CS 563 - Advanced Computer Security:
Security Measurement

Professor Adam Bates
Fall 2018

Administrative

Learning Objectives:

- Discuss two recent studies that use measurement methods
- Survey broad topics in the "security measurement" area

Announcements:

- Reaction paper was due today (and all classes)
- Feedback for reaction papers soon
- "Preference Proposal" Homework due 9/24
-

Reminder: Please put away (backlit) devices at the start of class

Measuring Internet Censorship

Reports suggest

Internet censorship practices

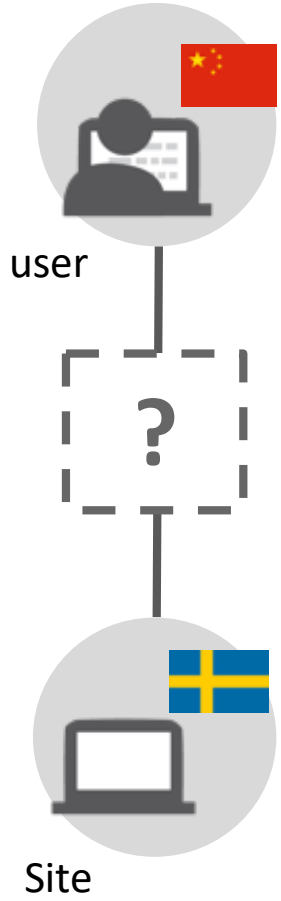
are diverse in their methods, targets, timing,

differing by regions, as well as across time.

Measuring Internet Censorship

Problem:

- How can we detect whether pairs of hosts around the world can talk to each other?



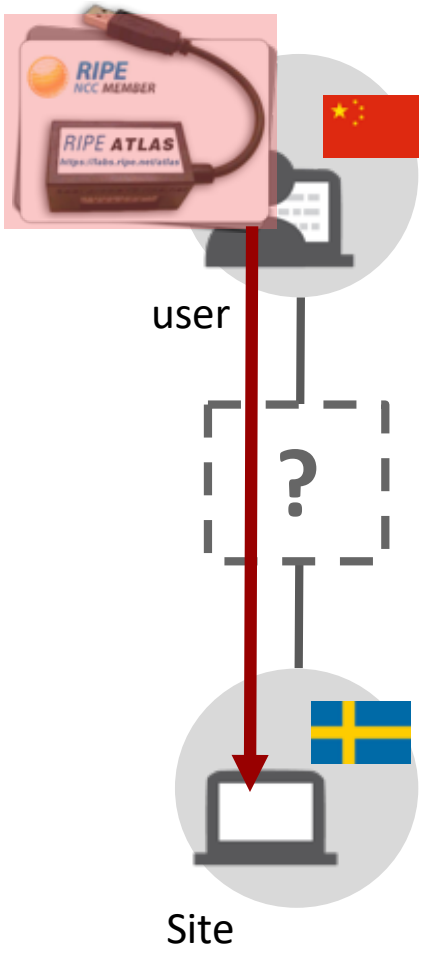
Measuring Internet Censorship

Problem:

- How can we detect whether pairs of hosts around the world can talk to each other?

State of the Art:

- Deploy hardware or software at hosts (RIPE Atlas, OONI probe)
- Ask people on the ground, or use VPNs, or research networks (PlanetLab)



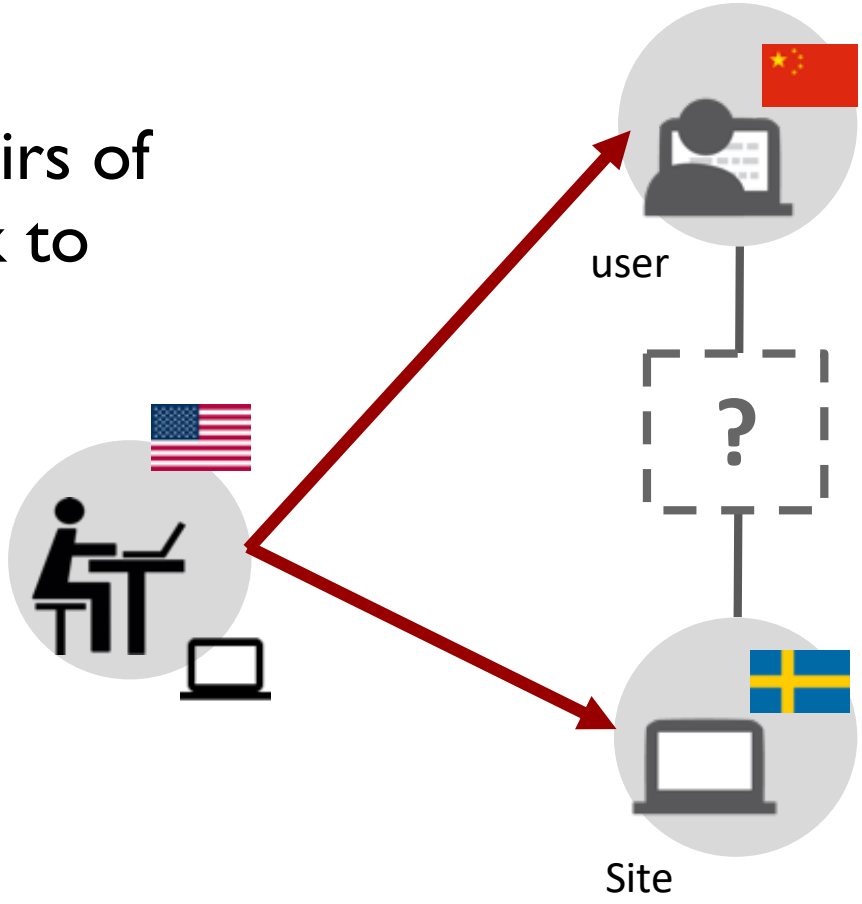
THREE KEY CHALLENGES:
Coverage, ethics, and continuity

Measuring Internet Censorship

Problem:

- How can we detect whether pairs of hosts around the world can talk to each other?

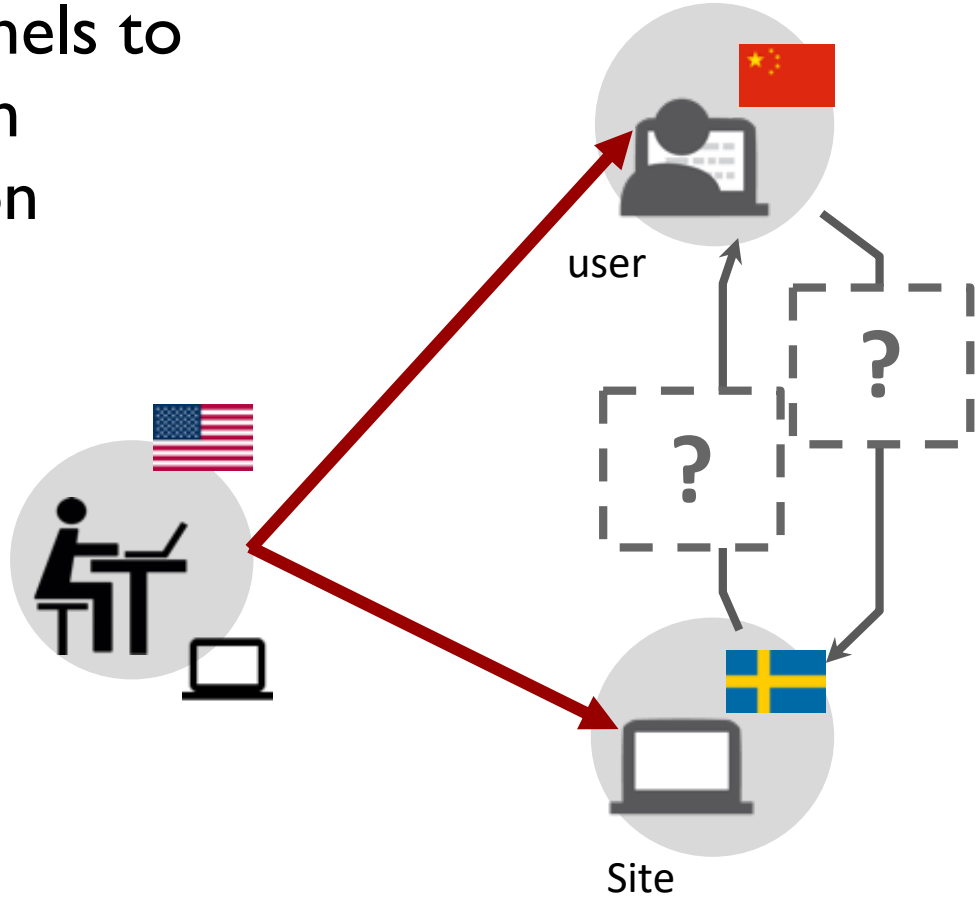
… from somewhere else in the world??



Hybrid Idle (Spooky) Scan

Spooky Scan: uses TCP/IP side channels to detect whether a user and a site can communicate (and in which direction packets are blocked).

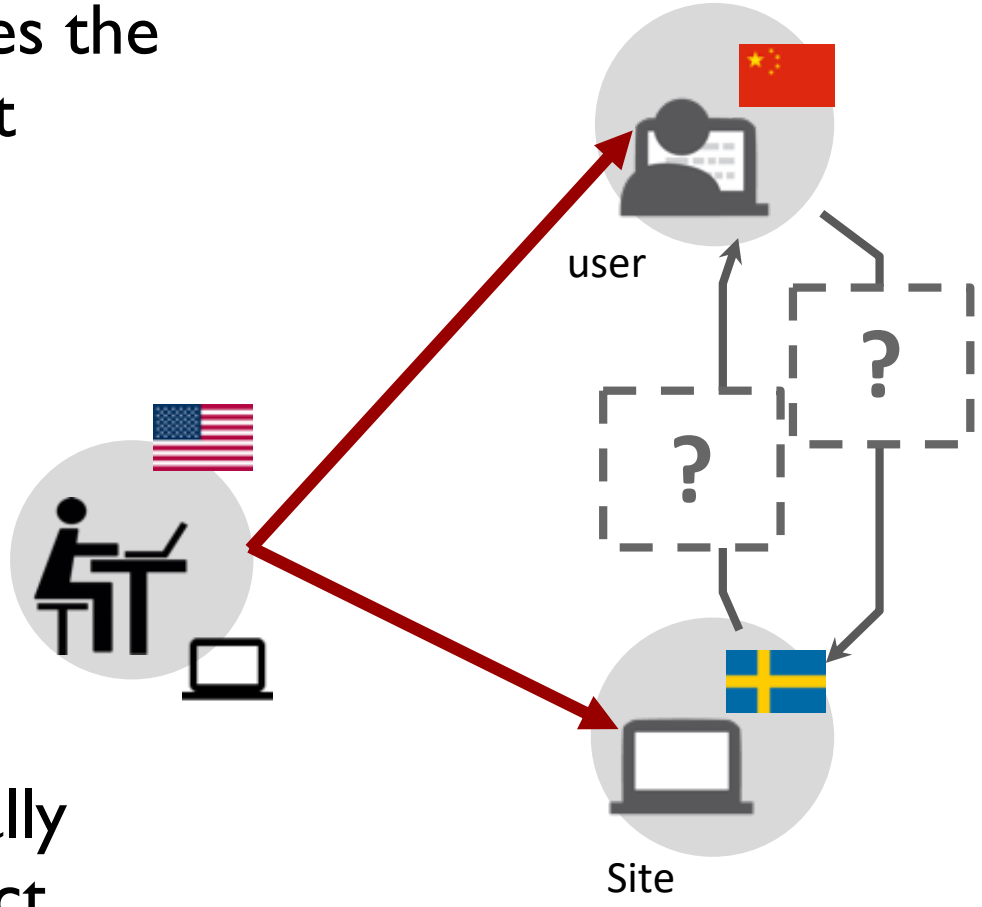
Goal: Detect blocking from off-path



* **TCP Idle Scan** Antirez, (Bugtraq 1998)
* **Detecting Intentional Packet Drops on the Internet via TCP/IP Side Channels**
  Roya Ensafi, Knockel, Alexander, and Crandall (PAM '14)
* **Idle Port Scanning and Non-interference Analysis of Network Protocol Stacks Using Model Checking**
  Roya Ensafi, Park, Kapur, and Crandall (Usenix Security 2010)

Hybrid Idle (Spooky) Scan

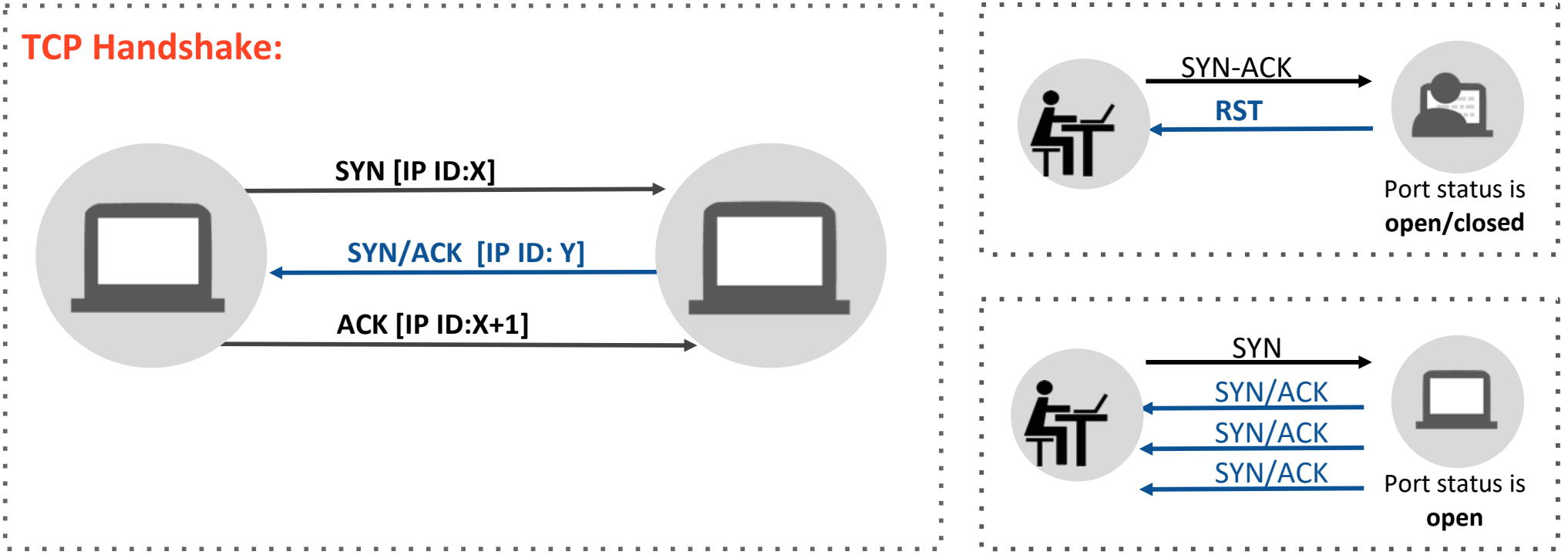
Augur is a follow up system that uses the same TCP/IP side channels to detect blocking from off-path.



Goals: Scalable, ethical, and statistically robust system to continuously detect blocking.

How does this work?

TCP/IP provides several building blocks:



How does this work?

Requirements for each participant:

“User” (Reflector)

Must maintain a
global value for IP ID

Site

Open port and
retransmitting SYN-ACKs

Measurement Machine

Must be able to spoof packets

Spooky Scans

Measurement machine

Reflector IP ID

Reflector

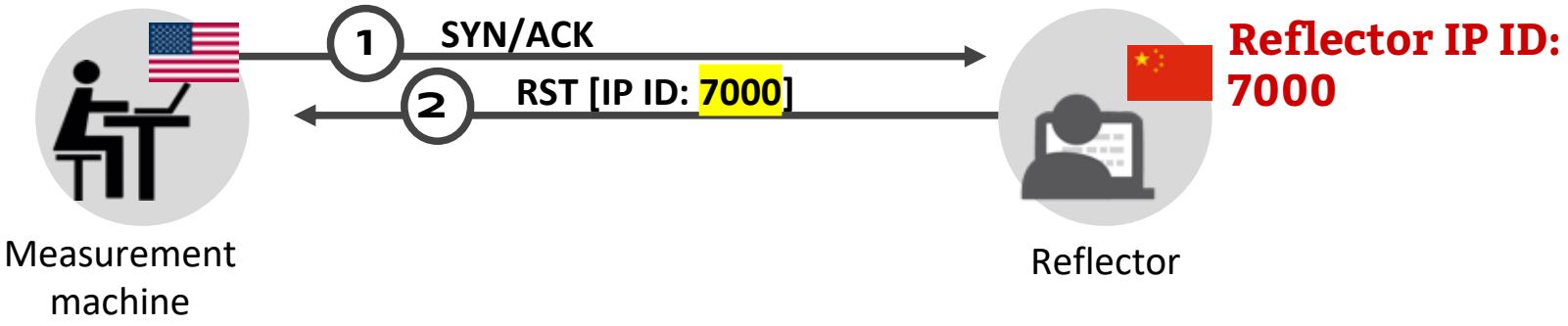
Site

No direction blocked

Spooky Scans

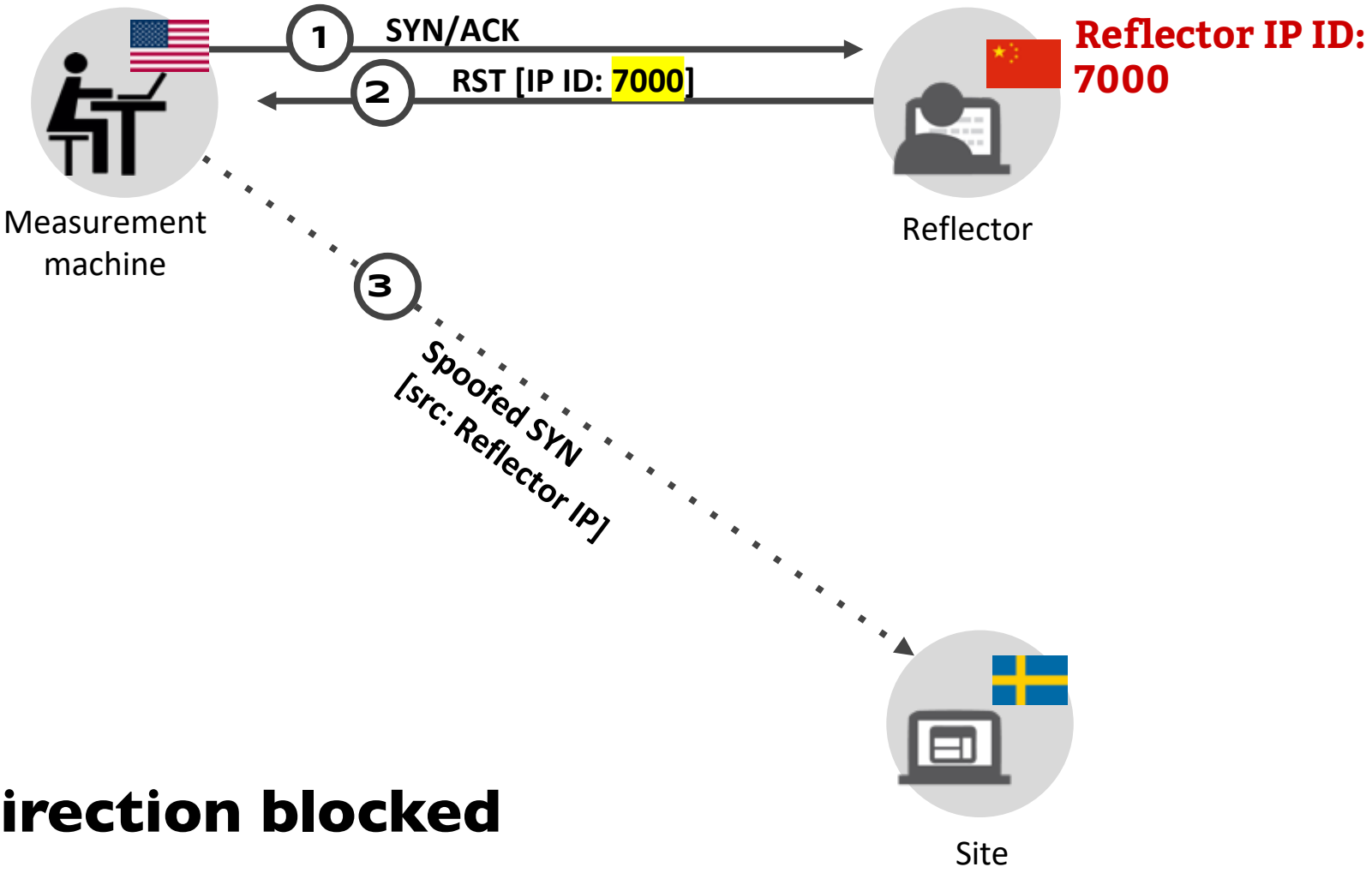
No direction blocked

Spooky Scans



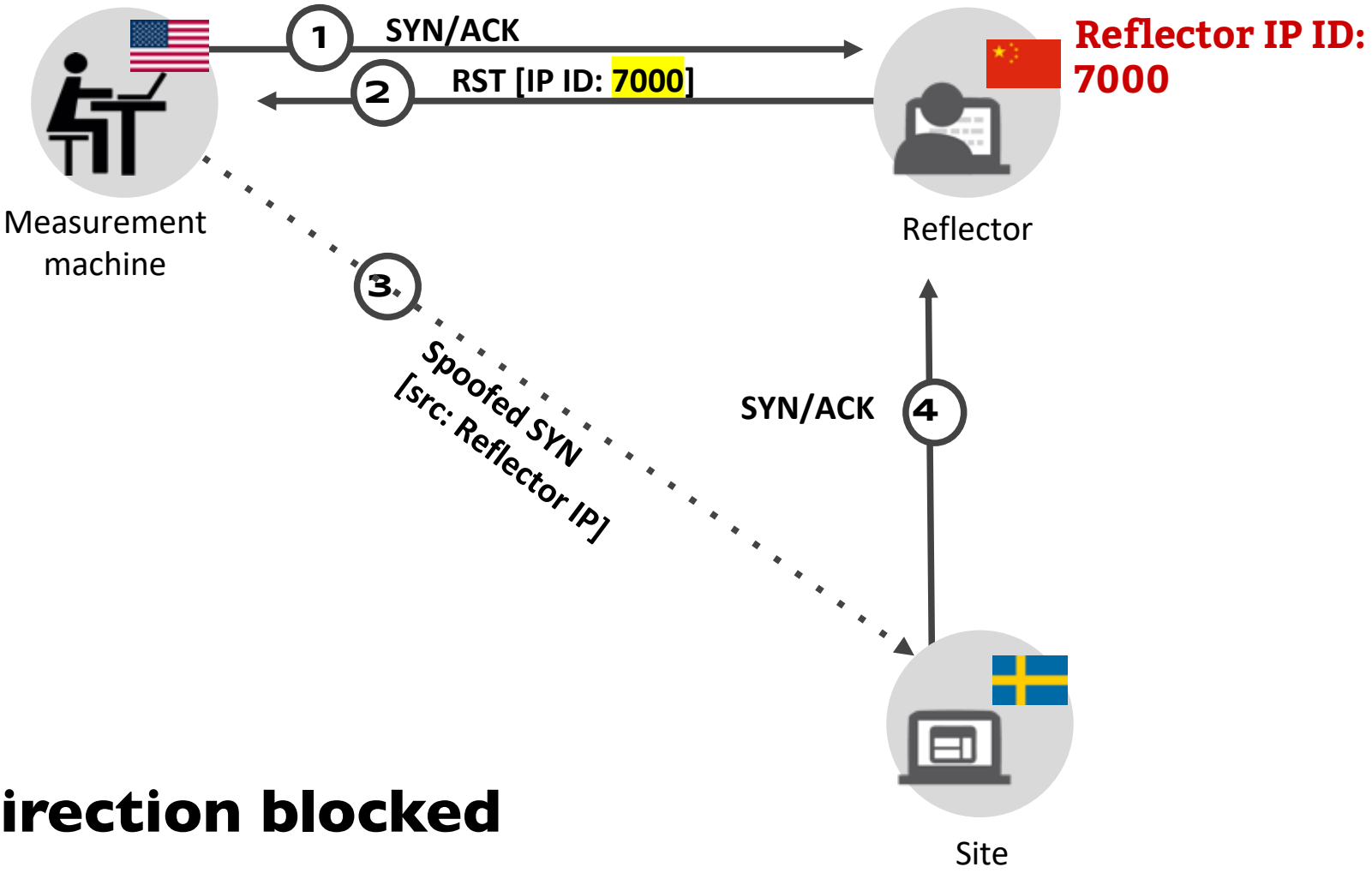
No direction blocked

Spooky Scans



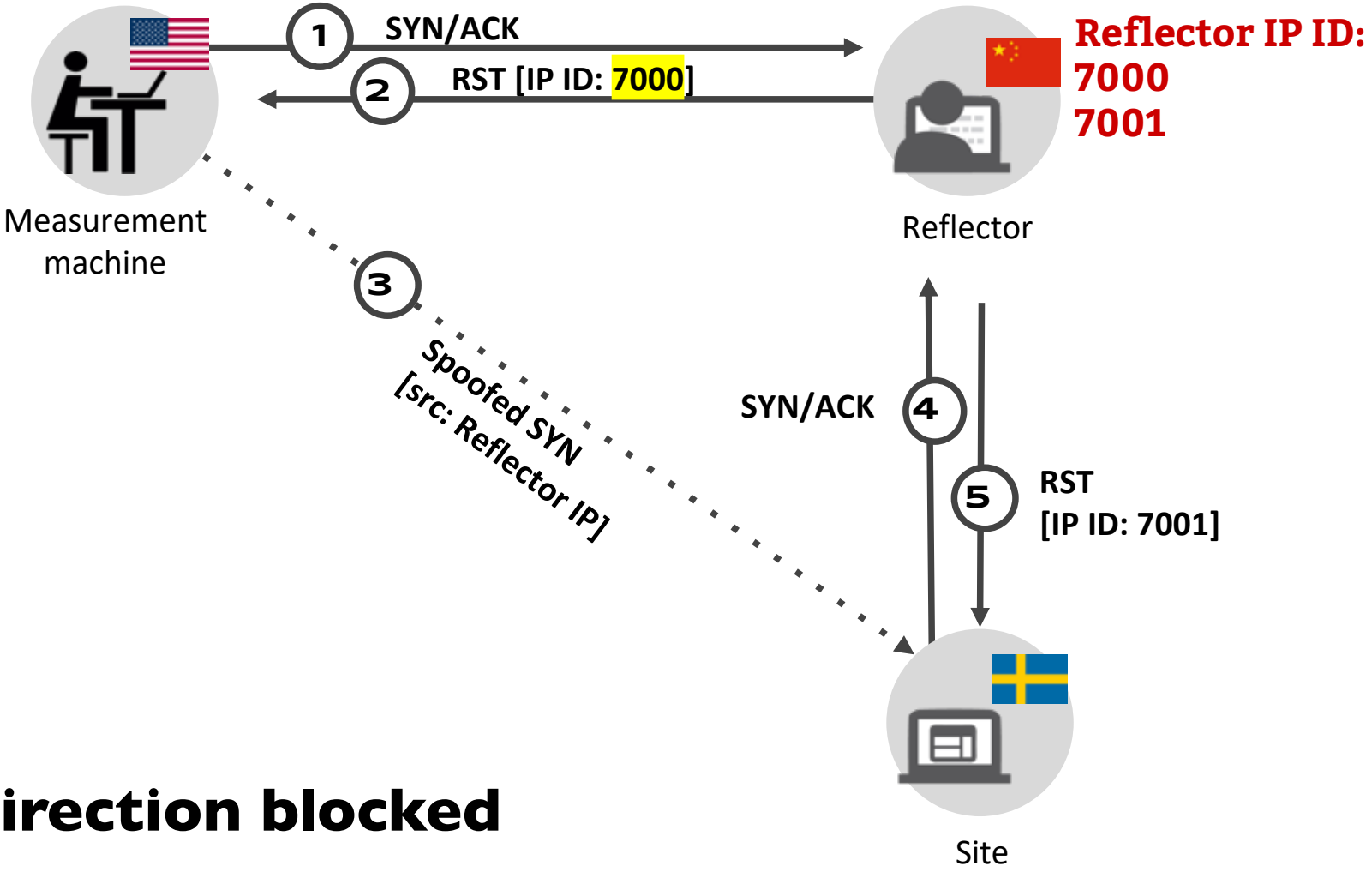
No direction blocked

Spooky Scans



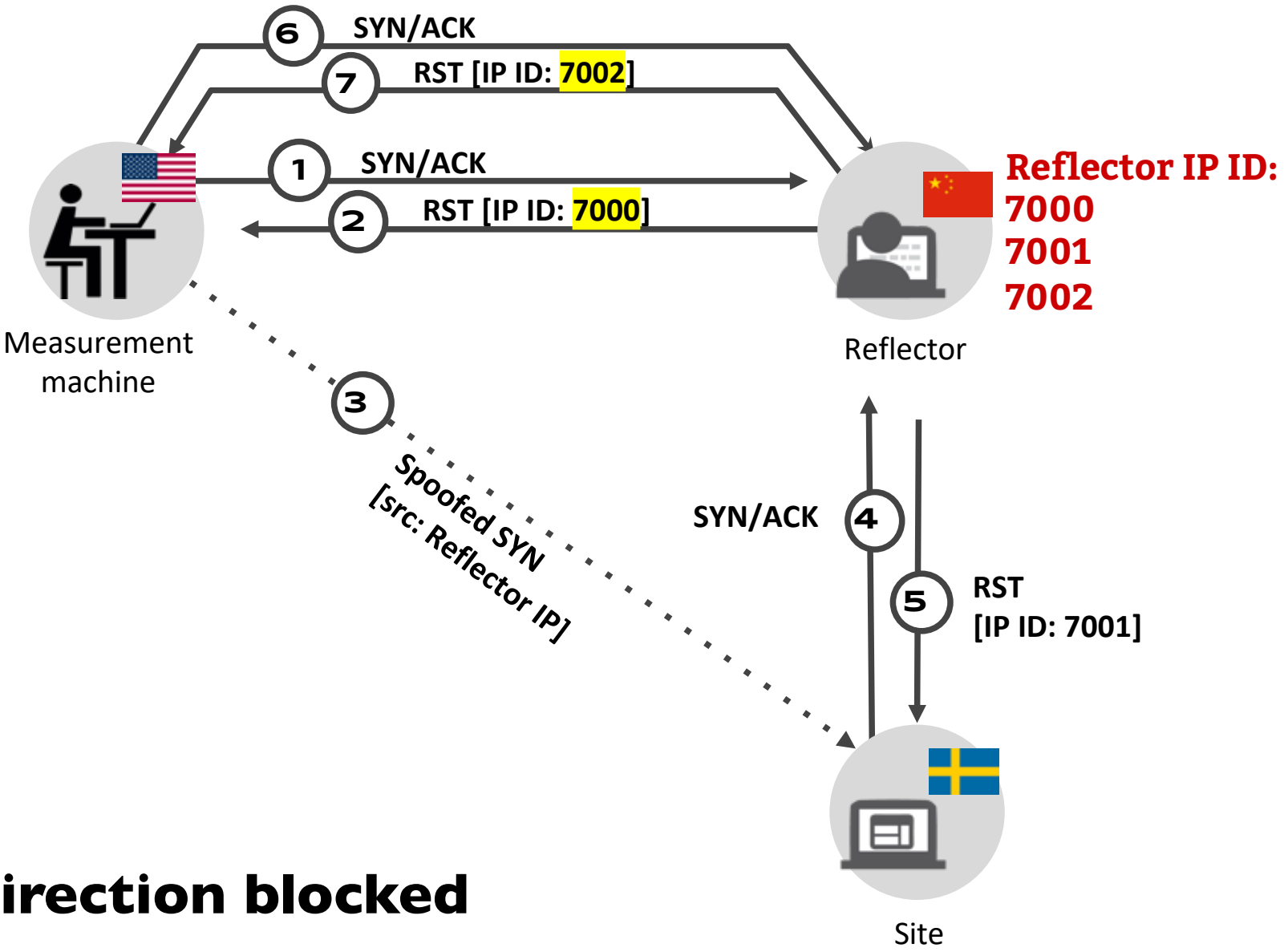
No direction blocked

Spooky Scans



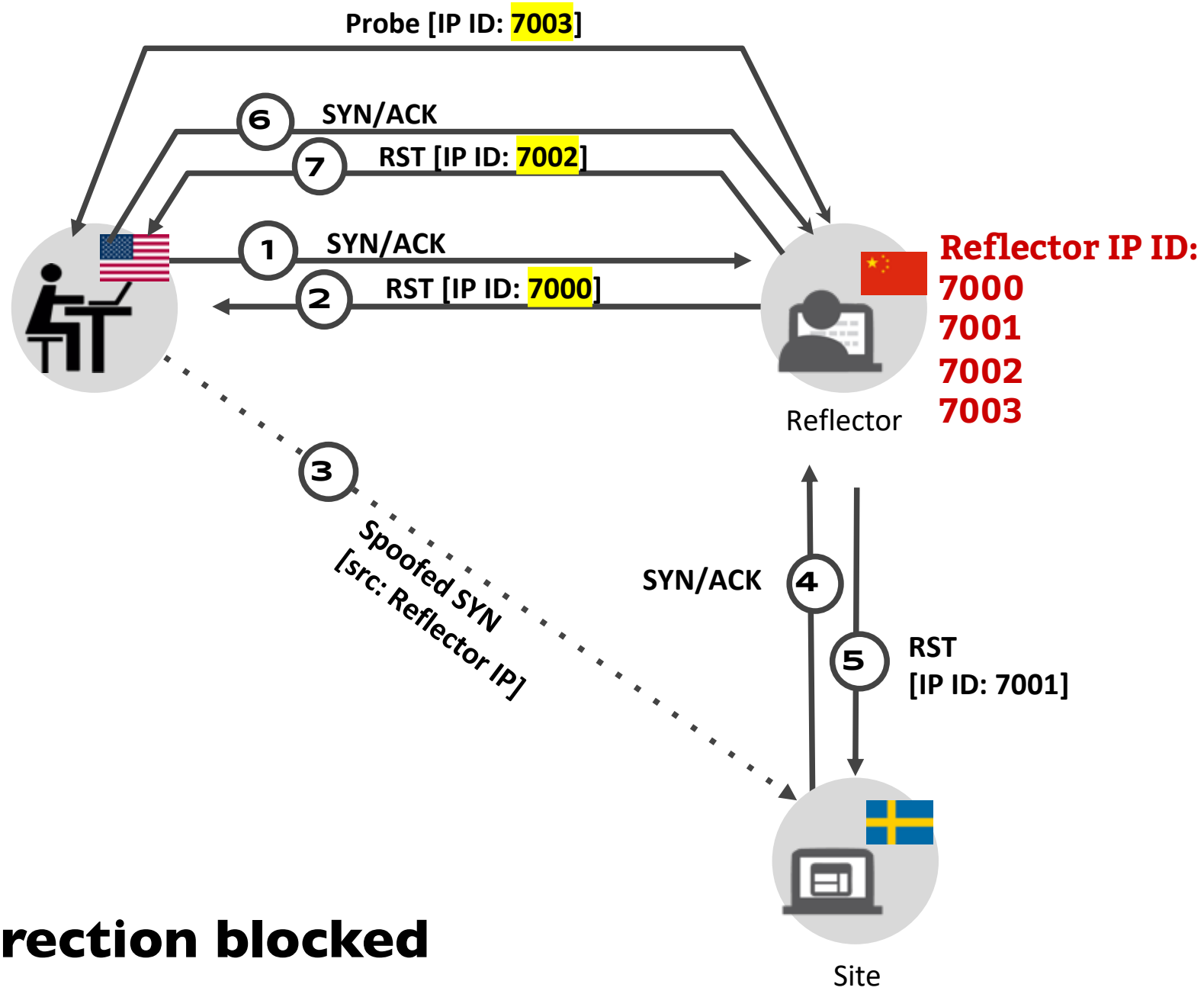
No direction blocked

Spooky Scans



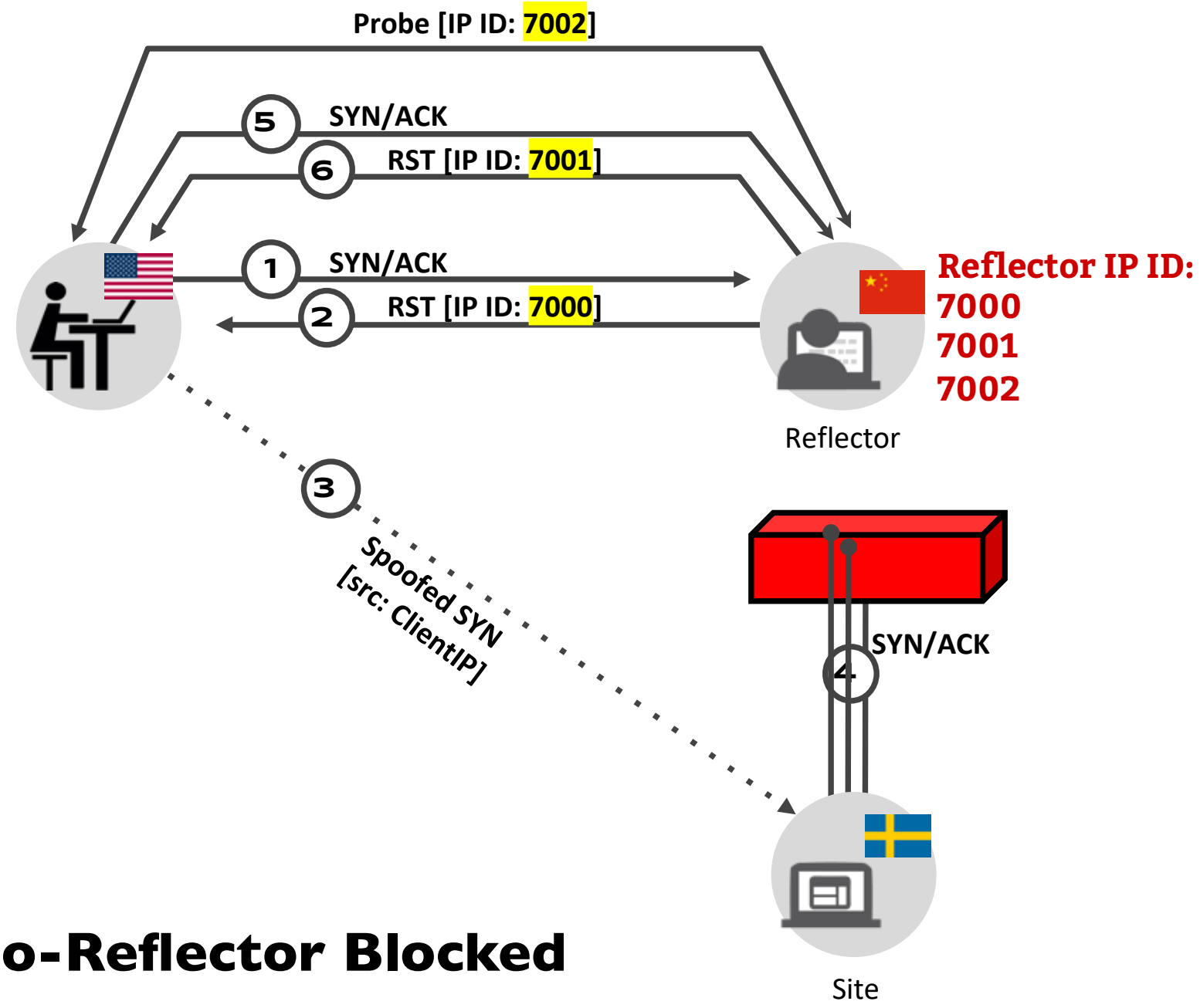
No direction blocked

Spooky Scans



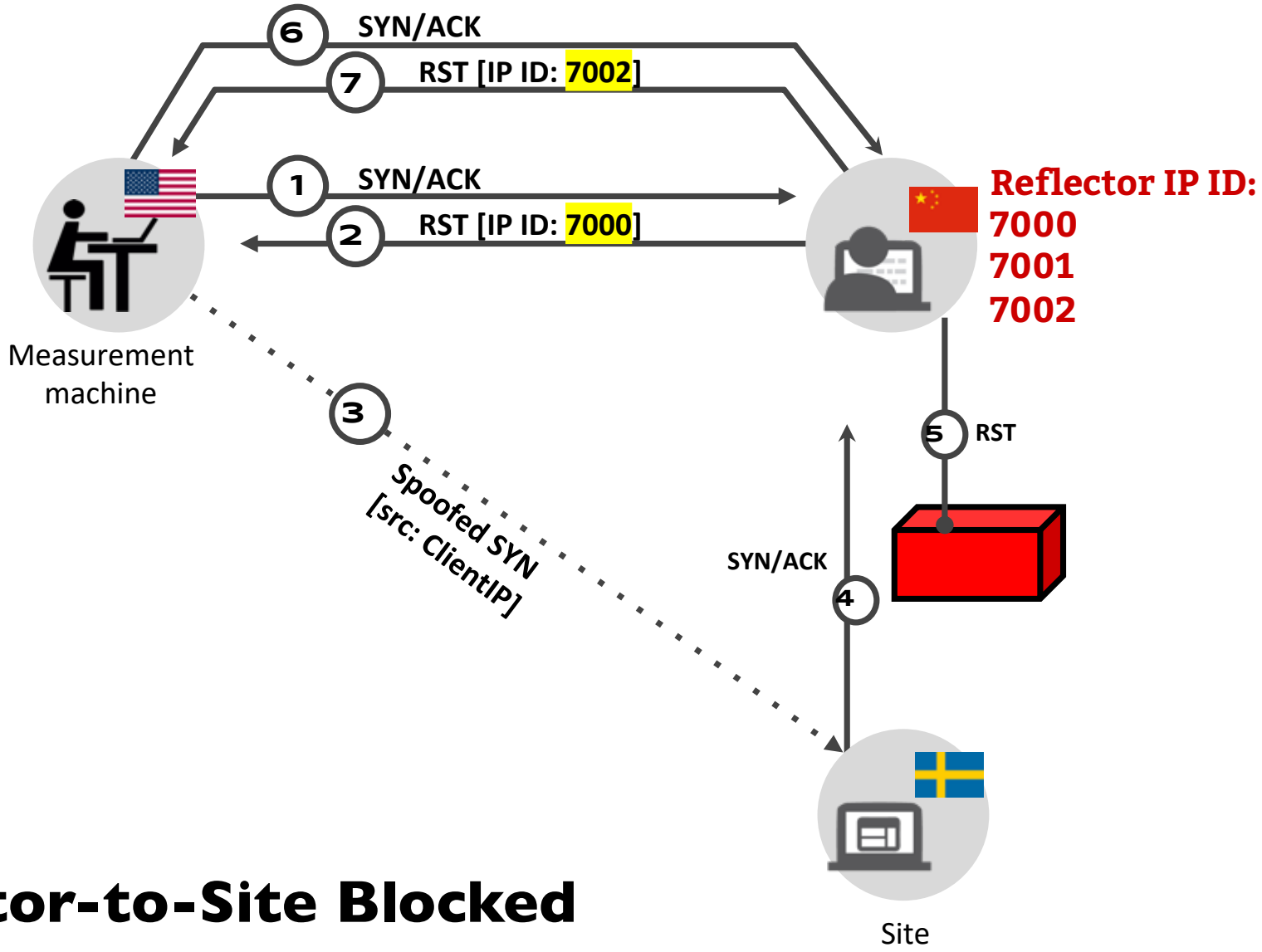
No direction blocked

Spooky Scans



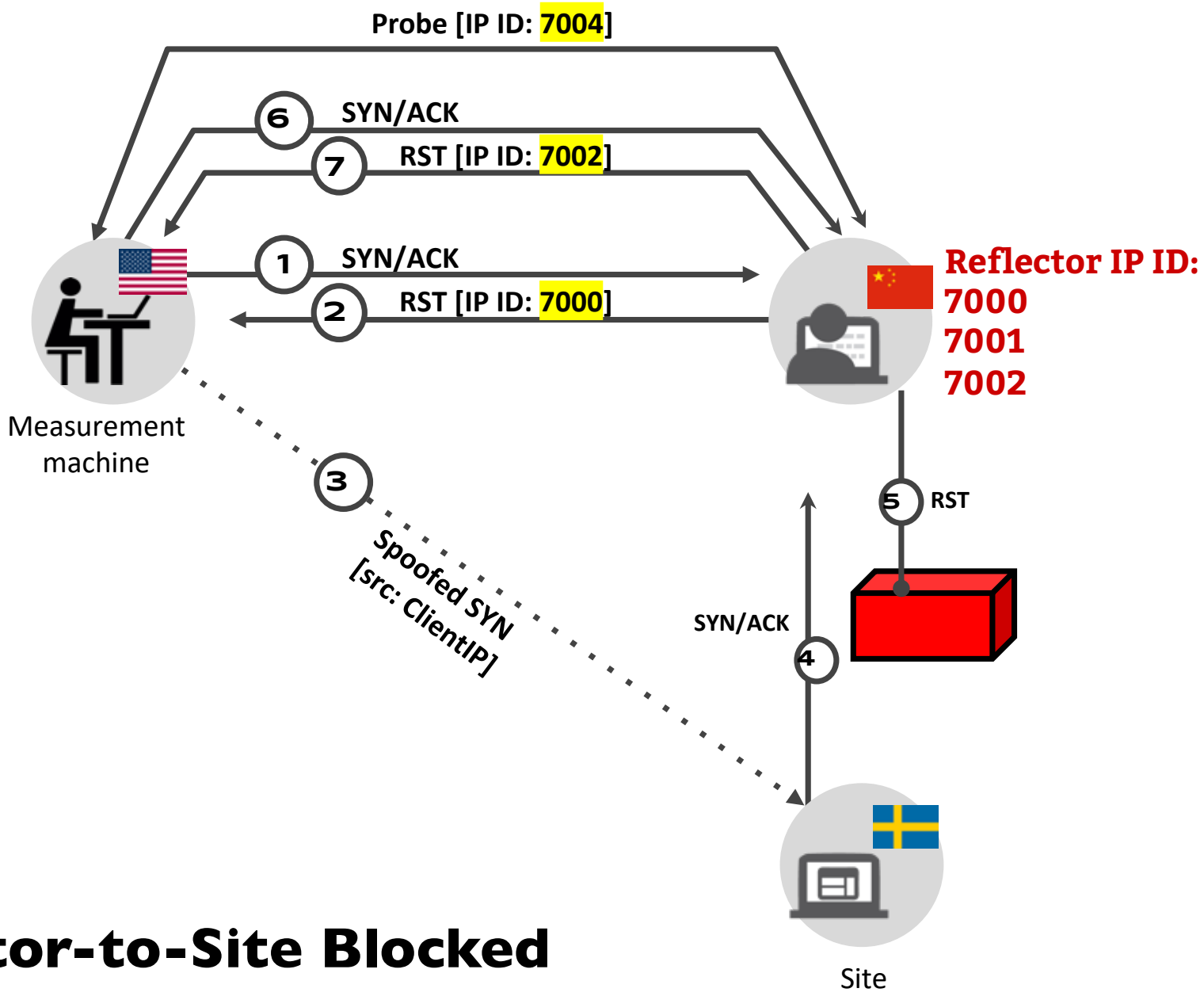
Site-to-Reflector Blocked

Spooky Scans



Reflector-to-Site Blocked

Spooky Scans



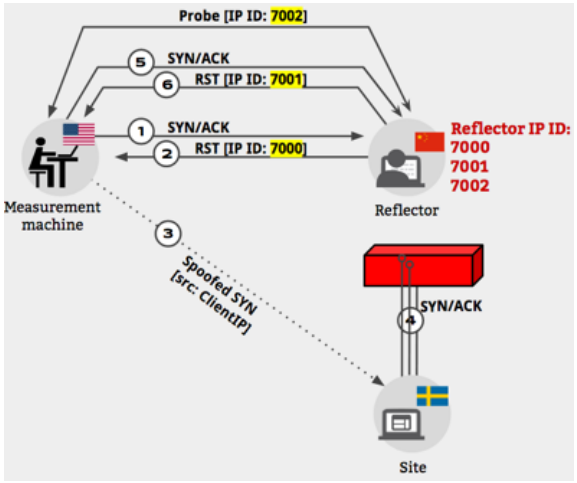
Reflector-to-Site Blocked

Spooky Scans

We can use the deltas for each IP packet ID to differentiate blockage:

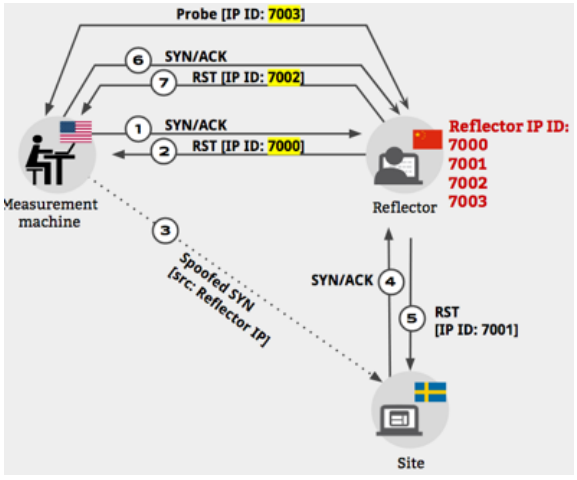
Site-to-Reflector Blocked

Δ IP ID1 = 1
Δ IP ID2 = 1



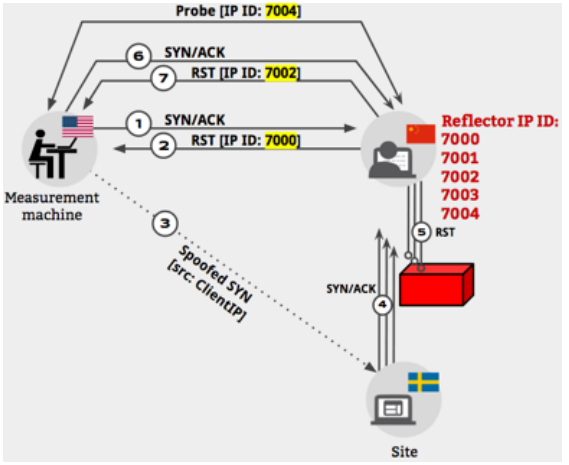
No Direction Blocked

Δ IP ID1 = 2
Δ IP ID2 = 1



Reflector-to-Site Blocked

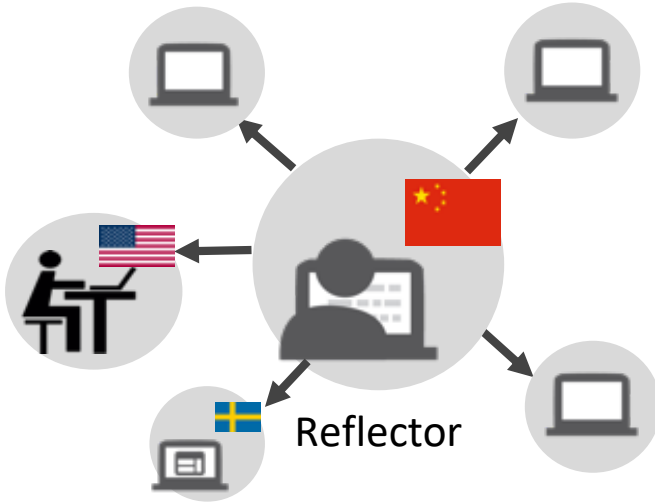
Δ IP ID1 = 2
Δ IP ID2 = 2



What about noise?

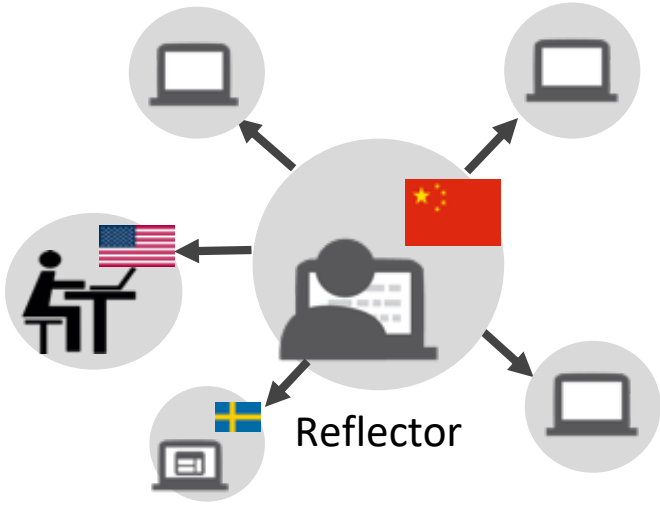
Reflectors will be making other Internet connections. How to cope?

- Amplify the signal by repeated probing (i.e., N probes instead of 1).
- Repeat the experiment to account for packet loss and other network pathologies.



What about noise?

Not all reflectors will have the same noise levels. How to adjust?



Probing Methodology:

Until we have high enough confidence (or up to):

Run
- For first 4s, query IPID every sec
  - Send 10 spoofed SYNs
  - Query IPID
- Query IPID

Repeat runs and

use Seq. Hypothesis Testing

to gradually build confidence.

Sequential Hypothesis Testing

Defining a Random Variable:

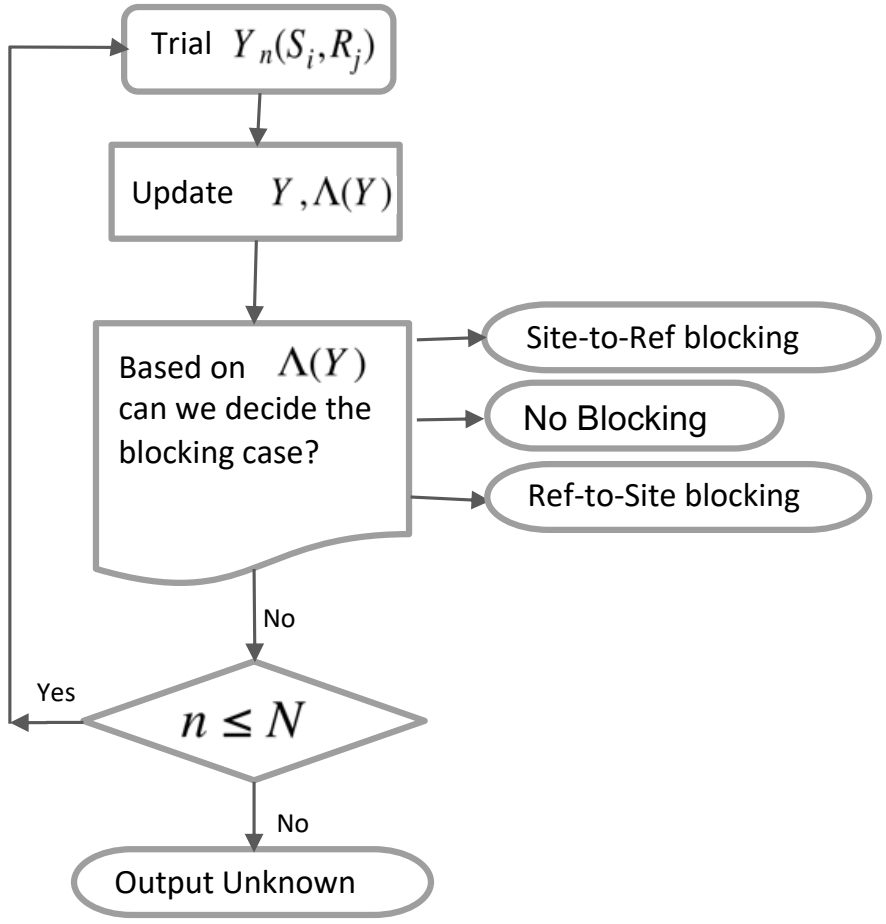
$$Y_n(S_i, R_j) = \begin{cases} 1 & \text{if no IPID acceleration occurs} \\ 0 & \text{if IPID acceleration occurs} \end{cases}$$

Calculate known outcome probabilities:

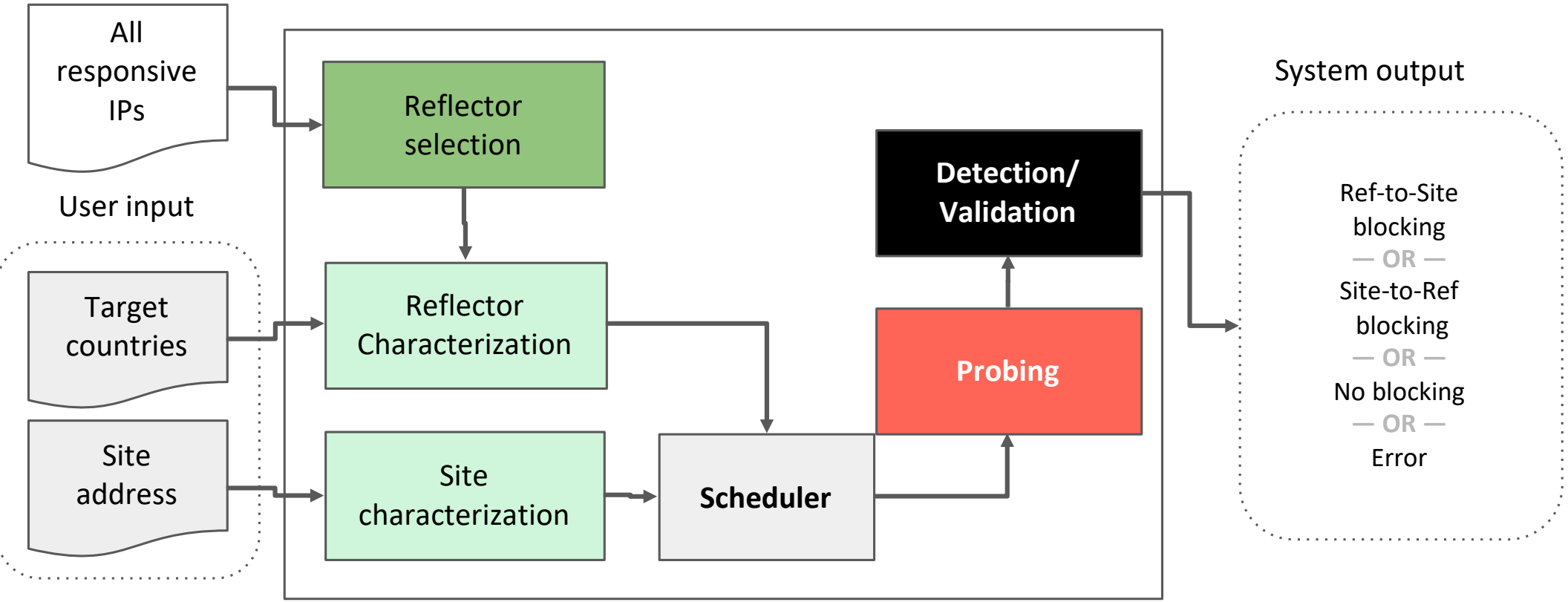
Prior 1: Prob. of no IPID acceleration when there is blocking
Prior 2: Prob. of IPID acceleration when there is no blocking

Maximum Likelihood Ratio

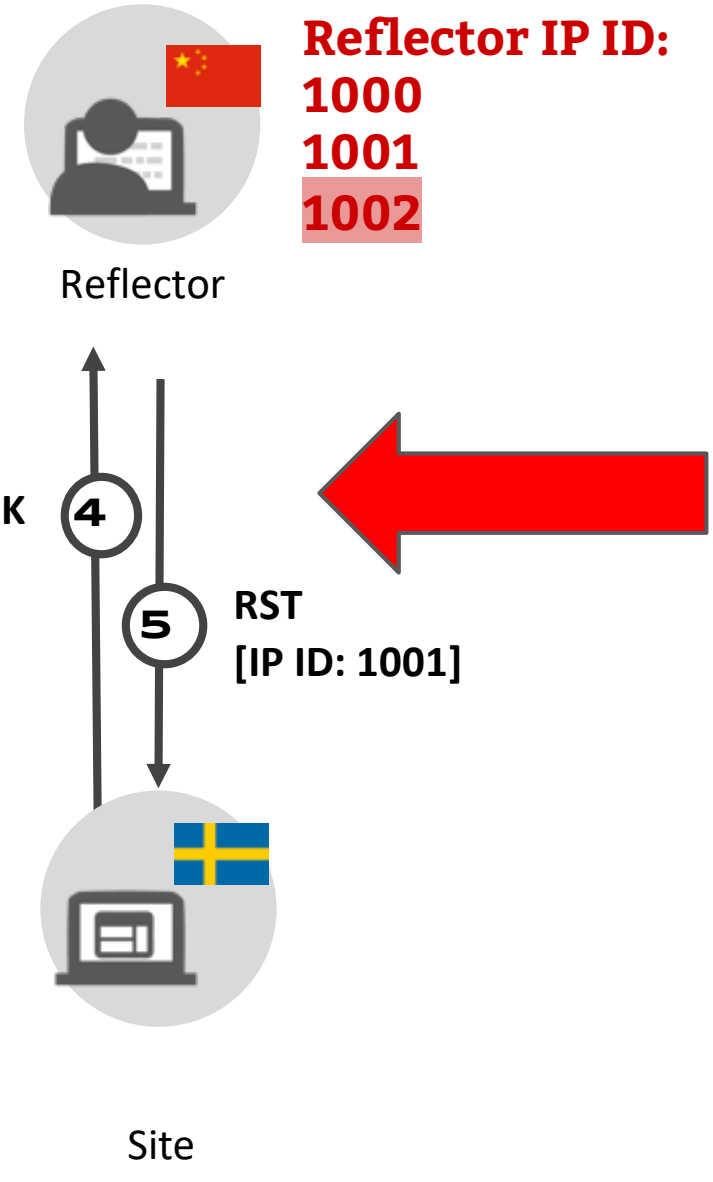
$$\Lambda(Y) \equiv \prod_{n=1}^{N} \frac{Pr[Y_n|Blocking]}{Pr[Y_n|No\ Blocking]}$$



Augur Framework



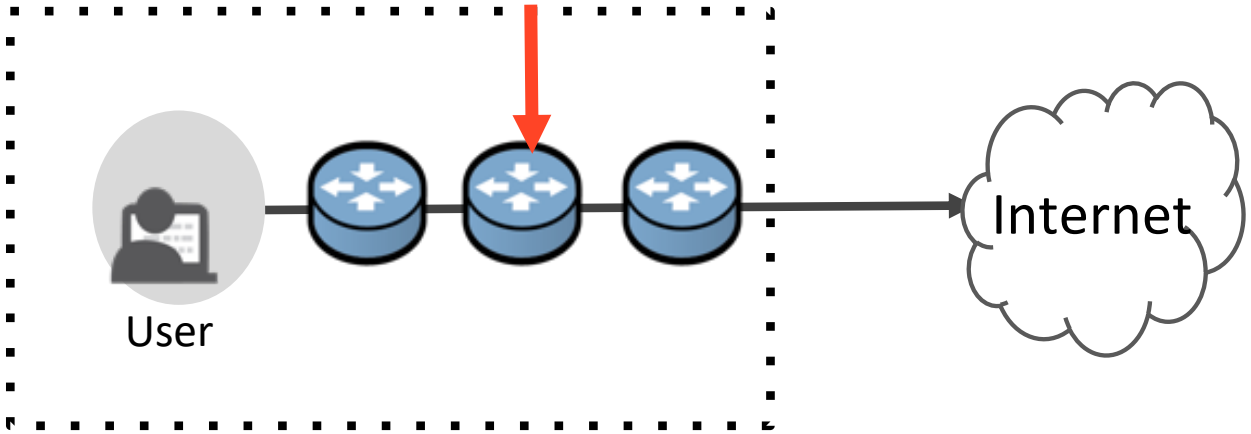
Ethical Considerations



Probing banned sites from users' machines creates risk for user?

Ethical Considerations

Solution: Only probe infrastructure devices.



Global IP ID	22.7 million	236 countries (and dependent territories)
Two hops back from end user	**53,000**	**180 countries**

Measurement Study

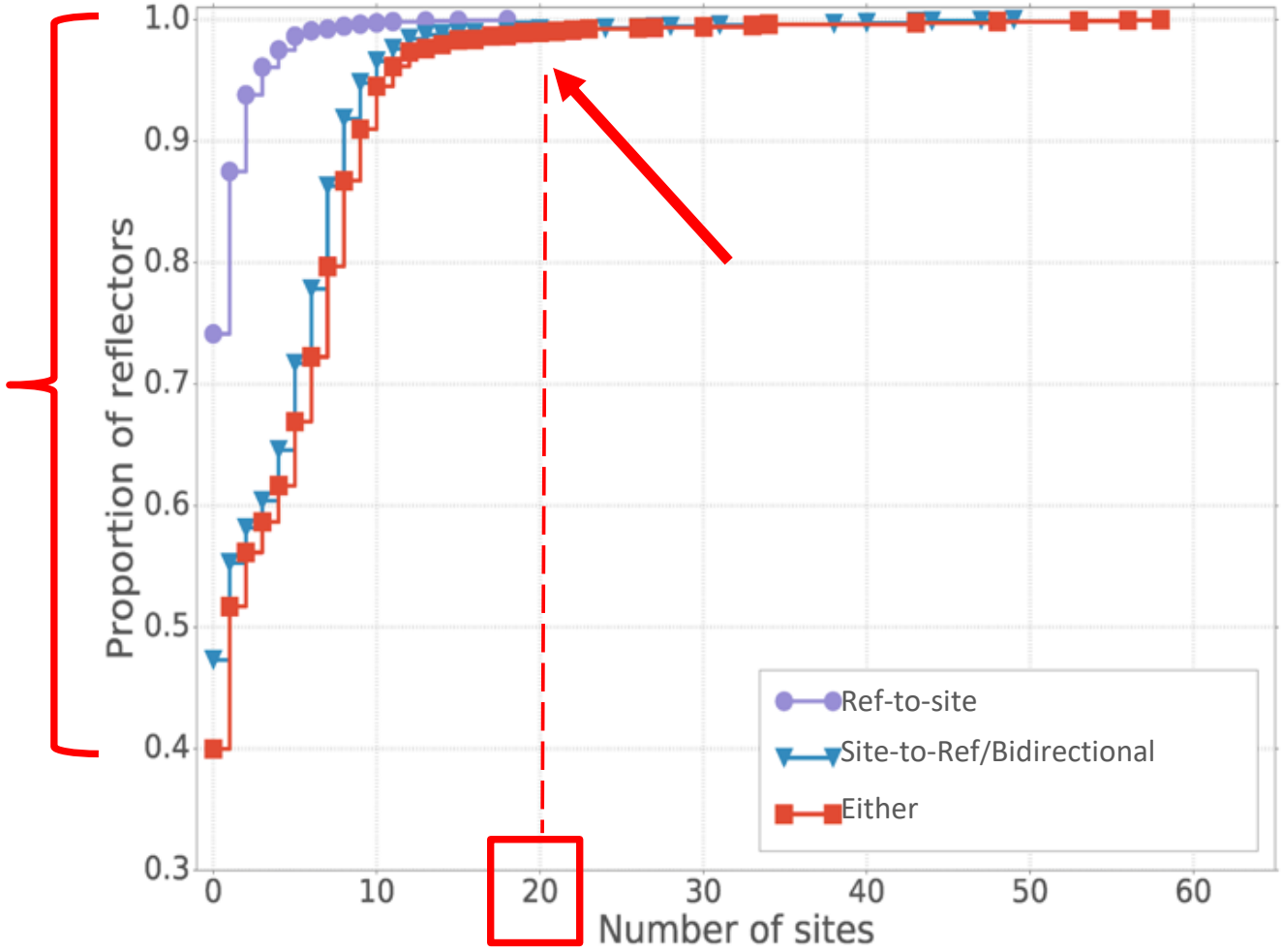
- 2,050 Reflectors
- 2,134 sites (Citizen Lab list + Alexa Top-10K)
- 47 Measurements per site per reflector
  - 207,600,000 measurements total
- ***How do we know Augur is working correctly?***

Validation Checks

One reflector shouldn't show all sites blocked

- 99% of reflectors experience disruption only for 20 or fewer website

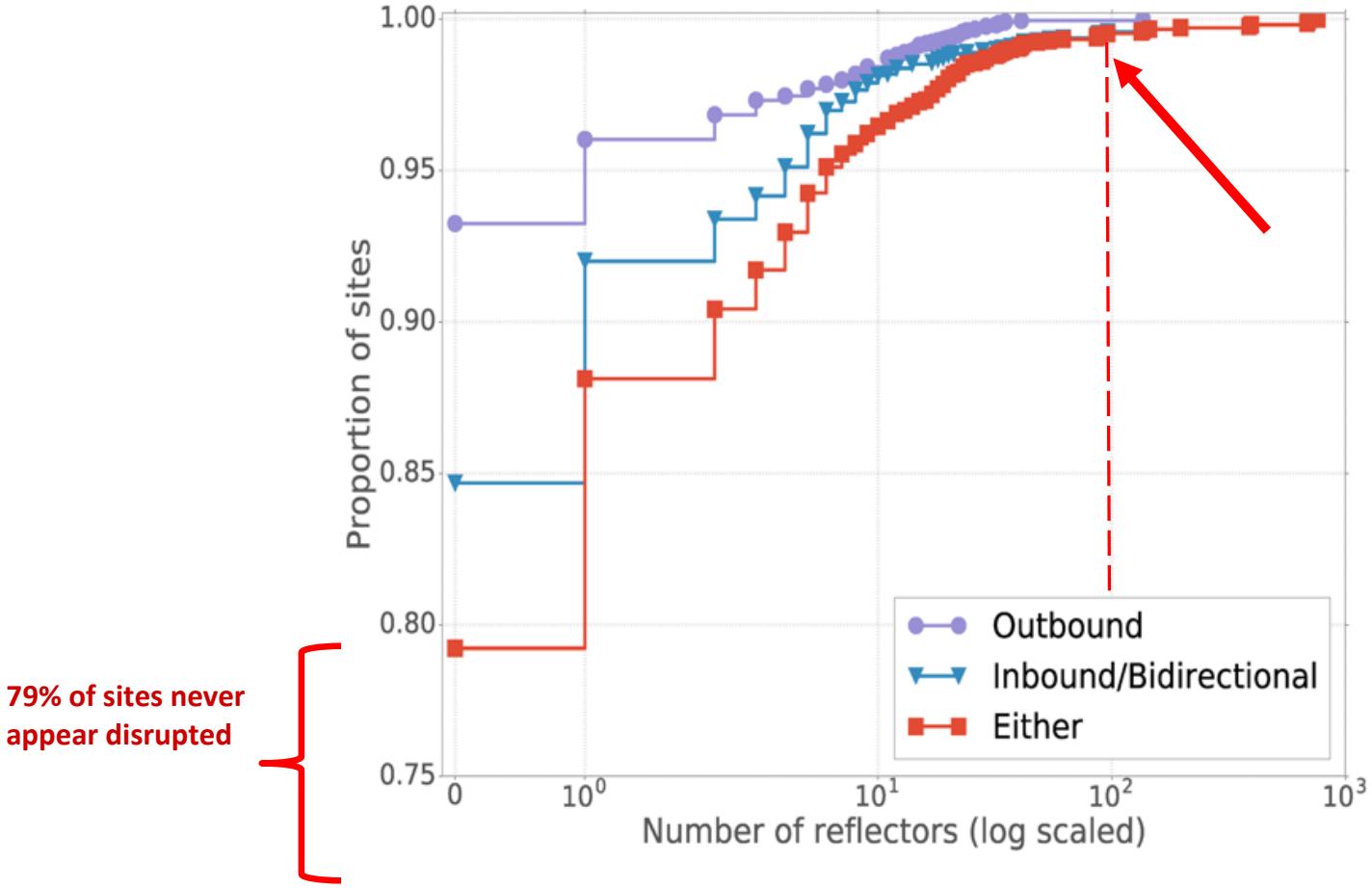
60% of Reflectors experience disruption



Validation Checks

Sites shouldn't be blocked across bulk of reflectors

- Over 99% of sites exhibit blocking by 100 reflectors (5%) or less

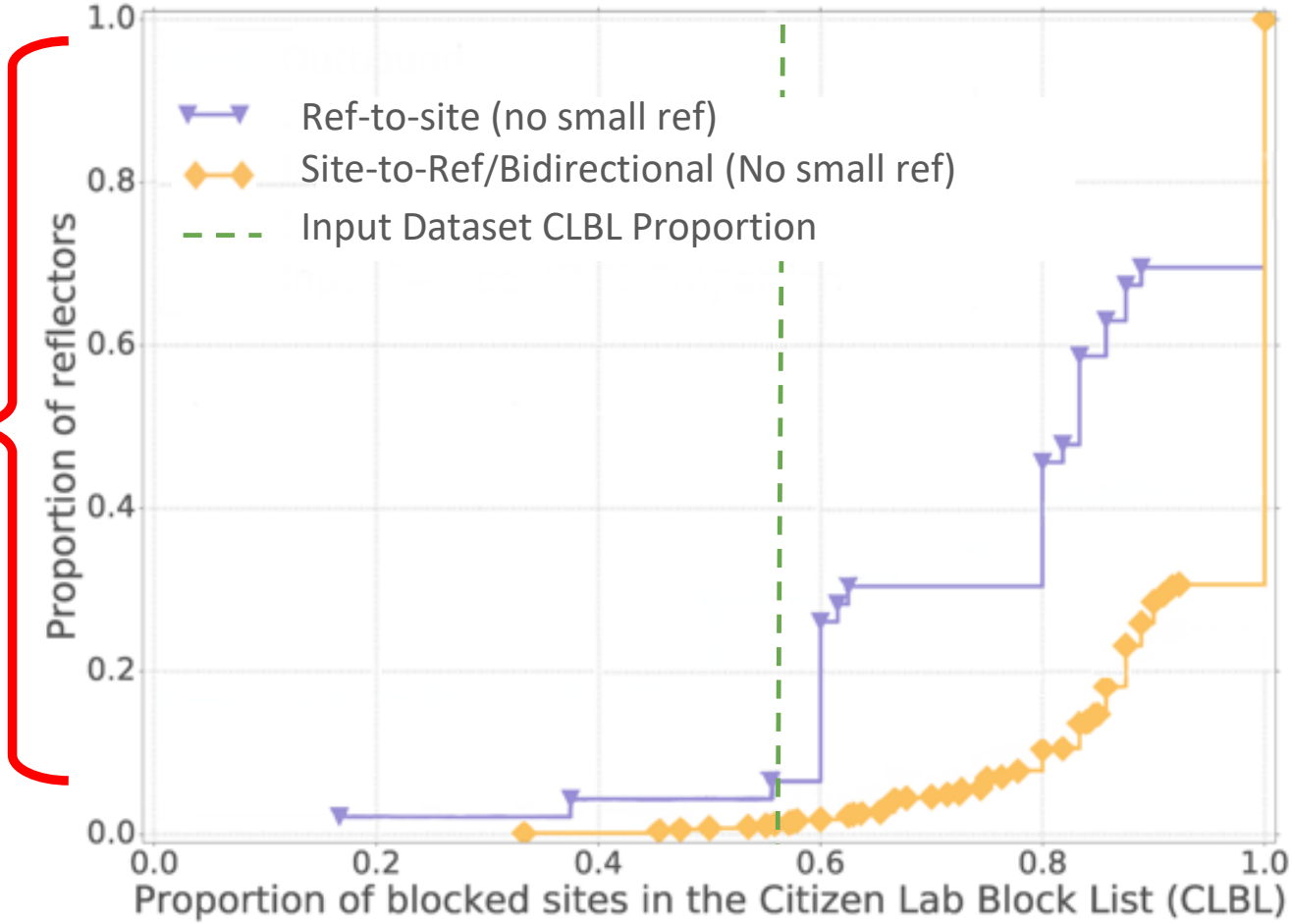


Validation Checks

There should be bias of blocking towards sensitive sites (CLBL)

- For 99% of reflectors, more than 56.7% of Site-to-Ref is towards CLBL

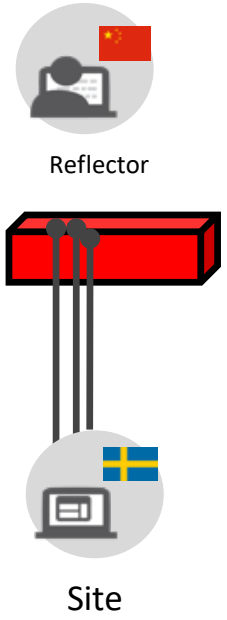
95% of reflectors, more than 56.7% of Site-to-Ref is towards CLBL



Results

Site-to-Reflector blocking

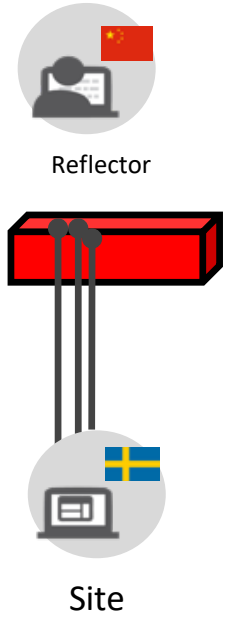
No.	Site	% Refs	% Cnt.	Class
1.	hrcr.org	41.7	83.0	Human Rights
2.	alstrangers.[LJ].com	37.9	78.8	Militants
3.	varlamov.ru	37.7	78.0	Foreign relations
	nordrus-norna.[LJ].com			Hate speech
4.	www.stratcom.mil	37.5	78.6	Foreign relations
5.	www.demonoid.me	21.7	58.5	P2P file sharing
6.	amateurpages.com	21.2	57.9	Adult contents
	voice.yahoo.jajah.com			Voice over IP
	amtrak.com			ALEXA



Results

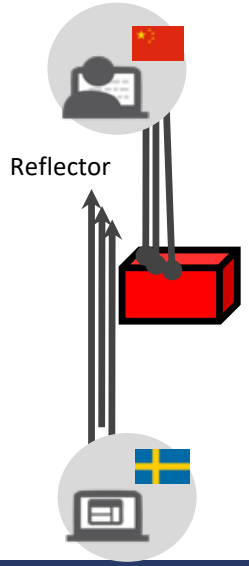
Site-to-Reflector blocking

No.	Site	% Refs	% Cnt.	Class
1.	hrcr.org	41.7	83.0	Human Rights
2.	alstrangers.[LJ].com	37.9	78.8	Militants
3.	varlamov.ru	37.7	78.0	Foreign relations
	nordrus-norna.[LJ].com			Hate speech
4.	www.stratcom.mil	37.5	78.6	Foreign relations
5.	www.demonoid.me	21.7	58.5	P2P file sharing
6.	amateurpages.com	21.2	57.9	Adult contents
	voice.yahoo.jajah.com			Voice over IP
	amtrak.com			ALEXA



Reflector-to-site blocking

No.	Site	% Refs	% Cnt.	Class
1.	nsa.gov	7.4	23.3	US Gov.
2.	scientology.org	2.2	6.9	Minority faiths
3.	goarch.org	1.9	4.4	Minority faiths
4.	yandex.ru	1.8	3.8	Freedom of Expression
5.	hushmail.com	1.8	4.4	Free email
6.	carnegieendowment.org	1.6	4.4	Political reforms



Questions

- ***Thoughts on Augur?***
- ***Has Augur enabled Internet-wide censorship detection forever?***
- ***How could censors evade Augur?***
- ***Are there kinds of censorship Augur can't detect?***
- ***Is Augur ethical?***

Cloud Computing

- Third-party cloud computing represents the promise of outsourced computation.
- It allows customers to purchase just the capacity they require, just when they require it.
- Cloud providers are able to maximize utilization of their capital investments by multiplexing many customer VMs across a shared physical infrastructure.
- It is a given that we need to be able to trust cloud providers to respect our private data…

… can we trust other users?

- We already know that 3rd Party cloud providers make their $$$ by multiplexing the machines in their monstrously large datacenters.
- Cloud computing creates threats of multi-tenancy, multiplexing the virtual machines of disjoint customers upon the same physical hardware.
- Could a customer be assigned to the same physical server as their adversary?
- Could the adversary exploit co-residency to extract confidential information?

Co-Residency Threat Model

- We trust the provider, its infrastructure and its employees.
- Adversaries are non-provider-affiliated malicious parties.
- Victims are running confidentiality-requiring services in the cloud.
- Everyone is a customer; both groups can all run and control many instances.
- We are not concerned with traditional threats and exploits here, even though they are alive and well in the cloud environment.
- 3rd Party Cloud Providers give attackers ***novel abilities*** , implicitly expanding the ***attack surface*** of the victim.
- Two kinds of attackers
  1. Casts a wide net in an attempt to attack somebody
  2. Focuses on attacking a particular victim service

Hey! You! Get Off of My Cloud!

1. Use Amazon EC2 as a case study.
   - U.S. Region
   - Linux Kernel
2. Achieve **PLACEMENT** of their malicious VM on the same physical machine as that of a target customer.
   - Determine where in the cloud an instance is likely to be located.
   - Determine if two instances are co-residents.
   - Intentionally launch an instance to achieve co-residence with another user.
3. Proceed to **EXTRACT** information and/or perpetrate all kinds of assorted nastiness.

[Ristenpart et al., CCS'09]

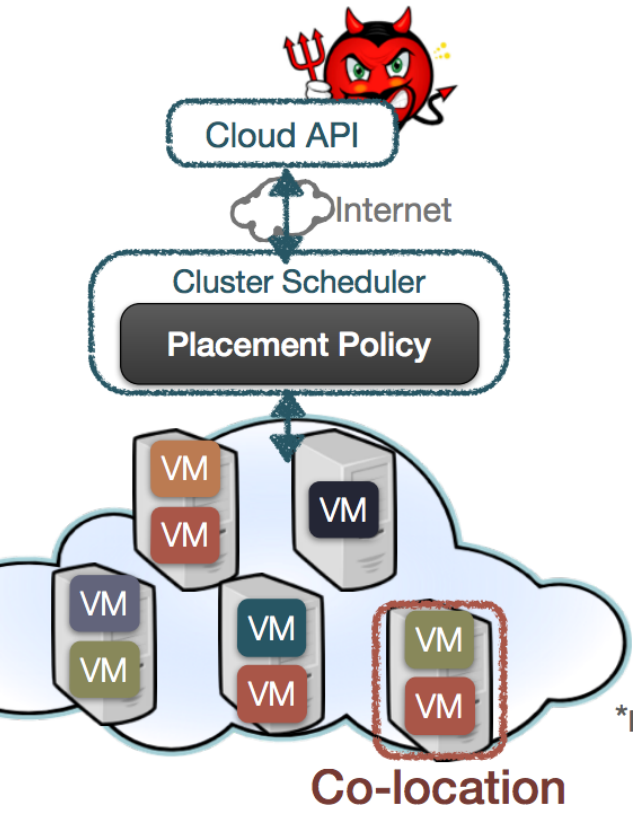
Hey! You! Get Off of My Cloud!

"Cloud Cartography"

- ***Hypothesis: different availability zones (and possibly instance types) are likely to correspond to different internal IP address ranges.***
- Since we already know that it's possible to infer the internal IP address of an instance associated with a public IP through the EC2's DNS service…
- If this hypothesis holds, an adversary can use a map of EC2 to determine the instance type and availability zone of their target, dramatically reducing the number of instances needed to achieve co-residence.

[Ristenpart et al., CCS'09]

Hey! You! Get Off of My Cloud!

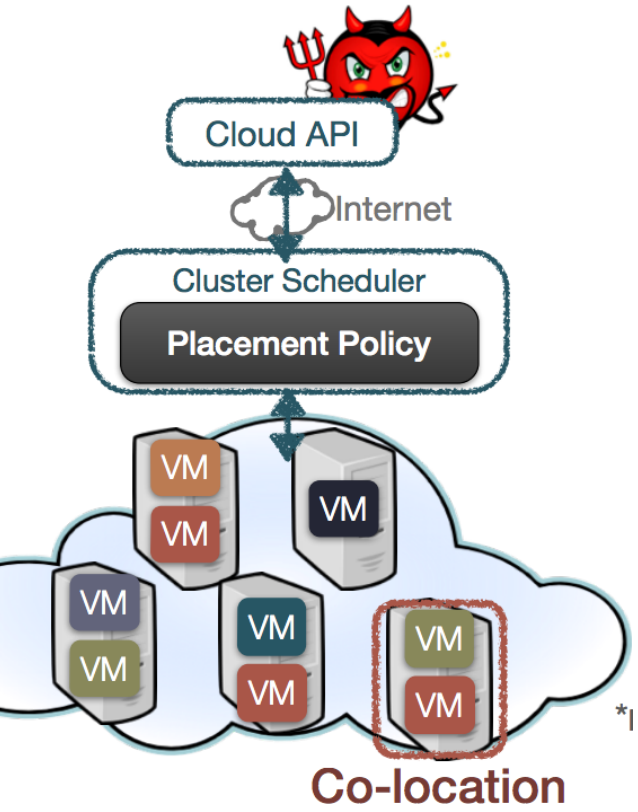


Limitations of prior work:

- Focused exclusively on Amazon EC2
- New countermeasures, including patching the side-channels originally used to detect co-residency.
- Increased scale of cloud — makes *cloud cartography*-based approach ineffective because the map got too big.

[Ristenpart et al., CCS'09]

[Varadarajan et al., Security'15]



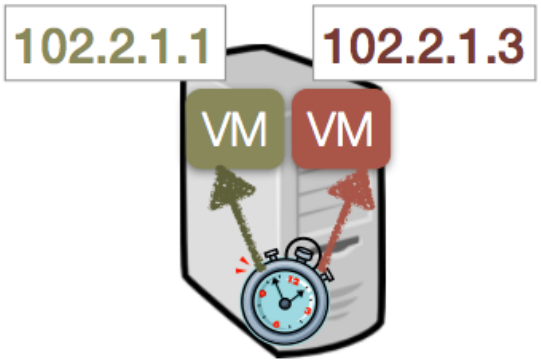
Novel contributions of this study:

- Performs black box testing of cloud scheduler to infer placement strategy
  - Enables intelligent attack strategy
- Presents new methods for co-residency detection that are more difficult to patch

Co-Residency Detection

1. Read shared state on two VMs

e.g., private IP addresses, shared TSC counters.



- Worked well in early days; was as simple as checking dom0 IP address.
- Less common now; many shared state channels have been patched.

2. Correlate performance of shared resources

e.g., network round-trip times, cache-based covert-channels.

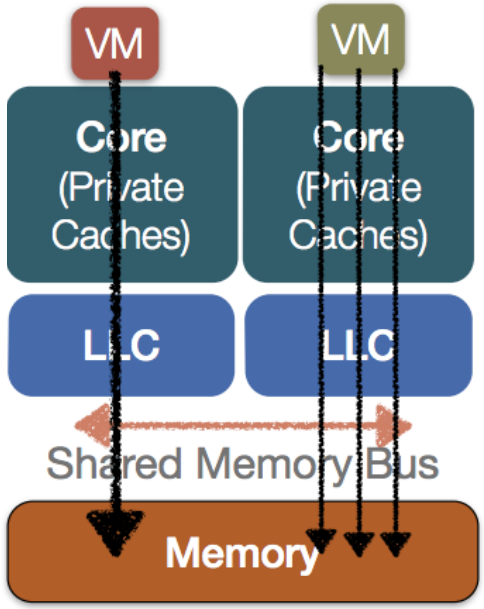
- Early effective techniques used L2 cache, i.e., “prime and probe.”
- Sharing is intrinsic to cloud computing; difficult to fix

Cooperative Detection

- In one class of co-residency detection schemes, VMs can collude to infer their placement.
  - (Works well for measurement studies but not attacks)
- Wu et al.'s memory locking covert channel does the trick!

Sender:

```
// allocate memory multiples of 64 bits
char_ptr = allocate_memory((N+1)*8)
//move half word up
unaligned_addr = char_ptr + 2
loop forever:
  loop i from (1..N):
    atomic_op(unaligned_addr + i, some_value)
  end loop
end loop
```



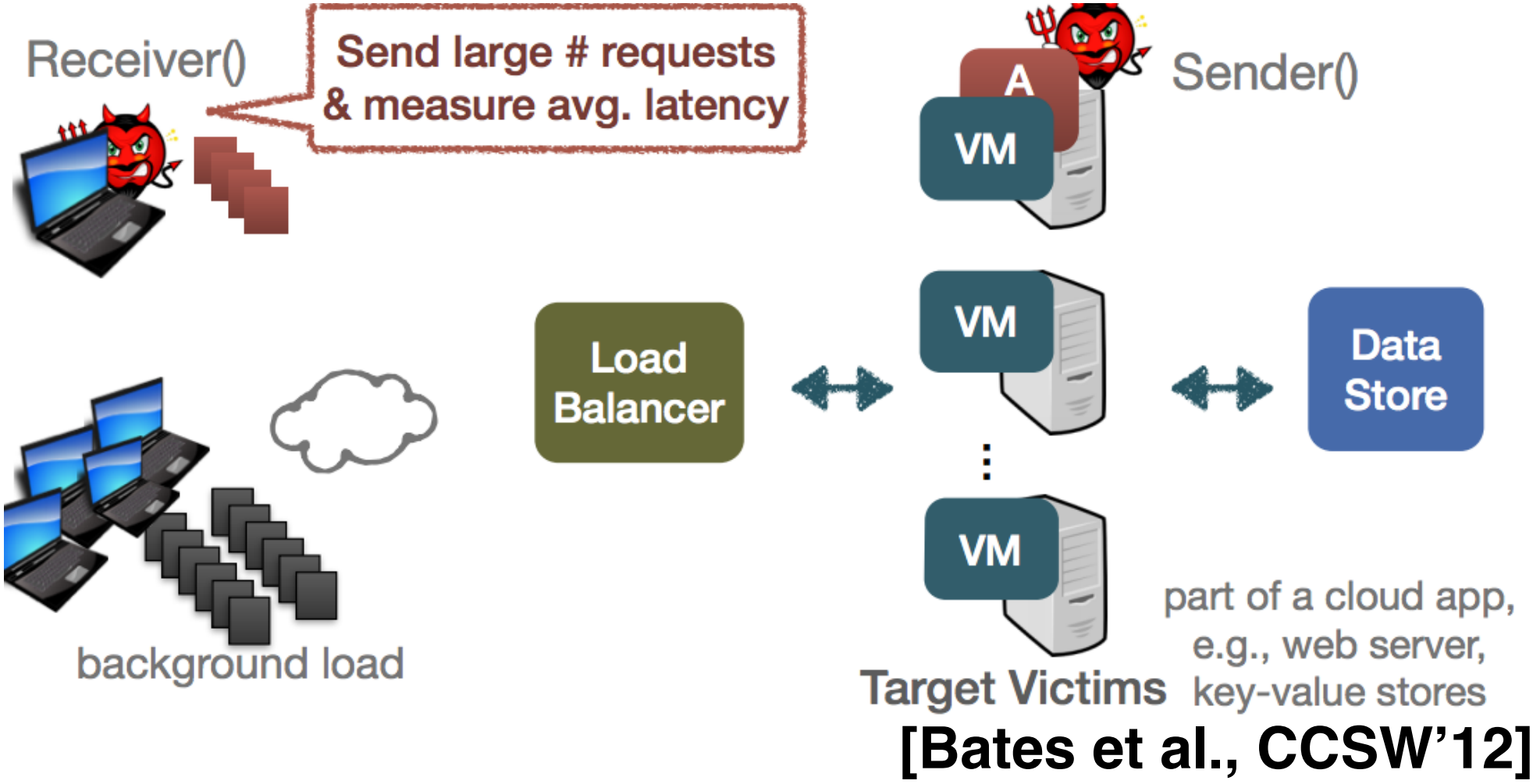
Receiver:

```
Observe() {
  s = start_time
  repeat N
    mem_access()
  done
  e = end_time
  bw = N/(e-s)
}
```

[Wu et al., Security'12]

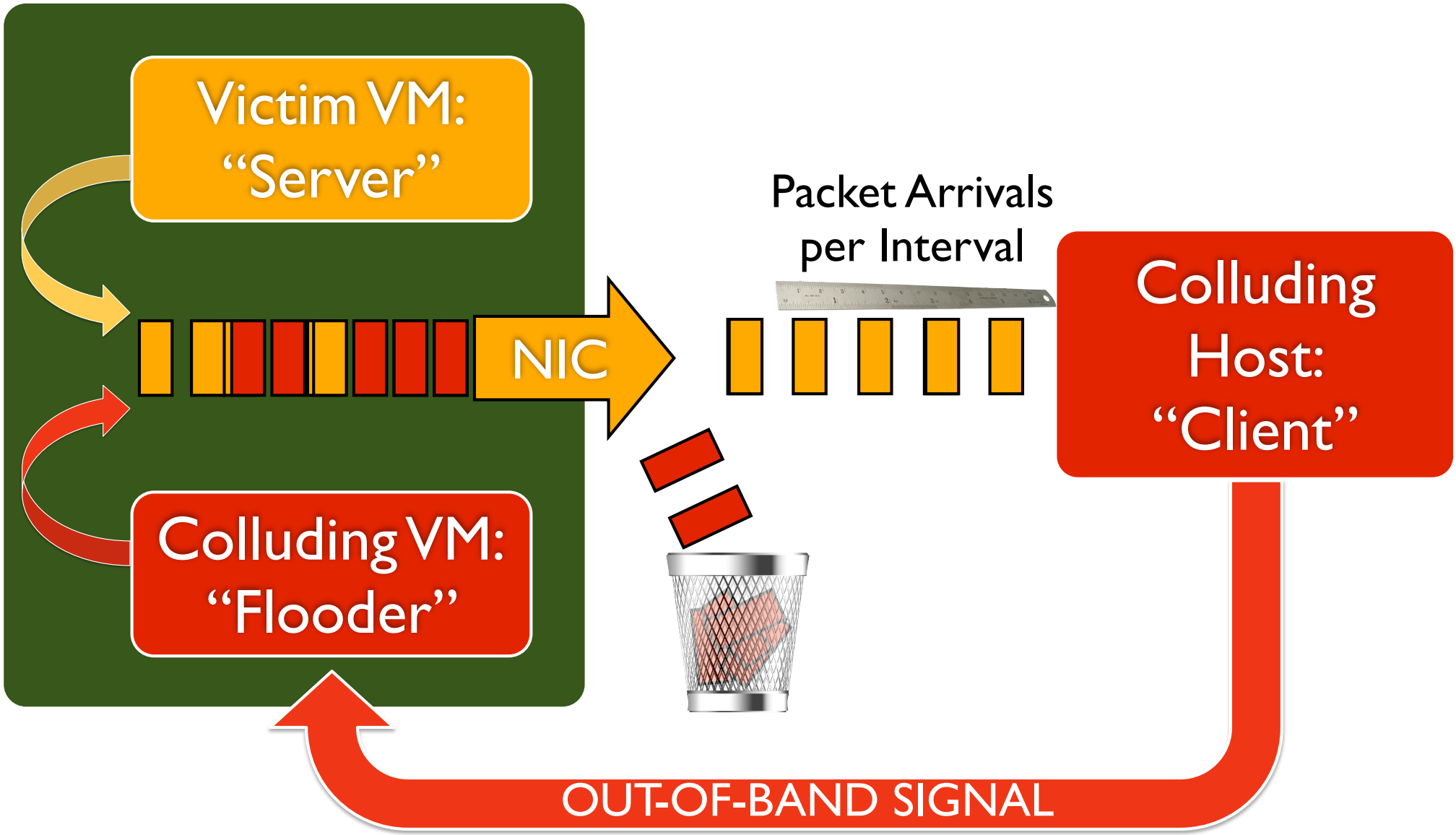
Cooperative Detection

- ***What about on an un-cooperative (victim) VM?***
- One possibility — embed a beacon into network activity!



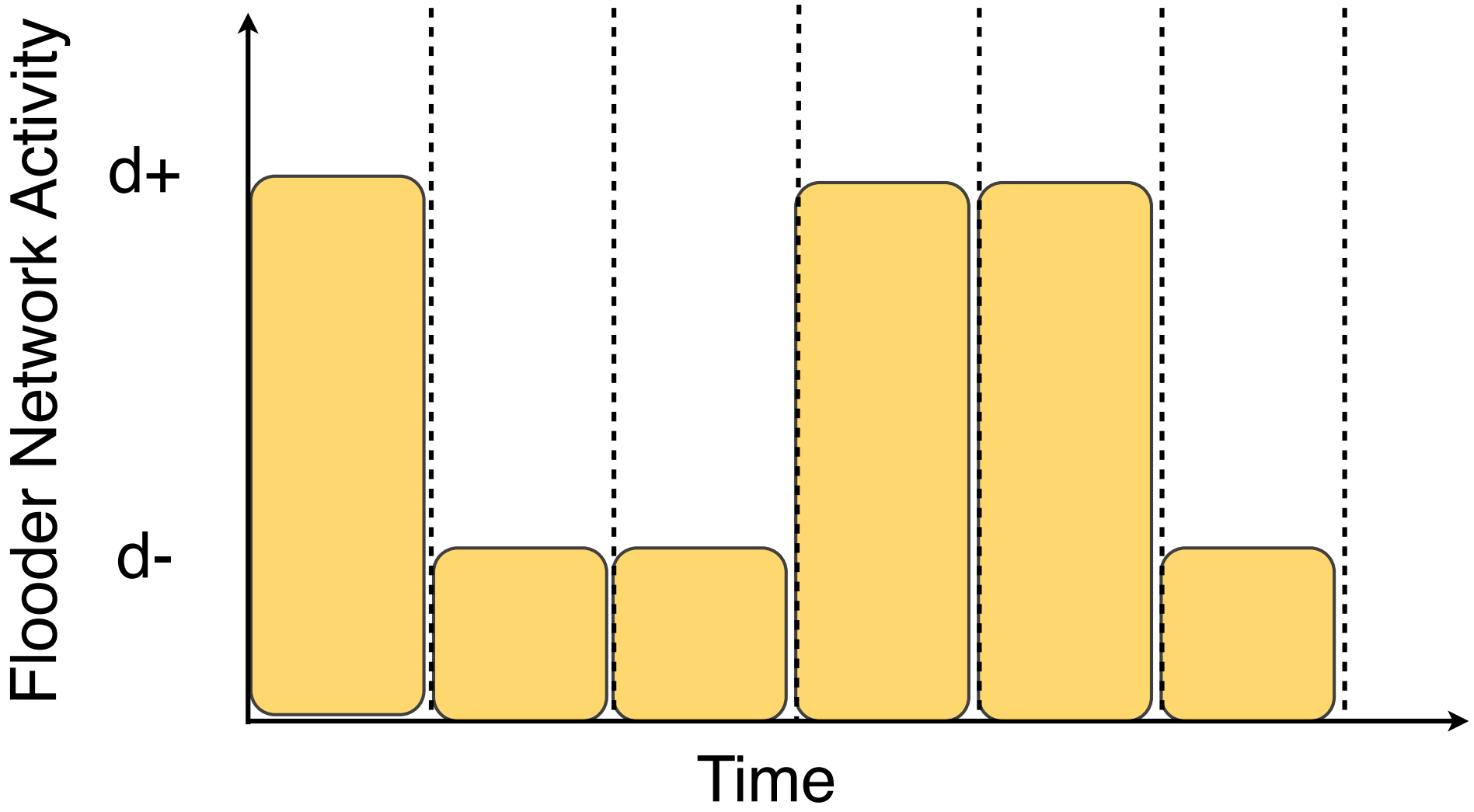
[Bates et al., CCSW'12]

Co-Resident Watermarking



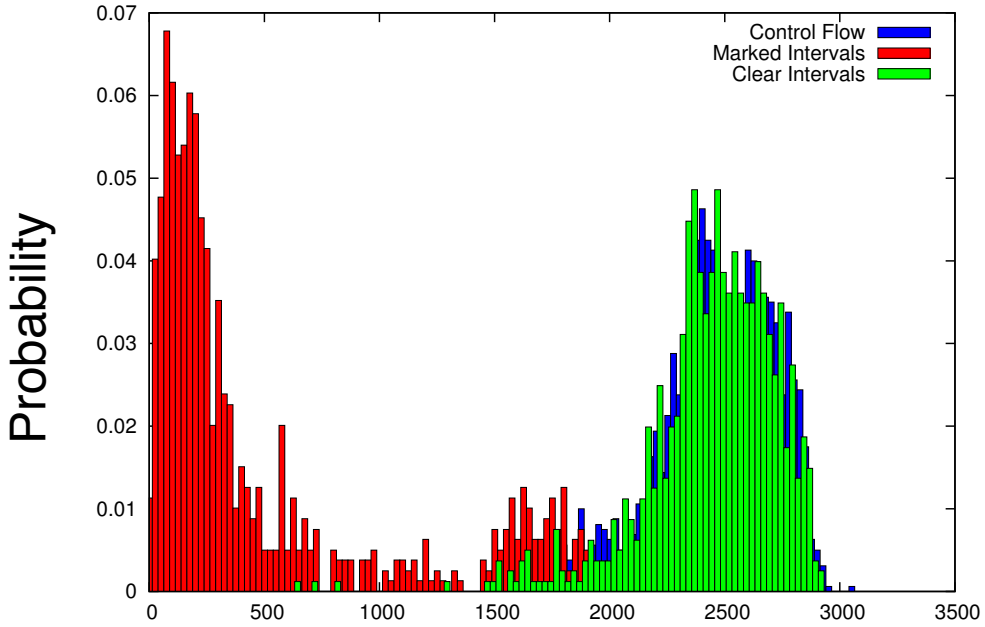
[Bates et al., CCSW'12]

Co-Resident Watermarking

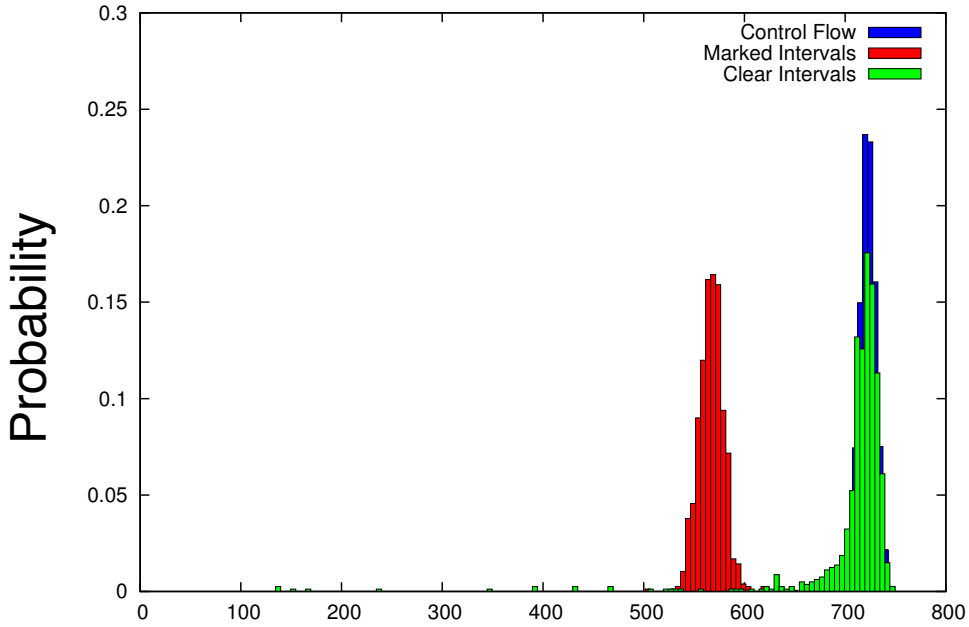


[Bates et al., CCSW'12]

Co-Resident Watermarking



ACISS (KVM)

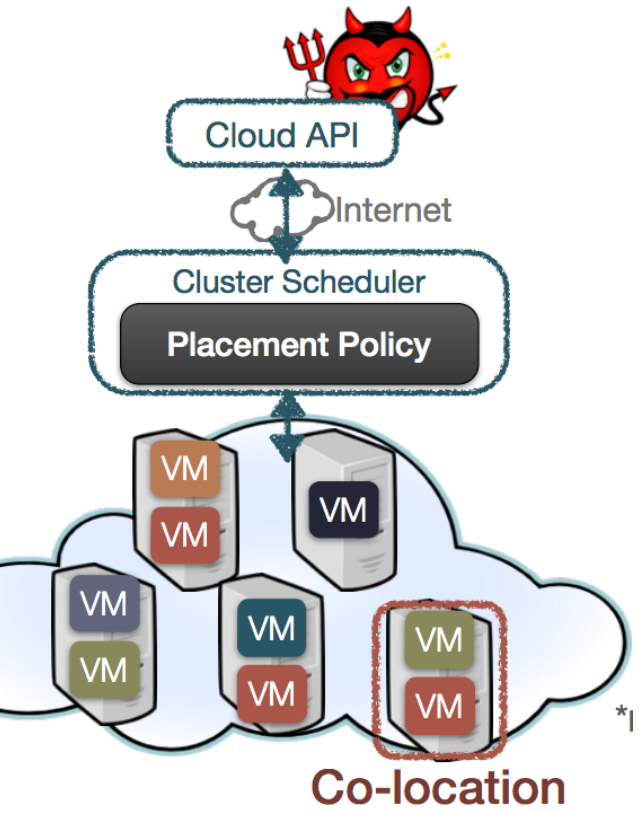


Futuregrid (Xen)

- *Co-resident watermarking* is a viable attack in production cloud environments.

[Bates et al., CCSW'12]

How hard *should* it be to co-locate?



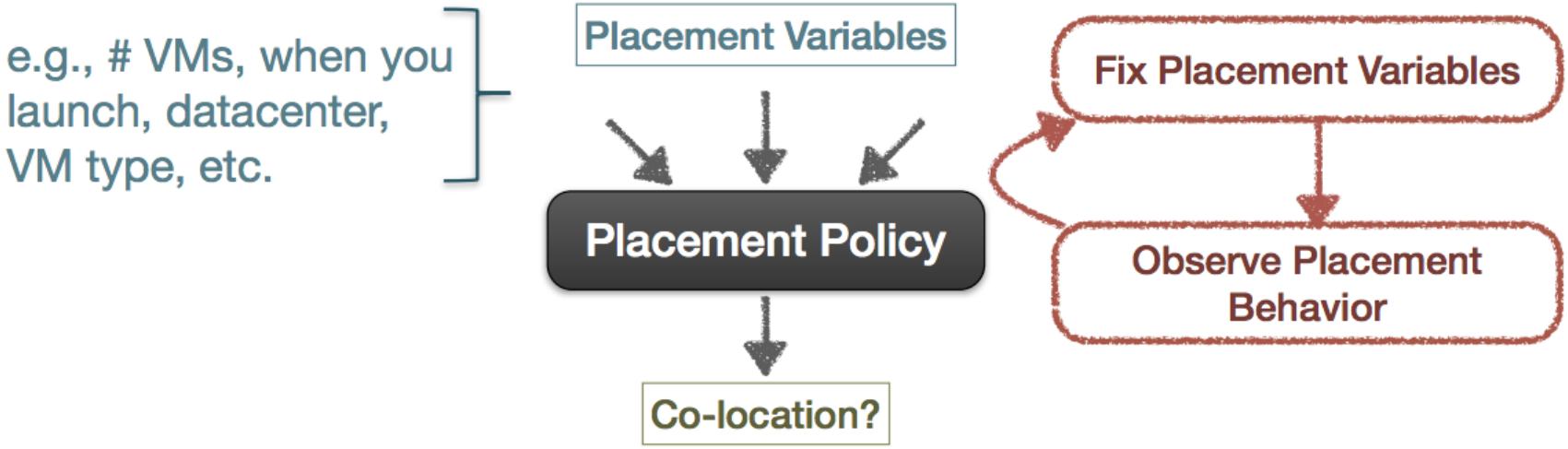
If a truly random placement policy was used…

- N = 50,000 machines
- v victim VMs and a attacker VMs
- Probability of Collision:

$$P_c = 1 - (1 - \frac{v}{N})^a$$

v	a = ln(1– P_c)/ln(1–v/N); P_c= 0.5
10	3466
20	1733
30	1155

Placement Study



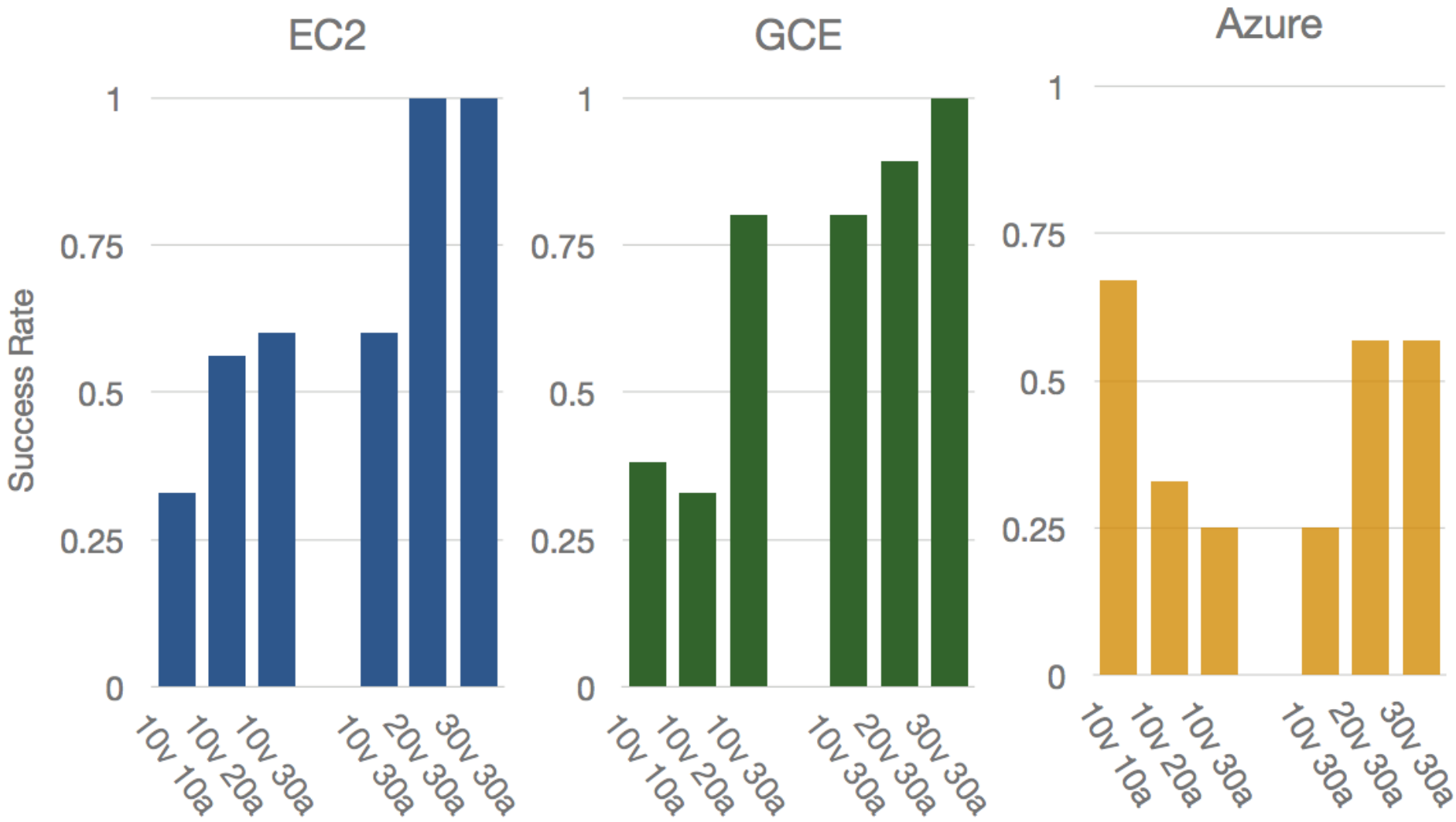
- 6 placement variables: # victim & attacker VMs, delay b/w launches, time of day, day of week, datacenter, cloud provider Small instance type
- 9 samples per strategy with 3 runs per time of day and 2 days of week (weekday/weekend).

Google Compute Engine

amazon web services EC2

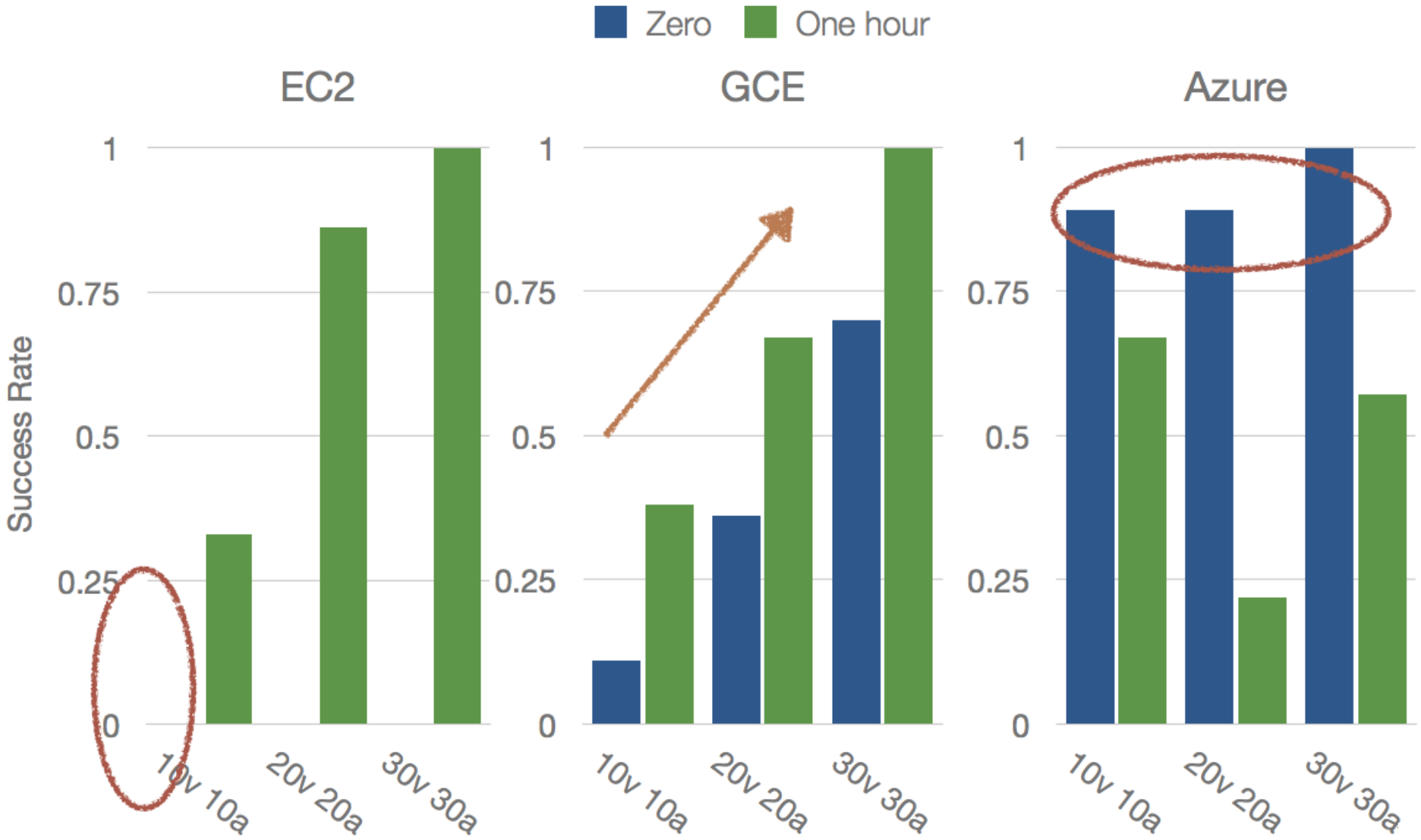
Microsoft Azure

Variable # of VMs



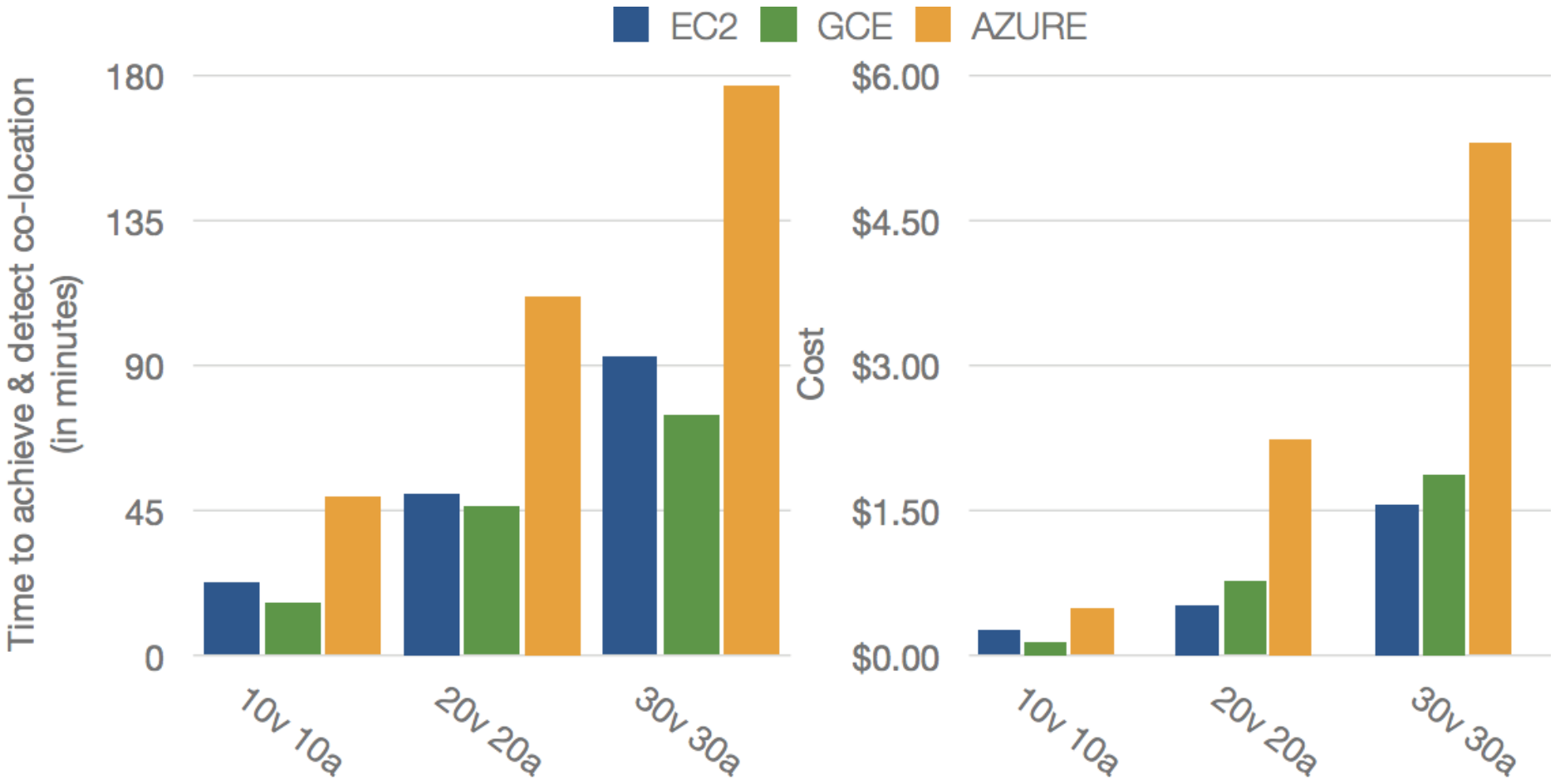
Co-location is possible with as low as 10 VMs and always achieve co-location with 30 VMs

Variable Delay between launches



Different clouds have wildly different temporal placement strategies

Attack Cost



Successful co-location as affordable as 14 cents.

Measurement: Looking Forward

- Where to look for literature: "Big 4" security conferences (IEEE S&P a.k.a. Oakland, USENIX Security, CCS, NDSS) and also major network conferences (e.g., IMC, SIGCOMM).
- Big Idea of measurement-based methodologies: Help us to better understand the state of security in the real world.
- Hot Topics in Measurement (not exhaustive):
  - Internet Ecosystem (e.g., TLS, HTTPS adoption, CDNs, DNS, Advertising)
  - Cloud Computing (e.g., side channels)
  - Software Development (e.g., longitudinal measurement of bugs in open source projects)
  - Malware, Spam, Botnets