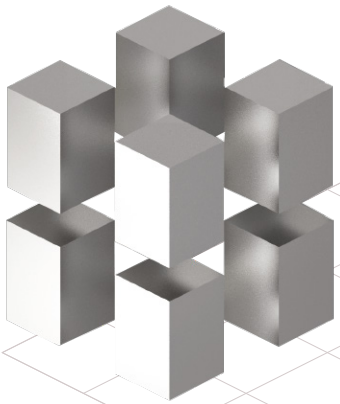
FOAM Technical Whitepaper



Draft 0.4

Foamspace Corp
info@foam.space

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

Revision: 7c33cd8

This technical whitepaper is a draft version and is subject to change and further revision. The statements made in this document should not be considered final. This document is only intended for persons who receive it directly from FOAM.

Contents

DRAFT

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

1 Introduction and Problem Statement

Foamspace is developing and deploying a decentralized Proof of Location protocol. We are committed to make availible spatial protocols, standards, and applications that offer a higher level of security and more resiliency than conventional geospatial technologies. Security and privacy threats threats related to location and positioning in many verticals, such as Internet of Things (IoT) technologies has not been sufficiently addressed so far. When considering blockchain and smart contract technology, secure localization and location verification is absent from the standard currently at play.

1.1 The Difference Between Static and Dynamic Proof of Location

As technology evolves and changes, maps need to change too. FOAM secures physical space on the blockchain, harnessing the power of Ethereum with a cryptographic software utility token used to provide computational work and verification to the network.

The goal of the Proof of Location solution is to provide the framework and infrastructure to support a decentralized, privacy preserving, highly accurate, censorship resistant alternative to GPS. Location is a fundamental infrastructure protocol needed to achieve the full vision of a decentralized 'web3' economy and can foster an ecosystem of applications built on top of a verified location standard.

Proof of Location is the primary utility arising from use of the Crypto-Spatial Coordinate and Spatial Index Visualizer elements of FOAM [see our website for more information]. Proof of Location will inherently be an iterative process which involves the use of token curated registries by users to contribute, verify and determine Proofs of Location.

In general any system for Proof of Location will need to bootstrap itself into existence from the weakest self attestation claims to strong fraud proof authority based claims. For FOAM the starting point is Static Proof of Location for Geographic Points of Interest, places and locations for a consensus driven map of the world. Yet a Dynamic Proof of Location system can account for Proof of Location in space and over time by accounting for mobile and dynamic location customers.

Foamspace hopes that the Cartographers and users will contribute the necessary individual work, resources, and effort themselves to contribute to the ongoing community-driven growth and upgrade of this important cartography project. With the addition and use of necessary radio hardware, as described in more detail below, Proof of Location could be expanded to further prove location status through a time synchronization protocol intended to ensure continuity of a distributed clock, whereby specialized hardware can serve as a Zone Anchor and synchronize nodes' clocks over radio to provide location services in a given area.

DRAFT

What follows is the potential outline of an expanded form of Proof of Location which is intended to supplement Static Proof of Location, potentially providing Proof of Location functionality to transitory things. It is described here for illustrative and descriptive purposes only and on a non-promissory basis. Its ultimate adoption would depend on a variety of factors, including FOAM user adoption, the organic community-driven expansion of the network and the requisite addition of radio hardware by individual FOAM users. As such, if or when it is adopted cannot be stated with any certainty.

This technical whitepaper is a draft and is provided for informational purposes only. For information regarding Foamspace, its platform or the FOAM token, please consult the current version of the FOAM Whitepaper. This technical whitepaper should be read in conjunction with the current Foamspace whitepaper.

For the remainder of the paper, Proof of Location will refer solely to Dynamic Proof of Location.

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

1.2 The Components of Dynamic Proof of Location

The goal of the FOAM Proof of Location protocol is to provide the framework and infrastructure to develop a decentralized, privacy preserving, highly accurate, censorship resistant alternative to the Global Positioning System (GPS). FOAM is a shared and open protocol that is not rent seeking and does not charge any centralized fees. We see FOAM as a fundamental infrastructure protocol needed to achieve the full vision of the web3 economy and that can create a winner take all market that fosters an ecosystem of applications built on top of a designed location standard.

The measurements of and problems of authenticating Time and Space are intricately intertwined. Our approach to Proof of Location rests on an autonomously self-stabilizing time synchronization protocol that is designed to ensure continuity of a distributed, Byzantine fault tolerant (BFT) clock. With a high-precision BFT clock signal, the network can use the relative geometry between beacons to compute a node's distance, thereby enabling a secure, spatially distributed location system

Features of FOAM Proof of Location

1. Trustless: Byzantine fault tolerant clock synchronization
2. Independent: Does not rely on GPS
3. Open: Anyone can utilize the network or offer utility services
4. Accountable: Economics structured to ensure honest behavior, verified with fraud proofs
5. Incentivized: Service providers remunerated for extending localization and verification zones

FOAM is designed to be a solution for blockchain based economies. Smart contracts that will execute autonomous code with geospatial data as an input will require secure location verification. If tokens are at stake the incentive to spoof location in a trustless system are raised. GPS is not suitable for blockchain based applications that will need precise and reliable location and is trivial to spoof on the client side.

Dynamic Proof of Location can provide consensus on whether an event or agent is verifiably at a certain point in time and space by producing a digital authentication certificate that is designed to be fraud proof, called a Presence Claim.

FOAM provides the tools, market framework and incentives for service operators to set up specialized hardware beacons to broadcast coverage and participate in the protocol rules. In the Proof of Location protocol these nodes are known as Zone Anchors and or Zone Authorities, depending on the ability to host a full node. Together, these entities form a Zone, local to an area that offers location verification services to the market while also bonding tokens to insure accordance with the protocol rules.

Zone Anchor beacons running the FOAM protocol will need to provide accurate time synchronization for a set period of time in order to not be seen as faulty. A distributed system is Byzantine fault tolerant when the coordination of untrustworthy participants will always convey honest information, given more than 2/3 act honestly. It is important that a time synchronization is able to self-stabilize if a number of nodes are broken or malicious.

All radio frequency (RF) location systems rely on clock precision from beacons; radio transceivers. The most accurate approaches require clock synchronization. FOAM uses a BFT clock synchronization algorithm to provide the best possible support for Radio Frequency Time of Flight algorithms. The Proof of Location protocol is open for Zones to autonomously form and operate as utility providers that compete for transactions fees by providing location verification services.

1.3 A New Multi-Sided Marketplace

Token mechanisms and cryptoeconomics have for the first time made it viable to incentivize an autonomous and self stabilizing system of nodes that can synchronize their clocks to offer secure location verification with accountability enforced by smart contract protocol rules. Proof of Location is intended to utilize token staking incentives to grow network coverage and utilize a verifier set for fraud proofs, and

DRAFT

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

enforce protocol rules. Safety deposits allow for attributable byzantine behavior in the form of slashing conditions.

A Proof of Location system is needed as a crucial infrastructure in our decentralized future and it can open new marketplaces of privacy preserving location data. Use cases may arise supply chain's, real time mapping for autonomous vehicles, Internet of Things data markets and location based consumer applications that integrate with blockchain require secure proofs.

1.4 Definitions and Paper Organization

The following terms constitute the core vocabulary of the Proof of Location protocol, which are reintroduced in context throughout the paper

A **Zone Anchor** is a device with a radio transmitter, a local clock, and a public key. A node is capable of engaging in a clock-synchronization protocol, requires a connection to a gateway .

A **Zone Authority** is a distinguished gateway gnode with internet access and "sufficient" computational power to maintain a shared State Machine. It has the ability to determine if a the state machine is in sync (see below).

A **Zone** is the quorum that maintains clock sync for a given region. Four or more Zone Authorities form a Zone, the quorum that maintains clock sync for a given region. Once synchronized, the Zone can determine the location of a requesting node by using time of arrival measurements to verifiably triangulate position.

A **Shared State Machine** is maintained by the Zone Authorities in a Zone on the state of synchronicity. A consensus algorithm is used to vote on writing to the shared state machine.

A **Root Chain** is the blockchain where FOAM token bonds, deposits, rewards and penalties take place. Participants must interact with this chain to be given access to participate as part of the validator set of a Zone. For now, imagine that the root chain is the public Ethereum blockchain.

The **FOAM Token** is used as a safety deposit to participate in the protocol correctly and contribute work, security and computation. The staking of the token is needed on the root chain to be granted access to the shared state machine of any given Zone.

A **Service Level Agreement** (SLA) is the form of staking smart contract Zone Anchors and Zone authorities bond to in order to offer location verification services. This agreement dictates the terms of service.

A **Beacon Public Key** is the known public key of a mobile beacon that wants to purchase presence claims from Zones.

A **Presence Claim** (PC) is a set of counter-signed Requests with the same Nonce, which is intended to provide enough data to constitute an exact localization. Issued by Zone participants to BPK for a fee. The presence claim is subject to fraud proof computation before being authenticated as a proof.

The **Verifiers** are computational engines that incentivized to check the time logs of Zones for fraud and finalize Proofs of Location. The verifiers need to have at least the same computational power as that of an Authority inside a zone. Because Verifiers compute locations from the time stamped data they can be said to be mining triangulations.

The **Slashing Conditions** are set faults that constitute a violation of protocol rules for Zone participants and Verifiers, which results in a loss of the deposited funds.

Proof of Location is a fraud proof authentication certificate that serves as a first class object on the blockchain and represents that an entity was at a certain space in time.

This organization of the rest of this paper is to highlight survey many of the problems posed by insecure geolocation and to offer insight into our solution: a Proof of Location system that maintains Byzantine consensus throughout a distributed network of synchronized clocks, while creating markets for local generation of triangulated positional data. We will first explain secure localization and location

DRAFT

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

verification, positioning techniques and conventional systems, consensus designs, time synchronicity, our token mechanism and the multi-sided marketplace that we hope will emerge from Proof of Location as a result of user growth and adoption.

2 Secure Localization and Location Verification

Localization refers to the problem of identifying a node's spatial position within some defined coordinate system. In order to compute a given spatial position, geolocation systems integrate signal information from a set of spatially distributed sources, solving for either transmission distance or angular incidence parameters. Within Wireless Sensor Networks (WSN) the location of beacons, or nodes, in the network is very important for the monitoring and collection of data, as events logged by a sensor occur within a certain location bound i.e. earthquake or water quality monitoring. Further, many network operations depend on the location of sensor nodes i.e. navigational routing and location-based authentication.

Standard commercial localization systems can be classified by *node-centric* and *infrastructure-centric* methods, with the former requiring sensor nodes to compute their locations own their own and the latter relying on centralized infrastructure. The majority of localization systems employed explicitly assume trusted environments, where data can be expected to be obtained correctly. The FOAM Proof of Location protocol is node centric and self stabilizing while making no assumptions about a trusted environment.

Localization systems can also be classified as *range-free* and *range-based*. Range-free solutions only rely on the existence of beacon signals to produce localized nodes. Range-based solutions use point-to-point distance or angle estimations between the signals a pair of nodes and is a more accurate approach. Measurements include signal propagation times, signal strengths, or angle of arrival. Secure localization attempts to mitigate against adversaries that attempt to disrupt the localization process through vulnerabilities in signal interception, jamming, replay packets or modifications.

In a node-centric and range-based localization system, nodes may be faulty or malicious and intentionally report fraudulent location data even if secure localization is used. Data authentication is required in valuable networks and location verification schemes can be employed to check the correctness of sensors' reported locations. Distance bounding protocols are a secure solution for this problem with a construction of a prover convincing a verifier of an assertion, in this case location where a verifier wants to check if the prover is within a certain distance of a claim.

Location verification should be designed to prevent dishonest behavior and be resistant to distance fraud attacks. Industry research has found that rhe most promising solution is measuring the return time of flight of in a challenge–response protocol.

DRAFT

2.1 Triangulation and Positioning

Localization requires distance or angle estimation of signals between nodes and position computation, computing a nodes position based on the measured distance information. Triangulation is one of the most commonly deployed position estimation techniques, which uses the geometric properties of triangles to estimate distance and location through signal measurement techniques such as Time of Flight (ToF) or Received Signal Strength Indicator (RSSI). Triangulation can be further broken down and categorized into (1) lateration and (2) angulation methods. Techniques utilizing distance are known as lateration methods, whereas angulation methods measure the angle of arrival of a signal from the beacon nodes of a known location for localization.

In lateration, positions are calculated by measuring the distance between reference beacons with known locations and a target. There are two main approaches to discovering the distance between a target node and the known locations of beacon nodes, one of which is Time of Flight (TOF) based and the other is Attenuation based. Attenuation methods used the Received Signal Strength Indicator (RSSI) to model the loss of signal strength over time to calculate distance. This method is simple with limited accuracy but is one of the most popular.

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

With Time of Flight (ToF) based position estimations lateration is simply the product of time taken by the signal to travel from an object to the reference point. ToF utilizes various algorithms such as Time of Arrival (ToA), Time Distance of Arrival (TDoA), and Round Trip Time of Flight (RToF).

In ToA, the distance between a mobile target and reference points (beacons) is measured using the propagation time of the signal between the reference point and the target. Since the velocity of the signal propagation is known, distance can be computed easily. Thereafter, the position estimate of the target is found out using trilateration. ToA requires highly accurate synchronization of the clocks of the sender and receiver. When atomic clocks are not available a two-way time of arrival method is reccomended, where the round-trip time of a signal is measured from the sender in a call and response, handshake, fashion.

Trilateration determines the distance between the transmitter and the receiver, which can be obtained from the time of arrival (ToA), the time of flight (ToF) or from the received signal strength indicator (RSSI). Three reference nodes with known positions are used to locate an unknown node. The position of the unknown is the intersection of three circles formed by the positions and distances to the reference nodes. Synchrony is a prerequisite in using trilateration for determining the location and, consequently, distances to the intended object.

The time difference of arrival (TDoA) approach uses two signals that travel with different velocities. The receiver is then able to determine its location similar to the ToA approach. For example, the first signal could be a radio signal (issued at t1 and received at t2), followed by an acoustic signal (either immediately or after a fixed time interval. TDoA-based approaches do not require the clocks of the sender and receiver to be synchronized and can obtain very accurate measurements. The disadvantage of the TDoA approach is the need for additional hardware.

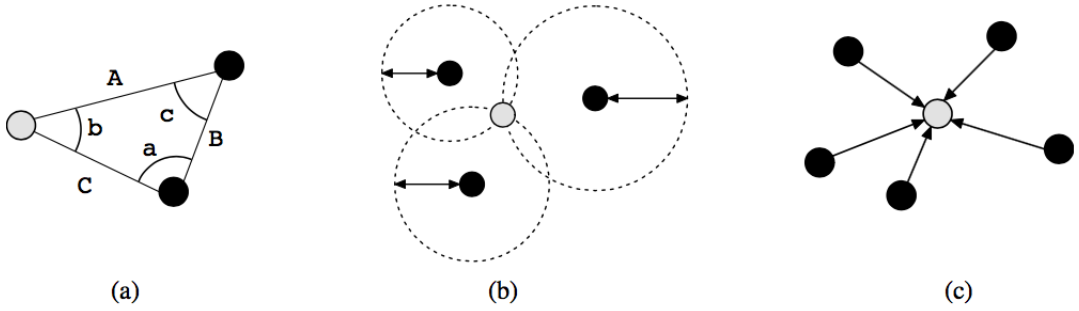


Figure 1: Position Estimation Methods: a) triangulation b) trilateration c) multilateration

Multilateration is quite similar to trilateration; however, the main feature to compute the location is the time difference of arrival (TDOA). The transmitters are synchronized to each other, whereas the receiver does not need to be. Thus, the location in this technique is the intersection of at least two hyperbolas (three antennas required). Multilateration is a technique for determining the position of a (mobile) device from a set of reference points whose positions are known, based on the ranges measured between the reference points and the device. The position of the device in two (three) dimensions can be computed if the device measured its distance to three (four) reference points. There are two multilateration schemes, surveillance vs. navigation. A surveillance regime requires 3 or more synchronized receivers, for navigation, those sites transmit and positional computation can be performed autonomously by the node in that signal field.

As distance is computed as a function of signal timing from spatially discrete sources, to most accurately computing such positions within geographic coordinate space precise clock synchronization is a requirement. Radio Frequency Time of Flight Distance-Bounding algorithms are the family of algorithms most robust against malicious actors. All Radio ToF algorithms rely on having precise clocks, some also rely on clock synchronization.

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

2.2 Global Navigation Satellite System Based Localization

In 1973, the United States was the first to deploy a geopositioning system that incorporated a constellation of orbital satellites to reach global coverage. To this day, satellite based localization systems remain the most prevalent, however, several state-backed competitors operate at various stages of Global Navigation Satellite System (GNNS) deployment. Russia's GLONASS program, initiated in 1976, is the only fully-realized geopositioning alternative. China began deploying their GNNS test network, BeiDou, in 2000, and has now mostly phased in a next generation satelite apparatus, COMPASS, that promises millimeter localization accuracy. The European Union's Galileo Project, begun in 2011, is now in partial deployment, and Indian Regional Navigation Satellite System.

The Global Positioning System (GPS) is a U.S.-owned utility consisting of 31 orbital satellites, base stations, and mobile receivers, which perform trilateration operations and made available for civilian and commercial use. Each GPS satellite goes around the world once every 12 hours, traveling roughly 7,000 miles per hour, 12,500 miles above sea level. What may not be immediately apparent, is that GPS technology works through time as much as it does space. Inside each satellite is a high-precision atomic clock, which sync regularly to master control stations on the ground. GPS receivers, common in today's smart phones, must pick up time-stamped signal data from a minimum of four overhead satellites. By using time stamps to calculate the time of arrival, a receiver can calculate a triangulated position. The accurate time in GPS signals is used to synchronized other systems, for example used to synchronize base stations in cell phone networks. A GPS receiver is able to receive the information constantly sent by the satellites, estimate its distance to at least four known satellites using ToA, and, finally, compute its position using trilateration. Once these procedures are executed, the receiver is able to determine its latitude, longitude, and altitude.

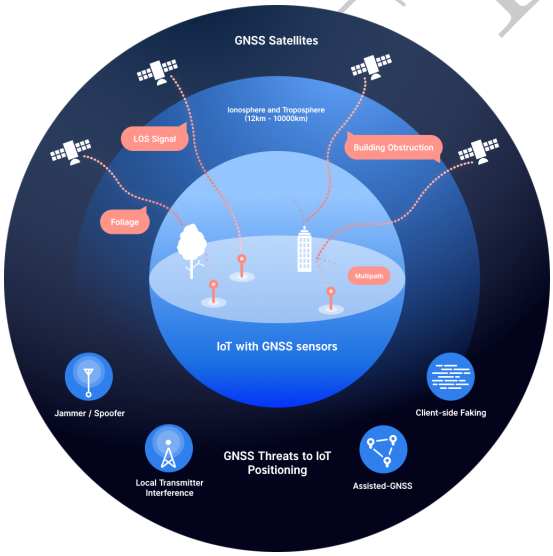


Figure 2: GNSS Threats to IoT Positioning

Ordinarily, GPS is incredibly reliable; however, problems and attack vectors with this system have become increasingly evident. Civil GPS is unencrypted, it has no proof-of-origin or authentication features, and despite dire warnings in the mainstream since at least 2012, the system remains extremely susceptible to fraud, spoofing, jamming, and cyberattack. Operational Control System (OCX), the next generation of GPS designed to address significant jamming and other cyber threats. However, the project has been continuously delayed, with a scheduled launch date now in 2022. Even so, the OCX design fails to address vulnerabilities, GPS competitiveness as a worldwide civil system will diminish.

The limitations of GPS requires at least four beacon signals to be overhead, which makes indoor localization nearly impossible. Urban density and skyscrapers also cause difficulties in receiving four messages and the issue of multi-path signals occurs within the vicinity of high rise buildings, Further, for a device, it can take multiple minutes to acquire an accurate coordinate. When it comes to power consumption, GPS is a drain on battery and is not feasible for low powered IoT devices.

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

In sum, the issues with depending on GPS for verified location are:

- A single point of failure
- Does not penetrate well indoors or underground
- Urban density increases signal multipath
- Energy intensive components are not suitable for devices with long maintenance cycles
- Spoofing, i.e deceive a GPS receiver by broadcasting incorrect GPS signals

2.3 The Necessity of Clock Synchronization

The importance of accurate time and synchronized clocks is paramount in a decentralized and autonomous location verification system. The ability for a network of distributed beacons to self-synchronize and thus offer accurate signaling for localization is essential when eliminating the reliance on external and centralized sources of location data, such as GPS.

Nodes will inherently experience drift in their local oscillators, frequencies vary unpredictably due to physical effects i.e. temperature, aging. This is known as *Clock Drift*. The lower the quality of the clock the higher the drift rate will be. Clock drift results in a gradual degradation of synchronicity and as a result the clock must be periodically resynchronized.

Time synchronization in Wireless Sensor Networks (WSNs) applications is crucial to many applications where state-of-the-art techniques such as GPS or network based timing protocols are not suitable. This has in recent years become an active area of research due to the increase in WSNs and the anticipated deployment of autonomous vehicles. A taxonomy of the state-of-the-art implemented time synchronization protocols surveys the current discourse.

DRAFT

3 Low Power Wide Area Networks

There are a number of radio technologies that can for localization and positioning systems without the use of GPS or GNSS. Non-GNSS positioning is a wide class of positioning systems without GNSS that encompasses everything from cellular signals to Ultra Wide band (UWB), WLAN, BLE, or Radio Frequency Identification (RFID) signals.These alternative position systems use a range of localization processes and techniques, which include Time of Arrival (ToA), Time Difference of Arrival (TDoA), Angle of Arrival (AoA) and Received Signal Strength (RSS).

Secure distributed systems will want to avoid GNSS in part because of intentional radio frequency interference, which includes mainly jamming and spoofing: "Jamming is the transmission of signals in the GNSS frequency bands with the intent of disrupting the system operation, whereas spoofing is the transmission of counterfeit GNSS-like signals with the intent of fooling the receiver to use false information for positioning calculations." Further, "GNSS positioning is degraded in urban areas and forests, where buildings and foliage, respectively, obstruct the signal propagation and cause multipath. Furthermore, conventional GNSS positioning is unavailable indoors and inside tunnels." [stated by the IEEE in Robustness, Security and Privacy in Location-Based Services for Future IoT: A Survey]

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

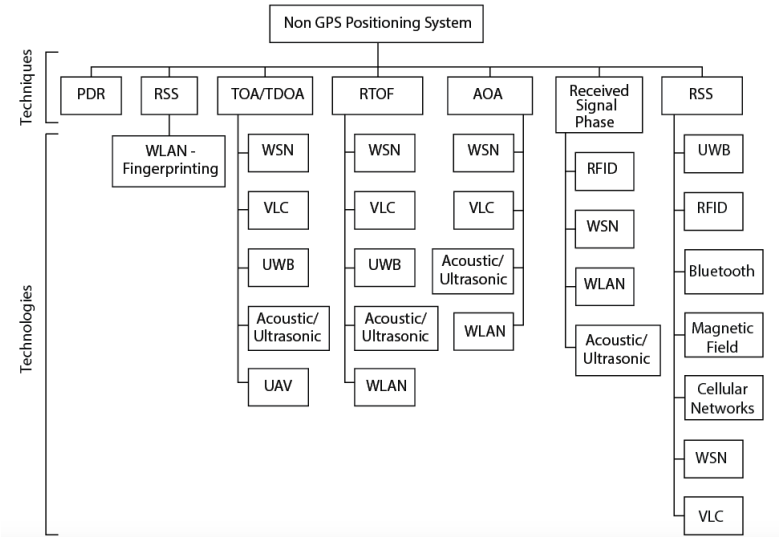


Figure 3: Non GNSS Positioning Systems

Among WiFi, RFiD and cellular radio, a new class of radio emerging and highly promising for internet of things devices called Low Power Wide Area Networks (LPWAN). LPWAN can offer the low power and longer battery life of bluetooth with the range of cellular. The trade-off is low throughput for high-capacity networks suitable for scaling. Another benefit of low power transmission is the access it allows to the unlicensed radio spectrum. LPWAN radios can operate on free radio waves without needing a license to offer coverage. Deploying a LPWAN, just like a blockchain, is permissionless.

LPWAN technologies offer unique sets of features including wide-area connectivity for low power and low data rate devices, not provided by legacy wireless technologies. LPWAN networks are unique because they make different tradeoffs than the traditional technologies prevalent in IoT landscape such as short-range wireless networks. IoT and M2M devices connected by LPWA technologies can be turned on anywhere and anytime to sense and interact with their environment instantly. It is worth clarifying that LPWA technologies achieve long range and low power operation at the expense of low data rate (typically in orders of tens of kilobits per seconds) and higher latency (typically in orders of seconds or minutes).

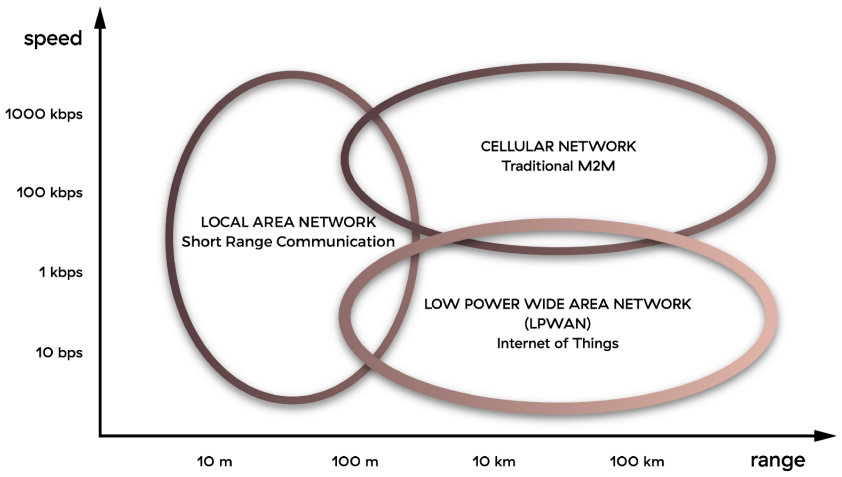


Figure 4: LPWAN Opportunities

One of the most promising new radios is a called LoRa, a physical layer technology that can travel 5–15km at 150 MHz and 1 GHz bands, which can provide bidirectional communication with a special chirp spread spectrum (CSS) techniques for long range with properties that make it harder to detect or jam. There is already the enterprise consortium called the LoRa Alliance, designing an open standard

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

and defining architecture and layers above the LoRa physical layer. Further there are open development communities in major cities around LoRa open libraries centered around the Things Network. Because these radios allow for bidirectional communication, mesh network topology significantly extends range.

LoRa is a physical layer technology that modulates the signals in SUB-GHZ ISM band using a proprietary spread spectrum technique developed and commercialized by Semtech Corporation. A bidirectional communication is provided by a special chirp spread spectrum (CSS) technique, which spreads a narrow band input signal over a wider channel bandwidth, signals vary frequencies over time with control over the tradeoffs between range and data rate.

4 Fault Tolerant Clock Synchronization

4.1 Time Synchronization

As localization is usually computed from an over-determined system of known locations and distances, we may focus now on how to come derive these values in a wireless sensory network. Here we look to distributed algorithms for clock synchronization, i.e. the process of synchronizing all the local clocks of nodes in a Zone. Even without fault tolerance concerns, this is a difficult problem which we by no means attempt to solve from scratch. Rather, we primarily look to the work of Malekpour and his paper *A Self-Stabilizing Hybrid Fault-Tolerant Synchronization Protocol* [5] as proof that it can be done.

Malekpour's work focuses on a Byzantine Fault Tolerant solution to clock synchronization of WSNs– i.e. a distributed algorithm for synchronizing clocks across a WSN even in the presence of a one third minority of Byzantine nodes. The algorithm is described in two phases, a coarse sync followed by a fine sync, at the end of which all fault-free nodes in the network will have local clocks synchronized to within the theoretical limit of one clock tick. This method allows the network of beacon's to determine its own geometry without external sources and then uses trilateration to accurately determine the current location.

As a BFT clock synchronization algorithm is key to any localization method employed in the FOAM protocol, we highlight the formal verification methods used in Malekpour's work. The algorithm presented in [5] is encoded as a *transition system* in a temporal logic known as Computation Tree Logic (CTL). This means that the algorithm is reified in the logic as a set of initial states, together with CTL propositions which govern what properties hold true at any state-transition graph rooted in that state. It is then possible to state propositions in CTL which are decidably proved or disproved by this transition system in one of many CTL solvers, properties such as

1. **System Liveness**: The system will stay alive longer than any bound on convergence – i.e. progress is always made.
2. **Convergence/Closure**: After convergence has occured, synchronization is maintained in all future states.
3. **Congruence**: It is possible for a node to tell based on local information only whether or not synchronization has been obtained system wide.

While the convergence and closure properties provide a system wide view to an external party, the congruence property provides an embodied view to any single node to locally compute if the total system has converged. Congruence is necessary for the interaction of self-stabilizing properties and higher order protocols and is what activates and deactivates the different synchronicity algorithms.

It is important to note that these clocks are synchronized in a zone-local manner and do not necessarily correspond to UTC time.

DRAFT

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

4.2 Fault Modes

We now outline in more detail how a Zone can be formed when nodes and corps are following the Malekpour protocol.

For a set V with K nodes, ie $|V| = K$ we form the complete graph $G(E, V)$ where $|E| = K * (K - 1)$. A complete graph means that there is a edge (or link) between every possible pairing of nodes. We assume that there are at most F faulty nodes and $F < \frac{K}{3}$ (see definition below for fault modes). Although it is likely possible to work with fewer assumptions on the connectivity of the graph, we use these assumptions in order to be able to apply the time-synchronization algorithm that we've choosen.

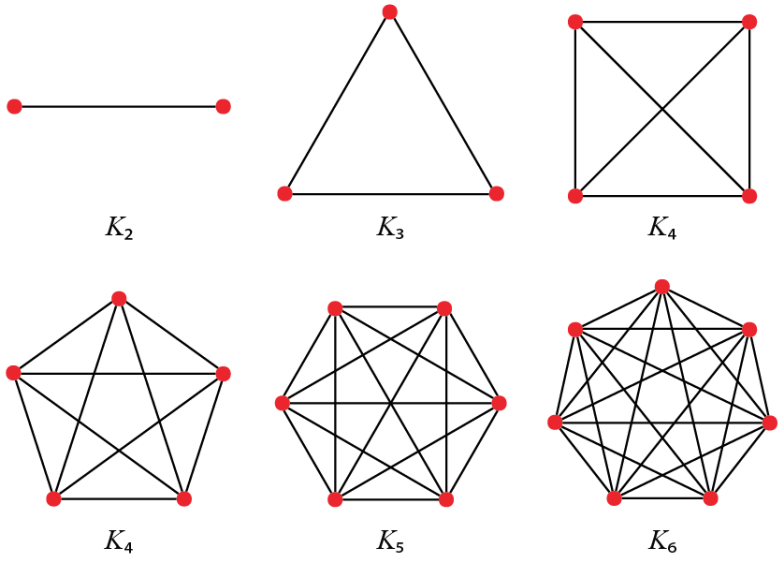


Figure 5: Complete Network Graphs

A good node is assumed to be an active participant and one that executes the algorithm correctly. A node can be faulty in the following ways:

- It can be **benign** and hence detectably bad-acting, meaning that it does not follow the protocol outlined below.
- It can be **symmetric** in which case all neighbouring nodes perceive it the same way, but they can't tell if it is bad-acting or not.
- It can be **bounded-arbitrary** (Byzantine) where it displays different behaviour depending on how it want to be perceived towards different neighbouring nodes.

Moreover, the $K * (K - 1)$ links can themselves be considered faulty. That can either be because the node that transmits the message is faulty, or because a good node is having trouble transmitting its messages.

In all of the following, we assume at most F faulty nodes *and* at most F faulty links.

With the assumptions above, a protocol is defined in [5] that provides a symmetric-fault (or hybrid-fault) tolerant, self-stabilizing synchronization process between the nodes.

5 Distance and location estimation in a zone

In a general wireless-sensor-network, it may be infeasible for all sensor nodes in a to have knowledge of their global coordinates. Therefore, many sensor networks rely on a subset of nodes that know their

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

global positions. These anchor nodes are then used by all other nodes to perform localization. Techniques that rely on such anchors are called **anchor-based** localization (as opposed to anchor-free localization). A large number of localization techniques (including many anchor-based approaches) are based on range measurements, that is, estimations of distances between several sensor nodes.

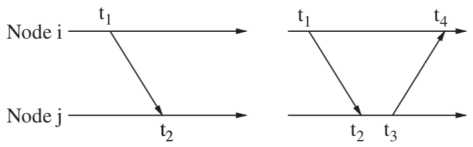


Figure 6: One-way distance estimation (left) and two-way distance estimation

Note that with one-way localization, the receiver node calculates its location, whereas in the two-way approach, the sender node calculates the receiver's location. Therefore a third message will be necessary in the two-way approach to inform the receiver of its location.

The precision available to a node measuring the distance to another node is determined by the transmission speed of the medium and the internal clock of the node. If we make the assumption that we use radio which transmits with the speed of light c and a clock of $24Mhz$ we get that the precision is $\frac{c}{24Mhz} = 12.49m$.

5.1 Anchor-based, one-way localization of nodes

Let $\mathbf{x}$ be the unknown position of a node. We can solve the linear problem $A\mathbf{x} = b$ where A, b are constructed from the known locations of the anchors n_i and the measured distance between the node and the anchor. Note that the location of each anchor is encoded in their message and that the distance between and an anchor can be estimated with the one-way time-of-arrival. The distance is given by $d(\mathbf{x}, n_i) = (T_{n_i} - T_{\mathbf{x}}) \times c$. As in GPS, where the accuracy or synchronization of the node's clock cannot be assumed to be on par with that of the anchors, we assume that we have access to at least four anchors. Then we can "mod out" the clock-error in the following way. For four anchors, draw spheres around each anchor with a radius of their respective distances. If there is no synchronization error, they will intersect in one point. Because of the error, they do not intersect in one point. However, because we can rely on that the relative time error of the nodes clock versus the anchors' is constant, we simply scale all spheres uniformly, either making them bigger or smaller, until they intersect in one point - moving the node's time into the past- or the future, respectively.

5.2 Single Signal Time Difference of Arrival

Single-signal difference of arrival (single-signal TDoA) is a procedure useful for establishing distances between nodes with synchronized clocks, but also for validating a presence claim of an entity external to a clock-synchronized zone. Consider the following figure, where t is a transmitter, r_t is the position of the transmitting node, r_i is the position of receiver i

DRAFT

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

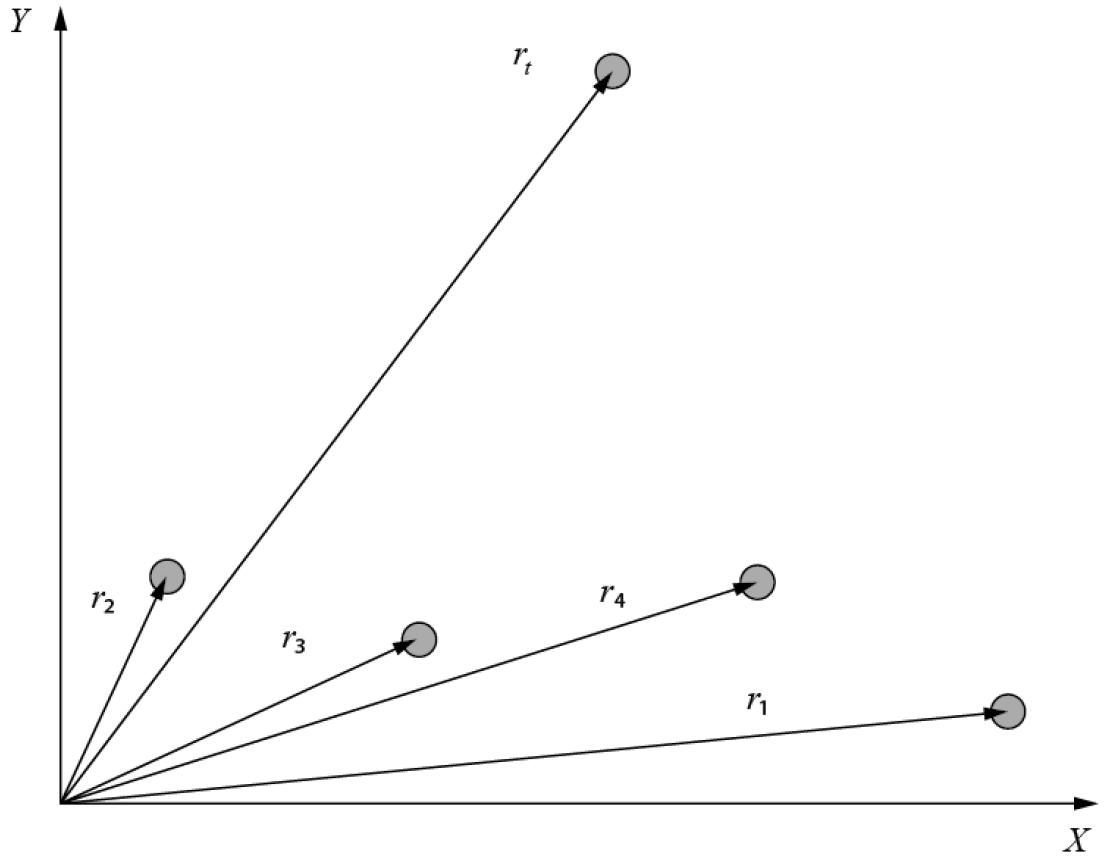


Figure 7: Single Signal Time Difference of Arrival

Assume we are in the first case, where the clocks of t, and all the r_i are synchronized. If t broadcasts a message containing the current timestamp which is received by r_i at time t_i, then we may compute the pairwise time difference of arrivals are computed as

$$\Delta_{i,j} = t_i - t_j = \frac{1}{c} \cdot (\|r_i - r_t\| - \|r_j - r_t\|)$$

This can be used to compute all of the pairwise distances between the nodes, from which it follows that knowning the location of any three of them determines the location of the fourth.

Alternatively, assume that the locations of all the r_i are known, and that t is trying to establish a proof of location from an initial claim. Now however, we may assume that t does not belong to the Zone, i.e. does not share a synchronized clock with the Zone. Then they may simply send out a broadcast ping, whose arrival time is recorded at each receiving node. Assuming that the Zone has agreed to the location of each receiving node in the Zone, the pairwise time differences can be checked against the known distances to establish the veracity of the transmitters claim.

Of course what we are describing is the ideal situation, which does not take into consideration signal noise or interference. However, there are substantial results, both theoretical and computational, which seek to accommodate these in a real world setting. See [2] for a survey.

As a note, this allows a user to use their location derived via GPS to issue a challenge to the Zone that their clocks are synchronized, and by extension that localization is possible. A users sends out a signal to the Zone. If the Zone is able to determine your position within a reasonable limit, you should be able to deduce that their clocks are synchronized. If they can not, then this constitutes as evidence that the Zone is faulty.

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

6 The Decentralized Location Verification Solution with FOAM

LPWAN technology is at the core of the FOAM vision for its ability to scale, cover large distances and remain available due to the low power. A node on the FOAM network will need to offer accurate time synchronization over radio transceivers. This kind of beacon is called a Zone Anchor. Four or more Zone Anchors form a Zone, the quorum that maintains clock sync for a given region. Once synchronized, the Zone can determine the location of a requesting node by using time of arrival measurements to verifiably triangulate position. The FOAM Proof of Location protocol prioritizes decentralization and security, thus the system must be designed to achieve secure localization, location verification, be node-centric and utilize range-based distance measurement techniques.

Neither location nor time attestation can be trusted without additional means of independent verification. Proof of Location is designed to eliminate any reliance on external data authentication. Decentralized application will need to rely on secure transactions within a trustless and autonomous execution environment. Both its centralized authority model and unencrypted uplink protocol preclude GPS from serving as a trusted and privacy-protecting data source for geospatial information. Existing localization techniques may be subject to malicious attacks, such as time spoofing. Failure to prevent fraud in GPS-based clock synchronization has the potential for critical failure and a violation of safety. Therefore, decentralized clock synchronization can be a valuable approach as an alternative GPS-based clock synchronization.

A location verification system based on decentralized clock synchronization must be designed as an open standard that allows qualifying participants to become their own service providers. The LPWAN technology exists today for such a system, what is missing are the economic incentives for users purchase, install and maintain a network of beacons. While this technology is available for today there is no incentive model to deploy at scale. Just as Bitcoin, Ethereum, and many other blockchains "hired" miners to run and operate the network, crypotoeconomic incentives are needed to grow decentralized location verification systems and markets. The FOAM Proof of Location protocol utilizes a Byzantine Fault Tolerant clock synchronization algorithm to provide the best possible support for RF ToF algorithms

In the FOAM Proof of Location protocol users operating Zones do so by entering into a Service Level Agreement (SLA) and backing the agreement with a safety deposit bond into a smart contract. This SLA enables autonomous service providers to open and maintain zones of coverage, promising to do so with a deposited bond. The SLA is governed by the protocol rules and enforces the protocol rules. This open and permissionless SLA system opens the incentives and user driven development of a decentralized, privacy preserving, highly accurate, censorship resistant alternative to GPS.

The FOAM Proof of Location has been designed to be radio agnostic. We see LPWAN as the optimal choice for a location alternative to GNSS. Further, we see LoRa as the most promising radio to use in a secure and decentralized network. See Appendix for full comparison of radios under consideration.

DRAFT

7 Consensus Design

7.1 Byzantine Consensus

The Byzantine General's Problem was first proposed by Lamport, Shostak, and Pease in 1982 as the difficulties of achieving distributed agreement over a compromised network. The breakthrough in this research came in 2008, from Satoshi Nakamoto, who introduced internet-scale distributed Byzantine Fault Tolerant (BFT) consensus with a blockchain scheme. Decentralized blockchain systems, such as Proof of Location, require Byzantine Fault Tolerance, which means it can withstand up to 1/3 of system wide faulty or malicious nodes and still follow the protocol correctly. Protocol Labs defines this design structures as:

Byzantine Consensus protocols are often structured in a sequence of rounds or epochs where participants propose and agree upon values in a sequence of epochs. Participants receive a sequence of state-changing

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

requests from clients, participants must propagate the effect of requests to each other, and all correct participants must come to agreement on the values or responses externalized to the clients.

7.2 Comparison of Blockchain Consensus Algorithms

If we abstract consensus, any protocol has the following `four properties`:

- Validator set
- Validator weight
- Validator criteria
- Validation verifiability

Proof of Work, for example, has an unknown validator set in terms of number of parties but it has known validator weight, i.e. the hashpower. But from one moment to the next you have no idea if the hashpower will double because there are no restrictions placed on that. The validation criteria are simply whether the blockhash founds meets the difficulty, and the validation verifiability requirement is having the chain synced, because without that you no idea of knowing whether the mined block is actually valid in the chain.

Proof of Stake has a known validator set, as a staker must be known by their public key and deposit amount before they can participate in the protocol, and the protocol itself governs validator weight through the maximum total stake metric. The validator criteria is having enough stake to participate and performing the staking process by signing, and being accepted into the validator set. The validation verifiability, like in Proof of Work, is mainly having the chain synced, as otherwise there's no way to know how much a validators have staked or whether they're even in the set of accepted validators.

In Proof-of-Stake blockchain protocols participants commit a stake, bond or deposit of tokens to operate as a validator and obtain voting power. Stakers accrue a larger reward by participating correctly in the consensus protocol. The required security deposit, the bonding of collateralized value, is at the core of PoS incentive systems. If faulty behavior is detected by a node, the bonded tokens are destroyed and forfeited by the protocol.

The Proof of Work security model is objective, the correct chain is the one with the most hash power. For Proof of Stake systems are "weakly subjective" as they require nodes to asses social information to come to a security conclusion. To address this, only nodes with a current and known bond are trusted and there is an unbonding delay period when withdrawing stake. In Proof of Stake protocols offer dynamic validator sets, participants can enter and exit. To account for Sybil attacks, these systems use the 3 E's:

1. Entry Cost
2. Existence Cost
3. Exit Penalty

The set of faults that constitute a violation of protocol rules are known as slashing conditions. However, if the validator operates correctly, it is eligible to receive newly minted tokens and transaction fees. The fault-tolerance assumption is described in terms of a fraction of the total tokens staked. With Proof of Stake systems there is flexibility in the protocol design to explicitly account for penalties of Byzantine behavior and program the the asymmetric risk and reward profiles of various actions.

The market incentives around Proof of Location and the FOAM token its the corresponding design is based on the Proof-of-Stake (PoS) research question of achieving decentralized consensus without depleting physical, scarce resources, for example the wasteful externalities of Proof of Work (PoW). Can decentralized computation achieve consensus and contribute "useful" work?

Proof of Authority is an alternative to Proof of Work where the validator set is trusted and privately agreed on which nodes can produce new blocks and secure the blockchain. Access to becoming a Proof of Authority validator can be operated in a permissionless way, as discussed below. **Proof of Sync** is defined below. **Proof of location** is defined below.

DRAFT

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

The unique problem introduced by Proof of Location is that for a part of the validator set, the physical distance between them plays a fundamental role, while in Proof of Work for example to validate having completed the work all that is needed is a communication channel, the physical distance between contributors is more or less irrelevant. Proof of Location has these same four consensus properties, but because the problem is more heavily constrained by physics, the projection of said properties is different.

For a given Zone the validator set is effectively the collection of Zone Authorities that share a "location horizon" - the maximum spherical range of the physical radio signal. Each Zone Authority has a location horizon, and the overlap in horizons constrains the possible validator set for a Zone. The validator weight is the the amount of tokens the validators have staked, and the validation verifiability comes from being in the horizon overlap for a Zone, as without that you have no way to tell whether the events truly happened at those times and in that Zone, you would merely be trusting a relayer.

Protocol	Validator set	Validator weight	Validator criteria	Validation verifiability	Users	Inputs	Chain output
Proof of work	Unknown number of nodes (miners)	Hashpower of each node. Not constant in time.	`sha(nonce) < difficulty`	Synced to the chain (verifying if the mined block is valid in the chain) ie valid `parentHash`	• ETH/account holders	• `[transaction]`	• Ordered subset of `[transaction]` and `stateRoot` • `parentHash` • `(nonce, miner)`
Proof of stake	Known by public key and corresponding deposit	Governed by protocol through maximum stake.	Having enough stake and correct signature to be accepted into validator set.	Synced to the chain - to verify `parentHash` as well as to verify that stake is valid.	• ETH/account holders	• `[transaction]` • `[(signature, stake)]`	• Ordered subset of `[transaction]` and `stateRoot` • `parentHash` • `[(signature, newBalance)]`
Proof of authority	Known number of nodes	The chain has to be signed off by the majority of nodes.	Signature of the block signer matches and hence accepted into validator set.	Synced to the chain.	• ETH/account holders with relationship to at least one authority	• `[transaction]`	• Ordered subset of `[transaction]` and `stateRoot` • `parentHash` • `[(signature, vote)]`
Proof of sync	Disjoint sets of (multiple) node(s) in a Zone.	The chain has to be signed off by the majority of nodes.	Signature matches validator set. In addition, the logs show that the nodes have performed the Malekpour protocol.	Signed receipts of all messages received ie. synched to the chain.	• Nodes	• `[(signature, timestamp)]`	• Ordered stream of `(signature, timestamp, validRelativeToSigne` • `r)` of which there is consensus on the synchronicity
Proof of location	Unknown number of participants (ie validators ala truebit)	Could be number of staked tokens or reputation.	Challenge with signatures is solved within horizon of speed of light, deterministically. Ultimate verdict by smart contract (costly).	Receipt that challenge has been inserted into logs (PoL contract)			

Figure 8: Comparison of consensus algorithms

The FOAM protocol encourages user implementation of synchronized clocks for Proof of Location as a decentralized, privacy preserving, and interactive alternative to GPS and means of measuring space and time for cartography and map making. The entirety of the FOAM protocol relies on Synchronous, Partially Synchronous and Asynchronous consensus. While the clock synchronization over radio and localization for nodes is determined with Synchronous consensus, the consensus algorithm for the replication of shared state machine for a single zone is partially synchronous. The time log data produced by a Zone is not considered final until it has been verified for fraud by a computation engine running trilateration and other triangulation location algorithms on the Zone's time data.

In this system, the security and consensus of the root chain is an asynchronous network. This can be the Nakamoto consensus of a Proof of Work chain like Ethereum or an established Proof of Stake blockchain like Ethereum's Casper. As such, FOAM employs Synchronous, Partially Synchronous, and Asynchronous consensus for Proof of Location.

8 Formation of a Single Zone

The following is a high level overview to serve as a theoretical foundation for the protocol, the assumptions presented are justified in the crypto-economics of the system. The practical description of Zone formation as it pertains to blockchain architecture is described in subsequent sections.

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

8.1 Synchronized Zones

Consider a set of nodes whose absolute positions are fixed and known. How can an observer tell whether or not they share synchronized clocks?

There are two distinct ways for an observer to determine if they are in sync:

The first one is simply for the observer to synchronize its own clock with that of the other nodes.

The second doesn't require the ability of clock-synchronization; suppose that the observer knew its absolute position, and it was in communication will all the nodes. It could ask them to each send out a series of timestamped messages, and use those timestamped messages plus the knowledge of their location to localize itself, á la GPS. The observer can then compare this calculated position to its known position, and if the two are reasonably close over multiple observations, it can conclude that these nodes' clocks are synchronized.

We would like the take the question of whether or not a set of nodes are in sync and make the answer subject to distributed consensus, similar to the question of determining the bitcoin holdings of a particular address at a particular time. But consensus by whom?

8.2 Example Scenario

Suppose for the moment that there are three Zone Authorities each with a set of Zone Anchors that require a gateway connection. For the sake of this example each corresponding set of Zone Authorities and Zone Anchors is called a Corp. Thus we can say there is, Green Corp, Red Corp, and Blue Corp. Suppose that each Corp authority is capable of receiving radio transmissions from all other nodes present, i.e. the Green Corp authority can receive messages from all Zone Anchors in Red Corp and Blue Corp, and so on. Furthermore, suppose each Corp's Authority is capable of bidirectional communication with every other Corp's Authority.

DRAFT

Assumptions

1. All nodes' and (and hence Authorities') relative distances (or absolute locations) are known in advance.
2. A Corp. always desires to be seen in the **synched state**.
3. Every Corp. has in its own interest that *every other Corp is in sync*.
4. The Corp.'s that make up a Zone share a State Machine

On Assumption 3 : It is assumed that every corp desires that every other Corp is actually in sync, not merely the fact the the state machine indicates the **SYNC** state for that corp. This full justification for this incentive is described in section 9. The ultimate purpose of being in sync is to provide location services, the accuracy of which is a function of the number of synced Corps involved.

Beyond this example, Zone Anchors and Zone Authorities are capable of joining together dynamically and fluidly and or may be deployed in altered configurations.

Here we discuss one aspect of the shared state machine within a Zone which models the relation between two propositions, *The Corp is in sync* and *The Corp is in an unknown state*. Below is a drawing of this simple state machine in the case of Blue Corp:

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

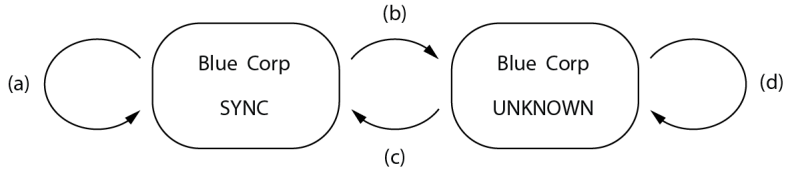


Figure 9: The four possible states of a corp

As in the diagram, let's label the states **SYNC** and **UNKNOWN**. Suppose that the state is currently **SYNC** for Blue Corp, and that the time as come for the state machine to run. We presume that a *judge* is responsible for any direct changes to this state machine, for example a smart contract. We outline the following procedure.

1. Blue Corp instructs their nodes that they are to issue a broadcast signal at time t, capable of being received by all other authorities.
2. When each local clock in BlueCorp arrives at time t, the nodes broadcast the message.
3. The Red Corp Authority receives the signal from node N_i at time t_i. They use this information to calculate their position x_{calc} and compare it to their known position x_{known}.
4. The Green Corp Authority receives the signal from node N_i at time t_i. They use this information to calculate their position x_{calc} and compare it to their known position x_{known}.
5. At this point some Corp, say Red Corp initiates a "transaction" to the state machine. This transaction consists of the pairs (N_i, t_i), as well as a vote for whether or not they agree to the synchronicity of the clocks based on the geographic data (to be dictated by a smart contract).
6. the other Corps would vote to agree or disagree, each presenting their received times to the judge as a basis for their decision.
7. the state of Blue Corp would transition along arrows (A) or (B) depending on the outcome.

A similar round exists for when Blue Corp is in the **UNKNOWN** state and desires to transition to the **SYNC** state. See [sec x] for further information about the shared State Machine of a Zone and how the voting round is structured.

DRAFT

8.3 Proof of Synchronicity

We formalize what's outlined above:

Definition 1 (Node)**.** *A **Zone Anchor** is a device with a radio transmitter, a local clock, and a public key. A node is capable of engaging in a clock-synchronization protocol.*
Definition 2 (Zone Authority)**.** *A **Zone Authority** is a distinguished node with internet access and "sufficient" computational power to maintain a shared State Machine. It has the ability to determine if a corp is in sync (see below).*
Definition 3 (Corp)**.** *A **Corp** is a set consisting of an Zone Authority together with at least two other Zone Anchors. These nodes are presumed be able to listen to messages from the Authority and rely on it for gateway access.*
Definition 4 (Zone)**.** *A **Zone** is a set of Corps that maintain a state-machine describing the mutual observed synchronized state between the Corps' and the Authorities.*

Example 8.1 (Authority relays)**.** We have three sets $\{a\}_0^3, \{b\}_0^3, \{c\}_0^3 = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ of four distinct Zone Anchors and Zone Authorities $\{a_0, b_0, c_0\}$ in each, thus forming three corps. The Zone consists of these three Authorities committing their readings of all the Corps.

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

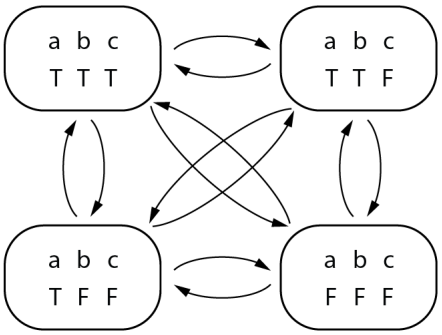


Figure 10: The possible states of a 3-corp zone (modulo permutations)

The state machine is represented in Figure 10

For a visitor to the Zone, they will be able to read the logs of three Zone Authorities and receive up to three sets of pings with up to four nodes participating in each distinct ping.

Set	Constituents
Nodes	$\{a\}_0^3 \cup \{b\}_0^3 \cup \{c\}_0^3$
Corps	$\{\{a\}_0^3, \{b\}_0^3, \{c\}_0^3\}$
Authorities	$\{a_0, b_0, c_0\}$
Zone	$\{\ldots, \{(v_{\mathbf{b}}^{a_0}, v_{\mathbf{c}}^{a_0}), (v_{\mathbf{a}}^{b_0}, v_{\mathbf{c}}^{b_0}), (v_{\mathbf{a}}^{c_0}, v_{\mathbf{b}}^{c_0})\}, \ldots\}$

Example 8.2 (Malekpour)**.** We have three nodes in total $\{a, b, c\}$, and each node is also an authority. Because the Malekpour protocol enables each node to determine if the rest of the nodes are in sync by joining it (congruence), every authority will be commiting their verdict of wether the other two nodes are in sync with it. Hence, the state machine looks like in the previous example in Figure 10.

For a visitor to the zone, they will be able to read the logs of three authorities and receive one set of pings with up to three nodes participating in each distinct ping.

Set	Constituents
Nodes	$\{a, b, c\}$
Corps	$\{\{a, b, c\}, \{b, a, c\}, \{c, a, b\}\}$
Authorities	$\{a, b, c\}$
Zone	$\{\ldots, \{(v_b^a, v_c^a), (v_a^b, v_c^b), (v_a^c, v_b^c)\}, \ldots\}$

Note that the above two examples are two different realizations of the same roles. As long as there are Authorities in Zones that can communicate with each other, two different realizations of a Zone remain compatible through transitivity.

We will talk further about relationships between Zones in a later section.

8.4 Messages and logs

Basic message components are defined here:

```
data Signature                        -- an ECDSA signature
data Incentive  -- a way of describing an incentive for a ZA to log/sign a presence claim
newtype Nonce     = Nonce UInt64      -- monotonically incrementing
newtype Timestamp = Timestamp UInt64 -- zone-local, ns resolution
data Location     = Location { latitude  :: Float32
                             , longitude :: Float32
```

DRAFT

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

```
}
```

An active participant - a Zone Anchor, broadcasts messages to other Zone Anchors to perform rounds of the protocol. Each message contains a cryptographic signature (nominally one which supports recovery of the public key from the signature for compactness' sake, such as ECDSA with the ubiquitous secp256k1 curve) to identify the node who broadcast it. It also contains a monotonically increasing numeric value (the nonce) which is used to identify the order in which a node broadcast the message relative its other messages. Lastly, each message contains a protocol term, which is the actual information the node is trying to convey. When we refer to an X message, we really mean "a Message whose ProtocolTerm is of type X" (or, a Message X in Haskell).

Thus when a Zone has been formed, it can be assumed to send out messages of the form:

```
data ProtocolTerm where
  SyncCoarse      ::                              ProtocolTerm -- Aka Init
  SyncFine        ::                              ProtocolTerm -- Aka Echo
  Ping            :: Location  -> Timestamp -> ProtocolTerm -- aka IAmHappy or also Echo? in Malekpo
  PresenceClaim   :: Incentive -> Timestamp -> ProtocolTerm -- sent by people wanting to prove their

data Message term :: ProtocolTerm -> *  = Message Signature Nonce t
```

A Zone's boundary is generally defined as the intersection of each Zone Anchor's set of points where one may intelligibly receive that Zone Anchor's radio transmissions.

9 FOAM Token Mechanism

In Proof of Location, the goal is to have consensus on whether an event or agent is verifiably at a certain point in time and space. The FOAM token incentivizes users to contribute computation and radio power towards secure location services in a way that aligns with financing alternative means of obtaining secure localization and location verification standards that are suitable for autonomous smart contracts.

As detailed in earlier in the paper, Proof of Location requires Proof of Time Synchronization. We introduced `Zone Anchors` `Zone Authorities` and a `Zone`. In this section we also introduce the definition of `Verifiers`, `Root chain`, Beacon Public Keys and Presence Claims.

A `Zone` is a set of Authorities with a network configuration as above, whose Authorities agree to participate in a shared state machine. "Zones" run concurrently and each Zone shares a state machine and is financially incentivized to remain in sync.

`root chain` is the blockchain where FOAM token bonds, deposits, rewards and penalties take place. Participants must interact with this chain to be given access to participate as part of the validator set of a Zone. For now, imagine that the root chain is the public Ethereum blockchain. Similar to other blockchain mining, Zone operators on the FOAM protocol are in essence providing comparable work to Bitcoin miners.

Beacon Public Key (BPK) Is the known public key of a mobile beacon that wants to purchase presence claims from Zones.

A presence claim (PC) is a set of counter-signed Requests with the same Nonce, which is intended to provide enough data to constitute an exact localization. Issued by Zone participants to BPK for a fee. The presence claim is subject to fraud proof computation before being authenticated as a proof.

`Verifiers` are computational engines that incentivized to check the time logs of Zones for fraud and finalize Proofs of Location. Because Verifiers compute locations from the time stamped data they can be said to be mining triangulations.

DRAFT

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

The purpose of this section is to

1. Introduce the economic incentives and market structure by defining the FOAM protocol token mechanism as it relates to the shared state machine of a `Zone` and the global FOAM protocol and ecosystem.
2. Further, we introduce our thinking and research about possible scaling solutions which is based on the active research of Plasma implementations and Tendermint consensus.

9.1 Token Incentive Structure

In Proof of Location, owning the **native software utility token**

1. Purchases the right to offer location services.
2. In addition, owning the token purchases the right to **verify** the network based on the consensus rules.

Location customers, i.e. users of applications and purchasers of Presence Claims do not need to use the FOAM token to do so. Zone's can accept payment in any token. The FOAM token is supply side driven and needed only when providing work to the network.

In both of the above scenarios, the token is deposited into a bond on the root chain, attesting to provide correct activity to the network. Improper activity will result in a slash of locked funds. Enforcement of fraud in the market activity by the protocol rules aligns incentives in that the most profitable outcome for all participants is honest behavior.

Thus the optimal utility of the token is through staking, meaning an owner will gain the biggest rewards/interest through participating in the the staking needed for either 1. and/or 2.

The role of the token is to incentivize providers of computation and ultimately offer location services. The token, a scarce resource, can be thought of as a piece of virtualized hardware needed to operate a validator. Each validator of each zone has a deposit; when a validator joins, its deposit is the number of deposited coins. After joining, each validator's deposit rises and falls with rewards and penalties.

The FOAM token is not rent seeking, meaning there is no service fee that goes to a central entity when using the token.

Users must run node and post a participation deposit to obtain transaction fees, which are therefore not passive income. Thus there is an inherent economic interest for the networks continued operation and avoidance of failure.

Given the cooperative incentives created by utility providers' safety deposits, the FOAM Protocol rule set is designed to be enforced by the network, punishing adverse behavior according to Slashing Conditions, the protocol defined punishments. In return, stakers are compensated for their work in the form of Block Rewards, the minting and distribution of a new FOAM Tokens and transaction fees.

Additional Features

- **Permissionless**: The Validator set of a `Zone` is open and permissionless, anyone who owns the token can bond and become a validator or join a `Zone`.
- **Network Bandwidth Based**: The token acts as a piece of virtualized hardware where the amount staked corresponds to bandwidth reservation of the network. The amount of tokens staked unlocks the ability to contribute more work and security to the network.
- **Provisions**: The FOAM protocol rewards and staking with the minting of new tokens and inflating the total supply. The minting of new tokens happens at the global level in relation to all Zones that have bonds on the `root chain`. Inflation, or protocol provisions, incentives staking to the highest degree because it is also a form of punishment to holders of tokens that are not being staked. New tokens distributed to the accounts of bonded tokens dilute the outstanding supply. The inflation

DRAFT

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

rate is set in relation to the total supply, which is continuously increasing. [further discussed in sec 9.3]

9.2 The Formation of Child-Chains

Each individual Zone will share a state machine local to their "location horizon" of physical coverage while also participate in the rule set of a parent and global public blockchain where a stake is made. This construction is known as Plasma. In order to join the validator set of a Zone, a bond must be placed on the `root chain` In this instance, defined as the public Ethereum blockchain. The `root chain` consensus mechanism is to record fraud proof computational results and enforce the protocol rules of Proof of Location. Out of the `root chain` the state machine of a Zone is maintained in the form of a side chain, child chain, peg etc. of the `root chain` where the root is used for on chain resolution to unavailability in the peg. It is possible for users to exit and withdraw their holdings on the child chain without consent in the event of faulty or malicious behavior. Because a bond must be placed on the `root chain`, child chains can be understood to operate as Proof of Authority side chains.

Each child chain, or Zone, is represented by a Plasma contract on the root chain, which define the protocol rules of Proof of Location and enforce faulty behavior through accountability and slashing conditions. Evidence of fraud occurring within a Zone presented to the root chain results in significant financial penalties. As stated in the Plasma whitepaper: ""to incentivize avoidance of Byzantine states, especially around correctness and liveness, it may be ideal to create a token per contract. This token represents the network effects in operating the contract, and creates an incentive to maximize security of this contract."

The first step is a staking or Plasma contract on Ethereum. Users deposit tokens to that contract for the benefit of becoming a validator in the underlying child chain. In this context the validators of the child chain are the Zone Anchors, which share a state machine i.e. blockchain [For the avoidance of doubt, 'validators' in this context are not a reference to 'Validators' as referred to elsewhere in this paper]. Zones that ensure their network continues to operate properly will have be able to receive receive transaction fees foe their location verification services. The plasma contract is able to track the state of the child chains and vice versa with bonds for local Zones defined by the fraud proofs in the global root chain, with "the value of the token derived from the net present discounted value of all future returns from staking."

DRAFT

9.3 Block rewards and transaction fees

The block reward occurs at block checkpoints on the `root chain` representing finality. Block rewards are a baseline incentive, as staking tokens also allows access to fees, which also are distributed on the `root chain` where Proof of Location contracts live. Blockrewards are simply all the transaction fees, ie presence-claim fees with additional tokens minted as provisions to stakers. **Provisions**:

- The block reward serves to bootstrap network growth and maximize Zone coverage. The block reward is higher on the edge of Zones than near dense nodes on the graph, encouraging additional nodes to join on the edge for the highest reward. Annual inflation is set at a decreasing and inverse rate in relation to the global coverage of Zones. The specifics of these rates will be discussed in a future paper.

The transaction fees that Zone Authorities receive are in exchange for service of issuing Presence Claims to the public key of a beacon and including this message in their time logs. A location customer will have a Beacon with a Public Key, the public address of a customers beacon that wants to have its message signed by a zone and included in the state machine produced by the Zone. The Presence claim issued and logged by the Zone will be retrievable by a verifier set checking for fraud proofs, which is required as the final step in the Proof of Location protocol.

BPK Signals are sent over radio and there is a hard physical limit to how many messages a Zone can receive, sign and process in any given round. Thus, there will be a market for location customer signature

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

fees, Presence Claims come at a cost. In a highly populated Zone, incoming messages will be prioritized by by the fee a customer is willing to pay. The location customer will have to pay the market rate and a Zone will determine what to accept and what not to accept. These payments serve as transaction fees and Zone specific payment channels can and will open up for regular customers that consume a high amount of Presence Claims. These fees are decided by market participants and can potentially be paid in any token. This fee will go to the Zone Authorities and Verifiers.

A Zone can breakdown and cease functioning properly if many malicious beacons broadcast high fees at once and the Zone only processes fraudulent requests and can never be paid for them. One requirement in this system is that the BPK makes a pre-commit of funds on the root chain that the Zone Authorities can become aware of. Otherwise, BPK can broadcast any arbitrarily high fee and effectively DDOS the network.

Verifiers in the FOAM Proof of Location protocol are any computational entity that is able to read the blockchain data produced by Zones and check Presence Claims for fraud proof. Similarly to the construction of a Zone, a verifier must make a deposit on the root chain to participate in the protocol. What a verifier is looking for is if the "SLA's" of Zones are being fulfilled, further they are conducting fraud proofs to see if the data is accurate, if the clocks were in fact in sync and if their corresponding location claims can be proven from the published data. Verifier is entitled to a percent of the fee a BPK paid for its Presence Claim.

10 State Machine Consensus and Service Level Agreements

In the FOAM Proof of Location Plasma implementation will be specified to accommodate Service Level Agreements (SLA) that are backed by a bonded deposit, the parameter for entrance, exit and participation in the validator set of child chains. The Zone must have tokens bonded to the SLA and agree to the associated slashing conditions if the evidence fraud or breaking the SLA is produced. There exists the possibility for users to exit and withdraw their holdings to the root chain from the child chain without consent in the event of faulty or malicious behavior. The SLA is a smart contract that specifies for how long the Zone will be open and available with its services. In Proof of Location the validator set needs to not only maintain a shared state machine but additionally offer radio coverage and location services. The SLA enables Authorities to become their own service providers for their area of coverage.

The the SLA specifies for how long uninterrupted coverage will be offered, and because this coverage requires the production of time logs, the amount of data any given SLA is supposed to generate by the end of the agreement can be predetermined. Because of this, it is possible to see if at a given point within the agreement the amount of time logs already produced match what the agreement specified. More on this verification process and fraud proofs in the section below.

In the sections above, we describe different modes of consensus formation. For simplicity, we describe the state-machine consensus in a Zone using Proof of Authority. Essentially, the relationship to the the root chain consensus mechanism is to record fraud proof computational results and enforce the protocol rules of Proof of Location. Out of the root chain the state machine of a Zone is maintained in the form of a side chain, child chain, peg etc. of the root chain where the root is used for on chain resolution to unavailability in the peg.

The emergence of a well formed Zone is an observable event on the blockchain. A well formed Zone is defined as a quorum and shared state machine between at least three Zone Authorities.

10.1 Tendermint

For the purpose of Proof of Location we have chosen to pursue Tendermint Core as the consensus mechanism for child chains, i.e. the shared state machine of Authorities in a Zone. This is an emergent and active area of research based off of https://github.com/cosmos/plasma

DRAFT

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

Tendermint Core is a Byzantine Fault Tolerant consensus algorithm developed in 2014 based on a Proof of Stake design. The main purpose and benefits to Tendermint are speed, consistency, safety and instant finality. These features empower the scale of public Proof of Stake blockchains:

> At the core, Tendermint works as a round-based voting mechanism which makes the consensus protocol. A round is broken up into a three-step process through which validators propose blocks, signal commitment intent and then sign to commit new blocks. This mechanism yields a secure state replication machine for atomic broadcast with an added layer of accountability. Safety faults are perfectly attributable in Tendermint.

When addressing the question of how distributed consensus on synchronicity is maintained by the Authorities of a Zone, the incentives of sharing and updating a state machine, as well as the structure of voting rounds are addressed by Tendermint core. The ultimate purpose of Authorities being in sync is to provide location services, the accuracy of which is a function of the number of synced entities involved.

11 Proof-of-location

As previously defined a presence claim is a set of counter-signed Requests with the same Nonce, which is intended to provide enough data to constitute an exact localization.Note, there are several minimal requirements for a presence claim to establish an actual localization– for example there must be at least four, the signatures must be valid, etc.

As a BPK will let a Zone Anchor know that it wants to produce a presence-claim, it sends out a message with its perceived location, a payment receipt and a nonce.

The Zone Anchors will receive the message and note the local time in which it received it. They may relay this distance-estimation to the rest of the Zone. Using single-source TDOA, it is possible for anyone with access to the Zone's logs to estimate the location of the beacon. Each Zone Anchor then transmits a message with its signed estimation of the position to the logs. It also sends the message back to the beacon.

Definition 5 (Proof-of-location)**.** *A* ***proof-of-location*** *is the following data, contained in a smart contract:*

1. *A geohash with precision score from its validator(s).*
2. *A reference to the Zone Authority that issued the presence-claim*
3. *A reference to a valid presence claim*
4. *A computational proof of the correctness of the presence-claim*

The diagram below illustrates how this hypothesis gets verified and put onto the blockchain with finality, acting as a proof-of-location.

DRAFT

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

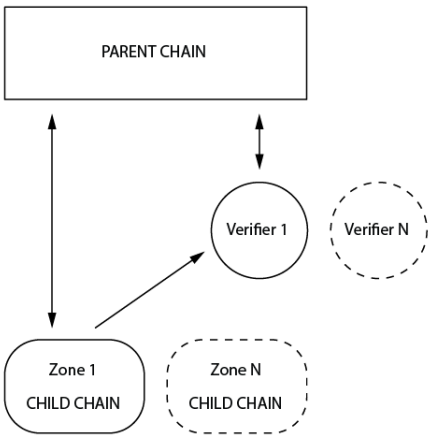


Figure 11: Relationship between Child Chains containing Zone Data and Unverified Presence Claims, Verifiers checking for Fraud Proof and the Parent Chain where Tokens are Staked

The Zone State Machine at the bottom is a child chain to the root chain and contains the data logs of time stamps as well as a source of unverified Presence Claims. At this stage, the Presence Claims constitutes a hypothesis to an entity's location in a Zone. This data is meant to be referenced by any computational engine capable of verifying that the presence claims are authentic (i.e. the signatures are valid and messages are well formed), as well as the fact that they constitute a valid trilateration for the claim's location.

The Verifier circles represents the set of computational engines that have staked on the root chain and are capable of checking the Zone data for fraud. The Verifier posts results to the root chain with a counter-factual verification contract. The purpose of this contract is to establish the validity of the presence-claims and to provide the proof-of-location contract with data and the verification credentials that match it while migrating the computation off of the blockchain. In more detail:

"Counterfactual verification is a technique used to verify a (possibly quite extensive) computation via a blockchain without needing the nodes of the blockchain to actually perform the computation themselves. Instead, the contract performing the counterfactual verification creates a set of incentives which would clearly result in certain responses if those responses existed."

Here we image a few possible implementations beyond counterfactuals by either implementing TrueBit or further customized fraud proofs. The point is that through a staking and fraud proof model, the financial threat that the computation is actually run and seen to be false should disincentives any fraudulence. Remember, the whole world is watching!

Finally, the proof-of-location contract acts a first class object for other blockchain applications. It comes into existence as a result of the verification stage, and represents a successful termination of the protocol.

DRAFT

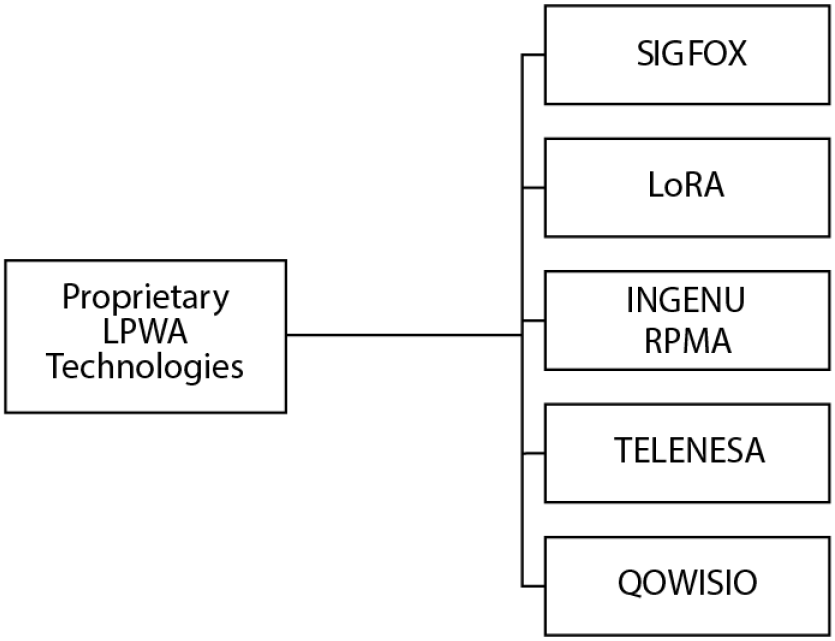
This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

12 Appendix

12.1 Hardware requirements

The BFT time-sync protocol is in general, hardware agnostic. Nonetheless, we assume that nodes have a dedicated radio transceiver with sufficient power and dynamic range to transmit and receive messages such that they can be received by all the other nodes they wish to form a Zone with. Furthermore, to allow a resolution of approximately 30cm when localizing, we prefer that the nodes have a clock source with a frequency of at least 1 GHz, as the speed of light (at which radio signals propgate) is approximately $\frac{29.98cm}{ns}$. Lastly, we assume that a node has some means of making its logs publicly available for validators which would presume a connection to the Internet is available (or at least some means of connectivity with a machine which does).

12.2 LPWAN Radio Comparison



This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

TABLE I
TECHNICAL SPECIFICATIONS OF VARIOUS LPWA TECHNOLOGIES (?=NOT KNOWN)

	SIGFOX	LORAWAN	INGENU	TELENSA
Modulation	UNB DBPSK(UL), GFSK(DL)	CSS	RPMA-DSSS(UL), CDMA(DL)	UNB 2-FSK
Band	SUB-GHZ ISM:EU (868MHz), US(902MHz)	SUB-GHZ ISM:EU (433MHz 868MHz), US (915MHz), Asia (430MHz)	ISM 2.4GHz	SUB-GHZ bands including ISM:EU (868MHz), US (915MHz), Asia (430MHz)
Data rate	100 bps(UL), 600 bps(DL)	0.3-37.5 kbps (LoRa), 50 kbps (FSK)	78kbps (UL), 19.5 kbps(DL) [39]	62.5 bps(UL), 500 bps(DL)
Range	10 km (URBAN), 50 km (RURAL)	5 km(URBAN), 15 km (RURAL)	15 km (URBAN)	1 km (URBAN)
Num. of channels / orthogonal signals	360 channels	10 in EU, 64+8(UL) and 8(DL) in US plus multiple SFs	40 1MHz channels, up to 1200 signals per channel	multiple channels
Link symmetry	×	✓	×	×
Forward error correction	×	✓	✓	✓
MAC	unslotted ALOHA	unslotted ALOHA	CDMA-like	?
Topology	star	star of stars	star, tree	star
Adaptive Data Rate	×	✓	✓	×
Payload length	12B(UL), 8B(DL)	up to 250B (depends on SF & region)	10KB	?
Handover	end devices do not join a single base station	end devices do not join a single base station	✓	?
Authentication & encryption	encryption not supported	AES 128b	16B hash, AES 256b	?
Over the air updates	×	✓	✓	✓
SLA support	×	×	×	×
Localization	×	✓	×	×

12.3 RoundTrip Time of Flight

Round trip time of flight (RTOF) is a method of establishing distance which does not depend on synchronized clocks between any parties. Rather, the transmitting node sends out a message marking its time of departure t_1^{trans} with its local clock. The receiving node marks it's time of arrival t_2^{rec} according to its local clock and replies at t_3^{rec} with an ack noting the total processing time $t_3^{rec} - t_2^{rec}$. The transmitting node receives this ack at t_4^{trans}, and can compute the total distance as

$$\begin{aligned}\Delta_{trans,rec} &= c \cdot \frac{(t_2^{rec} - t_1^{trans}) + (t_4^{trans} - t_3^{rec})}{2} \\ &= c \cdot \frac{(t_4^{trans} - t_1^{trans}) + (t_2^{rec} - t_3^{rec})}{2} \\ &= c \cdot \frac{(t_4 - t_1) + (t_2 - t_3)}{2}\end{aligned} \tag{1}$$

where the t_i are measured in absolute time, as the local clocks' differences cancel.

DRAFT

12.4 Reactive Systems

Definition 6 (Reactive system)**.** *A **Reactive System** (V, IC, RF) is a triple consiting of a finite set of variables V, an initial condition IC specifying the starting state of the system (i.e. the initial values of the variables), and a transition relation formula RF.*

with current state $V = \{pc, n, k\}$ where pc is the program counter. We can define the transition formula RF between V and the next state $V' = \{pc', n', k'\}$ as

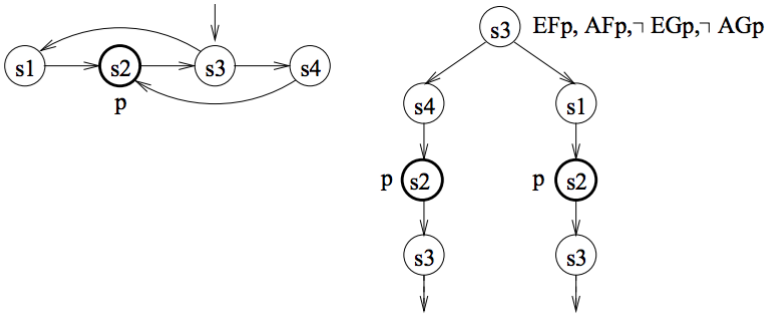
1. Gp (Globally p), assertion p holds in every state.
2. Fp (Future p), assertion p will hold in a future state.
3. pUq (p Until q), assertion p will hold until q holds
4. Xp (Next p), assertion p holds in the next state.

Here we give some examples with pictures to illustrate the semantics of CTL.

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

Example 12.1. The transition system T on the left is defined by:

- $S = \{s_1,\ s_2,\ s_3,\ s_4\}$
- $I = \{s_3\}$
- $R = \{(s_3, s_1),\ (s_4, s_2),\ (s_1, s_2),\ (s_2, s_3),\ (s_3, s_4)\}$

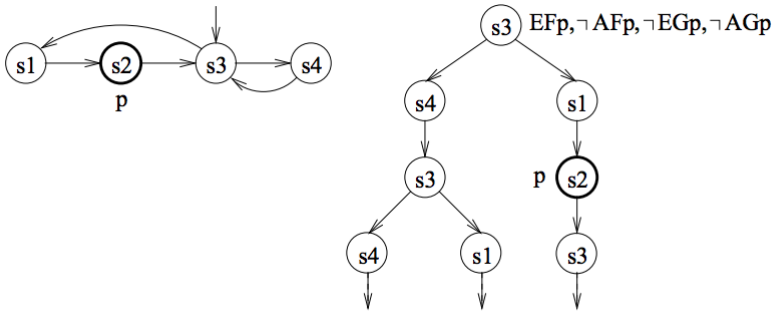


The property p is known to hold in s_2. Therefore, the following formulas are modeled by T, along with their English level interpretations:

- EFp – There exists a path where p holds in a future state.
- AFp – Forall paths, p holds in a future state.
- $\neg EGp$ – There is no path in which p holds in every state.
- $\neg AGp$ – It is not the case that p holds in every state of every path.

Example 12.2. The transition system T on the left is defined by:

- $S = \{s_1,\ s_2,\ s_3,\ s_4\}$
- $I = \{s_3\}$
- $R = \{(s_3, s_1),\ (s_4, s_3),\ (s_1, s_2),\ (s_2, s_3),\ (s_3, s_4)\}$



The property p is known to hold in s_3. Therefore, the following formulas are modeled by T, along with their English level interpretations:

- EFp – There exists a path where p holds in a future state (the right branch).
- $\neg AFp$ – Not all paths have p holding in a future state (e.g. the left branch).
- $\neg EGp$ – There is no path in which p holds in every state.

DRAFT

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

- $\neg AGp$ – It is not the case that p holds in every state of every path.

12.5 Rigidity

We take a moment to make a few observations on rigidity as it pertains to localization. By *rigity* we mean the property that the absolute position of an arbitraty node in the network – e.g. it's latitude, longitude – is guaranteed to be verifiable and computable given that the absolute positions of a much smaller subnetwork are known plus together edge weights representing distances between connected nodes. This is relevant given that the localization methods that a zone employs depend on known absolute positions of nodes belonging to that zone, and furthermore any localization data directly observable by a node is inherently local and relative.

Intuitively you can imagine rigity as follows. Suppose that the network is a weighted $\mathbb{K}_3$, in that every node is capable of communicating with every other node and computing the distance between itself and that node. Suppose that we have good reason to believe that v_1 and v_2 have the absolute position that they claim. Then there are still two possible embeddings of $\mathbb{K}_3$ that support this hypothesis (Figure)

Contrast this with the case of a $\mathbb{K}_4$, where the absolute position of any three nodes, together with the pairwise distances between all nodes, uniquely determines the absolute position of the fourth. This property is known as *global rigidity*, and occurs quite naturally in graphs in the plane of sufficiently high connectivity. Futhermore, in the plane there are strong enough characterizations that rigidty testing is polynomial in the vertex set [3].

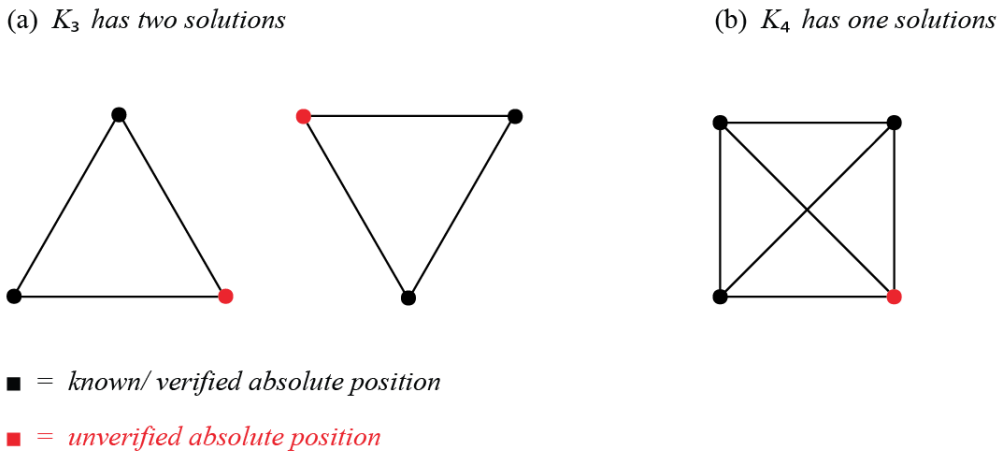


Figure 12: Rigidity Across Zones

12.6 Time Synchronization Protocol

Malekpour supposedly gives a more detailed description of the modeling process in [6], but we can go over the formulation of the desired propositions here:

1. **System Liveness**: The *convergence time* C , i.e. the amount of actual time after which you expect that the clocks of the good nodes all fall within the desired precision, is a derived parameter. The calculation is given in 3.6 of [5]. We define

$$ElapsedTime \;=\; (GlobalClock \;\geq\; C). \tag{2}$$

This draft document is not final and is subject to change upon the author's discretion. See the disclaimers on the last page of this document.

The **system liveness** proposition states that $AF(ElapsedTime)$, i.e. that every path has the property that eventually the convergence time is met. This in particular implies that it is possible to prove that convergence always eventually hapens event though we only have access to the local timers of the nodes.

2. **Convergence and Closure**: This is the most relevant proposition for the protocol, specifying whether or not the system will converge to the predicted precision after the elapse of convergence time, and whether or not it will remain within that precision thereafter. It can be stated as

$$\begin{aligned} AF(ElapsedTime) \wedge AG(ElapsedTime \ \rightarrow \ AllWithinProcision) \\ \wedge \ AG((ElapsedTime \ \wedge \ AllWithinPrecision) \\ \rightarrow \ AX(ElapsedTime \ \wedge \ AllWithinPrecision)) \end{aligned} \tag{3}$$

References

[1] Jeff Coleman et al. *Counterfactual Terminology* `https://github.com/ledgerlabs/state-channels/wiki/Counterfactual-Terminology`

[2] G. Mao et al. *Wireless sensor network localization techniques*. Computer Networks Vol 51, pp. 2529–2553, January 2007.

[3] Aspens et al. *A Theory of Network Localization*. IEEE Transactions on Mobile Computing, Vol. 5, No. 12, December 2006

[4] Mahya R. Malekpour *An Autonomous Distributed Fault-Tolerant Local Positioning System* 2017 NASA-TM-2017-219638, L-20782, NF1676L-26308

[5] Mahyar R. Malekpour *A Self-Stabilizing Hybrid Fault-Tolerant Synchronization Protocol*. 2015 IEEE Aerospace Conference, Big Ski, Montana, pp. 11, March 2015.

[6] Mahyar R. Malekpour *Model Checking a Self-Stabilizing Synchronization Protocol for Arbitrary Digraphs*. `https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20120016699.pdf`

Secure localization and location verification in wireless sensor networks: a survey

Location Verification using Secure Distance Bounding Protocols

LOCALIZATION SYSTEMS FOR WIRELESS SENSOR NETWORKS

Djenouri, Djamel, and Miloud Bagaa. "Synchronization protocols and implementation issues in wireless sensor networks: A review." IEEE Systems Journal 10.2 (2016): 617-627.

Low Power Wide Area Networks: An Overview

Robustness, Security and Privacy in Location-Based Services for Future IoT: A Survey

DRAFT

DISCLAIMER:
This technical whitepaper is a draft version and is subject to change by Foamspace without notice. This document is intended only for persons who receive it directly from Foamspace.
This document is for informational purposes only and is not an exhaustive discussion of Foamspace, its platform or the FOAM token -- while every effort has been made to ensure that the material in this technical whitepaper is accurate and current, no representation is made as to the accuracy of the materials. In addition, nothing in this document should be construed as professional advice or should be relied on. Foamspace does not guarantee, and accepts no legal liability whatsoever arising from or connected to, the accuracy, reliability, currency, or completeness of this technical whitepaper.

This technical whitepaper does not constitute an offer or solicitation to sell shares or securities in Foamspace, the FOAM Protocol or any related or associated company. The FOAM Token itself is not intended to be a security or otherwise represent equity, ownership, profits or proprietary rights in FOAM or the FOAM Protocol. None of the information or analysis presented herein is intended to form the basis for any investment decision, and no specific recommendations are intended. Accordingly, this technical whitepaper does not constitute investment advice or counsel or solicitation for investment, or purchase of, the FOAM Token or any financial product. This technical whitepaper does not constitute or form part of, and should not be construed as, any offer for sale or subscription of, or any invitation to offer to buy or subscribe for, any securities, nor should it or any part of it form the basis of, or be relied on in any connection with, any contract or commitment whatsoever. Foamspace and the FOAM Protocol expressly disclaim any and all responsibility for any direct or consequential loss or damage of any kind whatsoever arising directly or indirectly from: (i) reliance on any information contained in this technical whitepaper, (ii) any error, omission or inaccuracy in any such information or (iii) any action resulting therefrom.