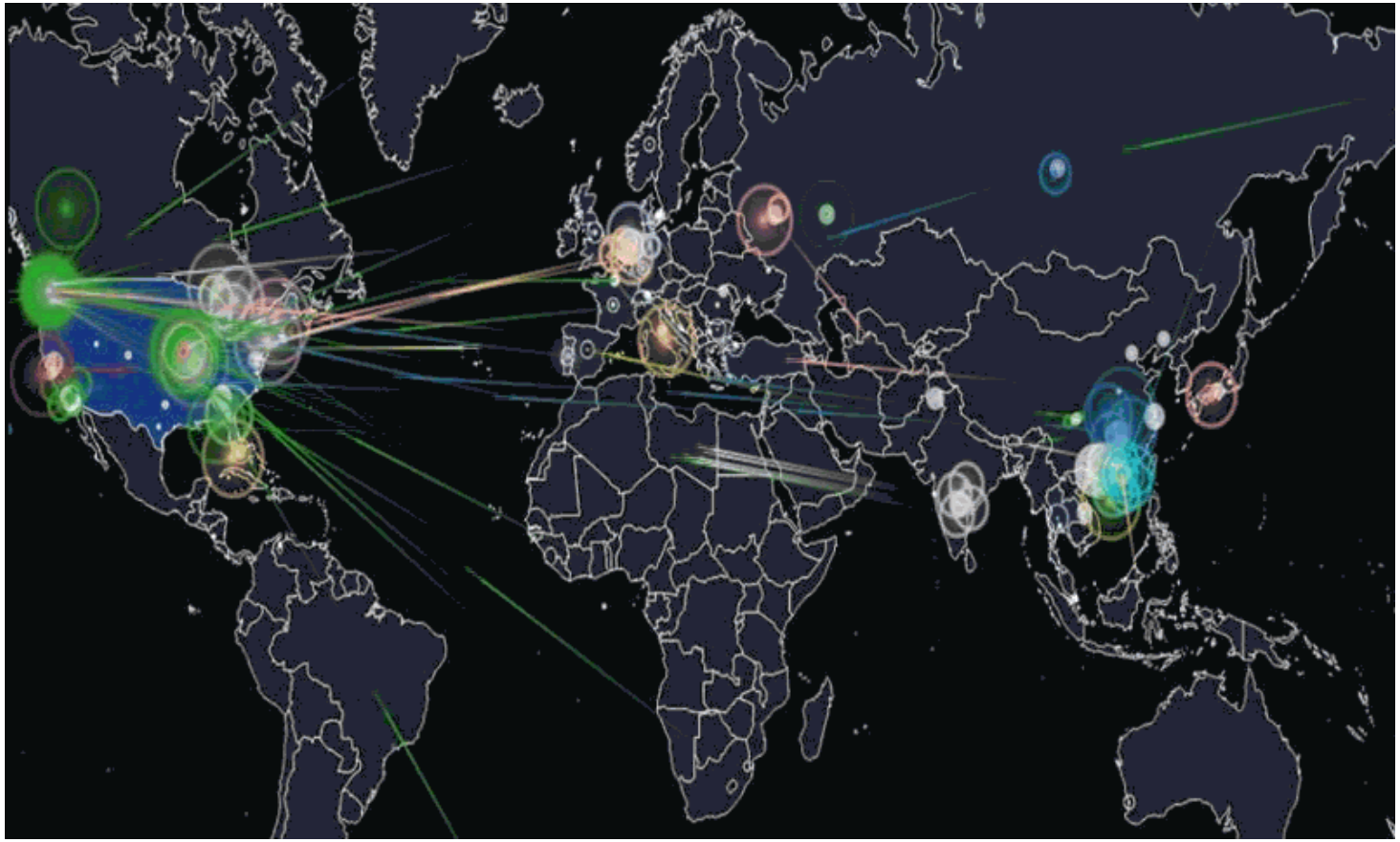
Georgia Institute of Technology

ShadowNet: Towards preventing IoT DDoS from the Edge

Ketan Bhardwaj, Joaquin Chung Miranda, Ada Gavrilovska

Today's Problem: IoT DDoS



- KrebsOnSecurity (Sept '16), Dyn (Oct'16), …
- Used IP cams, home routers (IoT devices)
- Use of weak passwords

DDoS using Mirai botnet on IoT devices

The Root Cause: Insecure Devices



- Malicious botnets infecting devices
- Accidental lapses in security policy enforcing

Lets face it !

It is impossible to ensure every device will always be secure

DDoS: Old Problem - New Face

- Have been around for more than a decade
  - Absorb DDoS attacks, counter IP spoofing, harden systems, firewalls, IDPS, load balancing, active resource monitoring …
- Recent DDoS attacks are application level attacks using IoT devices
  - We call them -- IoT-DDoS
- Bots uses massive amounts of seemingly legitimate traffic
  - Extremely challenging to classify as malicious traffic

Some Interesting Observations

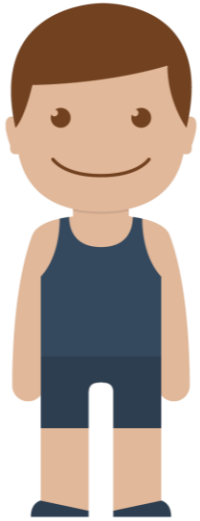
- **Damage is done** at the time existing solutions can help
  - Existing detection & mitigation require the attack to be in progress
- Only way forward (our goal) is to **prevent** attack from happening
- Well known that **prevention** is best near the source of the attack

Edge Computing

Lower Latency

Lower Backhaul Bandwidth

Lower Cost

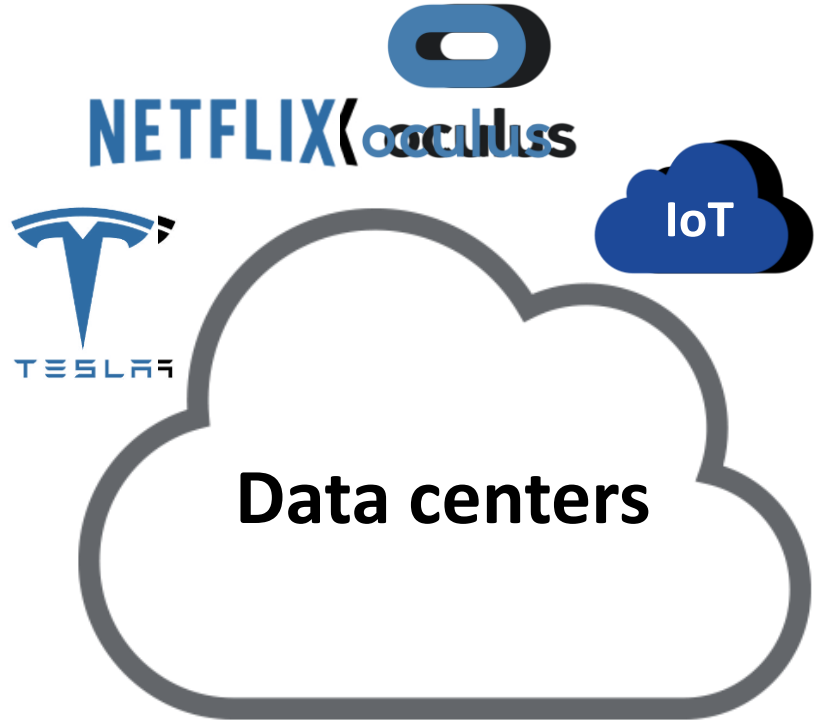




Edge Functions

The Edge

(Cellular Base stations, Enterprise, Personal)



Confidential

Can Edge Computing Help?

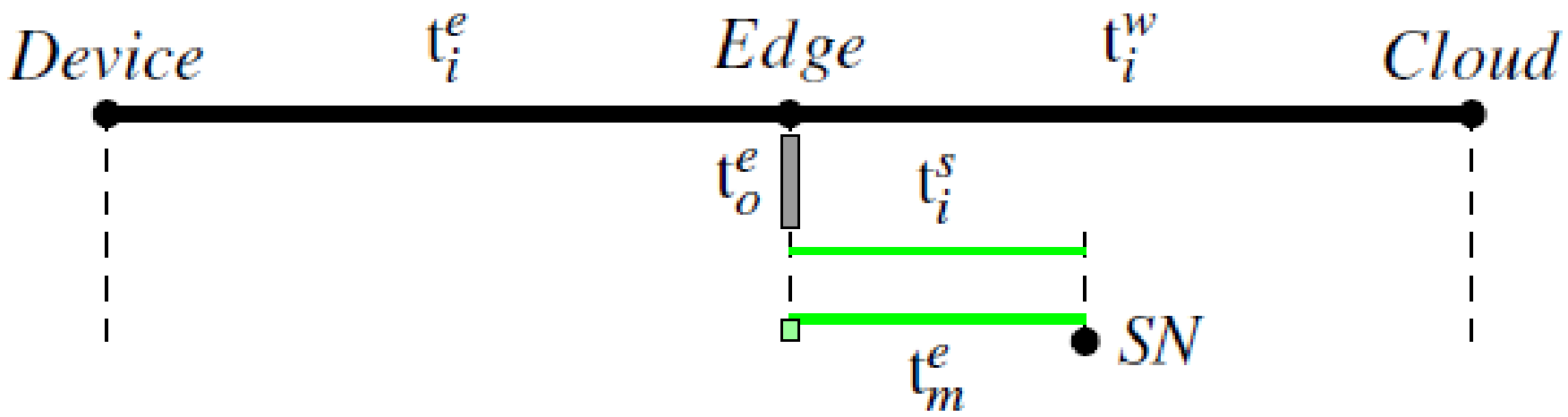
- Edge infrastructure is **closest** to the source of IoT-DDoS
  - A new vantage point for IoT-DDoS prevention

BUT

- A single edge location (function) has **limited visibility**
  - To be able to detect a IoT-DDoS
- Edge infrastructure at a location has **constrained** resources
  - To be able to mitigate a IoT-DDoS



Key Idea

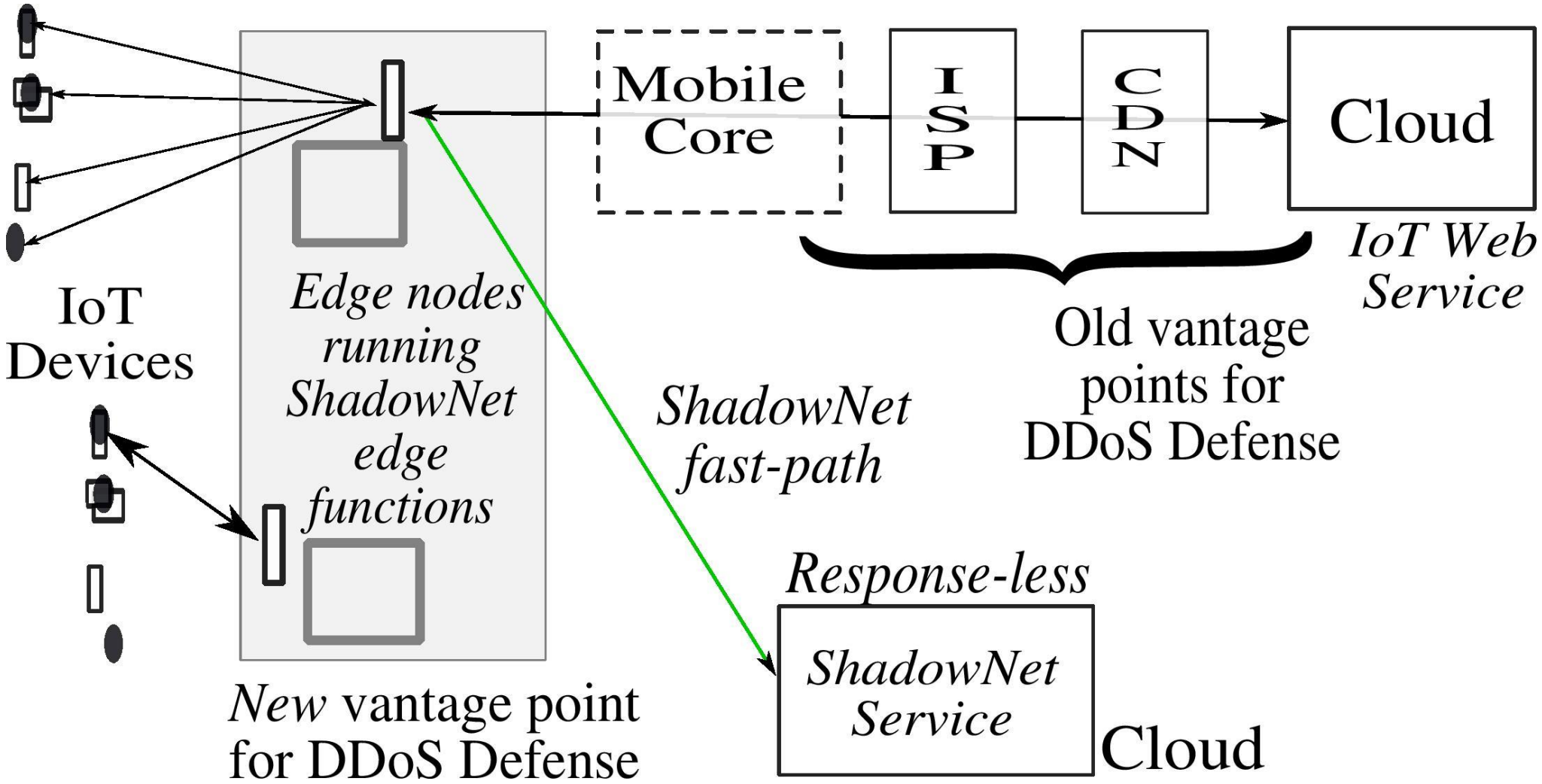


$$f\left(\Sigma\left(t_i^e + t_i^s\right).\, w_i^s\right) + t_o^e \leq f\left(\Sigma\left(t_i^e + t_i^w\right).w_i^w\right)$$

Only if all edge functions could inform a centralized service about the attack

faster than attack itself using minimal resources

ShadowNet System Overview

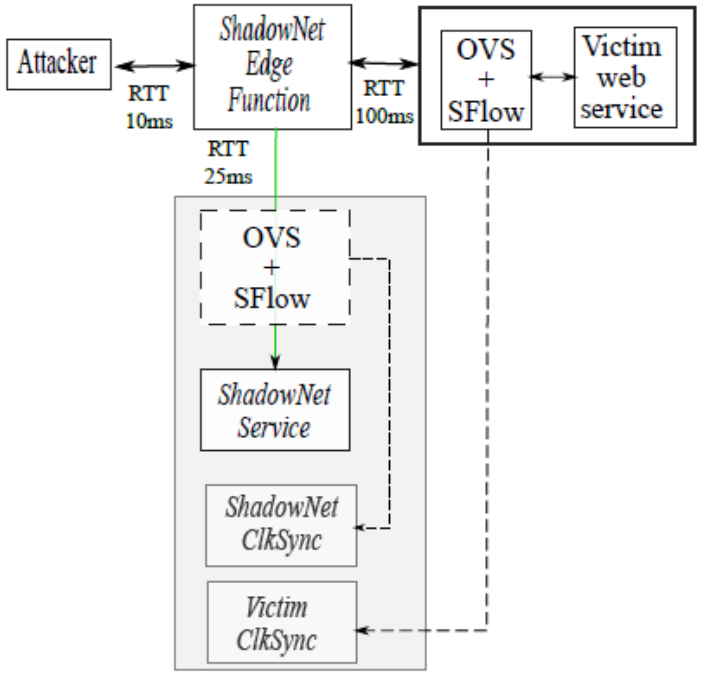


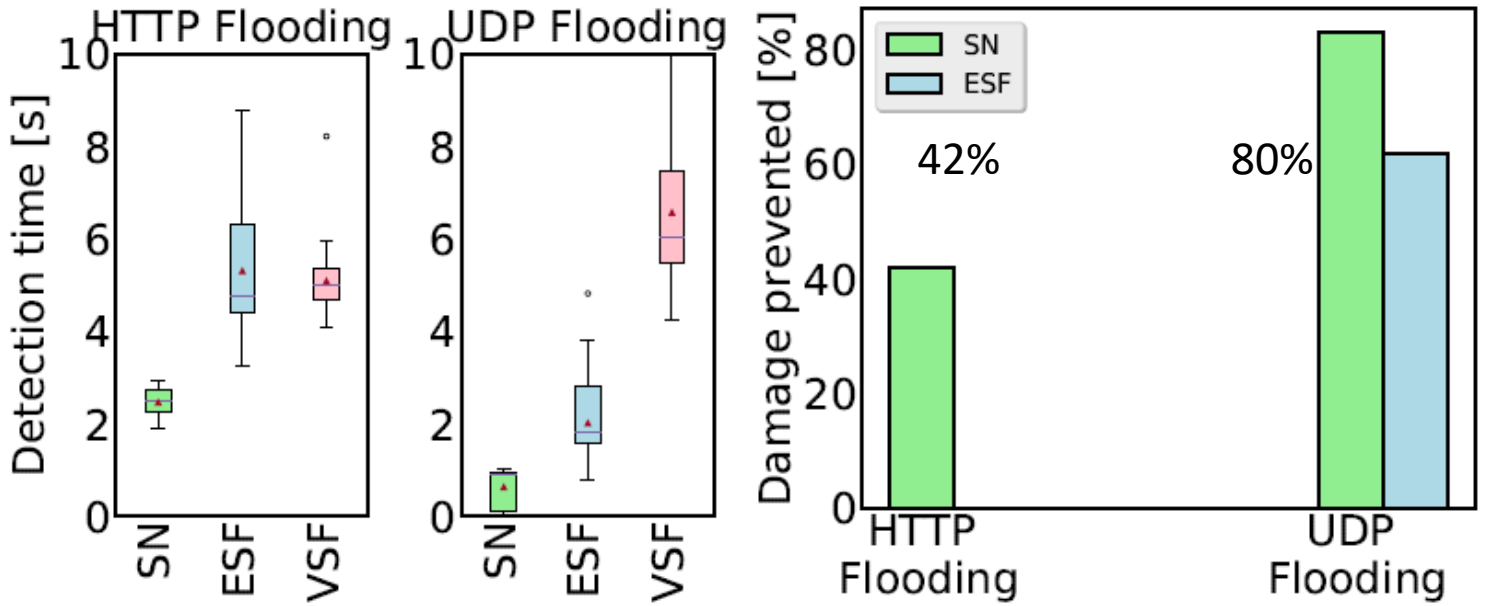
Design Elements

- Realizing **fast-path** for ShadowNet
  - Leveraging HW (network slicing) and SW mechanisms (lower protocol distance)
- Designing **scalable** ShadowNet service
  - Preferably over connection-less, response-less sink for shadow packets over fastpath
- Enabling **efficient** deployment of prevention and mitigation
  - Detection algorithms at ShadowNet service & just-in-time deployment

Up to 10x faster detection prevents 82% damage

- Setup: VMs on GENI platform with Openvswitch & sFlow monitoring
- Attacks generated using Bonesi: 252 attackers per edge function
- Bandwidth overhead UDP: 4.76 Mbps (0.62s), HTTP: 280 Kbps (2.46s)
- Resource overhead: +3.15% CPU, no perceivable memory overhead





Q/A, Feedback and Limitations

- Generalization of the idea: Knowing something **before** it happens
  - a unique and **new property** in networked systems
- Other potential **use cases** (?)
  - Faster ML inference/analytics, Faster perceived web page loading, …
- The **Achille's heal**: fast-path effectiveness