

draft-gjessing-taps-minset-05

S. Gjessing, M. Welzl

Presenter: Naeem Khademi

neat

TAPS @ IETF 99

Context (reminder)

- Charter item 2:
“Specify the subset of those Transport Services, as identified in item 1, that end systems supporting TAPS will provide, and give guidance on choosing among available mechanisms and protocols. Note that not all the capabilities of IETF Transport protocols need to be exposed as Transport Services.”
- Minimal set: We don't expose everything
 - that would yield a very complicated API
- Based on...
 - draft-ietf-taps-transport-usage-06
 - draft-ietf-taps-transport-usage-udp-03

Main change since -04

- This now begins with the minset
 - Textual description; **abstract API**
- Procedure to *derive* minset moved to appendix
 - Categorization (Functional, Optimizing, Automatable)
 - Reduction (removing transport features that do not require application-specific knowledge or prohibit falling back to TCP)
 - Discussion (how to handle peculiarities such as: sending messages, receiving a stream; not exposing multi-streaming but having a per-stream priority)

High-level view of miniset

- Need to begin with **flow creation** and **flow grouping**
 - And most configuration features only configure a group (e.g., can't configure timeout for a stream of an SCTP association)
 - Should use them early, ideally before connecting
- Need to allow **send-before-connect**
 - Abstract => could also be a parameter to connect
 - Specify “idempotent” to allow TFO
- **Limited connect / listen / close / abort semantics**
 - Support UDP, support streams
- At minimum: **send messages, receive bytestream**
- **Sender running dry** (related to SCTP & TCP LOWAT)
- Some queries related to **packet sizes**

minset abstract API

- **CREATE** (flow-group-id)
- **CONFIGURE_TIMEOUT** (flow-group-id [timeout] [peer_timeout] [retrans_notify])
- **CONFIGURE_URGENCY** (flow-group-id [scheduler] [capacity_profile] [low_watermark])
- **CONFIGURE_PRIORITY** (flow-id priority)
- **CONFIGURE_CHECKSUM** (flow-id [send [send_length]] [receive [receive_length]])
- **CONNECT** (flow-id dst_addr), **LISTEN** (flow-id)
- **CLOSE** (flow-id), **ABORT** (flow-id)
- **SEND_FRAME** (flow-id frame [reliability] [ordered] [bundle] [delack] [fragment] [idempotent])
- **RECEIVE_FRAME** (flow-id buffer)

Next version: will add this to initial configuration too !
(thanks to socket-intents folks)

miniset abstract API, cont'd

- **NOTIFICATIONS**

- Excessive Retransmissions
- ICMP Arrival (parameter: ICMP message); ECN Arrival
- Timeout (parameter: s seconds)
- Close; Abort
- Drain
- Path Change (parameter: path identifier)
- Send Failure

- **QUERY_PROPERTIES**

- maximum frame size that may be sent without fragmentation;
maximum transport frame size that can be sent; maximum transport
frame size that can be received; maximum amount of data that can
possibly be sent before or during connection establishment

Conclusion

- A TAPS system can do more than this
 - E.g., get input for “automatizing” decisions
- ... but TAPS system designers need to understand trade-offs
 - Don’t expose limiting functional features!
- minset document clarifies
- minset is the list of things to expose everywhere
 - or they can never be used!
 - Higher-level APIs AND lower-level APIs

Thank you

- Questions?

... we have one:

- Should this be adopted as a working group draft?