Spatial Path Selection and Network Topology Optimisation in P2P Anonymous Routing Protocols

Aleksandar Tošić[1,2,*] and Jernej Vičič[1,3]

[1]*University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies, Koper, Slovenia*
[2]*InnoRenew CoE, Izola, Slovenia*
[3]*Research Centre of the Slovenian Academy of Sciences and Arts, The Fran Ramovš Institute, Ljubljana, Slovenia*
E-mail: aleksandar.tosic@upr.si; jernej.vicic@upr.si
[*]*Corresponding Author*

Received 21 July 2021; Accepted 04 October 2021;
Publication 23 November 2021

Abstract

To anonymous internet traffic, many popular protocols route traffic through a network of nodes in order to conceal information about the request. However, routing traffic through other nodes inherently introduces added latency. Over the past two decades, there were many attempts to improve the path selection in order to decrease latency with little or no trade-off in terms of security, and anonymity. In this paper, we show the potential use of geo-sharding in decentralized routing networks to improve fault-tolerance, and latency. Such networks can be used as a communication layer for Edge devices computing huge amounts of data. Specifically, we focus our work on Low Latency Anonymous Routing Protocol (LLARP), a protocol built on top of Oxen blockchain that aims to achieve internet privacy. We analyse the existing network of Service Nodes(SN), observe cloud provider centralisation, and

Journal of Web Engineering, Vol. 21_1, 97–118.
doi: 10.13052/jwe1540-9589.2115

propose a high level protocol that provides incentives for a better geographical distribution mitigating potential cloud provider/country wide service dropouts. Additionally, the protocol level information about geographical location can be used to improve client's path (the string of nodes that will participate in the transaction) selection, decreasing network latency. We show the feasibility of our approach by comparing it with the random path selection in a simulated environment. We observe marginal drops in average latency when selecting paths geographically closer to each other.

Keywords: Lokinet, anonymous routing protocol, geo-sharding, fault tolerance, oxen.

1 Introduction

Since its inception, the internet went through many protocol, and infrastructural improvements that facilitate toady's scale. However, in the transformation process, not a lot of care was given to the privacy. Additionally, more sophisticated tooling leveraging advances in artificial intelligence, data science, and metadata analysis is constantly being created. Coupled with the federated centralization of the infrastructure, users privacy on the Internet is constantly being threatened. An important privacy consideration is the ability to conceal ones traffic and remain pseudonymous. To address these issues, routing protocols have been built, which route traffic through the network in an attempt to make the traffic indistinguishable, and hard to track.

The most known privacy preserving routing frameworks, such as The Onion Router (Tor), Invisible Internet Project (I2P) or Low Latency Anonymous Routing Protocol (LLARP), rely on service nodes for anonymization and obfuscation of transactions. These are plain computers with routing software operated by stakeholders with financial incentive or enthusiasts with no direct incentive. Most of these nodes are hosted by public computing providers that operate in data centers. Bigger providers can usually afford lower prices, thus concentrating routing nodes to a small number of data centers. Such concentration of routing nodes presents a single point of failure – SPOF, or at best, reduces fault tolerance of the network. We address this problem by providing protocol level incentives to decentralize the geographical location of routing nodes.

The second problem addressed by the presented research was lowering the average latency. Existing protocols have experimented with various path selection algorithms in an attempt to find the best trade-off between latency,

and security. The most secure path selection is obviously completely random. Improving the latency inherently structures and with traffic analysis potentially reduces privacy, and security.

The paper aims at showing the potential use of geo-sharding in decentralized routing networks to improve fault-tolerance, and latency. Settings with edge devices processing big data (or at least computing huge amounts of data locally) can use such networks as a communication layer. All communication about computations remains anonymous which in some cases is a prerequisite.

We focus our solution to the presented problems on incentive-based privacy preserving routing frameworks and propose a change in incentive distribution based on geo-sharding that takes into account the geographical proximity of the nodes. We use Lokinet as an example of an incentive based onion router, and compare traditional random path selection to ours by simulating the routing of requests on Lokinet.

The hypothesis of the presented research was that the proposed division of incentives will distribute the routing nodes evenly through geographical space. The presented simulation results show that geographically aware path selection can provide considerably lower latency. However, the performance is dependant on the even geographical distribution of routing nodes.

The rest of the paper is structured as follows: Section 2 introduces the reader to the research area and explains the motivation behind the research, Related works are presented in the next section, followed by the paper's main contribution in Section 4. In Section 4.2 the authors present a corollary that the distribution of the routing nodes will be more dispersed with the introduction of incentives based on geographical position of the nodes, the paper closes with discussion and further work.

2 Background And Motivation

The pioneers of the Internet were attempting to create a decentralized network i.e., ARPANET. The Internet has gone through a steep evolutionary process and while it serves as the backbone infrastructure that empowers the world, it has also lost some of the properties, namely decentralization, and privacy. Arguably centralization came with many benefits and was mostly driven by the search for efficiency. On the physical layer, the network is hierarchically clustered around gateways, the need for computing resources have accelerated centralization even further with cloud computing and efficiency obtained by centralizing server infrastructure. Cloud providers have high incentives to

centralize their infrastructure in data centers ranging from building space, and cost, proximity to high bandwidth Internet gateways, affordable electricity and other energy sources, and finally regulation of the jurisdiction [11]. The same way *Privacy* is becoming increasingly important with the amount of data, and metadata collected, and stored in these data centers. Internet companies have been gathering data, applying advanced data analysis in an effort to improve their products and services but at the same time raising valid concerns about user privacy. Additionally, the data is congested both physically (cloud providers) and institutionally by access being held by a few IT conglomerates. In recent years we witnessed to many privacy violations. Some were accidental data leaks, others violated privacy due to negligence, and in some cases breaches of security protocols. Concerns about the amount of trust put into cloud providers, and technology enterprises are valid. Moreover, from a cybersecurity point of view, centralization introduces single point of failures(SPOFs), and reduces fault tolerance.

The need for privacy, and decentralization was emphasized even further with Europe's general data protection regulation (GDPR) that stresses the need for business processes that handle personal data to be designed and built safeguard data through pseudonymization or even full anonymization where possible. Many countries followed with their own privacy preserving legislation such as Brazil's LGPD,[1] Australia's Privacy Amendment,[2] and Japan's Act on Protection of Personal Information[3] to name a few. The details of these regulations are not as important as the intent to preserve user privacy. However, there are many issues related to enforcing, and policing. The lack of transparency makes it hard to detect violations.

Internet privacy is a complex issue that can hardly be solved systemically. It is very unlikely that the backbone infrastructure will be decentralized and your IP address suddenly stop being mapped to identities. There is very little chance computing power and storage capacities will willingly decentralize without added incentives, and even less likely that private businesses and enterprises will provide transparency when handling private data. A big part of privacy is network identity (IP address), which is revealed to any Internet service clients interact with. Analysis of metadata poses a new threat from both a security perspective (an attacker breaking social ties to the user) as

[1] LGDP Brasil: https://www.lgpdbrasil.com.br/

[2] Privacy Amendment (Public Health Contact Information) Bill: https://www.aph.gov.au/Parliamentary_Business/Bills_Legislation/bd/bd1920a/20bd098#_Toc40190704

[3] Act on the Protection of Personal Information: https://www.ppc.go.jp/files/pdf/APPI_The_Every_Three_Year_Review_Outline_of_the_System_Reform.pdf

for privacy where an IP address can be tracked back to personal data such as geo-location, and in some cases even identity.

2.1 Internet Privacy

Today's Internet infrastructure does not guarantee any identity privacy. IP addresses are known to leak private information to a remote party [2]. Even the newer protocols being added make poor attempts at providing more privacy. As an alternative, many attempts at building overlay networks on top of existing infrastructures were made in order to preserve privacy. The primary focus of mixnets [6] and overlay networks is to conceal the network identify of clients. This is primarily done by either routing network traffic within the overlay network, or by mixing packets and bouncing them through proxies.

Mixnets [6] make use of a network consisting of a set of interconnected nodes referred to as mixers. Clients make network requests and send them to the mixers. The mixer then gathers requests from multiple clients, shuffles them, and sends them back out in random order to the next mixer and eventually to the destination of the request. In the most general form first introduced by David Chaum [6], a mixnet uses public key cryptography to seal massages in layered cryptography later refereed to as onion encryption. A message m in a 3 hop mixnet is constructed as: $m = E_pk1(h_1, E_pk2(h_2, E_pk3))$ where m is the onion encrypted message, h_i is a vector of ordered hops, and E_pki is the corresponding encryption the client performs using $i-th$ node's public key. At each hop, h_i performs $D_ski(m)$ to decrypt a layer of the onion, revealing the address of $h_i + 1$. However, mixnets can also conceal network identity, without encrypting messages as long as each mixer has sufficient requests to shuffle, making tracing the origin/destination increasingly harder as the number of hops increases. However, increasing the number of hops is a trade-off between the strength of the security and network latency. Overlay networks are known to introduce higher latency which theoretically is the sum of all latencies plus the sum of computation time needed at each hop in a given path, in practice even more.

2.2 Privacy Centric Routing Protocols

In general, most privacy routing protocols have fundamental shared components from an architectural point of view. The use of layered encryption is the most used solution to protect privacy of messages being transmitted in a way that every routing node only has partial information that is insufficient to

reconstruct the message, or the origin, and destination. However, all solutions require a peer to peer (P2P) that executes the protocol, and routes messages. The adversary models therefore mostly focus on sybil type attacks [9], where a malicious actor operates more then one node in the network. All types of attacks can be generalized to what information the adversary can learn about the network, network's traffic, and routing in case it runs a modified version of the protocol on multiple nodes. The value of privacy centric protocols is therefore in the P2P network of nodes that service the network by routing messages.

The motivation for developing, and using these networks can be divided into a few main categories:

1. ***Censorship resistance:*** Centralized networks inherently introduce single point of failures (SPOFs). These central points can also be used to censor communication. This can be addressed by introducing two main mechanisms, namely pseudonymity, or even anonymity where appropriate, and content encryption. The goal of the first is to prevent targeted censorship of a person/entity by not revealing their identity while the latter is used to prevent censoring specific information by protecting the contents of messages.
2. ***Personal privacy:*** Provide tools to protect privacy to those who value their privacy. This motivation has strengthened considerably over the past few years with news of popular platforms/companies violating user privacy in various ways.
3. ***Fear of retribution:*** Mostly aimed at providing a tool that enables users to voice their opinion without fear of retribution such as activist groups, whistle-blowers, and unofficial leaks.

2.3 Path Selection – State of the Art

Path selection is a process in which clients non-deterministically select k routing nodes to construct a circuit. The default path length of three hops states a reasonable trade-off between security and performance. Path selection is important as it impacts both security, and latency. A predictable path selection algorithm can potentially allow an adversary to de-anonymize the clients traffic by carefully aligning malicious nodes to be selected by the client. However, a perfectly random path selection can increase latency. Ideally, all clients in the network would select paths uniformly without a high performance penalty. Path selection algorithms have been studied to a great extent, in an attempt to find a balance between anonymity, and latency [24].

2.4 Path Selection Based on Geographical Information

The reason geographically inspired path selection is important for improving network latencies is due to centralization of nodes running on clusters. Moreover, even cloud providers are incentivized to centralise their servers [7]. Most relay nodes in routing networks are run as virtual private servers on popular cloud providers. Additionally, there is a persistent economic incentive for users to choose the cheapest provider inherently centralizing the network even further. Ideally, a protocol would incetify the opposite.

2.5 Analysis of Current State

In this section we take a closer look at some metrics of LLARP, which is further presented in Section 3.3. LLARP, and the reference implementation Lokinet [12]. We use this reference implementation of LLARP as our use case in this paper as it is the newest and most advanced protocol with a working implementation. All data presented was acquired by building and indexed database of the entire blockchain, which holds all information regarding the service nodes such as their public keys, IP address, activation, decommission, and de-registration events. Additionally, the unique properties the protocol has due to the blockchain integration allows for our main contribution to be adopted without risk of sybil attacks. Lokinet has been running steady for more then two years at the time of writing. It has undergone many updates and upgrades to reach a level of stability. Due to nodes being rewarded for their routing services, the node operators are inclined to reduce operating costs thereby maximizing profits. As a consequence the network converges to centralization on the level of virtual private server (VPS) providers. The same issues are present in other routing protocols, albeit to a lesser extent due to a higher level of adoption. Figure 1 shows a per country distribution of Lokinet service nodes at the time of writing. values are ordered in a descending order from left to right. The locations were obtained by mapping their IP addresses via a public location mapping database (ipstack). The concentration of nodes in certain countries could be coincidence, due to a high adoption of the protocol in those countries, or simply the economic incentives of operators to rent VPS from providers who's infrastructure is clustered in Germany, and France.

However, by mapping the IP addresses to known VPS provider IP ranges, it can be clearly seen that the geographical clustering is very well correlated with the VPS providers. Figure 2 shows the distribution of Lokinet service nodes across the top 20 most used hosting providers. The most used VPS

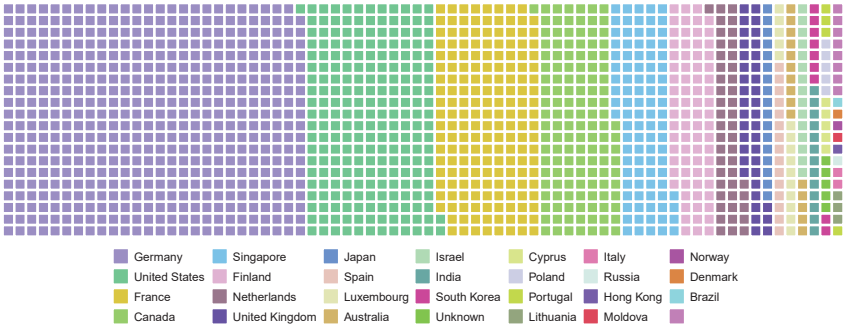

Figure 1 A waffle chart of all service nodes running the Lokinet protocol and their geographical location per country.

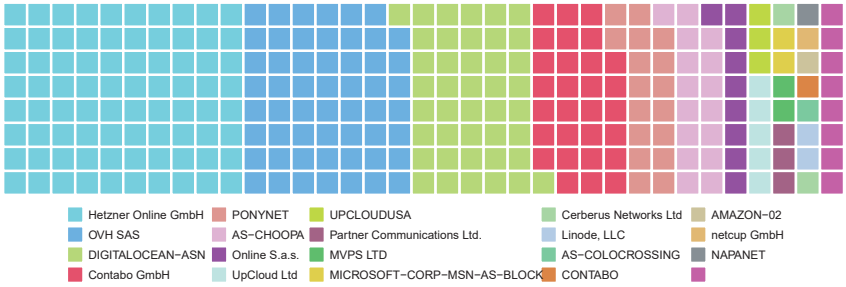

Figure 2 A waffle chart of all the top 20 virtual server providers by the number of Lokinet service nodes being hosted. The figure shows high centralization of the network on the service provider level.

provider is Hetzner Online GmbH,[4] which has by far the largest server infrastructure built in Germany, followed by OVH SAS,[5] which is the second biggest European VPS provider from France. Additionally, both Hetzner, and OVH are the cheapest VPS providers in Europe with prices as low as 3€/month for a cloud instance. Besides decreasing the fault tolerance of the entire network in case a larger outage occurs at any of the most used VPS providers, the security of the network is greatly decreased opening it up to a series of attack vectors such as timing attacks [26] or traffic analysis [16].

From the analysis of the current state we can draw the following conclusion:

1. Network nodes are mostly hosted on rented virtual private servers.

[4]Hetzner: https://www.hetzner.com/
[5]OVH: https://www.ovh.com/world/

2. The network is clustered due to centralized infrastructure of service providers.
3. The node operators prefer cheaper service providers over decentralization.

3 Related Works

In this section we review existing, and popular networks that all rely on different routing protocols used to construct overlay networks that facilitate anonymous and private message passing.

3.1 The Onion Router (Tor)

Tor is arguably the most popular anonymous routing network. The network maintains a high level of censorship resistance, and preserves internet privacy. The network is composed of different types of nodes, namely guard nodes, relay nodes, and exit nodes. A client then selects a randomized 3-hop path where the first node is a guard node, the second a relay, and finally an exit node. It then builds an *onion*, a data structure using layered encryption, and sends it to the guard node. The circuit routes the message thereby protecting the identity of the source, and destination.

However, the network is not completely decentralized. Tor relies on a group of servers called directory authorities. These nodes are operated by volunteers that are close to the Tor Foundation [8]. These directory authorities act as trusted reporters of the state of nodes in the network. When connecting to the network the first time, the client connects to one of the hard-coded directory authorities and learns about the consensus. Even though the ability to simply block all the relay nodes on the list was addressed by introducing bridges [14], the centralized registry decreases fault tolerance. An example was a credible threat received in 2014 threatening to block access to directory authorities [20].

3.2 Invisible Internet Project (I2P)

I2P takes a different approach by replacing the directory nodes with a distributed hash table (DHT), which is a distributed system that provides a lookup service of key-value pairs, to ascertain the network state. It uses garlic routing [19], which is an extension of onion routing and enables multiple messages to be bundled together, and uses unidirectional tunnelling. I2P also supports both TCP and UDP traffic capturing a much larger subset

of the Internet. Due to unsteady development, there is no formal support for exit nodes making I2P very limited for accessing the wider internet in an anonymous way. Additionally, most clients connecting to I2P also become routing nodes. This can be very limited when path selection is slow, and bandwidth is limited by the slowest router on the path. In part, I2P solves this by having a packet switches protocol unlike Tor, which is circuit-switched. Instead of establishing one tunnel and maintaining it, I2P creates multiple paths that can be used simultaneously. This gives I2P the ability to route around potential network congestion and node failures efficiently.

However, both I2P and Tor have not fully mitigated Sybil attacks. An adversary with a large number of relays can perform temporal analysis (timing attacks) that can greatly reduce privacy [27]. This is possible since both protocols rely on altruistic intention from the community to operate relays on their own cost, and a sufficiently motivated adversary can easily carry a Sybil attack by renting server instances from cloud providers.

3.3 Low Latency Anonymous Routing Protocol (LLARP)

LLARP, and the reference implementation Lokinet [12], is a newer protocol that inherits some properties from both aforementioned protocols while providing a solution to their drawbacks. Lokinet uses a custom blockchain that replaces Tor's directive authorities and I2P's DHT by storing the registry as blockchain staking transactions. Staking transactions are transactions submitted by relays called service nodes upon entering the network. Additionally, the protocol sits on the networking layer of the OSI standard allowing for a wider range of network protocols to use it as an anonymization layer. Traffic is routed through the service node network using onion routing. Network utilization is decreased by using packet level switching as opposed to tunnel based routing, because no additional time is spent building a tunnel and still achieving the same basic functionality. The main advantage is the intricate relationship between the Oxen blockchain, and service node operators. Operating a service node comes at cost thereby incentivizing operators to support the network with resources needed, and at the same time making a Sybil attack financially infeasible. This is done by rewarding nodes for their service, and penalizing them on a protocol level for bad behaviour [12]. At the time of writing, there are 1709 nodes in the network, with each node having staked a minimum of 15,000 oxen coins worth almost $30,000.[6]

[6]Coincodex: https://coincodex.com/crypto/loki/

3.4 Comparison of Path Selection Protocols

All three protocols rely on a network of nodes that relay messages to decouple the identity of the source and destination. However, in this paper we use Lokinet as our use case. Path selection can be very important in optimising traffic flow to both minimize latency and avoid network congestion. In Tor's initial path selection algorithm included categorizing relays by their bandwidth and up-time in an effort to build paths more frequently on nodes that can process the traffic, and are considered reliable making the network less prone to sybil attacks. This made the circuits more stable but it compromised anonymity as only a small selection of nodes were periodically chosen. An improved version of the algorithm prioritizes uniform path selection, improving the anonymity at the expense of speed and fault tolerance.

Relays in I2P actively probe one another to obtain a performance measure, which is used in path selection. Those include time to execute queries, tunnel construction success rate, and reliability of nodes. Nodes are categorized into high capacity, fast, and standard. The random path selection of I2P is biased towards faster nodes [25]. Due to the dynamic nature of performance metrics, even a low latent path can become congested. I2P addresses this in part by building new tunnels, instead of maintaining them for longer periods of time as in Tor.

On the other hand, path selection in Lokinet is done by randomly choosing nodes by their public key. This provides the highest anonymity. In general, completely random path selection would perform much slower [23]. However, service nodes in Lokinet are financially incentivized, and compensated for their operating costs making random path selection favor anonymity, and addressing bottlenecks on the network protocol level.

3.5 Limitations of Existing Approaches

All observed path selection implementations focus on uniform selection of paths to prevent potential tracing and improve anony-mity [18]. However, focusing on a single criteria leaves room for optimisation on the performance side. Many path selection improvements were proposed that take into account latency such as selecting middle node based on latencies between routers [22]. Another method suggested was to assign nodes a capacity as the median value of the peaks other nodes measured against it. However, the proposed solutions introduce additional burden and resource consumption on the entire network. Additionally, they are prone to sybil attacks, where an adversary can skew the metrics by simply advertising untrue measurements

of latency to force clients to build paths where all nodes are those controlled by the attacker deanonymizing the traffic. These issues are in part solved in Lokinet, assuming all nodes are of comparable capacity both in terms of computing, and bandwidth.

Using geographical location of relay nodes was proposed by Akhoondi et al. [1]. The authors propose a modified client-side path selection algorithm that uses existing databases to map IP addresses to geographical locations. In order to avoid clustering of paths, and potential path centralization they propose a simple clustering of nodes and selecting a path that selects a node within each cluster. This improves the latencies considerably and minimizes potential risks for an adversary to setup and operate nodes within all three geographical region. However, the proposed implementation centralized the network by querying a central database to map IP addresses to geo-location. Additionally, this creates a SPOF, where an attacker only needs to compromise the database being queried to falsify locations of nodes, forcing clients to build paths on malicious nodes only.

4 Methodology

This section presents paper's main contribution, a method that enables lowering latency through node selection based on geolocation. Privacy issues are properly addressed.

4.1 Geo-sharded Path Selection

The geographical closeness of routers empirically greatly improves latency and jitter. However, constantly routing traffic over geographically closer routers potentially reduces anonymity. Instead of using external services to map IP addresses to their estimated locations, we propose a protocol that uses geohashing to introduce approximate locations of routers without exposing their privacy. Geohashing, invented by Morton [15] and impemented and put in public domain by Niemayer [17] is a known technique of mapping the planet to a grid, and representing points within the grid as base64 strings (in this context called hashes, although geohashing breaks almost all properties of hash functions). To achieve this, we transform the continuous latitude and longitude space into a hierarchical discrete grid using a recurrent four-partition of the space. The most important property of geohashes is preservation of spatial hierarchy in the code prefix. In the context of relay nodes, a node can encode the latitude, and longitude coordinates to a geohash,

it then advertises as many bits of the hash it chooses too in order to preserve the privacy. Fewer bits will result in a greater bounding box description of the location. In Lokinet, IP changes are penalized, binding the geohash to the IP address will be sufficient to prevent most attacks. A basic path selection algorithm is outlined in Algorithm 1.

Algorithm 1: Naive path selection: construct a geohash from location, add vertices to path from bounded region, increase bounded region, until the length of the path is long enough.

Result: Naive path selection
`BuildPath`$(length, latitude, longitude)$
$g \leftarrow geohash(latitude, longitude)$; `// Construct a geohash from location`
$path = \{\emptyset\}$;
while $|path| \neq length$ **do**
  $path \leftarrow RelayMap(g)$; `// A hierarchical map mapping geohashes to IP addresses`
  $g \leftarrow g.substring(1)$; `// Reduce suffix to increase search bounding box`
end
return;

However, this has common shortcomings with other methods that focus on location aware path selection. Specifically, having clients build paths closer to them makes it less secure. An adversary can rent multiple virtual private servers in a cloud provider near the client forcing it to select a path in which all nodes are malicious. This enables the attacker to track the traffic linking the source to the destination. There is a trade-off between reduced latency, and centralization of nodes in the path. To address this concern we introduce a client side parameter that indicates the client's willingness to reduce security in favor of lower latency. In the most secure form, the algorithm first selects a subset of nodes that fall within the bounding box defined by the first character of their geohash, this roughly translates to a bounding box of 5000 square kilometers. It then randomly selects a node in the subset as the first on the path. To obtain the next node on the path, the client increases the accuracy of the query by adding an additional character thereby limiting the spacial query to a bounding box of approximately 1250×625 kilometers. This hierarchical type path selection greatly reduces the chance of selecting a path of only malicious nodes operated by an adversary since it is sufficient that only one node in the path be honest [4]. The algorithm

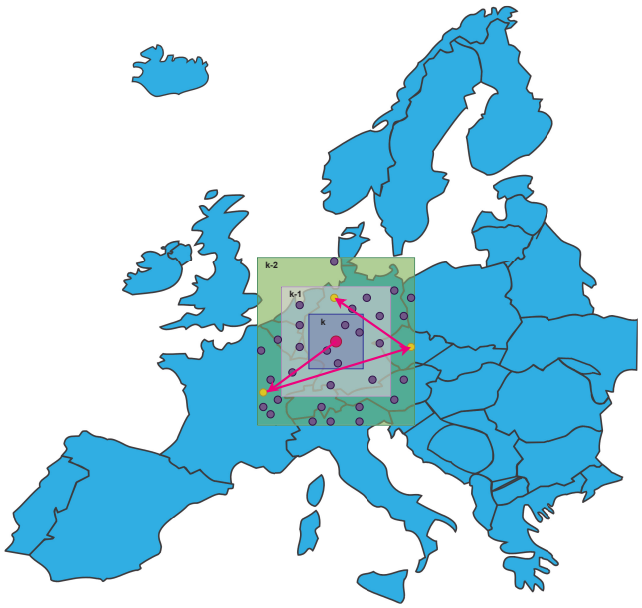

Figure 3 Graphical representation of hierarchical path selection based on geohashing. Parameter k describes the number of characters the client uses for querying other nodes in the proximity. Greater k values represent a smaller bounding box.

is simple and requires an additional $\mathcal{O}(max(|S_k, S_{k-1}|))$ to compute the relative set complement of each hierarchical query. By iterative decrements of k, the client queries an arbitrary small bounding box until the set S_k of retrieved points is nonempty. It then selects a random point from S_k. In the next iteration, the bounding box will expand by decreasing k as long as $|S_{k-1}| > |S_k|$. Then the next relay is obtained by randomly choosing a point from the relative complements $S_{k-1} - S_k$. A geometrical representation is shown in Figure 3.

4.2 Incentivized Geographical Diversity

Loki network's LLARP protocol with the unique features of financial incentives for node operators is likely to be the new evolution of onion routing inspired network. The current node operators join the network by submitting a staking transaction thereby locking up their funds. Inherently, the blockchain staking transactions build up a DHT type key value store of service nodes. Service node operators periodically divide a protocol level reward equally. Arguably building efficient path selection algorithm is only

part of the solution, an even geographic diversity plays a key role in reliable and efficient path selection. Lokinet's unique reward based protocol can incentify geographic distribution avoiding centralization of nodes operated by a few VPS providers and at the same time providing less represented areas the ability to select paths more efficiently, and even more importantly reduce latency and jitter. The protocol already uses swarm based techniques to construct subsets of the network responsible for querying other nodes in an effort to spot potential failures, changes of IP addresses that get penalized, etc. Adding a geohash to the staking transaction would enhance the DHT with additional geographic information. Nodes may choose their own level of accuracy within some predefined upper and lower bounds, depending on their privacy requirements. Describing a wide area where the node is located is sufficient. The swarm would then need to verify the location with a reasonable confidence area. The simplest way to achieve verifiable location assurance is to use a distance-bounding protocol. In the most general form, a distance-bounding protocol is an authentication protocol between a verifier and a prover in which the verifier can verify the claimed identity and physical location of the prover [5]. A new node entering the network would execute this protocol against other nodes in the current swarm, and submit proofs alongside the staking requirement. Previous research suggests other methods can be used even more efficiently and accurately [10]. Interest in geographically provable location protocols has risen substantially, as a result many variations of the initially proposed protocol were proposed [3].

In order to provide economic incentives towards decentralizing the network, the protocol needs an objective measure of geographic disparity. One of the more widely used functions for geographic diversity is the Nearest Neighbour Analysis. By computing the nearest distance of every point in a region and their averages. We then compare the average index with a standard uniform pattern. If the measured index is greater then the random(uniform) index, the spatial distribution tends to be uniform, otherwise it tends to aggregate.

The other option is using statistical methods such as chi-squared test between the observed point distribution and the random uniform distribution. This is done by partitioning the space into patches the same way geohashing does, then count the points in individual squares and test the distribution against the uniform distribution. The obvious drawback with this is how to dynamically choose the best partitioning resolution to avoid unreliable tests.

The less obvious measure would be to use geohashes. Geohashes sharing prefixes is a measure of proximity. To measure the geographic diversity a

new node would add to the network could be done by computing the average edit distance [21] between the newly added node and the rest of the network. The methods used by the protocol to evaluate the geographic diversity are beyond the scope of this paper. The aforementioned methods would need to be evaluated and tested on edge cases. However, once the measure is derived, an economic incentive can be provided to service node operators that contribute to the network by increasing the geographic diversity, and potentially penalize service nodes that increase clustering. Penalties are more important in cases where a new node's advertised geohash fails verification to prevent the problem commonly refereed to as nothing at stake problem [13]. Generally, if the protocol rewards geographic diversity, node operators will attempt to manipulate their locations data to increase their rewards. Unless there is a penalty for doing so, by game theoretical assumptions, everyone will chose to manipulate the data to maximize their reward.

5 Feasibility Study

This section presents a feasibility study of the proposed method. The simulation was implemented based on actual Lokinet nodes. The goal is to empirically estimate the potential improvements in latency by comparing the random path selection to the proposed spatial path selection algorithm.

5.1 Simulation

We simulate how latencies vary depending on the path selection algorithm. The tests were done using Lokinet as the reference implementation of LLARP. The process is presented as an itemized list of tasks that brought us to the complete simulation:

- *obtain a list of service nodes*; by running a lokid daemon, we can obtain a list of all service nodes and their IP addresses,
- *georeference each IP address*; the mapping of IP addresses to their approximate location was done using a free IP geolocation database, in our case the Geolite2,[7]
- *encode georeferences using geohashing*; The geohashing relies on string representation of geographical location, the accuracy of the location rises with the length of the string – geohash. In practice deleting

[7] GeoLite2 Free Geolocation Data: https://dev.maxmind.com/geoip/geolite2-free-geolocation-data?lang=en

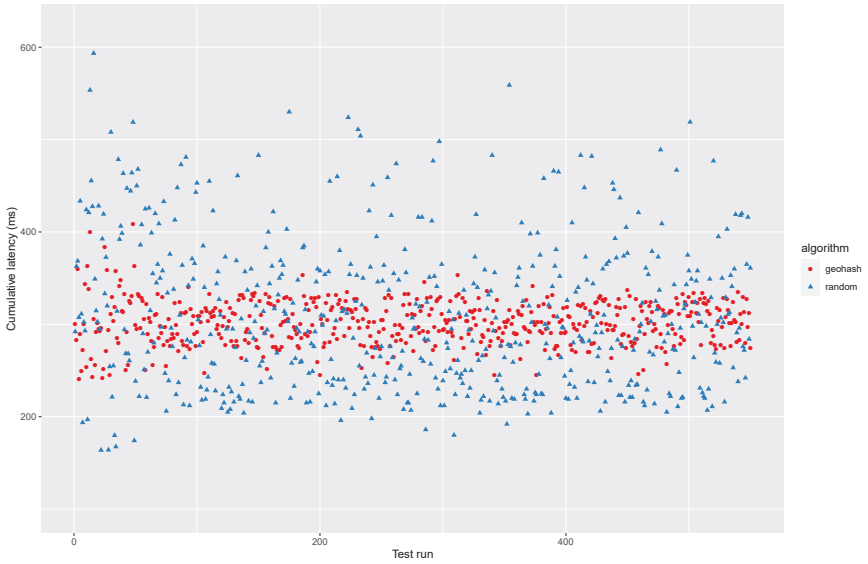

Figure 4 Simulation results showing how latency varies depending on the path selection algorithm. In total 50 paths were built using both algorithms and latency tests carried out on Lokinet by measuring RTT to a low latent public server.

characters from the end of the string simply increases the bounding box of the location.

- *choose a random reduction length for each geohash*; delete a random number of characters from each geohash the service node advertises to the network in an interval between [1, 5], which is to reduce the accuracy of the location descriptor to a larger bounding box,
- *build a predefined number of paths with each algorithm*; We built 50 paths with each algorithm. All paths were of length 3,
- *use basic ping to measure latency for each path*.

Figure 4 shows the simulation results for cumulative latency in milliseconds for a total of 100 paths (50 for each algorithm). The geohash based path selection was done from a client in middle of Europe, hence no bounding box selection was empty. The latencies between all pairs of nodes were obtained in the following way:

- We were able to gain access to approximately 13% of the Lokinet service nodes through a third party that wishes to remain anonymous.
- Each of these nodes measured latency to each other node in the network to produce a sparse graph.

- A full graph was constructed by estimating the missing edges with an average latency weighted with the estimated geographical distance.

We address the concerns with respect to security, and path selection in geographical areas with a low number of routing nodes in Section 6.

5.2 Results

Figure 4 illustrates the simulation results comparing the most secure random path selection with the proposed geo-based path selection. As expected, pseudo-randomly selecting 3 nodes to build a routing path introduces high variance in the observed latency. We observe that using geo-based path selection limits the latency spikes experienced in random selection. Geo-based paths were of same length where nodes were selected as:

1. First node is randomly chosen from a set of nodes n_1 that fall within the bounding box G_4 defined by the origin node's geohash of length 4. The bounding box is approximately a square of 39.1 km^2
2. Second node on the path is chosen from a set of nodes $N_2 = S_3 - G_4$ where S_3 is the set of nodes within the bounding box defined by the geohash of length 3 of approximate area of 156.5 km^2.
3. third node is randomly selected from the set of nodes $N_3 = G_2 - N_2$ where G_2 is the bounding box of geohash length 2, which is an area of approximately 1,252 km^2.

Moreover, we observe that even random path selection can produce low latent paths but with very high variance. Our path selection algorithm produces much more consistent latencies, which is desirable from a QoS perspective.

6 Conclusion and Future Work

In this paper we build upon the existing work on geo-location aware client side path selection in anonymous routing protocols. We identify the problem of geographic clustering of nodes with an insightful analysis of a live network (Lokinet). We focus our research on LLARP protocol and the reference implementation Lokinet due to it's Sybil attack resistance. We support our contribution with a simulation comparing geo-based path selection with purely random selection as the later has the most desirable security properties. While the proposed algorithm introduces some structure to the network, we argue that it retains most desirable properties of pure random selection within

each geo-shard. Our simulation results show that building geographically aware paths can lower the RTT latency considerably. Even more importantly, the latencies are much more consistent, which improves the quality of service.

In edge cases with a high geographic concentration of nodes had drawbacks in anonymity. To address this we propose an incentive structure built into the protocol to adapt rewards depending on geographic diversity of the network to strengthen the security of the path selection. The added rewards will subsidise potentially higher expenses for node operators in other locations as well.

However, we acknowledge the potential drawbacks of such path selection on the overall security, and privacy of the network. To address this shortcomings, we provide possible measures of spatial diversity and propose to use them on the protocol level to provide economic compensation for nodes that add to the geographic diversity of the network, as opposed to nodes, which reduce the spacial scattering.

The exact implementation that would dynamically increase rewards for distant nodes when the network gets packed, and reduce them gradually as the network becomes spatially diverse should be carefully studied from a game theoretical and economic point of view.

Acknowledgements

The authors gratefully acknowledge the European Commission for funding the InnoRenew CoE project (H2020 Grant Agreement #739574) and the PHArA-ON project (H2020 Grant Agreement #857188) and the Republic of Slovenia (Investment funding of the Republic of Slovenia and the European Union of the European Regional Development Fund) as well as the Slovenian Research Agency (ARRS) for supporting the project number J2-2504 (C).

References

[1] Masoud Akhoondi, Curtis Yu, and Harsha V. Madhyastha. Lastor: A low-latency as-aware tor client. In *2012 IEEE Symposium on Security and Privacy*, pages 476–490. IEEE, 2012.

[2] Nasser Mohammed Al-Fannah. One leak will sink a ship: WebRTC IP address leaks. In *2017 International Carnahan Conference on Security Technology (ICCST)*, pages 1–5. IEEE, 2017.

[3] Aiiad Albeshri, Colin Boyd, and Juan Gonzalez Nieto. Geoproof: Proofs of geographic location for cloud computing environment. In *2012 32nd*

International Conference on Distributed Computing Systems Workshops, pages 506–514. IEEE, 2012.

[4] Kevin Bauer, Damon McCoy, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Low-resource routing attacks against tor. In *Proceedings of the 2007 ACM workshop on Privacy in electronic society*, pages 11–20, 2007.

[5] Stefan Brands and David Chaum. Distance-bounding protocols. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 344–359. Springer, 1993.

[6] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, February 1981.

[7] Primavera De Filippi and Smari McCarthy. Cloud computing: Centralization and data sovereignty. *European Journal of Law and Technology*, 3(2), 2012.

[8] Maxence Delong, Eric Filiol, Clément Coddet, Olivier Fatou, and Clément Suhard. Osint analysis of the tor foundation. *arXiv preprint arXiv:1803.05201*, 2018.

[9] John R Douceur. The sybil attack. In *International workshop on peer-to-peer systems*, pages 251–260. Springer, 2002.

[10] Bamba Gueye, Artur Ziviani, Mark Crovella, and Serge Fdida. Constraint-based geolocation of internet hosts. *IEEE/ACM Transactions On Networking*, 14(6):1219–1232, 2006.

[11] Paul T Jaeger, Jimmy Lin, Justin M Grimes, and Shannon N Simmons. Where is the cloud? geography, economics, environment, and jurisdiction in cloud computing. *First Monday*, 2009.

[12] Kee Jefferys, Simon Harman, Johnathan Ross, and Paul McLean. Private transactions, decentralised communication. Technical report, OPTF, 2018.

[13] Wenting Li, Sébastien Andreina, Jens-Matthias Bohli, and Ghassan Karame. Securing proof-of-stake blockchain protocols. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pages 297–315. Springer, 2017.

[14] Jon McLachlan and Nicholas Hopper. On the risks of serving whenever you surf: Vulnerabilities in tor's blocking resistance design. In *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, pages 31–40, 2009.

[15] Guy M Morton. A computer oriented geodetic data base and a new technique in file sequencing. Technical report, International Business Machines Company New York, 1966.

[16] Steven J Murdoch and George Danezis. Low-cost traffic analysis of tor. In *2005 IEEE Symposium on Security and Privacy (S&P'05)*, pages 183–195. IEEE, 2005.

[17] Gustavo Niemeyer. Labix blog. https://web.archive.org/web/20080305223755/http://blog.labix.org/#post-85. Accessed: 2021-4-22.

[18] Andriy Panchenko and Johannes Renner. Path selection metrics for performance-improved onion routing. In *2009 Ninth Annual International Symposium on Applications and the Internet*, pages 114–120. IEEE, 2009.

[19] I2P Project. Garlic routing and "garlic" terminology. Technical report, I2P Project, 2014. Accessed: 2021-1-9.

[20] Tor Project. Possible upcoming attempts to disable the tor network. Technical report, The Tor Project, 2014. Accessed: 2021-1-9.

[21] Eric Sven Ristad and Peter N Yianilos. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532, 1998.

[22] Stephen Rollyson. Improving tor onion routing client latency. *Georgia Tech College of Computing, Tech. Rep*, 2006.

[23] Fatemeh Shirazi, Milivoj Simeonovski, Muhammad Rizwan Asghar, Michael Backes, and Claudia Diaz. A survey on routing in anonymous communication protocols. *ACM Computing Surveys (CSUR)*, 51(3):1–39, 2018.

[24] Robin Snader and Nikita Borisov. Improving security and performance in the tor network through tunable path selection. *IEEE Transactions on Dependable and Secure Computing*, 8(5):728–741, 2010.

[25] Helger Lipmaa UT, Michał Zając UT, Claudia Diaz KUL, Tariq Elahi KUL, Benjamin Weggenmann SAP, and Aggelos Kiayias. Design, modelling and analysis. Technical report, European Commision, 2016.

[26] Rungrat Wiangsripanawan, Willy Susilo, and Reihaneh Safavi-Naini. Design principles for low latency anonymous network systems secure against timing attacks. *Physical Sciences and Mathematics Commons*, pages 1–11, 2007.

[27] Philipp Winter, Roya Ensafi, Karsten Loesing, and Nick Feamster. Identifying and characterizing sybils in the tor network. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 1169–1185, 2016.

Biographies

Aleksandar Tošić. PhD candidate, teaching, and research assistant at University of Primorska, and Innorenew CoE. His fields of research are distributed, and decentralized systems, Peer to Peer networks, and distributed ledger technologies.

Jernej Vičič. Associate professor and research associate at the University of Primorska and Research Centre of the Slovenian Academy of Sciences and Arts. His research interests are quite broad ranging from language technologies to distributed systems.