Institute of Computer Science, University of Bern
Distributed Algorithms, Spring 2020

Prof. Christian Cachin
Jovana Mićić

Exercise 10

10.1 Byzantine randomized consensus (10pt)

A blockchain guru claims to have invented the following simple binary randomized asynchronous Byzantine consensus algorithm. It powers their new *Gurucoin* blockchain and its creators advertize the incredibly high speed and scalability of their revolutionary algorithm[1].

Recall that an asynchronous randomized protocol with Byzantine processes uses the randomized fail-arbitrary model. The processes use authenticated perfect links to send messages to each other.

Algorithm 1 *Gurucoin* randomized consensus for N processes, of which f are Byzantine.

Implements:
ByzantineRandomizedConsensus (*bc*), with domain $\{0, 1\}$

upon event $\langle$ *bc-Init* $\rangle$ **do**
round $\leftarrow 1$
$b \leftarrow \perp$
decided $\leftarrow$ FALSE

upon event $\langle$ *bc-propose* $\mid v \rangle$ **do**
$b \leftarrow v$
send message $[\text{VOTE}, \textit{round}, b]$ to all processes in Π

upon receiving $N - f$ msgs. $[\text{VOTE}, r, v]$ **such that** $r = \textit{round}$ from distinct processes **do**
let v^* be the majority value among the received v's
let m be the number of received VOTE-messages containing v^*
release the common coin with tag *round*
wait for obtaining the value *coin* of the coin with tag *round*
if $m \geq O$ **then** // overwhelming majority
$b \leftarrow v^*$ // switch vote to v^*
else
$b \leftarrow \textit{coin}$ // adopt the coin
if $m \geq G \wedge \textit{coin} = v^* \wedge \neg \textit{decided}$ **then** // good majority
trigger event $\langle$ *bc-decide* $\mid v^* \rangle$
decided $\leftarrow$ TRUE
round $\leftarrow$ *round* $+ 1$
send message $[\text{VOTE}, \textit{round}, b]$ to all processes in Π

[1] The story, all names, characters, and incidents portrayed here are fictitious. No identification with actual persons (...) and products is intended or should be inferred.

Overview. This algorithm is particularly simple because it uses only one exchange of votes in each round, in contrast to other protocols with two phases per round. Every process maintains its current vote and sends this to all others in each round. When the process receives an overwhelming majority (O) of votes for a value v^* it switches its own vote to v^*; otherwise, it adopts the coin for its vote. Furthermore, when the process observes a good majority (G) of votes for v^* and this v^* matches the coin of the round, then it decides, but continues to further rounds for helping other processes decide. (A separate mechanism to actually let the processes terminate would have to be added.) Assume that $N - f > O > G$.

a) Optimal resilience for Byzantine consensus? (2pt). One can show that the best possible fault tolerance for Byzantine consensus in this model is $N > 3f$. Why does *Gurucoin* not achieve this? In other words, show how the protocol violates the properties of consensus if $N = 3f + 1$ and $f = 1$, no matter how O and G are chosen.

b) Guru resilience? (3pt). The gurus set the protocol parameters to

$$O = N - 2f \qquad \text{and} \qquad G = N - 3f$$

and claim that the *Gurucoin* protocol achieves Byzantine consensus for $N > 4f$. Show why this is wrong. Use $N = 4f + 1$ and $f = 1$ for simplicity.

c) Actual resilience of *Gurucoin* (5pt). Using the settings of O and G from (b), find the smallest k such that the protocol achieves binary randomized Byzantine consensus with $N = kf + 1$ processes. Prove that *Gurucoin* is correct under this assumption.

Hint: Recall the randomized binary consensus protocol that tolerates crashes (Algorithm 5.12–5.13). Notice (1) that Byzantine processes may send any message, but there are only f Byzantine processes. Furthermore, (2) construct an argument similar to (but simpler than) the "observation" used to establish the correctness of Algorithm 5.12–5.13 (pp. 241–242) for showing validity and termination.

d) Bonus: General resilience of *Gurucoin* (+10pt). Formulate a condition on N and f, and find corresponding conditions replacing $m \geq O$ and $m \geq G$, such that the protocol achieves consensus with the *best possible* resilience (i.e., such that for given N, it tolerates the largest number of Byzantine processes).