Encoding SPAN Evolution Using Frequent-Collision Blockchains

Ryan Robinett and Tiago Royer

11 Dec 2019

Abstract

Blockchains are an implementation of a public distributed ledger, which allows for information to be agreed upon by several parties without the need of a central authority. When transposing the concept of blockchains to smartphone ad-hoc networks (SPANs), assumptions like relative stability of the underlying network are violated, which inevitably forces the blockchain to fork. In this context, instead of avoiding the creation of forks, we propose leveraging these long-term global forks as a way to record information about the topology of the undelying SPAN. We present a simulation package for analyzing the forking behavior, alongside with a protocol for performing local geographical authentication based on this forking behavior.

1 Motivation

Geographical authentication is the task of certify that someone is at a certain place; that is, the task of authenticating someone's geographical location. This problem has been considered for over 20 years [5]. The authors of [5] consider the problem of finding the location of an intruder; other applications include geographically restricted broadcasts [9], checkins at Foursquare, and withdrawing cash at an ATM.

In all these examples, there is a central authority which confers authentication services. Besides all the traditional problems with centralized solutions, like having a single point of failure, services like Foursquare have to essentially believe the user's word when attesting they were at a certain place [7]. ATMs circumvent this by essentially being a large network of totems, which is expensive [3].

The above solutions suffer from either having to trust too heavily upon the user's word, or from implementing an expensive network of totems to vet user dishonesty. The former makes a system unreliable, while the latter imposes a startup cost that is not realistic for small players trying to enter the market.

For reasons of decentralization and low cost, we would like a means of local geographic authentication where users—each user represented as a smartphone—mutually and simultaneously encode their geographic location using handshakes with nearby users. In this work, we present a blockchain-like distributed ledger protocol which, when implemented on top of some smartphone ad hoc network (SPAN), encodes user geotemporal information strictly in terms of the topology of that user's local blockchain. We further present a simulation package which—unlike any preexisting package know to the authors—allows for the simulation of a blockchain-like ledger system implemented over arbitrary SPAN topologies.

2 Blockchains over SPANs

Many smartphones today are equipped with wireless functionality specifically geared towards interacting with other smartphones in near geographic proximity. Assuming these connections faithfully represent geographic proximity[1], it follows immediately that, if each smartphone is a node u connected to node v if and only if u and v are within interaction distance—the set of neighbors of u serves as a valid representation of u's geographic location. Further, if two nodes u, v claim to be in the same geographic location (given the location is sufficiently granular), it is expected that the set of neighbors they report would significantly overlap. For this reason, recording "handshakes" with peers in geographic proximity should serve as a valid representation of one's location over time.

2.1 Three Evolving Topologies: Implications of Blockchain over a SPAN

The Bitcoin protocol is heavily dependent on a few assumptions concerning the topology of the underlying network. While it relies on the fact that no node can have global knowledge of what nodes are in the network due to the ease with which nodes may enter and exit the network, it also assumes that the network is stable enough to offer such features as never partitioning and never being susceptible to man-in-the-middle attacks [11]. This relative stability is what allows all nodes in the network to asymptotically agree on which blocks constitute the

[1] Breaking this assumption is an important problem which we do not attempt to address in this paper. We are only concerned with whether a blockchain-like distributed ledger implemented over a SPAN can reliably encode information about the SPAN's evolution over time using nothing more than the topology of local copies of the blockchain.

global blockchain. Further, this global blockchain naturally arises as the set-theoretic intersection of all local chains in the network.

Implementing a blockchain system over a SPAN, however, breaks all of the aforementioned assumptions. SPANs are highly volatile, and they can easily partition into disconnected components. This makes the idea of nodes globally converging to agreement on what is a global blockchain ridiculous, as the intersection of all local blockchains would cease to grow as soon as nodes disagree on which is the longest chain. It is guaranteed, however, that for all nodes u in the SPAN G, there exists some neighborhood $U \subset G$ containing u such that, if one takes the intersection

$$\mathbf{B}_U = \bigcap_{v \in U} \mathcal{B}_v$$

of all local blockchains $\mathcal{B}_v$ for all $v \in U$, it holds that $\mathbf{B}_U$ grows nontrivially for all timesteps[2].

The degree to which all subsets $U \subset G$ satisfy this property is the degree to which a global blockchain is well defined, and the degree to which most subsets U fail to do so can serve as a measure of the degree to which a global blockchain is poorly defined. For this reason, the authors present the idea of a *semi-global blockchain*: the global blockchain with regards to a connected subnetwork U of the network G, as arises as the set-theoretic intersection of all local chains $\mathcal{B}_v$ held by nodes $v \in U$.

Traditionally, blockchain networks strive mantain a single longest path starting from the root of the chain; that is, the goal is to avoid global forks. In the case of Bitcoin, for example, local, temporary forking still happens but one of those forks is usually quickly abandoned by the network [4]. In our case, since SPANs are often disconnected, the global blockchain will eventually fork. We actually embrace forking as a feature: each long-term fork represents a connected component of the underlying SPAN.

3 The SPANchain Simulator

We present a tool which allows for the simulation of the evolution of SPANs—represented as random geometric graphs—over time, and which, most saliently, allows for the simulation of a novel blockchain-like distributed ledger overlaid on this SPAN. While the authors are aware of 1) network simulation tools that accommodate dynamic network topologies [2] and 2) tools for simulating blockchain systems distributed over a network of nodes, we are unaware of any tool that allows for the simulation of a blockchain system implemented over arbitrary dynamic networks. Part of the reason for this is that SPANs, by nature, induce a network topology that fragments easily and from which nodes enter and exit with great frequency. For conventional blockchain systems, this inevitably leads to forking in the global chain—a property which, for cryptocurrency applications, is highly undesirable [4, 11]. Using our simulation tool, however, we show that the way global forks form in blockchain networks implemented over SPANs can encode geotemporal information about how the SPAN evolves over time.

In order to observe the behavior of this network under several scenarios, we wrote a simulator package SPANchain[3] for the protocol.

To the best of our knowledge, this is the first blockchain simulation framework which simulates blockchains implemented over SPANs, as well as the first to embrace forking as a feature rather than a bug. We also are not aware of any packages that implement blockchains over SPANs.

Although we implemented the protocol described in section 4 in the simulator, the package is written to be agnostic to the protocol for block formation, as well as processes by which the underlying topology evolves.[4] We hope that these packages can be used by other researchers to better understand how blockchains fork, and what information this forking encodes with respect to the history of the underlying topology.

4 Adapting the Bitcoin Protocol

This section formalizes how the SPANchain module models the Bitcoin protocol for block creation and validation. This adaptation loses the notion of a transaction—the primitive stored by the blocks in Bitcoin—and convolves the notion of a block and handshake into essentially the same primitive. This adaptation pivots from the Bitcoin protocol in such a way as to make this convolution well-defined while preserving some of the robustness of the blockchain against adversarial behavior shown by Nakamoto [11].

4.1 Exchanging Transactions for Handshakes

The blockchain is essentially a tree of blocks, containing solutions to cryptographic challenges. A *cryptographic challenge* is a pair

$$\mathcal{P} = (t_C, k_C)$$

where t is a timestamp and k is the public key of the problem creator. Each *block* is, in turn, a tuple of the

[2] To see that this is correct, it is sufficient to note that this trivially holds for $U = \{u\}$.

[3] The code for the simulator is publicly available at https://github.com/robbobbinett/geographic_auth_with_bc.

[4] We have, however, implemented a class of SPAN-nodes to specifically handle the evolution of the SPAN with respect to a random geometric graph model.

	Bitcoin P2P	SPAN P2P
P2P Topology	The Bitcoin protocol relies on the assumption that no node in the underlying network can have complete global knowledge of the state of the network, especially given nodes can spontaneously enter or exit the network. However, it is also assumed that this network never becomes disconnected and never becomes vulnerable to man-in-the-middle attacks. In this way, it is safe to assume that all local nodes can eventually agree on some global state information.	SPANs are highly volatile. The frequent making and breaking of connections causes the SPAN to fragment into disconnected components. It is hopeless, therefore, to assume that all the nodes will ever agree on global state information that is not granted *a priori*.
Local Blockchain (per node)	Very similar, up to slight differences in protocol	
Global Blockchain	The global blockchain is implicitly induced as the set-theoretic intersection of all local blockchains. It is guaranteed in probability that this intersection chain is always growing; failure to grow would imply that this global chain has forked.	Though the set-theoretic intersection of all local blockchains is well-defined, this intersection is likely to be noninteresting and stop growing after some timestep due to forking.
Semi-Global Blockchain (per neighborhood)	For any connected subnetwork G' of the network, the global chain of the restriction to G' always (asymptotically) agrees with the well-defined global chain.	Let G be the network. For each node $u \in G$, there exists neighborhoods $U, u \in U \subset G$ such that the global chain is well defined with respect to the restriction to U. We can define *semi-global blockchains* as the intersection of local chains with respect to such restrictions.

Figure 1: The Nakamoto paper assumes that the blockchain protocol is implemented over a relatively stable P2P network. If we attempt to implement a blockchain protocol over a SPAN, however, many of the notions that naturally arise in the case of Bitcoin no longer appear. The SPANchain simulator was written to allow for easy simulation of blockchains implemented over SPANs, together with graphic tools which the authors hope helps the research community come up with ways of studying semi-global blockchains in the case of a forking global chain.

form

$$B = (t_C, k_C, t_S, k_S, n, b),$$

where (t_C, k_C) form a cryptographic challenge, n is the nonce which solves the challenge (described below) k_S is the public key of the problem solver, t_S is the timestamp of when n was found, and b is a pointer to a parent block. Additionally, the empty block $(\varnothing, \varnothing, \varnothing, \varnothing, \varnothing)$ is a valid block. This is the global root of the blockchain tree.

We assume the existence of a cryptographically secure hash function H. A nonce n is a *solution* to the problem contained in a block B as above if $H(B) < \tau$, where τ is a predefined threshold value known *a priori* by the network.

4.2 Rate of Block Formation as a Function of τ

Nakamoto's original paper discusses regulating the rate of block formation in the global chain by periodically updating the threshold τ for which all blocks B must satisfy $H(B) < \tau$ [11]. Decker *et al.* subsequently show that the rate at which block collisions occur in the Bitcoin network can be modeled as a function of the rate of block creation. Putting these together, we surmise that our rates of block formation and global forking can be set arbitrarily high or low relative to the rates at which changes occur in the topology of the underlying SPAN. This allows us to simulate our peer-to-peer (P2P) message passing over the SPAN in such a way that, given network churn is kept low, we can run SPAN updates and block updates/propagation as discrete events that do not overlap or otherwise interfere.

5 Simulation

5.1 SPAN Connectivity Model

The protocol runs over an ad-hoc network, where each node is a smartphone and the nodes connect wirelessly to each other. In the simulation, we modelled this underlying infrastructure using a dynamic random graph.

The most commonly studied model of random graphs is the Erdős–Rényi model. Given parameters n and p, the model generates a graph with n vertices and each edge is added to the graph independently with probability p [1]. This model, however, has no information about node positions, which makes it a poor representative of ad-hoc networks [8].

Instead, we used random geometric graphs as our connectivity model. Given parameters n and r, the model randomly chooses n points in the unit square $[0,1]^2$, uniformly and independently, and adds an edge between each points which are at a distance r from each other. Random geometric graphs have been proposed as accurate models for ad-hoc networks [10, 8].

Nodes which are close to the borders of the unit square are affected by the so-called "border effect": if the distance of a node is smaller than r from any of the borders, then part of the "area of coverage" for this node (that is, the area which may contain adjacent vertices) lies outside of the unit square. This effectively truncates the degree of that node, because no nodes are generated outside the unit square. For smaller values of r this effect is less pronounced, because less nodes are affected.

In order to avoid the border effect, we used toroidal distances instead [10, 8]. The nodes are placed on the unit torus $[0,1)^2$ and distances will be measured according to a toroidal metric. So, for example, for connectivity radius $r = 0.1$, the nodes at $(0.01, 0.01)$ and $(0.99, 0.99)$ are adjacent in this metric.

We denote the toroidal random geometric graph model with parameters n and r by $T(n,r)$.

We want to model the underlying infrastructure to be dynamic, to represent the fact that nodes move around in the world. That is, we will have a sequence $G_0, G_1, \ldots$ of graphs over the same set of nodes, which represent the evolution of the network.

Using the toroidal model has the benefit that if the nodes move around randomly, but independently, then each resulting graph is still a geometric graph. More precisely,

Proposition 1. *Let $G_0 \in T(n,r)$ be a random graph. For each $i \in \mathbb{N}$, define G_{i+1} by translating each vertex $v \in V(G_i)$ by a random translation vector u_v, chosen independently according to some distribution, and connecting vertices which are within a distance of r of each other. Then, for each i, the graph G_i is a random geometric graph distributed according to $T(n,r)$.*

Note that each G_i will be distributed according to $T(n,r)$, regardless of the distribution of the translation vectors u_v (as long as each translation vector is chosen independently, and irrespective of the vector v).

However, there is nothing cohesive in the literature about using several different translation parameters at once. We used a simplified version of the dynamic model of [6]. Fixed a parameter s, G_{i+1} is generated from G_i by translating each vertex by a vector with norm s, chosen uniformly among all s-normed vectors in $\mathbb{R}^2$.

5.2 Visualizing the Interplay between Local and Semi-Global Chains

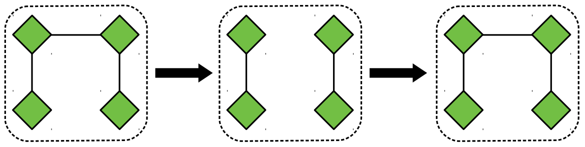
Due to the SPANchain module existing for less than a quarter of a year, it lacks a complete set of visualization tools for analyzing the behavior of semi-global blockchains over SPANs. Partly, this is due to our module being the first to accommodate forking as a feature rather than a bug; the only degree to which forking is well-understood in the literature is how one can best prevent it from happening. Though furthering our work will require rigorous measures of semi-global blockchain formation and diversity, we offer the following toy case as a proof of concept that the persistence of semi-global chains in the network (i.e. the degree to which a non-trivial global blockchain failes to exist) preserve information as to how the SPAN topology has changed over time.

The top of Figure 2 shows a connected SPAN which maintains its initial state long enough for ten blocks to form and propagate. The SPAN then partitions into two connected components, remains in this state long enough for ten more blocks to form and propagate, and finally reverts to its original state. The simulation randomly grants and deterministically propagates ten more blocks before terminating.

The bottom of Figure 2 shows the local blockchain held by a specific node in the described SPAN. The local chain recognizes all blocks indices zero through nine, as all problem formulations and blocks from all other nodes reached the node in question. However, this node does not recognize all of blocks ten through 19, as some of these were discovered by nodes in the other partition during the period of disconnection. By the time the two partitions joined back together, we see two branches in the local chain competing for status as global chain; one of these, which begins with the edge from node zero to node 21, represents the continuation of what had been the longest chain in the other partition. As half the nodes in the network recognize it as the longest chain, leaves on this chain become parent references for roughly half of the problem proposals created in the network. Though the branch which corresponds to the global chain in what had been the node's own partition continues to grow, it is actively competing with the continuation of the other partition's leading chain. This long-term coexistence of these two branches is strictly an artifact of the network's partition and reunification, and the lengths of the branches correspond to the duration of this separation.

6 Resilience against first-order attacks

In order to analyze adversarial behavior, we make the assumption that all nodes have about the same computational power. We consider the situation in which a



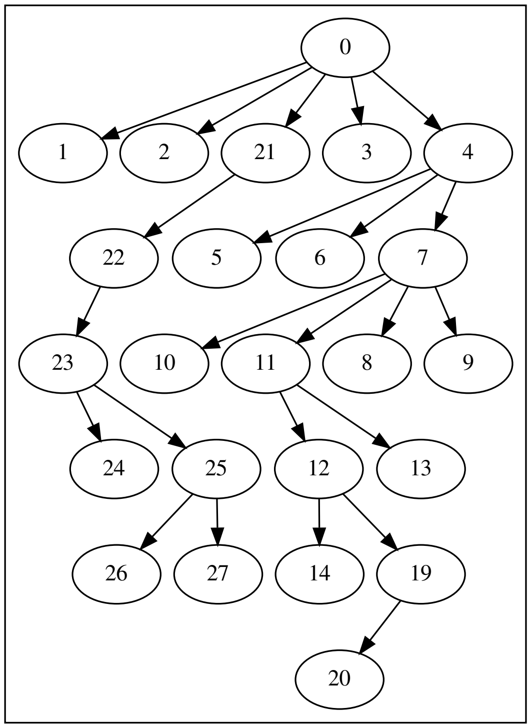


Figure 2: A proof-of-concept SPAN-distribted blockchain simulated in SPANchain. The simulation is performed in the script `horseshoe_partition.py`, visualization of the output is performed using `make_figures.sh`.

single node, without additional computational power, is trying to disrupt the network. That is, we analyze the protocol's resilience against first-order attacks.

A misbehaving node may only interfere with either the blockchain infrastructure or with the message-passing algorithm.

Since it is not possible to erase any blocks from the blockchain, the only way a misbehaving node could affect the ledger is by adding blocks. However, as we are supposing the malicious node has about the same computational power as the other nodes in the network, the villain would only be able to add blocks at the same rate as the other nodes, so these extraneous blocks would be a minority in the chain. Distributed consensus cannot be affected by a few blocks in the chain.

For the message-passing algorithm, the malicious node cannot forge or alter messages from other users due to cryptographic signatures. Dropping messages from other users is not effective; this is tantamount to making the network disconnected, which is a situation the protocol is designed to tolerate.

However, the node is able to generate several extraneous cryptographic challenges; that is, a single node could create several virtual identities (by generating several pairs of public and private keys) and pose cryptographic challenges to other nodes posing as each of these identities. Other nodes in the network would not be able to differentiate between these virtual identities and other honest nodes, and thus waste cycles trying to solve cryptographic challenges posed by nodes which do not exist. Although this attack does not erase previous blocks, it slows down the network, preventing it from doing meaningful progress.

7 Conclusions and Future Work

We have shown that when a blockchain protocol is implemented over an *ad hoc* network, it is possible to encode information about the network's evolution within nodes' copies of the local blockchain. We motivate the utility of using blockchains to encode this information in the context of local geographic authentication —fully acknowledging that implementing our protocol to the problem of geographic authentication requires hardware considerations which we do not address here. However, given sufficient exploration of the hardware side of the problem, it is plausible to use the protocol described in this paper to implement distributed geographical authentication.

More importantly, our work suggests that there exist applications for which a blockchain that sports "forking as a feature" are desirable. Given the nauseating volume of research currently focused on blockchain technology, we hope that our findings here—as well as the SPANchain simulator—provide a platform for blockchain researchers to solve more varied problems using blockchain variations.

Since using forking as a feature is a novel concept, there are several questions regarding its behavior over a SPAN which have no answer yet. For example, we may enquire, over a given SPAN, the average radius from a source node u for which non-trivial semi-global blockchains cease to exist. We may also wish to formalize means of counting discrete forks, or otherwise making an indiscrete forking index, for applications wherein "forking as a feature" is desireable. Tools for analyzing and visualizing these forks, of course, would also be required in such a wave of next steps.

As outlined in Section 6, making the protocol resilient to first-order attacks is an open problem. A possible solution involves adding a computational cost to enter and stay in the network; for example, honest nodes could simply try to solve cryptographic challenges from nodes which have appeared "recently" in the blockchain.

References

[1] Béla Bollobás. *Random Graphs*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2 edition, 2001. doi:10.1017/CBO9780511814068.

[2] Rachna Chaudhary, Shweta Sethi, Rita Keshari, and Sakshi Goel. A study of comparison of network simulator-3 and network simulator-2. *International Journal of Computer Science and Information Technologies*, 3(1):3085–3092, 2012.

[3] Armin Dammann, Simon Plass, Matthias R ockl, and Thomas Strang. Method for verification of identity of entity of user or device in relation to another verified entity, during geographical authentication, involves pre-determining measure of specific blur in identifier by updated position bit-accuracy, 2010. URL: https://patents.google.com/patent/DE102010044226B4/en.

[4] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *IEEE P2P 2013 Proceedings*, pages 1–10, Sep. 2013. doi:10.1109/P2P.2013.6688704.

[5] Dorothy E. Denning and Peter F. MacDoran. Location-based authentication: Grounding cyberspace for better security. *Computer Fraud & Security*, 1996(2):12–16, 1996. doi:https://doi.org/10.1016/S1361-3723(97)82613-9.

[6] Josep Díaz, Dieter Mitsche, and Xavier Pérez-Giménez. On the connectivity of dynamic random geometric graphs. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, pages 601–610, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[7] René Glas. Breaking reality: Exploring pervasive cheating in foursquare. 2015.

[8] Ramin Hekmat. *Ad-hoc Networks: Fundamental Properties and Network Topologies*. Springer Netherlands, 2006. doi:10.1007/1-4020-5166-2.

[9] Luke Irwin. The GDPR: US news sites block EU visitors, 2018. URL: https://www.itgovernanceusa.com/blog/the-gdpr-us-news-sites-block-eu-visitors.

[10] Hichem Kenniche and Vlady Ravelomananana. Random geometric graphs as model of wireless sensor networks. In *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*. IEEE, February 2010. URL: https://doi.org/10.1109/iccae.2010.5451758, doi:10.1109/iccae.2010.5451758.

[11] Satoshi Nakamoto et al. Bitcoin: A peer-to-peer electronic cash system. 2008.