
# ICARUS: Attacking low Earth orbit satellite networks

Giacomo Giuliari, Tommaso Ciussani, Adrian Perrig, and Ankit Singla, *ETH Zurich*

https://www.usenix.org/conference/atc21/presentation/giuliari

This paper is included in the Proceedings of the
2021 USENIX Annual Technical Conference.

July 14–16, 2021

978-1-939133-23-6

Open access to the Proceedings of the
2021 USENIX Annual Technical Conference
is sponsored by USENIX.

# ICARUS: Attacking low Earth orbit satellite networks

Giacomo Giuliari, Tommaso Ciussani, Adrian Perrig, Ankit Singla
*Department of Computer Science, ETH Zürich*

## Abstract

Internet service based on low Earth orbit satellites is generating immense excitement in the networking community due to its potential for global low-latency connectivity. Despite the promise of LEO satellite networks, the security of their operation has so far been largely neglected. In this context, we present ICARUS, a new class of denial of service attacks on LEO networks. ICARUS turns these networks' key benefits into vulnerabilities: an adversary can leverage the direct global accessibility to launch an attack from numerous locations, while the quest for low latency constrains routing, and provides predictability to the adversary. We explore how the adversary can exploit other unique features, including the path structure of such networks, and the public knowledge of the locations and connectivity of the satellite-routers. We find that a small amount of attack bandwidth can hamper communications between large terrestrial areas. Finally, we lay out open problems in this direction, and provide a framework to enable further research on attacks and defenses in this context.

## 1 Introduction

SpaceX Starlink [60,61], Amazon Kuiper [35], and others are deploying hundreds to thousands of satellites, each carrying high-capacity networking equipment. These low Earth orbit (LEO) satellite networks (LSNs) aim to provide global broadband Internet service. While global access would itself be a huge leap in connectivity, these networks also promise low-latency communications between otherwise well-connected regions. For instance, recent work [9] shows that for long-distance communication beyond a few thousand kilometers, such networks could provide lower latency than not only today's fiber Internet, but also specialized free-space radio networking used in the high-frequency trading industry.

Driven by the exciting potential of global low-latency networking, researchers are exploring many interesting problems related to LSNs, such as estimating their latency improvements [9, 28], and network topology design [10] and new routing strategies [25, 29, 33].

However, prior work has thus far largely ignored the aspect of the secure operation of these networks in the face of adversaries. We hence explore the resilience of LSNs to a large-scale volumetric distributed denial-of-service (DDoS) attack, carried out by a botnet of compromised satellite-enabled hosts. This threat has been extensively studied in terrestrial networks, but as we detail later (§ 3.4), an LSN is an extremely different environment: an adversary can leverage its global footprint to flexibly inject traffic into the network; knowledge about node (satellite) locations and network structure, far from being closely guarded, is easily available public information; and the low-latency objective both limits path diversity and reduces uncertainty about routing for the adversary.

With these unique opportunities for the adversary, also come new challenges: unlike a terrestrial transit network, the LSN connects directly to the clients injecting traffic into it, and can more easily detect and stop malicious behavior; at least initially, the client population will be much smaller than the Internet, limiting the adversary's resources; and the sparse structure of the inter-satellite network creates a risk for attack traffic flows to congest each other instead of the target(s).

Leveraging the unique opportunities and addressing the new challenges, we present ICARUS, a DDoS attack on LSNs. ICARUS draws on Coremelt [64] and Crossfire [31], and other DDoS work, but is customized for LSNs — the adversary carefully plans attack traffic using an LSN's structure, with the objective of congesting a target link or set of target links. The adversary hopes to do so at low *cost*, in terms of attack traffic volume, and with low *detectability*, in terms of change in the ingress bandwidth at individual satellites.

To help develop intuition in this new setting, we first analyze ICARUS in the scenario where the LSN uses single shortest-path routing. By simulating the attack on the LSN with the largest deployment to date, Starlink, we find that the adversary can successfully target the majority of its links, congesting both ground-to-satellite links and inter-satellite links (ISLs). Somewhat surprisingly, multi-target attacks that hamper connectivity between large regions are also feasible, at somewhat higher cost, but *without* increasing detectability compared to the single-target case.

In practice, LSNs may leverage multipath routing, potentially with randomized load balancing. We thus introduce a probabilistic version of ICARUS, that allows them to congest the target(s) with high probability, despite the uncertainty of routing. We analyze this attack against four intuitive routing schemes. Unfortunately, our experiments show that ICARUS is still able to attack targets at some additional cost compared to the single shortest-path scenario. The routing scheme that is most effective at increasing the attack cost—by 385% for the median target, *if* the adversary desires to minimize risk of detection—incurs a large latency increase on its paths, 1.32× in the median and up to 2.04× in more extreme cases. Despite this latency inflation, in the absolute, this increased attack cost still translates to less than 80 Gbps of attack traffic, and with a few tens of Mbps of bandwidth per bot, as promised by LSNs, translates to a few thousand bots. This is also the most pessimistic scenario for the attacker: there is no benign network traffic to lower the bar for congesting targets, the

LSN is willing to accept a large latency degradation, and the attacker seeks to minimize detectability rather than cost. In many settings, a few hundred bots will suffice instead.

Lastly, we briefly analyze additional attack opportunities that may arise from the dynamic nature of LSNs — as satellites move and paths change, there is natural flux in both the latencies of end-end paths, and in the traffic carried by each ISL. An adversary could potentially leverage latency changes to double the bandwidth of attack flows for a short period, in a manner similar to temporal lensing [55]. However, early results indicate little additional utility from this seemingly promising method. Further, an adversary may also time their attack flows to coincide with natural surges in link utilization from path changes. Our preliminary analysis indicates promise for this latter strategy, but a packet-level analysis is necessary to ascertain its impact.

Our main contributions are:

- We draw out the unique features of LSNs that introduce a vulnerability to DDoS, motivating a new analysis of a problem well-studied in terrestrial networks.
- We present the ICARUS attack, which exploits these unique features to successfully congest target links in an LSN. ICARUS addresses the new challenges the LSN setting poses for the adversary in terms of cost and detectability.
- We use the simple single shortest-path routing setting to develop intuition, and to show how connectivity between large regions may be hampered by ICARUS.
- We extend ICARUS to a randomized multipath routing scenario, and evaluate it against four intuitive routing schemes, showing that the attack still succeeds with high probability.

To the best of our knowledge, ICARUS is the first volumetric DDoS attack on upcoming LSNs. Beyond our particular attack methods and results, we hope that our analysis framework will aid further work on the vulnerability of LSNs to DDoS attacks, and on developing and evaluating mitigation strategies, e.g., routing that optimizes the latency-resilience tradeoff. Our framework is available on GitHub.

## 2 Background & related work

Our contributions lie at the intersection of the nascent research on large upcoming LSNs, and well-studied (for terrestrial networks) denial-of-service (DoS) attacks. We introduce the relevant background for both.

### 2.1 LEO satellite networks

In the proposed and under-deployment LSNs, hundreds to thousands of satellites will be arranged in equally-spaced *orbital planes*, each containing the same number of satellites. Satellites connect to terminal units (TUs) on the ground with radio links. At the same time, satellites may connect to each other by laser ISLs. Each orbit has a fixed *height* above sea level; under 2000 km is considered LEO. This relative closeness to the ground provides a great advantage in terms of communication latency, as the signal has to travel a much

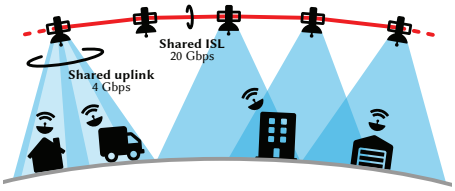



**Figure 1:** ***Satellite forwarding.*** *Satellite-enabled hosts communicate with the satellites overhead using radio up- and downlinks. Traffic is routed from satellite to satellite via ISLs.*

shorter distance to reach the satellite than, for example, to reach Geostationary satellites at 35,786 km.

In most proposed constellations, the orbital planes do not intersect the Equatorial plane at a 90° angle, and thus do not transit over the Earth's poles. Instead, a lower *inclination angle* is used to allow the satellites to spend more time at lower latitudes, thus improving coverage above more populous areas, whilst avoiding the polar regions.

Our analysis throughout uses the first shell of the SpaceX Starlink constellation, for which the above parameters are: 72 orbital planes, each with 22 satellites, a height of 550 km, and an orbital inclination of 53° [60, 61, 63].

**Satellite–ground connectivity**. Figure 1 shows a typical LSN communication scenario. Terminal units (TUs) are installed at the clients on the ground. Each satellite can simultaneously connect with multiple TUs, thanks to their multi-beam antennas [17]. We call the aggregate capacity incoming to and outgoing from a satellite its *uplink* and *downlink* respectively. Based on recent FCC filings [60, 61], our analysis assumes both up- and downlink capacities to be 4 Gbps.

TUs can communicate with overhead satellites that are above a *minimum elevation angle* over the horizon. The angle varies from constellation to constellation, and expresses a trade-off between the size of the coverage area or *footprint* of a satellite, and the rate of communication achievable. Greater elevation angles translate to smaller footprints, but also force shorter paths inside the atmosphere and allow for better frequency reuse, thus increasing capacity. For Starlink's shell I, the minimum elevation angle is planned to be 40°.

**Inter-satellite connectivity**. A key feature of the proposed constellations is the use of laser inter-satellite links (ISLs) on each satellite.[1] A typical approach is to have 4 ISLs per satellite, two connected to fore and aft satellites in the same orbital plane, and two to satellites in adjacent orbital planes [69]. The resulting ISL topology is used to transit traffic between TUs. As shown in Fig. 1, traffic can be forwarded from a TU to a satellite in view through an uplink, then across multiple ISLs, and finally to the destination via a downlink.

Some constellations, particularly during early stages of deployment, may forego ISLs, instead transiting data through a series of up and down transmissions between ground stations

[1] SpaceX recently launched ISL-capable satellites [59], and Telesat has started producing them [65].

and satellites; this is, for instance, the case for Starlink in its first stages of deployment. However, ISLs are expected to be a key component of future LSNs, including more mature deployments of Starlink, and are crucial to achieving many of their benefits [10, 28, 29]. We therefore focus our analysis on the setting with ISLs; no-ISL LSNs are a strictly easier attack target: ground–satellite links will have a much lower capacity than ISLs, with each satellite supporting a few Gbps of uplink at most, simplifying the adversary's goal of creating congestion. On the other hand, previous work [17] and laser equipment vendors' offerings [47] suggest that ISLs will be able transfer up to 20 Gbps full-duplex, greatly increasing the capacity of the network.

**Benefits: global low-latency broadband**. With their hundreds to thousands of satellites, each providing multi-gigabit connectivity, LSNs can blanket the globe with broadband Internet. Besides their potential to extend the Internet's coverage to under-served regions, LSNs are also generating excitement due to their potential for low-latency connectivity even in already well-connected areas. Prior work [10] estimates that LSNs could reduce latency for long-distance routes by more than 2× compared to today's Internet. Two factors contribute to this reduced latency: first, the speed of light in vacuum is roughly 50% higher than the speed of light in the glass medium of optical fiber. Second, a series of ISLs can often provide a more direct path between TUs than what fiber can achieve on the ground.

## 2.2 Denial of service

The term *denial of service* encompasses a large class of attacks. DoS attacks can target different resources at different layers in the network. We are interested in discovering and analyzing the threats to communication that are peculiar to LSNs, and that arise from the characteristics we have presented in the previous section. During a volumetric DDoS attack, a *botmaster* directs the traffic of thousands to millions of *bots* [3], i.e., compromised hosts, towards a target element in the network, overwhelming it. The *botnet* is usually composed of bots distributed across the globe; and targets can be servers, end-hosts, routers, etc. Traditional DDoS attacks send traffic to a target end-point. Newer attacks like Coremelt and Crossfire are designed to congest in-network elements instead—their approach to send traffic between bots or to initiate connections to a variety of public servers defies traditional methods to classify traffic into legitimate and unwanted.

**Coremelt & Crossfire**. In Coremelt [64], the adversary tries to congest a target inter-domain *link* by making use of legitimate-looking flows generated between the $N$ bots of a botnet. Given the knowledge of the $\binom{N}{2}$ paths between the bots, the adversary can initiate flows that will cross the target link. With sufficient resources, the adversary can successfully congest any link in the network.

Crossfire [31] expands on this idea. With Crossfire, the adversary is able to hinder communication between entire regions of a network, as bots simultaneously connect to strategically selected servers, thus creating overload on the links leading into the region under attack.

Both attacks have been notoriously hard to mitigate, since the adversary uses indistinguishable, legitimate traffic. In § 7 we argue that ICARUS poses an even larger threat, as the few mitigations available against Coremelt and Crossfire cannot be directly employed.

# 3 ICARUS adversary model

The ICARUS adversary controls a botnet of compromised hosts equipped with TUs that connect to the LSN. These bots are exploited by the adversary to send traffic over the LSN to congest links and disrupt communications between other TUs controlled by benign users.

## 3.1 The adversary's objective

The adversary seeks to generate legitimate-looking traffic that, by careful selection of source and destination bots, disrupts target communications by exhausting the capacity of certain LSN links. More concretely, the adversary may target:

- *A satellite uplink or downlink*. This is the easiest attack, as these are the lowest bandwidth links in an LSN.
- *An ISL*. Congesting ISLs is a potentially more disruptive attack, as they carry traffic for more source-destination pairs. However, they are also more difficult to attack than uplinks and downlinks, as they have higher capacity.
- *Multiple links*. We also consider multi-link attacks, in which the adversary selects a combination of target ISLs and/or up/downlinks to achieve a broader attack effect. For example, the adversary could try to congest *all* the links in the load-balancing set for a target source-destination communication, or to target all links connecting two large terrestrial areas.

Our simulations label a link *congested* when aggregate traffic demand across it exceeds its capacity. In practice, degradation of communication starts before this threshold.

## 3.2 Metrics of attack success

For any of the above targets, the adversary is interested in causing disruption at *low cost* and while *avoiding detection*.

**Cost**. Attack cost measures the resources an adversary has to deploy to successfully achieve their attack's goals. We express cost as the traffic volume (in Gbps) the adversary has to generate to be successful. With each bot generating tens of Mbps of attack traffic, as seen in initial measurements from Starlink [5, 11], an ISL's 20 Gbps worth of capacity translates to, e.g., 500 bots each sending 40 Mbps.

**Detectability and maxUp**. If the LSN operator sees large changes in how much traffic a certain satellite receives on its uplink, the operator could potentially localize and identify the compromised bots. Thus, to reduce detectability, the adversary would distribute attack traffic across numerous uplinks. To characterize detectability, we use the *maximum absolute*

*bandwidth increase* caused by the attack traffic across the uplinks of the LSN (maxUp). We use the absolute increase, rather than relative, to obtain comparable results from simulations with different baseline traffic models.

### 3.3 The adversary's capabilities & constraints

Unlike terrestrial networks, many aspects of an LSN's operations are public knowledge or can be inferred.

**LSN topology**. Satellite orbits are public-domain information constantly updated and published by NORAD [12]. From these, an adversary can precisely compute the positions of the satellites for a chosen attack time; the error in such computations is at most a few kilometers per day into the future [32]. As noted earlier (§2), the ISL interconnect is expected to be a typical cross pattern. The design capacity of all links is also known through public regulatory filings. Thus, we assume full advance knowledge of the LSN topology. Moreover, changes in the shortest paths happen on timescales of seconds. As the maximum forwarding latency in the network is about 125 ms, the adversary can look at the LSN in snapshots [10], and compute the attacks for successions of snapshots. They can thus avoid the complexity of continuous satellite motion.

**Routing**. The topology is known, as is the propagation delay across each link at all times. Therefore, given frequent measurements of latency between a large set of bots, the adversary can determine how routes are being chosen. This is a much simpler setting for network tomography than one where the topology and forwarding latencies are unknown.

If routing is deterministic, e.g., shortest-path routing, the adversary can compute the full forwarding path for any source-destination pair ahead of time. If routing is randomized, e.g., for load balancing across near-shortest paths, we assume the adversary knows the algorithm with which the *load-balancing set* is constructed. The adversary therefore knows in advance the possible paths traffic will take, but not the actual path selected for forwarding a particular flow.

**Bot locations & availability**. A key promise of LSNs is global coverage, thus providing diverse locations for potential bots. Even if terminals themselves are secure, it suffices to compromise the user devices connected through them, and therefore exploits used today to create botnets directly apply in the case of a satellite-enabled botnet. We thus assume that the adversary can compromise TUs at any location on land. Given that such TUs, even today [66], feature GNSS[2]-based self-location to aid signal acquisition from satellites, the adversary accurately knows the locations of their bots.

Regarding the size of the botnet, we do not constrain the number of compromised TUs at any location beyond the limits imposed by the capacity of uplinks. We will later show that successful attacks only require hundreds to thousands of compromised TUs—a very small fraction of the tens of millions of TUs expected to be available globally (SpaceX has sought permits for 5 million TUs in the US alone [62]).

[2]Global Navigation Satellite System.

**Attack control channel**. We assume that the adversary can coordinate the bot army through a command-and-control channel, which is typical for modern botnets. As we show later, the advance knowledge of the topology, the routing algorithm, and the bot locations, allows the adversary to pre-compute and disseminate to the bots their attack traffic commands ahead of time. Fine-grained time-synchronization across bots would only be needed if the adversary sought to create extremely short-term congestion (e.g., for tens of milliseconds) [36, 55]. This would easily be possible through GNSS.

### 3.4 Unique attack opportunities & challenges

The above discussion highlights that an adversary targeting LSNs has several advantages compared to one targeting traditional terrestrial networks.

- An adversary has full access to the network topology. The positions of the satellites are published regularly [12], and can be predicted with high accuracy into the future [32]. The design detail of the interconnections, moreover, can be deduced from the FCC requests that the satellite operators are mandated to file. In contrast, the topology of terrestrial networks is concealed, and may even be obfuscated [44].
- Given the LSN's global exposure, the adversary can recruit bots in diverse locations for generating attack traffic.
- The low-latency goal of LSNs leads to routing predictability, as shortest or near-shortest paths must be used.
- For already well-connected regions, the primary value of an LSN is low latency, as terrestrial fiber routes will be cheaper. This lowers an adversary's burden: between a target source-destination pair, the adversary must only deteriorate a small set of desirable (low-latency) paths, instead of needing to congest a cut in the network graph.
- Our analysis of the topology and path structure of an LSN shows that certain links are more vulnerable than others, providing easy targets for an adversary.
- Exploiting the system's predictability, communication with the bots can be asynchronous, making traditional detection of the command and control traffic difficult.

However, some aspects of the LSN setting also pose challenges to the adversary:

- Each bot has limited resources, and the pool of bots available is only a small fraction of Internet-connected hosts that could otherwise be purposed as bots. It is thus even more crucial for an adversary to keep attack cost low.
- The adversary needs to ensure that their limited attack resources are not wasted by self-congestion, whereby traffic from its bots congests links before reaching the target links.
- Congesting some satellite's uplink exploiting a large number of bots directly underneath is counter to the adversary's need to avoid detection. The LSN is a centrally-managed, intra-domain network, and therefore has better monitoring and policing capabilities than terrestrial transit networks.

Thus, the adversary has to be especially stealthy, and employ more distant and diffuse collections of bots.

- If the LSN uses randomized load balancing across multiple paths, this results in routing unpredictability for the adversary, requiring a probabilistic approach that would potentially consume more resources.

Our attack methods exploit the aforementioned unique opportunities, while addressing the new challenges.

# 4 ICARUS attack: shortest-path routing

We first discuss our ICARUS denial-of-service attack against an LSN in the simpler setting, where the LSN uses single shortest-path routing. While this provides low latency, the deterministic routing aids the adversary.

## 4.1 Attack mechanism

Paths in an LSN are typically stable on the order of seconds (§ 3.3). Thus, the adversary can compute the flows the bots must send on a static snapshot of the network, and use the same flow assignment for seconds before issuing a new one. Note that the adversary is also aware of the times when paths will change, and can plan their flow assignments for change-free periods. The assignments can be sent asynchronously to the bots for several periods in advance, as the entire system is fully predictable on a much longer timescale. For a particular system snapshot, the set of attack flows is computed as follows, similarly to Crossfire [31]:

**A1** *Link and path discovery.* The adversary uses the known inter-satellite topology to create a connectivity graph. They compute all the satellite in view of each bot, and add an uplink and a downlink for each reachable satellite to the connectivity graph. Finally, the adversary runs Dijkstra's algorithm for all the $N(N-1)/2$ pairs of bots to find the forwarding paths.

**A2** *Path filtering.* For a given target link or links, the adversary finds all bot-pairs whose shortest path traverses the target(s). The adversary retains only the paths that traverse the target(s) in the desired (attacked) direction.

**A3a** *Feasible attack flow computation.* The adversary must decide how much traffic to send on the chosen attack paths between its bot-pairs. The adversary checks if there exists a flow assignment such that (i) the target links are just above capacity, and (ii) no other links are congested. Satisfying these constraints ensures that the target(s) can be congested without the attack traffic self-congesting, i.e., being bottlenecked before reaching them. This approach minimizes cost in terms of attack bandwidth.

**A3b** *Reducing maxUp.* If **A3a** produces a feasible attack flow, the adversary iteratively optimizes maxUp. Recall that maxUp, $D$, is the maximum attack traffic volume across the uplinks. At each iteration $i$, the adversary lowers the permissible attack traffic volume on all uplinks, $D_i$; then they perform the feasibility check again. This process is repeated until the minimum value, $D_{min}$, is found, for which the attack is still feasible.

## 4.2 Evaluation setup

We evaluate the above attack strategy is terms of how its cost and maxUp depend on the attack scenario. The attack scenario varies the targets, i.e., either individual links or sets of links, and what type of benign traffic already uses the LSN.

Throughout, we use simulations on the SpaceX Starlink shell I constellation (§2) with a total of 1584 satellites. Note however, that our evaluation framework supports arbitrary constellations; an example set of results for a different configuration is included in §D in the Supplemental materials. We assume that bots and benign TUs can be located at any point on land. To keep the simulations tractable, we discretize the continuous space of geolocations to a regular geodesic grid. This is done in the form of a triangular tiling of the planet that, when restricted to land areas, amounts to a total of 1761 possible positions. (The restriction to land areas is driven by our below models of benign traffic, which are based on population and economic activity.) Each of the triangular tiles represents an area of 100,000 km$^2$. Since a single Starlink satellite covers an area ten times this size, this grid suffices to capture the nuances of different TU placement relative to the same satellite.

We test three traffic scenarios for benign traffic on the LSN:

- *Empty network:* This case, with no benign traffic, serves as a baseline, comparing against which, we can understand the impact of existing traffic.
- *GDP traffic:* A "gravity" sampling model, where probability of adding 10 Mbps of benign traffic between two locations is proportional to the product of their Gross Domestic Product (GDP) [48]. Each tile's centroid serves as a traffic source/sink location, with the aggregate GDP for the tile's area being this location's GDP weight. We then sample 250k times, discarding the samples that, if added, would exceed 90% of the capacity on the links. This process results in a utilization of 28% of the on-land uplink capacity. This model treats traffic as proportional to economic activity, representing the view that the biggest economic centers will drive traffic and revenue.
- *Pop traffic:* Another gravity model similar to the above, except the population is used as the weight instead of GDP [13]. This model represents the scenario where the LSN's primary role is improving coverage in regions that are populous but still poorly connected. For this model, we get 34% utilization.

All the simulations in the following are executed by our LSN simulation framework (~5000 lines of Python code).

## 4.3 Single link target scenarios

**Attacking an uplink.** The outcome of the attack on an uplink is easy to determine: either the adversary has enough bots in the area of coverage of the victim satellite uplink, or

the attack cannot succeed. This attack has therefore maximum maxUp in our model, as the adversary has to completely fill the uplink with attack traffic.

**Downlinks**. Unlike uplinks, downlinks can be reached from arbitrarily afar, and therefore the adversary can more easily find bots to congest them. In theory, the attack flows can self-congest: as the attack traffic flows towards the target downlink, more and more flows join paths, filling the ISLs leading up to the target. While this can violate constraint ii) in **A3a** (§ 4.1), it is unlikely for an attack on a downlink — all the ISLs preceding the target downlink can carry 5× the bandwidth needed to congest the downlink. As a result, our simulations reveal that *all* downlinks from satellites located over land are attackable in this manner. As our model does not allow TUs in the seas, there is no way to congest downlinks for the remaining satellites. We also verified that this result remains the same regardless of our three traffic models, as even with benign traffic, there is always enough leftover capacity in the network for the relatively small amount of attack traffic bandwidth required.

**ISLs**. ISLs have higher bandwidth than up-/down-links, making an attack more complicated and more costly to achieve. Not only is more attack traffic needed to cause congestion on an ISL compared to a downlink, but also self-congestion *does* often occur in practice for attack flows headed to a target ISL. This often results in there being no feasible attack flow. Consequently, of the 6336 possible ISL target links (1584 satellites, each with 4 ISLs, counting both directions), we find that in an empty network scenario, 5470 (86%) are feasible. With GDP or Pop traffic, this drops to 72% and 71% respectively. This is along expected lines: benign traffic reduces the available capacity on the attack flow paths, limiting the attack traffic that reaches the target.

An interesting finding from our simulations is that with all three models, of the end-to-end paths with at least one ISL, fewer than 0.6% do not contain at least one ISL the adversary can congest. Therefore, this attack can disrupt >99.4% of all communications that rely on an ISL.

**Cost and maxUp**. In the above, we have only discussed feasibility for each individual targeted link: for uplinks, feasibility depends simply on having enough bots under the target uplink; all downlinks over land are feasible to attack; and for ISLs, self-congestion limits feasibility, with the details dependent on the benign traffic. For the feasible attacks, we can also analyze attack cost and maxUp.

Figure 2 shows the cost and maxUp of attacks against all links that are feasible to attack (downlinks and ISLs together; uplinks are uninteresting as discussed, and thus omitted) for all three traffic models. From the cost graphic (top), we see that for the empty network, there are only two modes in the CDF — all downlinks cost 0.2 and all ISLs cost 1 to attack. (We present costs normalized to one ISL's bandwidth, i.e., 20 Gbps; recall that downlinks have one-fifth the capacity of an ISL.) This is as expected: in an otherwise empty network,

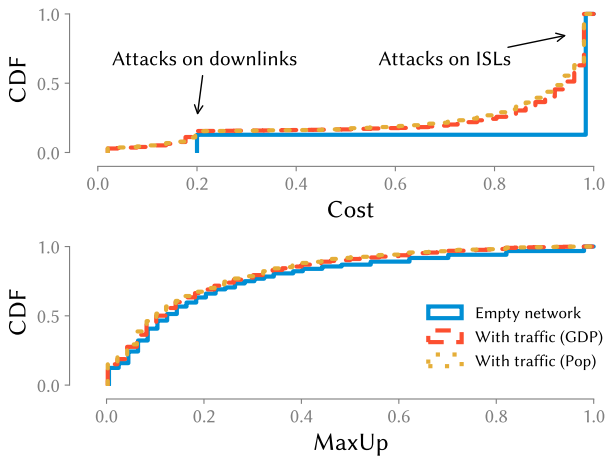



**Figure 2:** ***Cost and maxUp of single-target attacks.*** *While maxUp is similarly distributed either with or without baseline traffic, attack cost decreases significantly.*

congesting any link requires sending the target link's capacity worth of traffic. In the presence of benign traffic, for both traffic models, the attack cost is lower, as the adversary needs to add a smaller amount of traffic to the existing benign traffic. In terms of number of bots, a cost of 1 translates to a few hundred bots sending tens of Mbps each (§ 3.2). The maxUp plot (Fig. 2, below) shows the distribution of maxUp across attacks on different target links. We normalize maxUp to one uplink's bandwidth, such that needing to fully saturate a satellite's uplink is the worst case for the adversary, and means that maxUp is 1. The maxUp of an attack on the median target link is below 0.13 in the empty network scenario. Thus, the *maximum* satellite uplink bandwidth consumed across attack flows sent towards these target links comprises roughly one-eighth of an uplink's bandwidth. Adding benign traffic to the network with either traffic model further lowers the adversary's risk of detection.

While adding benign traffic decreases attack cost and lowers maxUp for those target links that are feasible to attack, it actually decreases the fraction of feasible target links. However, closer inspection shows that this is not a particularly severe problem for the adversary: the links that become less vulnerable when benign traffic is present are mostly above the oceans, and not the higher-value ones over the more populous regions (further analysis is presented in §A.1 in the Supplemental materials).

## 4.4 Multi-link region disconnection scenarios

Beyond attacking individual links and communications that traverse them, an adversary may seek to hamper all communication between two large geographic areas. To do so, they must congest *all* paths connecting the two regions simultaneously. This makes the attack more challenging: the adversary now has to carefully select the set of target links to congest, such that hampering region-region connectivity incurs low cost and has low maxUp. Figure 3 shows an example in which

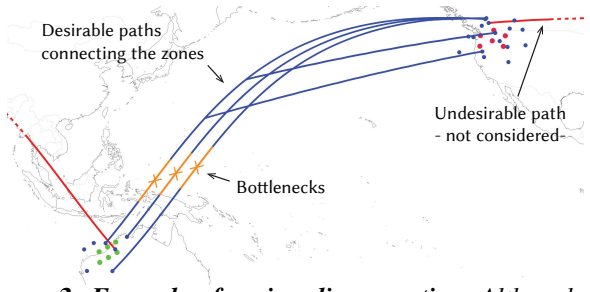



**Figure 3:** ***Example of region disconnection.*** *Although 12 distinct shortest paths are available between North-western America and North-western Australia, just 3 bottlenecks suffice to prevent communication between these large zones.*

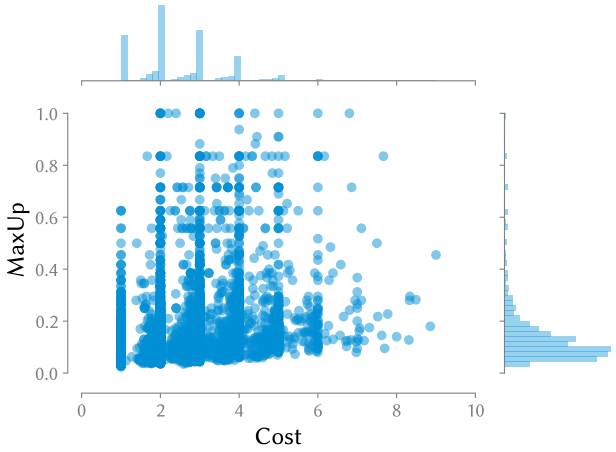



**Figure 4:** ***Cost and maxUp of zone disconnection attacks.*** *As shown by the marginals, maxUp is low for most attacks. The peaks shown in the cost marginal correspond to the number of bottlenecks in the paths between the zones.*

the adversary intelligently selects 3 bottlenecks shared across the 12 different paths, greatly reducing the attack cost. In many cases, however, the choice is not as obvious.

The adversary has to select the minimum set of links *it is able to attack simultaneously*, among the ones in the paths connecting the zones, such that their congestion results in a complete disconnection. Unfortunately, this problem is equivalent to the minimum set cover problem (we prove equivalence in §B in the Supplemental materials), which is NP-hard [34]. To avoid exponential explosion, we use a heuristic which computes a good approximation of the set of links to attack in polynomial time. Each link in the paths to congest is assigned a score, equal to the ratio between the maxUp of the attack on the link, and the number of paths that share the link. The lower this score, the stealthier it is to attack such a link. We therefore add the link with the lowest score to the attack set, we remove it from the network alongside all the paths it congests, and finally recompute the scores for all remaining links. We continue this procedure until all paths are congested. We run this heuristic three times, slightly varying the order in which links are removed, and keep the lowest maxUp result.

**Simulation results.** We iteratively sample 5000 pairs of regions, each comprising six points on the geodesic grid (§ 4.2). For each pair, we run the attack twice, inverting source and destination region, for a total of 9658 viable attack cases (some are discarded because the zones are overlapping). The results are shown in Fig. 4.

We find that 9208 of these zone pairs (95%) can be successfully attacked. The median maxUp of these attacks is 0.10, implying that attacking large regions does not expose the adversary more than the single-link attacks (where the maxUp was 0.13). The minor difference in maxUp arises because the zone construction algorithm prevents some of the corner cases that increase the median maxUp in the single-link attacks. Since zones are composed by multiple grid points, they have to be located farther away from the edges of satellite connectivity (high latitudes, or narrow corners of continents), thus giving the adversary more uplinks on which to distribute the attack traffic. Finally, the overall cost is mainly driven by the number of bottlenecks found (characterized by the spikes in the marginal in Fig. 4), which is consistently below 4.

## 5 ICARUS against more sophisticated routing

We use the single shortest path routing setting discussed thus far primarily to develop intuition about the vulnerabilities of an LSN, and to broadly understand an adversary's tradeoffs. In practice, the more likely scenario entails the LSN load-balancing traffic across multiple paths. We therefore consider several candidate routing schemes for such load balancing, and then formulate and evaluate a modified attack strategy suited for this more complex setting.

### 5.1 Candidate routing schemes

Developing, evaluating, and comparing new routing strategies is not our objective, and has been tackled in some depth in prior work (§ 8). However, to understand how load balancing across multiple paths impacts an adversary, we must frame some suitable routing methods. Here, we use one guiding constraint: the chosen non-shortest paths must still be "near shortest", i.e., not increase latency by a large amount. In the extreme, any path's latency must not exceed that of a terrestrial fiber route.[3] We thus evaluate the following multi-path routing variants:

- ***k*-SP** *Shortest paths:* This strategy load-balances traffic among the *k*-shortest source–destination paths.
- ***k*-DG** *Ground-to-ground disjoint paths:* This strategy only considers the *k* shortest paths that are node-disjoint, i.e.,without shared uplinks, downlinks, or ISLs.
- ***k*-DS** *Satellite-to-satellite disjoint paths:* This strategy is similar to the above, with the exception that overlap is allowed on uplinks and downlinks. Disjointness is enforced exclusively on the ISLs.

[3] This fiber latency is estimated, based on past measurement work [58], by multiplying the great-circle distance by a path-stretch factor of 1.53, and then dividing by the speed of light in fiber, $\approx 2c/3$.

**$k$-LO** *Limited-overlap shortest paths:* we implement the ESX algorithm by Chondrogiannis et al. [15], a heuristic that finds the shortest $k$ paths with a *similarity score* of no more than 50%.

Note that sometimes these algorithms yield fewer than $k$ paths because of the faster-than-fiber and path disjointness constraints. (For $k$-SP, only the former applies, and is only limiting for nearby end-points.) Each algorithm considers different types of near-shortest paths, varying the degree to which paths overlap. $k$-SP, which does not limit overlap, typically offers multiple paths with nearly the same latency as the shortest, but risks the shared links across these paths becoming the bottleneck. On the other extreme, the most restrictive scheme, $k$-DG, typically does not even provide more than 4 paths. Further analysis of the path structure of these algorithms is in §C in the Supplement. For each scheme, we assume that the LSN uses randomized load balancing (similar to ECMP) across the chosen path. Note that uniform random load balancing is the worst case for the adversary; any deterministic adaptive scheme will strictly reduce uncertainty for the adversary.

## 5.2 Probabilistic ICARUS

The use of load-balanced, uniformly randomized routing introduces uncertainty in the *link discovery* phase. As the network chooses the end-to-end path only at forwarding time, from a set of $k$ pre-computed paths, the adversary cannot know in advance which specific path will be taken by their attack traffic. We therefore present a probabilistic variation of ICARUS that accounts for this uncertainty. This attack performs attack-flow assignment in such a way that the target link will be flooded with high probability, while avoiding self-congestion among attack flows, and lowering maxUp.

**Congesting the target**. The adversary pre-computes all the load-balancing paths for all its bot source-destination pairs, and considers those $(s,d)$-pairs for which *at least* one path contains the target link. Consider an $(s,d)$-pair connected by $n \leq k$ paths, with $m$ of them crossing the target link. Then the event "the forwarding path chosen for this $(s,d)$-pair uses the target link" can be described as a Bernoulli random variable $X_{(s,d)}$ with probability of success $p_{(s,d)} = m/n$. Suppose the adversary uses an attack-flow set $\mathbb{A}$, comprising many $(s,d)$-pairs. Then the sum $Y = \sum_{s,d\in\mathbb{A}} X_{s,d}$, is a random variable describing the number of attack flows that are forwarded across the target link. The $X$ variables are independent but *not* identically distributed, with different $m$ and $n$ across $(s,d)$-pairs, therefore $Y$ follows a Poisson binomial distribution.

The adversary's goal is for the attack-flow set, $\mathbb{A}$, to be such that at least $T$ attack flows use the target link with high probability. Here, $T$ is determined by each attack flow's bandwidth, and the target link's capacity; in our model, for example, an ISL has a capacity of 20 Gbps; if a single $(s,d)$-pair can transmit 40 Mbps (3.2), $T$ is 500.

The target link is congested if $Y \geq T$. The adversary wants to bound their probability of failure to at most a small value, $\beta$, i.e., $P[Y < T] \leq \beta$. The adversary can get more certainty, i.e., lower $\beta$, at the expense of more resources.

The computation of $P[Y < T]$ is infeasible for non-trivial settings, but we can approximate it via the Chernoff (lower tail) bound [46]:

$$P[Y < (1-\delta)\mu] \leq \exp(-\mu\delta^2/2) \tag{1}$$

where $\mu = E[Y]$. By setting $\delta = 1 - \frac{T}{\mu}$ in eq. (1), we get:

$$P[Y < T] \leq \exp(-\mu(1-T/\mu)^2/2)$$

Note that $\mu = E[Y] = \sum_{(s,d)\in\mathbb{A}} p_{(s,d)}$ by the linearity of expectation. The adversary can easily calculate $\mu$ for any attack-flow set, $\mathbb{A}$. To ensure that $P[Y < T] \leq \beta$, the adversary can pick $\mathbb{A}$'s constituent $(s,d)$-pairs such that they satisfy:

$$\exp(-\mu(1-T/\mu)^2/2) \leq \beta$$

For $0 < \beta < 1$ and $1 \leq T < \mu$, this is equivalent to:

$$\mu > \sqrt{\ln\beta \cdot (\ln\beta - 2T)} + T - \ln\beta = \mu_{min} \tag{2}$$

**Avoiding self-congestion**. The adversary must also ensure that the attack flows do not cause self-congestion, i.e., congest other links before reaching the target. This problem, of *not* congesting non-target links, requires the same machinery as that above for congesting the target. For every non-target link, $l$, we simply need to repeat analysis similar to the above, except towards ensuring we *do not* congest $l$, i.e., $W_l < C_l$, where $W_l$ is a random variable describing the number of attack flows that traverse $l$, and $C_l$ is $l$'s capacity.

Say $\gamma$ is a parameter analogous to $\beta$ above, indicating the maximum acceptable risk of self-congestion. For any non-target link, $l$, $\mu_l$ (analogous to $\mu$ above) is the expected number of attack flows crossing $l$, such that $\mu_l = \sum_{(s,d)} p_{(s,d)}$ for $(s,d)$-pairs that traverse link $l$. Analysis similar to the above—but considering the upper tail—yields, the following condition:

$$\mu_l < \frac{1}{2}\left(-\sqrt{\ln\gamma \cdot (\ln\gamma - 8C_l)} - \ln\gamma + 2C_l\right) = \mu_{l,max} \tag{3}$$

**Probability of attack success**. An attack is successful if the target is congested without congesting any other link. We combine the probabilities computed so far as follows:

$$\begin{aligned}
P[\text{success}] &= 1 - P[\text{failure}] && (4)\\
&= 1 - P[Y < T \vee W_1 \geq C_1 \vee \cdots \vee W_N \geq C_N] && (5)\\
&\overset{\text{U.B.}}{\geq} 1 - \left(P[Y < T] + \sum_{l=1}^{N} P[W_l \geq C_l]\right) && (6)\\
&\geq 1 - (\beta + N\cdot\gamma) && (7)\\
&= 1 - \alpha && (8)
\end{aligned}$$

where eq. (6) is given by the union bound, and $N$ is the number of non-target links. For a high probability of success, $\alpha$ has to be small. In our setting, $N$ is in the range 1500–2000. We therefore choose $\gamma = 1/N^2 = 1/2000^2$ and $\beta = 0.1$, which gives $P[\text{success}] \approx 0.9$. Empirically, we find that the Chernoff bound is quite loose, and for these values of $\beta$ and $\gamma$ the success probability is higher (≥95%).

**Attack set construction**. The adversary must construct the attack-flow set, $\mathbb{A}$, such that it simultaneously satisfies the constraints on $\mu$, and for every non-target link $l$, $\mu_l$:

$$\mu = \sum_{(s,d)\in\mathbb{A}} p_{(s,d)} > \mu_{min} \quad \text{and} \tag{9a}$$

$$\mu_l = \sum_{(s,d)\in W_l} p_{(s,d)} < \mu_{l,max} \quad \forall \text{ non-target links } l \tag{9b}$$

Since $\mu$ is additive, we use a greedy approach:

- We sort the $(s,d)$-pairs by decreasing value of $p_{(s,d)}$.
- Iteratively, we add an $(s,d)$-pair to $\mathbb{A}$, checking that eq. (9b) holds. Otherwise, we discard the pair and move on to the next one.
- We repeat this until either eq. (9a) is satisfied—in which case the attack succeeds with probability $\geq 1-\alpha$, or there are no more $(s,d)$-pairs, and the algorithm fails to guarantee this probability of success.

The above approach yields the attack set with the minimum number of $(s,d)$ pairs. However, maxUp is not minimized, and is (nearly) $\mu_{max}/C_{\text{uplink}}$. If the adversary wants to minimize maxUp before cost, they can iteratively lower $\mu_{l,max}$ for the uplinks to the minimum value such that the construction of $\mathbb{A}$ still succeeds.

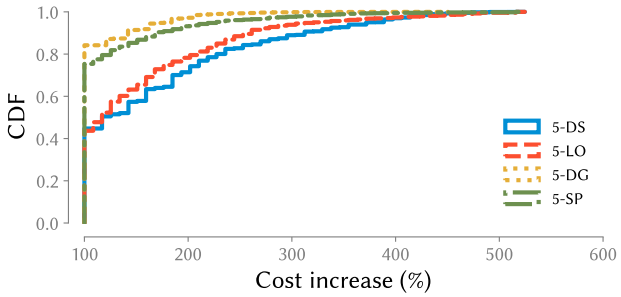
**Further optimization**. Due to the structure of the topology and the routing schemes, for each target link, there are certain $(s,d)$-pairs for which $p_{(s,d)} = 1$. In the above probabilistic analysis, such $(s,d)$-pairs needlessly contribute to making the probability bounds loose. Instead, they can be considered separately, and the above analysis done after accounting for those $(s,d)$-pairs.

## 5.3 Evaluation

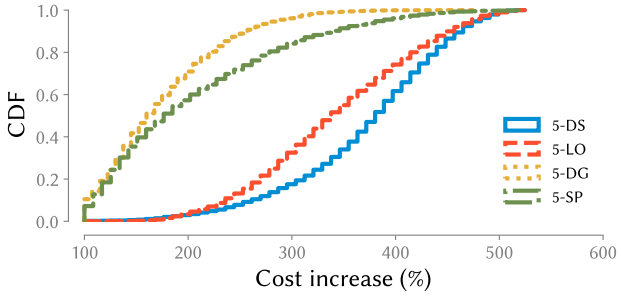
We evaluate probabilistic ICARUS using the same constellation, Starlink shell I, under the 4 routing schemes from § 5.1, with $k = 5$ paths, and uniform randomization across the paths. We present results for the empty network scenario, without any benign traffic, as it is the costliest for the adversary.

We target a maximum attack failure probability $\alpha$ of 10%, and measure the cost for the adversary in terms of the size of the attack set. A resilient routing scheme will force the adversary to incur higher attack cost for the same $\alpha$, while vulnerable routing schemes will incur cost closer to the single shortest-path routing case. Following this intuition, we present attack costs for different routing schemes in comparison to the latter.

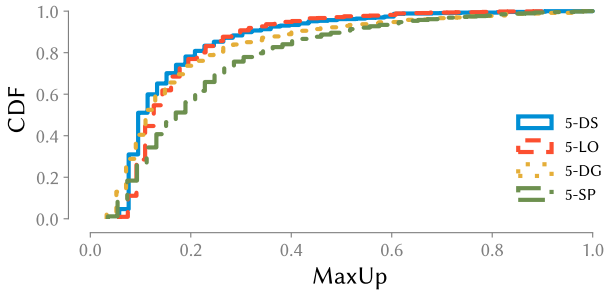
Fig. 5a and 5b show the attack cost for all 4 routing schemes, when the attacker optimizes for cost and for maxUp, respectively. Surprisingly, 5-DG and 5-SP perform similarly, even though they are very different: the former does not allow any overlap between paths for the same $(s,d)$-pair, while in the latter, path overlaps are not restricted at all. The commonality, however, is that the uncertainty in the forwarding path is low for both. In 5-SP, this is due to the lack of separation between the 5 absolute-shortest paths in the SN. In




**(a)** *Cost CDF for the attacks, when executed with cost minimization. For maxUp, please refer to text in § 5.2.*




**(b)** *Cost CDF for the attacks, when executed with maxUp minimization. The cost increases significantly compared to the previous figure, to the benefit of maxUp (below).*




**(c)** *MaxUp CDF for the attacks, when executed for maxUp minimization.*

**Figure 5:** ***Cost and maxUp of the attacks on all ISLs assuming different load-balancing strategies.***

5-DG, on the other hand, the low uncertainty comes from the lack of alternatives: between many $(s,d)$-pairs, this scheme is overly restrictive and does not provide 5 paths. The adversary can use such $(s,d)$-pairs in the attack-flow set, effectively side-stepping routing uncertainty.

The most resilient scheme in terms of increasing the adversary's cost, is 5-DS. This scheme strikes a balance between avoiding path overlap and being flexible enough to maintain enough path options: enforcing disjointness between satellites is less limiting that ground-to-ground disjointness. The median cost increase compared to single shortest-path routing is 385% with 5-DS. The price of this resilience is increased latency — paths often incur more than 1.5× latency compared to the shortest path (Fig. 4 in the Supplemental materials).

Unfortunately, despite incurring a high latency cost, while

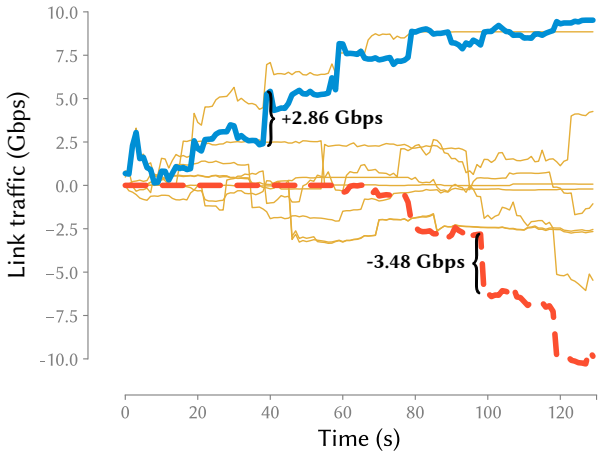



**Figure 6:** ***Load surges.*** *The figure shows the link traffic over time of the* 10 *links with the largest load surges. The biggest positive (—) and negative (– –) surges are also highlighted.*

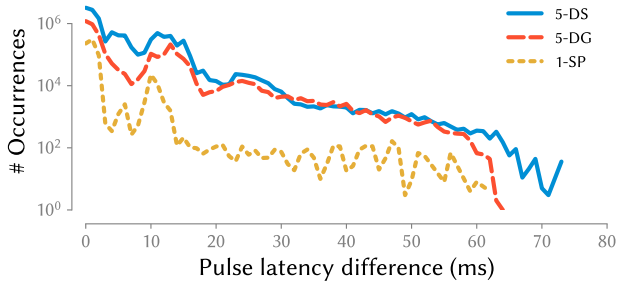



**Figure 7:** ***Feasibility of pulsing attacks.***

the percent-increase in cost seems large (385%), in the absolute, this is still under 80 Gbps of attack traffic, and botnets with a few thousand bots (depending on per bot bandwidth) can still congest the median LSN link. Note also that this cost is in the most difficult setting, i.e., without any benign traffic, with the LSN accepting a large latency deterioration, and with the attacker seeking to absolutely minimize maxUp. Further, valuable links above land are easier to attack. Figure 2 in the Supplemental materials shows *where* the cost of the attack is highest across the network's ISLs, with 5-DS. Attacking ISLs that are above land is easier because the adversary can always find a source-destination bot-pair for which there are few, or greatly overlapping paths (reducing uncertainty).

For maxUp, shown in Fig. 5c, 5-SP is the outlier, with higher maxUp than the other three schemes, for which results are similar. This is because other algorithms often have multiple uplinks available in the same load-balancing set, thus spreading the attack traffic and reducing its maxUp. In 5-SP, paths mostly overlap, and especially on the uplinks, preventing this effect.

## 6 Open problem: exploiting LSN dynamics

Thus far, the attacks we have presented consider only snapshots of the SN's state, with the adversary drawing on the relative stability of the network on the timescale of seconds to pre-compute attack flows. It is worth considering how/if the adversary may exploit the LSN's dynamics as an additional opportunity, rather than a minor hindrance that requires per-computed, albeit frequent, changes in the attack-flow set. We discuss two potential opportunities of this type.

**Load surges**. The motion of satellites naturally moves ISLs across low- and high-utilization areas. As a result, the network's links see large fluctuations in their utilization over time. Figure 6 shows how the load changes in time across the links with the biggest surges and drops over 130 seconds of simulation for the GDP traffic model. An adversary that knows or can reasonably guess the traffic matrix of the LSN can use these load surges to increase the effectiveness of low-volume attacks: well-timed bursts that coincide with the natural surges can cause high congestion, at a fraction of the cost. The effectiveness of such an attack depends on the the LSN's protocol stack, e.g., how long does congestion control take to detect such short, transient congestion, and how does it react and recover? Addressing this requires packet-level analysis, and is left to future work.

**Pulsing attacks**. Another dynamic feature of an LSN is the time-varying latency of paths. Consider a bot that has a path to a destination, $d$, across a target link. Over time, this bot's path to $d$ changes, with some of these changes still traversing the target, but potentially over a higher- or lower-latency path, resulting from satellite motion. When the path changes from a higher latency one to a lower latency one, the adversary can potentially double the effective attack bandwidth of the bot: traffic sent on the first and the second path will traverse the target link *simultaneously* for a short time interval. This "pulsing attack" bears similarities to *temporal lensing* [55], with the important difference that temporal lensing uses *external* infrastructure like DNS to create a time difference between the forwarding paths, while in LSNs this is provided by the network's inherent dynamics.

We briefly test the feasibility of pulsing attacks by running the following experiment:

1. At each time-step, we consider a target link, and compute all the paths across it.
2. We identify paths that will not be valid in the next time-step, and compute their replacement paths.
3. For each changed path, we compute the difference in latency from the source to the target link, between the original path and its replacement.

We run this procedure for all 6336 target links in a 130-sec simulation, and a time-step of 1 sec for evaluating changes. Figure 7 shows the results for a few routing schemes. For single shortest-path routing, achieving latency differences of tens of milliseconds across the path change is possible less than once per simulation second across the *entire* constellation. The multipath schemes increase this opportunity, but it is nevertheless small: even for the most favorable algorithm, 5-DS, there are only 8133 pulses with a duration above

50 ms. Since there are 6336 target ISLs, the adversary can benefit from little more than 1 pulse per ISL, on average, over the entire 130 sec of the simulation. Thus, while pulsing is possible in principle, its utility for an adversary is limited to occasionally impacting a small number of targets.

**Future outlook**. Our above analysis indicates positive potential for load surges, and a largely negative result for pulsing. Exploring other methods for an adversary to leverage LSN dynamism, as well as a deeper understanding of the impact of load surges, are left to future work.

## 7 Mitigations

We find that several traditional methods of addressing DDoS attacks are not applicable against ICARUS:

- *Overprovisioning* to absorb more than expected traffic is difficult: our model already accounts for ISLs having higher capacity than access links, and increasing ISL capacity or number will push against the cost, weight and power constraints of the satellites.
- *In-network filtering* [4, 26, 39, 42, 45, 54] requires computational resources, which are lacking at both the satellites and the TUs (TUs often need to be portable). More fundamentally, like Coremelt and Crossfire, attack traffic is indistinguishable from benign traffic. Therefore, even if satellites or TUs have the computational capabilities for filtering, it is unclear how they would distinguish malicious and benign traffic.
- *Capabilities* in the form of cryptographic tokens that provide access rights [2, 37, 38, 40, 52, 72] are not useful here either — the adversary compromises legitimate satellite-connected users to gain access to the LSN; thus inheriting these users' capabilities.
- *Cloud-based mitigations* [1, 16, 21, 24, 41, 51] offload the filtering of adversarial traffic to the cloud. Since TUs will often use the LSN as their last-mile provider, the burden of passing traffic through a cloud provider falls on the LSN, incurring additional bandwidth and latency overheads.

Thus, effective defenses against Coremelt and Crossfire—such as upgrading capacity or re-routing traffic to more capable networks—cannot be used to alleviate ICARUS' impact. In light of the above, we discuss two avenues that show promise towards neutralizing ICARUS.

**Resilient routing and network design**. Our experiments in § 5.2 show that resilient routing can greatly increase the attack's cost. While for our tested routing schemes, this comes at the expense of increased latency, we have only scratched the surface in exploring resilient routing; identifying routing schemes with the most favorable tradeoff curve remains an interesting direction for future work. Another possibility is to attack the problem using network topology design — Bhattacherjee et al. [10] have recently proposed to re-arrange ISLs, deviating from the traditional "cross" pattern, to improve forwarding latency. A similar strategy can be used in an adversarial setting: altering the structure of the network can improve resilience to attacks with minimal impact on latency.

**Traffic separation and differential pricing**. In this work, we assume that all TUs are equally capable, and optimized for low-latency communication. It is foreseeable, however, that not all hosts will require such an optimization. This consideration opens up the opportunity of creating different classes of service, with a premium service offering the lowest latency and highest resilience, at the expense of shifting other traffic to longer, less predictable paths. It remains to be seen, however, whether this introduces enough randomness in forwarding to make the attacks too costly to perform, or if, instead, the higher resource usage caused by more circuitous paths *increases* vulnerability.

## 8 Other related work

We discuss related work not already covered in § 2 or § 7.

**Attacks on satellite systems**. The most commonly recognized threat to satellite communications on the physical layer is *jamming* [30, 49, 56]: the adversary uses high-power antennas to induce noise on up- and downlinks, preventing the endpoints from reaching the ingress satellite. ICARUS is however beyond detection by a jamming analyzer as it uses protocol-compliant traffic from authorized (albeit compromised) sources. GNSS spoofing is another well-understood threat, with a long list of attacks and mitigations [57]. More recently, the attention has shifted towards the security of cryptographic standards used in satellite communications [18], and lack thereof [50].

Network-layer denial of service on LSNs has been addressed, to the best of our knowledge, only recently by Usman et al. [67]. However, their focus is on traditional geostationary orbit satellite networks, using only bent-pipe connectivity through satellites, without any notion of ISLs. The authors thus focus on ping flooding to exhaust control plane resources at GS network devices. In contrast, our work examines a completely different setting with modern LSNs, analyzing an entirely different breed of attacks: congestion-causing volumetric DDoS attacks from distributed bots. To the best of our knowledge, we are the first to frame and study this problem.

**Routing in satellite constellations**. Given the dynamic nature of links in LSNs, many works focus on the challenges of discovering and disseminating network status information, and optimizing forwarding latency [14, 19, 20, 22, 23, 27, 43, 53, 68–71, 73]. Barrit et al. [6–8] bring software-defined networking (SDN) to LSNs, and introduce the Temporospatial-SDN. The SDN controller uses the predictability of satellite orbits to maintain an accurate view of the LSN's topology, and aid routing decisions. Despite this past work, the subject of optimizing routing for resilience against an ICARUS-like attacker remains an open question.

## 9 Conclusion

We present ICARUS, the first volumetric DDoS attack against next-generation LSNs. We demonstrate ICARUS's disruptive potential across a variety of scenarios, with different LSN routing schemes, and different attack targets. ICARUS's potency stems from leveraging several unique characteristics of LSNs, whereby the adversary operates with greater information than available for terrestrial networks, and exploits the topology and path structure of LSNs. ICARUS also turns the low-latency and global coverage objectives of LSNs into vulnerabilities. Nevertheless, successful attacks of such networks require careful limitation of cost and detectability (characterized with the maxUp metric), as ICARUS does. In addition, randomized multipath routing increases an adversary's cost, but as we experimentally show, ICARUS also succeeds in that setting. We hope that our first steps in understanding the vulnerability of LSNs to DDoS will seed the ground for further research on this topic. To this end, we release our simulation framework, which is the first tool for analyzing DDoS attacks and defences on LSNs. It allows easy testing of the resilience of arbitrary constellation configurations and novel path-selection algorithms.


## Acknowledgements

We are grateful to our anonymous reviewers and our shepherd Gerd Zellweger for their insightful feedback and valuable suggestions. We gratefully acknowledge financial support from ETH Zurich and from the Zurich Information Security and Privacy Center (ZISC).


## References


[1] Amazon. Amazon AWS shield. https://aws.amazon.com/shield/, 2020.

[2] Tom Anderson, Timothy Roscoe, and David Wetherall. Preventing Internet denial-of-service with capabilities. *ACM SIGCOMM Computer Communication Review*, 34(1):39–44, 2004.

[3] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. Understanding the Mirai Botnet. In *USENIX Security Symposium*, pages 1093–1110, 2017.

[4] K. Argyraki and D.R. Cheriton. Active internet traffic filtering: Real-time response to denial-of-service attacks. *USENIX Annual Technical Conference*, pages 135–148, 2005.

[5] Ars Technica. SpaceX Starlink speeds revealed as beta users get downloads of 11 to 60mbps. https://arstechnica.com/information-technology/2020/08/spacex-starlink-beta-tests-show-speeds-up-to-60mbps-latency-as-low-as-31ms/, 2020.

[6] Brian Barritt and Wesley Eddy. Temporospatial SDN for aerospace communications. In *AIAA SPACE Conference and Exposition*, 2015.

[7] Brian Barritt, Tatiana Kichkaylo, Ketan Mandke, Adam Zalcman, and Victor Lin. Operating a UAV mesh & internet backhaul network using temporospatial SDN. In *IEEE Aerospace Conference*, 2017.

[8] Brian J. Barritt and Wesley Eddy. SDN enhancements for LEO satellite networks. In *AIAA International Communications Satellite Systems Conference*, 2016.

[9] Debopam Bhattacherjee, Waqar Aqeel, Ilker Nadi Bozkurt, Anthony Aguirre, Balakrishnan Chandrasekaran, P. Brighten Godfrey, Gregory Laughlin, Bruce Maggs, and Ankit Singla. Gearing up for the 21st century space race. In *ACM Workshop on Hot Topics in Networks - HotNets*, 2018.

[10] Debopam Bhattacherjee and Ankit Singla. Network topology design at 27,000 km/hour. In *International Conference on Emerging Networking Experiments And Technologies - CoNEXT*, 2019.

[11] Business Insider. Starlink's speed tests may look impressive, but experts say SpaceX's satellite-internet project is unlikely to win any federal subsidies. https://www.businessinsider.com/spacex-starlink-beta-speedtest-results-bandwidth-ping-latency-fcc-rdof-2020-8, 2020.

[12] CelesTrak. NORAD two-line element sets current data. https://celestrak.com/NORAD/elements/, 2020.

[13] Center For International Earth Science Information Network-CIESIN-Columbia University. Gridded Population of the World, Version 4 (GPWv4): Population Count, Revision 11. https://sedac.ciesin.columbia.edu/data/set/gpw-v4-population-count-rev11, 2018.

[14] Chao Chen, Eylem Ekici, and Ian F. Akyildiz. Satellite grouping and routing protocol for LEO/MEO satellite IP networks. In *ACM international workshop on Wireless mobile multimedia*, 2002.

[15] Theodoros Chondrogiannis, Panagiotis Bouros, Johann Gamper, Ulf Leser, and David B. Blumenthal. Finding k-shortest paths with limited overlap. *The VLDB Journal*, 29(5):1023–1047, 2020.

[16] CloudFlare. Comprehensive DDoS protection. https://www.cloudflare.com/ddos/, 2020.

[17] Inigo del Portillo, Bruce Cameron, and Edward Crawley. Ground segment architectures for large LEO constellations with feeder links in EHF-bands. In *IEEE Aerospace Conference*, 2018.

[18] Benedikt Driessen, Ralf Hund, Carsten Willems, Christof Paar, and Thorsten Holz. Don't trust satellite phones: A security analysis of two satphone standards. In *IEEE Symposium on Security and Privacy*, 2012.

[19] E. Ekici, I.F. Akyildiz, and M.D. Bender. Datagram routing algorithm for LEO satellite networks. In *Proceedings IEEE INFOCOM. Conference on Computer Communications*, 2000.

[20] O. Ercetin, S. Krishnamurthy, Son Dao, and L. Tassiulas. A predictive QoS routing scheme for broadband low earth orbit satellite networks. In *IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, 2000.

[21] S.K. Fayaz, Y. Tobioka, V. Sekar, and M. Bailey. Bohatei: Flexible and elastic DDoS defense. *USENIX Security Symposium*, pages 817–832, 2015.

[22] Daniel Fischer, David Basin, Knut Eckstein, and Thomas Engel. Predictable mobile routing for spacecraft networks. *IEEE Transactions on Mobile Computing*, 12(6):1174–1187, 2013.

[23] Daniel Fischer, David Basin, and Thomas Engel. Topology dynamics and routing for predictable mobile networks. In *IEEE International Conference on Network Protocols*, 2008.

[24] Y. Gilad, A. Herzberg, M. Sudkovitch, and M. Goberman. CDN-ondemand: An affordable DDoS defense via untrusted clouds. *Network and Distributed System Security Symposium - NDSS*, pages 1–15, 2016.

[25] Giacomo Giuliari, Tobias Klenze, Markus Legner, David Basin, Adrian Perrig, and Ankit Singla. Internet backbones in space. *ACM SIGCOMM Computer Communication Review*, 50(1):25–37, 2020.

[26] Deli Gong, Muoi Tran, Shweta Shinde, Hao Jin, Vyas Sekar, Prateek Saxena, and Min Suk Kang. Practical verifiable in-network filtering for DDoS defense. In *IEEE International Conference on Distributed Computing Systems*, 2019.

[27] Vidyashankar V Gounder, Ravi Prakash, and Hosame Abu-Amara. Routing in LEO-based satellite networks. *IEEE Emerging Technologies Symposium. Wireless Communications and Systems*, pages 22.1–22.6, 1999.

[28] Mark Handley. Delay is not an option. In *ACM Workshop on Hot Topics in Networks - HotNets*, 2018.

[29] Mark Handley. Using ground relays for low-latency wide-area routing in megaconstellations. In *ACM Workshop on Hot Topics in Networks - HotNets*, 2019.

[30] Todd Harrison, Katylin Johnson, and Thomas G. Roberts. Space threat assessment 2018. https://www.csis.org/analysis/space-threat-assessment-2018, 2018.

[31] Min Suk Kang, Soo Bum Lee, and V. D. Gligor. The Crossfire attack. In *IEEE Symposium on Security and Privacy*, 2013.

[32] T.S. Kelso. Validation of SGP4 and IS-GPS-200D Against GPS Precision Ephemerides. *AAS/AIAA Space Flight Mechanics Conference*, 2007.

[33] Tobias Klenze, Giacomo Giuliari, Christos Pappas, Adrian Perrig, and David Basin. Networking in Heaven as on Earth. In *ACM Workshop on Hot Topics in Networks - HotNets*, 2018.

[34] Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer-Verlag Berlin Heidelberg, 2012.

[35] Kuiper Systems LLC. Application of Kuiper Systems LLC forAuthority to Launch and Operate a Non-Geostationary Satellite Orbit System in Ka-band Frequencies. https://licensing.fcc.gov/myibfs/download.do?attachment_key=1773885, 2019.

[36] Aleksandar Kuzmanovic and Edward W. Knightly. Low-rate TCP-targeted denial of service attacks. In *Conference on Applications, technologies, architectures, and protocols for computer communications*, 2003.

[37] Soo Bum Lee and Virgil D Gligor. FLoc: Dependable link access for legitimate traffic in flooding attacks. In *IEEE International Conference on Distributed Computing Systems*, pages 327–338, 2010.

[38] Soo Bum Lee, Min Suk Kang, and Virgil D Gligor. CoDef: Collaborative defense against large-scale link-flooding attacks. In *ACM International Conference on emerging Networking EXperiments and Technologies - CoNEXT*, pages 417–428, 2013.

[39] X. Liu, X. Yang, and Y. Lu. To filter or to authorize: Network-layer DoS defense against multimillion-node botnets. *ACM SIGCOMM Conference on Data Communication*, pages 195–206, 2008.

[40] Xin Liu, Xiaowei Yang, and Yong Xia. NetFence: preventing internet denial of service from inside out. *ACM SIGCOMM Computer Communication Review*, 40(4):255–266, 2010.

[41] Z. Liu, H. Jiny, Y.-C. Hu, and M. Bailey. Middlepolice: Toward enforcing destination-defined policies in the middle of the internet. *ACM Conference on Computer and Communications Security*, 24-28-October-2016:1268–1279, 2016.

[42] R. Mahajan, S.M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker. Controlling high bandwidth aggregates in the network. *Computer Communication Review*, 32(3):62–73, 2002.

[43] S. A. M. Makki, Niki Pissinou, and Philippe Daroux. LEO satellite communication networks - a routing approach. *Wireless Communications and Mobile Computing*, 3(3):385–395, 2003.

[44] Roland Meier, Petar Tsankov, Vincent Lenders, Laurent Vanbever, and Martin Vechev. Nethide: Secure and practical network topology obfuscation. In *USENIX Security Symposium*, pages 693–709, 2018.

[45] J. Mirkovic, G. Prier, and P. Reiher. Attacking DDoS at the source. *International Conference on Network Protocols, ICNP*, pages 312–321, 2002.

[46] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, 2017.

[47] Mynaric. Flight terminals (space). https://mynaric.com/products/space/, 2020.

[48] W.D. Nordhaus and X. Chen. Global gridded geographically based economic (G-Econ) data set, version 4. https://sedac.ciesin.columbia.edu/data/set/spatialecon-gecon-v4, 2016.

[49] E.J. Ohlmeyer. Analysis of an ultra-tightly coupled GPS/INS system in jamming. In *IEEE/ION Position, Location, And Navigation Symposium*, 2006.

[50] James Pavur, Daniel Moser, Martin Strohmeier, Vincent Lenders, and Ivan Martinovic. A tale of sea and sky on the security of maritime VSAT communications. In *IEEE Symposium on Security and Privacy*, 2020.

[51] Radware. Radware's DefensePro DDoS Protection. https://www.radware.com/products/defensepro/, 2020.

[52] Barath Raghavan and Alex C Snoeren. A system for authenticated policy-compliant routing. *ACM SIGCOMM Computer Communication Review*, 34(4):167–178, 2004.

[53] M Rajanna, Shiva Murthy, and Kantharaju H C. Satellite networks routing protocol issues and challenges: A survey. *International Journal of Innovative Research in Computer and Communication Engineering*, 2:153, 2007.

[54] S. Ramanathan, J. Mirkovic, M. Yu, and Y. Zhang. SENSS against volumetric DDoS attacks. *Annual Computer Security Applications Conference*, pages 266–277, 2018.

[55] Ryan Rasti, Mukul Murthy, Nicholas Weaver, and Vern Paxson. Temporal lensing and its application in pulsing denial-of-service attacks. In *IEEE Symposium on Security and Privacy*, 2015.

[56] H. Rausch. Jamming commercial satellite communications during wartime an empirical study. In *IEEE International Workshop on Information Assurance*, pages 8 pp.–118, 2006.

[57] Desmond Schmidt, Kenneth Radke, Seyit Camtepe, Ernest Foo, and Michał Ren. A survey and analysis of the GNSS spoofing threat and countermeasures. *ACM Computing Surveys*, 48(4):1–31, 2016.

[58] Ankit Singla, Balakrishnan Chandrasekaran, P. Brighten Godfrey, and Bruce Maggs. The internet at the speed of light. In *ACM Workshop on Hot Topics in Networks*, HotNets, pages 1:1–1:7, 2014.

[59] SpaceNews. SpaceX adds laser crosslinks to polar Starlink satellites. https://spacenews.com/spacex-adds-laser-crosslinks-to-polar-starlink-satellites/, 2021.

[60] SpaceX. SpaceX Non-Geostationary Satellite System. https://fcc.report/IBFS/SAT-LOA-20161115-00118/1158350.pdf, 2016.

[61] SpaceX. SpaceX Non-Geostationary Satellite System, modification on satellite space station filing. https://fcc.report/IBFS/SAT-MOD-20190830-00087/1877671, 2019.

[62] SpaceX. Application for modification of blanket earth station authorization. https://fcc.report/IBFS/SES-MOD-INTR2020-02035/2612707, 2020.

[63] SpaceX. SpaceX Non-Geostationary Satellite System, modification on satellite space station filing. https://licensing.fcc.gov/myibfs/download.do?attachment_key=2274316, 2020.

[64] Ahren Studer and Adrian Perrig. The Coremelt attack. In *Computer Security – ESORICS*, pages 37–52, 2009.

[65] Telesat. Telesat Lightspeed. https://www.telesat.com/leo-satellites/, 2021.

[66] Thales. VasseLINK user manual. https://www.verasatglobal.com/wp-content/uploads/2019/02/Thales-Vesselink-User-Manual.pdf, 2020.

[67] Muhammad Usman, Marwa Qaraqe, Muhammad Rizwan Asghar, and Imran Shafique Ansari. Mitigating distributed denial of service attacks in satellite networks. *Transactions on Emerging Telecommunications Technologies*, 31(6), 2020.

[68] M. Werner. A dynamic routing concept for ATM-based satellite personal communication networks. *IEEE Journal on Selected Areas in Communications*, 15(8):1636–1648, 1997.

[69] Lloyd Wood. *Internetworking with satellite constellations*. PhD thesis, University of Surrey, 2001.

[70] Lloyd Wood. Satellite constellation networks. In *Internetworking and Computing Over Satellite Networks*, pages 13–34. Springer, 2003.

[71] Yipeng Wu, Zhihua Yang, and Qinyu Zhang. A novel DTN routing algorithm in the GEO-relaying satellite network. In *International Conference on Mobile Ad-hoc and Sensor Networks*, 2015.

[72] Abraham Yaar, Adrian Perrig, and Dawn Song. SIFF: A stateless Internet flow filter to mitigate DDoS flooding attacks. In *IEEE Symposium on Security and Privacy*, pages 130–143, 2004.

[73] Yuan Yang, Mingwei Xu, Dan Wang, and Yu Wang. Towards energy-efficient routing in satellite networks. *IEEE Journal on Selected Areas in Communications*, 34(12):3869–3886, 2016.