

draft-gjessing-taps-minset-04

S. Gjessing, M. Welzl

neat

TAPS @ IETF 98

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
- Mostly based on
draft-ietf-taps-transport-usage-03

Construction of a minimal set of transport features:

- 1. Categorization:** The transport features in the “usage” draft are categorized.
- 2. Reduction:** Remove all transport features that do not require application-specific knowledge or cannot be implemented with TCP.
- 3. Discussion:** The resulting list shows a number of peculiarities that are discussed
- 4. Construction of minimal set:** Based on the **reduced** set and the **discussion** of the transport features therein

Step 1 Categorization (reminder)

- Functional (e.g. unordered message delivery)
 - provide functionality that cannot be used without the application knowing about them, or else they violate assumptions that might cause the application to fail
- Optimizing (e.g. change DSCP)
 - Can't use without application-specific knowledge, but won't make an application "fail" (note: best effort)
- Automatable (e.g. Set Primary Path)
 - Could be used without application-specific knowledge

Step 2 Reduction

(App specific knowledge and fall-back to TCP)

- **CONNECTION Related Transport Features**
 - ESTABLISHMENT:
 - Connect + 4 other features
 - AVAILABILITY:
 - Listen
 - Specify which chunk types must always be authenticated
 - MAINTENANCE:
 - Disable Nagle + 17 other features
 - TERMINATION:
 - Close . . . + Abort . . . + Timeout

Step 2 Reduction cont'd

(App specific knowledge and fall-back to TCP)

- **DATA Transfer Related Transport Features**
 - SENDING DATA
 - Unreliably transfer a message + 7 other features
 - RECEIVING DATA
 - Receive data (with no message delineation)
 - Information about partial message arrival
 - ERRORS
 - Notification of send failures + 2 other features

Step 3: Discussion

- Sending Messages, Receiving Bytes
 - To be discussed after this presentation
- Stream Schedulers Without Streams
 - To be discussed after this presentation
(our draft explains relationship to connection setup, teardown, and priorities)
- Early Data Transmission
 - To be discussed after this presentation

Step 3: Discussion cont'd

- Sender Running Dry
 - SCTP has “Sender Dry” notification: special case of TCP’s unspec’d “TCP_NOTSENT_LOWAT” option where watermark = 0
 - Therefore suggest to offer a general, common method
- Capacity Profile: we suggest to generalize...
 - Disable Nagle algorithm (time-, not packet granularity)
 - Enable and configure a “Low Extra Delay Background Transfer”
 - Specify DSCP field

Step 3: Discussion cont'd

- Security
 - SCTP and TCP offer authentication
 - TCP (RFC 5925): todo
 - SCTP: specifying which SCTP chunks to authenticate creates protocol dependency
 - Suggest to specify:
Authenticate control information / data / both
TCP: always authenticates everything
 - Cookie lifetime: TBD
SCTP: client configures; TCP: server configures

Step 3: Discussion cont'd

- Packet size
 - “Specify DF field” (UDP(-Lite)): the only transport feature related to packet size
 - Necessary for apps doing PMTUD
 - A TAPS system should probably avoid automatically switching paths, and inform the application about any unavoidable path changes when an app uses this feature
 - Suggest to offer means to
 - query maximum unfragmented frame size
 - query maximum transport frame size

Step 4: Construction -- the Minimal Set of Transport Features

- The minimal set is based on the transport features from step 2 (reduction) and step 3 (the discussions)
- Flow Creation, Connection and Termination
- Flow Group Configuration
- Flow Configuration
- Data Transfer

Conclusion

- Should this be adopted as working group draft?

Thank you

Questions?