The Centralization of a Decentralized Video Platform – A First Characterization Of PeerTube

Michael Polinski
Northwestern University
USA
michaelp@u.northwestern.edu

Richard Jo
Northwestern University
USA
richardjo2023@u.northwestern.edu

Kevin McAfee
Northwestern University
USA
kevinmcafee2022@u.northwestern.edu

Fabián E. Bustamante
Northwestern University
USA
fabianb@cs.northwestern.edu

ABSTRACT

PeerTube is an open-source video sharing platform built as a decentralized alternative to YouTube. With software like Mastodon and Friendica, PeerTube is part of a series of federated social media platforms built partly in response to growing concerns about centralized control and ownership of the incumbent ones. In this paper, we present the first characterization of PeerTube, including its underlying infrastructure and the content being shared on its network. Our findings reveal concerning trends toward centralization that echo patterns observed in other contexts, exacerbated by the limited degree of content replication. PeerTube instances are mostly located in North America and Western Europe, with about 70% hosted in Germany, the USA, and France, and over 50% hosted on the top 7 ASes. We also find that over 92% of videos are stored without any redundancy in spite of PeerTube's native support for video redundancy.

CCS CONCEPTS

• **Computer systems organization → Dependable and fault-tolerant systems and networks**; • **Information systems → Social networks**; • **Networks → Peer-to-peer protocols**; *Network performance analysis*;

KEYWORDS

Decentralized web, Federated services, Video distribution, Characterization

1 INTRODUCTION

Consolidation trends on the Internet – from traffic and infrastructure to services and users – have raised concerns given their potential impact on failure resilience, market control and user privacy [9, 12, 29, 30]. Delivery of standard Internet services, from mail to DNS to web hosting, is consolidating in a handful of large providers, while web applications and platforms appear to function as near monopolies. In few places is this as obvious as in the context of social media [15].

Recently, partially in response to such concerns, a number of decentralized social media alternatives have surged in popularity. This includes Mastodon [26], Hubzilla [28], and PeerTube [23], all of which are part of what is known as the Fediverse [20], a notable subset of the "decentralized web".

Federated services are distributed systems designed to operate with a multiple-hub-and-spoke topology, where each *instance* (typically running on an independent server) communicates with its own registered users (the spokes) as well as other instances (the hubs). The instances in the *federation* have ultimate, independent control over their activity in the system. Federations are usually open, as is the case with the Fediverse. No approval is required by default for a new instance to begin interacting with the federation, but instances can also elect not to interact. To this end, federated services build on common decentralized protocols. OAuth [4] authorization, ActivityPub [1] and WebTorrent [8] are a few such protocols. Depending on the nature of the service, some instances offer open registration to users.

Such systems can offer a number of advantages, from enhancing user privacy and making data ownership more transparent to improving service resilience. The expectation is that federated alternatives to centralized services can increase user privacy by avoiding having a single data silo while bolstering resilience to different types of faults and attacks [13, 33, 37, 40],

There has been recent interest from the research community in characterizing federated services, exploring their potential and real benefits as well as their inherent challenges [10, 14, 18, 27, 34, 39]. Some of these recent studies have shown, surprisingly, a higher level of centralization than expected.

In this paper, we present the first characterization of PeerTube, a federated video sharing service, with a focus on (de)centralization. PeerTube is a distributed video sharing platform offered as an alternative to centralized services such as YouTube and Vimeo. As part of the Fediverse, PeerTube uses the ActivityPub [1] decentralized social networking protocol to federate with other instances as well as any other ActivityPub-compatible servers, like Mastodon servers.

While undoubtedly part of the decentralized web, PeerTube's intended function – i.e., long-form video sharing – translates into significantly different system design with different use models and sharing patterns than ecosystems focused on microblogging (like Mastodon [34]) or other content.

For instance, since videos occupy vastly more storage space than text-based content, PeerTube instances cannot simply replicate the videos of the channels or accounts their users subscribe to. Instead, PeerTube allows instance administrators to configure the "redundancy" policy – controlling the instance's replication of videos published by users on other instances. We present a brief introduction to PeerTube in Section 2.

Our multimodal characterization of PeerTube builds on two key datasets: (*i*) instance hostnames and configuration objects, and (*ii*) video and creator metadata. These were gathered while crawling the network of PeerTube instances. Section 3 describes our measurement approach.

We use the data to characterize the infrastructure underlying the PeerTube network (§4) and the hosted content (§5). We show that PeerTube is a rapidly growing platform with over 1200 live instances as of August 2024, and an active community as evidenced by the majority of the instances running the latest version of the PeerTube software. Despite the scale, we observe that the majority of instances are found in Europe and North America with IP addresses belonging to a handful of large cloud providers. *While others have observed similarly concerning trends toward centralization, in PeerTube this problem is exacerbated by the limited degree of content replication: over 92% of all videos are stored without any inter-instance redundancy.*

In summary, we make the following key contributions:

- We present the first characterization of PeerTube, with a focus on (de)centralization.
- We describe our measurement approach and the datasets we collect on the infrastructure, content and use of the platform.
- We present our findings, highlighting a concerning degree of centralization, made worse by a limited degree of content replication.

An understanding of PeerTube and its challenges as a distributed platform provides the context and motivation for new work to address them in the future.

2 PEERTUBE

PeerTube [22] is an open source, federated video hosting service whose development began in 2017 by a programmer known as Chocobozzz. Since 2018, the development of PeerTube has been supported by the French non-profit Framasoft. The first stable version released in October 2018.

PeerTube follows semantic versioning, and has had 5 major releases to date with minor releases typically between one and three months apart.

The architecture follows a client/server model with inter-server federation, much like email or XMPP. Servers, or *instances*, communicate with each other and with their clients (Fig. 1). They are created by an *instance administrator* on a virtual or bare-metal host. The administrator has control over the server's behavior and can configure registration, federation, and video redundancy policies.

Registered users can upload videos to channels on their home instance, which becomes that video's *origin* instance. The origin instance maintains control over the video and its associated data, including comments and the view count. Videos are bound to their origin instance, but an administrator can migrate the whole instance by copying the underlying database and video files. However, PeerTube allows other instances to copy the complete video and serve it to clients to improve performance and availability. Such instances are known as *caching* instances and are the basis of what PeerTube refers to as video *redundancy*, *duplication*, or *mirroring*, interchangeably. Redundancy is controlled by the administrator, who picks instances from which to mirror as well as one or more redundancy "strategies" with storage quotas. The three available caching strategies can fetch the most viewed, most recently added, or most actively trending videos. Note that, within the limit of the quotas, the server will not deprioritize caching large videos.

PeerTube servers communicate with each other using the ActivityPub protocol. The protocol defines semantics for operations between a set of social media entities, including posting content, following, liking, etc. In the case of PeerTube, when the administrator of an instance chooses to *follow* another, its users can see and interact with that instance's videos. The follower instance additionally *lists* that instance's videos, which causes them to be discoverable – whether through the search interface or otherwise – and viewable to users. Instance follows are unidirectional. The PeerTube documentation encourages administrators to maintain their own index of instances to follow, which can be fed as a JSON object to the PeerTube server. Framasoft's aforementioned tracker is one such index. Note that an instance following another instance does not imply that any video replication is occurring – listed videos are still served by the origin instance and any mirroring instances. However, any instance adhering to the protocol will report follow and replication events to the origin instance.

PeerTube clients include the first-party PeerTube web app as well as third-party apps for Android and other platforms. Clients communicate with the servers using a REST API, which provides authentication, search, commenting, and other core functionality. They stream videos using HTTP Live Streaming (HLS) or WebTorrent. PeerTube v6 removed WebTorrent completely in favor of HLS over WebRTC, which was already favored in PeerTube before this release. Both protocols enable clients to download chunks of the video from other instances that have a mirrored copy. The first-party client will also download the video from peers using WebTorrent or peer-to-peer HLS. These peers are the *swarm* of viewers downloading the same video, and each will advertise itself for discovery while it downloads a video.

3 DATA AND METHODOLOGY

For our characterization of PeerTube we build on two key datasets: (*i*) instance hostnames and configuration objects, and (*ii*) video and creator metadata. These datasets were

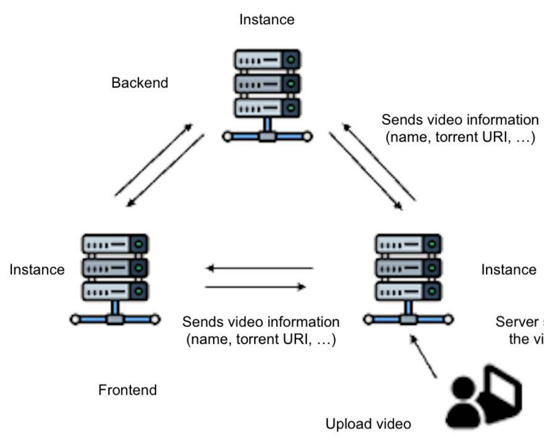


Figure 1: PeerTube Instances

gathered by crawling PeerTube instances, starting with a global list of instances from the Framasoft's tracker [6]. This site contains a list of over 1000 instances from around the world. [1] The crawl was performed in August 2024. We use the REST API exposed by each instance [5] to gather objects describing videos, registered accounts, and instance configuration. Instances expose a list of instances they are following and being followed by, which our crawler adds to its crawling queue recursively. For reference, the PeerTube API documentation provides the complete schema of all the objects [2] [3] [4] our crawler requests and parses.

This approach discovers instances in the "main" connected component of the instance graph, which includes the instances on the tracker's list, but may miss smaller connected components not reachable via the main component. We argue that members of such connected components may be better analyzed as private video-sharing servers or forums, not in the context of PeerTube as a federated social network.

Instances. A list of hostnames was gathered by crawling PeerTube instances and keeping track of unique servers (by hostname) whose configuration REST endpoints return valid responses. We geolocate these hosts based on their IP addresses using MaxMind's GeoIP location database [3], ipinfo.io, and RIPE IPmap [7], and record the results from ipinfo.io. All three sources disagree on the country-level location of 1% of the addresses, and two sources disagree on 17%. We gather an instance configuration object exposed by each instance's API. These objects give us information about the running software (and whether it is a PeerTube server at all) and options/policies set by the instance administrator.

[1]Some instances were not reachable and/or defunct at the time of our data collection.
[2]https://docs.joinpeertube.org/api-rest-reference.html#tag/Config
[3]https://docs.joinpeertube.org/api-rest-reference.html#tag/Instance-Follows
[4]https://docs.joinpeertube.org/api-rest-reference.html#tag/Video

Video Metadata. Video metadata is collected from API responses. Each video metadata object contains information including the number of views, likes and dislikes, account information for the creator publishing the video, and a video category. Video categories are strings, often from a canonical set (e.g. "News & Politics").

Every video is identified by a UUID that is globally unique on the PeerTube network. The collected video data contains all listed videos on each instance, including those from servers being followed by the instance (*non-local* videos). As a result, videos with the same UUID can appear multiple times in our list, regardless of whether they are mirrored or not. We key our dataset by UUID to obtain a list of unique videos while recording each video's *listing factor* – the number of instances listing that video. Since our crawler reaches each listing at a different time, gathering inconsistent view and like counts for the same video, we take the highest view/like count. This is a reasonably accurate estimate given the short total duration of a crawl (about 30 hours).

Notice that, because mirroring another instance's video implies listing it, our listing factor is an upper bound on the redundancy factor, the number of instances that hold a complete copy of a given video. However, we can calculate a more accurate redundancy factor for a subset of instances (88%) which have this data available – and publish valid HLS streams. In recent versions of PeerTube, instances expose multiple streaming sources including one from the originating instance, as well as a redundancies list with URLs to instances that mirror the video. For each video, we gather all unique streaming hosts (including the originating instance and caching instances) to calculate a redundancy factor.

Ethics. Our study relies on public infrastructure information and information on shared content that has been made publicly available. We focus on infrastructure and video metadata using aggregates and do not analyze individuals.

4 PEERTUBE INFRASTRUCTURE

In this section, we characterize the infrastructure behind the PeerTube federation collectively. We focus our analysis on instances, their configuration, and their dependencies, and review their geographic and network locations.

4.1 Instances

Our data collection approach discovered about 1200 total online PeerTube instances. Some of these instances are not visible on the PeerTube instance tracker hosted by Framasoft [6] since Framasoft delists instances under certain conditions. Furthermore, some discovered instances were unreachable; we have excluded these from our analysis.

As with other federated and decentralized services [16, 35, 36], it is virtually impossible to state an exact total count of PeerTube instances at a given moment considering their nonzero churn rate and the number of instances that are misconfigured or temporarily unresponsive.

Country	Instances	Percentage
Germany	268	24.5
United States	254	23.2
France	238	21.8
Finland	53	4.8
Canada	44	4
Russian Federation	34	3.1
Netherlands	26	2.4
Switzerland	26	2.4
Sweden	19	1.7
United Kingdom	16	1.5

Table 1: Top countries by number of instances. The top 3 countries host about two-thirds of all instances.

From the collected instance configurations, we noticed several trends. Only about 19% of instances had open public registration with many of the remaining instances being used only by small groups of users, and others likely offering exclusive registration in some other manner (judging by the number of users).

Additionally, we found that 70% of instances, at the time of writing, were running the latest major version of the PeerTube server software, and only 14% were more than one major version behind. 42% were running the latest minor release. This suggests a very active community of instance administrators, since the upgrade process is not automatic.

4.2 Geolocation, ASes, and Hosting

We analyze the distribution of instances across geographical locations and autonomous systems (ASes), thus also inferring information about the infrastructure providers powering the PeerTube network.

We find that the majority of instances are located in Europe and North America. Specifically, Germany, the United States, and France lead with well over half of the total.

Table 1 lists the top countries by number of instances as geolocated by ipinfo.io. Note that the development of PeerTube is supported by Framasoft, a French non-profit organization [23].

To determine the ASes hosting PeerTube instances, we resolve the hostname of each instance using a public DNS resolver. This yields a set of IPv4 addresses; we ignore IPv6 addresses since fewer than half of all instance hostnames have a AAAA record. We then query RIPEstat's API to determine the ASN and the name of the AS holder for every IP address. 4 of these IP addresses are advertised by more than one AS (making them multihomed).

Figure 2 shows the cumulative distribution function of instances per AS. We see that 62% of instances are hosted in the 15 most popular ASes, and over 50% of instances are hosted by the 7 most popular ASes.

Table 2 lists the most popular ASes, their ASNs, and the percentage of instances hosted there. The overwhelming majority of top ASes belong to large cloud providers. Unsurprisingly, the two most popular ASes belong to Hetzner Online and OVHCloud, a popular German and French cloud hosting provider, respectively. These are followed by Cloudflare's AS, which likely includes many instances that are solely using Cloudflare's DDoS mitigation services. We believe that the notable absence of US cloud hyperscalers, such as AWS and GCP, is caused largely by the current lack of competitive pricing, especially for transfer-out network bandwidth. For reference, at the time of writing, OVHCloud and Scaleway provided unlimited monthly transfer for at least some of their VPS product offerings. Most other popular providers charged 1–2 orders of magnitude less per byte of network traffic than the hyperscalers, often with 10TB or more included for free.

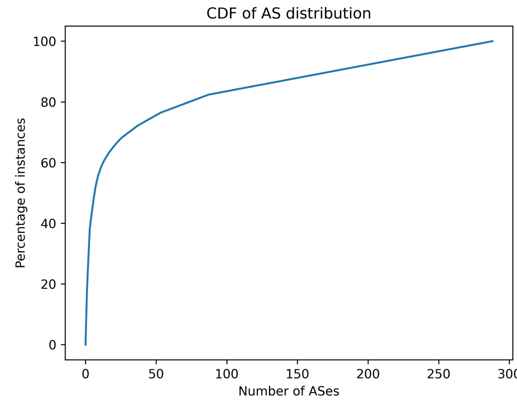


Figure 2: CDF of AS Distribution.

AS Holder	ASN	Percentage
HETZNER-AS - Hetzner Online GmbH	24940	17.5
OVH - OVH SAS	16276	10.7
CLOUDFLARENET	13335	9.9
AS12876 - SCALEWAY S.A.S.	12876	3.8
NETCUP-AS - netcup GmbH	197540	3.5
PROXAD - Free SAS	12322	3.3
CONTABO - Contabo GmbH	51167	3.1
DIGITALOCEAN-ASN	14061	2.1
AKAMAI-LINODE-AP Akamai Connected C	63949	1.9
AS-CHOOPA	20473	1.3
AS3215 - Orange S.A.	3215	1.3
OCTOPUCE-AS - Octopuce s.a.r.l.	28855	1.1
ORACLE-BMC-31898	31898	1
"GITOYEN-MAIN-AS - Association ""Gito"	20766	0.8
DTAG - Deutsche Telekom AG	3320	0.8

Table 2: Top ASes by number of instances; the top 5 host 45% of all instances.

4.3 Federation graph

We analyze the federation graph, in which each instance is a node and edges represent a "followed-by" or "following" relationship.

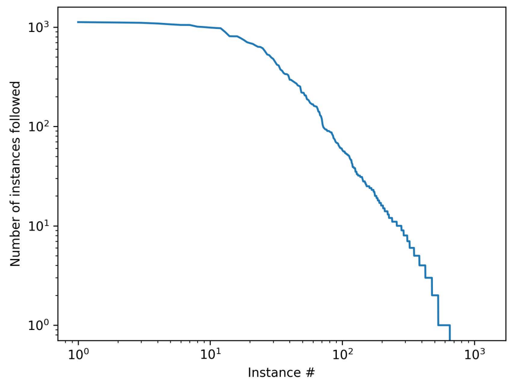


Figure 3: A log-log plot of the number of *follows* of an instance against instances ordered by their total number of follows. The inflection point separates the "super-followers".

We observe that the dominant follow pattern in the network consists of a few instances, which we call "super-followers", that follow nearly every other instance in the observable network. Upon manual inspection of their "About" pages, it's clear that many of these instances explicitly have the goal of being an aggregator providing a listing of videos across the PeerTube network, and some have few to zero local videos. Figure 3 plots the number of follows of each instance, simultaneously sorting instances by this metric. It shows a clear inflection point separating the "super-followers" from the remainder of the instances, which have the expected power law distribution with their number of follows.

Furthermore, there is a less obvious but noticeable pattern separating the commonly followed instances from those on the periphery of the network, some of which are followed by no other instance (including the "super-followers"). Appendix A describes this pattern in additional detail with the help of graph visualizations.

5 CONTENT

We now examine the content on the PeerTube network, focusing on redundancy. Then, we briefly examine the video production of the network over time and the popularity distribution of videos.

5.1 Redundancy

Compounding the problem of infrastructure centralization, we observe that content published on the PeerTube network is (*i*) listed across a few instances and (*ii*) redundantly copied on even fewer instances. Specifically, most videos (80%) are listed across 3 or fewer instances, and 95% of videos are listed across 6 or fewer. More problematically, PeerTube's redundancy feature is not widely used. Videos are redundantly stored across a low number of instances, with 92% of content stored on a single instance.

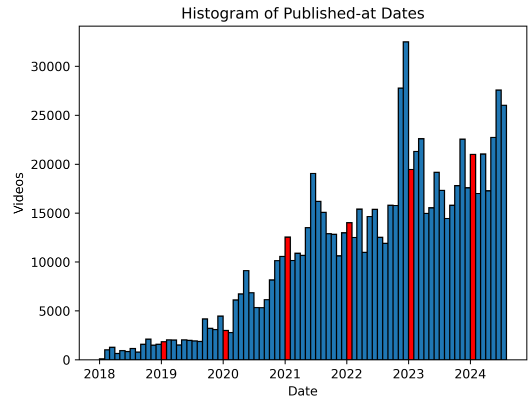


Figure 4: Videos published/month (Jan 2018 - Aug 2024)

First, a low listing factor indicates that content is displayed on few instances on average, which impedes the discoverability of content for users of other instances. Second, a low redundancy factor indicates that a limited number of instances store a local copy of each video, which restricts from where users can retrieve video.

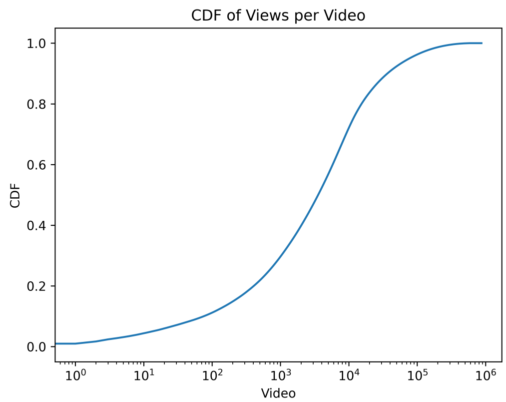
Although these results are concerning, they are consistent with the design of the PeerTube software, which has an "opt-in" policy for content federation across servers. This design choice acknowledges the significant cost of storage and bandwidth for video content, requiring administrators to configure a redundancy strategy. Unlike in other federated applications, content is not distributed across all servers automatically, centralizing content on a small set of servers that can handle it. This means the network as a whole likely has low resilience against failures in internet infrastructure as well as censorship, whether in the form of DDoS (distributed denial-of-service) attacks or simple blocking based on DNS, IP address, or TLS Server Name Identification (SNI) headers.

5.2 Production trends

The **total** number of videos published using PeerTube is growing quadratically year over year. Note that we do not have historical data, so videos no longer listed by any instance are excluded. The month-to-month publication rate generally follows the quadratic trend, but there are months with sudden spikes in videos published (Fig. 4). While the cause of each is not immediately clear to us, some are likely related to spikes of growth of the whole Fediverse, such as the spike in late 2022 after Twitter (now X) changed ownership. Though we did not inspect the video data for such activity, it is possible that instances may be spoofing videos (returning listings and metadata for nonexistent videos).

5.3 Popularity distribution

The 872,653 unique videos we discovered collectively held a total of 124,172,763 views in August 2024. Figure 5a presents a CDF of the number of views per video. As is typical in



(a)

Figure 5: CDF of views per video

such distributions, views are largely concentrated on the most popular videos. Over 3% of all views belong to just 5 videos. We also found that 305,282 videos did not have a single reported view.

Likes and dislikes offer another indicator of content popularity. We found a total of 264,393 likes across all videos for an average of 0.30 likes per video, and 13,725 dislikes for an average of 0.016 dislikes per video. We find likes and dislikes heavily concentrated on a small set of videos, with the majority of videos having no likes or dislikes. We observe no relationship between any of likes, dislikes, and views.

6 RELATED WORK

We present what is, to the best of our knowledge, the first characterization of PeerTube, a federated video sharing service. The service is offered as an alternative to centralized video hosting services such as YouTube and Vimeo. YouTube has been extensively studied [11, 17, 21, 24, 25, 32, 38].

PeerTube is part of a larger ecosystem of open-source, federated social media software platforms such as Mastodon [26], Pleroma and Diaspora [14]. PeerTube servers are an active part of the aforementioned Fediverse [20]. Raman et al. [34] present a large-scale study of Mastodon as an example of the decentralized web, exploring the feasibility and challenges of running federated social web systems. Fediverse platforms are in turn part of broader developments in decentralized systems. The IndieWeb [31] initiative and InterPlanetary File System are notable and key in these developments, as are various blockchain-based systems.

Trautwein et al. [39] describe the design and implementation of IPFS, an open-source, content-addressable P2P network that provides distributed data storage and delivery, and serves as a platform for various decentralized web applications such as social networking and content sharing. Balduf et al. [10] presents a large-scale measurement study of IPFS that shows similarly concerning trends toward centralization with almost 80% of IPFS DHT servers hosted in the cloud with the top three cloud providers hosting nearly 52% of all servers.

DTube is another video streaming platform that builds on IPFS and the Steem blockchain for content delivery and curation. Unlike the federated PeerTube, the platform is developed with the intent of being fully P2P. Viet Doan et al. [19] assess the viability of DTube. The work takes the perspective of an end user and measures the performance of DTube video streaming, which currently leverages IPFS HTTP gateways that act as a "CDN" to improve streaming performance over the direct use of IPFS. They find such a content caching model to be potentially competitive with centralized services like YouTube.

7 CONCLUSION

PeerTube's support for user-generated, long-form video presents a unique set of challenges different from those faced by other types of federated services, such as microblogging, ranging from the model of replication used to ensure availability to the approach clients use for content download. In this paper, we presented the first characterization of PeerTube, the content being shared using it, and the underlying infrastructure. Our study reveals clear trends toward centralization, including a reliance on large hosting providers, across few networks and countries, but with the added problem of near zero redundancy. These findings shed light on important aspects of PeerTube's current state, emphasizing the need for further examination and potential strategies to enhance its decentralization and content resilience.

7.1 Improving decentralization

Increasing video redundancy could be a straightforward way to enhance PeerTube's fault tolerance. For starters, increasing the use of PeerTube's native redundancy functionality could simultaneously provide additional available bandwidth to copied videos and reduce the risk of them becoming unavailable if the origin instance goes offline. This could be achieved by simply making it a suggested default for an instance administrator configuring a new server. For instances run on dedicated cloud servers or VPSs – the majority – it would make sense to allow PeerTube to use the entire disk for redundancy.

In addition, while more complex than simply altering instance redundancy defaults, PeerTube developers could consider implementing support for external video redundancy to gain the benefits of distributed storage. The peer-to-peer protocol IPFS [39] could provide such storage. This is not a new proposal; commenters in [2] discussed the use of IPFS [39] for video storage years ago. Section 6 further discusses IPFS and an existing IPFS-based video sharing network. To resolve the inherent reliability issues with peer-to-peer storage, the IPFS-backed Filecoin network could be used by instance administrators to pay for IPFS storage at competitive prices.

A FEDERATION GRAPHS

Two NEATO graphs (drawn with the Graphviz NEATO engine [5]) were created to help visualize the instance federation graph. Each of these graphs consists of 100 random instances sampled from the set of discovered instances **plus all directed neighbors** of each of the chosen instances. This sampling was done to reduce the visual crowding that makes these graphs otherwise impossible to interpret.

Figure 6 shows instances with directed edges to their followers (that is, an arrow is drawn from A to B if B follows A). The graph shows a clear cluster of instances in the center with a second ring of followers around it. Many of these followers are likely the "super-follower" instances. The central cluster is distinct from the instances on the periphery of the network, which are likely small and have fewer users and videos.

Figure 7 shows a similar visualization with the edges instead directed away from followers (that is, an arrow is drawn from A to B if A follows B). The central cluster mirrors the "super-follower" set in Figure 3.

On each graph, nodes are colored by country such that all instances from a given country will have the same color, helping to visually identify follower groups that share a country. From the node colors, there are no obvious clumps of "socially isolated" instances in a single country not followed by instances from other countries.

REFERENCES

[1] ActivityPub. https://activitypub.rocks/.
[2] Ipfs to store videos, issue 494. https://github.com/Chocobozzz/PeerTube/issues/494.
[3] MaxMind geolocation data. https://dev.maxmind.com/geoip/geolite2-free-geolocation-data.
[4] OAuth. https://oauth.net/.
[5] PeerTube API. https://docs.joinpeertube.org/api-rest-reference.html.
[6] PeerTube instances list. https://joinpeertube.org/instances.
[7] RIPE IPmap. https://ipmap.ripe.net/.
[8] WebTorrent. https://webtorrent.io/.
[9] Consolidation in the internet economy, Feb 2020.
[10] Leonhard Balduf ad Maciej Korczynski, Onur Ascigil, Navin V. Keizer, George Pavlou, Bjorn Scheuermann, and Michal Kroll. The cloud strikes back: Investigating the decentralization of IPFS. 2023.
[11] Vijay K. Adhikari, Sourabh Jain, Yingying Chen, and Zhi-Li Zhang. Vivisecting YouTube: An active measurement study. 2012.
[12] Jari Arkko. Centralised architectures in internet infrastructure. *IETF Internet Draft*, 2019.
[13] Eloise Barry. These are the countries where Twitter, Facebook and TikTok are banned. *Time Magazine*, January 2022.
[14] Ames Bielenberg, Lara Helm, Antholy Gentilucci, Dan Stefanescu, and Honggang Zhang. The growth of Diaspora - a decentralized online social network in the wild. In *INFOCOM Workshop*, 2012.
[15] Ian Bogost. So much for the decentralized internet. *The Atlantic*, July 2020. https://www.theatlantic.com/technology/archive/2020/07/twitter-hack-decentralized-internet/614593/.
[16] Fabián E. Bustamante and Yi Qiao. Designing less-structured p2p systems for the expected high churn. 16(3), June 2008.
[17] Pedro Casas, Pierdomenico Fiadino, Arian Bär, Alessandro D'Alconzo, Alessandro Finamore, and Marco Mellia. YouTube all around: Characterizing YouTube from mobile and fixed-line network vantage points. In *Proc. of EuCNC*, 2014.
[18] Lucio La Cava, Sergio Greco, and Andrea Tagarelli. Understanding the growth of the Fediverse through the lens of Mastodon. *Applied Network Science*, 6(64), September 2021.
[19] Trinh Viet Doan, Tat Dat Pham, Markus Oberprieler, and Vaibhav Bajpai. Measuring decentralized video streaming: A case study of dtube. In *Proc. of the IFIP Networking Conference (Networking)*, 2020.
[20] Fediverse.party. About fediverse - diversity is strength. https://fediverse.party/en/fediverse/.
[21] Alessandro Finamore, Marco Mellia, Maurizio M. Munafo, Ruben Torres, and Sanjay G. Rao. YouTube everywhere: impact of device and infrastructure synergies on user experience. 2011.
[22] Framasoft. Peertube documentation. https://docs.joinpeertube.org/admin-following-instances.
[23] Framasoft. PeerTube. https://joinpeertube.org/, October 2022.
[24] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. Youtube traffic characterization: a view from the edge. 2007.
[25] Danilo Giordano, Stefano Traverso, Luigi Grimaudo, Marco Mellia, Elena Baralis, Alog Tongaonkar, and Sabyasachi Saha. YouLighter: A cognitive approach to unveil YouTube CDN and changes. *IEEE TCCN*, 1(2), 2015.
[26] Mastodon GmbH. Mastodon. https://joinmastodon.org/, October 2022.
[27] Jahui He, Haris Bin Zia, Ignacio Castro, Aravindh Raman, Nishanth Sastry, and Gareth Tyson. Flocking to mastodon: Tracking the great twitter migration. 2023.
[28] Hubzilla. Hubzilla. https://hubzilla.org/, October 2022.
[29] Geoff Huston. DNS resolver centrality. APNIC Blog, September 2019.
[30] Geoff Huston. CDN and centrality. APNIC Blog, July 2021.
[31] IndieWeb. What is the IndieWeb? https://indieweb.org/, 2022.
[32] Ricky K. P. Mok, Vaibhav Bajpai, Amogh Dhamdhere, and K. C. Claffy. Revealing the load-balancing behavior of youtube traffic on interdomain links. 2018.
[33] Associated Press. Youtube blocked in brazil. *LA Times*, January 2007.
[34] Aravindh Raman, Sagar Joglekar, Emiliano De Cristofaro, Nishanth Sastry, and Gareth Tyson. Challenges in the decentralized web: The mastodon case. 2019.
[35] S. Rhea, D. Geels, Timothy Roscoe, and John Kubiatowicz. Handling churn in a DHT. 2004.
[36] Daniel Stutzbach and Reza Rejaie. Understanding churn in peer-to-peer networks. 2006.
[37] Josh Taylor. Facebook outage: what went wrong and why did it take so long to fix after social platform went down. *The Guardian*, October 5 2021.
[38] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo, and S. Rao. Dissecting video server selection strategies in the YouTube CDN. 2011.
[39] Dennis Trautwein, Aravindh Raman, Gareth Tyson, Ignacio Castro, Will Scott, Moritz Shubotz, Bela Gipp, and Yannis Psaras. Design and evaluation of IPFS: a storage layer for the decentralized Web. 2022.
[40] Decla Walsh. Pakistan blocks Facebook in row over Muhammad drawings. *The Guardian*, May 19 2010.

[5] https://graphviz.org/docs/layouts/neato/

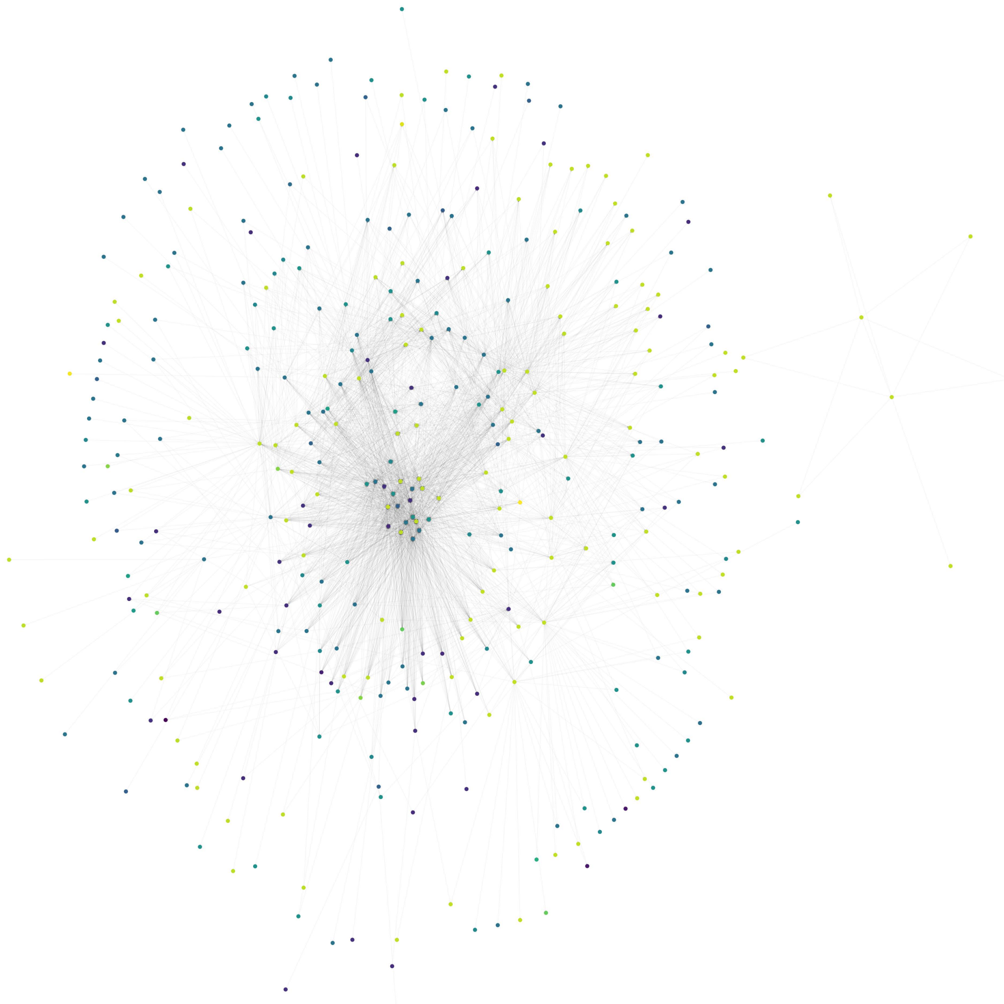


Figure 6: A NEATO visualization of instances with directed edges to followers.

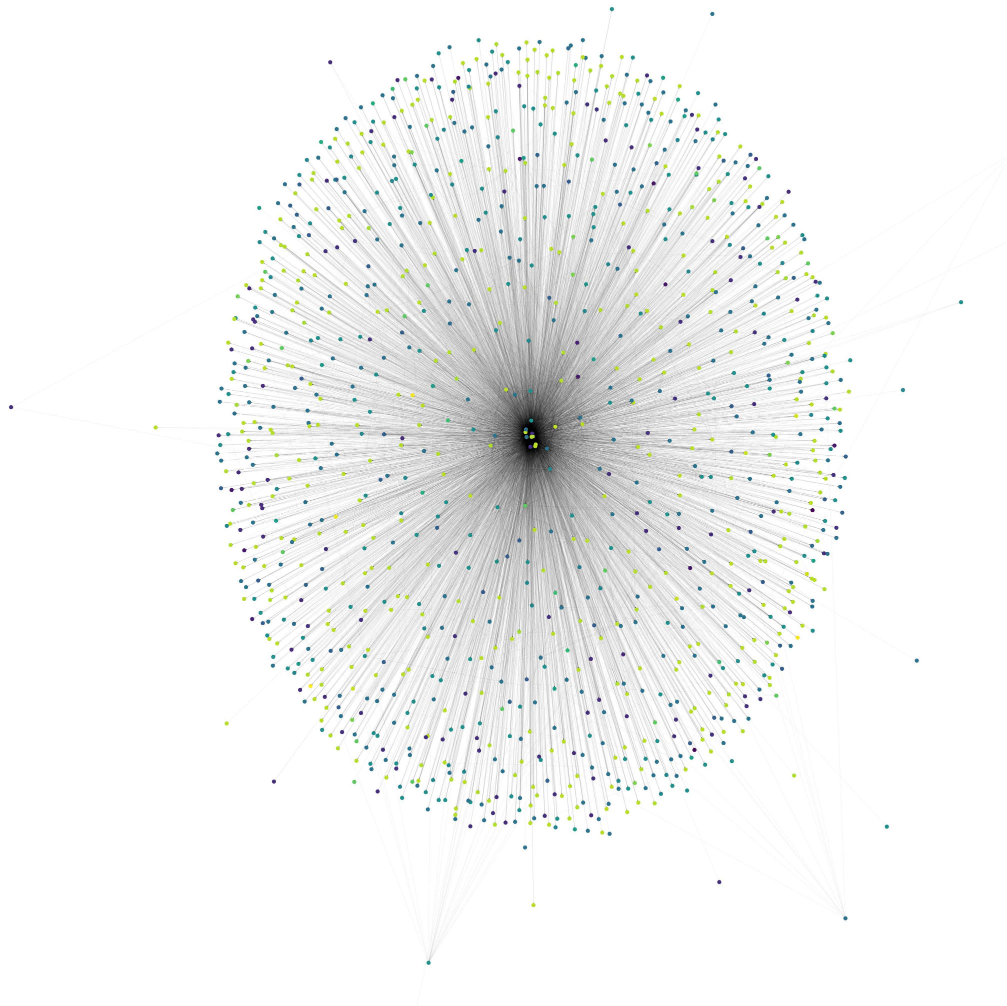


Figure 7: A NEATO visualization of instances with directed edges to the instances they follow.