Multipath TCP
Internet-Draft
Intended status: Standards Track
Expires: December 22, 2013

K. Xue
J. Guo
P. Hong
USTC
L. Zhu
F. Yu
Huawei

June 20, 2013

TMPP for Both Two MPTCP-unaware Hosts
draft-xue-mptcp-tmpp-unware-hosts-02

Abstract

Transparent MPTCP Proxy(TMPP) is an introduced network-based function, which is under MPTCP architecture. It can help two MPTCP-unaware hosts enjoy multipath support, and can be extensively used both in the access networks and operators' networks. Meanwhile, in MPTCP architecture with TMPP, TMPP needs to modify the received packets and transmit them again(just like gateway in NAT environment). In this document, we also discuss the guarantee for data transfer on TMPP's side. The consideration of data transfer can be expanded to the MPTCP architecture with proxy.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 22, 2013.

Table of Contents

1. Introduction

MPTCP can help increase the resilience of the connectivity and the efficiency of the resource usage [RFC 6182]. When two end hosts communicate with each other, the real connection may pass through multiple sections in the networks. There are differences in these sections since networks can be divided into wired and wireless, backhaul and core networks.

In [I-D.draft-hampel-mptcp-proxies-anchors], MPTCP proxy is only introduced to help two end hosts(One is MPTCP-capable, and the other is MPTCP-unware) communicate with each other.

In order to take the full advantage of MPTCP, we can use multipath transport in a wider environment. For example, when a household access device is used, the link between the access device and one end host(the end host connects Internet via this device) performs quite well. If the end host is MPTCP-unaware, it's appropriate to use an access device with MPTCP support in MPTCP environment. Under this situation, the mobile access device provides MPTCP function towards network side, while runs traditional TCP towards end hosts.

With this method of using MPTCP, there must be some scenarios in which both the end hosts are MPTCP-unaware. Currently, it's not supported for both two MPTCP-unaware hosts to enjoy multipath transport under MPTCP architecture[I-D.draft-hampel-mptcp-proxies-anchors]. Recently, [I-D.draft-ayar-transparent-sca-proxy] presents a new architecture, named SCA (Splitter/Combiner Architecture), which enables non-MPTCP-capable single-homed hosts to benefit from multipath by means of PEPs (Performance Enhancing Proxies) placed in the access networks. This draft corresponds to this case, but it is controversial since it's completely independent of MPTCP architecture.

This document complements the work of Proxy in MPTCP by introducing a kind of network-based function, TMPP (Transparent MPTCP Proxy), which is under MPTCP architecture, and can help two MPTCP-unaware hosts enjoy multipath support. Meanwhile, in MPTCP architecture with TMPP, TMPP needs to modify the received packets and transmit them again(just like gateway in NAT environment). In this document, we also discuss the guarantee for data transfer on TMPP's side. The consideration of data transfer can be expanded to the whole MPTCP architecture with proxy, which is also the further key problem for MPTCP proxy.

1.1. Background

With respect to proxy for MPTCP, there are three kinds of proxies according to whether the end points run MPTCP fuction.

o One end host runs MPTCP, and the other end runs traditional TCP. The MPTCP proxy allows two one end hosts separately run traditional TCP and MPTCP. to run ordinary TCP (the other end is MPTCP), Twith the proxy uses talking TCP to communicate with one end and MPTCP to communicate with the other end

o Both end hosts run MPTCP. MPTCP anchor allows continuing connectivity after two mobile hosts move simultaneously, or one end host moves and the other is behind a firewall.

o Neither of two end hosts runs MPTCP, which is not supported in current MPTCP architecture[I-D.draft-hampel-mptcp-proxies-anchors]. The proxy creates multiple paths between the proxy and the other (TCP) host.

[I-D.draft-hampel-mptcp-proxies-anchors] discusses relevant features and signaling enhancements needed for MPTCP proxies and MPTCP anchors, which are both especially suited for wireless access environments. MPTCP proxies and MPTCP anchors in that draft correspond to the first two cases.

MPTCP proxy provides multipath support for MPTCP-capable hosts on behalf of their MPTCP-unaware peers, aiming at facilitating incremental deployment of MPTCP, especially for wireless environments, where traffic is dominated by interactions between mobile clients and network-side servers.

MPTCP anchor permits subflow establishment for MPTCP connections when direct interaction between end hosts fails. This permits tolerance to local IP protocol restrictions and provides robustness in case of break-before-make mobility events.

Since proxy in [I-D.draft-hampel-mptcp-proxies-anchors] is designed for specific scenarios, which can't apply to the case when two end points are both MPTCP-unaware, although the split model of TCP-MPTCP-TCP seems to be the combination of two TCP-MPTCP models. The reason is that, according to [I-D.draft-hampel-mptcp-proxies-anchors], when the connection-initiating host is MPTCP-unaware, an initial SYN packet would be added with a MP_CAPABLE option in which PROXY flag is set when passing through an implicit MPTCP network node (if there is one residing on the direct routing path).When another implicit MPTCP network node inspects the SYN packet and finds the MP_CAPABLE option with PROXY flag set, it should not insert MP_CAPABLE to the SYN-ACK

response. This will lead to no proxy service supports for a connection whenever neither end hosts is MPTCP-capable.

In order to provide MPTCP support for a MPTCP-unaware couple of peers, new signaling enhancement for connection establishment is needed. At the same time, since there will be two network nodes working for MPTCP transport (see the split model in section 2), acknowledgement of data transfer turns to be complicated, then details in data transfer SHOULD also be considered. So not only connection establishment, but also data transfer SHOULD be designed complying with the fundamental MPTCP architecture and signaling. Meanwhile the consideration of data transfer can be expanded to the whole MPTCP architecture with proxy.

1.2. Terminology

TMPP: Transparent MPTCP Proxy.

2. TMPP

TMPP is a kind of MPTCP network functions. It interacts with MPTCP connections through MPTCP signaling. TMPP can reside on MPTCP network nodes.

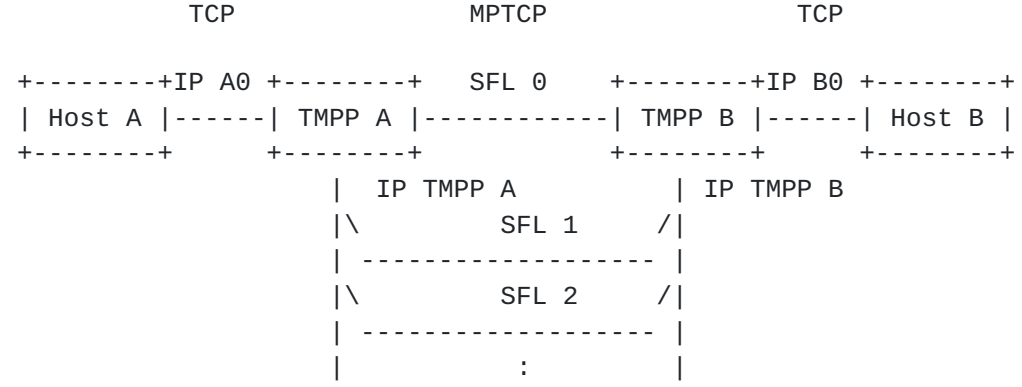

Figure 1: TCP-MPTCP-TCP split connection with TMPP

A couple of TMPPs work together to support MPTCP on behalf of MPTCP-unaware hosts. They split the connection between two MPTCP-unaware hosts into two TCP sections and one MPTCP section (Figure 1). All subflows are established by one TMPP and terminate at its TMPP peer. TMPP relays all packets arriving from one end host to another. TMPP's operations involve inserting or removing MPTCP options, translating locators (address and port) and sequence numbers, and allocating subflows to forward packets. As a result, TMPP MUST

maintain the translation relationship of locators and sequence numbers.

TMPP is designed as an implicit network function. It resides on the direct routing path between two communicating hosts. During the connection establishment on both two end hosts' side, they are treated as interacting with each other directly, while TMPP can obtain information about locators and MPTCP options via packet inspection, modify packets as necessary and thereby create the split connection.

3. Deployment Scenarios

TMPP is predominantly used when two MPTCP-unaware hosts are communicating with each other. At least one of them is located in mobile access networks enabling mobile access gateways with MPTCP function towards network side, or one network element in the network supporting multipath. In other words, the connection split-point may locate in the access side or the operators' networks.

3.1. TMPP Locates in the Access Networks

The mobile access gateway provides MPTCP function towards network side, and the multipath connection begins and ends both at the access gateway.

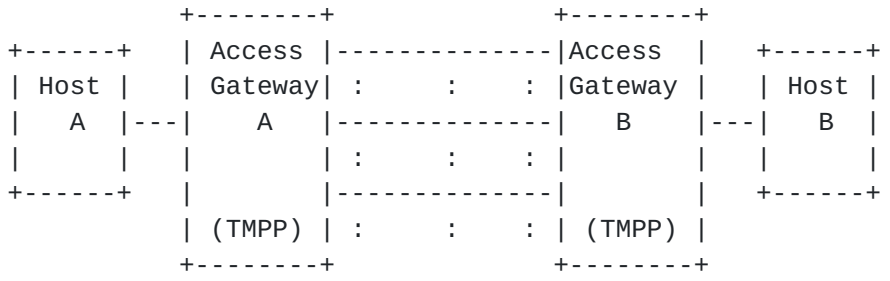

Figure 2: TMPP locates in the access networks

For instance,

o A household access device is connected to the Internet via multiple access methods, while the end device via a unique way to the access device.

o A vehicle network has several access methods, while the mobile devices hold by passengers can connect it via only one way (e.g. Wi-Fi).

While it's not realistic to make all end devices MPTCP-capable, keeping MPTCP function on network-side will be helpful by ensuring the network access device is MPTCP-capable.

In this scenario, it simply solves the problem by providing a MPTCP-capable access gateway, and it only needs network edge devices' support. However, it costs nuch because the edge device SHOULD have several SP signings.

3.2. TMPP Locates in the Operators' Networks

Currently, the bottleneck is the access side's entrance into the core network. Although the inside core network works well, the low efficiency in backhaul limits the whole system's performance. The earlier for the multiple paths to aggregate, the better, which is the same to separate, so it's suggested to put the split point into an operator network element. In this way, it will be convenient for operators' flexible management and charging. At the same time, since the multiple paths are managed by the operator, this single connection needs only one signing.

Here are two cases of this scenario.

1) The sender or the receiver is limited, the un-limited end host locates in Internet, and a MPTCP-capable P-GW manages multipath.

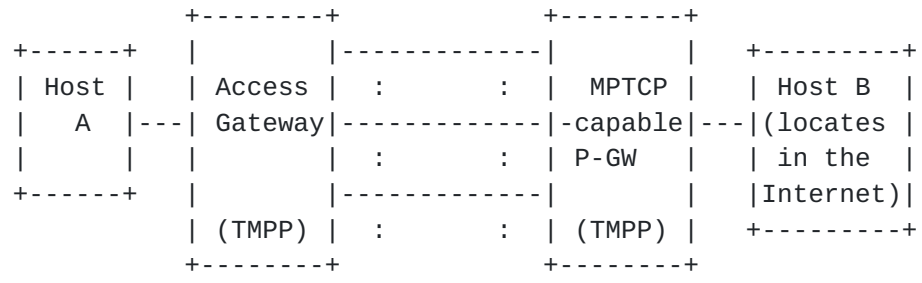

Figure 3: Only one end host is limited

2) Both the sender and the receiver are limited, and there are two MPTCP-capable P-GWs working for them separately.

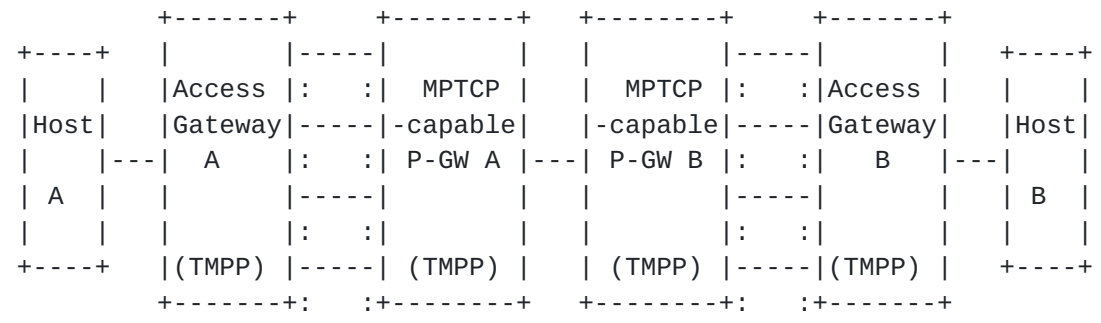

Figure 4: Both two end hosts are limited

4. Operation with TMPP

4.1. Connection Establishment

As mentioned in section 1.1, with implicit proxy proposed in [I-D .draft-hampel-mptcp-proxies-anchors], it's not supported when both end hosts are MPTCP-unaware, because the MPTCP network node refuses to add MP_CAPABLE option if the MP_CAPABLE option in the first SYN packet is added by some other MPTCP network nodes.

In order to provide chances for a connection between two MPTCP-unaware hosts also to enjoy multipath support, we make a change which requires another implicit MPTCP network node SHOULD also insert MP_CAPABLE to the SYN-ACK response even if finding the PROXY flag set in the MP_CAPABLE option in SYN packets.

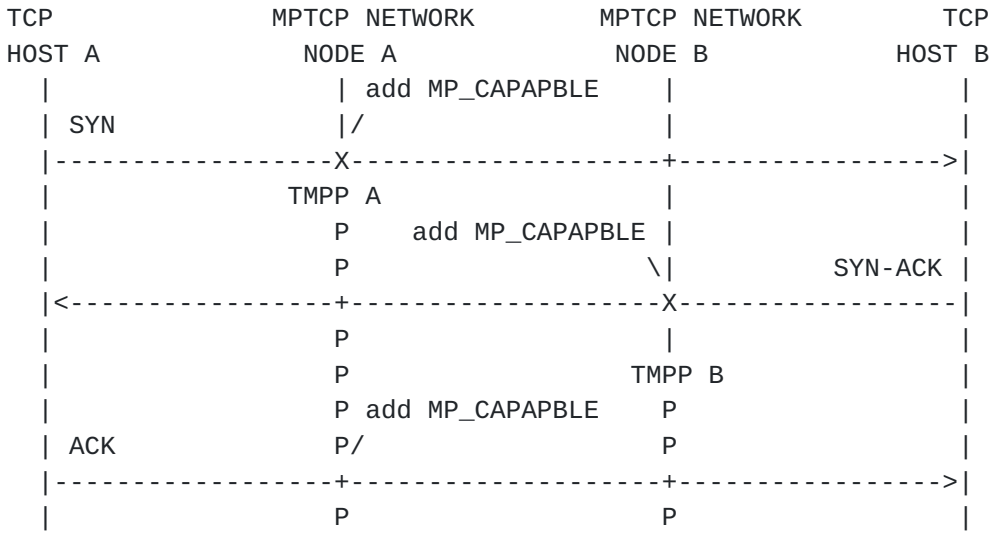

Figure 5: Connection initiation by MPTCP-unaware host with TMPP

The signaling of connection establishment is as follows:

o SYN

One MPTCP-unaware host (host A in Figure 2) starts a connection by sending a TCP SYN packet. TMPP resides on the routing path inspects the packet, and then caches the locators of host A and its peer (host B). Based on these locators, TMPP identifies and intercepts the peer's SYN-ACK response packet, as well as data packets to be transported after connection established. Here, TMPP does not change the locators contained on the packet (which is different from data transfer).

The closest TMPP to host A(namely TMPP A) is responsible for initiating multipath support. Inspecting there is no MP_CAPABLE option in SYN packets received from host A, it adds a MP_CAPABLE option (with FLAG set) into the SYN packet, then forwards the packet to host B.

Since another TMPP (TMPP B) potentially works for host B and resides on the routing path just as TMPP A does, SYN packet will be received by TMPP B before host B. TMPP B caches the locators of host A and host B, inspects the MP_CAPABLE option with proxy flag set, then becomes aware of that the host A is MPTCP-unaware and there must be a TMPP working for it. After obtaining this information, TMPP B forwards the SYN packet to host B without any change.

o SYN-ACK

After TMPP B receives the SYN-ACK response from host B, it creates a key on behalf of host B, inserts a MP_CAPABLE option (proxy flag is set) with this key into the SYN-ACK packet, and forwards the packet to host A.

When SYN-ACK packet is passing through TMPP A, TMPP A inspects the proxy flag, then caches the key produced by TMPP B for host B. The key will be used for subflow establishment.

o ACK

When the ACK response from host A arrives at TMPP A, TMPP A still SHOULD insert a MP_CAPABLE option (with PROXY flag set). Further, it produces a key on behalf of host A, and this key will be cached by TMPP B.

4.2. Subflow Management with TMPP

Splitting the established connection into two TCP sections and one MPTCP section, the couples of TMPPs become the end points for all further subflows. These subflows may be initiated by one TMPP and ended by the other TMPP. Both these two TMPPs must inform about their existence and IP addresses with each other. The PROXY flag on

those MP_CAPABLE options in SYN or SYN-ACK packets can help tell one TMPP that another TMPP exists and works for a MPTCP-unaware host. However, after connection established, TMPP is still unaware of another TMPP's IP addresses. As a result, TMPP needs to advertise its address via ADD_ADDR (as introduced in [I-D.draft-ietf-mptcp-multiaddressed]) to another TMPP.

In [I-D.draft-hampel-mptcp-proxies-anchors], considering reliability, SEEK_ADDR option is presented. This method is also accepted in this document.

4.3. Data Transfer

TMPP is the endpoint of all subflows, while runs a regular TCP connection with an end host, so TMPP SHOULD translate the transformation mode between MPTCP and TCP by replacing IP header and TCP header when forwards data packets.

MPTCP features acknowledgements at connection-level as well as subflow-level[I-D.draft-ietf-mptcp-multiaddressed], in order to provide a robust service to the application. When acknowledges a packet, TMPP receiver should send ACK to TMPP sender at subfow level as soon as the packet is accepted, while send date-level (i.e. connection-level) ACK until accept the end host's (the real end receiver) Acknowledgement.

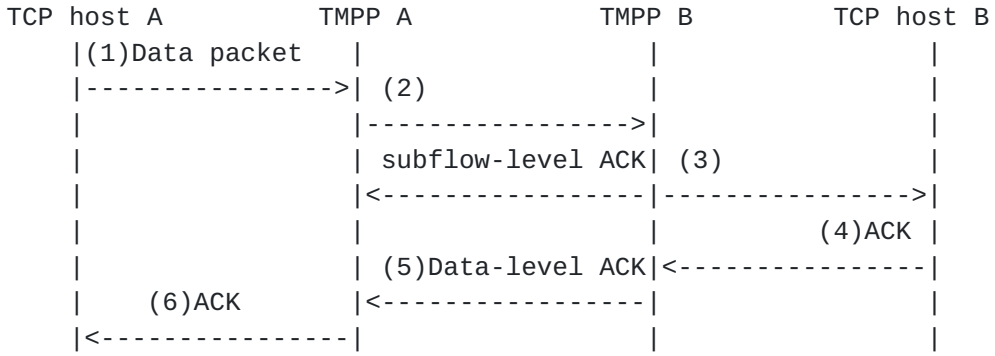

Figure 6: Data transfer with TMPP

o TCP host A sends a SYN packet that conveys A and B's IP addresses in IP head and ports and sequence number in TCP head to host B.

o TMPP A assigns the segments received from host A to subflows that already established between TMPP A and TMPP B, then changes IP head and TCP head (the locators will be changed to those of TMPP A and TMPP B's for the assigned subflows). The SN in the new TCP head and

the subflow sequence number in MPTCP option SHOULD be the SN of the specific subflow, while the data sequence number in MPTCP option SHOULD be the original SN of TCP flow (Figure 7). Since the insertion of MPTCP option and the changes of some of fields, the Check sum and TCP header length SHOULD also be reset accordingly. Then TMPP A maintains the translation relationship and sends the packet to TMPP B.

o With the reception of the packet from TMPP A, TMPP B acknowledges it at the subflow level, sending ACK to TMPP A (the ACK number is at the subflow level).

o At the same time, TMPP B recovers the locators to those of host A, removes MPTCP option, and sends the packet to host B.

o Host B responds with a simple ACK.

o TMPP B establishes and sends ACK to TMPP A at the data level.

o TMPP A establishes and sends a simple ACK to host A, with host B's locator.

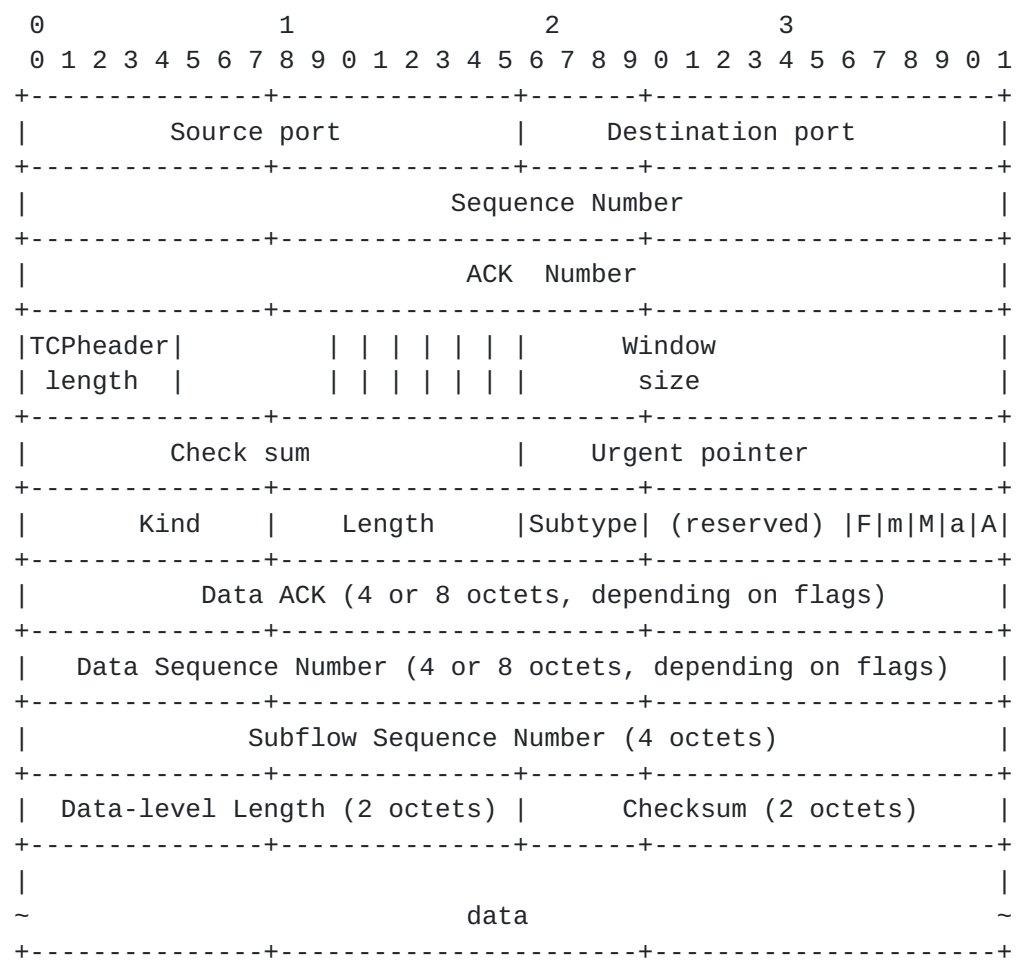

Figure 7: TCP header format

As a result of the buffer capacity limitation to the network devices, and in order not to bring too much overhead to these nodes, TMPPs are not required to cache data packets. In case TMPP B in Figure 6 doesn't receive the ACK response from host B, the lost packet should be retransmitted by the very beginner of the whole connection, i.e. host A. Optimizations could be negotiated in future versions of this protocol.

5. Security Considerations

It is recommended that before two TMPPs establish MPTCP connection, security preservation is provided by TLS/SSL, which can be further discussed in the next future work.

6. References

6.1. Normative References

[RFC6182] Ford, A., Raiciu, C., Handley, M., Barre, S., and J. Iyengar, "Architectural Guidelines for Multipath TCP Development", RFC 6182, March 2011.

6.2. Informative References

[I-D.ayar-transparent-sca-proxy]
Ayar, T., Rathke, B., Budzisz, L., and A. Wolisz, "A Transparent Performance Enhancing Proxy Architecture To Enable TCP over Multiple Paths for Single-Homed Hosts", draft-ayar-transparent-sca-proxy-00 (work in progress), February 2012.

[I-D.ford-mptcp-multiaddressed]
Ford, A., Raiciu, C., and M. Handley, "TCP Extensions for Multipath Operation with Multiple Addresses", draft-ford-mptcp-multiaddressed-03 (work in progress), March 2010.

[I-D.hampel-mptcp-proxies-anchors]
Hampel, G. and T. Klein, "MPTCP Proxies and Anchors", draft-hampel-mptcp-proxies-anchors-00 (work in progress), February 2012.

Authors' Addresses

Kaiping Xue
USTC
Room 305, EEIS Department, USTC West Campus
Shushan District , Hefei 230027
P. R. China

Phone: +86-551-3601334
Email: kpxue@ustc.edu.cn

Jing Guo
USTC
Room 305, EEIS Department, USTC West Campus
Shushan District , Hefei 230027
P. R. China

Phone: +86-551-3601334
Email: guojing1@mail.ustc.edu.cn

Peilin Hong
USTC
Room 305, EEIS Department, USTC West Campus
Shushan District , Hefei 230027
P. R. China

Phone: +86-551-3601334
Email: plhong@ustc.edu.cn

Lei Zhu
Huawei
Wireless network research department
Haidian District , Beijing 100085
P. R. China

Phone: +86-10-60611961
Email: lei.zhu@huawei.com

Fang Yu
Huawei
Wireless network research department
Haidian District , Beijing 100085
P. R. China

Phone: +86-10-60611961
Email: lei.zhu@huawei.com