TAPS ARCH & API open topics

Philipp S. Tiesel

IETF 105 - Montréal, Quebec, Canada

Listen filtering behavior (for TLS connections)? #338

The Listener object delivers connections, but doesn't provide a particular way to reject connections.

- The text suggests ReceivedConnection delivers a connection once the TLS server handshake is complete.

- Some implementation may want to rate limit, or modify/ block inbound connections.

Listen filtering behavior (for TLS connections)? #338

The Listener object delivers connections, but doesn't provide a particular way to reject connections.

Resolutions:

a. Match `ReceivedConnection` events with listen calls to allow back-pressure (analogous to `receive`)

b. Add additional events to control back-pressure. Should they be TLS specific?

c. Add Properties to control accept behavior?

Add Unidirectional Streams for Multicast Source and Sink support #150

We have a selection property to create unidirectional streams in the main document, but no way to query whether a connection is uni-directional.

- Is the his sufficient for Multicast?

API needs a way to handle "STARTTLS" #249

SMTP and XMPP (among other protocols, probably) have "STARTTLS" semantics, where a TLS layer can be inserted on top of an already-existing (unencrypted) connection sometime after some application-layer protocol negotiation has already happened.

- Is the current framer architecture sufficient to realise this?

Differentiating bad selection configuration from connection issues #307

Per the interface draft, an InitiateError gets thrown both when the set of (protocol) candidates is empty as well as when it is not possible to establish a connection.

Resolutions:

a. Add an `UnsatisfiableSelectionConfiguration` event

b. Add additional information the the `InitiationError` event

c. Do nothing and close the Issue

Consider API that takes padding policy as input #334

Since padding is (increasingly) supported by transports, this should be something a TAPS implementation provides.

Use cases for applications to control transport level padding include:

- Amplification prevention
- Traffic obfuscation

Consider API that takes padding policy as input #334

Since padding is (increasingly) supported by transports, this should be something a TAPS implementation provides.

Resolutions:

a. Add Selection and Message Properties to control padding.

b. Do nothing and ignore padding.

Should `send` return a Message Context? #336

PR #321 removed the unspecified `mgs_ref` and made `send` a `messageContext` object instead to to make matching of send error events and reply messages to their originating message easier.

1. There are concerns that an often unused return value is harmful in many languages.

2. Send errors are are currently 1:1 matched with `send` calls, not (partial) messages.

Should `send` return a Message Context? #336

1. There are concerns that an often unused return value is harmful in many languages.

Resolutions:

a. Remove the return value as `messageContext` can be created and passed explicitly.

b. Leave the return value as a convenience and make it optional.

Should `send` return a Message Context? #336

2. Send Errors are are currently 1:1 matched with `send` calls, not (partial) messages.
   - They can not just pass the message context from the send
   - Applications that do byte-wise partial send may get thousands of send errors

Resolutions:

a. Match Send Errors and Messages (instead of sends)

b. Separate Error and Message context

c. Allow to query original message context from the message context provided by the error (analogous to `GetOriginalRequest`)

How to handle Protocol stacks that are not equivalent #305

Per the architecture draft, only protocol stacks that are equivalent can be safely raced. What should happen if selection properties result in a candidate set that includes protocol stacks that are not intuitively equivalent?

Resolutions:

a. Define Protocol Stack Equivalence more rigorously.

b. Generate some kind of Error Event for configurations with unlike protocol stack candidates

c. Do nothing and close the issue.

Draft should point at existing implementations #145

It might be useful to reference existing implementations (e.g., Apple NWConnections and PyTAPS) and similar systems (e.g., NEAT, PostSockets, SocketIntents) that where used as input to the TAPS design.

Resolutions:

a. Add section to appendix

b. Add notes to acknowledgements

c. Do nothing and close the issue

Implementation should separate out protocol-specific bits #248

Right now, there are protocol specific examples mixed into the generic flow description; then, later, there is a section that has some description for protocol specific considerations.

Possible re-organizations:

a. Per-protocol mappings for each API call + per-protocol specific options. This is a guide to each protocol, such as a section for UDP and how each call is implemented for UDP, and what options UDP offers specifically.
   General discussion about Pre-Establishment, Establishment, Data Transfer, etc, that has less information about specific protocols.

b. Structure everything with the general topics (Pre-Establishment, Establishment, Data Transfer, etc), and have a rigorous list of each protocol's mappings at the end of each section. So, for Establishment, we describe racing, etc; and then go through and explain what Initiate() does for TCP, UDP, UDP Lite, QUIC, SCTP.