CLOSED state synchronization in H2

The problem

- H2 doesn't have provisions for acking a closed stream, leading to client and server not agreeing on when a stream is closed
- the problem really happens when transitioning from HREM to CLOSED
- but it may also happen when transitioning from HLOC to CLOSED (e.g if the peer already closed but its ES was not yet received, while data are still in flight to it)

Effects

- truncated responses when closing a connection after closing the last stream (server) (WU responses hitting a closed TCP socket resulting in RST being sent back and cancelling unparsed data in the TCP buffers)
  - https://www.mail-archive.com/haproxy@formilux.org/msg32456.html
- confused streamm dependencies when the server receives a PRIORITY mentioning a stream which is closed for it but not yet for the client.
  - https://github.com/httpwg/http-extensions/issues/751

Proposal

- use a PING frame at the end of closed streams when there is the need to be certain about the close. This is 100% backwards compatible with deployed implementations since it only uses mandatory protocol elements.

- the side interested in keep knowledge of the stream until the peer sees the close sends a PING frame after sending ES (probably useless on RST). This frame is composed of :

  32 MSB set to 0xDEAD1DAC (at choice)

  32 LSB set to the stream ID to ACK

- the sender then changes the stream's state to LAST_ACK instead of CLOSED, or HLOC1 instead of HLOC

- this is sent over TCP, there is no reordering, so the peer will ACK this once it has consumed all the data from the socket buffers. At this point the data are safe from any network activity.

- the sender (mostly server) receives the ACK and can either switch the stream from HLOC1 to HLOC2 or free it.

Notes

- it seems really pointless to implement extra states, these are only useful on a diagram

- an implementation shall instead keep one bit to remember if the ES flag was acked or not.

- there is no need for keeping any information related to ACKs sent as they are stateless.

- may lead to spurious false positives on protocol validation tools like h2spec or vtest.

Intent to implement

- in haproxy for graceful shutdown of connections. Not yet done, hopefully before 2.0, no showstopper estimated by now, only requires time.