Coursework Presentation

Oladotun Aluko

May 7, 2020

Outline

- Introduction and Background
- Traditional Consensus Mechanisms
- Exploring Proof-of-Reputation
- References

Introduction and Background

Background

- A fundamental problem with distributed systems and multi-agent systems is how to achieve overall system reliability in the presence of a number of faulty processes
- Consensus mechanism enable consensus to be reached regarding a shared state. This notion of a shared state has been generalized more into a concept known as **State Machine Replication (SMR)** [1]
- If all the participating nodes receive the same set of input messages in the exact same order then we have **Atomic Broadcast**

Background(cont.)

- Two crucial requirements to reach and maintain consensus among distributed nodes:
  - Deterministic state machine
  - Consensus protocol to disseminate inputs in a timely fashion. This translates into 4 properties:
    1. Validity
    2. Integrity
    3. Agreement
    4. Total Order

Background(cont.)

- Also, there are two sets of assumptions under which consensus protocols will function properly
    - Underlying Network Type: Synchronous, Asynchronous and Partially/Eventually Synchronous [2]
    - Properties of the consensus protocols: Consistency, Availability, and Fault Tolerance [3]
- In addition, there are two major fault-tolerance models within distributed systems
    - Crash failure (or tolerance)
    - Byzantine failure

Background(cont.)

- A consensus mechanism has four major groups of properties:
    - Structural properties
    - Block and reward properties
    - **Security properties** - Authentication, Attack Vector
    - **Performance properties** - Fault Tolerance, Throughput, Scalability, Latency, Energy Consumption [4]

Traditional Consensus Mechanisms

Proof-of-Work

- The idea of PoW was first presented in 1993 as a solution to email spamming
- A Proof-of-Work (PoW) mechanism involves two different parties (nodes): **prover** and **verifier**. The prover performs a resource-intensive computational task intending to achieve a goal and presents it to a verifier or a set of verifiers for validation that requires significantly less resource
- Limitations of PoW include:
    - Energy Consumption
    - Absence of penalty

Proof-of-Stake

- The core idea of PoS evolves around the concept that the nodes who would like to participate in the block creation process must prove that they own a certain number of coins at first
- Limitations of PoS include:
  - Collusion
  - Wealth Effect

Proof-of-Authority

- Proof-of-Authority (PoA) is a new family of BFT algorithms that has recently drawn attention due to the offered performance and toleration of faults
- It is currently used by Parity and Geth, two well-recognized clients for permissioned setting of Ethereum
- Still relatively new and it has not been rigorously tested
- Vulnerable to the Cloning Attack

Exploring Proof-of-Reputation

Proof-of-Reputation: Introduction

- Reputation can be defined as the rating of a member’s trustworthiness by others which can be managed centrally or decentralized
- Reputation serves as the incentive because, in the participant can write a block into the blockchain when it has the highest trust value in this block [5]

Proof-of-Reputation: Methodolody

- The protocol assumes three conditions:
    - Enrolment Control
    - Secure communication channel
    - Quick Bootstrap
- Design Overview for the protocol
    - Broadcasting Transaction
    - Building Blocks
    - Verifying Block

Proof-of-Reputation: Methodolody

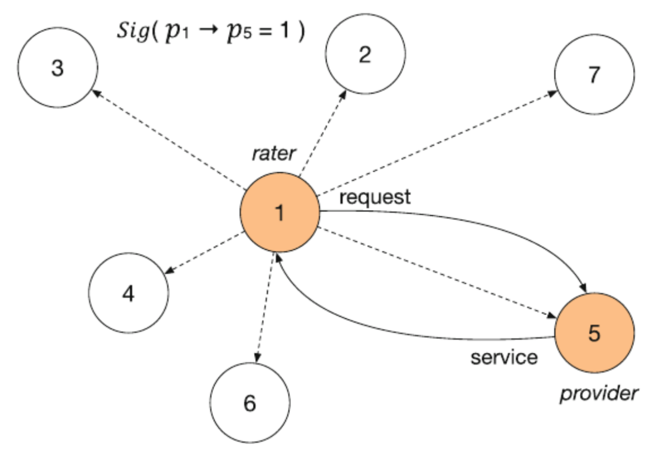

Figure 1: Broadcasting transactions step of p1 rating the service of p5

Proof-of-Reputation: Advantages

- Offers some advantages over traditional consensus:
  - There are no complex mathematical problems to be solved, which means the protocol is cost-efficient
  - No need to worry about the double-spending problem because reputation is an overall status of a node after a number of transactions, which can not be spent or transferred

Proof-of-Reputation: Experiments and Evaluation

Performance Evaluation

- Scalability
- Production Time
- Throughput

Proof-of-Reputation: Experiments and Evaluation

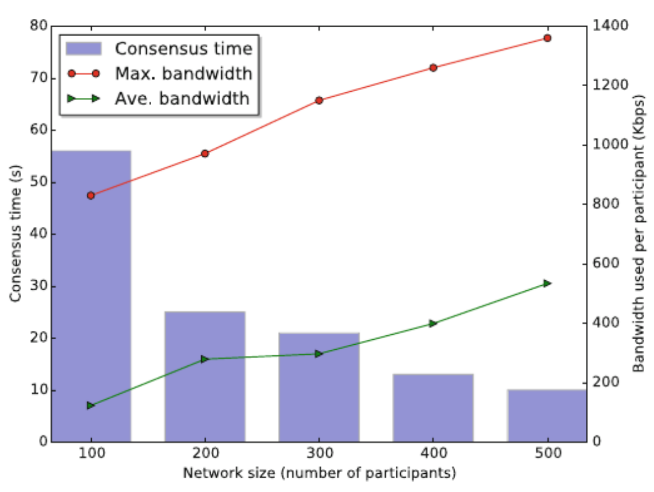

Figure 2: Consensus time and bandwidth of PoR with different network sizes

Proof-of-Reputation: Experiments and Evaluation

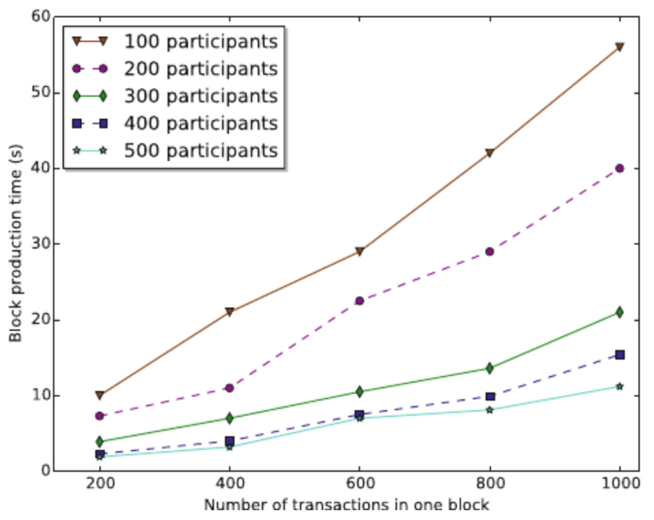

Figure 3: Average time to produce a block with different block sizes

Proof-of-Reputation: Experiments and Evaluation

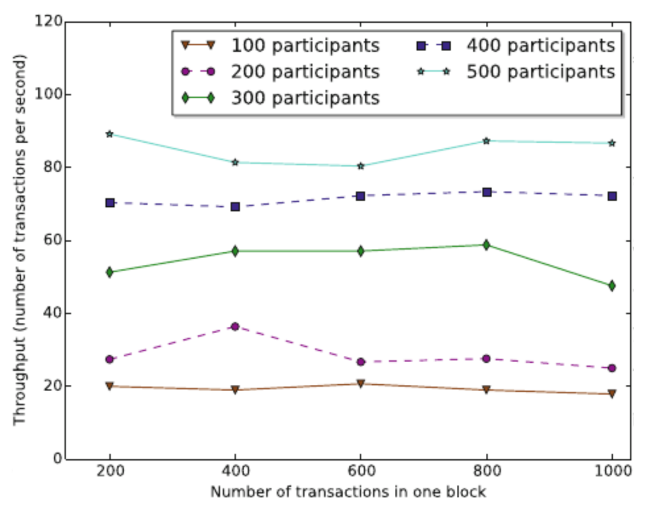

Figure 4: Throughput with different block sizes

Proof-of-Reputation: Experiments and Evaluation

Security Evaluation

- Bad-mouthing attack
- On-off attack
- Newcomer attack

- F. B. Schneider, “Implementing fault-tolerant services using the state machine approach: A tutorial,” *ACM Computing Surveys (CSUR)*, vol. 22, no. 4, pp. 299–319, 1990.

- C. Dwork, N. Lynch, and L. Stockmeyer, “Consensus in the presence of partial synchrony,” *Journal of the ACM (JACM)*, vol. 35, no. 2, pp. 288–323, 1988.

- M. J. Fischer, N. A. Lynch, and M. S. Paterson, “Impossibility of distributed consensus with one faulty process,” *Journal of the ACM (JACM)*, vol. 32, no. 2, pp. 374–382, 1985.

- M. Sadek Ferdous, M. Jabed Morshed Chowdhury, M. A. Hoque, and A. Colman, “Blockchain consensus algorithms: A survey,” *arXiv*, pp. arXiv–2001, 2020.

- F. Gai, B. Wang, W. Deng, and W. Peng, “Proof of reputation: A reputation-based consensus protocol for peer-to-peer network," in *International Conference on Database Systems for Advanced Applications*, pp. 666–681, Springer, 2018.