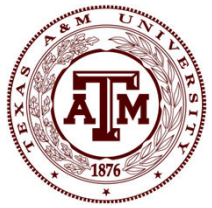
usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION

You Can Obfuscate, but You Cannot Hide: CrossPoint Attacks against Network Topology Obfuscation

Xuanbo Huang[1], Kaiping Xue[1], Lutong Chen[1], Mingrui Ai[1], Huancheng Zhou[2], Bo Luo[3], Guofei Gu[2], Qibin Sun[1]

[1] University of Science and Technology of China
[2] Texas A&M University
[3] The University of Kansas



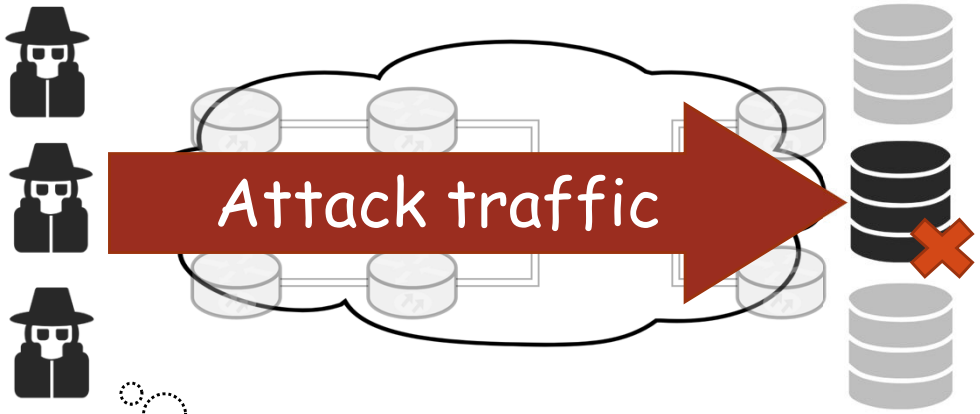
KU THE UNIVERSITY OF KANSAS

Outline

- **Background: DDoS Attacks**
- Proactive defense: Network Topology Obfuscation
- Motivations
- Security Analysis
- The CrossPoint Attack
- Experiment Setup and Results

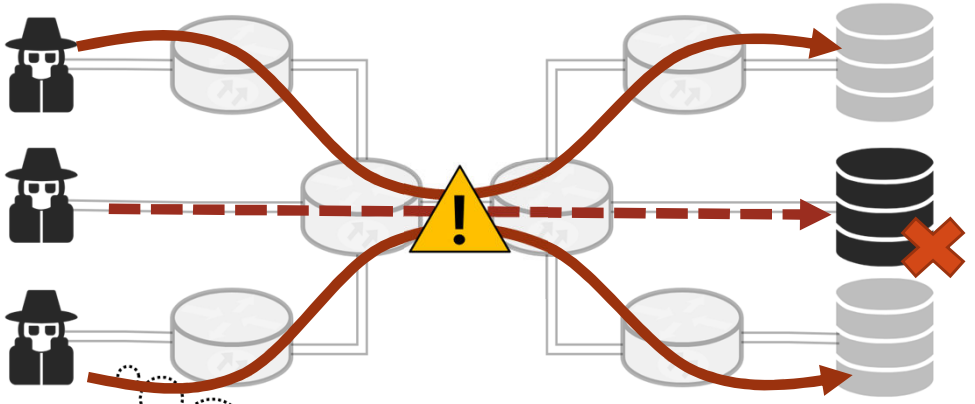
Background: DDoS attacks

Old DDoS



- Flood servers with SYN, UDP, ICMP ….
- Send high-intensity traffic.
- Might be defensed by IDS/firewall.

Link-flooding Attacks

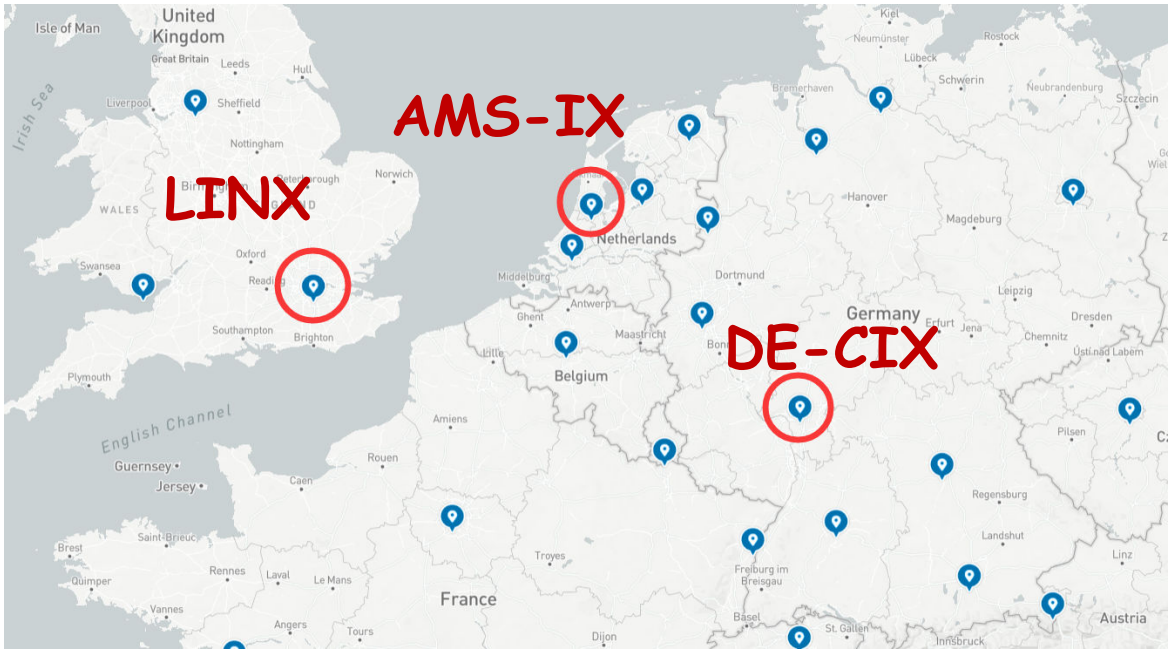


- Probe the topology with traceroute.
- Cut-off network connections.
- May not trigger end-host defense.

Background: Link-flooding Attacks

"Almost Broke the Internet"

- In 2013, CloudFlare reports a large-scale LFA that "*Almost Broke the Internet*".



The reported LFA attacked **four Internet eXchange Point (IXP)** in Asia and Europe.

[1] **Matthew Prince, The DDoS That Almost Broke the Internet.** https://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet

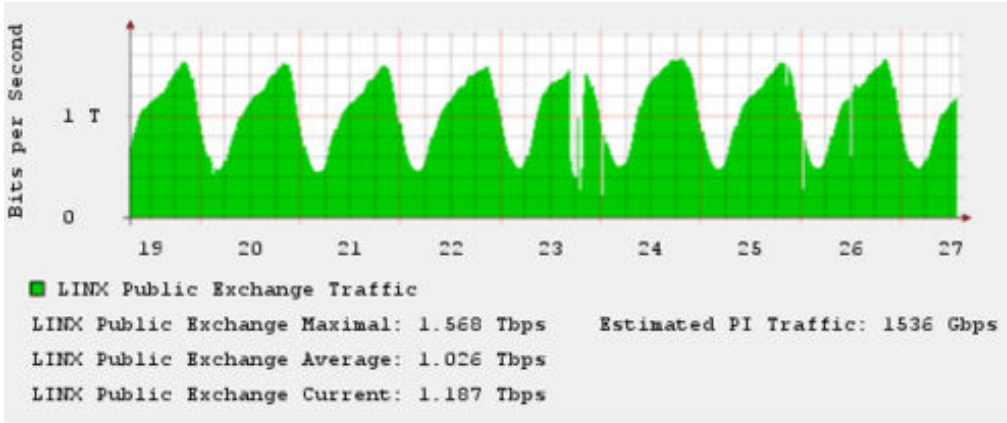
Background: Link-flooding Attacks

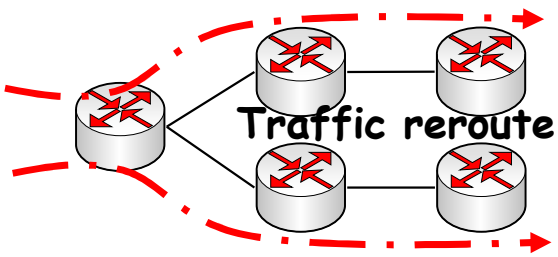
"Almost Broke the Internet"

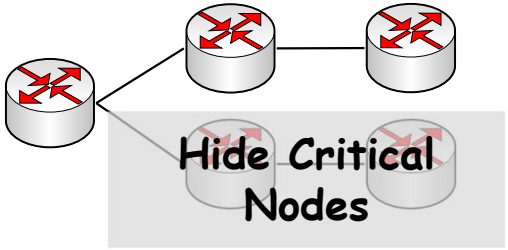
- In 2013, CloudFlare reports a large-scale LFA that "*Almost Broke the Internet*".

QUICK DEFENSE?

1. When detect problems, reroute traffic around the victim IX.
2. IX's IP address should not be announced as routable across the public Internet;







Traffic example of attacks & defenses on LINX in a week.

[1] **Matthew Prince, The DDoS That Almost Broke the Internet.** https://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet

Timeline: Link-flooding Attacks

Reactive DDoS defense

Reactive defense reroute the traffic

[The Coremelt Attack, 2009] ESORICS

[CoDef,2013] CoNext

[SPIFFY, 2016] NDSS

[TE, 2016] INFOCOM

[SIBRA, 2016] NDSS

[LinkScope, 2018] TIFS

[RADAR, 2018] TIFS

[RAC, S&P 2018]

[(In)feasibility of RAC, 2019] S&P

[Poseidon, 2020] NDSS

[Mew, S&P-2023]

[Ripple, Security 2021]

[The CrossFire Attack, 2013] S&P

[LinkBait, 2017]

[BottleNet, 2021] TIFS

Now

[NetHide, 2018] Security

[EqualNet, 2022] NDSS

Proactive defense hide the bottleneck

This paper focuses on the security of proactive defense.

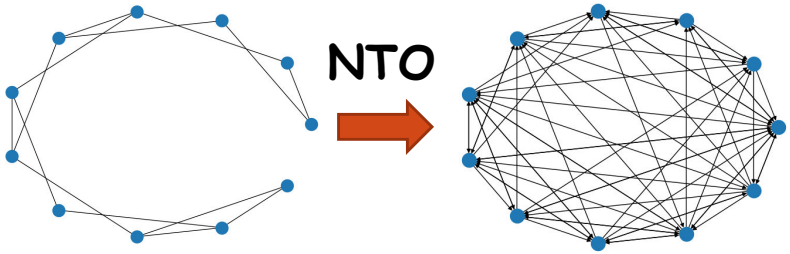
Proactive DDoS defense: Network topology obfuscation

Outline

- Background: DDoS Attacks
- **Proactive defense: Network Topology Obfuscation**
- Motivations
- Security Analysis
- The CrossPoint Attack
- Experiment Setup and Results

Proactive defense: Network Topology Obfuscation

Network topology obfuscation aims at hiding critical Internet nodes.



Physical topology

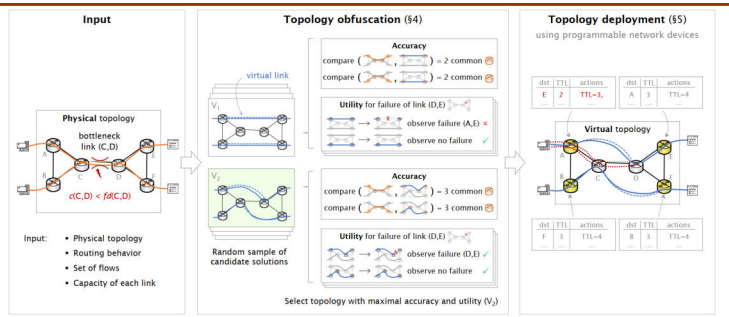
Obfuscated topology

dst	TTL	actions
E	2	TTL=3,
…	…	…

dst	TTL	actions
A	3	TTL=4
…	…	…

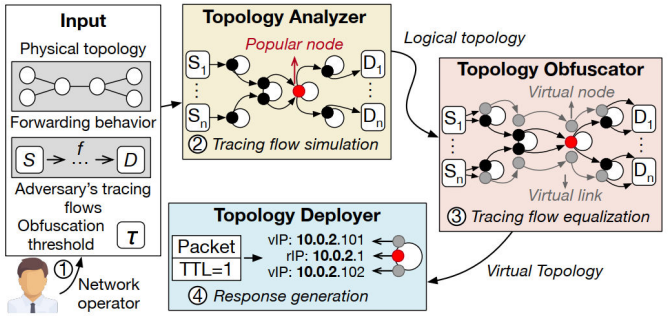
NetHide

USENIX Security 2018



EqualNet

NDSS 2022



Question: Do these SOTA NTO defenses provide adequate security?

Outline

- Background: DDoS Attacks
- Proactive defense: Network Topology Obfuscation
- **Motivations**
- Security Analysis
- The CrossPoint Attack
- Experiment Setup and Results

Motivations

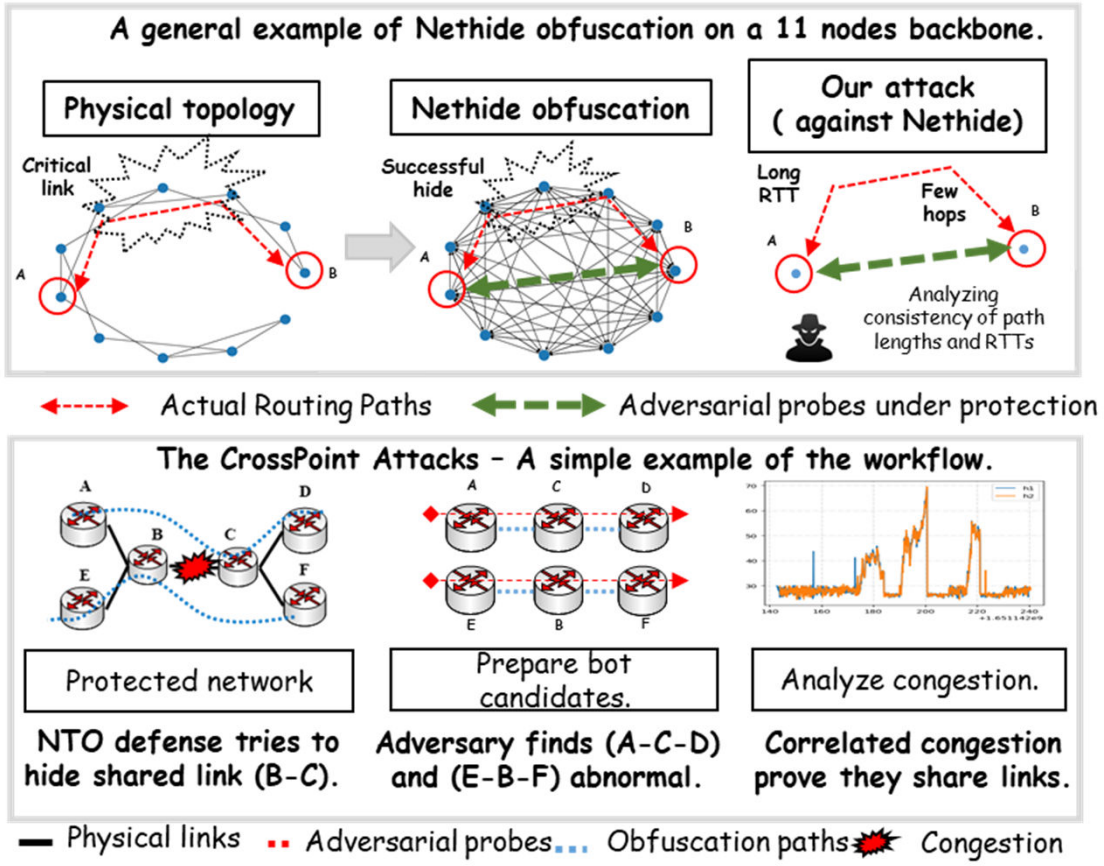
NTO schemes **CANNOT** hide the robust low-level network traffic patterns.

Insight I:

Crafted virtual paths exhibit ***statistical disparities*** compared to physical links in certain attributes.

Insight II:

Attackers can identify hidden physical bottleneck links through ***correlated congestion.***



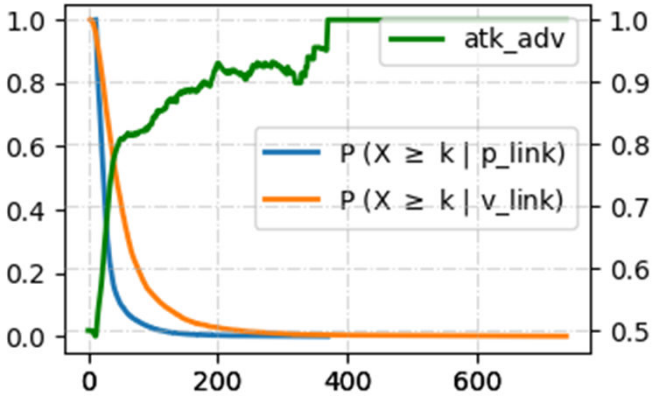
Outline

- Background: DDoS Attacks
- Proactive defense: Network Topology Obfuscation
- Motivations
- **Security Analysis**
- The CrossPoint Attack
- Experiment Setup and Results

Security analysis: Statistical disparities

Insight I: Crafted virtual paths may exhibit ***statistical disparities*** compared to physical links in certain static attributes (e.g., propagation delay, subnet IP…).

"***Hide links***" reduces hops but maintains the propagation delay.

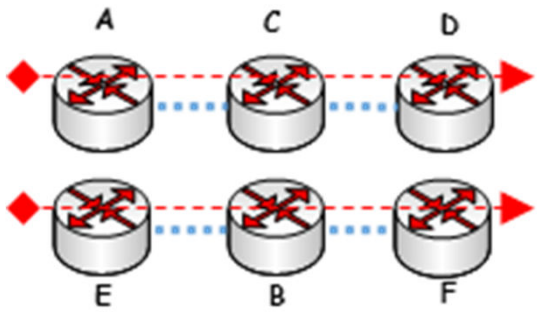


Use the statistical relationship between delay and hops as prior knowledge.

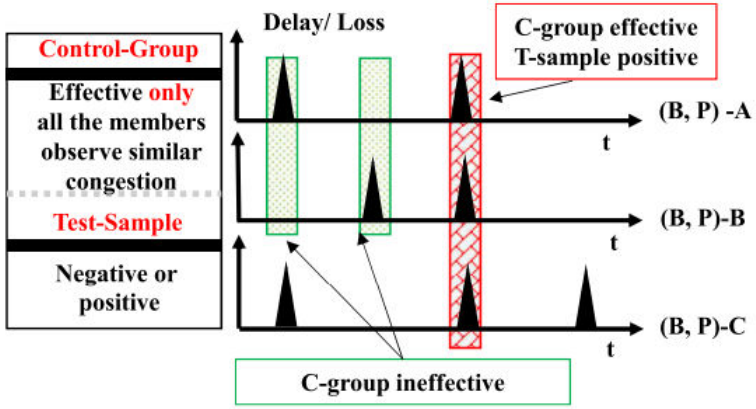
Identify some suspicious bots that pass through virtual paths.

Security analysis: Correlated congestion

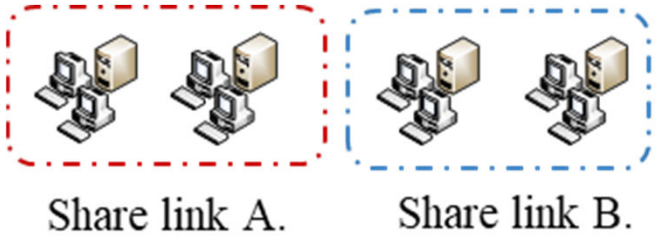
Key idea II: Attackers can identify hidden physical bottleneck links through ***correlated congestion.***



Send *ping* traces on the virtual paths identified in the previous step



Analyze the **correlation** of these ping traces.

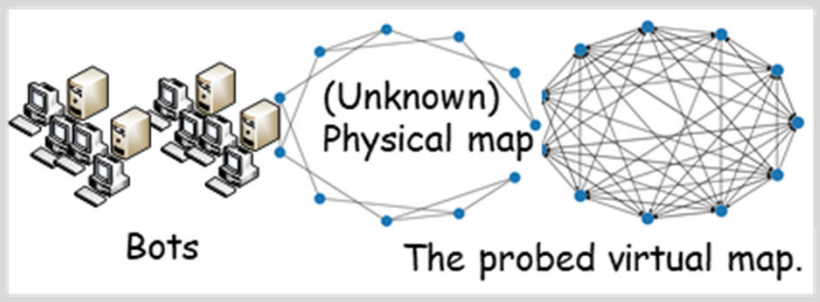


Aggregate correlated attack flows (share the same link) together.

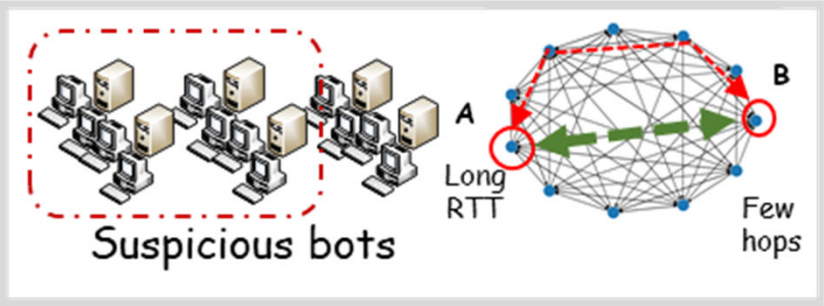
Outline

- Background: DDoS Attacks
- Proactive defense: Network Topology Obfuscation
- Motivations
- Security Analysis
- **The CrossPoint Attack**
- Experiment Setup and Results

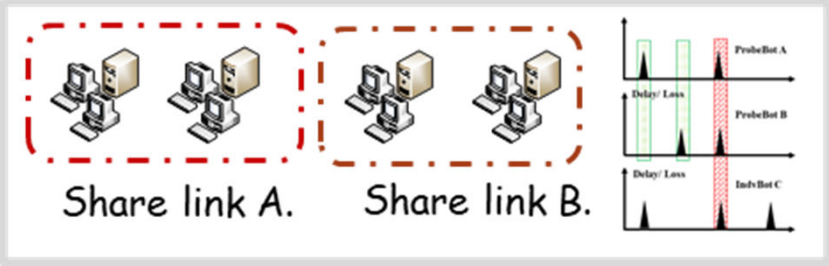
The CrossPoint Attack



STEP 1: Probing Protected Virtual Map with traceroute.

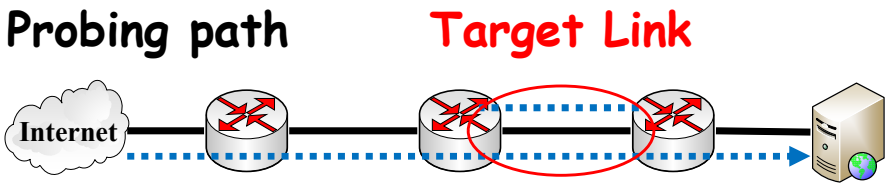


STEP 2: Detecting Virtual Links with Statistical Disparities (SD).



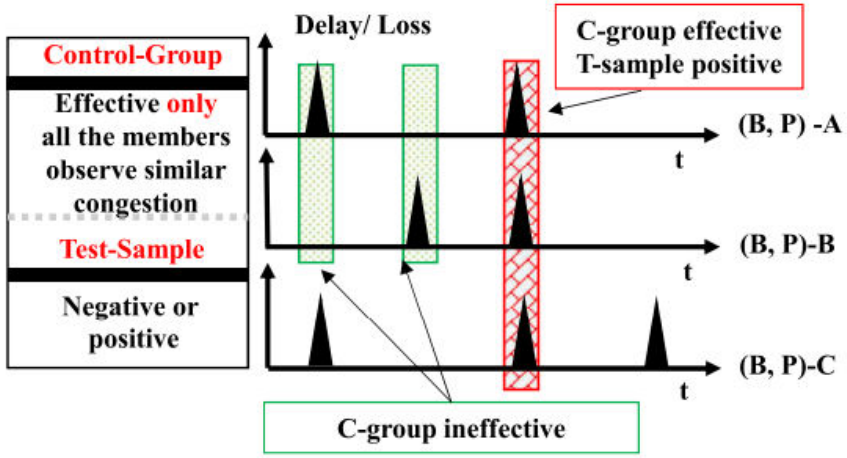
STEP 3: Identifying Physical Links using Correlated Congestion (CC).

Correlated congestion: location



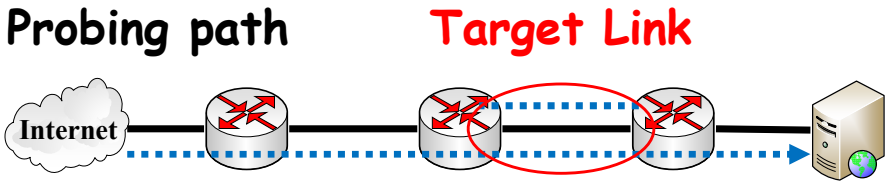
Congestion events can happen **everywhere**!!

How to locate congestion on a certain link?



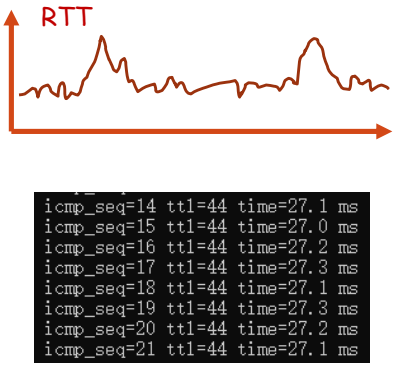
1. Identify at least two flows that share a link to serve as a control group.
2. Filter out congestion events that are not observed by all members of the control group.

Correlated congestion: noise

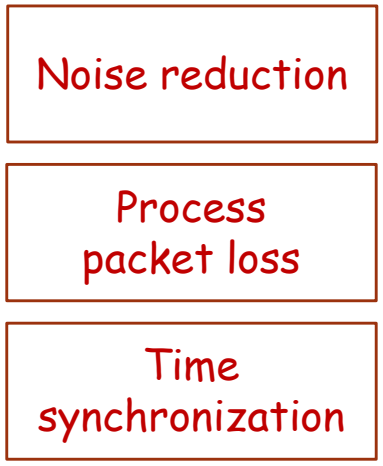


Internet is noisy!

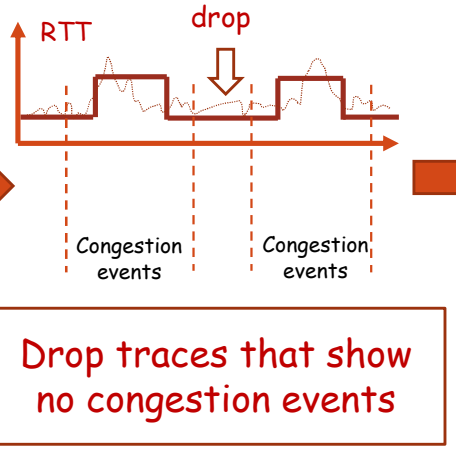
How to quantify the correlation ?



Ping traces



Preprocessing



Slicing

$$\mathbf{C_x} = E\{[\mathbf{X}(t) - \mu_x][\mathbf{X}(t) - \mu_x]^T\} = \begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{pmatrix},$$

where c_{ij} represents the covariance of $\mathrm{bs}_i(t)$ and $\mathrm{bs}_j(t)$, while μ_x is the mean vector of $\mathbf{X}(t)$. Next we calculate the PCC matrix as:

$$\mathbf{R}_x = \begin{pmatrix} \frac{c_{11}}{\sigma_1^2} & \cdots & \frac{c_{1n}}{\sigma_1\sigma_n} \\ \vdots & \ddots & \vdots \\ \frac{c_{n1}}{\sigma_n\sigma_1} & \cdots & \frac{c_{nn}}{\sigma_n\sigma_n} \end{pmatrix} = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nn} \end{pmatrix},$$

where σ_i is the standard deviation of $\mathrm{bs_i}$. To find C-group effective samples, we search in all congestion and calculate distance between the congestion and an ideal sample as :

$$D = \underset{i,j \leq n}{Max} \ (\Delta_x - \mathbf{R}_x)_{ij},$$

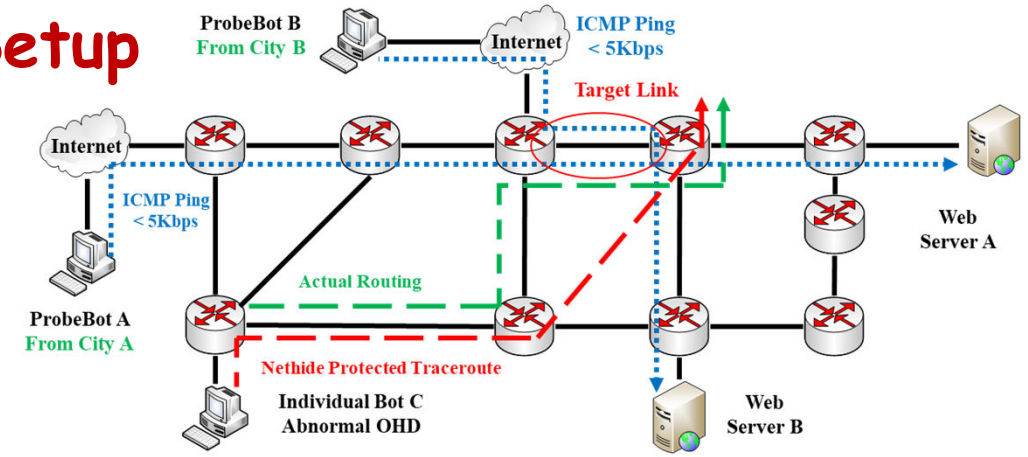
Person correlation coefficient

Outline

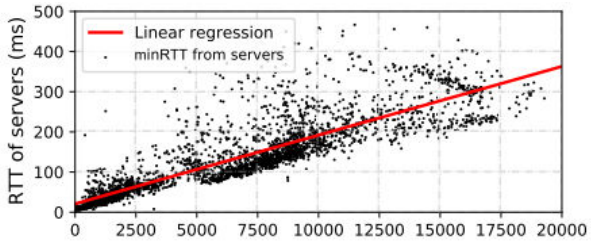
- Background: DDoS Attacks
- Proactive defense: Network Topology Obfuscation
- Motivations
- Security Analysis
- The CrossPoint Attack
- **Experiment Setup and Results**

Evaluations: Correlated congestion

Setup



Server id	Location
1,2	Shanghai, China
3	Hong Kong, China
4	Beijing, China
5	San Jose, U. S.



SID*	Date (2022)	Duration	Noise Congestion
1	05-18 — 05-24	120.2 h	10K+
2	05-20 — 05-21	29.15 h	163
3	05-22 — 05-24	29.15 h	955
4	05-22 — 05-23	29.12 h	1384
5	05-18 — 05-19	29.12 h	3053

Results

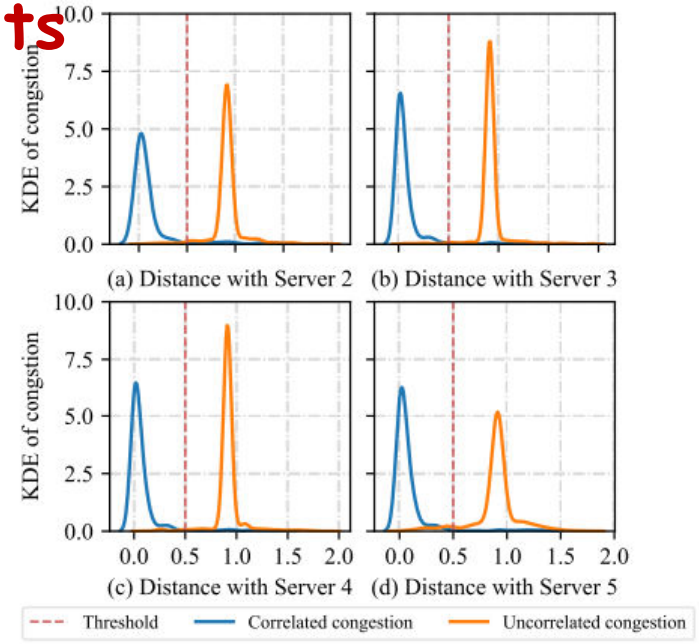


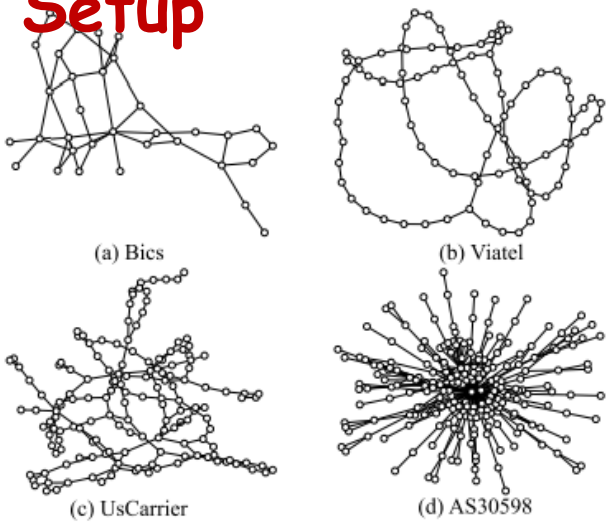
Table 2: Congestion classification results

Metrics	Accuracy	Precision	Recall	F1 score
Fig.7(a)	96.2%	98.3%	94.4%	96.3%
Fig.7(b)	97.5%	98.4%	96.8%	97.6%
Fig.7(c)	97.7%	98.4%	97.3%	97.8%
Fig.7(d)	95.4%	94.2%	97.3%	95.7%

Evaluations

Comprehensive performance

Setup

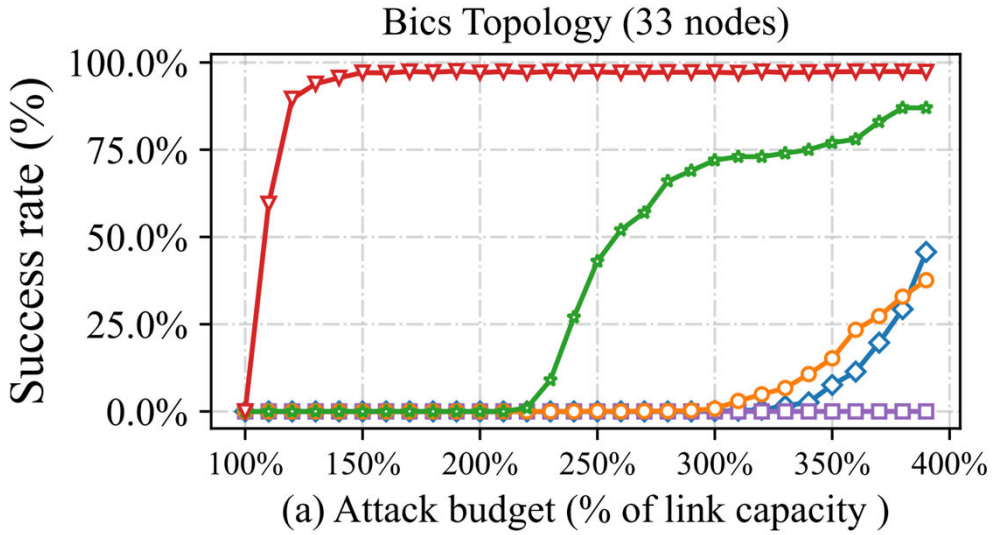


【DEFENSE】
Nethide
[Security, 18]

【DEFENSE】
EqualNet
[NDSS, 22]

【ATTACK】
Crossfire
[S&P, 13]

【ATTACK】
Coremelt
[ESORICS, 09]



Crossfire Random
(Proposed in Nethide)

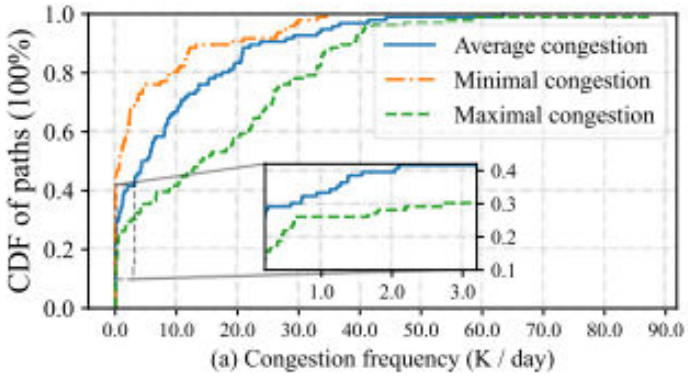
Coremelt Random

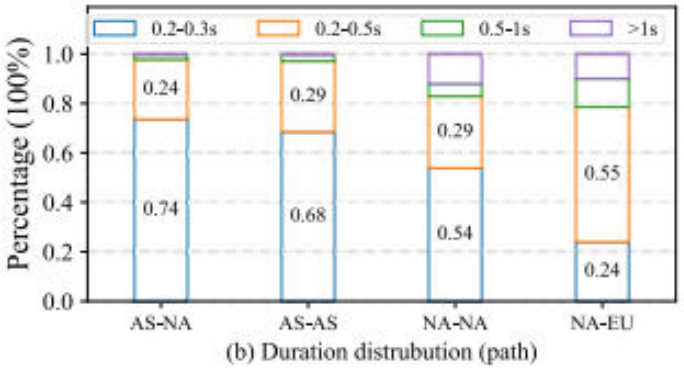
CrossPoint(SD)

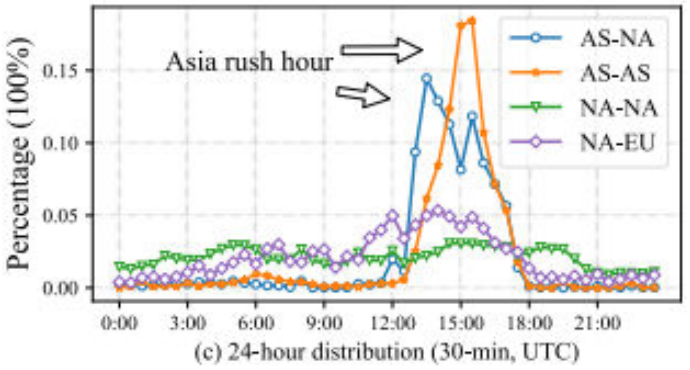
CrossPoint(SD+CC)

Crossfire

Evaluations: Measurement study







SETUP: 6 senders * 20 public servers (DNS, WEB, …) with 10 PPS ping.

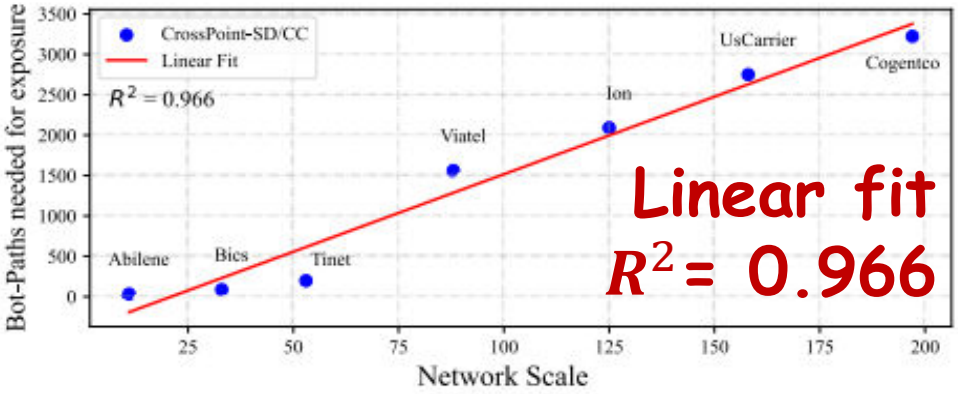
1. There are sufficient congestion events for the attacker to exploit.
2. The attacker can send pings at 10 PPS to observe most congestion events.
3. There are "rush hours" on some Internet paths.

Evaluations

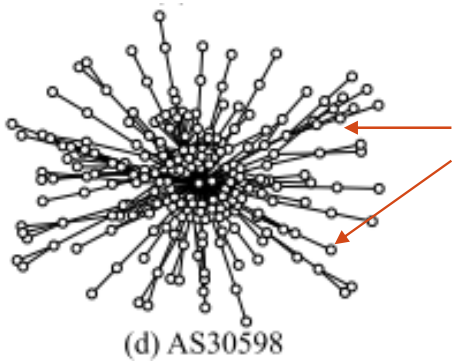
Scalability: Test the required bot-paths for a 90% success rate on 7 topologies.

Table 3: Topology used in scalability experiments

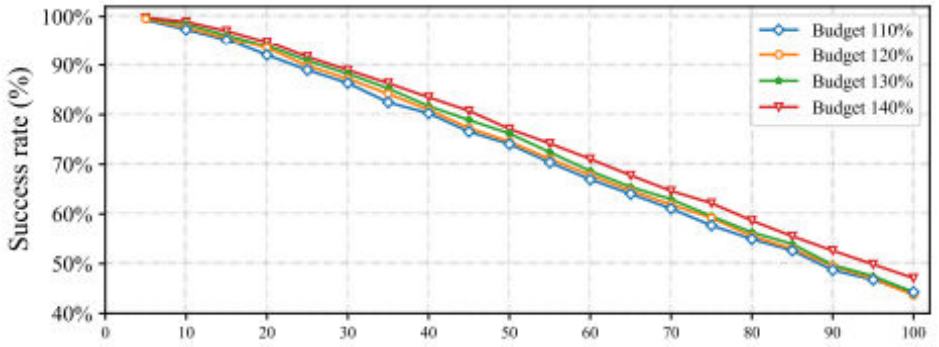
Topology	Nodes	Edges	Topology	Nodes	Edges
Abilene	11	14	Ion	125	146
Bics	33	48	UsCarrier	158	189
Tinet	53	89	Cogentco	197	243
Viatel	88	92			



Potential defense: Create fake congestion to mislead the attacker.


(d) AS30598

Simultaneous random congestion



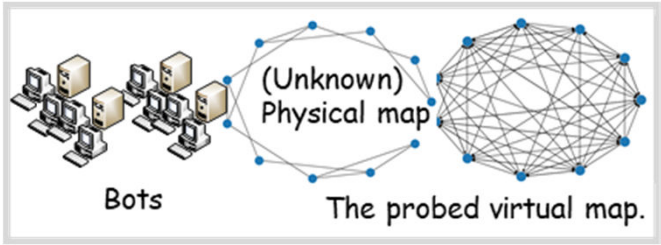
Summary

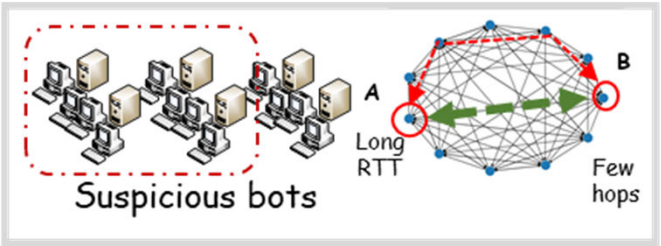
Insight I:

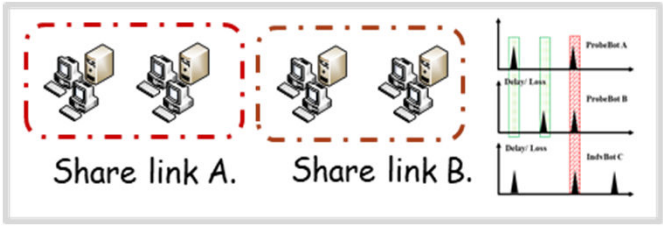
Crafted virtual paths exhibit ***statistical disparities*** compared to physical links in certain attributes.

Insight II:

Attackers can identify hidden physical bottleneck links through ***correlated congestion.***







STEP 1:

Probing Protected Virtual Map.

STEP 2:

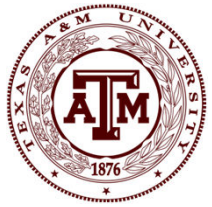
Detecting Virtual Links with Statistical Disparities (SD).

STEP 3:

Identifying Physical Links using Correlated Congestion (CC).

usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION

Thank you!
Q&A



KU THE UNIVERSITY OF KANSAS