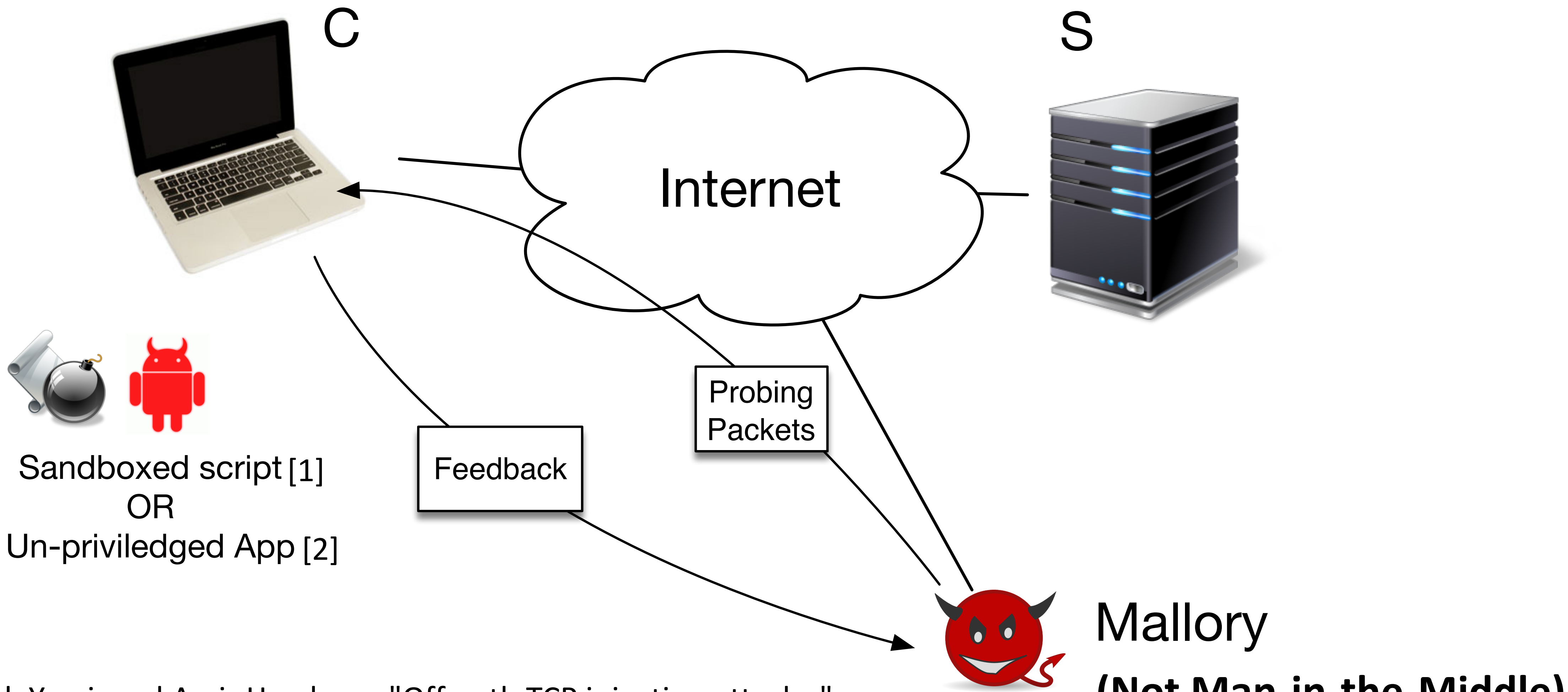
Off-Path TCP Exploit: How Wireless Routers Can Jeopardize Your Secrets

Weiteng Chen, Zhiyun Qian

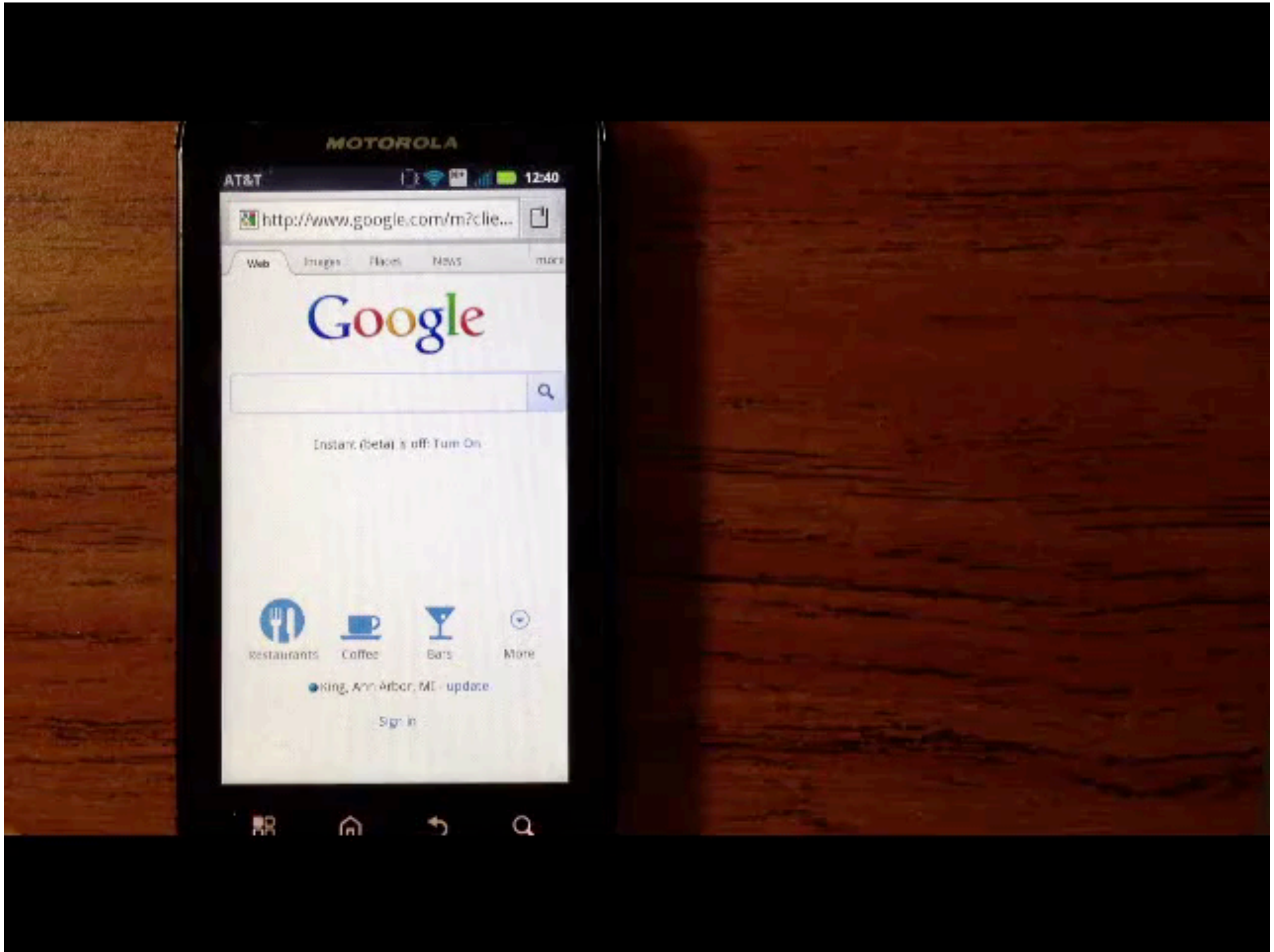
University of California, Riverside

Generic Threat Model



[1] Gilad, Yossi, and Amir Herzberg. "Off-path TCP injection attacks."
[2]Qian, Zhiyun, Z. Morley Mao, and Yinglian Xie. "Collaborative TCP sequence number inference attack: how to crack sequence number under a second."

An attack using packet counter side channel

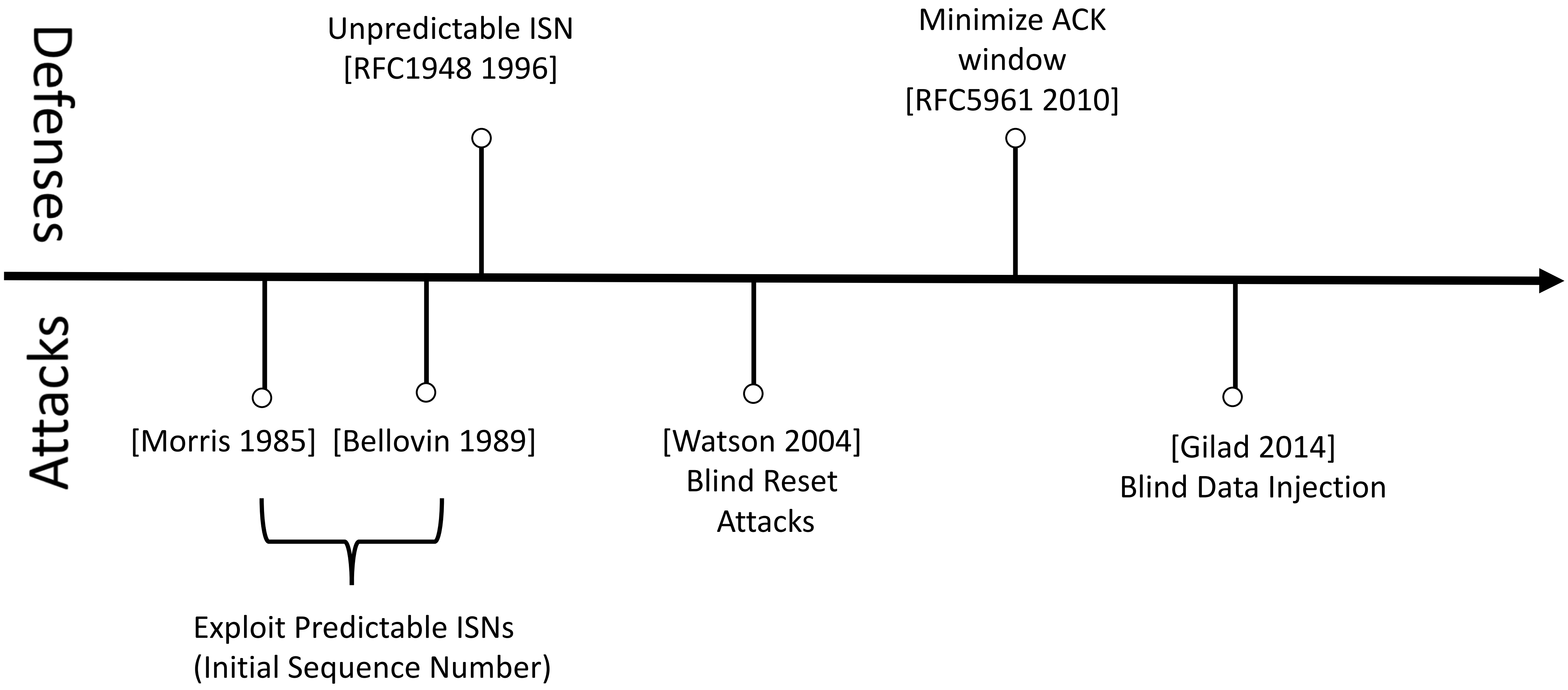


Building Blocks of Side Channels

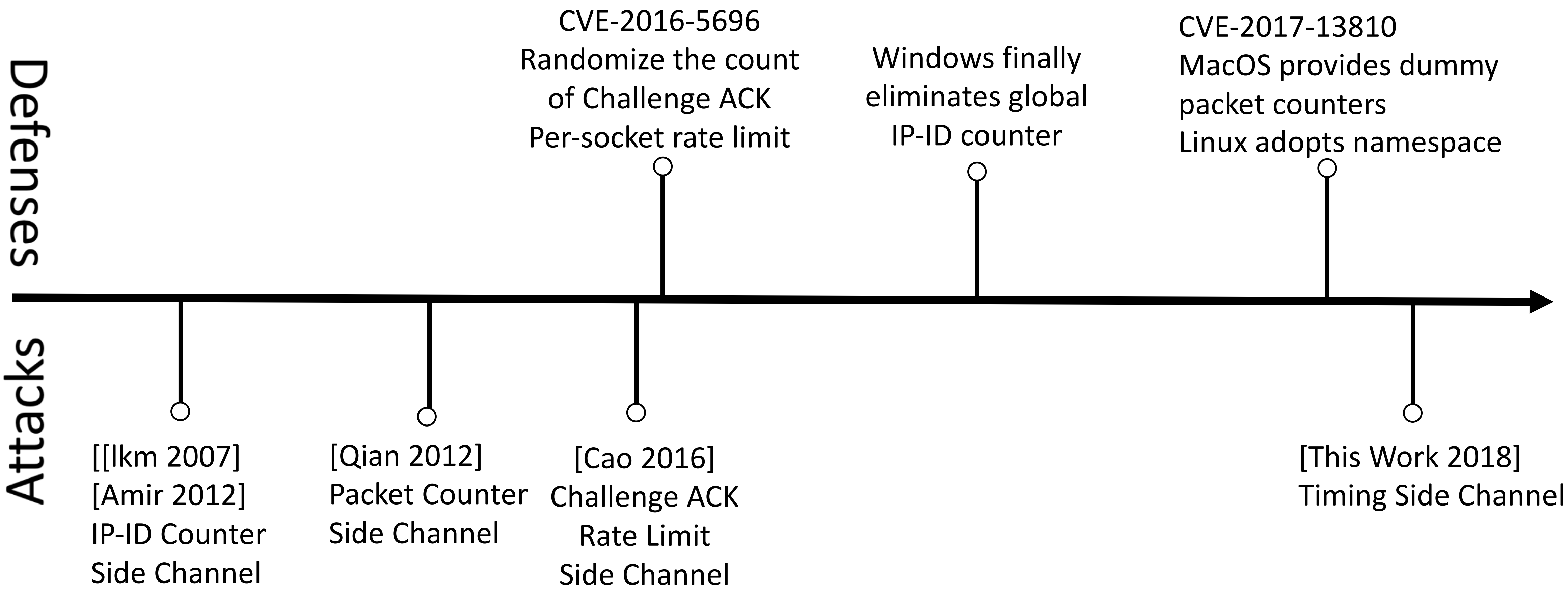
```
if (in_packet.seq is in rcv_window)
    // shared state change 1
else
    // shared state change 2
```

- Shared resources
  - e.g., Global IP-ID counter, Packet counter, Global challenge ACK rate limit
- Shared state changes observable to attackers
  - e.g., Javascript, Un-priviledged Malware

A Time-Line of TCP Injection Attacks



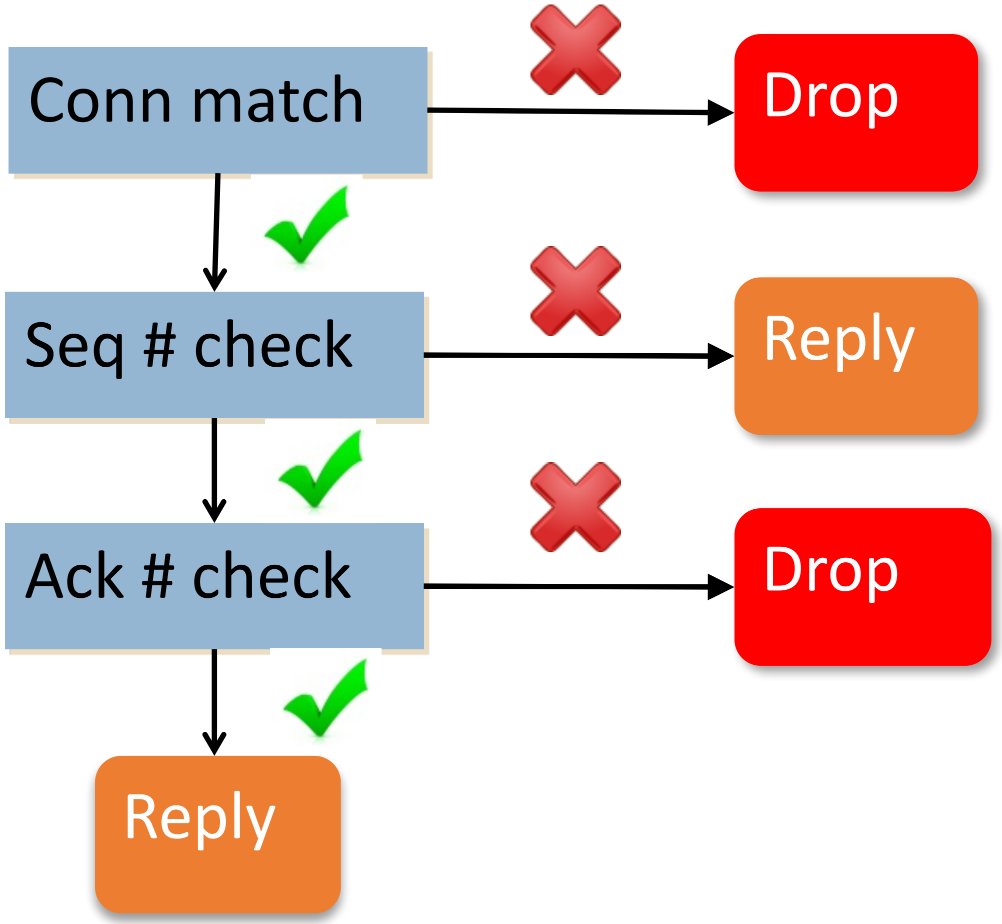
A Time-Line of TCP Injection Attacks (Cont)



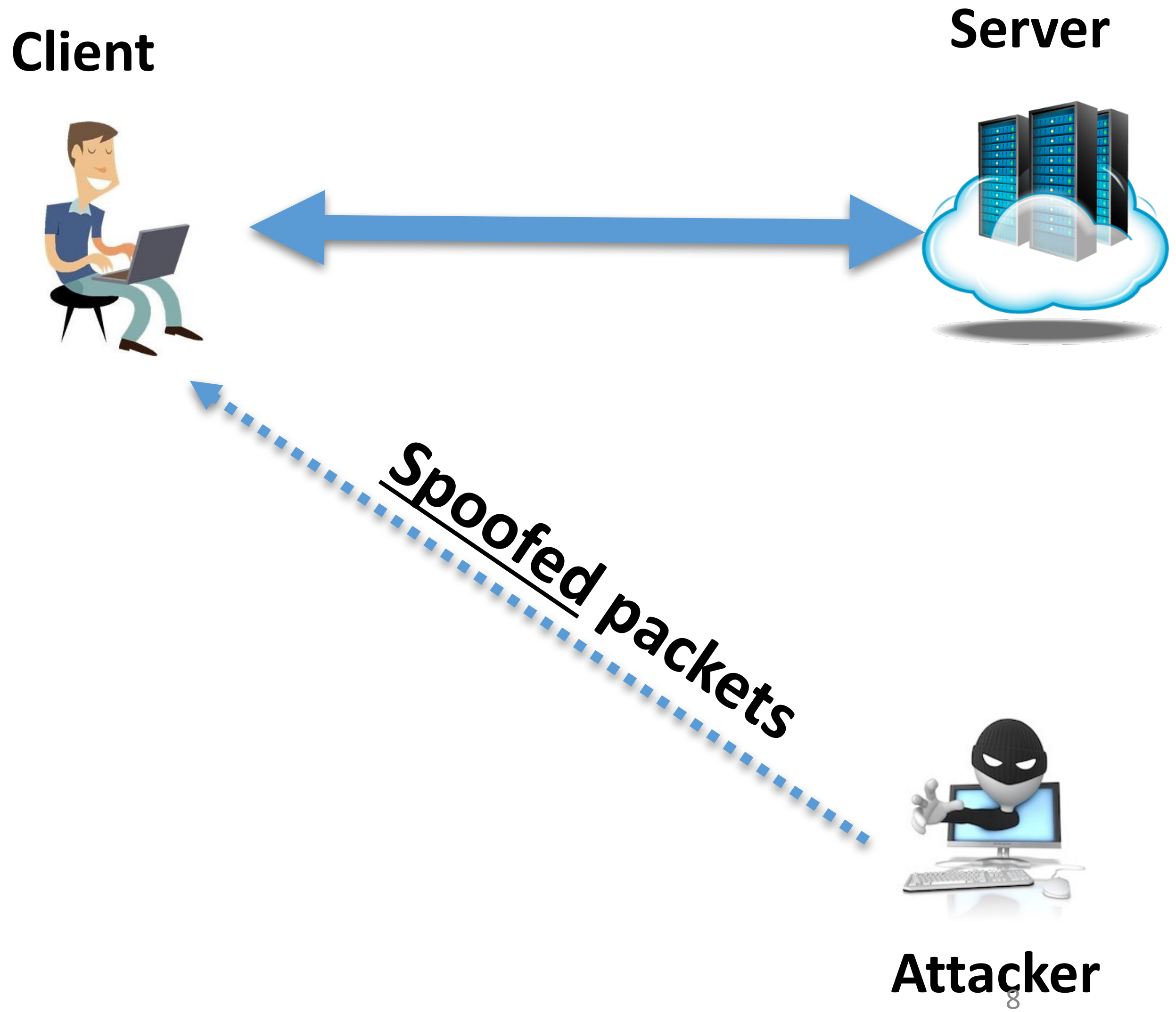
Off-Path TCP Injection Attacks

Side Channel	Requirement	Affected OS	Patch/Mitigation
Global IP-ID counter	N/A	Windows	Global IPID counter eliminated
Global challenge ACK rate limit	N/A	Linux	Global rate limit eliminated
Packet counter	Malware	Linux, MacOS	Namespace/dummy counter
Wireless contention (this work)	Javascript	Any	N/A

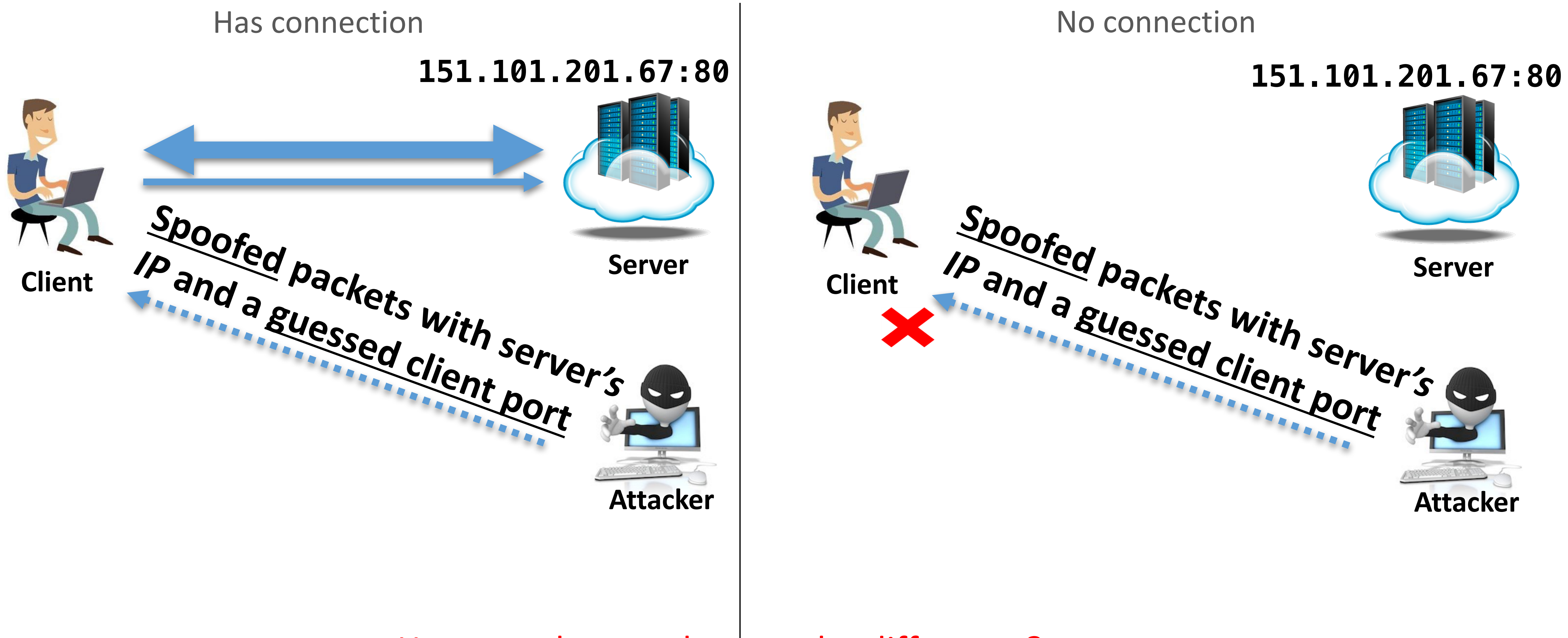
RFC 793: TCP Packet Receiving Basics



Simplified Processing Logic

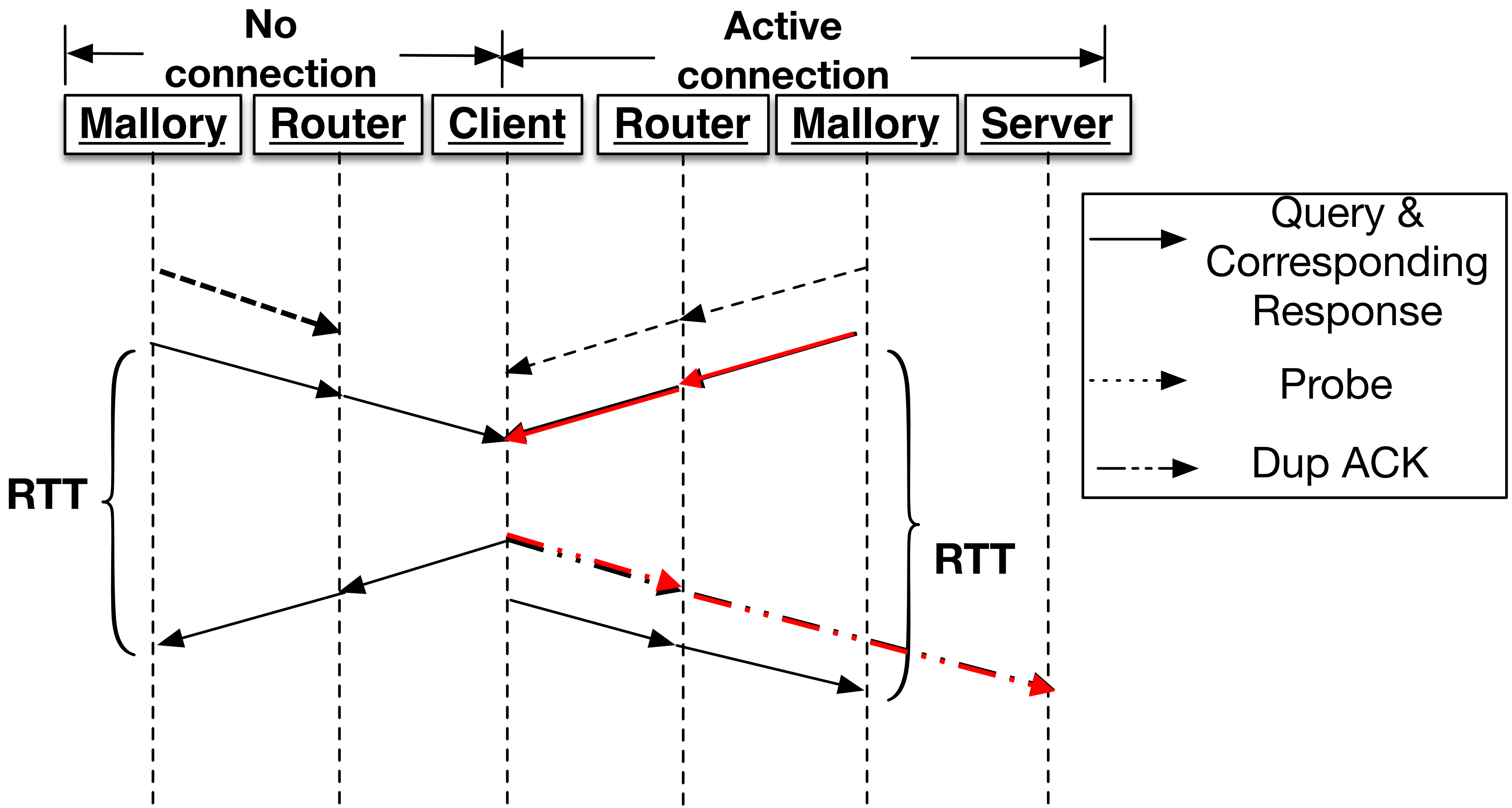


Port Number Inference



How can the attacker see the difference?

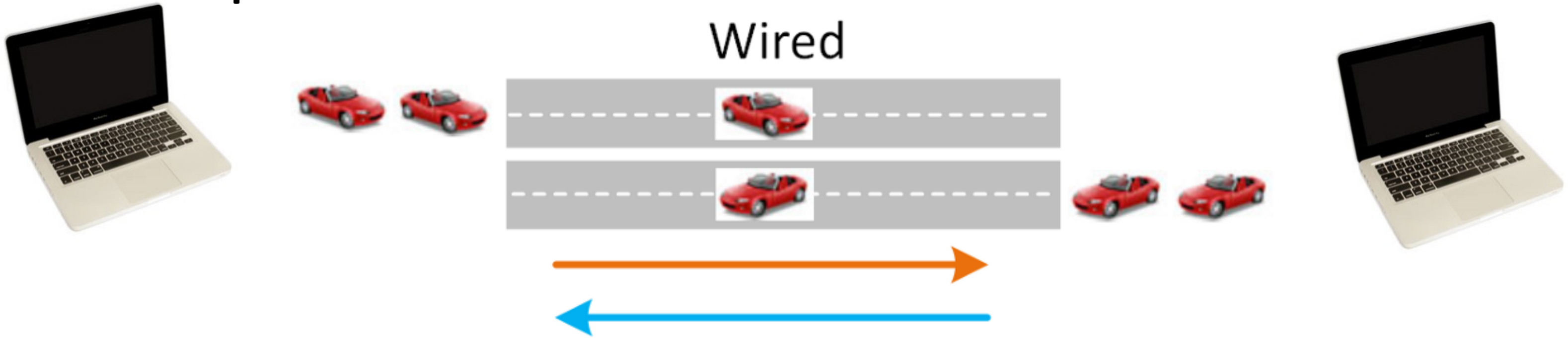
One Plausible Idea



Wireless Timing Channel

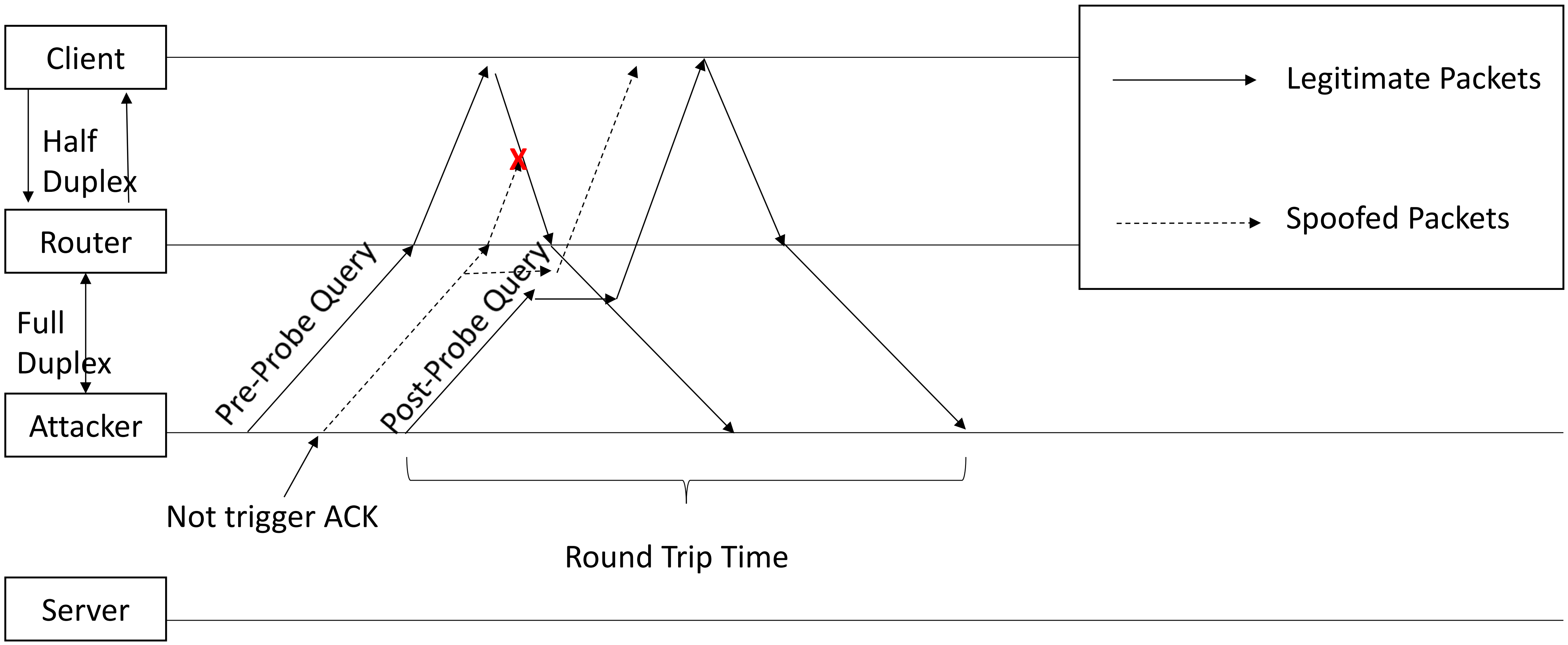
- **Half-duplex: A fundamental design of wireless protocol**
- **Shared Resource: The half-duplex wireless channel**

Full-duplex:

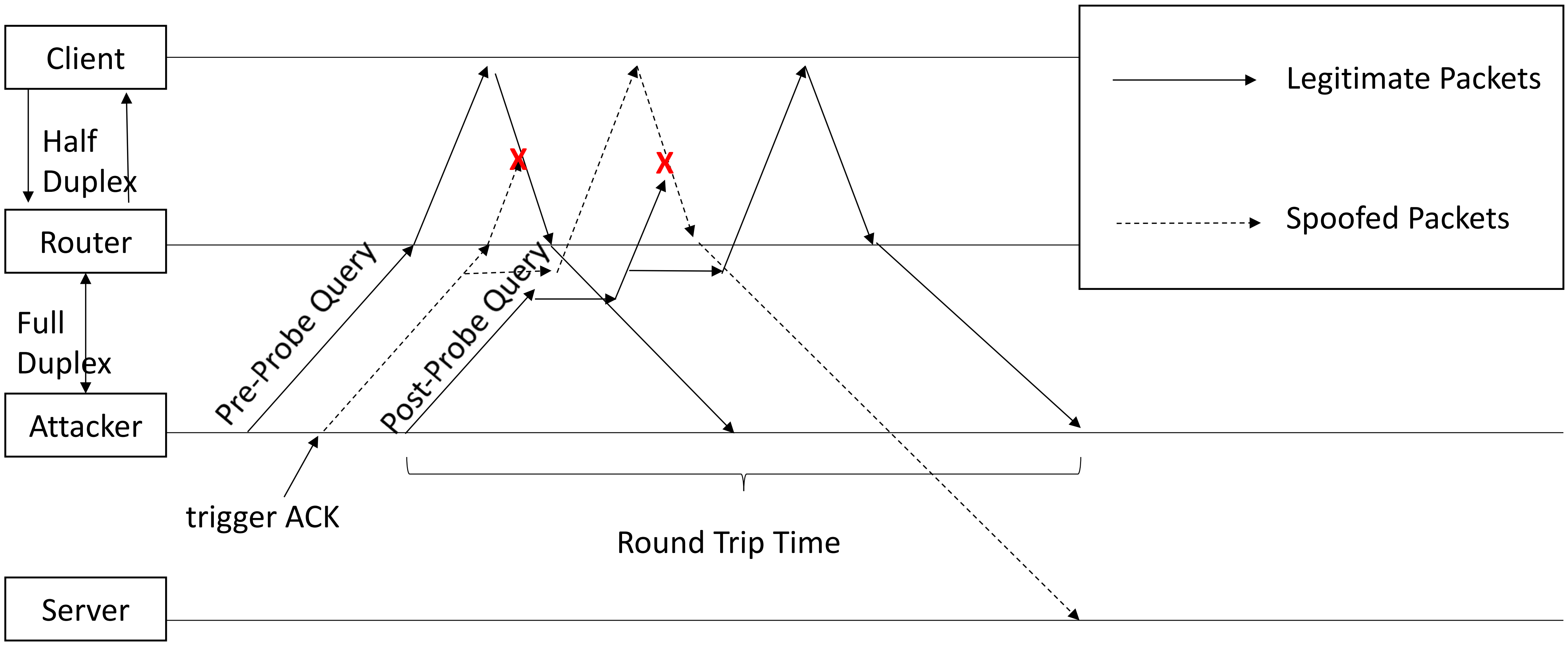


Half-duplex:

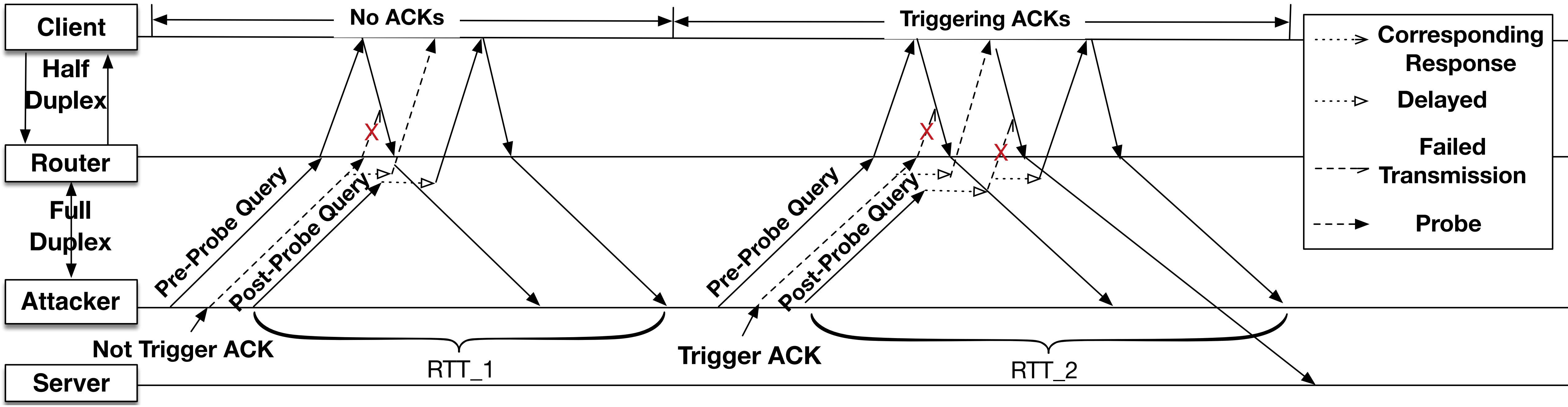
Probing Strategy



Probing Strategy (Cont)

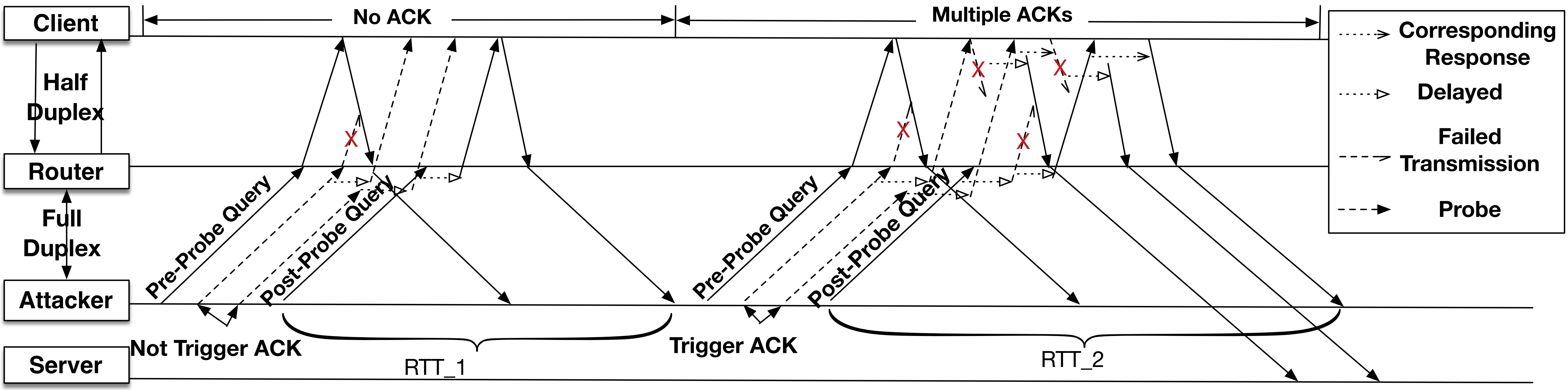


Timing Difference



- **Larger RTT ➔ Trigger ACK ➔ Correct Port Number ?**

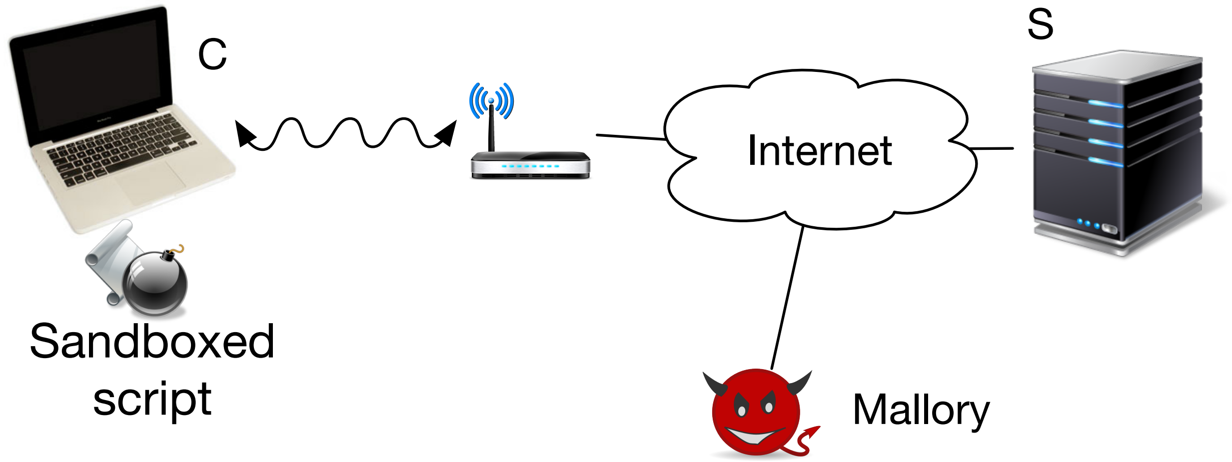
Timing Difference (Cont)



- **More Probing Packets ➔ More Contention ➔ Larger RTTS**

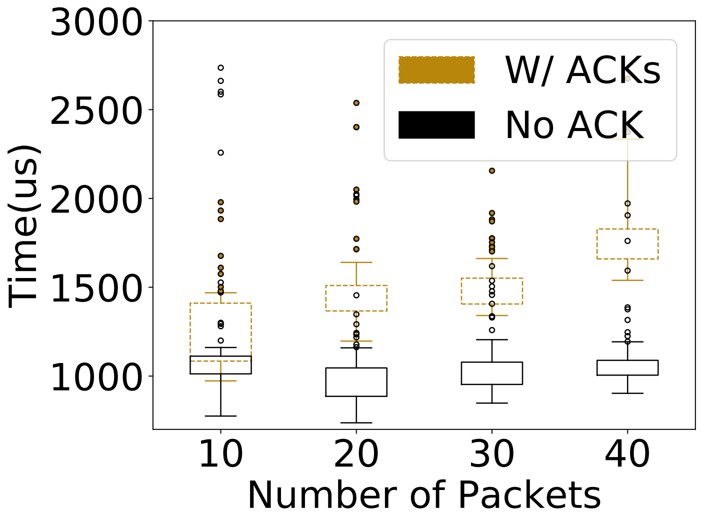
Empirical Test Results

- Setup:

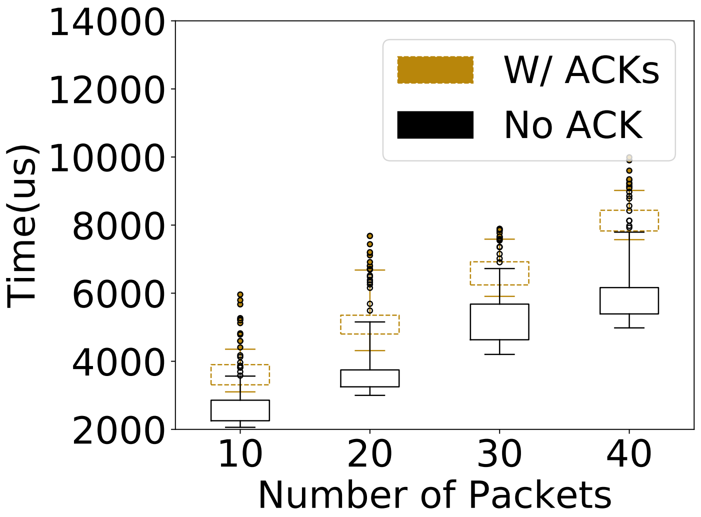


- 4 wireless routers: from Linksys, Huawei, Xiaomi, and Gee
- 2 machines: 2017 Macbook and 2017 Dell Desktop (Linux)
- 2.4GHz and 5GHz Wi-Fi

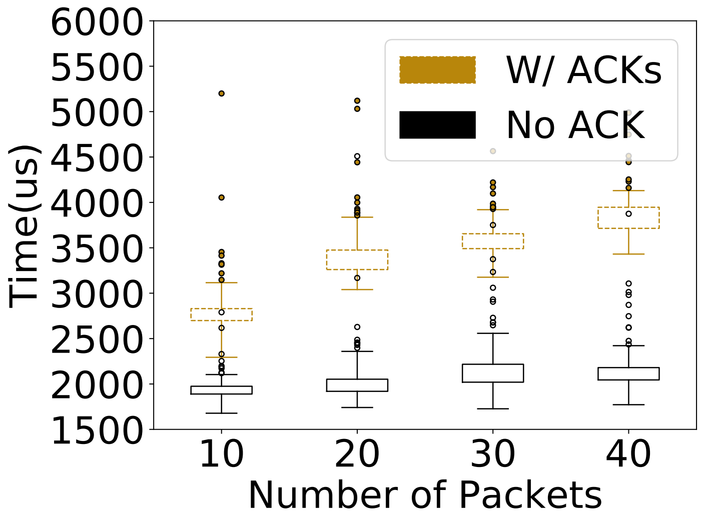
Empirical Test Results (Cont)



(a) RTT measurement of Linux using 5GHz network of a Linksys router

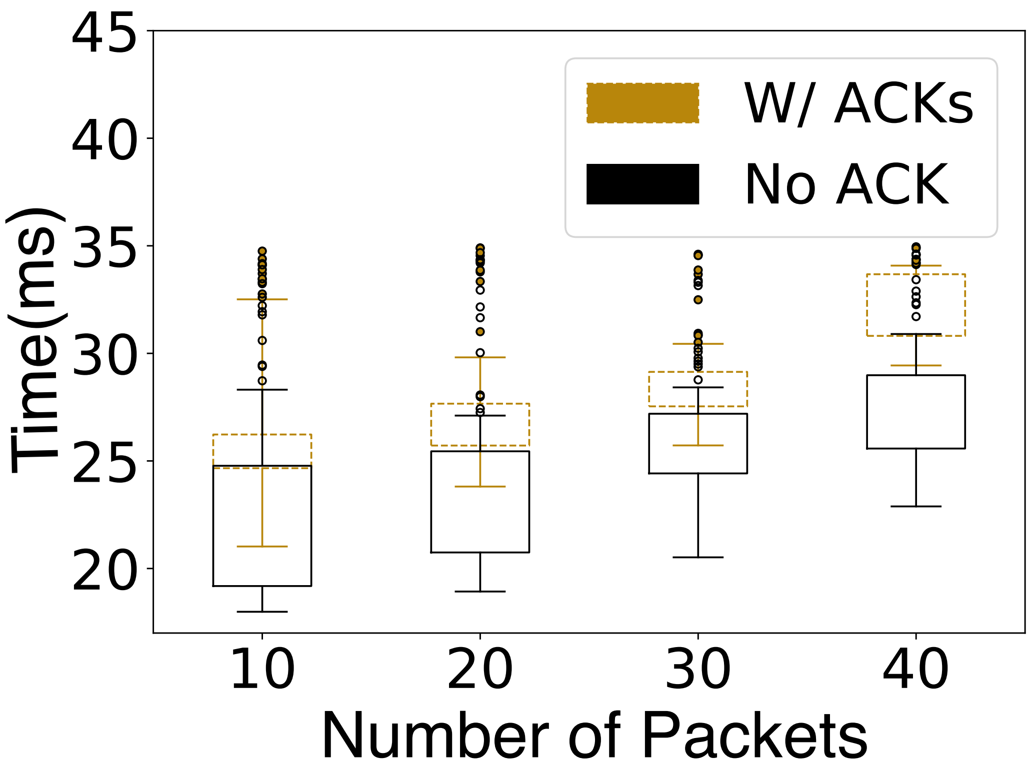


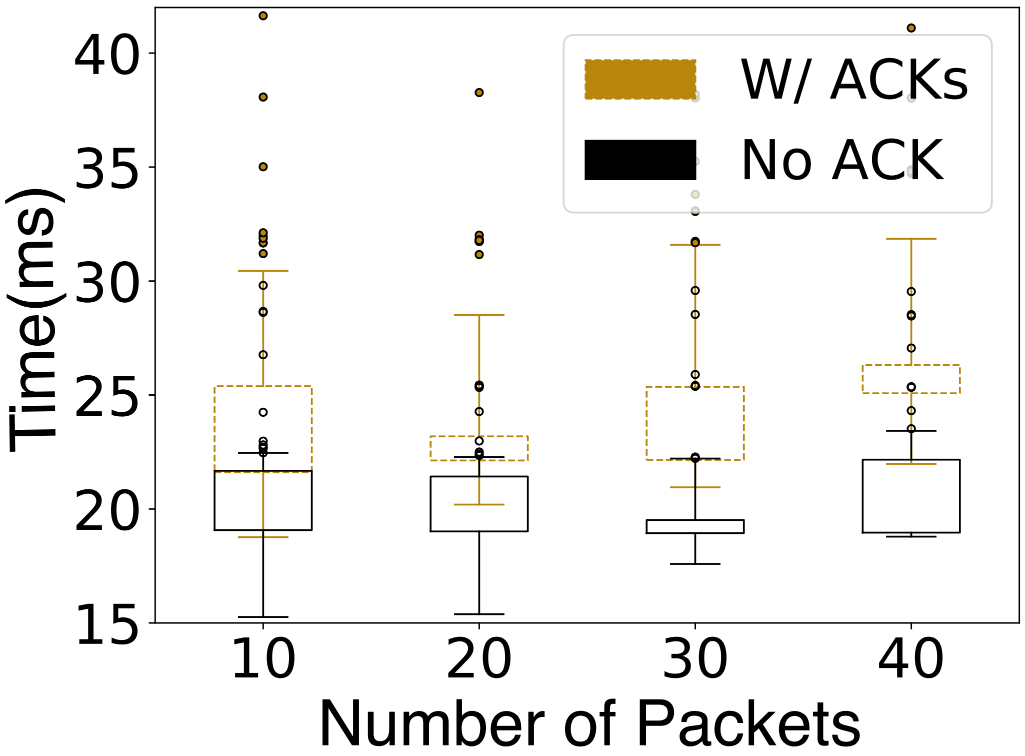
(b) RTT measurement of macOS using 2.4GHz network of a Xiaomi router



(c) RTT measurement of macOS using 5GHz network of a Huawei router

Empirical Test Results (Cont)

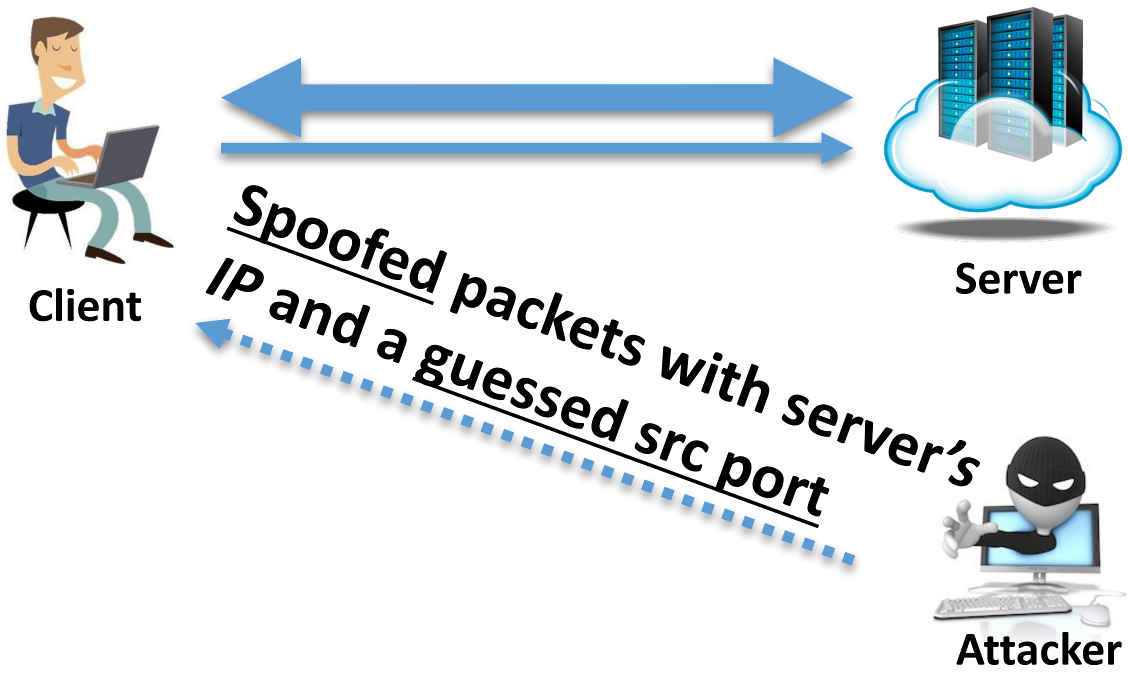




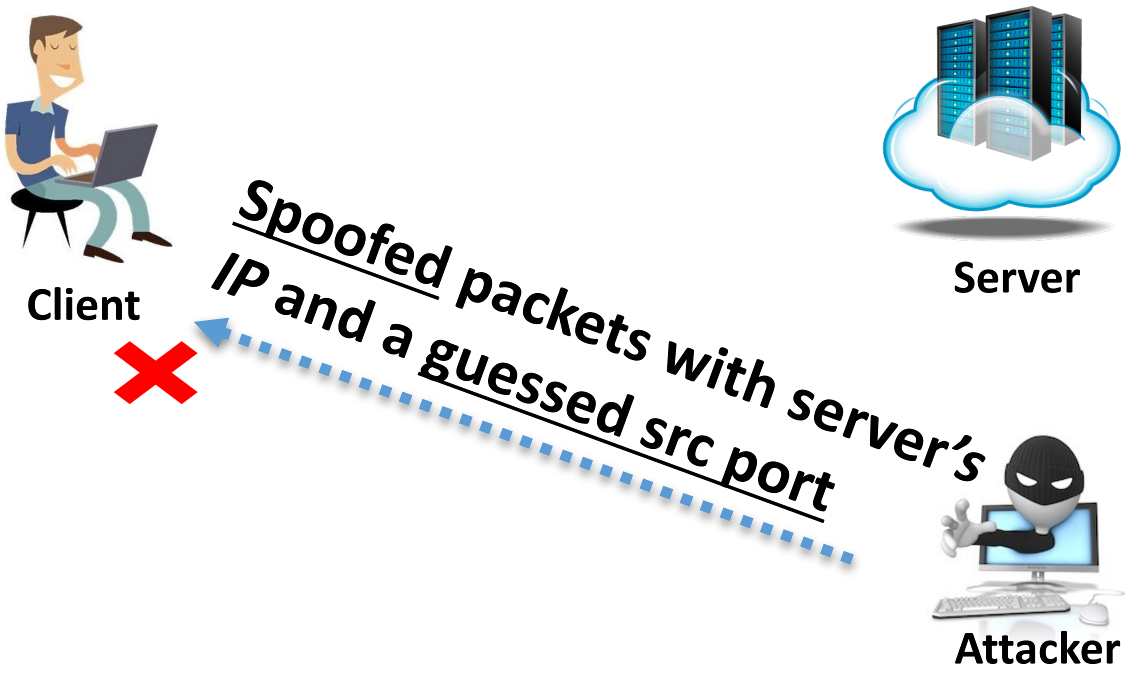
RTT measurement of macOS using 5GHz network of a Xiaomi router at two different locations with RTTs over 20ms

Port Number Inference

Has connection



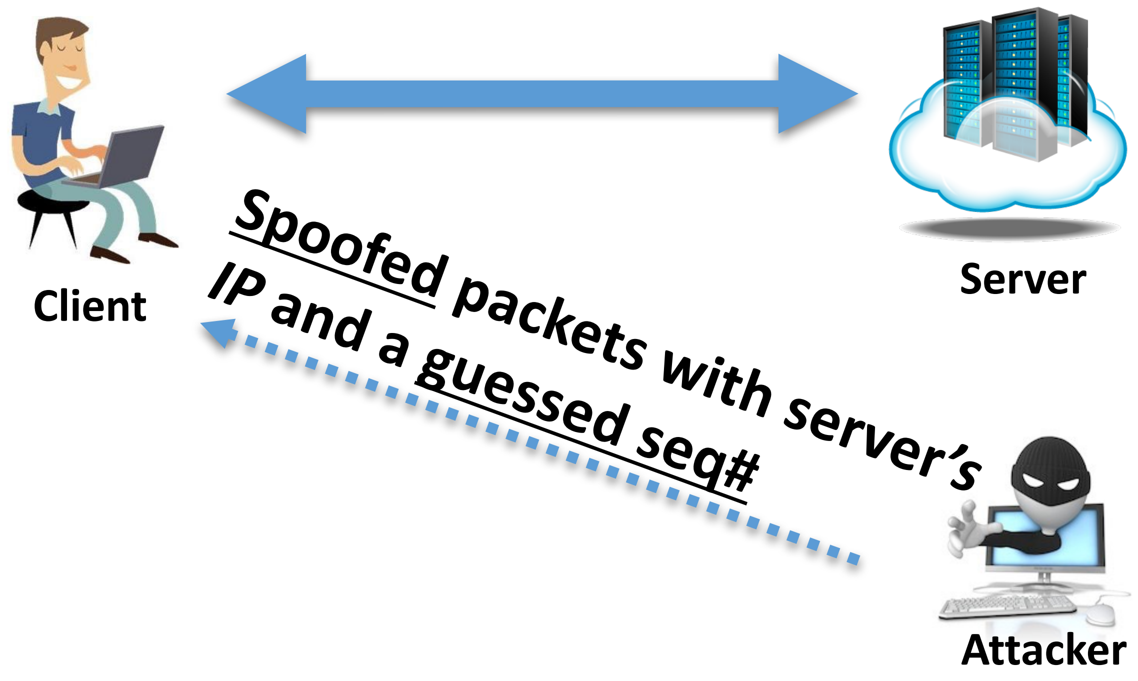
No connection



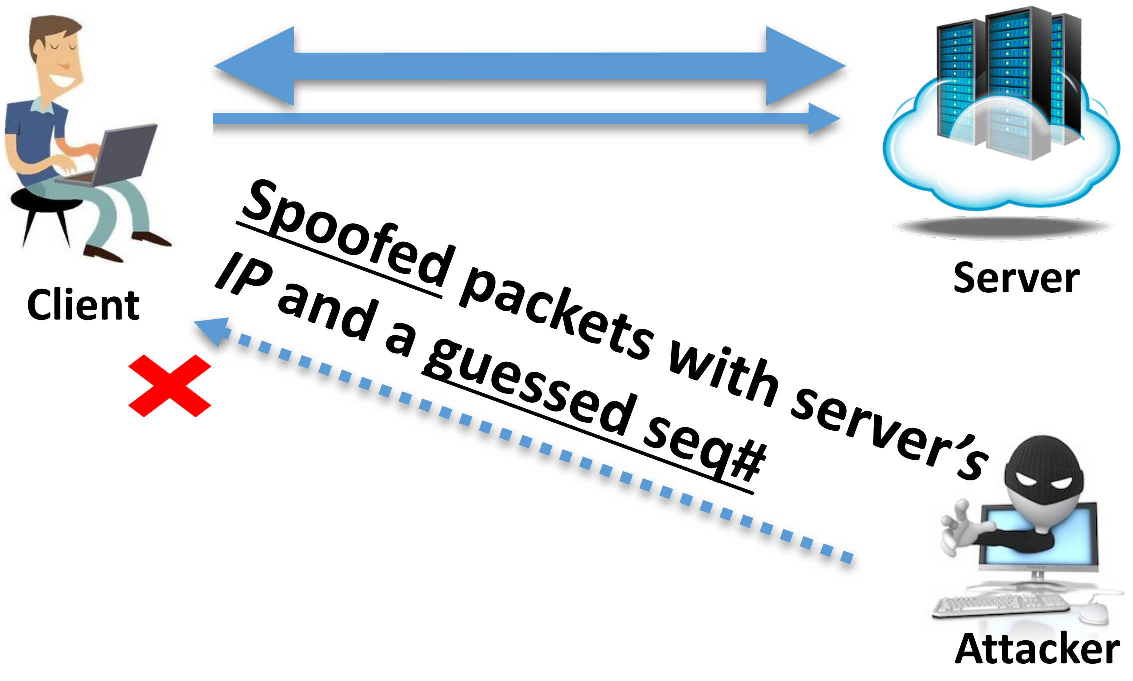
How can the attacker see the difference?

Sequence Number Inference

Seq in-window



Seq out-of-window



TCP Stack Implementations

No.	OS	FLAG	SEQ	ACK	PAYLOAD	#Responses
1	Linux	ACK|SYN|RST	Out-of-window	Any	1	10
3	Linux	ACK|SYN|RST	In-window	> SND.MAX	Any	0
10	MacOS	None|ACK	Out-of-window	Any	Any	10
11	MacOS	None	In-window	Out-of-window	Any	0
17	Windows	ACK|FIN|SYN	Out-of-window	Any	Any	10
18	Windows	ACK|FIN	In-window	Out-of-window	Any	0

Table. Behaviors on different OSes when processing 10 identical packets*

*:See the complete table in our paper

ACK Number Inference

- Implementations of ACK number check varies significantly from one OS to another
- Exploit HTTP specifications and behaviors of tolerant browsers
  - Brute-force ACK number
- Only takes a couple of seconds

Evaluation

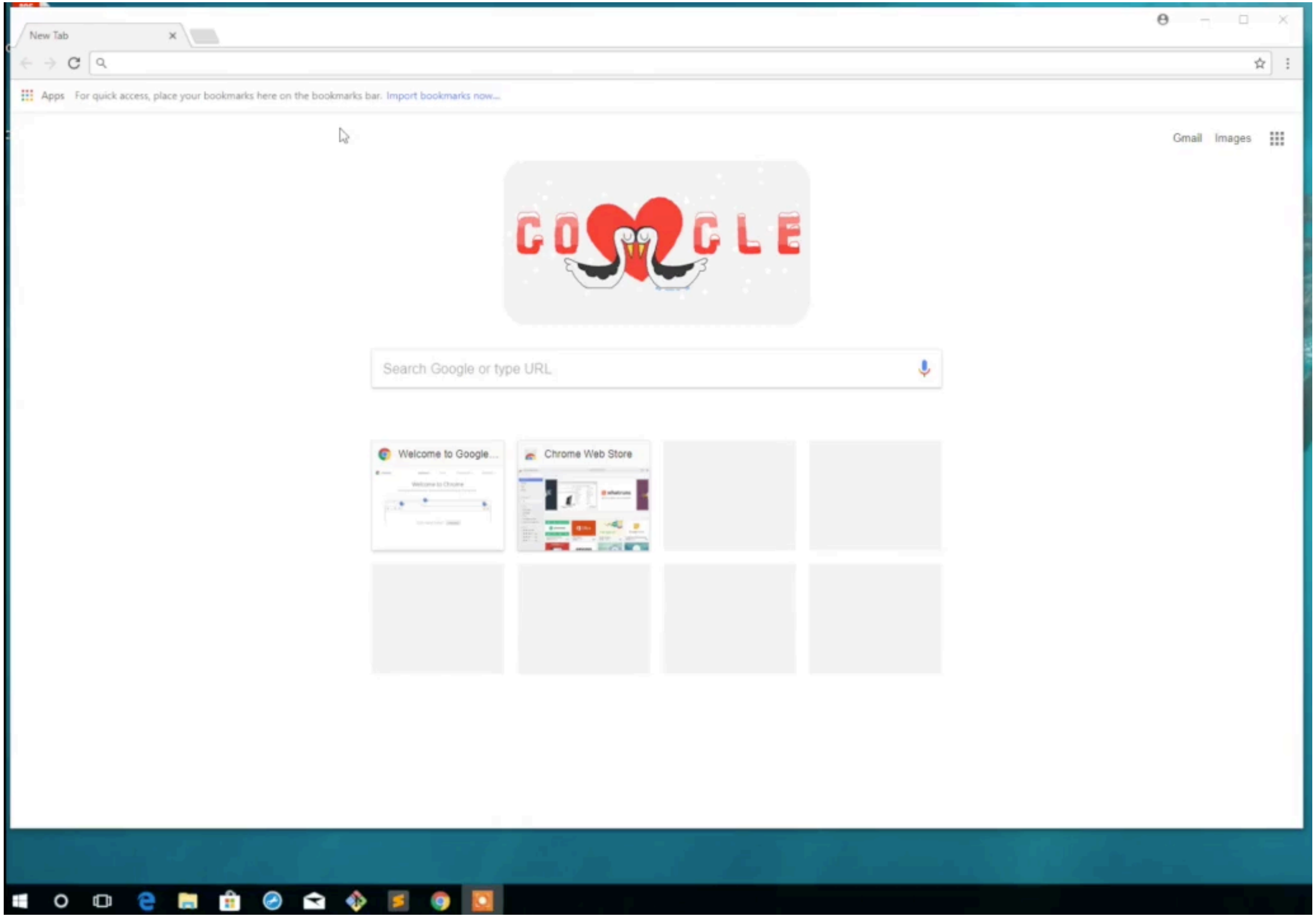
OS	Browser	Success Rate	Avg time cost (s)
Linux	Chrome/Firefox	10/10	188.80
MacOS	Chrome/Firefox	10/10	48.91
Windows	Chrome/Firefox	10/10	43.42

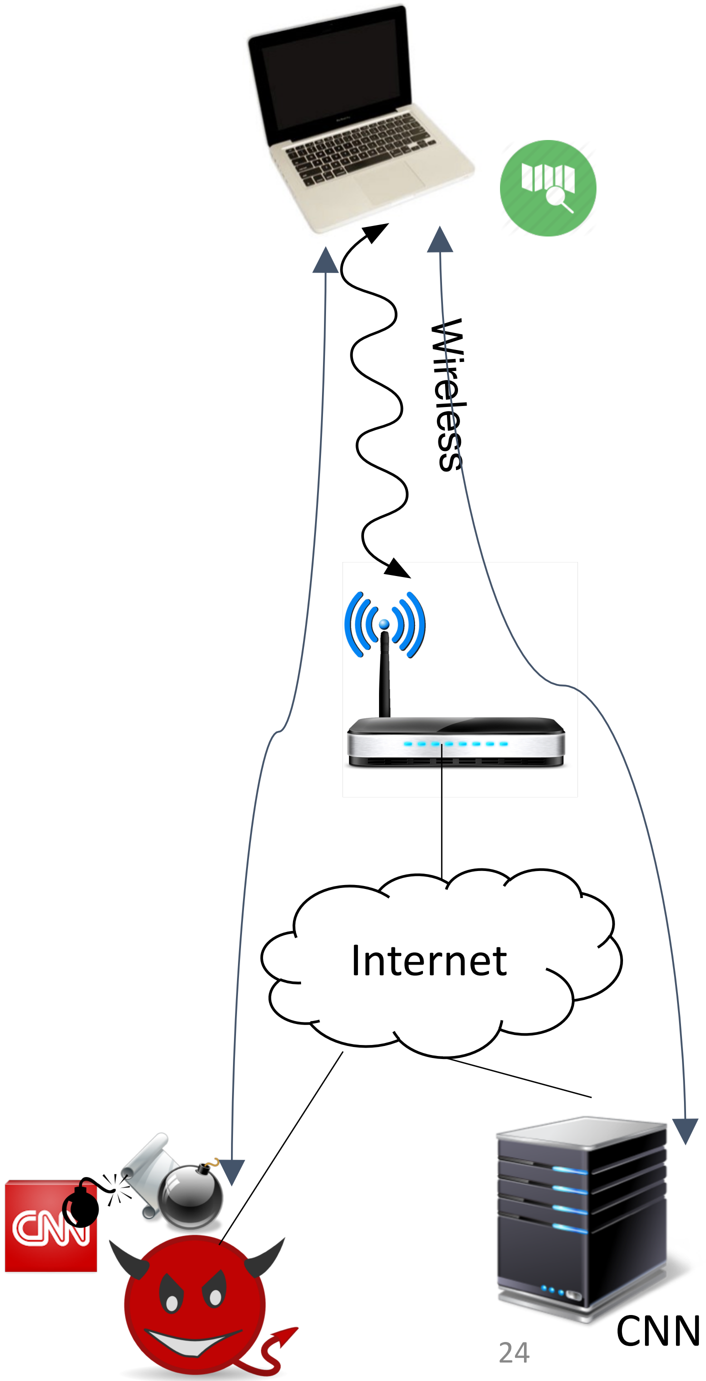
Local result

OS	Browser	Success Rate	Avg time cost (s)
MacOS	Chrome/Firefox	9/10	304.18

Remote result (RTT = 20ms)

Demo: Web Cache Poisoning





How bad?

- Teleconference with IEEE 802.11 working group
- **It’s not possible to be fixed at physical and MAC layers!**

Defenses/Mitigations

- Wireless Layer: Full-duplex Wi-Fi Technology
  - E.g., Frequency-division duplexing, different frequency sub-bands
- TCP Stack: Revisit TCP Specifications
  - E.g., Rate limit responses for incoming packets with out-of-window SEQ
- Application Layer: Deploy HSTS (HTTP Strict Transport Security)
  - Preventing access via the insecure HTTP protocol

Conclusion

- A new timing side channel inherent in all generations of IEEE 802.11 or Wi-Fi technology
- Comprehensive analysis of TCP stack implementations in macOS, Windows, and Linux
- Implement practical TCP injection attacks
- Propose possible defenses
- https://github.com/seclab-ucr/tcp_exploit

Q&A

Thanks for your attention!