BLOCKASTICS

Stochastic models for blockchain analysis

Pierre-O Goffard

March 28, 2022

Chapter 1

Introduction

A blockchain is a distributed ledger made of a sequence of blocks maintained by achieving consensus among a number of nodes in a Peer-to-Peer network. The blockchain technology has attracted a lot of interest after the advent of the bitcoin cryptocurrency in 2008, see Nakamoto [2008]. Since then, the blockchain concept has been used to develop decentralized systems to store and maintain the integrity of time-stamped transaction data across peer-to-peer networks. Besides the creation of a digital currency, blockchain applications include the sharing of IT resources, the registration of authentication certificate or the implementation of smart contracts.

A blockchain is

- Decentralized as it is maintained by a network. Nodes can be light or full nodes. Light nodes are blockchain users that broadcast transactions, full nodes are in charge of verifying and recording the transactions, see Figure 1.1.

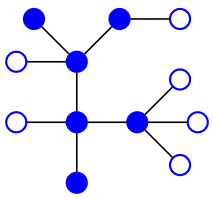


Figure 1.1: A network made of full nodes (blue) and light nodes (white)

  - A local copy is stored by each full node which grants security
  - The governance is not handled by a central authority

- Public or private. In public blockchain anyone can access the data, in private blockchain reading access is restricted.
- permissionned or permissionless. In permissionless blockchain, anyone can join the network as a full node.

- Immutable. Altering the information written in the blockchain is made difficult if not impossible.
- Incentive compatible. The process of reaching consensus is costly to the full nodes who must be compensated for their hard work.

The consensus protocols, at the core of the blockchain technologies, are the focus of these lecture notes. The goal is to evaluate consensus protocol according to three dimensions

1. Efficiency: The amount of data being processed per time unit
2. Decentralization: The fairness of the distribution of the decision power among the nodes
3. Security: The likelihood of a successful attack on the blockchain

Because consensus protocols involve random components, stochastic modelling is required to assess a blockchain system within the Efficiency/Decentralization/Security trilemma in Figure 1.2. As it is hard to improve one dimension without negatively impacting the other two, trade-offs

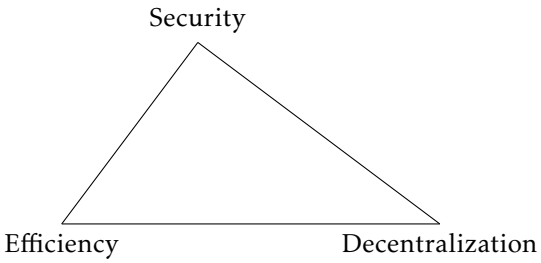


Figure 1.2: The blockchain trilemma

must be made. We will see how to use classical models of applied probability, including urn, epidemic, graph, queue and risk models, to provide numerically tractable indicators to quantify the efficiency, decentralization and security of blockchain systems. These indicators will then allow us to carry out sensitivity analysis with respect to the model parameters to optimize and improve blockchain implementations.

The main application of blockchain systems today is undoubtedly cryptocurrencies, the most well known of which being the bitcoin introduced by Nakamoto [2008]. Public and permissionless blockchain, like the bitcoin one, must be associated to a cryptocurrency. Indeed, to add a block to the bitcoin blockchains the full nodes compete to solve a cryptrographic puzzle using brute force search algorithm. The first node (referred to as a miner) who finds a solution, appends the next block and collects a reward expressed in cryptocurrency. Assuming this reward is worth something, it offsets the operational cost which is essentially the electricity consumed to run the computers 24/7. A cryptocurrency must be equipped with following features

1. No central authority (Decentralized network)

2. Ledger to record all the transactions and coin ownership (the blockchain)
3. A coin generation process (block finding reward)

   ↪ It creates an incentive compatible system to the full nodes
4. Ownership can be proved cryptographically, a wallet is secured with a public/private key system
5. Transactions can be issued by an entity proving ownership of the cryptographic unit through the private key
6. The system cannot process more than one transaction associated to the same cryptographic unit. It must be robust to double spending attack in which a fraudster is issuing two conflicting transactions to recover the funds she already spent

This charcaterization is given by Lansky [2018]. Cryptocurrencies draw their fundamental value from the fact that they

- provide transaction anonymity
- provide a reliable currency in certain regions of the world
- permit money transfer worldwide at low fare
- do not require a thrusted third party

An important implication of this architecture is disintermediation, it creates an environment where multiple parties can interact directly and transparently. Blockchain is therefore immediately relevant to banks and financial institutions which incur huge middlemen costs in settlements and other back office operations. Decentralized finance (DeFi) offers a new financial architecture that is non-custodial, permissionless, openly auditable, pseudo-anonymous and with potential new capital efficiencies. It extends the promise of the original bitcoin whitepaper Nakamoto [2008] of non-custodial transaction to more complex financial operations, see the SoK of Werner et al. [2021].

Blockchain is a research topic of interest to many communities. Computing science distributed ledger technologies (synonymous with blockchains) rely on distributed algorithms and enable cooperation within a peer-to-peer network. Linking blocks and checking the authenticity of data uses cryptographic functions which is another field of computer science. The establishment of an incentive system within a network of individuals adopting a strategic behavior naturally leads to problems of game theory similar to those solved by economists. The discussion on the nature of new financial assets such as crypto-currencies, utility tokens and non-fungible tokens, is also at the center of the concerns of researchers in finance and monetary economics.

We focus here on the use of mathematics to optimize blockchain systems which makes our problems very close to those encountered in operations research. These notes are organized as follows. Chapter 2 presents the various consensus algorithms. Chapter 3 focuses on the security aspects. In Chapter 3, we take a look at decentralization in Chapter 4. We close on efficiency with Chapter 5.

The topic of blockchain is of primary interest to computer scientists working on peer-to-peer networks and distributed algorithm. The problem of reaching consensus inside peer-to-peer networks is a classical problem in computer science. A consensus protocol just take advantage of the limited resources of the network which includes

- bandwidth
- computational power
- storage

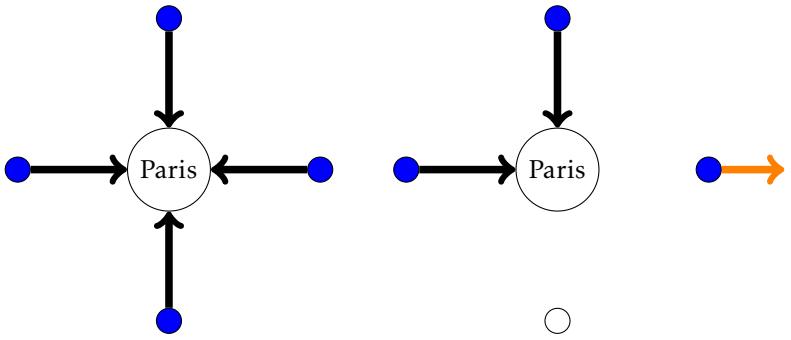
The most natural solution is to proceed to a majority vote via a system of message exchange. It was proposed a long time ago to solve "The Byzantine general problem" as framed by Lamport et al. [1982] in an abstract way. The issue is that exchanging messages inside a peer-to-peer network that can grow very large is not a practical solution. The colossal number of messages is prohibitive, it leads to communication overhead and the failure of some nodes by denial of service.

The goal is to find an algorithm which allows the node of the networks to agree despite the presence of crashing nodes and faulty nodes (also referred to as Byzantine node)

A group of generals from the Byzantine army is surrounding an enemy city. Communicating only by messenger, they must agree on a common battle plan. There may be traitors who will attack instead of retreat or non responding generals who will do nothing. For the project to be successful a majority of the general must either retreat or attack. The problem then reduces to finding an algorithm to ensure that the loyal generals reach an agreement. The problem is illustrated on Figure 1.3.

In a blockchain system, we have a large network of nodes that broadcast transactions which corresponds to pieces of information that will be written in the blockchain. light nodes, full nodes consensus and write information.

- Computer science
- Economics
- Applied math and operations research



(a) Coordinated attack

(b) Failed attack due to one traitor and one non-responding general

Figure 1.3: Illustration of the BYzantine general problem

Chapter 2

Consensus protocol

Transactions flow through the network of full nodes. After reviewing them, the full nodes must agree on the transaction that will be recorded in the next block. to do this, an algorithm must be designed so that consensus is reached. A consensus protocol must be based on one of the scarce resources available to the network peers which include

- bandwidth
- computational power
- storage

The first solution that comes to mind for reaching consensus is a majority vote based on a message exchange system. This solution has been proposed by Lamport et al. [1982] within the famous "Byzantine general problem". A voting system inside a large network involves a colossal number of messages exchanged leading to the consumption of all the bandwidth, the failure of some nodes by denial of service and delays in the synchronization of the network. Practical solution like the celebrated Practical Byzantine Fault Tolerance (PBFT) presented in Göbel et al. [2016] have been implemented in some blockchain systems. Despite these advances, a change in methods was needed to accommodate a network that could grow indefinitely.

Nakamoto [2008] solved this scaling problem by proposing a system based on the election of a leader. The Proof-of-Work (PoW) protocol appoints a leader based on its computing resources. Each node competes to solve a puzzle with a brute force search algorithm. The first node who is able to propose a solution append the next block. The search for a solution, referred to as mining, is associated with an operational cost borne by the nodes which is compensated by a reward expressed in the native blockchain cryptocurrency. The surge in cryptocurrency prices has led to a rush in block mining, leading to a major spike in the electricity consumption and electronic waste generation of blockchain networks. The blockchain network consumes as much electricity as countries the size of Thailand at the time of the writing. The need for a more environmentally

friendly consensus protocol therefore becomes pivotal. The use of data storage has been implemented within the Filecoin project of Protocol Labs [2017] via the Proof-of-Space (PoSp) and its variant like the Proof-of-Spacetime protocol. A leader is chosen depending on how much data she currently stored or for how long some data has been stored. Proof-of-Space (PoSp) is seen as a fairer and greener alternative by blockchain enthusiasts due to the general purpose nature of storage and the lower energy cost required by storage. The fact that most storage resources are owned by companies offering cloud storage solution poses a threat to the decentralized nature of the distributed ledger. The Proof-of-Interaction (PoI) protocol, proposed by Abegg et al. [2021], takes as leader the first node that is able to contact and obtain a response from a random sequence of nodes. This is a bandwidth-based alternative that is more scalable than majority voting. Along with bandwidth, computing power, and storage, a new resource has emerged with the advent of cryptocurrencies as a medium of exchange. The Proof-of-Stake protocol, described by Saleh [2020], selects a node with a probability proportional to the number of cryptocoins it holds.

This chapter is organized as follows. Section 2.1 gives a brief description of the voting based ways to get consensus by reviewing the "generals" problem. Section 2.2 goes through the leader based consensus protocols, including PoW in Section 2.2.1, PoSp in Section 2.2.2, PoI in Section 2.2.3, and PoS in Section 2.2.4.

2.1 Voting system

The problem of reaching consensus in a peer-to-peer network via a majority vote has been abstractedly compared to generals who must agree on a common battle plan. We start from the simple two general case before moving on the the situation of interest with several ones.

2.1.1 Two generals problem

Two generals wish to attack a city but they must agree on a timing to attack a city. They communicate via a messenger who must cross enemy territory at the risk of being intercepted. The first general G_1 sends a message to the second one G_2 saying

```
"I will attack tomorrow at dawn"
```

For the attack to succeed, both generals must attack at the same time. Because their communication medium is unreliable, then G_1 must await confirmation from G_2 in order to attack. If G_1 does not receive confirmation then she will not attack. G_2 is aware of that and respond

```
"I will follow your lead"
```

G_2 does not know whether the message went through and must wait for confirmation. This creates an infinite loop of messages and response, as on Figure 2.1. The two general problem is deemed unsolvable from a theoretical point of view and corresponds to a situation where two

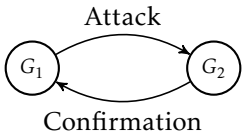


Figure 2.1: Message and confirmation loop

nodes communicate through an unreliable link. A practical solution for generals is to send many messengers hoping that at least one of them will succeed. This is only a thought experiment leading to the several general problem.

2.1.2 Byzantine General problem

The blockchain network contains more than two nodes, these nodes must agree on the transactions to confirm. In a permissionless blockchain the nodes do not trust each other. The problem of the previous section generalizes to more than two generals, assuming that some generals are traitors which corresponds to faulty nodes in the network. This problem is referred to as The "Byzantine general problem" and was coined by Lamport et al. [1982]. Assume that $n > 2$ generals must agree on a common battle plan for instance "Attack" (A) or "Retreat" (R) and that they can only communicate by two party messages. Denote by $m(i,j)$ the message sent by general i to general j. Each general j receives $n-1$ messages and applies a function f to determine the course of action, for instance

$$f(\{m(i,j);\ i=1,\dots,n\}) = \begin{cases} A, & \text{if } \sum_{i=1}^{n} \mathbb{I}_{m(i,j)=A} > n/2, \\ R, & \text{else.} \end{cases}$$

If there are no traitors, each general is communicating the same value to all the peers and consensus is reached as in Figure 2.2a. If one general is traitor, then he might not communicate the same value to all the generals and no consensus can be reached. It is the case for G_4 in Figure 2.2b. To handle such a situation, roles are given to the general. One of them become the leader and the other are the lieutenants. We aim at finding an algorithm such that

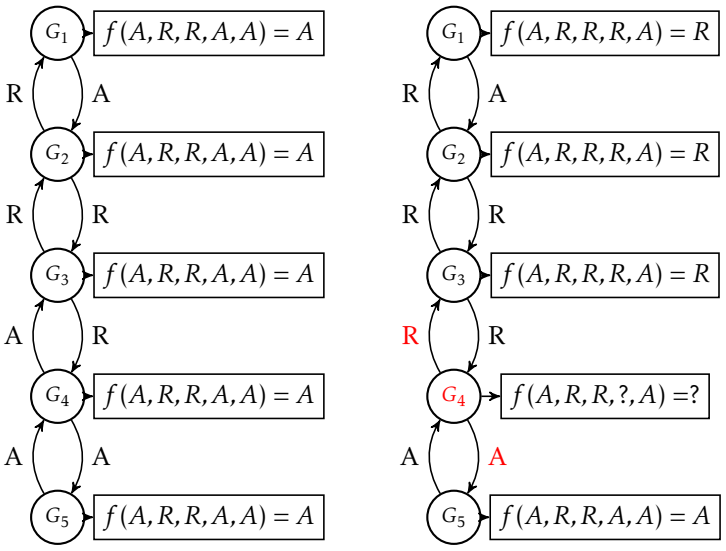
C1 All the loyal lieutenants obey the same order

C2 If the commanding general is loyal, then every loyal lieutenants obey the order he sends

A first result from Lamport et al. [1982] is the following

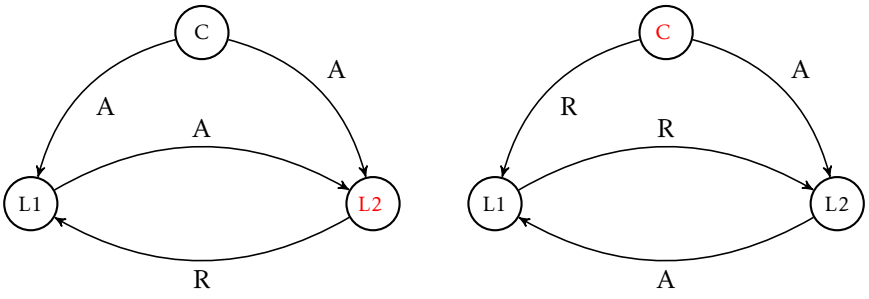
Theorem 1. *There are no solution to the Byzantine General problem for $n < 3m+1$ generals where m is the number of traitors.*

Proof. Consider the situation where $n = 3$ and $m = 1$. The traitor is either the commander or one of the lieutenants as shown in Unfortunately for Lieutenant 2, there is no way for her to tell apart the situation pictured in Figure 2.3a and Figure 2.3b and therefore no way to ensure both C1 and C2. We prove the result for $n > 3$ by contradiction. Assume that there is a way to verify



(a) No traitor (b) One traitor

Figure 2.2: Majority vote with or without a traitor



(a) Commander is loyal (b) Commander is a traitor

Figure 2.3: Majority vote with or without a traitor

both C1 and C2 with $3 < n < 3m + 1$. We then construct a solution with generals by having one general simulate the commander plus at most $m - 1$ generals, and the other two simulating at most m generals. One of the generals gather all the traitors and is therefore a traitor. The other two are loyal generals as they only simulate loyals general. We have built a solution with three generals that we know is impossible. □

Now we need an algorithm that allows $n > 3m + 1$ generals to deal with m traitors. We denote it by $OM(m)$, where OM stands for "Oral Messages".

2.2 Leader system

- Public blockchain,
- operational cost,

Algorithm 1 OM(m)

Initialize $m = 0$;
for $i = 1 \rightarrow n - 1$ **do**
    Commander sends $v_i = v$ to lieutenant i
    Lieutenant i set their value to v
end for
Let $m > 0$;
while $\pi_s(\theta) \neq \pi(\theta|\mathbf{x})$ **do**
    Search for π_{s+1} such that

$$\frac{1}{\sum_{i=1}^{N}(W_i^{s+1})^2} \geq \rho N, \text{ with } W_i^{s+1} \propto w_i^{s+1} = \pi_{s+1}(\theta_i^s)/\pi_s(\theta_i^s),\ i = 1,\ldots,K$$

    Compute $\widehat{\Sigma} = \mathrm{Cov}\left(\{(W_i^{s+1}, \theta_i^s),\ i = 1,\ldots,K\}\right)$
    for $i = 1 \rightarrow K$ **do**
        Sample $\tilde{\theta}_i \sim \{\theta_1^{(s)},\ldots,\theta_K^{(s)}\}$ with probabilities W_j^{s+1}, for $1 \leq j \leq K$
    end for
    for $i = 1 \rightarrow K$ **do**
        $\tilde{\theta}_i^* \leftarrow K_H(\tilde{\theta}_i, \cdot)$ where $K_H(\tilde{\theta}_i, \cdot)$ where $H = \frac{2.38}{\sqrt{d}} \cdot \widehat{\Sigma}$
    end for
    Compute $p_a = N^{-1} \sum_{i=1}^{K} \mathbb{I}_{\tilde{\theta}_i^* = \tilde{\theta}_i}$; $k = \max\left\{k_{\max}, \min\left[k_{\min}, \frac{\log(1-c)}{\log(1-p_a)}\right]\right\}$
    for $i = 1 \rightarrow K$ **do**
        $\theta_i^{s+1} \leftarrow K_H^{*(k-1)}(\tilde{\theta}_i^*, \cdot)$ where $K_H^{*(k-1)}(\tilde{\theta}_i^*, \cdot)$ corresponds to $k-1$ Metropolis-Hasting-Gibbs moves
        $W_i^{s+1} \leftarrow 1/K$
    end for
end while
Return $(W_1^t, \theta_1^t),\ldots,(W_K^t, \theta_N^t)$

- reward,
- incentive compatible
- Uses the scarce ressource of the network
  - Computational power (CPU, GPU)
  - Bandwidth
  - Storage space
  - Crypto coins

2.2.1 Proof-of-Work

2.2.2 Proof-of-SpaceTime

2.2.3 Proof-of-Interaction

2.2.4 Proof-of-Stake

Chapter 3

Security of blockchain systems

3.1 Double-spending in PoW

3.1.1 Random walk model

Double spending probability

Double spending time

3.1.2 Counting process model

Double spending probability

Double spending time

3.2 Blockwitholding in PoW

3.3 Nothing-at-stake in PoS

Chapter 4

Decentralization of blockchain system

4.1 Decentralization in PoS

Rich get richer? Polya's urn

4.1.1 Average stake own by each peer

4.1.2 Distribution of the stakes

4.2 Decentralization in PoW

4.2.1 Mining pools and reward systems

4.2.2 Mining pool risk analysis

Chapter 5

Efficiency of blockchain systems

5.1 A queueing model with bulk service

5.2 Latency and throughputs computation

Bibliography

Jean-Philippe Abegg, Quentin Bramas, and Thomas Noël. Blockchain using proof-of-interaction. In *Networked Systems*, pages 129–143. Springer International Publishing, 2021. doi: 10.1007/978-3-030-91014-3_9.

J. Göbel, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor. Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay. *Performance Evaluation*, 104:23–41, 2016.

Protocol Labs. Filecoin: A decentralized storage network. *White paper*, 2017. https://filecoin.io/filecoin.pdf.

Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, pages 382–401, July 1982. URL https://www.microsoft.com/en-us/research/publication/byzantine-generals-problem/.

Jan Lansky. Possible state approaches to cryptocurrencies. *Journal of Systems Integration*, 9(1): 19–31, jan 2018. doi: 10.20470/jsi.v9i1.335.

S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Available at https://bitcoin.org/bitcoin.pdf, 2008. URL https://bitcoin.org/bitcoin.pdf.

Fahad Saleh. Blockchain without waste: Proof-of-stake. *The Review of Financial Studies*, 34(3): 1156–1190, jul 2020. doi: 10.1093/rfs/hhaa075.

Sam M. Werner, Daniel Perez, Lewis Gudgeon, Ariah Klages-Mundt, Dominik Harz, and William J. Knottenbelt. Sok: Decentralized finance (defi), 2021.