CalcuLatency: Leveraging Cross-Layer Network Latency Measurements

to Detect Proxy-Enabled Abuse

Reethika Ramesh, Philipp Winter, **Sam Korman**, Roya Ensafi

USENIX Security '24

Global events affect the Internet in new ways every day. **Governments**, **network providers**, and **online threat actors** disrupt, tamper with, and monitor user traffic.

VPNs are Useful Tools

Users are turning to Virtual Private Networks (**VPNs**) as a **panacea** for security, privacy, and information restrictions.

VPNs are now a **multi-billion dollar** industry


techradar pro
A single VPN drop-out exposed breach scandal that cost Ubiquiti $4bn

How VPNs and Proxy Servers Are Used For Click Fraud

Leave no trace: how a teenage hacker lost himself online

VPN Misuse is on the rise

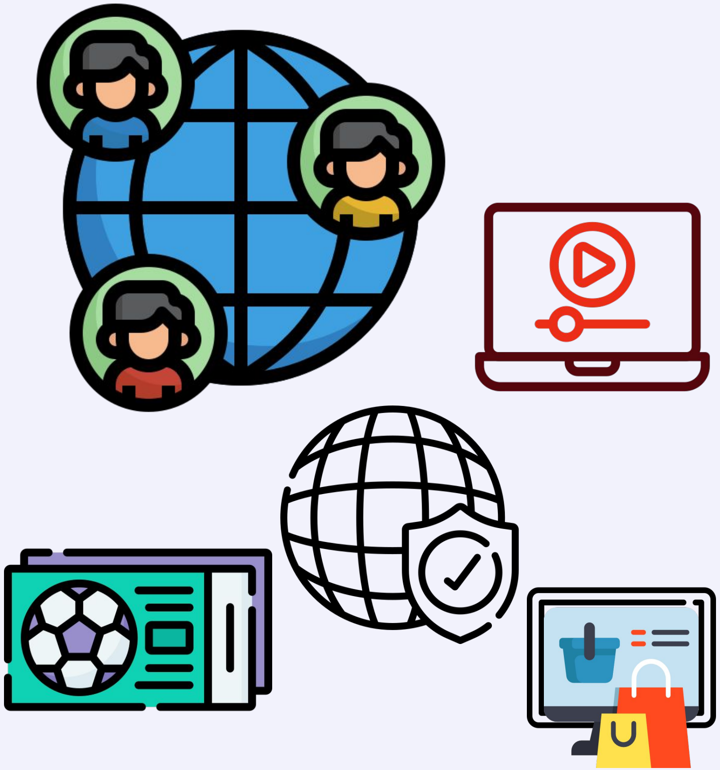
- VPNs were intended as a privacy-enhancing tool
- Bad actors misuse VPNs and hide behind them while carrying out nefarious activities

Challenges for

Server-Side Operators

Service providers impose certain restrictions on users:

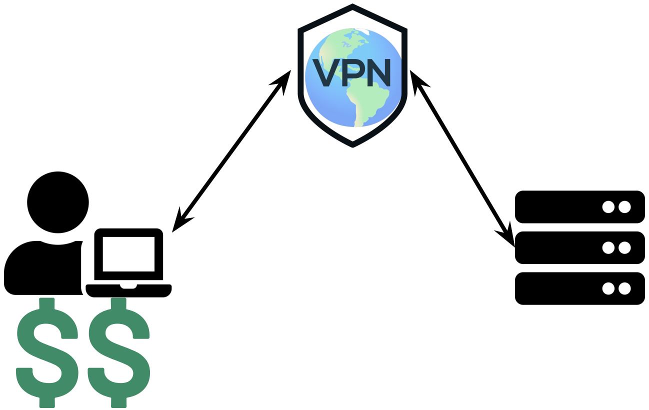
- Media licensing restrictions
- Geographic-proximity limitations
- E-commerce needs
- Security needs



Threats Due to Proxy and VPN Misuse

Attackers **fabricate their geolocation** to access geo-restricted content, or falsify activity to profit monetarily

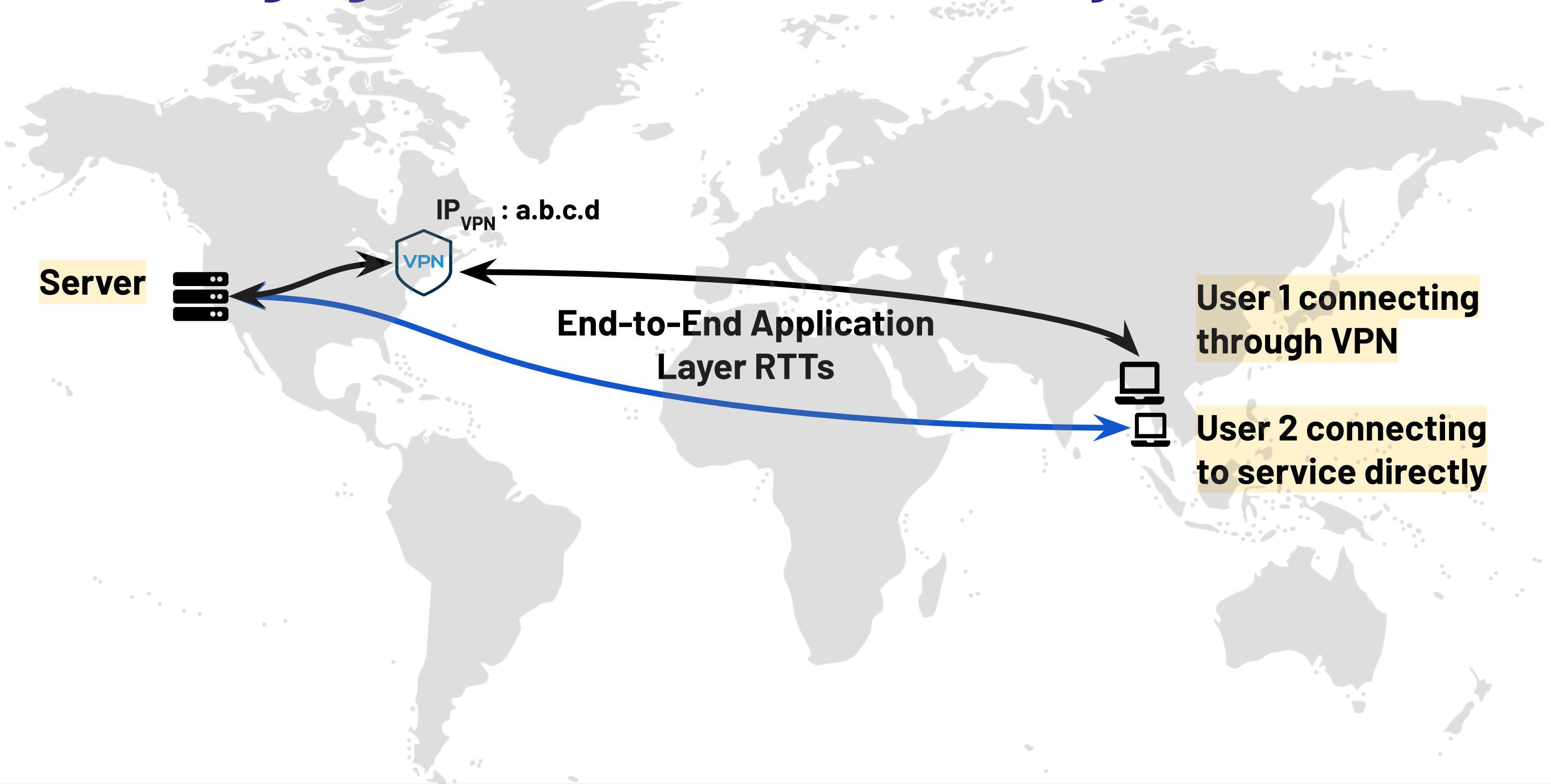
Balancing abuse-prevention techniques with a privacy-first approach is a hard challenge



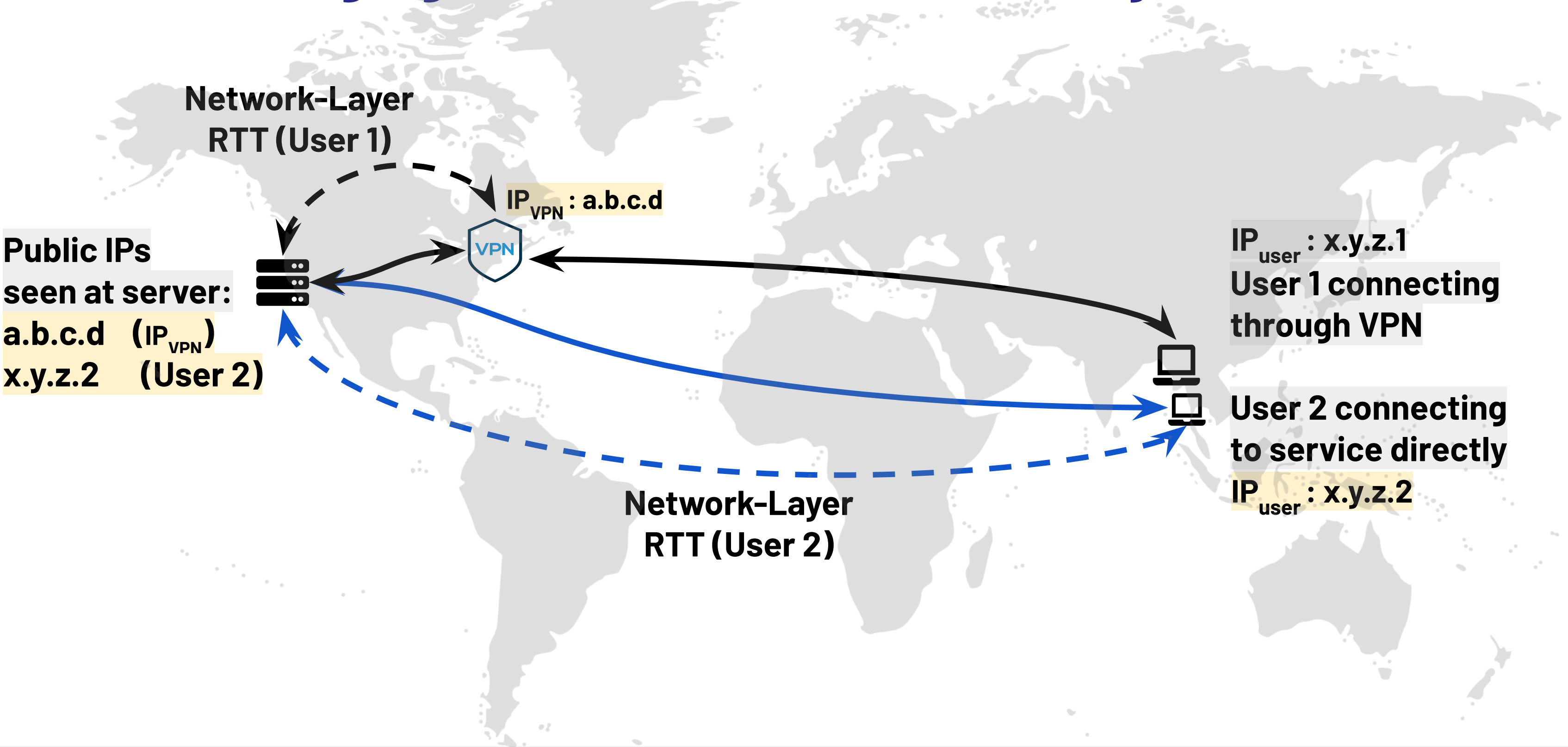
Can we use minimal connection features, such as latency, to infer proxy use, without jeopardizing user privacy?

Leveraging Different Network Latency Measurements

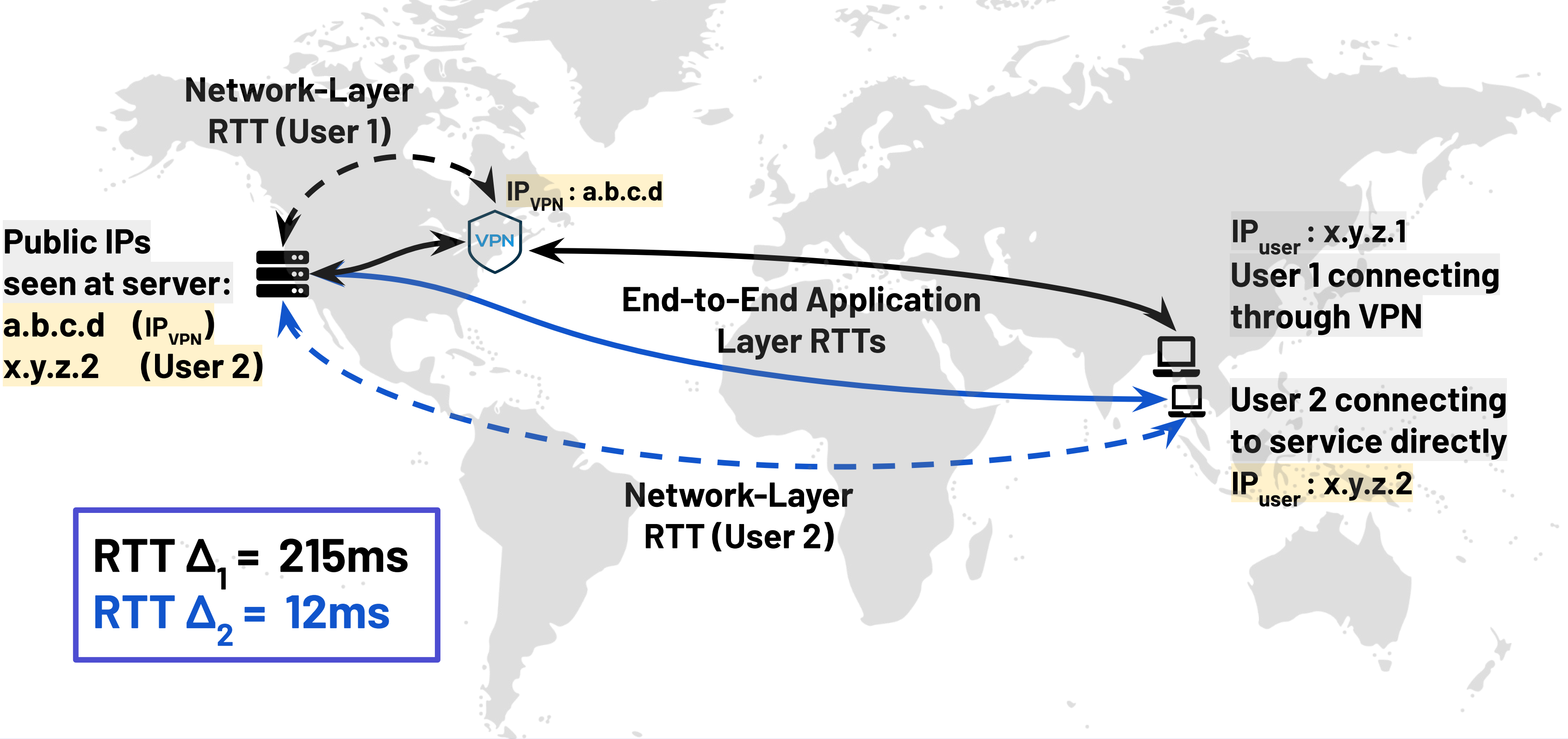
Leveraging Different Network Latency Measurements



Leveraging Different Network Latency Measurements



Leveraging Different Network Latency Measurements



RTT Δ_1 = 215ms

RTT Δ_2 = 12ms

System Assumptions

- **Require an application-layer connection** between the client and the server—e.g. HTTP(S), WebSocket
- We detect **long-distance, remote proxy** use, i.e. the proxy is geographically far from the user
- **Clients do not control the network behavior** of the proxy and therefore cannot selectively delay packets

Measurement Methods on Different Layers

Application Layer Latency

- WebSocket
- WebRTC
- HTTP page load times

Measurement Methods on Different Layers

Application Layer Latency

- WebSocket
- WebRTC
- HTTP page load times

Network Layer Latency

- TCP Handshake RTT
- ICMP Ping
- **Modified Traceroute**

Modified **Traceroute**—0trace

Traceroute: Send packets with incrementing TTL to determine **the path** and **the time taken** for a packet to reach a destination

Challenge: Get remote IP address' network stack to respond to unsolicited IP packets reliably

Modified Traceroute—Otrace

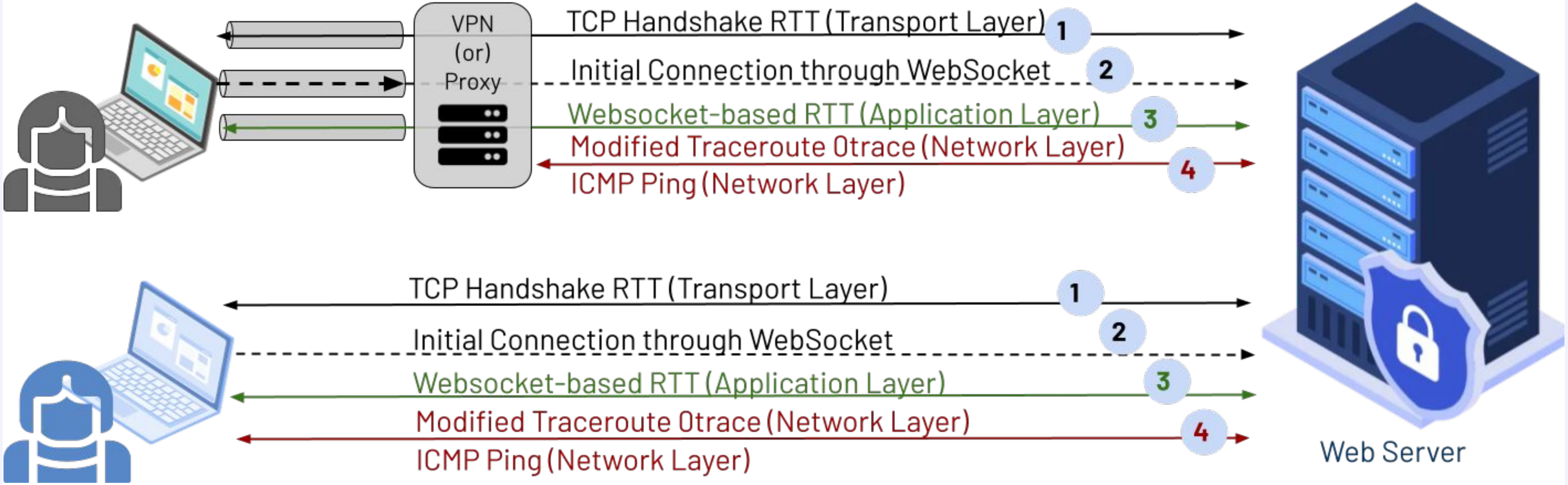
Traceroute: Send packets with incrementing TTL to determine **the path** and **the time taken** for a packet to reach a destination

Challenge: Get remote IP address' network stack to respond to unsolicited IP packets reliably

Otrace leverages existing TCP connections initiated by a client:

- Sends trace packets that **match the five-tuple** of an already established connection
- Can pass **stateful firewalls** and **traverse NATs**

CalcuLatency



Evaluation and Results

Two Sets of Evaluations

Building Block

Reliability of:

- WebSocket Pings
- TCP Handshake RTT
- Otrace Pings: Variance of Latency Across the Internet

CalcuLatency System

Evaluating the system in practice:

- Control Testbed Evaluation
- Real-world Crowdsourced Evaluation

Controlled Testbed

- Automated testing with Selenium from devices in 12 networks
- Tested on four popular browsers
- Four countries and geo-distributed servers from ten VPN providers

891 experiments: 337 VPN IPs in 82 ASes, 17 direct connections from 12 ASes in 4 countries

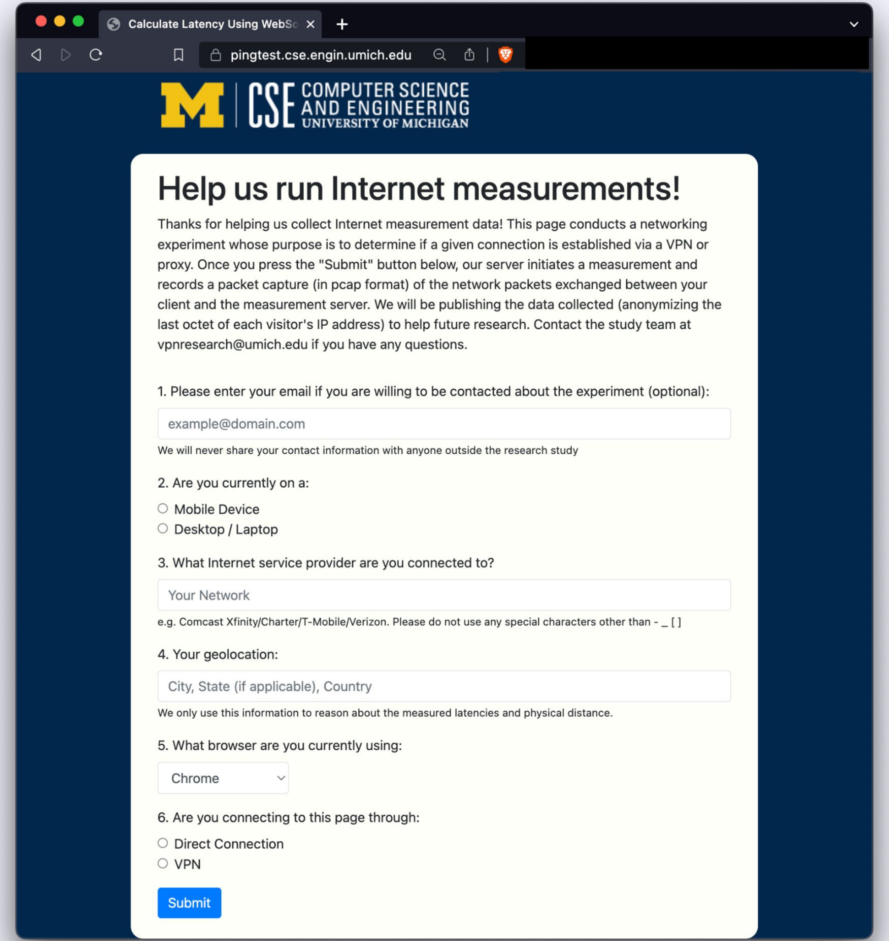
User Testing Locations:
USA, Canada, India, UAE

10 different **VPN Providers** offering WireGuard, OpenVPN, proprietary protocols, and own SOCKS5 implementation

Public Crowdsourced

- Deployed CalcuLatency system, hosted on a university subdomain
- Recruited users on Twitter and collected data for 15 days
- 37 countries from all (six) continents

283 experiments: 122 VPN IPs in 51 ASes, 161 direct connections from 93 ASes in 37 countries



Empirically viable RTT difference threshold is **50 milliseconds**

Evaluation Results

98% direct measurements:

ΔRTT < 50ms

89.1% and **63.9%** VPN measurements:

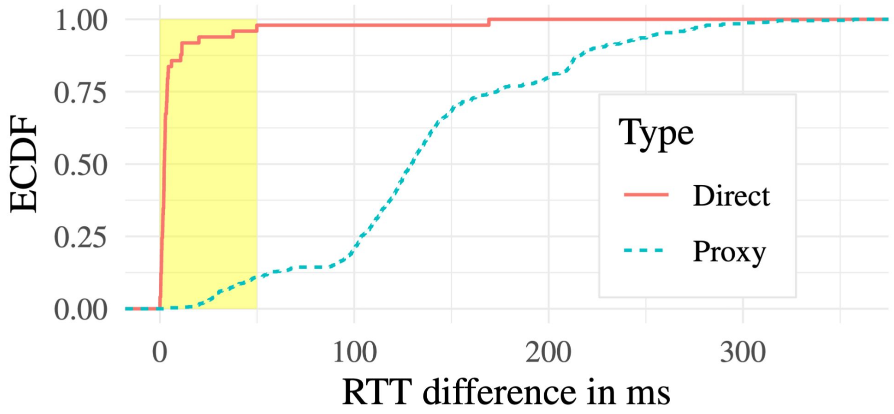
ΔRTT > 50ms

50ms covers almost all cases of direct measurements, i.e. low false positive rate

Controlled Testbed Evaluation

98% of direct measurements:
Δ RTT < 50 ms

89.1% of VPN measurements:
Δ RTT >> 50 ms

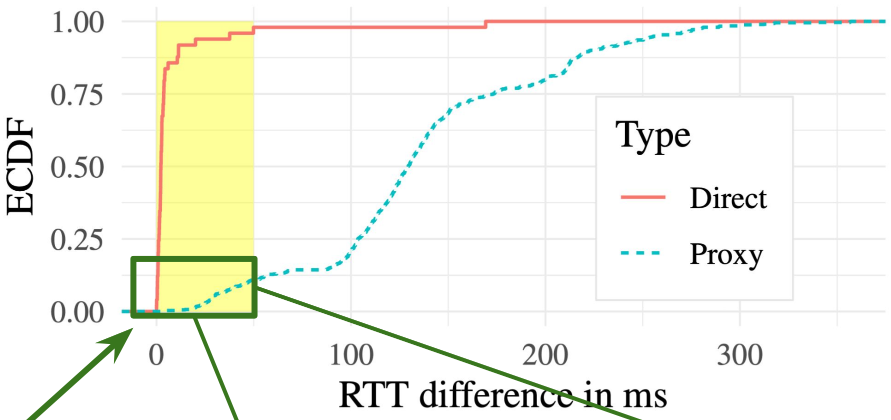


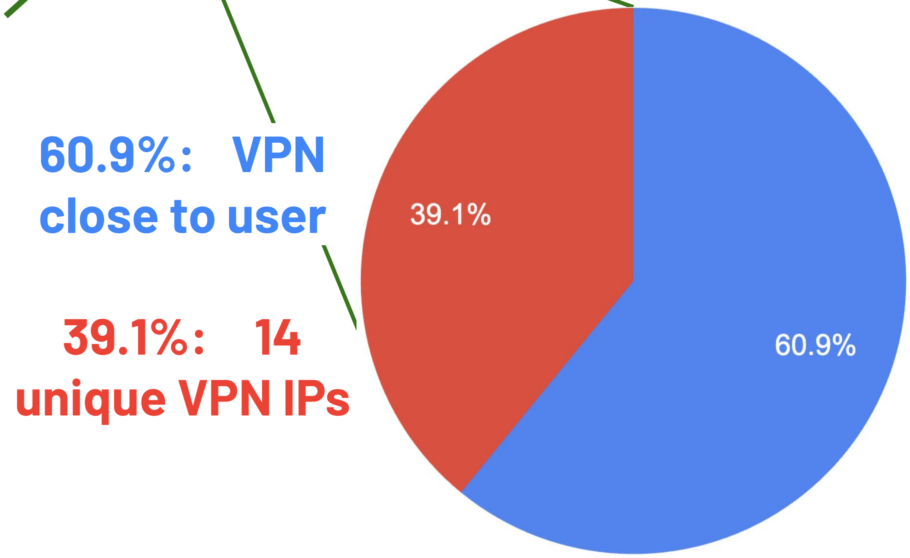
891 experiments: 337 VPN IPs in 82 ASes, 17 direct connections from 12 ASes in 4 countries

Controlled Testbed Evaluation

Of the remaining 10.9%, 60.9% of the time, the VPN was located very close to user

And 39.1% mapped to 14 unique VPN IPs





Controlled Testbed Evaluation

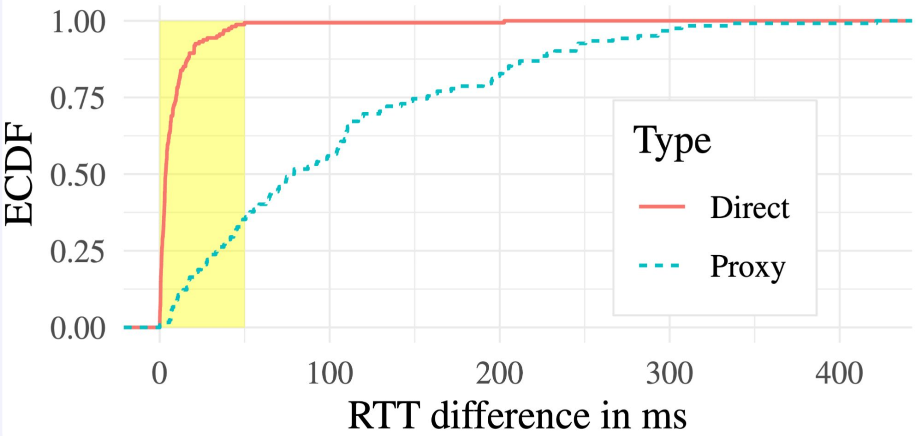
Investigating the **network layer RTTs** for these IPs: 6 of their **advertised VPN locations are an impossibility** based on speed of Internet approximations $\frac{4}{\square} * c$ [28]

[28] Katz-Bassett et al., ACM IMC 2006

Public Crowdsourced Evaluation

98.8% of direct measurements:
Δ RTT < 50 ms

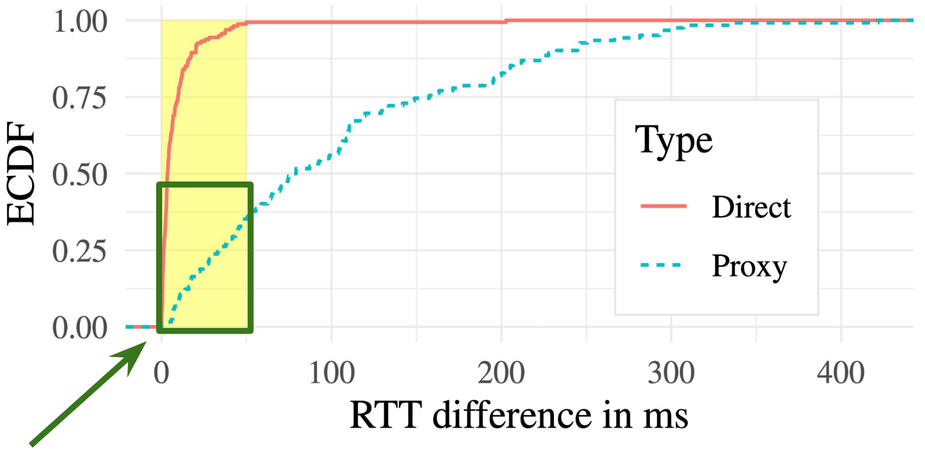
63.9% of VPN measurements:
Δ RTT >> 50 ms



283 experiments: 122 VPN IPs in 51 ASes, 161 direct connections from 93 ASes in 37 countries

Public Crowdsourced Evaluation

Of the remaining 44 experiments, in **34 of them, the VPN was located close to user** (outside our scope) and another 1 was actually a direct connection, and the remaining 9 are false-negatives



Network latency differences can be leveraged **as a first-step to identify clients connecting through remote, long-distance proxies**

Not all VPN traffic is abusive

- Not all VPN users are attackers—CalcuLatency is a labelling technique and is **not a catch-all solution**
- Users can evade detection by using VPNs close to their location which provides better performance and the requisite privacy and security features of a VPN

CalcuLatency: Leveraging Cross-Layer Network Latency Measurements

to Detect Proxy-Enabled Abuse

Reethika Ramesh, Philipp Winter, **Sam Korman**, Roya Ensafi

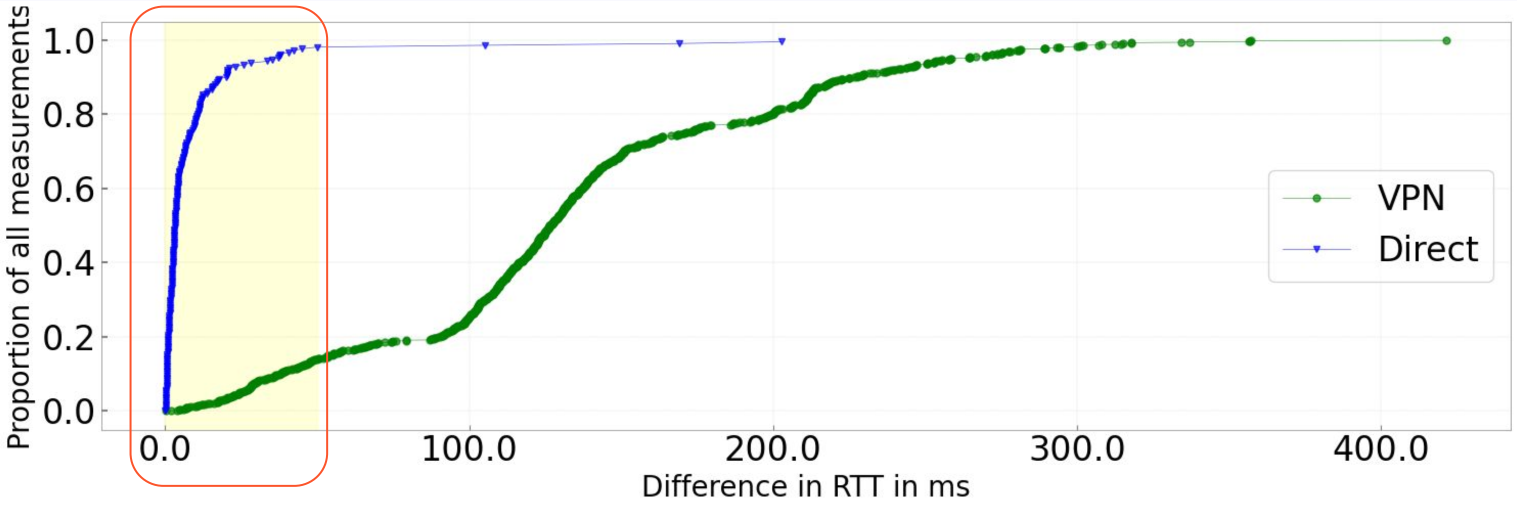
USENIX Security ’24

Evaluating Reliability of Each Measurement

WebSocket RTT	TCP Handshake RTT	Otrace Ping RTT

Results

Of 210 direct measurements, **only 3** had a RTT difference above 50ms
86% VPN measurements had an RTT difference above 50ms
Other 14%, majority were VPN server located close to the user (**not remote proxy**)



Building Block Evaluations

WebSocket Pings

We tested 10,000 sequential WebSocket echo requests to the client

TCP Handshake

Otrace Pings