SOCKS Protocol Version 6

draft-olteanu-intarea-socks-6-00

Vladimir Olteanu, Dragoș Niculescu
University Politehnica of Bucharest

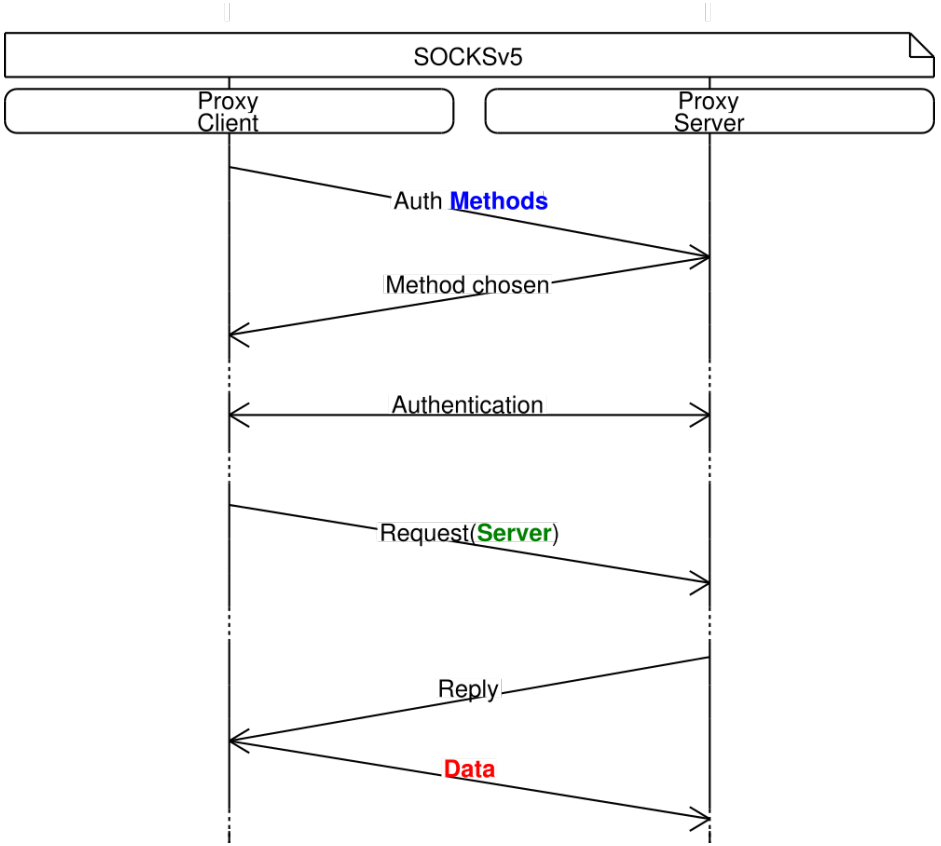
Motivation

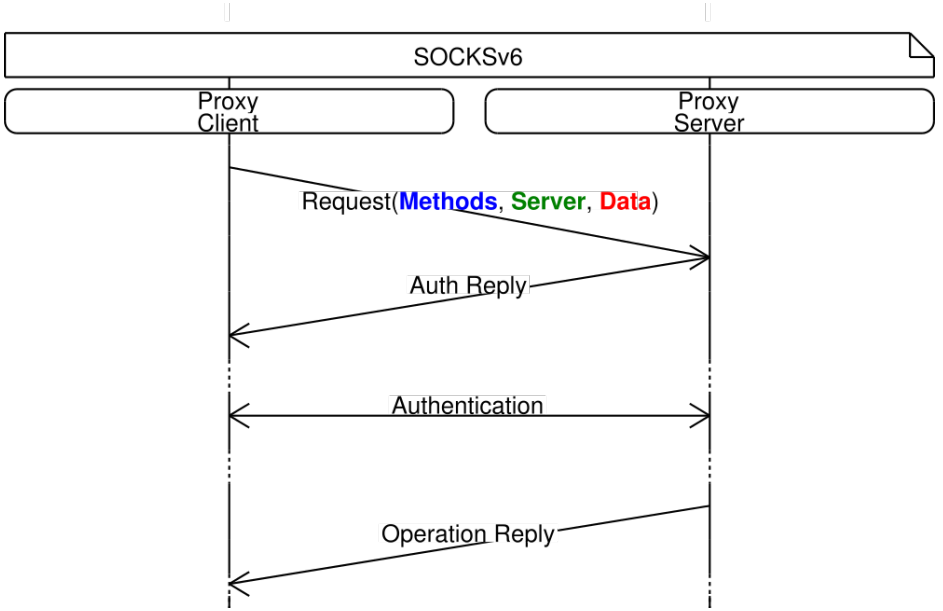
- SOCKSv5 makes liberal use of round trips
  - Authentication method negotiation
  - Authentication
  - Remote connection establishment
- 0-RTT authentication possible after pre-negotiation
- Hot use case: “Bond” 3G/4G/LTE and WiFi using MPTCP
  - Little to no MPTCP support on the server side
  - Use proxy to convert to regular TCP
  - Mobile networks have high latency

Improvements over v5

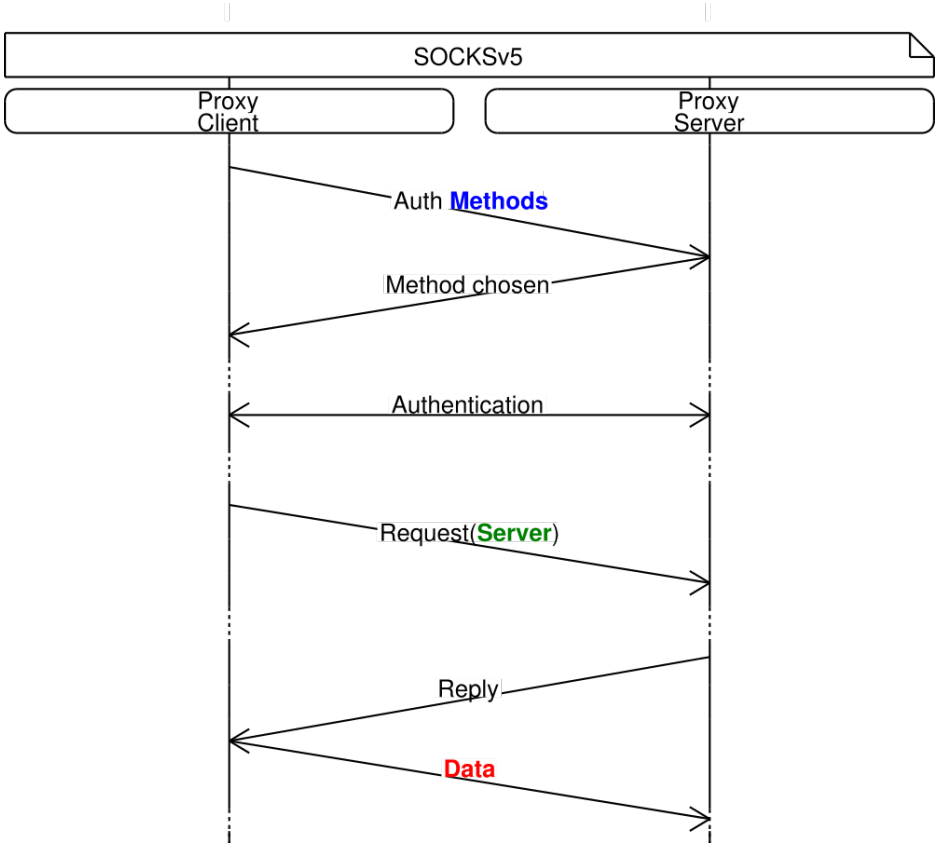
- Client sends as much information as possible upfront
  - Optimistic, doesn’t wait for authentication to conclude
  - Method advertisement, server address, some application data
- Client can specify if it wants TFO on the proxy-server leg
- Extensible: TCP-like options
- 0-RTT authentication support via options

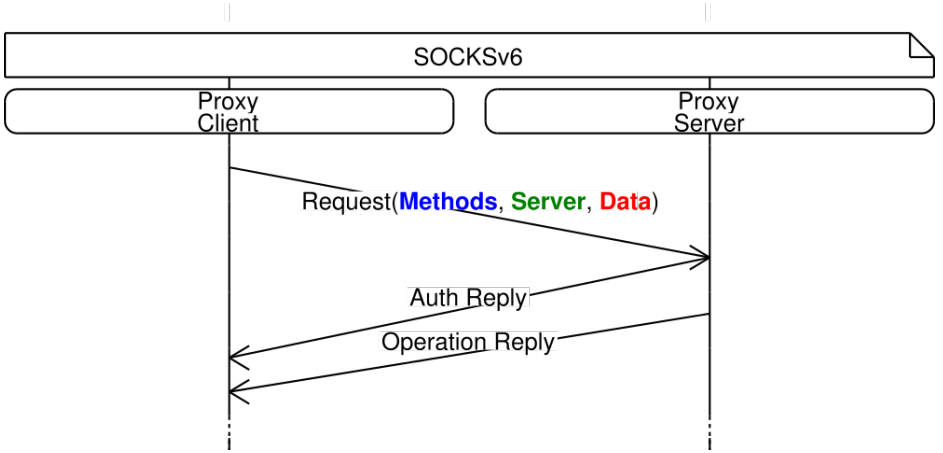
SOCKSv5 vs. SOCKSv6 [1/2]





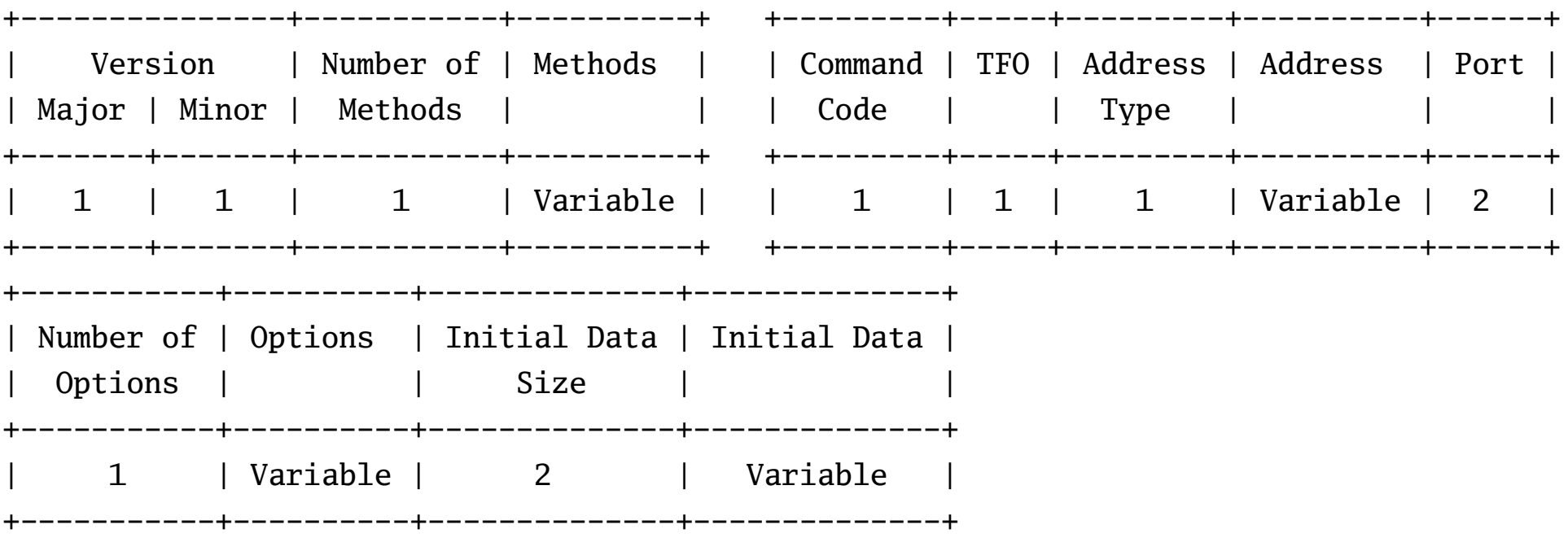
SOCKSv5 vs. SOCKSv6 [2/2]





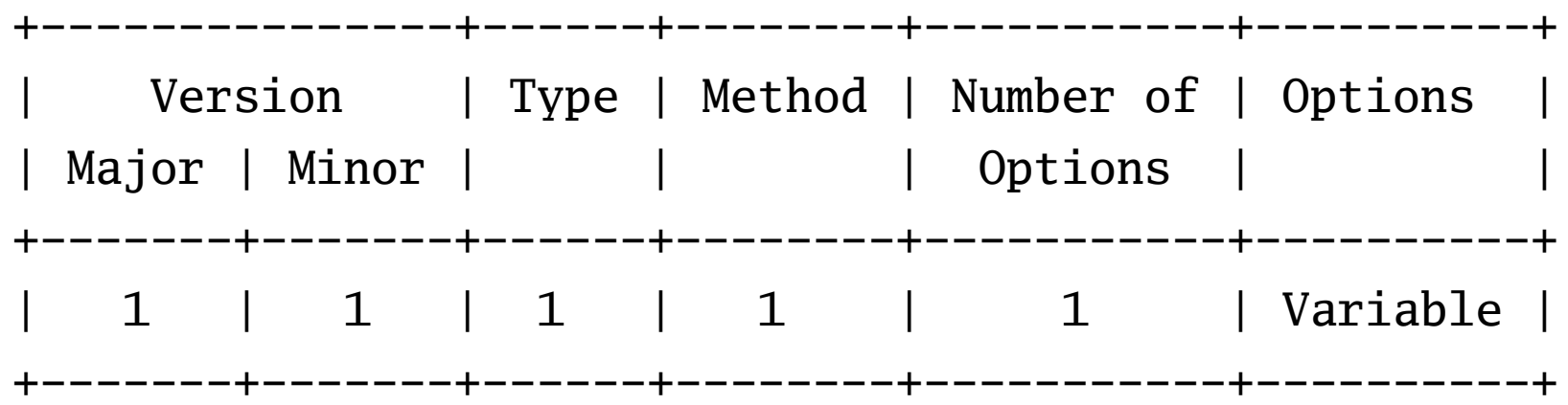
- Can include authentication data in the request on subsequent connections

SOCKSv6 Request



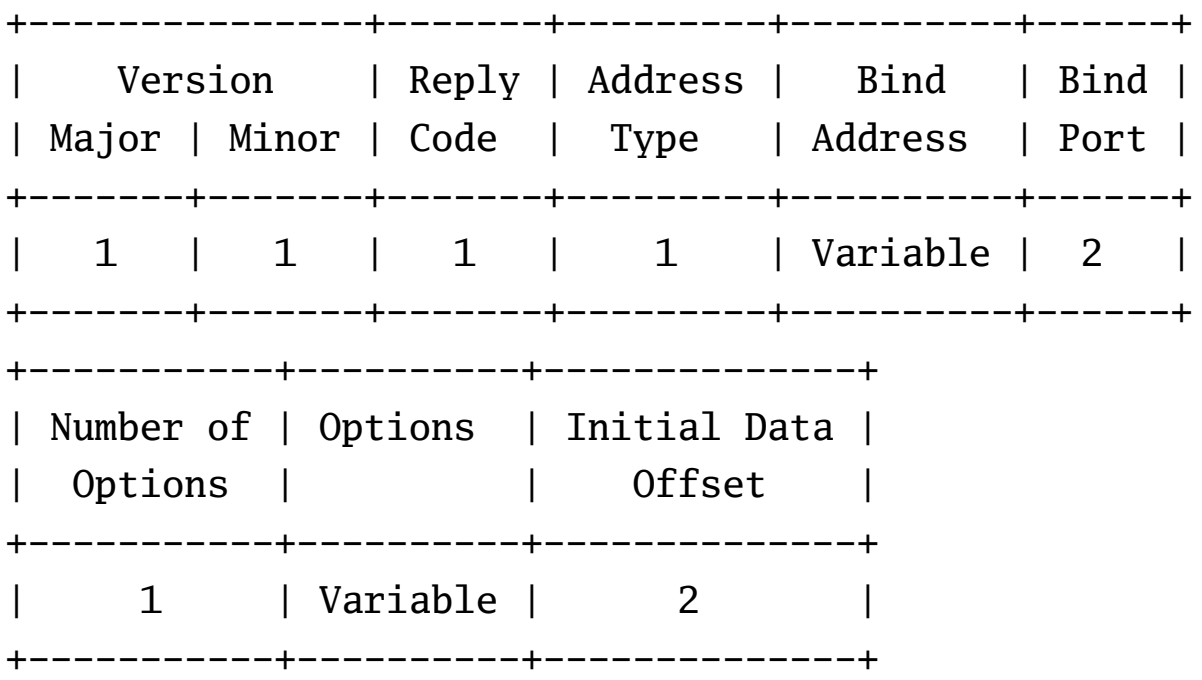
- Includes auth. method advertisement
- Includes initial data
- Options in TLV format
  - May include authentication data

SOCKSv6 Authentication Reply



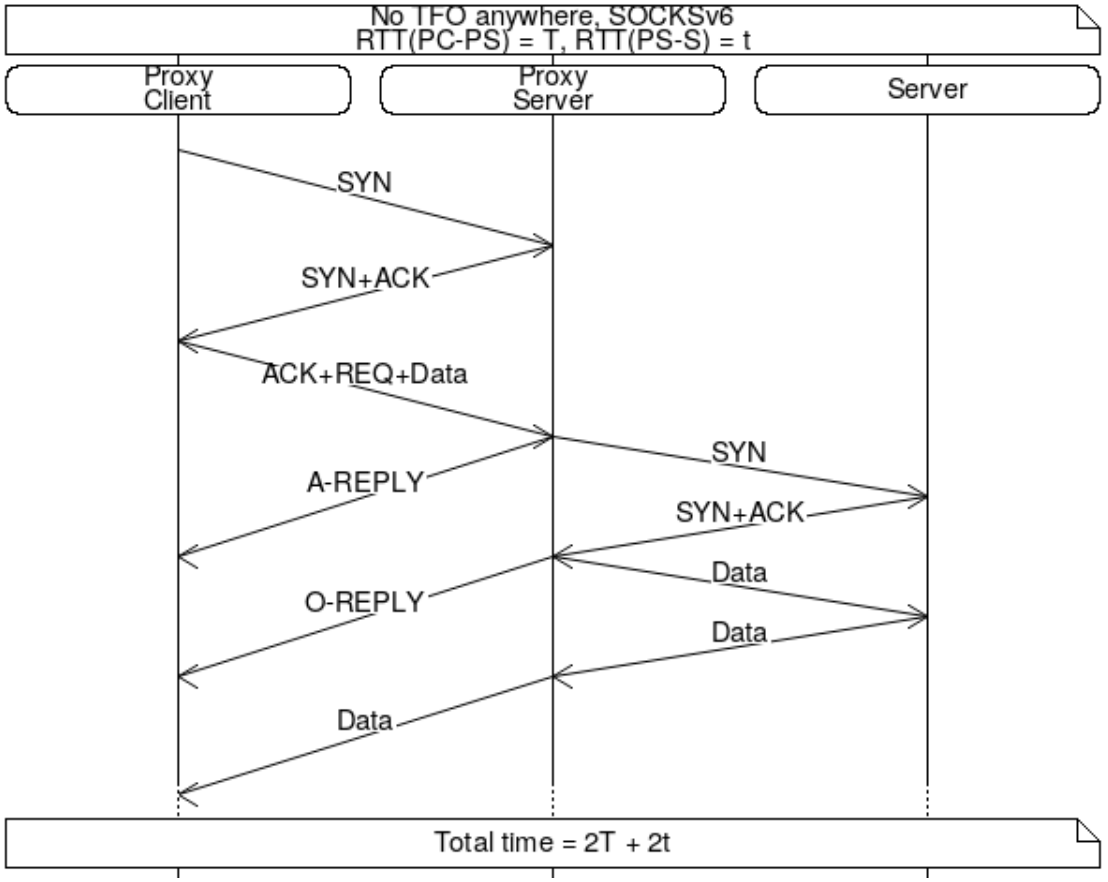
- Informs client whether more authentication is needed or not
  - If 0-RTT authentication failed: selects which authentication method to use
  - If 0-RTT authentication succeeded: informs client which method was used

SOCKSv6 Operation Reply



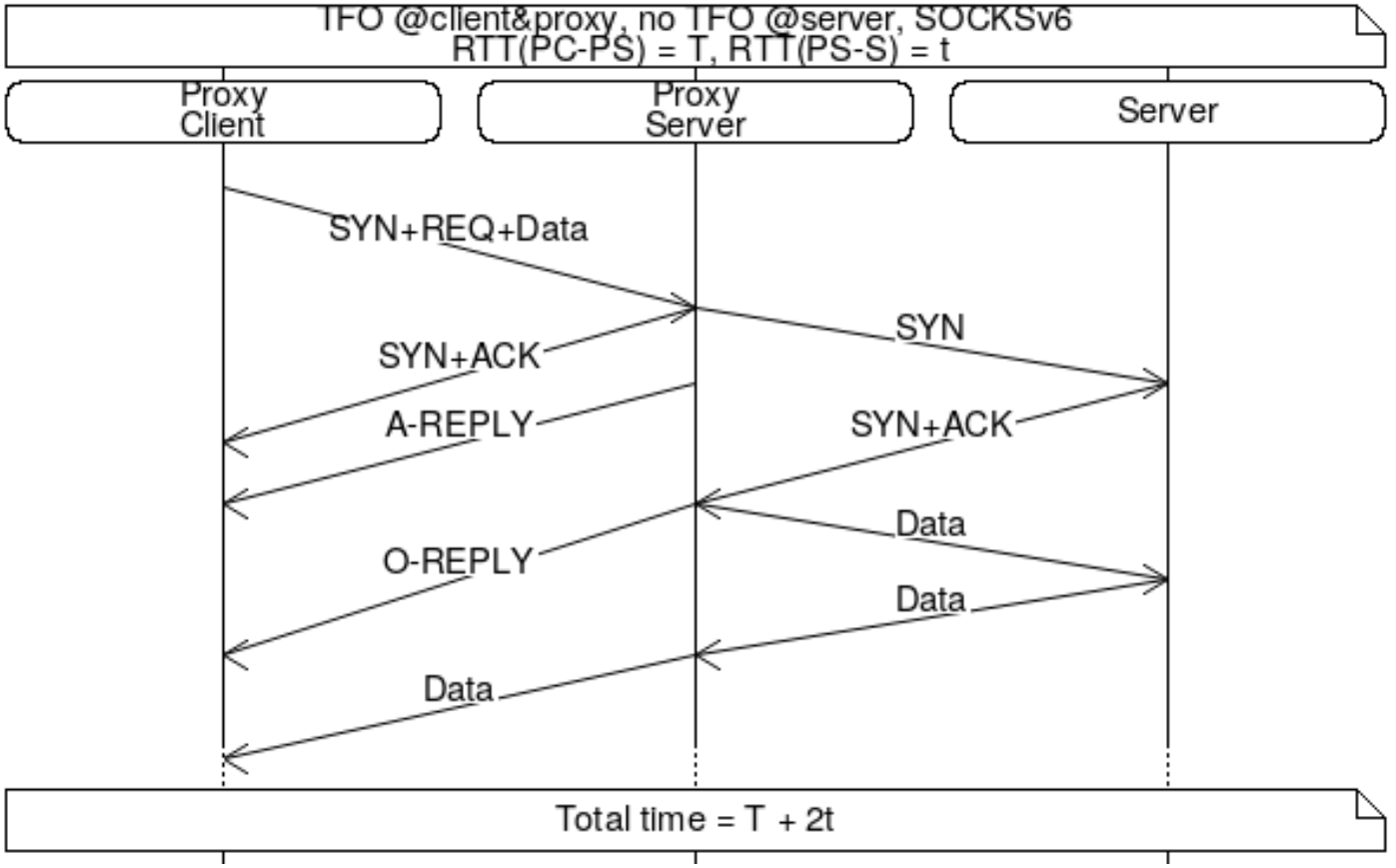
- Reply code indicates whether the connection was successful or not (and why: RST, timeout, etc.)
- Initial data offset lets the proxy avoid buffering data while the client authenticates

SOCKSv6 in action: no TFO anywhere



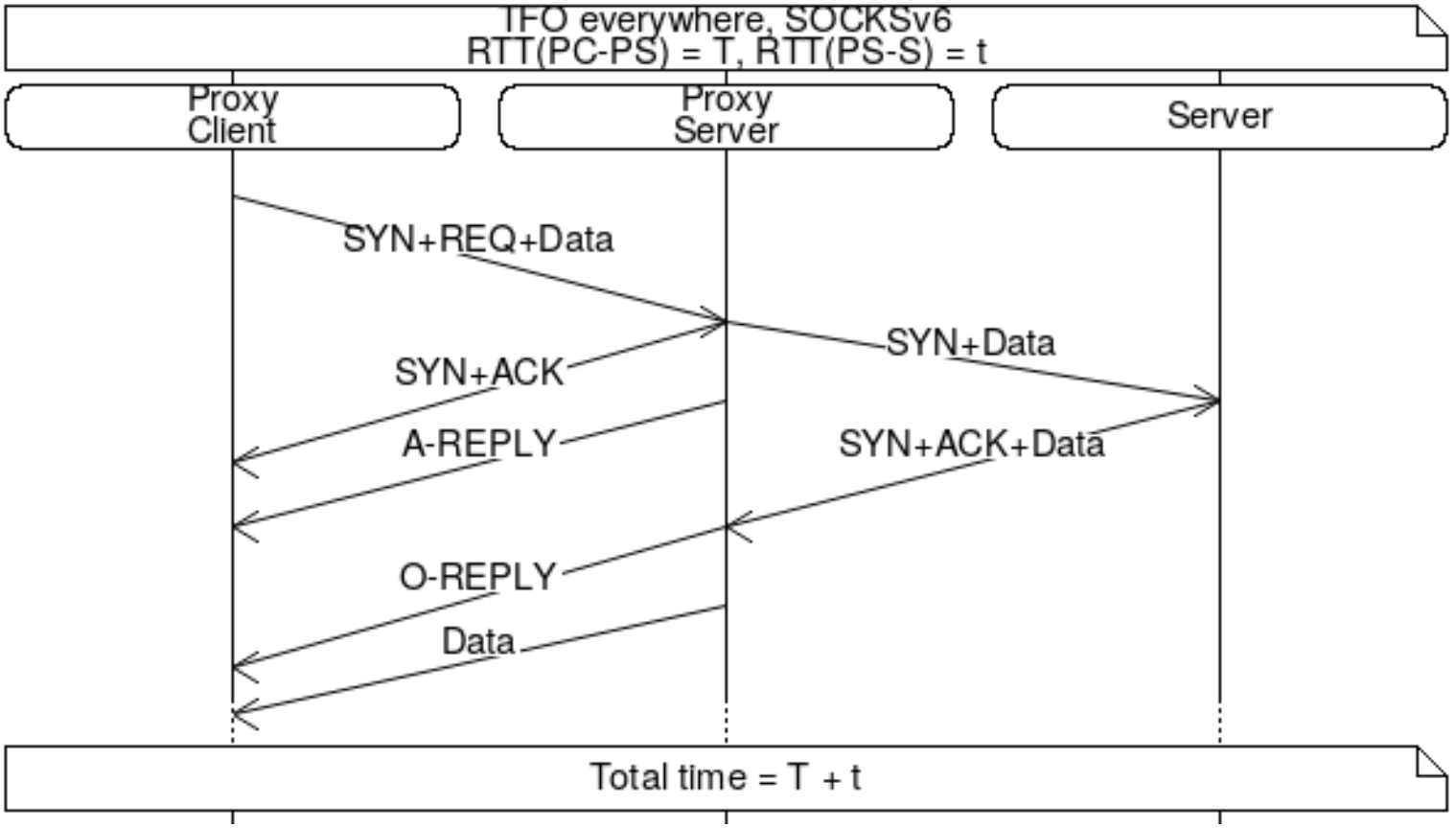
- Data reply in 2 RTTs
  - No worse than vanilla TCP

SOCKSv6 in action: TFO on proxy-client leg



- Data reply in 1 end-to-end RTT + 1 proxy-to-server RTT
  - **Negative overhead**: We save 1 client-to-proxy RTT, assuming the proxy is on path
  - Highly advantageous for mobile networks, where layer 2 has high delay

SOCKSv6 in action: TFO everywhere



- Data reply in 1 RTT
  - Same as when contacting the server directly

Multiple proxies

- Can run SOCKS over SOCKS (can be stacked indefinitely)
  - Client is responsible for authenticating with each proxy
  - Data reply in 2 RTTs w/o any TFO, 1 RTT with TFO on all legs
- ...or just configure the first proxy to go via a second proxy

Implementation

- Early prototype (some differences from draft)
  - Message library: https://github.com/45G/socks105
  - Proxifier + proxy: https://github.com/45G/shadowsocks-libev

Comparison to MPTCP-PM and 0-RTT TCP converters

- draft-boucadair-mptcp-plain-mode-10
- draft-bonaventure-mptcp-converters-00

- Similarity: No control data aside from initial exchange
- Different starting point: purely layer 5 protocol
  - All signaling is done using TCP data
  - TFO/SYN data not required, but highly beneficial
  - Midllebox doesn’t kill TCP => middlebox doesn’t kill SOCKS