BFT

IN5420 Distributed Blockchain Technologies

Michael Eikeland

March 2018

1 Byzantine fault tolerance

Byzantine fault tolerance (*BFT*) is the ability to tolerate failures in distributed computing systems. A failure is not a single thing, but for example constitutes scenarios such as dead nodes, malicious attackers or corrupted data. Failure will prevent a system from reaching consensus. BFT systems are well researched and have is proven to have impossibility results in certain models. Cryptocurrencies and blockchain technologies have challenged the view that deterministic BFT is needed in distributed computing systems.

2 PBFT

PBFT (Practical Byzantine fault tolerance) is a replication algorithm that is able to tolerate Byzantine faults. Earlier Byzantine fault tolerant algorithms had assumptions and requirements that were infeasible attain and accept in practice. PBFT however, as the name implies, offers a practical solution to the BFT problem without imposing too compromising restrictions. The proof-of-concept PBFT implementation in the paper had an overall performance loss of 3% under normal-case operation, which is arguably tolerable in order to achieve BFT. PBFT tolerates $\frac{n+1}{3}$ faulty nodes.

3 Proof-of-Work vs. BFT Replication

This paper compares the probabilistic Proof-of-Work (*PoW*) consensus fabric of blockchain technologies with established BFT replication algorithms. It addresses the performance issues typically associated with blockchains and scalability issues associated with BFT systems. The paper argues that they are at at opposite ends of in terms of scalability and performance. PoW consensus is great for node scalability but terrible at performance. BFT replication algorithms are the opposite; terrible at node scalability but great at performance. They establish comparative grounds and define properties in both systems. The properties are *Node identity management, Consensus finality, Scalability (no.*

of nodes), Scalability (no. of clients), Performance (throughput), Performance (latency), Power consumption, Tolerated power of an adversary, Network synchrony assumption and Correctness proofs.