# SOCKS Protocol Version 6 (update) draft-olteanu-intarea-socks-6-03

Vladimir Olteanu

Dragoș Niculescu

University Politehnica of Bucharest

# Changes in -03

- Mostly based on implementation experience

- More freedom w.r.t. which parts to support
- Timeliness for Token Window Advertisements
- Removed Salt options (AEAD mandatory in TLS 1.3)
- Renamed: Socket options => Stack options

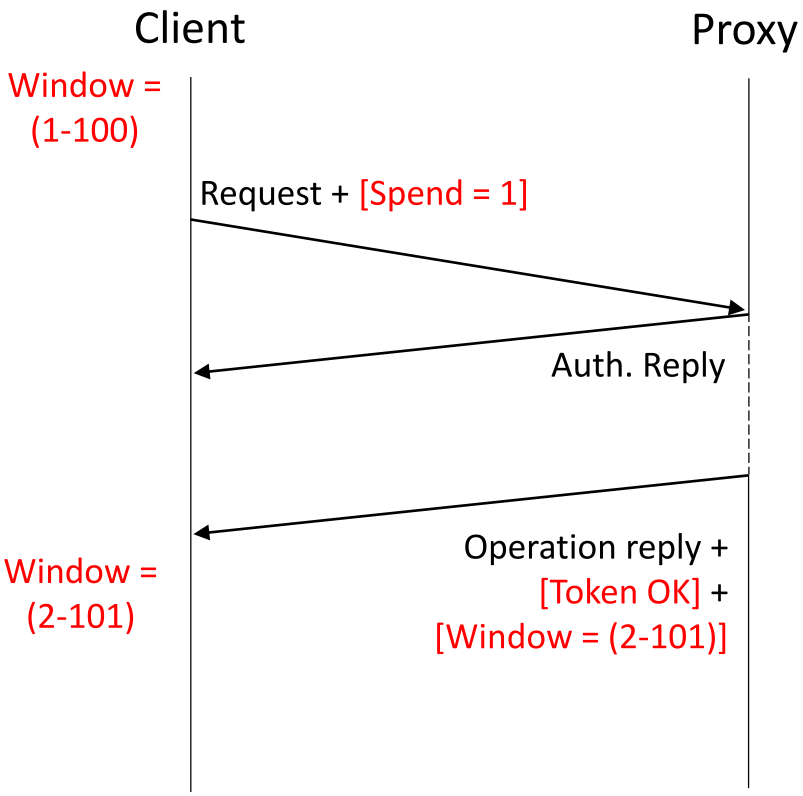
# Making options optional

- **Philosophy**: Unsupported options can be safely ignored
- Removed inter-dependencies and functionality overlap
- Just need v4 functionality?
  - Don’t support anything
- Need authentication?
  - Classic (as in v5): Auth. Method Options
  - 0-RTT: Auth. Data Options (enough for username + password)
- Avoid issues with TFO and/or TLS Early Data?
  - Idempotence options + some kind of authentication
- TFO? MPTCP proxy bypass? You get the idea…

# TFO on the client-proxy leg

- TFO payload can be replayed under rare circumstances
- Clients SHOULD NOT use TFO on the client-proxy leg unless:
  - Application protocol tolerates TFO
  - No application data in SYN payload
  - SOCKS over TLS without Early Data
  - Using Idempotence Options

# Idempotence options: refresher



Client

Proxy

Window = (1-100)

Request + [Spend = 1]

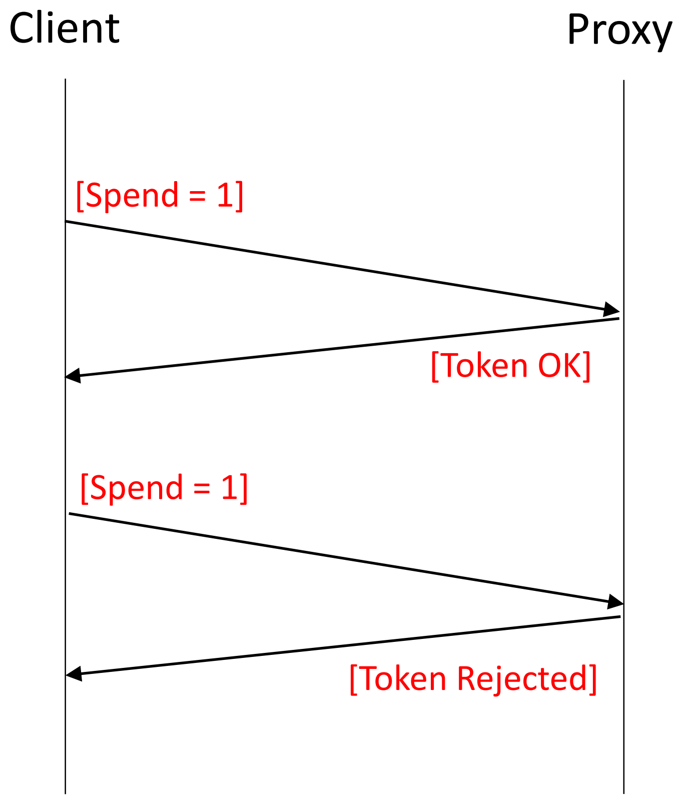
Auth. Reply

Operation reply + [Token OK] + [Window = (2-101)]

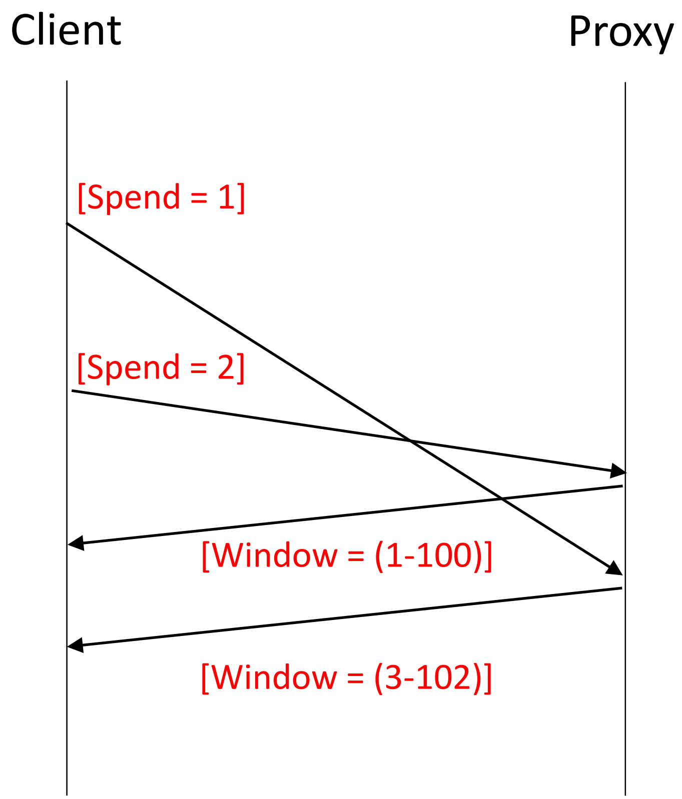
Window = (2-101)

(revision -02)

# Idempotence options: refresher

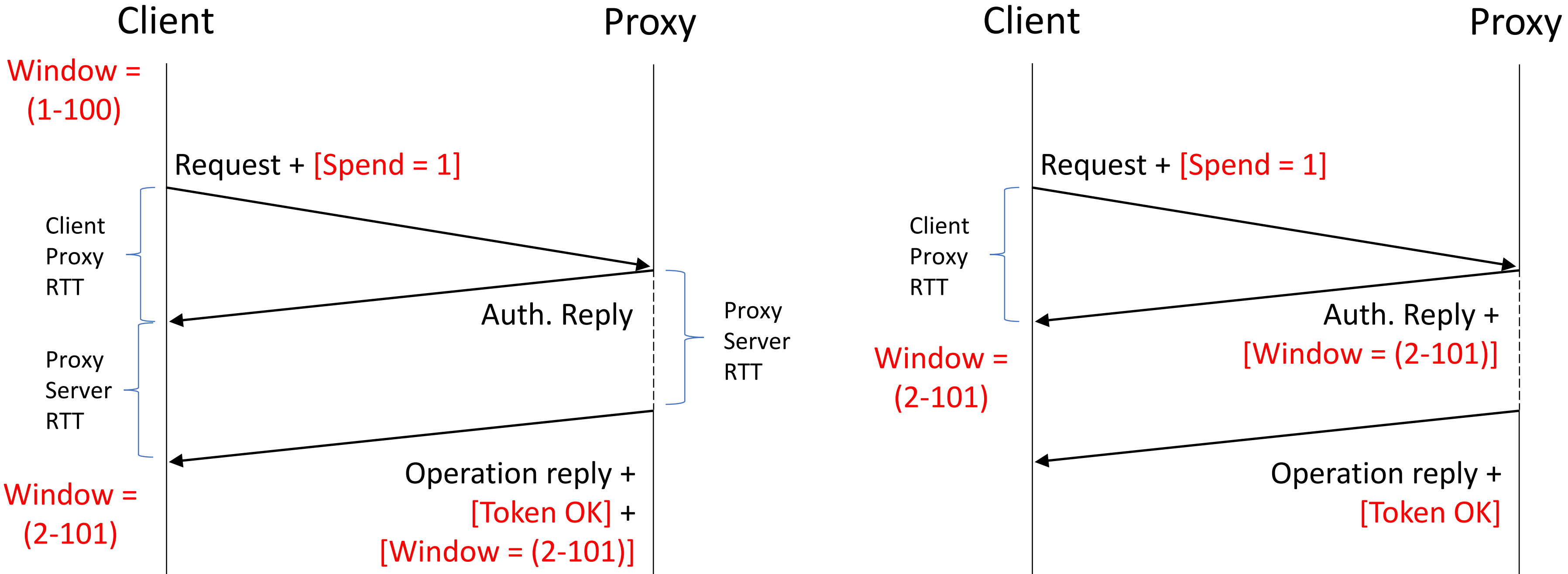


- Duplicates are rejected



- Reordering is tolerated

# Idempotence options: timeliness



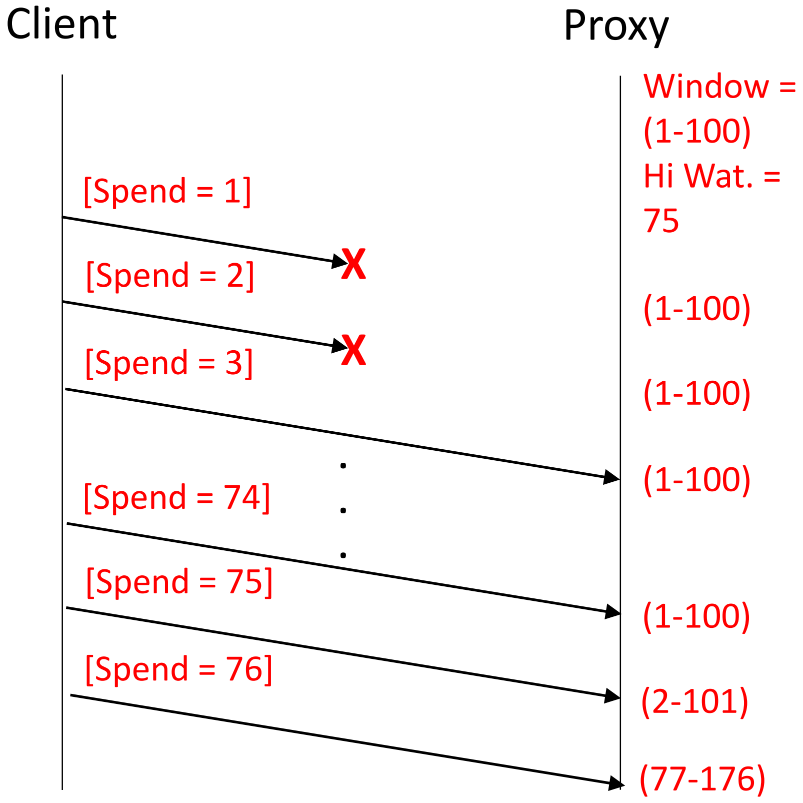
(revision -02)

(revision -03)

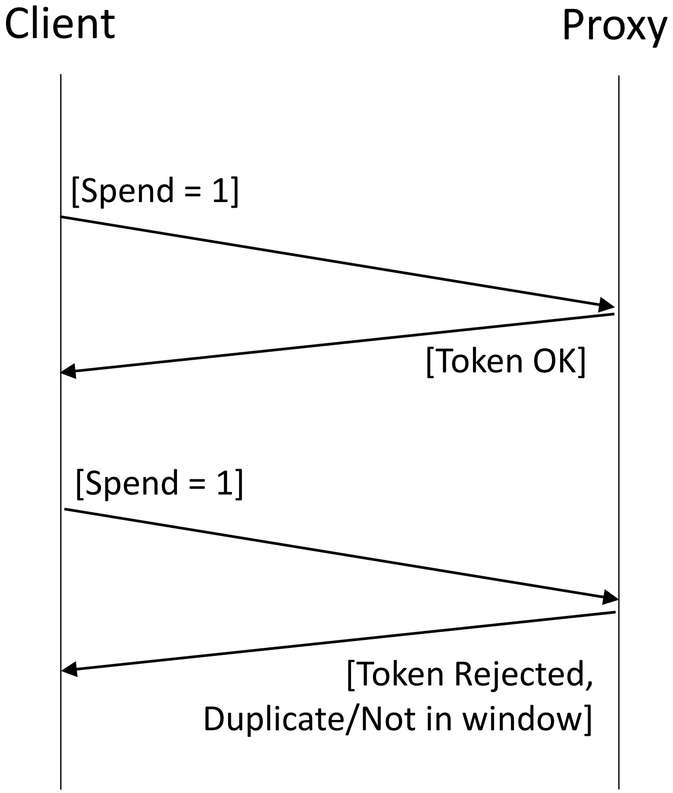
- Window Advertisements moved from Op. Reply to Auth. Reply

# Idempotence options: tracking used tokens

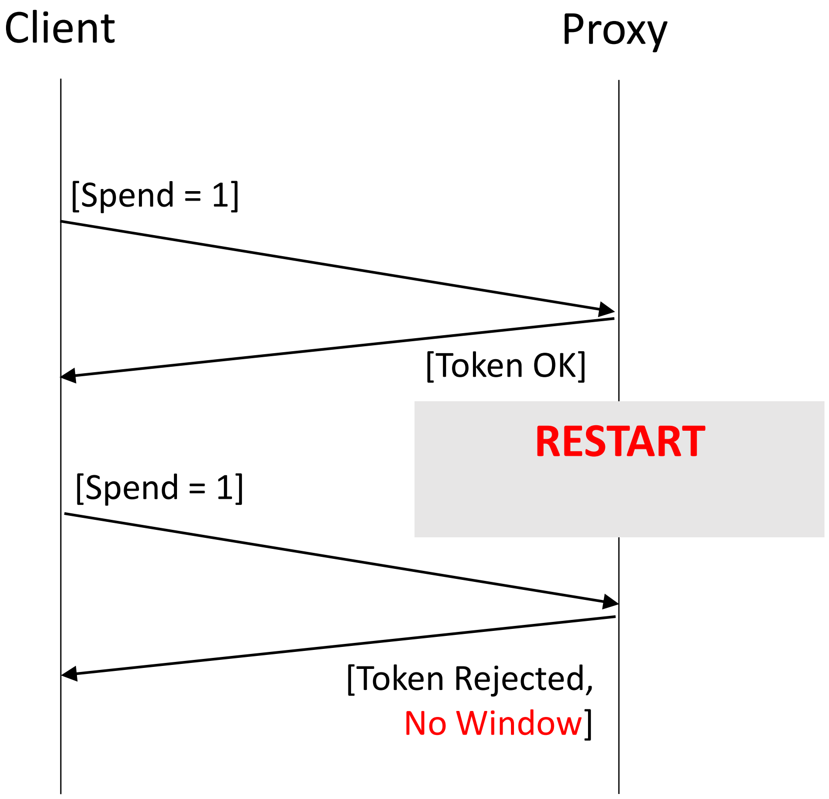
- Constant memory usage per user
  - Proxy only tracks tokens in window
  - Bitmap + few integers
- Use a **high water mark** to handle dropped requests
  - Not in draft



# Idempotence options: functionality downgrade

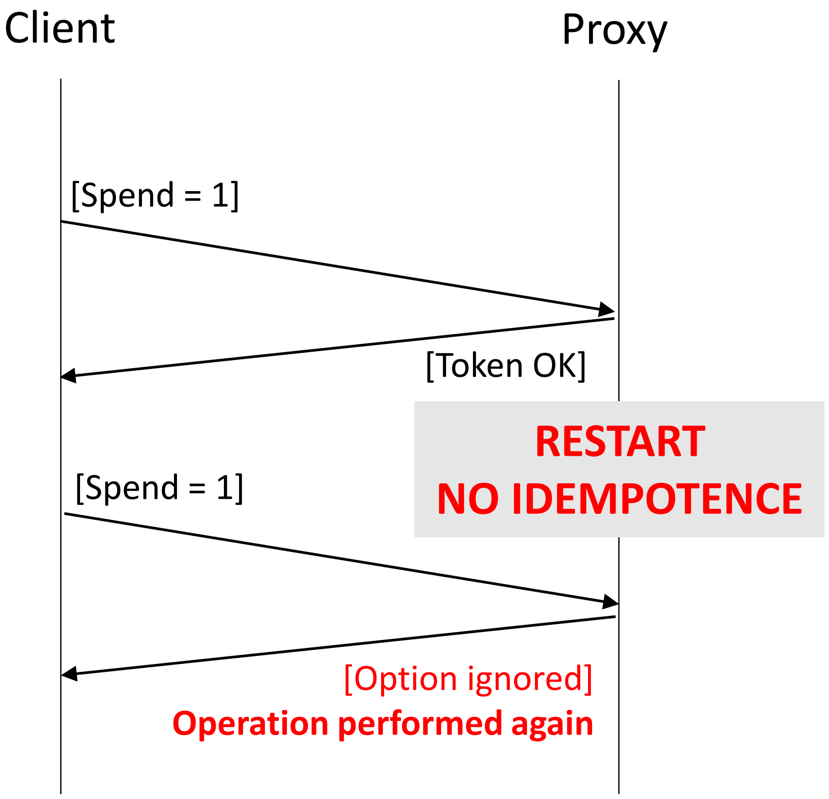


# Idempotence options: functionality downgrade



- Restarting the proxy **with support** for idempotence is **ok**
  - Mandatory under -02. But what about options being optional?

# Idempotence options: functionality downgrade



- Restarting the proxy **with support** for idempotence is **ok**
  - Mandatory under -02. But what about options being optional?
- Restarting the proxy **without support** for idempotence can be **problematic**
  - Possible under -03

# Idempotence options: functionality downgrade

- Prevent replays, rather than ensuring idempotence
  - Solution depends on use case
- TFO: Disable TFO for 1 MSL prior to the downgrade
- TLS Early Data: Kill TLS sessions

# Message library API example

- Fully-featured message library (C++ with C bindings)

## Creating a Request

```
uint8_t buf[1500];


struct S6M_Request req = {
    .code = SOCKS6_REQUEST_CONNECT,
    .addr = {
        .type = SOCKS6_ADDR_DOMAIN,
        .domain = "somesite.org",
    },
    .port = 80,
    .optionSet = {
        .tfo = 1,
    },
};

ssize_t size = S6M_Request_pack(&req, buf, 1500);
if (size < 0) {
    /* error */
}

/* send the request */
```

## Parsing a Request

```
uint8_t buf[1500];

/* receive the request */

struct S6M_Request *req;










ssize_t size = S6M_Request_parse(buf, 1500, &req);
if (size < 0) {
    /* error */
}

/* do something with the request */

S6M_Request_free(req);
```

# Implementation

- Message library (feature-complete): https://github.com/45G/libsocks6util
- Utility library: https://github.com/45G/libsocks6util
- Basic prototype based on Shadowsocks: https://github.com/45G/shadowsocks-libev
- Full-blown implementation in the works: https://github.com/vlolteanu/sixtysocks