Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Systems@ETH Zürich

Prof. T. Roscoe, Prof. R. Wattenhofer

Computer Systems

Assignment 9

1 Shared Coins

Advanced

1.1 Adaptive Adversaries

Consider the following synchronous consensus algorithm:

Algorithm 1 Synchronous consensus algorithm: Code for node i

```
1:  v_i ∈ ℝ                          ◁ input
2:  for j = 1, ..., f + 1 do
3:    if you are node j then
4:      broadcast v_i
5:    end if
6:    if you receive v_j from node j then
7:      v_i := v_j
8:    end if
9:  end for
10: output v_i
```

a) Show that this algorithm solves consensus against $f < n$ crash failures.

Suppose there exists a 1-round algorithm `dice_toss()` from which the nodes obtain a common uniformly random output in $\{1, \ldots, n\}$, such that before the nodes run `dice_toss()`, one cannot predict anything about the output of `dice_toss()`.

b) Suppose $f < n/2$. Randomize the algorithm using `dice_toss()` so that given any security parameter λ, in $\mathcal{O}(\lambda)$ rounds, the nodes achieve consensus with probability at least $1 - 2^{-\lambda}$. Explain why your algorithm works.

Now, consider running your randomized algorithm in the presence of an "adaptive" and malicious network adversary. The adversary listens to the nodes' communication while the nodes run the consensus algorithm. Whenever it wants, it can cause any node of its choice to crash, and it can make its choices depending on the nodes' communication. However, the adversary is not allowed to cause more than f nodes to crash, where $f < n/2$.

c) Does your algorithm work against the adaptive adversary? If not, why does it not work, and what are the assumptions you made when you designed your algorithm that the adaptive adversary breaks?

2 Quorum Systems

Quiz

2.1 The Resilience of a Quorum System

a) Does a quorum system exist, which can tolerate that all nodes of a specific quorum fail? Give an example or prove its nonexistence.

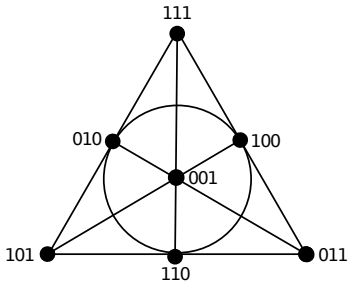
b) Consider the *nearly all* quorum system, which is made up of n different quorums, each containing $n-1$ servers. What is the resilience of this quorum system?

c) Can you think of a quorum system that contains as many quorums as possible?
Note: the quorum system does not have to be minimal.

Basic

2.2 A Quorum System

Consider a quorum system with 7 nodes numbered from 001 to 111, in which each three nodes fulfilling $x \oplus y = z$ constitute a quorum. In the following picture this quorum system is represented: All nodes on a line (such as 111, 010, 101) and the nodes on the circle (010, 100, 110) form a quorum.



a) Of how many different quorums does this system consist of and what are its work and its load?

b) Calculate its resilience f. Give an example where this quorum system does not work anymore with $f+1$ faulty nodes.

2.3 Uniform Quorum Systems

Definitions:

s-Uniform: A quorum system $\mathcal{S}$ is *s-uniform* if every quorum in $\mathcal{S}$ has exactly s elements.
Balanced access strategy: An access strategy Z for a quorum system $\mathcal{S}$ is *balanced* if it satisfies $L_Z(v_i) = L$ for all $v_i \in V$, for some value L.

Claim: An s-uniform quorum system $\mathcal{S}$ reaches an optimal load with a balanced access strategy, if such a strategy exists.

a) Describe in your own words why this claim is true.

b) Prove the optimality of a balanced access strategy on an s-uniform quorum system.