week14_lec1

Main Ideas

- Byzantine Agreement

Byzantine Agreement

In distributed computing, the Byzantine Agreement protocol is used.

A process for reaching a shared agreement on a distributed or decentralized multi-agent platform is known as distributed consensus in distributed systems. It is crucial for the message transmission system.

Our goal in this type of distributed system is to provide dependability, which means correct functioning even when one or more persons are problematic.

In a P2P network, the Byzantine Agreement problem may be viewed as a challenge of stimulating the broadcast channel. First and foremost, we could wonder why the sender isn't transmitting the message or information directly to each node in the network. So there are a variety of reasons why this isn't the best approach to broadcast in a network.

- General directs soldiers
- Victory is certain if all loyal soldiers assault
- If none attack, the Empire survives
- If some attack, the Empire perishes
- Gong will keep time
- But they don't have to fight all at once.

Byzantine Soldiers

- The enemy works by corrupting the soldiers
- Orders are distributed by exchange of messages
- Corrupt soldiers violate protocol at will
- Corrupt soldiers can’t intercept and modify messages between loyal troops

- The gong sounds slowly
- There is ample time for loyal soldiers to exchange messages (all to all)
- There is plenty of time for loyal soldiers to exchange messages (all to all)

The agreement states that the input set `V` must be broadcasted to all the nodes. There could be an obstacle and at max n entities could be corrupted.

- ***Agreement***

There are no differences in the outcomes of non-faulty operations.

- ***Validity***

If all nonfaulty processes start with the same v, the only nonfaulty process decision available is v.

- ***Termination***

At some time, all non-faulty processes come to an end.

There are various models of Byzantine agreement. They are all based on the same general protocol.

- Inquiry/Objective
- Network Model
- Protocol Model
- Security Model
- Adversary Model

Theorem 1: Byzantine agreement is not possible in inadequate graphs.

In a graph G (with n nodes), we state that Byzantine agreement is achievable if there are n devices, A 1,...,A n (which we name agreement devices) with the following properties:

Each agreement device A u accepts a Boolean input and returns a value of 1 or 0.

If node u of G runs A u in g, the behavior g of G is proper.

Any g of G system behavior with at least n-m correct nodes is a proper system behavior.

There is agreement since every correct node selects the same value. Validity: If all of the right nodes have the same input, then the value picked must be that input.

Impossibility of 1 out of 3

- If there are less than n/3 Byzantines, we can achieve an agreement for n players; otherwise, a legitimate choice cannot be reached.

Protocol for 1 out of 4

- The first message has been sent. The results are then analyzed once again. As a result, we have two rounds of message exchange:
- processors exchange input values - values from the first round are exchanged a second time.
- This is how the interactive consistency vector is calculated:

Connectivity Challenge

Consensus/agreement is feasible only if the network is (2t + 1)-connected in a (synchronous) P2P network with n nodes, t of which are (Byzantine) defective. When it comes to cryptography, (t+1)-connectivity is enough. In the worst situation, consensus takes > t rounds in a (synchronous) P2P network with n nodes, t of which are (fail-stop / Byzantine) defective.

There are other methods as well for the Consensus:

- Blockchain Based
- Quantum Byzantine Agreement