MPT Network Layer Multipath Library

Gábor Lencse
Szabolcs Szilágyi
Ferenc Fejes
Marius Georgescu (presenter)

What is MPT?

• MPT
  – is a network layer multipath solution
  – provides a tunnel over multiple paths using the GRE-in-UDP encapsulation [RFC 8086]
  – is different from both MPTCP [RFC 6824] and Huawei's GRE Tunnel Bonding Protocol [RFC 8157].

• Benefits of MPT
  – Path throughput capacity aggregation
  – Resilience to network failures
  – Better suits to real-time traffic than MPTCP
MTP can also be used

• As a router,
  – routing the packets among several networks between the tunnel endpoints, thus establishing a multipath site-to-site connection.

• As an IPv6 transition technology
  – The version of tunnel IP and the version of path IP are independent from each other, therefore MPT can also be used for IPv6 transition purposes.
MPT in the Networking Stack

```
+---------------------------------------------+        
|               Application (Tunnel)            |        
+-------------------------------+        
|                TCP/UDP (Tunnel)              |        
+-------------------------------+        
|               IPv4/IPv6 (Tunnel)             |        
+-------------------------------+        
|                 GRE         |   +---+        
+-------------------------------+        
|          GRE-in-UDP          |       |
+-------------------------------+        
|         UDP (Physical)       |    UDP (Physical)       |   +-----+        
+-------------------------------+        
|     IPv4/IPv6 (Physical)     |  IPv4/IPv6 (Physical)   |        
+-------------------------------+        
|      Network Access          |    Network Access       |
+-------------------------------+        
```
MPT Concept

• MPT implements a tunnel over several paths

• The tunnel traffic packets are mapped to one of the paths using
  – per packet based mapping (implemented)
  – flow-based mapping (planned)
  – combined mapping (planned)

• MPT data packet encapsulation
Connection specification

• A connection defines a communication session between two tunnel endpoints
• A connection may contain several paths
• All the parameters of the connection are stored in a “connection specification”, containing e.g.:
  – name, permissions, tunnel IP version, etc.
  – number of paths, their definitions, their weights, etc.
Control plane overview

• When the MPT servers are started they establish connections automatically (using configuration files)
• Connections may also be later established or teared down dynamically
• Paths may be added to / deleted from the connections
• There is an optional keep-alive mechanism for paths
Data Plane Overview – 1/2

+---------------------------------------------+
|    Tunnel Interface                         |
+---------------------------------------------+

#   \
#   \
#+---------------------------------------------#   \
#   \
# +---------------------------------------------+   \
# |   data packet          | forwarding to the  | #   \
#  |   reading             | tunnel interface   | #   \
# +---------------------------------------------+   \
#   \
# +---------------------------------------------+   \
# |   checking connection| packet reordering   | #   \
# |   specification      |     (optional)     | #   \
# +---------------------------------------------+   \
#   \
# +---------------------------------------------+   \
#<======# Control #

Data Plane Overview – 2/2

# +---------------------------------+---------------------------------+ # \/# MPT #
# \| /\ #<=====# Control #
# \/#   #
# +---------------------------------+---------------------------------+ #
# | path selection, | | data | #
# | GRE-in-UDP encaps. | | checking | #
# +---------------------------------+---------------------------------+ #
# \| / #
# \/# #
# +---------------------------------+---------------------------------+ #
# | physical data | | data packet | #
# | transmission | | reading | #
# +---------------------------------+---------------------------------+ #
# \| / #
# \/# #
# +---------------------------------+---------------------------------+ #
# | Physical Interface | #
# +---------------------------------+
Per packet based mapping

• A mapping decision is made for every packet
• The paths have their own weights
• The number of bytes sent to a path is proportional to the weight of the path
  – The draft contains
    • precise algorithm description
    • sample implementation (C code)
Flow-based mapping

• Flows are identified by the usual 5-tuple
• All the packets of a given flow are handled the same way
  – Rationale: provide different QoS for different classes of traffic e.g. VoIP, http, bit-torrent, etc.
    • See samples for possible scenarios in the draft
  – No implementation yet.
Packet reordering

• Different paths may have different delays
• The order of the tunnel packets may change
• Order-right delivery (optional)
  – Based on GRE sequence numbers
  – Done by the receiver, using reordering buffer
  – Controlled by parameters
    • reorder window (size of reordering buffer)
    • maximum buffer delay
  – Working well, parameter selection is still researched
MPT Results

• MPT has one working implementation [1]
• It was tested as a solution for:
  – path throughput capacity aggregation [2]
  – fast connection recovery [3]
  – elimination of the stalling events on YouTube video playback [4]

Thank you for listening!

• The Internet Draft is available:

• All comments are welcome!
  – On site to Marius Georgescu
  – In e-mail to Gábor Lencse: gabor-l@is.naist.jp