Byzantine Fault-Tolerance

Byzantine failures are the conditions in which there is a system service failure due to the lack of enough information for a global consensus. Byzantine failures do not always derive from a security problem and it can also be triggered by an electrical failure. Due to this fact, these failures make fault tolerance difficult. Thus, Byzantine failures are the most general and difficult failure modes. Byzantine fault tolerance defends against Byzantine failures. Therefore, a Byzantine fault tolerance system with correctly functioning components will be able to provide the system's service correctly. As an example, Byzantine Generals' Problem is an agreement problem in which Byzantine fault tolerance can be achieved if non-faulty generals have a majority agreement on their strategy.

Bitcoin as a peer-to-peer digital currency system, needs to overcome Byzantine failures. The solution of Bitcoin for reaching a coherent global view of the system state is to use proof-of-work. However, consensus latency and the theoretical peak throughput are still challenging in Bitcoin and are becoming more critical in other Bitcoin-like blockchains. In fact, proof-of-work and Byzantine fault tolerance are totally different in the terms of scalability. That is to say, blockchain based on proof-of-work consensus algorithm offers good node scalability with low performance, while Byzantine fault tolerance blockchain provides good performance for a limited number of participants and scale.

There have been some solutions for improving the scalability of both proof-of-work and Byzantine fault tolerance blockchains. GHOST, Bitcoin-NG and blockDAG were proposed for improving the performance of proof-of-work blockchains. However, GHOST still encounters lack of performance under high loads, and Bitcoin-NG has the possibility of forks creation in it which may result to security implications. Mixing proof-of-work blockchain with Byzantine fault tolerance protocol has also been proposed in order to ensure consensus finality. However, in these solutions there is also the problem of scaling in terms of the number of nodes.