Design of SOCKS Version 6

Vladimir Olteanu, Dragoș Niculescu - University Politehnica of Bucharest

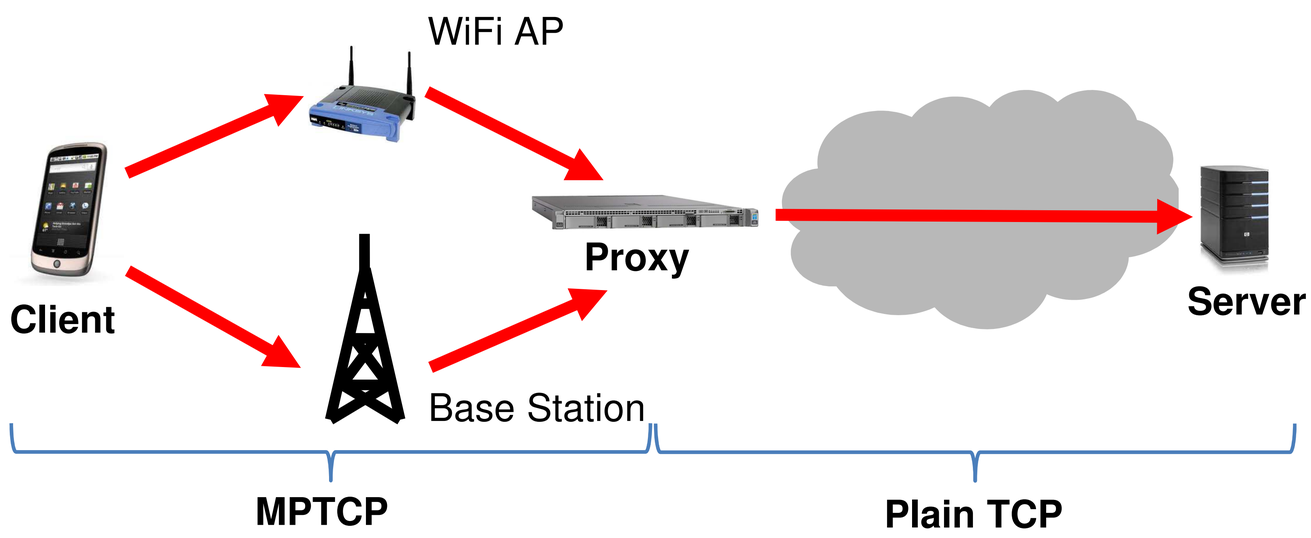
Motivation: Mobile MPTCP Deployment

"Bond" Cellular and WiFi for higher throughput
- Need proxy: most servers don't deploy MPTCP

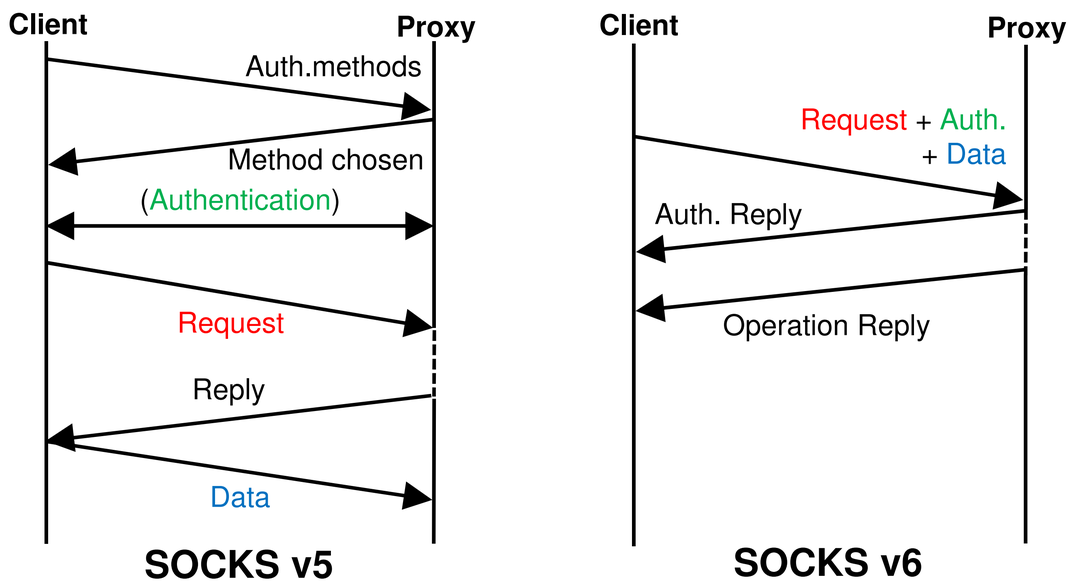
SOCKS v5 has high RTT overhead
- Problem exacerbated by high latency between phone and tower

Take advantage of TFO and TLS 1.3
- Fewer RTTs, but need special consideration

Proliferation of non-standard proxies, e.g Shadowsocks
- Unreviewed in terms of security



Basic Protocol



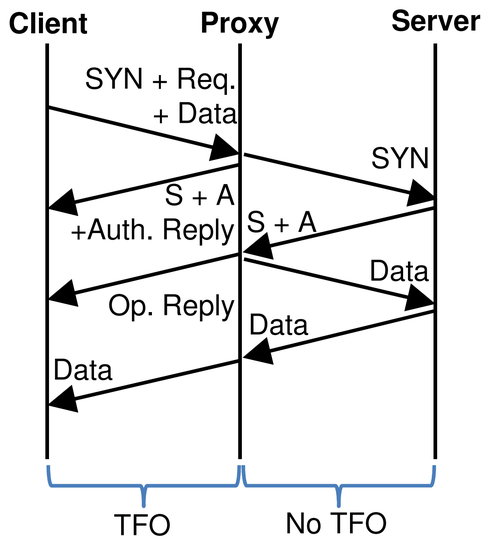
- **Send as much information upfront as possible**
- **Leverage 0-RTT authentication schemes**
- **Extensible: all messages can carry options**

Low RTT Usage

	TFO at proxy	TFO at server	Total RTT
TCP	-	No	2P + 2S
	-	Yes	P + S
SOCKS v6	No	No	2P + 2S
	Yes	**No**	**P + 2S**
	Yes	Yes	P + S

Time taken to receive a data response

- **No worse than TCP**
- **Outperforms TCP if TFO is unavailable at the server (see figure)**
- **TLS 1.3 adds no RTT overhead if using 0-RTT session resumption**



Idempotence Mechanism

0-RTT techniques have caveats
- TFO can lead to duplicate connections at the server
- TLS 0-RTT Data is prone to replay attacks

SOCKS Requests can optionally be made idempotent
- Replays become nearly impossible
- Makes TF0 and TLS 0-RTT Data safe to use

Clients request and then spend idempotence *tokens*
- Tokens are numbers in a 32-bit modular space
- A token can only be spent on one SOCKS Request
- Clients attempt to spend tokens in order

Proxies grant token *windows*
- Windows are contiguous ranges of tokens
- Only tokens inside the window are tracked
- New tokens are generated by *shifting* the window

Other Features

- **TFO on the proxy-server leg**: Clients can explicitly request that the proxy connect to the server using TFO.
- **MPTCP Proxy Bypass**: Clients can be informed if the server supports MPTCP; they can then contact the server directly.
- **MPTCP Scheduler**: Applications requiring low latency can request that application data be duplicated across all subflows.

Implementation and standardization

Code: https://github.com/45G
- **SOCKS v6 prototype**
- **Message library**

IETF Draft: draft-olteanu-intarea-socks-6

Thanks to: 45G

Executive Agency for Higher Education, Research, Development and Innovation Funding