CISC-650 – Computer Networks

Final Paper

Due: 12/8/2019

Eric Webb

[ew774@mynsu.nova.edu](mailto:ew774@mynsu.nova.edu)

Dr. Wei Lee

**Table of Contents.**

**Topic Title:** Page 3.

**Abstract:** Page 3.

**Introduction:** Page’s 3, 4, & 5.

**Bitcoin: History and Overview:** Page’s 5, 6, & 7.

**Bitcoin: Mining:** Page’s 7, 8, & 9.

**Bitcoin: Double Sha-256:** Page 9.

**Bitcoin: Coinbase Transaction:** Page’s 9, & 10.

**Bitcoin: Halvening:** Page 10.

**Bitcoin: Birth of ASIC’s and Flaws:** Page’s 10, 11, & 12.

**Birth of Alternate Coins, Decentralized Applications, and Smart Contracts:** Page’s 12 & 13.

**Etherium: Overview:** Page’s 13 & 14.

**Etherium: Sharding:** Page 14.

**Neo: Delegate Byzantine Fault Tolerance as a Practical Byzantine Fault Tolerance:** Page’s 15 & 16.

**Neo: Unspent Transaction Output:** Page 16.

**Conclusion:** Page’s 16 & 17.

**References:** Page 18.

**Certificate of Authorship:** Page 19.

**Topic Title:**

“Strength and Weaknesses in Blockchain Consensus Algorithms.”

**Abstract:**

This academic paper will be based on a technical analysis of popular blockchain consensus algorithms. The methodology for conducting research was be done via academic and professional literature, based from the IEEE and ACM databases. This paper will begin to discuss pros and cons of popular and upcoming blockchain consensus algorithms. Beginning with a brief history of blockchain and popular crypto currencies with their respective consensus algorithms. Examples of these algorithms are the Proof of Work, Proof of Stake, and Delegate Byzantine Fault Tolerance. These discussions explain high and low level overviews of how these algorithms work along with when and where you would implement one. The consensus algorithm’s vulnerabilities and security exploits will also be addressed. These topics will continue on how blockchain effects peer to peer communications because blockchain is essentially a peer to peer decentralized digital ledger and is a fast growing invigorating technology with the potential to disrupt the world market.

**Introduction:**

Since blockchains conception it has become a wide topic of discussion at many levels. Many times with the answers being abstracted away without truly grasping how the underlying architecture works. When blockchain was initiated the general public didn’t quite understand what it was, where it came from, what its future held, and what its future still holds. This narrative will begin to define what the fundamental basis of a blockchain is and how it is defined.

Blockchain has been a confusing topic for most because it is typically used in reference to the crypto currency Bitcoin. While it is true that Bitcoin uses blockchain it is not blockchains only application or specific architecture. There are numerous types of blockchains designed differently and for different purposes. It is widely agreed that blockchain in its most basic form is a decentralized digital ledger. These digital ledgers are essentially databases appending information to them via blocks. The blocks are added to the dataset effectively becoming chains of information. These database are stored within the blockchain community and are not centrally located under one entity. Each one of these databases keep a copy of the ledger and agree to make changes to it. (Zhang, Xue, & Liu, 2019, Pg. 2) That way if one entity has a different ledger the others can compare their copies to see where the change was made. This effectively makes blockchain pretty difficult to manipulate because it would have to effect multiple parties’ ledgers and not just one single node. Because blockchain transactions are stored as copies on multiple nodes within the network this is what makes blockchain public, verifiable, and decentralized. Blockchain is all about consensus and generally aims to answer the byzantine general problem. (Zhao, Yang, & Lou, 2019, Pg 1). In this problem multiples armies must gather a consensus on when to simultaneously besiege a city. They can only succeed if they all attack correctly at the same time or else they are destined to fail. Gathering a guaranteed safe consensus solves this problem. Throughout many conversations on blockchain theory there will be many references to the BGP mentioned above and how algorithms aim to solve it. This is because blockchain aims to create verifiable trust between multiple parties and synchronicities as questioned in the BGP. Having parties that both trust each other and can synchronize events will be crucial to blockchains evolution.

When developing blockchain solutions not only do you have to keep in mind the CIA security triad of Confidentiality vs Integrity vs Accessibility, but an architect has to keep in mind Blockchains trilemma of Decentralization vs Consistency vs Scalability. Blockchains will be designed and altered to fit specific needs around these aspects. A blockchain aims to be truly decentralized by not allowing a single entity to gain control of the network. No one should be able to gang up and control the network in an unfair fashion, this is the goal of decentralization, giving the smaller competitor the same opportunities as a big competitor. Scalability is the next part of the Trilemma and aims to grow a network to be a huge application without giving up security or performance. If an application is truly scalable it should be able to work just as efficiently with more nodes than less nodes on the network. Lastly, being truly consistent is the third factor in the blockchain trilemma. Having all this data stored in many locations means that all these locations must be consistent with their data. Book keeping measures should be used to cross reference this data especially when scaling up. Of course all these factors will differ in importance dependent on your specific application. Being truly decentralized might not be an important factor for your project if you care more about scalability or vice versa.

**Bitcoin: History and Overview.**

Now that this literature has broadly defined the byzantine general problem and blockchains DCS trilemma it will now discuss these applications in their literal references and history within the cryptocurrency Bitcoin and its Proof of Work Algorithm. When blockchain was conceived it snowballed to market essentially piggybacking off of the popularity of Bitcoin. Bitcoin became known worldwide for its revolutionary concepts of a decentralized ledger facilitating financial transactions. Behind the scenes driving this new technology was blockchain itself. This essentially allowed a community of users to verify transactions in a secure but decentralized manor. At a high over view this is done Bitcoins Proof of Work consensus Algorithm. This consensus allows competitors to use electrical energy and hardware to verify transactions and earn rewards for doing so without a central entity (Duy, Hien, Hien, & Pham, 2018, Pg 2). This is the process called mining where competitors are essentially proving their work and getting paid to do so. The beauty of this is it gives the control of the asset back to the people and not a central entity such as a government. The theoretical impacts of this technology were great debated because they challenged the status-qou of centralized ledgers such as a traditional banking system. It was argued that this technology could replace banks because instead of paying the banks fees for verifying transactions you can now have a community of trusted partners verify transactions and store of value at a fraction of the cost while undermining the monotony of a single organized entity. Of course the ramifications for this new technology are not only rarely heard out, but are not typically trusted either. Being a newer technology many questions about architecture and real world use come into play. Also companies are not so comfortable with giving up their legacy systems for a new experimental technology that would essential expose their book keeping to the world. Blockchain has also had a lot of kickback because the creation of this new technology is shrouded in mystery arguably because there is not an agreement on who creator was. The pseudonymous entity Satoshi Nakamoto was the alias used on Bitcoins white paper and the world is not sure whether it is one person or multiple entities under a guise of an alias (Bagaria, Fanti, Viswanath, Tse, & Kannan, 2019 Pg 1).

This has led to the background of blockchain technology and bitcoin mostly to have a low confidence and trust value among consumers. Why would the general public put their trust and money in something that had no definite creator? Even better question is why would the creator want to hide from all the fame and the glory? This would alarm most people from any security standpoint.

**Bitcoin: Mining**

Now that this literature has discussed what Bitcoin is, Proof of Work is, and how mining creates profit for the end user, we will now go into more detail about how these miners create candidate blocks that become eligible to add to the blockchain. These blocks must be properly constructed by a miner and has 6 parameters. The first being the version of the Bitcoin software, the second being the hash of the previous block, third is the merkle root which is a hash representation of all the transactions to be added in the candidate block. Fourth we have a timestamp. Fifth is a target hash threshold, this is what the network difficulty rate is based off of and tells the miners what they are competing against (Vilim, Duwe, & Kumar, 2016. Pg 1). This blocks header hashed value must be less than or equal to the networks target hash, if produces a number above the target hash the results are not accepted. Lastly for the sixth parameter there is the nonce value. This is a variable used in the proof of work algorithm for mining and is essentially a catalyst that triggers the action of an accepted transaction. Once those fields are filled out a block is sent to the network to be validated and if the other miners agree it is added to the ledger. These miners compete at the chance to solve hashing functions for a reward on the network. This is all effected by the network difficulty. When network difficulty increases, than more hashing power is needed to compete. This is a tad bit confusing from a technical aspect because it is quite backwards on what is really happening behind the scenes in the code. When more hashes are being discovered on the network the algorithm recognizes this and will increase the difficulty on the network. This is accomplished by actually lowering networks target hash. When you lower the networks target hash then blocks have less of a chance of getting a number below that value. This can be misleading because to increase network difficulty you have to decrease the networks target hash value. Conversely, to decrease hashing difficulty the network raises its target hash value so miners now have a greater chance of getting a number below that target hash number. This is where the nonce value comes into play. Miners compete to find a nonce value that produces a hash that is equal or less to the network target hash. This means that not only does your machine have to solve the function but its nonce value means it has to be the randomly chosen one in a group to do so as well. This is because that specific nonce will output a number to be compared to the target hash and if it is below that target hash number then they are selected to make a successful transaction. When a miner finds this winning nonce value it is referred to as the golden nonce. This golden nonce reserves the miner the right to add the block to the chain and receive a reward. All this increasing and decreasing of network difficulty is designed to keep mining from forming a linear relation in profits. An example of this would be someone coming along in Bitcoins infancy stage and quickly mining all the coins before the network grew competitive enough to prevent it. This If more powerful miners were to hit the network there would be a linear relation between the amounts of hash power and how bitcoin much they would get paid out compared to the other miners on the network which would effectively stomp out the little guy and harvest all the coins. Since the network difficulty is used industrial grade miners can’t just quickly mine the whole network because the sudden increase of hashing power would effectively raise the network difficulty. This is the decentralization aimed to stop big business from stomping on the little guy and keep the market competitive although this is not as true to day with the use of powerful miners and large mining pools.

**Bitcoin: Double Sha256**

Bitcoins Proof of work algorithm incorporates a double sha256 implementation for cryptographic hashing. This is because new recorded transactions will be added to a block header as a hashed value using sha-256 as the merkel root. Also in the header will be a hashed representation of the previous block. Now that the header has its initial references from two sha-256 hash’s the merkel root and previous blocks hash, it then takes the block header and applies sha-256 again effectively encapsulating the first two hashed values in the new sha-256 representation of the block header. If the blockheader has enough zeros in its little endiancy then it will be accepted to the ledger. If the hashed value of the block header does not have enough zeros it will be rejected and the nonce value increased to try again. (Vilim, Duwe, & Kumar, 2016. Pg 2).

**Bitcoin: Coinbase Transaction**

The first transaction in the merkel root is called the coinbase transaction and this allows for a miners verified transaction to get a reward from if the block is successfully mined. The Coinbase transaction is different in that instead of transactions having input and outputs, this transaction creates bitcoin as a reward for adding a block. The transaction of the coinbase will contain a block reward of Bitcoins to be sent to the miners address. It uses the recipient’s public key to send the value to when confirmed. Also in the coinbase transaction uses the fields for previous block hash and index but these are not needed since you are creating new bitcoins and there is no index or previous block. It is popular to see arbitrary data or messages hidden in those values from miners since they are not needed or used in the coinbase transaction. Again once the coinbase transaction is completed it is added to the merkel branch to create a merkel tree hashed value to be added to the block header. If that block headers hashed value is less than the target hash rate then the block that include the new coinbase transaction and other transaction are approved (Gjermundred, Chalkias, & Dionysiou, 2016 Pg 2).

**Bitcoin: Halvening.**

Bitcoins proof of work algorithm is designed to only mint 21,000,000 bitcoins in existence. To make acquiring all these coins more difficult a concept of halvening was introduced. This was designed to increase the assets value over time since they supply goes down traditionally increasing demand. When Bitcoin began miners used to get rewarded 50 bitcoins for adding a block. After that it halved to 25 and then to 12.5. The next halvening will make the block reward 6.25 bitcoins every time a block is added from the coinbase transaction. What this aims to do is decrease supply while keeping demand the same or higher. Now miners will only get half of what they used to for the same amount of work. This will in turn weed out unprofitable miners decreasing network difficulty till profitable again or force miners to sell their Bitcoin at a higher rate to make up for the losses. Effectively creating an equilibrium between market forces and mining hash rates.

**Birth of ASIC’s and 51 percent flaw.**

When Satoshi Nakamoto first white papered Bitcoin and the Proof of Work algorithm they had an idea of a decentralized digital ledger that would be inept from being controlled by any one single entity. While it is true that this consensus algorithm is optimized for decentralization it is not without fault, and is not as decentralized as it was designed to be. In the beginning the difficulty of the network was low, but as more and more nodes of miners appeared on the network the difficulty began to rise. This meant that you would have to produce more hashing power to compete on the network. Eventually people stopped using their Central Processing Units for computing hashing functions and started using their graphic processing units optimized for solving these functions. This lead to a higher difficulty on the Bitcoin network. Building from that was something that Satoshi Nakamoto did not mention in the whitepaper or maybe intentionally left out, the fact you could create machines optimized for solving the SHA-256 algorithm made up of many tiny specialized CPU’s on one machine. These machines Are called Application Specific Integrated Chip’s or ASIC for short. ASIC miners bring an unfair competitive advantage to the Bitcoin mining network because they typically produce hashing results exponentially faster than traditional CPU and GPU mining rigs. This is because the ASIC is essentially a machine with many CPU’s optimized for solving this one hashing function. This takes away from Bitcoin being truly decentralized because now everyday people must by industrial grade equipment if they want to compete on the Bitcoin network (Zamanov, Erokhin, & Fedotov, 2018 Pg 1).

This has led to the adoption of mining pools, where groups of miners get together and share hashing power at a chance of breaking a block and splitting the profits. These mining pools have become so large that they jeopardize Bitcoins credibility of being decentralized because they make up such large chunks of the network. This exposes one of the known vulnerabilities in Bitcoins Proof of Work architecture. This flaw is called the 51% vulnerability. The 51% vulnerability essentially means that if 51% of the network agrees on a lie then that lie becomes the truth. The Bitcoin whitepaper did not account for this because the Proof of Work algorithm was intended to be a fair competition with CPU miners but was then taken advantage of with the production of mining pools and ASIC miners. Now theoretically if mining pools were to get together they could all agree on a lie essentially making it true if they composed of 51% more of the market or not. Although other blockchain applications aim to build and solve this issue.

**Birth of Alternate Coins, Decentralized Apps, and Smart Contracts.**

The birth of Bitcoin and the Proof of Work consensus was the breakthrough phase in blockchains technological evolution but soon after that came the replication phase which ultimately led to inovation. Where other blockchain developers aimed at building off the weaknesses of Bitcoin to develop a more advanced blockchain for modern society. This led to the birth of alternate coins called altcoins. From the birth of these alternate coins came an advancement known in blockchain technology as decentralized applications and smart contracts. The technology allows for blockchain to not only be just a store of value but to actually deploy and run applications in a decentralized fashion. From a broad perspective a Decentralized Application that bases transactions of off smart contracts. A smart contract is back end server code that is ran on the decentralized blockchain network. The front end will have libraries that make reference to the back end API’s that can automate applications without a facilitator. For example, traditionally you would need a real estate broker to play the middle man in purchasing your home and acquiring a deed. In this instance if you have the money the deed for the home would be issued to you automatically from the smart contract without the need of a broker collecting fees. This is a broad example and the beauty of how smart contracts can create trust between entities and cut out the middleman in transactions. This was all made possible from a different type of consensus algorithm compared to Bitcoins traditional Proof of Work and it is called the Proof of Stake. The Proof of Stake eliminates using hashing algorithms that are taxing on the system and allow for transactions that can quickly and easily be verified. This is done through a staking architecture where nodes will put something of value at stake. If they are caught lying or doing something fraudulent they will lose their stake and no longer be able to verify transactions. This is in contrast to the Proof of Work using copious amounts of resources just to create a digital store of value. Proof of Stake uses a little bit of resources by instead of working for the transactions, they put money up front to back up miners will verify the transactions correctly (Lee, Jang, & Kim, 2019. Pg 1).  
**Etherium: Overview**:

One major competitor in then decentralized application market is a cryptocurrency called Etherium. Etherium is known for deploying smart contracts on its network using a language called “Solidity”. Solidity smart contracts give programs the fallibility to create secure transactions automatically without a third party verification system. One example of this could be an online slots machine that pays out winnings instantaneously to your cryptocurrency wallet without having to log into the proprietary websites exchange. A broader comparison would be that Proof of Stake is like a vending machine, where Proof of Work is like a cashier. The vending machine being less taxing but an upfront cost, while the cashier is literally working for their pay. This is achieved in Etheriums Proof of Stake consensus through the use of validators. Validators are nodes on the network whose votes weigh in proportion to how much stake they put up. Validators will then earn rewards for successful transactions that are processed and added to the chain. In a typical PoS consensus out of the group of validators a pseudo-randomly selected miner will process the transaction and get a reward. Of course there are known design flaws in Etheriums Proof of Stake design, first being the opposite of Bitcoins 51% network issue. Instead of owning 51% of the network someone can own 51% of the coins that are minted and will be able to create malicious intent on the network. They can do this because if they own 51% of the coins they can stake their 51% percent on the network and control the majority stake. (Lee, Jang, & Kim, 2019. Pg 3). Another exploit is the “Nothing at Stake” exploit where essentially miners can put up their stake one time and be able to stake multiple forks at the same time. So if someone creates a fork of the code and creates their own blockchain similar to the original it can mine on that forked network as well without have to put up any up to stake first. Of course as more designs of Proof of Stake consensus come out they aim to eliminate these flaws.

**Etherium: Sharding**

In Etherium’s Proof of Stake consensus it aims at creating a scalable applications without the loss of security by a concept of sharding. In the Proof of Work consensus all nodes store and process all transactions created many copies of a large digital ledger. This is provides a great amount of security but does not leave a lot of room for scalability. Sharding aims at grouping together nodes of the blockchain to validate transactions in independent groups. Over all increasing the blockchains throughput. A very simplified mathematical example of this would be if you had a PoW chain that has 100 nodes verifying every transaction a second versus a PoS consensus using ten groups of ten nodes solving ten transactions simultaneously a second. This is an example of how giving up security can increase throughput and scalability. (Lee, Jang, & Kim, 2019. Pg 1). The design of sharding uses validators to added information to the main chain. Typically there will exsist validators who are pseudo-randomly chosen staking nodes. Out of some of these validators nodes some will be randomly chosen to validate shards. One of the chosen group will be the catalyst randomly chosen to trigger the transaction adding the shard to the main chain. This will only be triggered if the remaining specially chosen validators in the shard agree in two thirds majority with shard validator sharing the shard data back to the main beacon chain.

**Neo: Delegate Byzantine Fault Tolerance as Practical Byzantine Fault Tolerance.**

Now that we have discussed Etheriums Proof of Stake we will discuss another flavor of Proof of Stake used by a blockchain company called Neo. Neo uses an implementation of PoS via its consensus deemed the Delegate Byzantine Fault Tolerance. The Point of a DBFT is that the blockchain should solve the byzantine general problem by continuing on and be resilient to issues brought up in the BGP. This is done by in the following process. Owners of the coins make transactions and are stored on book keeping nodes called delegates. These delegates verify the transactions and out of them a speaker is chosen. The speaker then verifies the transactions given by the delegates and compares them to the current hash rate for a new proposed hash rate. If the delegates agree with the speaker’s transactions and new proposed hashed rate at a two thirds majority then the transaction is added to the main chain. According to the Neo documentation the the speaker of the round is responsible for a broadcast called a prepare-request to initiate a newly proposed block. After getting the response the Speaker creates a prepared proposal block and if verified by 2/3 majority it will then move on to the Persisting stage where it will publish a new block and enter the next round of consensus. If a consensus is not reached or agreed upon then the nodes will initiate a change view proposal This will enter a new view with a new speaker and restart the consensus. This is an example of how Neo’s delegate byzantine fault tolerance performs byzantine fault tolerance by view changing of speakers and consensus creating redundancy if a block were to fail. (Tong & Yan, 2019, Pg 1).

**Neo: Unspent Transaction Output**

Prior to Proof of Stake consensus there was the traditional account balance model used in Bitcoins Proof of Work. The transactions were verified on the chain by directly recording each accounts assets. (Tong & Yan, 2019, Pg 1). Neo is widely popular because you can easily earn Gas for processing transactions on the Neo network from staking. To execute a transaction a fee in Gas must be paid. For this example we will show how unspent transaction output is used in Neo’s transaction asset Gas. Let’s say User A earns ten gas, they then send five gas to User B and then five gas to User C. The first transaction will see User A have an output of ten gas and be the original transaction index since this is where the gas was created. The Second transaction will see User A now only have an output of five gas while User B will now have an output of five gas as well, this will be the second transaction index from User A’s initial creation of gas in the first index. Lastly, in the third transaction you will no longer see User A in the outputs because they have used up their unused outputs. The remaining outputs are user B’s and user C’s five Gas waiting to be spent. This is a broad over view of how unspent transaction output is used in Neo’s Byzantine fault tolerant network.

**Conclusion.**

In conclusion the emergence of blockchain consensus has been an interesting topic for both research and debate. Covering topics of consensus in Proof of Work, Proof of Stake, and Byzantine Fault Tolerance shows how they aim to solve the Byzantine General Problem. The emergence of these technologies display the complexity and the ingenuity blockchain has brought to the market. As blockchain continues to scale, adapt, and be regulated it will find its niche in the growing market. Different applications will use different consensuses to specify their needs.

**References:**

Bagaria, V., Fanti, G., Viswanath, P., Tse, D., & Kannan, S. (2019, November 15). *Deconstructing the Blockchain to Approach Physical Limits.* doi: 10.1145/3319535.3363213

Duy, P. T., Hien, D. T. T., Hien, D. H., & Pham, V.-H. (2018). *Proceedings of the Ninth International Symposium on Information and Communication Technology*. New York NY: ACM. doi: 10.1145/3287921.3287978

Gjermundred, H., Chalkias, K., & Dionysiou, L. (2016, November 12). *Going Beyond the Coinbase Transaction Fee: Alternative Reward Schemes for Miners in Blockchain Systems*. doi: 10.1145/3003733.3003773

Lee, D., Jang, Y., & Kim, H. (2019, November 15).*A Proof-of-Stake (PoS) Blockchain Protocol using Fair and Dynamic Sharding Management.* doi:10.1145/3319535.3363254

Tong, C., & Yan, M. (2019, November 13). Neo Documentation UTXO. Retrieved November 24, 2019, from <https://docs.neo.org/docs/en-us/tooldev/concept/blockchain/utxo.html>.

Vilim, M., Duwe, H., & Kumar, R. (2016, June 9).*Approximate Bitcoin Mining* doi:10.1145/2897937.2897988

Zamanov, A., Erokhin, V., & Fedotov, P. (2018, February 1). *ASIC-resistant hash functions* doi: 10.1109/EIConRus.2018.8317115

Zhang, R., Xue, R., & Liu, L. (2019). *Security and Privacy on Blockchain*. New York, NY: ACM New York. doi: 10.1145/3316481

Zohar, A. (2019). *Recent trends in decentralized cryptocurrencies.* (49th ed.). New York, NY: ACM New York. doi: 10.1145/3055399.3079074

**Certification of Authorship**



Submitted to: Professor Wei Li

Student’s Name: Eric Webb

Date of Assignment: 12/8/2019

Title of Assignment: Final Paper

Certification of Authorship: I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas, or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for this course.

Student's Signature: ERIC WEBB