

Detecting VPN Traffic in Real-Time with Active Probing

Yuan Tian, Zechun Cao, Stephen Huang
University of Houston

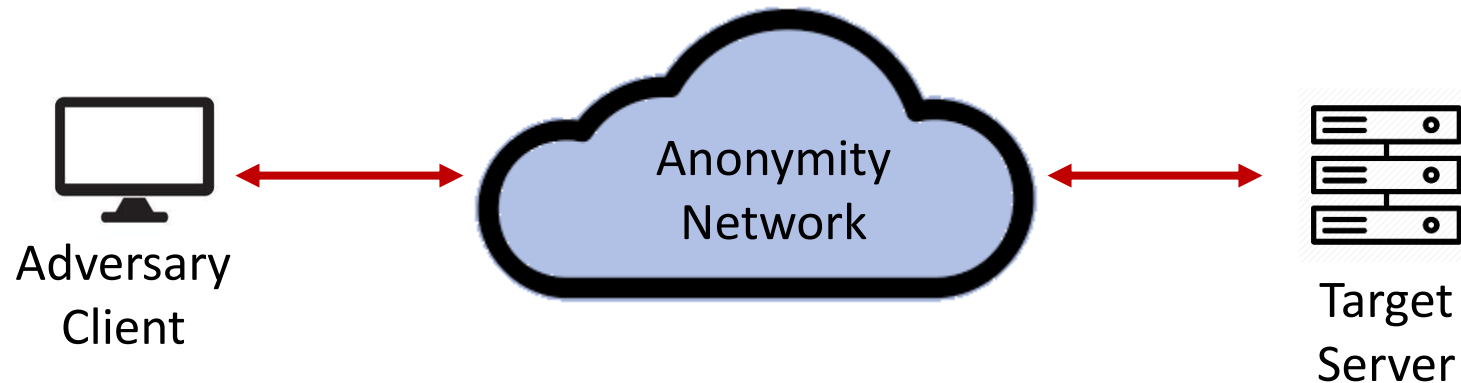
NCA

October 24, 2024

Outline

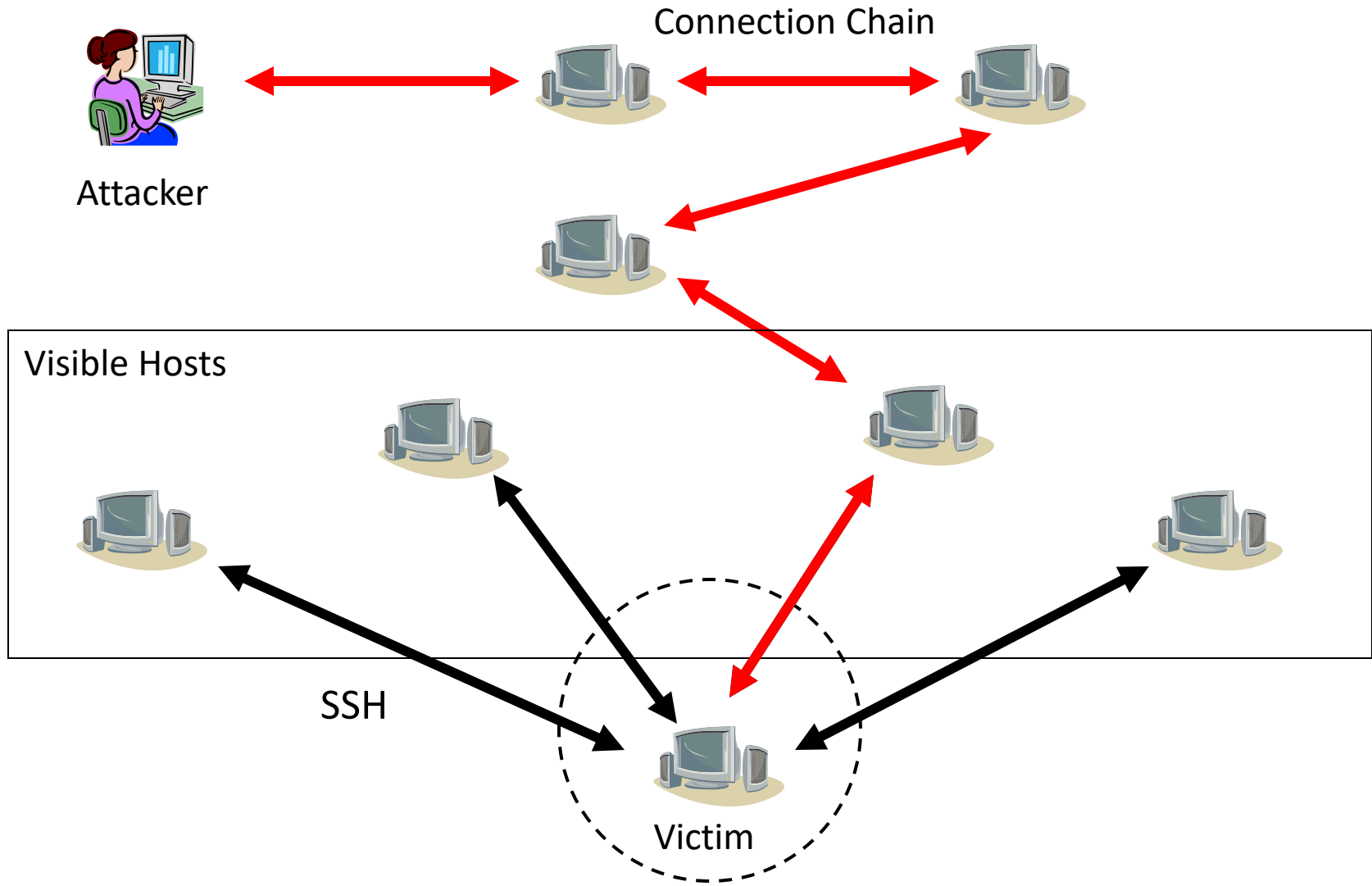
1. The Problem: Anonymity Network Hides Identity
2. Proposed Solution: Detect Suspicious Connections via VPN
3. Validation and Result
4. Conclusion

1. Anonymity Network

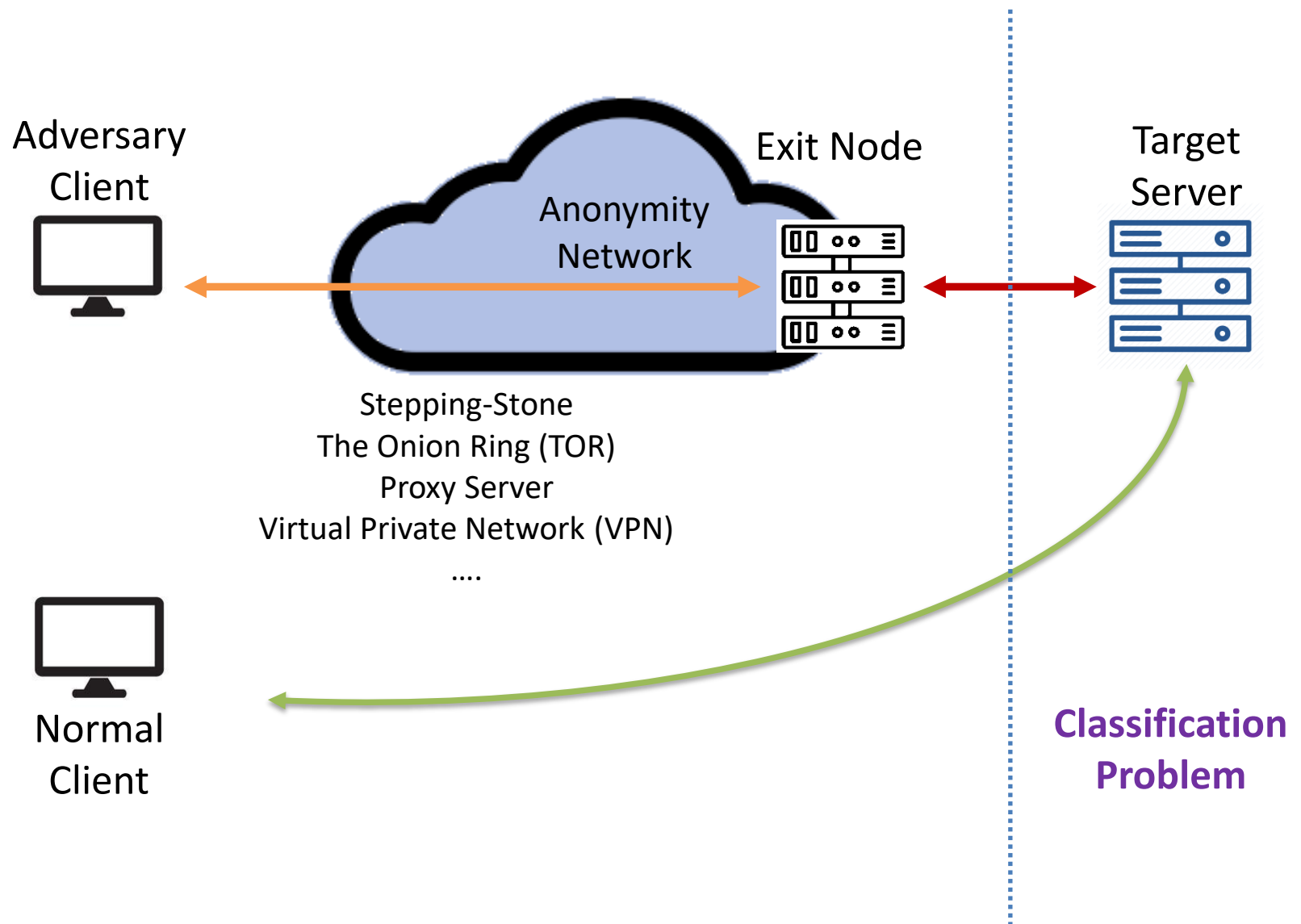


- An anonymity network was designed to provide privacy protection for the identity of the users.
- However, it may be used by intruders to hide their IP addresses.

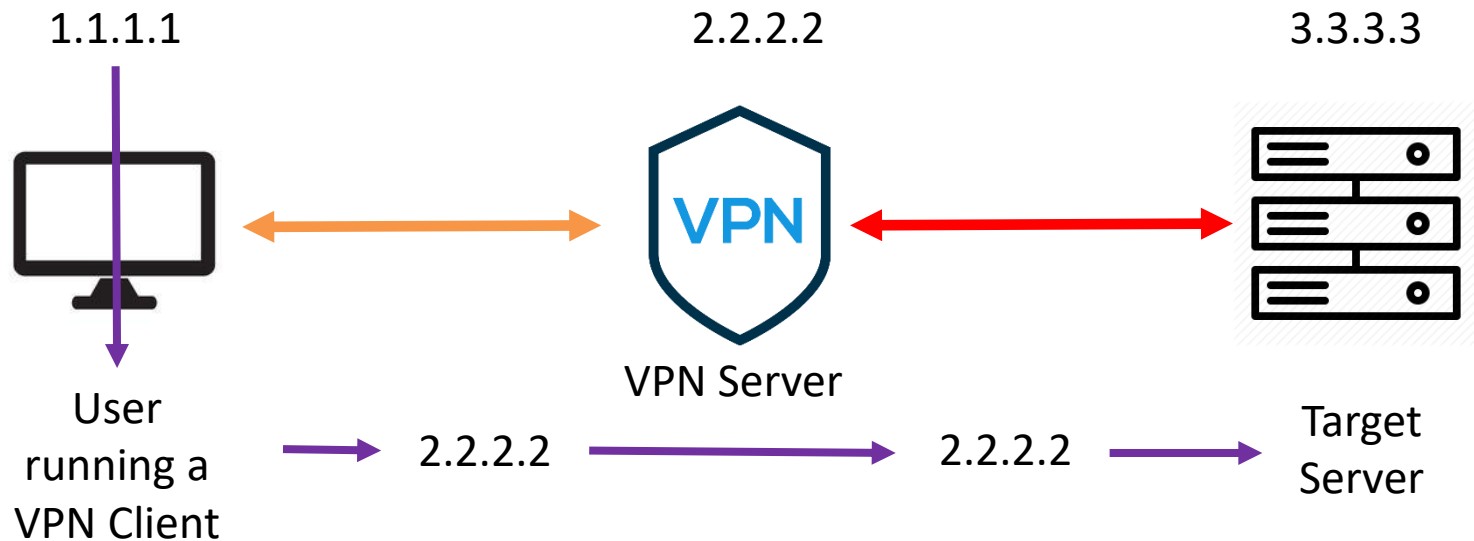
Stepping-Stone Network



Intrusion Detection Model



VPN as an Anonymity Network

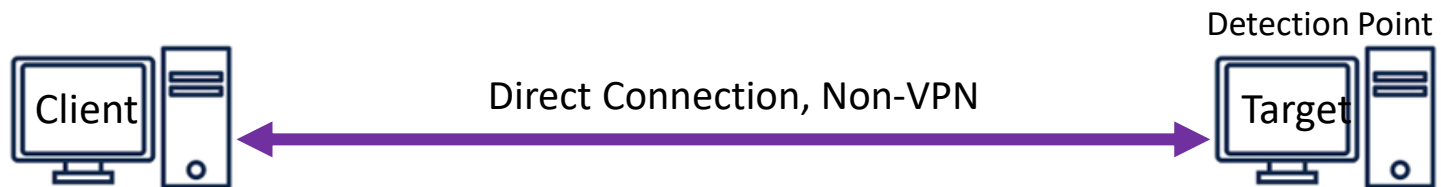


Routing SSH Through VPN

- TCP applications
 - Geo-spoofing: HTTPS
 - Fake user's location to gain particular privilege
 - Intrusion into servers: SSH
 - Data breach
 - Installing Malware
 - Ask for Ransom
 - Financial Fraud
- We chose SSH as our first application for detection because the damage is more severe.

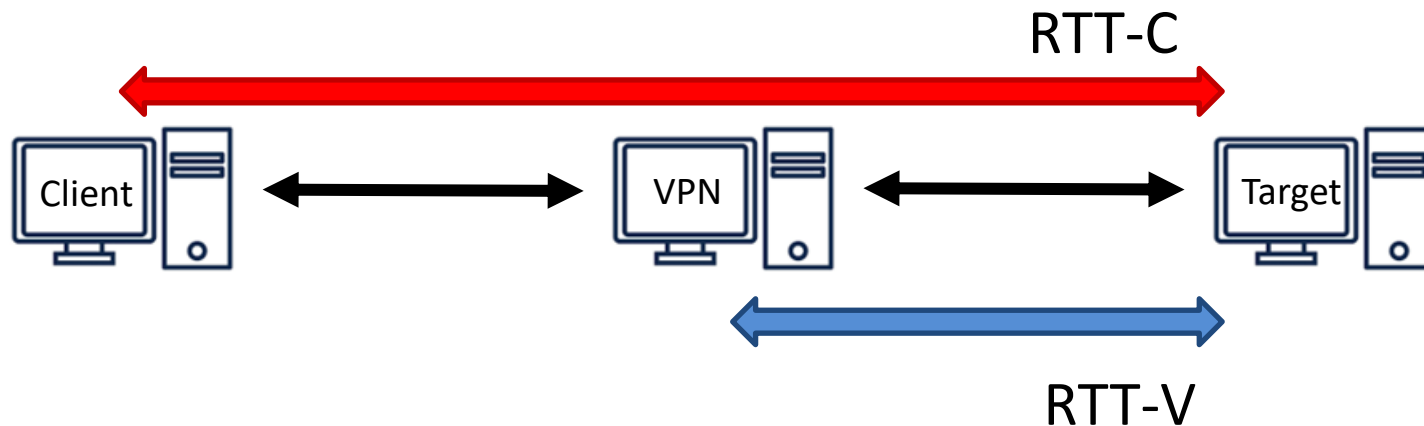
2. Proposed Solution

- Our solution is based on finding a discrepancy in the behavior of the network packets.
- We use Round-Trip Time (RTT) as a surrogate for network distance.
- If the client is connected to the Target server directly, the RTTs between them should not differ significantly.



RTT as a Measurement

- The Target server must exchange information with both the Client and the VPN.
- Routing through a VPN creates two RTTs from the Target server.
 - Target -> VPN -> Client -> VPN -> Target
 - Target -> VPN -> Target



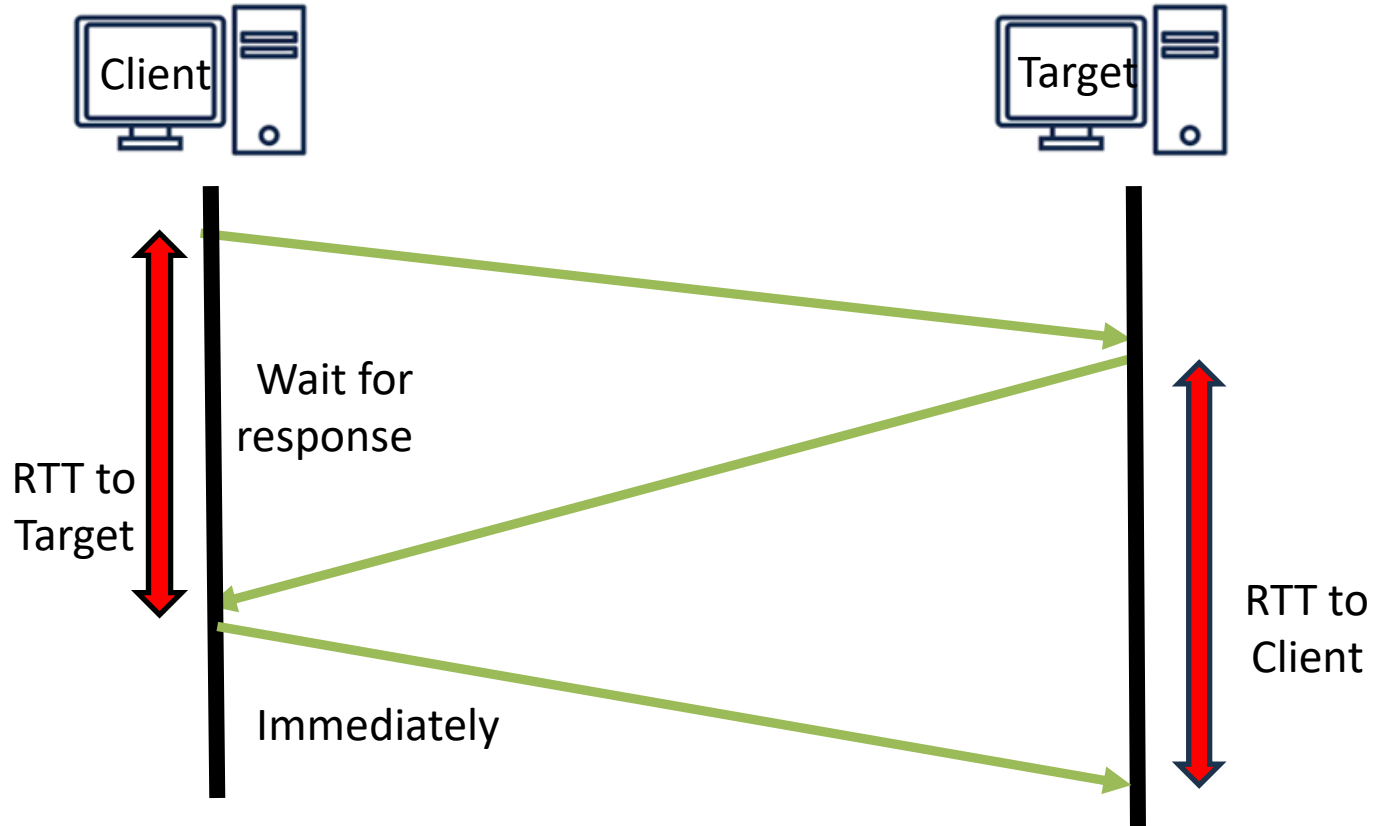
RTT Discrepancy

- The problem is reduced to compute RTT-V and RTT-C.
 - RTT-C is easier to measure since all packets sent to the VPN by the server are forwarded to the client.
 - RTT-V is more challenging.
- Since we do not know if there is a VPN, we must treat every connection as if it is a VPN.
- If there is a significant discrepancy between the two values, we know there is a VPN.
- We must find a way to measure the RTT-V.

Worst-Case Assumptions

- The VPN may not provide all the network services we usually count on.
 - The malicious user may compromise the VPN.
 - The malicious user may set up the VPN.
- The VPN may decline to send any response to a request, including
 - Ping, Traceroute, TCP connection, Etc.
- A non-malicious client machine is assumed to respond to most network communications.

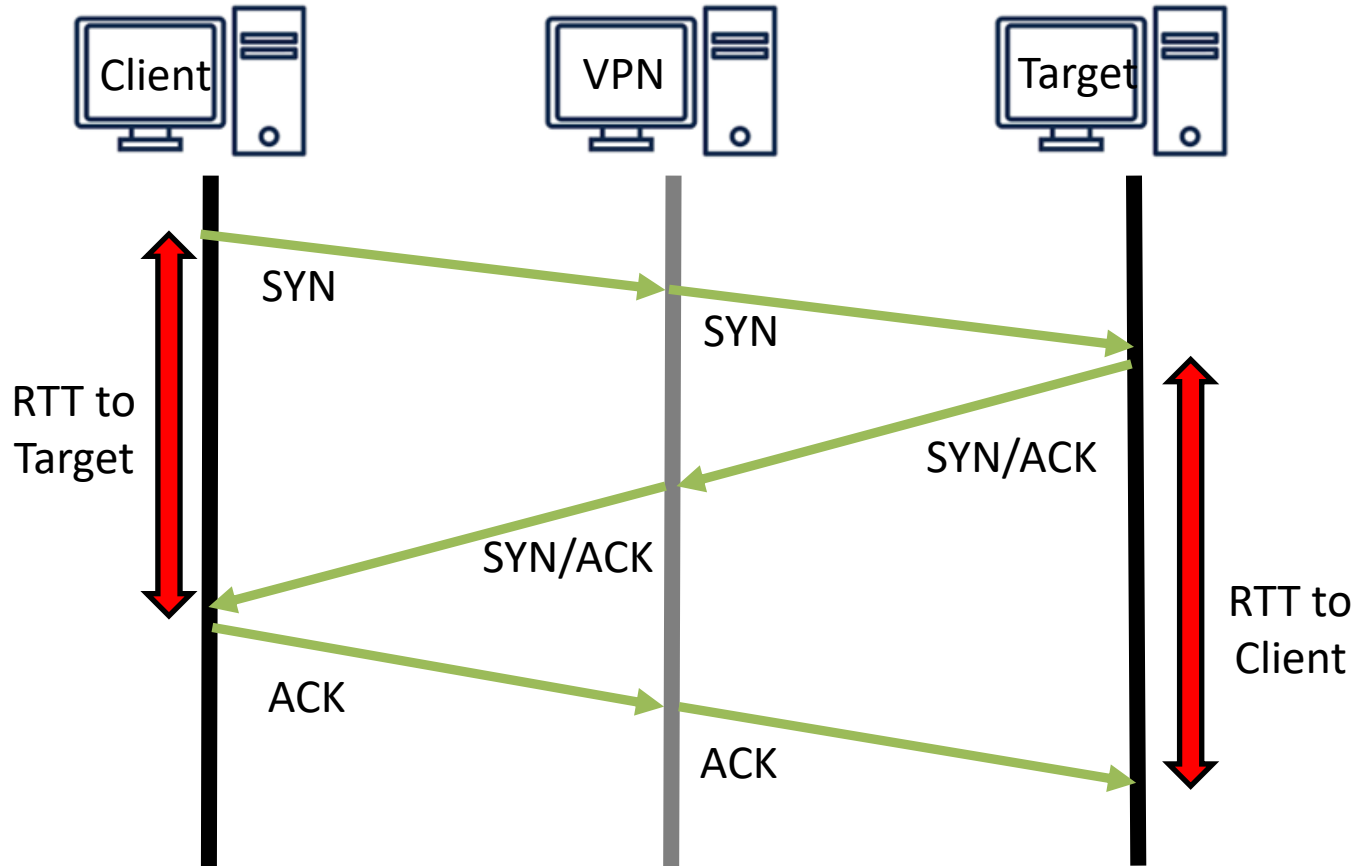
(a) RTT from the Target



RTT-C (Long)

- We have discovered three pairs of packets that will give us the RTT-C.
 - The three-way handshaking of the TCP protocol.
 - The Version Number exchange of the SSH protocol.
 - The Encryption Key exchange of the SSH protocol.
- All we need is one of the three to work. The two SSH protocol exchanges always work while the TCP protocol failed for one commercial VPN server.
- For simplicity, we will use the three-way handshaking to explain the calculation.

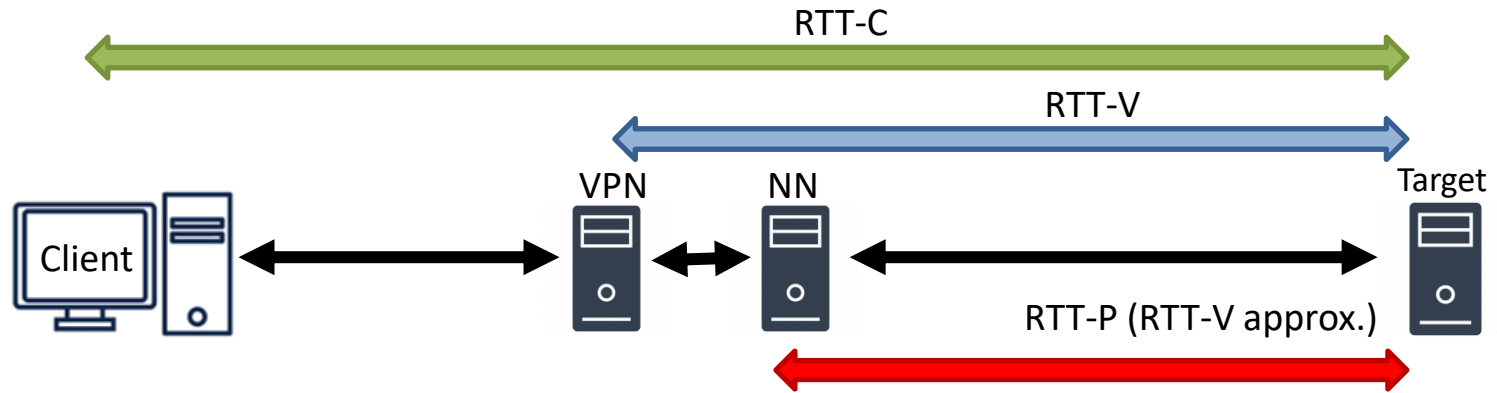
RTT-C (Long)



(b) RTT-V (Short)

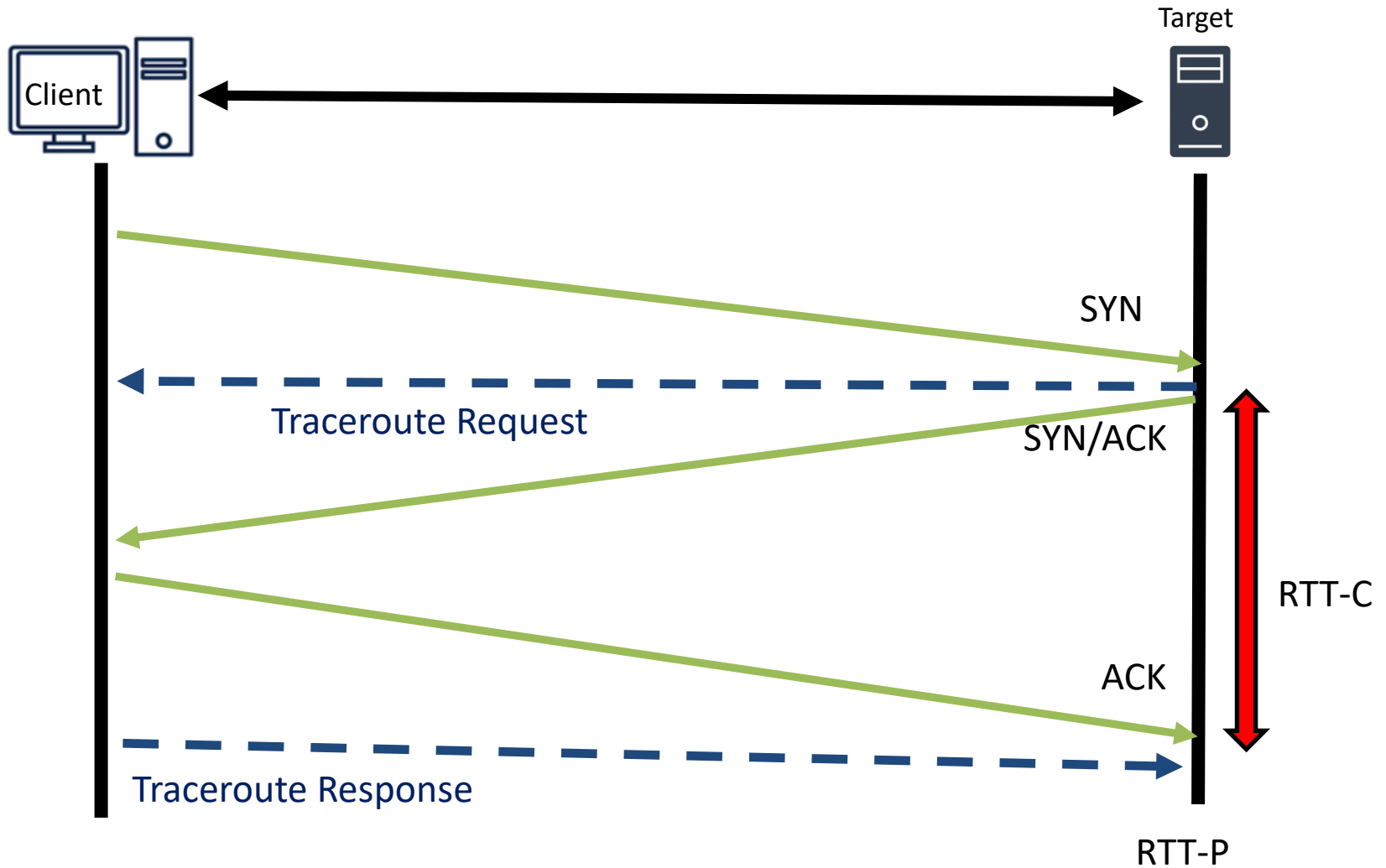
- Under the assumptions, we cannot trigger a compromised VPN to respond to a request.
- However, if we can find a good approximation, we may be able to distinguish VPN vs. non-VPN.
- Traceroute was used to do the probing. It may not reach the VPN but will try to get to the “nearest” neighbor.
- Is the approximation good enough?

Probing: Nearest Neighbor

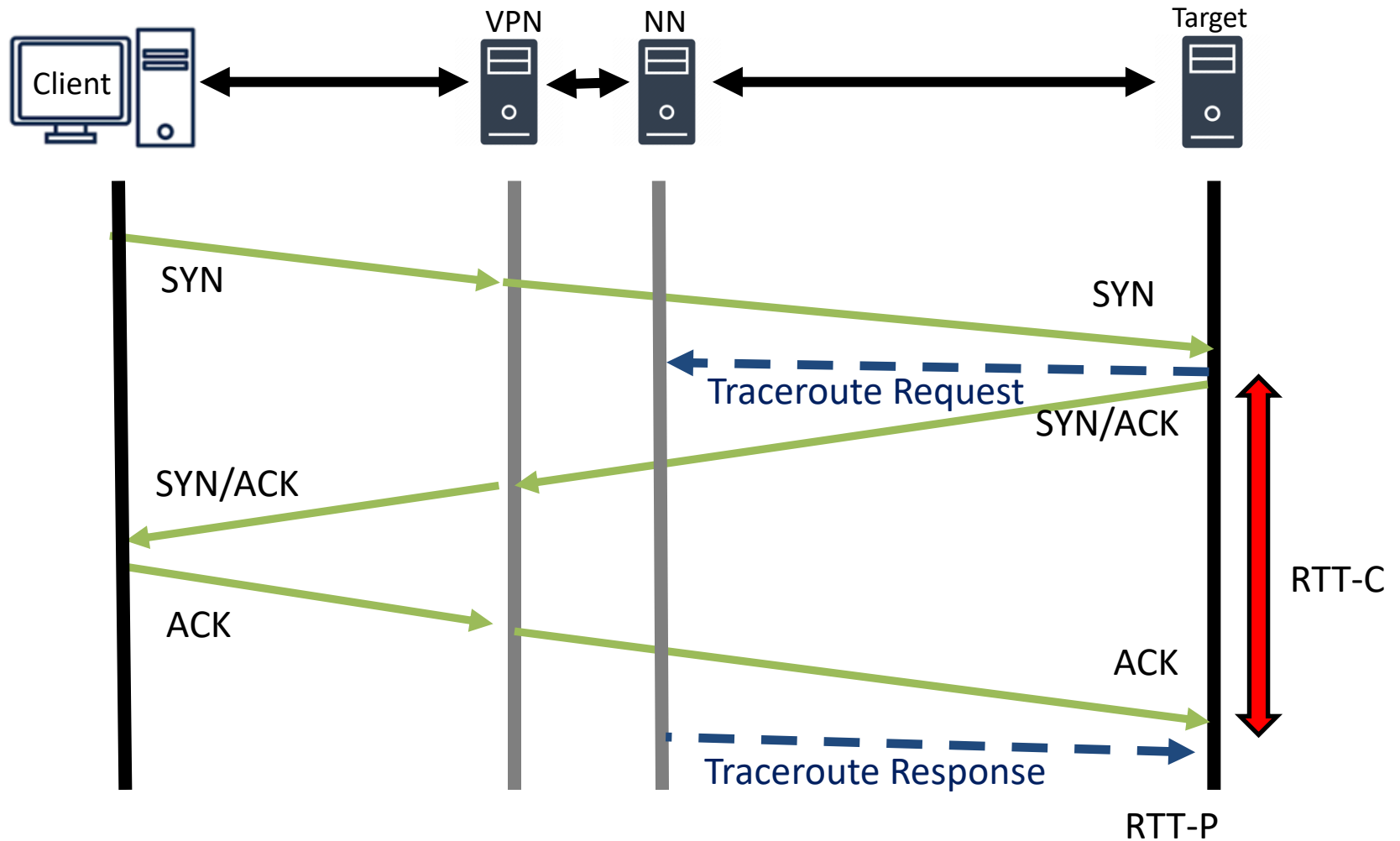


- In most of the experiments, the **RTT-P** & **RTT-V** differ by 3% in value.
- Because of the complexity of the probing, some **RTT-P** > **RTT-V**.

(c) RTT + Probing



RTT + Probing



The Algorithm

```
# Compute the three RTT-C;
if RTT-C is consistent using the protocol packets
    Choose one to be RTT-C
else
    return(VPN) # efficiency
# Probing to compute RTT-V
if contact to VPN available (e.g. Ping)
    compute the RTT-P to approximate RTT-V #efficiency
else # most costly
    send out a TraceRoute probe
    compute RTT-P to approximate RTT-V
# Decision
if RTT-C and RTT-V are close enough
    return(Non-VPN)
else
    return(VPN)
```

3. Validation and Result







- Validation is challenging.
- We notice that the ratio of distance is important. Variations of small values may have a bigger influence on the ratio.



- We decided to test four extreme cases for VPN and two cases for non-VPN.

Validation

- We tested four types of connections.

VPN			
	Config Type	# Config	# Data Point
	Long-Long	54	270
	Long-Short	54	270
	Short-Long	108	540
	Short-Short	54	270
	Total	270	1350
Direct Connection			
	Config Type	# Config	# Data Point
	Long	108	540
	Short	108	540
	Total	216	1080

Discrepancy Ratio R_D

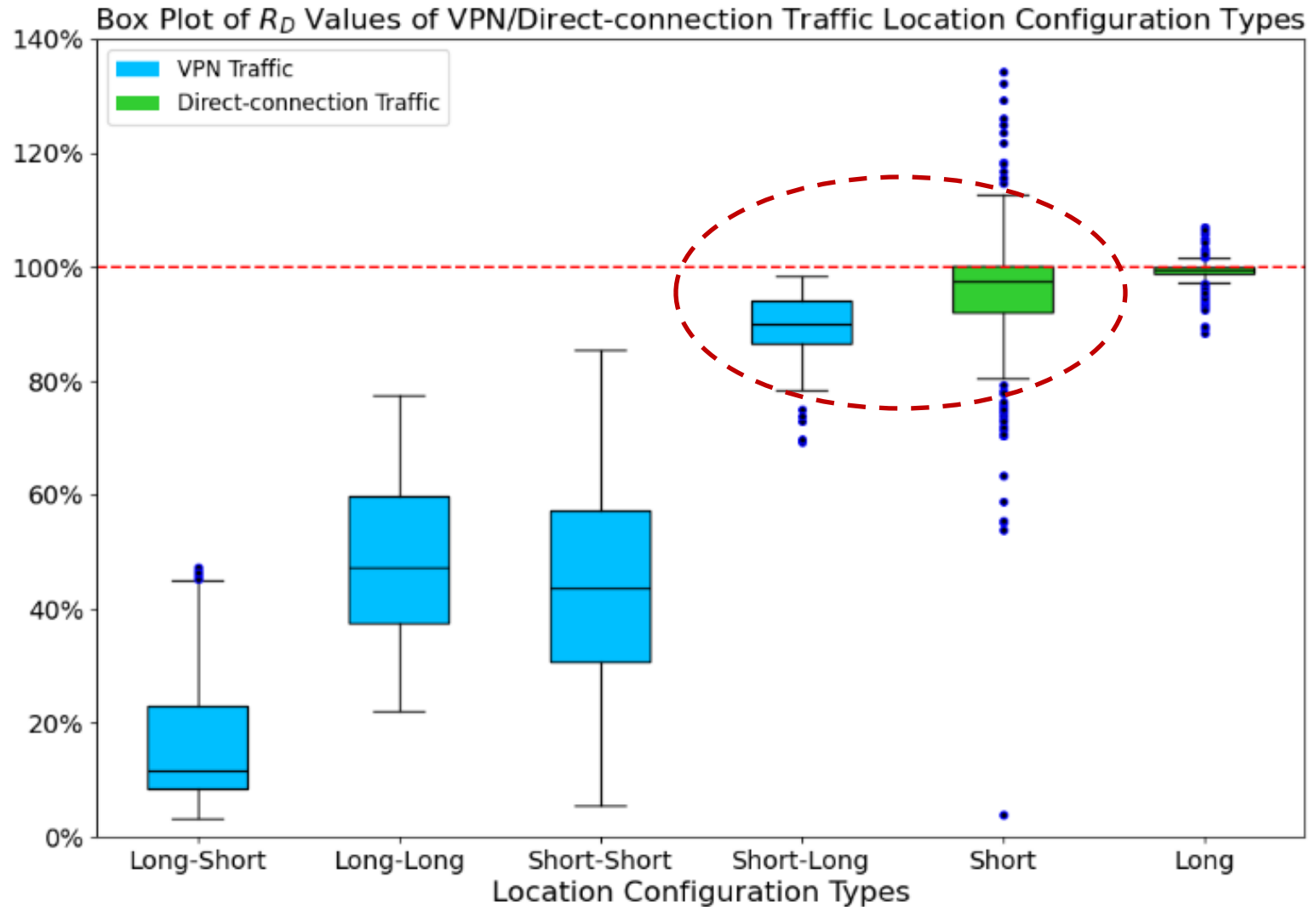


Table 2

		Train with VPN Location Config Types			
		Long-Long	Long-Short	Short-Short	Short-Long
Detect with Direct Connection Location Config Types	Long	1.0000	1.0000	1.0000	0.9630
	Short	0.9722	0.9981	0.9463	0.7805

- Cross-training data: 4 x 2 cases
- Average 95.75%, Worst 78.05%
- Modification of the ML Algorithm: Adding two attributes
 - Discrepancy Ratio
 - RTT-V
 - RTT-C

Table 2


		VPN Location Config Types			
		Long-Long	Long-Short	Short-Short	Short-Long
Direct Connection Location Config Types	Long	1.0000	1.0000	1.0000	0.9630
	Short	0.9722	0.9981	0.9463	0.7805

- Average 95.75%, Worst 78.05%

		VPN Location Config Types			
		Long-Long	Long-Short	Short-Short	Short-Long
Direct Connection Location Config Types	Long	1.0000	1.0000	1.0000	0.9657
	Short	1.0000	0.9988	0.9876	1.0000

- Average 99.40%, Worst 96.57%

Table 3: Accuracy vs. ML

ML Model		Accuracy	F1-score	Precision	Recall
S V M	RBF	0.9774	0.9797	0.9760	0.9837
	Linear	0.9691	0.9724	0.9688	0.9763
	Poly	0.9687	0.9718	0.9748	0.9689
	Sigmoid	0.7588	0.7749	0.8029	0.7504
 Random Forest	0.9868	0.9882	0.9854	0.9911	
Naïve Bayes	0.9407	0.9214	0.9320	0.9116	
Logistic Regression	0.9412	0.9224	0.9315	0.9139	
Neural Net	0.9827	0.9845	0.8448	0.6635	

4. Conclusion

- Algorithms
 - Finding Discrepancy in RTTs.
 - Computing RTT-C on the target side.
 - Using probing to estimate RTT-V.
- Advantage:
 - Real-time detection,
 - High accuracy rate,
 - Efficient (before any user data is transmitted).
- Our original goal was to use only the TCP protocol for detection, but we had to include the SSH protocol at the end.

Thank You

SHuang@cs.uh.edu

This research is sponsored by the National Security Agency (NSA) of the USA