draft-ietf-mptcp-api-01

Michael Scharf, Alan Ford

March 31, 2011

CHANGES COMPARED TO -00
MINOR UPDATES

- Port numbers in abstract function TCP_MULTIPATH_SUBFLOWS
  - Function returns both the IP addresses and the port numbers of subflows
  - -00 only mentioned the IP addresses
- Port numbers in abstract TCP_MULTIPATH_ADD
  - Function MAY indicate a desired port number
  - Port number MAY be different from existing subflows
  - Only a hint, implementation MAY not use addresses/ports, or only a subset
- Appendix: Additional entries in the candidate list for an extended API
  - REQ9 (new): Indication of app characteristics, e. g., amount of data
    Addresses Bob's comment on the mailing list
  - REQ11: Configuration of a subflow as a backup path

CHANGES COMPARED TO -00
MINOR UPDATES (CONTD.)

- Alternative approaches for avoiding non-MPTCP-capable paths
  - Try both MPTCP and TCP in parallel and respond to whichever replies first
  - Similar to the “Happy Eyeballs” proposal for IPv6
  - Mentioned as one possibility, not mandated

UPCOMING CHANGES
REVIEW OF JAVIER UBILLOS

- Concern about reassignment of an IP address to a different host
  - While the MPTCP connection persists
  - Re-assignment implications not specifically mentioned in the draft so far
- Issue 1: Risk of wrongly accepting a subflow destined to another host
  - Handshake should prevent it, and if not, it is similar to a sequence-number guessing attack
  - No API implications
- Issue 2: Address exposure to MPTCP-unaware applications
  - No issue if there is fate-sharing of the first subflow and the MPTCP connection, as the MPTCP connection will be closed upon address loss
  - Risks of not doing this are already documented and will be left as an implementation choice

➔ One or two sentences will be added in -02 in order to explicitly explain address re-assignment implications on applications

UPCOMING CHANGES
REVIEW OF MICHAEL TUEXEN

- Why is TCP_MULTIPATH_CONNID needed, i. e., REQ4? (Section 5.2/5.3)
  - With MPTCP, there is no single, static (address, port) pair for a connection
  - API should provide a static connection ID for MPTCP-aware applications to track its connections

  ➔ Draft already explains the rational behind this
- Why can one use TCP_MUTLIPATH_REMOVE only after connection setup? (Section 5.3.1)

  ➔ This is a bug in the current draft that will be fixed in -02

UPCOMING CHANGES
REVIEW OF MICHAEL TUEXEN (CONTD.)

- Vague definition of "list of addresses" and "list of pairs of addresses" in Table 1 (Section 5.3.1)
  - According to previous feedback, the MPTCP API is only described in an abstract way
  - The draft does not exactly define the data structures, but only describes the contained information, i. e., a list of IP (v4 and/or v6) addresses (and optionally ports)

  ➔ Already explained, but the authors could add a sentence as additional explanation
- Is it suggested to use the SCTP API for MPTCP? (Section 6.1)
  - Draft text: "API developers MAY wish to integrate SCTP and MPTCP calls to provide a consistent interface to the application."
  - But the draft also explains that not all SCTP functions can be mapped to MPTCP
  - Furthermore, MPTCP stacks may not necessarily support SCTP

  ➔ Proposal: Mention MPTCP/SCTP API integration in the appendix as a potential objective for the extended API

NEXT STEPS

- Two reviews - Thanks!!!
  - Javier Ubillos
  - Michael Tuexen

  ➔ Some clarifications needed that will be addressed in -02
- Still, the document is pretty stable
- Please provide further feedback