The Role of IPFS and P2P in Decentralized Infrastructures and Applications

João Leitão

NOVA LINCS & NOVA University of Lisbon

jc.leitao@fct.unl.pt @jcaleitao

https://mastodon.joaoleitao.org/@jcaleitao

https://asc.di.fct.unl.pt/~jleitao

NOVALINCS

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

TaRDIS

Disclaimer

I will be discussing the design and impact of both the Interplanetary File System (IPFS) and Libp2p which were developed by Protocol Labs.

While I have collaborated multiple times with Protocol Labs (mostly through the ProbeLab laboratory) I am not affiliated with them.

As usual, opinions shown here are my own.

Roadmap of this Talk

- Overview of IPFS (and Libp2p)
- What is it that we know about IPFS on the wild
- What are pressure points in IPFS and Libp2p
- Beyond IPFS and Libp2p on Decentralized Systems
- Takeaway Points

Roadmap of this Talk

- **Overview of IPFS (and Libp2p)**
- What is it that we know about IPFS on the wild
- What are pressure points in IPFS and Libp2p
- Beyond IPFS and Libp2p on Decentralized Systems
- Takeaway Points

IPFS

The Inter-Planetary File System is a file distribution system that operates at global scale featuring a fully decentralized architecture.

It is, in some sense, the (blob) storage of the Web3 and its design aligns with some of the key ideas of this movement:

- Independence of big operators
- Resistance to censorship

IPFS: Key Design Principles

The API supported by IPFS is like the one that you could expect from a standard Distributed Hash Table, with some details:

IPFS: Key Design Principles

The API supported by IPFS is like the one that you could expect from a standard Distributed Hash Table, with some details:

- Files are not stored by IPFS nodes normally, instead IPFS records “Content Record Providers” that state which nodes have content available.

IPFS: Key Design Principles

The API supported by IPFS is like the one that you could expect from a standard Distributed Hash Table, with some details:

- Files are not stored by IPFS nodes normally, instead IPFS records “Content Record Providers” that state which nodes have content available.
- Identifiers (CID) used to locate content are based on the content itself, which allows for quick verification of integrity upon retrieval.

IPFS: Key Design Principles

The API supported by IPFS is like the one that you could expect from a standard Distributed Hash Table, with some details:

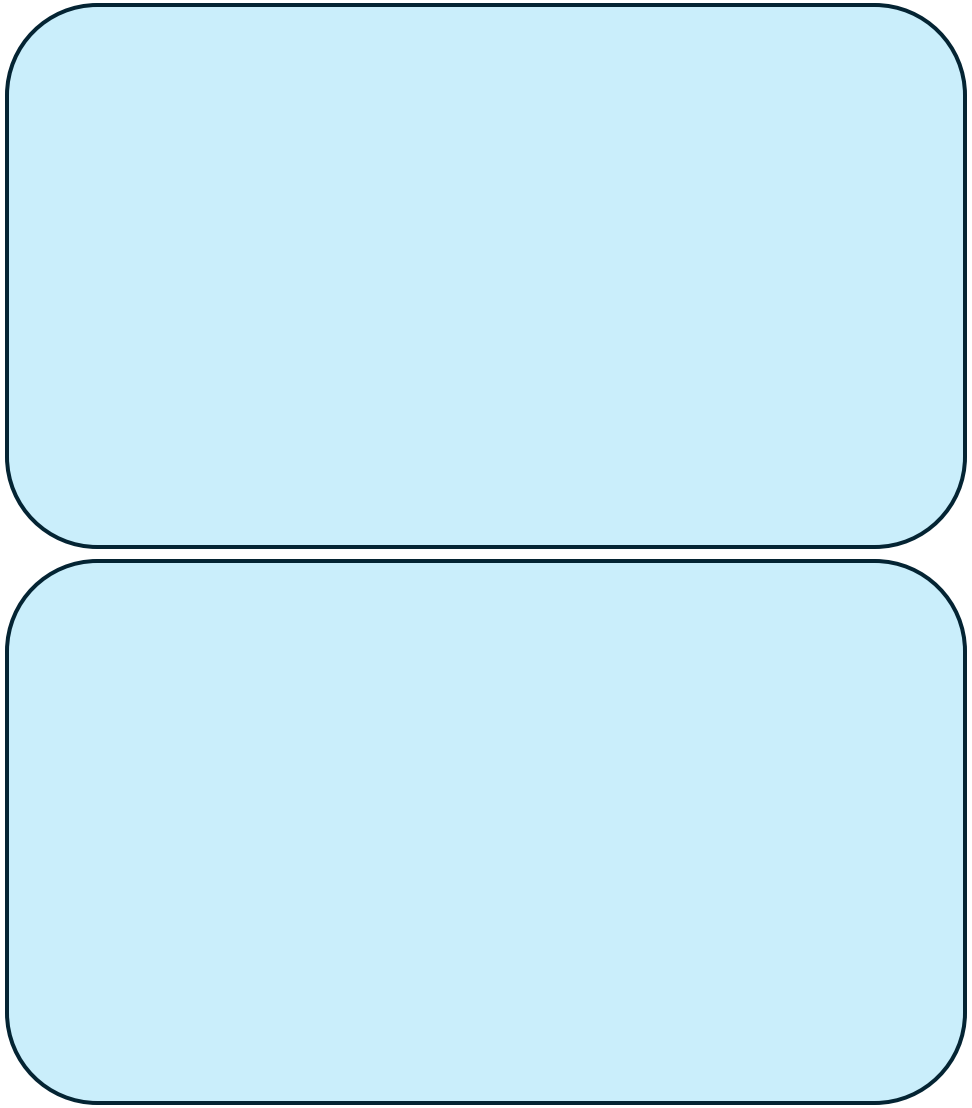
- Files are not stored by IPFS nodes normally, instead IPFS records “Content Record Providers” that state which nodes have content available.
- Identifiers (CID) used to locate content are based on the content itself, which allows for quick verification of integrity upon retrieval.
- Files are divided in blocks (each with a CID) whose relationship is recorded on a Directed Acyclic Graph (DAG) that also has a CID.

IPFS: Key Design Principles

The API supported by IPFS is like the one that you could expect from a standard Distributed Hash Table, with some details:

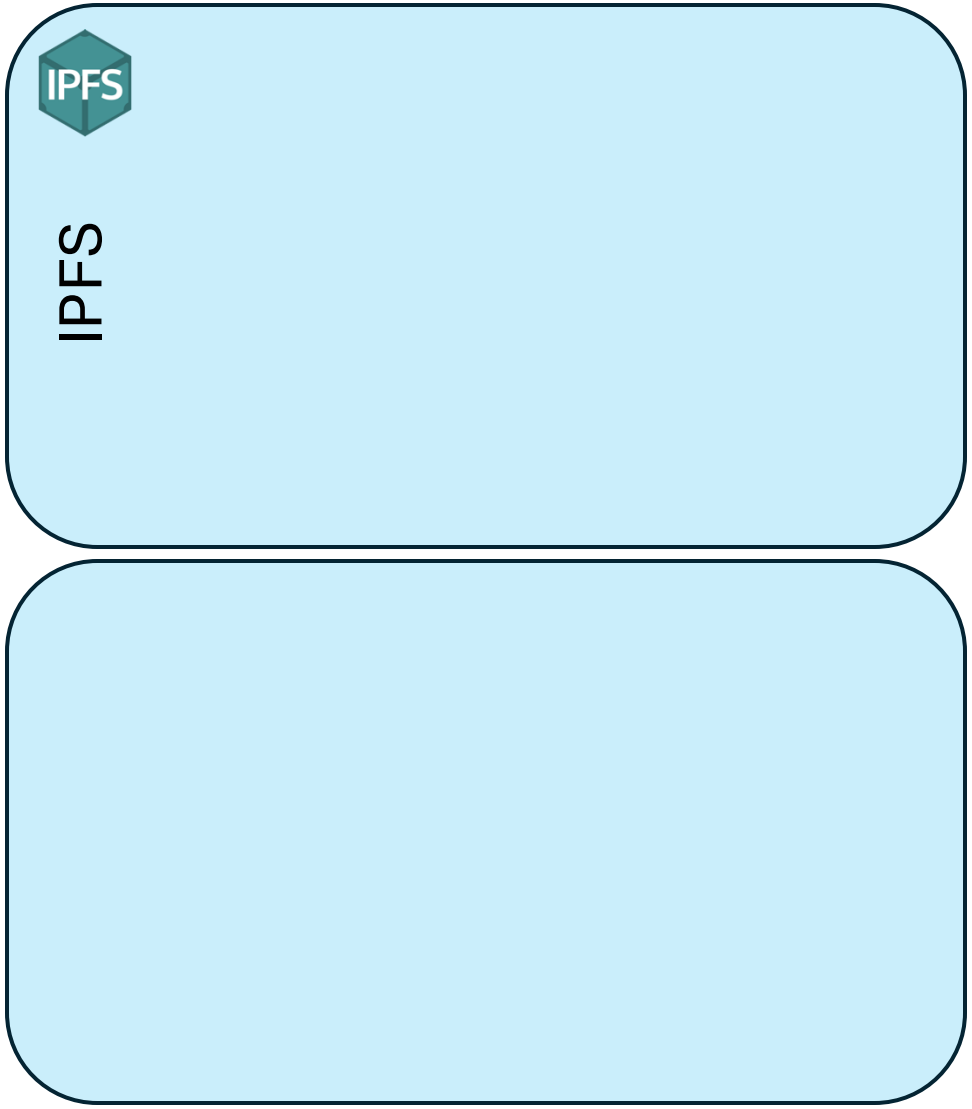
- Files are not stored by IPFS nodes normally, instead IPFS records “Content Record Providers” that state which nodes have content available.
- Identifiers (CID) used to locate content are based on the content itself, which allows for quick verification of integrity upon retrieval.
- Files are divided in blocks (each with a CID) whose relationship is recorded on a Directed Acyclic Graph (DAG) that also has a CID.
- Records are soft-state and need to be periodically refreshed to deal with churn.

IPFS: Architecture

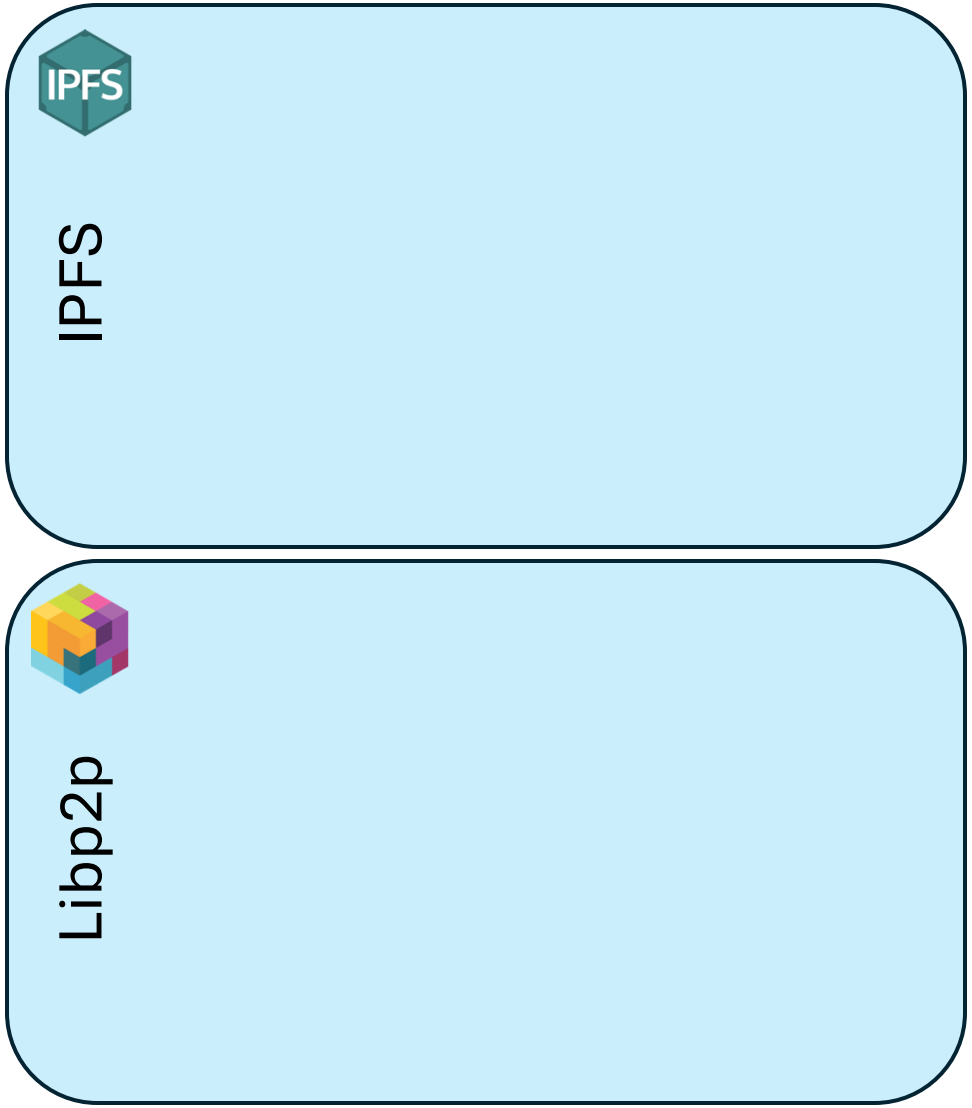


- The IPFS architecture is quite complex (but well documented).
- For the benefit of time, lets simplify this explanation abstracting some complexity and focusing on the main components.

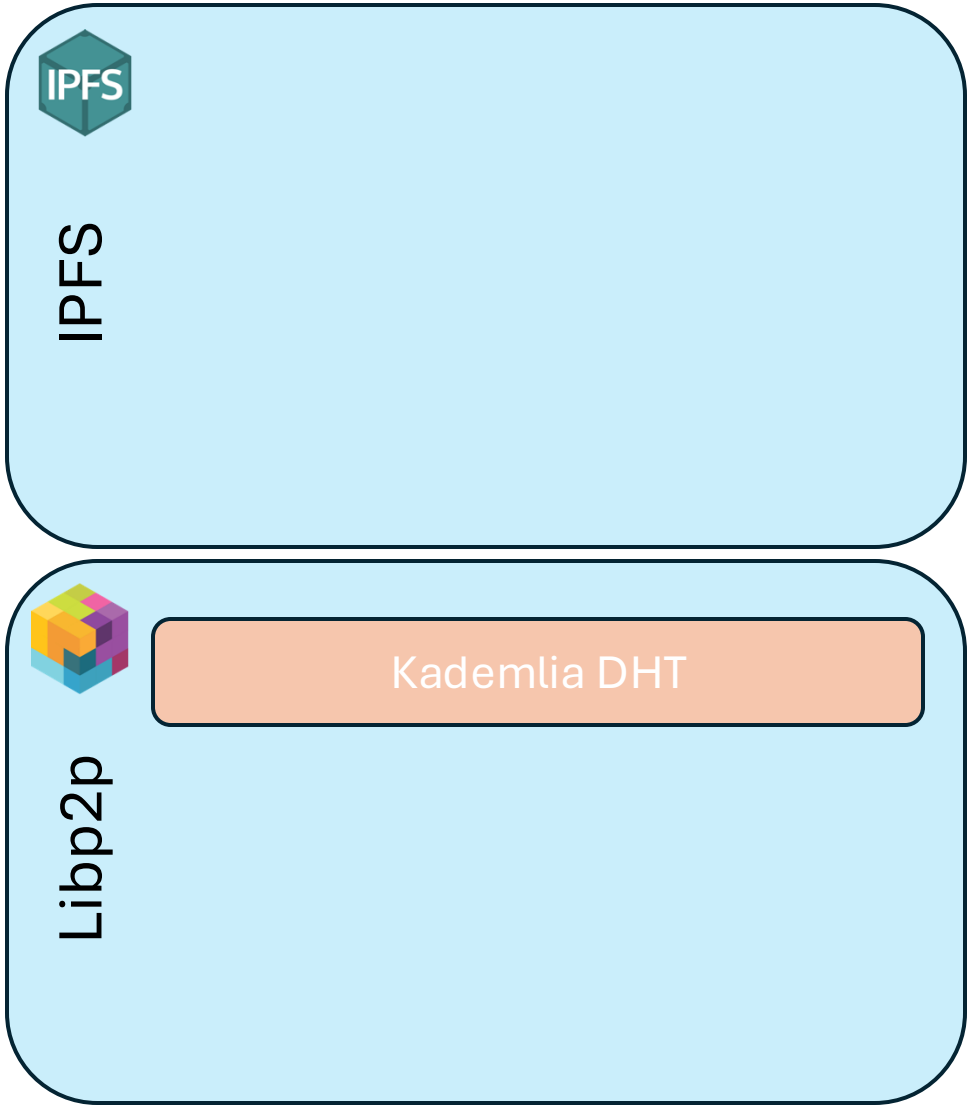
IPFS: Architecture



IPFS: Architecture

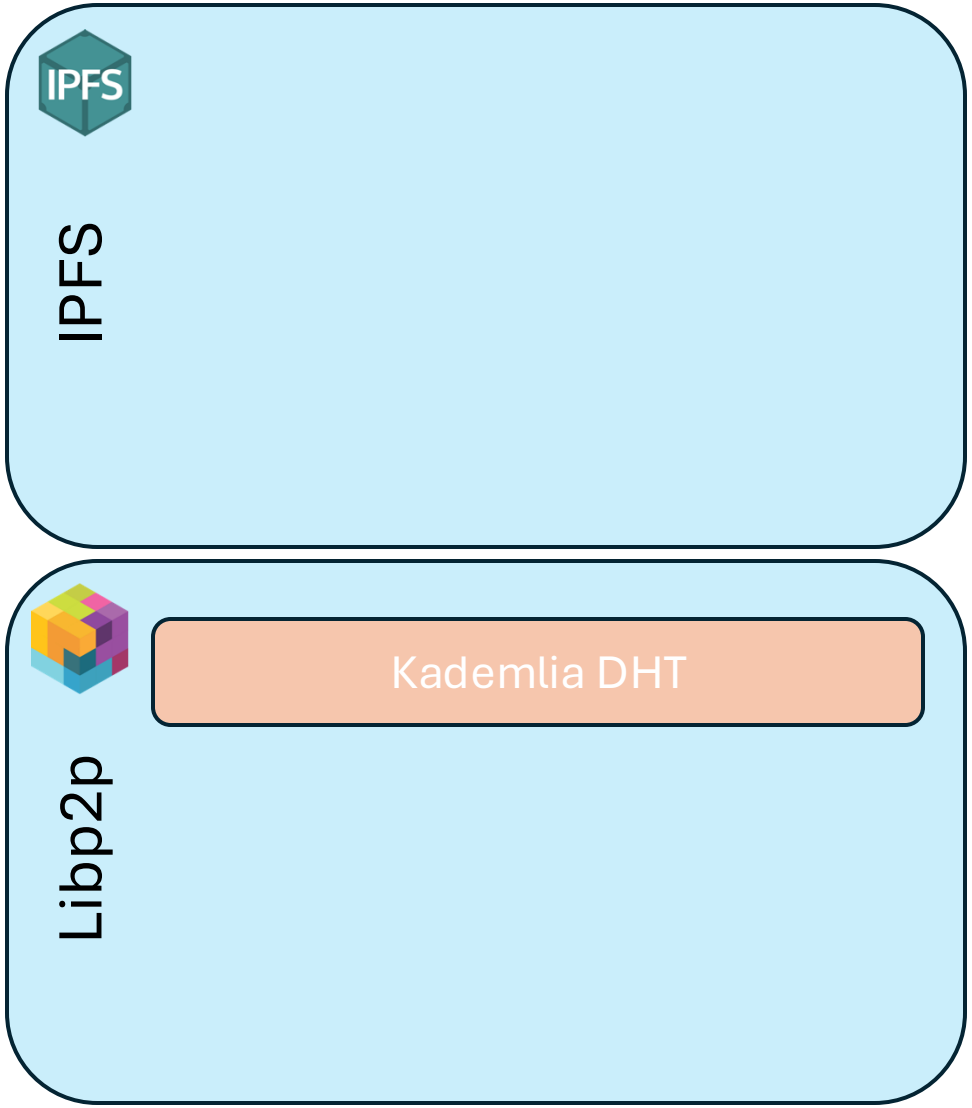


IPFS: Architecture



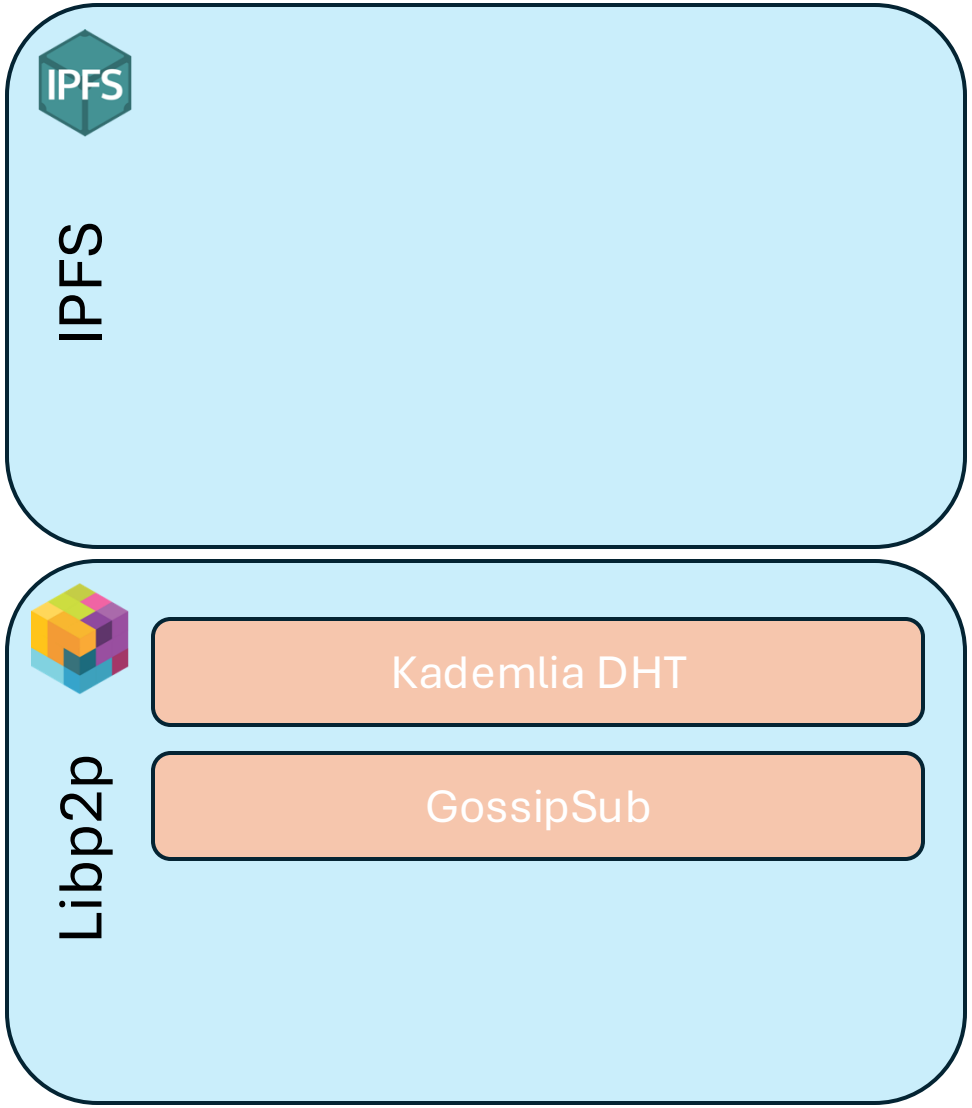
- IPFS operates on top of a DHT called Kademlia, which is quite interesting:
  - Routing tables have lots of information.
  - Contents of routing tables are (very) non-deterministic.
  - There is no specific signaling messages to manage the DHT.

IPFS: Architecture



- IPFS operates on top of a DHT called Kademlia, which is quite interesting:
  - Routing tables have lots of information.
  - Contents of routing tables are (very) non-deterministic.
  - There is no specific signaling messages to manage the DHT.
  - Content is stored in K nodes closer to the target identifier with K = 20.
  - Lookup tries to locate K nearest nodes and stops if content is found.

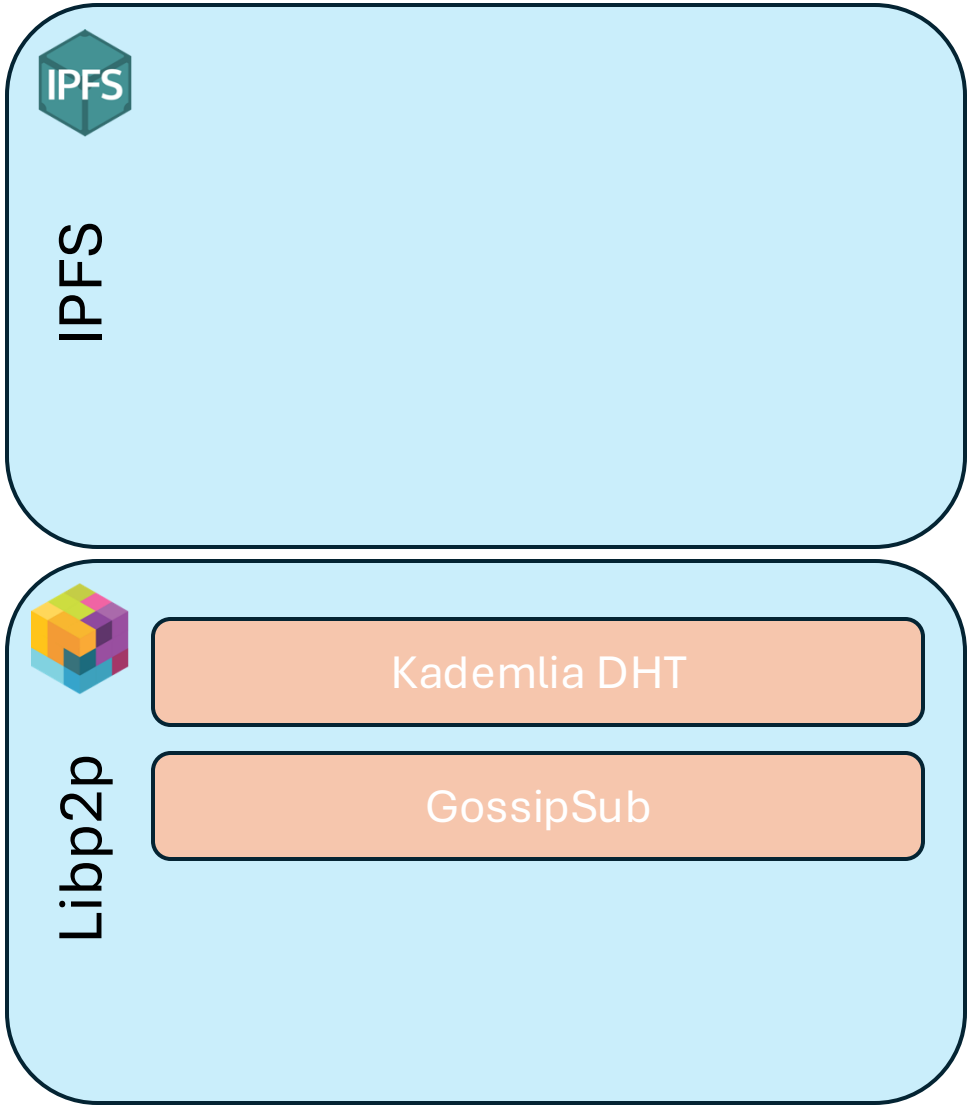
IPFS: Architecture



- Libp2p features many utilities, such as a publish-subscribe protocol
  - Topic-based
  - Designed to be Byzantine fault-tolerant
  - For each topic, a mesh is created among nodes interested in that topic.

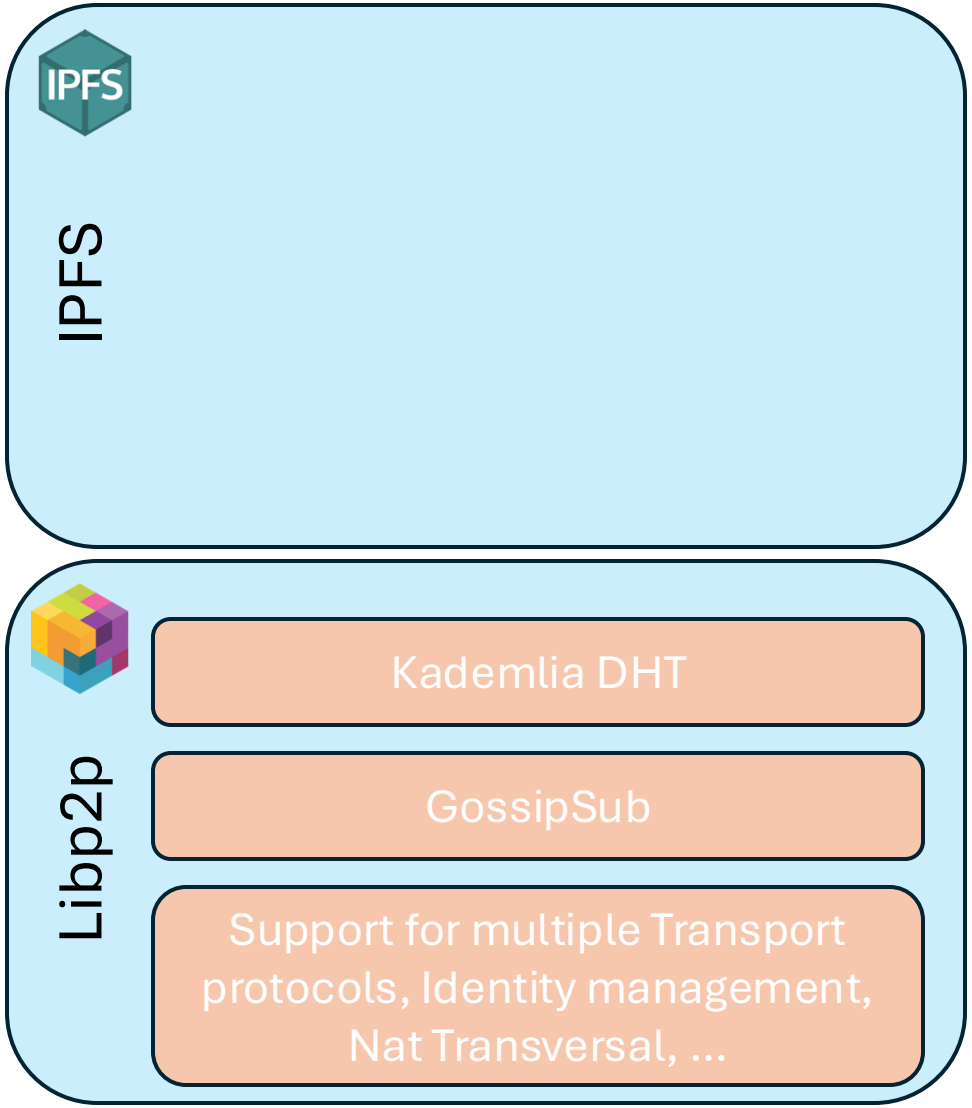
J. Leitao, J. Pereira and L. Rodrigues, "Epidemic Broadcast Trees," *2007 26th IEEE International Symposium on Reliable Distributed Systems (SRDS 2007)*, Beijing, China, 2007, pp. 301-310, doi: 10.1109/SRDS.2007.27.

IPFS: Architecture



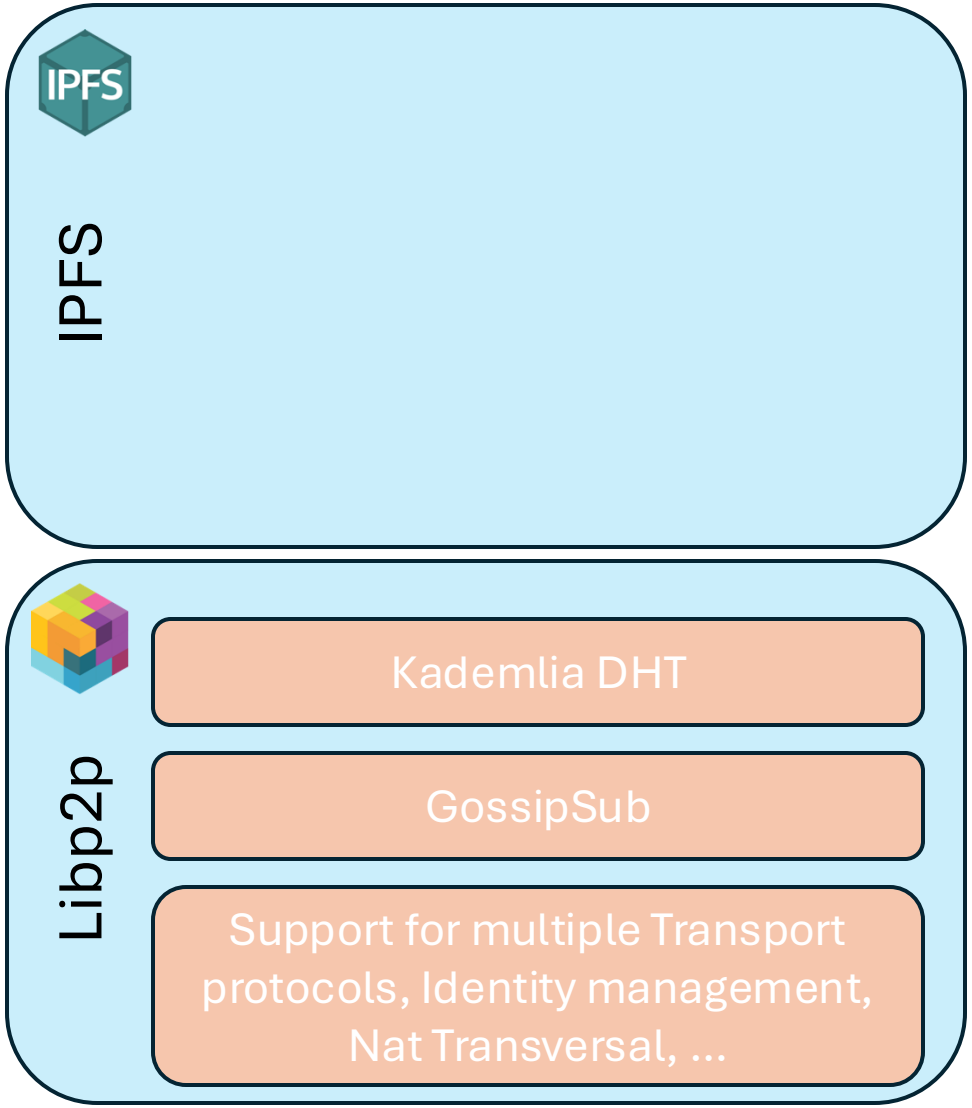
- Libp2p features many utilities, such as a publish-subscribe protocol
  - Topic-based
  - Designed to be Byzantine fault-tolerant
  - For each topic, a mesh is created among nodes interested in that topic.
  - Can use the DHT to bootstrap the mesh.
  - Interactions with neighbors in the mesh are analyzed to remove nodes that are misbehaving (using a score).

IPFS: Architecture



- Libp2p provides lots of support at the lowest level of the stack, some of which includes:
  - Identity management and verification (based on self-signed certificates)
  - Support for several transport protocols and multiplexing over these (UDP, TCP, QUIC, WebSocket)
  - Mechanisms for automated hole-punching, allowing NATed nodes to be reachable.

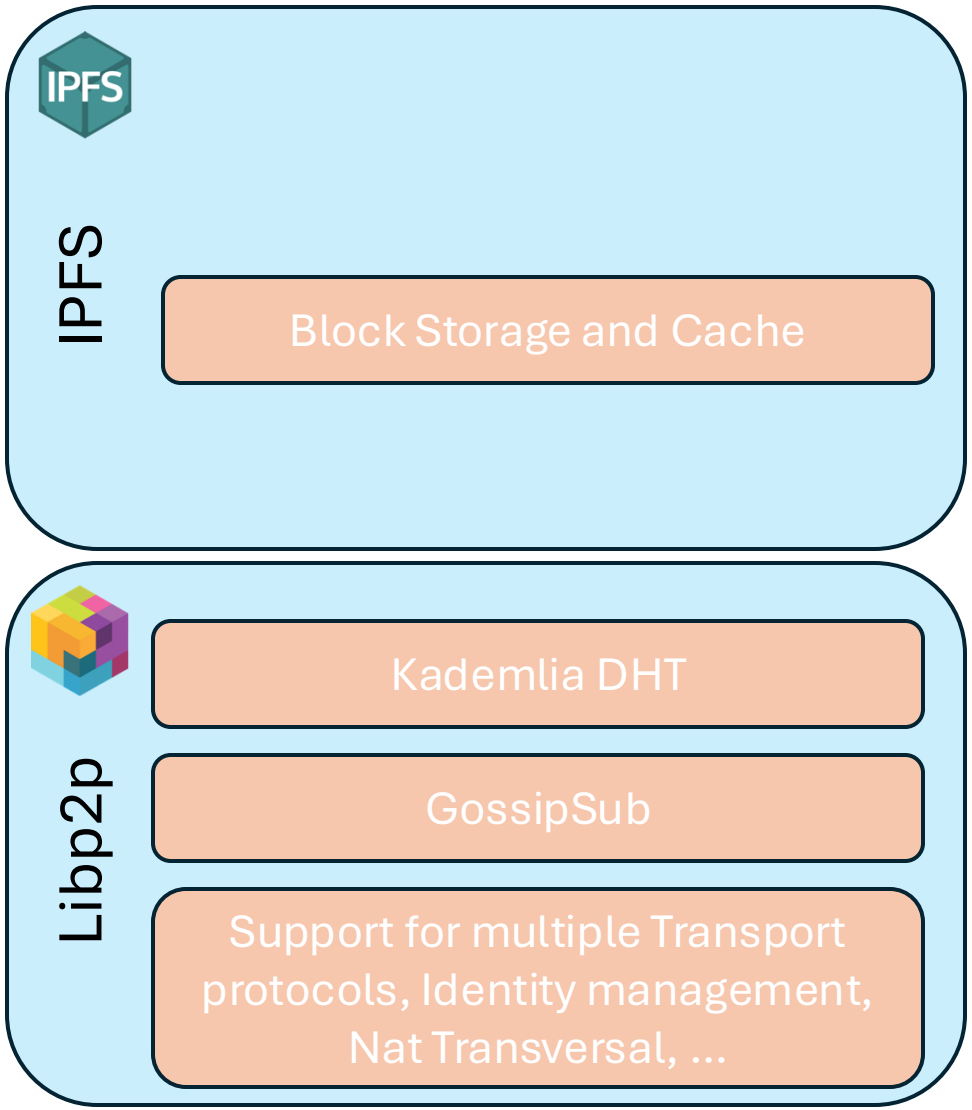
IPFS: Architecture



- Libp2p provides lots of support at the lowest level of the stack, some of which includes:
  - Identity management and verification (based on self-signed certificates)
  - Support for several transport protocols and multiplexing over these (UDP, TCP, QUIC, WebSocket)
  - Mechanisms for automated hole-punching, allowing NATed nodes to be reachable.

  - Currently being employed in many Web3 applications including Ethereum version 2.

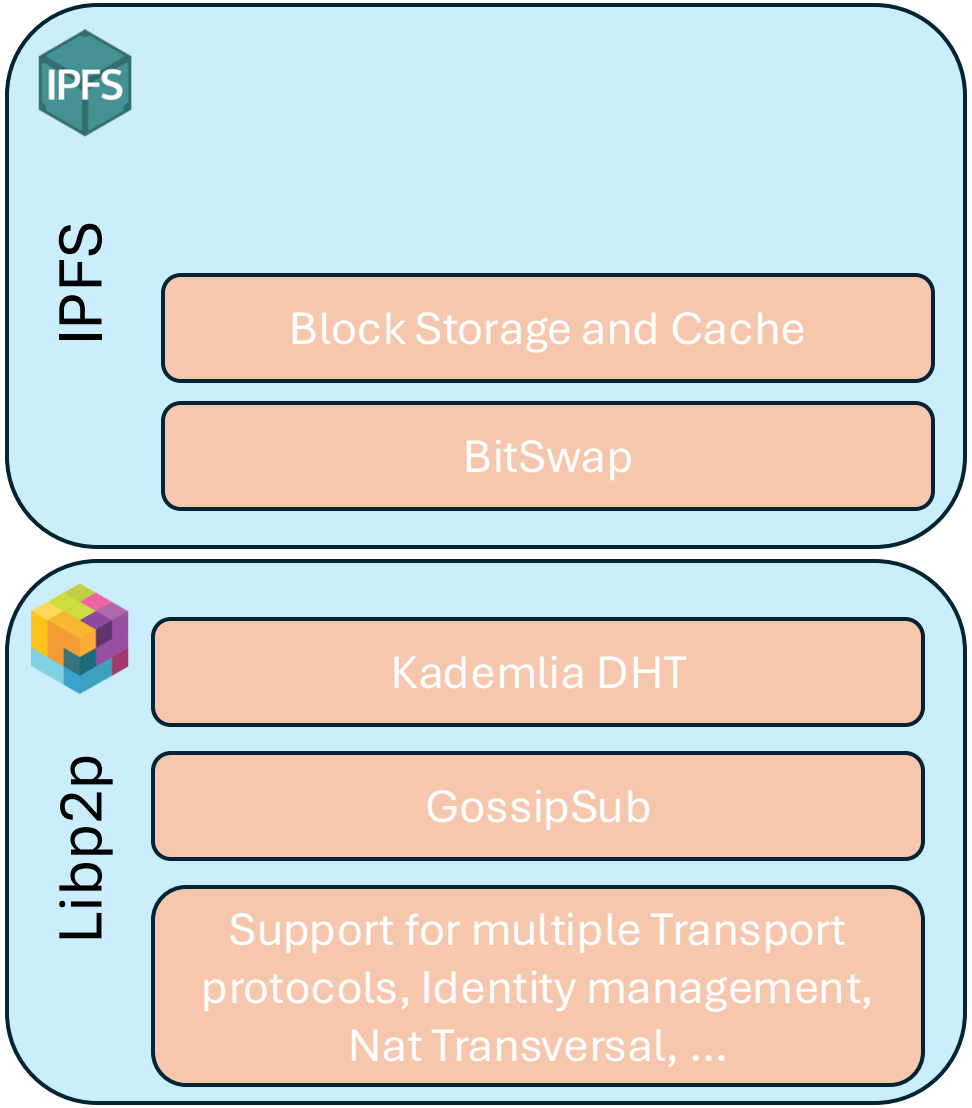
IPFS: Architecture



- At the core of the operation of IPFS (independently of what is provided by libp2p) is the Block storage and Cache.

- It stores file blocks for both files provided and accessed by the local node (and temporarily blocks that were fetched to assist other nodes to download them).

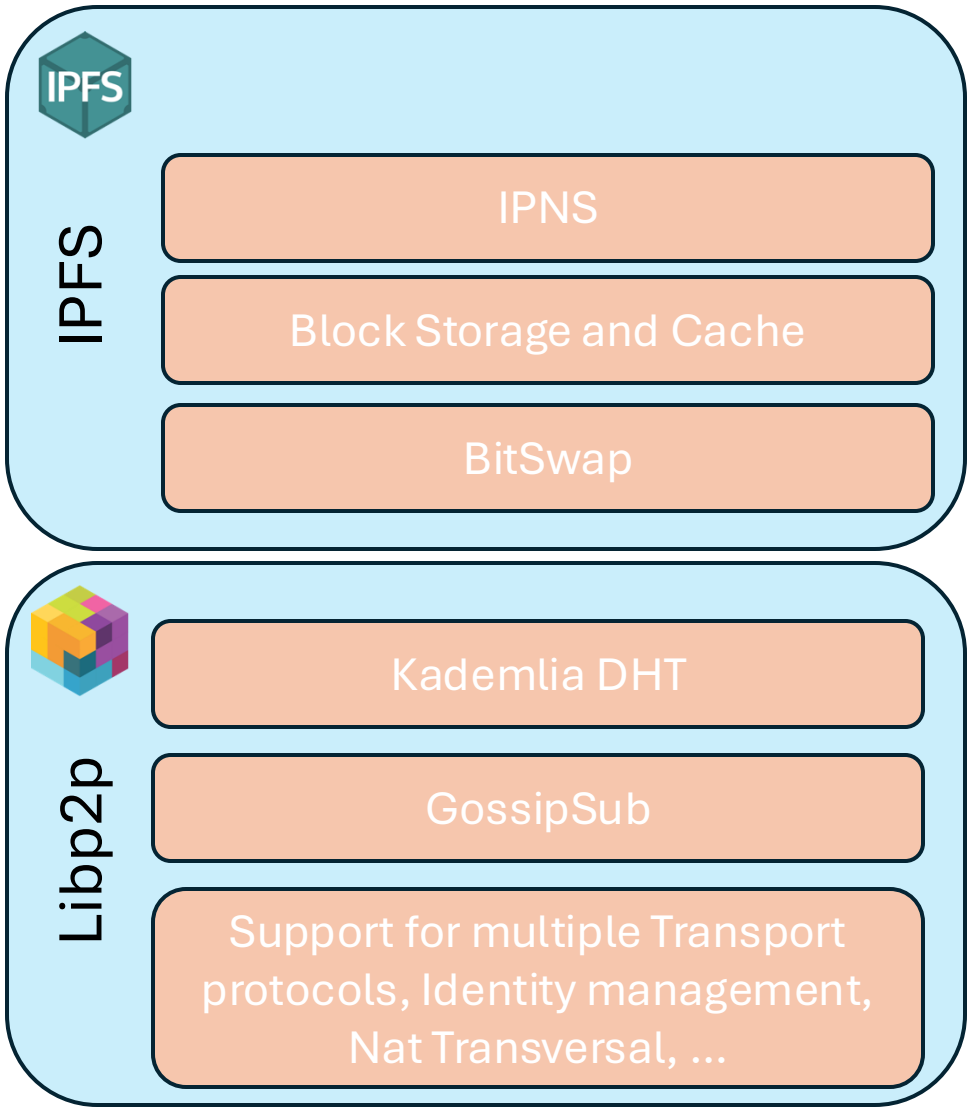
IPFS: Architecture



- Exchanges of blocks happen using the BitSwap protocol.

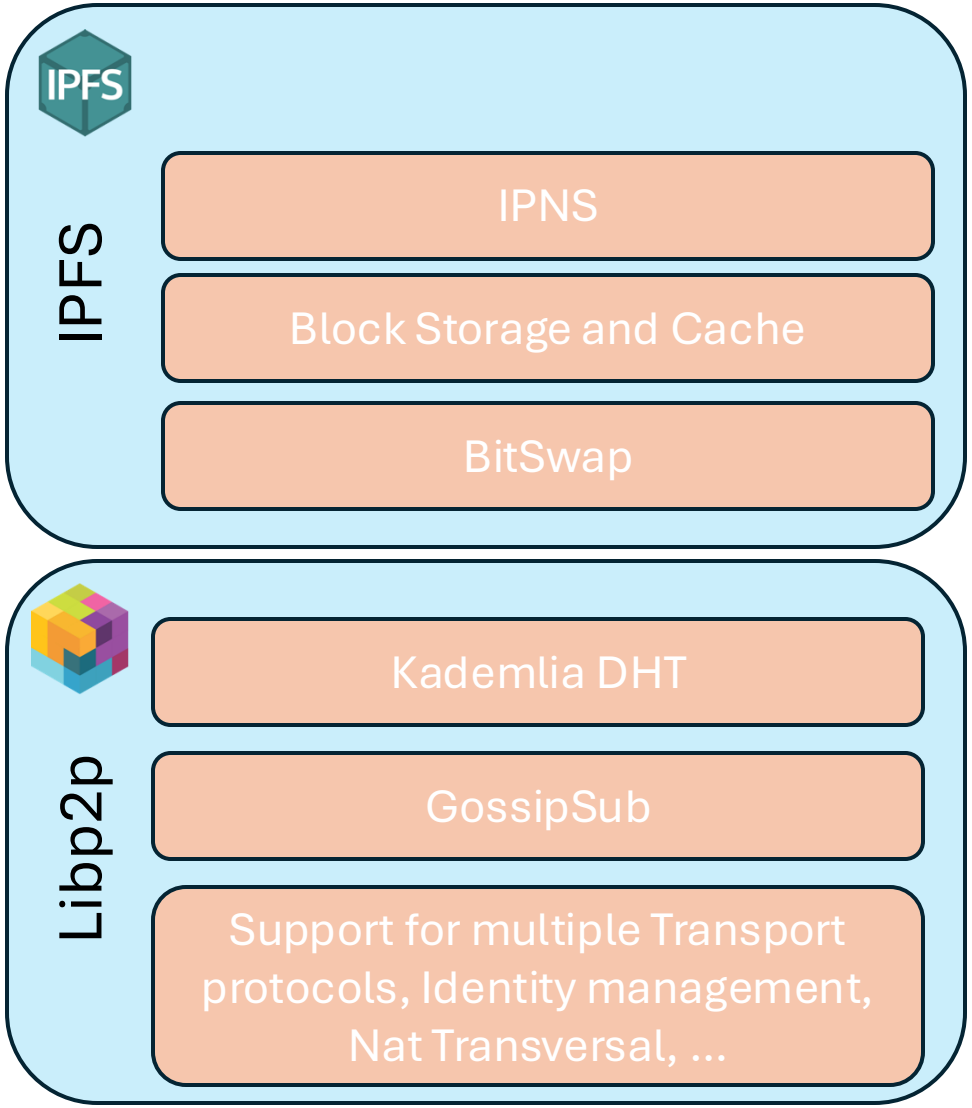
- BitSwap protocol is a gossip protocol that takes and adapts some of the ideas from bittorrent.
- For each file being downloaded, BitSwap creates a session populated with (some) direct neighbors and asks them for blocks associated to that file.
- If BitSwap is unable to fetch blocks it requests content providers to the DHT that are added to the session.

IPFS: Architecture



- Since CIDs are content based, and IPFS aims at being a way to serve web content in a decentralized fashion, whenever the root web page changes, its CID would change which makes it hard to navigate on web pages.

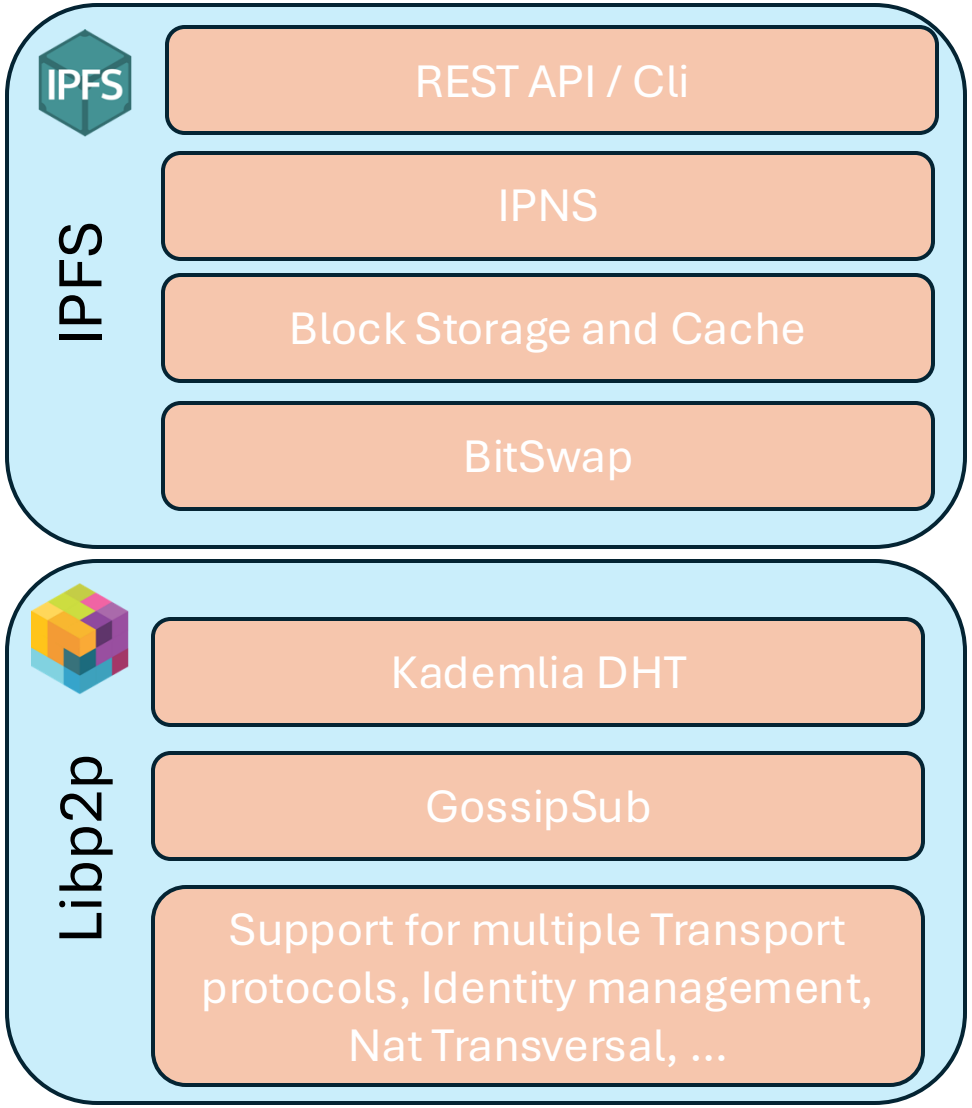
IPFS: Architecture



- Since CIDs are content based, and IPFS aims at being a way to serve web content in a decentralized fashion, whenever the root web page changes, its CID would change which makes it hard to navigate on web pages.

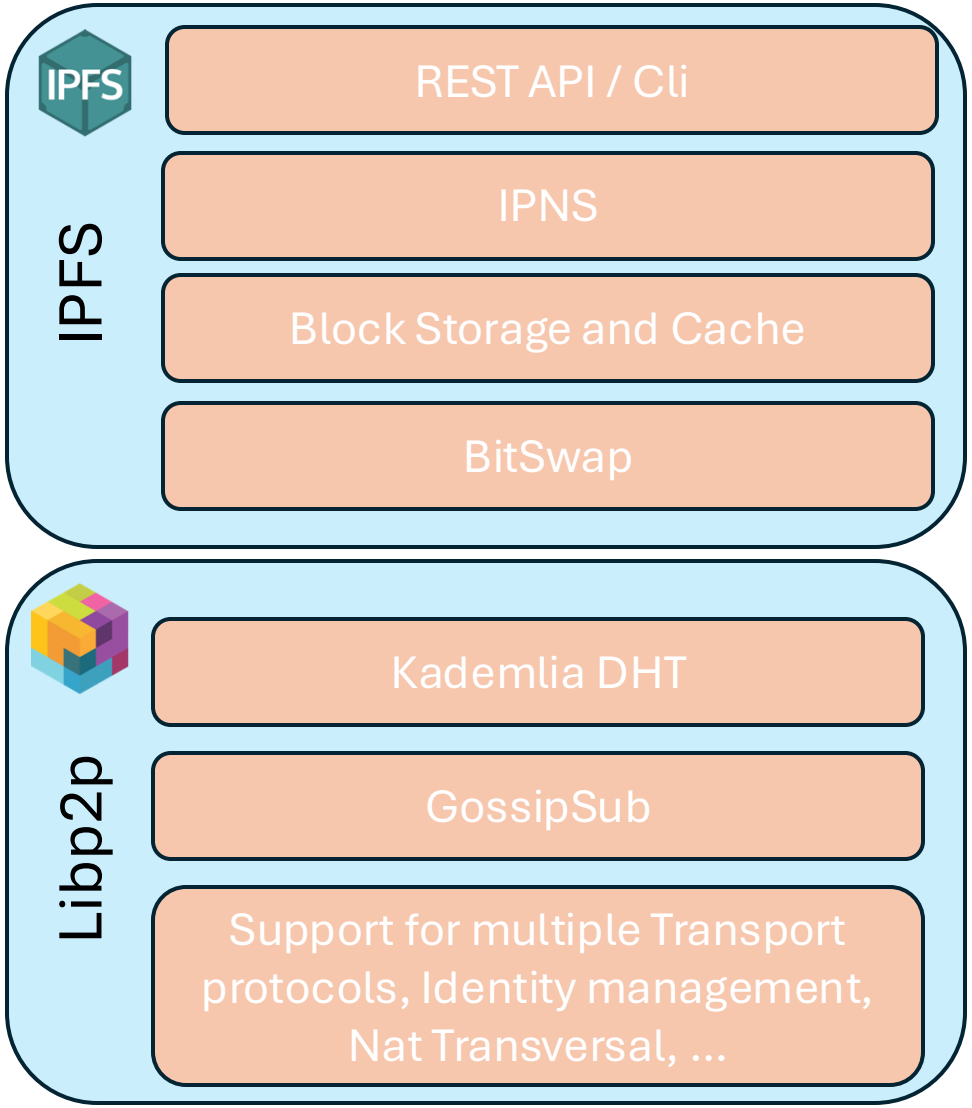
- To circumvent this, the Inter-Planetary Name System was created, that basically allows to create a persistent CID associated to content that can change over time.

IPFS: Architecture



- How can we interact with a IPFS node (speaking about Kubo).
- IPFS supports a command line cli interface to execute requests over the local IPFS node (publish content, retrieve a CID, garbage collect the block cache etc..)
- IPFS nodes also have a REST API by default blocked to access from the local node but that can be open to the world.

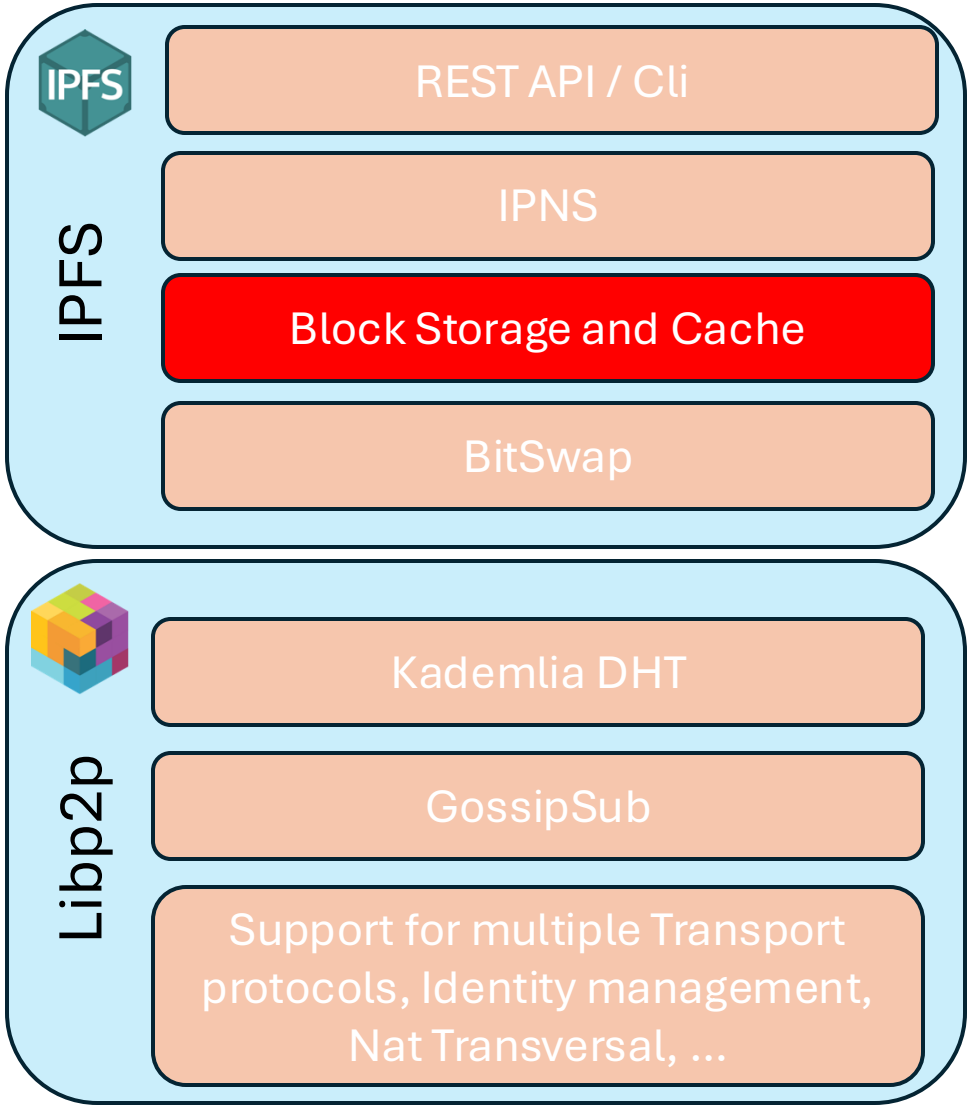
IPFS: General Operation



Publish Content to IPFS:

- Partition file into blocks and generate the DAG for the file.
- Generate a CID (via hash function) for all blocks (including the one with the DAG).

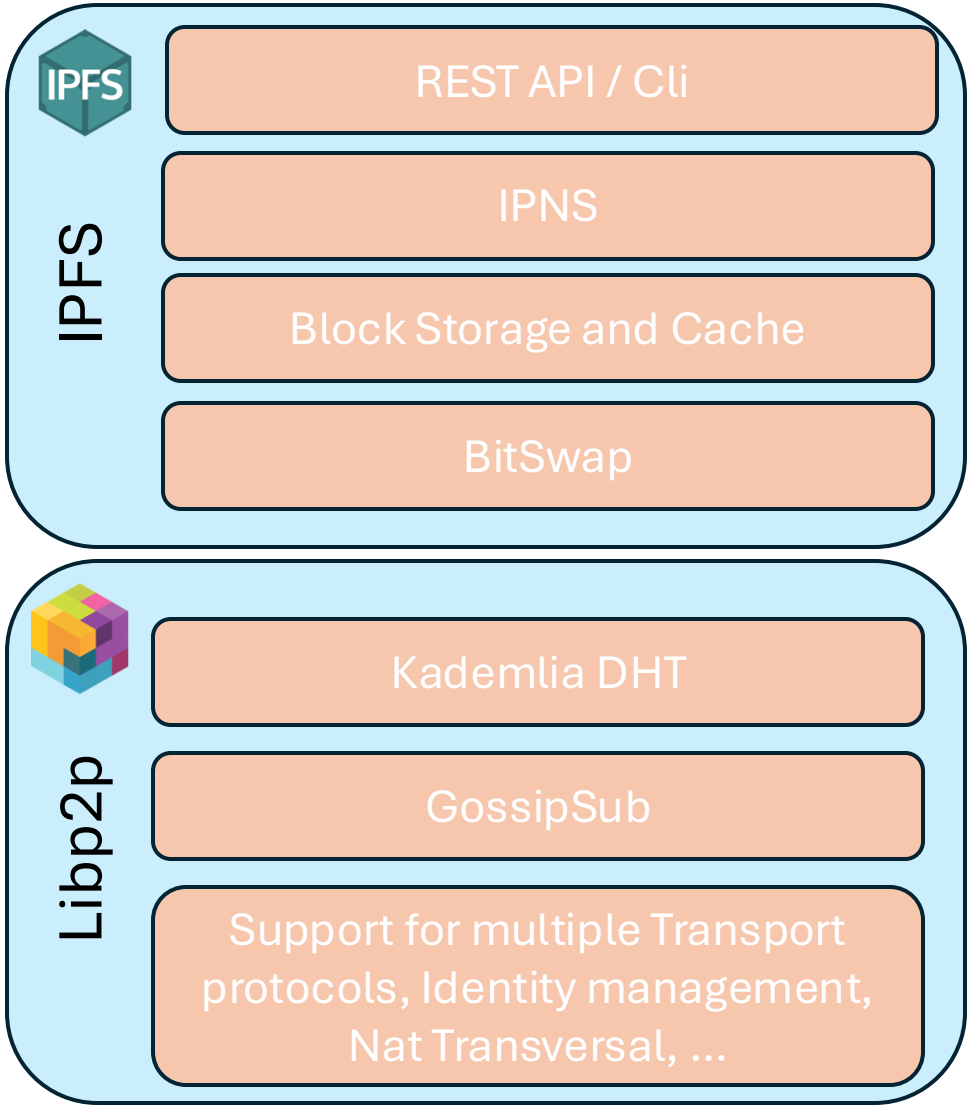
IPFS: General Operation



Publish Content to IPFS:

- Partition file in to blocks and generate the DAG for the file.
- Generate a CID (via hash function) for all blocks (including the one wiht the DAG).
- Store blocks on the local block storage.

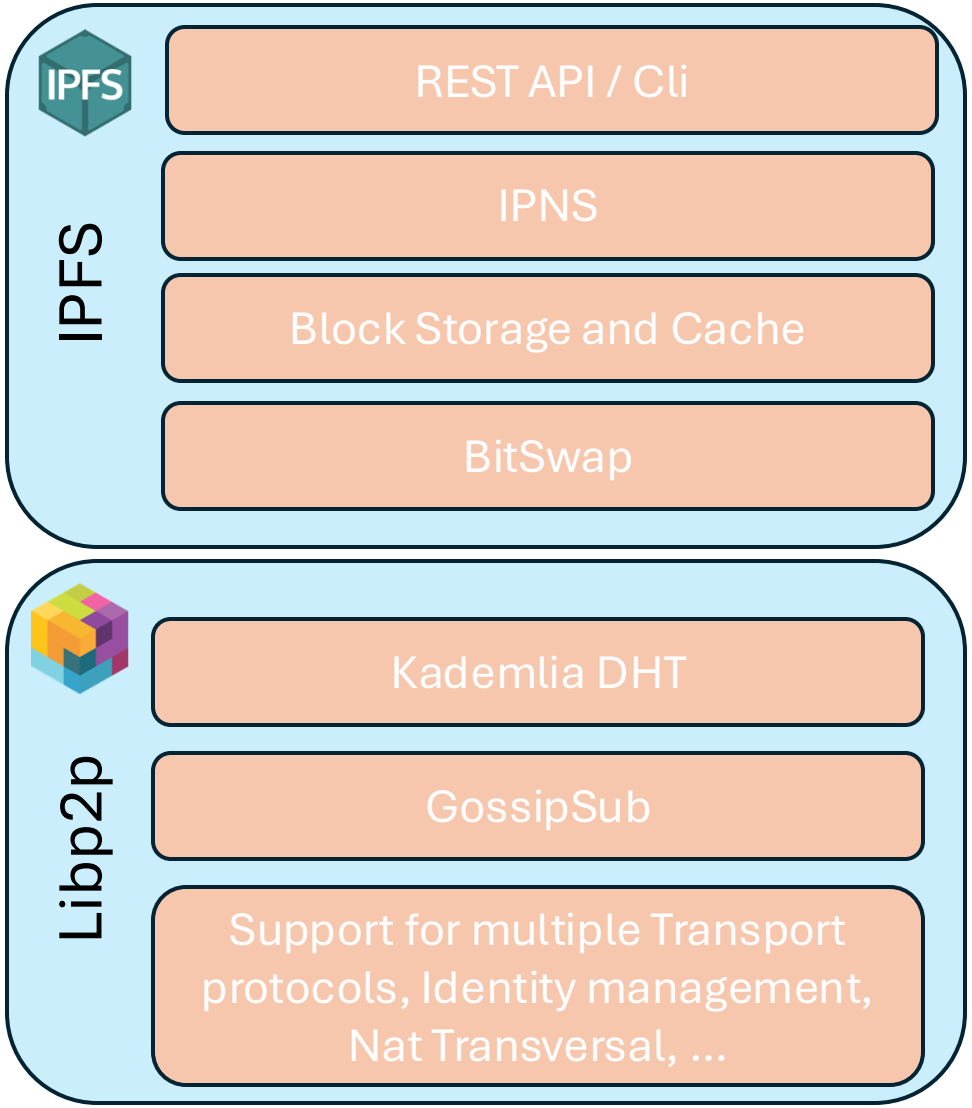
IPFS: General Operation



Publish Content to IPFS:

- Partition file in to blocks and generate the DAG for the file.
- Generate a CID (via hash function) for all blocks (including the one with the DAG).
- Store blocks on the local block storage.
- Generate provider records and publish them to K (k=20) closes nodes of those CIDs on the DHT.

IPFS: General Operation



Publish Content to IPFS:

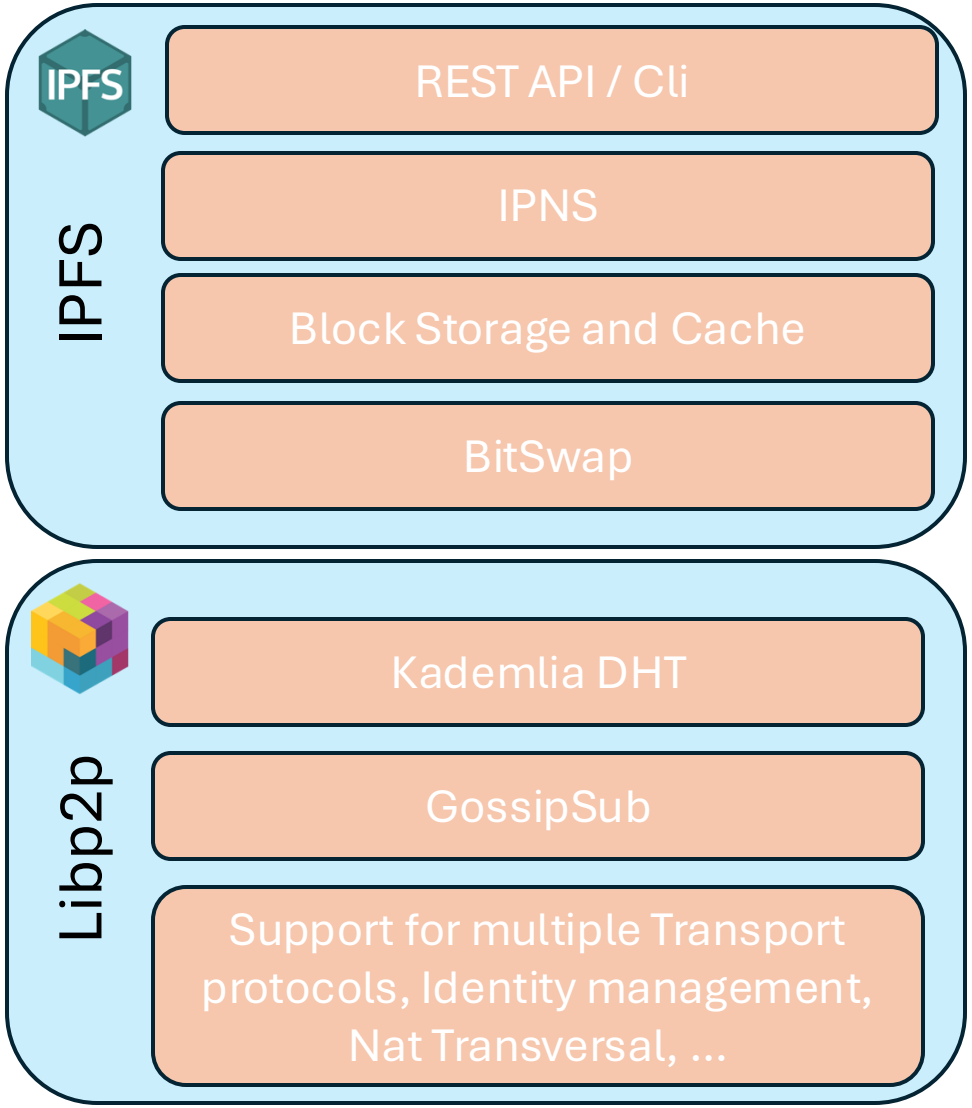
- Partition file in to blocks and generate the DAG for the file

Since the DHT relies on soft state to deal with churn, repeat the process every 12 hours.

This is a huge burden on clients that publish lots of content to IPFS.

- Generate provider records and publish them to K (k=20) closes nodes of those CIDs on the DHT.

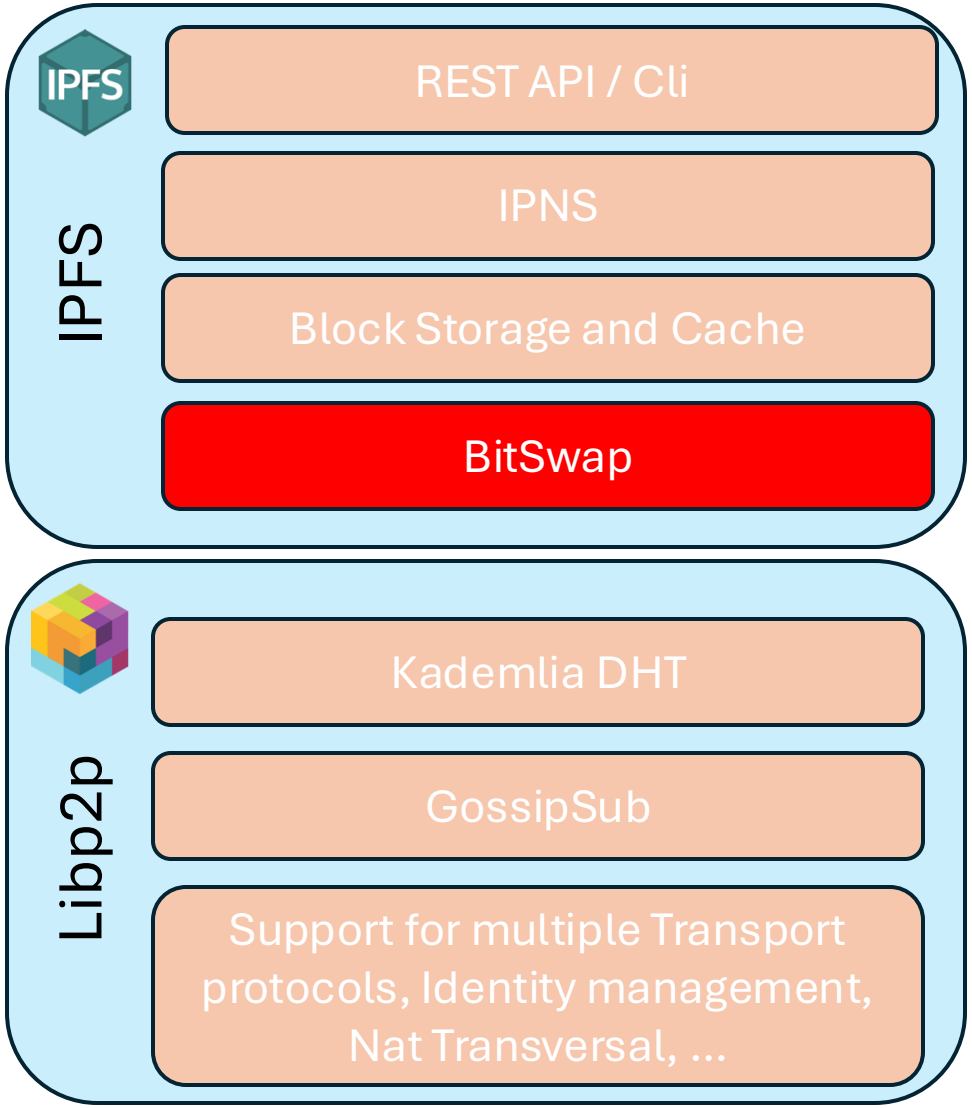
IPFS: General Operation



Fetch Content from IPFS:

- Access the CID of the DAG for the file (through external means).

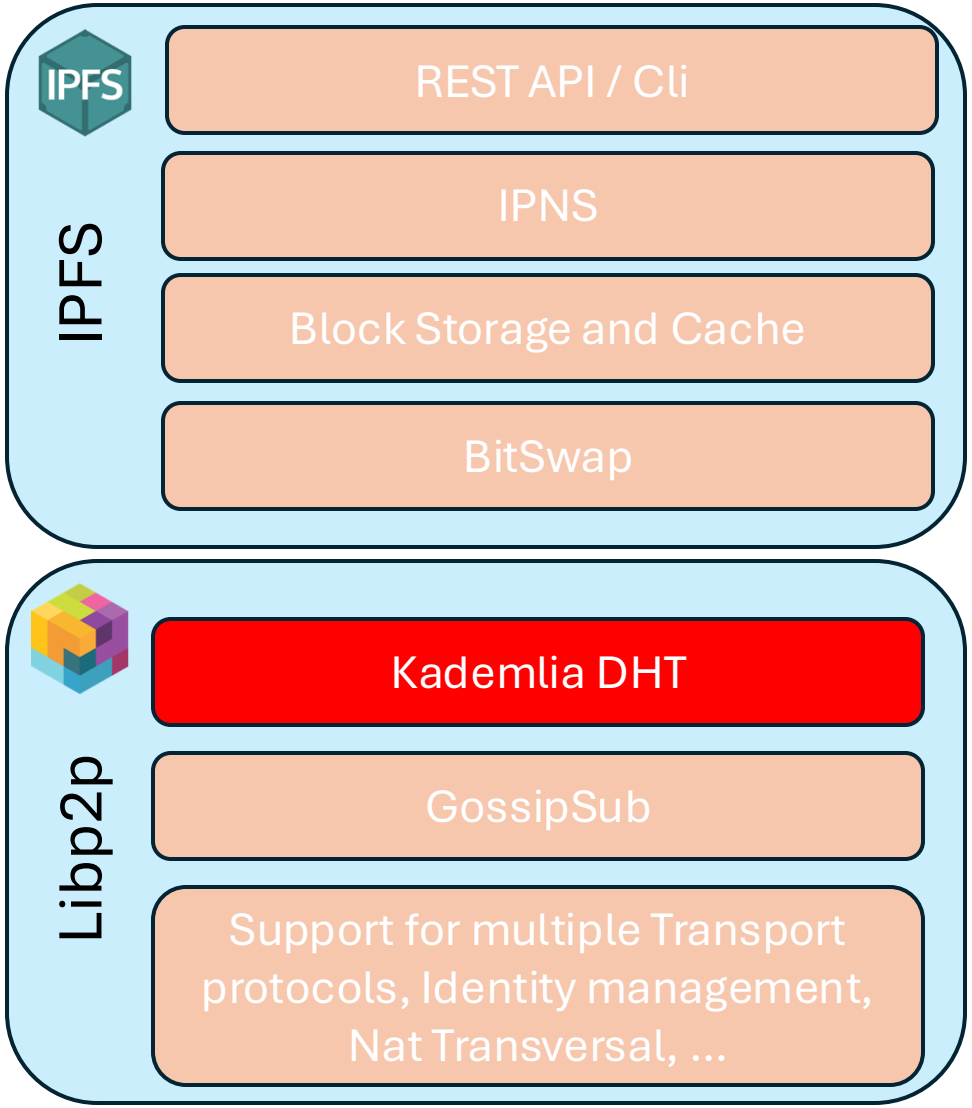
IPFS: General Operation



Fetch Content from IPFS:

- Access the CID of the DAG for the file (through external means).
- Try to fetch that block (and other referred on it) through BitSwap.

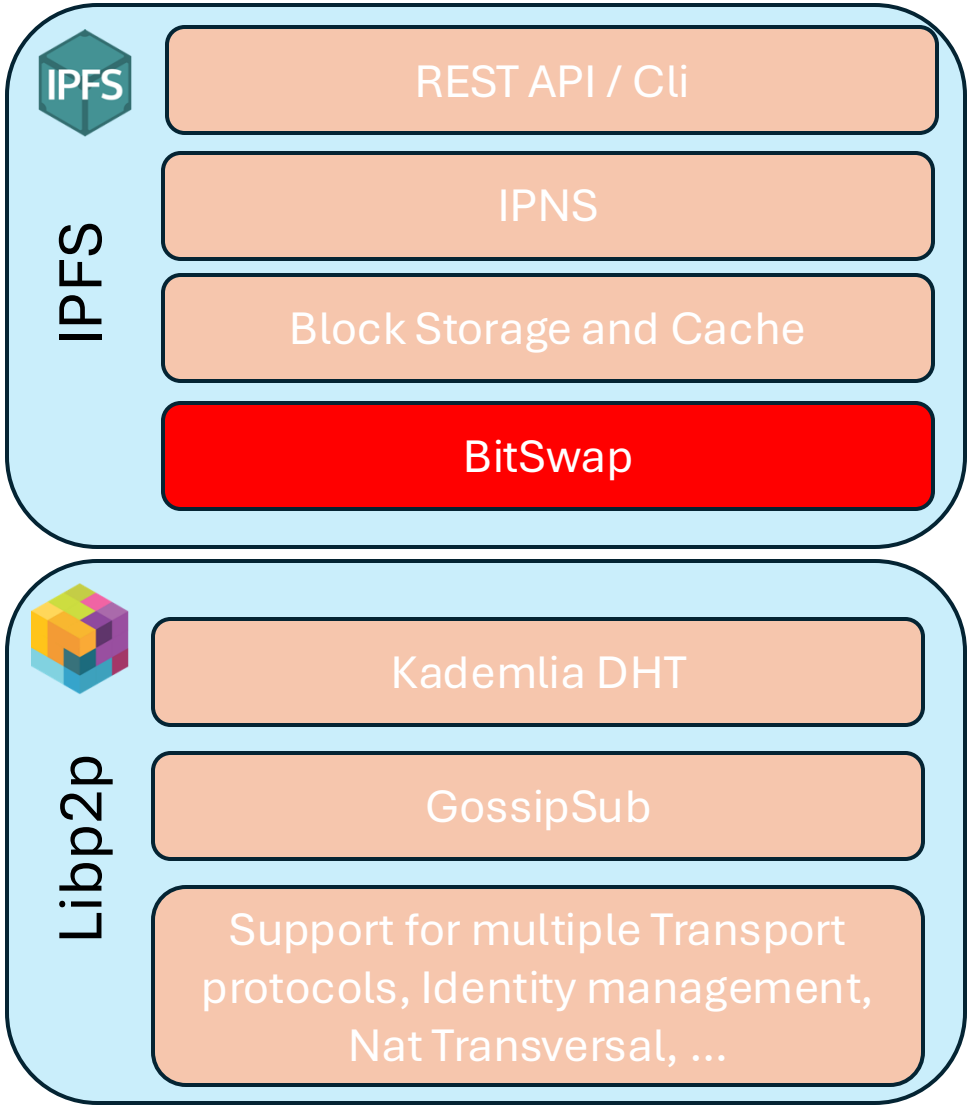
IPFS: General Operation



Fetch Content from IPFS:

- Access the CID of the DAG for the file (through external means).
- Try to fetch that block (and other referred on it) through BitSwap.
- If after 1s BitSwap is unable to start fetching the block from a neighbor, use the DHT to get a provider record.

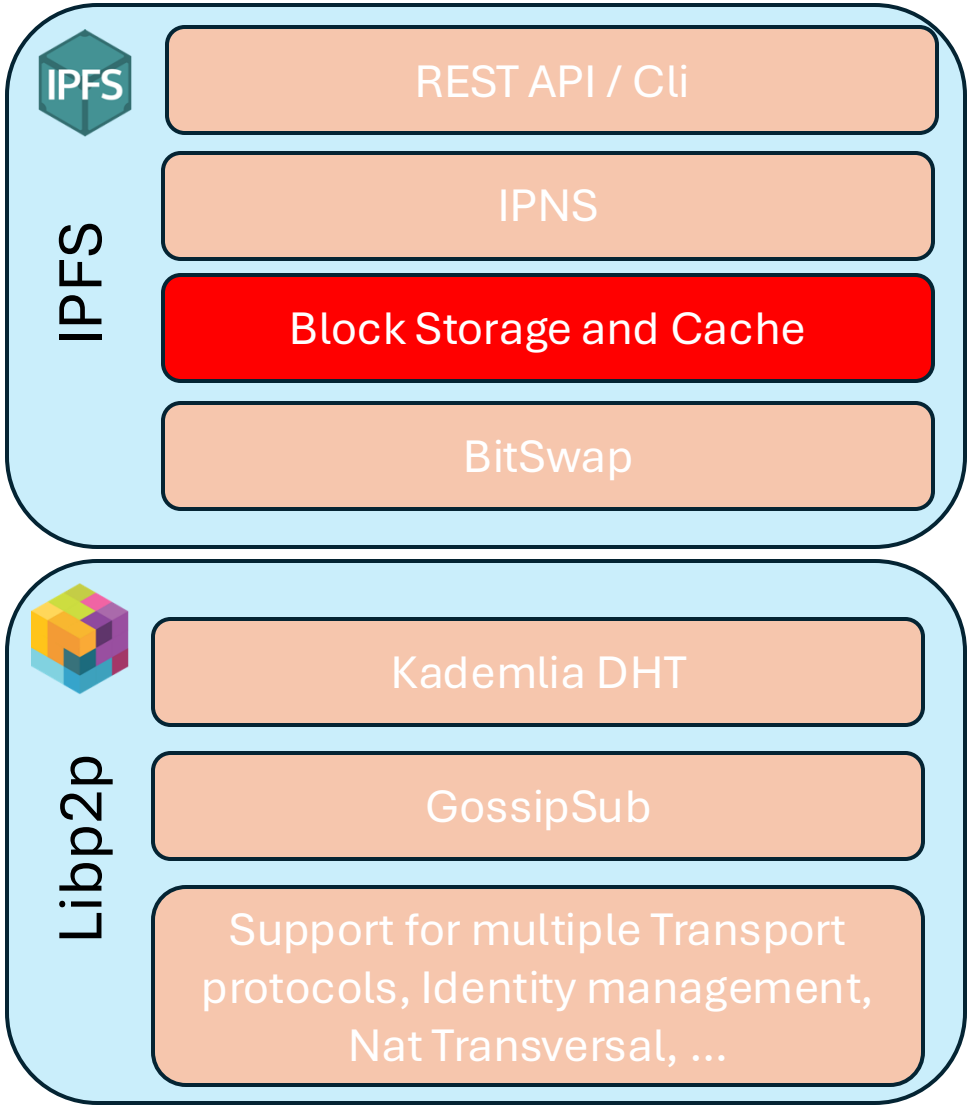
IPFS: General Operation



Fetch Content from IPFS:

- Access the CID of the DAG for the file (through external means).
- Try to fetch that block (and other referred on it) through BitSwap.
- If after 1s BitSwap is unable to start fetching the block from a neighbor, use the DHT to get a provider record.
- Add the peer(s) on the provide record to the BitSwap session.

IPFS: General Operation



- Blocks that are downloaded can be temporarily cached on the Block Storage and Cache (used to answer Bitswap requests).
- If a neighbor requests a block via BitSwap, a node can (optionally) try to fetch that block to help that neighbor (block becomes cached)
- After downloading content, I can **reprovide** it permanently (i.e., replicate a publish provider records to the DHT).

IPFS: Dealing with soft state

- The fact that provider records in the DHT have a time to live (12h as far as I know) means that providers of large number of files have a huge burden to periodically republish the content.

IPFS: Dealing with soft state

- The fact that provider records in the DHT have a time to live (12h as far as I know) means that providers of large number of files have a huge burden to periodically republish the content.

- This led to the emergence of Pinning Services:
  - Third party services.
  - Clients can store files in them, and they will be responsible for the periodic republishing.
  - Ensures that content is still accessible even if the client that publishes it is offline.

IPFS: Dealing with soft state

- The fact that provider records in the DHT have a time to live (12h as far as I know) means that providers of large number of files have a huge burden to periodically republish the content.

- This led to the emergence of Pinning Services:
  - Third party services.
  - Clients can store files in them, and they will be responsible for the periodic republishing.
  - Ensures that content is still accessible even if the client that publishes it is offline.
  - Usually, a paid service ☺

IPFS: Interactions via browsers

- Aiming to be a way to provide access to web content to common users in a decentralized way (and ensuring availability and censorship-resistance) means supporting “normal” users.

IPFS: Interactions via browsers

- Aiming to be a way to provide access to web content to common users in a decentralized way (and ensuring availability and censorship-resistance) means supporting “normal” users.
- Normal users can use a browser and probably not run an IPFS instance.

IPFS: Interactions via browsers

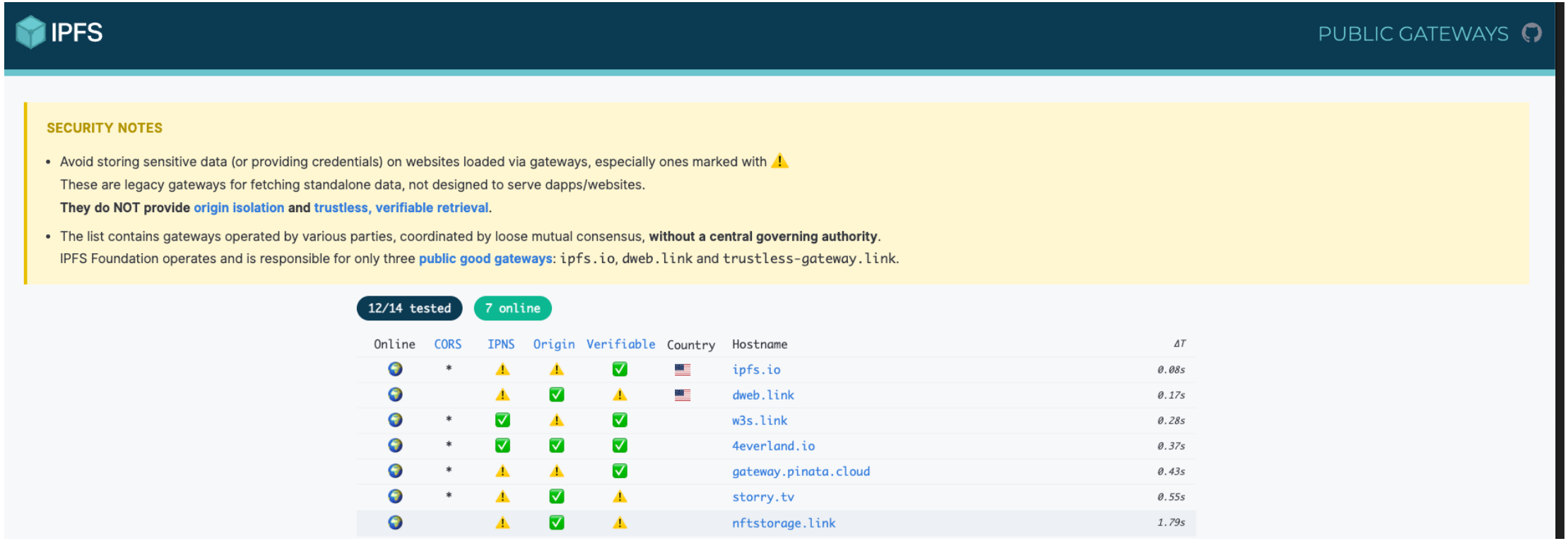
- Aiming to be a way to provide access to web content to common users in a decentralized way (and ensuring availability and censorship-resistance) means supporting “normal” users.
- Normal users can use a browser and probably not run an IPFS instance.

- Two ways to deal with this:

1. (Exoteric) a JavaScript version of IPFS can run directly on browsers.
2. (Standard) browser contacts an IPFS gateway (an IPFS node exposing its REST API) that will fetch the content from IPFS and return it to the client as a HTTP response.

IPFS: Gateways

- There is a big concern in keeping the system decentralized, so although PL operates a set of gateways, anyone can operate one and publish it.



IPFS: Impact of IPFS

- IPFS has had significant success, it has been launched in 2015, still operating now-a-days, with a variable number of servers throughout the world.

IPFS: Impact of IPFS

- IPFS has had significant success, it has been launched in 2015, still operating now-a-days, with a variable number of servers throughout the world.

- It has been somewhat informally adopted as the storage layer by the Web3 community:
  - Significant success in storing and providing NFTs.
  - Several decentralized services being built on top of it (e.g., youtube-like).
  - Big players publishing mirrors of sites to IPFS (e.g., Wikipedia)

IPFS: Impact of IPFS

- IPFS has had significant success, it has been launched in 2015, still operating now-a-days, with a variable number of servers throughout the world.

- It has been somewhat informally adopted as the storage layer by the Web3 community:
  - Significant success in storing and providing NFTs.
  - Several decentralized services being built on top of it (e.g., youtube-like).
  - Big players publishing mirrors of sites to IPFS (e.g., Wikipedia)
- Less legal impact:
  - Used as one of the storage systems for copyrighted academic works
  - Weird things being published there…

Story time

- I have been operating an IPFS public gateway from home for years (mostly to support some research activities)
- For some time, I had my public gateway published in the official list…

Story t

- I have bee (mostly tc
- For some

using ipfs.joaoleitao.org as a IPFS gateway for F-Droid Externos Caixa de entrada ×

Hans-Christoph Steiner <hans@guardianproject.info>
para mim, team@f-droid.org, Jochen, Michael

Hello,

F-Droid is rolling out beta support for IPFS Gateways and we found your gateway on the list of public ones. So we have included it in our beta test, let us know if that is a problem and we can remove it. For our official mirrors, we require a privacy policy, and we'd like to offer the same level of support with IPFS. Would you be willing to post a privacy policy on your IPFS gateway? Here are some examples you can draw from:

* https://ftp.lysator.liu.se/datahanteringspolicy.txt
* https://mirror.fcix.net/policy/
* https://plug-mirror.rcac.purdue.edu/info.html
* https://ftp.fau.de/datenschutz/

Thanks for public IPFS gateway!

.hc

--
Signal: +13478504872
PGP fingerprint: EE66 20C7 136B 0D2C 456C 0A4D E9E2 8DEA 00AA 5556
https://keys.openpgp.org/search?q=EE6620C7136B0D2C456C0A4DE9E28DEA00AA5556

Story time

- I have been operating an IPFS public gateway from home for years (mostly to support some research activities)
- For some time, I had my public gateway published in the official list…

…this was the OK story because soon enough…

Story time

Report received: (#13900737) Cloudflare is forwarding a notice from Public Interest Registry Externos Caixa de entrada ×

Cloudflare <notices@trustandsafety.cloudflare.com>
para mim

sexta, 27/01/2023, 19:20

Traduzir para português

##- Please type your reply above this line -##

Thank you for contacting Cloudflare's Registrar Support Services.

In order to expedite our investigation, we would ask that you visit our Abuse Reporting Form: https://www.cloudflare.com/abuse to submit your complaint. In order to make your experience efficient and expeditious, please be sure to select the appropriate Registrar category and include as much information about your complaint as possible. Example:

Registrar – Trademark Complaint (UDRP)

– Domain infringing on your TM or Copyright
– Domain Disputes between users and or entities

Registrar – Inaccurate WHOIS

– Reporting of invalid contact information of a domain

Registrar – General

– If you are looking to contact the Registrant directly for:
– the availability of their domain for purchase
– services they may provide

If you have a complaint regarding the content of a domain's website, please select the appropriate abuse category and provide as much detail as possible to expedite our investigation. Example: Copyright Infringement & DMCA Violations | Trademark Infringement | Phishing & Malware etc…

If you require assistance with managing your domain(s) and services, please visit our Customer Service Support page https://support.cloudflare.com/

IMPORTANT:

Please note that due to the volumes of reports that Cloudflare handles, you may not receive a response from us; however, don't worry, that means that we have all the information we need to proceed with our investigation.

Sincerely,

Cloudflare Trust & Safety

'ears

ial list...

Story time

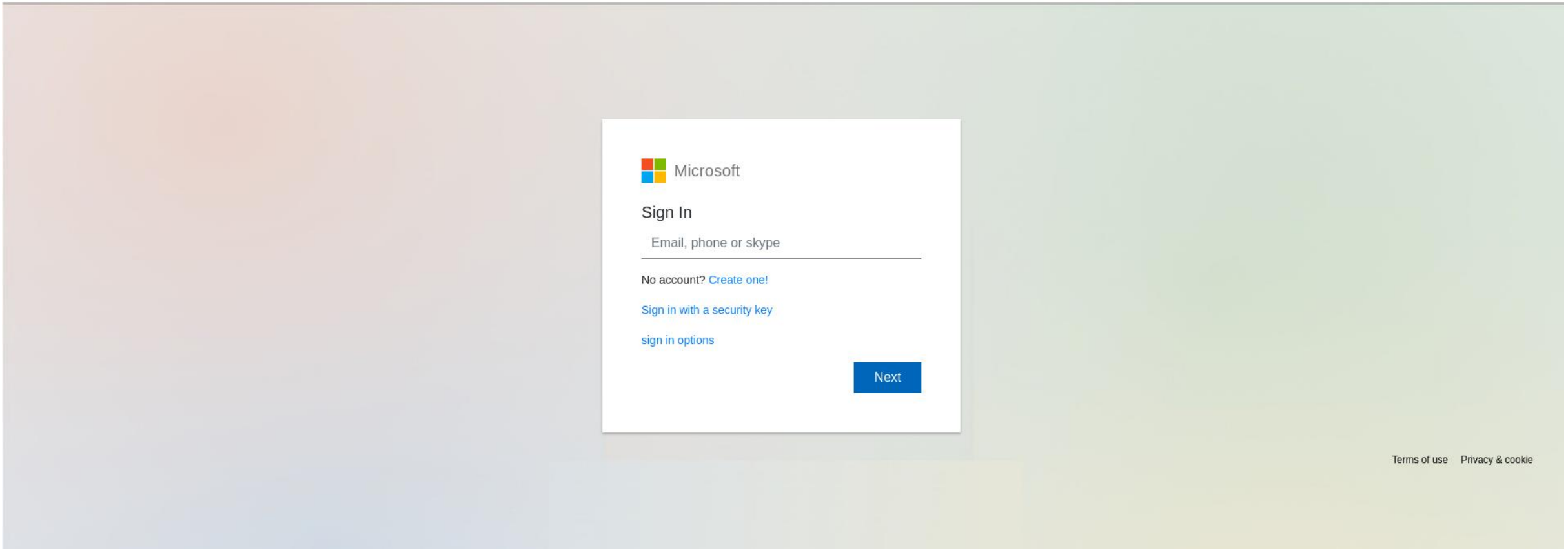
Report received: (#13900737) Cloudflare is forwarding a notice from Public Interest Registry Externos Caixa de entrada ×

'ears

Cloudflare <notices@trustandsafety.cloudflare.com>
para mim

sexta, 27/01/2023, 19:20

Screenshot taken at Thu, 26 Jan 2023 22:31:30 GMT from
openphish_premiumplus: https://bafybeiapf2imk3xqrrrmxbeipxnlbtf4s35dpfj7yled7vjk3cmsb7giue.ipfs.ipfs.joaoleitao.org/



Story time

Barao, Salome <sbarao@opsecsecurityonline.com> escreveu no dia quarta, 13/12/2023 à(s) 15:17:

=====>>>Por favor, não ignore este email. Ainda não recebemos resposta nenhum em relação a este assunto.

IMPORTANTE: Precisamos urgentemente que sejam eliminadas duas páginas fraudulentas (Phishing) do seu website!

Prezado Doutor Leitão,

Estamos escrevendo desde o Centro de Operações Anti-Fraude OpSec Online de parte da Norton Lifelock e Yahoo. Há alguns dias atras, escrevemos sobre este assunto, mas este problema ainda não foi solucionado.

Avisamos que o seu website foi comprometido (hackeado) e está hospedando uma fraude atacando clientes desavisados da Norton Lifelock para roubar informações de contas.

Segue a URL da pagina fraudulenta:

983355 **hxxp://bafkreifir4saecxkhxs4n33asx4kp6byuo2v7fmfhg4i7na2xf62q2hyqm [.] ipfs [.] ipfs [.] joaoleitao [.] org/**
984115 **hxxps://bafybeienbcpiqmw6jwvunfcwj2pk6kmpgiaqekf4agd52p5gsliwyw6wva [.] ipfs [.] ipfs [.] joaoleitao [.] org/#hghfg@yahoo [.] com**

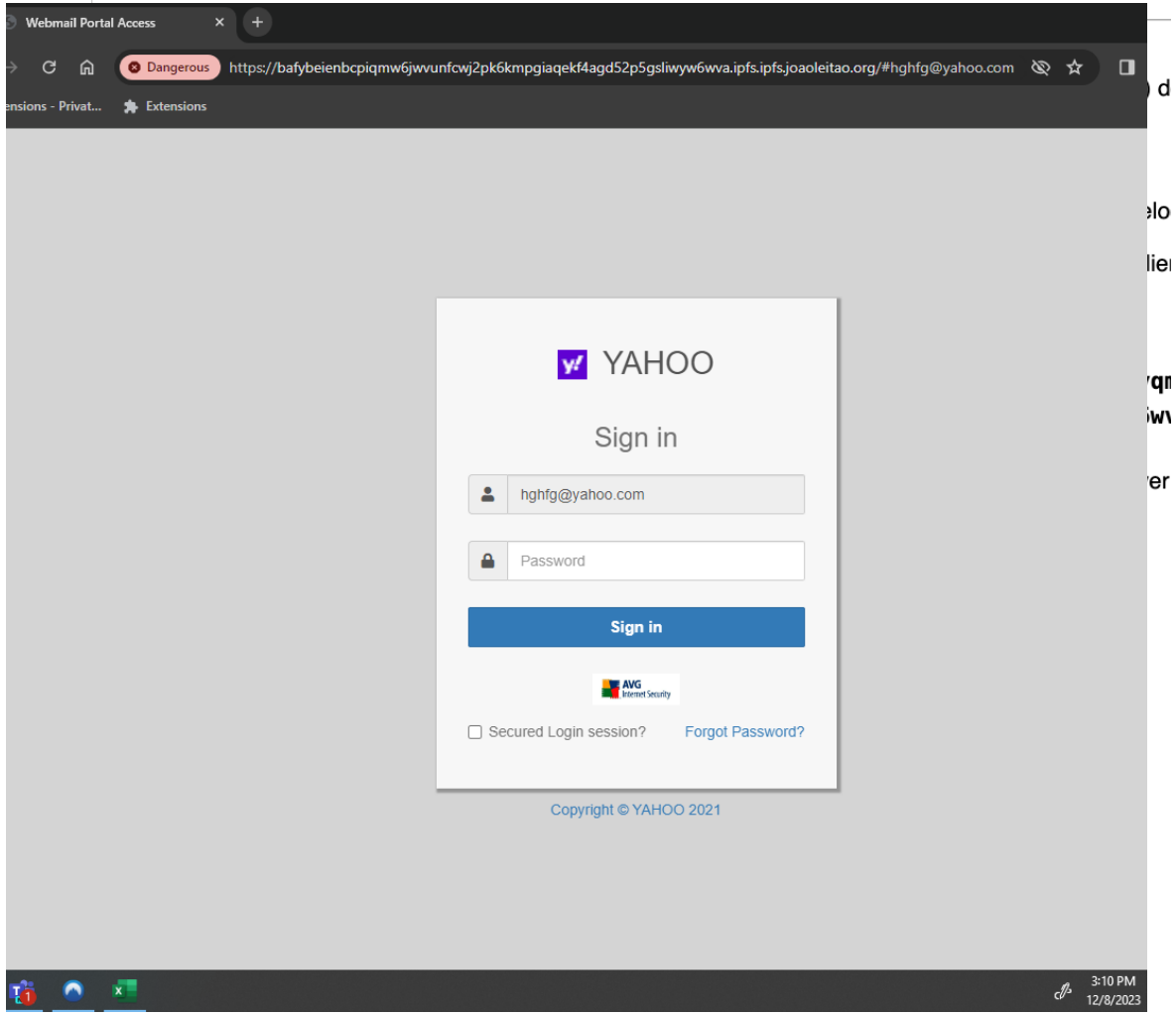
Como prova dessa fraude, uma captura de tela da mesma URL está anexada aqui para que possam ver que este phishing existe.

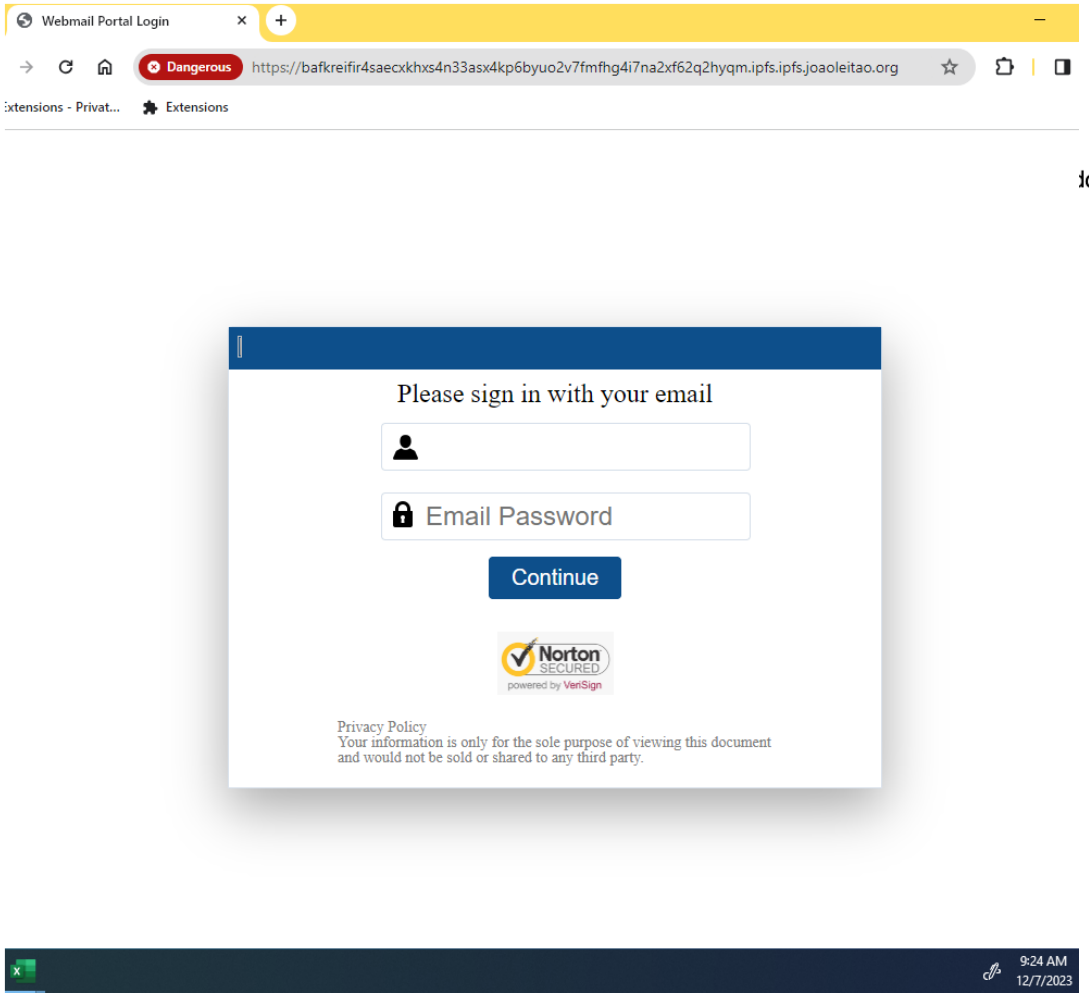
O endereço IP de hospedagem desse fraude é: 89.115.15.153
Por favor, exclua estes arquivos fraudulentos do seu site de modo a desabilitar a fraude.

Story time

Barao, Salome <sbarao@opsecsecurityonline.com> escreveu no dia quarta, 13/12/2023 à(s) 15:17:

=====>>>Por favor, não ignore este email. Ainda não recebemos resposta nenhum em relação a este as





Story time

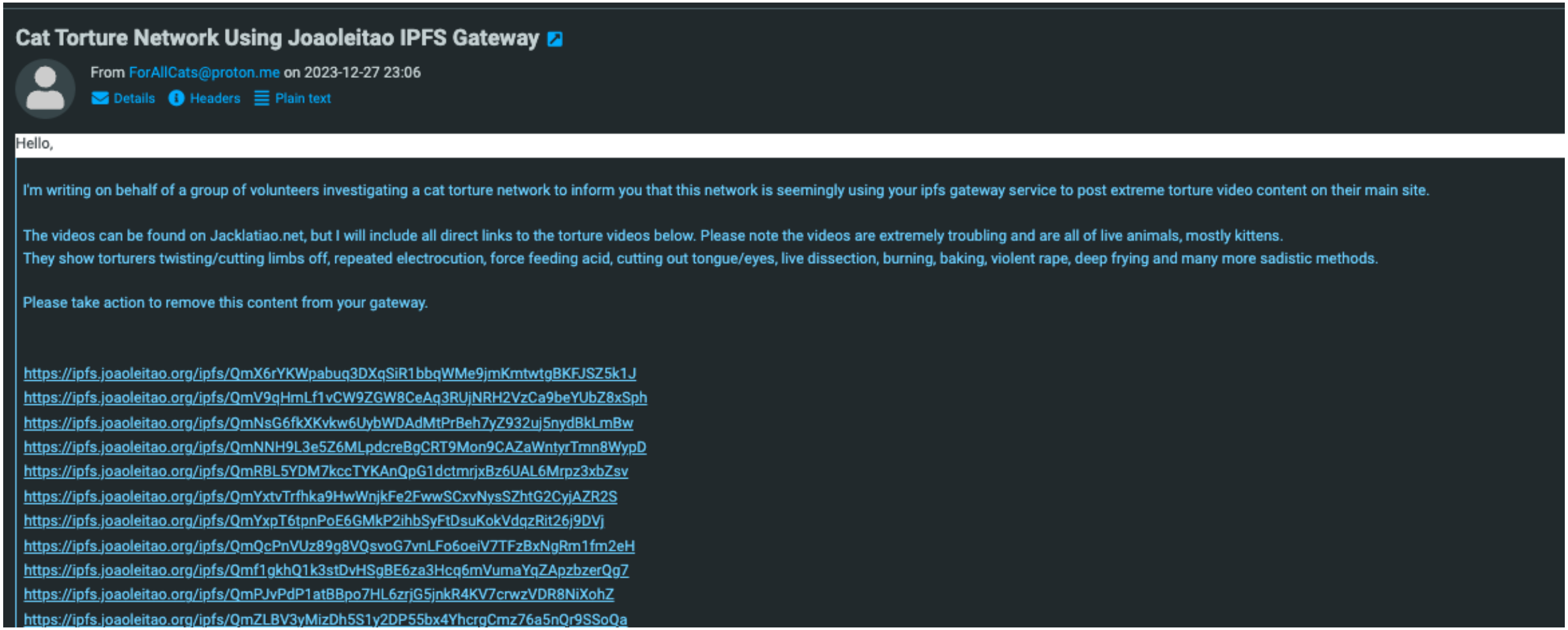
- I have been operating an IPFS public gateway from home for years (mostly to support some research activities)
- For some time, I had my public gateway published in the official list…

…this was the OK story because soon enough…

But no story is so revolting as a group that was publishing videos with cats being tortured on IPFS, most likely providing links to “clients” that pay to access that content.

Story time

- I have been operating an IPFS public gateway from home for years



Cat Torture Network Using Joaoleitao IPFS Gateway

From ForAllCats@proton.me on 2023-12-27 23:06

Details Headers Plain text

Hello,

I'm writing on behalf of a group of volunteers investigating a cat torture network to inform you that this network is seemingly using your ipfs gateway service to post extreme torture video content on their main site.

The videos can be found on Jacklatiao.net, but I will include all direct links to the torture videos below. Please note the videos are extremely troubling and are all of live animals, mostly kittens.
They show torturers twisting/cutting limbs off, repeated electrocution, force feeding acid, cutting out tongue/eyes, live dissection, burning, baking, violent rape, deep frying and many more sadistic methods.

Please take action to remove this content from your gateway.

https://ipfs.joaoleitao.org/ipfs/QmX6rYKWpabuq3DXqSiR1bbqWMe9jmKmtwtgBKFJSZ5k1J
https://ipfs.joaoleitao.org/ipfs/QmV9qHmLf1vCW9ZGW8CeAq3RUjNRH2VzCa9beYUbZ8xSph
https://ipfs.joaoleitao.org/ipfs/QmNsG6fkXKvkw6UybWDAdMtPrBeh7yZ932uj5nydBkLmBw
https://ipfs.joaoleitao.org/ipfs/QmNNH9L3e5Z6MLpdcreBgCRT9Mon9CAZaWntyrTmn8WypD
https://ipfs.joaoleitao.org/ipfs/QmRBL5YDM7kccTYKAnQpG1dctmrjxBz6UAL6Mrpz3xbZsv
https://ipfs.joaoleitao.org/ipfs/QmYxtvTrfhka9HwWnjkFe2FwwSCxvNysSZhtG2CyjAZR2S
https://ipfs.joaoleitao.org/ipfs/QmYxpT6tpnPoE6GMkP2ihbSyFtDsuKokVdqzRit26j9DVj
https://ipfs.joaoleitao.org/ipfs/QmQcPnVUz89g8VQsvoG7vnLFo6oeiV7TFzBxNgRm1fm2eH
https://ipfs.joaoleitao.org/ipfs/Qmf1gkhQ1k3stDvHSgBE6za3Hcq6mVumaYqZApzbzerQg7
https://ipfs.joaoleitao.org/ipfs/QmPJvPdP1atBBpo7HL6zrjG5jnkR4KV7crwzVDR8NiXohZ
https://ipfs.joaoleitao.org/ipfs/QmZLBV3yMizDh5S1y2DP55bx4YhcrgCmz76a5nQr9SSoQa

Takeaway from Story time

- Decentralized systems are a good way to ensure the right of free speech and defeat censorship on the Internet

- As always something that has the potential for good can be used for bad purposes.

Takeaway from Story time

- Decentralized systems are a good way to ensure the right of free speech and defeat censorship on the Internet

- As always something that has the potential for good can be used for bad purposes.

- IPFS has a module where a blacklist of CIDs can be imported from from Protocol Labs and that makes a IPFS instante to refuse to serve those contents.
- Evidently questions must be asked: Who controls this mechanism? Isn’t this mechanism something that can be used for censorship?

IPFS: Impact beyond IPFS

- IPFS led to the creation of Libp2p

- Libp2p has become a touch stone for innovation and development within the Web3 and decentralized world:

IPFS: Impact beyond IPFS

- IPFS led to the creation of Libp2p

- Libp2p has become a touch stone for innovation and development within the Web3 and decentralized world:

  - Many companies building products on top of Libp2p (e.g., Berty, a mobile app for group chats specifically tailored for anonymous and decentralized activism).
  - Libp2p has been used in many decentralized apps (including blockchain ecosystem)
  - Libp2p has been recently integrated as part of the second version of Ethereum (as far as I understand mostly for the use of GossipSub protocol).

Roadmap of this Talk

- Overview of IPFS (and Libp2p)
- **What is it that we know about IPFS on the wild**
- What are pressure points in IPFS and Libp2p
- Beyond IPFS and Libp2p on Decentralized Systems
- Takeaway Points

Measuring a Decentralized System on the Wild

- An interesting thing about having a fully decentralized system being adopted by a large community is that its hard to understand what is going on with the system.

Measuring a Decentralized System on the Wild

- An interesting thing about having a fully decentralized system being adopted by a large community is that its hard to understand what is going on with the system.
    - There is no central authority that tracks everything that happens in the system…

Measuring a Decentralized System on the Wild

- An interesting thing about having a fully decentralized system being adopted by a large community is that its hard to understand what is going on with the system.

  - There is no central authority that tracks everything that happens in the system…
  - …consequently, there is no observation point to obtain this information.

Measuring a Decentralized System on the Wild

- An interesting thing about having a fully decentralized system being adopted by a large community is that its hard to understand what is going on with the system.
  - There is no central authority that tracks everything that happens in the system…
  - …consequently, there is no observation point to obtain this information.
- Realistic options:
  - Passive measurements: Spinning nodes observe what they observe
  - Active measurements: Spinning nodes that interact actively with the system

Measuring a Decentralized System on the Wild

- An interesting thing about having a full decentralized system being adopted by a large community is that its hard to understand what is going on with the system.
  - There is no central authority that tracks everything that happens in the system…
  - …consequently, there is no observation point to obtain this information.
- Realistic options:
  - ~~Passive measurements: Spinning nodes observe what they observe~~
  - Active measurements: Spinning nodes that interact actively with the system

Active Measurement: Crawling the DHT

- My friend Dennis Trautwein has come up an idea to crawl the IPFS network through the DHT (remember that Kademlia nodes will reply with the 20 closest nodes that they are aware for a requested identifier). PL runs continually the **Nebula crawler** to keep an up-to-date vision of the system.

Protocol Labs
Research

- This allowed us to gather lots of information about the state of IPFS over time correlating with information that nodes expose (version of software) and other data sources (location of nodes, which nodes are in data centers, uptime of nodes, etc...)

Active Measurement: Crawling the DHT

- In 2022-23 what we knew about IPFS:
  - 150K+ nodes (many of them never reachable, potentially due to NATs)
  - 20K nodes acting as servers (mostly available)
  - Clients scattered across 152 countries (higher concentration on US, Germany)
  - Nodes executing in more that 2700 Autonomous systems

  - Only a small fraction of nodes (<20%) operating on cloud infrastructures.

Dennis Trautwein, Aravindh Raman, Gareth Tyson, Ignacio Castro, Will Scott, Moritz Schubotz, Bela Gipp, and Yiannis Psaras. **Design and evaluation of IPFS: a storage layer for the decentralized web.** In Proceedings of the ACM SIGCOMM 2022 Conference (SIGCOMM '22). Association for Computing Machinery, New York, NY, USA, 739–752. https://doi.org/10.1145/3544216.3544232

Active Measurement: Crawling the DHT

- In 2025 what we know about IPFS:
  - 300K+ nodes (many of hard to contact)
  - 46K nodes acting as servers (but 34K offline at least 90% of time)

https://probelab.io/ipfs/kpi/

Active Measurement: Crawling the DHT

- In 2025 what we know about IPFS:
  - 300K+ nodes (many of hard to contact)
  - 46K nodes acting as servers (but 34K offline at least 90% of time)

- Interestingly, the total number of unique nodes that can be found increased (including those acting as servers) but the reliable active servers have somewhat dropped (from close to 20K to around 12K).

https://probelab.io/ipfs/kpi/

How performant is IPFS (at the DHT Level)

- To understand this, we need to do an active measurement where we interact with the DHT.
- We set up a measurement campaign for close to 8 months from March 22 to October 22 where we setup 5 agents (on cloud infrastructures) throughout the World.

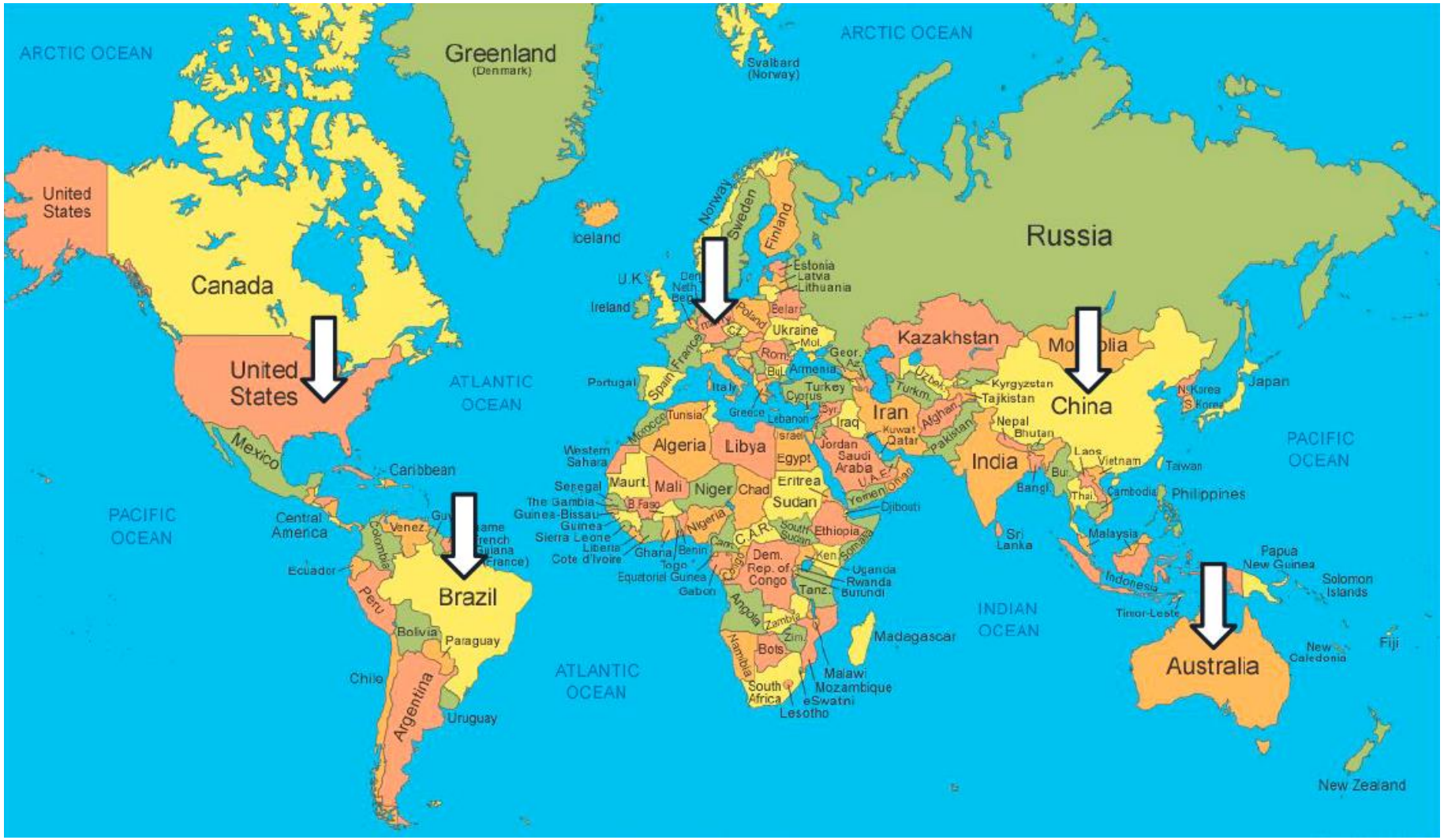
How performant is IPFS (at the DHT Level)

- To understand this, we need to do an active measurement where we interact with the DHT.
- We set up a measurement campaign for close to 8 months from March 22 to October 22 where we setup 5 agents (on cloud infrastructures) throughout the World.

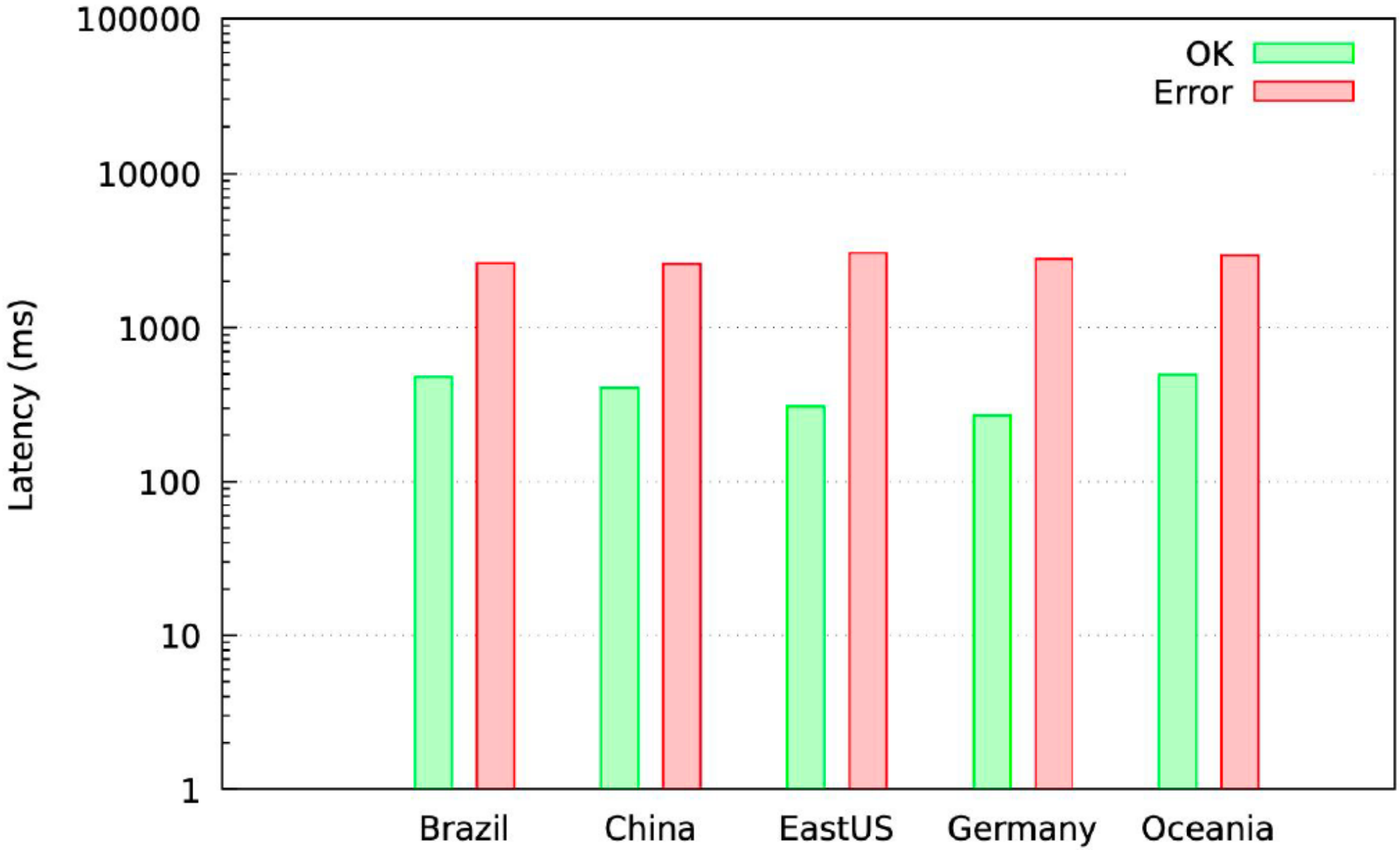
Each agent:

- Publish 1/5 of 265 keys generated by us to the DHT.
- Execute continuous lookups on the DHTs for the keys published by the other agents.
- We looked at the success rate of these lookups and latency of lookup and individual RPCs.

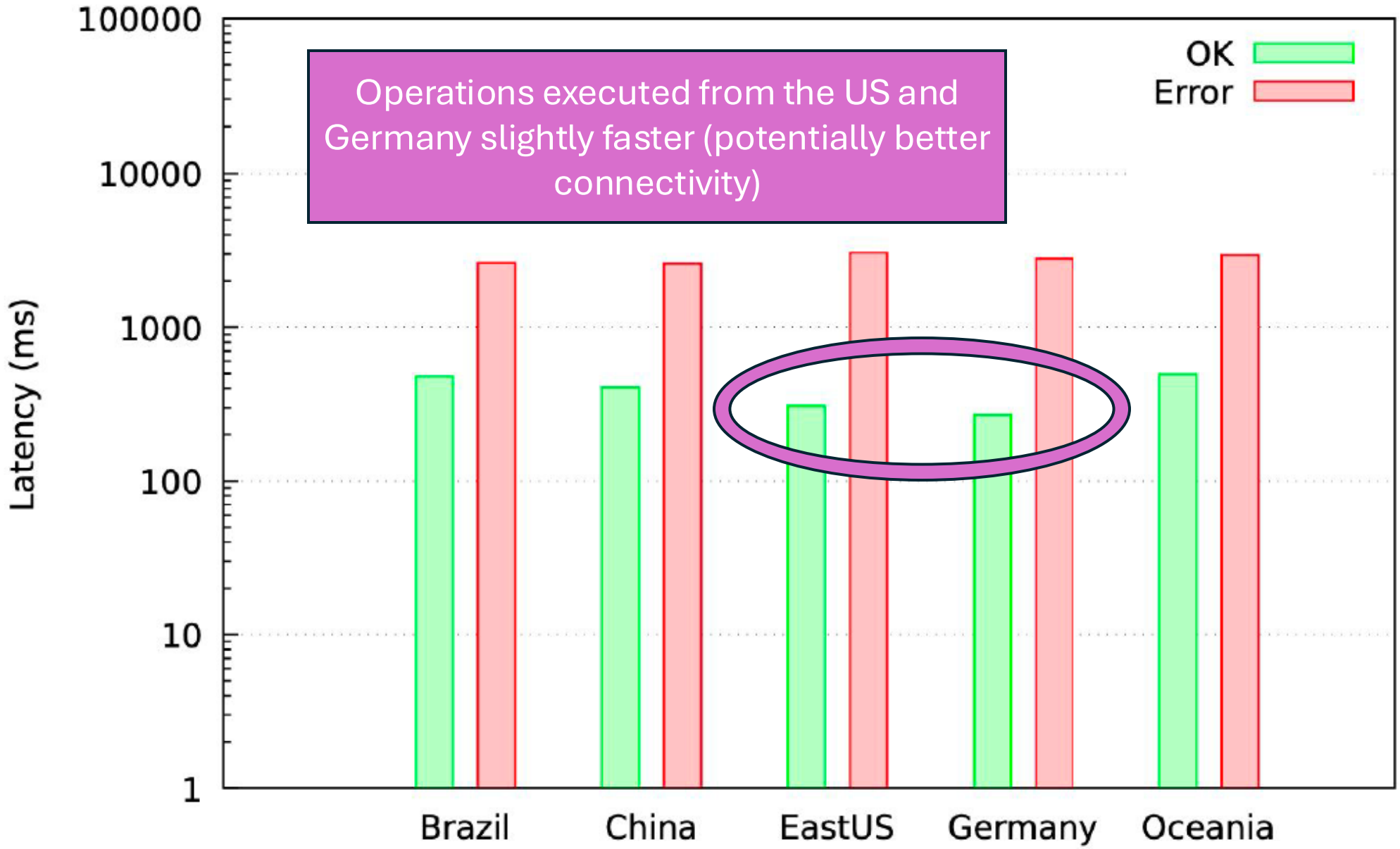
How performant is IPFS (at the DHT Level)



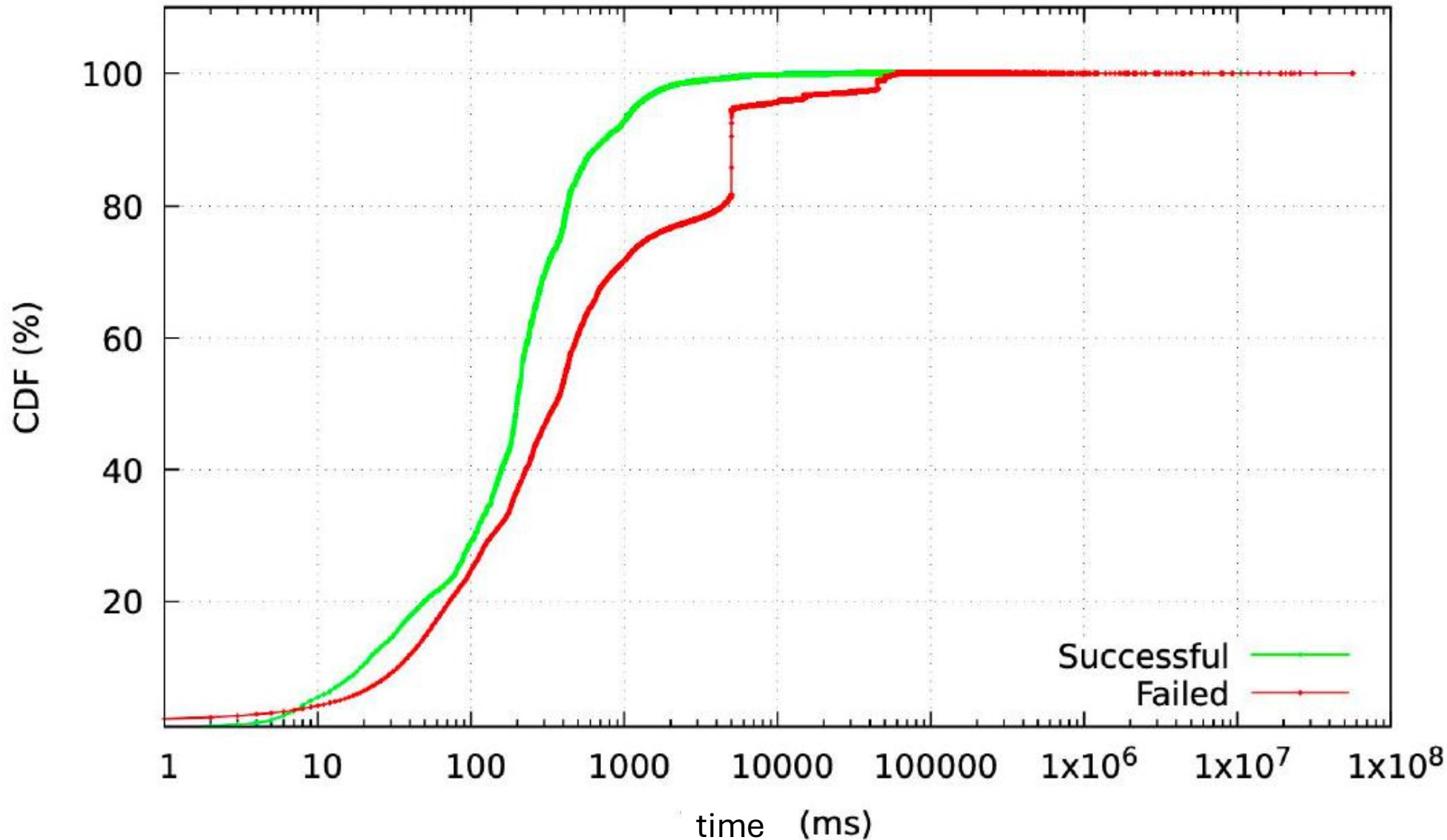
DHT Performance: Average Latency



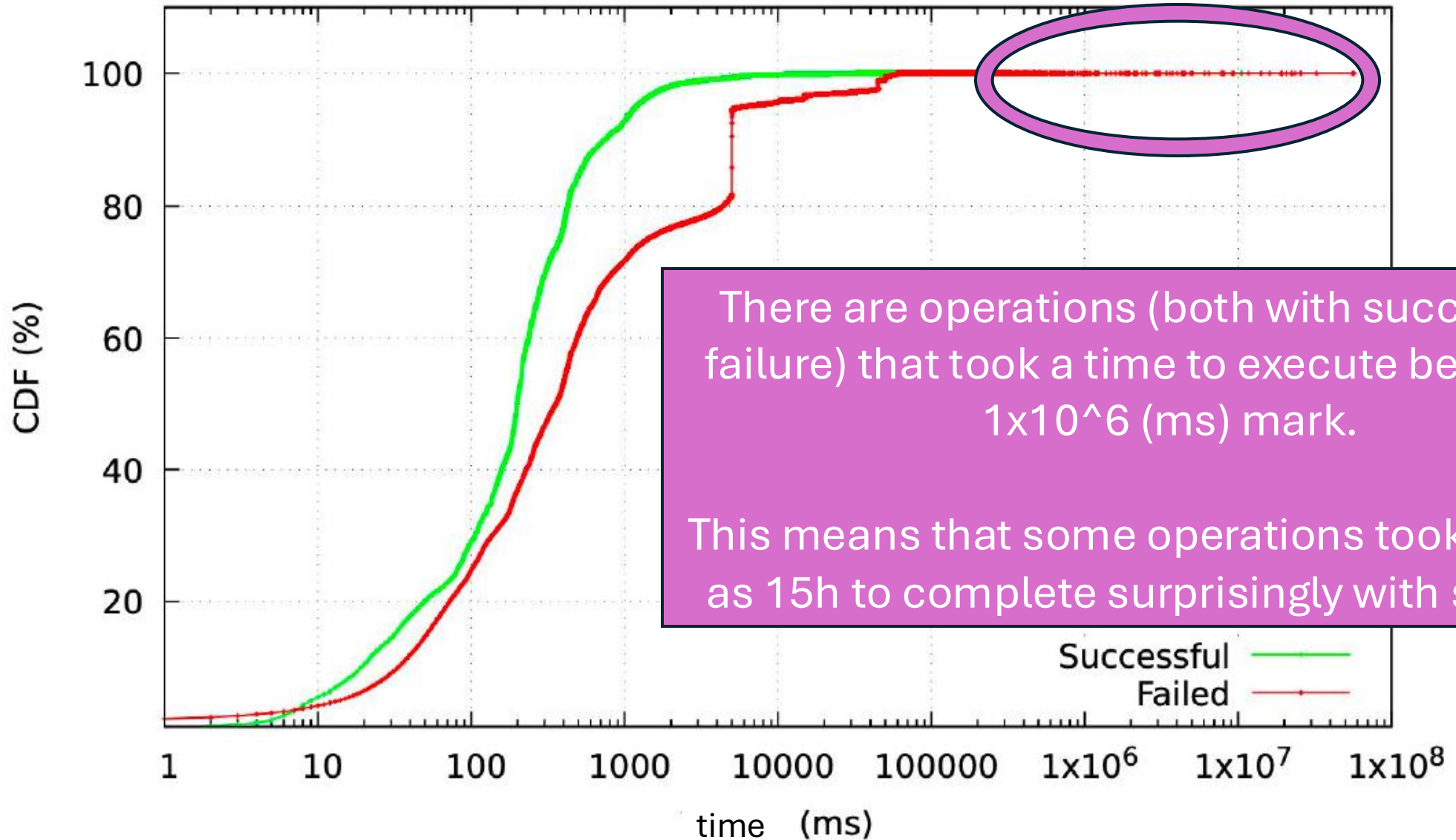
DHT Performance: Average Latency



DHT Performance: Latency CDF



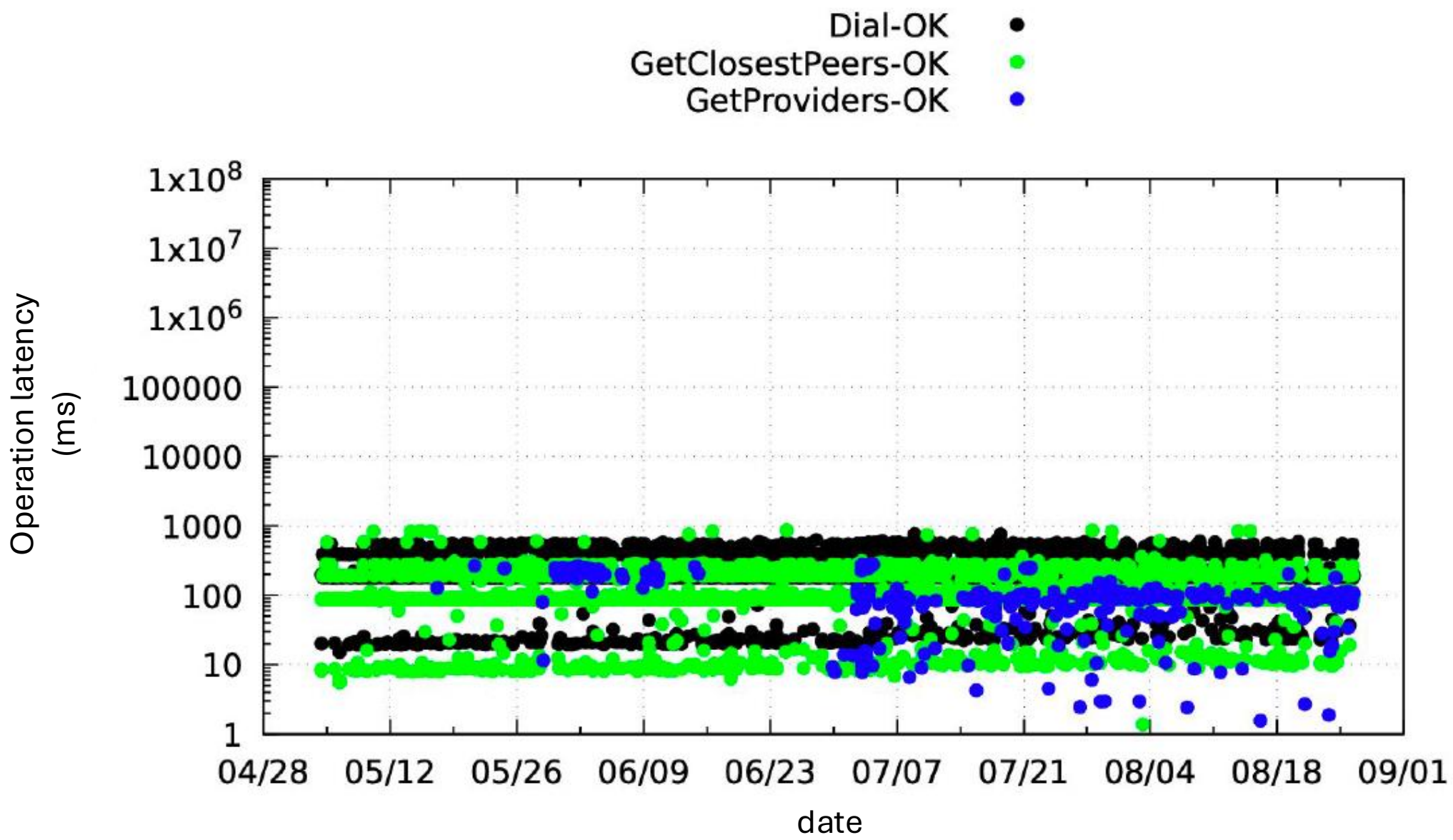
DHT Performance: Latency CDF



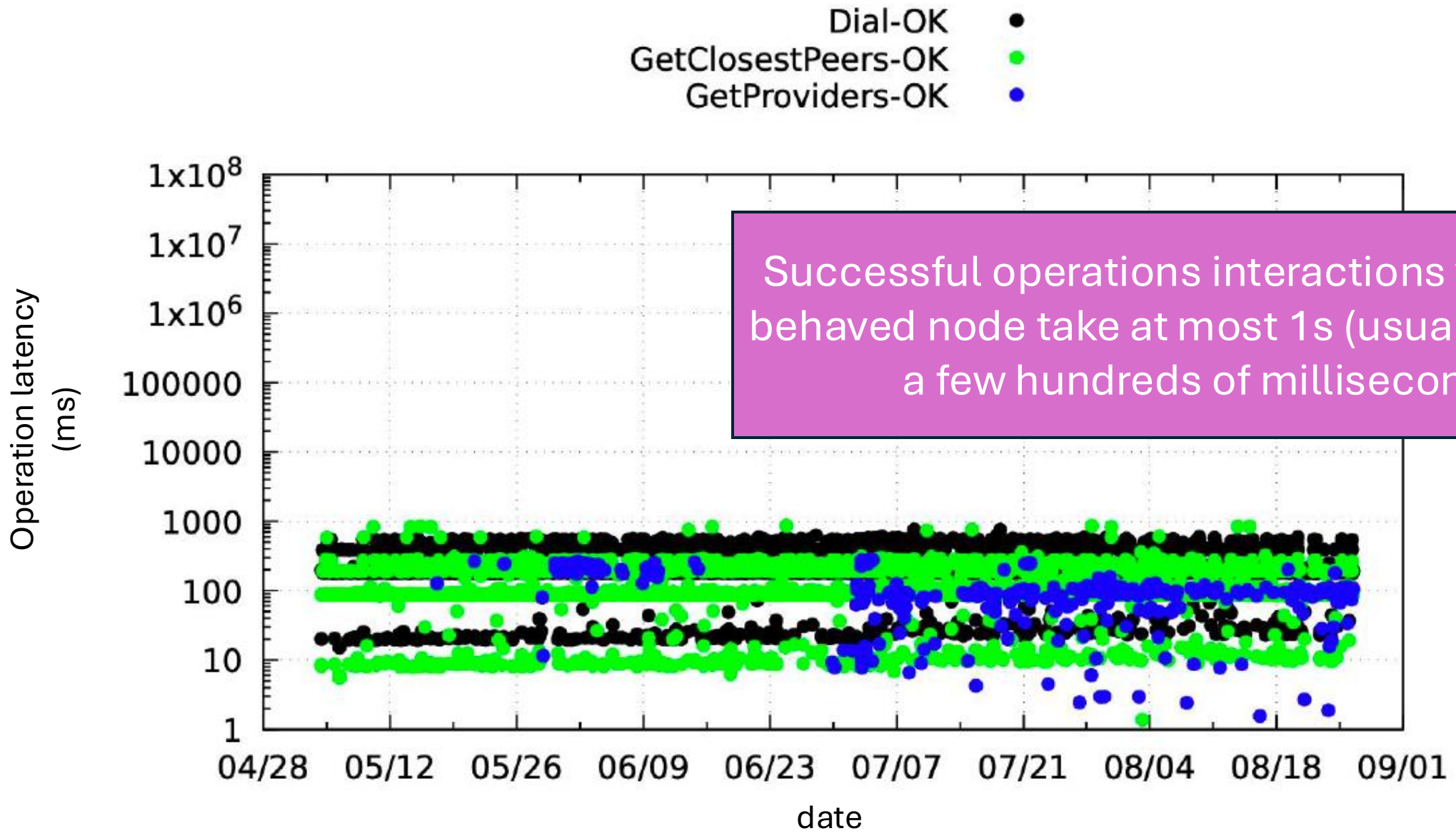
There are operations (both with success and failure) that took a time to execute beyond the 1x10^6 (ms) mark.

This means that some operations took as much as 15h to complete surprisingly with success.

DHT Performance: Behavior of a normal node

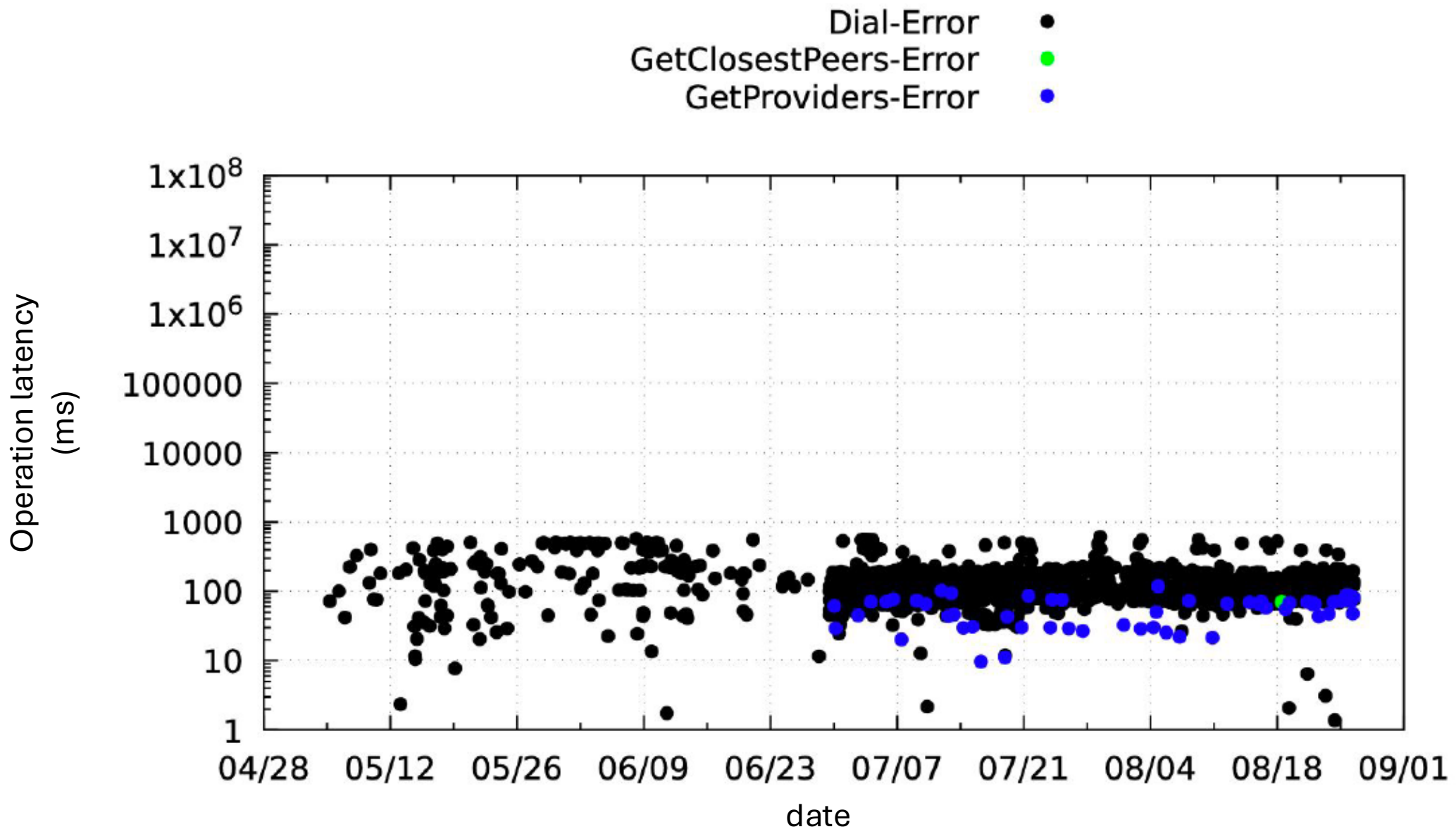


DHT Performance: Behavior of a normal node

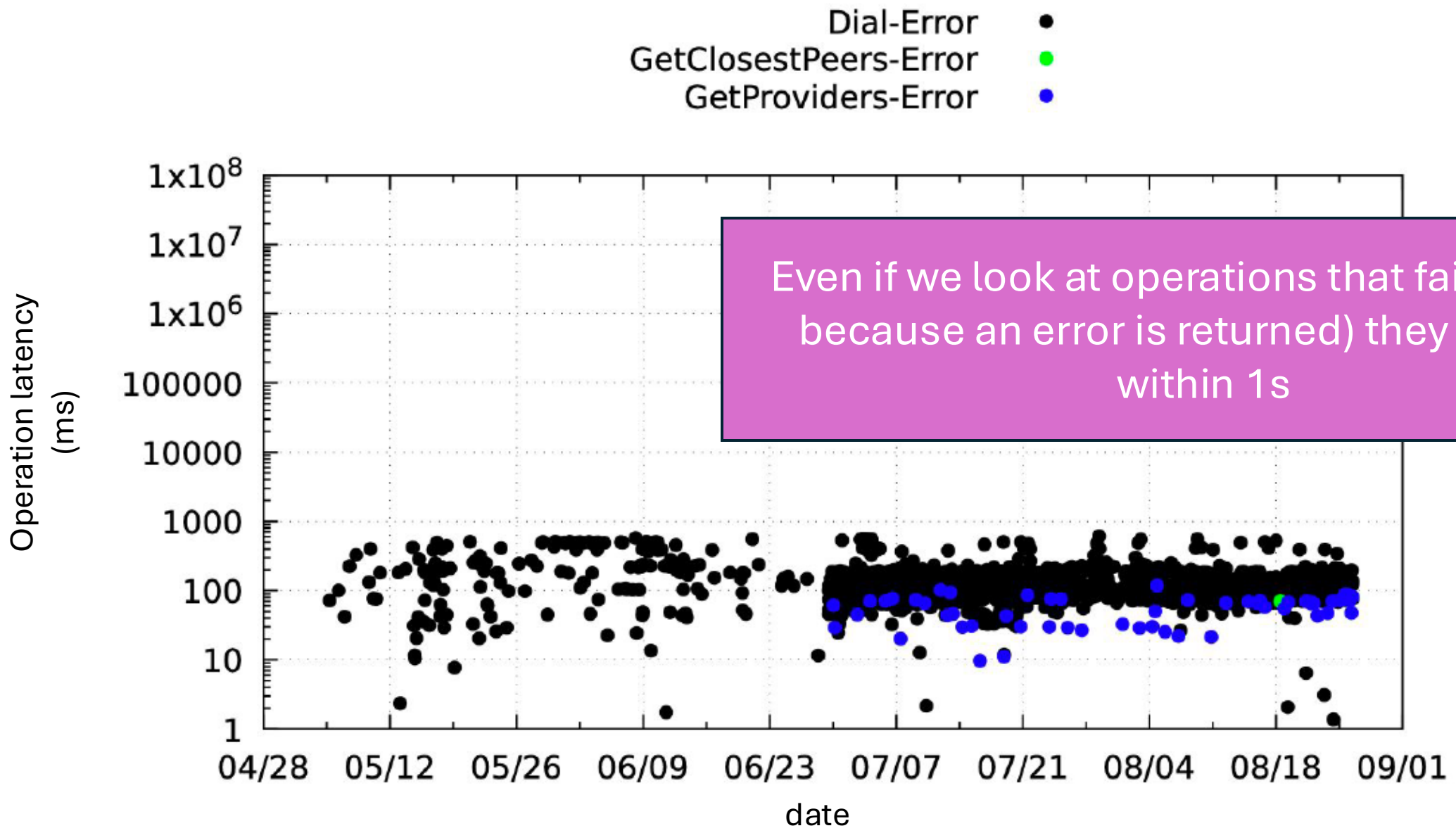


Successful operations interactions with a well-behaved node take at most 1s (usually less, only a few hundreds of milliseconds).

DHT Performance: Behavior of a normal node

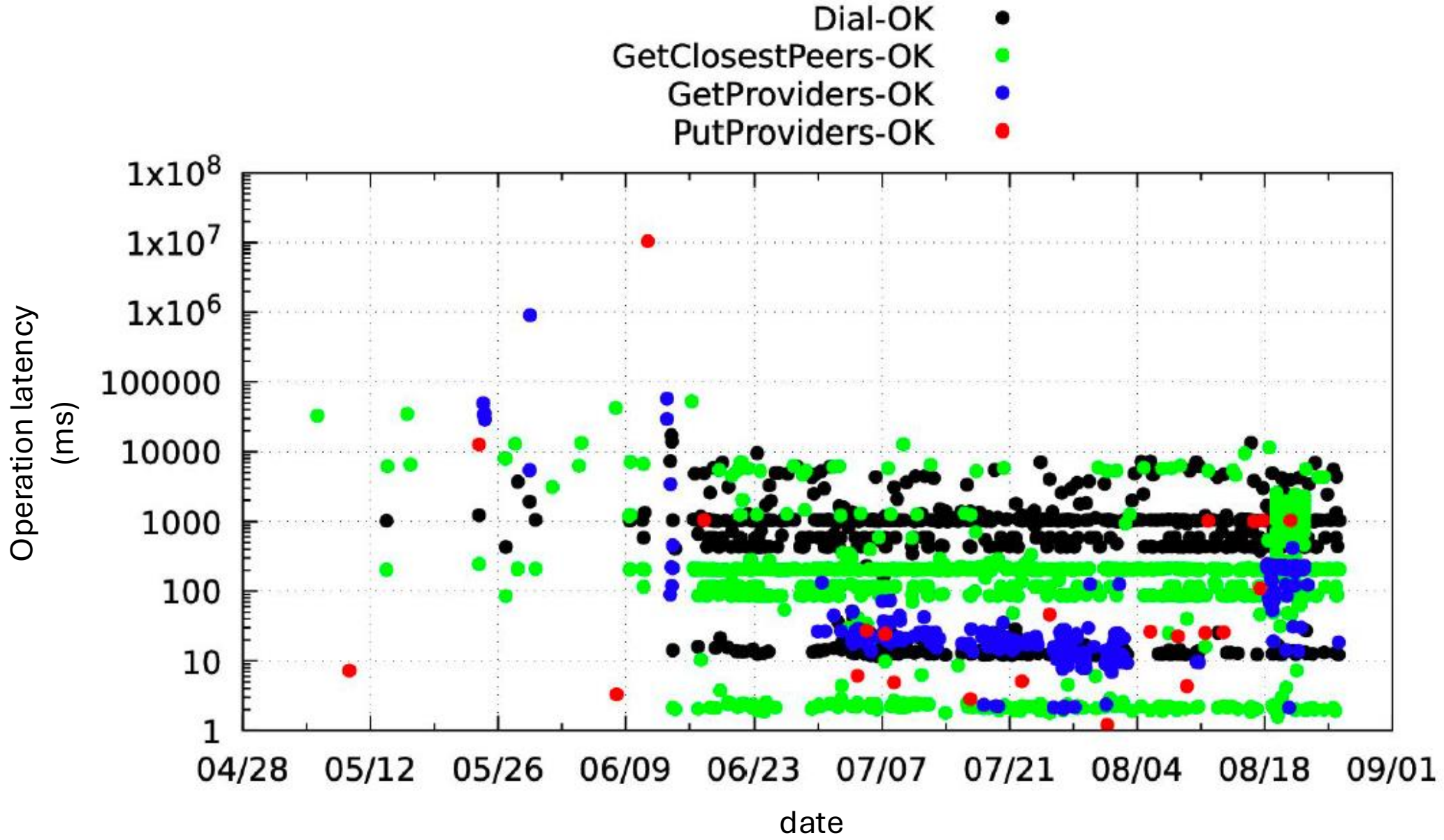


DHT Performance: Behavior of a normal node

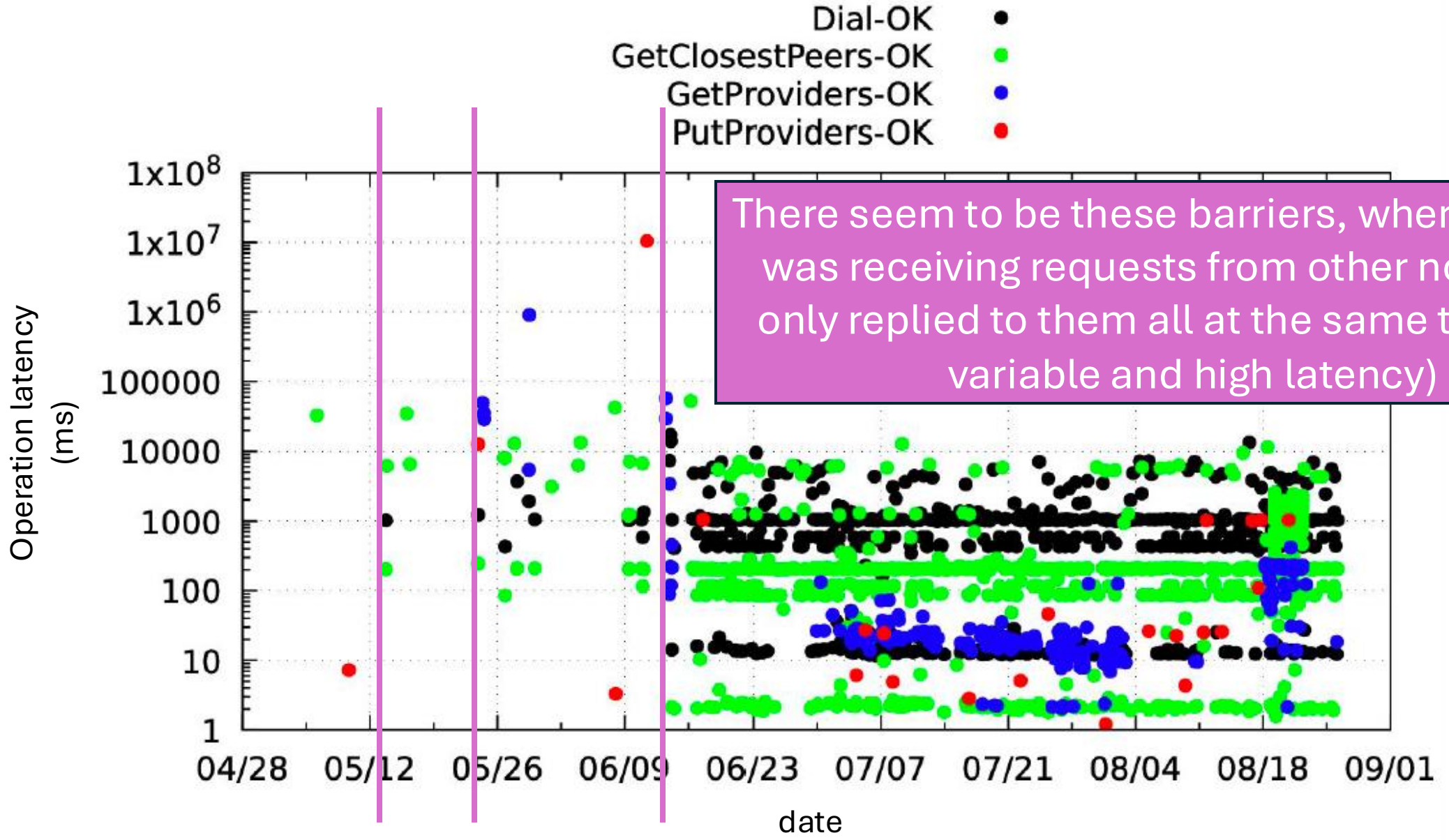


Even if we look at operations that fail (usually because an error is returned) they are well within 1s

DHT Performance: Behavior of a “slow” node

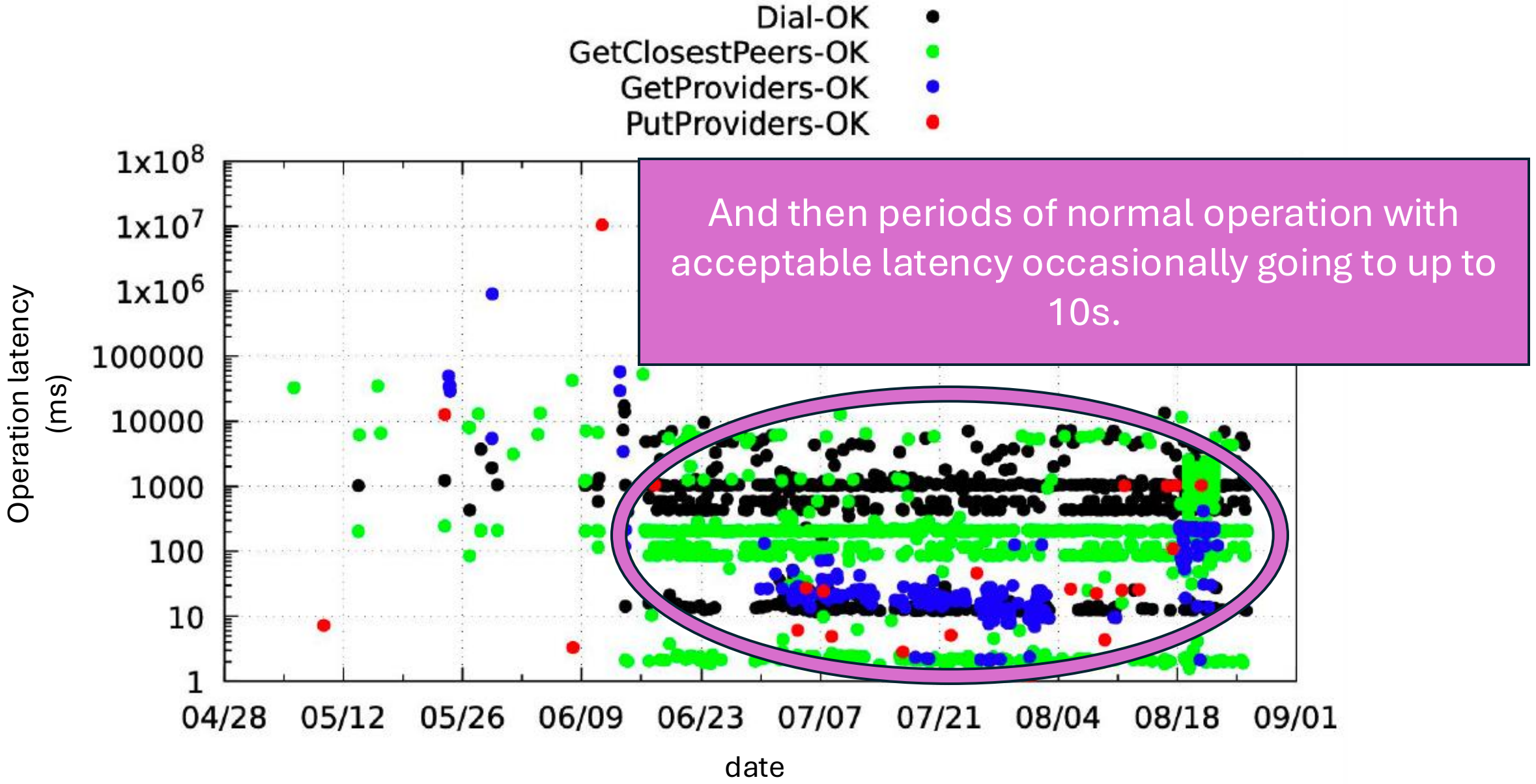


DHT Performance: Behavior of a “slow” node



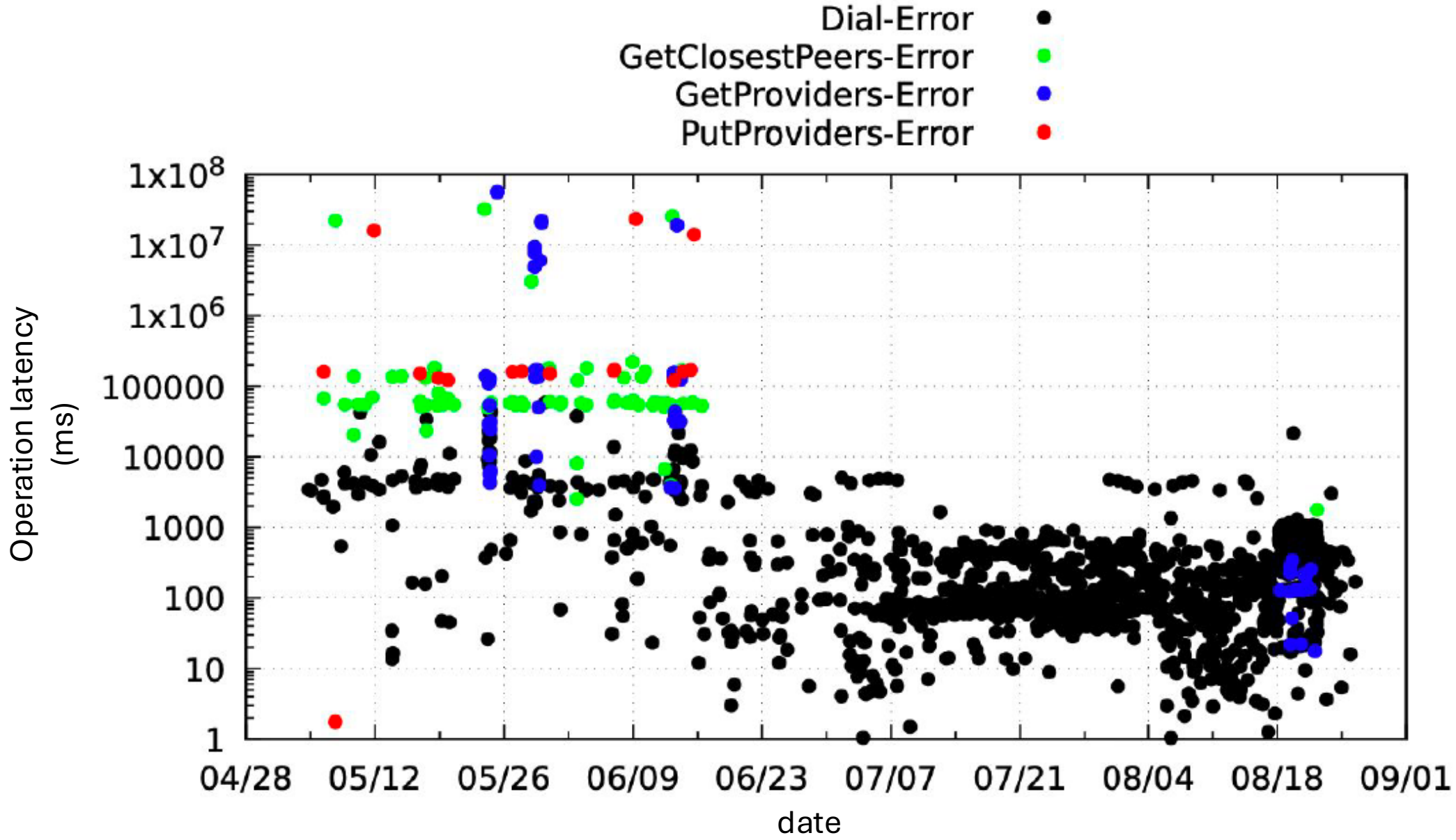
There seem to be these barriers, where the node was receiving requests from other nodes, but only replied to them all at the same time (with variable and high latency)

DHT Performance: Behavior of a “slow” node

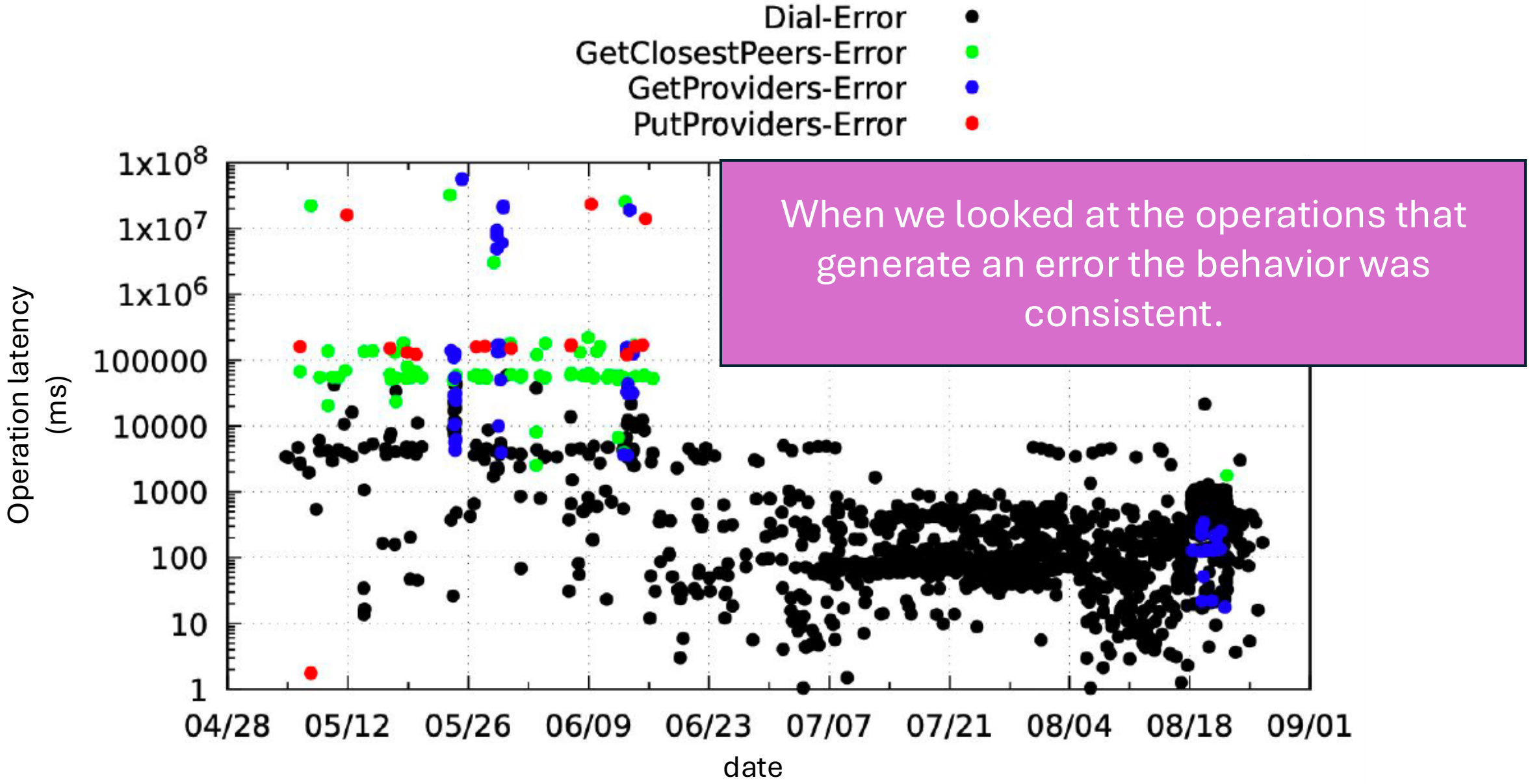


And then periods of normal operation with acceptable latency occasionally going to up to 10s.

DHT Performance: Behavior of a “slow” node



DHT Performance: Behavior of a “slow” node



When we looked at the operations that generate an error the behavior was consistent.

How performant is IPFS (at the DHT Level)

- In general, the DHT in IPFS is quite good. Most operations have success, and most interactions take a few hundreds of milliseconds to complete (interactions here means an RPC to a remote node).

- There are however anomalies, some nodes exhibit weird behaviors where they seem to occasionally to block:

How performant is IPFS (at the DHT Level)

- In general, the DHT in IPFS is quite good. Most operations have success, and most interactions take a few hundreds of milliseconds to complete (interactions here means an RPC to a remote node).

- There are however anomalies, some nodes exhibit weird behaviors where they seem to occasionally to block:
  - They could be byzantine nodes (but it did not look like it)
  - They could be trashing (but we could never validate this hypothesis)
  - At some point we considered that they could be too popular in the DHT (it was not true ☹)
  - It could be miss-configured nodes (but that does not explain the variable behavior)

What is the workload of IPFS

- A relevant question, now that we are aware of the size of the system and the performance of the DHT, which is a key component, is understanding the workload to which IPFS is subjected to.

What is the workload of IPFS

- A relevant question, now that we are aware of the size of the system and the performance of the DHT, which is a key component, is understanding the workload to which IPFS is subjected to.

- The fact that we have a fully decentralized system makes this very hard to estimate…

What is the workload of IPFS

- A relevant question, now that we are aware of the size of the system and the performance of the DHT, which is a key component, is understanding the workload to which IPFS is subjected to.

- The fact that we have a fully decentralized system makes this very hard to estimate… …except that we have the gateways (which are a popular mechanism to access IPFS, at least for common users).

What is the workload of IPFS

- We have conducted a measurement study where we used the logs of the ipfs.io gateway to study several aspects of the IPFS workload.

- What is the number of requests observed by a gateway
- Where are those requests being generated from
- What is the locality of requests (considering the location of providers)

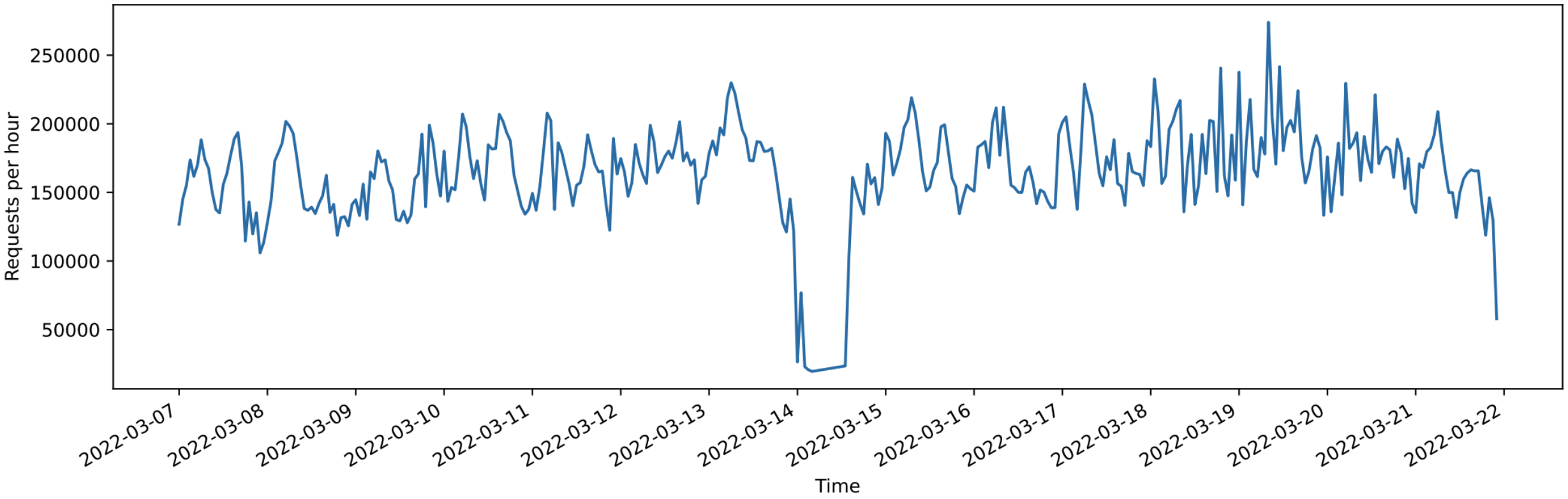
What is the workload of IPFS

- We have conducted a measurement study where we used the logs of the ipfs.io gateway to study several aspects of the IPFS workload.

- What is the number of requests observed by a gateway
- Where are those requests being generated from
- What is the locality of requests (considering the location of providers)
- Other aspects…

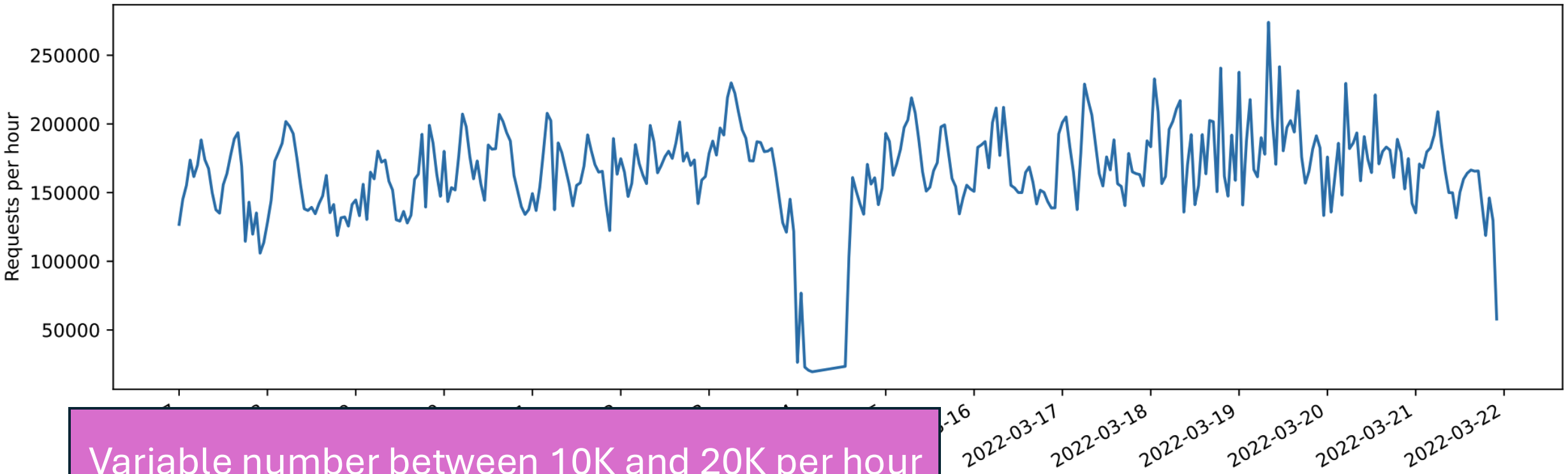
Pedro Ákos Costa, João Leitão, and Yannis Psaras. **Studying the Workload of a Fully Decentralized Web3 System: IPFS**. In Distributed Applications and Interoperable Systems: 23rd IFIP WG 6.1 International Conference, DAIS 2023, Held as Part of the 18th International Federated Conference on Distributed Computing Techniques, DisCoTec 2023. https://doi.org/10.1007/978-3-031-35260-7_2

What is the workload of IPFS



Global number of requests received by Gateway ipfs.io

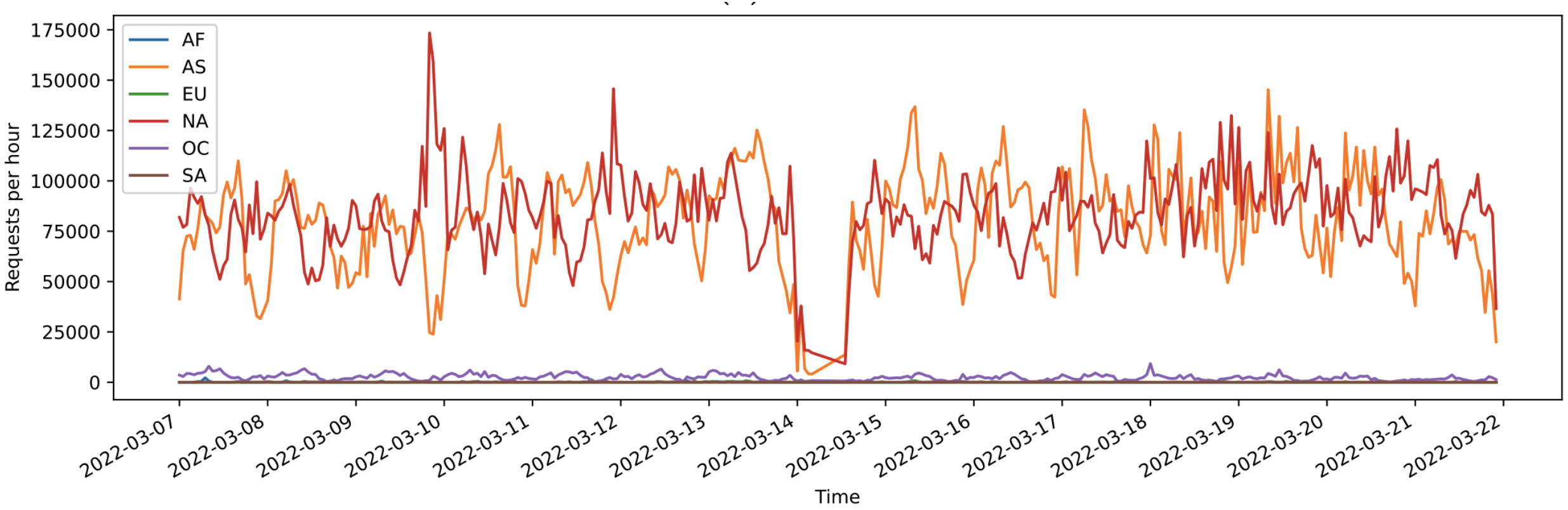
What is the workload of IPFS



Variable number between 10K and 20K per hour (remember that his is a single gateway of many available at the time)

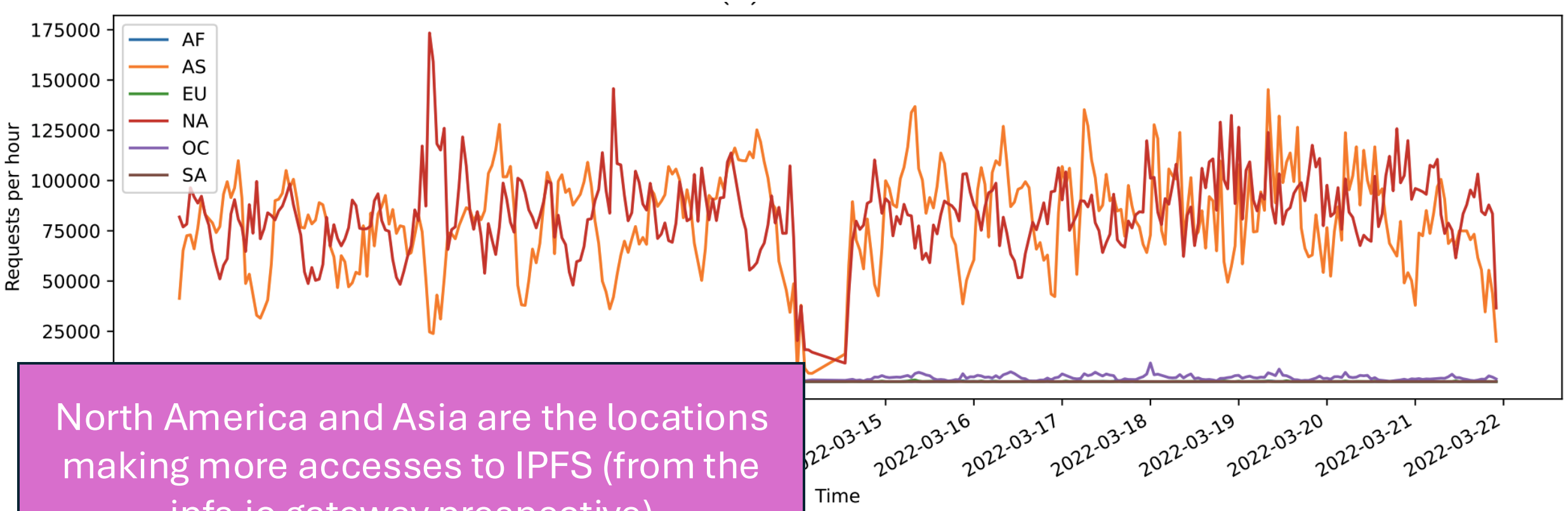
ed by Gateway ipfs.io

What is the workload of IPFS



Global number of requests received by Gateway ipfs.io
(per continent)

What is the workload of IPFS



North America and Asia are the locations making more accesses to IPFS (from the ipfs.io gateway prespective)

Global number of requests received by Gateway ipfs.io (per continent)

What is the workload of IPFS

- The other thing that we wanted to understand was the locality of requests, meaning what is the co-relation between the location of the requester and the location of the content being provided through IPFS.

- We did not know anything about this, although we all had a somewhat expectation that there would be some co-relation.

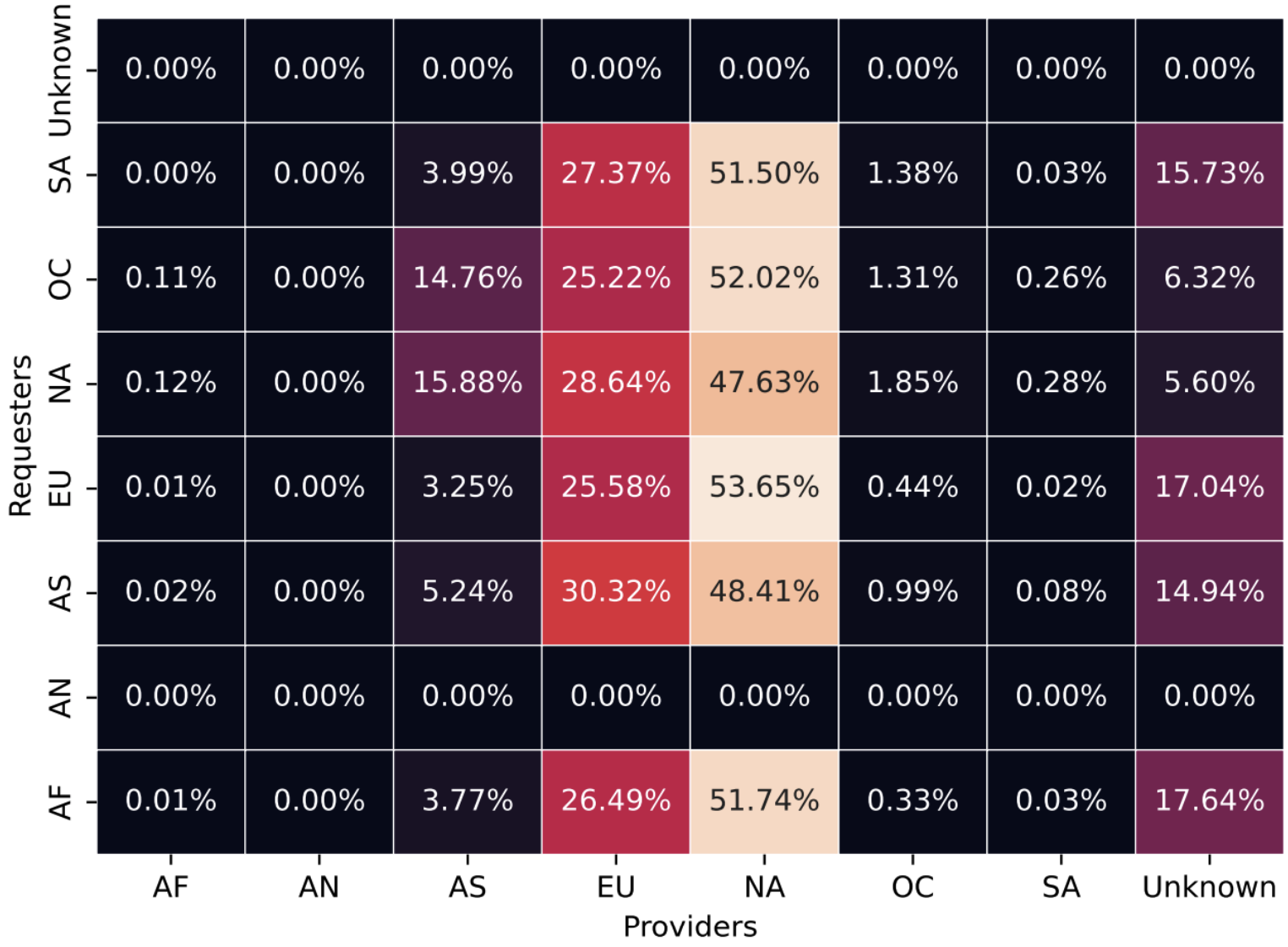
What is the workload of IPFS

- The other thing that we wanted to understand was the locality of requests, meaning what is the co-relation between the location of the requester and the location of the content being provided through IPFS.

- We did not know anything about this, although we all had a somewhat expectation that there would be some co-relation.

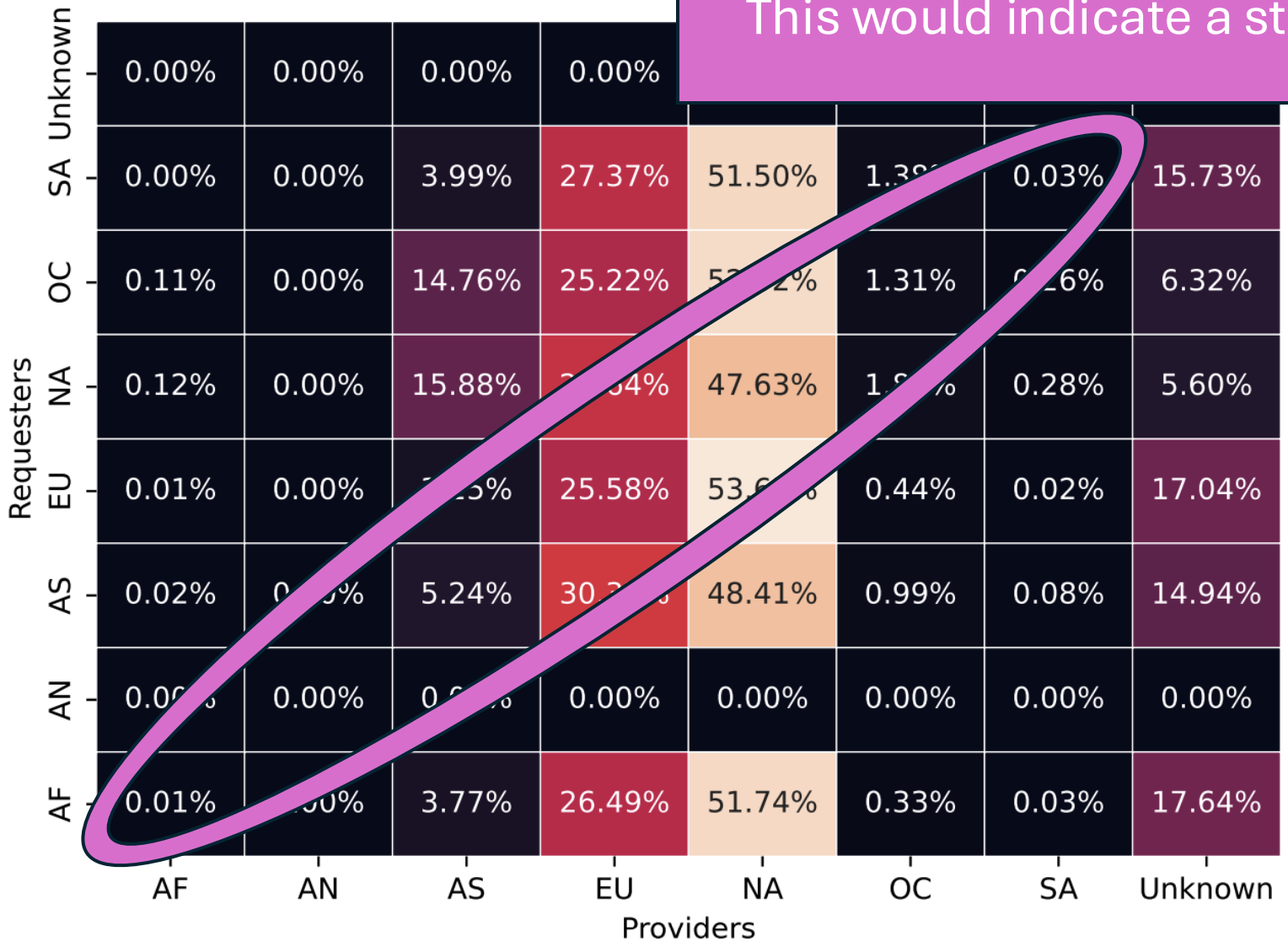
- To do this, we looked at the location of the IP making the request to the gateway, and then we tried to fetch the content provider record for that CID and lookup the location of the IP of the provider.

What is the workload of IPFS

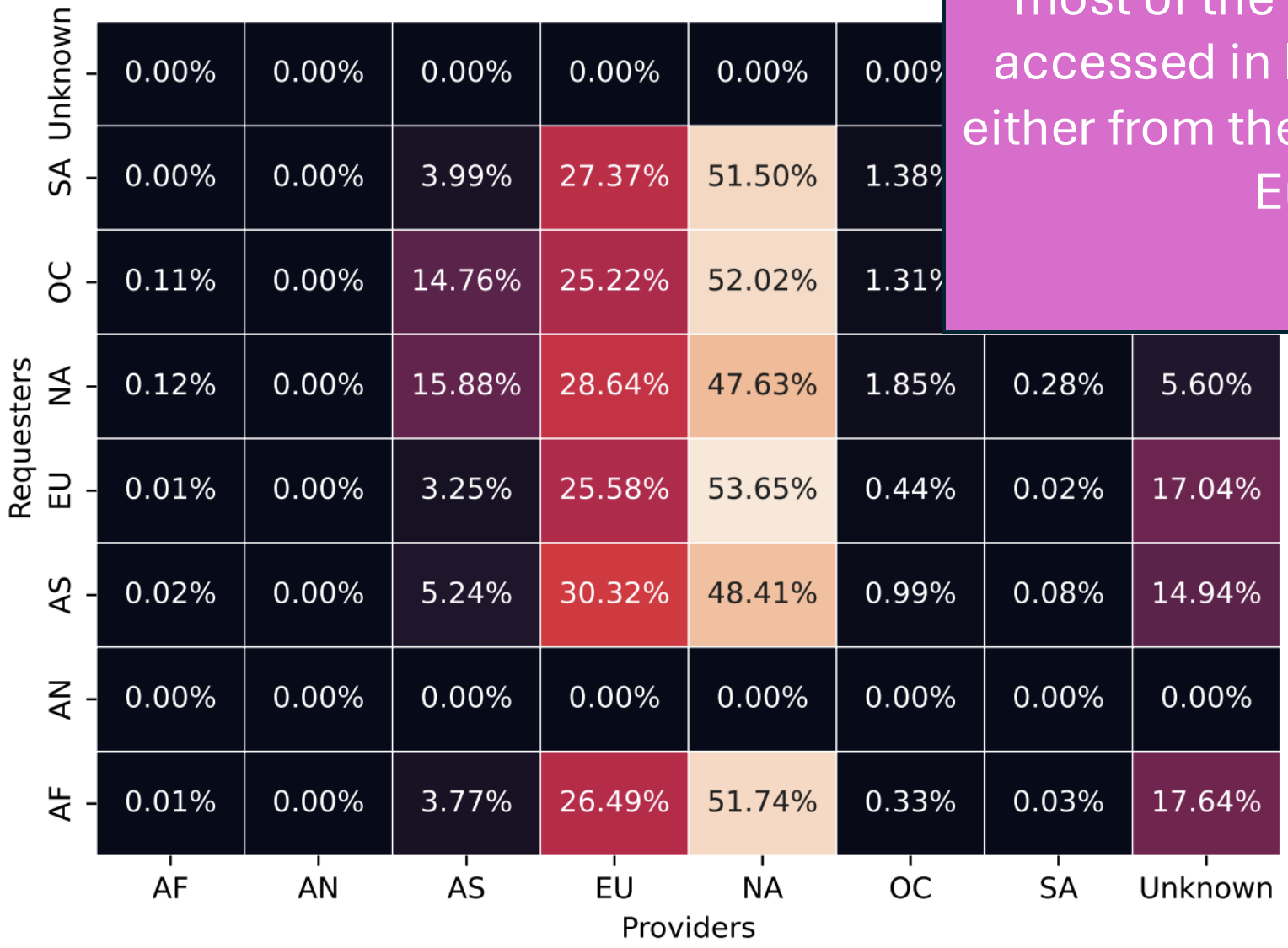


What is the workload of I

This would indicate a strong co-relation



What is the workload of IPFS



In fact, there is no co-relation, most of the popular content accessed in IPFS is published either from the North America or Europe

Roadmap of this Talk

- Overview of IPFS (and Libp2p)
- What is it that we know about IPFS on the wild
- **What are pressure points in IPFS and Libp2p**
- Beyond IPFS and Libp2p on Decentralized Systems
- Takeaway Points

What are the pressure points in IPFS and Libp2p

- I have shown you that the performance is variable and somewhat polluted by anomalies that appear concentrated on specific nodes on the network (at least for the case of IPFS).
- IPFS has a non-uniform access pattern to content (at least at Gateways)

What are the pressure points in IPFS and Libp2p

- I have shown you that the performance is variable and somewhat polluted by anomalies that appear concentrated on specific nodes on the network (at least for the case of IPFS).
- IPFS has a non-uniform access pattern to content (at least at Gateways)

- For libp2p, as a tool to build Web3 applications and decentralized systems, the library is a bit rigid:
  - It depends on the DHT for many operations, even things that should not require a DHT.
  - The structure of the library is quite hard to change (possible to add new protocols, but hard to replace existing protocols)
- Libp2p is now used in Ethereum, mostly motivated by the GossipSub protocol.

What are the pressure points in IPFS and Libp2p

- I have shown you that the performance is variable and somewhat polluted by anomalies that appear concentrated on specific nodes on the network (at least for the case of IPFS).
- IPFS has a non-uniform access pattern to content (at least at Gateways)
- For libp2p, as a tool to … decentralized systems, the library is …
  - It depends on the DHT … hat should not require a DHT.
  - The structure of the libr… ble to add new protocols, but hard to replace exis…
- Libp2p is now used in E…, mostly motivated by the GossipSub protocol.

But how good is really this GossipSub protocol?

Another collaboration with a company…

- Me and my team have started recently to collaborate with a startup in the Web3 domain that want to develop a solution on top of libp2p

- They are targeting decentralized web applications, which puts some requirements to their solution:
  - They want to propagate information through pub-sub using topics.
  - To avoid instabilities in the DTH they don’t want nodes to run the DHT.
  - Since they are targeting browsers, they want to use the libp2p JavaScript implementation.

Another collaboration with a company…

- Me and my team have started recently to collaborate with a startup in the Web3 domain that want to develop a solution on top of libp2p

- They are targeting decentralized web applications, which puts some requirements to their solution:
  - They want to propagate information through pub-sub using topics.
  - To avoid instabilities in the DTH they don’t want nodes to run the DHT.
  - Since they are targeting browsers, they want to use the libp2p JavaScript implementation.

- They requested our help to verify how good was libp2p GossipSub…

Evaluating GossipSub (with no DHT)

- Since we wanted to verify how effective was this solution in a realistic setting, but we did not want to run experiments on hundreds of nodes, we resorted to emulation:

  - Real code running on our (powerful) servers.
  - Code is encapsulated within docker containers.
  - Latency between nodes is emulated using Linux TC.
  - Latency model generated using Bonsai (a network model generator we that have built and validated with real measures from IPFS).

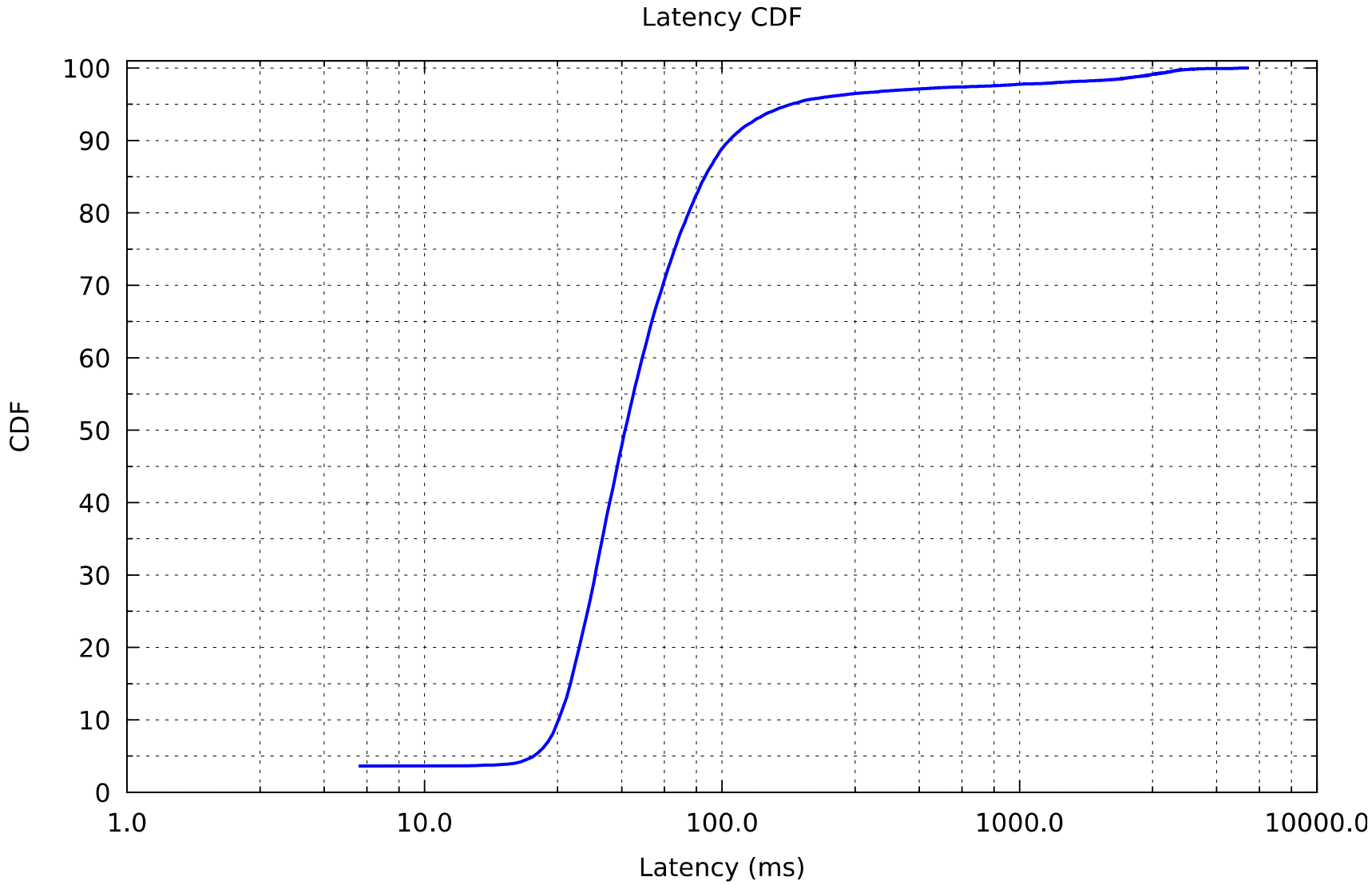
Evaluating GossipSub (with no DHT)

- Our setup was simple, we tested a variable number of nodes, publishing and receiving messages on a single topic.

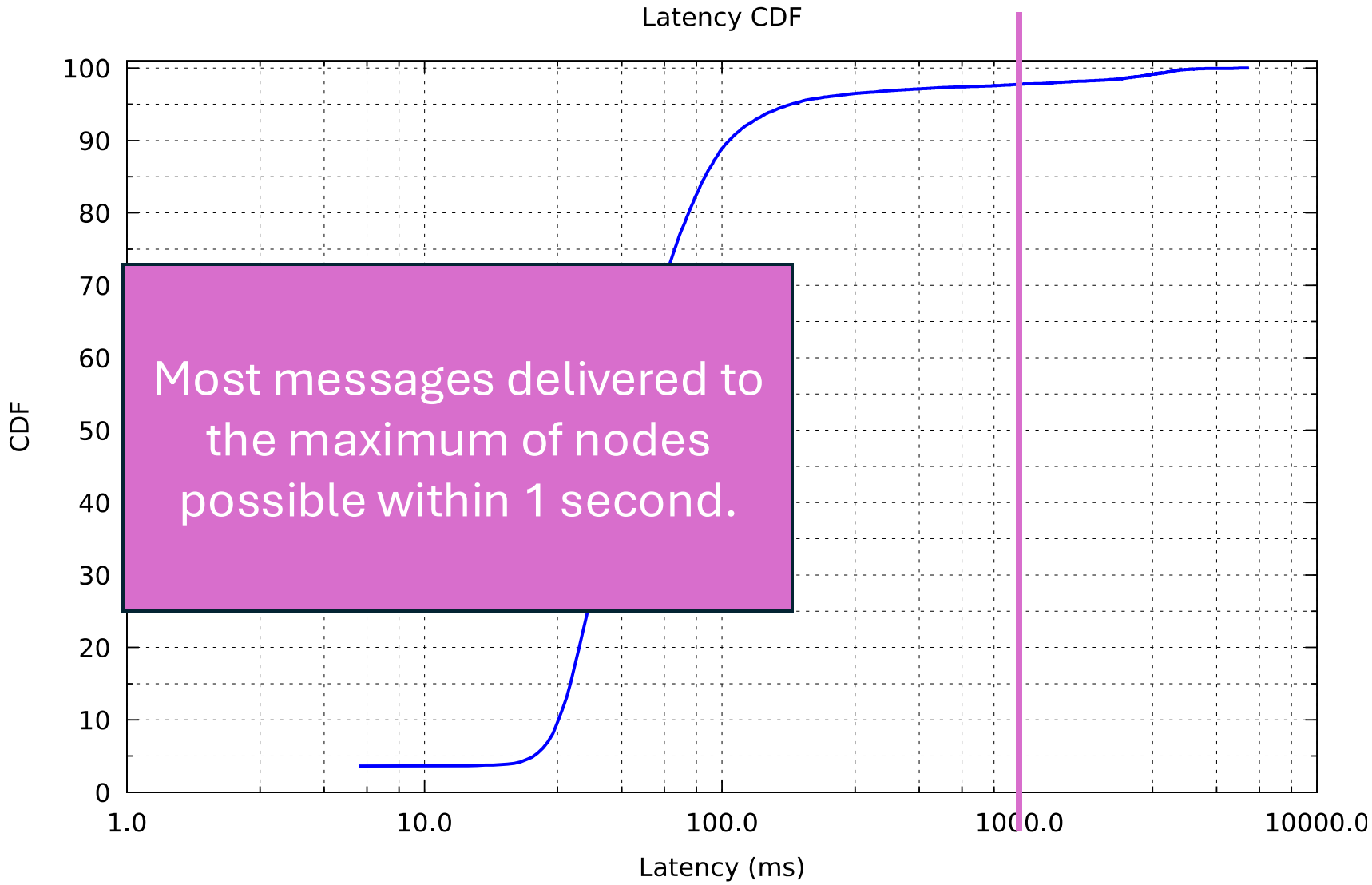
- Every nodes generates messages at the same rate (2 messages per second).

- We measure the reliability (delivery rate) and the maximum latency for messages.

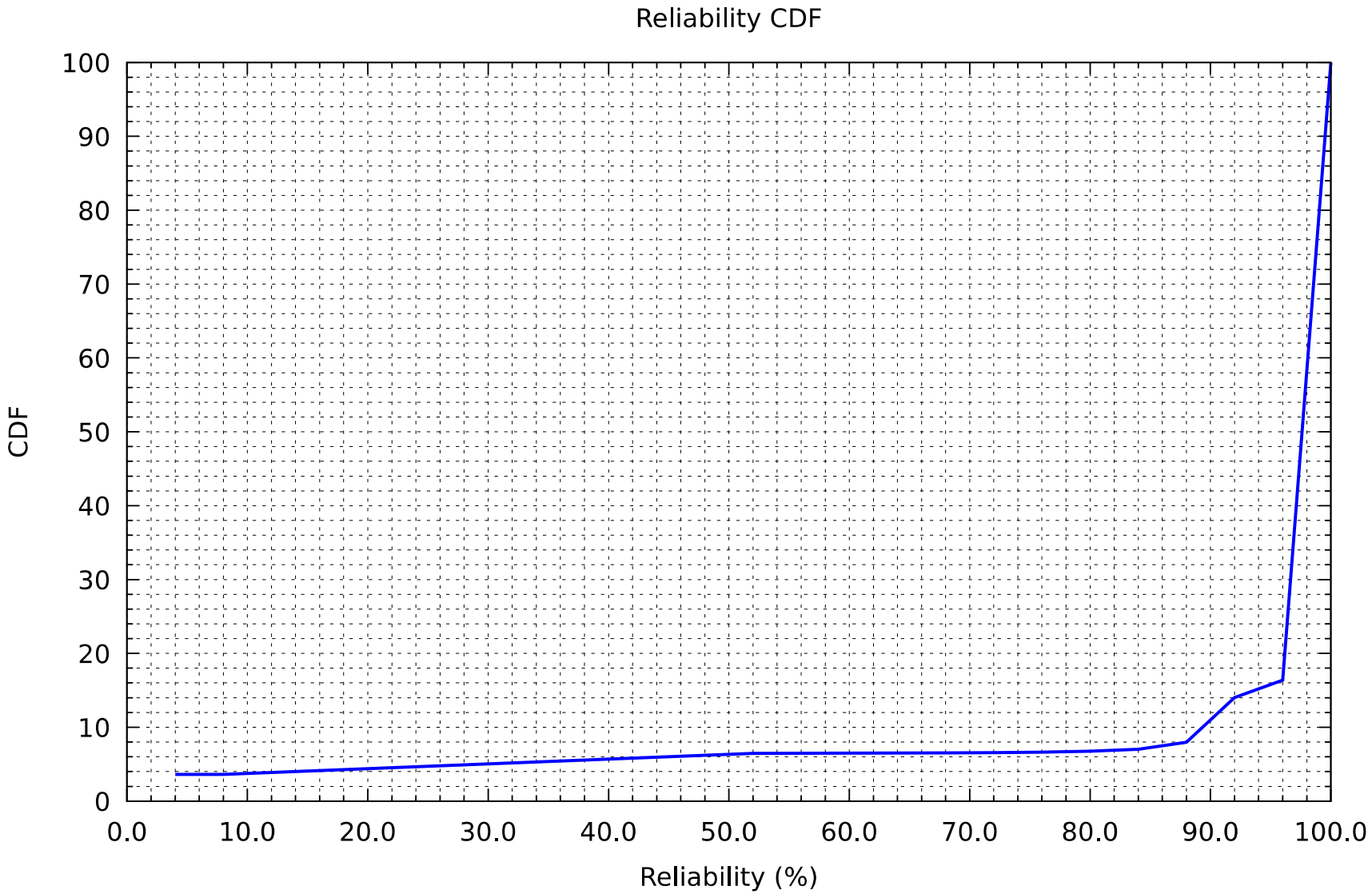
Evaluating GossipSub (with no DHT 25 nodes)



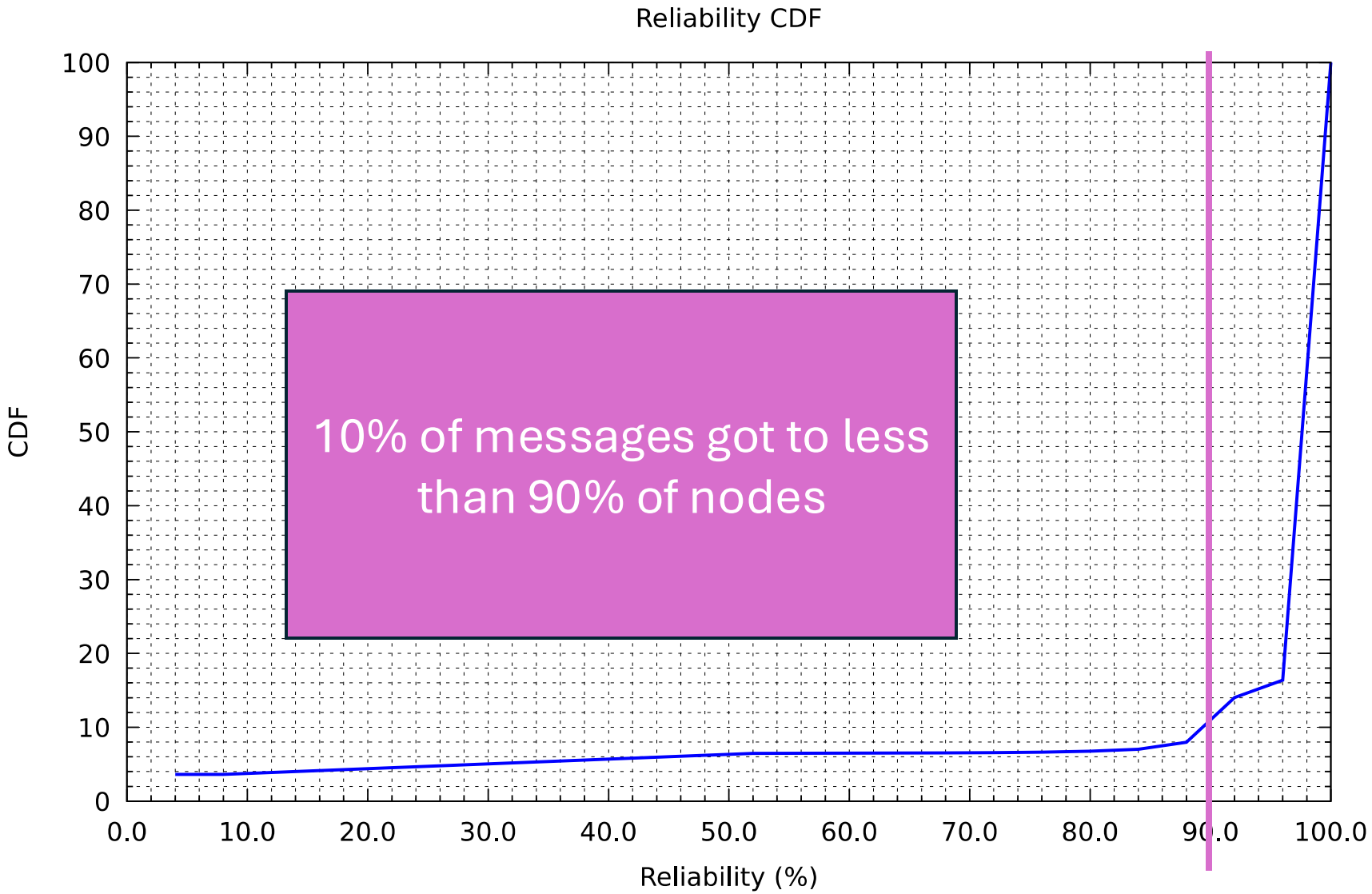
Evaluating GossipSub (with no DHT 25 nodes)



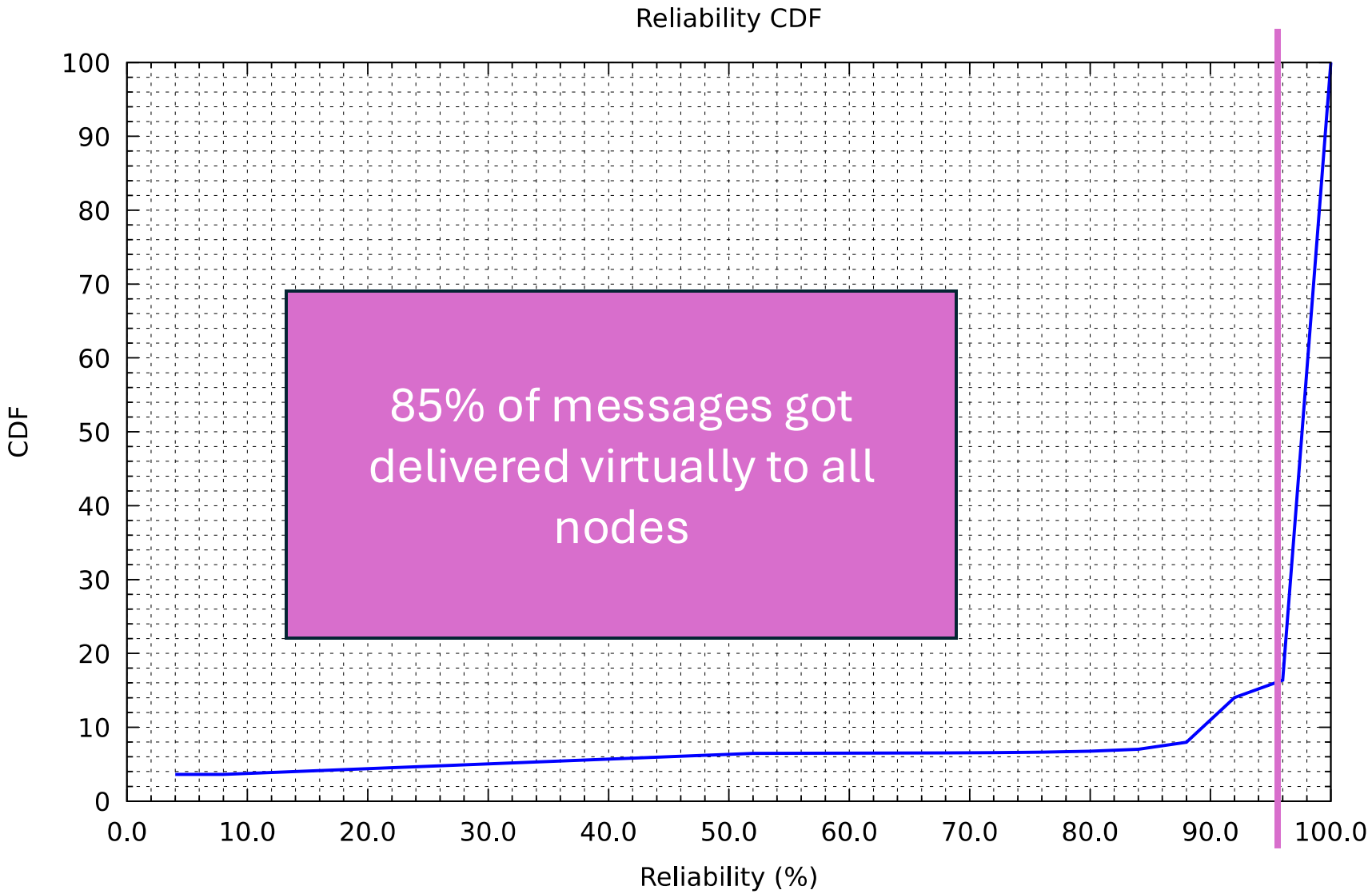
Evaluating GossipSub (with no DHT 25 nodes)



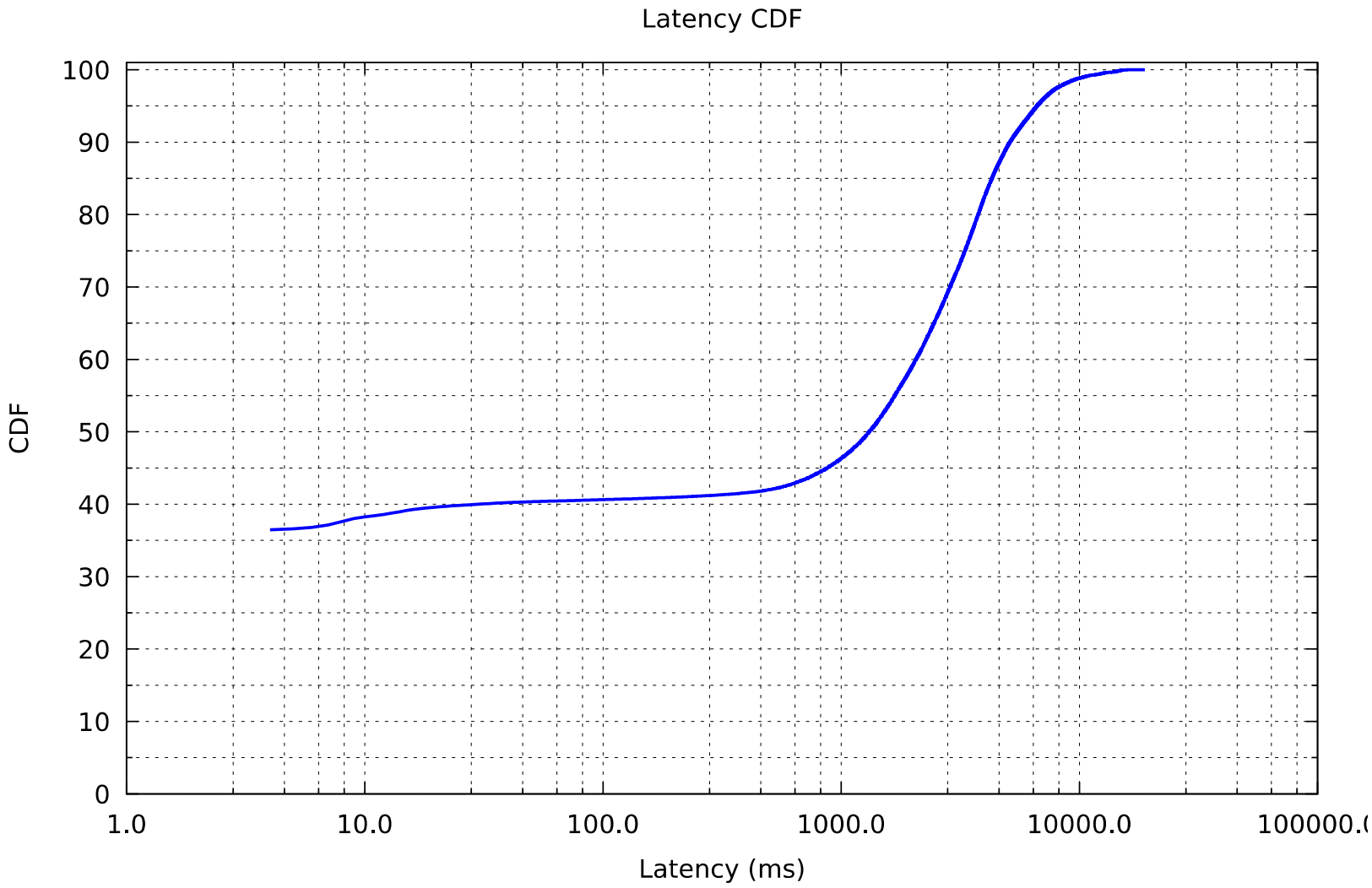
Evaluating GossipSub (with no DHT 25 nodes)



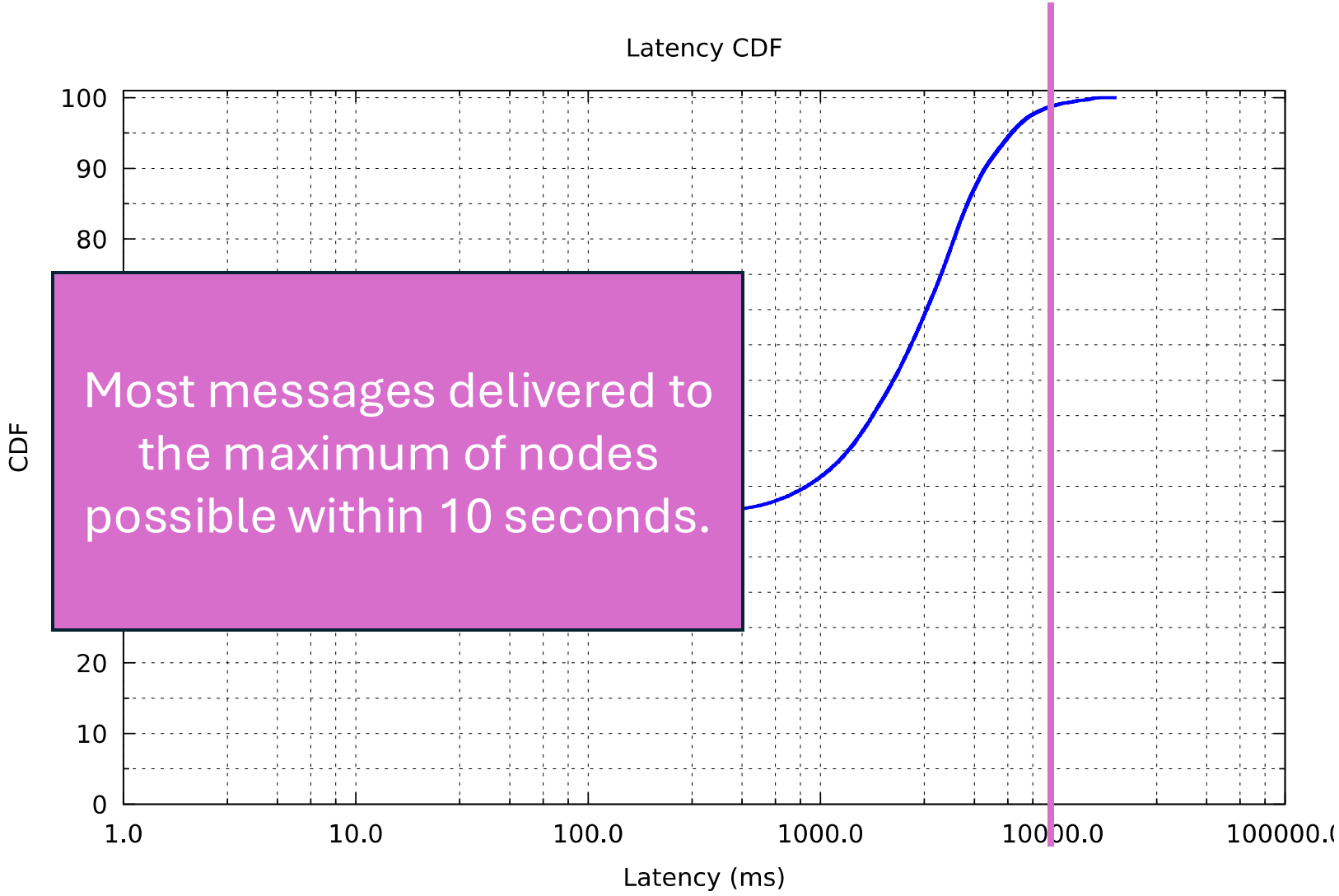
Evaluating GossipSub (with no DHT 25 nodes)



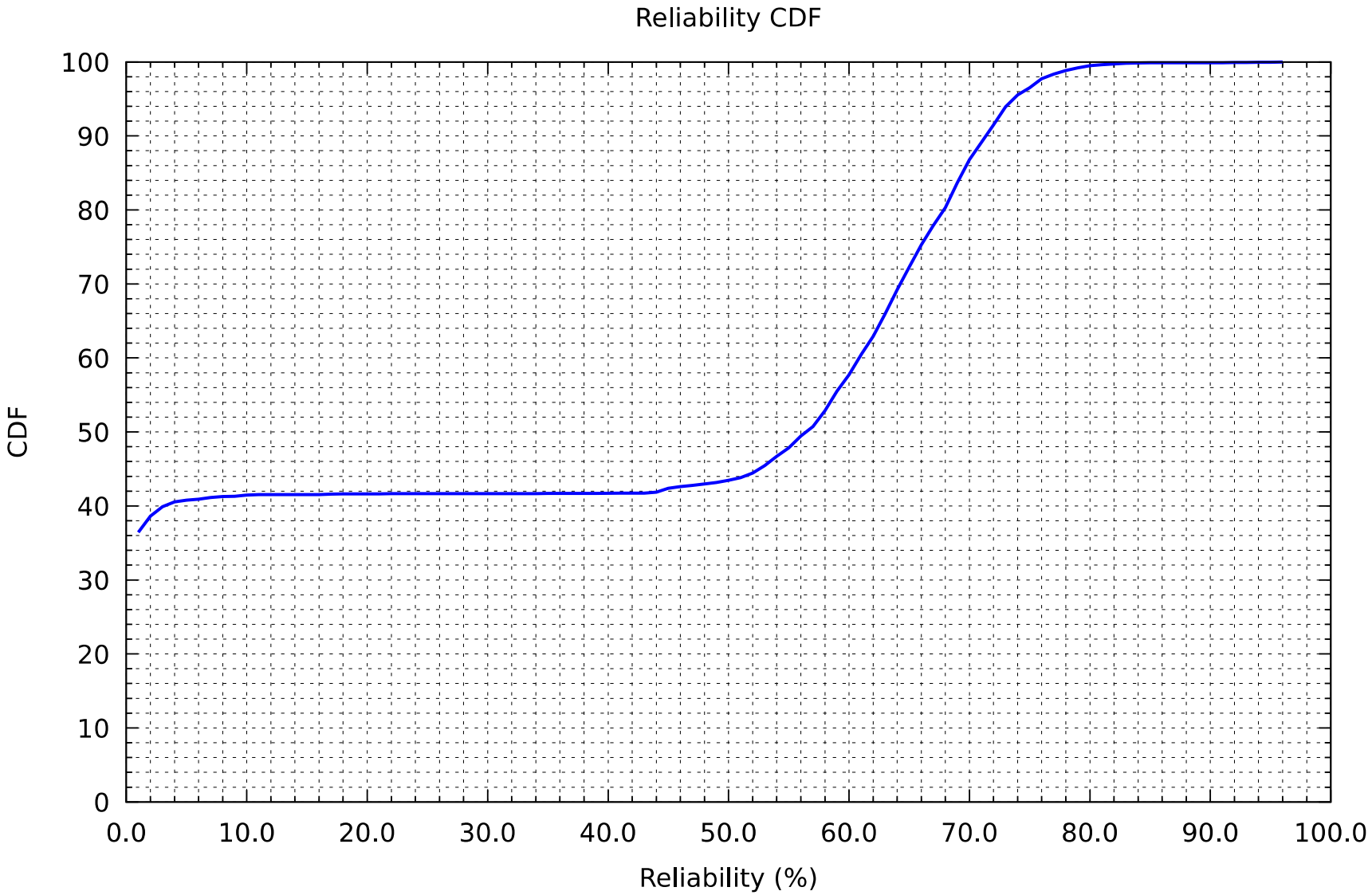
Evaluating GossipSub (with no DHT 100 nodes)



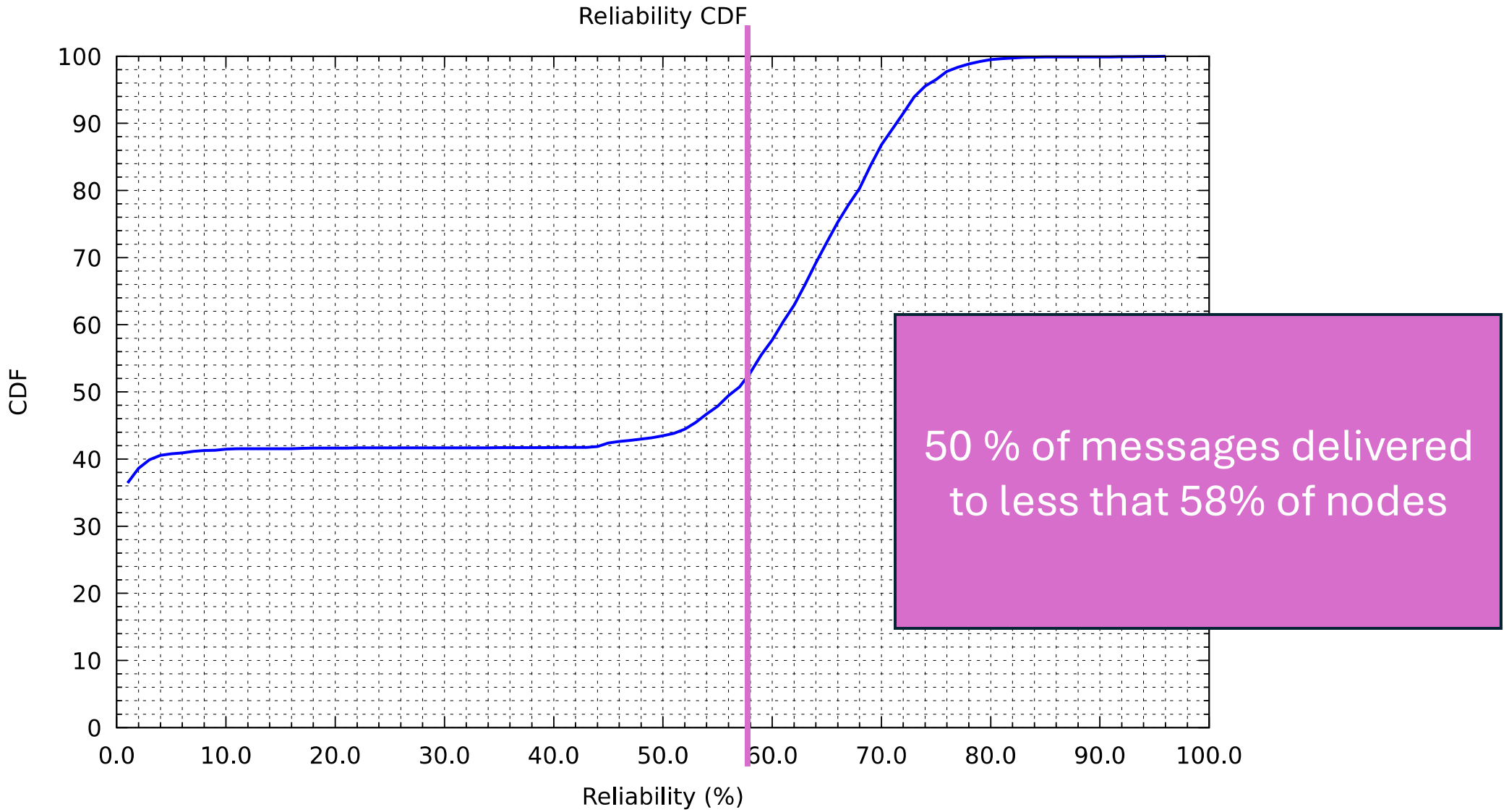
Evaluating GossipSub (with no DHT 100 nodes)



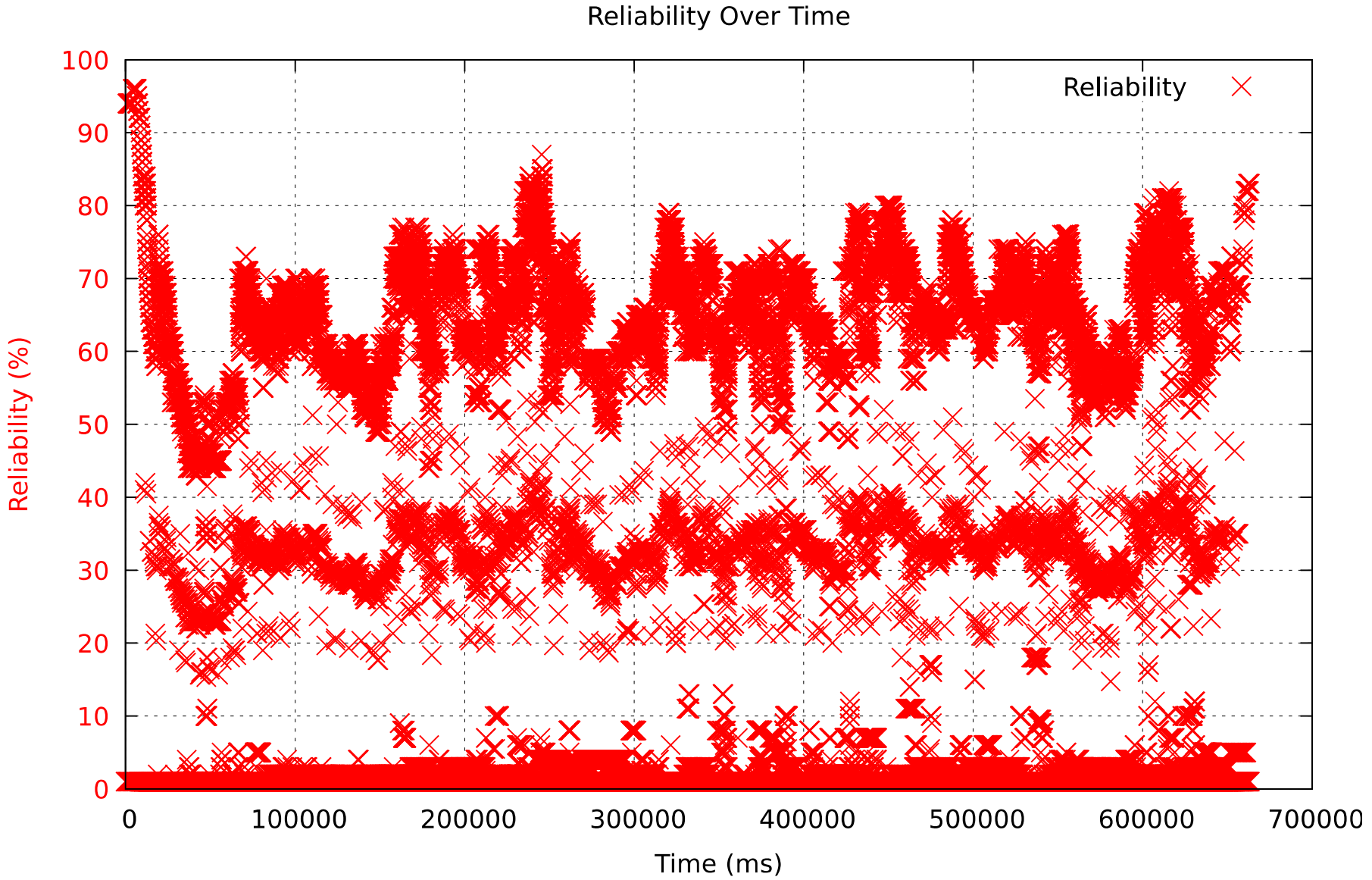
Evaluating GossipSub (with no DHT 100 nodes)



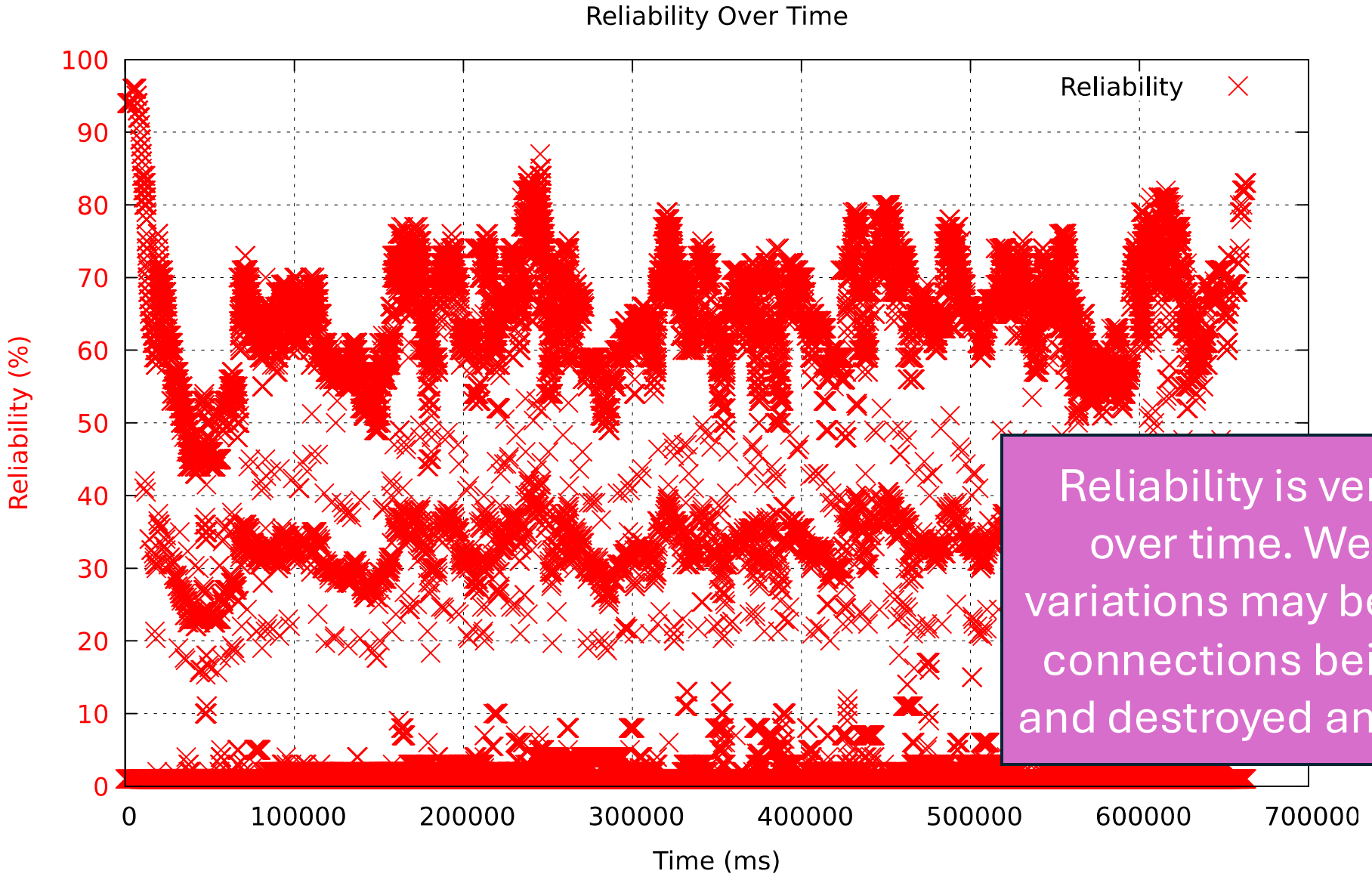
Evaluating GossipSub (with no DHT 100 nodes)



Evaluating GossipSub (with no DHT 100 nodes)



Evaluating GossipSub (with no DHT 100 nodes)



Reliability is very variable over time. We suspect variations may be caused by connections being created and destroyed among nodes.

Evaluating GossipSub (with no DHT)

- Connectivity of nodes at the GossipSub mesh is highly variable (most likely due to the lack of the DHT support, although… all nodes are publishing within the same topic, so it should be easy to find nodes on the same topic).

- Scalability is not great, nor it terms of latency, nor in terms of reliability.

Evaluating GossipSub (with no DHT)

- Connectivity of nodes at the GossipSub mesh is highly variable (most likely due to the lack of the DHT support, although… all nodes are publishing within the same topic, so it should be easy to find nodes on the same topic).

- Scalability is not great, nor it terms of latency, nor in terms of reliability.

- Results must be taken with a grain of salt because the Libp2p implementation in Javascript is known to not be the best around.

Evaluating GossipSub (with no DHT)

- Connectivity of nodes at the GossipSub mesh is highly variable (most likely due to the lack of the DHT support, although… all nodes are publishing within the same topic, so it should be easy to find nodes on the same topic).

- Scalability is not great, nor it terms of latency, nor in terms of reliability.

- Results must be taken with a grain of salt because the Libp2p implementation in Javascript is known to not be the best around.
- Unfortunately, hard to replace this protocol by another ☹

Roadmap of this Talk

- Overview of IPFS (and Libp2p)
- What is it that we know about IPFS on the wild
- What are pressure points in IPFS and Libp2p
- **Beyond IPFS and Libp2p on Decentralized Systems**
- Takeaway Points

TaRDIS European Project

- It is actually a great time to do research on decentralized systems, there are lots of open challenges and new domains.

- TaRDIS is a European Project where we are developing new technologies to support (smart) swarms of devices (highly heterogenous decentralized systems).

- Interesting use-cases: decentralized renewable energy markets, smart factories, decentralized machine learning platforms, swarms of satellites.

TaRDIS

https://project-tardis.eu/

Tools for building Decentralized Systems

- We are evolving a framework to build robust and highly modular distributed (and decentralized systems) called Babel.

P. Fouto, P. Á. Costa, N. Preguiça and J. Leitão, "Babel: A Framework for Developing Performant and Dependable Distributed Protocols," *2022 41st International Symposium on Reliable Distributed Systems (SRDS)*, Vienna, Austria, 2022, pp. 146-155, doi: 10.1109/SRDS55811.2022.00022.

Tools for building Decentralized Systems

- We are evolving a framework to build robust and highly modular distributed (and decentralized systems) called Babel.

- Easy to replace decentralized protocols by another design/implementation that offer similar functionality.
- Written in Java and ported to Android devices.
- Special features to deal with discovery of entry points and self-management of protocols.

Tools for building Decentralized Systems

- We are evolving a framework to build robust and highly modular distributed (and decentralized systems) called Babel.

- Easy to replace decentralized protocols by another design/implementation that offer similar functionality.
- Written in Java and ported to Android devices.
- Special features to deal with discovery of entry points and self-management of protocols.
- You can think: a much more flexible libp2p (only in Java for now).

Roadmap of this Talk

- Overview of IPFS (and Libp2p)
- What is it that we know about IPFS on the wild
- What are pressure points in IPFS and Libp2p
- Beyond IPFS and Libp2p on Decentralized Systems
- **Takeaway Points**

Main Takeaway points

- Decentralized system are clearly a relevant topic now-a-days
  - Lots of different systems and applications being explored
  - Different domains are emerging as viable for decentralized systems

Main Takeaway points

- Decentralized system are clearly a relevant topic now-a-days
  - Lots of different systems and applications being explored
  - Different domains are emerging as viable for decentralized systems
- IPFS and Libp2p have been key drivers in the Web3 domain
  - IPFS has significant success and adoption
  - Libp2p has been adopted to support development of many Dapps

Main Takeaway points

- Decentralized system are clearly a relevant topic now-a-days
  - Lots of different systems and applications being explored
  - Different domains are emerging as viable for decentralized systems
- IPFS and Libp2p have been key drivers in the Web3 domain
  - IPFS has significant success and adoption
  - Libp2p has been adopted to support development of many Dapps
- Security, Privacy, Misusage of Technology still open issues.
  - Tension between controlling misusage VS avoiding censorship

Main Takeaway points

- Decentralized system are clearly a relevant topic now-a-days
  - Lots of different systems and applications being explored
  - Different domains are emerging as viable for decentralized systems
- IPFS and Libp2p have been key drivers in the Web3 domain
  - IPFS has significant success and adoption
  - Libp2p has been adopted to support development of many Dapps
- Security, Privacy, Misusage of Technology still open issues.
  - Tension between controlling misusage VS avoiding censorship
- Performance and anomalies still affect these systems

Main Takeaway points

- Decentralized system are clearly a relevant topic now-a-days
  - Lots of different systems and applications being explored
  - Different domains are emerging as viable for decentralized systems
- IPFS and Libp2p have been key drivers in the Web3 domain
  - IPFS has significant success and adoption
  - Libp2p has been adopted to support development of many Dapps
- Security, Privacy, Misusage of Technology still open issues.
  - Tension between controlling misusage VS avoiding censorship
- Performance and anomalies still affect these systems
- More and more tooling (support) might be an essential driver for the future.

Thanks to all my collaborators past and present students

Yiannis Psaras
(Protocol Labs)

Nuno Preguiça
(NOVA)

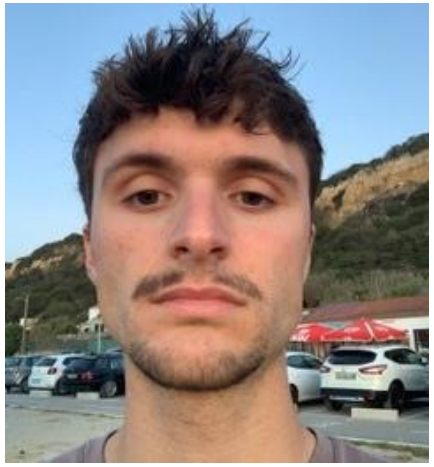
Pedro Fouto (PhD)

João Carvalho (MSc)

Pedro Ákos Costa (PhD)

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

NOVALINCS



Francisco Vale (Intern)

João Brilha (MSc)