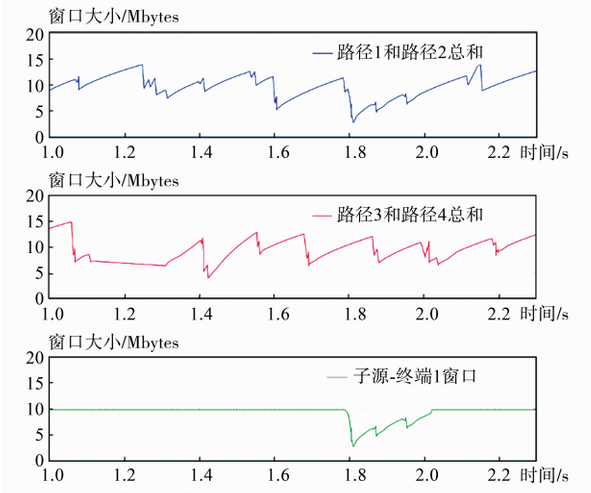

图12 子源-终端拥塞窗口变化分析

在图 12 中,对于多源协作机制,窗口之和可以大于终端接收的极限也就是 10Mbytes,当两条路径之和大于最大窗口时,子源会缓存这些数据等待终端获取. 图 12 子源-终端 1 窗口是子源对终端侧的拥塞窗口. 当缓存的数据量富裕时,则以接近最大窗口发送,当缓存数据低于阈值时则与对外侧拥塞窗口同步,图中 1.8s 时缓存数据低于阈值,窗口大小等于对外拥塞窗口. 子源-终端 1 对两条路径的利用率在本实验中可以达到 96%,远远高于传统 MPTCP.

由于网络可以控制终端的拥塞窗口,多源协作可以灵活调度带宽. 总的来说,多源协作机制不仅可以利用多个子源达成数据快速传输的目标,而且在单一子源内,可以对接入终端的服务内容进行区分和标识,对不同的服务采用不同的调度,达到服务质量最优,大大提高网络利用率和吞吐量.

6 结论

本文对 MPTCP 端到端传输进行了扩展,提出了一种多源协作传输控制机制,在端与端之间加入子源进行协作传输,将接入用户的拥塞窗口交由网络控制,以带宽弥补时延的方法,可大大提高网络利用率、提高网络吞吐量. 该机制继承了智慧协同网络的思想,并在传输层提出了一种可行方案,是对智慧协同网络的进一步完善和发展. 值得提出的是,多源协作机制通过引入一些存储冗余,增加传输路径的灵活性,来提高传输性能. 通过子源协作的优化配置,可以大大降低不必要的冗余. 在未来工作中,基于本论文提出的架构,更深入的性能分析、优化策略和测试验证将进一步研究和确认.

参考文献

[1] Transmission Control Protocol. IETF RFC793 [OL]. https://tools.ietf.org/html/rfc793,1981.

[2] Papastergiou G, Fairhurst G, et al. De-ossifying the internet transport layer: a survey and future perspectives [J]. IEEE Communications Surveys & Tutorials, 2016, 19(1): 619-639

[3] Zhang H, Dong P, et al. Promoting efficient communications for high speed railway using smart collaborative networking [J]. IEEE Wireless Communications, 2015, 22(6): 92-97.

[4] Han H, Shakkottai S, et al. Multi-path TCP: a joint congestion control and routing scheme to exploit path diversity in the internet [J]. IEEE/ACM Trans. Networking, 2006, 14(6): 1260-1270.

[5] Haddadi H, Bonaventure O. Recent Advances in Networking [M]. ACM SIGCOMM eBook, 2013.

[6] TCP Extensions for Multipath Operation with Multiple Addresses, IETF RFC6824 [OL]. https://tools.ietf.org/html/rfc6824, 2013.

[7] Xu C, Zhao J, Muntean G-M. Congestion control design for multipath transport protocols: a survey [J]. IEEE Communications Surveys and Tutorials, 2016, 18(4): 2948-2969.

[8] Khalili R, Gast N, et al. MPTCP is not pareto-optimal: performance issues and a possible solution [J]. IEEE/ACM Trans. Networking, 2013, 21(5): 1651-1655.

[9] 薛开平,陈珂,倪丹,张泓,洪佩. 基于 MPTCP 的多路径传输优化技术综述 [J]. 计算机研究与发展, 2016, 53(11): 2512-2529.
Xue K, Chen K, Ni D, Zhang H, Hong P. Survey of MPTCP-based multipath transmission optimization [J]. Journal of Computer Research and Development, 2016, 53(11): 2512-2529. (in Chinese)

[10] Honda M, Nishida Y, et al. Multipath congestion control for shared bottleneck [A]. IEEE PFLDNeT Workshop [C], 2009.

[11] Quan W, Wang K, Liu Y, Cheng N, et al. Software-defined collaborative offloading for heterogeneous vehicular networks [J]. Wireless Communications and Mobile Computing, 2018, 3810350: 1-9.

[12] 张宏科,罗洪斌. 智慧协同网络体系基础研究 [J]. 电子学报, 2013, 41(7): 1249-1254.
Zhang H, Luo H. Fundamental research on theories of smart and cooperative networks [J]. Acta Electronica Sinica, 2013, 41(7): 1249-1254. (in Chinese)

[13] 苏伟,陈佳,周华春,张宏科. 智慧协同网络中的服务机理研究[J]. 电子学报,2013,41(7):1255 – 1260.
Su W,Chen J,Zhou H,Zhang H. Research on the service mechanisms in smart and cooperative networks[J]. Acta Electronica Sinica,2013,41(7):1255 – 1260. (in Chinese)

[14] 郜帅,王洪超,王凯,张宏科. 智慧网络组件协同机制研究[J]. 电子学报,2013,41 (7):1261 – 1267.
Gao S,Wang H,Wang K,Zhang H. Research on cooperation mechanisms of smart network components[J]. Acta Electronica Sinica,2013,41(7):1261 – 1267. (in Chinese)

[15] Zhang H,Quan W,Chao H-C,et al. Smart identifier network:a collaborative architecture for the future internet [J]. IEEE Network,2016,30(3):46 – 51.

[16] Quan W,Liu Y,et al. Enhancing crowd collaborations for software defined vehicular networks[J]. IEEE Communications Magazine,2017,55(8):80 – 86.

[17] Peng Q,Walid A,et al. Multipath TCP:analysis,design,and implementation[J]. IEEE/ACM Trans. Networking,2016,24(1):597 – 598.

[18] Peng Q,Walid A,Low S. Multipath TCP algorithms:theory and design[A]. Acm Sigmetrics/International Conference on Measurement & Modeling of Computer Systems [C]. 2013,41 (1) :305 – 316,2013:305 – 316.

作者简介

权　伟　男,1987 年出生于山西运城,现为北京交通大学电子信息工程学院,下一代互联网互联设备国家工程实验室讲师,主要研究方向为未来互联网体系架构、车联网、能源互联网及网络分析. 先后在 IEEE Network、IEEE Wireless Communications、IEEE Transactions on Vehicular Technology、IEEE Communications Letters 等期刊发表学术论文 20 余篇,并担任多个国际期刊副主编和论文评审人,是 IEEE、ACM 及中国电子学会会员.
E-mail:weiquan@ bjtu. edu. cn

崔恩放　男,1995 年出生于河北沧州,现为北京交通大学电子信息工程学院,下一代互联网互联设备国家工程实验室博士研究生,主要研究方向为未来互联网体系架构、传输控制协议及车联网,申请多项国家发明专利.
E-mail:17111008@ bjtu. edu. cn

张宏科　男,1957 年出生于山西大同,现为北京交通大学电子信息工程学院教授,下一代互联网互联设备国家工程实验室主任,国家 973 首席科学家,多年来主要从事通信、计算机及信息网络科学等领域的理论和学术方面的研究,尤其是在新一代未来网络体系结构、关键理论、技术基础以及面临问题等方面有深入研究,先后在 IEEE Network、IEEE Transactions on Mobile Computing、IEEE Transactions on Parallel and Distributed Systems、电子学报等国内外重要刊物和学术会议上发表学术论文 100 余篇,授权发明专利 70 余项,并撰写网络理论与技术书籍多部.
E-mail:hkzhang@ bjtu. edu. cn