

Article

Protocol-Compliant Back-to-Origin Amplification in Satellite-Enabled CDNs

Bingru Jia , Xiaodan Yan * , Yongli Zhao , Ang Cao , Zhaochen Zhu and Yuqing Sun

School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China; 2024110591@bupt.cn (B.J.); yonglizhao@bupt.edu.cn (Y.Z.); ca0522@bupt.edu.cn (A.C.); zhaochenzhu@bupt.edu.cn (Z.Z.); yuqingsun@bupt.edu.cn (Y.S.)

* Correspondence: xiaodanyan@bupt.edu.cn

Abstract

Satellite-enabled content delivery networks (S-CDNs) are needed to improve content access for wide-area, remote, and highly dynamic users, for whom relying only on terrestrial CDN infrastructure is often insufficient. However, in conventional CDN systems, request acceptance and origin fetching are not tightly coupled. In satellite-ground environments with high latency, dynamic connectivity, and limited observability of requester state, low-cost and fully protocol-compliant requests may still trigger disproportionate backend work. This paper attributes the problem to open-loop request admission and studies two protocol-compliant paths. The range path is an instance of a known CDN cost asymmetry. The second path, called “Disconnection-induced amplification after request submission,” builds on known connection-decoupling behavior. Its attack realization is a protocol-compliant requester-side stop inspired by satellite coverage motion. The stop emulates the observable effect of a user leaving the service area after task triggering. The experiment evaluates this emulation. Experiments on an in-orbit prototype and a matched terrestrial configuration, together with supplementary analysis, show substantial back-to-origin amplification along both paths: the smallest byte-range request on uncached objects reaches a peak amplification of about 15,000, while the disconnection-induced path causes persistent satellite-side bandwidth occupation, a lower benign completion ratio, and higher downlink overhead. These findings indicate that cache misses in S-CDNs are not only performance events but also security-relevant points of backend commitment.

Keywords: satellite-enabled CDN; back-to-origin amplification; edge caching; cache miss handling; requester unreachability; network measurement



Academic Editor: Christos J. Bouras

Received: 4 September 2026

Revised: 2 October 2026

Accepted: 13 October 2026

Published: 15 October 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](#)

[Attribution \(CC BY\)](#) license.

1. Introduction

Low Earth orbit (LEO) satellites are becoming an important part of space-air-ground integrated networks, especially in scenarios such as maritime communications, remote access, and emergency communications [1,2]. Content delivery networks (CDNs), first introduced in the late 1990s, were designed to improve the performance and reduce the cost of delivering web content, media, and downloadable objects over the terrestrial Internet [3,4]. Related studies have further examined cache management and content placement as central mechanisms for improving CDN efficiency, responsiveness, and delivery cost [5,6]. To this end, CDNs deploy large numbers of edge servers around the world to cache and serve content from locations closer to users, thereby reducing user-perceived latency and improving delivery performance [5]. However, recent studies show that relying only on

traditional terrestrial CDNs together with Starlink access cannot stably guarantee user quality of experience (QoE) [7,8]. For example, one study observed that users in Africa may connect to ground stations in Europe and then access CDN servers attached to those stations, whose caches may contain content biased toward European demand [9]. This suggests that, in non-terrestrial network environments, if content delivery still depends mainly on the terrestrial CDN system, regional mismatch may arise between service paths and content supply, thereby weakening the potential benefits in latency and content adaptation. To reduce access latency and recurring origin traffic, existing studies have gradually pushed caching and content-delivery functions toward the satellite edge [10–14]. On this basis, studies on popularity-driven content placement, multi-hop content delivery, distributed inter-satellite cache-decision interaction, and matching-game-based cache placement have further provided a concrete foundation for satellite-side caching and content delivery design [15,16]. In addition, cooperative and resource-aware strategies further improve onboard caching capability under regional dynamics and resource constraints [17,18].

However, these designs place a known CDN problem on a more constrained service chain. Prior terrestrial studies have reported frontend–backend cost asymmetry [19–23]. The Range path here is one such mechanism, not a new amplification primitive. A satellite link does not inherently increase its byte ratio. However, the wasted retrieval crosses a shared satellite–ground link and may occupy limited onboard cache. The second path evaluates a coverage-loss-inspired, protocol-compliant requester-side stop after task triggering. It tests backend persistence in the studied service chain. It does not claim that natural satellite motion caused the requester loss. Legitimate mobility-related disconnections may complicate attribution. However, no detector or stealth advantage is evaluated.

This paper therefore asks whether protocol-compliant, low-rate requests can trigger disproportionate backend work in S-CDNs without relying on malformed packets or volumetric flooding. We study a Range path on uncached objects and a disconnection-induced path after request submission using an in-orbit prototype, matched ground experiments, and supplementary analysis. The second path uses a requester-side stop rather than an explicit reset. The ground experiments reproduce the backend-persistence behavior. The in-orbit experiment measures residual work on the satellite service path.

The paper makes four contributions. First, it separates known Range and connection-decoupling behavior from satellite deployment consequences. Second, it uses open-loop request admission as a common description of both paths. Third, for the disconnection-induced path, it designs and evaluates a coverage-loss-inspired requester-side stop as the attack realization. Fourth, it combines in-orbit measurements with matched ground experiments to distinguish CDN-intrinsic behavior from satellite-path resource consequences.

The remainder of this paper is organized as follows. Section 2 reviews the studies most relevant to satellite edge caching and back-to-origin amplification. Section 3 presents the system context, threat model, and root-cause framework. Section 4 describes the prototype and experimental methodology. Sections 5 and 6 present the two attack paths and their experimental characterization. Section 7 discusses the implications and limitations of this work. Section 8 concludes the paper.

2. Background and Related Work

2.1. Satellite Edge Caching and Content Delivery

Satellite edge caching has been widely studied as an important approach to reducing access latency and recurring origin traffic in space–air–ground integrated networks. Traditional terrestrial CDNs improve delivery performance by deploying large numbers of edge servers around the world and caching content as close to users as possible, thereby reducing user-perceived latency [3,5]. However, recent studies show that, in the era of LEO satellite

networks, relying only on traditional terrestrial CDNs together with Starlink-like access cannot stably guarantee user quality of experience. In some cases, users may even access CDN caches through distant ground stations whose cached content does not match local demand well [7–9]. This suggests that, in non-terrestrial environments, the design assumptions behind traditional CDNs, especially geographic proximity and relatively stable access paths, may no longer hold as well as they do on the terrestrial Internet. It also highlights the need to push caching and content-delivery functions further toward the satellite edge. Along this line, prior work has investigated cooperative multilayer caching, coverage-aware content placement, and prototype implementations of satellite edge caching systems [10–12,24]. Other studies have further examined popularity-driven content placement, multi-hop content delivery, and distributed cache-decision mechanisms in terrestrial–satellite and inter-satellite settings [15,16]. More recent efforts focus more on regional dynamics and onboard resource constraints, for example, through cooperative caching and resource-aware cache management in satellite edge networks [17,18]. In addition, other studies consider hybrid cache-aided satellite–terrestrial architectures, dynamic bandwidth allocation with edge caching, and content-aware routing supported by satellite caching [25–27].

Taken together, these studies show that moving caching and content-delivery functions toward satellite nodes can improve responsiveness and reduce repeated origin traffic. However, most existing work treats cache misses and origin retrieval mainly as performance-management issues, while paying less attention to whether the miss handling path itself may expose a protocol-compliant security trigger. This concern becomes more relevant in satellite environments. LEO satellites move continuously relative to the ground, so coverage and service relationships change over time, and a requester may leave the coverage of the serving satellite during an ongoing transfer. At the same time, satellite–ground links are bandwidth-limited and expensive, so extra origin retrieval after a miss often incurs a higher system cost than in terrestrial settings. These conditions, taken together, suggest that miss handling in S-CDNs is not only a performance issue but also a potential entry point for later security analysis.

2.2. Origin-Fetch Abuse and Backend Asymmetry in CDNs

A parallel line of work shows that security problems in CDNs and HTTP-based delivery systems do not arise only from volumetric abuse. They may also stem from request-processing logic and backend coordination mechanisms themselves. Prior studies have identified ambiguity and inconsistency in HTTP request handling [19], forwarding-loop vulnerabilities in CDN deployments [20], and weaknesses in CDN origin validation and backend trust relationships [21]. Most directly related to this paper, recent work further shows that CDN back-to-origin strategies can be abused to create substantial amplification effects even under syntactically valid requests [23]. Taken together, these studies suggest that when frontend request acceptance, backend resource commitment, and actual content delivery are not tightly coupled, attackers may exploit normal protocol interactions to induce backend work that is far more expensive than the frontend cost. As illustrated in Figure 1, the key distinction between cache-hit handling and cache-miss handling is that a local cache hit is handled entirely at the edge, whereas a cache miss may trigger costly origin retrieval over terrestrial network links. The closest match to our Range path is Small Byte Range amplification. A one-byte Range request can still induce object-level origin retrieval, with a reported maximum factor of 43,330 on a 25 MB object [22].

Beyond request parsing and origin-triggering logic, some HTTP delivery mechanisms may further enlarge this frontend–backend decoupling. In streaming or chunked delivery, systems often send data before the full content is ready. This improves responsiveness and processing efficiency. However, once the backend flow has started, the requester’s

receiving state may no longer control upstream origin communication or later delivery. In terrestrial CDNs, these issues are already security problems. They can consume origin bandwidth and cause billing loss, not merely inefficiency. In S-CDNs, an expensive link does not by itself increase the byte ratio. The added impact comes from contention on a shared satellite–ground link, pressure on limited onboard cache, and LEO motion. A requester may lose connectivity before a transfer completes. This possibility motivates the satellite deployment discussion. However, natural connectivity loss was not instrumented in our experiments. We therefore do not treat it as a validated satellite-specific mechanism.

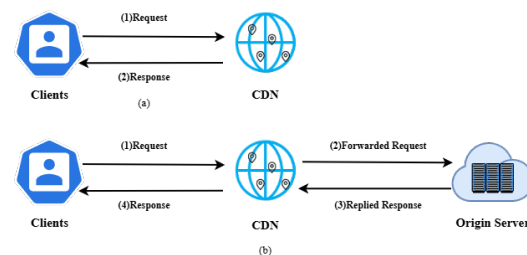


Figure 1. Multiple connection segments and simplified request process in a CDN environment. (a) Cache hit (HIT): the requested resource is available in the edge node’s local cache and has not expired. (b) Cache miss (MISS): the request triggers a fetch from the origin server when the resource is accessed for the first time, has expired, has been updated, or is not cached at the edge node.

2.3. Gap and Positioning of This Work

Although the two lines of work discussed above are closely related, neither directly resolves the question studied in this paper. On the one hand, existing research on satellite caching and satellite content delivery mainly focuses on performance issues such as cache deployment, content placement, cooperative delivery, and resource scheduling. Much less attention has been paid to whether cache-miss handling and origin retrieval themselves may constitute a security attack surface. In other words, prior work usually treats the cost of origin retrieval after a miss as an efficiency issue, rather than asking whether this path can also be exploited by fully protocol-compliant requests to induce disproportionate backend resource consumption.

Terrestrial CDN studies already explain the Range path. A legal small Range request, a cache miss, and coarse-grained origin retrieval create frontend–backend byte asymmetry [22,23]. The narrower question is how this known behavior, together with known post-stop persistence, interacts with shared satellite–ground resources and changing requester reachability. We therefore use matched Varnish/Nginx ground experiments to reproduce the Range/MISS, disconnection-induced, and mixed-operation behavior. The in-orbit measurements show the corresponding satellite-path consequences.

3. System Context, Threat Model, and Root-Cause Framework

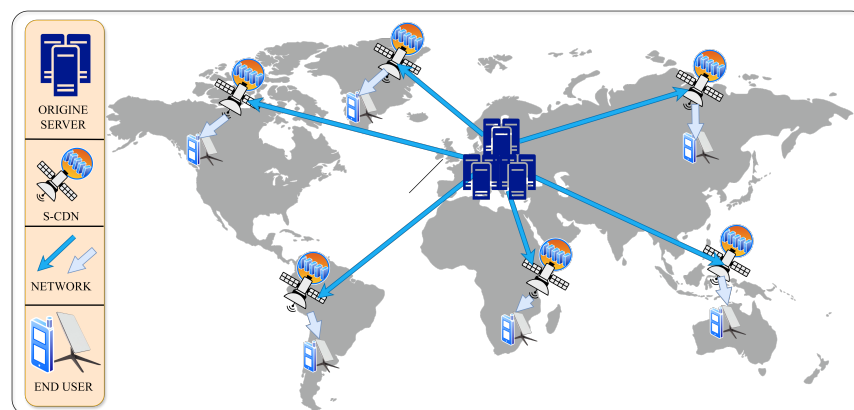
3.1. S-CDN Service Workflow

The S-CDN service chain studied in this paper consists of three basic entities: a requester, a satellite CDN edge, and an origin server. After a request arrives at the edge, the system first parses the request and performs a cache lookup. If the target object is found locally, the edge serves it directly to the requester. Otherwise, the edge enters the cache-miss handling path, creates a backend retrieval task, and fetches the object from the origin server. In this process, the edge acts not only as the receiver of frontend requests, but also as the executor of backend origin fetching and content forwarding. As a result, its miss handling behavior directly determines how subsequent system resources are consumed. The main differences between existing CDN studies and the present work are summarized in Table 1.

Table 1. Existing CDN studies and this paper, compared by aspect [20,22,23,28].

Aspect	Existing Studies	This Paper
Attack trigger	Small Range, HEAD, resize, large header, protocol conversion, active reset, or timed requests.	One-byte Range. Coverage-loss-inspired requester-side stop on the second path.
Protocol compliance	Legal HTTP; asymmetry from origin handling, not malformed packets.	Legal requests; UDP short session only in the studied prototype.
Cache state	Repeated origin fetch requires a miss.	99 misses and 51 hits; a hit stops the fetch.
Origin-fetch behavior	Whole-object fetch, header expansion, or origin connection kept after reset.	Whole-object fetch on a miss. Backend execution may continue after the requester-side stop.
Requester connectivity	Attacker stays active and keeps or resets the connection.	Range path: requester stays reachable. Second path: attacker stops consumption to emulate coverage departure. Natural coverage loss not instrumented.
Amplification metric	Byte ratio, connection count, or temporal concentration, kept separate. Small-Range maximum: 43,330.	Byte ratio grows with object size; no larger-than-ground claim.
Ground or satellite deployment	Commercial terrestrial CDNs; edge keeps the origin connection.	In-orbit edge and ground origin; bytes enter a shared satellite-ground link.
Experimental validation	Controlled tests on commercial CDNs and author-operated origins.	In-orbit experiments on one prototype. Matched ground experiments with the same core configuration for Range/MISS, disconnection, and mixed operation. Controlled analysis for trend decomposition.

As illustrated in Figure 2, the service workflow considered in this paper has two basic paths. Under a cache hit, content can be returned directly from the satellite edge to the requester. The service path is short, and no additional origin-side retrieval is needed. Under a cache miss, by contrast, the edge must first retrieve the object from the origin server and then forward it along the satellite service path to the requester. This cache-miss path is central to our analysis. It introduces additional origin-side work and cross-link data transfer, both of which are largely absent from the cache-hit path. For an S-CDN, this difference does not only imply higher access latency; it also means that the system begins to consume valuable back-to-origin bandwidth and related scheduling resources.

**Figure 2.** Service workflow in the S-CDN studied in this paper. On a cache hit, the satellite edge serves the requester directly; on a cache miss, it fetches the object from the origin and then forwards it to the requester.

In satellite environments, this cache-miss handling path deserves special attention. Compared with terrestrial CDNs, data fetched from the origin often has to traverse satellite-ground links that are capacity-limited and expensive. Therefore, the system cost of the same cache miss is usually much higher in satellite settings. In other words, a cache miss is not only a performance event but also the direct starting point of later origin-side cost and link-resource consumption. This point is especially important for the attack paths studied

in this paper, because although the two paths differ in surface behavior, both depend on the fact that once the system enters the miss handling stage, it may trigger substantially more expensive backend execution.

The prototype studied in this paper uses HTTP-style requests over a UDP-based short-session service. Under this service model, frontend request acceptance is more easily separated from strong delivery evidence than in a more tightly stateful setting. In other words, once a request is accepted and enters the cache-miss handling path, backend work may already have started, even though the system may not yet be able to sufficiently confirm whether the requester remains reachable and whether continued delivery is still meaningful. This does not necessarily imply a flawed system design. Rather, it is a natural consequence of using weakly stateful short-session services to preserve service continuity in a dynamic satellite environment.

It is important to note that this workflow is not, by itself, a source of malicious behavior. Under normal operation, it helps maintain content-delivery continuity in highly dynamic and intermittently connected satellite environments, and it can reduce service disruption caused by link fluctuations. However, when frontend requester cost and backend execution cost are not tightly coupled, the same workflow can also expose an attack surface. In this sense, cache-miss handling in an S-CDN is not only a normal component of content delivery but also the foundation for the later security analysis in this paper.

3.2. Open-Loop Request Admission and Execution Asymmetry

We describe the above root cause as open-loop request admission. Specifically, syntactic acceptance alone may trigger backend execution before sufficient delivery evidence is available. Once admitted, a request may lead to object-level task creation, origin-side communication, and cross-link data transfer, even though the requester contributes only a small amount of frontend traffic.

This mechanism creates execution asymmetry. The attacker-side cost is low because the triggering request is legal, lightweight, and low-rate. The backend cost, however, can be much higher because the edge may commit to expensive miss handling work, including origin retrieval and satellite-ground transmission. Importantly, this asymmetry does not depend on malformed packets, spoofing, or volumetric flooding. Instead, it arises because the system commits backend resources before obtaining sufficient evidence about actual downstream delivery.

The same root cause may appear in different forms. The first involves request granularity and origin retrieval granularity. A very small byte-range request may still induce near-object-level retrieval under a cache miss. The second involves downstream consumption and backend execution state. After task triggering, requester-side consumption may stop while backend work continues. These cases differ in surface behavior. However, both expose the same open-loop commitment of backend resources.

From this perspective, the security issue studied here is not simply a conventional traffic amplification problem caused by packet-level protocol abuse. Rather, it is a form of logic-level amplification caused by backend execution that starts too early and stops too late. This distinction is important because it explains why low-rate and fully protocol-compliant requests can still induce substantial origin-side cost in satellite environments.

3.3. Threat Model and Assumptions

We consider an attacker who can issue protocol-compliant requests to the S-CDN edge through the same service interface as ordinary users. Triggering the basic mechanisms does not require malformed packets, identity spoofing, or high sending rates. Instead, the

attacker exploits normal request semantics and the gap between frontend admission and backend execution.

The attacker is assumed to know accessible objects and may select likely uncached objects or vary query strings. These choices are subject to the cache-key and origin policies discussed below. The attacker may also stop consuming data after task triggering. The experiments implement this condition through a coverage-loss-inspired, controlled requester-side stop rather than an explicit reset. We use requester loss as an application-level description of absent downstream consumption. We do not treat it as evidence of a particular transport or orbital cause. Coverage switching, relative motion, and link obstruction are plausible deployment causes. However, they were not separately instrumented or reproduced in the present measurements. Such benign disconnects can occur during normal LEO operation. An attacker who deliberately stops receiving after task triggering may therefore resemble an ordinary coverage-loss event. We treat this as a threat-model implication, not a measured detection result.

Perfect knowledge of the cache state is not required. Exposed cache-status metadata can reveal whether a request was a hit or miss. Otherwise, response timing, object popularity, and cache lifetime may help estimate the miss probability. These signals are subject to link variability and cache updates. Probing may itself populate the cache, so it does not guarantee a later miss. Query-string perturbation requires distinct cache keys and valid target content from the origin. Cache-key normalization limits this opportunity.

Operational controls constrain the attack. Rate limiting and per-identity quotas restrict repeated miss-triggering requests. Authentication restricts access, but authorized users still need resource budgets. Cache-key normalization limits the number of distinct entries created by irrelevant parameters. Coalescing concurrent requests for the same key reduces duplicate fetches, but it does not remove the backend work of the initial miss. Admission control limits task initiation. Outstanding-task limits constrain accumulation. Timely backend cancellation limits persistence.

A single requester can trigger the studied execution asymmetry. However, sustained and meaningful resource consumption depends on the admitted request rate, miss probability, per-fetch cost, and available resources. The per-request amplification factor alone does not determine impact. Strict per-source or per-identity limits constrain sustained single-requester impact. Multiple requesters may increase aggregate pressure, but they are not necessary to trigger the mechanism. They also do not guarantee that global admission controls or resource budgets can be exceeded. For test-environment safety, the in-orbit experiments use a controlled single requester to validate the mechanisms. The high-load simulations instead examine how aggregate backend work affects benign traffic.

The defender in our setting is the S-CDN service chain itself, including the satellite edge cache and the origin path behind it. Our goal is not to study authentication bypass, cache poisoning, or generic denial-of-service flooding. Instead, we focus on whether protocol-compliant requests can induce disproportionate origin-side work, and whether such work can further degrade benign traffic under shared scheduling.

We also assume that the studied system does not always bind backend continuation tightly to real-time requester reachability. This assumption supports service continuity under unstable links. It also makes the attack surface meaningful. Under this setting, the attacker does not need to break the protocol. The attacker only needs to exploit normal cache-miss handling and the loss of downstream consumption after a request has been accepted.

3.4. Scope of Analysis

Under the above model, this paper studies two representative attack paths. The first is a range-based amplification on uncached objects. It captures request-to-origin asymmetry at

cache-miss admission time. The second is disconnection-induced amplification after request submission. It captures continued backend execution after requester-side consumption is deliberately stopped. We use these paths as representative manifestations of the same root cause. We do not claim that they are the only possible attacks in this model.

The first path highlights a mismatch between frontend request granularity and backend retrieval granularity. The second highlights a mismatch between requester-side availability and backend execution persistence. Studying the two paths together helps distinguish what is specific to one triggering condition from what is fundamental to the underlying service logic.

The analysis is intentionally limited to amplification triggered by protocol-compliant requests. We do not claim that all satellite content-delivery systems behave in the same way. We also do not claim that these two paths exhaust the full attack surface. The purpose of this section is to define the common system context and causal framework needed to interpret the later results. We aim to separate the known Range mechanism and known connection-decoupling behavior from the satellite deployment context. We then test both paths on the studied service chain. This section is not intended to present a complete defense architecture. Design implications are discussed later, but they remain secondary to the problem formulation and empirical characterization developed in the following sections.

4. Prototype and Experimental Methodology

4.1. In-Orbit Prototype and Terrestrial Baseline

We adopt a two-level experimental strategy that combines in-orbit validation with supplementary analysis. The in-orbit experiments show that the Range path and the disconnection-induced path can occur in a real satellite service chain. For the latter, the in-orbit run evaluates the coverage-loss-inspired requester-side stopping strategy described above. The supplementary analysis adds controlled repetition. This helps isolate the effects of cache state, object size, and requester-loss handling. This design preserves realism while improving interpretability and repeatability.

The prototype consists of three main entities: an in-orbit LEO node acting as the S-CDN edge, a ground origin server, and a requester. The satellite edge runs Varnish to perform cache lookup, request handling, and miss processing. The origin server runs Nginx and stores the original objects requested through the S-CDN path. The main hardware and protocol parameters of the onboard platform are listed in Table 2. The studied service uses HTTP-style requests over a UDP-based short-session service. In other words, requests follow HTTP semantics such as GET and Range, while the service itself does not rely on a strongly stateful end-to-end session model.

Table 2. Onboard S-CDN platform specification.

Component	Specification
CPU	FT-2000/4, 4-core, 2.0 GHz
Memory	8 GB DDR4
Accelerators	2 × Kintex KA200 FPGAs (8 GB GDDR6 each)
Storage	2 × 80 GB SSDs (mirrored)
Power Input	5 V DC/10 A
Telemetry	UART + JTAG
Data Link	100 Mbps UDP (bridged to RF)
Protocol Stack	Custom UDP-based short-session service with HTTP-style semantics

4.2. Protocol-Stack Representativeness and Cross-Stack Applicability

The custom UDP-based short-session service is the actual service stack used by the in-orbit prototype. We do not claim that it is a universal or standardized transport stack

for all current or future S-CDNs. Satellite broadband systems commonly expose IPv4 or IPv6 connectivity to users. Conventional Internet protocols may then be used. Depending on the deployment, a web service may use HTTP/1.1 or HTTP/2 over TCP. It may also use HTTP/3 over QUIC or an implementation-specific service above a satellite access and gateway layer [29–31]. Other satellite scenarios may use store-and-forward architectures with different session semantics. These are not direct substitutes for an HTTP CDN session and are outside the present measurements.

The UDP implementation is therefore a realization of the experimental service chain, not the claimed root cause. The common abstraction is an edge cache that accepts a protocol-compliant request and commits to cache-miss or origin processing. It may continue backend execution even when downstream delivery is no longer useful. The Range path mainly depends on the mismatch between requester-visible granularity and origin-retrieval granularity. It can therefore occur over TCP-based HTTP when the cache and origin apply the same coarse-grained miss handling. The disconnection-induced path depends more on transport and implementation. A stronger frontend cancellation signal may reduce persistence. However, it does not guarantee that an already-started cache-fill or origin-fetch task is rolled back.

HTTP/3 should also be distinguished from the custom UDP service. QUIC is a reliable, encrypted, multiplexed transport over UDP. It provides stream-level control and connection migration [30]. HTTP/3 maps HTTP semantics onto QUIC streams and defines request-stream cancellation [31]. These mechanisms can make frontend cancellation and path changes more explicit. However, cancellation is not guaranteed to propagate into CDN or origin-side execution. This remains an implementation issue. We therefore treat cross-stack applicability as a conditional architectural analysis. We do not claim that the measured amplification magnitude is identical for HTTP/1.1, HTTP/2, and HTTP/3/QUIC.

In the supplementary ground setup, we retain the same basic service logic, request semantics, and cache-miss handling behavior as in the in-orbit prototype. We also performed repeated experiments in a real Docker-based terrestrial deployment. This deployment used the same core Varnish/Nginx versions, containerized service chain, object set, request procedures, cache-miss path, and requester-stop conditions. Table 3 summarizes the results. These runs consistently reproduced the CDN-side byte asymmetry and backend persistence observed in orbit. Under the strictly controlled configuration, the Range/MISS comparison showed no observable increase in the byte-amplification ratio on the satellite path. We therefore attribute the mechanism to CDN service logic. The satellite setting changes the deployment consequence: the same unnecessary work crosses a constrained, shared satellite service path. The in-orbit evidence consists of recorded point observations rather than repeated independent orbital runs. This comparison is therefore descriptive. It is not presented as a statistical equivalence test.

Table 3. Isolation of CDN-intrinsic behavior from satellite deployment consequences. The terrestrial observations summarize repeated matched Docker experiments; the in-orbit column reports the recorded prototype observations.

Path	Repeated Ground Baseline	In-Orbit Observation	CDN-Intrinsic Component	Satellite-Side Consequence
Range/MISS	Object-scale backend fetch; about 1.50 MB/fetch	Same asymmetry; about 1.50 MB/fetch; no byte-ratio increase	Cache-miss handling; granularity mismatch	Shared satellite-ground path; onboard-cache pressure
Controlled requester stop	Backend work continued after stop	Residual work remained on measured satellite path	Frontend-backend decoupling; delayed cancellation	Satellite transmission consumed after benefit ends
Mixed operation	Same benign-service degradation direction	Same direction under shared scheduling	Persistent backend work vs. benign traffic	Shared satellite-side resource contention

4.3. Implementation Details and Reproducibility

The satellite edge ran Varnish 7.6 with VCL 4.1. The origin ran Nginx 1.27.4 in a Debian bookworm-based container. Varnish handled edge caching, and Nginx served the origin objects. The complete implementation parameters are summarized in Table 4.

Table 4. In-orbit prototype implementation parameters.

Item	Recorded Setting
Edge software	Varnish 7.6; VCL syntax 4.1; HTTP listener on port 80
Origin software	Nginx 1.27.4; origin address 172.18.0.3:80; objects mounted under /var/www/html
Cache storage	File storage; nominal capacity 1 GB; default TTL 120 s
Object TTLs	Static/image objects: 63 s; compressed, audio, and video suffixes: 30 min; grace period: 5 min
Cache key	Request URL followed by the Host value; query strings therefore remain part of the key
Backend timers	Connect timeout: 3 s; first-byte timeout: 9 s; between-bytes timeout: 1 s
Health probe	/every 5 s; probe timeout 1 s; window 5; threshold 3
Transport	Custom UDP short-session service; command ports 20000 and 20002; fixed 36-byte binary header

The VCL caches GET/HEAD, passes Authorization, and removes cookies for static resources. The cache key combines req.url and Host. URL normalization, request coalescing, and rate limiting are not configured. PURGE is restricted to local and internal addresses. The cache uses file-backed storage with a capacity of 1 GB and Varnish LRU eviction. The 100-miss block recorded 34 evictions. Runtime statistics use the SMA storage counters.

The origin used the arm64v8/nginx image with a bind-mounted configuration at /nginx-custom/nginx.conf. Its key settings were an HTTP listener on port 80, document root /var/www/html, and direct static-file serving from /home/Pictures. No proxy cache, URL rewrite, application session, or retransmission logic was added at the Nginx layer. Range processing used the standard Nginx HTTP implementation. In the representative probe, Range: bytes=0-0 produced 206 Partial Content, Content-Range: bytes 0-0/total, and Content-Length: 1.

The UDP service sends one protocol data unit per datagram in network byte order. Its fixed 36-byte header contains the synchronization word 0x1ACFFC1D, packet type, sequence number, file number, 6-byte file length, 6-byte data offset, 4-byte data length, and CRC32C. The payload follows the header. The command client uses ports 20000 and 20002, receives up to 4096 bytes per call, and has a 1 s receive timeout. This timeout is not the content-session requester-loss detector. Link-layer RTO and retransmission counters were outside the experimental scope and were not collected. The application-layer client has no retry loop or recorded RTO.

HTTP header lengths are variable. The range probe used Range: bytes=0-0. Its response included server, date, content type, validator, cache status, range, length, and connection fields. For the 100-miss block, Varnish recorded 17,121 aggregate request-header bytes and 62,238 aggregate response-header bytes. In orbit, requester loss was created by deliberately stopping downstream consumption after task triggering. Continued satellite-side transmission was then observed. The trace did not record a separate transport-layer loss signal, signal-strength transition, handover event, or calibrated coverage threshold. It also did not record synchronized timestamps for request submission, expected coverage loss, client-confirmed loss, and server-side detection. The experiment therefore cannot attribute the stop to satellite motion. It also cannot distinguish that stop from an active network shutdown, process exit, or a client that stops reading. In the Python 3.13 control model, disconnection occurs at approximately 64 KB with 16 KB jitter and a 32 KB lower bound. Detection and cancellation delays are 0.32 ± 0.11 s and 0.38 ± 0.14 s, respectively. Origin fetching remains enabled, and the attack grace period is 160 s. These are model parameters, not in-orbit transport thresholds.

4.4. Object Set, Cache-State Control, and Attack Realization

The object set consists of image objects ranging from 1 to 20 MB. This range is consistent with the current prototype and covers both small-object and large-object regimes for later analysis. In all experiments, the objects are stored at the origin server. The S-CDN edge serves them directly only on a cache hit; otherwise, it fetches them from the origin.

To describe cache state compactly, let r denote a request, let $k(r)$ denote its cache key, and let $\mathcal{C}(t)$ denote the set of objects cached at the edge at time t . A request is considered a cache hit according to Equation (1):

$$I_{\text{hit}}(r, t) = \begin{cases} 1, & k(r) \in \mathcal{C}(t), \\ 0, & k(r) \notin \mathcal{C}(t), \end{cases} \quad (1)$$

and the corresponding miss indicator is given by Equation (2):

$$I_{\text{miss}}(r, t) = 1 - I_{\text{hit}}(r, t). \quad (2)$$

For a request sequence $\mathcal{R}$, the cache-hit ratio is defined by Equation (3):

$$H = \frac{\sum_{r \in \mathcal{R}} I_{\text{hit}}(r, t_r)}{|\mathcal{R}|}, \quad (3)$$

where t_r is the arrival time of request r , and the miss ratio is $M = 1 - H$. In the service chain studied here, a miss is important because it may trigger backend retrieval. Let $s(r)$ denote the size of the requested object. Under object-level miss handling, the bytes fetched from the origin for a miss request satisfy Equation (4):

$$B_{\text{origin}}(r) \approx s(r), \quad \text{when } I_{\text{miss}}(r, t) = 1, \quad (4)$$

even if the requester-visible payload is much smaller. This property is central to the two attack paths studied in this paper.

A request is treated as effectively uncached if either its target object is absent from $\mathcal{C}(t)$ or its cache key is intentionally perturbed so that $k(r)$ does not match any existing cached entry.

To ensure stable cache-miss triggering, we reset or perturb the cache state between rounds when necessary. When the purpose is to evaluate request-triggered amplification under a miss, we select target objects and control the cache state so that the requested object is uncached at the time of arrival. This avoids mixing cache-hit returns with the backend cost induced by miss handling.

For the range-based amplification on uncached objects, the attack uses legal HTTP-style Range requests to trigger miss handling. We take Range: bytes=0-0 as the minimal representative case and later vary the requested range length in the sensitivity analysis. The goal is to test whether, under a cache miss, even a very small byte-range request can still induce near-object-level origin retrieval. To avoid conflating amplification with queuing effects, the core measurements for this path are conducted with sequential requests to distinct target objects.

For the disconnection-induced amplification after request submission, the experiment does not infer a transport-layer or orbital cause. To keep requests effectively uncached, we append a unique query suffix, such as ?uid=timestamp. Shortly after request submission and backend task creation, requester-side consumption is deliberately stopped. The stop uses a coverage-loss-inspired, protocol-compliant strategy rather than an explicit reset. We then observe whether the edge continues origin fetching and subsequent transmission even though meaningful downstream delivery is no longer supported. Coverage switching,

relative motion, and link obstruction motivate the strategy. They are not claimed as the experimentally realized trigger.

It is important to emphasize that neither path depends on malformed packets, abnormal fields, or high sending rates. Both use protocol-compliant requests. The first path exploits a mismatch between request granularity and origin retrieval granularity. The second exploits a mismatch between requester-side consumption and backend execution state. This distinction matters in the later analysis. It shows that the observed effect is not conventional flooding. Rather, it is execution amplification under open-loop request admission.

4.5. Experimental Scenarios and Data Collection

We use two types of experimental scenarios, as summarized in Table 5. The first is a pure-attack setting, which is used to quantify the origin-side work directly triggered by attacker requests. In this setting, benign traffic and unrelated scheduling effects are minimized, so that the measured amplification can be attributed as directly as possible to induced backend execution. The core measurements for the range-based path are mainly conducted under this setting.

Table 5. Experimental scenario settings.

Setting	Main Purpose	Typical Use in This Paper
Pure-attack	Measure request-triggered backend work under controlled conditions	Core amplification analysis for the range-based path
Mixed benign/attack	Measure spillover to benign traffic under shared scheduling	Collateral-impact analysis for the disconnection path

The second is a mixed benign/attack setting, which is used to evaluate collateral impact on ordinary traffic. In this setting, benign requests and attacker requests coexist in the same service chain. This allows us to examine whether extra origin fetching and continued execution triggered by attacker requests degrade normal service under shared scheduling. This setting is particularly important for the disconnection-induced path, because the harm of that path is not limited to wasted origin traffic. It may also appear as persistent consumption of satellite-side bandwidth and reduced service quality for benign users.

In the in-orbit experiments, the main objective is to establish direct evidence for the two paths in a real satellite service chain. For the range-based path, we measure the relation between very small requester traffic and the induced volume fetched from the origin under sequential cache-miss requests. For the disconnection-induced path, we record the satellite-side bandwidth trend after requester loss in order to determine whether backend work persists after the requester becomes unreachable.

The supplementary ground analysis is used for controlled decomposition. For the range-based path, we vary the requested range length. We then compare bytes fetched from the origin with requester-visible traffic. For the disconnection-induced path, we vary object size and evaluate both amplification and collateral effects under mixed traffic. This design separates the effects of cache state, request granularity, object size, and requester loss. It does not change the core service logic of the studied service chain.

4.6. Measurement Metrics

For a request, or a batch of requests, that triggers B_{origin} bytes of origin retrieval, we report two amplification metrics:

$$A_{\text{req}} = \frac{B_{\text{origin}}}{B_{\text{req}}}, \quad A_{\text{vis}} = \frac{B_{\text{origin}}}{B_{\text{vis}}}, \quad (5)$$

where B_{req} denotes attacker-generated request bytes and B_{vis} denotes requester-visible bytes, including response headers and returned payload when applicable. These metrics are defined in Equation (5). We use A_{req} as the primary metric because it more directly captures adversarial cost. We report A_{vis} when requester-visible traffic is also important for interpretation, especially in range-based supplementary analysis.

For budget-constrained supplementary experiments, we further report the total induced origin traffic defined in Equation (6):

$$B_{\text{origin}}^{\text{tot}} = \sum_{i=1}^n B_{\text{origin},i}, \quad (6)$$

and the budget-level amplification

$$A_{\text{budget}} = \frac{B_{\text{origin}}^{\text{tot}}}{U}, \quad (7)$$

where $B_{\text{origin},i}$ is the origin-side retrieval induced by the i -th request, n is the number of requests that can be issued under the corresponding budget, and U denotes the fixed requester-side budget used in that batch. The budget-level amplification is defined in Equation (7). In our supplementary Monte Carlo analysis, this metric is used to quantify how much aggregate origin-side work can be induced under a bounded requester-side cost.

For mixed-traffic experiments, we further evaluate benign completion ratio and downlink overhead:

$$R = \frac{N_{\text{completed}}}{N_{\text{total}}}, \quad (8)$$

$$O = \frac{B_{\text{link}}}{B_{\text{app}}}, \quad (9)$$

where $N_{\text{completed}}$ and N_{total} denote the numbers of completed and total benign sessions, respectively, B_{link} denotes downlink transmitted bytes, and B_{app} denotes delivered application-layer bytes. While A_{req} , A_{vis} , and A_{budget} characterize attacker-side efficiency under different accounting views, R and O capture whether the same execution asymmetry spills over to benign traffic under shared scheduling. The completion ratio and downlink overhead are defined in Equations (8) and (9).

This distinction is especially important for the disconnection-induced path. In that case, the damage is not limited to wasted origin traffic. It may also appear as prolonged contention after requester loss. For this reason, the later evaluation does not stop at amplification alone. It also examines whether continued backend execution after requester unreachability degrades benign-service performance under shared scheduling.

4.7. Statistical Reporting and Evidence Classes

We distinguish point observations from repeated terrestrial experiments and repeated controlled simulations. The in-orbit Range/cache-state records and downlink trace are point observations from one recorded experiment. We report them without artificially constructed variance or confidence intervals. The real Docker-based terrestrial experiments were repeated under the matched configuration. They consistently reproduced the same

outcomes. The detailed terrestrial Range/MISS record reported here is one representative complete run. It contains 150 requests (99 misses, 51 hits) and 100 backend fetches. The repeated ground runs establish descriptive reproducibility. They do not provide a statistical equivalence test against the single in-orbit records. The controlled supplementary analyses have repeated samples. The Range Monte Carlo outputs use 1800 trials per configuration for the range and object-size panels, 1400 trials for the budget panel, and 2200 trials for each robustness panel. Their shaded intervals are empirical 10th–90th percentile ranges, not confidence intervals. The default mixed-traffic full-stack simulation uses 50 scenario repetitions per object size. A scenario repetition, not an individual correlated session, is the primary statistical unit. The raw output records the repetition identifier and random seed. The summary CSV reports both session counts and repetition counts.

Results for a repeated sample include the mean, standard deviation, relevant percentiles, and a two-sided 95% confidence interval (CI) across scenario repetitions. Completion ratio uses a Wilson binomial interval. Downlink overhead is calculated within each repetition as total downlink bytes divided by total delivered application bytes. It is then summarized across repetitions. For comparisons between the default attack and no-attack simulations, we report effect sizes and Welch and Mann–Whitney tests. We apply Holm adjustment to the family of comparisons. We also provide two exploratory summaries: an equal-weight global condition analysis across object sizes and a size-trend analysis. These do not replace the pointwise results. For the seed-matched ground/satellite counterfactual, we compute paired differences $\Delta = x_{\text{sat}} - x_{\text{ground}}$ over 50 replications. These counterfactual comparisons are simulation results. They are not additional in-orbit or same-configuration terrestrial measurements.

To directly test feeder-link collateral impact, we performed a preregistered confirmatory sensitivity experiment at a 2 MB object size. It compares a no-attack condition (F0) with a high-intensity feeder-interference condition (F5; attack arrival rate 2.10/s and backend attack cap 90). The experiment uses 120 paired scenario repetitions. The benign catalog was enlarged to 10,000 objects, and the benign concurrency cap was set to 1000. This makes source-to-satellite origin contention, rather than cache-hit admission, the main source of comparison. The primary endpoint is the completion ratio of benign cache-miss transfers within a fixed 180 s deadline. The paired difference is tested with a paired *t*-test and a Wilcoxon signed-rank test. Completion time median and P95 are secondary endpoints. This condition is a controlled high-load sensitivity case, not a universal default workload.

5. Range-Based Amplification on Uncached Objects

5.1. Attack Path and Triggering Conditions

We first study the range-based amplification on uncached objects, which captures request-to-origin asymmetry at cache-miss admission time. The attack exploits a mismatch between fine-grained frontend demand and object-level backend retrieval under a cache miss. This mismatch can arise naturally in content-delivery scenarios where the requester needs only a very small part of an object, while the edge still relies on coarse-grained miss handling to retrieve content from the origin.

The attack uses a legal HTTP-style Range header to request only a very small prefix of the target object. We use Range: bytes=0-0 as the minimal representative case in the core path description and later consider larger small-range settings in the sensitivity analysis. The key precondition is that the target object is uncached at the S-CDN edge when the request arrives. Under this condition, the request is syntactically valid and lightweight at the frontend, but it may still trigger expensive backend work once the edge enters the cache-miss handling path.

As illustrated in Figure 3, the attack path can be summarized in four steps. First, the attacker selects an object that is not currently cached at the edge. Second, the attacker sends a very small range request for that object, with Range: bytes=0-0 as the minimal representative case. Third, after parsing the request and confirming the cache miss, the edge creates a backend retrieval task and initiates origin fetching. Fourth, although the requester asks for only a minimal byte range, the edge may still fetch most or all of the object from the origin under object-level miss handling, and then return only the requested range to the requester. The essential mismatch is therefore between frontend request granularity and backend retrieval granularity.

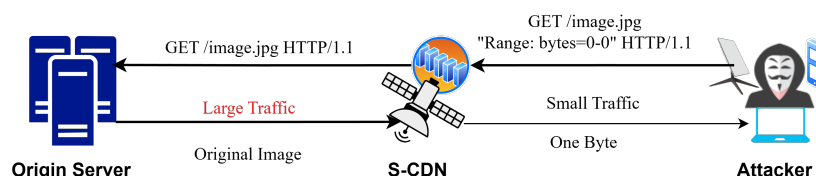


Figure 3. Range-based origin amplification under cache miss in an S-CDN.

This path matters in the satellite setting, but not because it creates a larger byte ratio. The ratio is still set mainly by object size and miss handling granularity. In an S-CDN, the fetched bytes cross a shared satellite–ground link and may occupy limited onboard cache. The satellite setting therefore changes the consequence of the same miss, not the amplification formula.

To expose this effect clearly, we enforce cache misses and use sequential requests to distinct target objects. This design avoids conflating request-triggered amplification with parallel queuing effects or cross-request interference. As a result, the measured backend cost can be attributed more directly to miss handling itself, rather than to contention among concurrent transfers. It also allows the later evaluation to focus on the causal question of interest: whether a very small range request can still induce near-object-level origin retrieval under a cache miss.

5.2. In-Orbit Evidence

We next provide direct in-orbit evidence for the range-based amplification on uncached objects. This subsection characterizes the path along three dimensions: request granularity, object size, and cache state. The question here is straightforward: in the operational satellite service chain, can a very small requester-side input still induce substantial origin-side work once the request falls on the miss path?

It should also be noted that the raw measurements reported in this subsection were collected on the satellite platform and then downlinked to the ground. The subsequent aggregation and plotting were completed on the ground. Therefore, the figures below still reflect actual in-orbit measurements, although the visualization itself was produced offline.

The left-hand plot in Figure 4 shows how request granularity and object size affect the amplification strength. As defined in (5), we use A_{req} to quantify requester-side amplification. Its upper subplot shows that, for a fixed object size, A_{req} decreases as the requested Range becomes larger, and this trend remains consistent across different object sizes. For example, for a 16 MB object, the amplification drops from about 1.5×10^4 at 1 B to about 8.0×10^3 at 16 KB. For an 8 MB object, it decreases from about 8.2×10^3 to about 3.8×10^3 . Even for a 4 MB object, it still declines clearly, from about 4.5×10^3 to about 1.4×10^3 . This result indicates that, under forced misses, reducing the requester-side granularity compresses the frontend cost much more than it reduces the backend work, and the amplification is therefore strengthened.

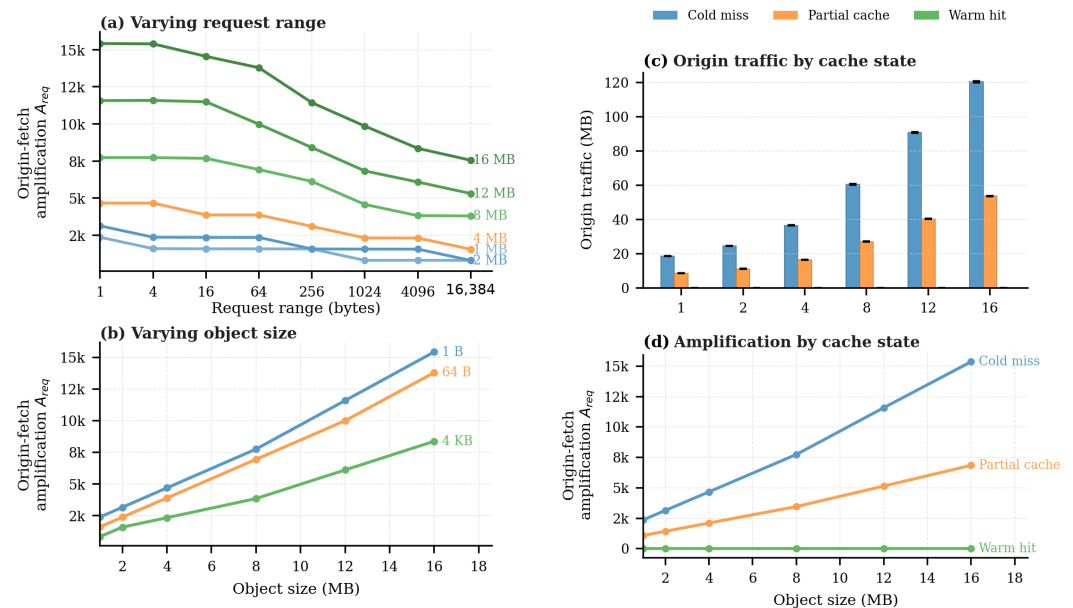


Figure 4. In-orbit sensitivity of range-based amplification. Panels (a) and (b) show request-granularity and object-size sensitivity, respectively. Panels (c) and (d) show origin traffic and requester-side amplification by cache state, respectively. In (c), the green warm-hit bars lie at the zero baseline because a cache hit does not fetch from the origin.

Its lower subplot shows the complementary trend from the object-size direction. For a fixed Range length, the amplification increases steadily with object size. For a 1 B request, A_{req} rises from about 2.6×10^3 at 2 MB to about 1.5×10^4 at 16 MB. For a 64 B request, it increases from about 1.9×10^3 to about 1.35×10^4 . Even when the requested range is enlarged to 4 KB, the amplification still grows from about 9×10^2 to about 6.5×10^3 . This shows that the effect is not determined by “small requests” alone. Instead, it is jointly driven by the mismatch between a tiny Range input and object-scale miss handling. When the object is larger, the origin-side retrieval cost grows, while the requester-side input remains very small. As a result, the overall asymmetry becomes more pronounced.

The right-hand plot in Figure 4 further compares the in-orbit behavior under different cache states. It shows that, under Cold miss, the origin traffic increases rapidly with object size, from about 18 MB at 1 MB objects to about 120 MB at 16 MB objects. Under Partial cache, the origin-side cost is reduced, but it still grows from about 9 MB to about 54 MB. In contrast, under Warm hit, the origin traffic remains close to zero throughout. The corresponding amplification also increases from the low-thousand range to about 1.5×10^4 at 16 MB under Cold miss; under Partial cache, it is about 3.3×10^3 , 5.1×10^3 , and 7.1×10^3 for 8 MB, 12 MB, and 16 MB objects, respectively. Under Warm hit, the amplification remains close to zero. In other words, cache hits can effectively suppress this path, but as long as the system stays in a miss or partial-miss state, a very small byte-range request may still trigger origin-side cost far beyond the requester-side traffic.

These in-orbit results show that the effect is not an isolated spike. It recurs across request granularities, object sizes, and cache states. This links the effect to miss handling rather than transient measurement noise. Cache state also sharply changes the magnitude. Warm hits suppress the path. Misses and partial misses leave a large request-to-origin gap in the operational satellite chain. To separate the CDN-intrinsic component from the satellite-specific consequence, we next compare the core Range/MISS measurement with the matched terrestrial experiment.

The repeated ground Range/MISS experiments use the same core Varnish/Nginx configuration, object set, cache state, and request procedure as the in-orbit experiment.

They consistently reproduce the same behavior. A one-byte client-visible Range response triggers approximately object-scale backend fetching on a cache miss. Average backend bytes per fetch are approximately 1.50 MB in both settings. Thus, no observable increase in the byte-amplification ratio is attributable to the satellite environment under this controlled configuration. The repeated ground requester-stop and mixed-operation experiments likewise reproduce the same qualitative outcomes. These comparisons identify the byte asymmetry and backend persistence as CDN-intrinsic behavior. The satellite-specific significance is that the resulting unnecessary work occupies shared satellite-ground transmission and limited onboard resources.

5.3. Controlled Decomposition Under Cache-Miss Assumptions

To explain these in-orbit trends, we conduct a large-scale supplementary analysis under controlled cache-miss assumptions. Unlike the direct in-orbit measurements, this subsection is used to isolate why request granularity matters and to test whether the effect remains under different budgets, workload distributions, and miss handling policies. The experiments retain the HTTP-style request semantics introduced earlier and the object-level miss handling assumption, while varying the requested Range length, object size, requester-side budget, object-size distribution, and miss handling policy. The amplification measures used below follow Equation (5), while the budget-constrained result uses the metric defined in Equation (7).

For a single request r under this setting, the miss handling asymmetry implies that the visible amplification can be approximated by Equation (10):

$$A_{\text{vis}}(r) \approx \frac{s(r)}{B_{\text{req}}(r) + B_{\text{hdr}}(r) + \ell(r)}, \quad (10)$$

where $s(r)$ is the target object size, $B_{\text{req}}(r)$ is the request bytes, $B_{\text{hdr}}(r)$ is the response-header and other visible overhead, and $\ell(r)$ is the requested Range payload length. Under object-level miss handling, the numerator remains close to the full object size, whereas the denominator grows with requester-visible traffic. Therefore, reducing the requested Range length tends to increase amplification. Although the figures below mainly report A_{req} and A_{budget} , both reflect the frontend-backend gap created by coarse-grained miss handling.

In these experiments, the cache-miss ratio is fixed at 1.0, so every request is treated as a miss. Compared with the earlier supplementary presentation, the updated analysis further decomposes the effect from six aspects: request-granularity sensitivity, object-size sensitivity in terms of requester-side amplification, object-size sensitivity in terms of mean origin fetch, budget-constrained accumulation, distribution robustness, and policy robustness.

Figure 5a first shows the sensitivity of requester-side amplification to request granularity. For each object size, A_{req} decreases as the requested Range becomes larger, and this trend remains consistent across all object sizes. Larger objects also produce higher amplification curves throughout. For example, the 16 MB case remains substantially above the 8 MB, 4 MB, and 1 MB cases over the full range. This result shows that, under forced misses, shrinking the requester-side granularity reduces frontend cost much faster than it reduces backend work. As a result, the asymmetry becomes stronger as the requested byte range becomes smaller.

Figure 5b complements this result from the object-size dimension. It fixes three representative request granularities, namely, 1 B, 64 B, and 4 KB, and then evaluates A_{req} as the target object becomes larger. In all three cases, requester-side amplification increases steadily with object size. The smallest requested range consistently yields the highest amplification, while the 4 KB case remains the lowest. This confirms that larger objects strengthen the trend already observed in Figure 5a.

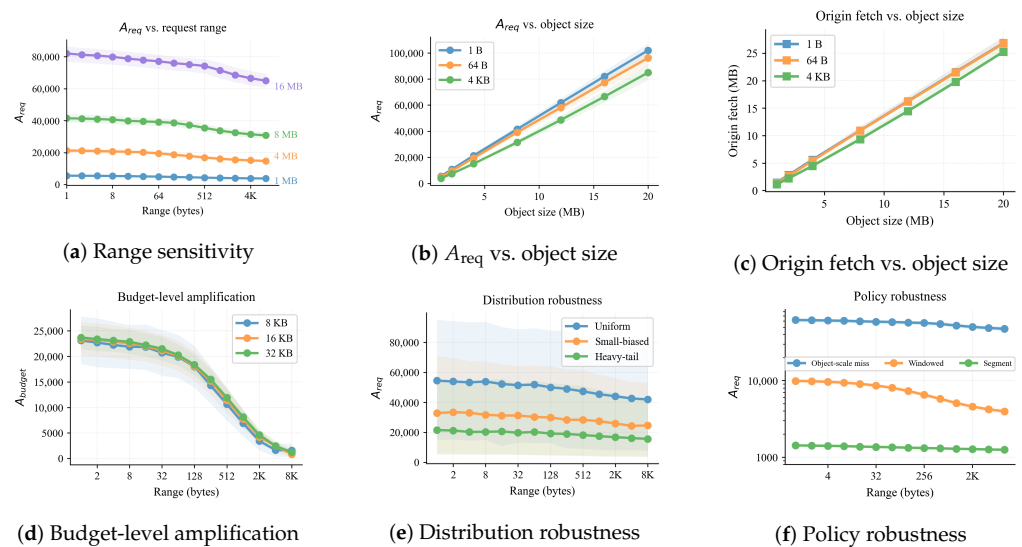


Figure 5. Monte Carlo supplementary results under forced cache misses. Panels (a–f) use 1800, 1400, and 2200 trials per configuration, respectively. Shaded bands are empirical q_{10} – q_{90} ranges.

Figure 5c further reports the corresponding mean origin-fetch volume under the same three Range settings. The mean origin fetch grows clearly with object size in all cases. At the same time, the three curves remain close to each other, which indicates that backend retrieval remains primarily object-scale under forced misses. This explains why amplification rises so strongly with object size: the origin-side work increases with the object itself, while the requester-side input remains very small, especially for fine-grained byte-range requests.

Figure 5d next examines whether this per-request gap accumulates into a strong aggregate effect under a fixed requester-side budget. It reports A_{budget} for three representative budgets, namely, 8 KB, 16 KB, and 32 KB. This metric is used to show that smaller Range requests are harmful not only on a per-request basis but also in terms of cumulative origin-side work under a bounded requester-side budget. The trend is clear. Smaller Range values allow more requests to be issued under the same budget and therefore yield higher budget-level amplification. As the requested range becomes larger, A_{budget} drops substantially for all three budgets. This result is consistent with the single-request analysis. Smaller byte ranges not only increase per-request amplification, but they also allow the attacker to pack more requests into the same requester-side budget. The backend load therefore accumulates more strongly at the aggregate level.

Figure 5e evaluates robustness across different object-size distributions. It compares uniform, small-biased, and heavy-tail settings. Although the absolute levels differ, all three cases follow the same overall pattern: smaller Range values lead to higher A_{req} . The object-size distribution changes the strength of amplification, but not the underlying rule. This suggests that the observed effect is not tied to one specially chosen workload mix.

Figure 5f evaluates robustness across different miss handling policies. More aggressive miss handling yields stronger amplification, whereas more conservative policies reduce it. However, even under the more conservative settings, smaller ranges still provide a clear amplification advantage. This indicates that the effect does not depend on a single extreme implementation. As long as fine-grained requester input is met by coarser backend execution under a miss, small byte-range requests will systematically increase amplification.

Taken together, these supplementary results explain the in-orbit effect: smaller ranges raise A_{req} because requester-side cost shrinks much faster than origin fetch, while larger objects widen the gap because retrieval remains close to object-scale. The same pattern per-

sists under fixed budgets, alternative object-size distributions, and different miss handling policies, so it is not an artifact of one parameter choice.

5.4. Amplification Denominators and Interpretation

The supplementary results also help interpret the amplification metrics reported in this paper. Under request-only accounting, the amplification reaches the 10^4 level. Under client-visible accounting, it remains substantial at $A_{\text{vis}} = 308.7\times$. The difference between A_{req} and A_{vis} is therefore explained by denominator choice rather than by any inconsistency in the measurements. The former counts only attacker-generated request bytes, whereas the latter also includes requester-visible response-side traffic when applicable.

Taken together, the in-orbit and supplementary results show a clear request-to-origin asymmetry under cache misses. The requester contributes only a small amount of traffic, whereas the induced backend work is governed by object-level retrieval behavior rather than by requester-side byte demand.

At the same time, the result should be interpreted together with its triggering conditions. The strongest effect appears when the requested object is uncached and the miss handling logic retrieves most or all of the object from the origin. We do not claim that all S-CDNs must exhibit the same amplification magnitude under all configurations. Rather, the contribution of this section is to show, through direct in-orbit evidence and controlled supplementary decomposition, that such amplification is feasible, measurable, and substantial in the studied S-CDN service chain.

6. Disconnection-Induced Amplification After Request Submission

6.1. Attack Path and Requester-Loss Semantics

We next study the disconnection-induced amplification after request submission. This path captures continued backend execution after the requester becomes unreachable. Unlike the range-based path, the key issue here is not request granularity. Instead, the main question is whether backend work can continue after task triggering even when meaningful downstream delivery is no longer supported.

As illustrated in Figure 6, the path can be summarized as follows. The attacker first submits a legal request for a large object, which causes the edge to enter the cache-miss handling path. Once the request has been accepted, the edge may create a backend retrieval task and initiate origin fetching. Shortly after task triggering, the requester becomes unreachable. If backend continuation is not tightly coupled to requester reachability, the system may continue to fetch data from the origin and may also continue subsequent transmission attempts even though the requester is no longer able to receive the object.

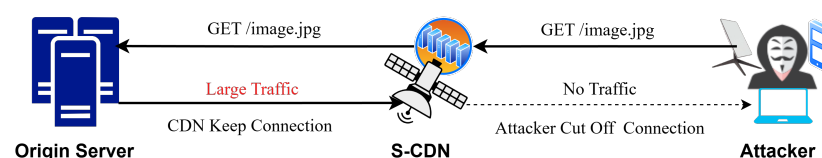


Figure 6. Disconnection-induced origin amplification in an S-CDN. After a request is accepted and miss handling is triggered, the requester may become unreachable while the edge continues origin fetching and subsequent transmission.

This path is consistent with the decoupled structure of content-delivery systems. In practice, the frontend requester–edge interaction and the backend edge–origin interaction are not always terminated at exactly the same time. Whether a backend origin connection is closed immediately after requester-side loss is often implementation dependent rather than strictly dictated by a single uniform rule. As a result, once origin retrieval has started,

requester-side loss does not necessarily imply immediate termination of backend work. Under such conditions, data may continue to flow on the origin-to-edge path even though the requester no longer benefits from the transfer.

More generally, HTTP delivery behavior can further enlarge this gap. The HTTP/1.1 specification permits chunked transfer, in which data is sent progressively before the full content is ready [32]. Such progressive delivery is useful for improving responsiveness and supporting latency-sensitive services [33], while prior work on HTTP handling also shows that implementation subtleties can affect how request and transfer state is interpreted across a service chain [34]. Most directly, prior work has shown that such transmission behavior can allow a CDN to maintain backend origin communication even after requester-side termination, thereby creating a practical amplification vector [23]. For the path studied here, these observations are not used to claim dependence on any single mechanism. Rather, they show that in a decoupled content-delivery architecture, requester-side state changes and origin-side continuation need not remain tightly synchronized.

In this paper, the measured event is a controlled requester-loss condition that implements the coverage-loss-inspired attack strategy. Downstream consumption is deliberately stopped after task triggering. We do not claim an explicit transport reset or a naturally observed orbital outage. The strategy emulates the observable frontend effect of a user leaving the moving satellite coverage area. This application-level condition is not equivalent to a measured transport-layer outage. Coverage switching, relative motion, or link obstruction could produce a similar state in an operational S-CDN. However, the present trace does not distinguish these causes. It therefore does not validate natural passive loss.

This point is operationally relevant in satellite environments because residual retrieval traffic can cross a shared satellite-ground path. The present measurements show occupation of that path after a controlled requester-side stop. They do not quantify a satellite-specific cost premium or the incremental effect of natural coverage loss.

To keep the requests effectively uncached, each request is assigned a unique query suffix when needed, for example, through a `?uid=timestamp` field. This prevents the edge from serving repeated requests from an already cached copy and ensures that each attack request exercises the miss handling path. Under this setup, the path tests whether accepted requests can continue to consume origin-side and satellite-side resources after requester loss.

The essence of this attack is therefore not an abnormal request format, but a mismatch between requester reachability and backend execution state. The requester becomes unreachable after task triggering, yet the edge may still continue origin retrieval and later transmission attempts. Compared with the range-based path, which highlights a mismatch between frontend request granularity and backend retrieval granularity, this path highlights a different but equally important form of open-loop backend commitment in S-CDNs.

6.2. In-Orbit Evidence of Continued Backend Execution

The in-orbit experiment for this path focuses on one question: does satellite-side bandwidth consumption persist after requester consumption is deliberately stopped? The experiment tests execution persistence in a real satellite service chain. It does not test whether a naturally occurring coverage transition causes the requester-loss condition.

In the experiment, the attacker sends legal requests for large objects to the S-CDN edge. When needed, the attacker uses unique query suffixes to keep the requests effectively uncached and trigger miss handling reliably. Shortly after the request is accepted and the backend task starts, the experiment deliberately stops requester-side consumption. Under this condition, the coupling between requester state and backend execution may be weak. Origin retrieval and follow-on transmission may then continue even though meaningful downstream delivery is no longer supported.

Figure 7 provides direct in-orbit evidence of this effect. The horizontal axis shows seconds since the first logged sample, and the vertical axis shows the change in downlink bytes. After requester-side consumption is deliberately stopped, the measured bandwidth does not immediately return to idle. Instead, it remains elevated for a sustained period and exhibits clear continued activity before eventually dropping back. This pattern indicates that backend execution is not interrupted immediately when requester-side consumption stops. The request has already been accepted and miss handling has already been triggered.

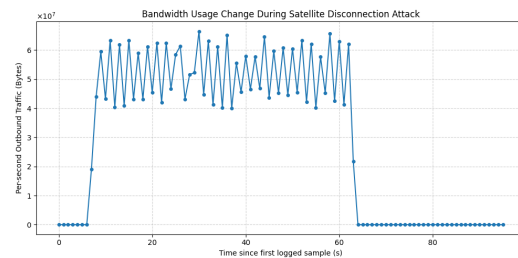


Figure 7. In-orbit evidence under the disconnection-induced path: downlink-byte trend showing continued satellite-side bandwidth consumption after requester-side consumption is deliberately stopped.

This observation is consistent with the root cause developed earlier. Once the system accepts a request and commits backend work, the subsequent origin retrieval and related transmission may continue without timely interruption, even though the requester no longer benefits from the transfer. In this sense, the path is not merely a hypothetical control-flow issue. It appears as a directly measurable source of continued bandwidth consumption in the real satellite service chain.

This result shows that the coverage-loss-inspired requester-side stopping strategy can leave residual backend work on the measured satellite service path. It does not show that satellite motion or coverage variation caused the requester loss. It also does not distinguish the controlled stop from active network shutdown, process exit, or stopped reads. The experiment therefore supports a claim about execution persistence and satellite-path resource occupation. It does not show that passive coverage loss provides additional attack capability or resource consumption.

More importantly, the significance of this result goes beyond bandwidth waste alone. The persistent activity in Figure 7 provides the practical basis for the later mixed-traffic analysis. If backend execution continues after the controlled requester-side stop, the resulting resource occupation may interfere with benign traffic under shared scheduling. The in-orbit result therefore establishes the disconnection-induced path. It also evaluates its coverage-loss-inspired attack realization in the prototype. Natural mobility-induced loss remains to be validated separately.

6.3. Mixed-Traffic Collateral Impact

The main harm of the disconnection-induced path is not limited to wasted origin traffic. Once requester-unreachable tasks continue to consume backend and downlink resources, benign traffic may also be affected under shared scheduling. We therefore evaluate this path under mixed benign/attack traffic, rather than considering amplification alone.

The results in Figure 8 are controlled supplementary statistics. They are not in-orbit raw measurements or the real terrestrial Range/MISS baseline. The default full-stack simulation uses 50 scenario repetitions per object size. The plotted curves summarize one value per scenario repetition. The corresponding session and repetition counts are retained in the summary CSV. In this analysis, attack flows are cache-miss requests that eventually settle. They are grouped by object size. We distinguish three settle reasons: *origin_full*,

cancel_effective, and grace_timeout. For benign flows, we measure completion time, completion ratio, and downlink overhead as defined in Equations (8) and (9). The separate seed-matched counterfactual uses 50 paired replications to compare ground and satellite profiles. It is used only for uncertainty and difference reporting. It is not a replacement for the real ground experiment.

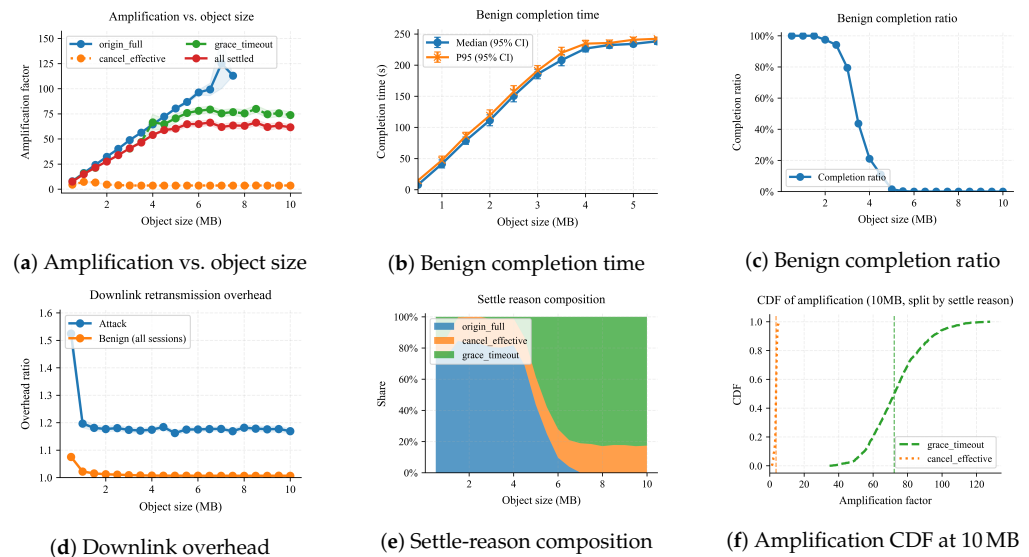


Figure 8. Controlled supplementary statistics for the disconnection-induced path under mixed benign and attack traffic. Curves summarize 50 scenario repetitions per object size; shaded regions or error bars denote 95% confidence intervals where shown, with scenario repetition as the statistical unit. Panel (c) shows the completion ratio only; no confidence interval is plotted in that panel.

The confirmatory feeder-link experiment is reported separately from the default mixed-traffic curves. It uses 120 paired repetitions for F0 and F5 at 2 MB. Its primary endpoint is the fixed 180 s deadline-based completion ratio of benign cache-miss transfers. The default sweep therefore describes size-dependent behavior under the modeled workload. The confirmatory experiment tests whether a deliberately high feeder-load condition produces a measurable collateral effect.

Figure 8a–d show the main mixed-traffic effects. For a given object-size group s , let $\mathcal{S}_s$ denote the set of settled attack samples in that group. The corresponding mean amplification is given by Equation (11):

$$\bar{A}(s) = \frac{1}{|\mathcal{S}_s|} \sum_{i \in \mathcal{S}_s} A_i, \quad (11)$$

Here A_i is the amplification of attack sample i . Figure 8a shows that mean amplification increases with object size. The rise is clear at first, but the curve later enters a high plateau rather than continuing to grow linearly. Figure 8b,c show that benign service quality deteriorates as object size increases. Completion time becomes longer, especially in the tail. Completion ratio drops sharply and approaches zero in the large-object regime. Figure 8d shows the corresponding downlink-overhead trend. Because overhead is a byte ratio, its attack-baseline difference is small even when contention affects completion time and completion probability. The point estimates therefore support spillover as a possible consequence of controlled requester-loss tasks. The strength of the spillover depends on object size and is not statistically significant for every pointwise comparison.

As a global check, we gave each object size equal weight. We then compared the 50 repetition-level averages between the default attack and no-attack conditions. Across all sizes, the completion-ratio difference was -0.0069 (95% CI $[-0.0195, 0.0057]$, Holm-

adjusted $p = 0.854$). The corresponding downlink-overhead difference was -3.72×10^{-4} (unadjusted $p = 0.036$; Holm-adjusted $p = 0.144$). These default-workload results do not justify a universal claim of statistically significant degradation. The confirmatory feeder-link sensitivity experiment addressed a different, prespecified question. Under high-intensity source-to-satellite interference, the 180 s completion ratio decreased from 0.9202 to 0.8949. The paired difference was -0.0253 (95% CI $[-0.0395, -0.0111]$, paired t -test $p = 0.000680$, Wilcoxon $p = 0.000196$; $n = 120$ pairs). The corresponding median completion time increased by 7.61 s, and the P95 increased by 7.52 s. These results support a statistically significant collateral effect under the stated high-load feeder condition. The default sweep remains a condition- and workload-dependent analysis.

The high-size plateau in Figure 8a is explained by the structural change shown in Figure 8e,f. Let $\mathcal{R} = \{\text{origin_full}, \text{cancel_effective}, \text{grace_timeout}\}$ denote the set of settle reasons. For each size group s , the share of settle reason $r \in \mathcal{R}$ is given by Equation (12):

$$P(r | s) = \frac{N_{r,s}}{\sum_{r' \in \mathcal{R}} N_{r',s}}, \quad (12)$$

where $N_{r,s}$ is the number of settled attack samples of size group s with settle reason r . Figure 8e shows that this composition changes substantially with object size. For smaller objects, `origin_full` occupies a larger share, meaning that many attack requests complete with full origin retrieval. As object size increases, `grace_timeout` becomes dominant, while `cancel_effective` remains present but no longer leads.

This change in composition directly reshapes the overall amplification trend. In particular, the mean amplification for size group s can be written as Equation (13):

$$\bar{A}(s) = \sum_{r \in \mathcal{R}} P(r | s) \bar{A}(s, r), \quad (13)$$

where $\bar{A}(s, r)$ is the conditional mean amplification of settled attack samples with size group s and settle reason r . Figure 8f further examines the 10 MB case and shows that the amplification distribution is not unimodal. Instead, it is a mixed distribution shaped by different settle reasons. Specifically, `cancel_effective` is concentrated in a lower amplification range, whereas `grace_timeout` occupies a higher range. This explains why the overall curve enters a plateau at large sizes. The plateau is therefore not a statistical artifact. It is the result of a real change in execution outcomes.

These results reveal the modeled causal chain for the disconnection-induced path. As object size increases, strong amplification becomes more likely. Continued backend execution can delay benign transfers and lower benign completion probability under shared scheduling. The confirmatory feeder experiment shows that completion-rate loss and completion-time increase are statistically significant under high source-to-satellite interference. The range-based path mainly creates request-to-origin asymmetry under cache misses. In contrast, the disconnection-induced path exposes a broader system-level risk. Origin-side work persists after controlled requester loss. It may occupy shared resources after downstream delivery has ceased. For an S-CDN, this is not only wasted retrieval. It is a combined failure mode involving ineffective origin fetching, prolonged feeder contention, and service degradation for ordinary users under high load.

6.4. Large-Object Persistence and Interpretation

The large-object regime is especially important for understanding this path. The corresponding results show that the observed behavior is not a short-lived fluctuation caused by a few outliers. As object size increases, the tail of the amplification distribution becomes more persistent. The overall amplification profile also shifts more clearly. This

pattern is consistent with the mechanism discussed above. Small objects may finish quickly or be terminated early, which limits wasted work. For larger objects, delayed interruption allows more origin retrieval and downlink transmission to continue after requester-side consumption stops.

This interpretation motivates the coverage-loss-inspired attack realization in satellite environments. Moving coverage can produce benign requester disconnections after submission. The present experiment does not validate a natural coverage-loss trigger. Its satellite-specific threat-model implication is therefore one of attribution. An attacker can deliberately disconnect after task triggering in a way that resembles a user leaving coverage. Similar disconnects may also occur during normal mobility. The malicious event may therefore be harder to distinguish from routine service variation. We do not evaluate a detector or quantify this stealth advantage. The experiment shows that the proposed strategy can leave work on the satellite–ground service path. Measuring natural-loss frequency and detection difficulty requires a separately instrumented study.

Taken together, the in-orbit results and supplementary statistics show that back-end execution may outlive controlled requester-side consumption. The matched ground requester-loss and mixed-operation experiments reproduce the same trend. This confirms that the persistence arises from service-chain logic, not from a uniquely satellite mechanism. In the measured S-CDN prototype, the residual work occupies shared satellite resources. The study does not quantify an additional effect caused specifically by natural coverage loss.

For the seed-matched counterfactual, the satellite–ground difference in benign completion ratio was -0.0568 ± 0.0561 at 2 MB, -0.7737 ± 0.0848 at 4 MB, and -0.9524 ± 0.0344 at 5 MB (paired 95% CI half-widths; $n = 50$). The corresponding 1 MB interval included zero. At 6–10 MB, the completion-ratio difference remained below -0.997 , with 95% CIs excluding zero. No benign completion-time or completed-session overhead estimate was available, because no benign transfers completed in the satellite counterfactual. For downlink overhead, the paired differences were -0.1711 ± 0.0037 , -0.0874 ± 0.0018 , -0.0430 ± 0.0011 , and -0.0337 ± 0.0013 at 1, 2, 4, and 5 MB, respectively. All four intervals excluded zero. These significance statements apply only to the controlled counterfactual simulation. They do not apply to the single in-orbit trace.

7. Discussion

7.1. A Unified View of the Two Paths

Although the two attack paths differ in their immediate triggers, they expose the same underlying problem in the studied S-CDN service chain. The range-based path shows that a very small frontend request can induce near-object-level origin retrieval under a cache miss. The disconnection-induced path shows that backend work may continue after requester-side consumption is deliberately stopped. The first path exploits a mismatch between request granularity and origin retrieval granularity. The latter involves a mismatch between requester reachability and backend execution state. In both cases, the system commits backend resources before obtaining sufficient evidence about actual downstream delivery.

This commonality is important because it clarifies that the attack surface is not tied to malformed packets, spoofing, or high-rate flooding. Instead, it is rooted in how the service chain handles accepted requests under cache misses and uncertain delivery conditions. From this perspective, the observed amplification should be understood as a consequence of open-loop request admission and execution asymmetry. The range-based path highlights admission-time asymmetry, whereas the disconnection-induced path further reveals execution persistence after task triggering.

The two paths also differ in their operational impact. The Range path mainly repeats a known request-to-origin gap. A requester can induce substantial backend work with very little frontend traffic. Its satellite impact lies in where those bytes go, not in a larger ratio. The disconnection path has a broader effect. After a controlled requester-side stop, backend work may continue. The resulting traffic can occupy satellite-side transmission resources and degrade benign service under shared scheduling. The persistence itself is related to known connection-decoupling behavior. The attack-realization contribution is the coverage-loss-inspired requester-side stop, not a new persistence mechanism. Natural coverage loss was not instrumented. The present results therefore do not establish it as the experimental trigger or quantify an additional attack capability. The satellite-specific security implication is the plausible attribution ambiguity created by a normal background of mobility-related disconnects. This was not experimentally evaluated.

Taken together, the results suggest that cache misses in S-CDNs should not be viewed only as a performance event. Under weak requester observability and costly satellite-ground transfer, miss handling can also become a security-relevant commitment point. This insight is useful beyond the two specific paths studied here, because it indicates that other protocol-compliant request patterns may trigger similar asymmetries if backend work is admitted too early or interrupted too late.

7.2. External Validity and Limitations

The results should be interpreted with several limitations in mind. First, the in-orbit validation is obtained from one prototype service chain. This improves realism, but the reported magnitudes should not be read as universal constants for all S-CDN implementations. Different systems may vary in cache policy, miss handling logic, requester observability, transport protocol, and backend interruption behavior. In particular, the custom UDP-based short-session service is the actual protocol stack of the measured prototype. It is not a claim about a universal S-CDN standard. The cross-stack discussion in Section 4.2 therefore separates the transport-specific manifestation from the broader open-loop admission condition.

Second, the object set in this study is image-dominant and spans 1–20 MB. This range is sufficient to expose both small-object and large-object regimes, but it does not cover all content types that may appear in future S-CDNs. Different object classes, object lifetimes, and content-delivery policies may change the exact magnitude or shape of the observed amplification. Nevertheless, these limits do not weaken the core finding that protocol-compliant requests can induce disproportionate backend work under cache misses and requester loss.

Third, the in-orbit requester-loss condition was deliberately induced. It was not caused by an observed coverage transition. No separate link-loss signal, calibrated coverage threshold, or synchronized event timeline was recorded. The experiment therefore cannot distinguish natural coverage loss from active network shutdown, process exit, or stopped reads. It also does not quantify additional attack capability or resource consumption caused specifically by passive loss. A future validation should trigger and instrument a real handover or coverage-loss event. It should record request submission, predicted and observed link loss, client-side loss confirmation, server-side detection, and backend cancellation. The controlled supplementary analysis isolates range length, object size, mixed traffic, and settle-reason composition. It does not replace that validation.

Fourth, the present study focuses on two representative paths. It does not claim to exhaust the full attack surface or all protocol stacks. We deliberately selected these paths because they reflect two distinct but fundamental manifestations of the same root cause. The Range path is expected to be more portable across HTTP transport choices.

The disconnection-induced path depends more strongly on how frontend cancellation is propagated into backend execution. We did not experimentally port the service to HTTP/1.1, HTTP/2, or HTTP/3/QUIC. We also did not evaluate other store-and-forward architectures. Other protocol-compliant request patterns may also interact with miss handling, cache policy, or delivery uncertainty in ways that induce backend amplification. Cross-stack experiments remain future work.

Even with these limitations, the results remain meaningful for two reasons. The in-orbit experiments show that the two controlled behaviors can occur in an operationally relevant satellite service chain. The supplementary results show that the observed behavior is systematic rather than anecdotal. They also show that the large-object regime and mixed-traffic effects follow interpretable structural patterns rather than isolated anomalies. Together, they support the main claim of the paper: in S-CDNs, backend work triggered under insufficient delivery evidence can become a measurable and security-relevant source of origin-side and satellite-side cost.

7.3. Lightweight Design Implications

The purpose of this paper is not to propose a complete defense architecture. Even so, the results suggest several lightweight design implications for future S-CDN implementations. The first implication is that miss handling should be more evidence-aware. A cache miss should not automatically imply immediate backend commitment when requester observability is weak and origin-side transfer is expensive. Even lightweight delivery-aware checks can help reduce cases in which accepted requests trigger backend work that is no longer meaningful.

The second implication is that requester reachability should influence backend continuation, not only frontend admission. The disconnection-induced path shows that the main cost may arise after task triggering rather than at the moment of acceptance. For this reason, interruption conditions and continuation policies matter as much as admission conditions. If backend work is allowed to persist without timely reaction to requester loss, the system may continue to consume scarce satellite-side resources long after the requester has ceased to benefit from the transfer.

The third implication concerns cache admission and retrieval granularity. The range-based path shows that a small requester-visible demand can still induce near-object-level retrieval under a miss. This suggests that retrieval policy and cache policy should be considered together. If the system always commits to coarse-grained origin retrieval for fine-grained requests, it enlarges the gap between requester cost and backend cost. Reducing this gap does not necessarily require heavy protocol redesign, but it does require explicit awareness that cache-miss handling is also a security-sensitive decision point.

More broadly, the findings indicate that future S-CDN designs should balance two goals that are often treated separately: service continuity under unstable links and bounded backend commitment under uncertain delivery. Mechanisms that improve continuity are still necessary in satellite systems, but their costs should not be treated as negligible. In environments where origin traffic is expensive and requester reachability is difficult to observe continuously, the system should avoid assuming that every syntactically valid request deserves the same level of backend commitment.

These implications are intentionally lightweight and directional. The Range path is a known CDN problem observed in orbit. The second path contributes a coverage-loss-inspired requester-side stopping strategy and demonstrates persistence on the measured satellite service path. Natural coverage loss was not instrumented. Neither path is claimed to have a larger byte ratio than the matched ground deployment.

8. Conclusions

This paper studies protocol-compliant back-to-origin amplification in S-CDNs. Matched ground experiments show that the Range/MISS asymmetry and post-stop persistence arise from CDN service logic. The satellite prototype shows that the resulting work occupies a constrained shared service path. The Range path is a known CDN mechanism. The disconnection-induced contribution is the evaluated coverage-loss-inspired requester-side stopping strategy. Natural coverage loss was not instrumented. The attribution ambiguity from benign mobility-related disconnects therefore remains a bounded implication and future-work question.

Author Contributions: Conceptualization, B.J. and X.Y.; methodology, B.J., Y.Z. and X.Y.; software, B.J. and A.C.; validation, B.J., A.C. and Y.Z.; formal analysis, B.J. and X.Y.; investigation, B.J., A.C. and Z.Z.; data curation, B.J. and Y.S.; writing—original draft preparation, B.J.; writing—review and editing, X.Y. and Y.Z.; visualization, B.J. and Y.S.; supervision, X.Y. and Y.Z.; project administration, Y.Z.; funding acquisition, Y.Z. and X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Fundamental Research Funds for the Central Universities, under Grant Nos. 510224057 and 530424001, and by the National Natural Science Foundation of China (NSFC), under Grant Nos. 62425105 and 62206019.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the findings of this study are not publicly available because they were collected from an in-orbit prototype service chain and contain deployment-specific measurements. Further details may be provided by the corresponding author upon reasonable request, subject to applicable access restrictions.

Acknowledgments: The authors thank the technical staff who supported the in-orbit prototype and ground supplementary experiments.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Azari, M.M.; Solanki, S.; Chatzinotas, S.; Kodheli, O.; Sallouha, H.; Colpaert, A.; Mendoza Montoya, J.F.; Pollin, S.; Haqiqatnejad, A.; Mostaani, A.; et al. Evolution of Non-Terrestrial Networks from 5G to 6G: A Survey. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 2633–2672. [\[CrossRef\]](#)
2. Cui, H.; Zhang, J.; Geng, Y.; Xiao, Z.; Sun, T.; Zhang, N.; Liu, J.; Wu, Q.; Cao, X. Space-Air-Ground Integrated Network (SAGIN) for 6G: Requirements, Architecture and Challenges. *China Commun.* **2022**, *19*, 90–108. [\[CrossRef\]](#)
3. Dilley, J.; Maggs, B.M.; Parikh, J.; Prokop, H.; Sitaraman, R.K.; Wehl, W.E. Globally Distributed Content Delivery. *IEEE Internet Comput.* **2002**, *6*, 50–58. [\[CrossRef\]](#)
4. Adler, M.; Sitaraman, R.K.; Venkataramani, H. Algorithms for optimizing the bandwidth cost of content delivery. *Comput. Netw.* **2011**, *55*, 4007–4020. [\[CrossRef\]](#)
5. Nygren, E.; Sitaraman, R.K.; Sun, J. The Akamai Network: A Platform for High-Performance Internet Applications. *ACM SIGOPS Oper. Syst. Rev.* **2010**, *44*, 2–19. [\[CrossRef\]](#)
6. Thomdapu, S.T.; Katiyar, P.; Rajawat, K. Dynamic cache management in content delivery networks. *Comput. Netw.* **2021**, *187*, 107822. [\[CrossRef\]](#)
7. Bose, R.; Fadaei, S.; Mohan, N.; Kassem, M.; Sastry, N.; Ott, J. It's a Bird? It's a Plane? It's CDN!: Investigating Content Delivery Networks in the LEO Satellite Networks Era. In Proceedings of the 23rd ACM Workshop on Hot Topics in Networks, Irvine, CA, USA, 18–19 November 2024; pp. 1–9. [\[CrossRef\]](#)
8. Bose, R.; Mohan, N.; Ott, J. Poster: Twinkle, Twinkle, Streaming Star: Illuminating CDN Performance over Starlink. In Proceedings of the 2024 ACM on Internet Measurement Conference, Madrid, Spain, 4–6 November 2024; pp. 759–760. [\[CrossRef\]](#)
9. Mohan, N.; Ferguson, A.E.; Cech, H.; Bose, R.; Renatin, P.R.; Marina, M.K.; Ott, J. A Multifaceted Look at Starlink Performance. In Proceedings of the ACM Web Conference 2024, Singapore, 13–17 May 2024; pp. 2723–2734. [\[CrossRef\]](#)

10. Zhu, X.; Jiang, C.; Kuang, L.; Zhao, Z. Cooperative Multilayer Edge Caching in Integrated Satellite-Terrestrial Networks. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 2924–2937. [\[CrossRef\]](#)
11. Zhao, R.; Ran, Y.; Luo, J.; Chen, S. Towards Coverage-Aware Cooperative Video Caching in LEO Satellite Networks. In Proceedings of the GLOBECOM 2022—2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022; pp. 1893–1898. [\[CrossRef\]](#)
12. Vu, T.X.; Poirier, Y.; Chatzinotas, S.; Maturo, N.; Grotz, J.; Roelens, F. Modeling and Implementation of 5G Edge Caching over Satellite. *Int. J. Satell. Commun. Netw.* **2020**, *38*, 395–406. [\[CrossRef\]](#)
13. Luglio, M.; Romano, S.P.; Roseti, C.; Zampognaro, F. Satellite multi-beam multicast support for an efficient community-based CDN. *Comput. Netw.* **2022**, *217*, 109352. [\[CrossRef\]](#)
14. Zhang, L.; Guo, Z.; Su, H.; Zhao, W. CAAST: Optimizing data communications in satellite networks through Cache and Anycast. *Comput. Netw.* **2025**, *259*, 111066. [\[CrossRef\]](#)
15. Ji, Z.; Wu, S.; Jiang, C.; Wang, W. Popularity-Driven Content Placement and Multi-Hop Delivery for Terrestrial-Satellite Networks. *IEEE Commun. Lett.* **2020**, *24*, 2574–2578. [\[CrossRef\]](#)
16. Liu, S.; Hu, X.; Wang, Y.; Cui, G.; Wang, W. Distributed Caching Based on Matching Game in LEO Satellite Constellation Networks. *IEEE Commun. Lett.* **2018**, *22*, 300–303. [\[CrossRef\]](#)
17. Tang, J.; Li, J.; Chen, X.; Xue, K.; Zhang, L.; Sun, Q.; Lu, J. Cooperative Caching in Satellite-Terrestrial Integrated Networks: A Region Features Aware Approach. *IEEE Trans. Veh. Technol.* **2024**, *73*, 10602–10616. [\[CrossRef\]](#)
18. Abreha, H.G.; Maity, I.; Chougrani, H.; Politis, C.; Chatzinotas, S. Resource-Aware On-Board Content Caching in Multi-Layer Satellite Edge Networks. In Proceedings of the ICC 2024—IEEE International Conference on Communications, Denver, CO, USA, 9–13 June 2024; pp. 3943–3949. [\[CrossRef\]](#)
19. Chen, J.; Jiang, J.; Duan, H.; Weaver, N.; Wan, T.; Paxson, V. Host of Troubles: Multiple Host Ambiguities in HTTP Implementations. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 1516–1527. [\[CrossRef\]](#)
20. Chen, J.; Zheng, X.; Duan, H.; Liang, J.; Jiang, J.; Li, K.; Wan, T.; Paxson, V. Forwarding-Loop Attacks in Content Delivery Networks. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 21–24 February 2016.
21. Guo, R.; Chen, J.; Liu, B.; Zhang, J.; Zhang, C.; Duan, H.; Wan, T.; Jiang, J.; Hao, S.; Jia, Y. Abusing CDNs for Fun and Profit: Security Issues in CDNs' Origin Validation. In Proceedings of the 2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS), Salvador, Brazil, 2–5 October 2018; pp. 1–10. [\[CrossRef\]](#)
22. Li, W.; Shen, K.; Guo, R.; Liu, B.; Zhang, J.; Duan, H.; Hao, S.; Chen, X.; Wang, Y. CDN backfired: Amplification attacks based on http range requests. In Proceedings of the 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Valencia, Spain, 29 June–2 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 14–25.
23. Lin, Z.; Lin, Z.; Liu, X.; Chen, J.; Guo, R.; Chen, C.; Xiao, S. CDN Cannon: Exploiting CDN Back-to-Origin Strategies for Amplification Attacks. In Proceedings of the 33rd USENIX Security Symposium (USENIX Security 24), Philadelphia, PA, USA, 14–16 August 2024; pp. 5717–5734.
24. Vu, T.X.; Maturo, N.; Vuppala, S.; Chatzinotas, S.; Grotz, J.; Alagha, N. Efficient 5G Edge Caching over Satellite. In Proceedings of the 36th International Communications Satellite Systems Conference (ICSSC 2018), Niagara Falls, ON, Canada, 15–18 October 2018; pp. 1–5. [\[CrossRef\]](#)
25. Ngo, Q.T.; Phan, K.T.; Xiang, W.; Mahmood, A.; Slay, J. Two-Tier Cache-Aided Full-Duplex Hybrid Satellite-Terrestrial Communication Networks. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 1753–1765. [\[CrossRef\]](#)
26. Vu, T.X.; Maturo, N.; Chatzinotas, S.; Grotz, J.; Christophory, T.; Ottersten, B. Dynamic Bandwidth Allocation and Edge Caching Optimization for Nonlinear Content Delivery through Flexible Multibeam Satellites. In Proceedings of the ICC 2022—IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2022; pp. 1143–1148. [\[CrossRef\]](#)
27. Tang, J.; Li, J.; Zhang, L.; Xue, K.; Sun, Q.; Lu, J. Content-Aware Routing Based on Cached Content Prediction in Satellite Networks. In Proceedings of the GLOBECOM 2022—2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022; pp. 6541–6546. [\[CrossRef\]](#)
28. Fielding, R.; Nottingham, M.; Reschke, J. HTTP Semantics. RFC 9110, RFC Editor, June 2022. Available online: <https://www.rfc-editor.org/rfc/rfc9110> (accessed on 12 October 2026).
29. Thomson, M.; Benfield, C. HTTP/2. RFC 9113, RFC Editor, June 2022. Available online: <https://www.rfc-editor.org/rfc/rfc9113> (accessed on 12 October 2026).
30. Iyengar, J.; Thomson, M. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000, RFC Editor, May 2021. Available online: <https://www.rfc-editor.org/rfc/rfc9000> (accessed on 12 October 2026).
31. Bishop, M. HTTP/3. RFC 9114, RFC Editor, June 2022. Available online: <https://www.rfc-editor.org/rfc/rfc9114> (accessed on 12 October 2026).

32. Fielding, R.T.; Nottingham, M.; Reschke, J. HTTP/1.1. RFC 9112, RFC Editor, June 2022. Available online: <https://www.rfc-editor.org/rfc/rfc9112> (accessed on 12 October 2026).
33. Jansen, J.; Subramanyam, S.; Bouqueau, R.; Cernigliaro, G.; Cabré, M.M.; Pérez, F.; Cesar, P. A Pipeline for Multiparty Volumetric Video Conferencing: Transmission of Point Clouds over Low Latency DASH. In Proceedings of the 11th ACM Multimedia Systems Conference, Istanbul, Turkey, 8–11 June 2020; pp. 341–344. [[CrossRef](#)]
34. Jabiyevev, B.; Sprecher, S.; Onarlioglu, K.; Kirda, E. T-Reqs: HTTP Request Smuggling with Differential Fuzzing. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual, 15–19 November 2021; pp. 1805–1820. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.