EVENT client message

Proposal: 0004
Authors: Alexander Færøy, David Goulet
Status: Awaiting implementation

Implementation: TBD

Introduction

This proposal details the addition of a client event message that can be sent back to the main process. Theses messages can be anything that reflects the ongoing status of the PT after bootstrap.

Motivation

There are various reasons for having a way for a client PT to send back event messages. One of the main motivation comes from the Tor Browser and UX team that want to have a way to better diagnose the PT bridge connection state and, if needed, provide visual feedback to the users if any action is required.

Proposed solution

This proposal introduces to the pt-1.0.txt specification an EVENT message that can be sent back to the main process with a specific format. We'll be also proposing an event message type named "CONNECTION" that reports back, using the EVENT message, the state of the PT connection to the bridge.

Design

We propose to add section 3.3.4 to the pt-1_0.txt file with the following:

```
3.3.4. Pluggable Transport Client Event Messages

   These messages are for a client PT to be able to signal back to the parent
   process via messages to whatever means is used to communicate (stdout,
   ExtORPort, ...) that an event has occurred client side.

   An event is anything that the PT could want to report back that can be
used
   as feedback of what is going on in the PT so better decision can be made
by
```

the parent process and improve UX.

The format of the message is kept simple to be extendable:

EVENT <Type> <OptArgs>

The <Type> is a subevent describing what is the event type providing more context for what the event is for or coming from.

The following section described different event <Type> that this message can use.

3.3.4.1 CONNECTION type.

A client PT connects to a bridge so this message is used to report back to the parent process the status of that connection.

CONNECTION <transport> <address> <port>

The "CONNECTION" message is used to signal that a specific PT <transport> connection has been initiated towards a bridge at <address> and <port>.

Example:

EVENT CONNECTION obfs4 198.51.100.1 19999

CONNECTION-ERROR <transport> <ErrorType> [<ErrorMessage>]

The "CONNECTION-ERROR" message is used to signal that a specific PT <transport> connection had an error occurred after being launched. This means that in order to have such a message, first the PT must have sent a "CONNECTION" event.

The <ErrorType> is a Keyword indicating the context of the error. For the connection type, the following values are recognized:

CONNECTION_IOERROR -- The TCP connection got some IO errors.
CONNECTION_REFUSED -- The TCP connection was refused.
CONNECTION_TIMEOUT -- The TCP connection timed out.

The <ErrorMessage> is an optional human readable string that can describe how the connection failed adding semantic to the error type.

Example:

EVENT CONNECTION-ERROR obfs3 CONNECTION_TIMEOUT Connection timeout after 30 sec

```
      EVENT CONNECTION-ERROR obfs4 CONNECTION_IOERROR Connection was reset
      EVENT CONNECTION-ERROR obfs4 CONNECTION_REFUSED

CONNECTION-SUCCESS <transport>

   The "CONNECTION-SUCCESS" message is used to signal that a specific PT
   <transport> has successfully established a connection to the bridge and
   thus is ready to be used.

   A "CONNECTION" message must have been sent before this. If a
   "CONNECTION-ERROR" occured, the parent process should not expect to
   receive this message.

   This message only indicate that the transport has now connected to the
   bridge and can be used by the parent process to relay traffic.

   Example:

      EVENT CONNECTION-DONE obfs4
```

Effect on API Compatibility

We are extending the API so no backward breakage.

Effect on IPC Compatibility

By adding a new message that is passed through the IPC used, the main application could receive unknown messages and thus should ignore them if unrecognized.