The coordinated attack and the jealous amazons

Alessandro Panconesi
DSI - La Sapienza
via Salaria 113, piano III
00198 Roma, Italy

In this lecture we shall discuss two problems that arise in the context of distributed computing, the *coordinated attack* and the *jealous amazons* problems. At the outset the first might look easy while the second impossible. In fact, the opposite is true!

1 The coordinated attack

Two armies, denoted as A and B are camped at the foot of a hill. A third army C is lodged inside a fortress on the top. If A and B attack *simultaneously* they will win, otherwise they will be annihilated. In order to agree on a time to attack, the two armies can communicate by sending messengers. A messenger requires 30 minutes to deliver a message but it can be ambushed by the enemy and killed. A and B want to agree on a time to attack. Is this possible at all?

This is a typical coordination problem arising in a distributed system that is trivial in the absence of faults but that becomes more interesting if one is looking for a *fault-tolerant* solution. In this case it is the communication medium that it is not secure, for it is not certain that messengers arrive at their destination. The problem is not at all a theoretical one. In [1], perhaps the first paper to state the problem formally, the following example is given. Army A is a teller machine in Frankfurt from which a customer wants to withdraw, say, a million marks. Army B is a computer in Tokyo in charge of the customer account. Certainly A and B must act in a coordinated fashion, i.e., either A pays and B debits or A does not pay and B does not debit, otherwise somebody will be in real trouble!

Let us formalize the problem. We have a distributed system in which A and B are two processes communicating via messages. The system is *synchronous* that is, messages are delivered within 1 time unit or not at all (they are lost). Each process has its own *input bit* $x_i, i \in \{A, B\}$ and is to to produce an *output bit* $y_i, i \in \{A, B\}$, subject to the following conditions:

> **Coordination:** The output bit are the same and are produced simultaneously;
>
> **Non-triviality:** if $x_A = x_B = b$ then $y_A = y_B = b$;
>
> **Limited-bureaucracy:** The protocol terminates.

The reader should think about why the Non-triviality condition is necessary. Finally, we need to model a transmission channel that works intermittently:

> **Fairness Assumption:** if infinitely many messages are sent then infinitely many are delivered.

Now that the problem is formalized the question whether a protocol satisfying Coordination and Non-triviality exists becomes a mathematical question, amenable to a proof.

Theorem 1. *There is no protocol for the Coordinated Attack that satisfies Coordination, Non-triviality and Limited-bureaucracy.*

Proof. Consider the case when $x_A = 0$ and $x_B = 1$. Clearly in this case the number of exchanged messages must be strictly greater than zero. Assuming that a protocol exists, let us pick the execution of the protocol in which the number of messages that were successfully delivered is the smallest. Let e be this execution and let m be the last message to be delivered in e. In case there are two such messages let m be any one of them. Suppose now without loss of generality that m was sent from A to B. Consider the execution e' in which m is not delivered. Since B receives one message less, it can notice a difference. But, if B sends messages to A, we can prevent them from reaching the destination without violating the Fairness Assumption. Therefore as far as A is concerned nothing is changed– the input bit and the sequence of messages received are identical. It follows that in e' A decides the same thing as in e and does so at the same round r. Because of the Coordination condition, this forces B to decide at round r the same output bit. But e' has one message less than e, contradicting the minimality of e and thereby proving the theorem. $\star$

The theorem can also be proven in the following way. The above reasoning shows that given any execution e_0 in which the processes decide, the last message can be removed obtaining a new execution e_1 in which the processes decide the same value at the same moment. The last message is defined as the last message sent in chronological order, ties broken arbitrarily. It follows that all messages can be removed, thereby proving that if a protocol works then it works by exchanging zero messages. But this is clearly impossible (see the exercise).

2 The Jealous Amazons

In the kingdom of the Amazons a cuckold is a cuckold, irrespective of sex. As it is well-known, the state of being a cuckold displays a noteworthy lack of reflexivity: when somebody is a cuckold everybody knows about it, except for the cuckold. One day the Queen of the Amazons summoned the population to the main square: "In our kingdom there are women that are cuckolded by their partners. For the sake of social order you are not allowed to communicate about it in any manner. But whenever you are certain that you are the cuckold, you shall shot your partener at the strike of midnight of that day". The Amazons went back to their homes, resuming their normal activities. Nobody spoke about or mentioned the problem. There were 17 women whose companions were unfaithful. At midnight of the seventeenth day, 17 shots reverberated simultaneously in the air and all the culprits were gone. How is it possible?

This puzzle is interesting because it seems that the information conveyed by the Queen does not add anything to what everybody already knows. The puzzle is solved by looking at the general case.

Theorem 2. *If in the Kingdom of the Amazons there are k cuckolds, the culprits are shot at midnight of the k-th day after the speech of the Queen.*

Proof. Suppose $k = 1$. Then, as soon as the Queen speaks the cuckold knows to be a cuckold and at midnight her partner is shot. Suppose $k = 2$ and let A and B be the two cuckolds. At midnight of the first day A does not hear any shots. Therefore she knows that B sees another cuckold, which makes two cuckolds. Since A sees only one cuckold she now knows that she must be the other one. Therefore at midnight of the second day A and B shoot their cheating partners.

Inductively, if $A_1, \ldots, A_k$ are cuckolds they all see $k-1$ cuckolds. When, at midnight of day $k-1$, they hear no shots they know that they must be cuckolds. ⋆

Problems and Exercises

Problem 1. *Why is the Non-triviality condition needed in Coordinated Attack?*

Problem 2. *Give a formal proof that if $x_A \neq x_B$ there can be no protocol for Coordinated Attack using zero messages.*

Problem 3. *Suppose we are looking for a protocol for (a modification of) Coordinated Attack, satisfying the following conditions.*

- *Limited Bureacracy,*
- *Non-triviality, and*
- *Agreement: $y(A) = y(B)$, i.e. the decision must be the same but it can be taken at different times.*

It is assumed that after deciding the protocol stops. The system is synchrnous and the Fairness Condition holds. Does a protocol exist?

Problem 4. *Is there a protocol for the Coordinated Attack for* asynchronous *systems?*

Problem 5. *Suppose we are looking for a protocol for (a modification of) Coordinated Attack, satisfying the following conditions.*

- *Non-triviality,*
- *Agreement, and*
- *Limited Dithering: the decision must be taken within finite time but the protocol can be executed forever. In particular, a process can keep sending and receiving messages after deciding. Such protocols are called* quiescent*.*

The system is synchronous and the Fairness Condition holds. Does a protocol exist?

References

[1] Jim Gray, Notes on Data Base Operating Systems, 1977