Lecture 23

1 Overview

Today, we will discuss a classical problem in distributed systems, the Byzantine generals problem.

2 The problem

Here, we imagine an army camped outside a city, divided into several divisions. The generals leading the division wish to decide whether to attack or not on any given day based on their observation. Some number of them are perhaps traitors and wish to mess up their plans, by, for example, having half of the good generals attack and the other half not.

That is, the good generals wish to have an algorithm that gives the following condition.

A. All the loyal generals follow the same plan.

Of, they may also wish to have the following condition.

B. A small number of traitors cannot force the generals to adopt a "bad" plan.

This latter condition is vague. For example, majority can be foiled by even one traitor.

Still, our solution is one based on messages. Each general can send a message to each other general. We need the an algorithm based on communicating thusly such that each general computes a value $v(i)$ for each general.

Condition A, then becomes the following.

1. Any two loyal generals use the same values, $v(1), v(2), \ldots, v(n)$ to make their decision.

Condition B becomes the following

2. If the ith general is loyal, then the value used for $v(i)$ by any loyal general is the value that the general sends.

This can then be reduced to the following problem, referred to as "The Byzantine Generals Problem."

A commanding general wishes to send a command to each of $n-1$ lieutenants, such that

IC1 All loyal generals obey the same command.

IC2 If the commanding general is loyal, then every loyal lieutenant obeys the command s/he sends.

These conditions were called interactivity conditions.

3 Lower Bound

Oral messages are communicated orally general to general. They are not signed written messages that can then be forwarded, without corruption.

We argue that for three generals cannot achieve interactive consistency with even 1 traitor. The values are “attack” and “retreat”.

Say, a lieutenant A receives an “attack” command from the commander. And the other lieutenant says the commander said to retreat. If the other lieutenant was a traitor, A would be forced by “attack” to meet IC2.

On the other hand, if the commander was a traitor, and sends attack to one lieutenant and retreat to the other. Then, the two would take different actions, in order to acheive IC2, but would then violate IC1.

This is not a proof. But with great difficulty can be made into one. We can, however, use this result to prove that any group of $3m$ generals are not tolerant to $m+1$ traitors.

This is done by contradiction. We then construct a solution to the 3 general case. We call the $3m$ algorithm, the Albanian generals algorithm, and the 3 general problem the Byzantion generals. Here, each Byzantine general simulates 1/3 of the Albanian generals. Since, only one of the Byzantine generals is a traitor, only m of the Albanian generals are traitors. Thus, the Albanian generals corresponding to the loyal generals obey IC1 and IC2, and thus the corresponding Byzantine generals can as well.

We note that this problem cannot easily be solved by changing the problem to included timing. For example, that all lieutenants attack within 10 minutes of each other. Once again, one can simulate a solution to the Byzantine generals problem using a solution to this problem. And thus, one cannot solve it with more than 1/3 fraction of traitors.

4 An oral message algorithm.

Again, the messages are delivered with the following guarantees.

- A1 Every message that is sent, is delivered correctly.
- A2 The receiver knows who sent it.
- A3 The absence of a message is known.

We assume a function majority which given a set chooses an element that occurs most often. We only need the condition that if more than half of a set is some value, that we choose that value.

Now, here is an algorithms.

OM(1)

1. Commander sends a value to each lieutenant.
2. Lieutenant uses the received value as its value, or RETREAT if it received no such thing.

OM(m), $m > 0$

1. Commander sends a value to each lieutenant.

2. Lieutenant sends the received value , or RETREAT if it received no such value, via $OM(m-1)$ where it is the commander.

3. Consider a lietenant, if it receives v_j as the value of the commander in the $OM(m-1)$ algorithm from lieutenant j is the commander. It chooses the majority of such values.

Consider, $m = 1$ and $n = 4$. Consider that one of the lieutenants is a traitor. Then, he lies about the values when he tells the other lietenants when he runs $OM(0)$. They, each still get at least two values (their own, and one from another good lieutenant) and thus agree with the commander.

If the commander is a traitor. He can only send inconsistent messages. But, here, two out of three agree, and the lietenants will recursively find the right majority.

The algorithm is implemented as a recursion. The message is essentially labelled with the stack of the recursion, i.e., the sequence of commanding generals in this particular call.

We will now show this algorithm works.

Lemma 1
For any k and m, OM(m) satisfies IC2 when there are more than $2k + m$ generals and at most k traitors.

Proof:

This, by induction. If the commander is loyal, OM(0) works for $m = 0$.

For $m > 0$, each loyal lieutenant runs OM(m-1) on n-1 generals, the commander is loyal so $n > 2m + k$, and $n - 1 > 2k + (m - 1)$, and thus satisfies IC2. That is, every loyal lieutenant gets a valid v_j for each loyal lietenant.

Since $(n-1) > 2k + (m-1)$, the majority of these lieutenants get the same value of v_j.

□

Lemma 2
OM(m) satisfies IC1 and IC2 when there are at least 3m generals and no more than m traitors.

Proof:

By, induction on m. OM(0) works for no traitors.

If the commander is loyal, we are done since IC2 is implies IC1 in this case.

If the commander is not so nice, there are at most m traitors and one is the commander. Thus, at most $m - 1$ lieutenants are traitors, and $3m - 1 > 3(m - 1)$. Thus, the algorithm works recursively and all the lieutenants receive the same values. Thus, they all agree to do the same thing. This implies IC1.

□

5 Signed Messages

We add the condition here.

[A4] (a) A loyal generals signature cannot be forged.

(b) Anyone can verify a generals signature.

Here, we assume a choice function over a set of values where if the set is singleton, it chooses that value, and if it is empty it chooses RETREAT.

Here, one can tolerate up to $m - 2$ bad generals and still get interactive consistency.

The idea is that each general receives messages of the form, $v(1), v(i_1), v(i_2)$... where $v(1)$ is a message from the commander and all the other $v(j)$ is the signature of j. If the number of signatures is less than the number k of bad generals, the general signs this message and forwards it.

This protocol ensures that they all agree. If a general ever sees two different messages from the commander it defaults to retreat. Now, we argue that each good general sees the same set of signed values from the commander.

Question 1: Prove this statement.