GEO protocol

Max Demianiuk Dima Chizhevsky

August 30, 2018

Abstract

This document describes GEO – a protocol for highly efficient decentralized processing of peer-to-peer payments in a distributed network of nodes that is not based on a common ledger or consensus process. GEO can both serve as a highly efficient trustless payment network layer for existing blockchain systems, and facilitate payments in fiat or other non-blockchain-based units of exchange through a network of distributed trust. By emphasizing routing features, cross-unit exchange and transaction atomicity, GEO enables efficient inter-blockchain exchange of value. It also provides a unique mechanism for onboarding users onto the cryptofinance ecosystem. Because the GEO network does not make use of specialized hubs for routing, even your smartphone can act as a payment processing node, eliminating the need for large intermediaries. GEO is blockchain-agnostic, and integration with a large number of blockchain networks is trivial.

Contents

1 Introduction

The fundamental need in payment processing is ensuring against the possibility of an actor misrepresenting their holdings to others and ultimately spending the money they don't have. The three known ways to prevent participants from making false claims of payments (also known as the double-spend problem) are:

- Using a trusted intermediary to hold a database that at all times expresses the global truth (this is the trust-based model);
- Using a byzantine fault-tolerant consensus protocol, such as the one used by the Bitcoin cryptocurrency (this is the trustless model); and, finally,
- Using a distributed loan accounting mechanism, such as the one used in the historical Hawala payment network in the Middle East (this is the distributed-trust model).

Only the trustless and the distributed-trust models could be said to enable peer-to-peer payments. The era of peer-to-peer payments started with the Bitcoin protocol, which enabled fully trustless transfers of an abstract asset between network participants. Additionally, the Bitcoin network is completely open and highly censorship-resistant: it is impossible for anyone who does not possess vast (and well-known) amount of economic power to prevent users from participating in the Bitcoin network or from sending payment transactions. On the other side, the trustless approach to P2P payments has a number of drawbacks, fully described in cryptocurrency literature. To summarize: it is computationally expensive and wasteful; it cannot operate on lightweight and mobile devices; it gives rise to highly volatile asset prices; it is often slow; and it requires secure key-management – a big hurdle to broad adoption.

Besides, due to technical and other constraints, the level of interaction between different cryptoassets is insufficient, drastically limiting implementation of the technology. The GEO Protocol delivers an alternative that is based on the distributed trust model of peer-to-peer payments, but when applied to blockchain-based cryptoassets, it becomes a fully trustless system for payments, based on the innovation of blockchain layer 2 networks.

Thus, GEO is able to become a viable link between distributed ledgers, and also between various kinds of non-crypto assets (for example, fiat money, commodities, securities).

1.1 Market overview

We live in a world where technologies have already reached the level that is sufficient for sending funds (money) easily and safely — as easily as sending a message, even between smartphones. Nevertheless, the existing infrastructure adapts very slowly to the new digital landscape. At present, the financial world is mostly built on trusted intermediaries who account for assets and ensure the transactional processing. However, those intermediaries are very poorly synchronized between themselves (sometimes there are even manual operations still present), there are no universal standards of interaction, etc. The highlights characterizing the current state are:

- Weak integration. Due to a high cost of integration, just a few payment service providers cooperate with each other. Fees are charged for transactions (commissions for international transfers are especially high), and the transaction speed is extremely low (in some cases, it could take several days to get money from a sender to the receiver). A payment can even get lost during the execution of a transaction between the parties. The fraudulent activity level is high.
- Monopolization of the payment processing market. Monopolies and the closed-source software limit the interaction between the market players and the consumers. The lack of open-source processing technologies is a deterrent for innovations in the sector. It is possible to create a great mobile application but many financial institutions will not be able to use it simultaneously.
- Extreme overregulation of the system plus multiple international restrictions.
- Uneven development of the infrastructure.

The emergence of trustless technologies of accounting for assets and payments allowed performing protected transactions without intermediaries. Due to the regulation complexity and geographical restrictions of the legacy finance system on the one hand, and the high availability level of the newly emerged trustless technology on the other, its use grows actively, and has, in fact, led to the creation of a whole new economical and technological ecosystem that we call the crypto industry.

At the same time, despite the new technical possibilities of digitizing and trustless transfers of assets, the characteristics of the known technologies of distributed registries are objectively insufficient to serve retail payments, the exchanges, national currencies, etc.

Moreover, each “crypto asset” exists as a closed-source system due to the lack of the industry-wide standardization and the difference in technological approaches. In this sense, the crypto industry is even less flexible and suitable for synchronization then the legacy finance.

So, at the moment, going beyond is connected with high costs and long wait times, or is impossible at all, though one can send an asset relatively easily within a country or a blockchain registry. It is worth mentioning that it is difficult for crypto projects to use derivatives from an asset, such as IOUs [18]: that makes asset liquidity management much more complicated. To build a sustainable and efficient trustless infrastructure for exchanging and transferring assets, a protocol is required to be:

- Open (allow to achieve mass adoption and be flexible and customizable);
- Easily integrated (the ability to make transactions between different ledgers);
- Atomic (double-spend proof);
- Scalable; With higher maximum flow capacity (comparing with the existing processing technologies);
- Lightweight client (where devices such as an ordinary smartphone would be able to act as a processing node).

1.2 Background and motivation

For the last few years, technologies in the crypto sector have been evolving rapidly, mostly in the direction of consensus mechanism improvement. At the same time, considering some functional constraints of on-chain technologies and a demand for off-chain solutions have emerged, especially for state-channel-based protocols.

After an analysis of existing projects, we have concluded that state channels are not, in fact, trying to solve all limitations. Some problems still exist, e.g. liquidity inside a state channel network and the lack of interoperability of different crypto assets.

The key features of the GEO protocol are:

- There is no common ledger — it is an off-chain protocol. Data is distributed over the network, and is stored by the nodes between which the channels are installed. Such an approach allows to reach high throughput, as well as accessibility for less powerful devices.
- Transactions are executed by local consensus of nodes that participate in these transactions.
- The protocol provides full atomicity for a multi-ledger network and allows to conduct complex transactions with several assets.
- Composite channels between two nodes. That’s a combination of state channels and trustlines (IOU channels for various digital and physical assets, including fiat) that allows building a more flexible infrastructure, as well as the integration of the existing ones. Such a technology provides an opportunity to make trustless transfers of tokenized assets using state channels, simultaneously implementing scalable trustlines technology with the ability to create non-tokenized assets.
- Implementation of post-quantum cryptography.

Due to its architecture, the GEO Protocol allows building an infrastructure for various dApps and solutions such as: payment systems, cross-chain decentralized exchanges (DEX), rating systems and loyalty programs, delegated democracy, decentralized credit networks, clearing systems, and IoT solutions.

At the moment, our team is working on a concept that fits this description exactly: "*We are not followers of a certain crypto-asset or technology. On the contrary, our goal is to create a flexible infrastructure protocol that connects different ideas and ledgers, including centralized ones. We believe that only an open and equally accessible solution will connect all industry participants evolutionarily (as it happened with HTTP or SMTP)*"

1.3 Related work

Lightning network

The most famous implementation of the state channel is the Lightning Network[6], which was recently launched on mainnet. The network handles the routing of multi-hop payments across a distributed network of nodes, secured using the hashed time-locked contracts (HTLCs) approach. It uses modified Dijkstra's algorithm [22] (an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks) and onion routing Sphinx[23] to securely and privately route HTLCs within the network. By itself, HTLC is an atomicity solution for Lightning. The key differences between Lightning and the GEO Protocol are:

- Lightning is built on top of Bitcoin and can be implemented only by blockchains with same hash function as those of Bitcoin. GEO Protocol is blockchain-agnostic, able to connect different ledgers and exchange different kinds of assets;
- In terms of the topology collection method, GEO nodes need to know only their first-level connections, unlike Lightning, where the gossip protocol is implemented, which requires storing more topology information and refreshing it constantly. This may lead to network overload and become a scalability issue.
- GEO utilized a different approach to achieve atomicity. While Lightning uses HTLC that may cause a loss of intermediary funds in case of disconnecting from the network, GEO Protocol is relying on observers, network participants with a separate protocol that ensures the full atomicity of payments.
- GEO Protocol supports atomic multi-path payments, and Lightning Network does not. At present, these types of operations it only exist in a form of proposal for Lightning[24].
- There are no transaction fees in the GEO Protocol, while the Lightning Network requires transaction fees.

Interledger

Interledger Protocol (ILP)[17] proposed a protocol for secure payments between different ledgers through an arbitrary chain of ledgers and connectors that can use various types of relations in order to reach the final destination — a receiver.

ILP routing is similar to BGP routing. It's performed by special nodes — connectors (they could be run either by a person or by a large enterprise and may act as a DEX). Each connector has its own routing table where the path and next hop are determined. Tables are created when a new connector is added based on the tables that belong to other connectors or can be configured manually. When a packet is received, the connector sends it further based on its table and using the Longest prefix match rule.

Interledger uses HTLA (hash time-lock agreement), an agreement based on trust between participants, but with all payments divided into small parts, so that if one of intermediaries steals the money, then they will lose their reputation. GEO Protocol supports full atomicity due to the Observers Chain — a private blockchain with BFT consensus [15; 16] that provide payment finality.

Preliminarily, ILP had the atomic mode that provides atomicity for payment chains in which the participants can agree upon a group of notaries. However, due to concept complexity of the implementation in a cross-chain environment, as well as the fact that users would have to trust notaries, this concept has not yet been implemented.

Celer Network

Celer Network[14] constructs generalized state channels technology that aims to scale different blockchains. The main difference is the ability to scale smart contracts. Celer is based on the Backpressure algorithm[9; 10] that is aimed to achieve high throughput instead of finding the shortest path as most path-based projects do. In a nutshell, it works like this: each node in each point in time calculates congestion of its first level. During the calculation, the transactions queue and the channel imbalance are taken into account. When a calculation is completed, the node sends a transaction to a node with the lowest congestion. This process repeats until a node reaches receiver node or its first level (in this case, congestions of receiver is believed to be 0).

The key difference between GEO Protocol and Celer Network lies in the mission of the projects. Celer Network aims to scale every blockchain, but the goal of GEO is to make the transfer of different assets as easy as possible and to connect different ledgers.

Another difference lies in the way atomicity is achieved. Celer uses HTLR (hash time lock registry) that is the extension for Sprites's PM [4] (Preimage manager) — something like an arbitrator for the HTLC that would allow delegating the function of taking decisions regarding the expiration of the lock contract period from each individual node to the central registry, and thus avoiding the problem where one of the payment participants loses money whileoffline.

In the HTLR, there are two dependency endpoints — IsFinalized, QueryResult. The first one returns if a preimage has been registered before the block number, the second one returns if a preimage has been registered. Potentially, these two functions can be united into one. It should be noted that the HTLR is always on-chain.

Related research

There are a few papers that are describing routing algorithms for distributed networks.

Landmark[8] routing was proposed as one of the options for decentralized payment routing in several payment channels. The key idea of Landmark routing is the definition of the shortest path from the sender to the receiver through an intermediate node called Landmark — usually a well-known node with high connectivity.

SpeedyMurmurs[2] complements the previous shortest path routing algorithms by accounting for the available balances in each payment channel. In the protocol, the embedding prefix tree is used for routing — node coordinates tree, in which coordinates are assigned in the form of vectors, starting with an empty vector at the landmark/root. Each internal node of the spanning tree is its children and appends the enumeration index of a child to its coordinate to obtain the child coordinate. The distance between the two coordinates corresponds to the length of the shortest path in the spanning tree between them. When changing the network topology (especially when removing nodes), there are often situations when one needs to update the information on a large part of the nodes (the prefix tree). This approach is sensitive to malicious modification of the prefix tree and the generation of duplicate coordinates (there must be a central register of already issued coordinates to solve this problem).

Flare [19] is a routing algorithm that was proposed for the Lightning network by the ACINQ team. It also aims to find the shortest path, but uses totally new approach. The node constructs its own routing tables where it can find a path to the first (second or even more) level nodes. When two nodes have to make a transaction between each other, they exchange their tables and search for intersections. If there are no such intersections, they can use other routing tables using special nodes – beacons (any basic node that agreed to be a beacon for a particular user). The process repeats until path is found.

All the proposed solutions have made a significant contribution to the development of the entire industry and in specific directions as well.

Nevertheless, the above-mentioned constraints such as atomicity gaps, the ineffectiveness of topology collection and the interoperability issues are crucial and relevant for rapid market evolution.

2 GEO Protocol

The GEO Protocol provides the ability to implement a decentralized peer-to-peer network that allows its members to transfer atomic assets, including the exchange of different types of assets. While designing the basic principles of the protocol, we addressed the limitations of existing distributed systems, including most blockchain-based systems, and their scalability and transaction throughput challenges.

In the GEO Protocol, consensus is reached only between the parties who are directly involved in the transaction. At the same time, they do not have information about the state of the rest of the network. There is no common ledger for the assets that present in the network, as well as the general source of information about the network itself. There are two types of channels in the GEO network:

- Trustlines — channels between two parties in the network, that provide for simple and fast assets transfers, but are not connected to any external ledger and, as a result, are not related to any external environment.
- State channels — channels between two parties in the network that use trustlines as their basis, but are also related to external ledger, and are mirroring their balances to it.

Both types of channels might be used by any pair of participants of the network simultaneously and for processing various types of assets.

Due to the state channels, one can bypass the existing limitations such as scalability and interoperability. The main idea of the GEO Protocol is not to attach to one blockchain, but to allow users to have access to multiple channels between two nodes in different assets.

However, taking advantage of distributed technologies, we do not want to ignore the existing financial solutions of the real world. Therefore, an important approach of the GEO Protocol is the transfer of financial relationships to decentralized realities through the use of IOUs. This solution, taking advantage of distributed systems, allows creating a multi-equivalent credit network without a central issuer. Users form such a network independently with the help of their existing real-life relationships.

The Geo Protocol has combined two technologies: the flexibility of real financial relations (digitizing the selective trust of the real world) and the security of decentralized solutions in the trustless environment. We call these kinds of connections between two nodes "composite channels." The name fully reflects its multi-component nature, since there are different types of channels, equivalents and assets present in the connections.

2.1 Roles and key components

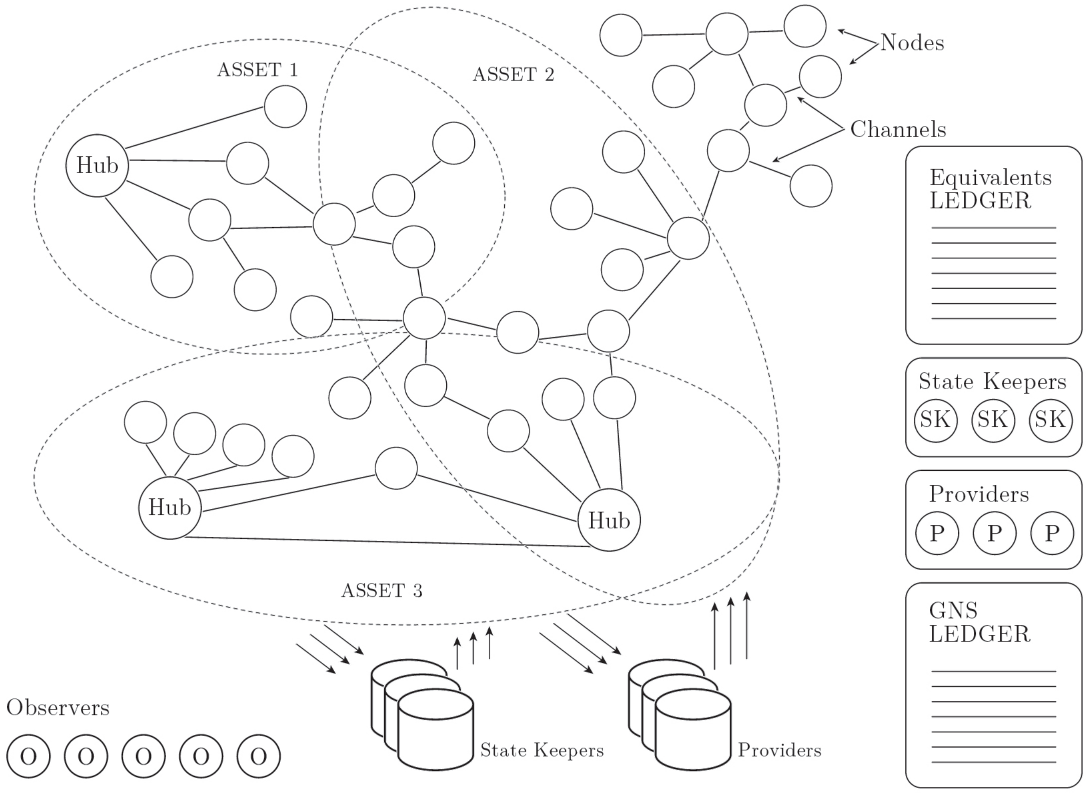

Figure 1: Key roles interaction in GEO Protocol

Basic components

1. A node, or a network node, is a basic “buiding brick” of the GEO network. As a participant in the peer-to-peer network, a node interacts with other participants of the GEO Protocol. It is installed on a specific hardware device. The node application is so lightweight that it can be installed on an ordinary smartphone.

2. A channel is a connection between two nodes. A channel is an agreement to automatically perform a transaction on demand in the future, in a specific equivalent and quantity. A channel can be one-directional or bi-directional. Here are some characteristics of channels in GEO:

   - They are set in a specific equivalent or measure units;
   - There could be an unlimited number of channels between two nodes (in different equivalents);
   - Channels are trusted, one-directional by default;
   - The channel can be a composite channel (that includes both trustline and state channel).

3. GNS (Geo Name Service) is the registry of cross-accounts with identification, information about the provider, users, observers and additional data for custom solutions. It is a kind of a virtual passport for the GEO network nodes:

   - Initially, records are stored in an external ledger (Testnet on Ethereum)
   - Minimal information is required: ID/user identifier/provider
   - Entries in the register can be created by providers only
   - Custom solutions are possible
   - Ability to provide selective access to individual values

- Ability to delegate records to specific values: ID/user identifier/provider/custom information
- Ability to supplement the "virtual personality" with custom information, i.e. an opportunity to extend the record (for example, private key on BTC, identity number, medical records)

Roles:

1. Participants: the basic role allowing equitable interaction with the network within the protocol. Each participant must activate the node to use the network. A participant may:
   - Create or delete a channel
   - Generate transactions
   - Participate in the implementation of other transactions (automatically)
   - Assume any other role in the system
2. Hubs: nodes with a large number of first level connections. Their main function is logistics: they provide greater connectivity and network capacity. A hub can receive a reward for its services.
3. Observers: a separate protocol that protects the network from a certain type of attack. It is not necessary to be a node in order to be an observer on the GEO network.
4. GNS Provider: a separate protocol. The provider stores the IP address table for the GEO network nodes. Because of this, GEO network routing is ensured.
5. State keeper: a separate protocol. Allows the nodes of the GEO network to delegate the state of their open connections. A state keeper can sign transactions for the node, and also ensure that the channels are closed correctly while the node is offline. At the same time, it cannot intercept funds. This service protects the network from some attacks and ensures its greater reliability and availability.

Events:

1. Transaction
   - Can only be done through channels (using trustlines or composite channels).
   - Can be carried out through chains up to six hops (meaning that there could be up to five intermediary nodes between sender and receiver of a transaction). The simulation results have shown that, to conduct a transaction for each node, it's enough to use 3-6 hops between two nodes. In further versions of the protocol, it will be possible to determine a custom number of hops.
   - Can be carried out using multiple hops.
   - Can be split into several paths (in case of insufficient payment capacity of a single path, a transaction can be sent through several paths without affecting its atomicity).
   - 100% local consensus is reached only between the nodes involved in the transaction independently from the rest of the network.
   - Is reflected as a simultaneous change of balances on all channels participating in the transaction (there may be situations where some nodes commit the transaction later: dropouts, network failures, etc.)
   - Transactions can be in one equivalent, or in different equivalents (cross-equivalent).
   - Duration of a transaction on GEO is only a few seconds, double-spending is impossible (transactions are atomic).
2. Clearing (closing cycles): automatic netting (within the protocol) of accumulated balances in a closed chain. After the cycle is found, a process similar to a transaction is initiated.

3. Creating (or deleting) a channel:
   - Can be created only by initiator node.
   - An exchange of crypto-signatures for future changes in the balance through the channel is necessary.
4. Creating an equivalent (users can create GEO BTC or GEO LTC that are equal to BTC or LTC, and freely exchange them between network participants).
   - Equivalents list is stored on a separate Ethereum contract (initially).
   - Any user can create their own equivalent of any asset (user initiates a creation of a smart-contract record and then can exchange the newly created equivalent across the network).
5. Recording into the GNS Registry:
   - Entry into the register is made by the providers only according to the contract.
   - Can be initiated by either users or applications.
   - Can be supplemented by additional information.
   - GNS records contain lists of users, providers and observers.

2.1.1 GNS

Geo Name Service (GNS) is an independent decentralized participant identification system designed to serve the financial infrastructure of the GEO protocol. GNS was built with the aim to enable each user to create a decentralized and self-sovereign identity with the following features:

- Decentralized and self-sovereign. Independence from centralized authorities and identity providers. No one can own and control the user's identity except the user him/herself;
- Privacy and security. A user decides to whom he/she is willing to grant an access.
- Interoperability. The identity is compatible with different ledgers.

Currently, GNS has much wider functionality and is a crucial part for the entire GEO network. All the operations between participants and with tokens happen here. GNS performs the following tasks:

- Keeping a registry of participants, their identification data, as well as their pseudo-addresses in financial services and services, the benefits of which are used by a (specific) participant.
- Defines the main roles in the network. GNS holds the lists of users, providers and observers.
- Routing requests between GEO network members from "gray" areas of the Internet. This aspect of the system is important for the GEO network as a whole, since it allows one to create and maintain a direct connection between Internet participants without having a static ("white") address.
- Providing necessary identification data to the services on behalf of and by agreement with network participants (authentication in various services).
- Excessive backup of the participant data of the GEO network to prevent irreversible loss of credentials.

Identification of participants is provided by a separate independent protocol. This solution allows nodes of GEO network to be involved simultaneously in payment and other systems with different interaction logic.

Basic mechanics

GNS is a structured cumulative distributed decentralized register, providing high-level access to the GEO network infrastructure and consisting of three main subsystems:

- An identification system that generates and distributes unique identifiers for members of the GEO network. It can also be used to map the identity of the GEO network to the identity of other systems (Bitcoin, Ethereum, etc). The main area of responsibility of this subsystem is the deduplication of the IDs of the GEO network members in the public registry.

- A high-speed distributed registry of public IP addresses that processes real-time requests of network participants and allows direct (p2p) connection of GEO network participants from “gray” Internet areas (for this purpose we use NAT, a method of remapping one IP address space into another by modifying network address information in the IP header of packets while they are in transit across a traffic routing device). The main area of responsibility of this subsystem is providing of IP addresses to the participants in real time.

- IP routers. The role of distributed routers on the network comes to hold a “key-value” table in the following format: ”Participant Subzone: Current Public IP Address and NAT back port.”

Each service provider maintains this table independently through its infrastructure in order to provide direct access to GEO Protocol participants who have chosen this provider as a representative in the public Internet segment.

Entities

- Node / GEO Node is a participant of the GEO network; it needs real IP addresses of other nodes in the network. It uses its provider for retrieving IP addresses of known members via their GNS names.

- Provider is a high-level service, working in the public Internet, that provides names resolution for the nodes in the network by request. Each provider maintains its own high-throughput map (Name → IP Address).

- GNS Registry that contains high-level GNS entries. GNS is built using the Ethereum blockchain.

- Observers are entities with their own private blockchain that are responsible for conflict resolving and are providing payments atomicity.

Architecture

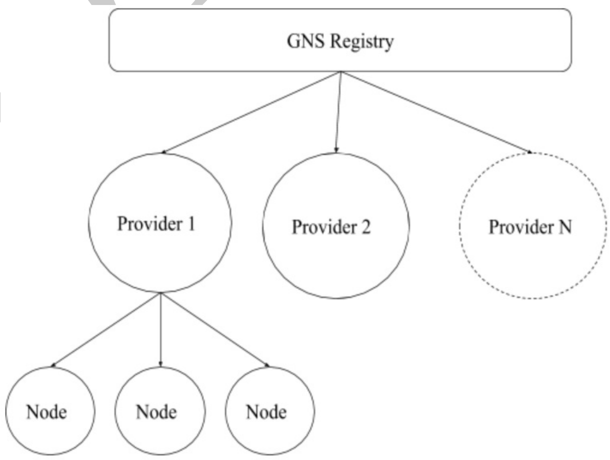

Figure 2: GEO Protocol Architecture

2.1.2 Providers

Each GEO node knows its internal provider’s alias and pings it from time to time to update its global IP address in the internal provider’s database and to bypass the possible NAT.

Provider caches node's global IP addresses for some time and shows it to the other nodes by request. Once cached, each IP address is probed from time to time by the provider to keep the connection alive. If a remote node doesn't respond to the ping, then the connection is considered as obsolete and would be removed from the cache. This addressing technique is needed to provide NAT-agnostic addressing for IPv4 networks. This is the IP address discovering flow:

- The node knows its contractor's global alias or provider-specific alias and sends the discovering request to its provider. Optionally, the request might contain a fields list, which should be returned. By default, the whole record is returned back to the node in case of success.

- Provider looks in its internal aliases namespace and, in case alias is present there — returns the whole record if no additional fields specifications are present in the request, or only a subset of fields.

- If no alias is present in the provider-internal namespace, then provider parses the requested alias and extracts global alias, goes into global addressing zone (blockchain) and looks for the specified record. In case of success, the provider transfers the request to providers behind it and waits for the response. In case of success, the retrieved record would be cached for some time and returned to the originating node.

Users may not be registered in this system if their circle of relationships is limited, and find a local provider that will also not be registered with GNS. But in this case, access to the rest of the network for them will be significantly limited, since other participants will not be able to find them on the Internet and convey necessary information.

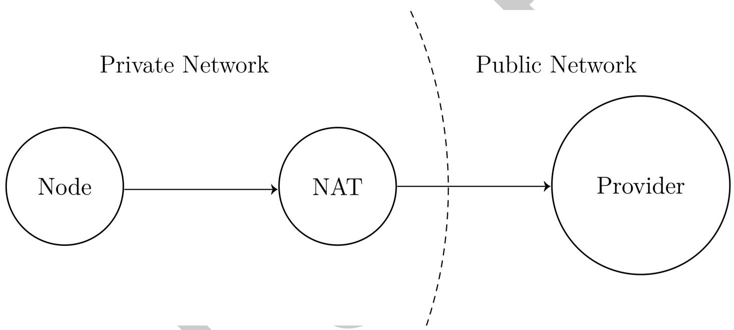

Figure 3: Providers arrangements

2.1.3 Observers

Since GEO Protocol considers the usage of mobile phones working in mobile networks that are much more often subject of destructive network fluctuations than devices with a permanent Internet connection, the system should include solutions for leveling the situations of partial losses of network packets, their damage, etc. However, this kind of situation usually is not permanent and can be resolved at the level of the exchange protocol.
All payments can be conditionally divided into two categories:

- Reversible by node's request
- Irreversible without a general agreement

The purpose of the stage of issuing debt checks and their subsequent signing is the formation of a set of signatures of all participants in the transaction under an agreement to change their own balances in favor of the receiving party. Difficulties arise in the event a part of the nodes issued their own debt checks, signed a general agreement on the transaction and transferred it to the payment coordinator, and then, due to a destructive influence, did not receive a full set of signatures of the remaining participants in the transaction (the coordinator left the network or deliberately delayed the beginning of the operation). In this case, the nodes are in a state of uncertainty, because they cannot track the further fate of the operation and, therefore, cannot make a final decision about its completion. Also in order to avoid collision of funds, they are forced to keep reserves on their own trustlines/channels until the clarity of the operation is ascertained. Theoretically, waiting may take a long time, and certainty may never come. This is a classic problem of protocols operating on the principle of a two-phase/three-phase commitment. In this case, it is possible for an attacker to compromise the network with the subsequent "freezing" of liquidity by initializing payments (i.e. assuming the role of coordinator) and delaying the beginning of the operation. Observers are members of the GEO ecosystem, working on a separate protocol, participating in the search for or consensus building for operations, whose parties, because of destructive influence, could not reach consensus on their own.

Principle of operation and role

- An appeal to an observer occurs only in the event when there is a suspicion of unfair behavior of other participants. Thus, observers are deprived of any information about transactions that have been completed successfully without their participation.
- An appeal to an observer can be generated by any participant of a payment at any time after transaction start and until it is completed, but it is assumed that participants will seek help from an observer only in case of a problem situation. There are several events foreseen in the algorithm upon the occurrence of which an appeal to an observer is inevitable. But, at the same time, this is significant optimization: any potentially problematic situation in achieving consensus can be reduced to a single solution that is predictable in terms of time and efficiency.
- Each appeal to an observer must contain an operation identifier (unique ID) and a list of participants. Information about the amount of payment, purpose, payment topology (the ways in which the transaction is being performed), as well as who is the sender and who is the recipient is missing. Thus, observers can collect very limited amount of information about ambiguous transactions.
- The observer's role is to ask for a list of signatures from all of transaction participants. Having received the observer's request, a participant can send them a package with signatures (i.e. signatures collected from other participants during the transaction processing). In case a participant does not respond, their vote can be delegated to another participant of the transaction. Thus, the observer's goal is to collect complete list of signatures of the transaction participants. If 100% of signatures are collected by the observer, he/she should inform those participants who applied to him/her. In case of failure observer must repeat request attempt to the participants in a time span of up to 10 minutes from the moment of the first application for this operation. If 100% of signatures are not collected during this timespan, the observer has to generate a special reject packet informing all payment participants about the cancellation of the transaction. The observer's decision

is considered as prevailing. After receiving a reject package signed by the observer, participants can discard reserves, cancel a transaction and free up channel liquidity for other transactions.

Thus, an observer can cancel a transaction without the consent of all payment participants, but has no right to confirm the transaction without the consent of all participants and, accordingly, can't not affect the balances of the network node.

So the decision for each doubtful or problematic transaction can be made in a strictly predefined timespan.

2.1.4 Registry of equivalents

The GEO protocol provides the interaction of nodes in different units: a node can open a trustline to its counterparty in any equivalent – USD, BTC, mile, kWh, etc. The difference is the fact that clearing occurs only within the limits of one equivalent. The goal of creating multi-equivalence is to provide transactional liquidity and improve the processes of interaction between various systems.

Equivalents do not have to reflect existing fiat currencies or cryptocurrencies. Everything that is acceptable for servicing a particular economic mechanics of one or another segment of the network participants can be used as an equivalent. Equivalents are the context in which the relationship between two nodes is expressed.

Creation, and also the total number of equivalents in the system, is virtually unlimited. Each equivalent is assigned a set of properties (name, unit of measure, other conditions), depending on the context and purpose. It is not possible to open two or more trustlines in one equivalent for one counterparty, and the number of links in different equivalents is unlimited.

This architecture allows making smart contracts involving several equivalents, which also allows creating more complex systems of interaction. In a simplest form, it could be, for example, a decentralized marketplace; in more complex versions, there could be programmable processes involving several cryptoassets, as well as external conditions (which can be digitized).

Bitcoin	0001	First cryptocurrency
Ethereum	0002	Decentralized computer
USD	0003	Fiat dollar
Watt	0004	Energy
...	...	...
nameN	idN	short description

Addition mechanism

A list of equivalents is created in the blockchain (Ethereum). The name of the equivalent is added to the smart contract for this, the protocol refers to the registry, after which it is possible to freely exchange this equivalent. Anyone can add a new equivalent. Nodes freely decide in which equivalent and to whom to open the channel. When adding a new equivalent, the fee in ETH is charged to prevent DDoS attacks.

2.2 Types of interrelation channels

2.2.1 Trustlines

GEO Protocol allows one to create a distributed peer-to-peer network, supported by community members. Operations in the network are conducted by nodes — devices connected to the network on behalf of participating holders. A node in the network could be installed on any device.

Trustlines

Assets and other values in the network are transferred between the participants with the help of special channels — trustlines. From the technical point of view, a trustline is an accounting primitive that stores incoming trust amount (the sum a counterparty trusts you with), outgoing trust amount (the sum you trust your counterparty with), and the balance between these two after each operation.

In layman's terms, a trustline is a digitally expressed willingness of a participant to accept obligations (IOUs) of another network member without exceeding the predefined confidence limit.

Trustlines may be created on the basis of both personal social ties and organizational business relations. At its core, a trustline is a smart contract signed by both parties. For now, there are only two variables of interaction in the protocol: the equivalent and the limit, but in the future the number of variables will be expanded, in turn allowing the creation of complex systems of interaction. To establish a new channel, two nodes must complete the following steps:

- Create an end-to-end secured communication channel for p2p data transfers between participants
- One node must create outgoing trustline
- Counterparty node must accept or reject incoming trustline
- Then both nodes must synchronize their trustlines and ensure that no operations is possible or done insecurely
- Also nodes must set trustline capacity (maximum amounts of value they trust each other with)

A trustline might be one-directional or bi-directional. A one-directional trustline represents trust flow from one node to another without any backward trust flow, for example, only from node A to node B.

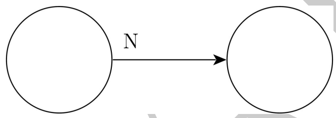

Figure 4: Bi-directional trustline

A bi-directional trustline represents trust flow in both directions. In cases when two nodes trust each other (not necessarily the same amount), instead of two one-directional trustlines, only one bi-directional trustline will be created in the GEO.

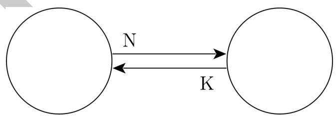

Figure 5: Bi-directional trustline

One bi-directional trustline is more efficient then two one-directional trustlines for the following reasons:

- Only one common keys pool is used
- Simpler accounting and audit logic
- Less space is used on the node's devices

Each operation on the trustline must be signed by both parties. Based on which crypto system is utiized, each operation has to utilize its own key pair, and so, to be able to process several operations, nodes have to establish key pools. Depending on the node configuration, key pools might contain virtually any number of keys, from several keys to up to several thousands.

When an error occurs, nodes enter into audit state, during which they re-synchronize their balances, outgoing and incoming amounts. A node can open a trustline to its counterparty in any equivalent. The number of such trustlines is unlimited, provided each of them will be nominated in a different equivalent (cross-unit): it will not be possible to open two trustlines by one counterparty in the same cross-unit, because it will make the procedure of finding payment paths much more complicated.

2.2.2 State channels

This section describes how to build a universal network for off-chain-transaction processing, which allows implementation of a second-level layer for the majority of existing blockchain systems, provided there are some minimal integration improvements on the blockchain side.

Let's assume there are two members, Alice and Bob, who want to install a payment channel for exchanging tokens in some blockchain (token - TKN, blockchain). Alice and Bob do not trust each other. At the same time, the tokens that they want to exchange already exist and are serviced by a third-party decentralized solution (blockchain), so Alice and Bob want to exchange them, not their equivalents. To do this, Alice and Bob install the GEO node and the necessary extension for communication with the external blockchain that serves their token.

- Through GEO, Alice (side A) and Bob (side B) form a multisig transaction to open the multisig address in the blockchain. The purpose of this operation is to atomically create an address simultaneously belonging to both A and B, where both parties will send funds (not necessarily in a proportional amount), and to specify the addresses to which the funds will be withdrawn in the event of the closure of the channel.
- Since the operation is created and signed by both parties, the situation when one party's funds are frozen in the channel while the other has not yet arrived is impossible, so additional temporary blocking of funds is not necessary. Since the multisig address is derived from existing blockchain addresses, it is always possible to verify the validity of the signature of both parties in the transaction. The correctness of this operation is blockchain's responsibility. Creating a transaction through GEO allows parties to agree on an operation outside of the blockchain network, so they can send only one channel opening transaction to the network instead of two operations for opening and refilling the channel by each of the two parties.
- After the channel establishment, the parties start exchanging assets in the GEO network. At the commit stage in GEO, the parties create and sign a transaction in a format suitable for export to the blockchain. Herewith the current state of the balances of the parties and the transaction sequential number (0 for the first transaction in the channel, N is subsequent, N tends to infinity) between them are fixed in the transaction. When exported to the network, any transaction of this kind triggers the mechanism for the channel closure.
  - Important: Since the parties are fixing a state of a channel (the balances of the parties), they must conduct only one operation at one point in time. Otherwise, the integrity of history and balances can be violated, and the parties will have an opportunity of unfair behaviour in the network. This condition is checked by the module for the GEO protocol, which implements communication with the blockchain.
  - Important: Since the GEO Protocol implies the possibility of force canceling the operation (force rejection by an observer), the signature of the state for the blockchain must be followed strictly after irreversibility of the operation is guaranteed.

Once one of the parties decides to close the channel, it exports the last transaction from the history to the blockchain network. As a result, the blockchain starts the procedure of the channel closure. In this case, the funds are not transferred to the settlement addresses indicated in the settlement transaction instantly, but a waiting procedure of 500 blocks is started (or any other block interval equivalent to time sufficient for notifying the counterpart, since different blockchains have different conditions for finding blocks), and so there are two possible outcomes:

- Cooperative closure — the parties mutually agree to close the channel. The party that initiates the closure (let's assume it was side A) sends the last transaction to the network, the party that confirms the closure (side B) waits for the closure request on the network, checks the balances of this transaction, and, if everything is correct, sends a transaction to the network confirming the closure of the channel (a separate type of operation for which only the signature of the response party is required). After that, the funds from the multisig address will be sent to the addresses indicated in the settlement transaction, and the channel will be considered to be closed.

- Important: After the channel is closed, the parties can no longer carry out operations on the GEO network. The GEO protocol module that communicates to the blockchain is responsible for verifying this condition.

- Non-cooperative closure — the party that must confirm the closure (side B) by receiving a channel closure request published in the blockchain network conducts a condition check and finds that the balance specified in this request does not match the expected balance (party A has sent to the network an outdated transaction, possibly for the purpose of fraud). In this case, side B sends its version of the last transaction to the blockchain. Having received two (or more) requests to close the channel, the blockchain prefers the request an internal transaction number of which is larger (a sign of a newer operation), and restarts the waiting procedure from the receiving side (this time for side A). Thus, in case of suspicion of fraud, the parties may exchange transactions in the network with the hope that the blockchain will accept the transaction that is favorable for one them as the final one. However, this is a finite process. First of all, it is expensive; also, sooner or later, one of the parties will run out of signed operations with a higher number.

In the worst-case scenario, the funds will be unlocked in the blockchain after 500 blocks. Important: In the proposed version, there is no punishment for the participants of the network for sending outdated transactions. After all, the blockchain will always set the current balance of the parties based on the last published transaction confirmed by both parties.

Advantages of the state channels:

- Universality. Extensions for different blockchains may use cryptography and solutions adopted in their ecosystem. The proposed solution does not impose a need for a common format for everyone.

- Ease of implementation. The proposed solution requires support of relatively simple primitives on the blockchain side. According to our observations, most modern blockchains can implement them through either a smart contract, or by customizing the internal logic.

2.2.3 Composite channels

There are two types of channels in the GEO Protocol, namely trustlines and composite channels (a combination of trustlines and state channels). A trustline reflects an equivalent of the particular asset (stored in registry of equivalents). At the same time, trustline obligations are not tied to any external ledger.

A composite channel has a trustline as a fundamental technology, and is also complemented with the logic of ledger to which the channel itself is attached. A channel is able to use native technology of the platform to conduct the operation. For example, Bitcoin's channel can use HTLC, and a channel on Ethereum can use the technology of common state channels for this platform. The channel is opened between two nodes. The number of channels is approximately equal to the number of nodes involved in the operation.

Thanks to this feature, users have a unique opportunity to build a composite channel infrastructure that combines the scalability of trustlines, trustlessness of state channels and the possibility of using an unlimited number of various tokenized assets, as well as to create equivalents of non-tokenized assets.

This complex system of the GEO Protocol will allow building various applications for exchange of different assets and equivalents (such as DEX with fast cross chain connectivity). For example, let's say you need to pay in dollars for chicken at the Chinese market, yet the seller only accepts yuan: the GEO Protocol will solve this issue. Or, on one channel you can receive electricity in watts, and on another you can pay for this electricity (the bill is a smart contract). All this is made possible by composite channels. The advantages of the composite channels are:

- Flexibility that allows the use of different types of connections and combinations thereof

- Cross-chain interoperability — simultaneous operations with several assets

- Scalability — one does not need to wait until a block is mined by each ledger, since all actions occur off-chain

3 Technical stack

3.1 Payment algorithm

This algorithm is responsible for the payment itself. In other words, this is how a transaction actually occurs. The process starts with discovering possible network paths between coordinator and receiver. After that, the algorithm calculates the maximum payment flow (i.e. how much money could be sent through each of the discovered paths). This calculation is based on channel capacities of each individual middleware node located on the way from the coordinator to the receiver. If no path was found, the payment algorithm execution stops.

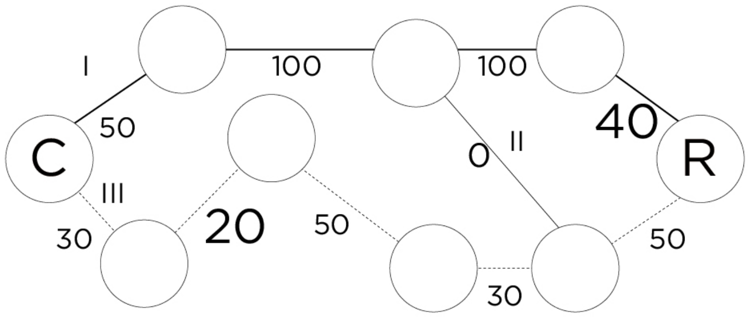

Figure 6: Possible paths and maximum flow capacity

In the next step, coordinator attempts to reserve the required transaction amount on several (or all, if needed) discovered paths. The reservation means freezing the necessary part of the capacity of each trustline to ensure that no other possible transaction, which may occur at the same time using the same trustline, would interfere with the ongoing one. To do so, coordinator sends a reservation request to the first node down the path and waits for a response. If no response is received during a predefined time frame, then the whole path that contains this node is considered "unavailable" (this is true for each consequent node on a path). Otherwise, the algorithm proceeds.

When the reservation request is received, each middleware node checks the possibility to approve it, so it performs one of the three possible actions:

- If the amount available in the requested trustline is sufficient, the node accepts the reservation, and sends the response to the coordinator, confirming the whole sum.
- If the amount available in the trustline is insufficient to cover the whole requested sum, the node accepts the reservation partially, i.e. sends back the confirmation response but only for the sum that is available.
- Otherwise, the node rejects the reservation.

This approval procedure is repeated by every consequent node down the path. After completing this step, the coordinator sends the Final Reservations Configuration (FRC) to all the nodes involved. The node cancels the operation if no FRC was received. Otherwise, the node must validate it (the validation procedure is described below). After receiving the FRC, the node will do the following:

- Create debt receipt for the neighbouring node with the amount that has been previously reserved in the corresponding trustline.
- Sign it with a one of public keys from the pool of its public keys with the counterparty.
- Send signed debt receipt to the neighbouring node.
- Receive signed debt receipts from all neighbours.
- Check all received debt receipts according to the following requirements:
  - The receipt amount must be equal to the previously reserved amount

- Receipt's Transaction ID must match the current Transaction ID Receipt's Equivalent ID must match the Equivalent ID of the reservations, in the related trustline
- There are no duplicates of the signed debt receipts for this operation on the node

Now the reservations are complete, and we could be sure that the whole path will be available (as long as no middlewaries will suddenly go offline, of course). Reservations will be canceled automatically after a few seconds of possible waiting to prevent long funds freezing. Nodes also must submit their Public Keys (that will be used for transaction signing) to the coordinator. In turn, the coordinator generates Participants Public Keys List (PPKL) of each node involved in the transaction and sends it to them. It is necessary to ensure the possibility to check whether the transaction was signed by the right node.

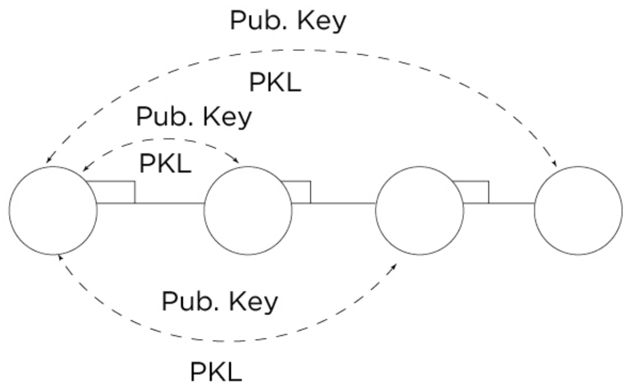

Figure 7: Keys exchanging

After checking this list, each node serializes the data to a stable storage and signs the transaction. Serialization is the standard process that is used in traditional database systems to prevent information loss. Simply put, one makes a copy of data and, if problems emerge, the data can be recovered from this backup.

When serialization is done, the coordinator waits for the signed vote lists from the nodes (here all nodes confirm that they are ready to perform the actual transaction).

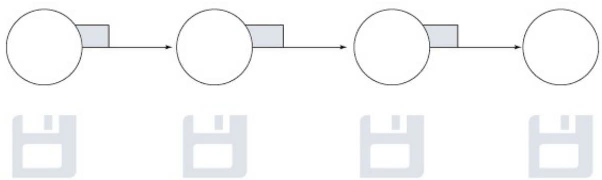

Figure 8: Serialization

The coordinator merges every newly received vote list with the common one. Both the coordinator and the receiver serialize data to the stable storage as well, and also sign the transaction. If signatures of all participating nodes were received, the transaction is considered approved. In this case, the coordinator stores the final vote list in its trustlines registry, and propagates it to all nodes of the transaction. Then, they return all reserves into balances on all trustlines/channels related to the transaction, thus executing the transaction.

Now each node has all the needed proof for requesting debt refund from its counterparties. This process has to be initialized with each neighboring node the balance of which has been changed. After this final step of the payment algorithm, the transaction is considered completed.

3.2 Maximum flow prediction algorithm

Each node in GEO stores information only about those nodes with which it has trustline connections. This allows solving the scalability problem and achieving high TPS of the network. In order to transfer funds between two nodes that are not connected with a trustline directly, and to calculate the maximum payment flow for such a transaction (it is needed because you can't send the amount of funds you want due to limited incoming and outgoing amounts on each intermediary trustline), the sender node needs to obtain network topology that allows it to perform such a calculations.

Let's consider the process of obtaining topology in more detail. Let's suppose that two nodes that are not directly connected to each other want to make a transaction. Both the coordinator and the receiver know their first-level connections only, the nodes with which they have direct trustline connections. The coordinator sends a request for a topology to the receiver. The receiver sends a reply to the coordinator about its first-level connections and the amount of funds from each node it can receive. Amounts are calculated based on the information about incoming and outgoing trust, as well as the current balance. The receiver then sends requests to its own first-level connections, so they could also submit their topology to the coordinator. Those first-level connections in turn send these requests to their first levels, that is, to the second level in relation to the receiver. This level is final. It receives request and sends information about its topology to the coordinator. In total, the topology information is sent by the receiver and its first- and second-level connections. Same thing happens with the first- and second-level connections of the coordinator. So we have the topology of all 6 hops (the coordinator, the first level of the coordinator, the second level of the coordinator, the receiver, the first level of the receiver, the second level of the receiver). In other words, the coordinator node collects information about its neighbors and neighboring neighbors, and the receiver node acts the same way, and sends its information to the coordinator, thus forming a topological map.

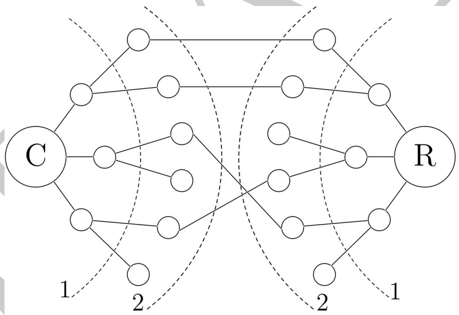

Figure 9: Coordinator, Receiver, their first and second levels

After that, using the modified Edmonds-Karp algorithm[21], the maximum flow is calculated. In terms of statistics, the average time to collect topology for a node with 10-20 trustlines is approximately 200 milliseconds, and in some cases, when the node has 200 trustlines, the time can take up to 1 second.

If the distance between the coordinator and the receiver is less than 6 nodes, one node may be the first level for the coordinator and the second for the receiver (or vice versa). This node may have to send information twice.

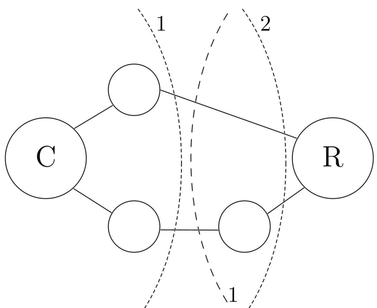

Figure 10: Overlapping coordinator and receiver levels

To avoid this, the nodes save the cache in the GEO network. The cache records information about to which node and which data has already been sent. When a new query is received, a comparison is made and, if some data has already been sent, the node sends only the information in which the changes have occurred (if there were any). After a certain period of time, the cache is deleted.

3.3 Routing algorithm

In the GEO network, nodes that are linked via a trustline are not necessarily connected by a common physical data channel. Rather, the majority of network participants are expected to use the existing Internet topology (especially the 3G/4G network) for efficient routing of their assets.

A problem associated with logical topology is, in particular, that it is much more prone to changes and reconfiguration than static networks. If we assume that the first-level connections could be created/changed/deleted/recreated with little probability, then, with the increase of the remoteness level, the probability of such a change tends to 1.

Simply put, from the point of view of any network member, the connection at its 5th-6th level of remoteness will be changed relatively often, and its routing tables will become obsolete over time, which means that there is a need of an update mechanism. The social and economic nature of the network topology will contribute to the frequent changes in the topological tree; therefore, in addition to updating the routing tables, the question of the time effectiveness of such a solution arises. Even if we hypothetically assume that routing tables can be placed in the memory of the end-user devices, and that their updating takes up to one day (which is pretty time-effective for the projected number of entries 150^6), then, because of the changes frequency this process turns into a kind of "streaming of topology changes." There are three obvious shortcomings of this kind of solution:

- The amount of traffic consumed
- The need to constantly keep large amounts of irrelevant information (routing tables)
- Permanent inconsistency of information from routing tables for nodes at remote levels

When faced with similar tasks, some systems opt to delegate computing power to third-party services, which entails the direct need for their financial motivation and, therefore, affects the network fee. The GEO Protocol aims to create a decentralized solution that maximizes usage of the network nature to create a map of possible payments while ensuring the formation of the first-data portions during the first few seconds after a transaction is initialized, rather then delegating computing power to third-party services.

Ability to predict the maximum flow

One of the decisive factors in decentralized credit networks is the ability to quickly predict the maximum payment flow between any pair of network nodes. The difficulty is that with an increase in the number of participants and operations on the network, the frequency of change in channel states increases proportionally. The operational complexity of predicting flows also increases exponentially, and in some cases quadratically, just like in the difficulties in routing described above.

At the same time, while it is possible to cache network topology for a relatively long period of time (for example, by using the above-mentioned routing tables), the state of the channels is very difficult to cache because each cached value on one node leads to potential distortions of information about the payment flow on other nodes. In general, the nature of these distortions depends on a number of factors, such as the length of the cache, the way information is collected from the network, etc.

Proposed solution

The solution offered by the GEO Project aims to rethink the way of the maximum flow prediction and combine it with the suitable payment paths between participants finding process.

A high-level solution requires several important refinements: the coordinator and the receiver can be mutually addressed at the level of the data network (the Internet in most cases): the coordinator can send a data packet directly to the receiver, and vice versa.

Algorithm

Next is a high-level description of the algorithm for predicting the max flow and collecting payment paths. The above description is for informational purposes only (for a more detailed description, see the technical description of the maximum flow prediction algorithm).

- Coordinator analyzes its first level of channels for maximum capacity of sending funds to the network. If it equals 0, the operation is interrupted, because, in relation to any node, its maximum capacity is zero.
- Coordinator sends a message to the receiver informing it about the beginning of the flow prediction process. The receiver performs a similar check for the maximum incoming flow on their side. If it is 0, the receiver tells the coordinator that the operation should be canceled, since none of the channels/trustlines can be used.

At this stage, the maximum flow limits can already be outlined: they cannot be greater than the sum of all outgoing flows on the coordinator side, and at the same time they cannot be larger than the sum of all incoming streams on the receiver's side.

If both the coordinator and the receiver have a non-zero potential flow, they both begin to collect network data simultaneously. The sequence of operations performed is as follows:

- Coordinator sends a message initiating flow prediction to the nodes that have channel/trustline with non-zero outgoing flow. This message is as short as possible (just a few bytes) and contains only the operation type, a short identifier and an address (or GNS record) for sending the return message.
- Having received a request to predict the flow, each node produces a similar operation with its own first line of connections, except for the following cases:
  - The sender of the request does not get the answer.
  - The message contains information about the maximum flow at the current level (how much money could be sent down by the path's part that the message has already traveled through). At the following levels, this indicator could be reduced in accordance to the payment flow of those levels.
  - The message includes information about the current message level (distance in hops). Due to this parameter, it is possible to limit the maximum distance of the packet. According to the protocol, this package must pass only 3 hops (maximum path length is 6 hops so when both coordinator and receiver start topology collection from their sides, each of them need information only about their first, second and third levels, and if they combine their 3 hop topologies, they will get the whole 6-hop look of the network).
- The receiver performs a similar set of actions with its first level of connections.
- The purpose of the algorithm is to collect information about the capacity of the intermediate channels on the path from the receiver and the coordinator to their 3rd level of nodes. Nodes, which will be common to both coordinator and receiver, send the collected packet back to the coordinator.
- By gathering the network topology and capacity of the channels, the coordinator builds a topology and, with the help of the modified Edmonds-Karp algorithm, predicts the maximum flow.

Thus, the coordinator, as the initiator of payment, takes the greatest computing load, as well as the higher load on the network and traffic. All intermediary nodes and the receiver perform trivial actions and spend a minimum of their computing resources.

3.4 Cyclic clearing algorithm

To better understand the cyclic clearing algorithm, let's consider its operation principles in the following example. Suppose that we have three sides: Alice (side A), Bob (side B) and Charlie (side C). In our hypothetical case, Alice owes 10 TKN to Bob, Bob owes 10 TKN to Charlie, and Charlie owes 10 TKN to Alice.

When each party pays their debt, the overall balance of the participants will not change. Accordingly, by accepting what no one owes to anyone or, in other words, by completing the cycle, we can avoid the need for additional transfer of funds, thereby reducing the load on the network.

The GEO system looks for such potential mutually settled cycles of debt obligations, and clears them automatically. The cycle itself is a simplified payment along one path, where the coordinator and the receiver is the same node. That is, payment occurs in a circle, writing off balances accumulated as a result of past actions of other nodes. Since GEO supports payments length of up to 6 hops, its cycles may be as long as 6, 5, 4 or 3 nodes.

Let's consider the procedure of the algorithm in more detail in an example of a cycle of 5-6 nodes. Since a node only knows its first-level connections, it is, first of all, necessary to collect the wider network topology. Suppose that a node wants to build the clearing cycle and then close it.

In the first-level connections of the node, there are creditors (those who were paid by the node), and debtors (those who paid this node). The further course of action is as follows: the node sends packets to its debtors and creditors. Each packet contains information about the path that this packet has already passed, maximum payment flow along this path, how many hops this packet has to go through, how many hops this packet already went through, whether this packet has to be sent to debtors or creditors (packets for debtors can be sent only to debtors, the same goes for creditors). After receiving the packet, every node modifies the current number of hops the packet already went through, its current path, its current payment capacity, and sends it to its own first-level connections (which are the coordinator's second-level connections). This process repeats until the maximum number of hops from the coordinator's node are achieved.

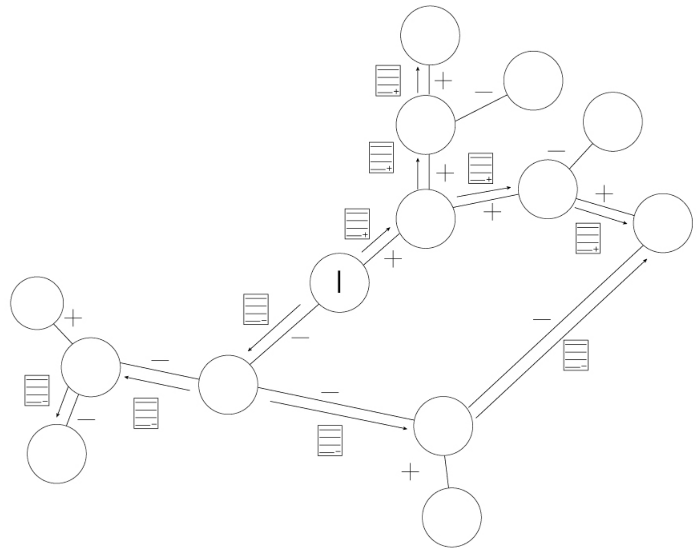

Figure 11: Example of packets flow

When this happens, every packet returns to the coordinator. So now the coordinator node has gathered enough information to build the network topology that surrounds it. The information contains the list of neighboring nodes of up to 5 or 6 hops deep, as well as paths between them with negative balances, paths with positive balances, and payment capacities of each of these paths. Then, to build the clearing cycle, the coordinator looks if there are any pairs of debtor/creditor paths that involve the same nodes. When it finds such pairs, it can build the cycle. This algorithm is run once a day, because there are potentially not many cycles for 5 and 6 nodes.

The algorithm for 3 and 4 nodes is slightly different. It starts after each payment. If the node has transferred funds, it potentially has a promissory note. For a better understanding, let's consider the following example. In our network, there are 6 nodes with a topology like this:

After a payment, node I owes 50 to node A. To build a clearing cycl,e I now is looking for its neighbors that owe I (E and C in our example). Then, it checks whether these nodes are neighbors of node A. In our case, C and E are both neighbors. After that node, I (the initiator node) sends the list of potential cycle participants (C and E) to node A, which in turn chooses from this list of participants those to whom A owes (only C in our example) and returns the report with which nodes could be used for the cycle, and A's balances with them. When response is received, node I sees that it can build the cycle I → A → C → I, where I owes A the sum of 50, C owes I the sum of 100, and A owes C the sum of 60.

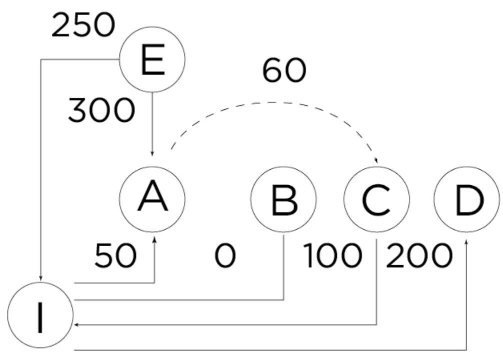

Figure 12: Network topology example

Maximum closing capacity is the smallest sum in the cycle, which is 50 in our example. So, when the cycle is closed, I will settle the sum of 50 with A, A will settle 50 with C, and C will settle 50 with I. The picture below demonstrates the resulting outstanding balances (those that should be actually repaid) after the clearing cycle is closed:

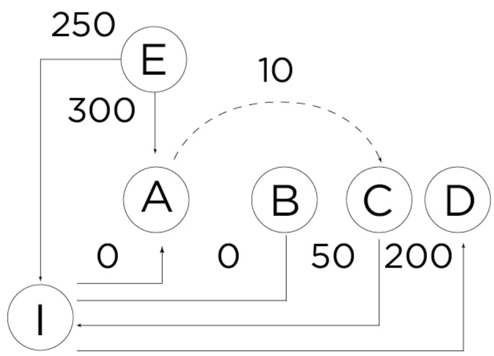

Figure 13: Result of cycle clearing

3.5 Cryptography

The cryptography method is the Lamport signature[20], which is resistant to quantum computing, matches the network requirements regarding the amount of traffic used, and the digest of which is sufficiently optimal in size.

Current cryptanalysis of the Lamport signature reports some redundancy in relation to the brute force attack performed by quantum computers. In turn, this leads to the redundancy of traffic used in each operation, protected by this signature. At the same time, in order to prevent a significant decrease in the crypto resistance due to an incomplete adherence to the classical algorithm, we decided to implement the classical Lamport signature algorithm without weakening and decreasing the digests received. In order to save traffic, the Lamport signature protects only those operations that lead to a change in balances on trustlines. Operations on the exchange of service information are protected by more crypto-weak functions (SHA256), which is overall sufficient at the current technological level. A certain class of operations is not protected by cryptographic methods at all as there is no need.

The pilot version of the protocol does not provide network participants with the ability to change cryptographic primitives, but it seems reasonable to give them the opportunity to independently choose cryptographic primitives that will be used for different kinds of counterparties. This will allow network members to make their own decisions on security issues and traffic usage. At the same time, the standard version of the protocol will be released with certain cryptographic primitives set by default, which, according to the authors of GEO Protocol, are sufficient for the safe interaction of network participants for today's technological level.

The Lamport signature is one-off and should not be reused to confirm more than one operation, since with each additional operation, the risk of compromising it increases. Therefore, GEO Protocol introduced the mechanism for proactive signature pre-generation at the time of the opening of the LD. By default, the number of pre-generated signatures is 1024, which allows us to confirm (or reject) 1024 direct or intermediary transactions on the network. Upon exhaustion of this limit or its approach to completion, participants repeat signature pre-generation (reinstall the context of trust). It is wise to conduct this procedure at the slightest suspicion of compromised data of one of the counterparties. Thus, the amount of data required to store information on one LD is about 1024 * (16kB + 8kB) = 24MB. GEO Protocol assumes that both counterparts mutually trust each other's public keys.

4 Use Cases

The GEO protocol provides an infrastructure for applications of various types and purposes. Due to the design of the system, channels and trustlines can be used not only for payments, but also for useful computing, information exchange, voting, etc.

A. Payment solutions

Non-blockchain, fast, real-time, double-spending proof, distributed multi-attendee payment crypto-protocol with time predictable 100% participant consensus. GEO Protocol helps users safely send and receive payments in P2P marketplaces. It greatly enhances the buying/selling process with decentralized escrow for secure payments, third-party dispute resolution, and very low transaction costs.

B. Interoperability protocol

GEO protocol may act as a cross-chain protocol, enabling interaction and interoperability among different blockchains. This makes instant payments across a network of participants easy and inexpensive.

C. Cross-chain DEX

The structure of the network allows the exchange of assets between two participants quickly and safely. The very process of exchange is similar to the technology of atomic swap. Therefore, it is expected that one of the first applications on the GEO Protocol will be a decentralized exchange.

D.Identity management

GNS, which is part of the GEO ecosystem, allows us to upload user information and delegate access to personal data in the GEO ecosystem. Thus, it will be possible to create a digital passport.

E. Rating systems

The transitivity of trust reflects the amount of value that can be trusted to a node. It has a numeric measurement, so the platform gives a tool to evaluate the rate or amount of trust, loyalty or support.

F. Clearing systems

The Geo Protocol enables the possibility of implementing the automatic clearing with elimination of double expenditure.

G. IoT solutions

P2P protocol enables the scalable, secure, private and highly trusted method to perform IoT transactions with the participation of an unlimited number of nodes.

H. Mobile money operators transactions (protocol level)

P2P transaction protocol using SMS or any mobile app to perform transfer of funds. Nodes can be represented by phone numbers. The access to all online services is provided by duplication of node accounts to the cloud.

I. Mesh networks

Due to unique network architecture of GEO Protocol, it will be possible to create an infrastructure for mesh networks in the future.

J. dApp scaling solutions

The technology of state channels is actively developing, allowing more complicated operations to be carried out in the blockchain. In the future, many decentralized applications will carry out the main part of their calculations, which do not require the protection by entire blockchain in these channels.

K. Loyalty programs

We are providing a platform for building loyalty programs and a developer interface, enabling customization of loyalty application for any need. Using the GEO Protocol, commercial brands will benefit from simple development and customization, low management costs and the elimination of liabilities associated with unredeemed items.

L. Delegated democracy

With GEO Protocol, one can create a voting system and a governance mechanism for decision making. In addition, there is the possibility of anonymous delegation of votes in such a way that no one could possibly know what power their voice really has.

M. Decentralized credit networks

GEO Protocol allows us to implement a system of P2P lending, credit unions and credit systems with a guarantor. It is also possible to create a social and credit network — an alternative economic system built on social relations, which includes all of the above elements of credit and payment networks.

5 Conclusion

In this work, we presented the GEO Protocol, a decentralized P2P network for fast and secure exchange of various data (financial and non-financial). It brings together existing legacy financial systems and data registries.

GEO Protocol implements the mechanics of multihop-transaction processing between several participants. By default, GEO Protocol implies the consent of all participants to cooperate on the principle of debt obligations using the technology of trustlines.

It is also supplemented with components for implementing the logic of asset exchange in the absence of trust and/or the delegation of arbitrage to an external service through state channels — an offline scaling solution that allows implementing a second layer for most existing blockchain systems.

In order to make use of and eliminate the limitations of existing technologies, the GEO Protocol provides an opportunity to use composite channel infrastructure that combines the scalability of trustlines and trustless state channels, uses an unlimited number of various tokenized assets and creates equivalents of non-tokenized assets.

Due to its structure, the GEO protocol allows building of an infrastructure for various public applications and solutions, such as: payment applications, cross-chain decentralized exchanges (DEX), voting services and loyalty programs, credit and clearing systems, and interactions between different blockchains and IoT solutions.

References

1 Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, Dawn Song, and Roger Wattenhofer. On scaling decentralized blockchains. In FC, 2016.

2 Stefanie Roos, Pedro Moreno-Sanchez, Aniket Kate, Ian Goldberg. Settling Payments Fast and Private: Efficient Decentralized Routing for Path-Based Transactions. https://arxiv.org/pdf/1709.05748.pdf

3 Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. http://bitcoin.org/bitcoin.pdf

4 Andrew Miller, Iddo Bentov, Ranjit Kumaresan, and Patrick McCorry. “Sprites: Payment Channels that Go Faster than Lightning”. http://arxiv.org/abs/ 1702.05812.

5 Jeff Coleman. State channels. https://www.jeffcoleman.ca/state-channels/

6 Joseph Poon and Thaddeus Dryja. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments, 2016. https://lightning.network/lightning-network-paper. pdf.

7 Patrick McCorry, Surya Bakshi, Iddo Bentov, Andrew Miller, and Sarah Meiklejohn. “Pisa: Arbitration Outsourcing for State Channels”. In: IACR Cryptology ePrint Archive 2018. https: //eprint.iacr.org/2018/582.

8 P. F. Tsuchiya. The Landmark Hierarchy: A New Hierarchy for Routing in Very Large Networks. In SIGCOMM, 1988. http://www.cs.cornell.edu/people/francis/p35-tsuchiya.pdf

9 M. J. Neely and R. Urgaonkar, ”Optimal Backpressure Routing in Wireless Networks with Multi-Receiver Diversity,” Ad Hoc Networks (Elsevier), vol. 7, no. 5, pp. 862-881, July 2009.

10 Tassiulas and A. Ephremides, ”Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks, IEEE Transactions on Automatic Control, vol. 37, no. 12, pp. 1936-1948, Dec. 1992.

11 Stefan Dziembowski, Sebastian Faust, Kristina Hostakova. Foundations of state channel networks. https://eprint.iacr.org/2018/320

12 Vitalik Buterin. A next generation smart contract and decentralized application platform. https: //github.com/ethereum/wiki/wiki/White-Paper, 2013.

13 K. Croman et al., “On scaling decentralized blockchains”, in International conference on financial cryptography and data security, 2016. https://fc16.ifca.ai/bitcoin/papers/CDE+16.pdf

14 Celer Network: Bring Internet Scale to Every Blockchain. https://www.celer.network/doc/ CelerNetwork-Whitepaper.pdf

15 Castro, M., Liskov, B., et al. Practical byzantine fault tolerance. In OSDI (1999), vol. 99, pp. 173–186.

16 Christopher Copeland and Hongxia Zhong. Tangaroa: a byzantine fault tolerant raft. http://www.scs.stanford.edu/14au-cs244b/labs/projects/copeland_zhong.pdf, 2016.

17 Stefan Thomas and Evan Schwartz. A Protocol for Interledger Payments. https://interledger.org/interledger.pdf, 2015.

18 Ryan Fugger. Money as IOUs in Social Trust Networks A Proposal for a Decentralized Currency Network Protocol, 2004. http://archive.ripple-project.org/decentralizedcurrency.pdf

19 Pavel Prihodko, Slava Zhigulin, Mykola Sahno, Aleksei Ostrovskiy, and Olaoluwa Osuntokun. Flare: An Approach to Routing in Lightning Network. https://bitfury.com/content/ downloads/whitepaper_flare_an_approach_to_routing_in_lightning_network_7_7_2016.pdf

20 L. Lamport, Constructing digital signatures from a one-way function, Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, Oct. 1979.

21 Edmonds, Jack; Karp, Richard M. (1972). "Theoretical improvements in algorithmic efficiency for network flow problems". Journal of the ACM. Association for Computing Machinery. 19 (2): 248–264

22 E. W. Dijkstra. A note on two problems in connexion with graphs. Numerische Mathematik, 1:269–271, 1959

23 George Danezis, Ian Goldberg. Sphinx: A Compact and Provably Secure Mix Format.

24 Olaoluwa Osuntokun. AMP: Atomic Multi-Path Payments over Lightning. https://lists.linuxfoundation.org/pipermail/lightning-dev/2018-February/000993.html