



#ASLI ENGINEERING

The Two General's Problem



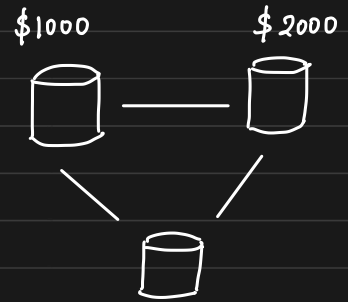
BY

ARPIT BHAYANI

Distributed Consensus: Two General's Problem

Reaching consensus is extremely important in any distributed network.

eg: we cannot have two datanodes in a cluster such that one thinks price = \$1000 while the other thinks price = \$2000



Depending on which node the request hits, the user would see the corresponding value, giving an inconsistent view. Somehow, the nodes need to agree on one value.

Reaching consensus is easy, when there are NO failures.

it becomes impossible problem when communication links are unreliable.

Two General's Problem

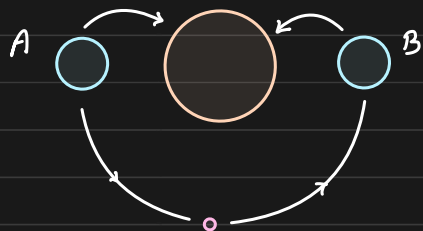
Say, there are two generals who want to attack enemy from different directions.



The only way to conquer is when both the armies attack simultaneously.

If one attacks but the other doesn't, the enemy will win.

The generals can communicate only via the foot soldiers. This messenger can be captured by the enemy and hence the 'message' can be lost.



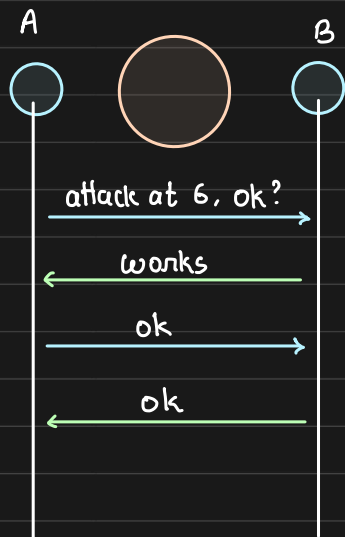
How would the generals co-ordinate their attack over this unreliable network?

When no messages are lost

if communication channel is reliable, then all general can send each other messages to agree upon the attack.

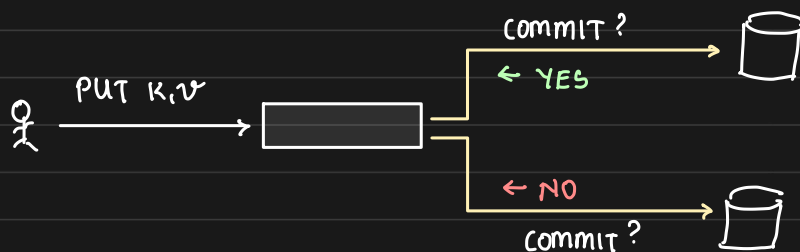
After rounds equal to the 'diameter' of the network, every general would know where and when to attack.

This algorithm does not work when the communication channel is unreliable, ie when messages can be lost.



Real world analogy

Two general problem = commit in a distributed database



We have to ensure, either

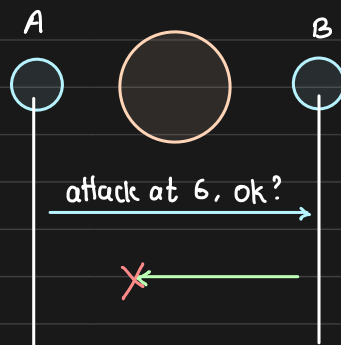
1. everybody commits
2. everybody aborts

So that our distributed database always remain consistent

When messages are lost

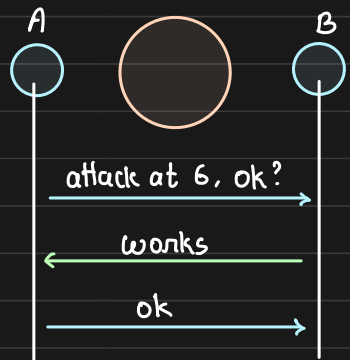
A sent message to B, but B's response got lost, so, should A attack?

also should B attack?



A sent message to B, B responded,
but did A receive B's decision?

B can be sure about this, only
after it receives an ACK from A



But, how can A be sure that B agreed

hence it waits for an ACK from B

and the ACK for an ACK from A and so on....

This is the classic Two General's Problem

Two nodes can never come an agreement if network is unreliable

How should the generals decide?

A always attacks and it sends lots of messages to increase
the probability that one will go through.

if none of the messages go through,

A attacks, B does not

We are just increasing the odds, but not deriving
a concrete solution

But distributed consensus is possible, but how?

we make some probabilistic assumptions about the loss of messages

eg: if we assume failure rate of 0.5

we send 2 or 3 messages instead of 1

just to cover the odds



Hence, in real world, we always retry.

↑
To over compensate for network unreliability