An Analysis and Comparison of Two Decentralized Peer-to-Peer Networks: FreeNet vs. ZeroNet

Kevin Situ, V00867500

University of Victoria, Victoria, BC, Canada

April 9, 2021

Abstract

This paper describes, discusses and compares the architecture, protocols and qualitative behaviors of two decentralized peer-to-peer networks: "FreeNet", created in 2000 and "ZeroNet", from 2015. These two networks were both created to address issues surrounding privacy and denial of service that arose from the current internet. We will provide a general overview of the two networks features and protocol details and then describe and compare the feature set, user experience, and the creation of websites for the respective networks.

1 Introduction

With the increasing prevalence of the Internet, information has never been easier to access than before and the importance of it grows each day. However, with the increased power and potential of the Internet, the flaws and weaknesses with the system become visible as parties seek to exploit and manipulate it to accomplish their own desires. Some of these parties seek to censor information, deny access to services or expose the identities of individuals and compromise user's privacy. Additionally, the current internet does not focus much on

providing privacy for it's users and needs to be accomplished by other systems built on top of it.

Various networking technologies have been developed to address these issues and although there are countless number of them that have been tried over the years, this paper will focus on two systems one from twenty years ago, FreeNet, and one more recent, ZeroNet and compare the differences between them. The motivation for focusing on these two networks stems from the desire to adequately assess two networks, one from a relatively long time ago and one from a relatively current time. Ideally, these two networks would be somewhat popular and show some form of improvement or differentiation from other technologies. In this way, we can learn from what has worked, what could work, and see what lessons have been learned.

The paper will be organized as follows. First, section two will describe some background information on the general nature of the type of networks that FreeNet and ZeroNet are as well as some information on related works. The next two sections will describe FreeNet and ZeroNet, their architecture and some details about their protocol. Section five will discuss and compare the two networks qualitatively. Lastly, section six will conclude the paper and provide some ideas for future work.

2 Background

In the traditional internet network model, computers mainly focus on a client-server model where communication and resources are mainly done by users talking to a centralized server. Although this model is simple to setup and easy to understand, this model has several issues. By having a central server, this makes the network more vulnerable as attackers can simply attack these central point of failure. Servers can be taken down by denial of service, whether that be by the actions of an attacker or by real visitors visiting a site in a sudden surge of popularity. The types of networks that FreeNet and ZeroNet do not follow this model but

instead use a different paradigm called Peer-to-Peer (P2P) networking. More specifically these P2P networks are decentralized in that they have no centralized points of failure.

Peer-to-Peer networks are characterized by the fact that nodes (commonly called Peers) in the system are now both a client and a server. Typically, these nodes are equally privileged and provide their own resources to the system to function. Tasks which were formally done on a central server are now split up and done by the network. By doing this, attackers now have to attack the network as a whole to prevent access to a service or at least decipher which specific nodes in the network might be handling that service. In some cases, sites might be served by many nodes thus furthering the resilience of a node. Decentralized P2P networks are special cases of P2P networks that have no centralized functionality at all. Some previous P2P networks used a centralized server to coordinate some functionality of the system. In the case of both of these networks, we will see later how they are decentralized.

2.1 Related Work

Several types of related networks are relevant to the networks discussed in this paper. Early P2P networks such as Napster[1][1] and Gnutella[2] were fundamental in influencing and demonstrating the capability and popularity of P2P networks, proving the feasibility of such networks. However, both of these networks had flaws. Napster's reliance on centralized servers as indices[3] led it to being taken down quickly while Gnutella's scalability was a big criticism[4].

Some important technologies that ZeroNet relies upon include BitTorrent[5], a P2P network which was one of the earliest to allow downloading files from multiple sources, distributing load and work across the network by splitting pieces of files. Another technology important to ZeroNet is the BIP32 address scheme[6]. This technology was originally intended as a scheme to create addresses for hierarchical deterministic wallets for Bitcoin but ZeroNet uses this as the address instead of using IP addresses.

[1]Archived from Original: http://www.napster.com/

Relevant networks to this project include such technologies as I2P[7][8] (Invisible Internet Project), a fork of FreeNet intended as a replacement for it's communication layer that developed into its own popular network utilizing garlic routing, tunnels and series of hops through routers to obfuscate sender and receivers. Tor[9] is another relevant privacy focused technology that is frequently compared to FreeNet and that ZeroNet supports using.

Finally, in terms of relevant work done in comparing FreeNet or ZeroNet with other networks there has been some work done with respect to FreeNet. Hu et al. compared FreeNet with various other darknet networks by classifying user behaviors[10]. Negi[11] performed a short qualitative comparison of FreeNet with other anonymous communication networks, detailing a number of factors such as speed and usage to community support.

3 FreeNet

FreeNet, created in 2000 by Ian Clarke while he was at the University of Edinburgh, is "a cooperative distributed information storage and retrieval system designed to address ... concerns of privacy and availability"[12] incorporating "location independence and transparent lazy replication". The system also is made to respond adaptively to usage patterns by replicating, moving and deleting files as necessary and according to demand.

3.1 Architecture

FreeNet's architecture is implemented as a P2P network of nodes. Nodes can query and store data files that are named by location-independent keys which are obtained by hashing descriptions of the files. Each node maintains a local datastore, which is lent to the network to be used for reading and writing, and a dynamic routing table which contains addresses of nodes and the keys that they hold.

The nodes communicate through queries which get passed between nodes with each node making a local decision of where to send the query next creating a chain of requests. These

queries are assigned a pseudo-unique random identifier so that nodes can recognize previous queries and reject ones which they have already seen to prevent loops. Queries also have a "hops-to-live" count that act as a limit so that query requests do not get passed endlessly.

3.2 Protocol Details

FreeNet has three important protocols that it follows to function properly. First is the "Retrieve Data" protocol. To retrieve data, FreeNet uses hashed strings to get keys for files. The input strings should be short and describe the file in question and then hashed. Once getting the key, users send a query for that key. Nodes receiving a request query first check its local data store then if it does not exist it checks its routing table and sends the query to the entry in the table that has the most similar key. If a query is successful, then a node returns the file along the path created by the query. While the file is being returned, each node along the path caches the file in its local datastore and creates a routing table entry associating the actual data source with the requested key. Also on the return route, nodes can change the reply message to claim itself or another node as the data source. There are some cases for failure states for this protocol. If a node fails to send a message to a node in the routing table, the node will try the second-nearest key in the table and if that fails as well, then the third-nearest and so on. If the node goes through all the entries, it will return a failure to the preceding node and the preceding node repeats the same process. If the query exceeds it's hop limit then a failure result is sent back to the original request.

Next is the "Store Data" protocol. An insert message is first comprised of a key and a hop count. The key is created from a hashed description string. The insert message is sent to a user's own node first, which will check it's own datastore to see if it already has a matching key. If the key is not found then a node checks its routing table and routes the insert message to the entry in the table with the nearest key. If the key collides with some node further along, then the node behaves as if it was a "Retrieve Data" message and returns the file along the created path. If the key is not found after completing the number of hops

in the hop count, then an all clear message is returned to the original insertion. The user then knows to send the file along the created path and each node can store the file. Nodes cana again claim that they or another node was the source of the data.

Lastly is the "Managing Data" protocol. FreeNet has some procedures to handle storage issues. A node's datastore and routing table is managed as a least recently used cache, with items being sorted by the order that they were last requested. This behavior leads to files which are not accessed often being dropped from the system.

4 ZeroNet

ZeroNet[13], created in 2015 by Tamas Kocsis, is a P2P network based mainly off of BitTorrent and Bitcoin-based-cryptography but with some differences. In most P2P networks, and the BitTorrent network which ZeroNet is based off of, the system is mainly file-sharing based. However, ZeroNet is more focused on site publishing. ZeroNet aims to solve the current Internet's issues by removing centralized points of failure, being free of hosting costs, and un-censorable. Additionally, the network supports offline browsing if the site has been visited, and anonymity through Tor.

4.1 Architecture

ZeroNet's architecture is very similar to BitTorrent[14]. It shares concepts like peers and trackers where peers are users/nodes running a ZeroNet client and trackers are a special type of server that assists in the communication between peers. In ZeroNet specifically, trackers are used to help keep track of where copies of sites are in the network. Another important piece are ZeroNet sites or "Zites" which act similarly as files in a BitTorrent system. Sites are created with a private/public BIP-32 address instead of an IP address. These BIP-32 key pairs can be used to sign and publish changes to a site. One important piece of a ZeroNet site is a file named "content.json" which holds all other file names, hashes and the site owner's

cryptographic signature.

4.2 Protocol Details

ZeroNet has a few helpful protocols to help the system function[15][16]. First is the site retrieval process. To retrieve sites, created sites are announced to trackers on creation and every twenty minutes. The tracker returns a list of all the nodes that it knows to hold the site. Nodes requesting a site contacts a tracker with the BIP-32 address. Nodes also have "Peer Exchange" to retrieve sites, which, instead of relying on central trackers to find sites, relies on groups of peers collaborating to share and update each other on files.

Once a peer finds a node hosting a site through the previous methods, the peer asks for the content.json file via TCP/IP or Tor. The peer validates the content.json by verifying the site owner's cryptographic signature and the site's BIP-32 address. Once the peer has finished validating the json file, it then downloads the rest of the file and verifies that they are downloaded correctly with a SHA512 hash from the json file. After downloading, the peer then announces itself to a tracker saying that it also serves the site.

To update existing files, a site owner signs a new content.json file and begins sending that file when it is contacted by visitors. Visitors check that the file is valid and newer then the ones that the visitors might have and then downloads the changed files. Additionally, peers can send a listModified message to other peers to retrieve only the content.json file to see if any files have been modified.

5 Comparison

Now with a brief understanding of FreeNet and ZeroNet, we begin discussing and comparing the two networks. The comparisons are organized first by a general feature comparison. Next, we compare by user experience, which includes factors such as popularity of the networks, ease of usage, content and support. Following this, we discuss the general creation process

of creating two websites on the respective networks. Lastly, we end with a brief comparison of how the two networks are being blocked or censored in several jurisdictions.

5.1 Feature Comparison

There are some important distinctions between the two networks, mainly is that the networks are focused on two different purposes. FreeNet's features compliment the idea of being able to insert a file or website into the network and then have that user immediately leave the network. The file or website will be propagated around the network. On the other hand, ZeroNet requires at least one peer to have the site and be online for a site to be accessible. For both of these networks, files and sites do not last forever in the network. If a file is not called upon frequently enough in FreeNet, then eventually the file will be forgotten by the system because of its least recently used cache. With ZeroNet, if all hosts of a site leave the network then there will be no way to download a site. More generally, FreeNet and ZeroNet are expressly different in the types of files they are to have on the network. ZeroNet is designed for websites and not general files. In fact, Wang et al.[16] considers the platform to be a site publishing platform rather than a file distribution network. FreeNet on the other hand is in fact a full fledged information storage system. Another fundamental difference is the privacy of publishers in the two systems. In FreeNet, privacy is assured as the file is distributed across the network and each node having a random chance to declare that they were the source of the publishing. This obscures which node originally published a file. In ZeroNet however, privacy for a publisher can only be obtained after a large amount of peers start hosting the site. If a new file is published and it only has one peer then it is relatively safe to assume that the peer was the publisher of the file.

5.2 User Experience

FreeNet's user experience is fairly nice as well. On first launch, FreeNet prompts the user with a series of questions on how large they would want their local datastore to be as well

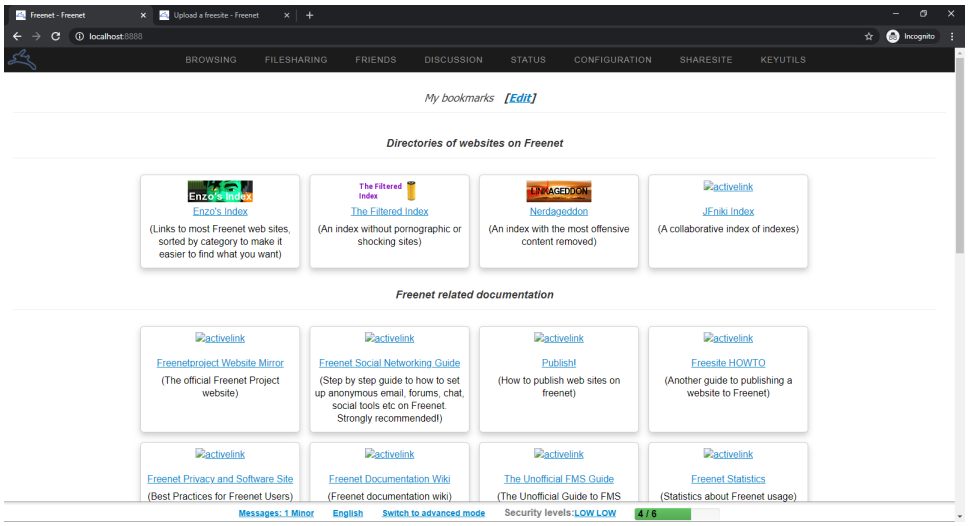

Figure 1: FreeNet landing page

as questions on whether or not they have/would like to implement a data cap. Providing these controls up front during startup allows user's to decide on how much they wish to contribute their local resources rather than letting the network have unfettered control over their machine and connection. After this setup wizard is completed, users are redirected to a landing page where they can learn more about FreeNet, change settings, or quickly jump to important sites. An example of the landing page is shown in figure 1. By providing user's with a easy to understand and helpful landing page like this, user's can quickly find popular websites before downloading them or do most of their functions such as creating a website through an intuitive UI rather than a command line. FreeNet's settings, download manager and other features are mainly hosted at the top bar of the landing page. Along terms of popularity and support, FreeNet has enjoyed long lasting support considering the project started in 2000. However, this age shows as users come and go, dead links and old information is abundant and still the main source of information with no idea on whether it is still correct or not. Fortunately, because of the large user base it is not difficult to have questions answered or to find more recent information. One example was trying to find a pre-compiled binary of the jSite tool which is used in creating FreeNet websites. Downloading the tool from the official FreeNet website results in a dead link and the only way to actually

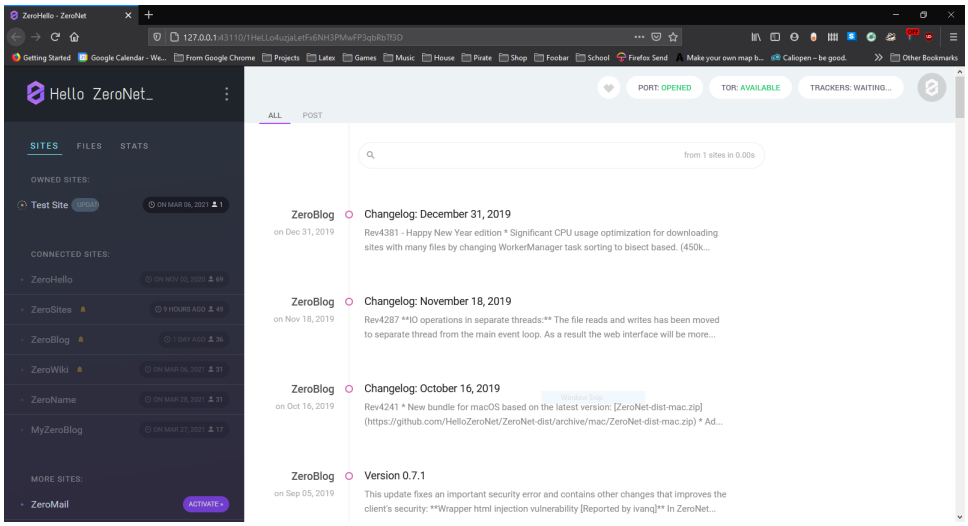

Figure 2: ZeroNet landing page titled "HelloZero"

download a binary was from the developer's personal FreeNet website. Even then, the binary was out of date and needed to be updated by compiling from source.

ZeroNet also has a solid user experience. On startup and first launch, the default landing page is a page called HelloZero (see figure 2). At this website, users can create new sites, find popular ones, view, manage, edit and delete local files relating to other sites and many other functions. Users can also see exactly which sites are being downloaded, and see statistics on bandwidth usage. The landing page of ZeroNet's is, aesthetically and functionality-wise, better in my opinion compared to FreeNet's. Visually, the site looks cleaner and the functions for most of ZeroNet are all hosted on this single landing page. Each site also has a HelloZero button that hovers in the top right which when clicked will bring user's back to the landing page. The HelloZero button also can be dragged leftward to open a site settings menu. This site settings menu lets site owners handle settings as well as providing statistics on the number of files, the bandwidth and number of peers. In terms of popularity and support, in my opinion it seems that the network is fading in these terms. Updates to the network have stopped with the last major one being in 2019. Most posts from the official ZeroNet accounts on social media have also stopped been updated with the last posts on Twitter and Facebook being from 2018. With this lack of support from the developer, it also seems

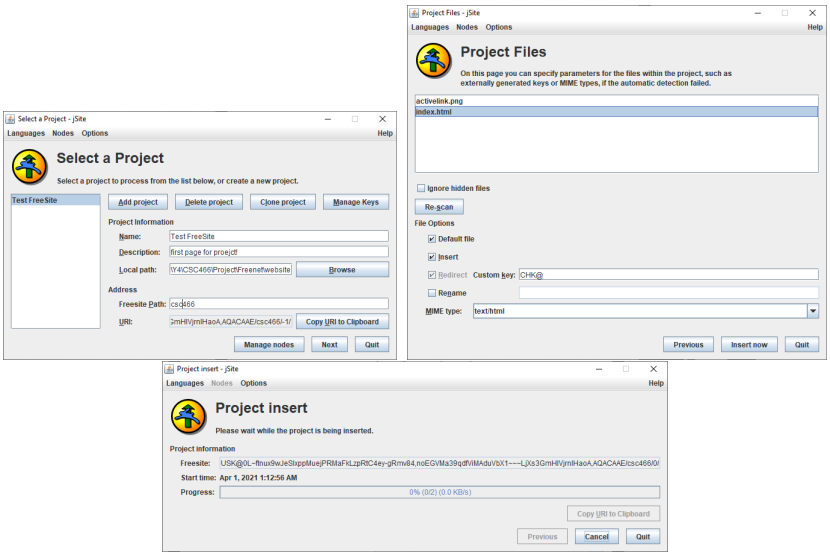

Figure 3: FreeNet website insertion via jSite

inevitable that eventually the user base will fade away as well.

5.3 Website Creation

For FreeNet, website creation is a little more difficult but ease of use can be achieved through some community plugins and tools. One recommended way to create a website is through the use of a tool called "jSite"[17]. To create a site, there are some recommended steps to do first. One recommended step is to include an "Activelink"[18] image to your own website which helps pre-fetch the website whenever that image is accessed elsewhere. If a user is able to view this image then they know that the website that is linked to that image is already loaded. The image must be 108x36 in pixel size and be a png. After creating that image, one simply creates a generic HTML page and can now use the jSite tool. Using jSite, user's create a new project, fill out the forms and then can simply insert the HTML file and Activelink image into the network. Figure 3 gives an example of creating a website using jSite.

For ZeroNet the website creation process is fairly easy and streamlined. There are two supported ways to create a site[14], the first way is to create through the HelloZero web

interface and the other is to manually create by running a command on the ZeroNet application in the command line. To create a site through the HelloZero interface go to the top left and click on the three dots menu then select the "Create new, empty site" option. In figure 4, which shows the three dots menu, we can see the option near to the bottom of the list. After the site is created, the site will appear on the left hand menu bar of the HelloZero. By

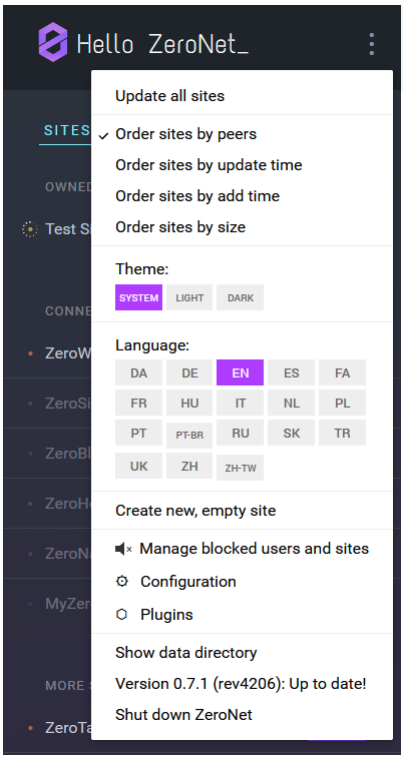

Figure 4: HelloZero three dot menu listings

clicking on this listing in the menu, it will bring a user to the newly created site. Opening the site settings menu by dragging the HelloZero button and then scrolling to the bottom of the menu shows the sign and publish options. ZeroNet will automatically handle the private key when using this method. Figure 5 shows the options in the menu.

To use the command line, a user runs these following commands:

```
zeronet.py siteCreate
zeronet.py siteSign <public_key>
zeronet.py sitePublish <public_key>
```

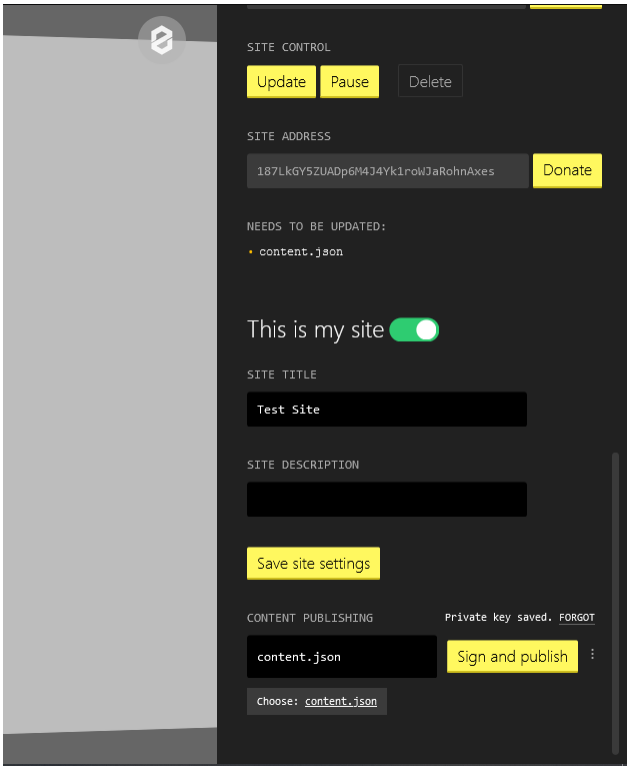

Figure 5: The ZeroNet site settings menu

to create, modify and update a site. After running the first command, ZeroNet creates a site and the program returns a public and private key pair as well as the files for a basic website in a subfolder labelled by the public address. The basic website files are basic html and javascript. Once a user finishes editing their website they run the next two commands to sign and publish the changes. These two commands will require the user to enter the private key each time. The two created websites are actually on their respective networks and they can be accessed [2] through the URLs noted in appendix A.

[2]Accessing the websites requires active connections to the respective networks; in the case of ZeroNet, the site has no Peers so one will have to wait until I connect to the network to seed the site

5.4 Censorship

In terms of actions taken against these two networks, both FreeNet and ZeroNet has faced censorship. While it is difficult to block the networks as they are designed to be resilient against such types of censorship, this has not dissuaded entities from doing so. The Chinese national firewall has blocked the FreeNet website, where one would normally download the software to connect to FreeNet for many years and in 2005 was known to have blocked an older protocol and some French internet service providers are known to block Peer-to-Peer networks in general[19]. ZeroNet also faces similar issues with the ZeroNet website being blocked in China as well[20]. Additionally, the Russian government is seeking to block "Darknet" technologies with the list including both FreeNet and ZeroNet[21].

6 Conclusion

In this paper, we discussed Peer-to-Peer networks, the difference and advantages between them and conventional client-server networks. Next we focused on two such networks, FreeNet and ZeroNet which were then described in detail, focusing on the networks architecture and protocols. Afterwards, the two networks were compared, focusing on the feature set of the two networks, the user experience which includes popularity, ease of usage, and support, and lastly the experience of creating two websites for the respective networks. Finally the comparison ended with a discussion and note on the censorship of these networks in various jurisdictions.

6.1 Future Work

In terms of future work, there is still much to be done in relation to these specific two networks as well as other privacy and anonymity focused technologies. For FreeNet and ZeroNet comparisons, the next step is clearly to step up from a qualitative approach to a quantitative approach, detailing important aspects such as the latency, loss, jitter, bandwidth

usage, number of uploads or downloads, and number of incoming or outgoing connections of the two networks. Such an approach could be done by hosting nodes on two machines at different addresses and measuring the differences.

In terms of other related networks, much work is still to be done on researching newer technologies. ZeroNet, released in 2015, is relatively old as technology is moving at an ever increasing pace and it has a large lack of research done. Even newer technologies, are most likely hardly even mentioned. It is important to document these new technologies so as to gain an understanding of how problems are being approached and to see how an application of an idea functions under real usage cases. With hope, these lessons learned from upcoming technologies can be used to create a better future internet.

A URLs for Created Websites

FreeNet: `http://localhost:8888/USK@0L~ftnux9wJeSIxppMuejPRMaFkLzpRtC4ey-gRmv8`
`4,noEGVMa39qdfViMAduVbX1~~~LjXs3GmHIVjrnIHaoA,AQACAAE/csc466/0`

ZeroNet: `http://127.0.0.1:43110/187LkGY5ZUADp6M4J4Yk1roWJaRohnAxes`

B References

[1] I. Napster. "Napster." (1999), [Online]. Available: https://web.archive.org/web/20000301102915/http://www.napster.com/.

[2] The Gnutella Developer Forum. "The annotated gnutella protoocl specification v0.4." (2000), [Online]. Available: http://rfc-gnutella.sourceforge.net/developer/stable/index.html.

[3] S. Saroiu, K. P. Gummadi, and S. D. Gribble, "Measuring and analyzing the characteristics of napster and gnutella hosts," *Multimedia Systems*, pp. 170–184, 2003. DOI: 10.1007/s00530-003-0088-1. [Online]. Available: https://doi.org/10.1007/s00530-003-0088-1.

[4] J. Ritter. "Why gnutella can't scale. no, really." (2001), [Online]. Available: https://www.cs.rice.edu/~alc/old/comp520/papers/ritter01gnutella-cant-scale.pdf.

[5] B. Cohen, "Incentives build robustness in bittorrent," 2003. [Online]. Available: https://www.bittorrent.org/bittorrentecon.pdf.

[6] P. Wuille. "Bip32: Hierarchical deterministic wallets." (2012), [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki.

[7] jrandom (Pseudonym), *Invisible internet project (i2p) project overview*, Aug. 2003. [Online]. Available: https://geti2p.net/_static/pdf/i2p_philosophy.pdf.

[8] ——, "I2p: Introduction." (2003), [Online]. Available: https://geti2p.net/en/docs/how/tech-intro.

[9] The Tor Project. "Tor." (2021), [Online]. Available: https://www.torproject.org/.

[10] Y. Hu, F. Zou, L. Li, and P. Yi, "Traffic classification of user behaviors in tor, i2p, zeronet, freenet," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 418–424. DOI: 10.1109/TrustCom50675.2020.00064. [Online]. Available: https://ieeexplore.ieee.org/document/9343185.

[11] N. Negi, "Comparison of anonymous communication networks - tor, i2p, freenet," *International Research Journal of Engineering and Technology*, vol. 4, no. 7, pp. 2542–2544, 2017.

[12] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability Berkeley, CA, USA, July 25–26, 2000 Proceedings*, H. Federrath, Ed., pp. 46–66, 2001. DOI: 10.1007/3-540-44702-4_4. [Online]. Available: https://doi.org/10.1007/3-540-44702-4_4.

[13] T. Kocsis. "Zeronet." (2015), [Online]. Available: https://zeronet.io.

[14] ZeroNet. "Zeronet documentation." (2020), [Online]. Available: https://zeronet.io/docs/.

[15] ——, "Zeronet: Decentralized web platform using bitcoin cryptography and bittorrent network." (2015), [Online]. Available: https://zeronet.io/files/ZeroNet_Presentation.pdf.

[16] S. Wang, Y. Gao, J. Shi, X. Wang, C. Zhao, and Z. Yin, "Look deep into the new deep network: A measurement study on the zeronet," V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, and J. Teixeira, Eds., Cham: Springer International Publishing, 2020, pp. 595–608, ISBN: 978-3-030-50371-0.

[17] David Roden (Bombe). "Jsite." (2020), [Online]. Available: https://github.com/Bombe/jSite.

[18] S. Oliver. "Activelink." (2017), [Online]. Available: https://github.com/freenet/wiki/wiki/Activelink.

[19] The FreeNet Project Inc. "Freenet: Help." (2021), [Online]. Available: https://freenetproject.org/pages/help.html.

[20] P. Hill. "China blocks access to zeronet website and tracker." (2017), [Online]. Available: https://www.neowin.net/news/china-blocks-access-to-zeronet-website-and-tracker/.

[21] A. Baydakova. "Russia seeks to block 'darknet' technologies, including telegram's blockchain." (2020), [Online]. Available: https://www.coindesk.com/russia-seeks-to-block-darknet-technologies-including-telegrams-blockchain.