Lecture 3: Consensus I: Byzantine General Problems, Dolev-Strong

Lecturer: Shumo Chu *Scribes: Kerem Celik, Radha Kumaran*

3.1 Byzantine Generals Problem

The Byzantine Generals Problem is a thought experiment used to model how distributed systems can reach agreement even in adverse conditions.

3.1.1 Allegory

There exists a group of generals devising a military strategy. Suppose one of the generals is a *commanding general* that proposes the order. The generals, via passing messages, must agree to collectively attack or retreat, in the presence of disloyal generals such that two properties hold:

1. If the commanding general is loyal, all generals agree with their order
2. All loyal generals agree on order

3.1.2 Formal Definition

Distributed consensus: Problem in which multiple processes must agree on some value in the presence of failures

Byzantine fault: Condition of a component where it can fail in an arbitrary way

Synchronous network: There is a latency upper-bound for message delivery. If an honest node sends msg in round r, the recipient will receive msg in round $r + 1$.

Authenticated setting: There exists a public key infrastructure (PKI) where messages can be signed by a node and verified by others with its public key.

The Byzantine Generals Problem, also known as Byzantine Broadcast, requires a solution to allow distributed consensus in a network where nodes can suffer Byzantine faults. Specifically, in a network of n nodes, with one selected as the designated sender, in the presence of honest nodes, which follow the protocol, and $f < n$ corrupt nodes, which can undergo Byzantine faults and whose existence is not known beforehand, the following properties must hold:

1. If the designated sender is loyal, all nodes agree with its value
2. All honest nodes must agree on same value

3.1.3 Naive Solution

Suppose we describe a round-based authenticated Byzantine Broadcast protocol where a designated sender receives bit b as input and all nodes must output a bit after the protocol to signify consensus. We assume message validation using a PKI is done upon every method receipt and sent messages are signed.

Round 1. Designated sender receives bit b as input and broadcasts it

Round 2. For each node, if it receives a single bit b', broadcasts the vote b' else broadcasts the vote 0

Round 3. Output the bit that got majority vote else output 0

A simple attack can be constructed for this protocol. Suppose there are three nodes, S, C_0, C_1 where S is the designated sender and is corrupt.

Round 1. S sends 0 to C_0 and 1 to C_1

Round 2. C_0 votes 0, C_1 votes 1. S votes 0 to C_0 and 1 to C_1

Round 3. C_0 counts 2 votes for 0 and outputs 0. C_1 counts 2 votes for 1 and outputs 1.

3.2 The Dolev-Strong Protocol

The Dolev-Strong Protocol is a round-based protocol that solves the Byzantine Generals problem in a synchronous and authenticated setting.

3.2.1 Setup

There are n nodes numbered $1, 2, \ldots, n$, and we assume that node 1 is the designated sender. Each node i maintains an extracted set $extr_i$ which contains all the distinct valid bits chosen so far. $\langle b \rangle_S$ denotes a bit b that has valid signatures by the set of nodes $S, S \subseteq [n]$. f denotes an upper bound on the number of corrupt nodes.

3.2.2 Protocol

- **Round** 0: Sender sends $\langle 1 \rangle_1$ to all nodes.
- **For each round** $r = 1$ **to** $f + 1$:

  For every message $\langle \tilde{b} \rangle_{1,j_1,j_2,\ldots,j_{r-1}}$ that node i receives with r signatures from distinct nodes:

  If $\tilde{b} \notin extr_i$:

  Add $\tilde{b}$ to $extr_i$.

  Send $\langle \tilde{b} \rangle_{1,j_1,j_2,\ldots,j_{r-1},i}$ to everyone.
- **At the end of round** $f + 1$:

  If $|extr_i| = 1$, node i outputs the bit in $extr_i$, else node i outputs 0.

3.2.3 Intuition

If the nodes had to output at the end of f rounds rather than $f + 1$, then the following attack could be constructed, where the sender is one of the f corrupt nodes:

- **Round** 0: Sender sends $\langle 1 \rangle_1$ to all nodes.
- **For each round** $r = 1$ **to** $f - 1$: The corrupt nodes send no messages.
- **Round** f:

  The set of corrupt nodes $\mathcal{F}$ choose an honest node v and make v receive $\langle 0 \rangle_{i_0, i_1, \ldots, i_f}$ $(i_0, \ldots, i_f \in \mathcal{F})$.
- **At the end of round** f:

  $extr_v = 0, 1$, so v outputs 0. $extr_j = 1$ for all honest nodes j, so they output 1.

However, when the algorithm runs for one more round, each message must have $f + 1$ signatures. This means at least one honest node (say node i) must have signed the message in round $r < f + 1$, and propagated this message with $r + 1$ signatures to all the other nodes. Therefore, all the honest nodes would have added b to their extracted sets at the beginning of round $r + 1$.