Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

A GENERALIZED AGENT BASED FRAMEWORK FOR MODELING A BLOCKCHAIN SYSTEM

Chaitanya Kaligotla
Charles M Macal

Decision and Infrastructure Sciences
Argonne National Laboratory
9700 S Cass Ave
Argonne, IL 60439, USA

ABSTRACT

We describe an agent-based conceptual model of a Blockchain system. Blockchain technology enables distributed, encrypted and secure logging of digital transactions in an add-only ledger record. We model and simulate an expanding Blockchain network, describing the participating agents, the transactions, and the verified add-only public ledger record achieving decentralized consensus. All of the essential details of the functioning of the Blockchain including the behaviors and decisions made by agents deciding to join as well as participate in the market are detailed in the prototype model. The aim of this paper is to illustrate the essential elements and functioning of a Blockchain system, implement a generalized simulation and a measure of Blockchain efficiency from an agent choice and energy cost perspective. Our preliminary results indicate that mining choice (transaction block to verify) coupled with proof of work incentives are critical for energy efficiency.

1 INTRODUCTION

The term "Blockchain", although quickly becoming ubiquitous, has no standard technical definition in the literature (Narayanan and Clark 2017). Rather, it is commonly used as an umbrella term to describe systems which resemble a use-case for a distributed public ledger like *Ethereum* (Buterin 2013), or, some crypto-currency like *Bitcoin* (Nakamoto 2008). The technical characteristics of Blockchain typically denote a network-based distributed public ledger system characterized by append-only transactional records, achieved through distributed consensus (trust) across a decentralized network.

Blockchain systems are generally thought to be secure by design (through the use of cryptographic hashing) and exemplify a distributed computing system with a high fault tolerance (not sensitive to component failures). They are highly suited for record keeping and processing transactions with a high degree of security, low risk, and at greater speeds. The unique value in this technology is recognized as a natural extension of the Internet as a fabric of daily life.

Blockchain technology is increasingly being touted as a solution to many modern problems, from alternate currency systems, to applications in healthcare, to helping modern governance. Some states and countries are exploring Blockchain technology to aid governance (https://spectrum.ieee.org/computing/networks/illinois-vs-dubai-two-experiments-bring-Blockchains-to-government). Blockchain technologies are at the cusp of technological advances and have been the prime focus of a significant amount of R&D and entrepreneurial investments. The various technologies themselves, apart from the applications, are still in development, and the challenge involves the effective intersection of a number of distinct scientific and technical areas - cryptography, computer science, social sciences, economics and financial markets.

978-1-5386-6572-5/18/$31.00 ©2018 IEEE

1.1 Research Goal

We use the term Blockchain to refer specifically to a dynamic system, consisting of underlying elements (defined later) which are dynamic, and interact with each other resulting in a dynamic system state. In this paper, we describe the various elements of a Blockchain, illustrate their dynamics, and develop a generalized representative model of this system. This is similar to the description of Ethereum in Wood (2018), as a specialized version of a secure transaction-based state machine.

Our specific aim in this paper is to: i) develop an Agent Based model of a Blockchain System to illustrate the essential elements, dynamics and functioning of the system, ii) demonstrate the use of ABM methodology to simulate Blockchain systems, and iii) characterize Blockchain systems for average energy cost per verified transaction. To this end, we briefly describe the relevant literature related to Blockchains in Section 2, highlighting the development of the underlying technologies, concepts and ideas. In Section 3, we present our generalized agent based model of a Blockchain System, describing the various components, their interactions and dynamics of the system. We briefly describe our prototype simulation using *Mathematica* (Wolfram Research 2018) in Section 4, and discuss results in Section 5.

2 LITERATURE

In this section, we describe the features and conceptual building blocks of a Blockchain System, highlighting the development of Blockchain ideas through the relevant literature. The development of Blockchain technology is rooted in diverse ideas across extant literature. Narayanan and Clark (2017) show that nearly all of the technical basis of Bitcoin (Nakamoto 2008) originated in the academic literature. Figure 1 shows a brief time line of the development of the critical technological and concepts related to Blockchain.

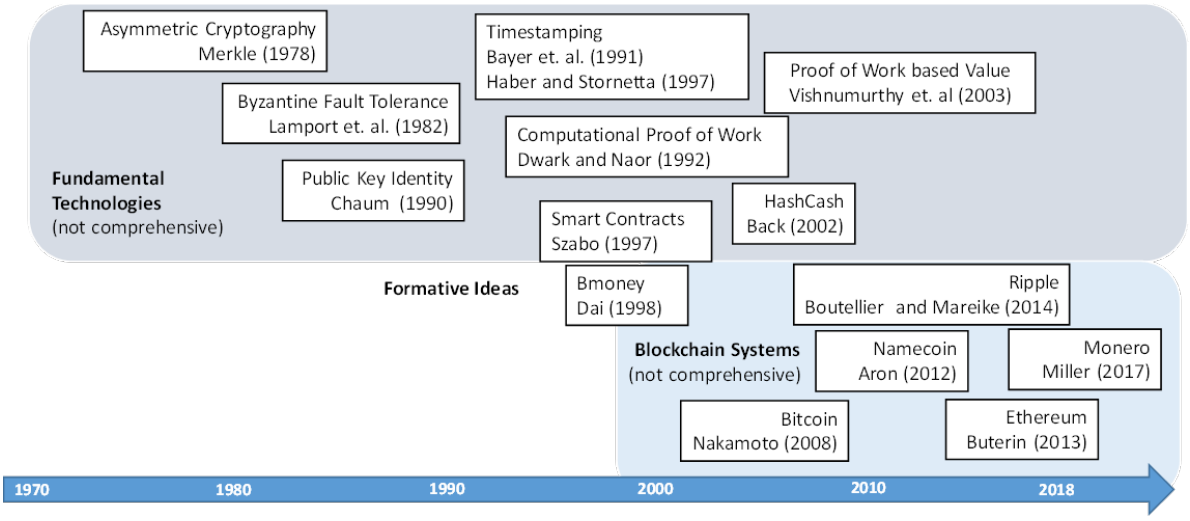

Figure 1: Timeline of literature showcasing fundamental concepts, ideas and development related to Blockchain systems.

2.1 Features of Blockchain Systems

A Blockchain system can be described as a distributed peer-to-peer network using cryptography to either securely host applications, store data, or digitally transact instruments of value (like contracts). Central to this system is the idea of the *Public Ledger*, which is a record of the transactions in a Blockchain system. The properties of the public ledger in the Blockchain system derives from features of the system – timestamping, consensus and fault tolerance, and proof of computational work.

Ledger: A Ledger is a place to record all transactions that happen in a system. In the context of a Blockchain System, a ledger is generally defined as a globally distributed data structure, collectively maintained by independent participants, irrespective of mutual trust. The public ledger in a Blockchain system however has some unique features: i) it is *distributed*, i.e., it exists across the network with no central authority, and ii) it is *immutable*, i.e., the ledger is append only, i.e. new transactions can be added, but historical transactions which are part of the ledger cannot be removed, modified or reordered.

Cryptographic Hashing and Merkle Trees: Cryptographic hashing is a method of secure verification of any data in a computer system, and is based on concepts of asymmetric cryptography (Merkle 1978), which is the basis of secure messaging in a peer-to-peer computer network. Merkle Trees (Merkle 1980) are data structures used to record the canonical order of transactions in the Blockchain Ledger , which are then cryptographically hashed to serve as a basis for secure verification of historical transactions. As a result of the Merkle Tree structure, knowledge of the latest hash of the ledger is sufficient to enable secure verification of the whole ledger from distributed sources of the hashed ledger. Another key property of this data structure is that it enables verification of a single historical transaction in the ledger.

Timestamping: Timestamps of transactions in any record are critical for a variety of applications – patents, contracts, documents, etc, as it enables verification of individual transactions in the ledger. A critical property of the data structure of the public ledger in the Blockchain System is that all transactions are timestamped, originating from Haber and Stornetta (1991), Bayer et al. (1993), and Haber and Stornetta (1997). Haber and Stornetta (1991) describes a digital notary service, describing the timestamping of transactions or documents, following the structure of a Merkle Tree. Here, the creator of each document or transaction cryptographically hashes the time of creation, its timestamp, to the previously broadcast document or transaction, which has signed its own predecessor, thereby forming a recursive chain of immutable timestamped transactions or document records. Thus, if you are given a single item in the chain by a trusted source (e.g., another user or a specialized timestamping service), the entire chain up to that point is locked in, immutable, and temporally ordered. Haber and Stornetta (1997) also introduced the concept of grouping transactions or documents into batches or blocks, with documents in each block having essentially the same timestamp. They made it efficient using (Merkle 1980), where within each block, documents can be linked together with a binary tree of hash pointers, called a Merkle tree, rather than a linear chain.

Consensus and Fault Tolerance: Another critical property of a distributed public ledger is that of consensus, where all participants agree on the exact order or structure of the pubic ledger absent a central authority. To illustrate, consider a case of a distributed ledger where groups of nodes disagree on the latest block of transactions in the public ledger. This difference is called a *fork*, and could be because of an adversary, network latency, or error. Blockchain Systems are designed to avoid forks, as that violates immutability and verification. Fault tolerance (Lamport et al. 1982) is based on the idea of the number of nodes that can fail (or deviate from consensus) on a network, without effecting the integrity or operation of the network. Castro and Liskov (1999) introduced a practical Byzantine fault tolerance concept accommodating for both Byzantine faults and an unreliable network. Research in fault tolerance in Blockchain Systems is currently a critical area of focus. Virtually all fault-tolerant systems assume that a strict majority or super-majority (e.g., more than half or two-thirds) of nodes in the system are both honest and reliable.

Proof of Computational Work: Dwork and Naor (1993) first introduced usage of cryptographic proof of computation expenditure (proof of work) as a means to deter spam. Critically, their method allowed a receiver to make a verification without having to rely upon *trust* of the sender. The central idea was that email recipients would process only those emails that were accompanied by proof that the sender had performed a moderate amount of computational work. This would be a disincentive for a spammer to send a large number of emails that would require higher computational power and costs. Vishnumurthy et al. (2003) demonstrated the use of proof-of-work to secure a digital currency, augmenting digital signatures and a ledger in order to ensure that the historical record couldn't be corrupted and that malicious actors could not spoof payment or unjustly complain about service delivery. Proof of Work was typically achieved

by a stochastic set of puzzles to be solved by one party, and verified by the other. Independently, Back (2002) introduced a system to use proof of work as a form of currency. Nakamoto (2008) developed this proof-of-work to secure crypto-currency, and Bitcoin thus became the first widely adopted global decentralized transaction ledger.

2.2 Development of BlockChain Systems

Dai, Wei (1998) is often cited as the genesis for the idea of decentralized consensus achieved by creating money or value by solving computational puzzles. Szabo (1997) is well recognized as an early pioneer in Smart Contracts, as was Miller, Mark (1997)'s ideas in the legal domain. Smart contract is a term used to describe a program (using Blockchain technology) that is capable of facilitating, executing, and enforcing an agreement. Smart contracts are tempting from a social contract perspective – that algorithmic enforcement of agreements could become a significant force in human cooperation. Ethereum (Buterin 2013) began as a general implementation of such a crypto-law system.

Nakamoto (2008) was the first to combine multiple concepts described above to implement and achieve distributed consensus (called Nakamoto consensus) for the order of transactions. This work is a watershed moment in technology, and since then, Bitcoin was widely adopted and continues to be the largest Blockchain System by value and number of transactions. Other projects built on Bitcoin's success; numerous other currencies were introduced through alterations of the protocol. Some of the best known are Litecoin and Primecoin, discussed by Sprankel (2013). Other projects have repurposed the protocol, Aron (2012), for example, describes the the Namecoin project which aims to provide a decentralized name-resolution system. Additional work has been done in the area with discarding the decentralization foundation, as in Ripple (Boutellier and Heinzen 2014), which has sought to create a "federated" system for currency exchange, effectively creating a new financial clearing system. It has demonstrated that high efficiency gains can be made if the decentralization premise is discarded. Buterin (2013) and Wood (2018)'s development of the Ethereum paradigm, which provides a platform for a variety of applications, is rapidly growing in popularity and second only to Bitcoin in terms of value and transactions.

Current research focus includes dealing with scalability issues (Vukoli 2016), security issues such as Byzantine fault tolerance (Eyal and Sirer 2014), and analysis of Blockchain Systems (Miller et al. 2017).

3 A GENERALIZED MODEL FOR BLOCKCHAIN SYSTEM

In this section, we describe and formalize a generalized model for a Blockchain System.

3.1 Model Description

We consider two types of agents in the model: market agents and miner agents. Market agents make bilateral agent-to-agent transactions; pool markets are not addressed in the current Blockchain model presented here. Miner agents select blocks of candidate transactions to attempt to verify. Mining tends to be a computer-intensive operation and requires a significant amount of resources including electricity to run a miners computers. Miner agents can also be market agents simultaneously mining transactions and engaging in transactions. Market agents do not have computing power and do not engage in mining; however, it is possible that agents may change roles in the model. Market agents may become miner agents, and miner agents may become market agents as the simulation progresses, to reflect real-world possibilities. In the model presented here, we assume a fix number of agents throughout, and agents retain their original roles throughout the simulation. There are interesting questions about factors that are under an agents control regarding network growth (when do agents decide to join or quit the network?) as well as in the level of resources, such as computing power, that agents should commit to in order make money. These will be addressed in future versions of the model that consider the additional decision making capabilities of Blockchain agents.

Verifying a transaction consists of ensuring that the agents involved in a transaction, specifically the agent that is sending the value of the transaction to the receiving agent, have adequate levels of resources (currency) to cover the transaction. Verification is a process conducted by miner agents by which the history of all transactions that the sending agent of a transaction has ever been involved in, is verified to ensure that the agent sending the value has adequate value (currency) in their wallet, to cover the payment. This ensures that the value offered is truly available to the agent, and that the currency cannot be double counted. Verifying the transaction means that the transaction can be trusted, a feature of any Blockchain. When a miner agent successfully verifies a transaction block, they receive a reward from the network for their service. The reward consists of a fee (for example, gas in the Ethereum network) that the sending agent puts up as an incentive to encourage the transaction to be selected for mining. Successful miner agents split the fee between them; miner agents who are unsuccessful in being the first to verify the transactions they selected to miner from the pending transaction queue receive nothing and endure the costs involved in the mining process.

3.2 Formal Model

Let $\{1,2,\ldots,T\}$ be a set of times, arbitrarily defined to be discrete times. We define a Blockchain System, **BC**, to consist of market agents, **X**, miner agents, **Y**, transactions, τ, in the pending transaction list **PTQ**, and the public ledger **L**.

3.2.1 Market Agents X

We consider a market agent $x \in \mathbf{X}$ to be an agent in a bilateral transactional market, where they may assume the role of buyer or seller in any specific transaction. We make an explicit assumption that the number of participating agents, $|\mathbf{X}|$, is sufficient to create a market (market clearing conditions are satisfied). We do not consider a case where a buyer(seller) cannot find a trading partner to execute a sell (buy) transaction. Each market agent has two characteristic properties: identifier (i), a unique identifier for an agent, and wallet (ω), containing the agents' current balance of currency or value. The state of a market agent i at time t is denoted by $x^t_{i,\omega}$.

3.2.2 Miner Agents Y

We consider a miner agent, $y \in \mathbf{Y}$, on a compute network on which a BlockChain system exists. Miner agents posses the following characteristic properties – identifier (i), a unique identifier for an agent, and wallet (ω), containing the agents' current balance of currency or value, $pwr \in 1,2,\ldots,\mathbb{N}$ denoting the compute power required by a miner agent to solve a cryptographic puzzle as a proof of computation work, and mining cost, $cost$, the cost associated per unit of compute power. The state of market agent m at time t is denoted by $y^t_{m,\omega,pwr,cost}$.

3.2.3 Transactions τ, Transaction List PTQ and Public Ledger L

Transactions are denoted by $\tau_n(i,j,v,g,t,\sigma)$, where n is the unique transaction identifier, i is the transaction source, j is the the transaction recipient, v is the value, g is the gas value (transaction fee), and $\sigma \in 0,1$ is a verification value, where the transaction has either been verified, or not. $\mathbf{PTQ}^t$ is a vector of unverified transactions. $\mathbf{PTQ}^t \triangleq [\tau_1,\tau_2,\ldots,\tau_n]$, where $\tau(\sigma) \neq 1$ Miner agents select transactions to verify from the $\mathbf{PTQ}^t$. Once a transaction has been verified, it is removed from from the Pending Transaction Queue between t and $t+1$. Public Ledger is a vector of verified transactions, which were previously in the Pending Transaction Queue. $\mathbf{L}^t \triangleq [\tau_1,\tau_2,\ldots]$ where all $\tau(\sigma) = 1$ are verified.

3.2.4 Appending to Public Ledger – Mining

The process of mining, carried out by miner agents, is the process of verification of as yet unverified transactions in the PTQ. A miner agent, selects a fixed number of transactions ($blocklength$) from the PTQ to form a candidate block of transactions to verify. The miner agent then begins the process of verifying the candidate block of transactions. When the candidate block is verified by at least μ other agents, and the candidate block is valid (none of the transactions in that candidate block have already been added to the Public Ledger), then the candidate block gets added to the Public Ledger. Once a transaction is part of the candidate block that has been added to the public ledger, the transaction has a verify value of 1, is removed from the PTQ, and no further verifications are possible. That implies that all candidate blocks with the transaction in question are no longer valid. Figure 2 depicts the process of mining.

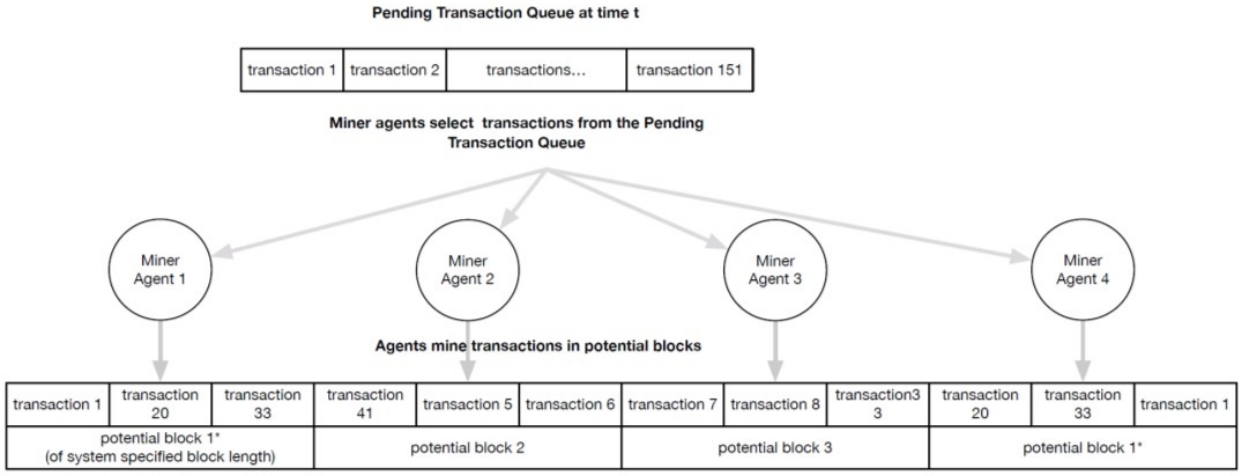

Figure 2: Mining Transactions - Mining is a process of dedicating effort to bolster one series of transactions (block) over other potential competing blocks (candidate blocks). *In this version of the blockchain simulation, we assume that transactions are orderless in a block.

3.2.5 System State Dynamics

Let $\mathbf{BC}^t$ denote the state of the Blockchain System at time t. Analogously, the superscript t is used to denote the state of each subsystem at t. Thus system state at time t is $\mathbf{BC}^t = \{\mathbf{X}^t, \mathbf{Y}^t, \mathbf{PTQ}^t, \mathbf{L}^t\}$ and we can thus characterize system state transitions as $\mathbf{BC}^t \rightarrow \mathbf{BC}^{t+1}$. Figure 3 depicts this transition. At time t, market agent x_i transacts with x_j through transaction τ_n. At time $t+1$, the Pending Transaction Queue is updated to include τ_n. Through mining process $V(\ldots)$, miner agents from $\mathbf{Y}$ verify transaction τ_n after $t+Nt$ where $N = 2,3\ldots$. Once τ_n is verified, the system state becomes $\mathbf{BC}^{\mathbf{t+1}}$.

4 BLOCKCHAIN SIMULATION AND EXPERIMENTS

The Blockchain model simulates the operation of a Blockchain as it unfolds over a time period. In the results presented here, we simulate a Blockchain for a period of 10 minutes at 15 second intervals. At the end of each interval the state of the Blockchain is updated. In this sense, the model is a time-stepped simulation to most closely reflect how real-work Blockchains are structured, such as Ethereum.

At the end of each time interval, a series of events occurs. An agent submits bilateral transaction with another agent in the network. Candidate transactions are placed in a pending transactions queue and must be verified before they are added to the Blockchain, the permanent record and public ledger of the network. Miner agents, who have high-performing computers, select blocks of transactions to attempt to verify. Miners compete against other miners to be the first to verify selected blocks, which consist of a

fixed number of candidate transactions. When a sufficient number of miners (a number set by the network protocol) have verified the same block, that block is permanently added to the Blockchain. The Blockchain is the public ledger in that it contains the all the transactions that have been previously verified over the entire history of the Blockchains existence. Details on the implementation of the simulation model are provided in the Appendix A.

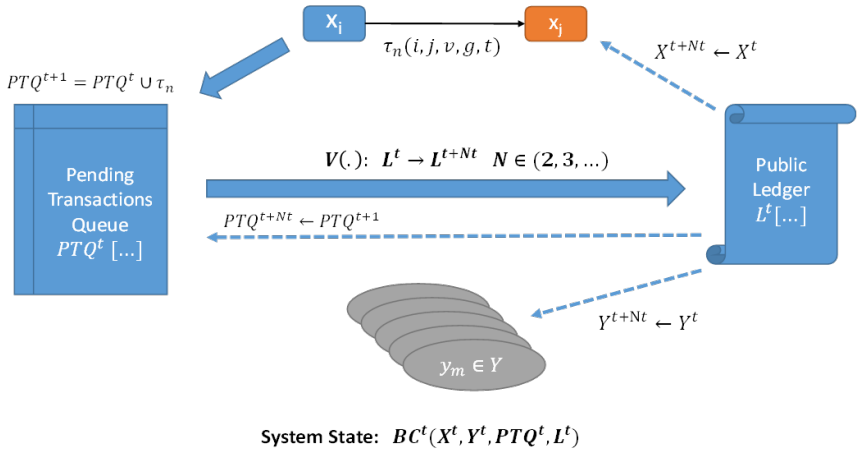

Figure 3: Elements of a generalized Blockchain system depicting the process flow for a single transaction.

4.1 Scenarios

There are several interesting questions regarding the protocols of how a Blockchain should be setup and the impacts that the setup has on the Blockchains' performance. Performance can be measured by several criteria, including the efficiency of the transaction verification process, energy consumption of mining to verify new transactions and blocks, total value of the Blockchain, etc. There are also interesting questions concerning the strategies that miners could have in terms of the computing performance needed to verify the most transactions, which blocks to choose to verify for possible rewards if processed efficiently relative to other miners, etc. In this section, we show the results of a few computational experiments to look at alternative transaction block selections strategies that miners may adopt and to see the effects on the block verification rate and energy consumption.

Energy consumption of Blockchains is a major concern, as some have estimated that the total consumption of electricity spent on Blockchain verification computing using the bitcoin network, is equal to the entire power consumption of the country of Denmark (https://www.washingtonpost.com/news/energy-environment/wp/2017/12/19/why-the-bitcoin-craze-is-using-up-so-much-energy). These results are preliminary and representative of how the model can be used to explore alternative Blockchain designs and to understand the effects of various agent behaviors that miners might adopt.

We ran three scenarios in which we varied how agents selected blocks of candidate transactions to verify. Miners agents then attempted to verify the transactions in the block. If enough miners successfully verified the block, the miners were rewarded by the network. Miners may verify a block, and consume a considerable amount of electricity to power computers in the process, but not receive a reward if the transactions have already been verified by miners using faster performing computers. The scenarios are as follows:

LARGEST Agents choose the transaction blocks with the highest rewards (gas) if successfully verified with the intent of maximizing their returns.

POLAR Miners with the highest performing computers choose the transaction blocks with the highest rewards, and miners with the lowest performing computers choose the transaction blocks with the lowest rewards.

PARTITION Partition the space of transaction returns to match computer performance That is, miners with the highest performing computers choose the transaction blocks with the highest rewards, miners with the mid-performing computers choose the transaction blocks with the middle rewards, and miners with the lowest performing computers choose the transaction blocks with the lowest rewards.

5 RESULTS AND DISCUSSION

5.1 Simulation Results

In this section we describe some representative results from simulations of the Blockchain model. Figures 4,5 and 6 show the results for simulating the Blockchain for 10 minutes (600 seconds) in which the protocol is to update the system every 15 seconds. Upon an update, new transactions are placed in the pending transaction queue for miner agents to select transaction blocks for verification.

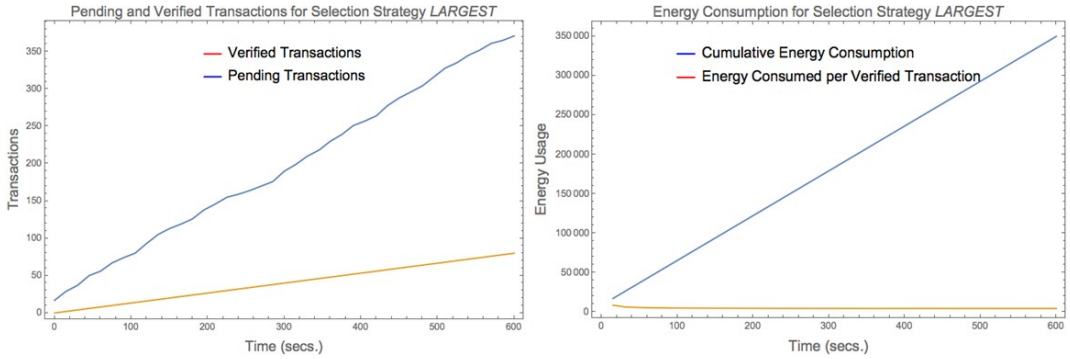

Figure 4: Transactions and Energy Consumption for Miner Strategy LARGEST.

The number of transactions verified is somewhat higher for the POLAR scenario than for the LARGEST scenario (compare Figures 4 and 5), as would be expected, as less miner agents lose out to miners with faster performing computers, but the difference is relatively small. Cumulative energy consumption is very similar, and the energy consumed per transaction is slightly lower for POLAR than it is for LARGEST.

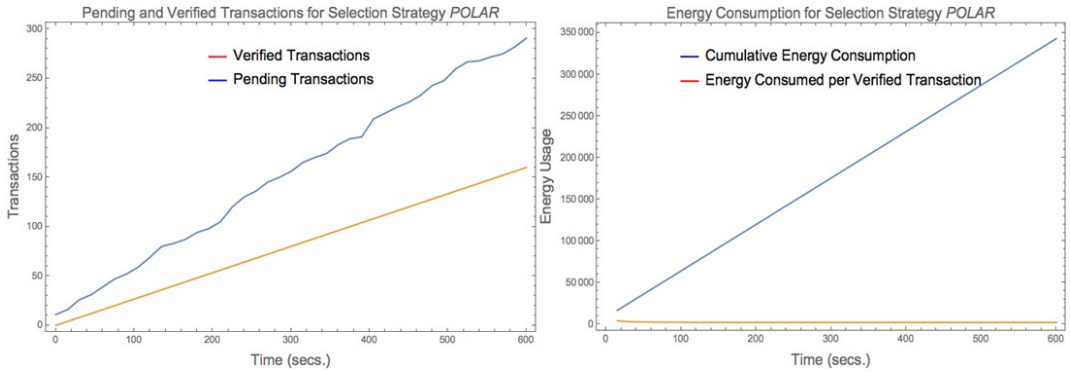

Figure 5: Transactions and Energy Consumption for Miner Strategy POLAR.

The PARTITION scenario in which agents distribute their block selections in a way that most closely meets their relative computer performance capabilities, produces significantly more transactions at a correspondingly lower energy per verified transaction rate (compare Figures 5 and 6). This implies that if miners probe the system and choose transactions to verify at various levels, they may discover their highest

attainable levels of reward that they could receive. This suggests that proof of work incentives set by the Blockchain System can significantly influence energy efficiency. Figure 7 compares the number of transactions verified across all miner block selection strategies.

5.2 Discussion and Future Work

We developed a generalized AB model for a Blockchain system, described our implemented simulation, and demonstrated brief computational experiments relating to energy efficiency of miner agents in the Blockchain system. We find that the strategy scenario in which agents choose verification blocks relative to their computing power is the most energy efficient, showing lower energy per verified transaction, than two other strategies. Our preliminary results suggests that proof of work incentive strategies for miner agents (which dictate mining choice) are crucial for energy efficiency.

More broadly, we demonstrate the strength of using Agent Based Modeling methods to simulate and study Blockchain Systems. Future work will involve further development of our model, including calibration with Bitcoin or Ethereum data.

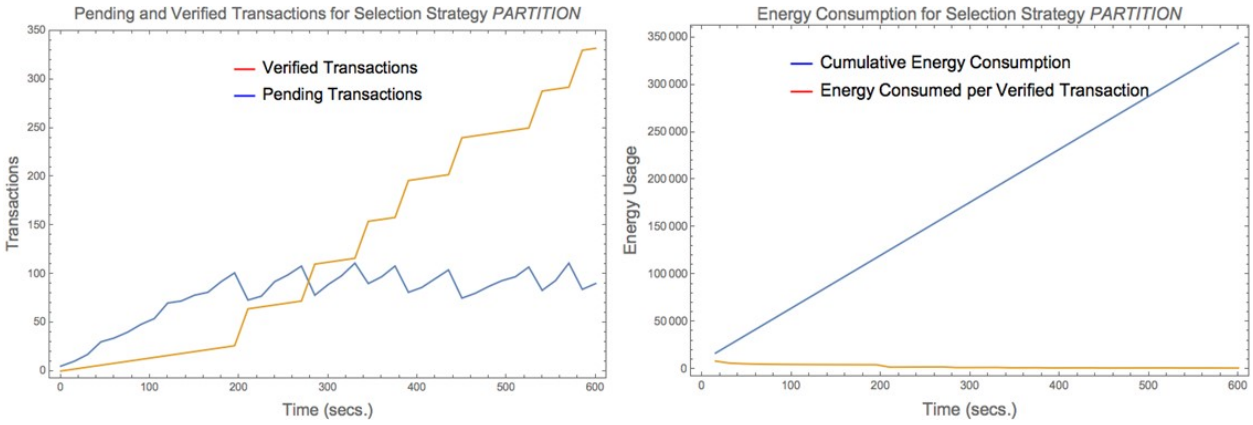

Figure 6: Transactions and Energy Consumption for Miner Strategy PARTITION.

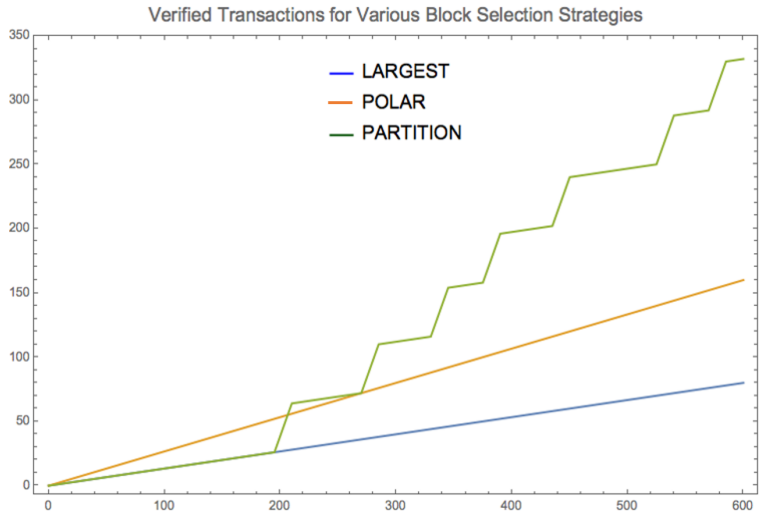

Figure 7: Number of Verified Transactions by Block Selection Strategy.

ACKNOWLEDGMENTS

This work is supported by the U.S. Department of Energy under contract number DE-AC02-06CH11357.

A APPENDICES

A.1 Implementation

We take an object-oriented-like approach to modeling the Blockchain. We implement the model using *Mathematica* (Wolfram Research 2018). We briefly describe the main implementation details here in pseudo-code.

A.1.1 Agents and Objects

There are five main object classes in the model: agent, wallet, transaction, network and simulation. Each class is defined by a set of characteristics and methods that are applied to the respective class. In the following description the expression at the head of a bracket indicates the class, and the variables inside the bracket refer to characteristics, or attributes, of the class. $agent[id, type, wallet[currencyBal], computingPower, miningCost, totalMineCost, blockList, tEndVerify]$ where: id is the unique identifier for an agent. $type$ refers to the agent type, defined as either market agent or miner agent. $wallet[currencyBal]$ is the agents wallet. The wallet contains the agents current balance of currency. $computingPower$ is the relative computing power of the agent. The computing power is used to determine how long and how many resources are required to verify a transaction. $miningCost$ is the relative mining cost per time unit of effort applied to verify a transaction. $totalMineCost$ is the cumulative total mining cost of effort the agent has applied to verify transactions. $blocklist$ is the block being evaluated by a miner agent. A block consists of a number of unverified transactions selected by the agent. $tEndVerify$ is the time that the current block (if any) being evaluated by an agent will be completed. $transaction[idt, agenti, agentj, tCreate, tVerify, gas, value]$ where: idt is the unique identifier for a transaction. $agenti$ is the sender of value in the transaction. agent i sends values to agent j. $agentj$ is the receiver of value in the transaction (agent j receives values from agent i). $tCreate$ is the time when the transaction is created. $tVerify$ is the time when the transaction is verified. gas is the value received by the miners who verify the transaction. $value$ is the value of the transaction, sent by agent i and received by agent j. $network[networkReward, agentsToVerifyTrans, blockLength, gasLevels, deltaT]$ where: networkReward is the network reward split among the miner agents who verify a block of transactions. agentsToVerifyTrans is the number of miner agents the network requires to verify a block. blockLength is the number of transactions in a block. gasLevels is a list of relative gas levels agents can choose from to indicate the priority of a transaction, highGas = 10, mediumGas = 5, lowGas = 2 deltaT is the time increment between the Blockchain system updates (15 secs.) $simulation[runName, nAgents, fracMiner, simScenarios, tBegin, tEnd, simualtionResults]$ where: runName is the name for the simulation run. nAgents is the number of agents in the simulation. fracMiner is the fraction of agents who are miner agents. tBegin is the beginning time for the simulation. tEnd is the ending time for the simulation. scenarioResults is the log of results for the simulation run.

A.1.2 Behaviors, Methods, Decision & Actions

This following methods implement the main logic of the model.
$BlockchainSimulation[tNow, selectTransBlockRule]$: The steps of the simulation are implemented through a series of high level functions calls. See Steps in Simulation section below. $newTrans[nT, tNow]$: Generate nT new transactions at time tNow. $selectTransBlock[agent[id, miner], blockSelectRule, tNow]$: Miner agent selects a block of candidate transactions to verify from the pending transactions queue. Miner agents have various behavioral rules for selecting transactions to verify that give them the best chance of being the first to verify a block and receive a network reward. $verifyBlockOfTransactions[block]$: Miners verify a block of candidate transactions. $transferValueToJ[transaction[idt, agenti, agentj, tCreate, tVerify, gas, value]]$:

Transfer value of a block of verified transactions to receiving agents. *processRewards*[*minersList*, *reward*]: Process rewards to miners who are the first to successfully verify a block of candidate transactions.

A.1.3 Steps in the Simulation

1. Initialize time (tNow = 0), agents, agent wallets, verified initial transactions with 0 gas.
2. Generate transactions between agents. When agent i sends a candidate transaction to agent j, deduct the gas and the value of the transaction from agent i's wallet.
3. Miners select next block of transactions to verify.
4. Time Loop:++tNow
   - 4.1 Verify transaction and block status.
   - 4.2 Determine required verifications.
   - 4.3 The value of each of the transactions in the block is added to the respective agent j's wallet. These values were previously removed from the corresponding agent i's wallets at the initialization of the block verification.
   - 4.4 The agents who verified the transaction block receive credit. The gas value of the transactions in the block and the network reward is distributed to the successful miner agents by increasing the currency value in their wallets. Take top N miners for reward based on fastest processing time.
   - 4.5 Empty transactions in rewarded miners and non-rewarded miner verification blocks. Losing miner agents verified transactions but were too late to receive credit. Empty these agents transaction block.
   - 4.6 The transactions in the block are marked verified and automatically moved out of the PTQ and into the Blockchain.
   - 4.7 Generate next set of transactions between agents.
   - 4.8 When agent i sends a candidate transaction to agent j, deduct the gas and the value of the transaction from agent i' s wallet.
   - 4.9 Miners select next block of transactions to verify.
5. Repeat Time Loop (Step 4) till simulation Stop Time.

REFERENCES

Aron, J. 2012. "Bitcoin Software Finds New Life". *New Scientist* 213(2847):20.

Back, A. 2002. "Hashcash - A Denial Of Service Counter-Measure". *White Paper*.

Bayer, D., S. Haber, and W. S. Stornetta. 1993. "Improving The Efficiency and Reliability Of Digital Time-Stamping". In *Sequences Ii*, edited by R. Capocelli, A. De Santis, and U. Vaccaro, 329–334. Springer New York.

Boutellier, R., and M. Heinzen. 2014. *Pirates, Pioneers, Innovators and Imitators*, 85–96. Cham: Springer International Publishing.

Vitalik Buterin 2013. "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform".

Castro, M., and B. Liskov. 1999. "Practical Byzantine Fault Tolerance". In *Proceedings Of The Third Symposium On Operating Systems Design and Implementation*, 173–186. Berkeley, Ca, Usa: Usenix Association.

Dai, Wei 1998. "B-Money Proposal". White Paper.

Dwork, C., and M. Naor. 1993. "Pricing Via Processing Or Combatting Junk Mail". In *Proceedings Of The 12th Annual International Cryptology Conference On Advances In Cryptology*, Crypto '92, 139–147. London, Uk: Springer-Verlag.

Eyal, I., and E. G. Sirer. 2014. "Majority Is Not Enough: Bitcoin Mining Is Vulnerable". In *Financial Cryptography and Data Security*, edited by N. Christin and R. Safavi-Naini, 436–454. Springer Berlin Heidelberg.

Haber, S., and W. S. Stornetta. 1991. "How To Time-Stamp A Digital Document". In *Advances In Cryptology-Crypt0' 90*, 437–455. Berlin, Heidelberg: Springer Berlin Heidelberg.

Haber, S., and W. S. Stornetta. 1997. "Secure Names For Bit-Strings". In *Proceedings Of The 4th Acm Conference On Computer and Communications Security*, 28–35. Zurich, Switzerland: Acm.

Lamport, L., R. Shostak, and M. Pease. 1982, July. "The Byzantine Generals Problem". *Acm Trans. Program. Lang. Syst.* 4(3):382–401.

Merkle, R. C. 1978. "Secure Communications Over Insecure Channels". *Communications Of The Acm* 21(4):294–299.

Merkle, R. C. 1980. "Protocols For Public Key Cryptosystems". In *Security and Privacy, 1980 IEEE Symposium On*, 122–122. IEEE.

Miller, A., M. Möser, K. Lee, and A. Narayanan. 2017. "An Empirical Analysis Of Linkability In The Monero Blockchain". *Arxiv Preprint* 1704.

Miller, Mark 1997. "The Future Of Law". Talk Delivered At The Extro 3 Conference.

Nakamoto, S. 2008. "Bitcoin: A Peer-To-Peer Electronic Cash System". *White Paper*.

Narayanan, A., and J. Clark. 2017. "Bitcoin's Academic Pedigree". *Communications Of The Acm* 60(12):36–45.

Sprankel, S. 2013. "Technical Basis Of Digital Currencies". *Technische UniversitÄT, Darmstadt*.

Szabo, N. 1997. "Formalizing and Securing Relationships On Public Networks". *First Monday* 2(9).

Vishnumurthy, V., S. Chandrakumar, and E. G. Sirer. 2003. "Karma: A Secure Economic Framework For Peer-To-Peer Resource Sharing". In *Workshop On Economics Of Peer-To-Peer Systems*, Volume 35.

Vukoli, M. 2016. "The Quest For Scalable Blockchain Fabric: Proof-Of-Work Vs. Bft Replication". In *Open Problems In Network Security*, edited by J. Camenisch and D. Kesdoan, Volume 9591, 112–125. Springer International Publishing.

Wolfram Research, Inc. 2018. "Mathematica, VErsion 11.3". Champaign, Il, 2018.

Wood, G. 2018. "Ethereum: A Secure Decentralised Generalised Transaction Ledger Byzantium Version". *Ethereum Project Yellow Paper*.

AUTHOR BIOGRAPHIES

CHAITANYA "CK" KALIGOTLA Ph.D., holds postdoctoral appointments at the Decision and Infrastructure Sciences Division at Argonne National Laboratory and the Consortium for Advanced Science and Engineering (CASE) at the University of Chicago. He obtained his Ph.D. in Decision Science from INSEAD in Aug 2016. CK also has Master's degrees in Business, and in Decision Sciences, and a Bachelor's degree in Industrial Engineering. His email address is ckaligotla@anl.gov

CHARLES "CHICK" MACAL Ph.D., PE, is the Director of the Social, Behavioral and Decision Sciences group at Argonne National Laboratory. He is a member of the INFORMS-Simulation Society, Association for Computing Machinery, the Society for Computer Simulation International, and the System Dynamics Society. He is on the editorial boards of Transactions on Modeling and Computer Simulation, Simulation, and Complex Adaptive Systems Modeling. He has a Ph.D. in Industrial Engineering & Management Sciences from Northwestern and a Masters Degree in Industrial Engineering from Purdue. His email address is macal@anl.gov