CSE 6117 Distributed Computing Assignment 4

Omar Abid
Student ID: 211295573
CSE Login: omarabid

02-12-2016

The Problem

We consider the algorithm presented in class that solves the Byzantine agreement, with f being the maximum number of Byzantine failures allowed. We also have the conditions:

- Complete Network
- Synchronous Network
- n processes

And satisfies the following two properties:

- *Agreement:* Every correct process outputs the same value.
- *Weak validity:* If every correct process has input v, then every correct process outputs v.

And a stronger validity property dictated as:

- *Strong Validity:* The output of each correct process is the input of some correct process.

1 Show that the assumption that $n > 4f$ is really crucial for that algorithm's correctness. In other words, for every $n \leq 4f$, construct an execution that violates agreement or weak validity.

Let us assume an execution where $n = 4f$, then initially, every process has some preference and each process sends their preference including themselves to every other process. Then the algorithm will attempt to pick the majority value in the first round given all the preferences it has received from every other process.

If there is no majority value, and this is possible in our given condition $n = 4f$, since the number of correct processes must have at least $n - f > n/2 + f$ processes to have a majority and we end up with an incorrect inequality $n > n$ (after substituting $n = 4f$ into the first equality). Assume that this process has input v. If it does not receive a majority, then in the second round this process will choose the preference given by the phase leader. Now let us assume that the phase leader is faulty, if this is the case, then the current process p_i will change it's $new_preference = pref(p_j)$ where $pref(p_i) \neq pref(p_j)$.

Violation of the weak validity Hence we have showed that there is some execution where some process with input v outputs v' where $v \neq v'$, thus the initial assumption of $n > 4f$ is crucial for this algorithm's correctness.

2 Show that the algorithm does not guarantee strong validity even when $n > 4f$.

In a case when even the condition $n > 4f$ is imposed, then we arrive at a similar problem as above. Assume initially that all n processes have different preferences and that there are atleast $n - f$ correct processes.

In the first round, everyone once again sends their preferences and then attempts to pick a majority value. Since everyone's preferences are different, no majority value has been decided and that process updates its own preference to the preference of the phase leader. If it is the phase leader, it keeps its own preference.

Consider that the phase leader that you have gotten your value from is a failing process. It has a choice of outputting either the (fake) majority value or an arbitrary (fake) default value since it deviates from a regular execution. Regardless of the outcome, we have now set the input of this correct process to a value given by an incorrect process.

Violation of Strong validity In subsequent iterations, the output of this correct process will only be the input of some incorrect process. If a correct process sees this input, it will reject it since it has received input from other correct non faulty processes indicating a preference. Hence only incorrect processes will accept this value and we have a violation of the strong validity.

3 Suppose that inputs of correct processes can come from the set $\{1, 2, 3, 4\}$. Prove that no algorithm can guarantee strong validity and agreement when $n = 12$ and $f = 3$.

Consider the initial preferences of the processes where you have an even distribution of the input values chosen {1,1,1,2,2,2,3,3,3,4,4,4}. In such a scenario, each process sends their preference to all other processes but there is no majority. If the failing processes are the ones with the inputs 4,4,4. In this case, we cannot rely on a phase king either since we run into the same problem; this process has gotten some $< f + n/2$ copies of the message v. Hence we rely on some default value. If the default value chosen is the maximum, then the algorithm will fail since we have chosen the values of the incorrect process. If the default value chosen is some other value than the max (like the minimum), we can construct another execution, where the failing processes have initial inputs 1,1,1. This logic follows for all failing processes having some initial input value.

The key to this is that:

- There is some execution regardless of the algorithm wherein there is no majority value present
- Within this execution, we cannot rely on some default value given by one process since it may be a faulty process.

Violating Strong Validity From the same reasoning of part 2 of this assignment, there's some execution where one processes output will be the input to only incorrect process(es).

Agreement property is violated as we see that there are some correct processes $n - f = 9$ that will output different values because of different initial preferences and an inability to resolve onto an agreement using a majority vote or some form of default value.