MPTCP Protocol – Updates

draft-ietf-mptcp-multiaddressed-03

Alan Ford, Costin Raiciu,

Mark Handley, Olivier Bonaventure

Where we're at

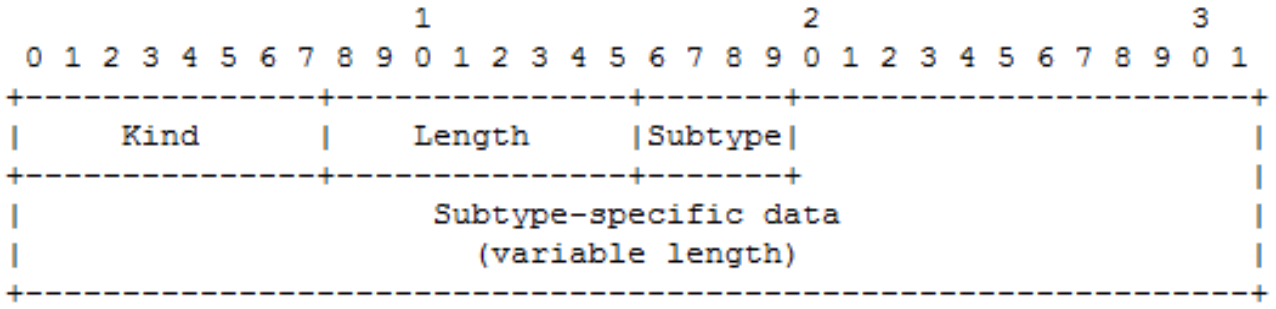
- Stabilising design
- Gathering implementation experience
- Improving document clarity

Changes since -02

- Single option type
- Optimising signals (merging data-level ACK, FIN and sequence number mapping), plus decoupled DATA FIN from subflow FIN
- Confirmed security solution from interim meeting, including crypto agility
- Checksum changes
- More discussion on error handling and heuristics

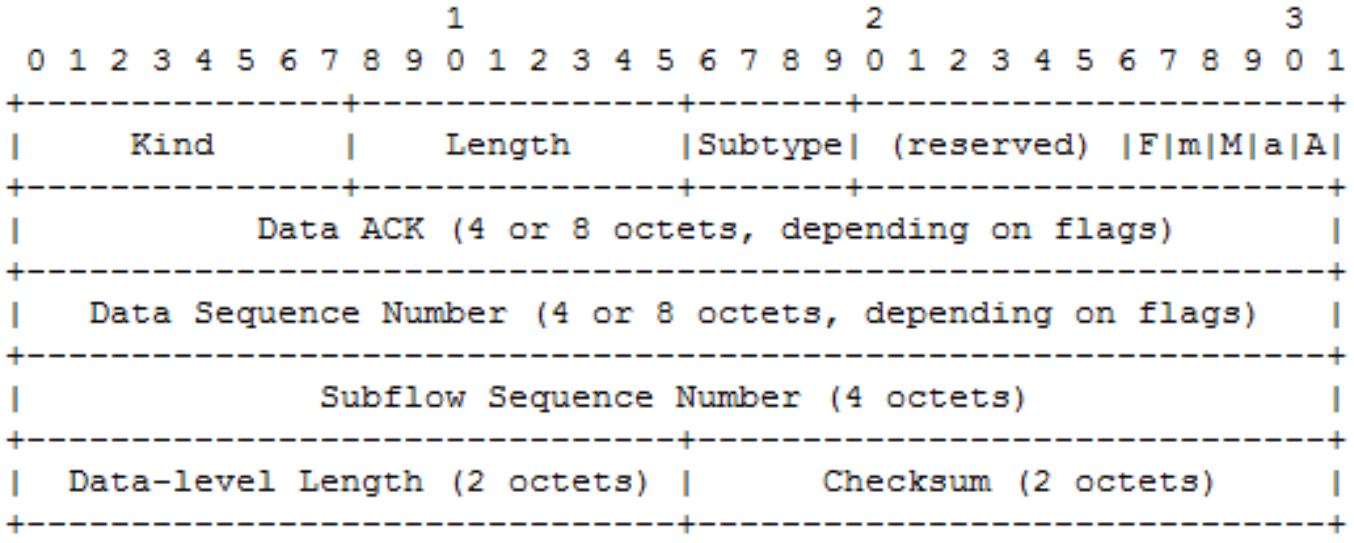
Single Option Type

- All MPTCP messages are now a “subtype” of a single IANA-assigned TCP option type:



Data Sequence Signal

- Merging the old DATA ACK, Data Sequence Mapping, and DATA FIN options into one
- Saves option space when multiple signals are needed



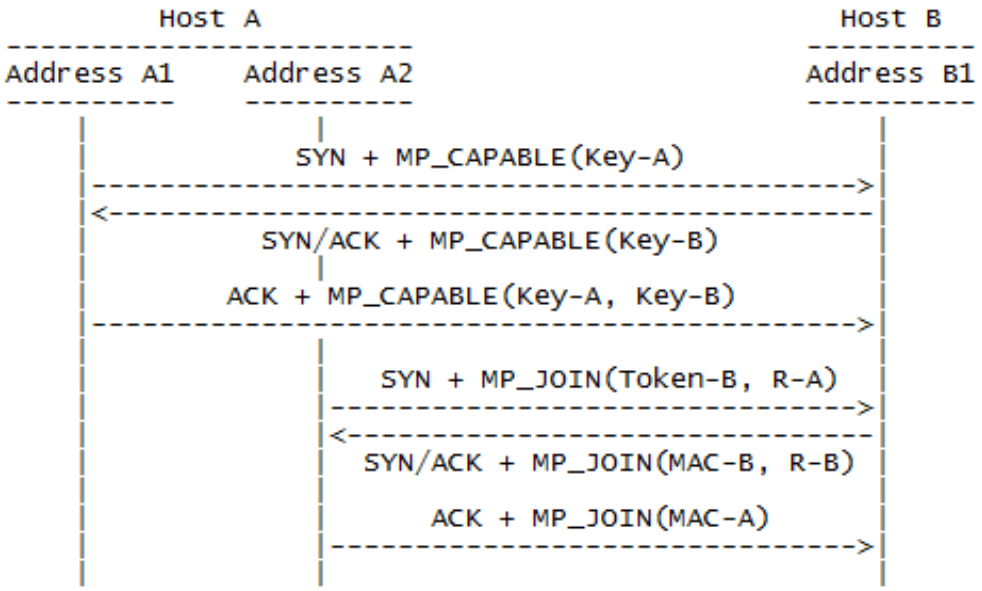
Data Sequence Signal (2)

- Constituent components are optional, and lengths can be variable (32 or 64 bits)
  - Guidance clarified for when to use each length
- DATA FIN is no longer coupled with subflow FIN
  - Ensures all data is received before subflows are closed with FINs
  - Can be sent with a subflow FIN if there is no other outstanding data

Security Solution

- Interim WG meeting in December converged on -02 proposal as being broadly the most appropriate, given the constraints
- Refinements in -03:
  - Reduced to a 3-packet exchange (SYN/SYN-ACK/ACK) with no need for using the payload
  - Allocated bits in MP_CAPABLE to select crypto algorithm (SHA-1 being only current option)

Security Solution (2)



- Key-A, Key-B = 64-bit random numbers
- Token-B = SHA-1(Key-B) [truncated to 32 bits]
- R-A, R-B = 32-bit random nonces
- MAC-A = MAC(Key=(Key-A+Key-B), Msg=(R-A+R-B)) [full 160 bits]
- MAC-B = MAC(Key=(Key-B+Key-A), Msg=(R-B+R-A)) [truncated to 64 bits]

Other Changes

  - Reduces overhead on known safe links
  - Added pseudo-header to include checksum of data-level sequence numbers
- Clarified rules on duplicate ACKs for signalling
  - Don't send more than 2 simultaneously
  - Don't treat MPTCP-only duplicate ACKs as congestion
- Many improvements to document flow, context, terminology, error handling, etc
- Appendix on TCP control block data structures

Heuristics etc

- We don't have a huge amount of heuristics yet
  - Detailed guidance is probably appropriate for a separate document after large-scale experiments
- Receiver and Sender Considerations
  - Buffer sizing, receive window, etc
  - Early discussion – several strategic options
- Initiating subflows
  - Port usage
  - When to do so (discussion and sender-side proposal from Bob Briscoe: opening subflows based on buffered data)
- Handing failures

Where next?

and accessible now – content subject to

- Pleased of reviewers.
- Please! :-)