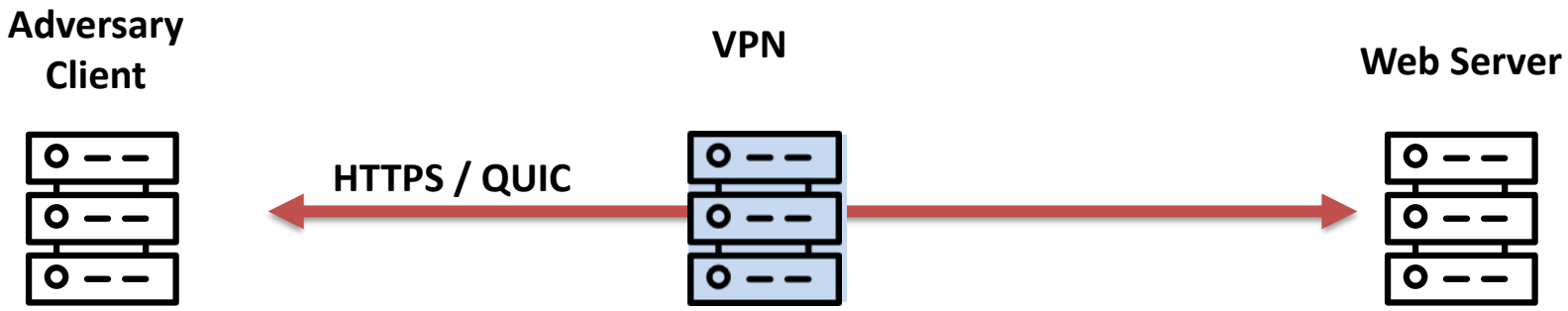
A Framework for Detecting Secure Web Traffic Over VPNs: HTTPS and QUIC



Yuan Tian, Stephen Huang
University of Houston

UNIVERSITY of HOUSTON

Outline

1. Introduction: Cybersecurity and Anonymity Networks
2. Secure HTTP Protocols
3. Solution Based on Discrepancy of RTT
   a. HTTPS
   b. QUIC
4. Validation and Performance
5. Conclusion

VPNs were once the standard for remote access, but today they are one of the most exploited entry points for cyberattacks.

1. Introduction

- Typically, an anonymity network (VPN or TOR) is used to hide the identity of the hacker.
- There are several recent attacks using anonymity networks and geo-spoofing.
  - Workers loyal to North Korea have infiltrated global organizations via a fraudulent remote work scheme, violating international sanctions and posing cybersecurity risks to employers.
- Geo-spoofing allows attackers to appear in trusted regions.
- Avoiding Geo-Restriction to access services from permitted jurisdictions.

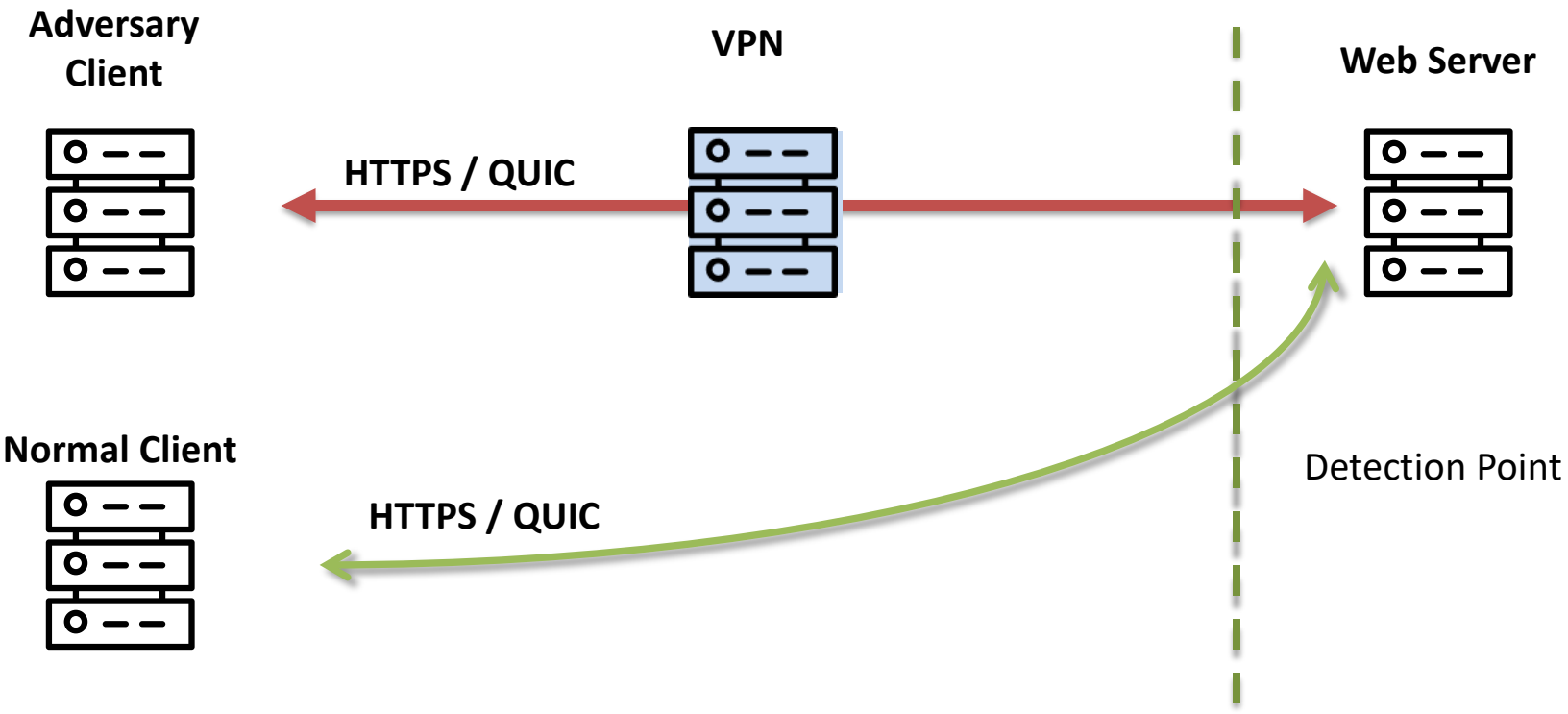
Laptop Farm

Cyber Security | Cyber Security News

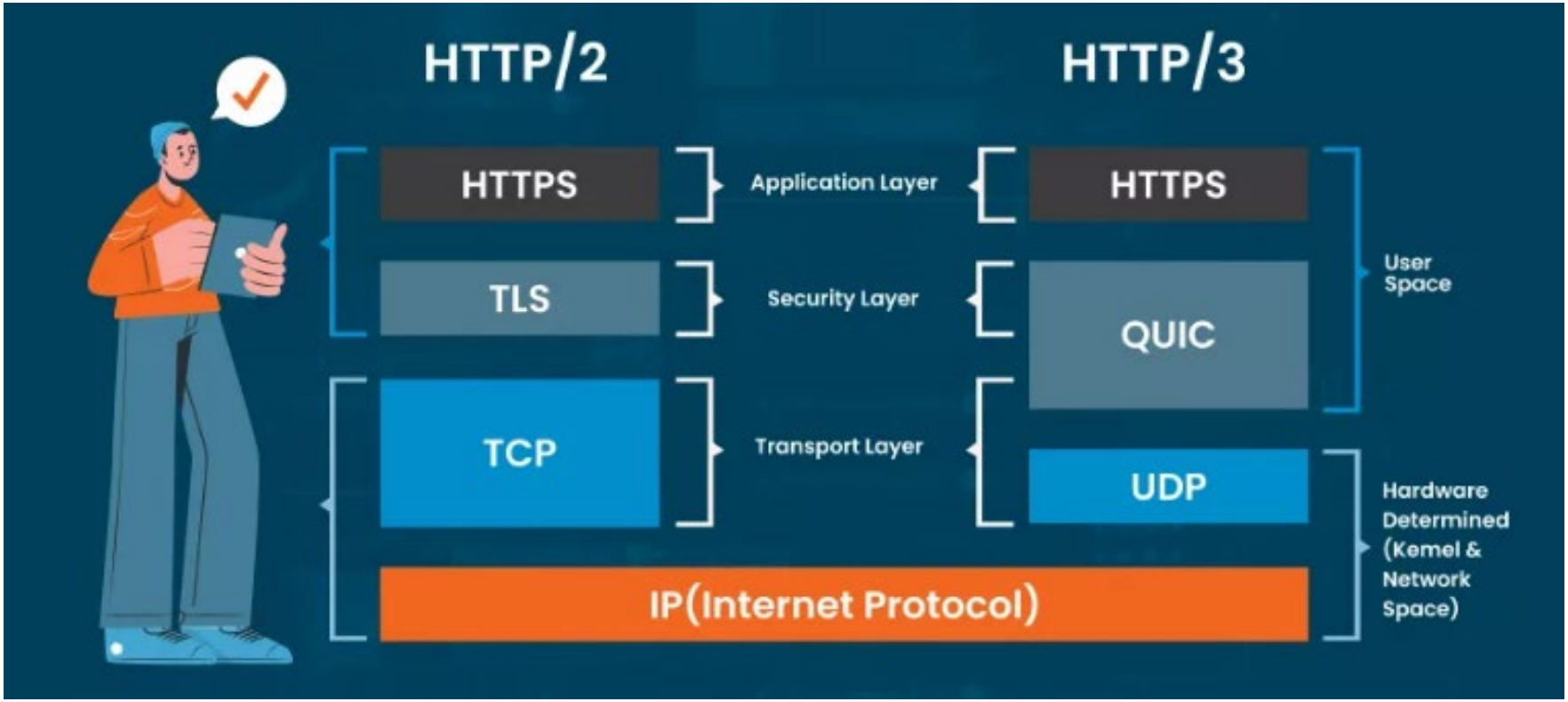
U.S. DOJ and Microsoft Dismantle 29 Laptop Farms and 200 Computers Used by North Korean IT Workers

By **Kaaviva** - Iulv 1. 2025

Classification Problem



2. Secure HTTP Protocols



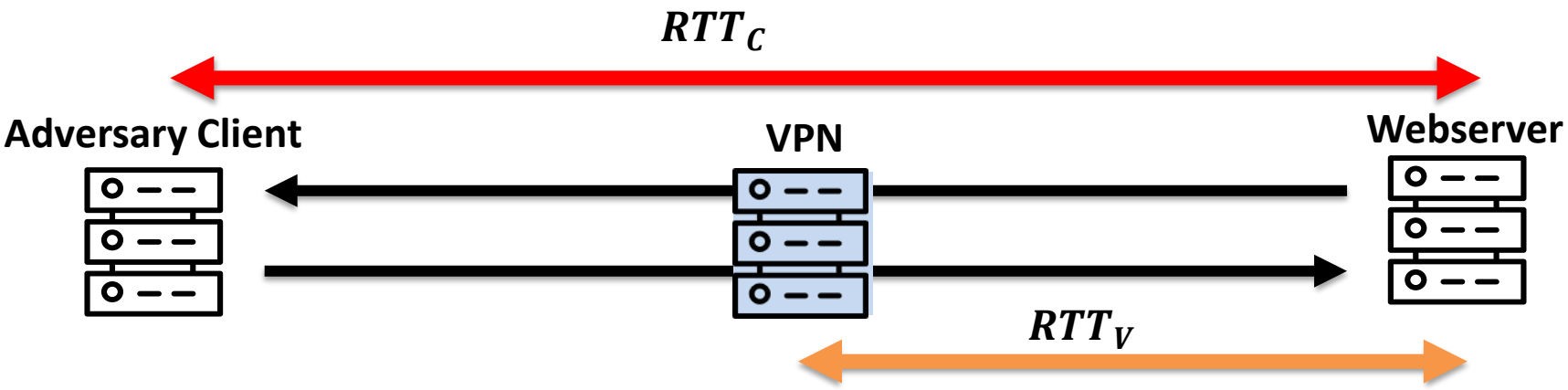
https://cheapsslweb.com/blog/what-are-quic-and-http-3-difference-between-http-3-and-http-2/

Virtual Private Network (VPN)

- Our worst-case assumption:
  - Unlike the TOR network, for which there is only one, a VPN can be installed on a "private" machine, making it stealthy.
  - Turning off all network contacts of the VPN, such as ping and traceroute.
- An active probe can be initiated, which can find a reasonable estimate of the round-trip time (RTT).
  - Use traceroute to the nearest neighbor.

RTT Discrepancy Ratio

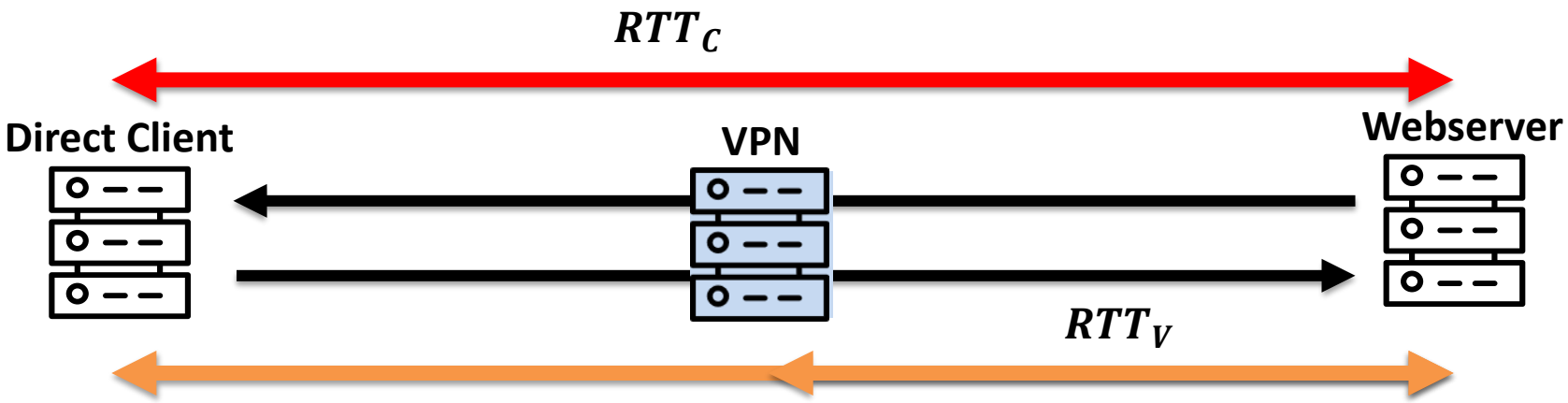
- RTT_C: RTT between the web server and the client
- RTT_V: RTT between the web server and the VPN



- With VPN: $\frac{RTT_V}{RTT_c} << 1$

RTT Discrepancy Ratio

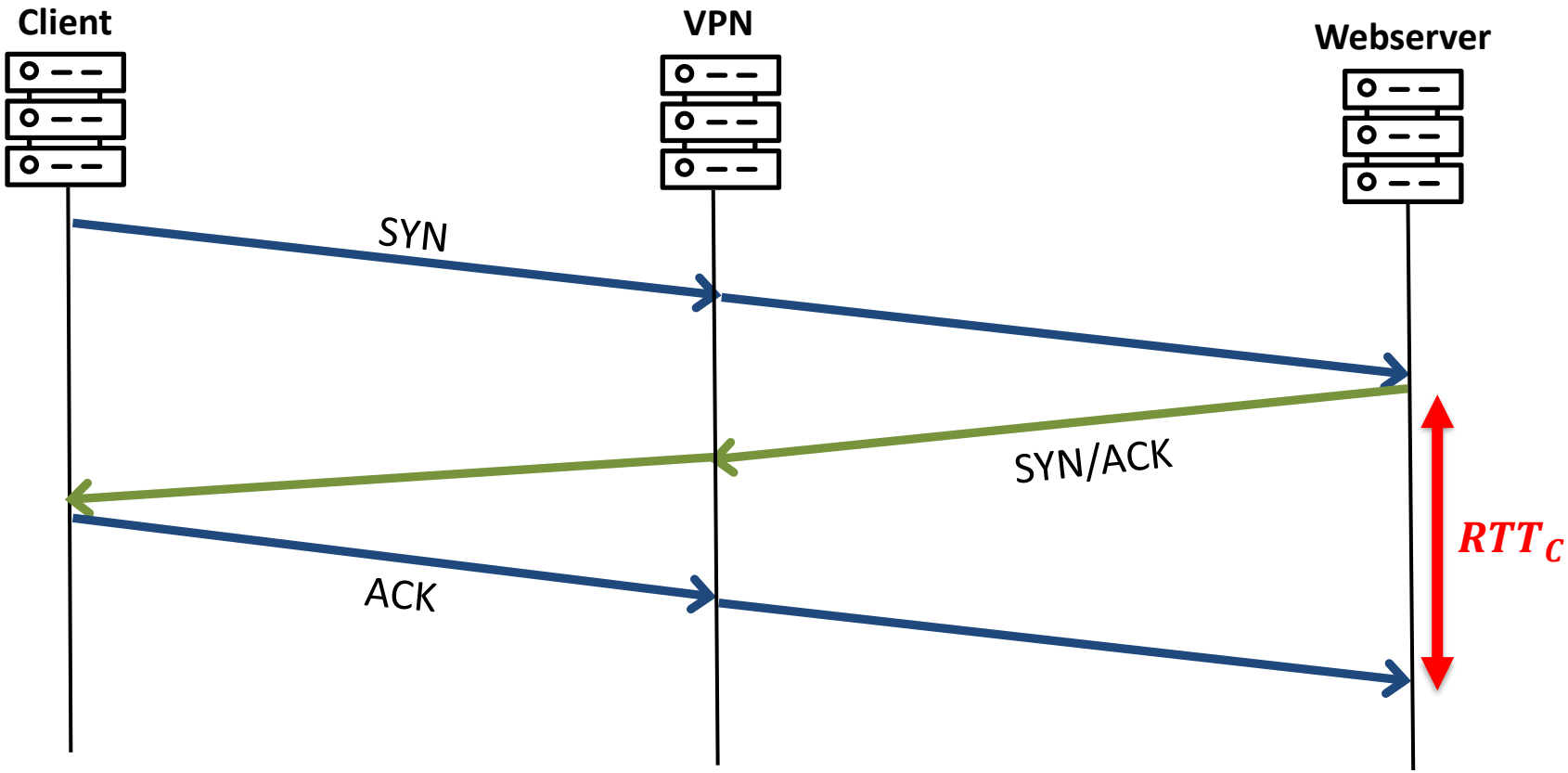
- The ratio can be used to discriminate whether there is a VPN.



- Without VPN: $\frac{RTT_V}{RTT_C} \sim 1$

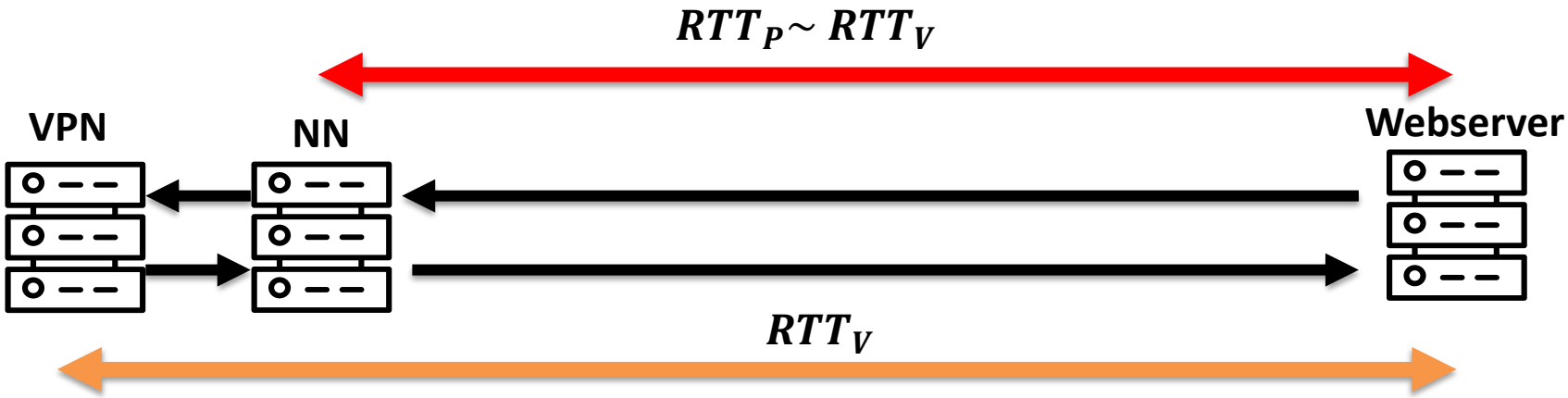
Calculating RTT_C

- The TCP three-way handshake provided the RTT_C on the web server (w/ or w/o VPN).



Active Probe for RTT_V

- To calculate the RTT, we need an ACK from the VPN. If the VPN does not send an ACK, we cannot measure the RTT_V.
- If a traceroute does not reach the VPN, it will get to the nearest neighbor.
- Experiment shows that the difference is minimal.

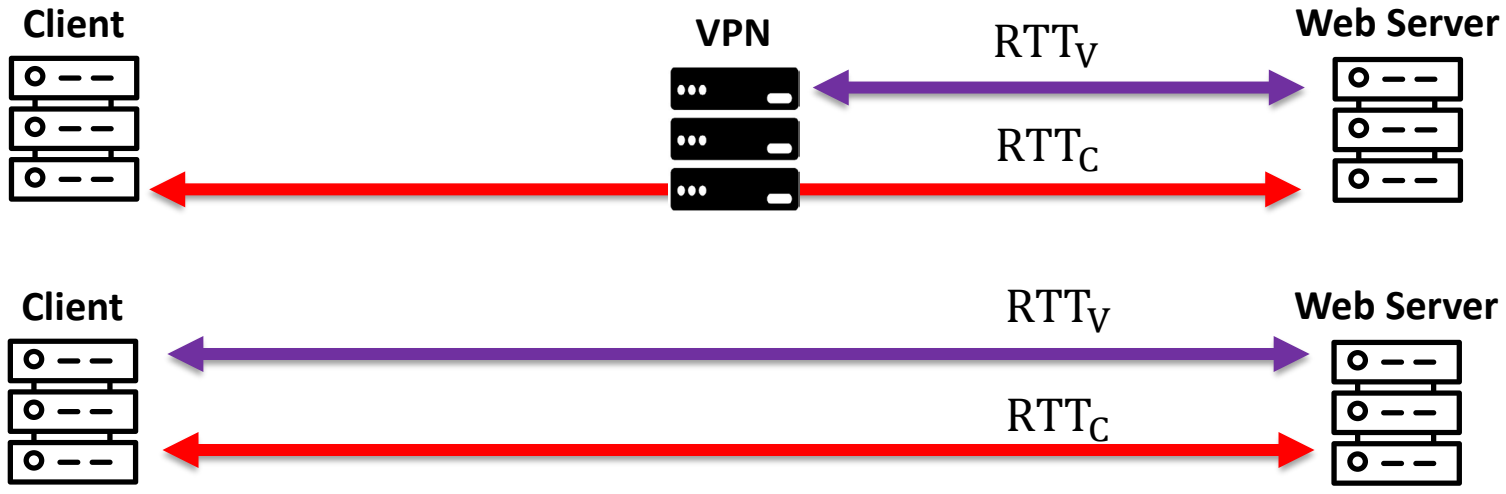


VPN Configurations

- In most VPN systems, a proxy (or a load balancer) is deployed before the VPN backend.
  - This will alter the TCP three-way handshake behaviors so that the proxy does not reflect end-to-end latency between the client and the web server
  - QUIC is not a TCP-based protocol; thus, the proxy has no impact on QUIC.

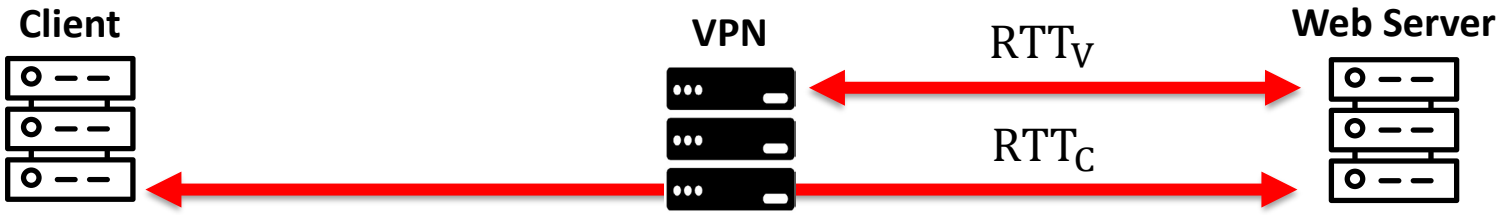
3. Solution Based on Discrepancy

- Question: Did the client connect to the web through a VPN?
- Our solution is based on the discrepancy of RTT: RTT_V vs. RTT_C
- How do we measure the two values?



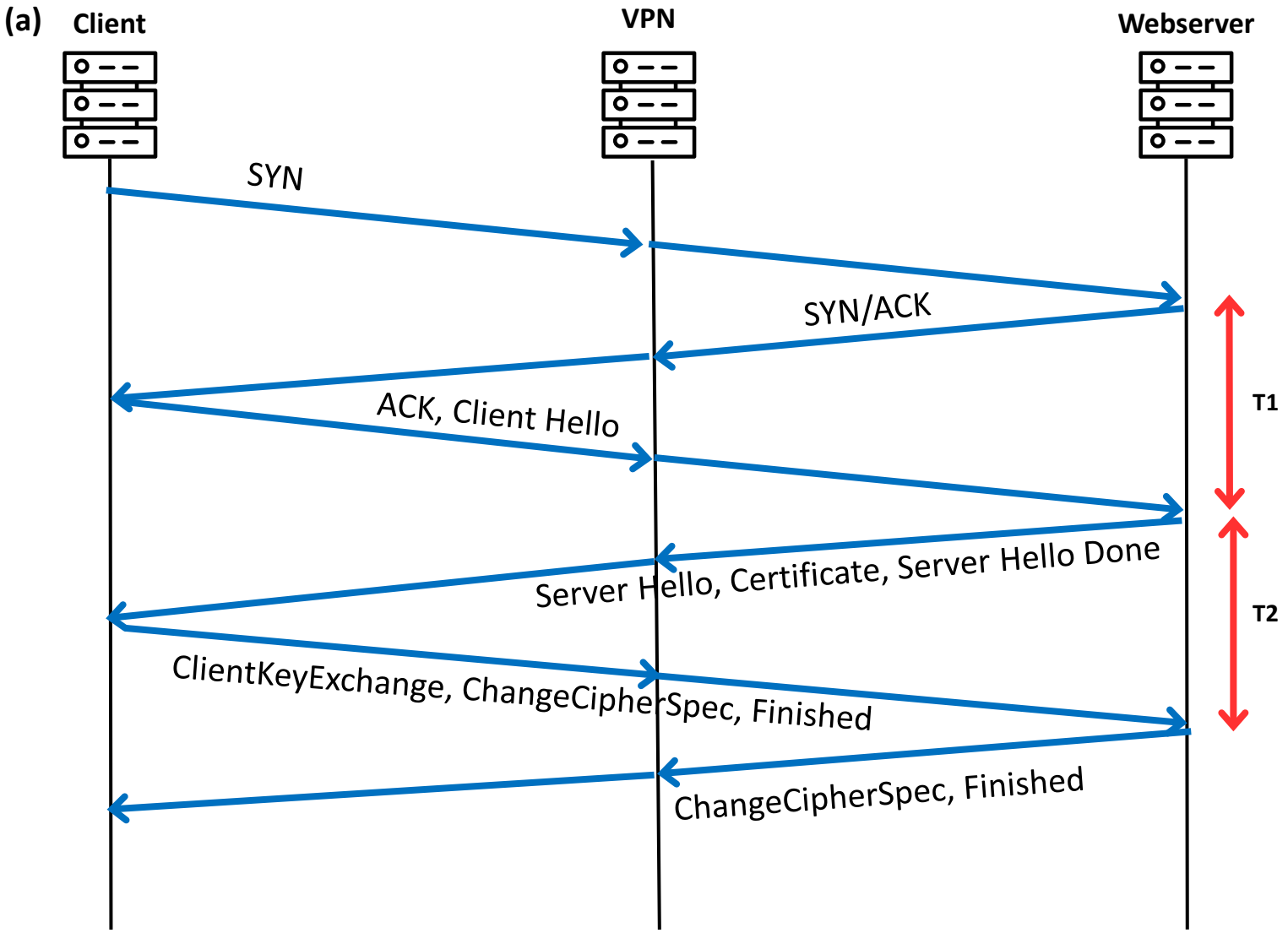
(a) HTTPS

- All we need to do is to find a good estimation of RTT_V and RTT_C
  - We can obtain RTT_C from TCP and HTTPS protocol handshake.
  - We can use the Active Probe to find an approximate value of $RTT_P \approx RTT_V$.

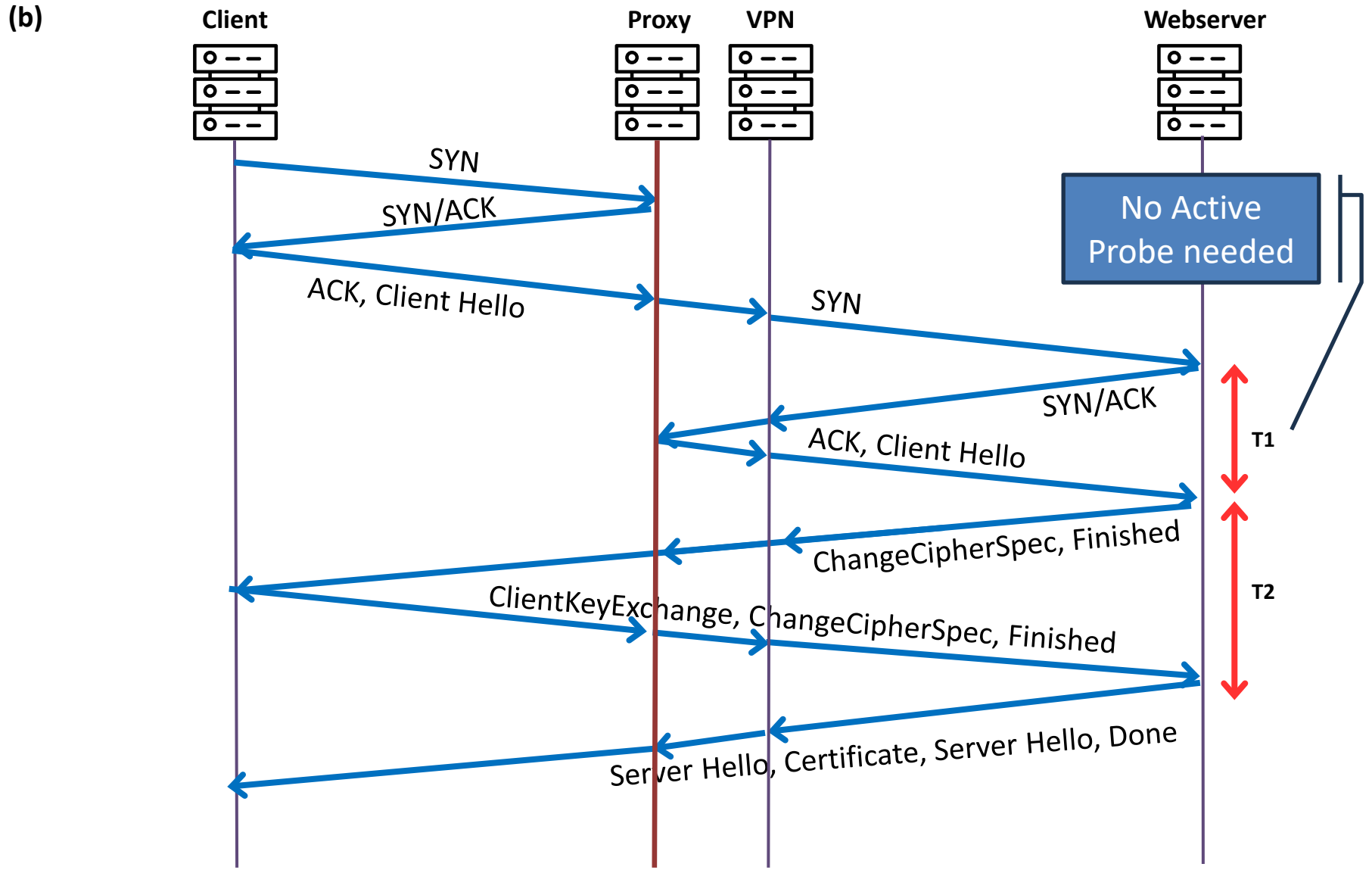


- However, VPN configuration complicated the algorithm
  - Stand-alone VPN
  - VPN with a Proxy server

(a.1) HTTPS VPN


(a)
Client
VPN
Webserver
SYN
SYN/ACK
ACK, Client Hello
Server Hello, Certificate, Server Hello Done
ClientKeyExchange, ChangeCipherSpec, Finished
ChangeCipherSpec, Finished
T1
T2

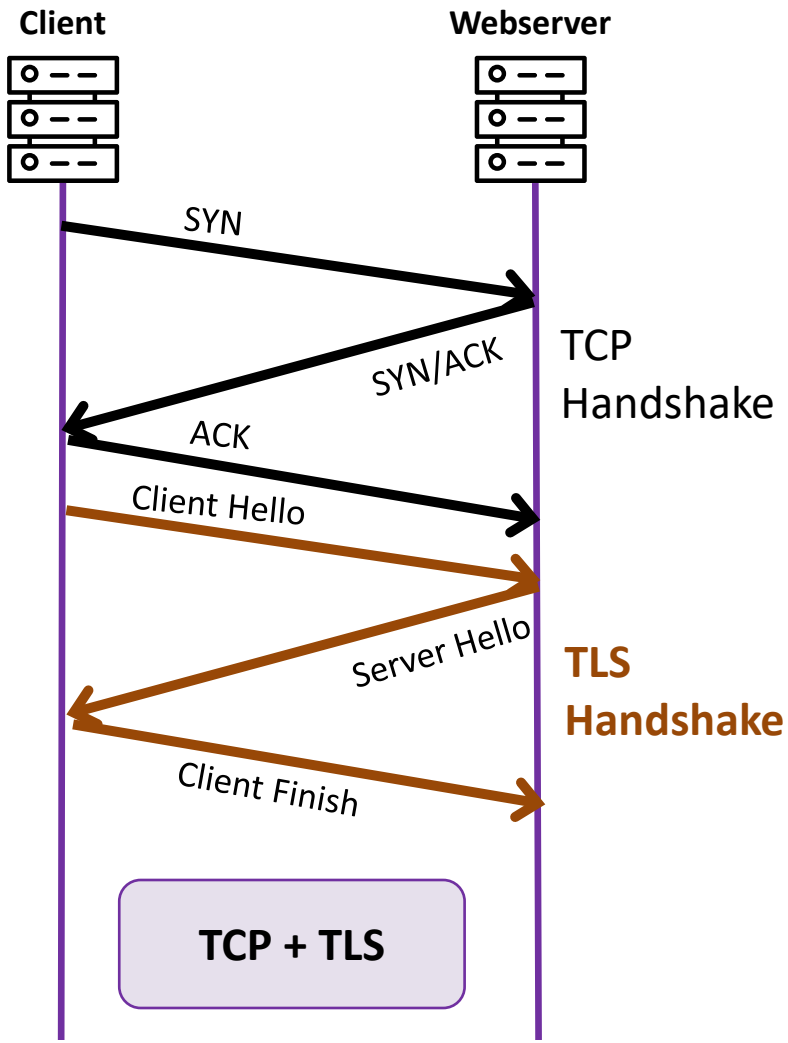
(a.2) HTTPS VPN w/ Proxy

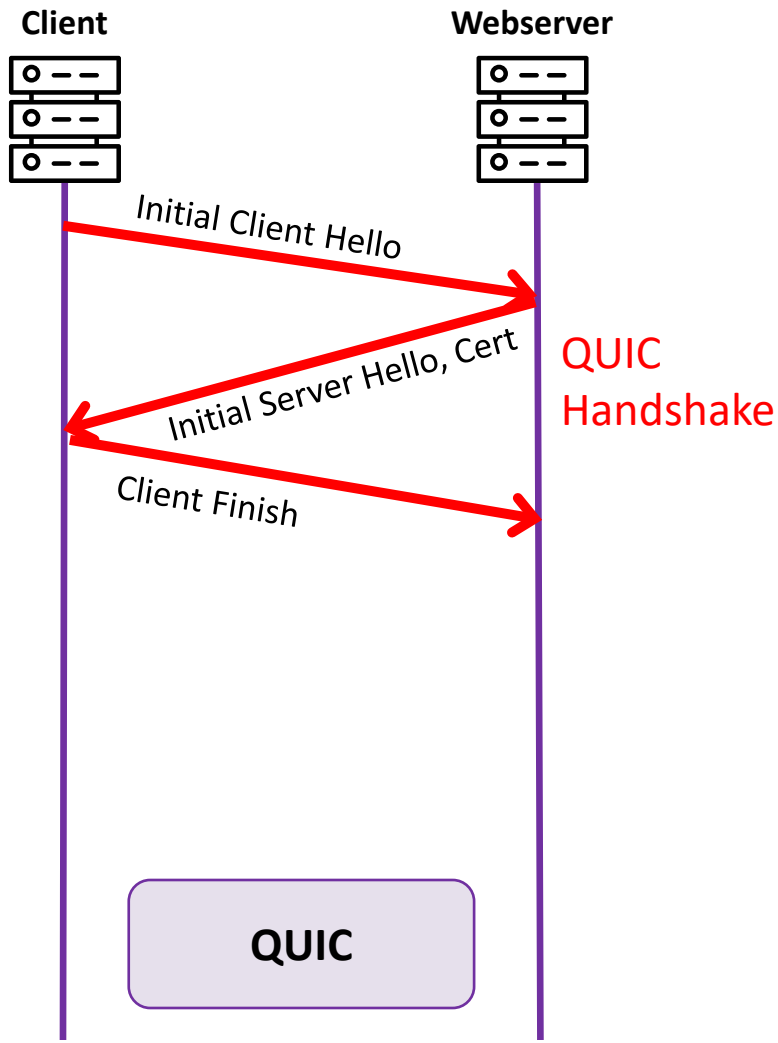


(b) QUIC

- QUIC stands for Quick UDP Internet Connections.
- QUIC effectively packs more application-level information into each UDP packet than HTTP/2 does into TCP packets. (Efficiency)
- The first successful connection between a QUIC client and server is always a 1-RTT handshake.
- Subsequent connections may be 0-RTT if the condition is right.

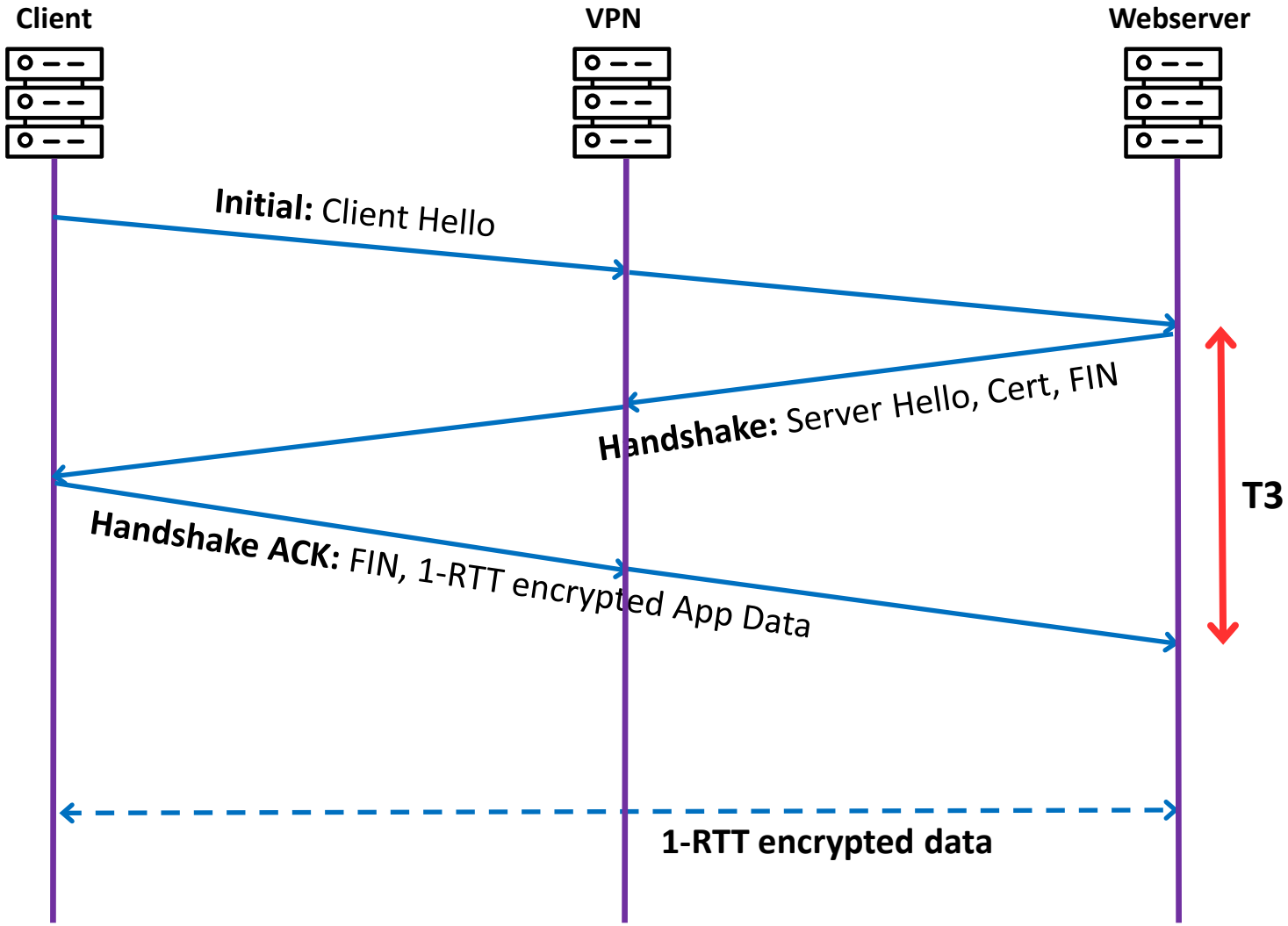
QUIC vs TCP+TLS





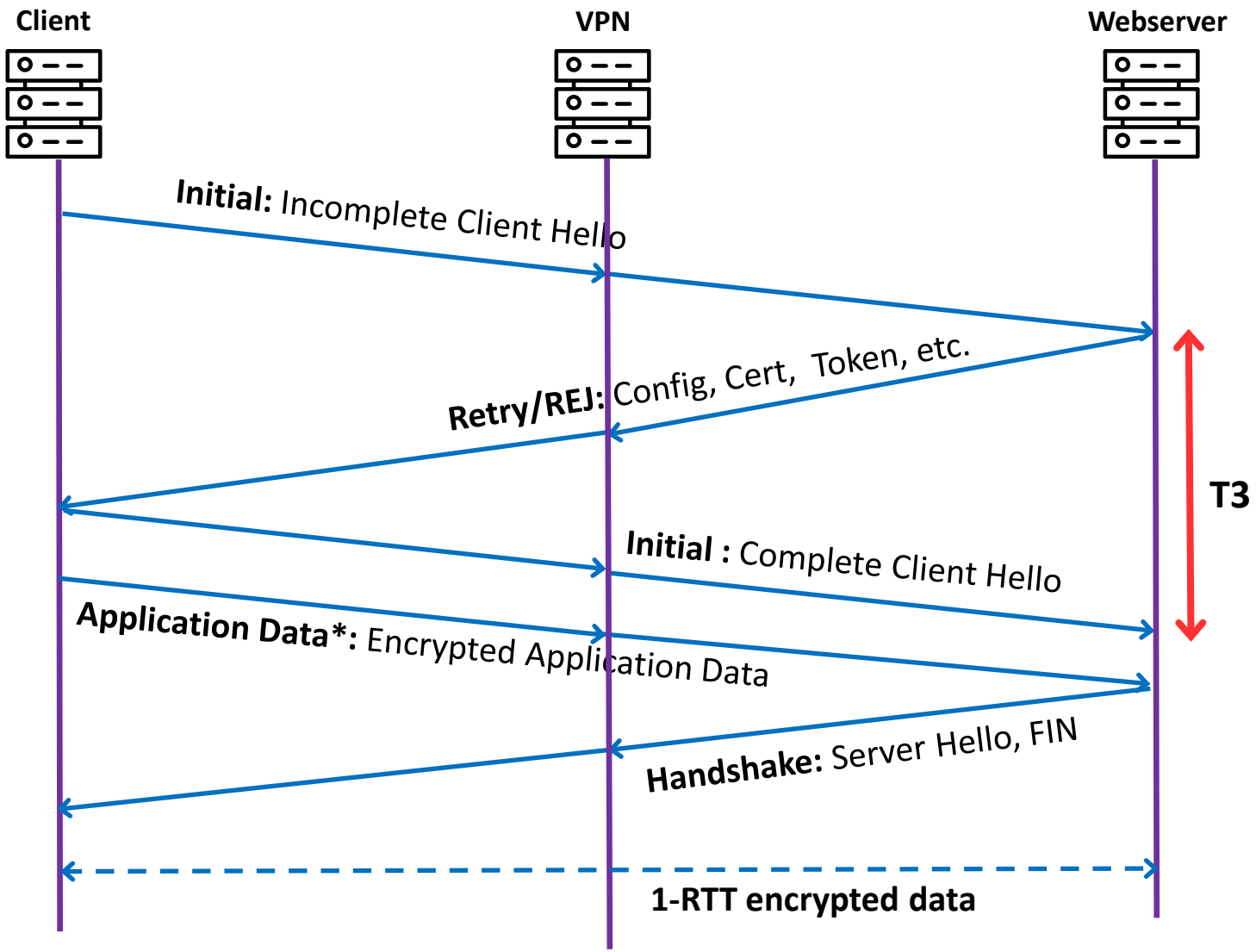
QUIC 1-RTT

(a) Standard 1-RTT QUIC handshake.



QUIC 1-RTT w/ Retry

(b) 1-RTT QUIC handshake with Retry.



Machine Learning Features

ML Attributes for HTTPS: (for w/ or w/o proxy)

- **T1**: RTT from the TCP three-way handshake
- **T2**: RTT from the TLS handshake
- **RTT_P**: RTT measured via traceroute-based active probing
- **RTT_P/T1**: discrepancy ratio between RTT_P and T1
- **RTT_P/T2**: discrepancy ratio between RTT_P and T2

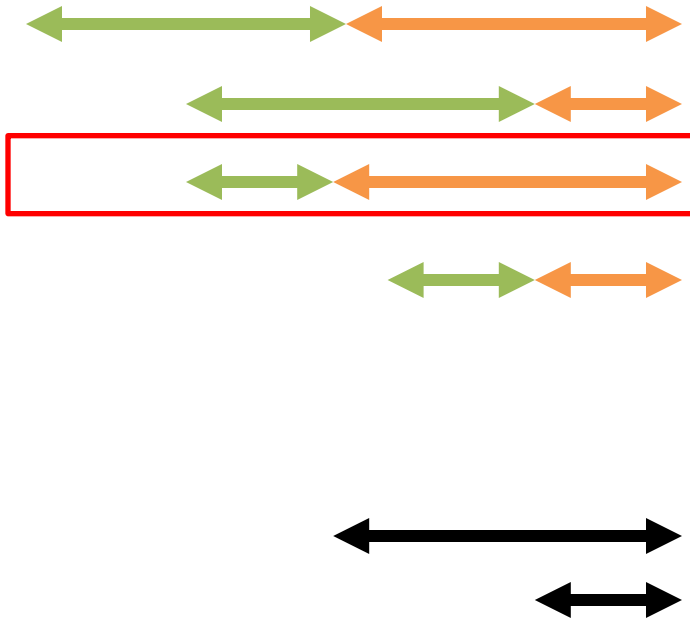
ML Attributes for QUIC:

- **T3:** TLS-layer RTT derived from the QUIC handshake
- **RTT_P:** RTT measured via traceroute-based active probing
- **RTT_P/T3**: discrepancy ratio between RTT_P and T3.

- Note that both absolute and relative numbers are used.

4. Validation and Performance

- Using the RTT ratio may be sensitive to slight variations in RTTs.
- We tested four extreme cases of connection ratios.



VPN		
Config Type	# Config	# Data Point
Long-Long	60	300
Long-Short	60	300
Short-Long	120	600
Short-Short	60	300
Total	300	1500

Direct Connection		
Config Type	# Config	# Data Point
Long	150	750
Short	150	750
Total	300	1500

Classification Accuracy

HTTPS

ML Model		Accuracy	F1-score	Precision	Recall
SVM	RBF	0.9910	0.9909	0.9946	0.9873
	Linear	**0.9943**	**0.9943**	**0.9960**	**0.9926**
	Poly	0.9853	0.9853	0.9867	0.9840
	Sigmoid	0.8997	0.8989	0.9041	0.8943
Random Forest		**0.9963**	**0.9963**	**0.9973**	**0.9953**
Naïve Bayes		0.9890	0.9775	0.9925	0.9630
Logistic Regre.		0.9897	0.9777	0.9923	0.9637
Neural Net		**0.9967**	**0.9967**	**0.9980**	**0.9967**
XGBoost		**0.9963**	**0.9963**	**0.9980**	**0.9947**

Classification Accuracy

QUIC (HTTP/3)

ML Model		Accuracy	F1-score	Precision	Recall
SVM	RBF	0.9446	0.9428	0.9428	0.9127
	Linear	**0.9355**	**0.9320**	**0.9824**	**0.8873**
	Poly	0.9151	0.9131	0.9466	0.8842
	Sigmoid	0.8555	0.8536	0.8797	0.8315
Random Forest		**0.9553**	**0.9549**	**0.9638**	**0.9467**
Naïve Bayes		0.8890	0.8685	0.9265	0.8175
Logistic Regre.		0.8903	0.8685	0.9280	0.8163
Neural Net		**0.9480**	**0.9471**	**0.9661**	**0.9293**
XGBoost		**0.9527**	**0.9525**	**0.9419**	**0.9733**

5. Conclusion

- This paper developed an innovative algorithm that detects secure HTTP traffic through a VPN, all while keeping the VPN server's identity under wraps.
  - Different VPN configuration (w/ or w/o proxy)
  - Different Secure HTTP protocol (HTTPS vs QUIC)
- The algorithm is real-time; it completes the detection before any application data is transmitted.
- We found very high detection rates on selected extreme cases using several ML algorithms. The testing was done using several commercial VPN servers.
- The detection algorithm runs on the web server; thus, it can detect geo-spoofing, prevent copyright and licensing infringement, but is not likely to be used to impose censorship.

Thank You

Stephen Huang

SHuang@cs.uh.edu