Nakamoto consensus over MANET
(Gaza Strip case)

Amro Saeed

Abstract:

In this think piece we're trying to build a blockchain as an interface between theoretical computer science and human behavior exploiting the fact that humans are not infinitely rational and in particular they have computational constraints in what they are able to think (Byzantium) in a finite amount of time specially in highly hostile and rapid congested moving environment (*Lipschitz continuity)*, discovering to what extent we can reach with resembling a Nakamoto consensus within such blockchain constituents leveraging the rationality of Nash equilibrium that can be achieved in a game setting where players (population & main players) are under a severe conflict scenario, and force a some kind of consensus over a set of incentives shared among good players and the bad ones being excluded in consequence to their startegies.

We analyzed the state of art BlockDAG system design (Wayfinder){1} for its unprecedented achievement towards building Tamper-proof Provenance-Aware Storage for Mobile Ad Hoc Networks and charactrized its pros and cons with regard to use it as a medium of cryptocurrency exchange.

We also argue we can enhance Wayfinder BlokDAG with techniques used to Detect Sybil Attacks using Proofs of Work and Location in VANETs{2}, only under the umbrella of a theoretical game based scenario where the players (nodes) play anonymously and share a very hostile Geo-politically pounded environment like in the case of Gaza strip

Game setting:

Mediterranean Sea
Gaza
Dayr al Balah
1950 Armistice Line
Khān Yūnis
ISRAEL
Rafah
EGYPT
Gaza International Airport
Abū 'Awdah (Joz Abu 'Awdah)
0 3 6 km
0 3 6 mi

Place: Gaza-strip a Palestinian enclave on the eastern coast of the Mediterranean Sea.
Area: 365 Km2
Population: 2.048 million (2020)
Density: 5000 citizen/Km2 (the densest in the world)
Currency: Israeli New Shekel (ISL = 0.0000062 BTC)
Technology: Only MANET devices (WWW is not reliable)
Hostile/Conflict scenarios: Following the takeover of Gaza by Hamas, the territory has been subjected to a blockade, maintained by Israel and Egypt.

Because of the Israeli–Egyptian blockade, the population is not free to leave or enter the Gaza Strip. Israel maintains direct external control over Gaza and indirect control over life within Gaza: it controls Gaza's air and maritime space, and six of Gaza's seven land crossings. It reserves the right to enter Gaza at will with its military and maintains a no-go buffer zone within the Gaza territory. Gaza is dependent on Israel for its water, electricity, telecommunications, and other utilities

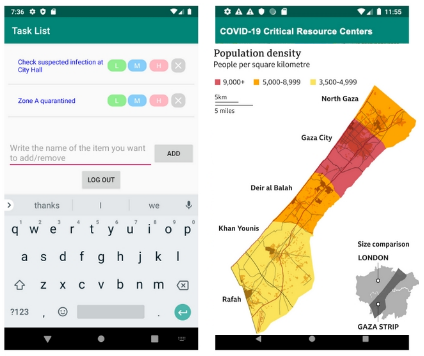
Introduction:

The concept of Nash equilibrium is paramount in Game Theory. This is because the exacting brand of rationality that it exudes, Another reason is because the existence Nash's mathematical proof for that it became the standard way of looking at economic problems

Another contender to Nash equilibrium to force players to act honestly and reach consensus on a globally distributed and persistent shared ledger without diminishing security measures from Byzantine nodes to double spending problems in a certain game setting is Nakamoto consensus but unfortunately, MANET devices (backbone of networking infrastructure in suppressed areas) are often deployed in challenging conditions and most of the time in a mobile state, and their connectivity may be intermittent (Additionally, the absence of reliable up/down link to the world wide web itself) which make it impossible to share a decentralized distributed ledger that is tamper-proof to sypil attacks, byzantine fault tolerance and preventing double spending by different bad players.

Prototype:

We developed a core concept of game, a Mixed-reality (Metaverse) type of Dapp as a frontend for users and a blockchain solution as the backend, the application prototype originally built by the wayfinder team showing two applications for emergency first response: a Task List for command and control and an Annotative Map (see Figure 2). The Task List app allows users to post and complete prioritized jobs on a shared list. The Annotative Map app allows users to place prioritized annotations on a shared map. Both applications use 2P+ sets (CRDTs, Conflict Free Data Types) and run with or without network infrastructure in a completely disconnected environment. These applications run on smartphones, servers, desktops, and potentially on IoT sensor devices.



(a) Task List (b) Annotative Map

Fig. 2: Screenshots of sample applications.

Code base:

In order to implement Wayfinder into a framework that could be installed on manet devices as well as on servers, they chose Java for their code base, though we conjecture that the best way to offload computations leveraging validity proofs (e.g. zkSTARKs) to static devices (Desktop servers) cairo language would be the best choice as a code base, their goal was to demonstrate that two concurrently running applications could rely on the same underlying WBD. (without the notion of double spending problem) But in this paper we worked on taking their further, as we noticed in Gaza-strip game setting we can besides creating a shared list for emergency response we provide a stable coin as a cryptographic secure currency and levitate the stakes in the game.

Wayfinder BlockDAG (WBD):

Wayfinder is not suitable to supporting cryptocurrencies: it can detect but not prevent double-spending. However, we believe that for many applications the CRDT consistency properties combined with tamperproofness is sufficient. While Wayfinder only provides a partial “happens-before” ordering on operations, Wayfinder can be used to hold devices and their users accountable because blocks are tamperproof. Through accountability, Wayfinder incentivizes users to behave well. Also, the properties of CRDTs prevent inconsistencies in the data structures used by applications.

Wayfinder demonstrates a design, implementation, and initial evaluation of a blockchain specifically for the low-connectivity, low-power, high data rate manet setting. The blockchain techology should be tamperproof, but it should also tolerate network partitions well and use a low-power consensus mechanism. Instead of resolving forks, it will need to permit them, resulting in a Directed Acyclic Graph (DAG) structure of the blockchain rather than a linear one. There are other blockchain designs that have embraced a DAG structure. However, they do so to increase the transaction rate, not to tolerate partitions. The cost of this partition tolerance is that the types of applications that can be implemented with the blockchain are limited to ones that only require a partial ordering of logged events

In Wayfinder, devices store state and communicate through a shared storage layer. Because the network is partitionable but consistency is still desired, Wayfinder embraces Conflict-free Replicated Data Types (CRDTs) [28]. A CRDT has the property that two states of a CRDT can be merged into a new CRDT with intuitive semantics. A good example of a CRDT is an append-only set. Elements can be added concurrently in partitions of a network, and upon reconciliation the sets can be merged by taking the union of the sets. There is an extensive variety of CRDTs defined in the literature.

Wayfinder maintains a partial ordering between operations and can define CRDTs that exploit this partial ordering. A Wayfinder CRDT (WCRDT) is an object whose state is uniquely determined by a partially ordered set of operations. Wayfinder maintains a block DAG, that is, a Directed Acyclic Graph of blocks. Each block contains a transaction, which

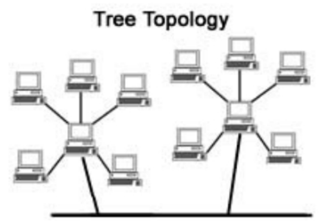
Wayfinder BlockDAG (WBD) peer review:

GAZA BlockDAG (GBD)

In our proposal we argue that by enhancing the node with proof of work sypil detecting techniques and a trajectory generation method within local network partition and among local neighboring nodes we can allow Wayfinder BlockDAG to allow forks locally but maintain a linear blockchain structure with a presistent ledger of transactions ordering resultimg in rendering it suitable for cryptocurrencies

Proposed Network Topology

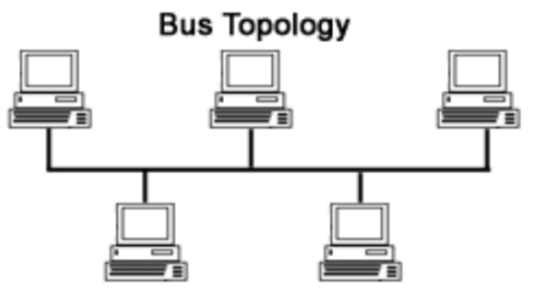
Allowing different network graph partitions separately in a Tree topology formation to fork we help devices to tolerate network partitions well and use a low-power consensus mechanism.



Limiting Branching: Ideally, the WBD would be strictly linear like a conventional blockchain, providing a total order on all operations.

The core contribution of this proposal is by using a week form of proof-of-work messaging system between every two nodes we can effectively diminish sypil attacks locally (withing neighbored nodes) and with additionally generating a trajectory proof on imaginary coordinates (that forms a map of time and speed limits) we can prevent whole network graph sypil attack

Allowing whole network graph partitions to coincide in a Tree topology formation we help devices to tolerate network partitions well and use a low-power consensus mechanism besides more importantly preventing Devices that can try to create cycles in the block DAG or add new blocks relentlessly and create a block DAG that wildly branches instead of approximating a linear blockchain formation cycles inside the acyclic graph by a whole network sypil attack (51%)



is a sequence of operations on possibly multiple WCRDTs. **The order of operations within a block and the directed edges between blocks encode the partial order between all operations.** More precisely, an operation o 1 on a WCRDT is before another operation o 2 if and only if:

o 1 and o 2 are part of the same transaction (and therefore in the same block) and o 1 comes before o 2 in the transaction; or
o 1 and o 2 are in different transactions and there is a path of the block containing o 2 to the block containing o 1 . Therefore, the Wayfinder block DAG (WBD) exactly determines the state of each WCRDT stored in it.

Wayfinder BlockDAG (WBD) peer review:

CRDTs : entanglement properties

Wayfinder only provides a partial "happens-before" ordering on operations,To this extent we intent to levitate the partial ordering of operation by exploiting our proposed network schema (Tree & Bus toplogies) to ensure blocks propagated withing Tree & Bus formations follow a determined chronological order as CRDT itself and foster its entanglement techniques to coup with network partitioning.

Another complexity is how to ensure tamperproofness: blocks should never get dropped from the WBD during reconcilation

Using both of techniques (generation of trajectories inside a Mixed-reality realm using proof-of-works) we force our **GAZA BlockDAG** disseminates blocks of CRDTs as in figure (B) "decentralized over tree & bus kind of graph formation", Instead of Wayfinder BlockDAG as in figure (C).

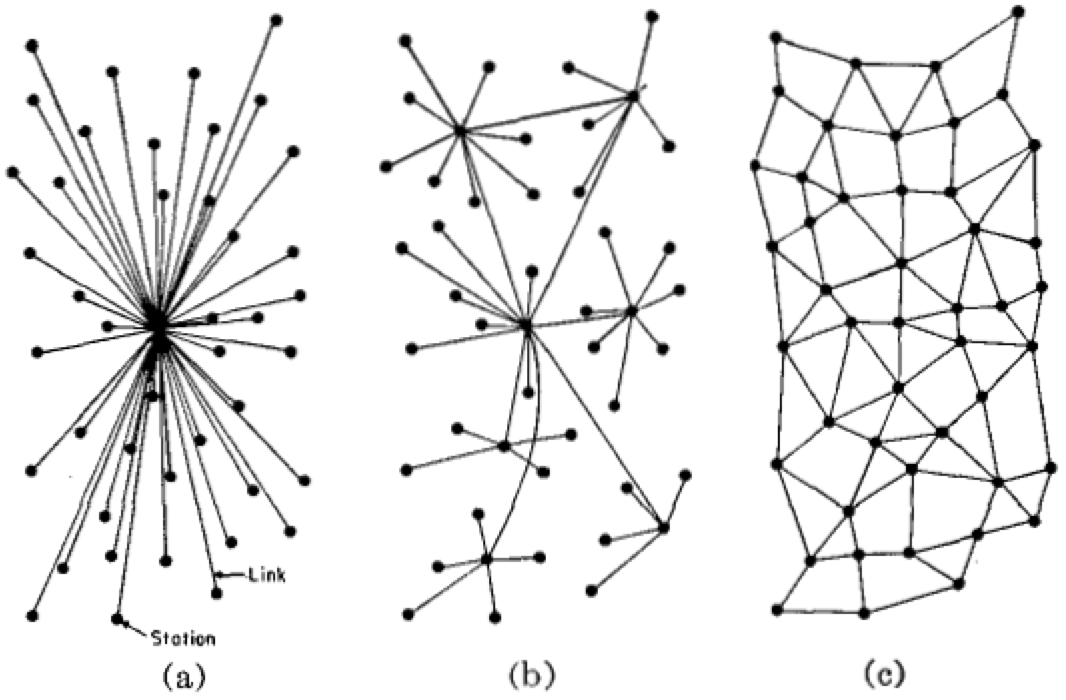


Fig. 1—(a) Centralized. (b) Decentralized. (c) Distributed networks.

GAZA BlockDAG like Wayfinder, it has three tiers of abstraction. From top to bottom, these tiers are:

1) The Application Tier provides an interface for applications to create and use smart contracts .

This layer contains a CRDT library including a new variant of 2P sets .

2) The Block DAG Tier maintains the WBD with byzantine fault tolerance.

3) The Reconciliation Tier gives an efficient reconciliation algorithm among.

A. Application Tier

CRDTs enable devices to independently update their states without any remote synchronization and guarantee consistency as long as their concurrent operations commute.

1) 2P+ Sets: A simple example of a CRDT is a 2P-Set . It is implemented by two append-only sets: an ADD set and a REMOVE set. The set difference between these two sets determines the current state of the CRDT state machine. Note that in a 2P-set, once an element is removed, it can never be added again.Leveraging causal relationships between operations, we refine the notion of a 2P set and introduce a new CRDT, a 2P+ set. When an add and delete on the same element are executed concurrently in normal 2P sets, then delete wins. However, if an add happens causally after a delete of the same element, the element is added back to the set, unlike a regular 2P set. This can be formalized as follows: with each +x (add x) and −x (remove x) operation in the causal graph of 2P+ operations (where each operation may depend on a set of other operations), we associate an add set and a delete set. They are defined as follows:

$$+x.add = \bigcup_{t \in +x.deps} t.add \cup \{x\}$$

$$+x.delete = \bigcup_{t \in +x.deps} t.delete \backslash \{x\}$$

$$-x.add = \bigcup_{t \in -x.deps} t.add \backslash \{x\}$$

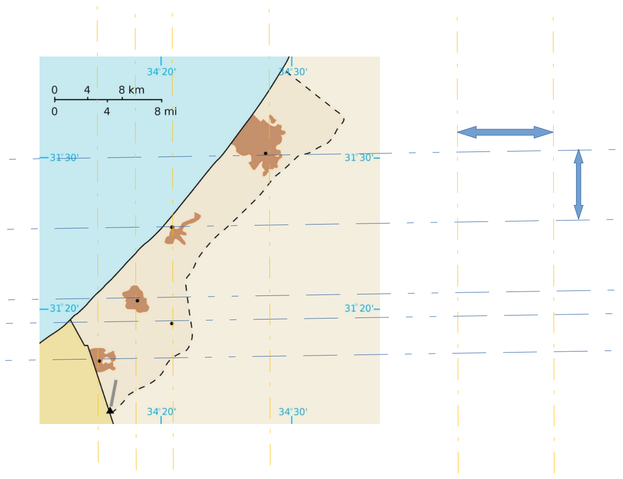
$$-x.delete = \bigcup_{t \in -x.deps} t.delete \cup \{x\}$$

The first of these specifies that the add set after some particular add(x) operation is the union of the add sets of its ancestors and x itself. The others are defined similarly. The application-visible content of the 2P+ set after an operation is then the difference between its add set and its delete set. The 2P+ can be further generalized in various ways. For example, it is trivial to define a 2P− set in which addition wins
instead of delete to resolve concurrent conflicting operations. Going further, we can define nP+ and nP− sets for values of n ≥ 2. One can think of these as prioritized sets, with n priorities. The priority of elements can be changed simply by moving them. We have found these very useful for building applications . We envision doing similar generalizations for a variety of other existing CRDT objects.

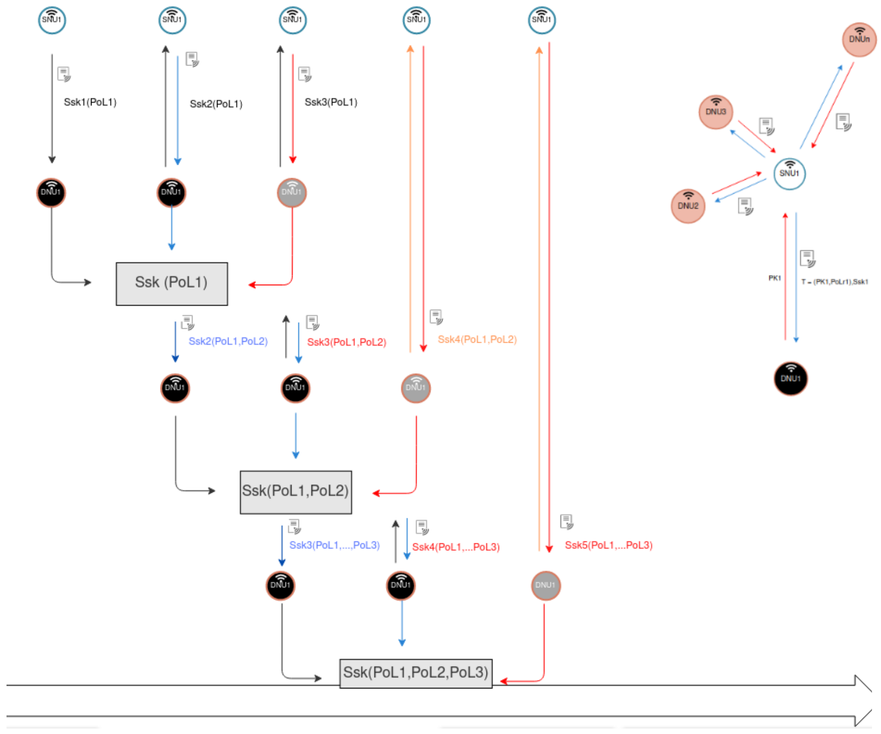
Wayfinder BlockDAG (WBD) peer review: GAZA BlockDAG Mixed-reality and TEE

2) Smart Contracts: One defining feature of blockchains is the smart contract. Wayfinder can also support a smart contract mechanism. Wayfinder smart contracts are programs that act upon WCRDTs and are themselves stored in WCRDTs in the WBD. Each smart contract tracks the state of certain other WCRDTs maintained by the WBD and takes actions such as accessing a local database on the device or operating physical actuators. A challenge is how to implement such secure interactions between the WBD and the physical environment in which it is deployed. To solve this, we plan to leverage Trusted Execution Environments (TEE). A TEE running on an manet device can provide secure sensor readings, including possibly GPS location and time.

Imaginary coordinates (that forms a map of time and speed limits)



GAZA BlockDAG node Trajectory generation



Trusted Execution Environment:

A trusted execution environment is a secure area of a main processor. It guarantees code and data loaded inside to be protected with respect to confidentiality and integrity.

Simulated temporality:

To share a case of temporality in MANET devices, we have to care for that internal hardware might run at a particular frequency (a number cycles per second, HERTZ); As software and the underlying operating system triggered according to this internal clock, In a case sensitive situation like integrating partial ordering of operations locally (neighborhood nodes – tree formation) and with additionally generating a trajectory proof on imaginary coordinates (that forms a map of time and speed limits) (whole network – bus formation) its crucial to have a restrictive form of shared temporality and also a shared spatiality between all the participants

Shared temporality:

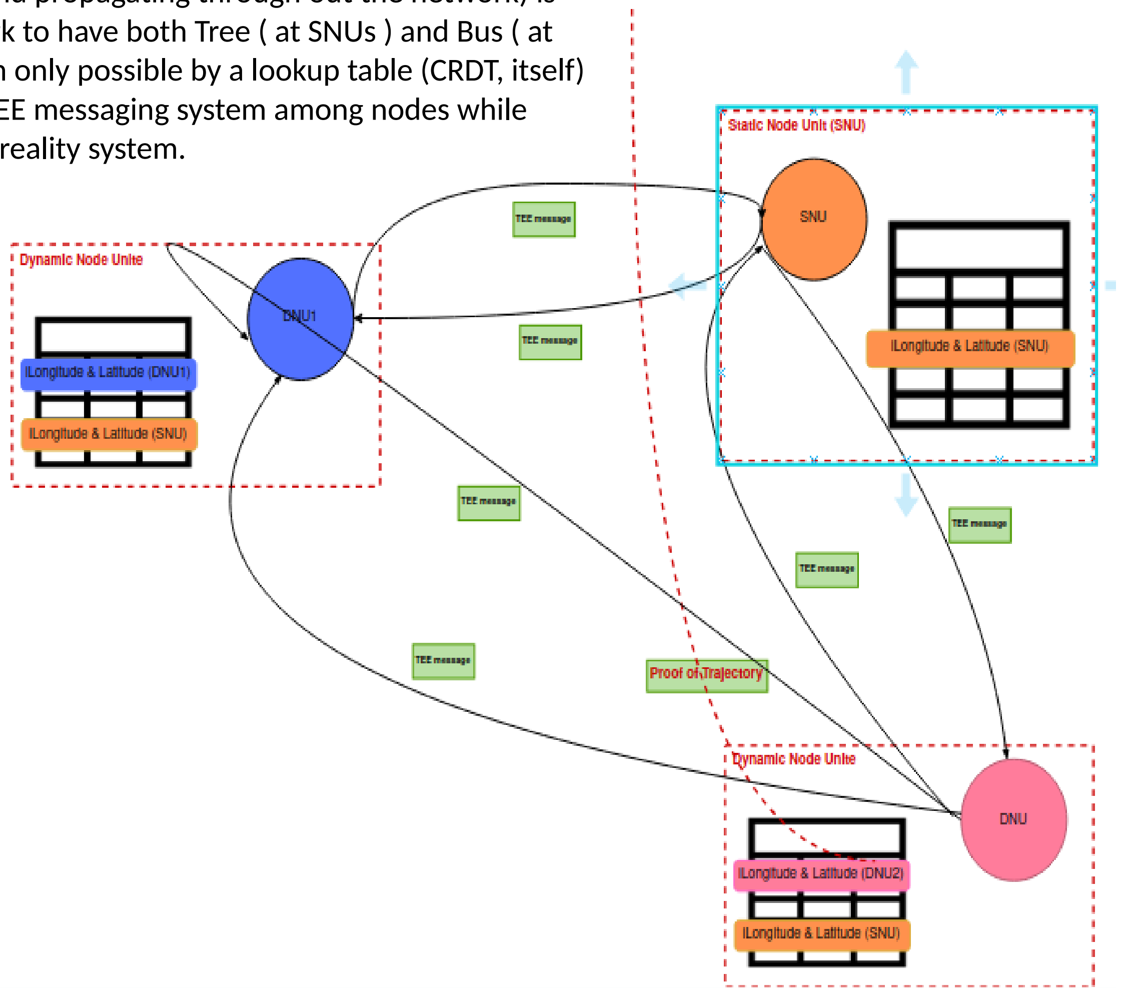
Interacting agents sharing an imaginary sense of physical laws constrained (MANETs propagation across the network graph speed and time) metaverse should cumulatively share abstraction of times, means agents are able (forced) to share simultaneously real-world time and exactly (one) virtual time, In this regard for a metaverse abstraction to be considered a real-time verse agents (nodes) must wait for other agents (nodes) to complete their actions in virtual-time (Internal clock frequency) before a consensus reached out on the transactions within.

Shared spatality:

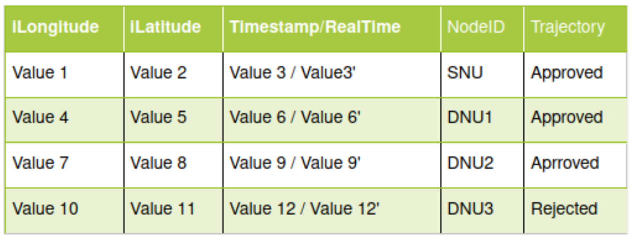
In our metaverse creation proposal for it support the network proposal it has to enjoy a fundamental physical spatial properties to support a restrictive frame of reference (time and speed of nodes propagating/ordering transaction) among meatverse agents (Nodes), such spatial properties like (containment, topology, distance, orientation and movement) without favorably depend on external source of data like GPS system (sensors are preferred), and to reach that shared spatiality nodes propagate through through seemingly continuous space in discrete steps the equivalent would be moving from pixel to pixel in the virtual frame of reference

Nodes dynamics: GAZA BlockDAG Mixed-reality

Nodes dynamics (Oscillating and propagating through out the network) is the main cause for the network to have both Tree (at SNUs) and Bus (at DNUs) and that discrimination only possible by a lookup table (CRDT, itself) set of rules on timestamped TEE messaging system among nodes while propagating within the mixed-reality system.

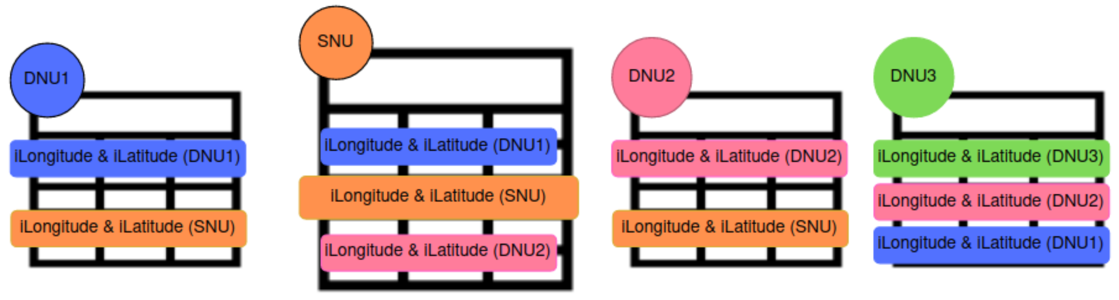


Lookup table pinged



lLongitude	lLatitude	Timestamp/RealTime	NodeID	Trajectory
Value 1	Value 2	Value 3 / Value3'	SNU	Approved
Value 4	Value 5	Value 6 / Value 6'	DNU1	Approved
Value 7	Value 8	Value 9 / Value 9'	DNU2	Aprroved
Value 10	Value 11	Value 12 / Value 12'	DNU3	Rejected

Each node at every oscillating phase where it is SNU or DNU maintain a loohup table for its CRDTs to reconcile among others and propagate the network


DNU1
lLongitude & lLatitude (DNU1)
lLongitude & lLatitude (SNU)
SNU
lLongitude & lLatitude (DNU1)
lLongitude & lLatitude (SNU)
lLongitude & lLatitude (DNU2)
DNU2
lLongitude & lLatitude (DNU2)
lLongitude & lLatitude (SNU)
DNU3
lLongitude & lLatitude (DNU3)
lLongitude & lLatitude (DNU2)
lLongitude & lLatitude (DNU1)

SNUs know by the exestence of its imagenary coordenations in >= 50% nodes in its neighborhood while its coordinates do not chnge for a time window on the other side DNUs exists in < 50% and maintains a changing coordinates

B. Block Tier

1) Ensuring an Acyclic WBD: The WBD is maintained by a collection of devices, each identified by a public key. Each block in the WBD is created and signed by a device and is uniquely identified by a cryptographic hash. Each block contains the hashes of its ancestor blocks, exactly forming the edges in the DAG. The properties of cryptographic hashes prevent cycles in the WBD. A correct device will only accept (i.e., store) a block if it has all the ancestor blocks. One or more blocks do not have ancestor blocks—it is up to each device to decide which of those blocks it will accept. Typically a device is configured with one such block, called the genesis block, and thus there is a path from each block on the device to the configured genesis block.

2) Permissioned Growth: The WBD contains an authorization mechanism. Each WCRDT defines an Access Control Matrix (ACM) that itself must be a WCRDT. Its state is uniquely determined by the partially ordered set of operationson the ACM contained in the WBD. In its simplest form, anACM is a 2P+ set that determines which devices are allowed to operate on the WCRDT and its ACM.
A device is authorized to delegate rights that it has to other devices, rather than create new rights.
There are special WCRDTs called “registries” that describes a set of WCRDTs maintained by the WBD. A registry defines operations to add and remove WCRDTs from that set and also contains an ACM specifying the devices that may execute those operations. A registry is always created in a genesis block.
In regards to WCRDT operations, an operation is permissible in a block if the author has a valid standing within the ACM and the operation is in the set of allowable operations. A block with a disallowed operation is considered invalid. Since an operation is signed by its owner, this is considered a Proof-of-Misbehavior (PoM) of that owner.
Note that while it may appear that the WBD is a permissioned block DAG, any device can create a new self-signed genesis block and participate. Two disconnected WBDs can be connected by creating transactions that depend on blocks in both WBDs. However, each individual WCRDT is protected through the relevant ACMs.

Wayfinder BlockDAG (WBD) peer review:

GAZA BlockDAG (GBD)

Wayfinder ensuring its BlockDAG acyclic nature by using properties of cryptographic hashes inside a small portion of the network graph (which make the device only honest to itself “not much of a use”) because the chain of blocks hashes are created within the device even with leveraging a TEE, a TEE have no means to outsource truth from the surronding nodes nevertheless the whole network (51%), that’s why we levarage the use of proof-of-work for trajectory generation of a node locally relative to its neighbours and forced large porion of network node to oscillates between static & dynamic phases (SNUs & DNUs) to exploit the propability for the fact that most of nodes will behave in the static mode when they are hocked to the mobile network, and for more assurance we implemented the system in a mixed reality with physical laws constrains where none of any sophisticated sypil attacks mounted by any party that is more than incitevized in such hostile environment

Wayfinder protocol tries to force permissioned growth through an Access Control Matrix (ACM) simply because maintaing a partial ordering on operations on CRDTs over large (wide) network partitions requires not only the device to controlled within (TEE) but also the WBD (that's controversially share some sort of ledger immutability “even though with small network partitions at the edges”) to have a lookup table of miss behavioral nodes (PoM)

Moreover the protocol developers tried to leverage right delegations of some nodes to other (very unlikely approach if you lookout for economic maturity of the protocol underlying “PoS type blockchains”) even they went further besides rendering the whole system as permission-ed like blockchain and after couping with all attacks that results from using a proof-of-workless BlockDAG like creating a layer of discriminately nodes “survivor list” a group of devise that proves they hold a list of tamperproof block ansectors

GAZA BlockDAG (GBD) Vs Wayfinder (WBD)

3) Limiting Branching: Ideally, the WBD would be strictly linear like a conventional blockchain, providing a total order on all operations. Given the partitionable nature of the environment, this goal is not achievable while allowing progress at all times. However, we impose various constraints to encourage the WBD to conform to a shape which is “long and thin”. In particular, for any two blocks of a correct device, there is always a path from one of the blocks to the other. In other words, a correct device does not generate concurrent transactions.

4) Tamperproving: Each device maintains an instance of a WBD and opportunistically gossips with other devices to disseminate blocks. A device will not try to reconcile with another device if it has a PoM for the other device. A correct device never drops blocks, but Byzantine devices may drop blocks from their WBDs. Also, even devices that do not drop blocks may crash or break in some unrecoverable fashion. This poses a problem, as applications may rely on blocks on the Wayfinder block graph to be tamperproof. Digital signatures prevent blocks from being modified, but not from being dropped. A block is tamperproof only if it is stored on a correct device. Note that because correct devices only store blocks if it stores all its ancestors blocks, a tamperproof block implies that its ancestors are tamperproof as well.

5) Checkpointing: So far we have assumed that all devices have unlimited storage. Besides the problem that this might be an unrealistic assumption, it also leads to potentially excessive amounts of communication needed to reconcile WBDs. To solve this problem, a WCRDT can support checkpoint operations that summarizes its state in a way that still allows partitionable operation. In particular, checkpointing preserves the ability to compute a unique state from the partially ordered operations on the WCRDT. Most WCRDTs already natively support this as they do not need to maintain the history of operations in order to compute the state. For example, in 2P+ sets, it is only necessary to store the add set and the remove set—not their histories. The problem is how to ensure that a Byzantine device does not create invalid checkpoints.

In Wayfinder protocol researchers forced to build WBD with proofs of misbehavior & a survivor list of nodes among other various constraints to force WBD to conform a “long and thin” and reach some sort of respectable consensus among nodes in such network types “MANETs” we argue that leveraging CRDTs entanglement properties allows WBD to form a “long type” shape with the addition of block time was more than enough, though imposing those types of constraints (PoM, ACM, and Survivor lists) renders Wayfinder “Very thick” to any fair/unfair economic activity to be held on top of its protocol.

An Attack vector on WBD:

A device can generate concurrent transactions, that’s simply infeasible under our setting because every transaction is bounded locally (neighborhood) by computationally expensive proof-of-work and traceable back to the signature the device used to sign these blocks after generating its trajectory locally and globally (Metaverse heuristic), so its forced to disseminate hashed blocks of tamperproofed items (TEE/Metaverse + transactions on CRDT operations)

Wayfinder BlockDAG system allows a wide range of Byzantium type of behavior (nodes have the choice to drop blocks), In our scenario this is not feasible simply because the chain of blocks consists of forced to act truthfully on blocks (hashes) from the the beginning of the economic activity whether by generating proofs-of-work, validating the existence of a trajectory inside a shared metaverse physical frame of reference, all under the umbrella of a theoretic game that by its rule allow all players to reach nash equilibrium not by controlling their strategies but by eliminating adversaries chances to tamper the system (Gaza-Strip special case) and to this extent nodes has to propagate truthful blocks upon communication not only within the edges of the graph but also through out the network congested core rendering tamperproofness of the block (with the assumption that all devices have unlimited storage) and here comes Checkpointing.

Checkpointing:

The concept of checkpointing originally used by Satoshi Nakamoto for providing Bitcoin protocol, Checkpoints are when block hash values up to a specific point in time are hard coded into the official Bitcoin client. The client takes all transactions confirmed up to the checkpoint as irreversible.

Fortunately Most WCRDTs already natively support this as they do not need to maintain the history of operations in order to compute the state.

On the contrary to Wayfinder nodes in our scheme cannot create invalid checkpoints as the CRDT client library it self already concensed upon propagation

6) Rate Limiting: A Byzantine device can try to create WCRDTs and/or operations on those WCRDTs at an unlimited rate. Even if we limited the rate at which a device can add operations to the WBD, it could authorize a never ending stream of new devices to add more operations (aka a Sybil attack).
Taking advantage of the fact that applications usually create transactions at a known maximum rate, we employ the following solution currently: each WCRDT specifies at what rate each device is allowed to add operations to the WBD. When a device authorizes another device, it must split its rate with the new device. Therefore, the two devices together cannot increase the rate at which the WBD grows.
To support this, we assume that devices have clocks that are loosely synchronized with real time. Each block has a timestamp. If b 1 is an ancestor block of b 2 , then the timestamp on b 1 must be before that of b 2 . Also, correct devices only consider blocks that have timestamps before the time on their clocks. Each device can compute the rate at which other devices are adding operations to a WCRDT and can use this to detect devices exceeding their rates. Such detections can be used as PoMs as well because they can be verified by any device.

GAZA BlockDAG (GBD)

The approach Wayfinder take to rate limiting a byzantine device to create CRDTs or operations on those CRDTs is anti-innovative as it poses limits on applications developers to prop the chain for with requests for responces as many today applications core developers try to index their chains effectively and with ease of use paving the road for application side developers to build intuitive applications for end users, a bizarre limitation for a supposedly smart contract ecosystem

C. Reconciliation Tier

Wayfinder’s Reconciliation Tier is responsible for reconciling block DAGs and witness sets between devices. In a mobile ad hoc network, reconciliation is initiated every time two devices are within each other’s communication radius. This is different from a static network infrastructure where devices gossip, or periodically synchronize updates, with one another. As the period when two devices are within communication range may be limited, reconciliation must be fast. In the following section, we discuss approaches we took to address this challenge.

1) Reconciling WBDs: One approach to reconcile the WBDs of two devices is to exchange all blocks. This Send All Blocks Protocol (SABP) can be improved by devices also exchanging acknowledgments to prevent sending blocks that have already been acknowledged. While simple and reliable, it is inefficient as devices are generally expected to have many of the same blocks already and thus the protocol may end up sending blocks that the peer device already stores.

GAZA BlockDAG (GBD)

Firstly: In our proposed scheme a we do mot suffer from a witnesses set to reconcile on, we only reconcile between ledgers of transactions across the network

Wayfinder implementation currently uses two types of communication protocols Hashed Vector Timestamp but reverts to FSRP if the protocol fails to reconcile on its Own.

Wayfinder paper looked at different settings for communication protocols to reconcile WBDs like Send All Blocks Protocol (SABP) but cant exploit its simplicity and reliablity as a consequences of using Acyclic graph that encourages forks instead of resolving them, on the contrary in our scenario we limit forks for the beginning on many levels even nodes cannot drop blocks and cannot create invalid checkpoints whether they chose to participate in the network game or not .

Thats why in our solution we tend to use SABP as our networking protocol and enjoy the ease of just allowing network nodes to ping each other for reconcilation on the CRDTs, and avoiding FSRP compared to Wayfinder case as it is computationally expensive and slow FSRP is slow because it goes through phases rather than streaming blocks one after another, and in the case of using Hashed Vector Timestamp (HVTP) its not reliable, it may fails if devices detects that they have different Hashed Vector Timestamps (also its unlikley in our case compared to Wayfinder but it remains a waste of computational resources.

Techniques used to Detect Sybil Attacks using Proofs of Work and Location in VANETs:

we studied several techniques for preventing sybil attacks in VANETs and they formulate their solutions in two steps:

1. Trajectory Generation
2. Sybil attack detection

Their network model states as follows: (But is not achievable in our state of infrastructure)

RSU: (Road Side Unit)

- Provides wireless access to users within its coverage
- RSUs are interconnected (RSU backbone network). Not available by any means in hostile environment and not even favorable as it resembles a single point of failure

OBU: (On Board Unit)

- Can communicate with RSUs and other vehicles via wireless connections.

Off-line Trust authority: (Can not exist in such hostile environment else it be considered as a single point of failure)

- Responsible for system initialization
- Connected to RSU backbone network
- Does NOT serve vehicles for any certification purpose

Our Network Model (Nodes): GAZA BlockDAG (GBD)

Leveraging TEE each MANET device oscillates between two states whether its static within the network partition or its dynamic out of neighbor coverage and into others

static node unit (SNU):

Provides wireless access to users within its coverage

dynamic node unit (DNU):

On continuous travel route through out the network graph a different but yet pounded speeds.

Definition of Trajectory:

A MANET device anonymously authenticates itself using its trajectory.

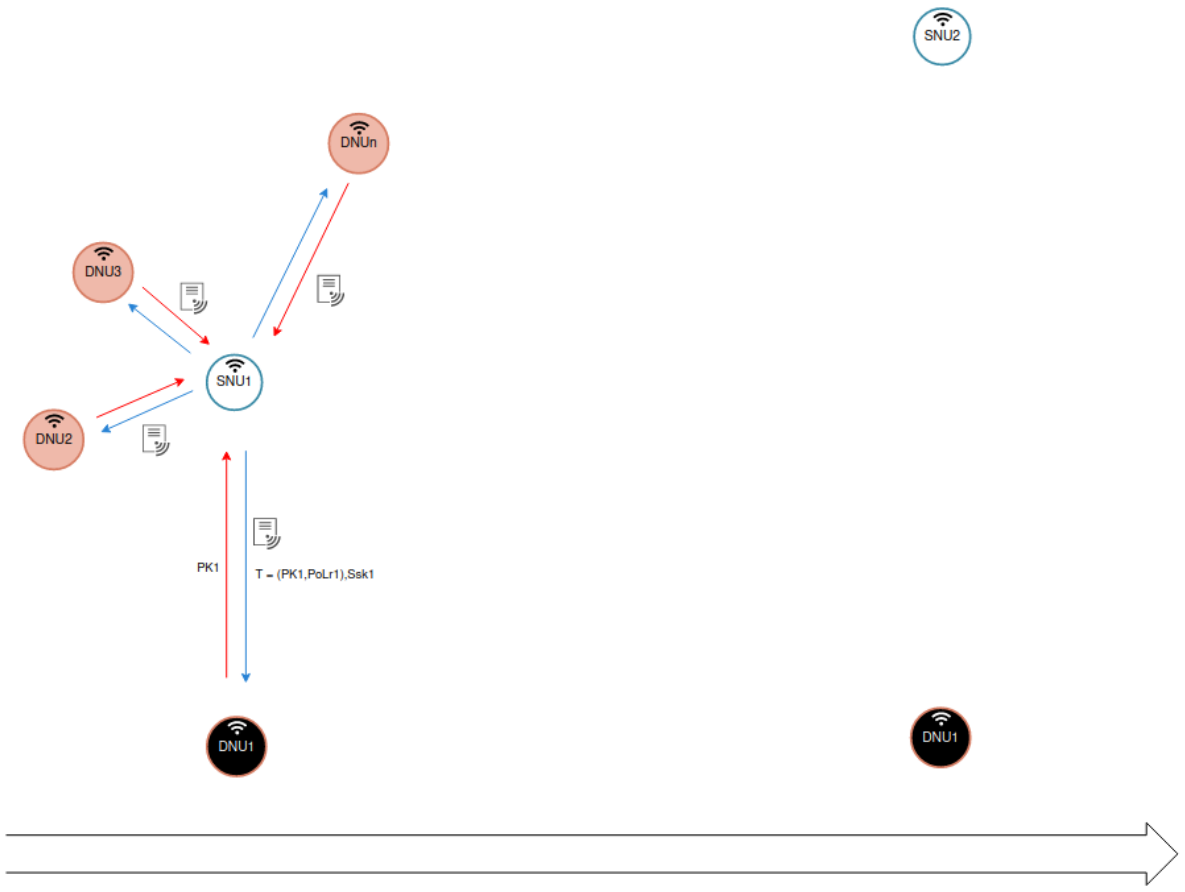
- When passing by a SNU, a DNU obtains an authorized message as proof of presence at particular SNU at a given time.

- A set of consecutive authorized messages form a trajectory

Obtaining Authorized Timestamped Messages:

1. DNU1 generates Private/Public Key pair SK/PK
2. DNU1 requests an authorized message by submitting PK
3. SNU1 generates Proof-of-Location: PoL
4. SNU1 signs on PK, PoL : $S(PK, PoL)$
5. SNU1 issues authorized message T

$T = PK, PoL, S$ to DNU1



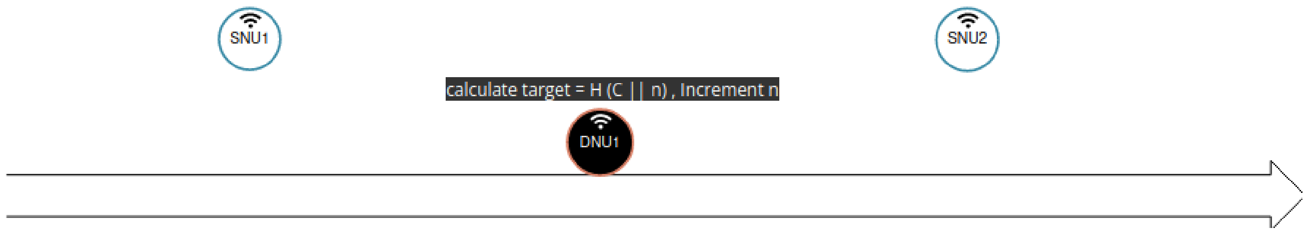
Proof-of-Work

Goal: Prevent DNUs from creating multiple trajectories at a time and disseminates concurrent blocks across the network.

DNU:

Upon receiving of T
- generate challenge $C = H(T)$
- start running the PoW algorithm:
- Calculate target = $H(C \,||\, n)$ Increment n
while increment n.
- Keep the lowest target.

The longer it takes DNU to traverse from SNU to SNU , the lower the value of target should become due to the probabilistic behavior.



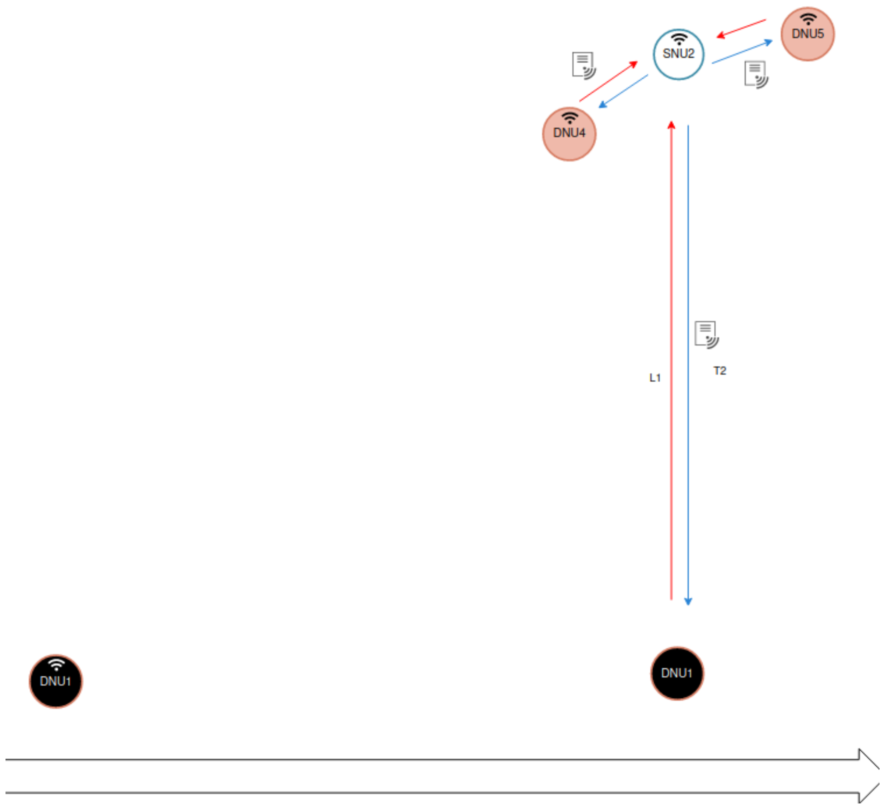
Proof-of-Work Verification

DNU:

1. generates Private/Public Key pair $SK2$ /$PK2$

2. signs on previously obtained authorized message $T1$, $PK2$, target, n:
Ssk1(T1,PK2,target,n)

3. requests a new authorized message by submitting $L1$ = ($T1$, PK 2, target, n , S), Ssk1



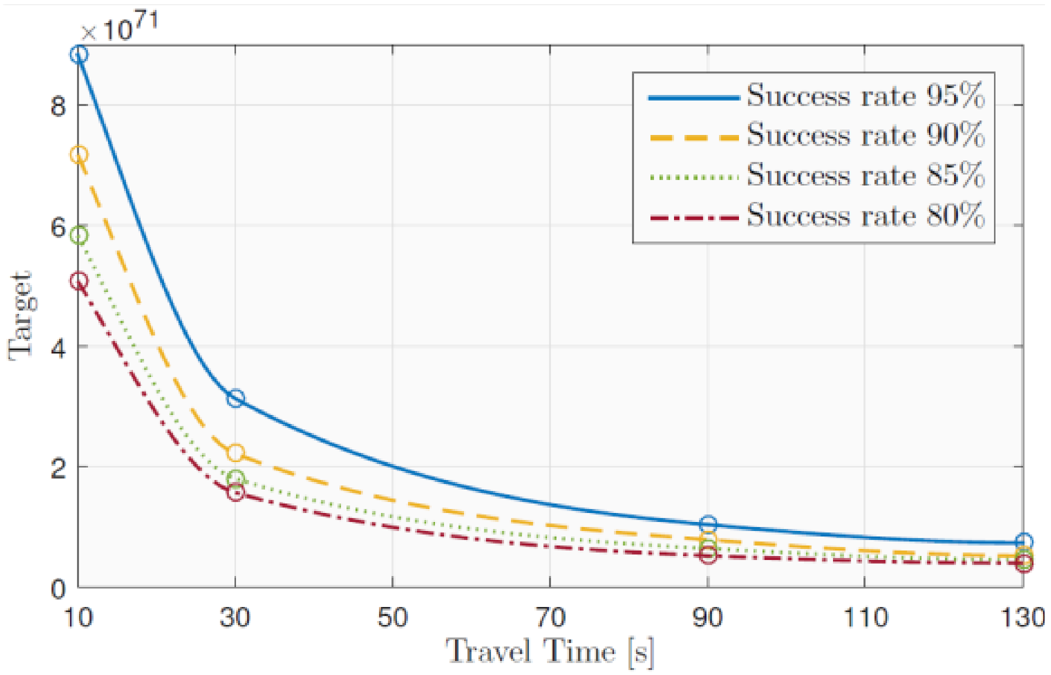
Proof-of-Work Verification

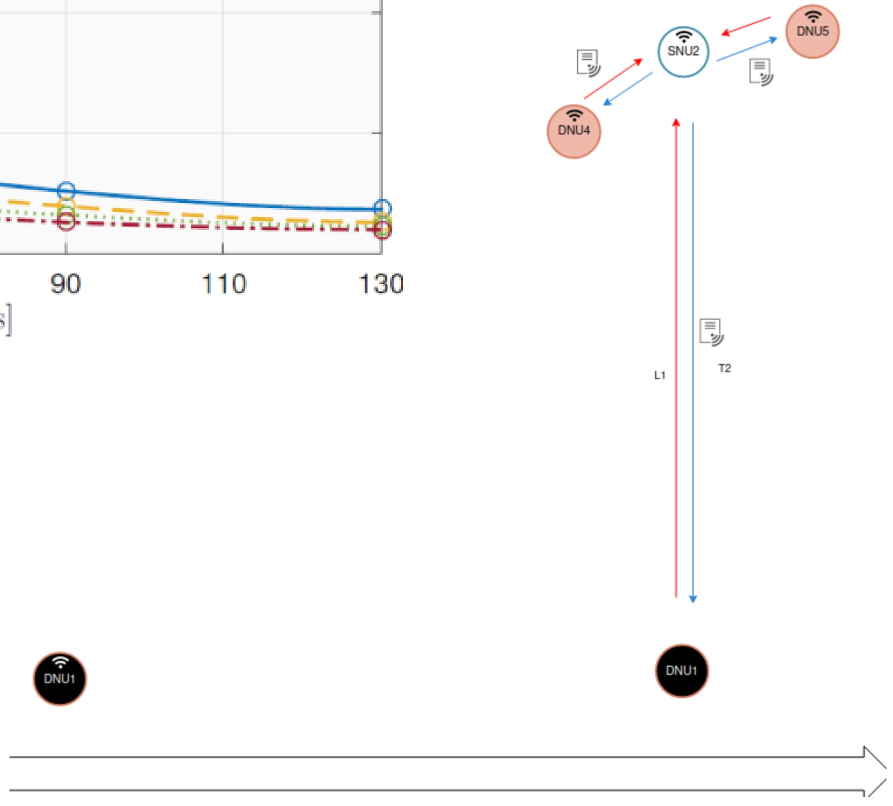
SNU:

1. Verify if H (n||c) = target

2. Determine travel time using ts of PoL1 : T = tc - ts

3. Look up expected target and check if target <= target lookup





Message Verification

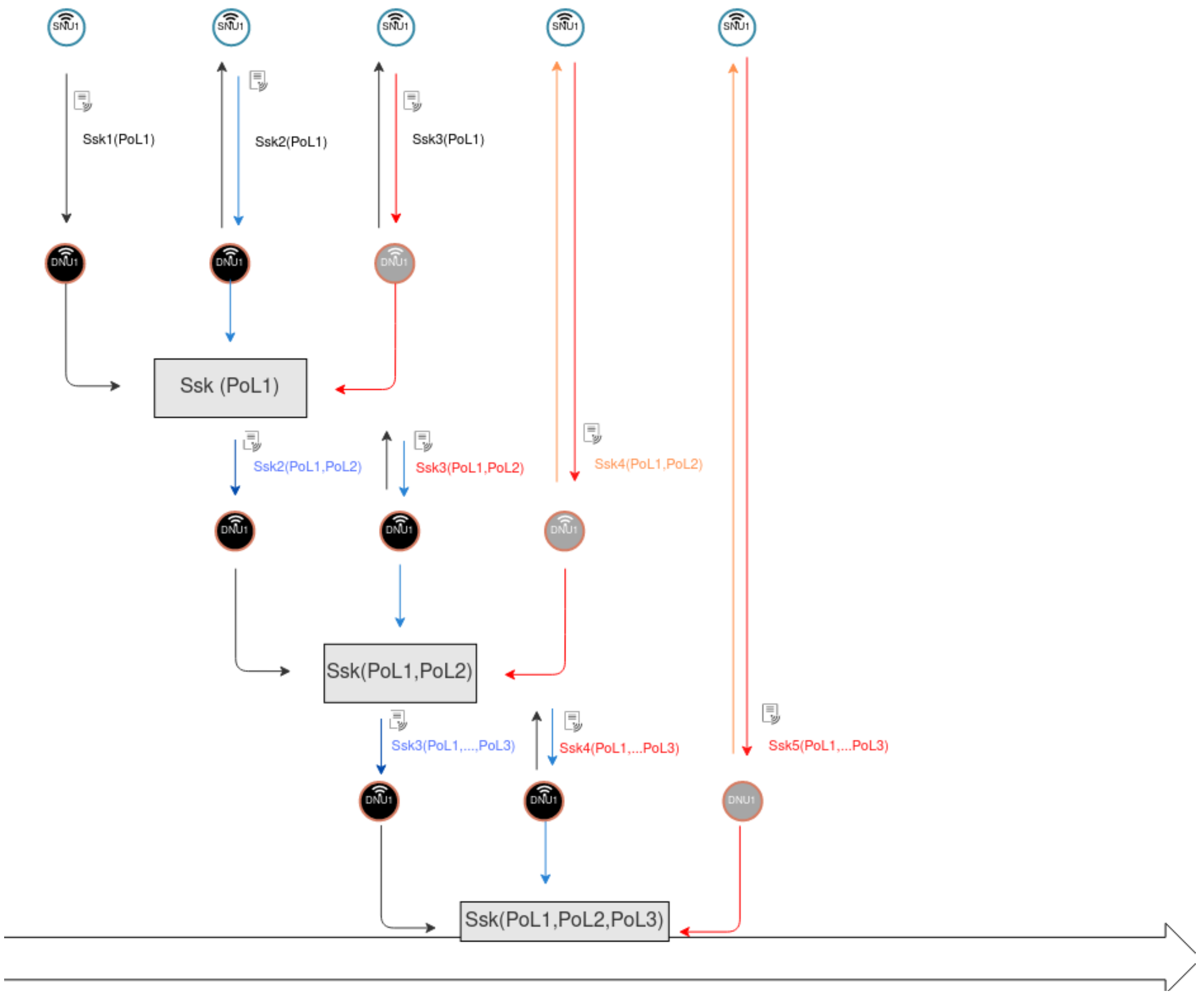
DNU:

1. generates Private/Public Key pair SK/PK
2. sign on previously obtained authorized message T1, PK2, target, and n: Ssk1(T1, PK2, target, n)
3. requests a new authorized message by submitting L1 = (T1, PK2, target, n), Ssk1

SNU:

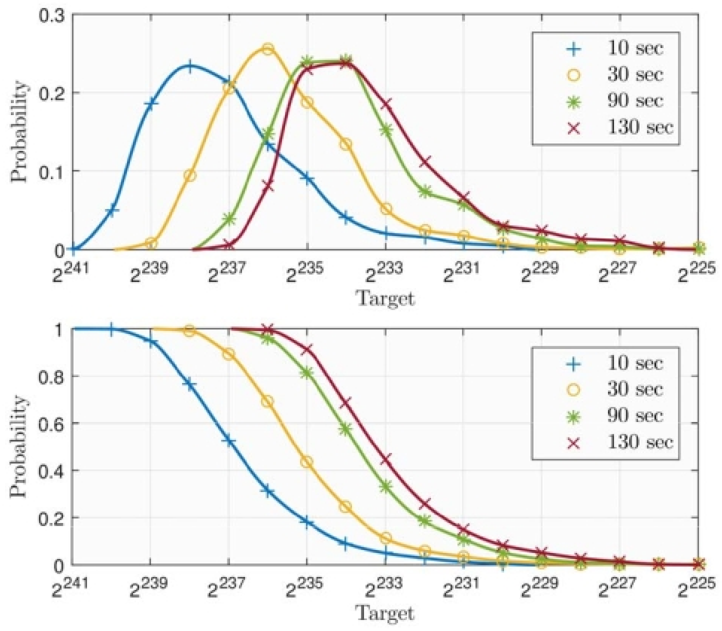
1. verifies proof of work
2. verifies Sskv1 and Ssk1
3. generates Proof-of-Location: Polr2 = ts, tagr2
4.signs on (PK2, PoLr1, PoLr2) : Ssk2 (PK2, PoLr1, PoLr2)
5. issues authorized message T2 = (PK2, PoLr1, PoLr2, Ssk2) to DNU

Collaborative Trajectory Generation:



Selection of PoW Targets and mapping it into lookup table

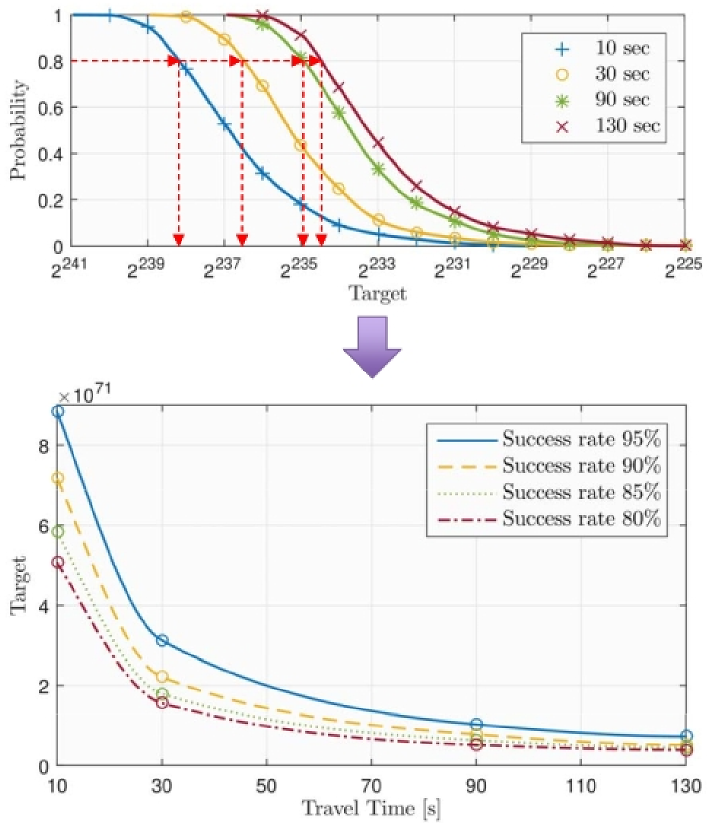
1. Run PoW algorithm for constant times to obtain probability distributions.



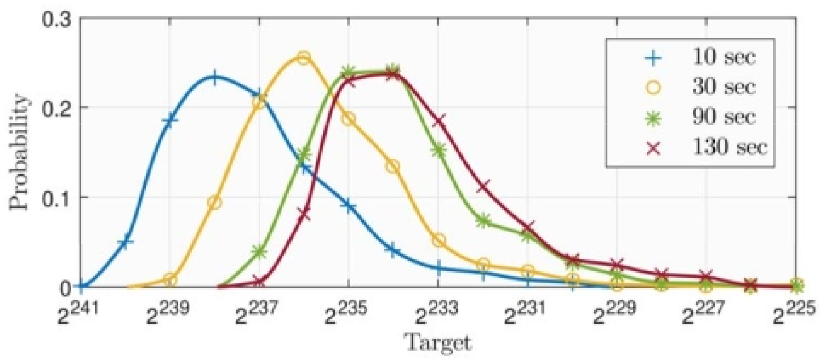
<u>Experiment Setup:</u>

- Raspberry Pi 3
  (1.2 GHz processor, 1 GB RAM)
- Travel times:
  10 sec, 30 sec, 90 sec, 130 sec
- Number of samples:
  1000 per travel time

2. Map data into Target Lookup Table



3. Selection of PoW Targets by the algorithm

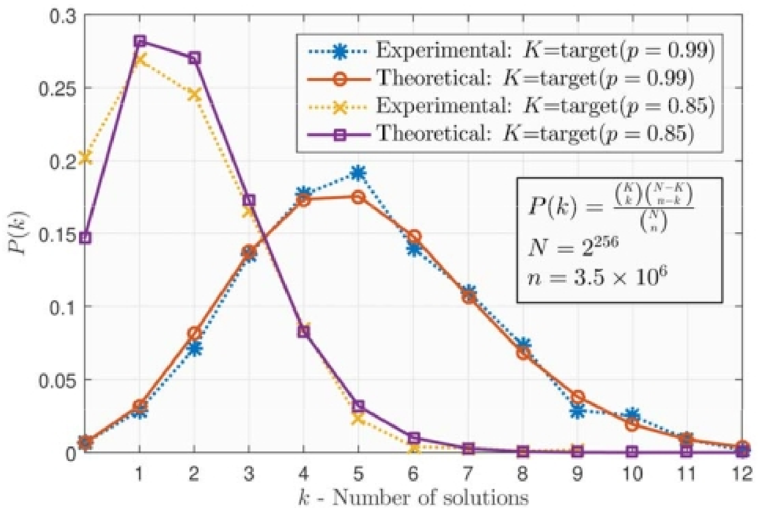


Mathematical Model:

Hypergeometric Distribution: $P(k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}$

$N = 2^{256}$; Output range of SHA-256
K = target ; Target at given probability
n = Number of hashes per travel time on RPi 3
$k = 1$; Number of solutions

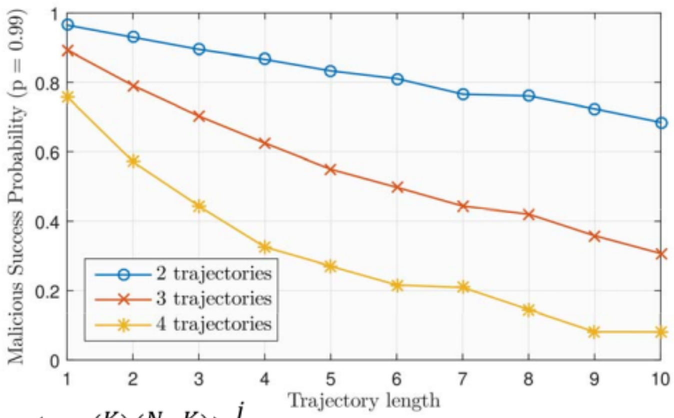
Run PoW algorithm for constant time and obtain the number of solutions found for defined target values



Hypergeometric Distribution: $P(k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}$

$N = 2^{256}$; Output range of SHA-256
K = target ; Target at given probability
$n = 3.5 \times 10^6$; Number of hashes per 90 sec. on RPi 3
k = Number of solutions

4. Probability of generating k trajectories of length j



Experiment Setup:

- Raspberry Pi 3
  (1.2 GHz processor, 1 GB RAM)
- Travel time between two RSUs:
  90 sec.
- Trajectory length: 1…10
- Number of simultaneous trajectories:
  2, 3, 4
- Number of samples:
  1000 per target

$$p = \left(\sum_k^0 \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}\right)^j$$

$N = 2^{256}$; Output range of SHA-256
$K = 16.74 \times 10^{70}$; Target at given probability
$n = 3.5 \times 10^6$; Number of hashes per 90 sec. on RPi 3
k = Number of solutions
j = Trajectory Length

Sybil attack detection

During a conversation (initialized by a DNU or an SNU):

1. Participating nodes should provide their trajectories timestamped for verification
2. The conversation holder verifies each trajectory
3. The conversation holder conducts online Sybil attack detection
4. Proceeding with the conversation

Trajectory T of DNU\SNU v:

Sskn (PKn, PoL1, PoL1, PoL2, PoL3, …., PoLn)

Tv = PKN, PoL1, PoL2, PoL3, ...PoLN, Sskn

Sybil attack detection

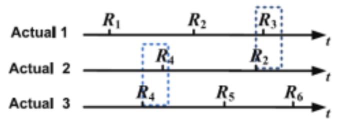
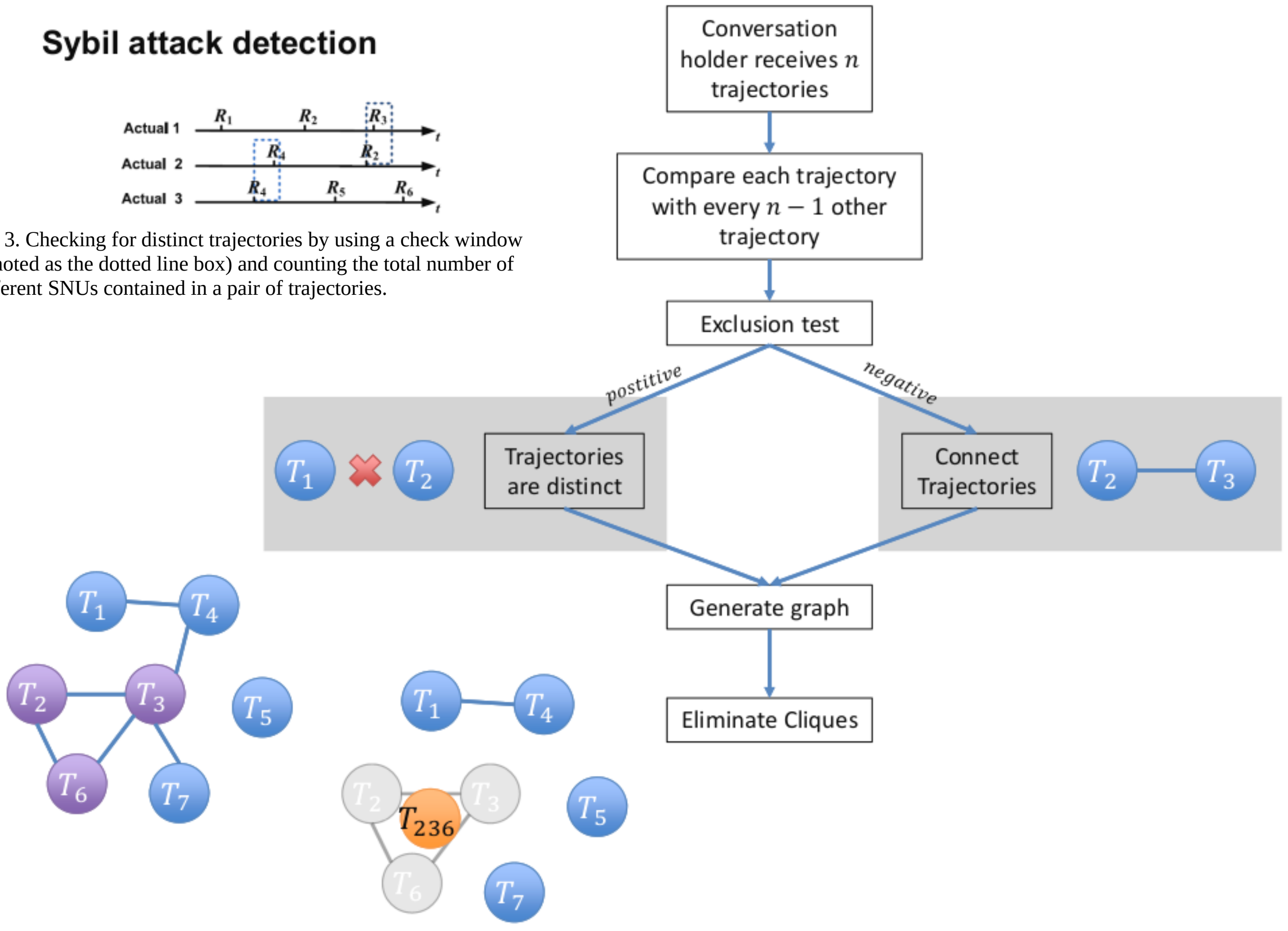


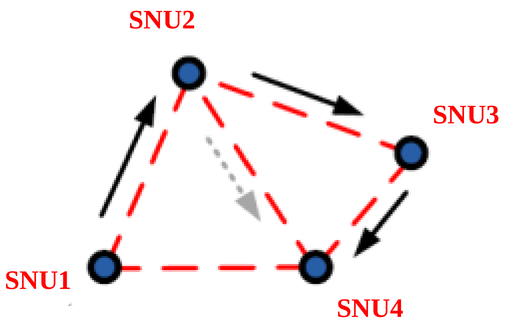
Fig. 3. Checking for distinct trajectories by using a check window (denoted as the dotted line box) and counting the total number of Different SNUs contained in a pair of trajectories.



Sybil attack detection:

Features of forged trajectories:

1. A forged trajectory is a proper subset of the actual trajectory
2. Any two forged trajectories cannot have two distinct SNUs at the same time (otherwise the malicious DNU would appear at two locations at the same time)



Features of actual trajectories:

1. It is very hard (if not impossible) for a single vehicle to traverse between a pair of RSU's
shorter than a time limit
=> traverse time limit: the shortest time for a vehicle to travel between any pair of RSUs in the system
2. Within a limited time period, the total number of RSUs traversed by a single vehicle is less than a limit
=> trajectory length limit: the maximum number of RSUs involved in a trajectory within an event
Both limits can be measured based on the distance and speed limitations of each road segment and the layout of RSU deployment

Disclaimer:

All experimental results are held by the original papers under review writers

References:

1. Tamperproof Provenance-Aware Storage for Mobile Ad Hoc Networks by,
Danny Adams, Gloire Rubambiza, Pablo Fiori, Xinwen Wang, Hakim Weatherspoon, Robbert van Renesse, Department of Computer Science, Cornell University.

2. Detecting Sybil Attacks using Proofs of Work and Location in VANETs by,
Mohamed Baza ∗ , Mahmoud Nabil ∗ , Niclas Bewermeier ∗ , Kemal Fidan † ,
Mohamed Mahmoud ∗ , Mohamed Abdallah ‡
Department of Electrical and Computer Engineering, Tennessee Tech University, Cookeville, TN, USA
Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN, USA
Division of Information and Computing Technology, College of Science and Engineering, HBKU, Doha, Qata

3. Approximate Nash Equilibria in Anonymous Games by,
Constantinos Daskalakis ∗
EECS, MIT
costis@mit.edu
Christos H. Papadimitriou †
EECS, UC Berkeley
christos@cs.berkeley.edu