EE 374: Fundamentals of Blockchain Infrastructure **Stanford, Spring 2025**

Lecture 1: Introduction and Course Overview

March 31, 2025

Lecturer: David Tse Scribe: Trevor Golob

1 Introduction

In today's lecture we will go over the course logistics and discuss the context in which Bitcoin was developed. We will discuss two key aspects of Bitcoin, namely, proof-of-work and the longest chain protocol, their drawbacks and the efforts to improve on them.

2 Logistics

- **Instructor:** David Tse
- **Lectures:** 3:00 – 4:20 pm, Mon/Wed (Room 540/108)
- **David's Office Hours:** 4:30 – 5:30 pm, Mon/Wed (Room 264 Packard)
- **Nusret's Office Hours:** 4:30 pm, Thurs
- **Website:** `https://web.stanford.edu/class/ee374/index.html`
- **Discussion Section:** 4:30 – 5:30 pm, Tuesday (beginning week 2)
- **Communication Tools:** Piazza for questions; Gradescope for submissions
- **Prerequisites:** Basic probability (e.g., CS109, EE178)

3 Course Requirements & Grading

- **Lecture Scribe + Tweet:** 10%
- **Homeworks:** 4 or 5 total, due Friday 5 pm (20%)
- **Midterm:** 24-hour take-home exam (35%)
- **Project:** Poster + report, with a poster presentation on June 4th (35%)

In the project, students will analyse an existing blockchain protocol. They will then suggest improvements to this protocol given the class material.

4 Course Description and Outline

The vision of blockchains is to allow billions of people to interact with minimal trust of third parties. Since the invention of Bitcoin by Satoshi Nakamoto in 2008, much innovative infrastructure has been built to fulfill this vision. This course is a rigorous treatment of the fundamental concepts behind these innovations. A particular focus is on the problem of distributed consensus and how to make it permission-less, secure and scalable.

The course is divided into 3 parts:

1. Bitcoin as a payment system and as a consensus protocol. Security analysis. Dynamic availability.

2. Proof-of stake protocols. BFT consensus. Accountability.

3. Scaling blockchains. Data availability. Zero-knowledge and optimistic rollups. Security sharing and re-staking.

5 Motivation for Bitcoin

In 2009, a white paper appeared in the P2P Foundation forum with the name 'Bitcoin: A Peer-to-Peer Electronic Cash System' by an author named Satoshi Nakamoto; this was assumed to be a pseudonym, and the author's true identity remains unknown [2]. Despite this, his paper went on to start an entire field. The paper was about digital currency and how one could implement one. It would be different from fiat currencies managed and legitimated by governments; instead of relying on a centralized authority, the electronic cash system would be maintained by decentralized nodes. At the time of the release of the paper, 2008/2009, the United States was in the middle of a financial crisis which caused many to question the trust placed in the government to maintain currency.

$$\begin{gathered}\text{Centralized authority} \rightarrow \text{decentralized nodes}\\ \text{Centralized trust} \rightarrow \text{decentralized trust}\end{gathered} \tag{1}$$

6 Relevant Definitions

Bitcoin was not only an innovation in applications (digital currency), but an innovation in technology (consensus protocol). Here are two useful definitions discussed in class.

Definition 6.1 (Consensus Protocol)**.** The protocol used by a set of distributed participants use to come to agreement on state or a ledger.

Definition 6.2 (Ledger)**.** An ordered sequence of events. In the case of Bitcoin, the events are transactions. An example Bitcoin transaction is 'Alice Pays Bob 25 BTC'.

Observation: Provided there is consensus on the start date of the ledger, the total ownership of all tokens on the network can be determined by the ledger.

7 Bitcoin as a Consensus Protocol

While Bitcoin may have been one of the first digital currencies, it was certainly not the first consensus protocol. Much work had been done on the subject; albeit, in a different context, the context of data centers. For instance, one of the earliest works in the consensus literature, ‘The Byzantine Generals Problem’ by Leslie Lamport discusses a consensus protocol, where the goal is to prevent the failure of one component from taking down the entire system. [1] However, the consensus protocols Lamport and his descendants worked on were built for *permissioned* systems like data centers. In contrast, Nakamoto built his protocol in the *permissionless* context, meaning anyone could interact with the network and join as a consensus participant.

Before we further compare the two protocols, we recap the two failure types discussed in the lecture: *crash faults* and *Byzantine faults.*

A *crash fault* is a type of system failure that occurs when a node stops operating. These can occur for a multitude of reasons including power outages, network failures or cyber attacks. In the colorful analogy of Byzantine generals preparing to attack a city, a crash fault would be akin to a general getting too drunk and falling asleep only to wake up after the attack has failed. Maybe his sabotage is not malicious, but it results in failure none-the-less.

A *Byzantine fault* is a system failure where the node could not only stop operating but behave maliciously towards the system. Returning to the analogy from before, if one of the attacking generals told another general to attack the day before, thus leading to his destruction, this would be considered a *Byzantine fault.*

Although Lamport’s original work does consider malicious actors, as his work is in the permissioned setting (e.g., a data center setting), Byzantine failures were not the primary concern, as nodes would have to be vetted to join the network. In contrast, since Nakamoto was building a digital currency, Bitcoin as a consensus protocol must be resistant to malicious actors, as people could be financially incentivized to attack the network. Thus, Nakamoto’s application is the first use case of consensus protocols, where the full strength of adversaries *has to be* considered.

8 Two Key Features of Bitcoin

Two key features of Bitcoin are the use of proof-of-work and the longest chain protocol.

Proof-of-work, originally posed as a solution to email spam, provides resistance to Sybil attacks. A Sybil attack is an attack on a computer network, where an adversary creates multiple pseudonymous identities to gain an advantage over other nodes in the network. proof-of-work prevents this by requiring each node to solve a cryptographic problem in order to contribute to the chain of events. This requires CPU power, which makes Sybil attacks economically challenging to execute. One way to think about this is a ‘one CPU, one vote’.

The longest chain protocol is a way for blocks to determine which chain to accept as ‘real’ in the event of a split. It instructs the nodes to continue to build onto the chain with the most blocks and thus the most compute power. Therefore, nodes can ensure that they continue to validate the chain with the most ‘votes’ attesting that it is the correct chain.

Given the two features above, we gain an interesting security property: Bitcoin is secure if $> 50\%$ of compute power is honest.

9 Drawbacks

The drawbacks associated with Nakamoto's approach disscussed in class were.

1. Energy Consumption
2. Low Throughput
3. Confirmation Latency

The *energy consumption* problem is a result of proof-of-work. Since the nodes must constantly solve cryptographic puzzles to validate transactions, people will be dedicating compute power and energy to the network. *Low throughput* comes from the network's decentralization. Transaction rates on the Bitcoin network are in part limited by block size. Throughput could be increased by increasing the block size, but doing so would require more resources to store and validate each block. This could reduce the number of full nodes, potentially compromising the network's decentralization. The transaction rate on the Bitcoin network is roughly 7 tx/s, which when compared to that of popular credit card networks is far slower. VISA, for example, has a transaction rate of around 65,000 tx/s. The final drawback discussed is confirmation latency. The probability that a chain containing a specific transaction is reversed decays over time but starts quite high. Thus, one has to wait until 'confirming' that the payment has been processed. Something interesting to note is that the aforementioned probability, namely that of a transaction being invalidated, is never zero.

10 Solution to Drawbacks

The solution to the energy problem is rather ingenious. There are other resources that can be used as a basis for 'voting' in the network. The tokens themselves can be used to do this! This is known as proof-of-stake (PoS) and is a method used by newer digital currency networks. This method also has a side benefit of solving the problem of confirmation latency.

$$\begin{gathered}\text{proof-of-work} \rightarrow \text{proof-of-stake} \\ \text{One CPU, one vote} \rightarrow \text{one coin, one vote}\end{gathered} \tag{2}$$

Examples of proof-of-stake protocols are Ethereum and Tendermint. PoS as a side benefit can also solve the problem of confirmation latency. The throughput problem will be covered in future lectures.

References

[1] R. S. Leslie Lamport and M. Pease. The byzantine generals problem. *SRI International*, 1982.

[2] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Self-published*, 2008.