SOCKS Protocol Version 6 (update)

draft-olteanu-intarea-socks-6-01

Vladimir Olteanu, Dragoș Niculescu
University Politehnica of Bucharest

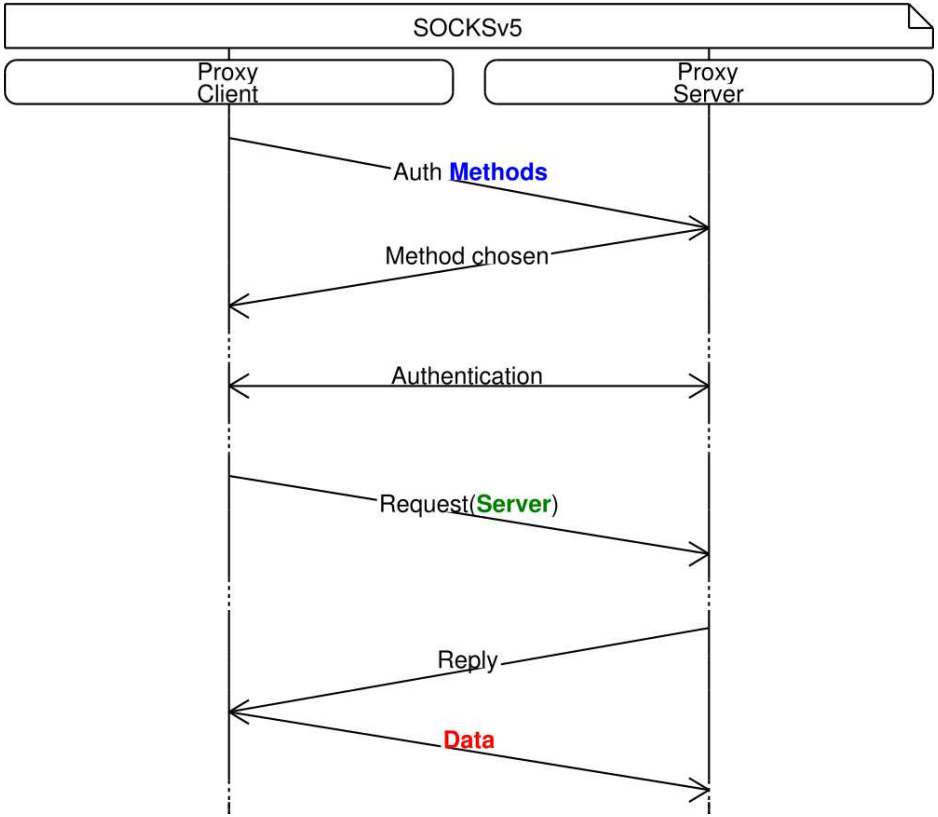
Motivation

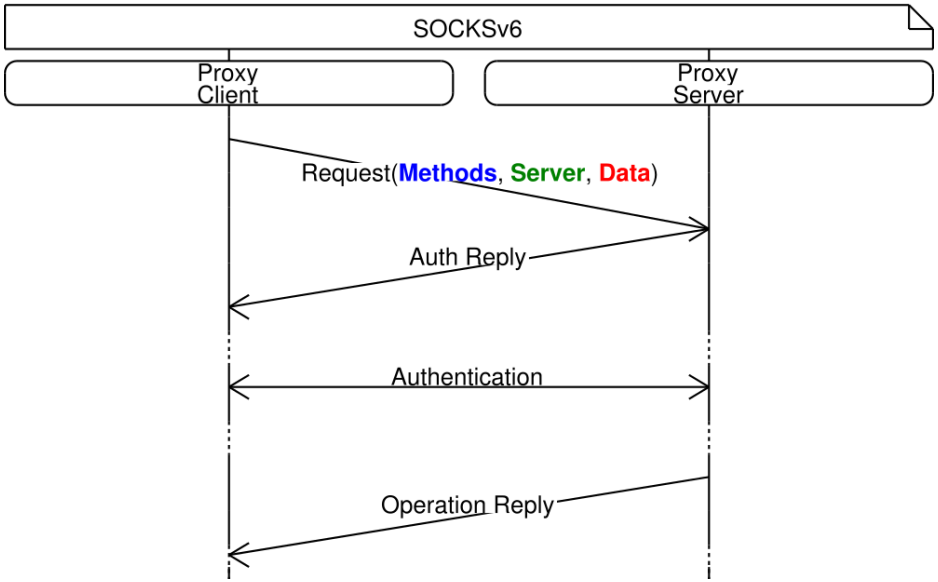
- SOCKSv5 makes liberal use of round trips
  - Authentication method negotiation
  - Authentication
  - Remote connection establishment
- 0-RTT authentication possible after pre-negotiation
- Hot use case: “Bond” 3G/4G/LTE and WiFi using MPTCP
  - Little to no MPTCP support on the server side
  - Use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

Improvements over v5

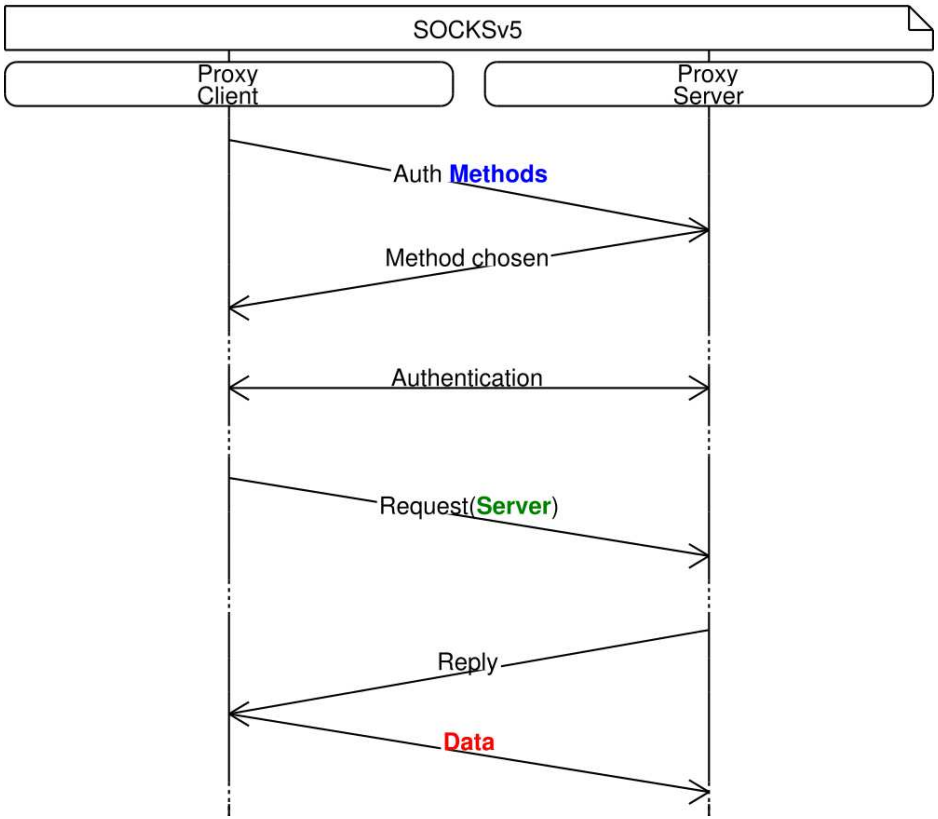
- Client sends as much information as possible upfront
  - Optimistic, doesn’t wait for authentication to conclude
  - Method advertisement, server address, some application data
- Client can specify if it wants TFO on the proxy-server leg
- Extensible: TCP-like options
- 0-RTT authentication support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

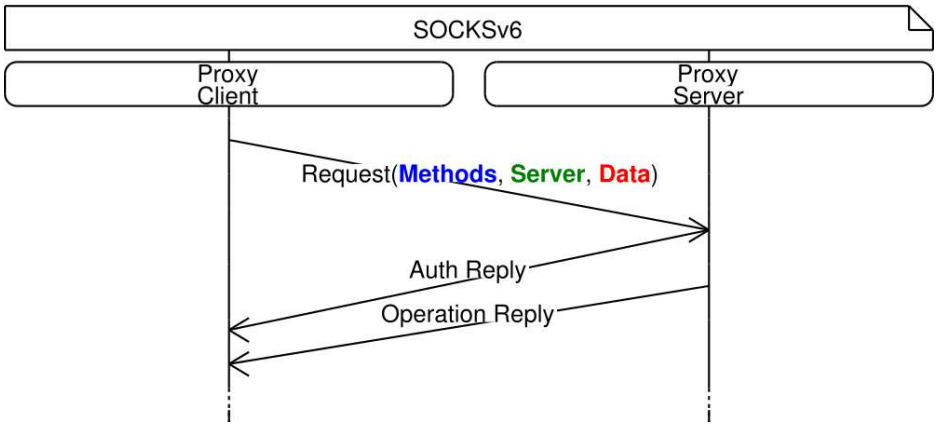
SOCKSv5 vs. SOCKSv6 [1/2]





SOCKSv5 vs. SOCKSv6 [2/2]





- Can include authentication data in the request on subsequent connections

Security

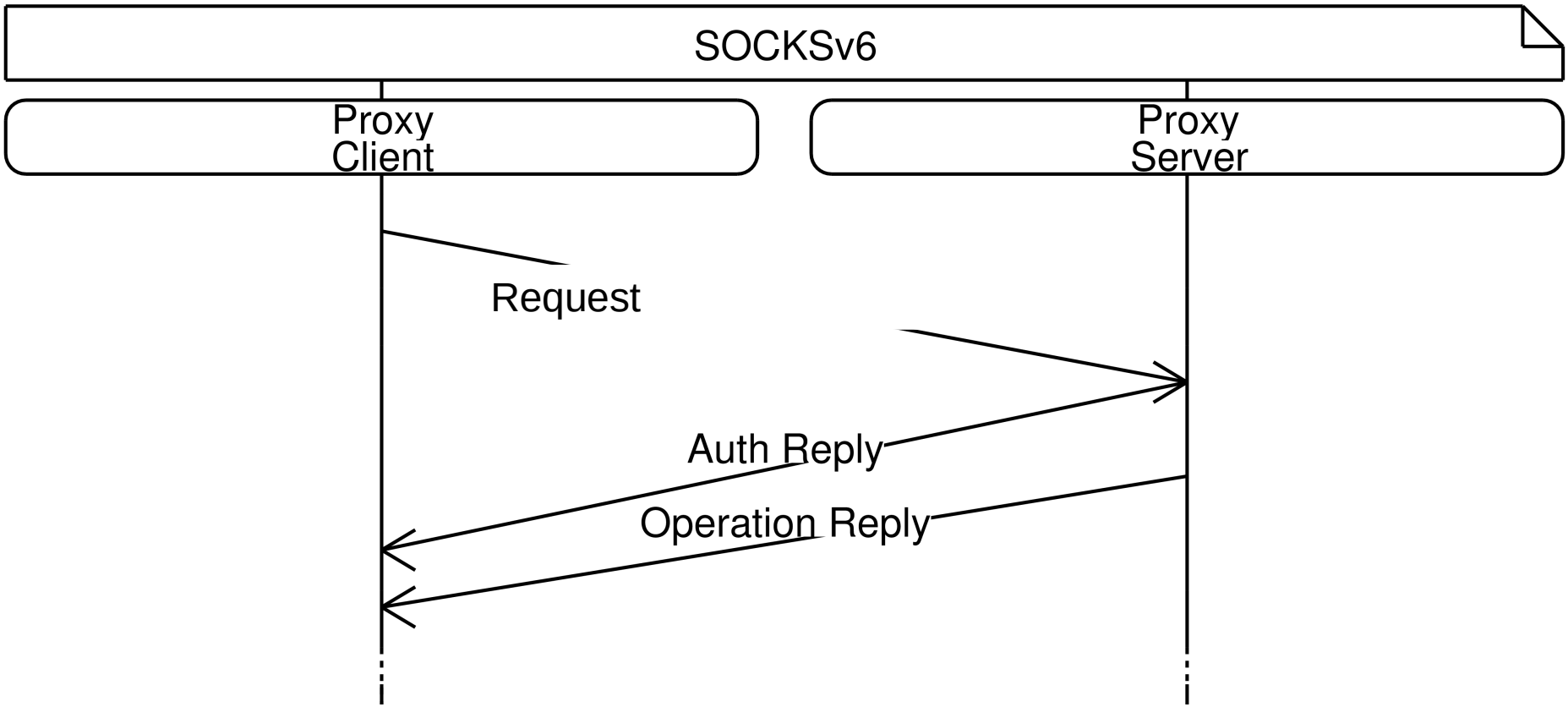
- Deprecate support for encryption
- Just run SOCKS over TLS

- TLS 1.3 has support for early data
  - 0-RTT overhead
  - Prone to replay attacks
- Need mechanism that makes SOCKS requests idempotent

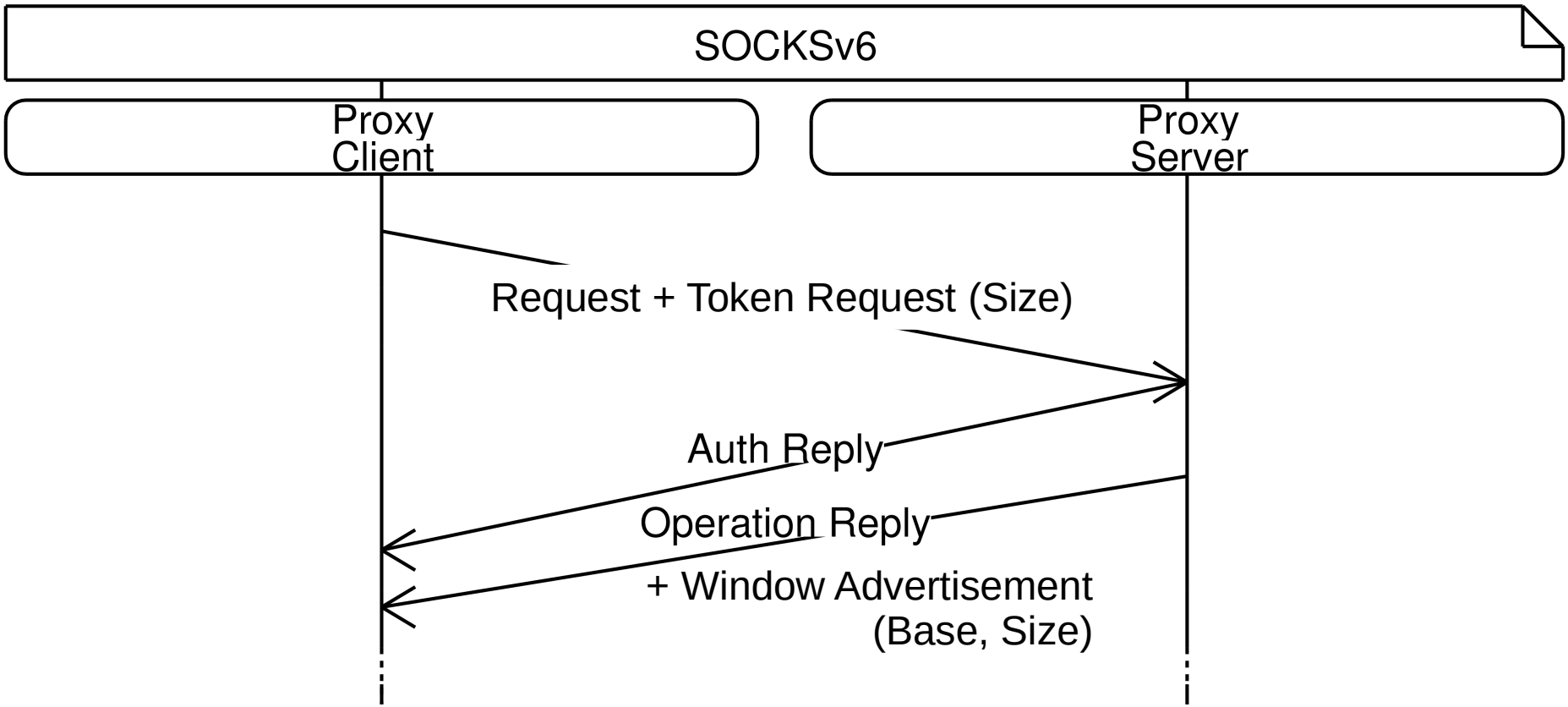
Idempotence options

- **Authenticated** clients can be granted single-use Tokens
  - Tokens are assigned on a per-user basis
- A Token can only be spent on a single operation
  - Proxies and clients keep track of spent Tokens
- Part of SOCKS Requests and Operation Replies

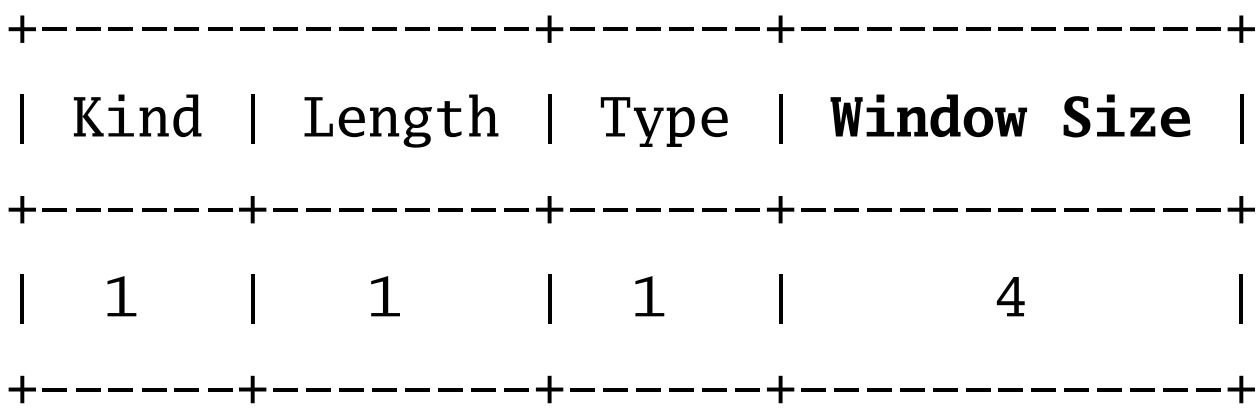
Requesting Tokens



Requesting Tokens



Token Request



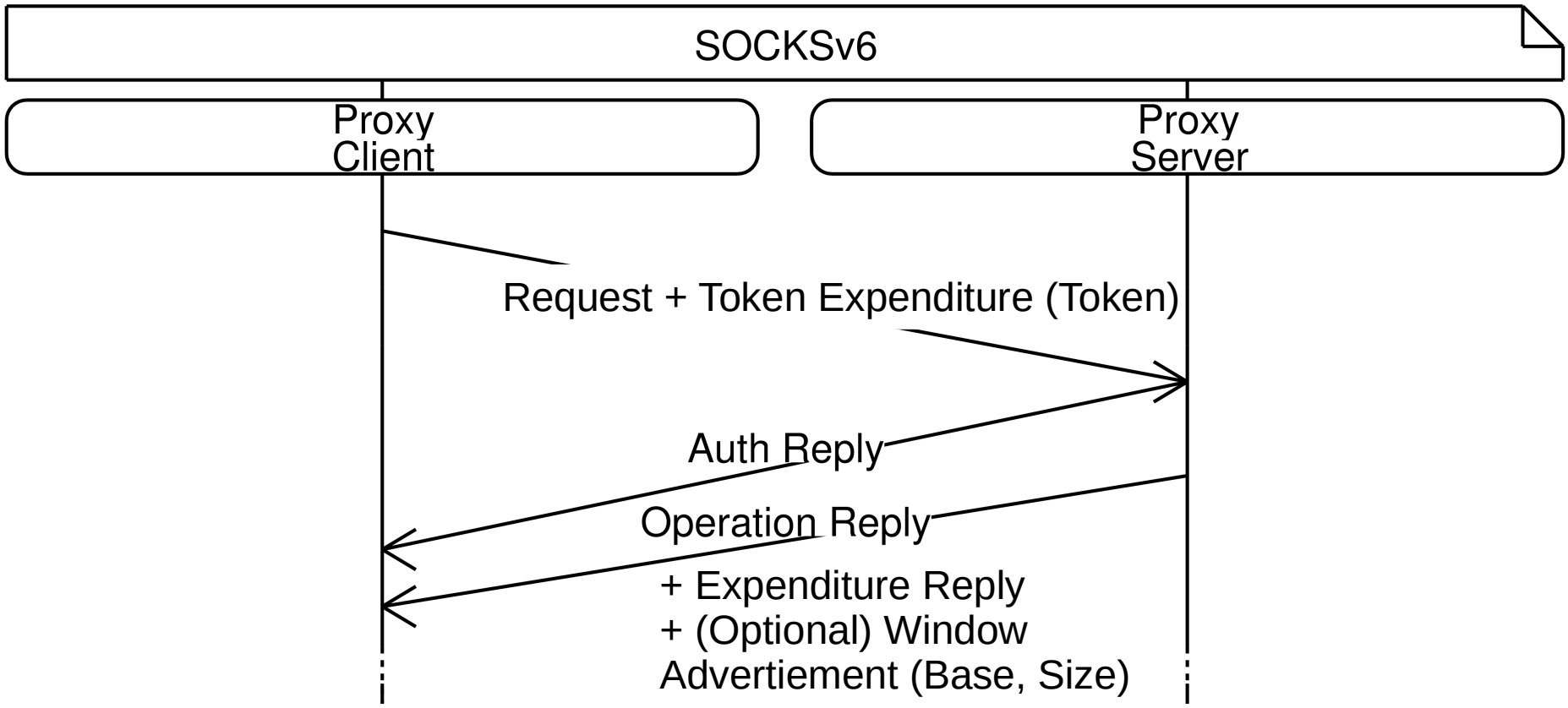
- Client starts by requesting a number of tokens
  - Can be done as part of a NOOP request
  - Secure, as long as TLS early data is not used

Token Window Advertisement

Kind	Length	Type	**Window Base**	**Window Size**
1	1	1	4	4

- Proxy offers a number of consecutive Tokens
  - Window Base: first token
  - Window Size: number of tokens
- E.g.: base=10, size=3 means that the following tokens are available: 10, 11, 12

Spending Tokens



Token Expenditure

Kind	Length	Type	**Token**
1	1	1	4

- Client spends Tokens on Operations
  - Clients SHOULD attempt to spend tokens in order

Token Expenditure Reply

Kind	Length	Type	**Response Code**
1	1	1	1

- Server replies:
  - Duplicate or out-of-window tokens are rejected

Shifting the token window

Kind	Length	Type	**Window Base**	**Window Size**
1	1	1	4	4

- Proxies can **unilaterally increment** the Window Base
  - Lowest-order Tokens are discarded, new high-order Tokens are created
  - Send unsolicited Token Window Advertisements to let clients know
- Use cases
  - Ideal: Lowest-order Tokens are spent; shift the base past them
  - The client has begun spending higher-order tokens; shift window past low-order gaps

What’s next for MPTCP?

- Options for influencing the proxy’s behavior
  - Path Manager
  - Scheduler
- Better reverse proxy support
  - Ability to listen() on a socket and have connections forwarded

Comparison to 0-RTT TCP converters

- draft-bonaventure-mptcp-converters-02

- Similarity: No control data aside from initial exchange
- Different starting point: purely layer 5 protocol
  - Can be run over TLS
  - TFO data not required, but highly beneficial
  - Midllebox doesn’t kill TCP => middlebox doesn’t kill SOCKS

Extra Slides

Token Space

- Tokens are
  - 32-bit unsigned integers
  - in a 32-bit modular space
- x < y if (y-x) < 2^31