Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

30th September 2019
University of Lübeck

Enhanced Performance and Privacy for Core Internet Protocols

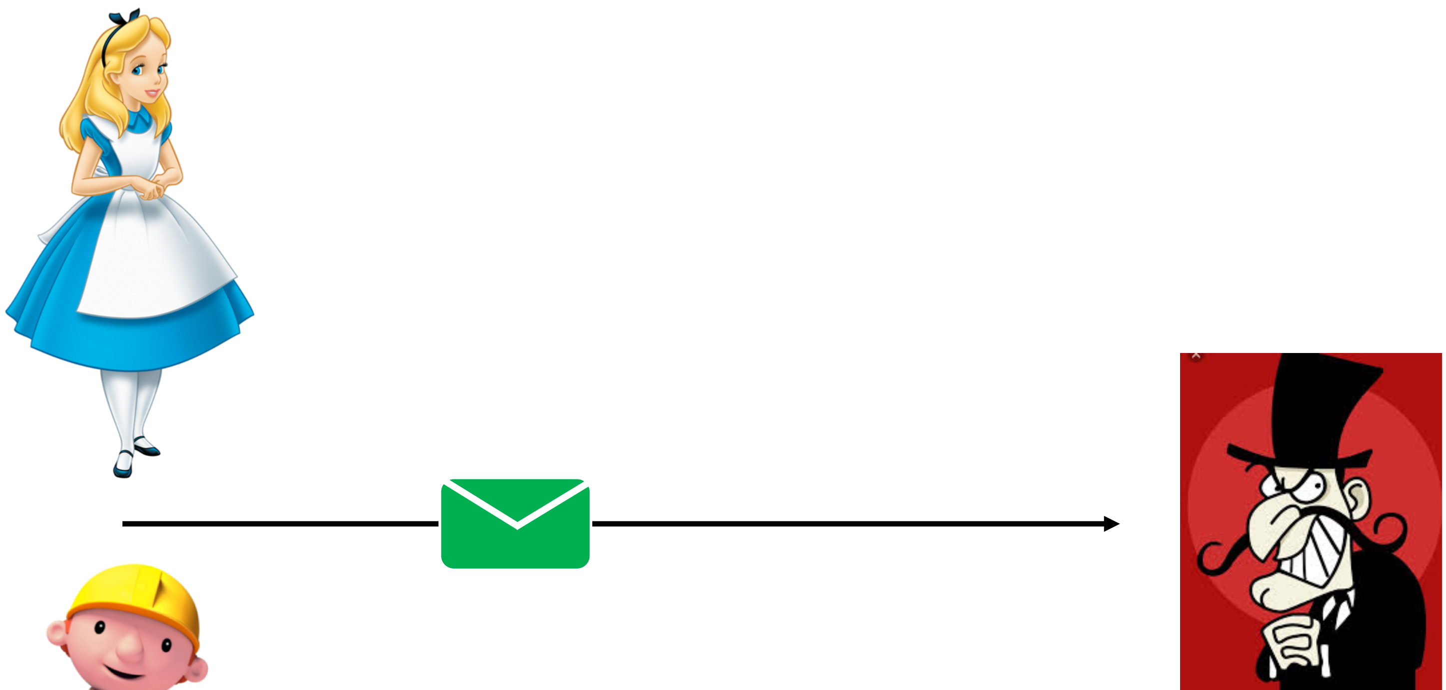
Erik Sy

The Right to Informational Self-Determination

- Individuals have the right to determine in principle the disclosure and use of their personal data (German constitution)
- “Self-determination is an elementary prerequisite for the functioning of a free democratic society” (Census Act, German Federal Constitutional Court)

Do core Internet protocols comply with our right to informational self-determination?

Motivation

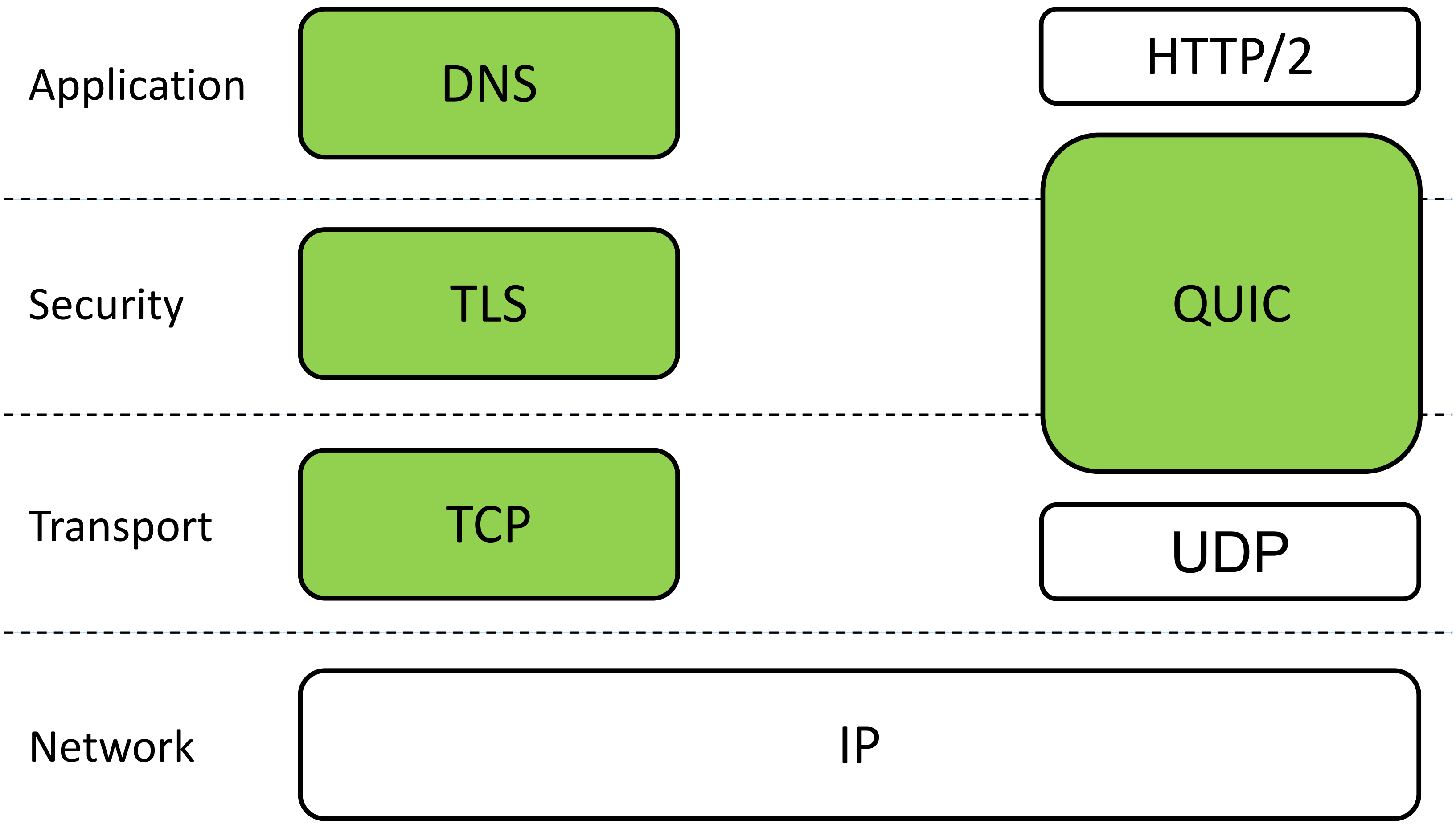


Transport protocols should not allow distinguishing Alice and Bob as the sender of a message.

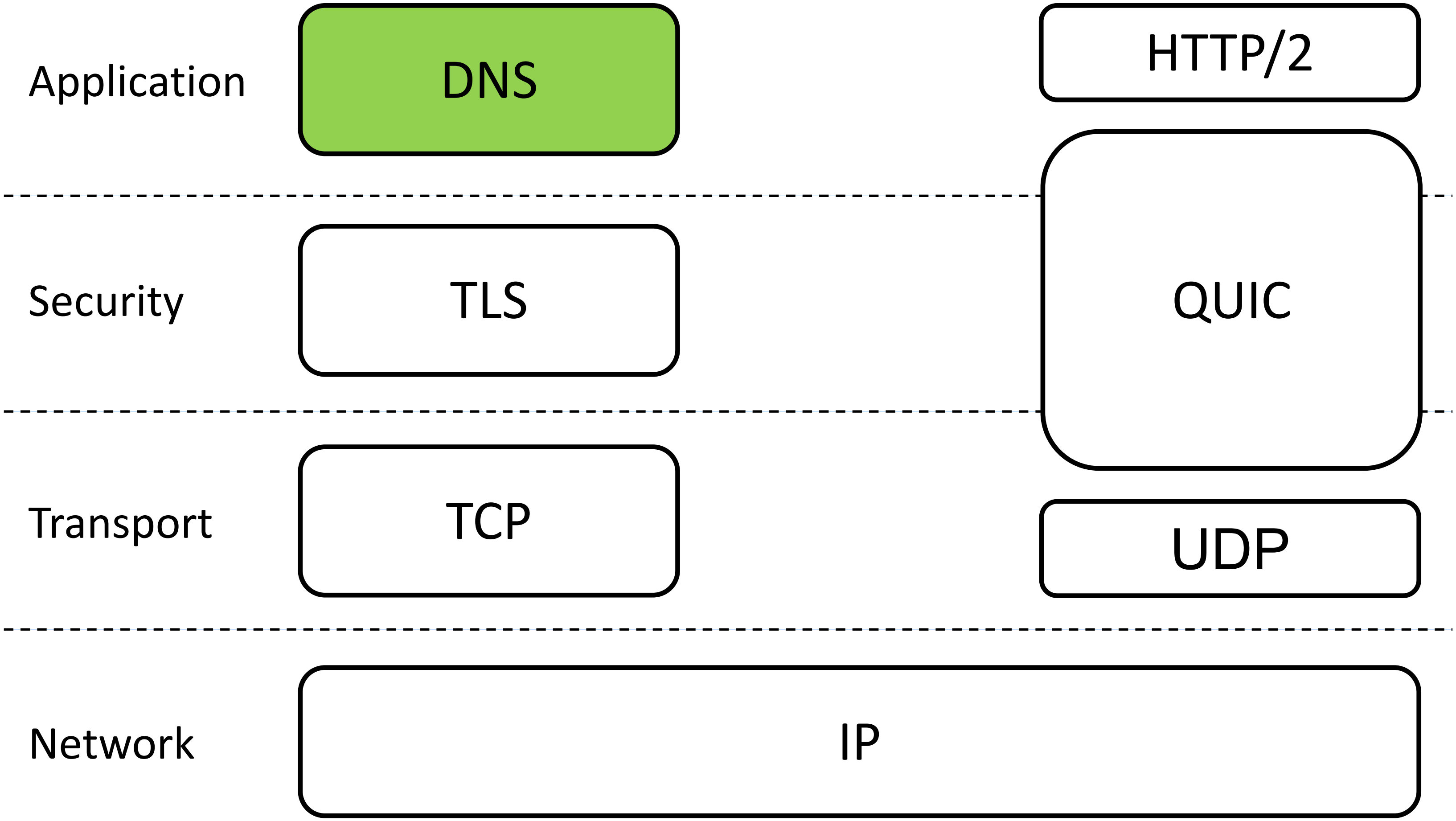
Motivation

- Increase the quality of experience for web users
  - The delay of the connection establishments presents a significant overhead of an average web flow

Investigated Protocols

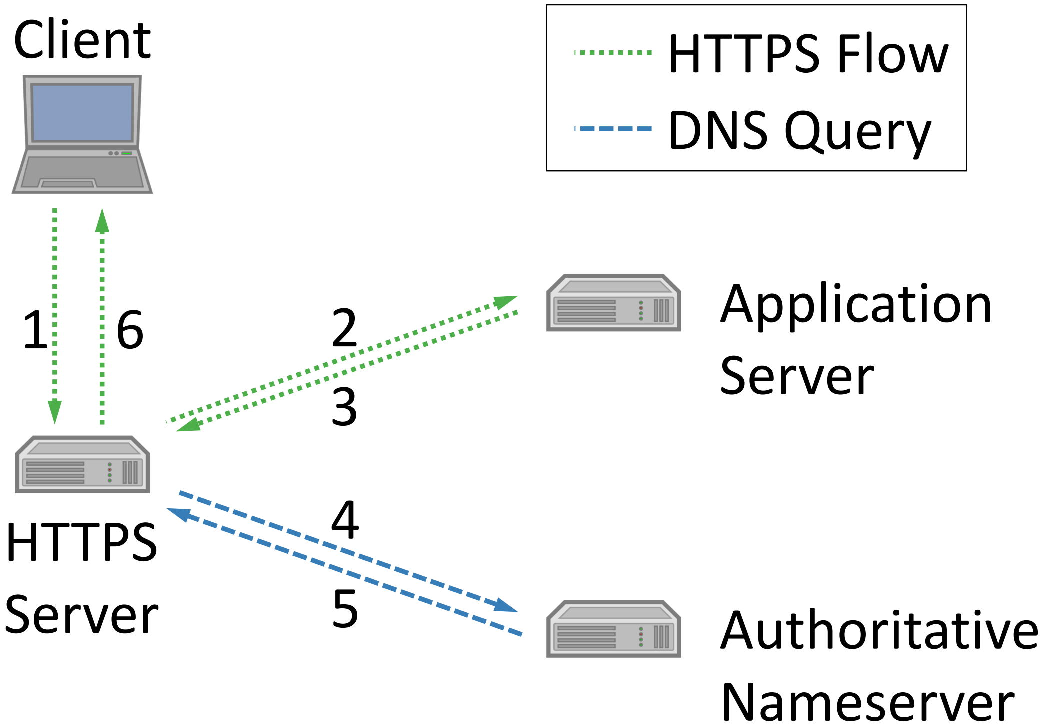


Introducing Resolver-Less DNS



Introducing Resolver-Less DNS[1]

- Web server provides relevant DNS records to it's clients
  - Improves client's privacy posture towards resolver & reduces delay



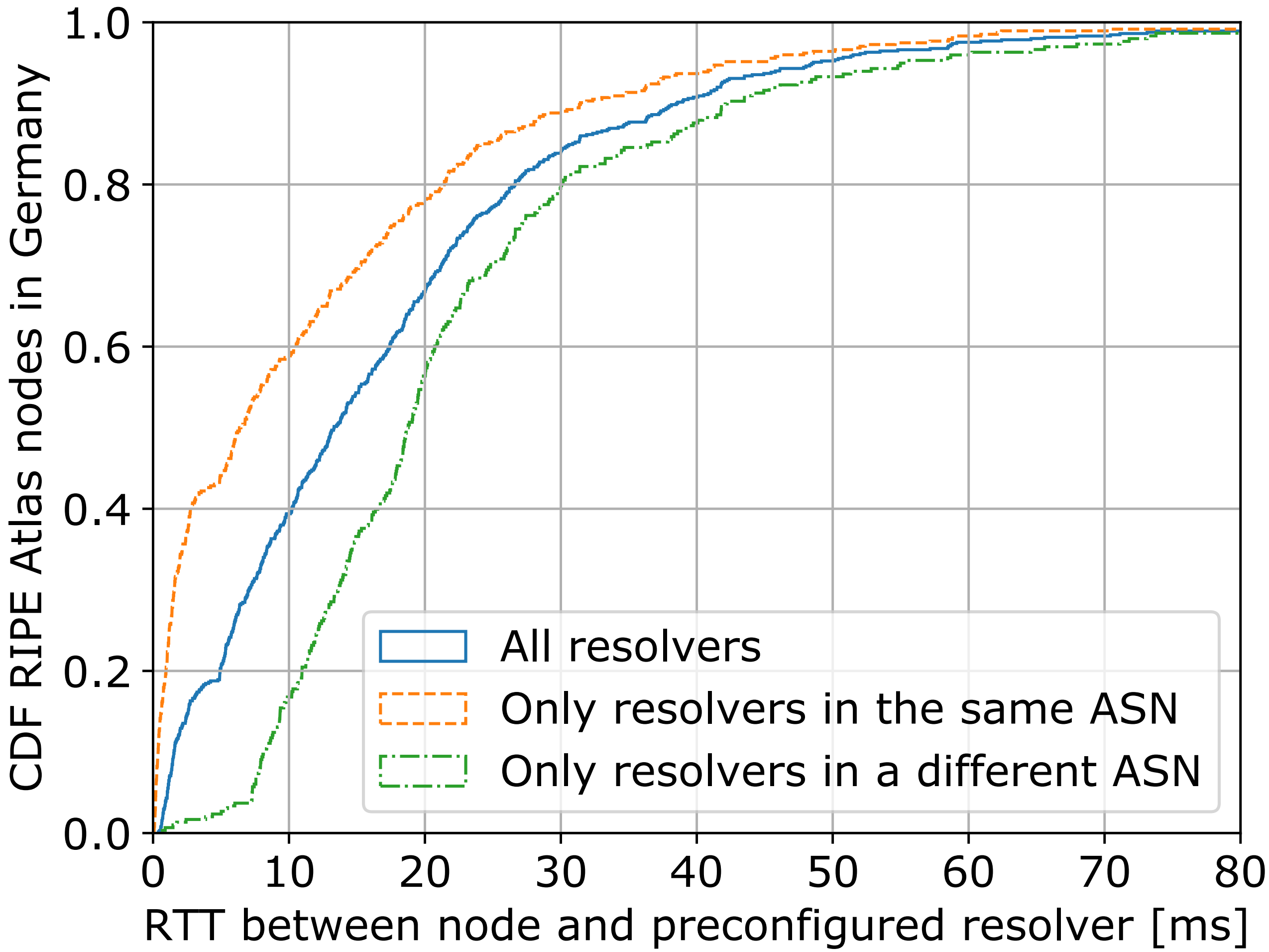
1: Sy, Erik "Enhanced Performance and Privacy via Resolver-Less DNS" (2019)

Validation Mechanisms of Resolver-Less DNS

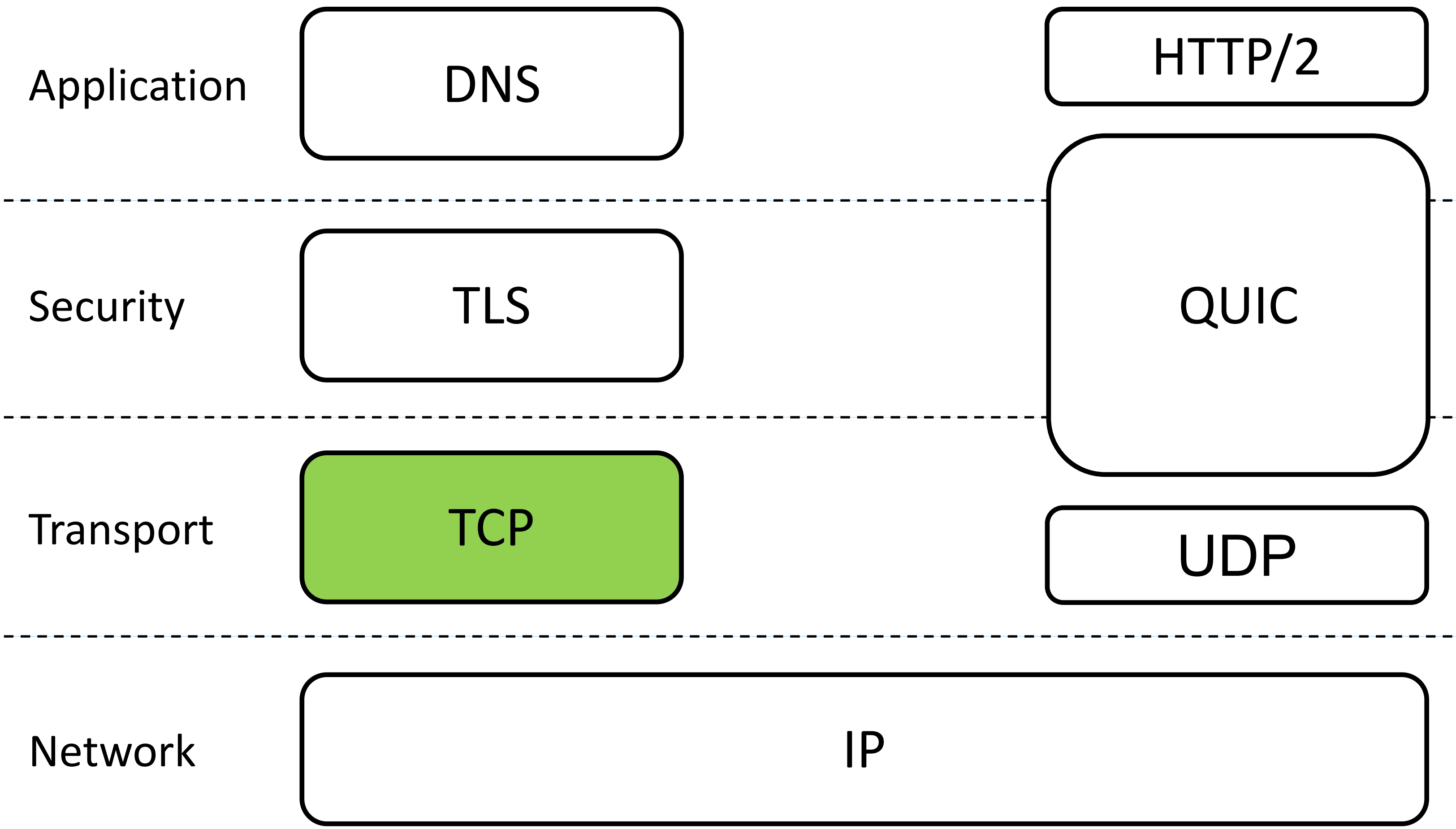
- Client does not send application data to presented IP address before a successful validation of the used DNS record
- Preferred validation mechanism uses server authentication during connection establishment
- Fallback validation mechanism includes traditional DNS lookup to make a comparison between both DNS records

Performance Evaluation of Resolver-Less DNS

- 1% of clients saves at least 80ms per DNS query compared to status quo

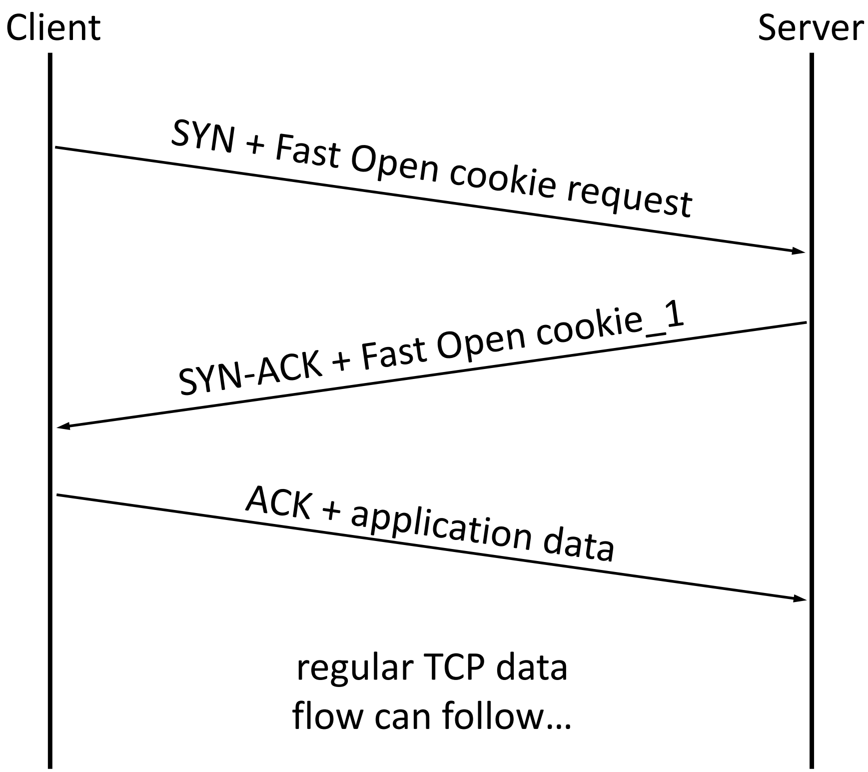


Introducing TCP Fast Open (RFC 7413, Dec 2014)

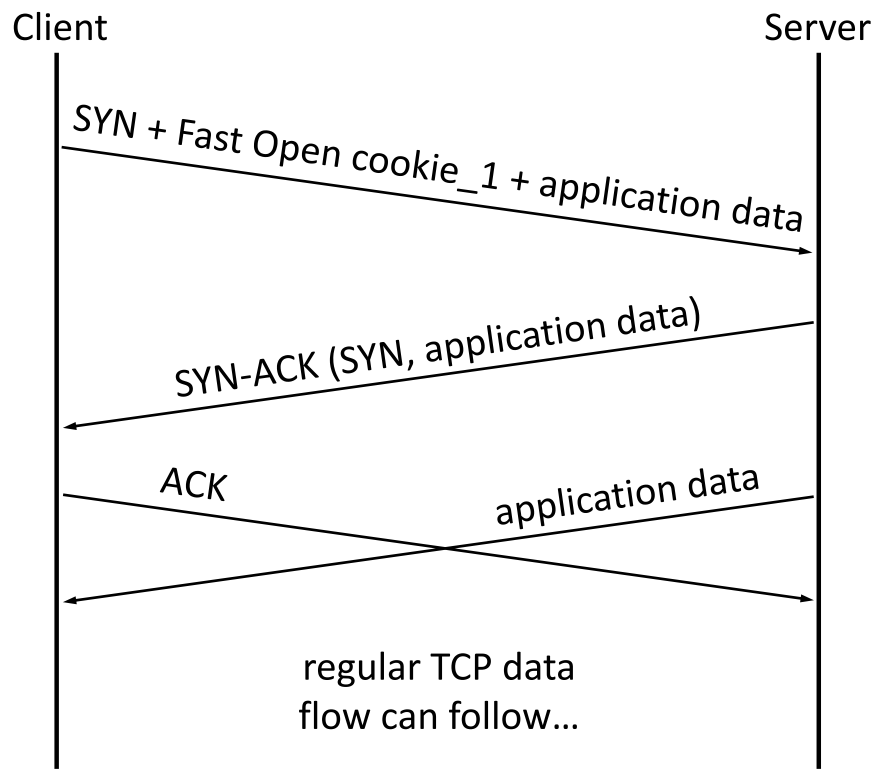


Introducing TCP Fast Open (RFC 7413)

- Allows validating the client's IP address without an additional round trip



1) Initial handshake



2) Abbreviated handshake

User Tracking via TCP Fast Open

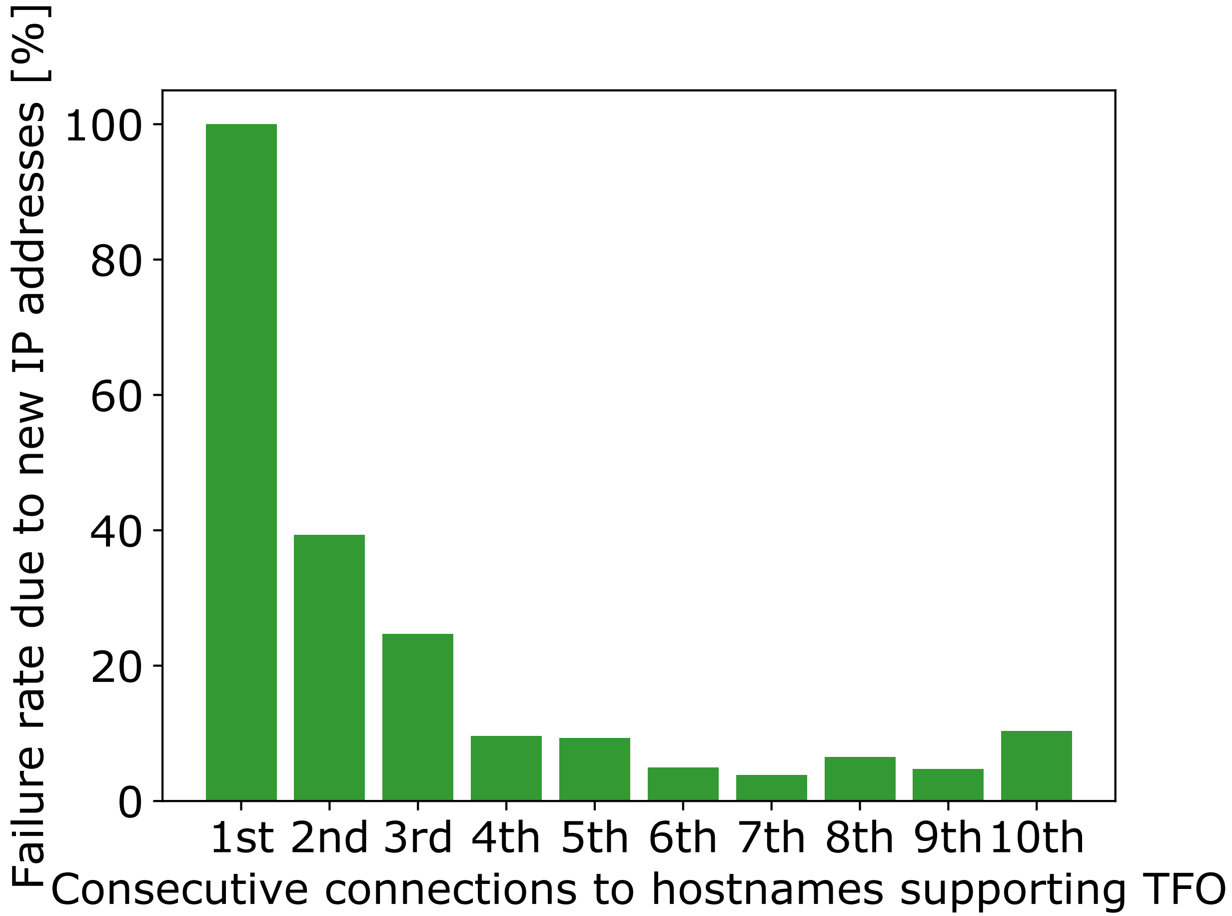
- Main findings[2]
  - Fast Open cookies present a kernel-based tracking mechanism
  - Tracking feasible for network observer
  - Feasible tracking periods are unrestricted
  - Enables tracking across private browsing modes, browser restarts, and different applications

- Reactions by browser vendors
  - Mozilla stopped using TFO within Firefox
  - Microsoft stopped using TFO within the private browsing mode of Edge

2: Sy et al. “Enhanced Performance and Privacy for TLS over TCP Fast Open” (2019)

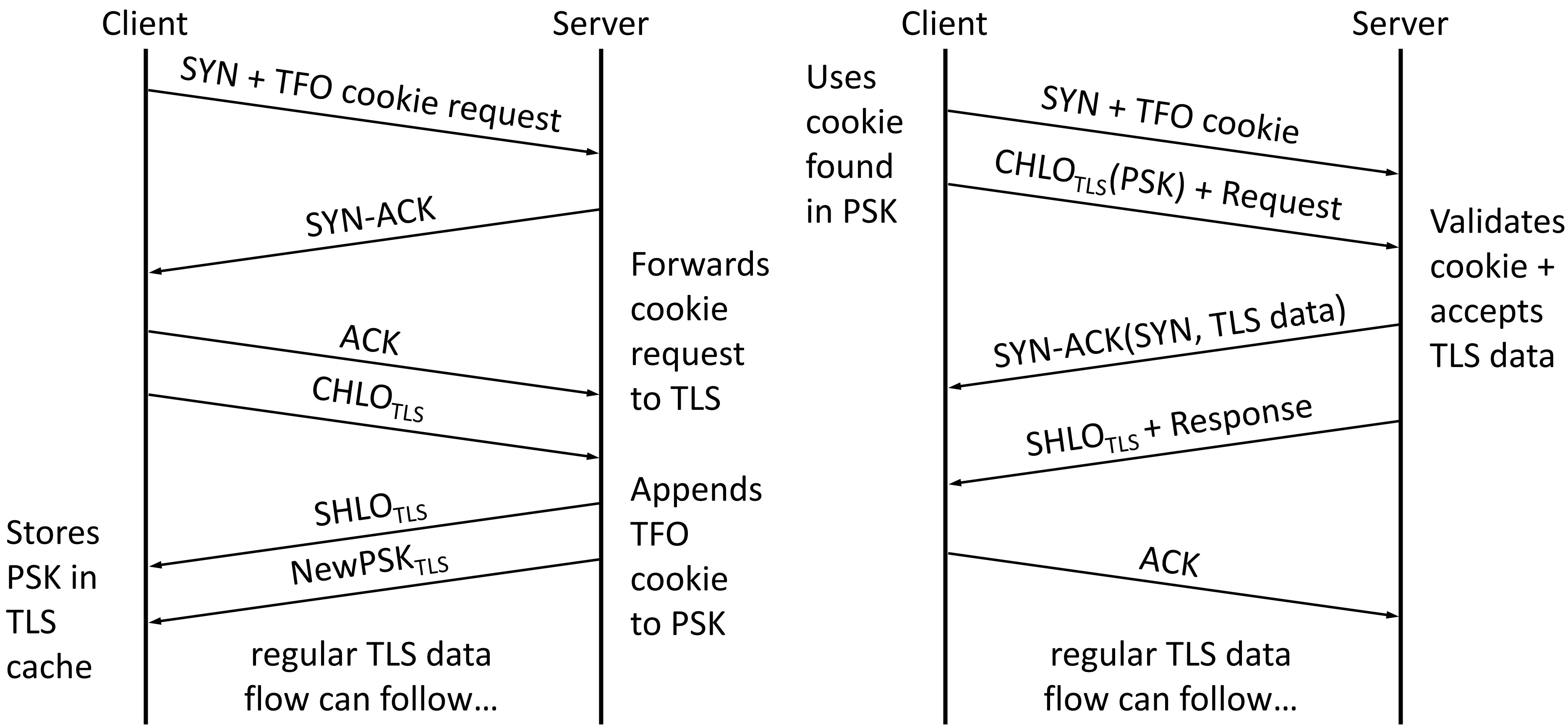
Performance Limitation of TCP Fast Open

- Requirement of matching server IP address for abbreviated handshakes does not anticipate real-world load balancing



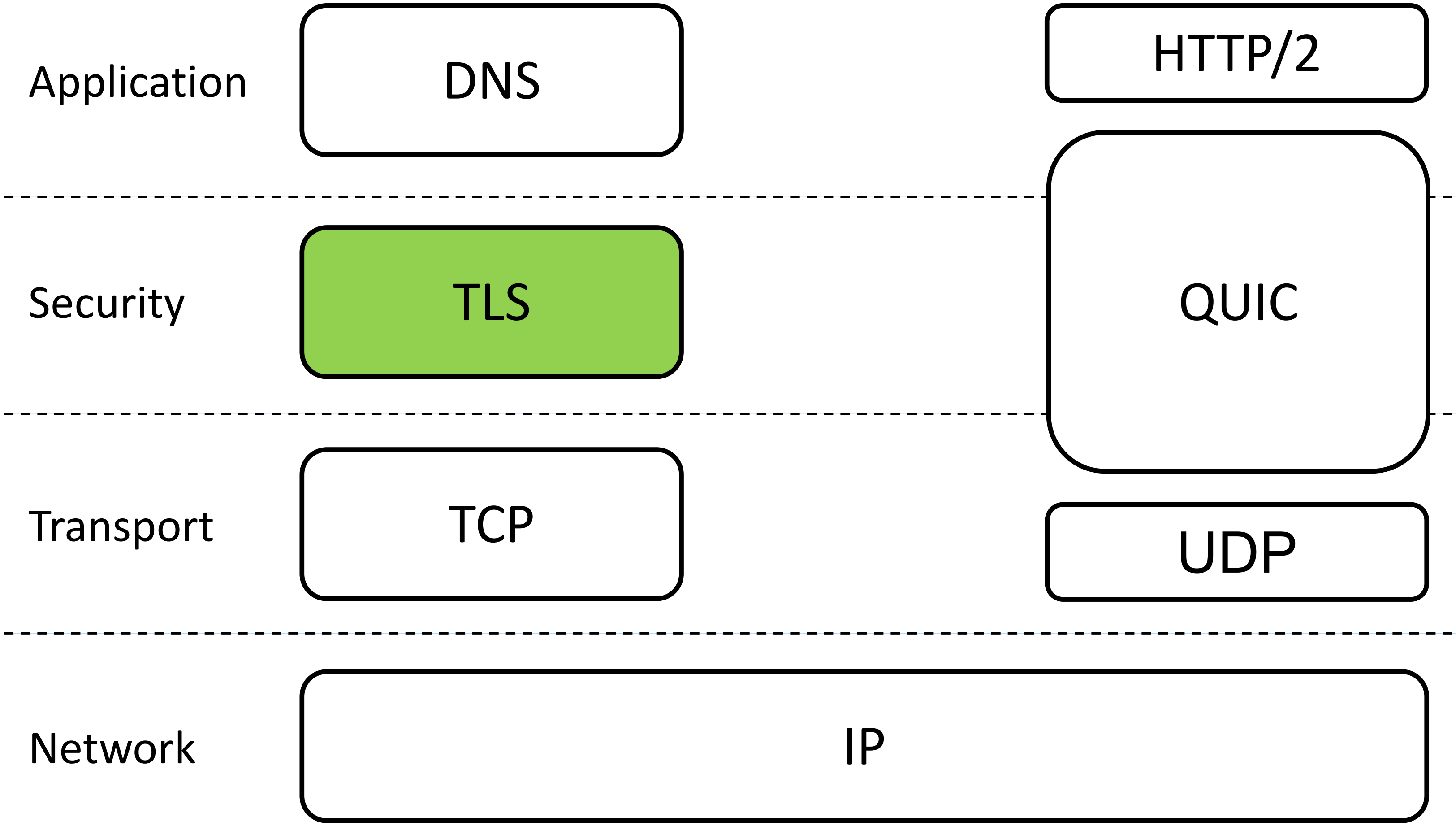
Proposed TCP Fast Open Privacy

- Cross-layer approach to mitigate privacy and performance issues of TFO



1) Initial handshake   2) Abbreviated handshake

Introducing TLS Session Resumption



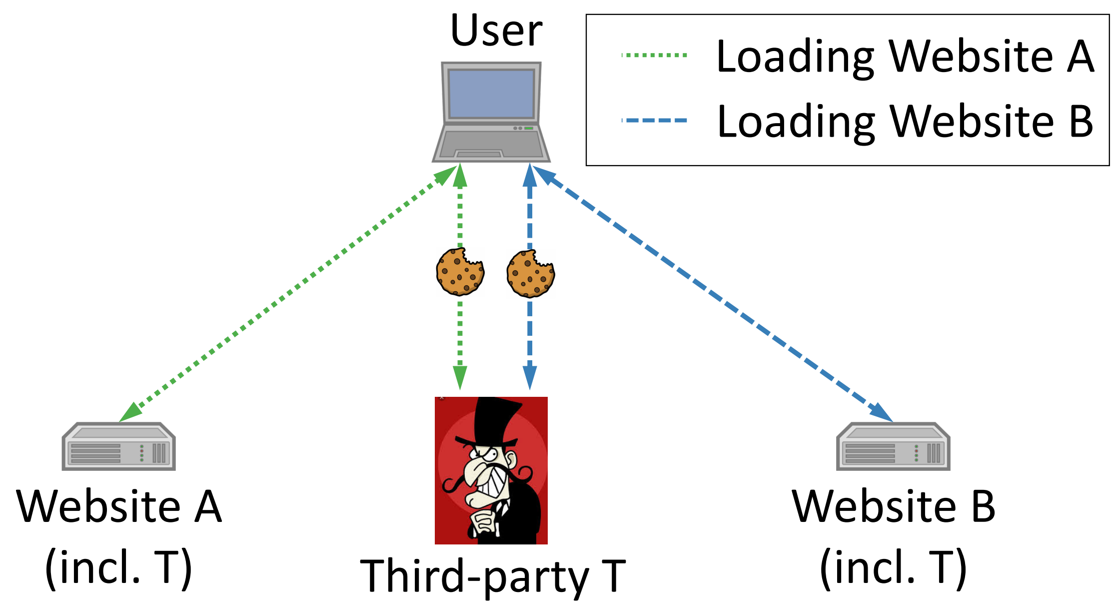
Introduction to TLS Session Resumption

- Allows a client-server pair to establish a new TLS connection with a previously exchanged symmetric key
  - Reduces the delay and the computational overhead of TLS handshakes
  - Server can uniquely identify clients based on this secret key

- Deployment on the Internet
  - 96% of TLS-enabled Alexa Top Million Sites support TLS resumption
  - All popular web browsers support this feature, which is included in every TLS version

Tracking via TLS Session Resumption

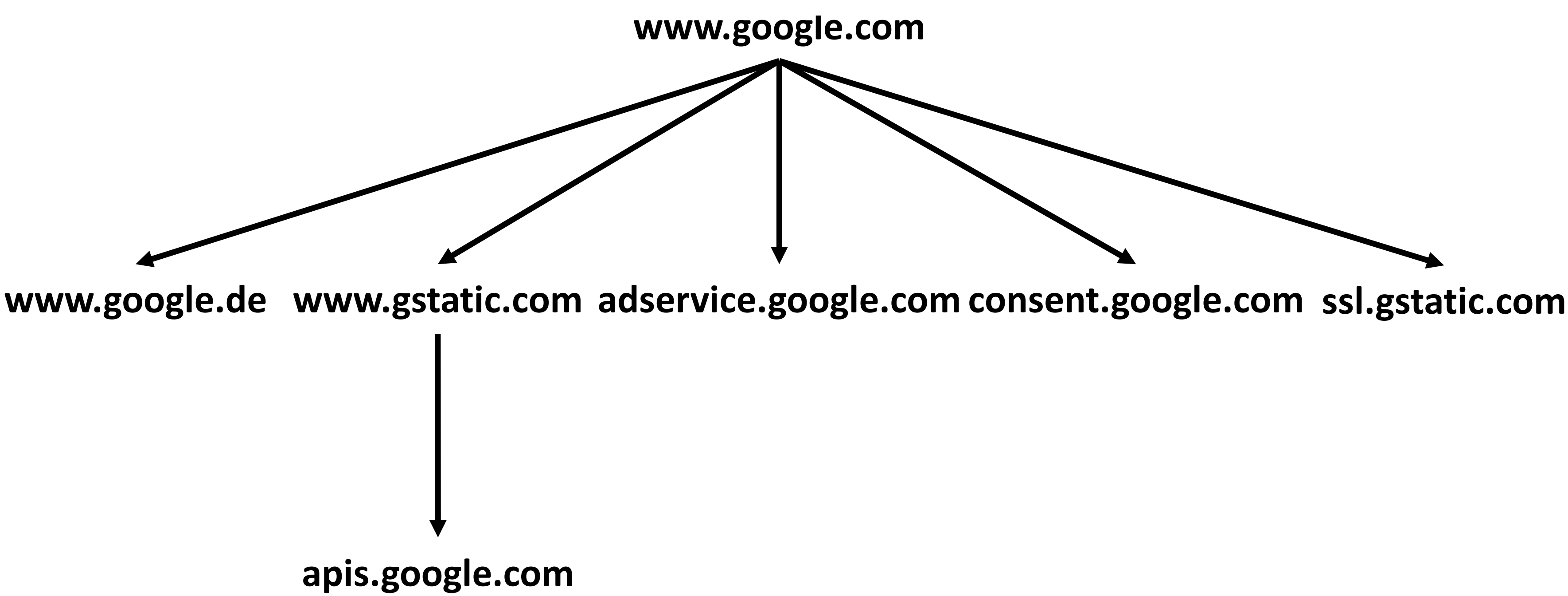
- Main findings[3]
  - Safari and Firefox can be tracked for at least 24h using this mechanism
  - Prolongation attack extends feasible tracking periods
  - Only TLS v1.3 protects against tracking by network observer
  - Most browsers do not protect against third-party tracking via TLS SR



3: Sy et al. “Tracking Users across the Web via TLS Session Resumption” (2018)

Domain Trees of popular Websites[4]

- Alexa Top 1K Site requires on average 20.24 connections to different hosts
- These hostnames form on average 9.49 TLS trust groups[1]



4: Sy et al. “Enhanced Performance for the encrypted Web through TLS Resumption across Hostnames” (2019)

Proposed TLS 1.3 Extension

- TLS 1.3 allows resumptions across hostnames, if the corresponding hostnames can be validated via the same server certificate
- Server signals that a group of hostnames mutually support TLS resumptions
  - Presented server certificate needs to be valid for theses hostnames
- SAN-list of certificate can be used to define this group
  - Adds complexity to the generation of server certificates
  - Helps to avoid resumptions to hostnames for which the cert is not valid
- Extension for the NewSessionTicket frame

Performance of TLS 1.3 Connection Establishments

- Elapsed time

Network latency	Initial	1-RTT resumed	0-RTT resumed
0.3 ms	29.2 ms	6.3 ms	6.6 ms
50 ms	190.1 ms	160.1 ms	109.6 ms
100 ms	340.8 ms	310.3 ms	209.7 ms

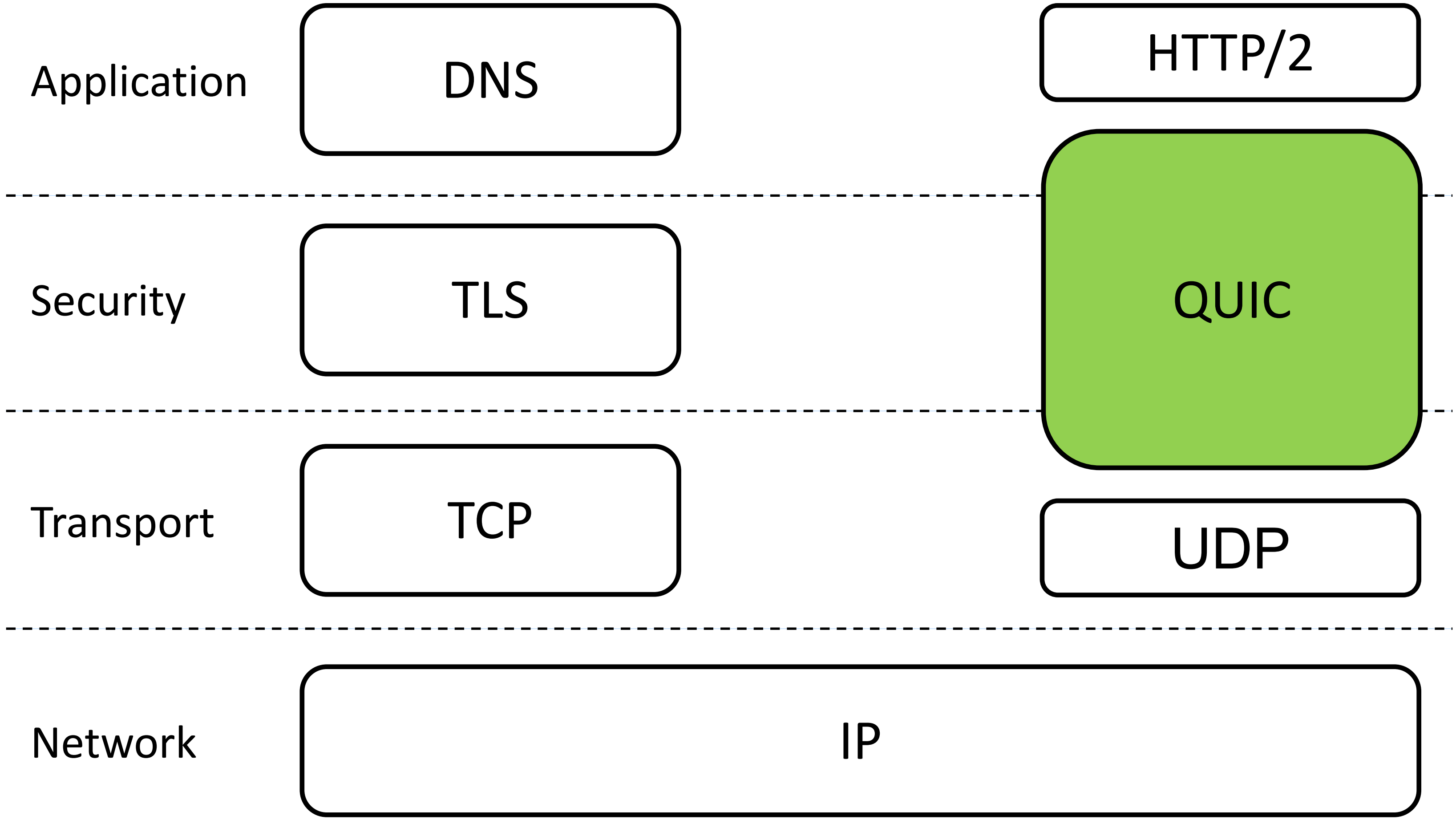
- CPU time

Peer	Initial	1-RTT resumed	0-RTT resumed
Server	7.8 ms	2.3 ms	2.6 ms
Client	9.2 ms	2.4 ms	2.5 ms

Results for an average Website

- Converts about 58.7% of the required full TLS handshakes to resumed connection establishments
- Reduces the required CPU time for the TLS connection establishments by about 44%
- Reduces the elapsed time to establish all required TLS connections by up to 30.6%

Introducing QUIC



Introduction to the QUIC Transport Protocol

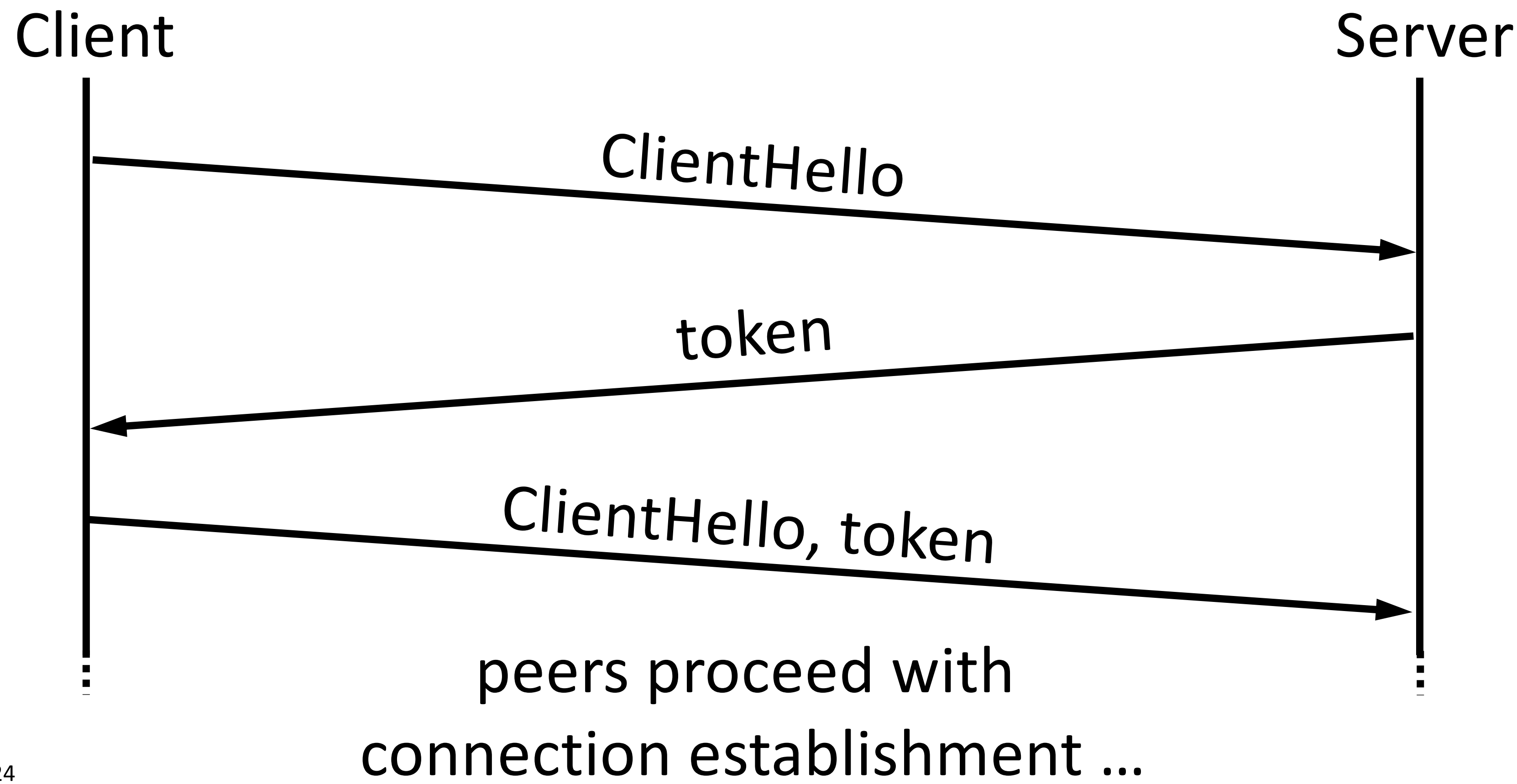
- QUIC is going to replace TLS over TCP in HTTP/3

- Improves problems of TLS over TCP
  - Protocol Entrenchment
  - Implementation Entrenchment
  - Handshake Delay
  - Head-of-line Blocking
  - Mobility

- Google's QUIC protocol is already widely deployed on the Internet
  - Accounts for 7% of global Internet traffic
  - Supported by Google Chrome (approx. 60% browser market share)

Tracking via Source-Address Token

- Source-address token speed up the validation of the client's IP address in subsequent connections between the same peers



Tracking via QUIC’s Server Config

- QUIC’s server config contains a public key used to bootstrap the cryptographic connection establishment
- Client reuses server config across different connections
- Tracking feasible if server distributes unique server configs/ server config identifiers to its clients

Tracking via QUIC

- Main findings[5]
  - Default configuration of Chrome enables unlimited tracking periods
  - Third-party tracking feasible via this mechanism for Chrome
  - Network observers may track user's via QUIC's server config

- Reactions by browser vendors
  - Google Chrome restricts feasible tracking periods to one week

5: Sy et al. "A QUIC Look at Web Tracking" (2019)

Shared Client IP Address Validation[6]

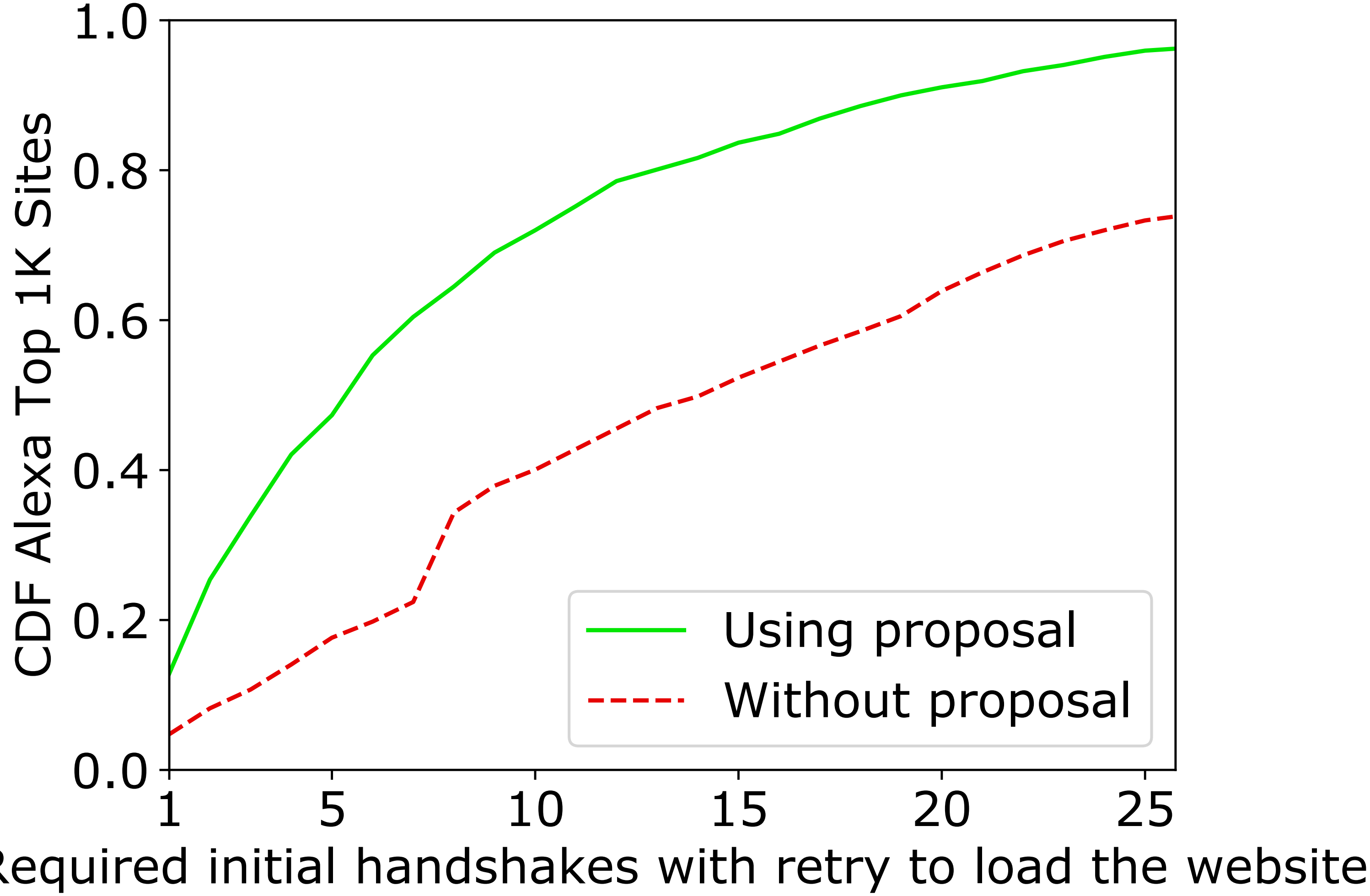
- QUIC server having a TLS trust-relation accept source-address tokens generated by each other
  - Each accepted source-address token allows client-server pair to save a round trip time during the connection establishment

- Novel QUIC transport parameter is used to inform the client about other hosts accepting a provided validation token

6: Sy, Erik “Surfing the Web Quicker Than QUIC via a Shared Address Validation” (2019)

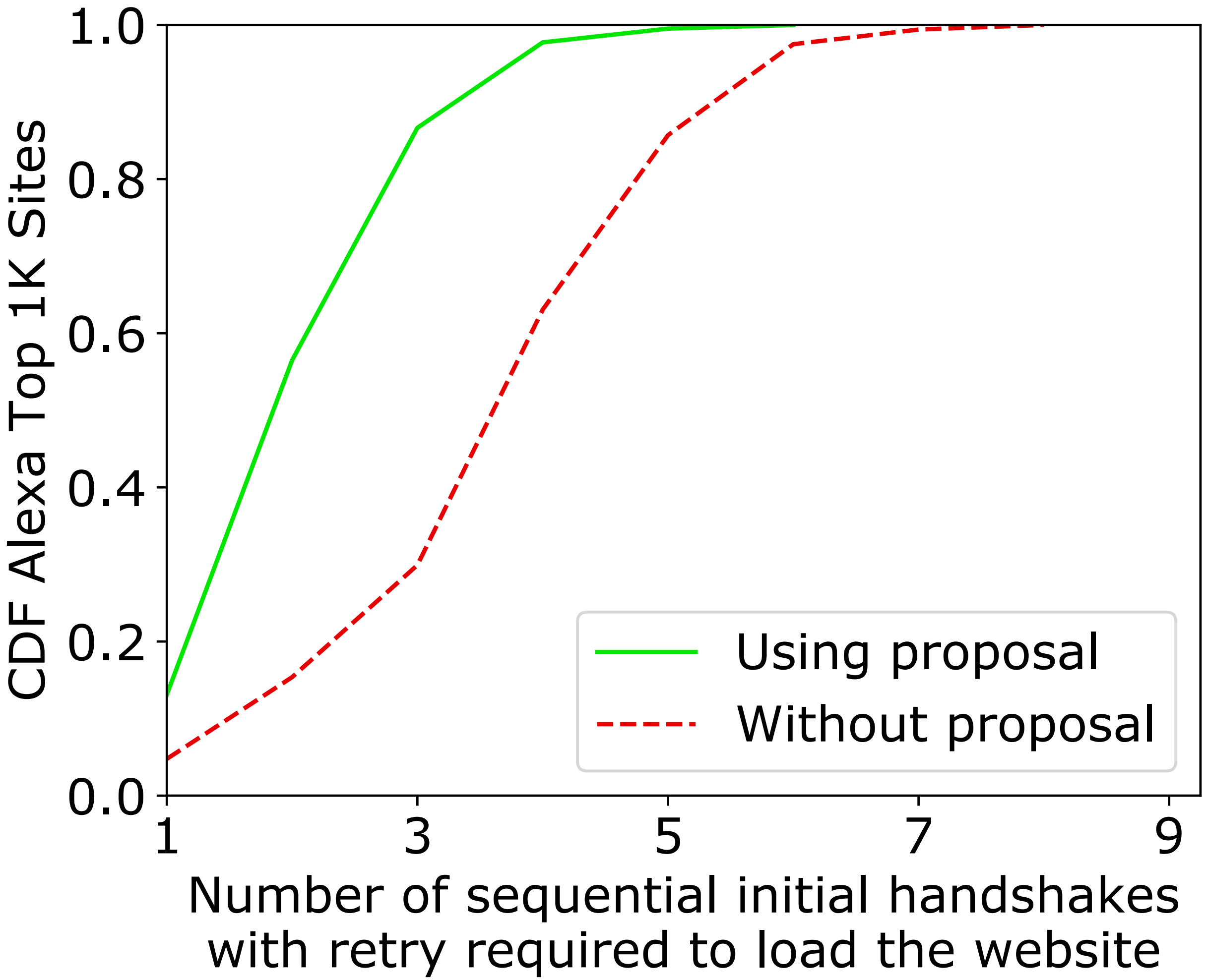
Performance Improvements for the average Website (1/2)

- Proposal saves a round-trip time on 58.75% of the established connections



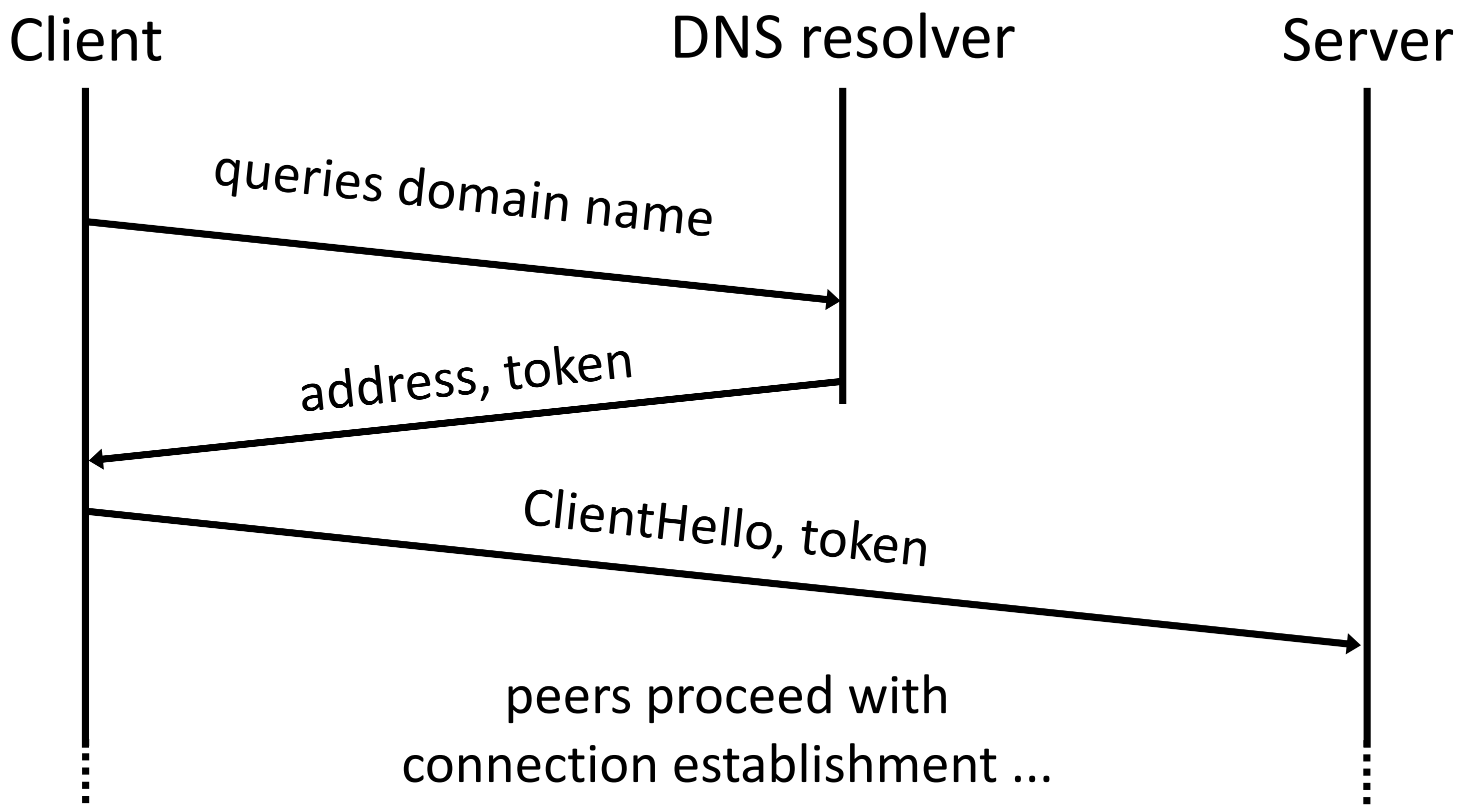
Performance Improvements for the average Website (2/2)

- Longest path of sequential connections with retry is reduced by 39.1%



Shared IP Address Validation using Out-Of-Band Token[7]

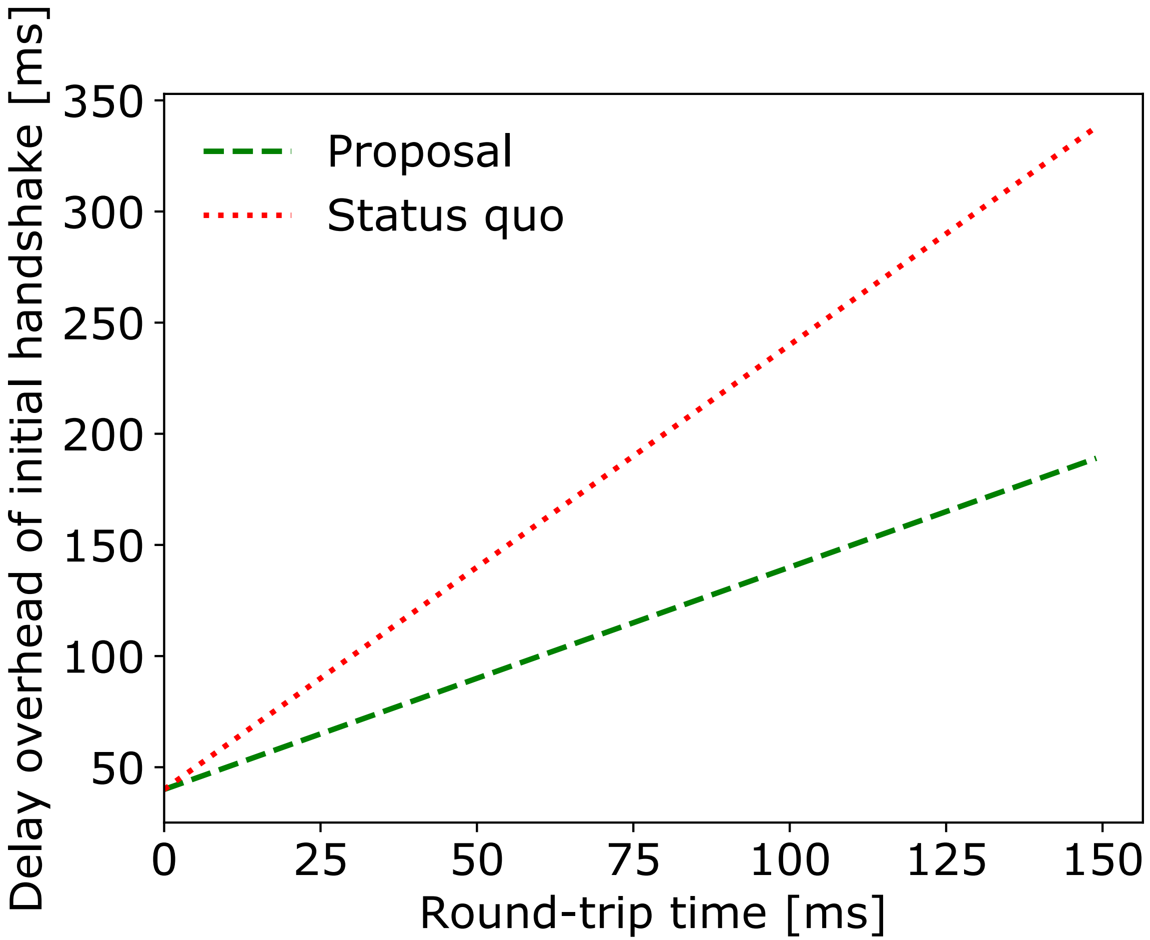
- Distribution of out-of-band validation token via DNS resolver or other QUIC server



7: Sy et al. “QUICker Connection Establishment with Out-Of-Band Validation Tokens” (2019)

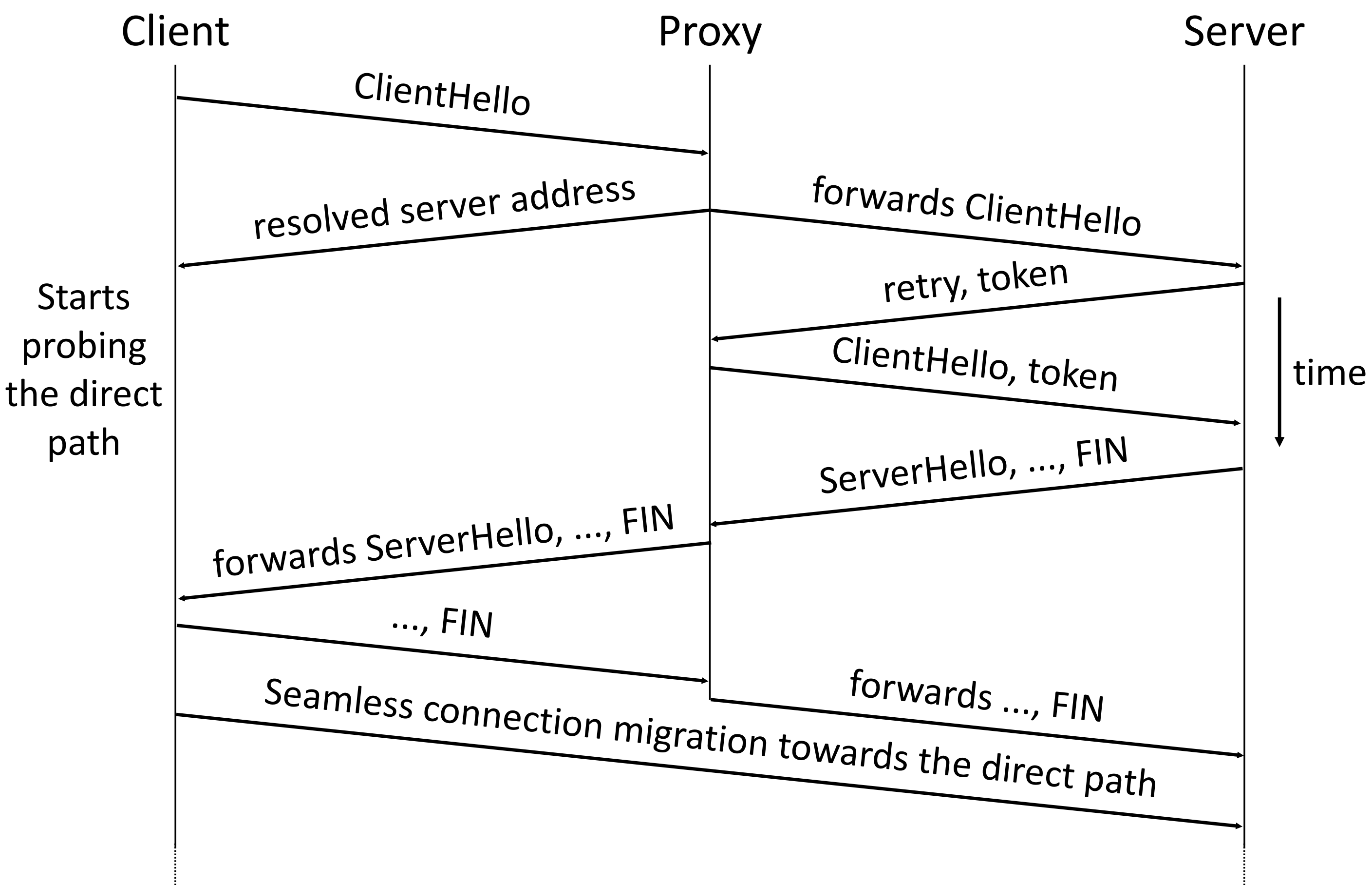
Performance gains based on Out-Of-Band Validation Token

- Each initial QUIC connection establishment can save up to a RTT



Introducing the QuicSocks Design[8]

- Assumes a QuicSocks Proxy colocated with the DNS resolver

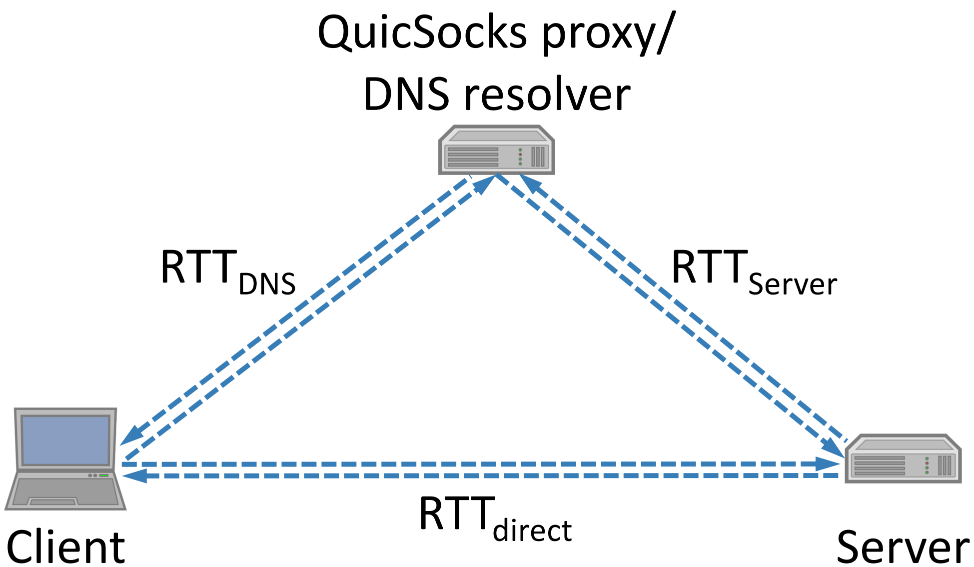


8: Sy et al. “Accelerating QUIC's Connection Establishment on High-Latency Access Networks” (2019)

Analytical Performance Evaluation

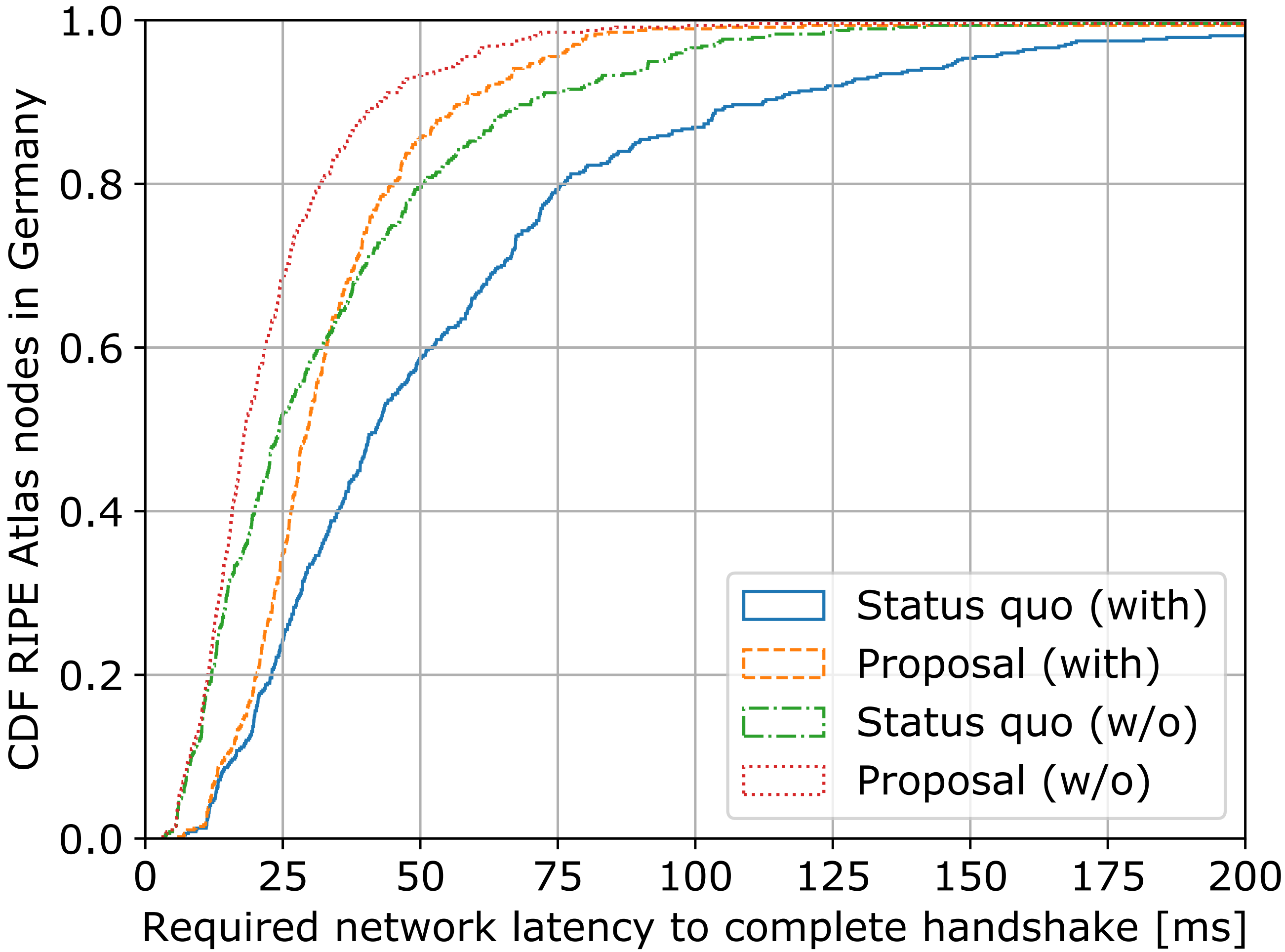
- Proposal achieves better performance if $RTT_{Server} < RTT_{direct}$

Stateless retry	Latency to establish connection (incl. DNS)	
	Status quo	Proposal
w/o	$RTT_{DNS} + RTT_{direct}$	$RTT_{DNS} + RTT_{Server}$
with	$RTT_{DNS} + 2^{*} RTT_{direct}$	$RTT_{DNS} + 2^{*} RTT_{Server}$



Empirical Performance Evaluation

- 24.3% of nodes saves at least 15ms without and 30ms with stateless retry

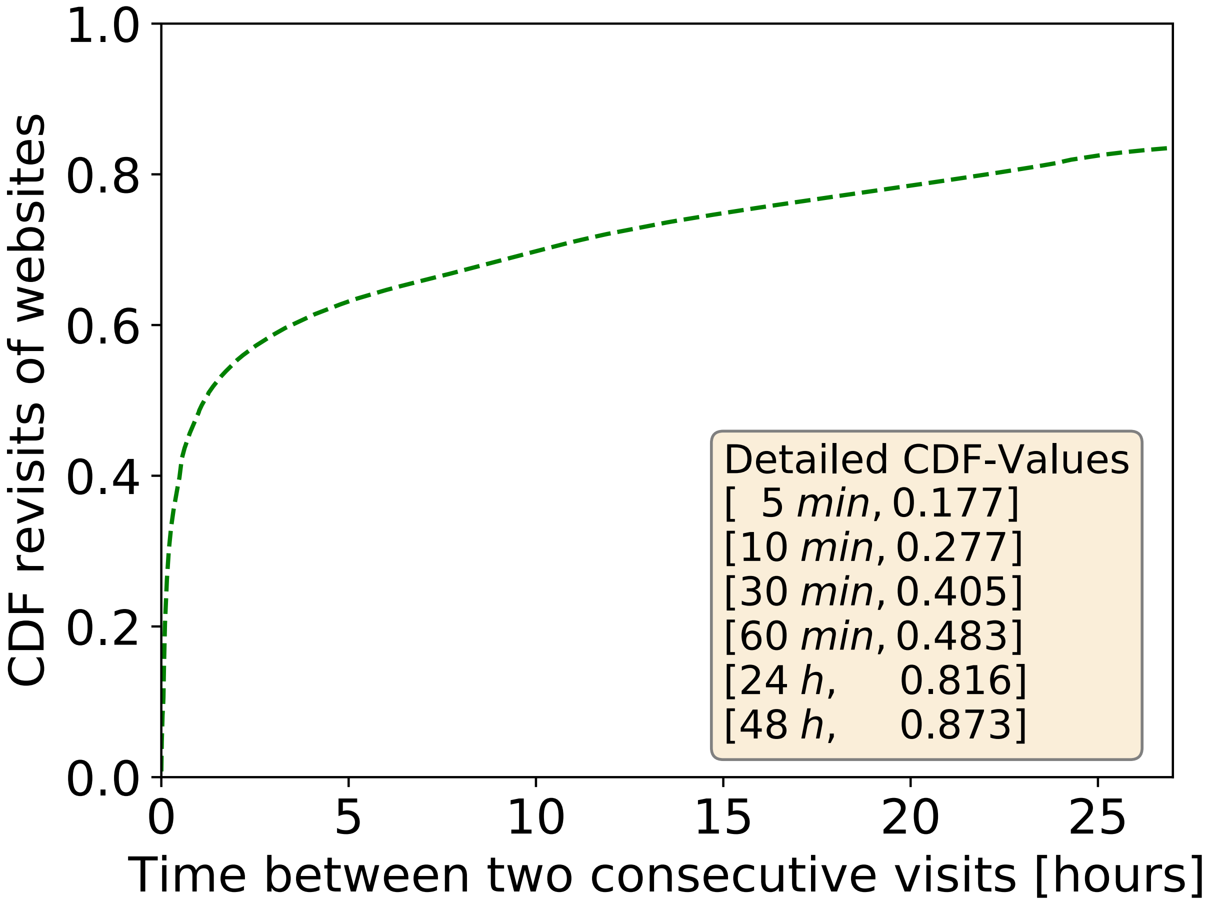


Recommended Privacy Protections

- Deactivate TCP Fast Open
- Applications restricting tracking via HTTP cookies should apply the same limitations to tracking via the presented mechanisms in TLS and QUIC
- Deploying resolver-less DNS

The Performance versus Privacy Trade-off

- Short lifetime for the investigated tracking mechanisms provides already significant performance gains while limiting feasible tracking periods



Conclusion

- TCP Fast Open, TLS, and QUIC contain mechanisms that can severely harm the privacy of users
- Popular browsers do not sufficiently protect against these privacy risks
- Investigated mechanisms should be used with a short expiration time to balance the performance versus privacy trade-off
- Several performance optimizations are feasible for core Internet protocols

Thank you

Questions and Answers

E-mail: Luebeck@erik-sy.de

Slides: https://erik-sy.de/luebeck