An MPTCP Option for Network-Assisted MPTCP Deployments: Plain Transport Mode

draft-boucadair-mptcp-plain-mode-06
IETF 95-Buenos Aires, March 2016

M. Boucadair (Orange)
C. Jacquenet (Orange)
D. Behaghel (OneAccess)
S. Secci (Universite Pierre et Marie Curie)
W. Henderickx (ALU/Nokia)
R. Skog (Ericsson)
O. Bonaventure (Tessares)
S. Vinapamula (Juniper)
Outline

- Rationale
- The Plain Mode MPTCP Option
- Where to convey the option
- Handling UDP packets
- Some issues
- Next steps
Network-Assisted MPTCP: Rationale

• Given
  – The MPTCP penetration rate is close to null at the server side, and
  – Network Providers do not control customers’ terminals
• A network-assisted model is attractive to offer bonding services

• ASSUMPTION: All access networks are managed by the same Network Provider
How many times did you hear: “MPTCP is not my friend, because …”?

• When you discuss with one of your favorite vendor(s)
• Each time you read a benchmark about bonding solutions
  – Excerpt from a document released in February 2016 by HGI ([link](link))

<table>
<thead>
<tr>
<th>Pro</th>
<th>Con</th>
</tr>
</thead>
</table>
| • Defined (IETF RFC 6824, IETF RFC 6356)  
  • Implemented  
  • Can be implemented end-to-end avoiding deployment of a new network element (HAG)  
  • Works on a per application basis, so can perform dynamic, per application steering. | • Current implementations do not exploit MPTCP’s full potential  
  • Simple implementations may not provide significant advantages over regular TCP  
  • Policies need to be created and tuned by the Operator. No standard to help.  
  • Requires 2 IP addresses  
  • Jitter and latency will be greater than that of the highest of the 2 paths  
  • Only works for TCP |

  – Some of the above comments are “odd”, but the one about UDP is a valid one

• **This document proposes an MPTCP extension so that connections can carry any kind of traffic (UDP, in particular) without requiring any encapsulation scheme**
One Single Option, Multiple Uses

• The option is called: Plain Mode (PM)

  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +---------------+---------------+-------+-------+---------------+|     Kind      |     Length    |SubType|D|Flag |   ...   Port (2 octets, optional)   | +-------------------------------+
  |---------------|---------------|-------|-------|---------------|
  | Kind          | Length        |SubType|D|Flag | Protocol      | +---------------------------+
  | Address (IPv4 - 4 octets / IPv6 - 16 octets) | +--------------------+
  | Port (2 octets, optional) | +-------------------+

  – **D-bit (direction bit):** indicates whether the enclosed IP address and/or port number are the original source (D-bit is set) or destination (D-bit is unset) IP address and/or port
  – **Protocol:** Indicates the protocol that is carried in the MPTCP connection, e.g., 6 (TCP), 17 (UDP)
  – “**Flag**”: A set of reserved bits for future assignment as additional flag bits
  – **IPv4/IPv6 Address:** Includes a source or destination IPv4/v6 address
  – **Port:** May be used to carry a source or destination port number; valid for protocols that use a 16-bit port number
One Single Option, Multiple Uses

Outgoing SYN/without source address preservation at the Concentrator

A pool of external IP addresses is configured

ccf(concentrator customer-facing interface); cif(concentrator Internet-facing interface)
One Single Option, Multiple Uses

Incoming SYN/without address preservation at the Concentrator

A mapping entry is instantiated

<table>
<thead>
<tr>
<th>dst: IP@s</th>
<th>src: IP@d</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst: IPcpe@1</td>
<td>src: IP@ccf</td>
</tr>
<tr>
<td>PM(D=1; IP@d)</td>
<td>dst: IP@cif</td>
</tr>
</tbody>
</table>

A mapping is maintained for this external IP address and port

D retrieves the external IP address and port of H1 using, for example, a rendezvous service

dcf(concentrator customer-facing interface); cif(concentrator Internet-facing interface)
One Single Option, Multiple Uses

- Address preservation is required in IPv6 deployments, in particular
- Does not break applications with address referrals
One Single Option, Multiple Uses

A pool of external IP addresses is configured

Outgoing UDP packet/without source address preservation at the Concentrator

A mapping entry is instantiated

A mapping entry is instantiated

ccf(concentrator customer-facing interface); cif(concentrator Internet-facing interface)
Where to Convey the PM Option?

• In SYN segments (RECOMMENDED)
  – The CPE and the Concentrator should maintain a state
  – The option should be included in this order:
    • Dedicated option space, if there is enough room left
    • In the SYN payload, otherwise

• It may be tempting to include the option in all segments (stateless)
  – ..but this design leads to an overhead
  – Some implementers reported that it is complex to integrate in an MPTCP stack
Carrying UDP Traffic

- Dedicated subflows are established to carry UDP traffic
  - These sub-flows can be established prior to the receipt of UDP packets (optimize 3WHS), or
  - Initialized upon receipt of an UDP datagram elected to the bonding service: SYN with data in payload (RECOMMENDED)
- UDP packets are “transformed” into TCP packets by the CPE/Concentrator and which carry the PM Option with the “Protocol” field set to 17
  - UDP header is swapped to a TCP header
- To avoid UDP fragmentation, it is RECOMMENDED to increase the MTU by at least 12 bytes the accommodate the overhead of the UDP/TCP header swapping
- Some TCP features may be disabled by the CPE or Concentrator such as reordering: deployment-specific
Carrying UDP Traffic: Some Open Issues

• **Issue#1:** Include multiple payloads in the same MPTCP message or not?
  – The current version assumes a simple mode with “1:1” header swapping

• **Issue#2:** Do we need to indicate explicitly the payload boundaries?

• **Issue#3:** The behavior to follow if swapping UDP/TCP headers leads to fragmentation
  – Not an issue if the MTU is well configured?
  – Declare these packets as not candidate for the bonding service?
  – Fragment the transformed packet and reassemble it before extracting the corresponding UDP packet?
  – Declare it out of scope of the specification?
Some Recommendations & Assumptions

• For IPv4 bonding services, the *default behavior does not assume address preservation* – i.e., Only one instance of the PM option will be present

• The solution *relies upon IETF BCPs and recommendations*, especially:
  – RFC4787, RFC5382, RFC6888, and draft-ietf-tsvwg-behave-requirements-update
  – CPE and Concentrator NAT capabilities are not altered

• Whether the CPE/Concentrator preserves *DSCP marking or rewrites it is deployment-specific*

• The support of features such as *MSS clamping is implementation-specific*
Incoming Connections

• In order to allow for incoming connections, means to instruct the concentrator about how to forward incoming traffic to the appropriate CPE are required

• **Compatibility with UPnP IGD is RECOMMENDED**
  – SOCKS-based deployments will require an interworking function (which does not exist!)

• **Reuse existing code/protocols, e.g.:**
  – Port Control Protocol (RFC6887)
  – UPnP IGD/PCP Interworking Function (RFC 6970)
Recap

• No tunnel, no encapsulation
• No out-of-band signaling for each MPTCP subflow
• Carries any protocol (incl. UDP) for the benefit of massive MPTCP adoption
• Accommodates various deployment contexts
• Prototype implementations are underway
What’s Next?

• Request mptcp WG adoption
• Comments and contributions are welcome
Appendix
Why not my favorite protocol: SOCKS, for example

- Too chatty
- UDP bonding is not natively supported
- Need for UPnP IGD-SOCKS interworking

TCP SYN/SYN ACK/ACK (to IP.dest, port-dest)

(MPTCP 3 way handshake (SYN/SYN ACK/ACK)
including the Multipath extension for TCP
selection message/version=05,
Number of methods supported, list of methods
METHOD selection (version=05, method=02)
Authentication Request (login, password)
Authentication Ack (status=SUCCESS)
Socks COMMAND (CONNECT to IP.dest::port-dest)
Socks REPLY (status=SUCCEEDED)

HCPE establishes MPTCP subflow on DSL

TCP flow

MPTCP on DSL

On LTE path

(MPTCP 3 way handshake (SYN/SYN ACK/ACK
IP.SRC=IP.LTE, PORT.DEST=1080)
Including the Multipath extension for TCP
selection message/version=05,
Number of methods supported, list of methods
METHOD selection (version=05, method=02)
Authentication Request (login, password)
Authentication Ack (status=SUCCESS)
Socks COMMAND (CONNECT to IP.dest::port-dest)
Socks REPLY (status=SUCCEEDED)

HCPE establishes MPTCP subflow on LTE

TCP flow

MPTCP on LTE

TCP Relay and MPTCP distribution

TCP flow

MPTCP on DSL

MPTCP on LTE

TCP flow