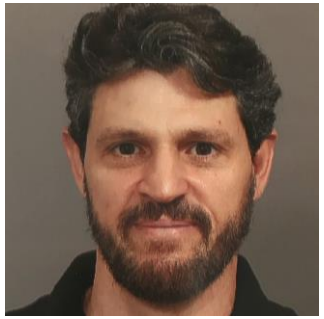
usenix THE ADVANCED COMPUTING SYSTEMS ASSOCIATION Security Symposium 2022

Anycast Agility: Network Playbooks to Fight DDoS

ASM Rizvi*
USC/ISI



Leandro Bertholdo*
University of Twente

Joao M Ceron
SIDN Labs

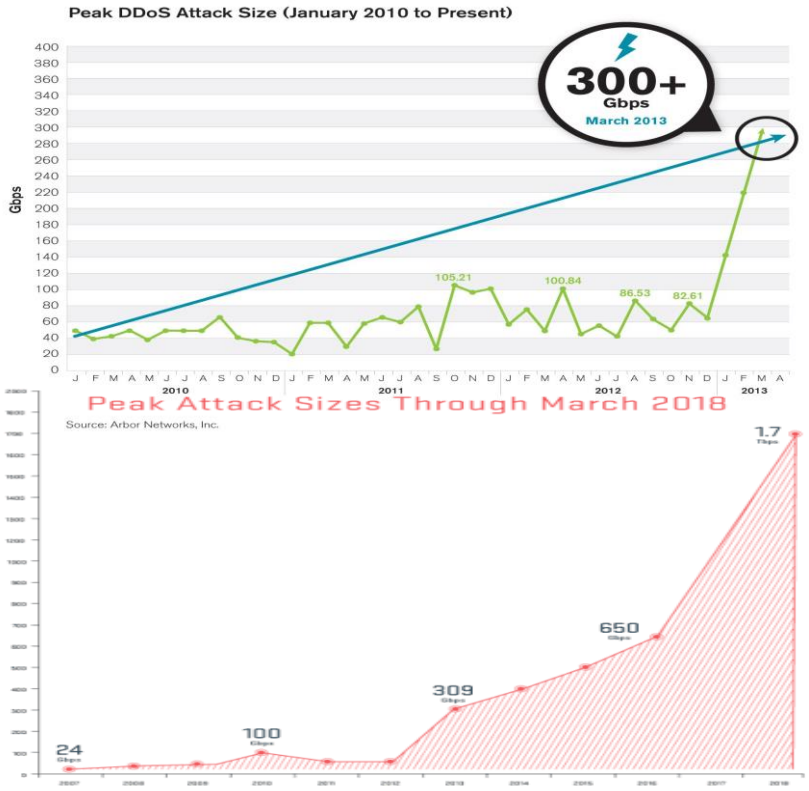
John Heidemann
USC/ISI

12 August, 2022

* Indicates equal contribution

USC Viterbi School of Engineering Information Sciences Institute ant.isi.edu UNIVERSITY OF TWENTE. SIDN

Distributed Denial-of-Service (DDoS) is Bad... and Getting Worse



AWS said it mitigated a 2.3 Tbps DDoS attack, the largest ever

The previous record for the largest DDoS attack ever recorded was of 1.7 Tbps, recorded in March 2018.

- DDoS is big
  - Automated botnets
- DDoS is getting bigger
  - IoT devices
  - Github attack exceeds 1.35Tbps rate in March 2018
  - Amazon gets 2.3 Tbps DDoS in February 2020
- Cheap: booters offer DDoS-as-a-service
  - starting at $1/attack [Santanna et al, 2015]

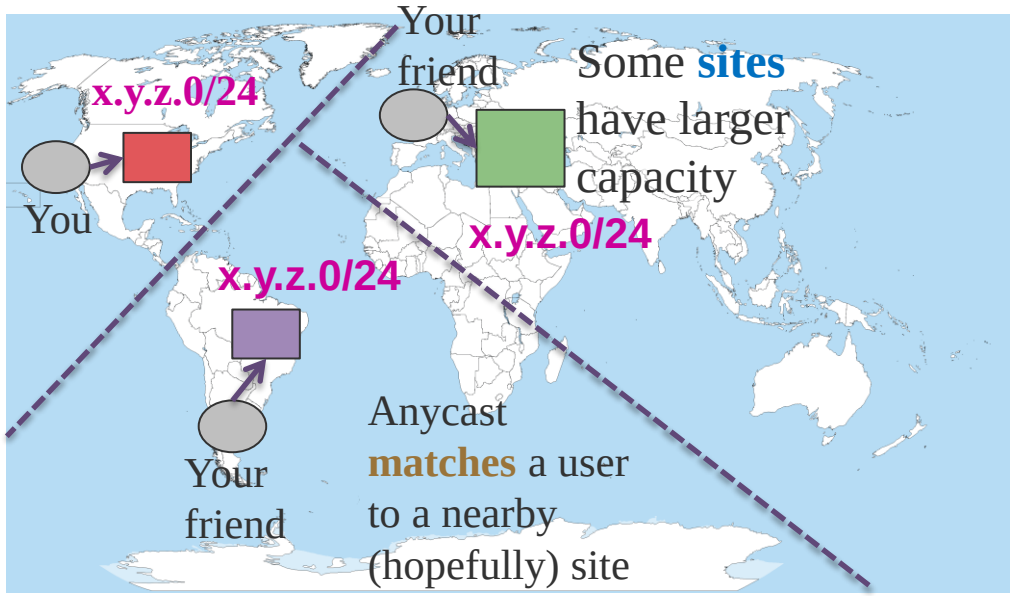
Understanding Anycast as a DDoS Defense

- Today, many use third-party scrubbing
  - Good, but can be expensive
  - And costs some control (privacy and sovereignty)

- We explore **anycast** as a DDoS defense
  - Multiple sites for capacity
  - Routing spreads traffic during attack (traffic engineering)
  - A privacy-preserving alternative to third parties
  - …and what they do themselves (but have not described)

Background: What is Anycast?

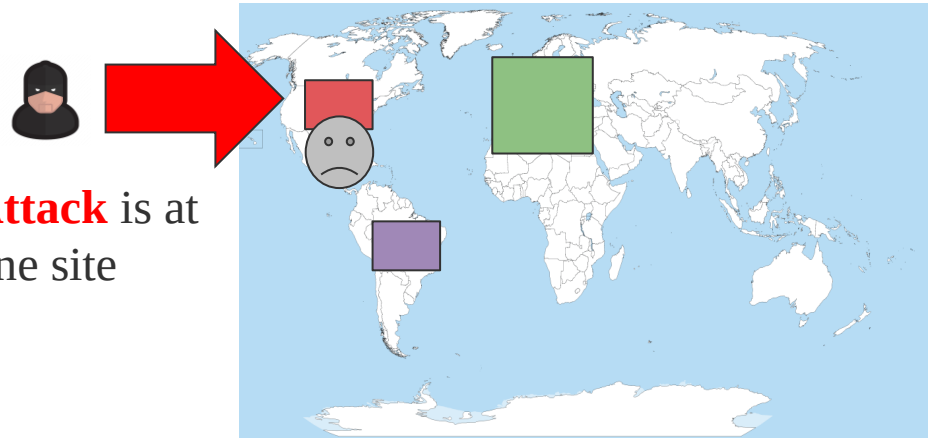
- Replication of the same service
  - Multiple physical sites
  - BGP matches users to sites
  - Spread load over sites
- Anycast is widely used
  - Akamai, Cloudflare, Google, Azure, Facebook, Amazon...
  - Most large DNS operators (all root operators, many ccTLDs, etc.)
  - BGP is the way to control traffic in anycast.
- But how does BGP distribute users?
  - Normally
  - And especially when under attack (our goal)



Anycast **divides** the world into catchments.

A DDoS Attack

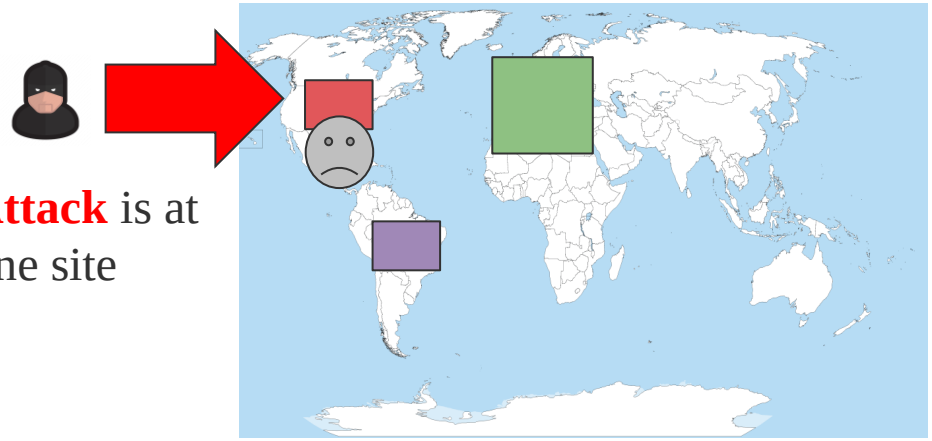
One site is **overwhelmed**



Attack is at one site

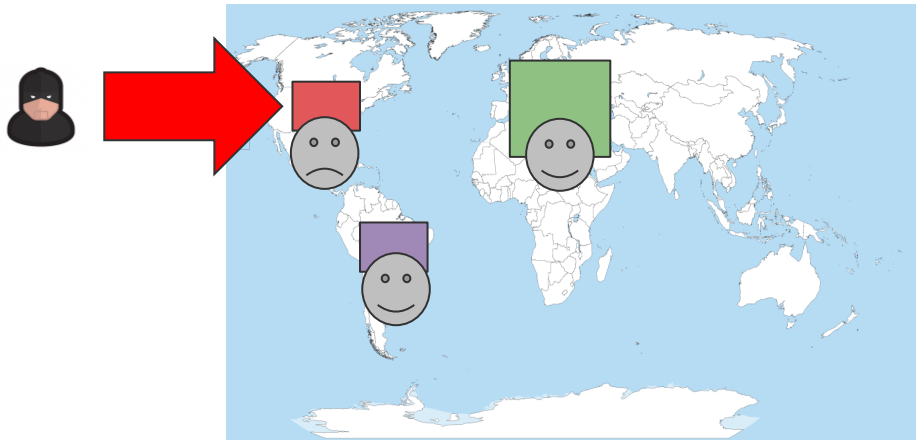
Defense 1: Absorb at One Site

One site is **overwhelmed**



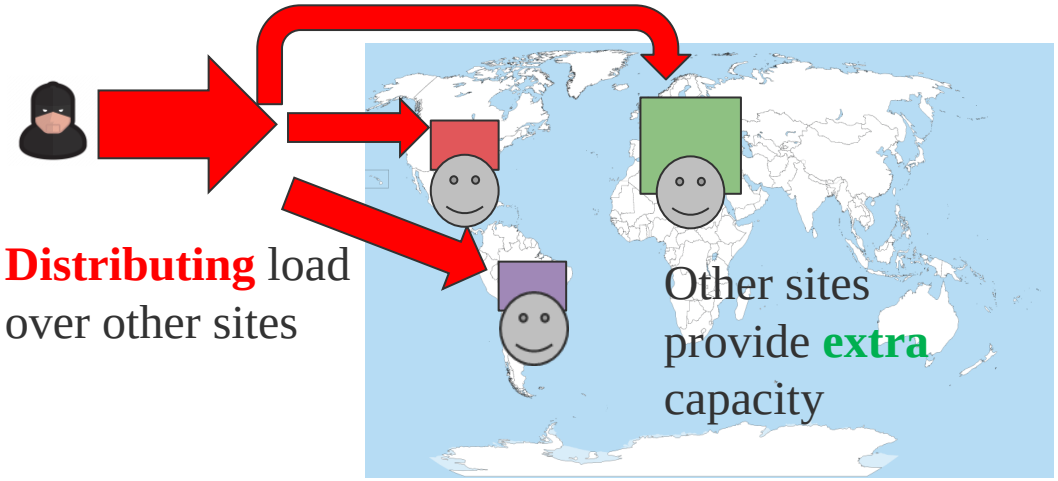
Attack is at one site

One site is hurt, but others are OK!



Defense 2 (Spread) and 3 (Shift)

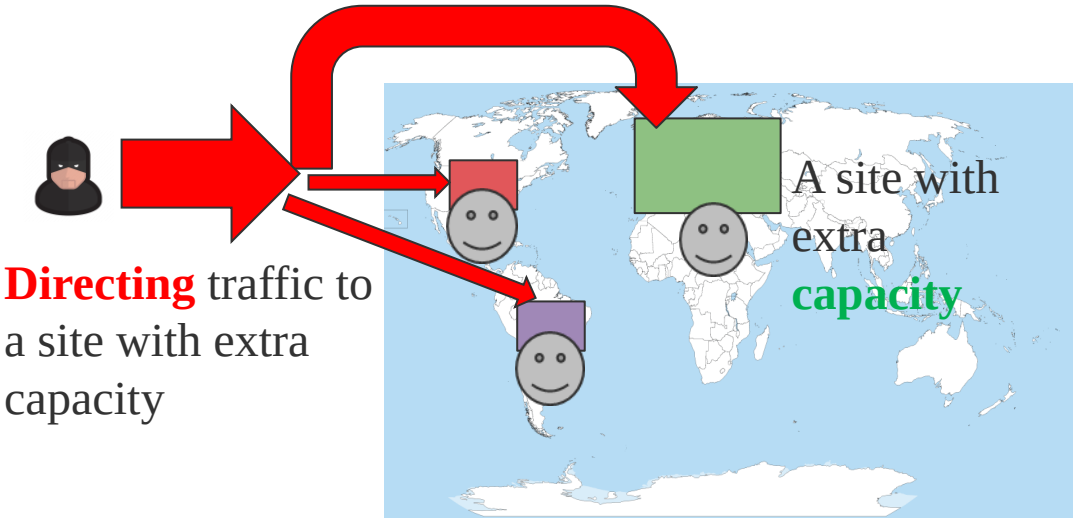
When other sites provide extra capacity



Distributing load over other sites

Other sites provide **extra** capacity

When a specific site has extra capacity



Directing traffic to a site with extra capacity

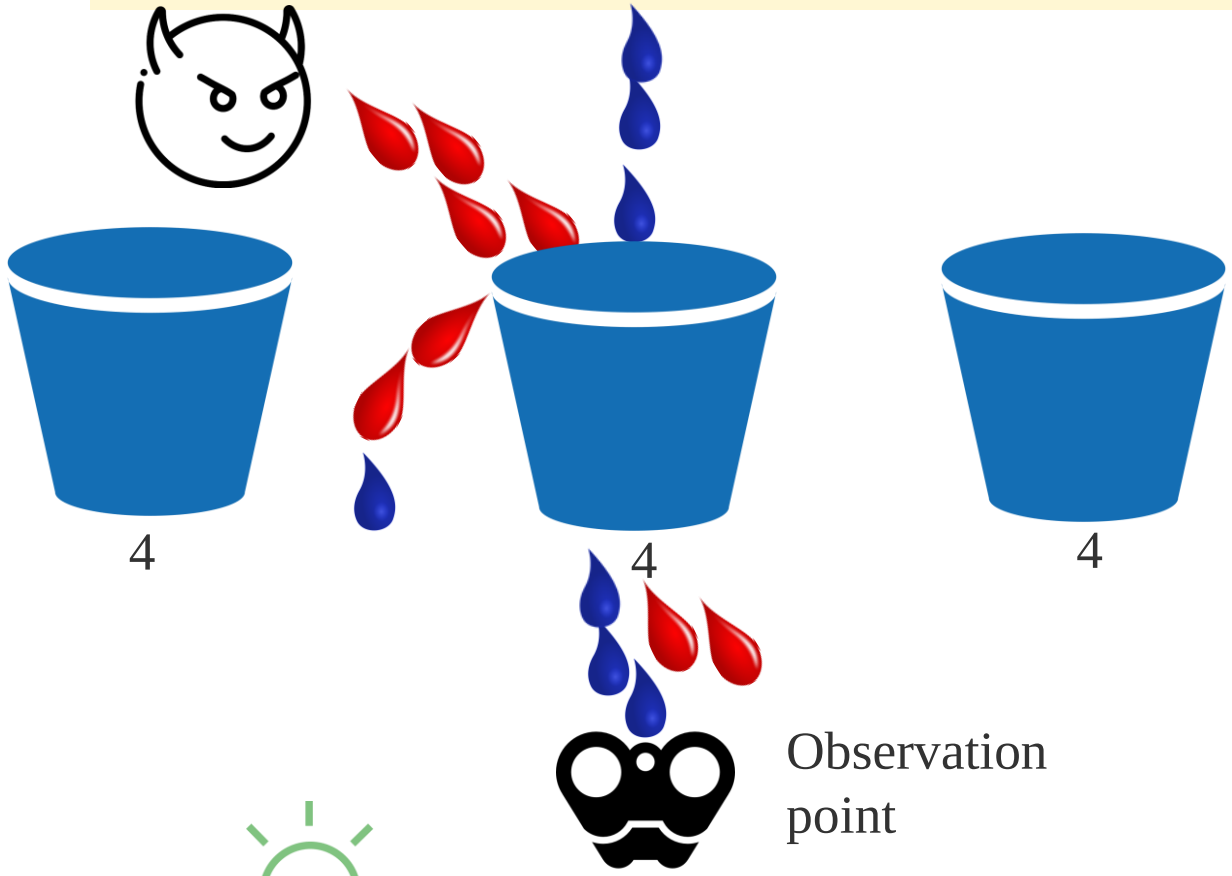
A site with extra **capacity**

How does the defender absorb, spread, or shift?

A Demo of The Defense System with Spreading

https://vimeo.com/735280790

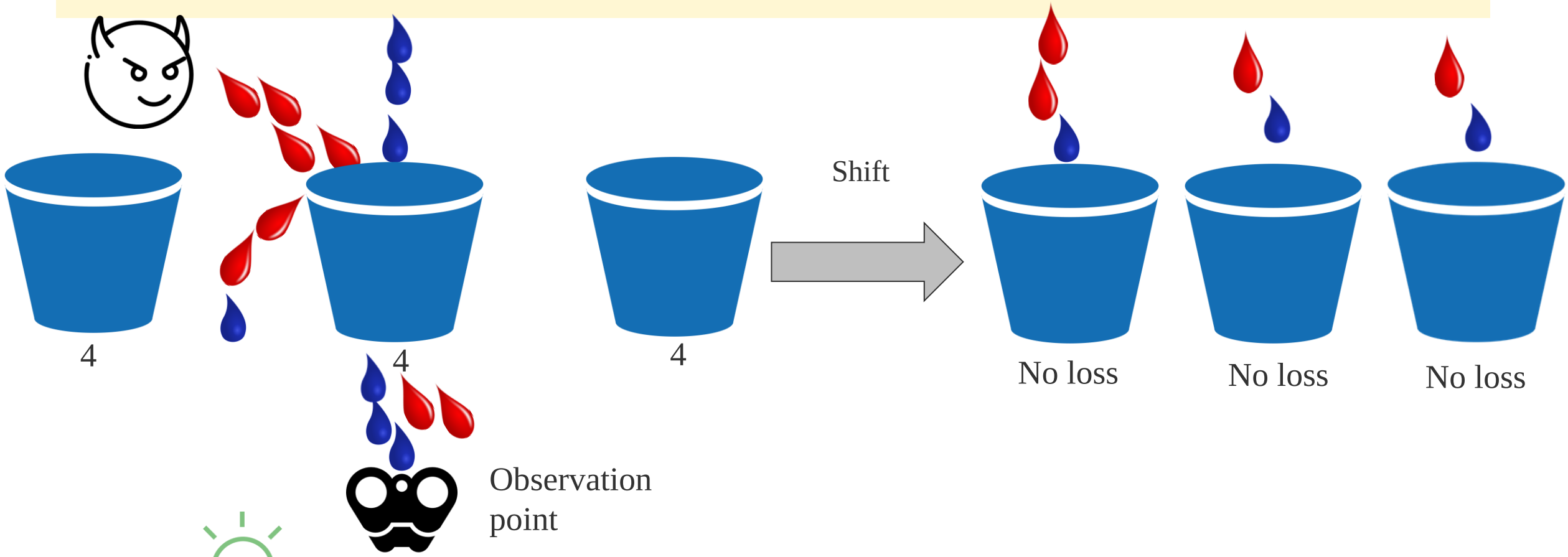
Challenge 1: Unknown Offered Load



- What you directly observe is “at full capacity with 50% attack traffic”
- But the truth is “at 175% capacity with 100% attack traffic, and 75% legitimate”
- Direct observation under-estimates

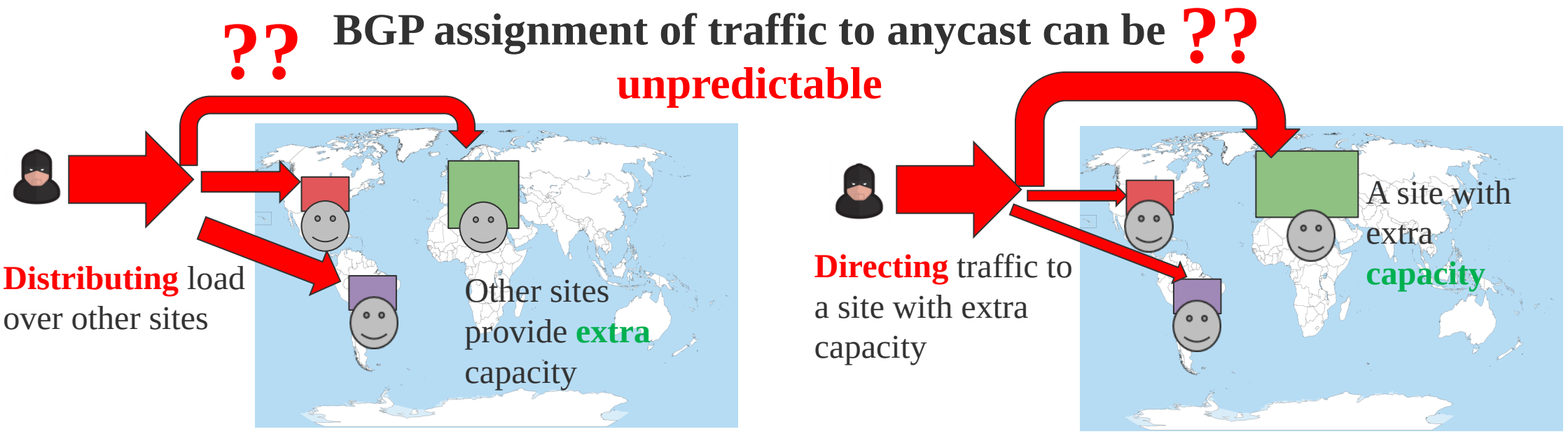
Our **contribution**: proposing a way to estimate the true offered load

Challenge 2: Controlled Traffic Engineering



Our **contribution**: we help the operator to get the **right** shift

Challenge 3: Selecting from Different Defenses



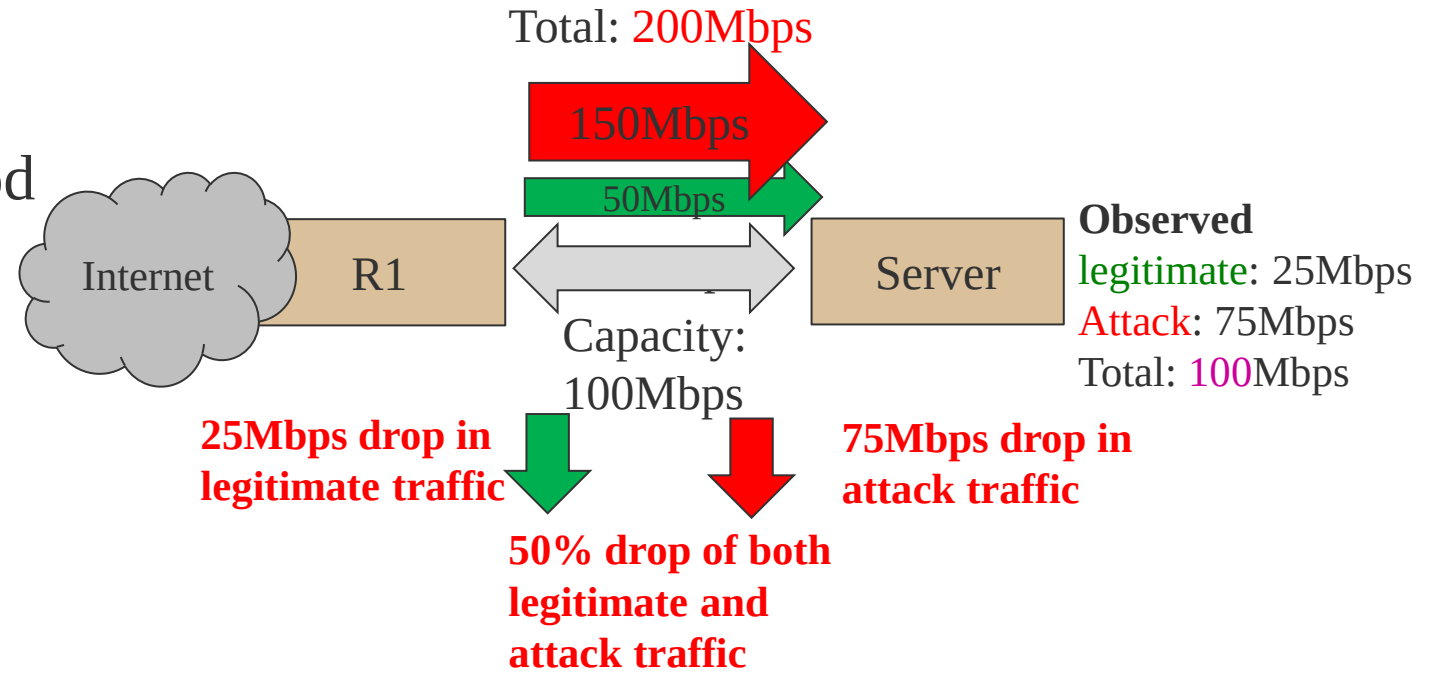
Our **contribution**: how to build a **BGP playbook** to predict anycast ahead of time

Our Contribution

- New approach to estimate the offered load (challenge 1)
  - Allows us to plan a defense
- Define a BGP playbook (challenge 3)
  - Allows us to execute the correct defense
- Show a BGP playbook works in a real DDoS event (challenge 2)
  - Effectiveness of our approach

Methodology: Estimating Offered Load

- Problem: upstream loss is invisible
- Insight:
  - Sites have predictable known good traffic
  - Infer attack size by change in this traffic
- Details in paper section 3.3



Methodology: Understanding Traffic Engineering (TE)

- We used three TE techniques
- Each TE method has tradeoffs (details in section 6)
  - Path prepending
    - Available in all sites but no granular control
  - Community strings
    - Not available in all sites but provide granular control
  - Path poisoning
    - Filtered when poisoning Tier-1 ASes, and provide limited control

Methodology: Building a Playbook

- Combining all the TE techniques
  - TE methods: path prepending, community strings, and path poisoning
- Mapping the impacts
  - Utilized Verfploeter to find out the traffic distribution after deploying TE methods
  - Details in section 3.2, 3.4, and 5

Routing Policy	Traffic to Site (%) AMS	BOS	CNF
(a) Route-server	15	35	55
(b) All-IXP-Peers/Poison transits	15	35	45
(c) 2xPrepend AMS	25	35	45
(d) 1xPrepend AMS	35	25	35
(e) -1xPrepend BOS	45	45	15
(f) -1xPrepend CNF	45	5	45
(g) Transit-1	45	25	35
(h) Transit-2	55	15	25
(i) Poison Tier-1/Transit-2	35	25	35
(j) Poison Transit-1	55	25	25
(k) Baseline	65	15	15
(l) 1,2xPrepend BOS	65	5	25
(m) 1,2,3xPrepend CNF	75	15	5
(n) -1,-2,-3xPrepend AMS	85	5	5

A sample playbook

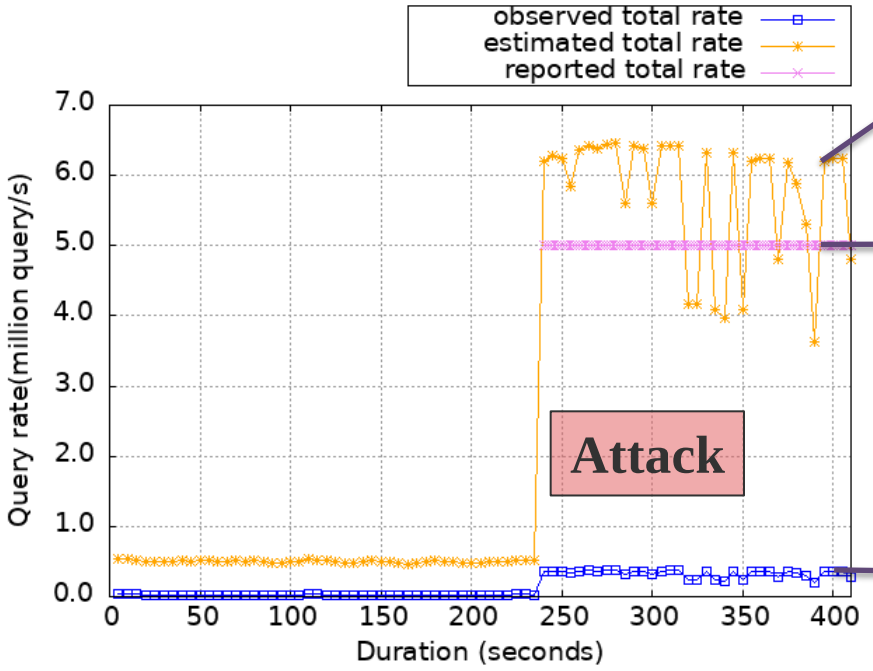
Announcing only to **Transit-2:**
AMS: 55% traffic
BOS: 15% traffic
CNF: 25% traffic

Validation and Results

- Validation: estimating offered load
  - Can we estimate correctly?: section 4.1
- Result: control and coverage of TE methods
  - Path prepending, community strings, and path poisoning (section 6.1, 6.2, 6.3)
  - How can we construct a playbook? (section 6.4)
- Result: deployment considerations
  - What is the impact of anycast configurations and number of sites? (section 7.1, 7.2)
  - How is the stability of a playbook? (section 7.3)
- Result: defense at work
  - Can we defend real-world attack events using the playbook? (section 8)

Offered Load Estimates are Accurate

- Question: **does estimation work?**
- Experiment:
  - Replayed captured packet trace collected at B root
  - Measured observed traffic rate and access fraction to estimate
  - Compared the estimation with the reported rate
- Answer: **yes**
- Attack is root DNS attack from 2015-11-30 with data from B-root



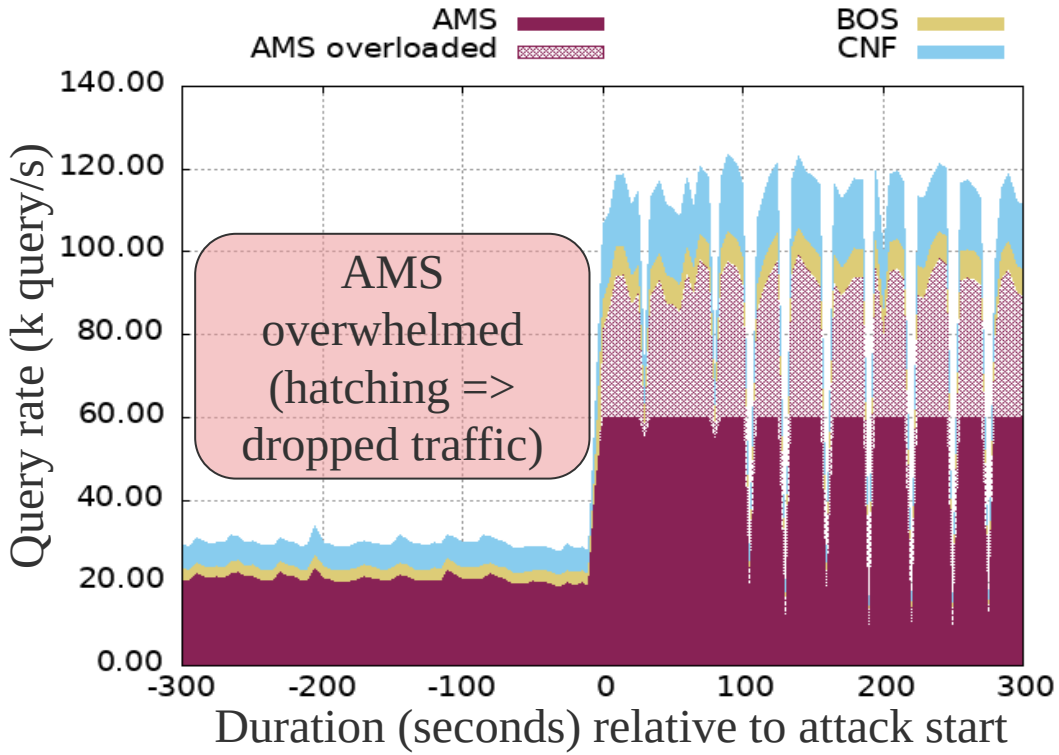
Our estimate: ~5 M query/s

True Offered load 5 M query/s

Observation: 0.35 M query/s (very low) because loss in upstream

Using a Playbook to Defend

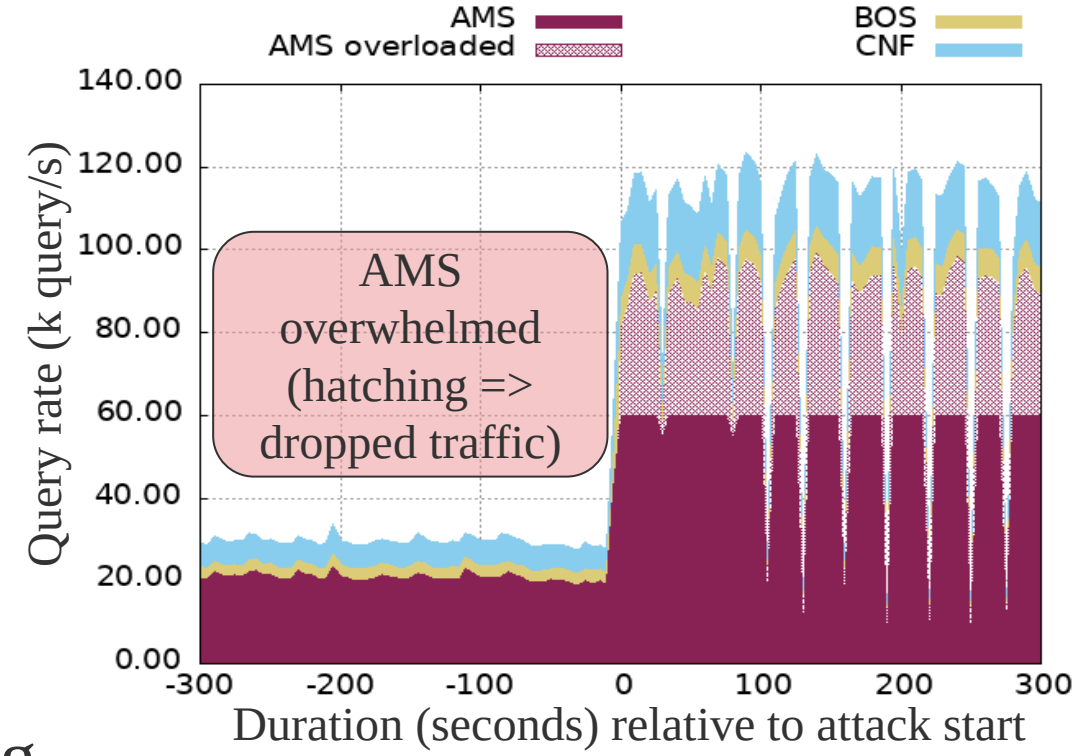
- Question: how to use a playbook during an attack?
- Experiment:
  - Simulate a DNS attack
    - B-root event from 2017-03-06
    - More events in section 8 of the paper
  - Against a 3-site anycast system
    - Each site has ~60k queries/s capacity



Let's look at the BGP playbook.

Solution: Playbook to Get Routing Options

Routing Policy	Traffic to Site (%) AMS	BOS	CNF
~~(a) Route-server~~	~~15~~	~~35~~	~~55~~
~~(b) All-IXP-Peers/Poison transits~~	~~15~~	~~35~~	~~45~~
(c) 2xPrepend AMS	25	35	45
(d) 1xPrepend AMS	35	25	35
(e) -1xPrepend BOS	45	45	15
(f) -1xPrepend CNF	45	5	45
(g) Transit-1	45	25	35
~~(h) Transit-2~~	~~55~~	~~15~~	~~25~~
(i) Poison Tier-1/Transit-2	35	25	35
~~(j) Poison Transit-1~~	~~55~~	~~25~~	~~25~~
~~**(k) Baseline**~~	~~65~~	~~15~~	~~15~~
~~(l) 1,2xPrepend BOS~~	~~65~~	~~5~~	~~25~~
~~(m) 1,2,3xPrepend CNF~~	~~75~~	~~15~~	~~5~~
~~(n) 1, 2, 3xPrepend AMS~~	~~85~~	~~5~~	~~5~~

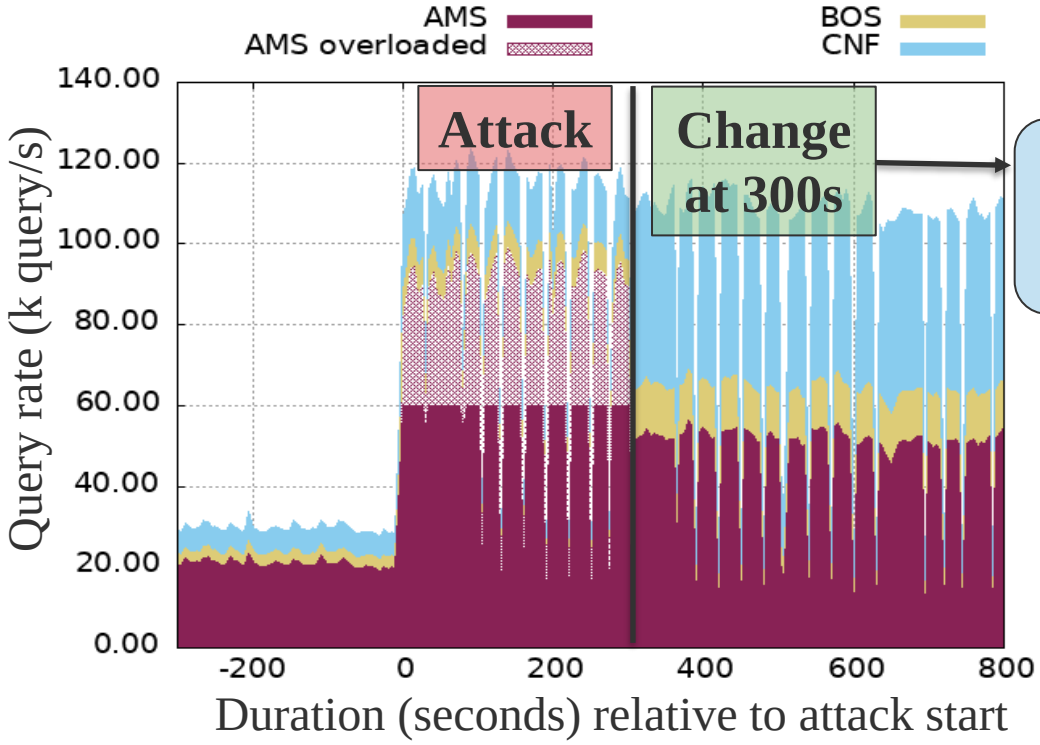


- Goal: lower traffic at AMS
- Several options work: c, d, e, f. g
- We pick **g** to avoid overloading other sites

Outcome after Applying a New BGP Policy

BGP changes at 300s; new traffic balance => no more drops (no hatching)

Routing Policy	Traffic to Site (%) AMS	BOS	CNF	
~~(a) Route-server~~	~~15~~	~~35~~	~~55~~	✗
~~(b) All-IXP-Peers/Poison transits~~	~~15~~	~~35~~	~~45~~	✗
(c) 2xPrepend AMS	25	35	45	✓
(d) 1xPrepend AMS	35	25	35	✓
(e) -1xPrepend BOS	45	45	15	✓
(f) -1xPrepend CNF	45	5	45	✓
(g) Transit-1	45	25	35	✓
~~(h) Transit-2~~	~~55~~	~~15~~	~~25~~	✗
(i) Poison Tier-1/Transit-2	35	25	35	✓
~~(j) Poison Transit-1~~	~~55~~	~~25~~	~~25~~	✗
~~(k) Baseline~~	~~65~~	~~15~~	~~15~~	✗
~~(l) 1,2xPrepend BOS~~	~~65~~	~~5~~	~~25~~	✗
~~(m) 1,2,3xPrepend CNF~~	~~75~~	~~15~~	~~5~~	✗
~~(n) 1, 2, 3xPrepend AMS~~	~~85~~	~~5~~	~~5~~	✗



No loss
Some AMS traffic => CNF

Conclusion

- New method to **estimate attack size** from known good traffic
- Propose **BGP playbook** to plan reactions to DDoS
- **Evaluations against real attacks**
- More information in paper
  - **Artifacts:** https://zenodo.org/record/6473023#.YtyvnBzMI5s