MPTCP D. Wing
Internet-Draft R. Ravindranath
Intended status: Standards Track T. Reddy
Expires: April 10, 2014 Cisco
A. Ford
Unaffiliated
R. Penno
Cisco
October 07, 2013

# Multipath TCP (MPTCP) Path Selection using PCP
draft-wing-mptcp-pcp-00

Abstract

MultiPath TCP (MPTCP) allows a host to use multiple interfaces to transfer data. Without knowledge of the characterisitcs of each network path, the MPTCP stack has to send data to learn those characteristics. This document communicates network characteristics using Port Control Protocol(PCP) to allow the MPTCP stack influence its functions.

Table of Contents



# 1. Introduction

Multipath Transmission Control Protocol (MPTCP) [RFC6182] pools multiple TCP paths within a transport connection, and is transparent to the application. Multipath TCP is primarily concerned with utilizing multiple paths end-to-end, where one or both of the end hosts are multihomed. It may also have applications where multiple paths exist within the network and can be manipulated by an end host. An MPTCP connection begins similarly to a regular TCP connection and if extra paths are available, additional TCP subflows are created on these paths, and are combined with the existing session, which continues to appear as a single connection to the applications at both ends. MPTCP provides greater throughput by using multiple paths, and also resilience against path failure. The latter property additionally provides mobility functionality.

MPTCP identifies multiple paths by the presence of multiple addresses at hosts. The discovery and setup of additional subflows will be achieved through a path management method. Section 3.3.8 of [RFC6824] discusses MPTCP policies to share traffic over the available paths. MPTCP may use all paths (for maximum throughput) or a subset of paths (for network resiliency). The path selection is mostly based on local policy, OS behavior, and the MP_PRIO option.

The MPTCP API document [RFC6897] discusses the requirements for MPTCP-aware applications to select multiple paths that can provide the required flow characteristics; for example, 5Mbps of upstream/ downstream bandwidth, low loss, low delay, etc. Appendix A.3 of [RFC6897] lists two requirements (REQ-8, REQ-9) for an advanced MPTCP API which would enable the application to select paths based on the link characteristics like bandwidth, latency, etc.

This draft defines the on-the-wire protocol for such an advanced MPTCP API. It uses PCP flow extensions [I-D.wing-pcp-flowdata] to select the best path when multiple paths are available. This would be particularly relevant for applications that are highly interactive but require specific link characteristics such as certain minimum upstream or downstream bandwidth, delay, loss, or jitter characteristics. In such a situation, the MPTCP stack can use PCP to find a interface that provides the necessary characteristics. The network could even acquire the required charactertics (e.g., by assigning bandwidth to the user). The MPTCP stack may start one or more additional subflows that are not immediately used, but are available as "hot standby" for resilience and recovery purposes. PCP can be used to find those additional paths that meet the flow characteristics to handle future failover.

Readers are assumed to be familiar with MPTCP and PCP [RFC6887].

## 2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This document uses the terminology defined in MPTCP Architecture [RFC6182], Multipath TCP [RFC6824] andPort Control Protocol [RFC6887].

## 3. MPTCP stack using PCP

This section describes the algorithm a MPTCP stack can use with PCP extensions. The application would signal the flow characteristics to the MPTCP stack. For example, the MPTCP stack would receive an abstract request from the application to provide a low-latency, low-jitter, n-Mbps of upstream bandwidth and m-Mbps of downstream bandwidth service. The MPTCP stack would send PCP flow extension requests to the default router on each interface, receive PCP flow extension responses indicating the network characteristics, and tune the MPTCP stack accordingly to favor certain interfaces over other interfaces.

```
                       Host
     +-----------------------------------------+
     |                                         |
     |                                         |
     |   +-------------------------------+     |
     |   |           Application         |     |
     |   +-------------------------------+     |
     |         ^                               |
     |         |                               |
     |         v                               |
     |   +---------------+---------------+     |
     |   |   MPTCP       | PCP Client    |     |
     |   |   stack   <------->           |     |
     |   + - - - - - - - + - - - - - - - +     |
     |   | Subflow (TCP) | UDP           |     |
     |   +-------------------------------+     |
     |   |       IP      |      IP       |     |
     |   +-------------------------------+     |
     +-----------------------------------------+
```

Figure 1: MPTCP stack using PCP

The below steps briefly describe how a MPTCP stack uses the PCP FLOWDATA option:

1. The application requests the MPTCP stack to setup a connection towards a server/remote peer. The MPTCP stack discovers all the available interfaces and gathers the source addresses from these interfaces. This includes addresses from different interfaces (in the case of the host having multiple interfaces), or from the same interface (Multihoming), and also confirms that PCP Flow Extensions is supported.

2. The application signals the required flow characteristics to the MPTCP stack via a API (such as the abstract API described in Appendix A of [RFC6897]). After getting the flow characteristics, the MPTCP stack uses the PCP client to send PCP MAP opcode with FILTER (section 11 of [RFC6887]) and FLOWDATA options (section 3 of [I-D.wing-pcp-flowdata]) to signal the flow characteristics like bandwidth, loss, delay, etc to multiple PCP servers.

3. After receiving the PCP Flow extension responses from multiple PCP servers, the MPTCP stack sorts the source addresses according to the link characteristics.

4. The MPTCP stack picks the source address from the above sorted list and uses the procedures explained in [RFC6824] to send a SYN with MP_CAPABLE flag set to indicate to the server (peer) that this host is MPTCP capable, in order to initiate the primary subflow.

5. If the server supports MPTCP then the stack will either choose to create subsequent subflows using the sorted source address list from step 3 for resiliency purposes, or for use in parallel with the primary subflow to exchange data at a higher throughput. The choice here will likely depend on the stack's interpretation of the application's required flow characteristics.

6. Any changes to the path characteristics that the PCP client receives will be indicated to the MPTCP stack which then may chose to migrate a subflow or consolidate subflows.

7. MPTCP stack can use PCP to communicate with PCP-controlled NAT to learn external IP address, port and adverstise in ADD_ADDR MPTCP option to the remote peer. MPTCP stack can also use PCP to communicate with PCP-controlled firewall to permit incoming TCP connections from the remote peer.

```
                         +---------+
+-----------+            |         +----------- { ISP-A }
| App       |-<network>--+ router |
+-----------+            |         +----------- { ISP-B }
                         +---------+


  App with MPTCP stack      PCP server      PCP Server
       and PCP client          (ISP A)         (ISP B)      TCP server
 ------------------------        |              |               |
```

```
    Address A     Address  B       |              |               |
    ---------    ----------        |              |               |
      |              |             |              |               |
      |              |             |              |               |
      |-PCP MAP + FLOWDATA-------->|              |               |
      |              |             |              |               |
      |              |--PCP MAP + FLOWDATA------->|               |
      |              |             |              |               |
      |<-SUCCESS-------------------|              |               |
      |              |             |              |               |
      |              |<-SUCCESS-------------------|               |
      |              |             |              |               |
      |    ISP A is the best path indicated by PCP|               |
      |              |             |              |               |
      |------------TCP SYN(MP_CAPABLE)--------------------------->|
      |<-----------TCP SYN+ACK (MP_CAPABLE)---------------------->|
      |------------TCP ACK (MP_CAPABLE)-------------------------->|
      |              |             |              |               |
      |              |             |              |               |
      |     Setup additional subflows             |               |
      |              |             |              |               |
      |              |------------TCP SYN(MP_JOIN)--------------->|
      |              |<-----------TCP SYN+ACK (MP_JOIN)-----------|
      |              |------------TCP ACK (MP_JOIN)-------------->|
      |              |             |              |               |
```

Figure 2: MPTCP stack using PCP

# 4.  Multiple Interfaces

An MPTCP session begins similarly to a regular TCP connection.  If multiple paths are available, the MPTCP stack can use PCP flow extensions [I-D.wing-pcp-flowdata] to determine the best path.  The advantage is PCP can be used to select the most suitable paths instead of having MPTCP stack try out all paths.  When a host has multiple interfaces available (for example 3G/4G, WiFi, VPN etc), an MPTCP application or the MPTCP stack can choose the interface for the primary subflow and interfaces for subsequent subflows according to the path characteristics, as discussed in the previous two sections.

## 4.1.  Interface Availability

A MPTCP stack using the procedures described in [I-D.deng-mif-api-session-continuity-guide] will be notified whenever existing interfaces become unavailable or new interfaces are available.  For example the MPTCP stack implementation in the Linux kernel is aware of the changes in the availability of interfaces and can react accordingly.

In such cases the MPTCP stack can use PCP to consolidate sublows or migrate an existing subflow, as described below.

#### 4.1.1. consolidate subflows

When a new interface is discovered, the MPTCP stack can use PCP flow extensions to determine the link characteristics of the new path. If the new path can provide the required flow characteristics then MPTCP could reduce the number of subflows in use. For example, assume three subflows were in use to meet the application bandwidth demand: the primary path providing bandwidth of 2Mbps, the secondary path providing 1Mbps, and the tertiary paths 2Mbps. If PCP determines that the new path can provide 3Mbps, then one subflow can be set up in the new path and, and some of the subflows can be migrated to this new path and thus reduce the number of subflows by closing the old ones. Other factors like jitter, delay, and loss MAY also be considered in the decision to migrate subflows.

#### 4.1.2. migrating an existing subflow

When a existing interface becomes unavailable, the MPTCP stack picks the unused interfaces and uses PCP flow extensions to determine the interfaces which can provide the required flow characteristics. MPTCP stack will follow the previously described steps to pick one or more of the unused interfaces for creating additional subflows.

## 5. Switch-over

It is possible that the characteristics of a link might change over time, and the MPTCP stack might want to move the subflow to a different interface. For example, if a competing high-bandwidth flow has finished, more bandwidth is available for the MPTCP flow; the DSL line rate might have improved (or degraded); the link speed may have been dynamically increased (or decreased). When link quality changes in such a fashion, a PCP server will send PCP response which could carry a FLOWDATA option where the data fields contain different values from the first response. Upon receiving PCP response, the MPTCP stack can tune its behavior (e.g., increase or decrease traffic on the interface that is now more or less favorable).

## 6. Using MP_PRIO mechanism of MPTCP along with PCP

MPTCP has a priority mechanism, MP_PRIO, for setting a path to be backup flow. This allows additional subflows to be set up but not used until no higher priority subflows are available, allowing fast fail-over. The MP_PRIO value of a subflow can be changed during the lifetime of the session. A PCP server could send a notification to the PCP client whenever path characteristics change, thus the PCP client can indicate the same to the MPTCP stack which could change the MP_PRIO values for the associated subflow(s) and trigger switch-over appropriately.

## 7. PCP Instance ID usage in MPTCP flows

The instance identifier field in PCP flow extensions would help the PCP server to co-relate multiple subflows that are part of the same MPTCP session. The instance ID can be also be used by the service provider to co-relate all the subflows of a MPTCP session.

## 8. IANA Considerations

None.

## 9. Security Considerations

Security considerations discussed in [RFC6887] are to be taken into account.

Security considerations discussed in [RFC6824] are to be taken in to account when creating new TCP subflows.

## 10. References

### 10.1. Normative References

[I-D.ietf-pcp-proxy]
Boucadair, M., Penno, R., and D. Wing, "Port Control Protocol (PCP) Proxy Function", draft-ietf-pcp-proxy-04 (work in progress), July 2013.

[I-D.wing-pcp-flowdata]
Wing, D., Penno, R., and T. Reddy, "PCP Flowdata Option", draft-wing-pcp-flowdata-00 (work in progress), July 2013.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

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

[RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.

[RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, January 2013.

[RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.

[RFC6897] Scharf, M. and A. Ford, "Multipath TCP (MPTCP) Application Interface Considerations", RFC 6897, March 2013.

## 10.2. Informative References

[I-D.deng-mif-api-session-continuity-guide]
Deng, H., Krishnan, S., Lemon, T., and M. Wasserman, "Guide for application developers on session continuity by using MIF API", draft-deng-mif-api-session-continuity-guide-03 (work in progress), October 2012.

[RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.

[RFC6356] Raiciu, C., Handley, M., and D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols", RFC 6356, October 2011.

## Authors' Addresses

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California  95134
USA

Email: dwing@cisco.com

Ram Mohan Ravindranath
Cisco Systems, Inc.
Cessna Business Park,
Kadabeesanahalli Village, Varthur Hobli,
Sarjapur-Marathahalli Outer Ring Road
Bangalore, Karnataka  560103
India

Email: rmohanr@cisco.com


Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka  560103
India

Email: tireddy@cisco.com


Alan Ford
Unaffiliated

Email: alan.ford@gmail.com


Reinaldo Penno
Cisco Systems, Inc.
170 West Tasman Drive
San Jose  95134
USA

Email: repenno@cisco.com