AWALLET that supports ACOIN

A Decentralized Wireless Algorithmic *PoF*® WHITEPAPER

Prof.PCWY. Mr. Jean Michel-FLOC'H ACoin Blockchain Analyst Team

ACoin Systems, Inc. (To be incorporated)

Release V.1.1.5 (11th September 2021☑ Revised draft)

Abstract

We are on the almost decades when Bitcoin has first released. The creating blocks, peer to peer either recorded in open transparent ledgers or smart contracts kind of ecology has evoke our daily lifestyles, before I starts, the market cap. figures, of Crypto up to date are approx. USD2 trillion, and participants of Crypto is around 0.694%; 50 million people. Because the value of Blockchain or any other technology in reflections has to be accounted and compared with actual economy, and Bitcoin was first designed and planned to evolve the banking intermediate cost, therefore we must starts to share our concept here by connecting Blockchain technology with Crypto economy.

To starts further, we must agree that either Blockchain or traditional kind of server storage systems requires algorithmic process. Bitcoin and any other kinds of algorithm hash rate are process through GPU mining. These requires tremendous power supply as each mining machine needs 1350V to operates, and the carbon emissions during mining ruins our environments. Many countries has started to ban mining, and therefore is either we quickly go deep to find a solution or Crypto will gradually lose it "shines".

Acoin Blockchain mission is to carry out the legacy that Bitcoin has created but further more solve the problems regarding low power green mining and create and collaborate with more merchandisers for more users friendly and wider usage.

The ACoin network is a decentralized wireless network that works and enables devices anywhere in the world to wirelessly connect to the Internet and geolocate themselves to algorithm hash rate in low power supply of around 5W, that's USD1 electric fee per month. Powering ACoin network is a Blockchain with native protocol coin incentivizing a dual sided consumption between PoF® (Proof of Flow) and EoM (Ease of Mine), ACoin ecosystem embraces Bitcoin and reorganize and innovates with green mining and rollout for publicity to enroll as miners and not "centralized mining by miners".

Our secure and open-source primitives enable developers to build and set up home low-power, Internet-connected devices quickly and cost-effectively. The ACoin network has a wide variety of applications across media, gaming, NFTs and is carrying out Bitcoin's principals that's creating a real decentralized world, this time not only the coin itself, but also mining peer to peer.

1. Introduction

The word decentralize has been so common nowadays, it is the same with the 80s~90s when people talks about Internet. A multitude of platforms, technologies, and services are moving from centralized proprietary systems to decentralized, open ones. Peer to peer networks such as BitTorrent paved the way for Blockchain networks and Crypto-currencies to be built. Now Bitcoin, Ethereum, and other Blockchain networks have shown the value of decentralized transaction ledgers. Existing Internet services such as file storage, identity verification, and domain name system are being replaed by modern Blockchain-based versions. While software-level decentralization has moved quickly, physical networks are taking longer to affect. These networks are more complicated to decentralize as they often require specialized hardware to function.

The ACoin network is a wide-area wireless networking system, a Blockchain, and a protocol coin. The Blockchain runs on a new consensus protocol, called the ACoin Consensus Protocol, and a new kind of proof, called *Proof-of-Flow*®. The Miners who are identify by wireless network flow in a cryptographically verified physical location and time submit proofs to ACoin network, and the Miners submitting the best proofs are elected to an asynchronous byzantine fault tolerant concensus group receive encrypted transactions submitted by other Miners and forms them into blocks at an extremely high transaction rate. In addition to the Blockchain protocol, the ACoin protocol, Ethernet, provides a bi-directional data transfer system between Decives and the Internet via a network of independent providers that does not rely on a single coordinator, where:(1) Devices pay to send & receive data to the Internet and geolocate themselves, Miners earn coins for connecting with ACoin network flow, and for validating the integrity of the ACoin network.

Note: This Whitepaper represents a continuous work in progress. We will endeavor to keep this document current with the latest development progress. As a result of the ongoing and iterative nature of our development process, the resulting code and implementation is likely to differ from is represented in this paper.

1.1 Key Components

The ACoin network is built around the following key components:

Proof-of Flow® We present a computationally algorithmic *Proof-of-flow*® that allows Miners to prove they are providing wireless network mining in home. We anchor these proofs using a *Proof-of-IP.Identity*®. *on* that allows miners to prove they are accurately representing time relative to system on the network in a cryptographically secure way.

ACoin Network

We demonstrate an entirely new purpose-built Blockchain network built to service Ethernet and provide a system for authenticating and identifying devices, providing cryptographic guarantees of data transmission and authenticity, offer transaction primitives designed around Ethernet, and more.

ACoin Consensus Protocol

We present a novel consensus protocol construction that creates a permissionless, high throughout, censor-resistant system by combining an asynchronous byzantine fault tolerant protocol with identities presented via *Proof-of-Flow*®.

Ethernet

We introduce a open-source and standards compliant for ACoin network protocol, Ethernet, designed for low power Devices across territories and areas. This protocol is designed to run on existing commodity LAN/WAN available from dozens of manufacturers with no proprietary technologies or modulation schemes required.

Proof-of Territory®

We outline a system for interpreting the physical *geolocation* of a Device using WHIP without the need for expensive and power-hungry satellite location hardware. Devices can make immutable, secure, and verificable claims about their location at a given moment in time which is recorded in the Blockchain.

DWN

We will further adopt and present a decentralized wireless network (DWN) that provides wireless access to the Internet for Devices by way of multiple independent Miners and outlines the ACoin network and Ethernet specification by which participant in the Acoin network should conform. Routers pay this network of Miners are rewarded with newly-minted coins for providing network flow and delivering Device data to the Internet.

1.2 System Overview

The ACoin network is a *decentralized wireless network* built around Ethernet on a purpose-built Blockchain with a native coin.

Devices take the form of hardware containing a LAN/WAN and firmware compatible with Ethernet, and spend coins by paying Miners to send data to and from the Internet.

Miners earn coins by through wireless network flow via purpose-built hardware which provides a bridge between Ethernet and Routers, which are Internet applications.

Devices store their private keys in commodity key-storage hardware and their public keys in the Blockchain.

Miners join the network by asserting their LAN/WAN IP-derived location, a special type of transaction in the Blockchain, and staking a coin deposit.

Miners specify the price they are willing to accept for data transport and *Proof-of-Territory*® services, and Routers specify the price they are willing to pay for Device's data. Miners are paid once they prove they have delivered data to the Device's specified Router.

Miners participate in the creation of new blocks in the Blockchain by being elected to an asynchronous byzantine fault tolerant consensus group.

Miners are rewarded with newly minted protocol coins for blocks that are created while they are part of the consensus group.

A Miner's probability of being elected to the consensus group at a given epoch is based on the quality of the wireless network flow they collected.

The Blockchain employs *Proof-of-Flow*® to guarantee that Miners are honestly representing the wireless network flow they are creating.

(Figure 1) shows a visual representation of the ACoin network.

2. The ACoin DWN

We will be introducing the core components of the DWN.

2.1 Participants

There are three types of participants in the ACoin network: Device, Miner, or a Router.

Devices send and receive encrypted data from the Internet using hardware compatible with WHIP. Data sent from Devices is *fingerprinted*, and that fingerprint stored in the Blockchain.
Miners provide wireless network flow to the ACoin network via purpose-built hardware, called Spotflows®, which provide a multiple-range bridge between WHIP devices and the Internet. Users join the ACoin network as Miners by purchasing or building a Spotflow® that conforms to WHIP, and *staking* a coin deposit proportional to the density of the other Miners operating in their territory or area. Miners participate in the *Proof-of-Flow*® process to prove that they are continuously providing wireless network flow that Device can use. Miners join the ACoin network with a score that diminishes as blocks pass without valid proofs being submitted. At a given epoch, a new group of Miners are elected to a *consensus group* which mine new blocks in the Blockchain and receive the block reward and

transaction fees for any transactions included in the block once mined. As a Miner's score drops their probability of being elected to the consensus group and mining blocks diminishes.

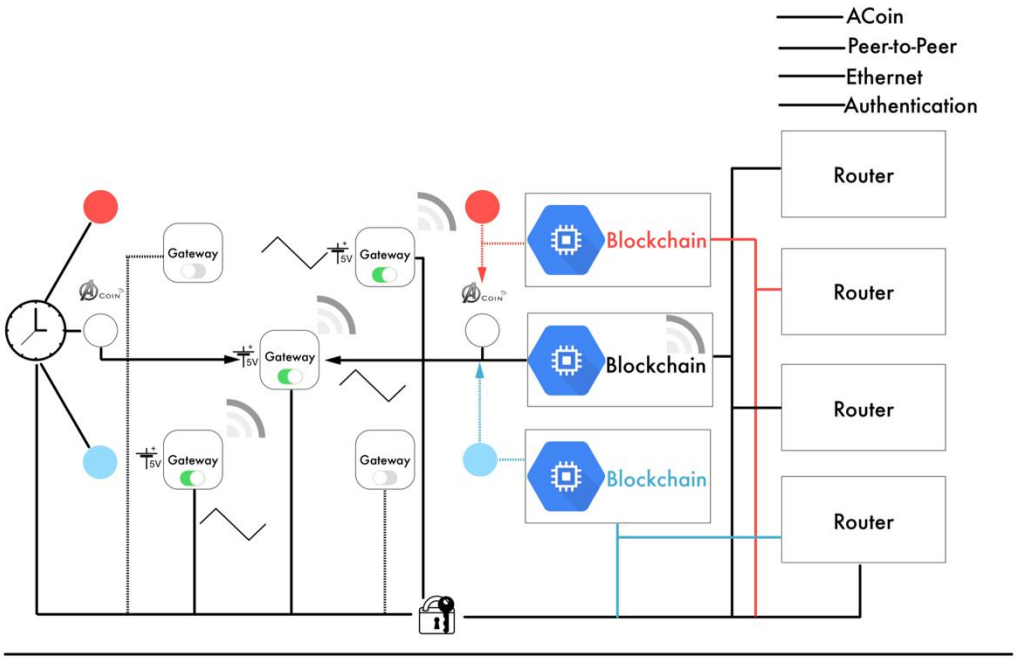


Figure 1. *System Overview*

Routers are Internet applications that purchase encrypted Device data from Miners. In locations with a sufficient number of Miners, Routers can pay several Miners to obtain enough copies of a packet to geolocate a Device without needing satellite location hardware, which we call *Proof-of-IP.Identity*®. Routers are the termination point for Device data encryption. Devices record to the Blockchain to which Routers a given Miner should send their data, such that any Spotflow® on the ACoin network can send any Device data to the appropriate Router. Routers are responsible for confirming to Spotflow®s that Device data was delivered to the correct destination and that the Miner should be paid for their service.

2.2 Blockchain

The ACoin network is a distributed ledger designed to provide a cost-effective way to run application logic core to the operation of a DWN, store immutable Device data fingerprints, and furnish a transaction system. The Acoin network is an immutable append-only list of transactions which achieves consensus using *Acoin Consensus Protocol*. Users internal and external to the DWN have access to the Blockchain, which is new protocol built from scratch specifically for the DWN.

The Blockchain consists of blocks which contain a header and a list of transactions. There are several kinds of transactions.

At a given epoch a given block consists of:

Block Version
Block Height
Previous Block Hash
Transactions *1...n* Merkle Hash
Threshold signature by the current consensus group

As the *Proof-of Flow* is valuable to the network , Miners are required to submit their proofs at regular intervals. All Miners have a score, which decays over time, and is boosted by submitting Proofs-of Flow to the Blockchain. At a fixed epoch, a HoneyBadgerBFT consensus group of the highest scoring Miners is elected. For that epoch, all transactions are encrypted and submitted to the consensus group for inclusion in the Blockchain. The consensus group is responsible for decrypting transactions using *threshold decryption*, agreeing on the validity an ordering of transactions, forming them into blocks, and appending them to the Blockchain for which the members of the consensus group receive a reward.

As the consensus group is validating transactions without having to provide an associated block-proof (beyond a threshold signature), there is partically no settlement time, and the transaction throughput is extremely high compared to a *Nakamoto Consensus* Blockchain such as Bitcoin or Ethererum. The ACoin Consensus Protocol is outlined in below.

2.3 Physical Implementation

The ACoin network is also a physical network installation. The participants in the ACoin network can be thought of as follow:

Ethernet The ACoin network uses a ethernet protocol, called *LAN*. LAN has no restrictions in modern usage, low-power, ethernet protocol suitable for use with commodity open-standards hardware. Ethernet compatible hardware can communicate with Blockchain server over many close and square miles in any environments or hundreds of any miles in rural settings. Ethernet compatible hardware connects with Blockchain server uses strong public key cryptography and authentication occurs using the ACoin Blockchain, and data is encrypted end-to end between the device and corresponding Internet-hosted router.

Spotflows are physical network devices that provide close and wide area wireless flow and participate in the ACoin network. Spotflow®s transmit data back and forth between Routers on the Internet and Devices while generating *Proofs-of-Flow*® for the ACoin network. Spotflow®s are manufactured using commodity open-standards components with no proprietary hardware. Spotflows

can co-operate and geolocate Devices using the ACoin network without any additional required hardware. Each Spotflow® can support thousands of connected Devices, and provide flow over many square miles. Miners operating Spotflow®s specify the price they are willing to accept for transport and *Proof-of-Territory*® services for Devices.

Devices exist in the form of hardware products that contain a compatible Ethernet transceiver and communicate with Spotflow®s on the ACoin network. Ethernet is designed to facilitate low power data transmission and reception, Devices can exist in a variety of forms, depending on the product or use case, and a variety of transmission and reception strategies can be employed to optimize for transmission/reception frequency. Device manufacturers are encouraged to use hardware-based key storage which can securely generate, store, and authenticate public/private key pairs without leaking the private key.

Here, we expand on the components of the network.

2.4 Ethernet Protocol

2.4.1 Motivation

LAN/WAN technologies are available today. These technologies focus on creating all ranges, low power Internet communication for sensors and other smart Devices. The PSTN and Ethernet were designed for ACoin Consensus purposes. The result is specify in two areas: switching techniques and network access methods. In Ciruit-Switching; the bandwidth geolocations, through modest, is guaranteed.

Traffic	Bandwidth Required	Burst Support	Latency
Data	Variable bandwidth needs	Extremely important	Not important

In ACoin Ecology we believed that participants in contributions are the most important factors, thus WiFi network has significantly higher data rates and therefore creating mass participations are necessary. We do not consider an open alliance built on top of proprietary hardware to be an acceptable compromise. While such as IEEE 802.15.4 used in the first generation of our wireless products:

1. **New types of cables**: While coaxial is still an option, newer installations use less expensive unshielded twisted pair or higher capacity fibre.
2. **New toplogies**: Cabling for the newer Ethernet standards use a star, a bus-star hybrid called a tree, and even a ring.
3. **Increased bandwidth**: The standards now define speeds between 1 Mbps and i Gbps (soon 10 Gbps).
4. **Support for full-duplex operation**: The original standard supported half-duplex only. (On full-duplex networks, CSMA/CD is not required.)
5. **Expansion of the distances supported**: Ethernet is no longer restricted to the LAN. It is now deployed in MAN network, and will soon provide the underlying service in WAN (wide area network) environments as well.

6. **Support for new applications**: Gigabit and 10 Gigabit Ethernet are able to provide transport for Blockchain data.

This open solutions eventually drove the creation of a new protocol.

2.4.2 Outline

We introduce Ethernet. Ethernet is a highly secure, all-range, low power, bi-directional network protocol that is compatible with a all range of existing transceivers operating in the original standards. Authentication with the network uses modern public-key encryption and key pairs, with the public keys for all participants stored in the Blockchain.

Ethernet Standardisation

Current Standards

Standardisation is a key to the wide acceptance of Ethernet. The original standard, IEEE 802.3, was finalised in 1983. It has been updated repeatedly since then. The scope of this paper doesn't permit a discussion on each supplement, but a brief description of the most important ones follows. If you want more information, the complete standard is available from the IEEE (www.ieee802.org/3/).

Suppliment	Name	Speed	Max Line Length	Medium	Duplexing
802.3i	10BASE-T	10 Mbps	100 metres	UTP,Category 3+	Half-and full-duplex
802.3u	100BASE-TX	100 Mbps	100 metres	UTP,Category 3+	Half-and full-duplex
	100BASE-TX	100 Mbps	2 kilometres	Multimode fibre	Half-and full-duplex
802.3z	1000BASE-TX	1 Gbps	5 kilometres	Singlemode fibre	full-duplex
	1000BASE-TX	1 Gbps	220-550 metres	Multimode fibre	full-duplex
	1000BASE-TX	1 Gbps	100 metres	UTP,Category 3+	full-duplex

Emerging Standards

The EEE 802.3 committee has two groups working on the other standards that you may find interesting:

1. The 10 Gb/s Ethernet Task Force is working on a standard for 10 Gigabit Ethernet (802.3ae). For more information, see the group's web site at grouper.ieee.org/groups/802/3/ae/.
2. The Ethernet in the First Mile Working Group (802.3ah) is preparing a standard addressing Ethernet to the home. For more information, see the group's web site at www.eee802.org/3/efm/.

Relationship to the OSI 7-Layer Model

Data networking professionals often categorize network services according to the OSI 7–Layer Reference Model, which is also sometimes called the OSI 7–Layer Reference Model. An in–depth discussion of this subject is beyond the scope of this white paper. For those who ar curious, Ethernet fits into Layer 1 and Layer 2 of this model. For more information, there are many good books on the

subject, and shorter discussions can be found on many internet web sites.

We choose bandwidth to accomplish the following goals:

Range Bandwidth allows for extremely all–range communications, with data rates that scale both up and down depending on the density of Spotflow®s.

2.4.3 Implementation

Ethernet supports several data rates, channel bandwidths, and error–correction techniques. Spotflow®s and Devices dynamically negotiate the combination of these options using a *signalling packet* delivered at the lowest bandwidth and symbol rate to ensure maximum range for the initial communication.

The full Ethernet specification will be made available by the Decentralized Device Network Alliance.

2.5 Spotflow®s

Spotflow®s are physical network devices operated and extracted by Miners that create Ethernet flow over all areas. They transmit data back and forth between Routers on the Internet and Devices on the network, process Blockchain transactions, and create *Proofs–of–Flow*® for the Acoin network. Spotflow®s is connected to the Internet network. Spotflow®s can connect to the Internet using TCP/IP capable backhaul, such as WiFi or Cellular. Each Spotflow® contains a frontend chip capable of receiving to several Mbps at a time and can hear all traffic transmitted within the spectrum. In this configuration modulation and demodulation is done in software. The benefit of this structure is that Spotflow®s can hear any Device traffic transmitted within the Mbps range, and no synchronization between the Spotflow® and Device needs to occur. This allows Devices to remain inexpensive and relatively simple and reduces protocol overhead. If a Miner wishes to minimize their Spotflow® hardware costs, synchronized LAN/WAN hopping schemes are also permitted within the specification as a cheaper alternative to a more expensive frontend.

Spotflow®s require a LAN/WAN IP.address is used in conjunction with other techniques to verify that a Spotflow® is, in fact, providing network low in the location it claims. Because LAN/ WAN IP.address location messages are impossible to fabricate and so not necessarily prove that Blockchain algorithmic data is being created, multiple mechanisms are not required to validate this work as described in more detail in.

LAN/WAN IP.address information is also correlated with packet arrival events to provide *Proof–of–Territory*® for Devices if multiple Spotflow®s observe the same packet. This allows devices to locate themselves, and therefore provide accurate territory data at a fraction of the cost of competing methods.

We will make both a complete open–source reference design and a finished product available at launch of the ACoin network.

2.6 Devices

A Device is any LAN/WAN IP.address hardware capable of communicating with Spotflow®s via Ethernet. Ethernet is designed to facilitate low power data transmission and reception, so typically devices would exist in the form of sensors that can function for perpetually.

Ethernet is designed such that Devices can be manufactured using commodity hardware available from a wide variety of vendors with very low-cost bill of materials (BOM).

2.7 Routers

The task of defining a Local Area Network (LAN) domain is accomplished using a router. Routers ar located at the service provider's central office and interface with the LAN router located at the customer's premises. Routers pass traffic only to the intended destinations, and block all broadcasts as configurated. Multiple routers are common within the customer's LAN domain, used as needed to segment large LAN installations. The internet is built using many thousands of routers that define all networks and services that make up this vast global information resource.

Routers are internet-deployed applications that receive packets from Devices via Spotflow® and route them to appropriate destinations such as an HTTP or MQTT endpoint.

Routers serve several functions on the ACoin network，including：

Authenticating Devices with the ACoin network;

Receiving packets from Spotflow®s and routing them to the internet;

Delivering downlink messages, including OTA updates, to Devices via Spotflow®s;

Providing delivery confirmations to ensure transport transactions are honest;

Providing authentication and routing mechanisms to third-party cloud services.

Storing and making available a full copy of the Blockchain ledger by acting as *full node*.

When a Spotflow® receives a data packet from a Device on the Acoin network, it queries the Blockchain to determine which Router to use given the Device's ACoin network address. Anyone is free to host their own Router and define their Device' traffic to be delivered there by any Spotflow® on the Spotflow® network. This ability allows users of the ACoin network to create VPN-like functionality whereby encrypted data is delivered only to a Router (or set of Routers) that they specify and can optionally host themselves.

Routers can implement a system called a channel which handles the authentication and routing of data

to a specific third party Internet application. These channel implementations can take advantage of a Device's onboard hardware security to create a secure, hardware-authenticated connection to a third party which would otherwise be difficult to implement directly on an embedded microcontroller. We will make available an open source reference implementation of a Channel that can be used to build additional interfaces to Internet services.

3. *Proof-of-Flow®* and *Proof-of-IP.Identity®*

In the ACoin network, Miners must prove that they are providing wireless network flow that Devices are able to use to communicate with the Internet. Miners do this by complying with the *Proof-of-Flow®* protocol which the ACoin network and other Miners audit and verify. We use a *Proof-of-IP.Identity®* to ensure that Miners are correctly representing their time in relation to others on the network, and obtain cryptographic proof of dishonest behavior. Several components of the ACoin network, such as *Proof-of-Flow®*, use *Proof-of-IP.Identity®* as a cryptographic "anchor" that root those occurrences with a cryptographic time proof. With a combination of *Proof-of-Flow®* and *Proof-of-IP.Identity®* we can obtain cryptographic proof of the approximate location and time of events occurring within the ACoin network.

3.1 Motivation

Most existing Blockchain networks such as Bitcoin and Ethereum use a Proof-of-Work system that relies on an algorithmic puzzle that is asymmetric in nature. These proofs are extremely difficult to generate, but simple for a third party to verify. Security on these networks is achieved by the network-wide consensus that the amount of computing power required to generate a valid proof is difficult to forge, and as subsequent blocks are added to the Blockchain, the cumulative difficulty of the chain becoming prohibitively difficult to fabricate.

These computation-heavy proofs are, however, not otherwise useful to Blockchain networks. We define useful as work that is valuable to a Blockchain network beyond securing the ledger. While there have been attempts in other networks to turn mining power into something useful, such as Ethereum executing small programs called smart contracts, the majority also extremely wasteful, as the determining factor in the work is typically computational power, which consumes massive amounts of electricity and requires significant hardware to execute.

The proofs used in the ACoin network must be resistant to *Sybil attacks* in which dishonest Miners create pseudonymous identities and use them to subvert the ACoin network and gain access to block rewards to which they should not be entitled. This is particularly difficult attack vector to manage in a physical network like the ACoin network. We must also be resistant to a new attack vector: *alternate reality attacks*, which exist where a dishonest group of Miners are able to simulate that wireless network flow exists in the physical world when it in fact does not. An example of this would be running the mining software on a single computer and simulating GPS coordinates and RF networking.

We later propose the ACoin Consensus Protocol that uses *Proof–of–Flow*® to both secure the Blockchain and provide an extremely useful service to the ACoin network, which provides wireless network flow that Devices can use to send data to and from the Internet.

3.2 Inspiration

Proof–of–Flow® is an innovative proof that allows Miners to prove that they are providing wireless network flow *W* in a specific region to a challenger, *C*. *Proof–of–Flow*® is an interactive protocol where a set of targets T_n assert that *W* exists in a specific GPS location *L* and then convinces *C* that T_n are in fact creating *W* and that said flow must have been created using the wireless RF network. ***Proof-of-Flow***® is the first such protocol that attempts to prove the varacity of miners in a physical space, and then use it to acheive consensus on a Blockchain network.

With ***Proof-of-Flow***® we aim to solve for the following:

Prove that Miners are operating RF hardware and firmware compatible with WHIP;

Prove that Miners are located in the geography they claim by having them communicate via RF; and

Correctly identify which version of reality is correct when there is a conflict

Proof of Flow® is inspired by the Helium Systems which devises a system of service of forming People's Network.

ACoin combine ***Proof-of-Flow***® with ***Proof-of-IP.Identity***® a proof that allows Miners on the ACoin network to achieve cryptographic time consensus among decentralized clients. We aim to achieve rough time synchronization in a secure way that does not depend on any particular time server, and in such a way that, if a time server does not misbehave, then clients end up with cryptographic proof of that behavior.

3.3 Constructing *Proof-of-Flow*®

With the ***Proof-of-Flow***® protocol, we aim to construct a proof that takes advantage of the following characteristics of LAN/WAN IP.address that are unique and different to Internet communication:

1. LAN/WAN IP.address has no limit physical propagation and, therefore, distance;
2. The strength of a received LAN/WAN is inversely proportional to the square of the distance from the transmitter; and
3. LAN/WAN travels at the speed of light with (effectively) no latency

Our goal is to verify whether Miners in a physical region are acting honestly and mining via wireless network flow compatible with Ethernet. To do this, we might requires a challenger *C* deterministically constructs a multi–layer data packet *O* which begins at an initial target, *T1*, and is broadcast wirelessly to a set of sequential targets, Tn, each of which are only able to decrypt the outer–most layer of *O* if

they were the intended recipient. Each target signs a receipt, K_S, delivers it to C, removes their layer of O, and broadcasts it for the next target, Essentially an "envelope of envelopes" only decipherable by the intended recipient

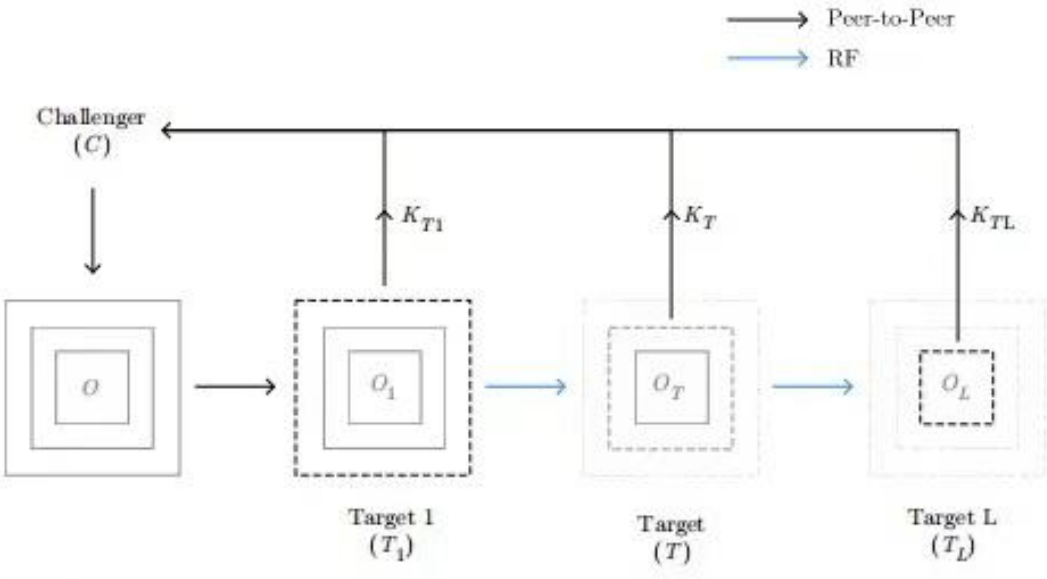


Figure 2. *Multi-Layer Data Packet Deconstruction*

3.3.1 Selecting the Initial Target

We aim to deterministically locate a geographic reference target, T, for the challenger, C. Both C and T are Miners in the ACoin network. T does not need to be geographically proximate to C. To locate T, C initially seeds verifiable entropy, η, into the selection process by signing the current block hash with its private key. Since the probabilities associated to each miner from a discrete probability distribution (Equation 1), C uses the probability associated to each eligible Miner to locate T and applies the inverse cumulative distribution function using a uniform random number generated via η. This allows us to ensure that we always target potentially dishonest Miners as they have a lower score, thus increasing their probability of being targeted by C. Given that a Miners score is diminishing linearly over time, it is necessary to create this inverse relationship to give low–scoring Miners an opportunity to participate in the process and increase their score. This diminishing score also incentivizes all the participants to spend receipts to C and broadcast the remainder of O.

3.3.2 Constructing the multi-layer challenge

Once T has been selected, C must construct a multi–layer challenge, O. O is a data packet broadcast over the ACoin network and received by geographically proximate targets T_n. Geographically proximate is defined as within a radius of T, a network value T radius . Each layr of O, O_l , consists of a threetuple of $E(S,\Psi,R)$, where E is a secure encryption function using the Elliptic–Curve Diffie–Hellman (ECDH) derived symmetric key, S is a nonce, Ψ is the time to broadcast the next layer of the challenge and R is the remainder of O consisting of recursive three–tuples. The maximum number of O_l *is bounded by a network value, Omax.*

The construction logic of *O by C is as follows:*

1. A set of candidate nodes, T_n, are selected such that all members of T_n are within a contiguous radio network that also contains T;
2. Two targets, T_1 and T_L, are selected by finding the highest scoring targets in T_n furthest from T;
3. A weighted graph, T_g, is constructed from T_n such that members of T_g in radio range of each other are connected by an edge weighted by the value of $1-(score(T_a)-score(T_b))$;
4. The shortest path between T_1 to T to T_L is computed using Dijkstra's algorithm using the edge weights from the previous step;
5. An ephemeral public/private keypair E_K and E_{K-1} are generated;
6. A layer O_l is created and added to O, and S is encrypted with the combination of the public key of T_L, retrieved from the Blockchain as T_{LK} and E_{k-1} as an ECDH exchange to compute a shared secret, know only to both parties C and T_L; and
7. The previous step repeats with additional layers added to O until all $T_L \rightarrow T_1$ have a layer O_l included in O

The resulting O can be visually represented.

3.3.3 Creating the Proof

Once O has been constructed, it is delivered to T_1 via the ACoin network and immediately broadcast by T_1 via the ACoin network. WHIP is not a point-to-point system, so several Miners within proximity of T_1 will hear O. In this example, only the specific target T will be able to decrypt E and send a valid receipt back to the challenger, C.

We describe the approximate flow of *Proof-of Flow*® creation as follows:

1. *T1* receives O from C via the ACoin network, decrypts the outermost layer and immediately broadcasts it R via the ACoin network;
2. T hears O and attempts to decrypt the value of E by using its private key where p_k: $E_{pk}(S,\Psi,R)$;

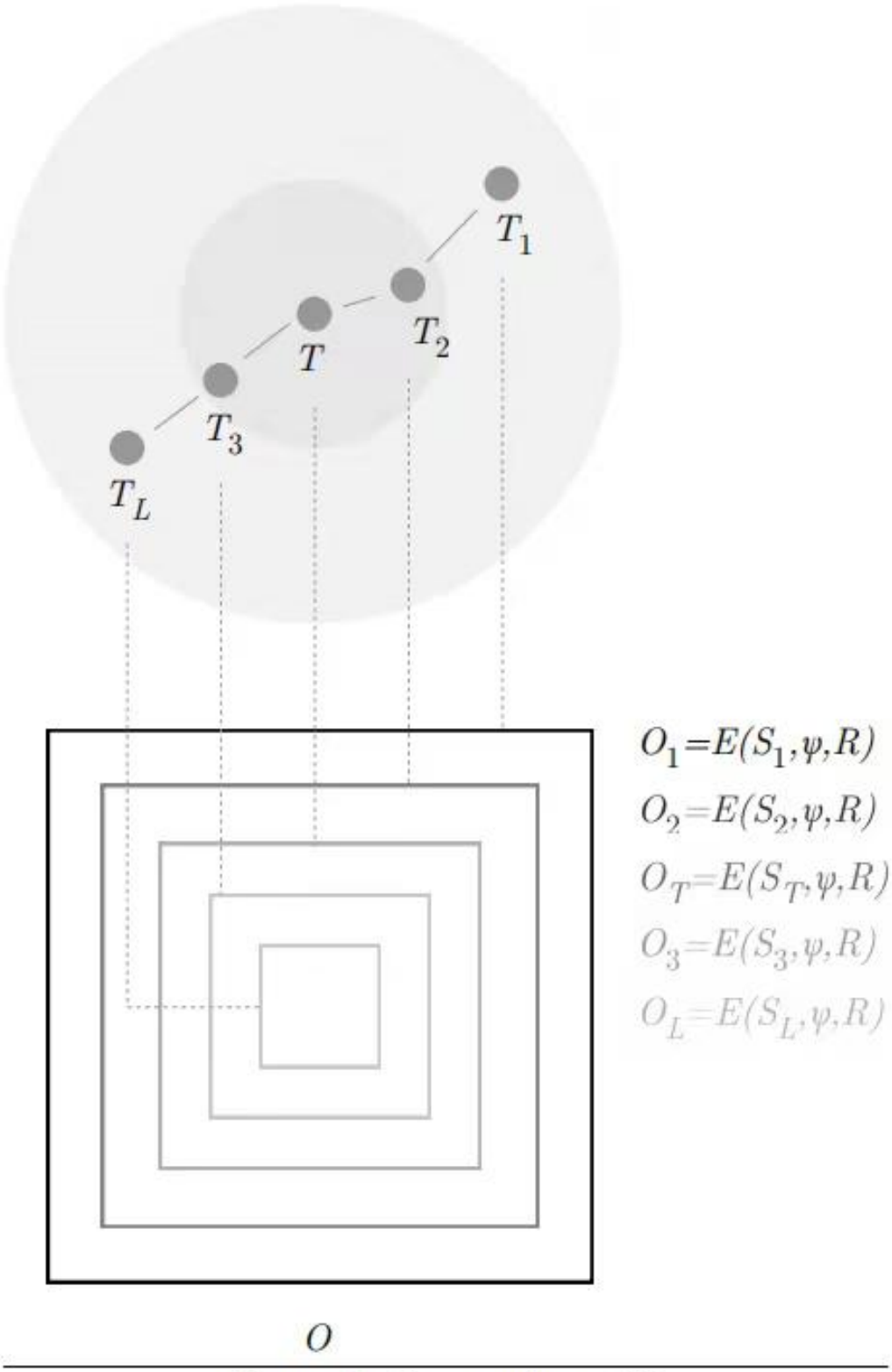


Figure 3. Construction of O

3. T records both time of arrival β and the signal strength v of O;
4. If successful, T then creates signed receipt $K_{\S}$, where $K_{\S} = (S||\beta||v)$ signed by the private key of T;

5. *T* submits *K§* to *C* via ACoin network, removes the outer most layer, and wirelessly broadcasts the remainder *O*; and
6. These steps repeat for *T1*.. *T*... *TL*, with *TL* being the last target in the graph

C expects to hear responses from *Tg* within a time threshold 入, otherwise it considers the *Proof–of–Flow*® to have concluded. Because *C* is the only party with complete knowledge of *O*, upper bounds of the values for *β* and v are assigned by *C* which are used to verify that each layer of *O* was transmitted approximately where and when it was expected. The upper bound for *β* is limited by the speed of light τ between *Tn* and *Tn–1*. Thus we know that, subject to some slight delays from reflection or multipath, the packet should not arrive at *Tg* later than τ multiplied by the geographical distance *D* plus some small episilon value, v = τ x (*D* + ę). For v , because of the inverse–square law, we can calculate the maximum RSSI (Received Signal Strength Indication) possible for a packet transmitted, 乄, from *Tg – 1* to *Tg* as 乄 = 1/D2. Spotflow®s that are closer than expected, or which are transmitting at a higher power to mask their location disparity, are unlikely to get 乄 correct, given that they do not know who the next layer *O* is addressed to.

Once *TL* has delivered receipt to *C*, or 入 has elapsed, the *Proof–of–Flow*® is completed. The collection of signed receipts, *K§*, constitute the *Proof–of–Flow*® that *C* will submit to the ACoin network.

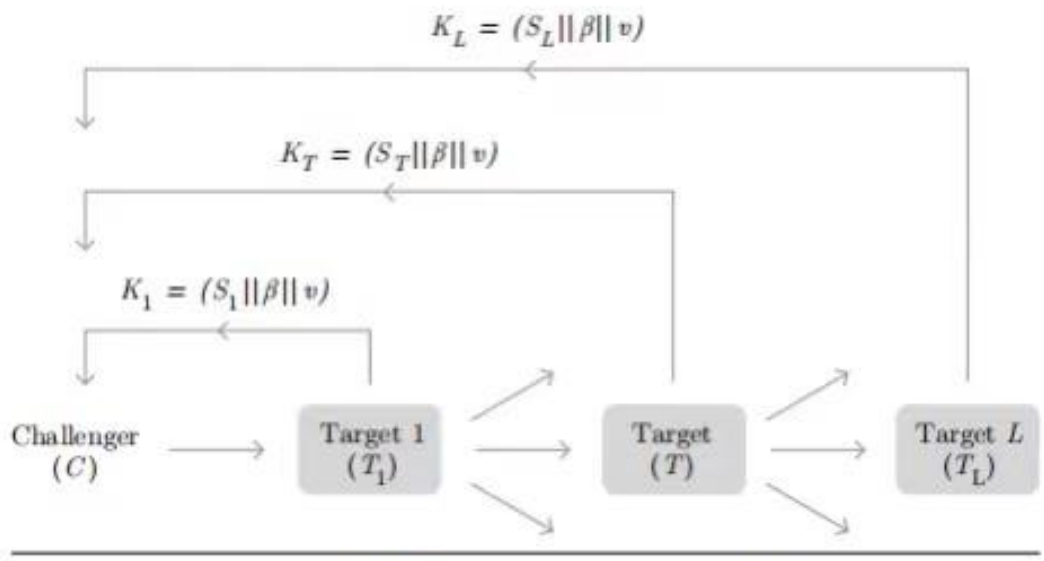


Figure 4. *Proof-of-Coverage* flow

3.3.4 Scoring

The score allocated to a Miner, and therefore the resulting score of the ***Proof-of-Flow***®, is an integral part of the ACoin Consensus Protocol described. When Miners join the ACoin network, they are assigned a score, Φɱ. We consider any Miner with a score greater than Φɱ to be an honest miner. This score depreciates according to the number of verifications the Miner has as well as the height since its last successful verification. As Φɱ decreases the probability of the Miner *M* being the target for *C* increases, such that the ACoin network continually attempts to prove that the lowest

scoring Miners are acting honestly, and giving Miners a reasonable chance to improve their scores.

In order to acheive this behavior we define the following invariants:
M, Miner
V , number of successful verifications for *M* –number of failed verifications for *M*
h, height since the last successful verification for *M*

If we assume that the ideal verification interval for any Miner is close to 240 blocks (4 hours if we assume a 60 second block time), we scale these invariants to fit the scoring functions:

V ' v /10.0
h', *h*/480

Using the above we can now construct a staleness–factor, *ő* , which would be used in determining the score of the Miner *M*.

$$\text{Ő}m = \begin{cases} -(8.h')2 & v'=0 \\ v'.(1-h'2/min(0.25,v')) & v'>0 \\ v'.(1-10.v'.h'2) & v'<0 \end{cases}$$

The above conditions strictly adhere to the following principles:
1. A negative v indicates that Miner is consistently failing verification.
2. If v = 0, then we do not have any trust information, therefore, we use a steep parabolic curve for the decay dependent on *h*'.
3. If v > 0, then it implies that the Miner has been successfully verified consistently, hence, we use an inverse parabolic curve that crosses the Y axis at 1, where the width of the parabola increases as a factor of v up to 0.25. This implies that the more positive verifications the Miner has accrued, the slower its score decays as a factor of *h*'.
4. Finally, if v < 0, then this is the inverese of the above case, wherein, a Miner has consistently been failing verification. Therefore, we use a similar parabola as above; however, the width of the parabola decreases as a factor of v , leading to a higher score decay for the Miner as a factor of *h*'.

[Figure 5] shows the trends for each of the above functions.

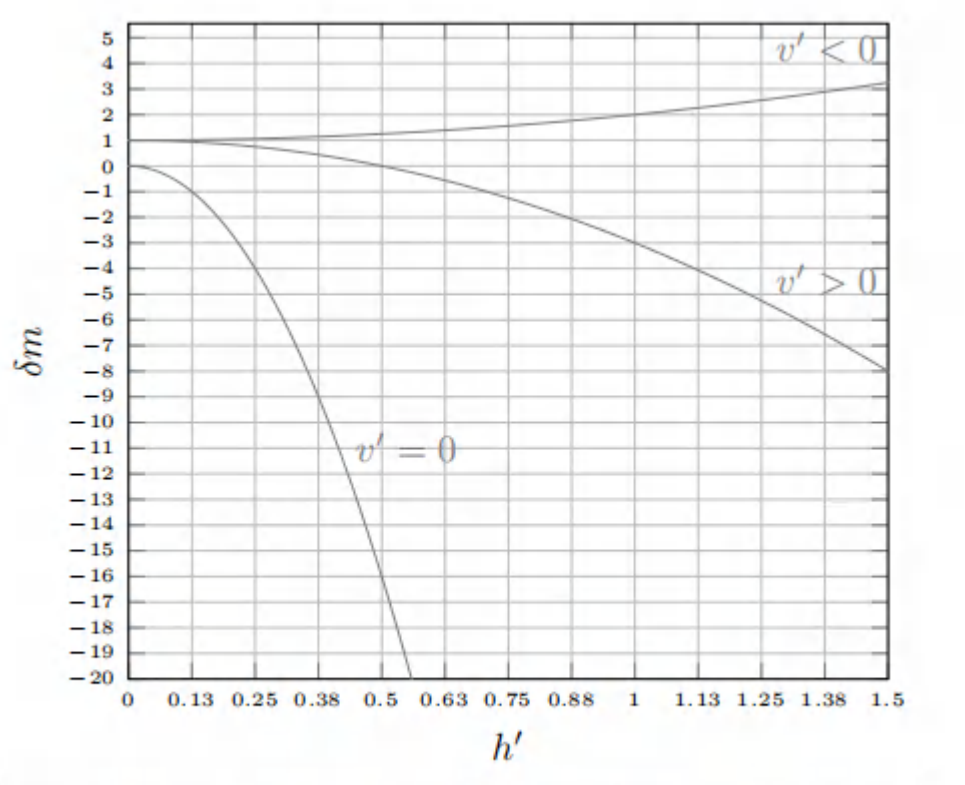


Figure 5. *Trendlines for the scoring functions*

Adhering to the above set of rules, we define the following scoring function, which is essentially a variation of a sigmoid curve fluctuating between values (0,1):

$$\Phi m = \frac{\text{artcan } (2.\acute{o}\ m) + 1.58}{3.16}$$

This scoring function yields [Figure 6], which shows the variation of the score with the staleness–factor:

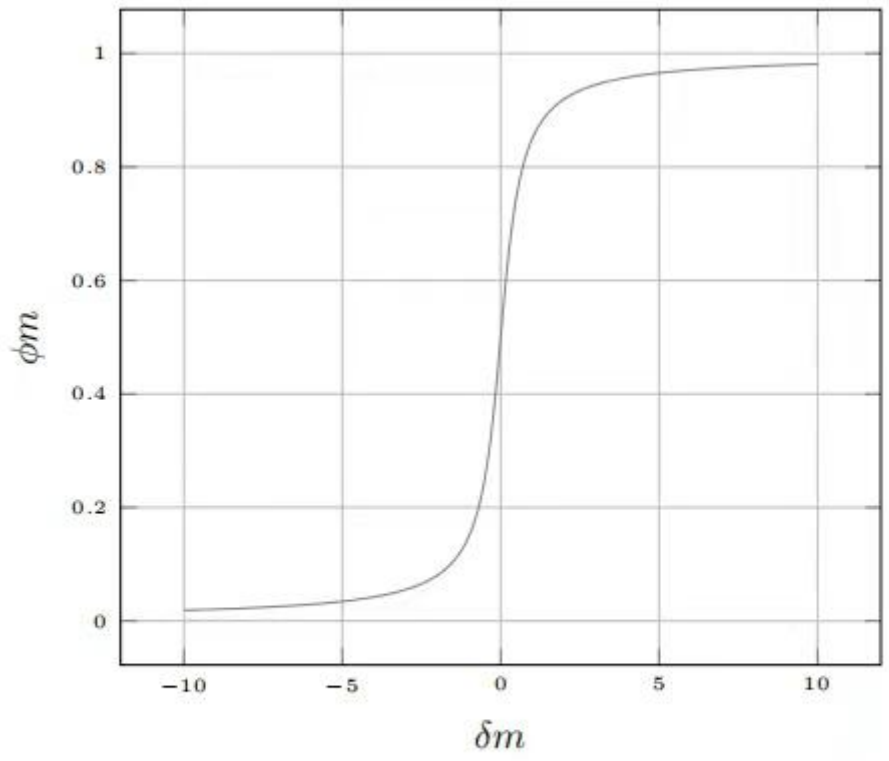


Figure 6. *Scoring algorithm and the resulting staleness factor*

[Figure 7] shows a snapshot of a random subset of the ACoin network at any blockchain height ***h***. The Miners represent random locations with an illustrated score, while the edges are calculated using Dijkstra's algorithm.

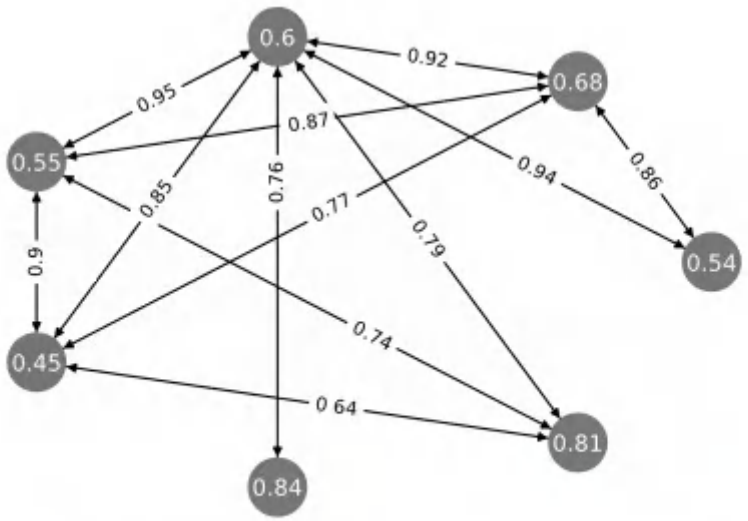


Figure 7. *Snapshot of a random subset of the initial network*

After 10,000 iterations the ACoin network appears as represented in [Figure 8].

The goal of this system is to ensure that the scoring algorithm considers that some Miners may attempt to act dishonestly. However, because the calculated edge–weights (via Dijkstra's algorithm) and the target selection mechanism ensure that we only boost the score of a Miner when it is being verified by other high scoring Miners, we believe that the system will favor legitimate Miners and deter dishonest ones.

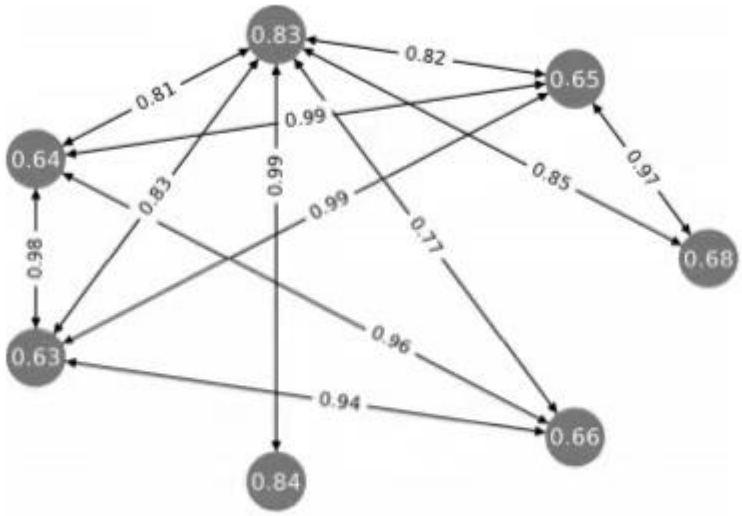


Figure 8. *Snapshot of a random subset of the network after 10000 iterations*

3.3.5 Target Selection

Due to the way scoring decays, there is a possibility that a given Miners' score may become stale as that Miner may not be verified within a reasonable interval. We therefore structure the target selection mechanism to give Miners a statistically target as their score decays. This is accomplished by biasing the probability of Miners being selected as potential targets based on their individual scores.

Let the set of miners be defined as:

$$M = \{m_1, m_2, m_3 \ldots m_n \mid n > 1\}$$

Let the set of miner scores be defined as:

$$S = \{\Phi m, m \in N\}$$

We assign the target selection probability to each miner in the following way:

$$P(m) = \frac{1 - \Phi m}{}$$

$$n - \sum_{i=1}^{n} \Phi m_i \tag{1}$$

The above equation ensures that the Miner with the lowest score is assigned the highest probability of being selected as a potential target while the opposite holds for the Miner with the highest score.

Furthermore, it also asserts that the probabilities are inversely proportional to the score of an individual Miner. This allows us to successfully target potentially low scoring Miners and improve the overall balance of the scoring system.

Another valuable aspect of assigning the probability as shown above is that all the probabilities together from a discrete probability distribution. A discrete probability distribution satisfies the following equation:

$$\sum_{i} P(M = i) = 1$$

3.3.6 Verifying the Proof

Once TL has delivered K_S or 入 has elapsed, the *Proof–of–Flow*® is considered complete. When C submits this proof, via a special type of transaction, all receipts K_S from $T_1 \ldots T_L$ are included in the transaction published to the ACoin network. As all the steps originally completed by C are deterministic in nature with verifiable and recreatable randomness, it is simple for verifying Miner, V, to recreate the original steps and verify that the proof is legitimate.

Verifying Miners in the consensus group who see the proof transaction are able to verify the *Proof–of–Flow*® by recreating the following steps:

1. The verifying Miner, V, reconstructs the set of Miners N;
2. The random seed η can be verified by V to have been created at approximately the correct time by the private key of C;
3. V then selects T from N, as seeding with η will result in the same target selection;
4. The set of candidate T_n, are reconstructed from which T_1 and T_L are determined;
5. Dijkstra's algorithm is used to reconstruct the graph T_g; and
6. The K_S receipts contained in B_c are verified to have been signed by the private keys of $T_1 \ldots T \ldots T_L$

Assuming these steps are completed successfully, the *Proof–of– Flow*® is verified the score of C is adjusted appropriately.

3.4 Constructing Proof-of IP.Identity®

To achieve cryptographic time consensus among decentralized clients, we implement a simplified form of Internet's Roughtime. Roughtime is protocol that aims to acheive rough time synchronization in a

secure way that does not depend on any particular time server, and in such a way that, if a time server does misbehave, clients end up with cryptographic proof of that behavior.

This section describes the construction of the *Proof–of–IP.Identity*® protocol.

3.4.1 Creating the Proof

We outline the approximate process to achieve cryptographically secure time as follows:

1. To begin, a Miner M pseudo–randomly picks two Miners M_1 and M_2, with whom to prove contact IP.Identity;
2. It is assumed M has a public key for M_1 and M_2 otherwise M should obtain it from the blockchain;
3. M generates a nonce, R, which is a SHA512 hash of the *Proof–of–Flow*®, which M has partially constructed;
4. M then generates a salted hash commitment, K, called the proof–kernel, where $K = H(\boldsymbol{R}\|\boldsymbol{M_1}\|\boldsymbol{M_2})$;
5. M sends K to M_1. M_1 replies with T, a signed message including the current time T_1 and K; and
6. M knows that the reply from M_1 was not pre–generated because it includes the nonce R that M generated

Because M can not trust M_1, it will ask for another time from M_2.
1. For the second request, a new nonce R is generated using T truncated to 512–bits, blinded by XOR'ing a randomly generated 512–bit number;
2. M then generates a sub–proof–kernel, $L = H(\boldsymbol{R}\|\boldsymbol{T}\|\boldsymbol{K})$, and sends it to M_2;
3. M_2 replies with U, a signed message including the current time T_2 and L; and
4. U is now a proof artifact that shows that M desired and then proved a ID.Identity between M_1 and M_2

With only two servers, M can end up with proof that something is wrong, but no idea of the correct time But with half a dozen or more independent servers, M will end up with chain of proof of any server's misbehaviour, signed by several other, and enough accurate replies to establish the correct time, T_t.

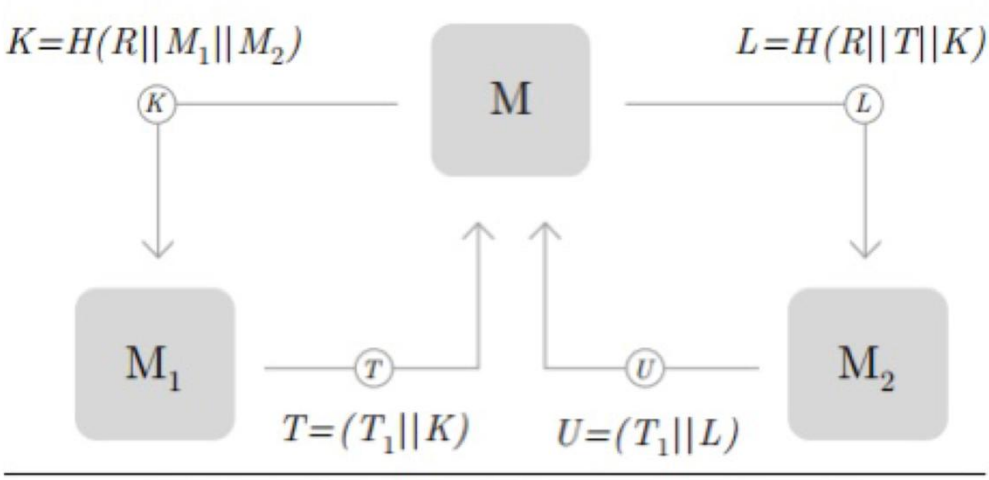


Figure 9. *Creating* ***Proof-of-IP.Identity***

3.4.2 Verifying the Proof

If we assume that the times from M_1 and M_2 are significantly different, and the time from M_2 is before M_1, then M has proof of misbehaviour. The reply from $M2$ implicitly shows that it was created later because of the way that M constructed the nonce. If the time from M_2 is after M_1, then M can reverse the roles of M_1 and M_2 and repeat the process to obtain, assuming steady clocks, a misordered proof as in the other case.

To verify the correct time, it is necessary for M to repeat the time synchronization process with enough Miners to gain consensus on the correct time:

1. A Miner M again pseudo–randomly selects n Miners $M_1 \ldots M_n$;
2. M generates a salted hash commitment, K, and delivers it to $M1$, where $K = H(\boldsymbol{R}\|\boldsymbol{M_1}\|\boldsymbol{M_2})$;
3. M_1 again responds with T, a signed message containing the current time T_1 and K;
4. M generates a sub–proof–kernel, $L = H(\boldsymbol{R}\|\boldsymbol{T}\|\boldsymbol{K})$, and sends it to the next Miner M_n;
5. The next Miner replies with U, a signed message including the current time and L;
6. These steps repeat through M, until at least three time responses, Tn, are monotonic; and
7. T_n can then be confirmed to be T_t, the correct time

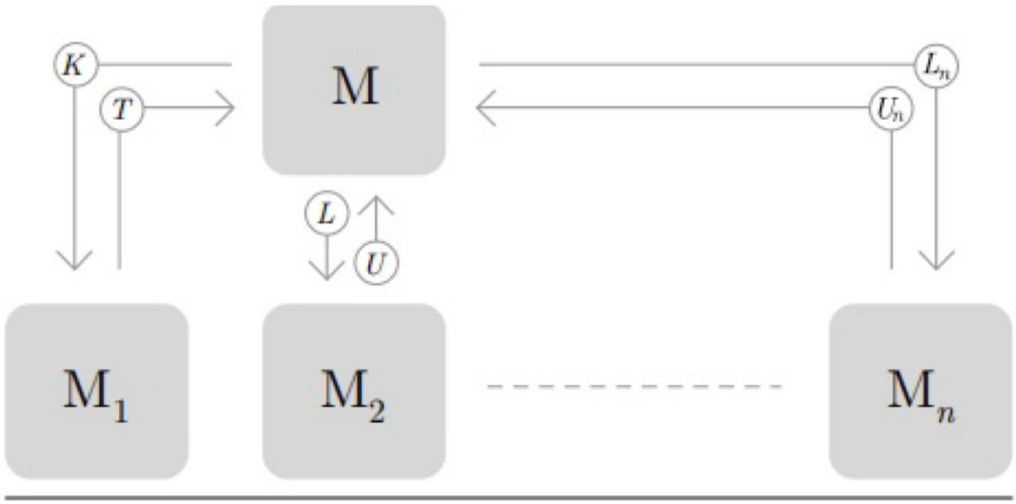


Figure 10. *Verifying* ***Proof-of-IP.Identity***

3.4.3 Utilizing the Proven Time

Once the correct time, T_t, has been determined via *Proof–of–IP.Identity*®, it is used by *M* and included during proof construction as described. The randomness, *η*, used to compute *O* and, thus, obtain the *Proof–of– Flow*® is tied to the previous block, which contains T_t. This allows us to prove, with relative certainty, that some piece of data *D* was created between the time of the previous block b_t and T_t. *D* in the case is the *Proof–of–Flow*®. Thus, we know that *D* must have been constructed between b_t and T_t. This ensures that the *Proof–of–Flow*® cannot be pre- computed.

4. *Proof-of-Territory*®

Using *Proof–of–Flow*® and *Proof–of–IP.Identity*®, we achieve cryptographic proof of Miners territory and cryptographic time consensus among Miners. We can take advantage of these proofs to determine the physical geolocation of WHIP–compatible Devices and generate a new type of proof based on the Devices geolocations. We call this *Proof–of Territory*®

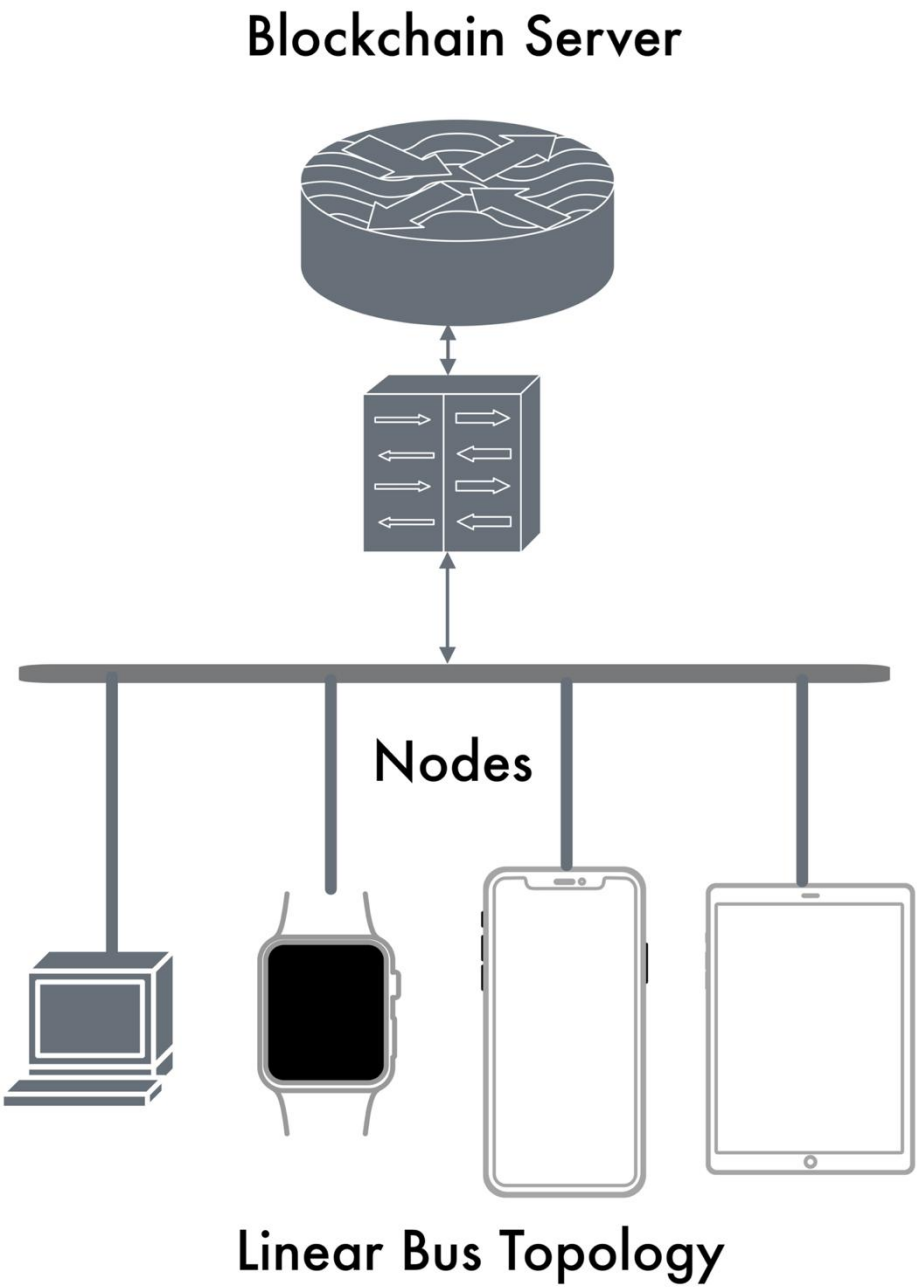
4. Motivation

Territory tracking is one of the most valuable use cases for low power Devices. It is expected that there will be at least 70 million asset tracking devices shipping by 2022.

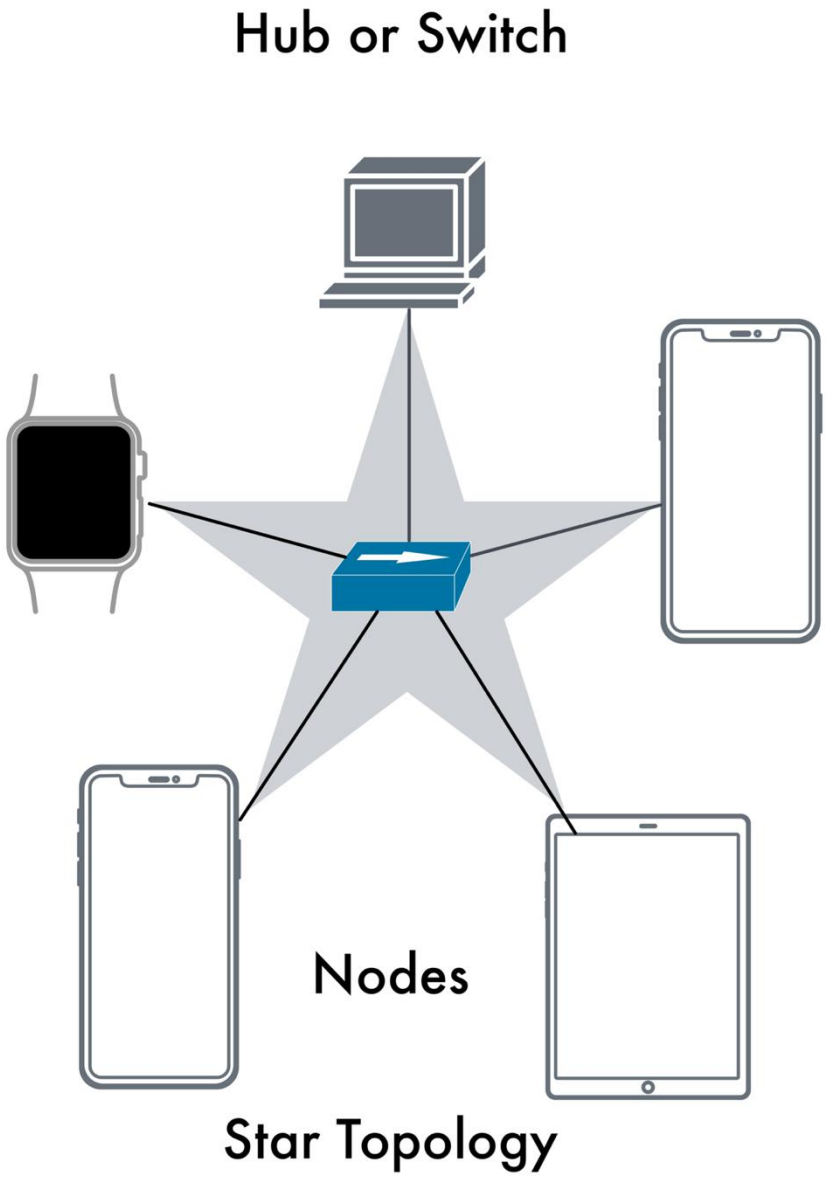
Ethernet Topology

Linear Bus The original Ethernet standard specified a linear bus. This topolgy is seldom used in new installations. A cable break on a linear bus brings down the whole network, and cabling costs can be

reduced by using twisted pair cables in a star configuration.



Star The star topology is the most common. It mitigates Ethernet distance limitations, can use inexpensive unshielded twisted pair cables, and the entire network doesn't go down if a cable breaks or is disconnected.

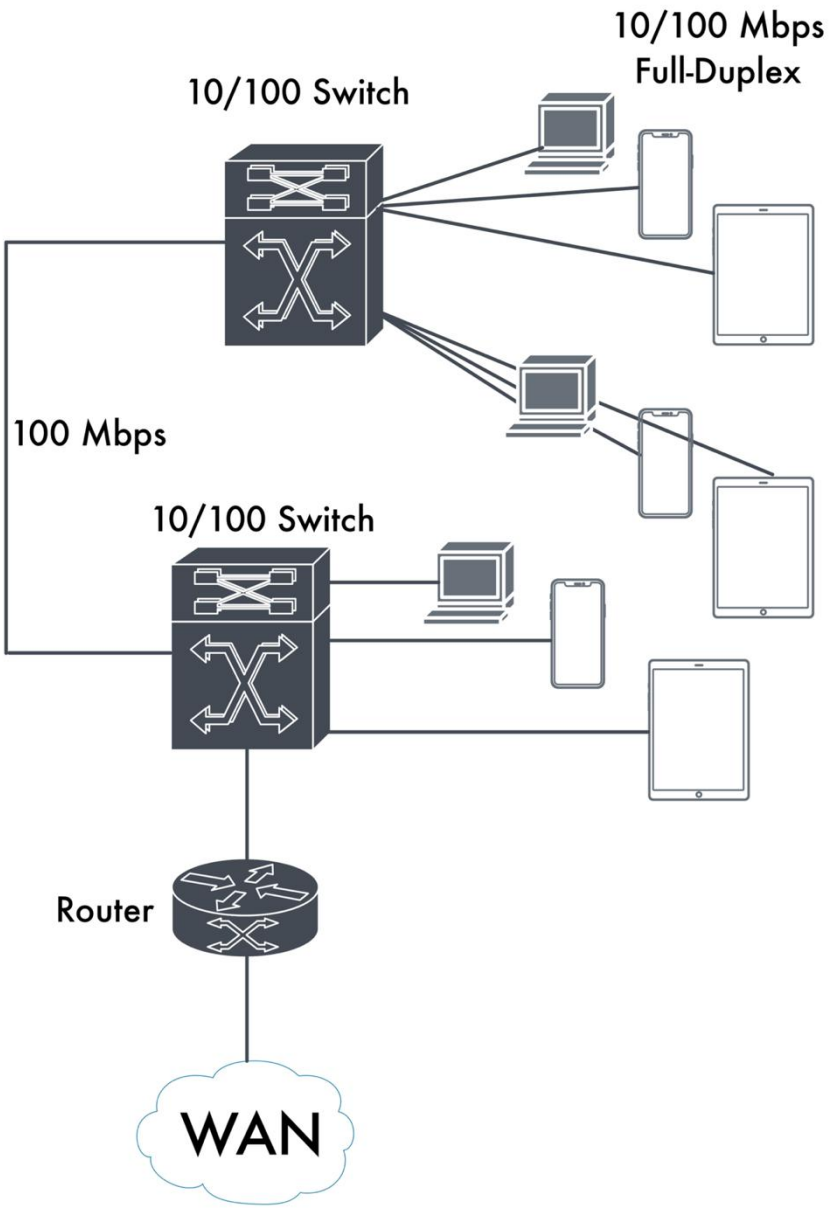


Bridges Like repeaters, bridges straddle two Ethernet segments. Unlike repeaters, they make intelligent decisions about which frames to forward and which to discard. Bridges reduce LAN traffic by dividing it into two segments. They perform a service similar to switches, though most often bridges support one network boundary only; switches support four or more segments.

Switches Though they are multi–port devices like hubs, switches are multiport bridges. Rther than broadcasting a frame out every port as hubs do, they forward the frame to its intended destination only.

This means that each port becomes a seperate collision domain. Bandwidth is shared only with stations using that port. Ports that host only a single station can be configured for full–duplex communication, which means collisions can't occur. This arrangement also means that bandwidth doubles:

1. A 10 Mbps connection provides 10 Mbps in each direction.
2. A 100 Mbps link, provides 100 Mbps in each



Ethernet Switches and Routers

We are interested in low power implementations of territory services that, in conjunction with an immutable distributed ledger, can be used to verify location and time. Given the above factors, we adopt Ethernet is an acceptable mechanism for these requirements and select Miners consensus via identifying their IP Address login instead.

4.2 Constructing *Proof-of-Territory®*

Our goal is to verify the physical geolocation of a given Device, *D*, without using GNSS hardware. To do this, we rely on the fact that we have already determined and proven the physical geolocation and cryptographic time consensus of a given Miner, *M*, using the *Proof–of Flow®* and *Proof–of–IP.Identity®* protocols described.

4.2.1 Precise timestamping of Ethernet data

There are a handful of techniques used by positioning systems with the use of Ethernet, *Time of Arrival (ToA)*, and *Time Differential of Arrival (TDoA)*. These techniques use Ethernet transmissions, usually received by one or more receivers, combined with various algorithms based on characteristics of those transmissions.

Our conclusion is that TDoA is the most accurate but challenging technique to implement. TDoA, in simple terms, relies on the variance between a transmitter and several receivers. As such, it is critical to accurately timestamp RF packets Devices emit, and synchronize the clocks of miners on the ACoin network.

An example timestamping flow is as follows:

1. A Device, *D*, broadcasts a packet *P* containing artitrary data via the ACoin network;
2. Several Miners, M_n, hear *P*, and record a timestamp T_n of their reception time of *P*;
3. T_n is created based on the nanosecond time received via GNSS and stamped using raw radio sample data received by the Spotflow® radio frontend;
4. A signed transaction including *P* and T_n are delivered to the router R belonging to *D* by M_n; and
5. *R* has now received several copies of *P*, each of which has a slightly varying value of T_n

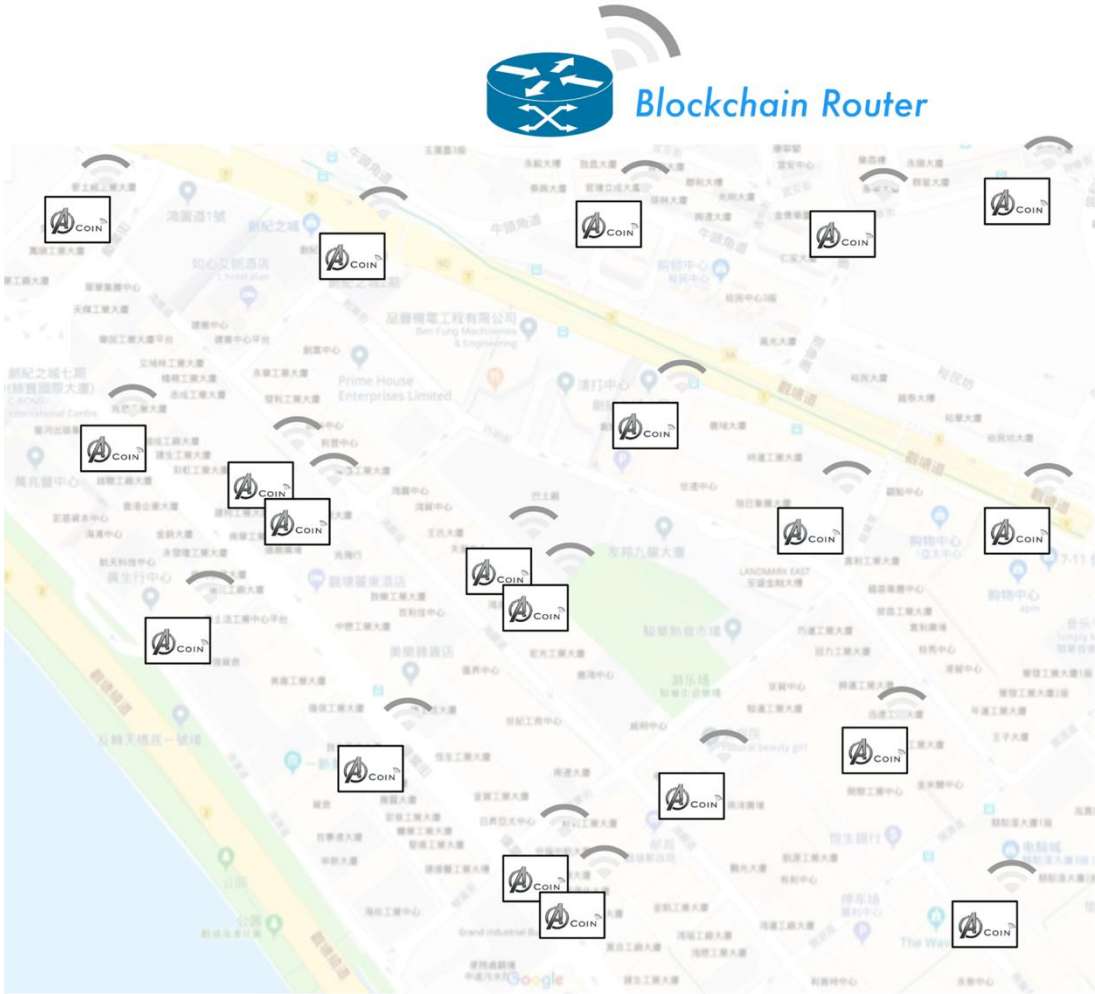


Figure11. *Geolocation via Ethernet*

Typically, it is challenging to accurately record these timestamps as any nanosecond–level variance in the timestamp can lead to significant variance in the resulting territory solution. To achieve this level of precision it is necessary to use high–bandwidth raw in–phase and quadrature (I/Q) data from the Miner's radio hardware and a fast enough processor to sample this data, identify an appropriate packet, and record the timestamp. Typically, a Field Programmable Gate Array (FPGA) is used as the processor for this data as these types of processors are able to process data in a deterministic way.

However, FPGAs are fairly expensive, power hungry, and emit significant heat. Instead, our Spotflow® mining hardware uses a novel technique using commodity low-cost components to process I/Q data and achieve timestamping at this level of precision. As a comparative example, an existing low-cost LoRaWAN access point is only capable of providing timestamp data accurate within several milliseconds of precision-as radio waves travel at the speed of light, each millisecond equates to approximately 300,000 meters of physical distance, which we deem practically useless for any accurate geolocation. Further information on the techniques, components and schematics use in out Spotflow® will be released as open source software at launch of the ACoin network.

4.2.2 Using timestamps to derive Territory

Now that the Devices Router, R, is in possession of a variety of signed messages, which include the precise timestamps, Tn, it is possible to solve for the location of the Device D. A variety of TDoA algorithms exist such as [20],[21],[19] and [22]. If a sufficient density of Mn and, therefore, Tn are recorded for a given packet, the territory of D can be derived down to a few meters depending on a variety of factors. We encourage the interested reader to read the cited papers for further details on TDoA algorithms, as they are beyond the scope of this whitepaper.

4.2.3 Verifying *Proof-of-Territory®*

Once R has computed a territory of D, it may become necessary to verify that the reported territory of D was accurate at the given moment in time. As the ***Proof-of Territory®*** is deterministic and derived from information publicly available in the Blokchain it is possible to reconstruct every step involved:
From the signatures contained within the timestamped packets, Tn, every Miner involved in providing timestamps can be verified;
By inspecting the asset_location transaction, the claimed GPS location of those Miners can be determined; and
The ***Proof-of-Flow®*** and scores for each Miner can be retrieved from the Blockchain and inspected

By auditing the above steps the router operator can cryptographically prove (or disprove) the territory of each of the Miners involved in providing the components for ***Proof-of-Territory®*** for a given Device D. The accuracy of the proof will depend heavily on the number of Mn involved and, therefore, Tn received. Additional RF factors, such as reflections and multipath, can significantly affect the accuracy of the location calculation.

5. Transactions

Transactions in the ACoin network provide functionality that enables address-to address transfers of protocol coins, similar to many existing Blockchain networks, but also provide a set of primitives that enable core functionality that is critical to the operation of a DWN. We will first address ACoin's need for microtransactions and purpose a new solution.

5.1 The ACoin Network's Need for Microtransactions

Devices Pay Per Packet The goal of the ACoin network is to offer Internet data transport fees (the fee paid by Devices to Miners) that are an order of magnitude less than anything currently available for this type of service. This transport fee would need to be metered per–packet in order to allow for maximum flexibility – this way, a Device could transact with the Miner, even just to send or receive a single packet without having previously established a relationship with the Miner.

All Transactions Occur On-Chain The ACoin network is built on the philosophy that all transactions should occur ***on-chain***; that is, blocks should be sized and mined with Ethernet such that every transaction which occur on the ACoin network should be stored in the Blockchain. To accomplish this goal, the cost of mining must be low, blocks must be large enough to encapsulate a large number of transactions, and blocks must be created frequently enough that transactions are processed quickly.

Allow Devices to Persist Data to the Blockchain Because the ACoin network services a specific use, the DWN, blocks must additionally be able store fingerprints of data sent from the Devices along with the transaction, which pays a Miner for its transport service. We believe that this holistic ***tamper-proof*** data trail will enable entirely new use cases where the authenticity and veracity of sensor data is critical.

5.2 Limitations of Existing Solutions

Now that we have discussed the requirements of transactions within the ACoin network, we outline the existing solutions for micropayments on a Blockchain and address their short comings as they apply to the ACoin network.

Heavyweight Transactions This first option is suitable only for larger transactions as the service fee is smaller than the payment. This method does not work well for very small transactions as whoever pays the transaction fee ends up potentially paying more for the transaction fees than the value being exchanged. This is similar problem to buying small–value items using credit cards today. The vendor pays a minimum fee on each credit card transaction, and under a certain charge they lose money on the transaction. These heavyweight transactions are clearly not suitable for use as a micro transaction system within the ACoin network.

Zero-fee Transactions While highly desirable from a device perspective, a true zero–fee Blockchain would be fraught with spam transactions. It would be trivial to write a script to pollute the Blockchain with transactions meant only to waste space on the Blockchain and increase congestion on the network. Some ostensibly zero–fee Blockchain implementations solve this issue in clever ways, such as offloading the work of processing and verifying transactions to the transactions themselves. However, these implementations have their own issues, for example IOTA has not yet proved it is capable of operating this type of a system without the need for a centralized coordinator.

State Channels State channels [31] allow two parties to exchange value, usually in small increments at a time, with very limited risk. If one party thinks the other is acting dishonestly, it can publish the

final transaction in the state channel to the Blockchain and close the channel.
At most one payment is usually at risk. However, there are several downsides: the payer has to lock up significant funds for the lifetime of the state channel, meaning they may be unable to open state channels with other parties or pay other dues; transactions in the state channel do not appear on the main chain at all; and these implementations are relatively complex to execute well (note that neither Lightning nor Raiden have become widely used yet).

Payment in Arrear Payment in arrear, after the services have been rendered, is an extremely risky method in a decentralized pseudo–anonymous system. There is no mechanism to gain certainty around the intent or honesty of the entities transacting, nor do you know if the entities control the requisite funds when the debt comes due. This model only works when the parties involved trust each other or have some other recourse to recover funds.

5.3 Types of Fees in ACoin

In this section we outline the types of fees needed on the ACoin network, and propose solutions that take advantage of the unique characteristics of the ACoin Consensus Protocol.

5.3.1 Transport Fees

Devices using the ACoin network to send and receive data to and from the Internet must pay Miners what is known as a ***transport fee***. This fee compensates the Miner for delivering data packets between the Device and the intended router on the Internet, and is unrelated to the ***transaction fee*** that Miners earn for mining transactions as part of blocks that are recorded to the Blockchain. The fee is negotiated between the Router to which the Device belongs, and the Miner, as Devices are not directly connected to the Blockchain.

Miners set the price they are willing to accept to transport data to and from the Internet on a per–byte basis.

A Devices router pays Miners the transport fee on transmission or reception of the data. This means that the Miner will receive the transport fee prior to the transaction being mined in a block and recorded into the Blockchain. This entails some risk for the Miner, as they must believe that the transport payment is not malicious or fraudulent prior to it being confirmed in the Blockchain. However, given how low the per–byte transport amount is likely to be, the risk seems tolerable. A Miner can blacklist a Device or organization address if they continually abuse the system.

An example transport fee process is as follows:

1. A Miner, ***M*** hears a packet, ***P***, broadcast by Device ***D***;
2. M uses the address of ***D***, attached to ***P***, to identify a router, ***R***, as the owner of ***D***;
3. M sends the signature, ***K(P)***, of ***P*** and an offer of a ***n*** coins for transport to ***R***;
4. R receives ***K(P)*** and the payment offer and determines if it accepts the packet for the offered price,
5. Assuming ***R*** accepts the packet at the offered price, it constructs a transaction ***T*** of value n payable

to ***M*** and sends it to the Miner; and
6. Once ***M*** sees the transaction in the reply it delivers ***P*** to ***R*** and submits ***T*** to the consensus group for inclusion in the ACoin network

5.3.2 Transaction Fees

Transaction fees are an essential part of most Blockchain implementations. They incentivize Miners to include a transaction in their draft block and ensure that spam transactions do not pullute the ACoin network.
To determine the appropriate fee for a new transaction, the transactor will take the medium of the past δ packet transport fees, within some margin of error. Until δ packet transports have occurred on the ACoin network, the fee will be fixed at a constant value $\mathbb{Q}$. By anchoring the transaction fee to the current fees being charged for transport on the ACoin network, we root them in reality. The ACoin network's primary purpose is to facilitate a network of wireless Internet coverage. In order to accomplish this in the long term, all of the economics of the system must align to make it practical for the primary users to transact on the ACoin network. If one set of fees were to outstrip the others, then the ACoin network would quickly lose its utility for the key user segment.

To enable Miners and other light clients to determine an appropriate fee, full nodes will expose a fee suggestion API. This way resource constrained entities that do not maintain a complete copy of the Blockchain entities that do not maintain a complete copy of the Blockchain will not need to compute the fee from the most recent transactions. During the block submission process, Miners in the consensus group will verify the correctness of the block and ensure that no fee has deviated beyond the acceptable threshold of δ.

Due to the censorship-resilience built into the *ACoin Consensus Protocol*, there is no incentive to include larger transaction fees. Unlike Bitcoin, where miners cherry-pick the transactions with the largest fees from their mempool to include in their blocks, ACoin miners cannot see the contents of the transactions without collaborating with other members of the consensus group to decrypt them. Transactions with incorrect fees (either too high or low) will be rejected prior to the block being appended to the Blockchain.

5.3.3 Staking Fees

The ***assert_location*** transaction, mentioned below has a special type of fee calculation, a ***dynamic*** fee. Because the ACoin network reaches maximum ***usefulness*** at a specific density of Spotflow®s, we want the fees to incentivize the ACoin network density to be as close to the ideal as possible. To that end, the transaction fee for asserting a territory can be thought of as the y coordinate on a curve with the formula:

$$y = (x - D)_4 + F$$

where D is the ideal Spotflow® density and F is the unit fee for a territory transaction. A sample graph of this function where $D = 3$ and $F = 1$ follows:

As can be seen, Spotflow® near the ideal network density are cheap to add, but establishing a new

network or overpopulating a network gets expensive very quickly. This serves to dis-incentivize Sporflow® deployments that are not beneficial to the network. In particular, *Alternate Reality Attacks* and warehouses full of Miners become prohibitively expensive.

Miners who have not asserted their territory, and therefore not paid the staking fee, will not be considered for inclusion in the consensus group.

Miners who move physical territory will need to assert a new territory, and pay the new staking fee.

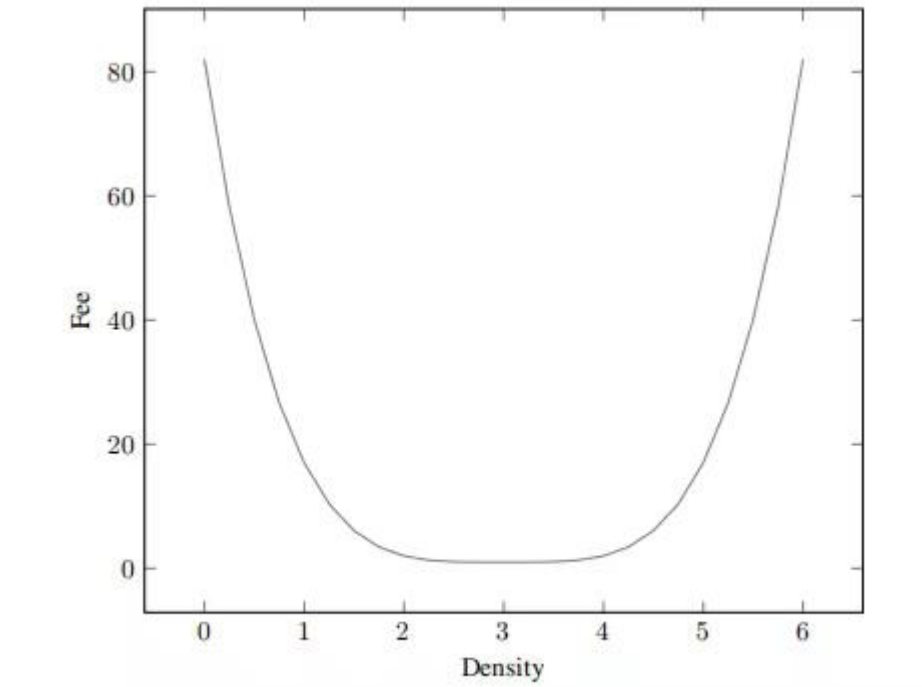


Figure 12. *Staking fee vs Miner density*

5.4 Primitives in the ACoin Network

Having discussed the philosophy of our transaction system and presented our approach to facilitating microtransactions on the ACoin network, we now delineate the transaction primitives and their properties.

add_Spotflow® Registers a new Spotflow® on the ACoin network, adding it to an existing account that will be responsible for supplying its stake (required for mining) an will receive mining rewards and fees earned by the Spotflow®.

Property	Description
Spotflow®_address	the public key address of the Spotflow® being added to the network
Owner_address signatures	the address of the owner account mutual signatures of the owner and Spotflow®

assert_territory Asserts a Spotflow® territory in the form of geographic coordinates, requiring a dynamic stake

Property	Description
Spotflow®_address	the address asserting its territory
nonce	a monotonically increasing integer
IP.address	IP. address of the Spotflow®
signature	the signature of the Spotflow®

payment Moves coins from one account, the payer, to another account, the *payee*, including the requisite fee.

Property	Description
payer_address	the address of the sender
payee_address	the address of the recipient
nonce	a monotoically increasing integer
value	an integer-based representation of the coins to send
signature	the signature of the sender

5.5 Light Clients and Full Nodes

Until now, we have discussed how to deal with microtransactions in a cost–effective way, however we have not yet addressed how to deal with the inevitable continuously increasing size of the Blockchain. One requirement for the ACoin network is that all transactions occur on–chain. This means that the size of the full Blockchain will eventually grow quite large. This is compounded by the fact that all Miners on the ACoin network are Spotflow® devices, relatively limited in computation power and storage space.

We solve this constraint by allowing mining nodes to operate as ***light clients*** on the Blockchain, pruning old blocks and transactions as needed and keeping only the latest ledger values. They will communicate over the peer–to–peer network with ***full nodes*** which maintain a complete history of the Blockchain to verify transactions.

This raises a question: who is responsible for operating full nodes, and what is their incentive to do so? Routers are software–only applications with access to scalable, cloud–based storage and will be required to operation full nodes in order to fulfill their purpose. We will operate a set of hosted routers that will make it easy for developers to launch products without needing to deploy their own router. However, many enterprise developers, who are required to maintain a higher standard of privacy, will want to host their own router. Together, these routers will form a network of full nodes capable of supporting resource constrained Spotflow®s and wallets operating light clients.

6. ACoin Consensus Protocol

Instead of an extremely computationally expensive and power hungry ***Proof-of-Work***, Miners generate

Proofs-of-Flow®. In this section we present how these useful proofs can be used to create permissionless network consensus.

6.1 Motivation

Many current generation Blockchains rely on a computationally difficult ***Proof-of-Work*** to protect the ACoin network against Sybil attacks, also known as ***Nakamoto Consensus***. The fact that the ***Proof-of-Work*** is computationally expensive to create, but cheap to verify means that in order to propose a new valid block to the ACoin network there is evidence that a significant amount of computation has been expended. Due to the fact that computation is limited by hardware cost, power cost, physical space and computational efficiency of modern technology, Sybil attacks become impossible. However, this approach, while fundamental to the mainstream adoption of Blockchain technology, has several downsides. Chief among the downsides is the power consumption; it is estimated that the Bitcoin network is consuming more power than many small countries. Bitcoin's Proof–of–Work is so wasteful it is now on the list of the top uses of electricity in the world and whenever the value of Bitcoin goes up, so do the resources devoted to mining it.

Related to the power problem is the mining pool problem. Many Blockchains have mining pools where users band together to, in parallel, mine a single block and listing the pool's address as the party to get paid. The pool then shares the block reward with the members of the pool. This ends up defeating many of the advantages of decentralization as both Bitcoin and Ethereum have come to be dominated by less than 10 mining pools each. These large pools effectively prevent independent parties from mining blocks on their own. This means that the consensus protocol for these Blockchains is effectively controlled by a very small number of mining pools and risks becoming further centralized.

More recently there has been increased momentum around making Blockchain consensus protocols less wasteful and more useful to the network. Filecoin has a Proof–Spacetime and Ethereum is moving towards a Proof–of–Stake approach.

For the ACoin network, we desire a consensus protocol with the following attributes:

Permissionless Nodes should be able to freely participate in the ACoin network without permission or approval from any other entity, as long as those nodes operate in accordance with the consensus rules.

Extremely decentralized in nature Network consensus should be designed such that there is no incentive available for taking advantage of macro–economic factors, such as cheaper access to electricity in certain geographies, and that simply buying more hardware in the same territory is either ineffective or cost prohibitive. Additionally, it should be impossible for mining pools to form and for groups to collaborate in mining blocks.

Byzantine Fault Tolerant The protocol should be tolerant of Byzantine failures such that consensus can still be reached as long as a threshold of actors are acting honestly.

Based on useful work Achieving network consensus should be ***useful*** and ***reusable*** to the network. Work performed in Nakamoto Consensus–based systems is only useful for the particular block being mined and is not otherwise useful or reusable on the network. An ideal consensus system would contain work that is both useful and reusable to the network beyond simply securing the Blockchain.

High confirmed transaction rate Our ideal consensus protocol would be able to process a very high number of transactions per second, and once a transaction is seen in a block it would be considered confirmed. Many existing Blockchains require a lengthy ***settlement time*** while the network achieves consensus which is not ideal in a system like the ACoin network, which may experience a very high number of transactions and where waiting for a transaction to settle is not tenable.

Transactions are censor-resistant Ideally, Miners would not be able to censor or otherwise pick and choose transactions prior to mining them. This would not only nullify any attempts to nefariously censor transactions, but would allow for otherwise unattractive transactions (such as fixed–fee transactions) to be included in the Blockchain.

The remainder of this section lays out our construction of a consensus protocol with these design goals in mind that we refer to as the ***ACoin Consensus Protocol***.

6.2 ACoin Consensus Protocol

We propose a unique consensus protocol around ***Proof-of-Flow®*** to capture the useful work of verifying the ACoin network as a replacement for Proof–of–Work, combined with a variant of the ***HoneyBadgerBFT (HBFT)*** asynchronous byzantine fault tolerant protocol.

6.2.1 HBFT

HBFT is an asynchronous atomic broadcast protocol designed to achieve optimal asymptotic efficiency, initially presented in 2016, In HBFT, the setting assumes a network of N designated nodes with distinct well–known identities (P_0 through P_{N-1}). In our HCP instantiation, this network of nodes is known as the consensus group C. The consensus group receives transactions as input, and its goal is to reach common agreement on an ordering of these transactions and form them into blocks to be added to the Blockchain.

The protocol proceeds in rounds, where after each round, a new batch of transactions is appended to the Blockchain. At the beginning of each round, the group chooses a subset of the transactions in its buffer and provides them as input to an instance of randomized agreement protocol. At the end of the agreement protocol, the final set of transactions for this round is chosen.

HBFT relies on a ***threshold encryption*** scheme that requires transactions be encrypted using a sharded public key, such that the consensus group must work together to decrypt it. This means that no individual node is able to decrypt or censor a particular transaction without colluding with the majority of the group.

6.2.2 Applying *Proof-of-Flow*® to HBFT

In the ACoin network, miners are required to submit ***Proofs-of-Flow***® to the ACoin network at an epoch, Δp. These proofs are submitted as a special type of transaction, and subsequently recorded to the Blockchain. Miners increase their scores as they submit valid proofs to the ACoin network. At an epoch, Δc, the highest scoring Miners, N, are elected as the new HBFT consensus group, C.

By using ***Proof-of-Flow***® to elect the members of C we are essentially substituting for well–know identities in the HBFT protocol. As we desire a permissionless network, we can use ***Proofs-of-Flow***® to determine whether Miners are acting honestly and reward the most honest Miners at a given epoch by electing them to the HBFT consensus group.

6.2.3 The Consensus group

During Δc, the currently elected consensus group is responsible for creating blocks and appending them to the Blockchain. All new transactions on the ACoin network are submitted to the current members of the consensus group. New blocks are created by C at a fixed interval Δb and recorded to the Blockchain. A Coin block reward is spilt among the members of C for every block submitted, along with thesum of all fees contained within valid transactions. In the unusual case that there are no transactions during Δb, an empty block is appended to the Blockchain.

6.2.4 The mining process

Once the consensus group C has been elected for a given Δc epoch, a distributed key generation phase occurs to bootstrap a threshold encryption key TPKE. TPKE is a cryptographic primitive that allows any party to encrypt a transaction to a master public key PK, such that C must work together to decrypt it. Once $f + 1_1$ correct members of C compute and reveal decryption shares, $\delta_{i,}$the transaction can be recovered. Once PK is generated via the TPKE. Setup function, a block containing PK is immediately submitted to the Blockchain. Each member N_m in C receives a ***secret key share***, SK_i, of PK.

Miners on the ACoin network submit new transactions t to C. Each member of C takes a random subset of the first B transactions in the queue and applies the TPKE.Enc(PK, t) $\longrightarrow e$ function and submits them to the other member of C. Once the members of C receive at least $N - f$ e they run the TPKE.DecShare (SK_j,e) $\longrightarrow \delta$ function to produce their decryption share. Members broadcast their δ to the other members of C, and once f + 1 members have seen δ shares they can proceed to the TPKE.Dec function using PK, e and the δ shares and attempt to decrypt the transaction. Each member of C appends decrypt the transaction. Each member of C appends decrypted transactions to its own instantiation of the next block kept in a local buffer. Double–spend and other malformed transactions are removed rom the blocks at this stage.

As members of the group cannot decrypt e on their own, a given member cannot censor a transaction prior to its inclusion in the candidate block without $f + 1$ members of C colluding as transactions are

received. Any honest member of C that has t in the first B of its transaction queue will eventually be able to include t in a block as the other members of C cannot decrypt the transaction until it has been agreed to, at which it is too late to censor it. As the members of C for a Δ_c epoch are selected base on their submitted *Proofs-of-Flow*®, making the members unpredictable, this type of collusion would be extremely challenging to execute.

Once $f + 1$ nodes have agreed on the transactions for the block, a TPKE threshold signature is obtained over the block. This certifies that enough nodes to exceed the byzantine fault threshold have agreed on a block. Memebers of C that are censoring or disagreeing on the contents of the block will produce an incompatible signature share that cannot be used to count towards the signature threshold. This block is then gossiped via the ACoin network to all Miners and added to the Blockchain.

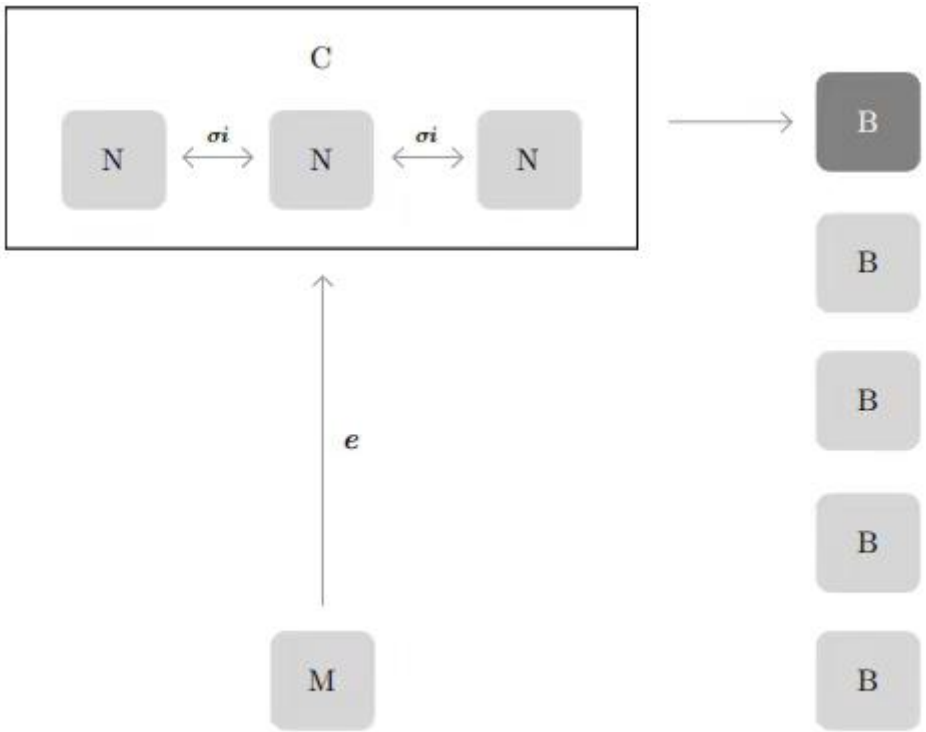


Figure 13. *The Consensus Group and Mining*

6.2.5 Conclusion

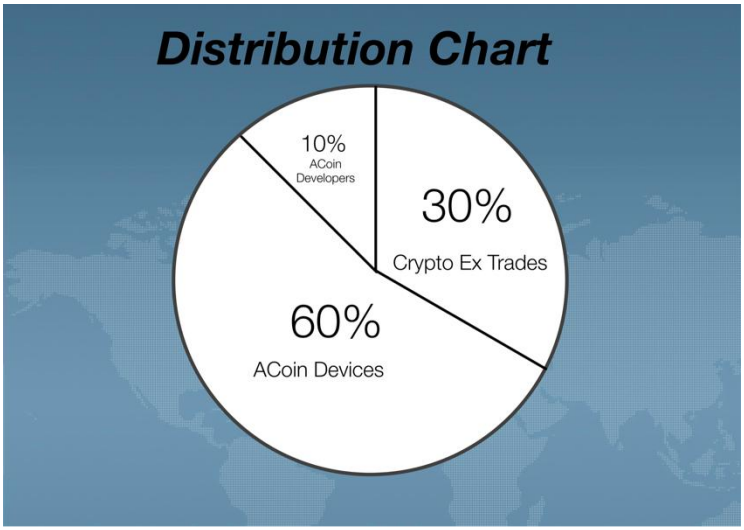
We have presented the ACoin Consensus Protocol which combines a modern, asynchronous and highly efficient byzantine fault tolerant consensus protocol with a noval mechanism for substituting permissioned identity with a useful and reusable Proof–of Flow®. The resultant protocol satisfies the design requirements of being permissionless, decentralized, byzantine fault tolerant, based on useful work, and with a very high–rate censor proof transaction mechanism.

7. REWARDS DISTRIBUTION

ACoin is created and mined by AChain as a side chain of Etherum, the Acoin Ecological Team build this Chain in order to support various Decentralised Applications that may undergo to evolute the

traditional APPs in Apps stores.

Total Mined : One Trillion



Acoin Developers will be rewarded in the 10% in the next 120 months; Each disbursement of rewards will be released in the first day of release in monthly basis.

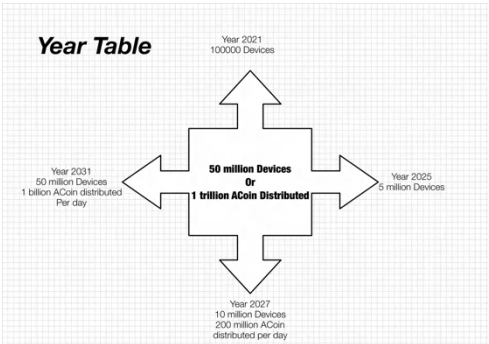
Hive devices will distribute to PoF® contributors in the following:–

1000 sets	every 1000 sets 20% up to 10000 sets	100000 sets touch base
• 10000 ACoins per day	• Max. 200%; 200000 ACoins per day	• Max. 2000%; 2000000 ACoins per day

All Distributions, activities, open source codes, contractual will be public published during set up. ACoin community will handle the developments of ACoin ecology.

7.1. Expire Countdown

The Algorithmic Devices that supports the distribution of ACoin will touches its final baseline when it reaches 50 million sold sets or 1 trillion of ACoin is totally distributed.



8. Mission

This paper presents a well thoughtful and design layout that we are contributing our know how and knowledge of Blockchain, and our believes in Blockchain is changing the world.

Almost all of the idea was inspired by Helium, we in fact modify and input more elements in our paper, instead of a singular chain for a singular mining set, we will implant multiple chains in our singular mining set. We would also like to express our deepest thanks for HIVE® and BQEX® and BCTV® for giving us all the supports that we requires.

ACOIN FULLY SUPPORTERS