Ben-Gurion University of the Negev

DEFCON

HYDRA: Persistent and Multi-Zone Disruption in LEO Constellations Using Minimal Resources

Roee Idan, Rami Puzis, Asaf Shabtai, Yuval Elovici

CBG Cyber@ Ben-Gurion University of the Negev

2025

Why Satellite DDoS Needs a Closer Look?

- LEO satellites power global internet, banking, disaster response, and defense
- But they rely on **predictable links** with **limited capacity**
- DDoS isn't just about volume — it's also about **targeting bottlenecks**



Can you disrupt communication between targeted geographical zones using only a few bots?

Yes - HYDRA shows how a small botnet, placed smartly, can do massive damage

Cyberthreats on Satellites

BleepingComputer

Viasat shares details on KA-SAT satellite service cyberattack

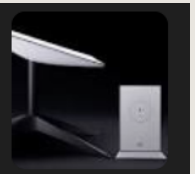
US satellite communications provider Viasat has shared an incident report regarding the cyberattack that affected its KA-SAT...

30 Mar 2022

Space.com

Elon Musk: Russia is ramping … ttacks on Starlink in Ukraine

SpaceX's Starlink … cyberatt… …e are under increasing



LILY HAY NEWMAN SECURITY AUG 28, 2024 11:19 AM

Notorious Iranian Hackers Have Been Ta…

In addition to its long-standing password spraying … malware dubbed "Tickler." …eloped custom

These attacks are no longer just theoretical they already affected real-world systems and disrupting critical infrastructure.

The Washington Post

Russian hacker group exploit… …steal data, hide tracks

The Turla group has hijacked data beamed from commercial satellites in Africa and the Middle East.

9 Sept 2015



China-Based Cyber Espionage Campaign Targets Satellite, Telecom, Defense Firms

Threat group Thrip is using three computers based in China to steal data from targeted companies in Southeast Asia and the US, Symantec says.



Jai Vijayan, Contributing Writer

June 20, 2018

5 Min Read

ENDPOINT SECURITY

CBG Cyber@Ben-Gurion University of the Negev

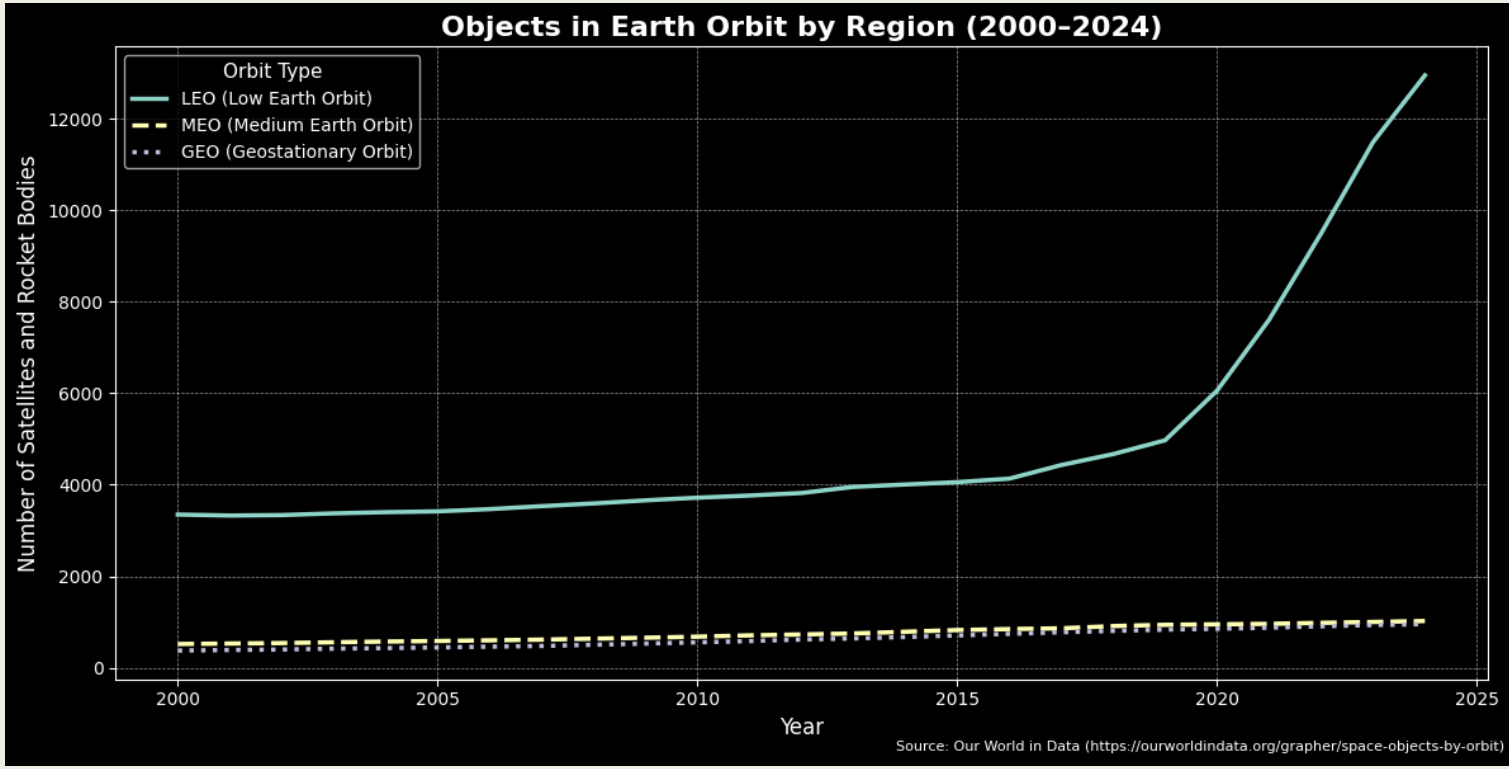
HYDRA

What You'll Hear in This Talk

- Why satellite networks are vulnerable to DDoS
- How link-flooding attacks work in space
- The HYDRA attack framework – 4 attack models
- Results from large-scale LEO network simulations
- What defenders can do to fight back

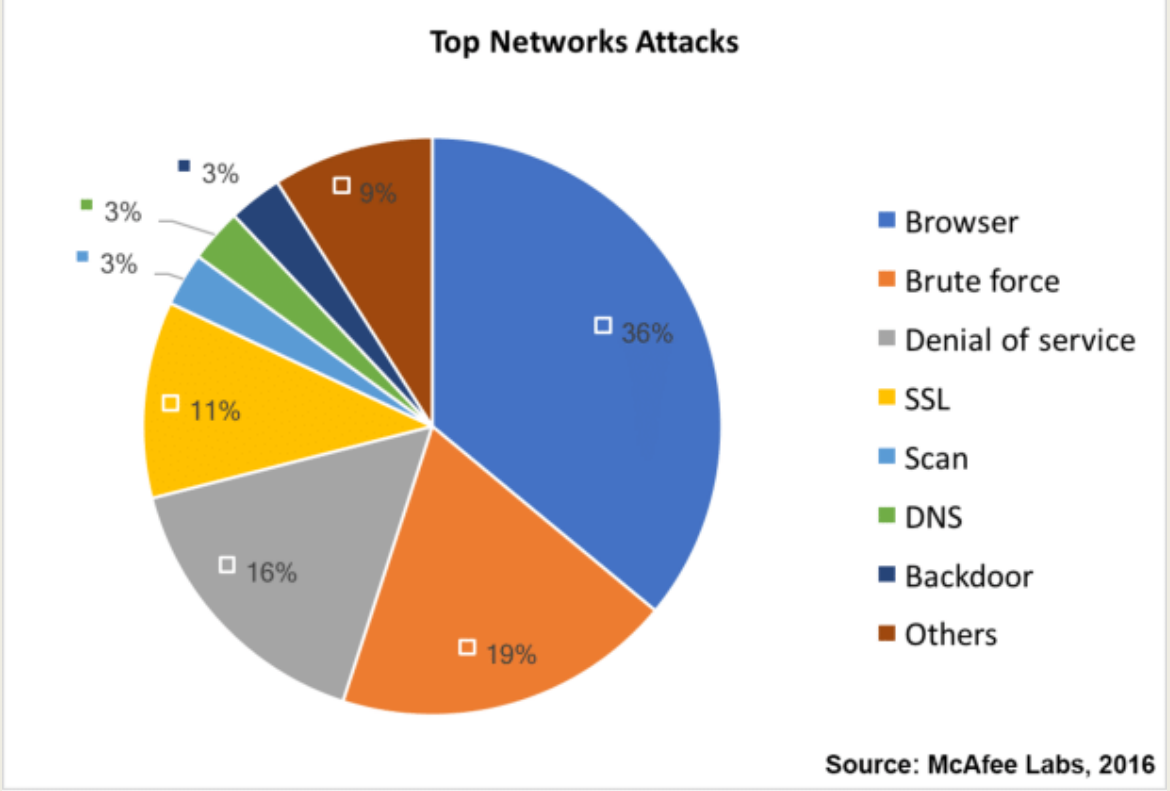
What Makes LEO Satellite Networks Unique?

- Thousands of satellites orbiting up to 2,000 km altitude
- Connect to ground via GSLs (ground-satellite links) and to each other via ISLs (inter-satellite links)
- Usually each satellite connects via four ISLs, two within its orbital plane and two to adjacent planes forming a sparse, time-varying network.
- This constrained structure and continuous orbital motion create topological pressure points, where congestion is likely to emerge.



Cyberattack Types: The Terrestrial Landscape

- **Volumetric Attacks** – Overwhelm bandwidth with junk traffic (e.g., UDP floods)
- **Protocol Attacks** – Exploit TCP, DNS, or HTTP behaviors (e.g., SYN floods)
- **Application Layer Attacks** – Target web login forms, APIs, or app logic
- **Link-Flooding Attacks** – Congest critical paths between regions
- Most existing DDoS defenses are designed for **endpoint- and volume-based attacks**



Why Focus on Link-Flooding Attacks?

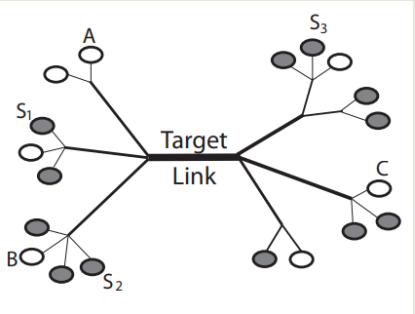
- **Volumetric attacks** are loud and easily detected
- Satellites in these networks are primarily used to route data between ground locations.
- **LFAs** exploit routing, not endpoints — and use normal-looking traffic
- **Sparse, predictable links** make satellites ideal targets for LFAs

Link-Flooding Attacks: From Coremelt to Space

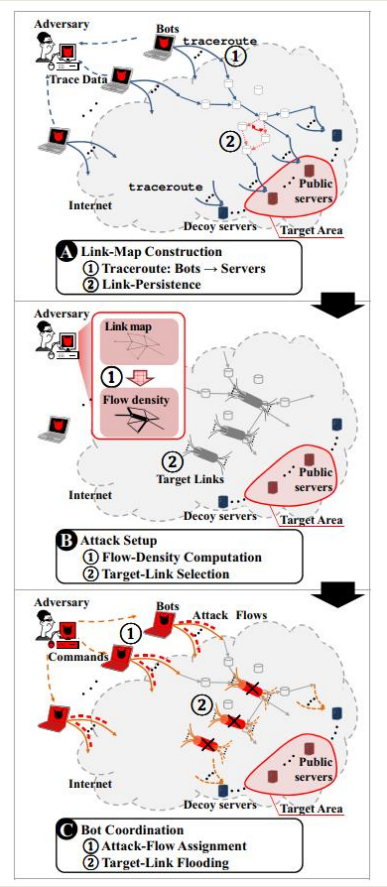
- **Coremelt** et.al. (2009) – Bots flood each other across core links ⟶ infrastructure meltdown
- **Crossfire** et.al (2013) – Bots send low-rate traffic to real servers ⟶ congest key paths

---

- Target: links, not endpoints
- Same concept works in LEO — attacker times traffic to hit weak orbital paths



Coremelt et.al (2009)



Crossfire et.al(2013)

Why Satellites Are Especially Vulnerable to LFAs

- **Orbits and routing paths are predictable** (public orbital data)
- **Limited bandwidth**: 4–25 Gbps for GSLs, 10–100 Gbps for ISLs
- Routing tends to follow shortest paths for lower latency, enabling high-accuracy path prediction
- Global botnet reach: bots can join from anywhere on Earth

Dynamic topology = **constantly evolving bottlenecks**



Why Most LEO Traffic Still Uses Ground Routing — For Now

Current Status

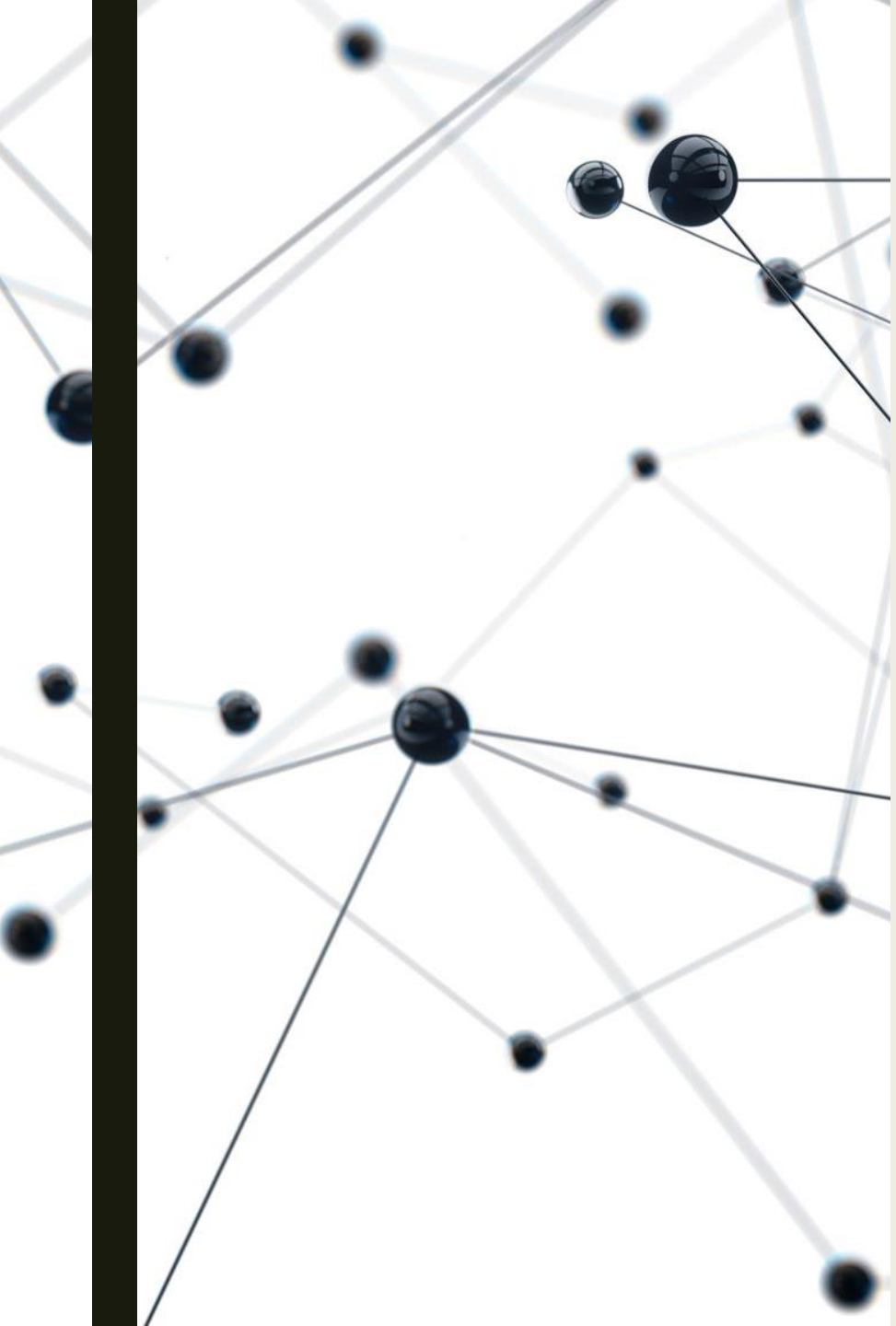
- In over **70% of cases**, Starlink routes traffic to the **nearest ground station** (Izhikevich et.al. 2024)

Reasons:

- Routing in LEO constellations is **harder,** the network changes constantly
- GS-to-GS routing is **more mature and widely deployed**

Future Outlook

- Prior work shows ISLs can reduce end-to-end latency by over 2× (Network topology at 27,000 km/h)
- **This latency gain comes from two factors:** light travels ~1.5× faster in vacuum than in fiber, and ISLs can take more direct routes than ground cables.
- More and more satellites are equipped with ISLs
- End-to-end satellite routing is expected to carry more traffic in the coming years, especially for:
- **Long-distance** or **remote-region** communication
- **Latency-sensitive** use cases (e.g., defense, finance)
- **Geopolitically isolated regions**



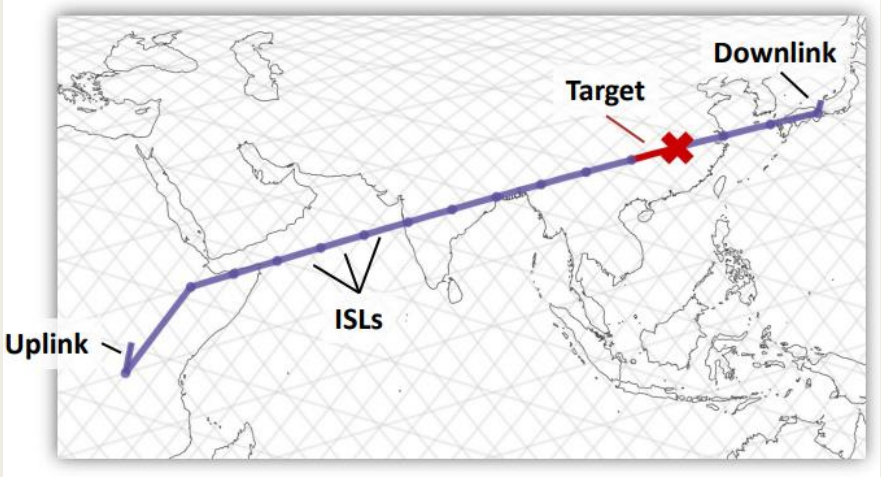
The Problem: Disrupting Zones via Satellite Links

- Goal: Block communication between geographic regions (e.g., Europe ↔ Asia)
- Use ground bots to congest satellite links at the right time
- Attack must **adapt to the dynamic network** (satellites move every minute)
- Use **as few bots as possible** to avoid detection

Foundations: Prior Work on Satellite LFA DDoS Attacks

- **ICARUS (USENIX 2021)** – Demonstrated link-flooding attacks are feasible in LEO
- **DoSat (ACNS 2024)** – Used temporal lensing to enhance short-term disruptions
- **StarMaze (IEEE IWQoS 2024)** – Exploited orbital patterns to congest satellite rings
- **SKYFALL (NDSS 2024)** - Showed how attackers can disrupt LEO traffic by congesting shifting GSL bottlenecks

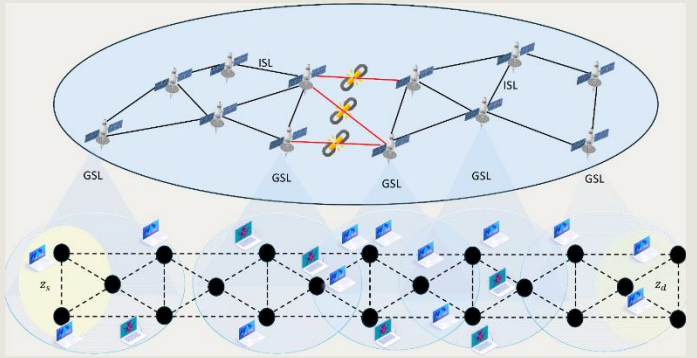
These works prove LEO vulnerabilities — but don't consider how few bots are needed or how long and effectively the attack can be sustained



The Missing Piece: Planning Across Time

- Real attackers want to conserve resources and stay undetected
- Need to sustain disruption over time, not just one moment
- Must adapt to satellite motion and routing shifts
- Goal: Minimize botnet size while maximizing impact
- This is a **discrete optimization** problem, not brute force

So This Is Where HYDRA Steps In



- HYDRA models the satellite network as a **dynamic graph** over time
- **Nodes** represent satellites and ground-based terminals (bots) - N_S, N_{GPO}
- **Links** include:
- GSLs: ground-to-satellite uplinks
- ISLs: inter-satellite links (changing by snapshot)
- Each time snapshot t is represented as a graph
- Paths between source and destination zones are extracted as
- These paths become **attack targets** for planned congestion

$$\text{GSLs: } L_{GSL} \subseteq 2^{N_{GPO}} \times N_S$$

$$\text{ISLs: } L_{ISL} \subseteq N_S \times N_S$$

$$G_t = (N_S \cup N_{GPO}, L^t_{GSL} \cup L^t_{ISL})$$

$$\mathcal{F}_t(g_s, g_d) = (gsl_s, isl_1, \ldots, isl_k, gsl_d)$$

What's the smallest set of bots I need, and where should I place them, to congest all of these paths over time?

Threat Model & Optimization Formulation

Attacker Model

- Controls a **global botnet** of upload-only ground devices (GPOs)
- Knows **satellite orbits and coverage** from public TLE data
- Estimates **routing and capacity** using topology snapshots
- Plans attacks **offline** and executes traffic in real time
- Goal: **Congest all paths** between zone pairs using **minimal bots**

Optimization Formulation

- Paths between zones: $P_t = \{\mathcal{F}_t(g_s, g_d) : g_s \in z_s, g_d \in z_d\}$
- Expected available capacity:
$$\mathcal{E}_t(l) = c_l - \sum_{g_s,g_d:l\in\mathcal{F}_t(g_s,g_d)} \mathbf{TM}_{t,g_s,g_d}$$
- Residual Capacity during attack:
$$\mathcal{R}(l) = \mathcal{E}(l) - \sum_{g_s,g_d:l\in\mathcal{F}_t(g_s,g_d)} \mathbf{ATM}_{g_s,g_d}$$
- Zone congestion function:
$$\mathcal{C}_P(\mathbf{ATM}) = \begin{cases} 1 & \text{if} \quad \forall\, p \in P \;\; \exists\, l \in p,\; \mathcal{R}(l) = 0, \\ 0 & \text{else.} \end{cases}$$
- How much flow from each geo location: $o(g_s) = \sum_{g_d} \mathbf{ATM}_{g_s,g_d}$
- How many compromised users were used in the attack:
$$\mathcal{S}(\mathbf{ATM})_{g_s} = \left\lceil \frac{o(g_s)}{\beta} \right\rceil$$
- **Minimize botnet objective:**
$$\mathbf{ATM}^* = \arg\min_{\mathbf{ATM}} \|\mathcal{S}(\mathbf{ATM})\|_1$$

Strategy 1 — SS-BMP:
Snapshot Disruption with Minimal Bots

- Solve the simplest case: **one snapshot, one target zone pair**
- Goal: **Fully congest all paths** between zones in that snapshot
- Use as **few ground bots as possible**
- This gives us a **lower bound** on required botnet size
- Models static attack feasibility in a time-frozen network

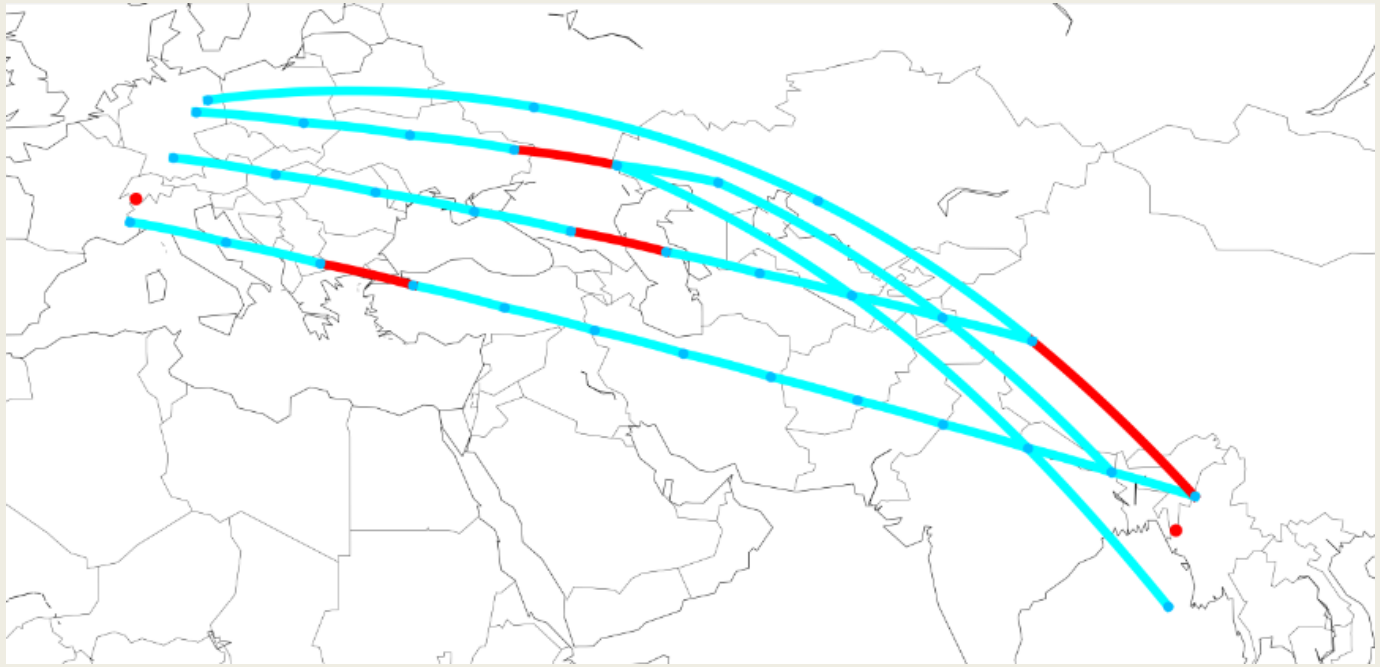
$$\vec{x^*} = \arg\min_{\mathbf{ATM}} \|\mathcal{S}(\mathbf{ATM})\|_1$$

$$\text{s.t.} \qquad \forall\, l \in L,\ \ \mathcal{R}(l) \geq 0,$$

$$\mathcal{C}_P(\mathbf{ATM}) = 1$$

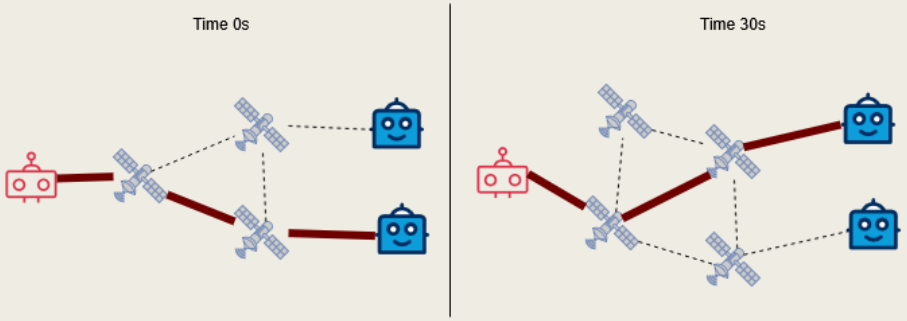
Results: HYDRA vs. ICARUS in a Single Snapshot

- HYDRA reduces required botnet size by **~50%** compared to ICARUS
- Achieves the same disruption with **fewer bots and lower total traffic**
- Slight increase in per-bot uplink bandwidth (e.g., +120 Mbps maxUp)
- Demonstrates **cost-efficiency** and **precision** of HYDRA's optimization



Strategy 2 — C-BMP:
Continuous Disruption Over Time

- Satellite topology changes constantly ~ we captured it every 30 seconds ⟶ paths changes
- Goal: **Maintain disruption across many time snapshots**
- HYDRA selects a **shared botnet** that works across time
- Minimizes total botnet size while ensuring attack success in each snapshot
- Captures temporal structure of the network and shifting bottlenecks

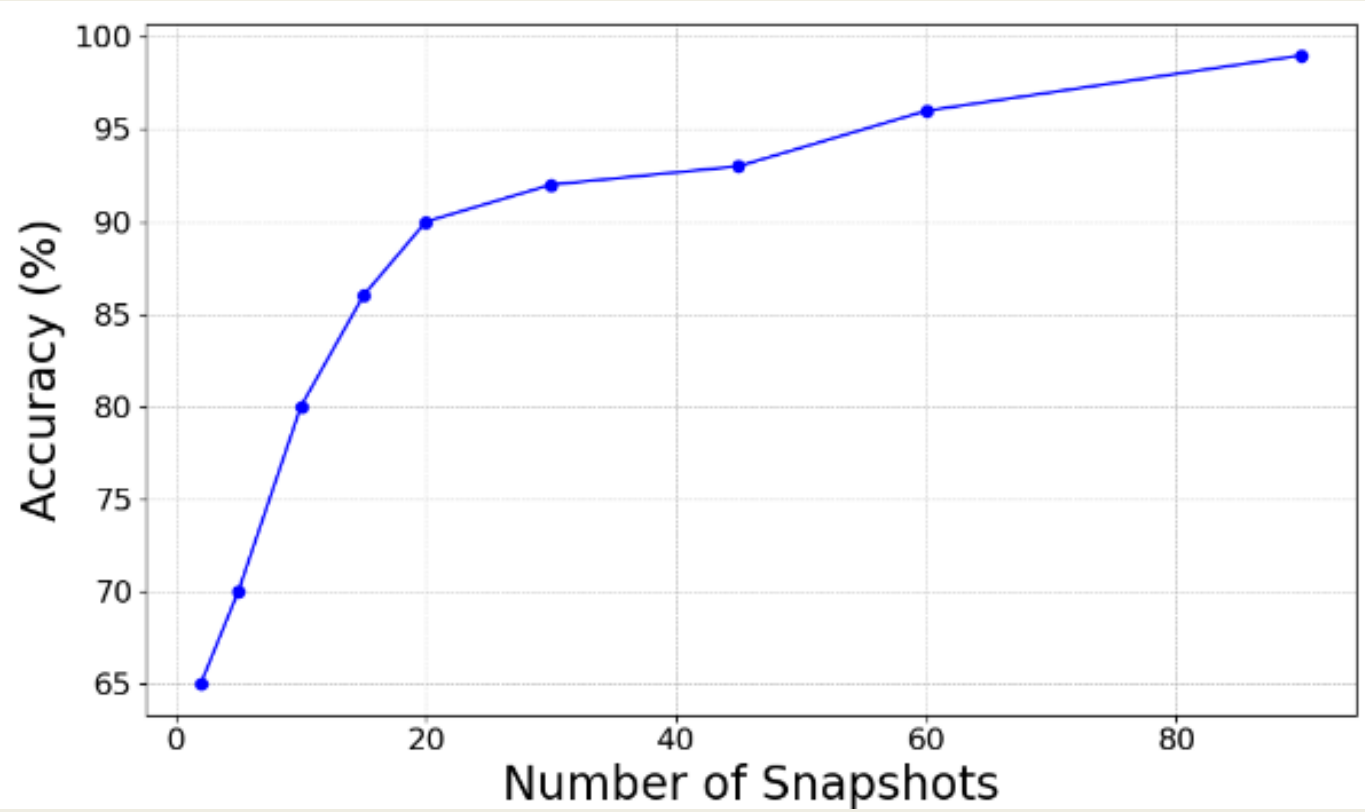


$$\vec{x}^* = \underset{\{\mathbf{ATM_t}\}_{t\in T}}{\arg\min} \ \|\mathcal{S}_{\max}(\mathrm{ATM}_T)\|_1$$

$$\text{s.t.} \quad \forall t \in T, \ \forall l \in L_t, \ \mathcal{R}_t(l) \geq 0,$$
$$\forall t \in T, \ \mathcal{C}_{P_t}(\mathbf{ATM}) = 1$$

Results: HYDRA C-BMP — Sustained Disruption Over Time

- Evaluated over **90 minutes** using snapshots every 30 seconds
- HYDRA maintains **~97% success rate** across the full window
- Only a **~2% drop in effectiveness** over longer durations
- Uses a **single, shared botnet** across all snapshots
- Shows tradeoff: **more snapshots = better accuracy, slower planning**



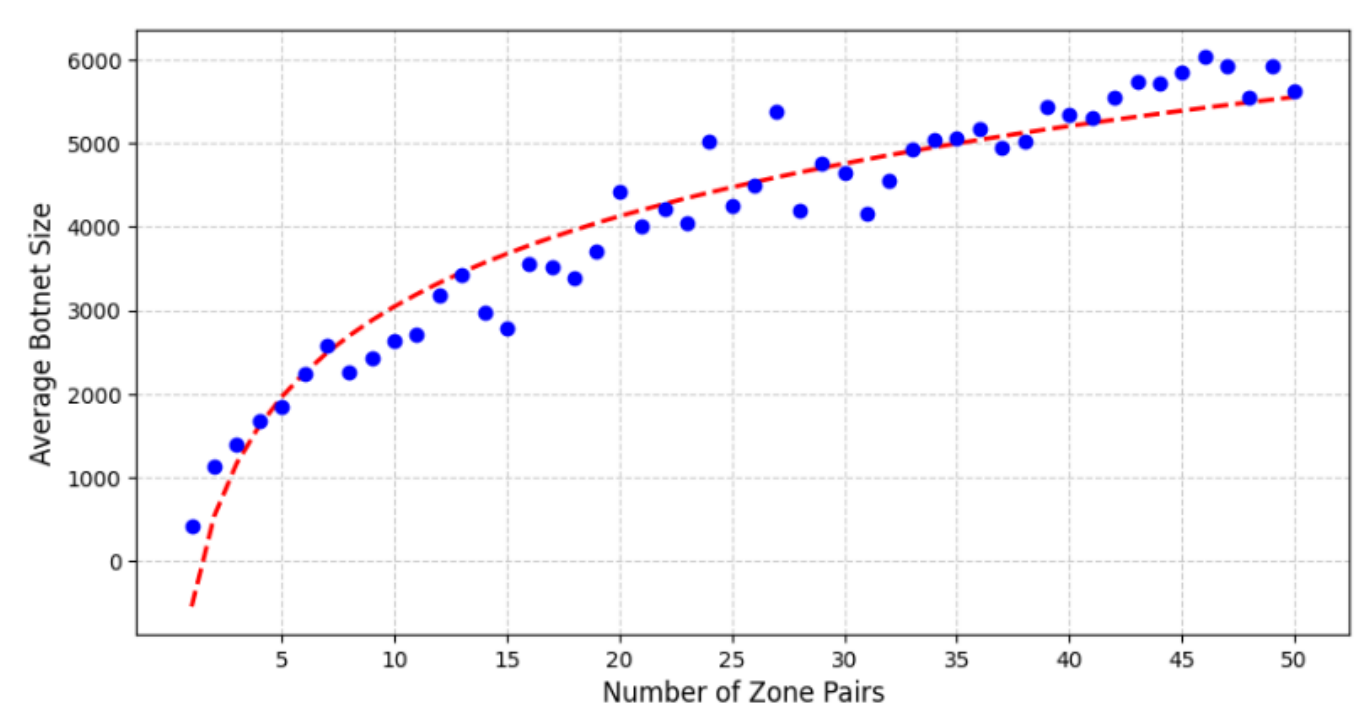
Strategy 3 — F-BMP:
Flexible Botnet for Multi-Target Adaptability

- Design a **single botnet** that can attack **multiple zone pairs**
- Bots are selected for their **coverage and impact across targets**
- HYDRA finds overlapping paths to **maximize reuse**
- Attack success remains high across thousands of random target pairs
- Enables **on-demand disruption** with minimal exposure

$$\vec{x}^* = \underset{\{\mathbf{ATM_a}\}_{a\in A}}{\arg\min} \; \|\mathcal{S}_{\max}(\mathrm{ATM}_A)\|_1$$

$$\text{s.t.} \qquad \forall a \in A, \;\; \forall l \in L_a, \;\; \mathcal{R}_a(l) \geq 0,$$
$$\forall a \in A, \;\; \mathcal{C}_{P_a}(\mathbf{ATM}_a) = 1$$

Flexible Botnet Results: Scaling Across Zones



- HYDRA creates a **single botnet** that can be reused to attack **many different zone pairs**
- The growth trend is **logarithmic** — attackers can scale reach **without scaling exposure**
- Efficient overlap: many zone pairs share **paths and links**, enabling bot reuse
- Demonstrates HYDRA's ability to maintain **attack flexibility** with **minimal resource expansion**

Example: Botnet Optimized for Global Flexibility

- HYDRA selects **bot locations** to maximize path overlap across many zone pairs
- Optimized using 48 zone pairs; tested on 3,500 new pairs
- Most bots are located in **high-coverage regions** with wide satellite visibility
- Achieves **85% attack success rate** with this fixed botnet
- Demonstrates attacker efficiency using **shared infrastructure**



Strategy 4 — SIMU-BMP:
Simultaneous Multi-Zone Disruption

- Goal: Attack **multiple zone pairs at the same time**
- HYDRA identifies **shared links** across targets → bottleneck overlap
- Optimizes one botnet to congest all target paths simultaneously
- Reduces total botnet size by **up to 50%** vs. solving each pair separately
- Enables **broad impact with minimal resource growth**

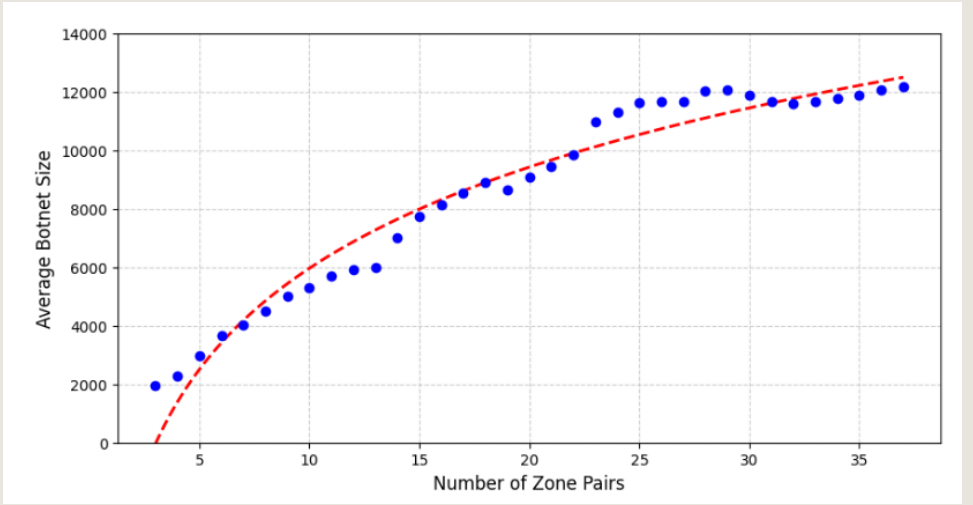
$$\vec{x^*} = \arg\min_{\mathbf{ATM}} \|\mathcal{S}(\mathbf{ATM})\|_1$$

$$\text{s.t.} \qquad \forall\, l \in L,\ \ \mathcal{R}(l) \geq 0,$$

$$\mathcal{C}_{P_A}(\mathbf{ATM}) = 1$$

Results: SIMU-BMP — Coordinated Attacks with Shared Resources

- Evaluated on **simultaneous attacks across multiple zone pairs**
- HYDRA achieves full disruption with **50% fewer bots** than solving independently
- Overlapping paths between zone pairs allow the same bots to support multiple attacks simultaneously.
- Optimization exploits **structure in the network**, not brute force
- Demonstrates highest **impact-to-resource ratio** of all four strategies



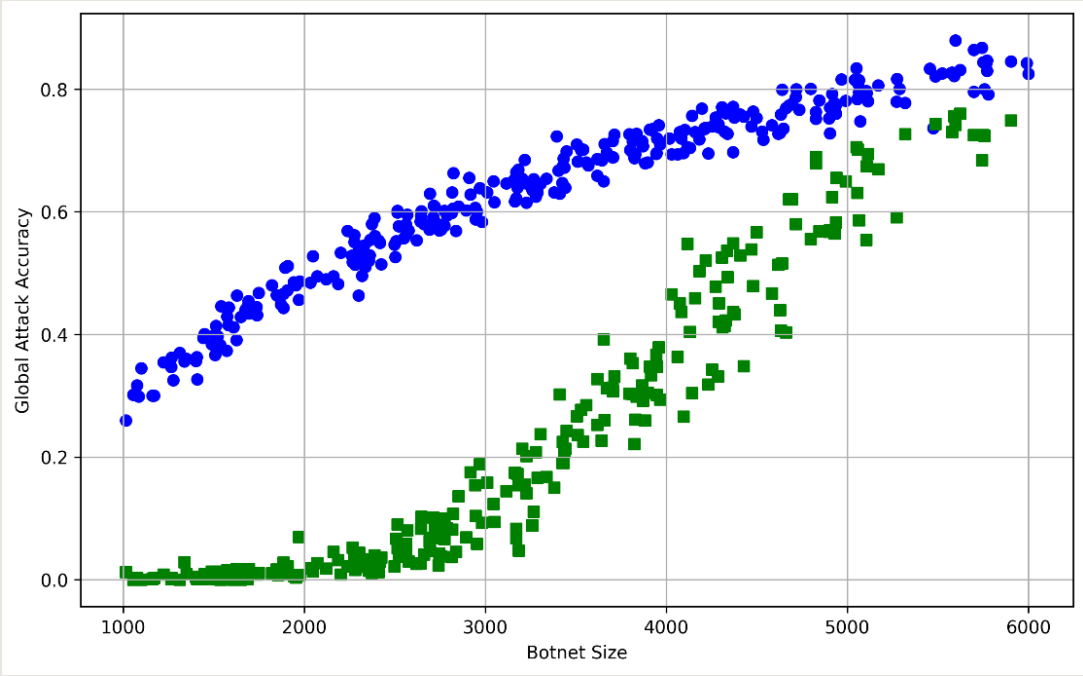
Recap — Four Optimized Attack Strategies

- **Strategy 1 — SS-BMP:** Snapshot Disruption with Minimal Bots
- **Strategy 2 — C-BMP:** Continuous Attack Over Time
- **Strategy 3 — F-BMP:** Flexible Botnet for Multi-Target Adaptability
- **Strategy 4 — SIMU-BMP:** Simultaneous Multi-Zone Disruption

All strategies share the same core optimization model

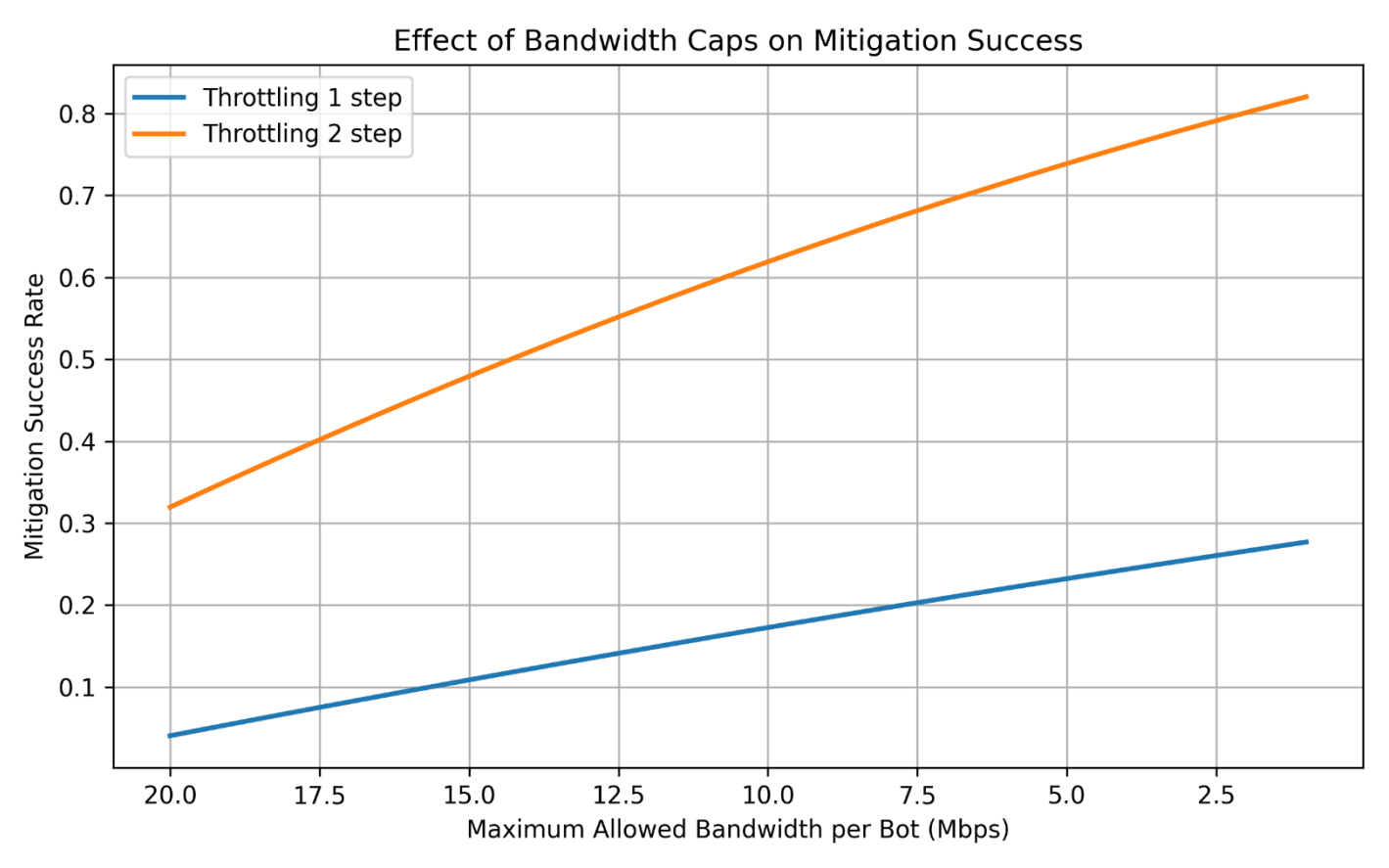
Adaptive Routing as a Mitigation Strategy

- Link-flooding attacks rely on **predictable routing paths** to target critical links with precision.
- Adaptive routing introduces **path variability**, making it harder for attackers to predict and plan effective congestion.
- This increases the **cost and complexity** of executing LFAs, particularly in time-sensitive or multi-zone attacks.
- However, when structural bottlenecks persist across routing configurations, adaptive routing alone may be insufficient.



Incremental Bandwidth Throttling Improves Mitigation

- **Step 1:** Throttle all senders observed on congested links → **~30% mitigation success**
- **Step 2:** Also throttle senders that appeared later congested links
- **Result:** Cumulative throttling improves mitigation to **over 80%**
- Achieved **without blocking** or botnet detection





Conclusion:
Small Botnets, Big Disruption

- HYDRA is a new framework for planning **link-flooding attacks** in satellite networks
- Designed to disrupt **zone-to-zone communication over time** using a **minimal botnet**
- Supports **flexible, continuous, and simultaneous** attack strategies
- Achieves **persistent, scalable disruption** with a compact, shared botnet
- Highlights real vulnerabilities in satellite routing — and how **optimization can amplify them**

Q & A

Thank you

Roee Idan
Cybersecurity Researcher