Known Startup State for HTTPS TLS Negotiation

HTTPbis WG, IETF 86, Berlin
31 July, 2013

Osama Mazahir
Matthew Cox
Gabriel Montenegro
(Microsoft)
Review: Unknown Startup State

• Needless complexity if the protocol does not start at a known state at both client/server

• Best to not allow the protocol to “overstep” itself
  • “overstep”: send more than you have credit for, open more streams than the receiver allows for, etc.

• Let’s not abandon protocol correctness in the quest for speed (besides, no need to)

• Can lead to more overstepping with future extensions with unpredictable consequences

• Solved for HTTP Upgrade case
  • addressed in -04 by HTTP2-Settings header being required with the Upgrade request.
HTTPS – TLS Negotiation case

• The client MUST send a SETTINGS frame once TLS negotiation is complete.
• But – the server does not have opportunity to send initial preferences before receiving client frames.
• The client could open too many streams or send too much data to the server.
• Solution: the server send its settings during TLS.
• Note: The client sends nothing within TLS handshake
  • simply uses SETTINGS frame as usual upon start of the HTTP/2 session
Alternatives

• **Agree on Defaults.**
  • Tried this: too much divergence between positions.
  • Hard to pick a default appropriate for the future.

• **Wait one RTT.**
  • The client sends its SETTINGS and w-a-i-t-s... for the server SETTINGS before initiating any real operation.
  • Not an alternative because of latency.

• **OOB methods for client to fetch server SETTINGS:**
  • DNS
  • Well-known URI, Webfinger, etc
    These add too much latency and cannot be relied upon to always be there or available.

• **Send SETTINGS within TLS.** Most dependable and straightforward.
HTTPbis – TLS Liaison

• For HTTPS, need a capability to convey SETTINGS within the TLS handshake.
• Recommendation: HTTPbis to request the TLS WG to address this requirement.