

Routing Centralization and Censorship in IPFS

Jan Matter (ETH Zurich), **Muoi Tran** (Chalmers/GU)

SSS '26 | 9 October 2026

In 2017, Spain censored IPFS at its gateway

1. Spain blocked referendum websites ahead of the vote
2. The Catalan government mirrored the content on IPFS's P2P network
3. The censor then **blocked IPFS's central HTTP gateway** (ipfs.io)



Photo: Generalitat de Catalunya, Attribution, <https://commons.wikimedia.org/w/index.php?curid=62597004>

InterPlanetary File System (IPFS) is a widely adopted peer-to-peer file system

Servers

~25,000

DHT server nodes, 2026¹

Store content and serve it to
other peers

Clients

~400,000

DHT client nodes, 2026¹

Fetch content but do not
serve others

Applications

2

browsers: **Brave, Opera**

Apps: **Audius, Matters
News, Filebase, ...**

¹ ProbeLab, IPFS network report, Sept 2026.

IPFS aims to be decentralized, but it is centralized in practice

Decentralized in principle

- **Self-certifying content addressing (CID):** anyone can verify content's integrity
- **Kademlia DHT:** peers find content providers without any central server
- **Replication:** many providers, so in principle no single party can block a CID

Centralized in practice

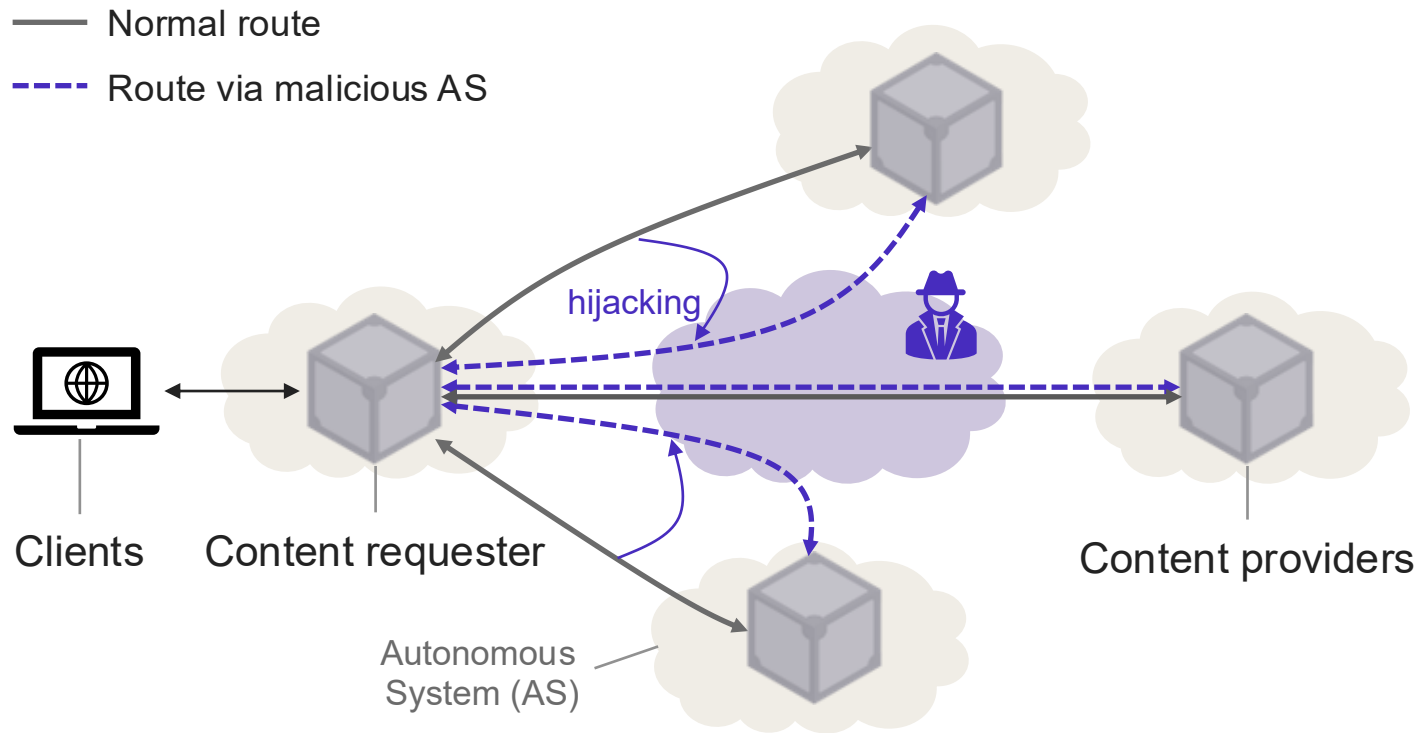
3 in 4

nodes are hosted by a few cloud providers¹

Indexers and HTTP gateways: centralized components for speed and convenience implemented by IPFS maintainers

¹ Shi et al., Centralization in the Decentralized Web, ACM WWW 2025.

Internet routing can be another dimension of centralization



Internet routing

~80K autonomous systems (ASes)¹
route any traffic on the Internet via
Border Gateway Protocol (**BGP**)

- **Interception:** already on-path ASes
- **Hijacking:** false routes draw traffic
- **Known availability attacks:**
in Bitcoin, Ethereum, Tor,...

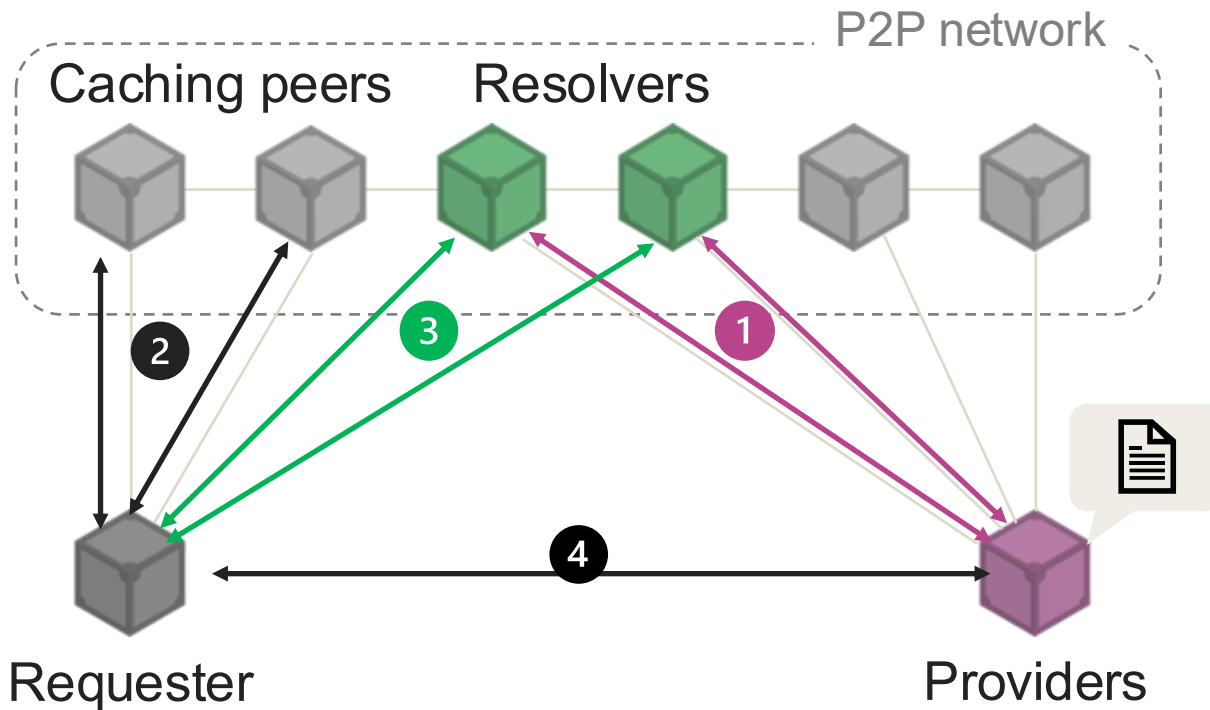
Can an AS really censor IPFS?

¹ CAIDA AS Rank (Oct 2026)

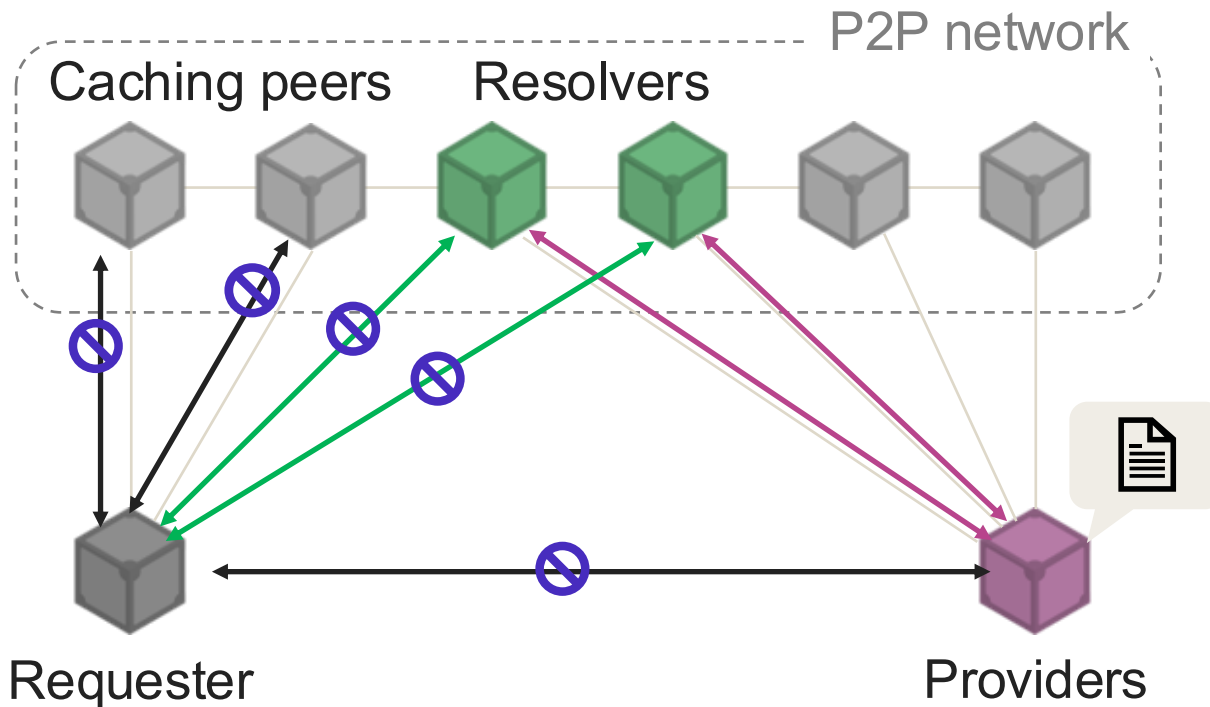
Background: Content publishing and retrieval in IPFS

Retrieval steps

- 1 Provider adds a record to 20 resolvers
- 2 Requester asks caching peers (Bitswap)
- 3 Else, fetches the record from resolvers
- 4 Requester downloads from a provider



Content censorship requires blocking caching peers and providers or resolvers



Attack vectors

Provider vector

Caching peers

+

Providers

Resolver vector

Caching peers

+

Resolvers

Both together: every retrieval path is blocked for maximal censorship

We measured the live IPFS network for popular CIDs

3 datasets of 1,000 popular CIDs

From Bitswap monitors,
one week apart, July 2025

- **Caching peers**

Our unlimited-connection supernode asks every peer via Bitswap

- **Resolvers**

30 closest DHT peers (20 + margin)
and the IPNI indexer (cid.contact)

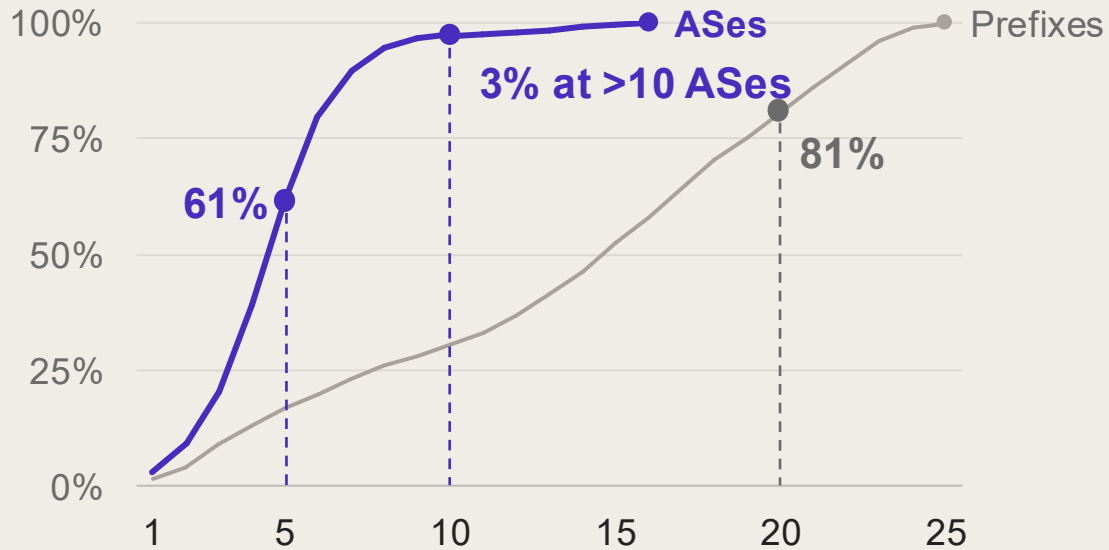
- **Providers**

A standard CID lookup returns
their peer IDs and addresses

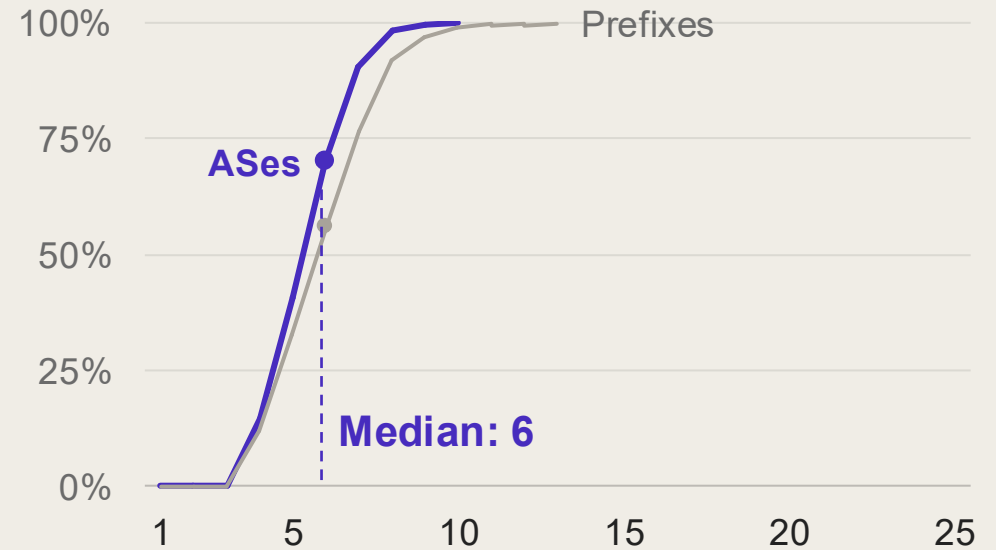
**IP prefix
and AS**
of every address

Most popular CIDs sit in a few ASes and prefixes

Providers and caching peers



Resolvers and caching peers



Both charts x: distinct ASes or prefixes per CID y: cumulative % of CIDs

Hosting is concentrated

As prior work found, most CIDs sit in a handful of ASes

Only a thin tail is widely spread

Very few CIDs spread beyond 10 provider ASes

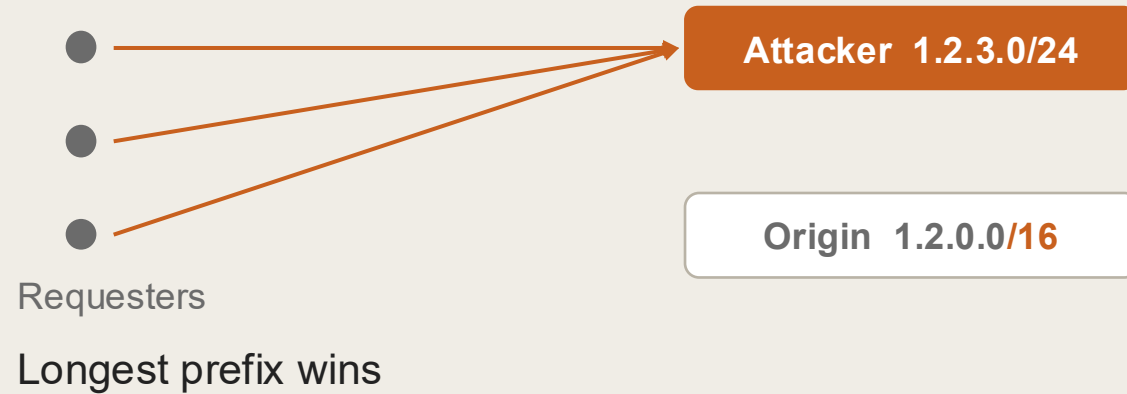
Resolvers share few prefixes

Unlike providers, their prefixes are as few as their ASes

A target's prefix decides which AS can hijack it

Wins from anywhere

position-independent



Unprotected

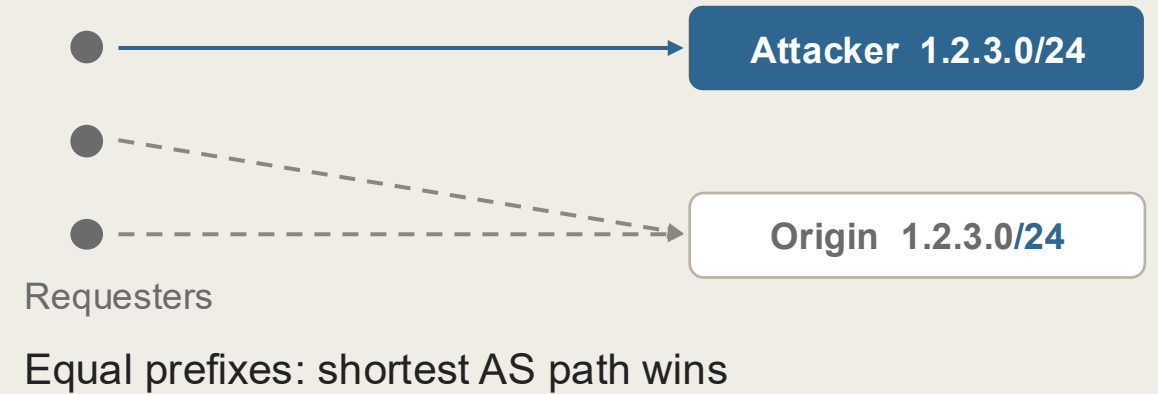
- 1 **Origin shorter than /24**
Attacker announces a /24 inside it

RPKI-protected

- 3 **Below ROA max-length**
A more-specific route stays RPKI-valid

Wins only nearby

position-dependent



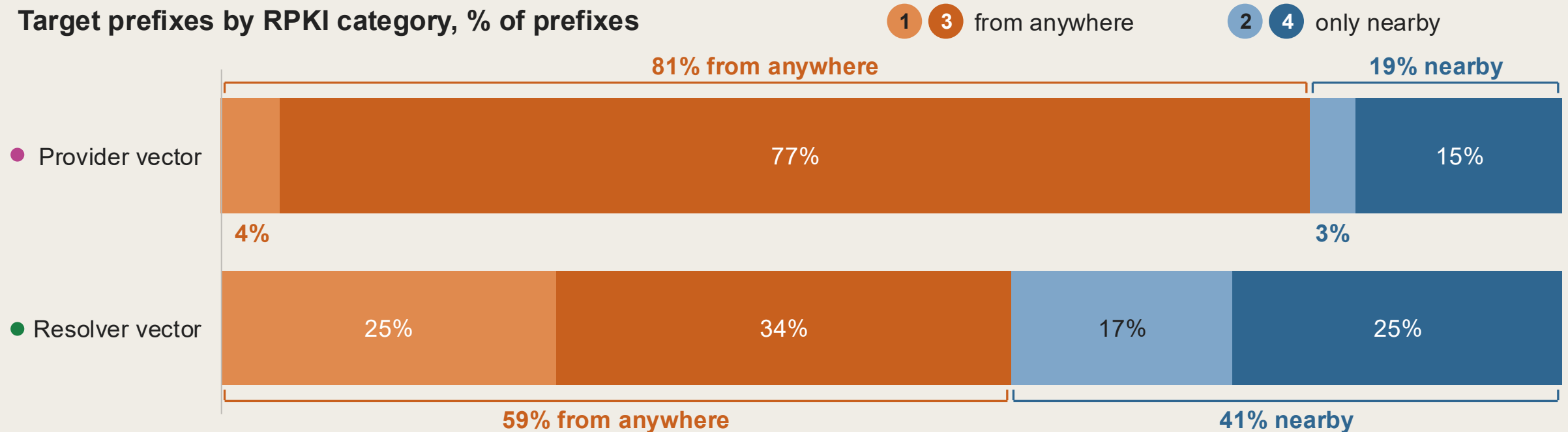
- 2 **Origin exactly /24**
Longer routes are filtered by operators

- 4 **At ROA max-length**
No valid more-specific route exists

RPKI: signed records (ROAs) naming which AS may announce a prefix, and up to which length. IPv4 shown; /48 for IPv6.

Most targets can be hijacked from anywhere

Target prefixes by RPKI category, % of prefixes



Providers: RPKI rarely helps

Most have a ROA, but its max-length still allows a more-specific route

Resolvers fare better

but most of their prefixes can still be hijacked from anywhere

5% of CIDs: fully exposed

all provider-vector prefixes are hijackable from anywhere

Three questions about the censorship attacks

Feasibility

Can one AS block a CID for every requester?

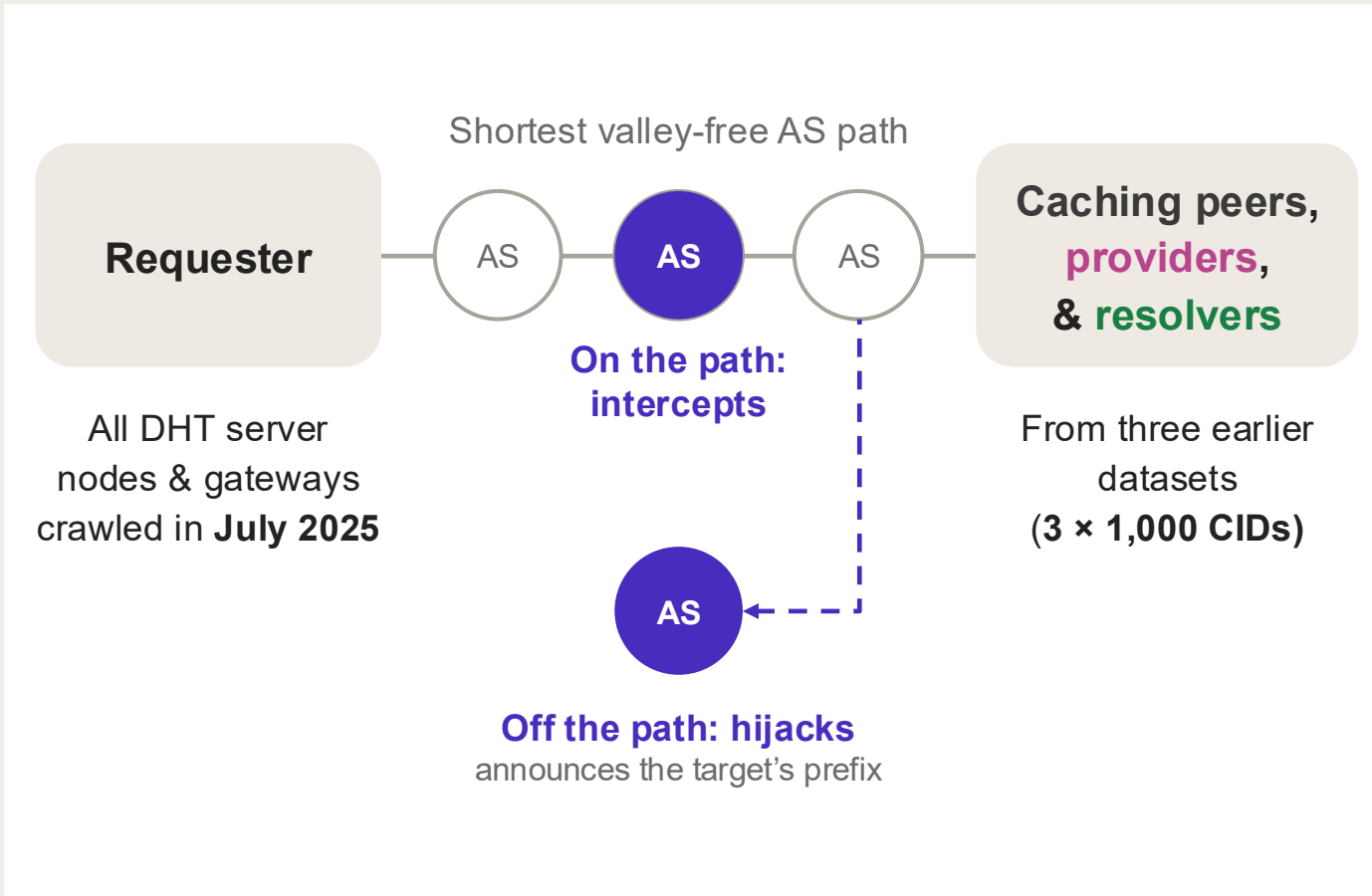
Effectiveness

If not, how many requesters can it block?

Cost

How many prefixes must it hijack to block a CID?

We simulated each top-100 AS as a censor



One request

A requester fetches a CID from its peers or provider

Standard routing model

Valley-free paths; ties go to the lower AS number

Passive interception

An on-path attacker drops traffic in either direction

BGP hijacking

From all ASes: success depends on prefix categories

Measure for each AS

The **share of requesters blocked** per CID

Feasibility:

Complete censorship works only for a few CIDs

0.7–7.8%

of CIDs can be blocked for all requesters

Range across three datasets of 1,000 CIDs each

What these CIDs share

Every provider prefix is hijackable from anywhere

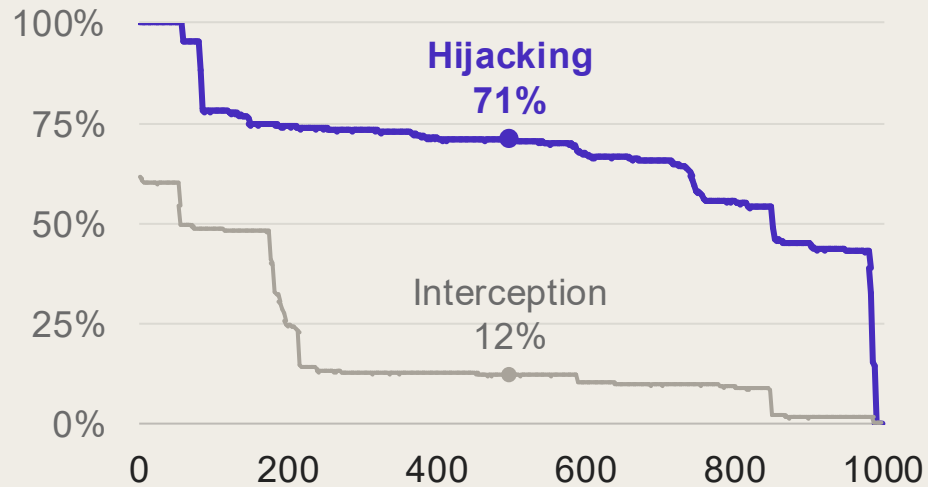
- 1 **Unprotected, origin shorter than /24**
Attacker announces a /24 inside it
- 3 **RPKI-protected, below ROA max-length**
A more-specific route stays RPKI-valid

None had a provider in category 2 or 4, which only nearby ASes can hijack

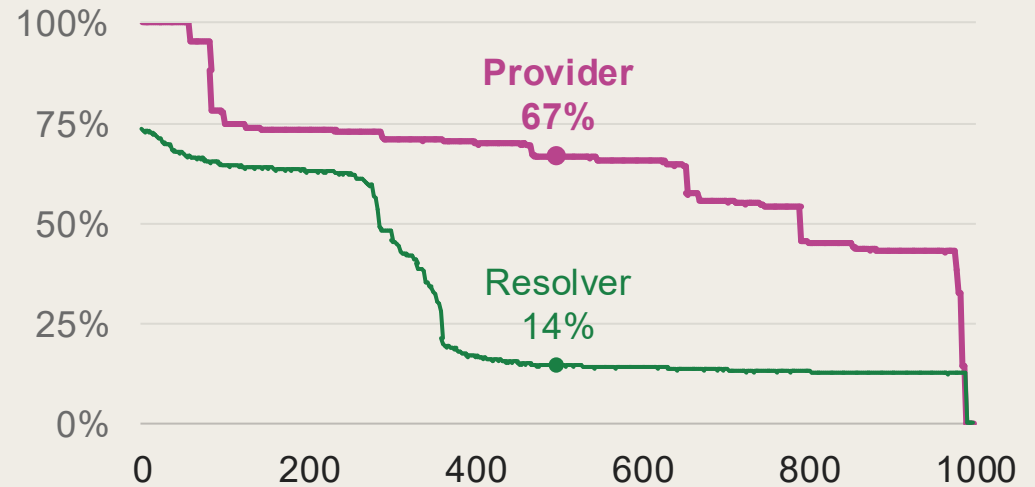
Effectiveness:

Hijacking lets one AS block most requesters

By routing method



By attack vector



Both charts x: CIDs, ranked by share blocked y: requesters blocked by the strongest AS, dataset 1

Hijacking does the work

Interception blocks few, and adds little to hijacking

The provider vector is the weak point

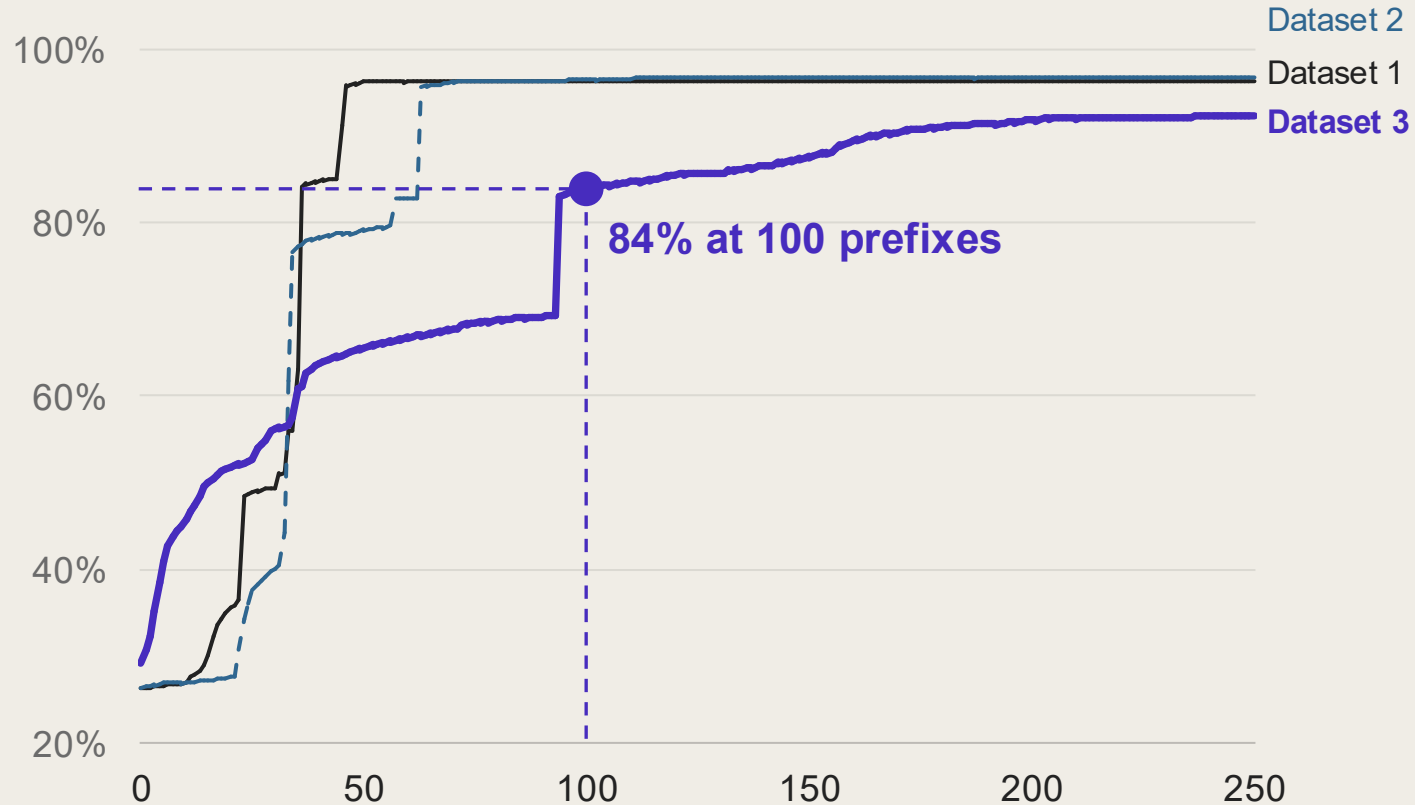
Attacking providers is enough; resolvers are harder to hijack

The attacker position is also crucial: AS174 (a Tier-1 AS) blocks over twice as much as most other ASes

Cost:

Hijacking about 100 prefixes is enough

Requesters blocked, as share of the blockage with all prefixes



x-axis: hijacked IP prefixes, added greedily by largest gain

100

of 2,791 target prefixes suffice

$\geq 84\%$

of the maximum blockage in all datasets

Diminishing returns

Another 150 prefixes add at most 8 points

Each weakness the attack exploits has a fix

What the attack exploits

Hijackable prefixes

Most provider prefixes sit in **categories 1 and 3**, so any AS on the Internet can hijack them



Countermeasure

Harden one provider

1

Serve at least one copy from an **RPKI-protected, max-length** prefix, where hijacks rarely succeed

Concentrated providers

A CID lives in **a few ASes and prefixes**, so one well-placed AS sits on most paths to it



Replicate widely

2

Spread copies over **many ASes and prefixes**, so no single AS sits on the path to all of them

We test both in practice: 1 owner-run **public IPFS clusters** 2 protocol-level **network-wide pinning**

One RPKI-hardened provider defeats hijacking

Public IPFS cluster	ASes	Prefixes (IPv4 / IPv6)	Requesters blocked
Project Gutenberg (Spanish)	15	23 / 7	0
IPFS Websites	2	5 / 4	0
Wikipedia	1	4 / 3	0

One shared, protected prefix

All three share **147.75.63.0/24**

Its ROA makes more-specific hijacks **RPKI-invalid**

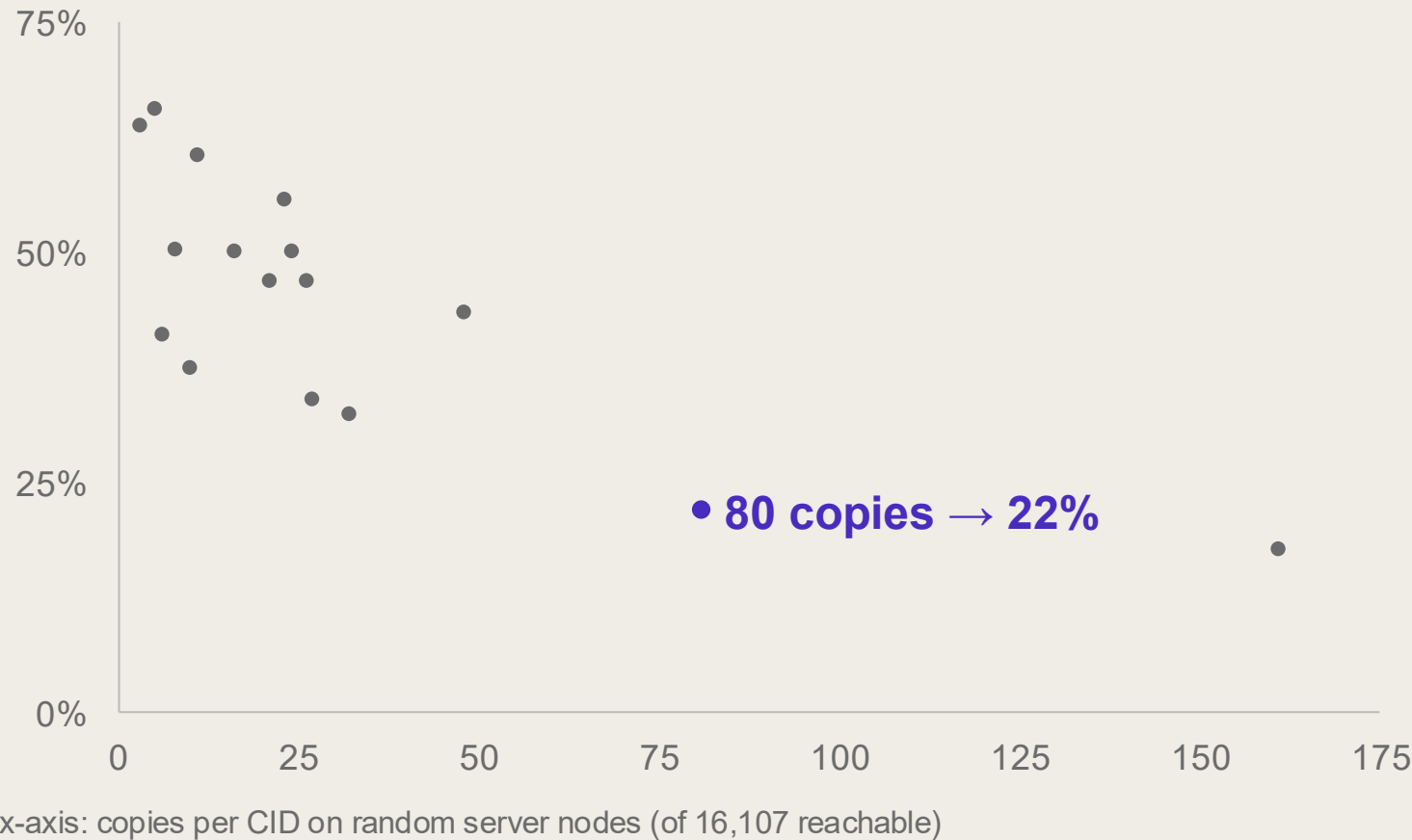
One AS is enough

Wikipedia sits in only **1 AS**, yet attacks still block **0**

Its **RPKI-valid prefix** protects it, not its AS count

80 random copies reduce censorship to 22%

Requesters intercepted by the strongest AS, per CID



Our experiment

Replicate each CID on more **random DHT server nodes**, then rerun the attack

0.5%

of server nodes cuts censorship to 22%

Plus hardened providers

A few RPKI-protected providers further push blocking close to zero

Several follow-up directions remain open

Limits of our model



Equal weighting

Every requester and CID counts the same



Static snapshot

No re-provisioning, so results are upper bounds



Server nodes only

DHT clients are missing from the requester set



One AS at a time

Coalitions of ASes could block even more

Open problems



Domestic censors

Local censors force users to tunnel out

Does IPFS still work over tunnels, and can censors fingerprint its traffic?



Content discovery

Predictable resolvers and a central indexer

How do we make content discovery as hard to censor as data delivery?

Takeaway: One AS can censor IPFS effectively, but hardened providers mitigate it

Why it works: IPFS hosting concentrates on a few ASes and largely unprotected prefixes

Effective

>75%

of tested content is blocked by one AS from most requesters

Cheap

≈100

hijacked IP prefixes achieve over 80% of the blockage with all prefixes

Mitigable

≈0%

of requesters blocked once replicas include hardened providers



CHALMERS
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF
GOTHENBURG

Routing Centralization and Censorship in IPFS

Jan Matter (ETH Zurich), **Muoi Tran** (Chalmers/GU)

SSS '26 | 9 October 2026