Bangladesh University of Engineering & Technology

Computer Science and Engineering

BYZANTINE GENERALS PROBLEM

Supervisor:

Abdur Rafi

Report written by:

2105123-Shatabdi Dutta Chowdhury

2105124-Shadman Abid

2105129-Sarowar Alam Roki

Date: December 19, 2024

Contents

Byzantine Generals Problem

Abstract

In a distributed computer system, some components may behave unpredictably or maliciously, sending conflicting information that disrupts overall operations. This challenge can be presented as to achieving agreement among system nodes communicating through unreliable channels, even when some produce faulty results. We need to develop a protocol that ensures consensus among trustworthy components, regardless of disruptions. Consensus is possible with basic communication methods if the number of faulty components remains below one-third of the number of total components. With secure mechanisms like cryptographic authentication, consensus can be achieved even with an arbitrary number of faulty components.

Introduction

Reliable computer systems must be designed to withstand failures, even when some components behave erratically. A challenge arises when faulty components send conflicting information to different parts of the system. This problem is often abstracted as a distributed agreement challenge, famously referred to as the Byzantine Generals Problem.

Suppose *several divisions of an army are encamped around a city, each divison led by its general. The generals must coordinate a unified strategy—whether to attack or retreat—using only messengers for communication. However, some generals may act as traitors, sending contradictory messages to sow confusion and disrupt the consensus. The loyal generals must devise a strategy to ensure that all their divisions act in unison, regardless of any attempts by traitors to undermine the plan.*[Lamport 1982]

Problem Formulation

To address the Byzantine Generals Problem, we model the system as a group of generals coordinating through unreliable channels. Each general observes

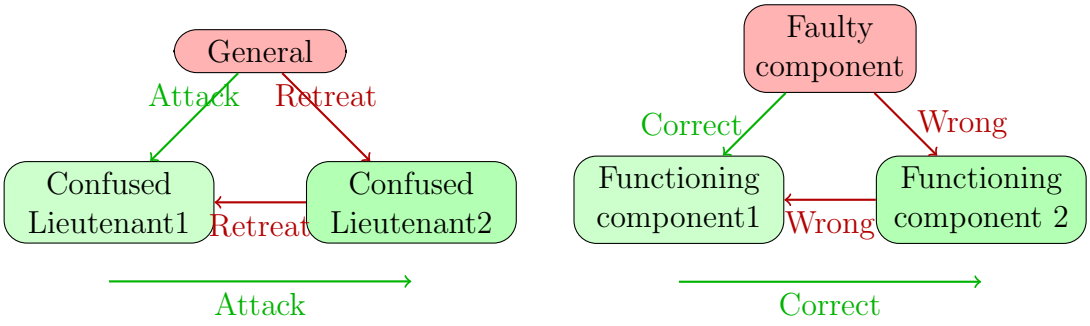
external conditions and communicates their observations to others, forming a collective decision.Let us assume there are n generals , hence , a commanding general must send messages to n-1 generals such that :

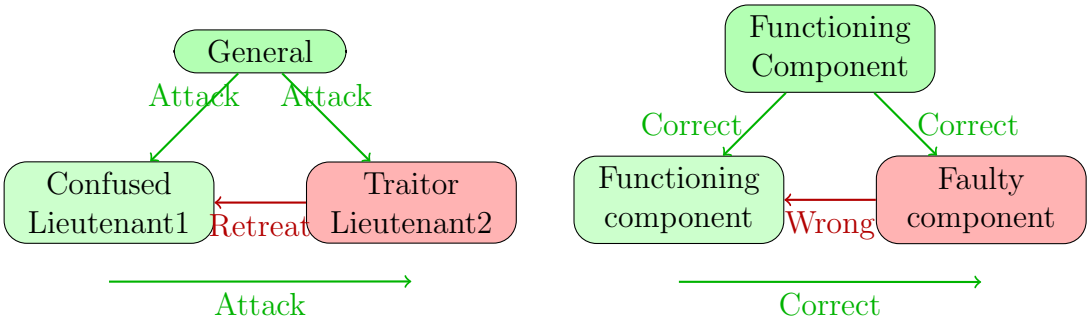
Interactive Consistency Conditions

1. **Consistency (C1)**:All loyal lieutenants obey the same order.
2. **Correctness (C2)**: If the commanding general is loyal , then every loyal lieutenant obeys the order he sends.

These conditions ensure that malicious actions cannot disrupt agreement or lead loyal participants to an undesirable decision. However, satisfying these conditions is complex because traitorous participants can send contradictory information, potentially misleading different groups of loyal generals.

The Byzantine Generals problem relates to the distributed network in a sense that , the faulty components can be mapped to the traitor generals .To put in the analogy of the Byzantine Generals Problem , the functioning components must produce the correct output even if some of the components fail just as to the Loyal Byzantine Generals who must come up with a consensus even if some of them deceive.





To ensure consistency and correctness, every general must use the same method for processing and combining information. For instance, if the decision involves a binary choice, a majority vote can help achieve consensus, if the number of traitorous participants is limited to less than one-third of the total entities involved. The solution to this problem must ensure every loyal participant receives consistent and accurate information, even if some pariticipant deliberately send false or conflicting data. This requires robust communication protocols and mechanisms to authenticate messages, minimizing the influence of malicious actors and preserving the integrity of the decision-making process.

Impossibility Results

The Byzantine Generals Problem may seem deceptively simple, but its complexity lies in the surprising fact that if the generals communicate using only oral messages, no solution can work unless more than two-thirds of the generals are loyal. For instance, with only three generals, no solution can tolerate even a single traitor. An ***oral message*** is one whose contents are entirely controlled by the sender, allowing a traitorous sender to transmit any message. Such a message is analogous to the type of communication typically used by computers.

(a) Case 1 (b) Case 2

Figure 1: No solution Scenario with Minimum i.e. 1/3 Traitors of the Total Entities.

Let us consider two scenarios:

1. In the first scenario, the commander is loyal and sends an "attack" order, but Lieutenant 2 is a traitor who tells Lieutenant 1 that the commander sent a "retreat" order.

2. In the second scenario, the commander is a traitor who sends an "attack" order to Lieutenant 1 and a "retreat" order to Lieutenant 2.

In both cases, Lieutenant 1 cannot distinguish between the two scenarios. Thus, if Lieutenant 1 obeys the "attack" order in the first scenario, he must also obey it in the second scenario.Similar argument applies to Lieutenant 2 also. This situation violates the integrity conditions:

Integrity Conditions

- **IC1**: All loyal generals obey the same order.
- **IC2**: If the commander is loyal, then every loyal lieutenant obeys the order sent by the commander.

Hence, no solution exists for three generals with one traitor. This argument may appear plausible but lacks rigorous proof. For a formal proof, refer to Lamport 1982. Using this result, it is proven that no solution with fewer than $3m + 1$ generals can handle m traitors.

1 Mathematical Proof

Now , the proof is through contradiction. And here we will be showing ***No solution with fewer than*** $3m + 1$ ***generals exists where m is the number of traitors.***

Hence the proof is through contradiction , we assume that there is a solution that works with $3m$ or fewer generals and use it to construct a 3 generals solution , which we know is impossible from Lamport 1982 .

Now , to avoid confusion we name the assumed solution , with $3m$ or fewer generals where m is the number of traitors , as Albanian Generals and constructed solutions as Byzantine Generals . That is starting from an algorithm that allows $3m$ or fewer Albanian generals to cope with m traitors we reach a solution for Byzantine generals that allow 3 generals to handle 1 traitor.

Let us assume ,each Byzantine general simulates $\approx m$ Albanian generals sothat each Byzantine general is simulating m Albanian Generals where the Byzantine Commander is simulates the Albanina commander and his $m-1$ lieutenants simulate the $m-1$ Byzantine lieutenants. Again each of these Byzantine lieutenants simulates at most m Albanian generals Problem recursively. Since at base case Byzantine Generals problem can work for 1 traitors Lamport 1982 , so , Albanina Generlas can handle m traitors.

Specifically:

Byzantine Commander: $m-1$ Albanian Lieutenants + Albanian Commander,
Byzantine Lieutenants: At most m Albanian Lieutenants each.

Since at most one Byzantine general is a traitor, only m Albanian generals can be traitors but we assumed a solution exists for $3m$ general or fewer generals where m is the number of traitors i.e.the number of traitors can be more than m that is , it can work for more traitors than m which is a contradiction. By satisfying IC1 and IC2 for the Albanian generals, the corresponding conditions hold for the Byzantine generals. Thus, a contradiction is reached.

2 Approximate Agreement

In Byzantine Generals Problem , even achieving approximate agreement is equally challenging. For approximate agreement, Let us assume the following

modified conditions:

Modified Conditions

- **IC1'**: All loyal lieutenants attack within 10 minutes of one another.
- **IC2'**: If the commander is loyal, then all loyal lieutenants attack within 10 minutes of the commanded time.

Using a similar approach, it can be shown that approximate agreement for three generals with one traitor is also impossible. Suppose the commander sends an attack at $1:00$ and a retreat at $2:00$. Each lieutenant executes the following:

1. If the received time is:
   - $\leq 1:10$, then attack.
   - $\geq 1:50$, then retreat.
   - Otherwise, defer the decision.
2. Ask the other lieutenant's decision. If decided, follow their decision; otherwise, retreat.

By contradiction, this approach also constructs a three-general solution for the original Byzantine Generals Problem, which is impossible. Hence, no solution exists for fewer than $3m + 1$ generals to handle m traitors.

The impossibility of solving the Byzantine Generals Problem, even approximately, highlights the critical role of loyalty thresholds in achieving consensus. The mathematical insights presented here serve as the foundation for future research on fault-tolerant algorithms.

Solutions

Oral Messages

We aim to solve the Byzantine Generals Problem using oral messages. Let us define an oral message system with the following assumptions:

- **A1**: Every message that is sent is delivered correctly.

- **A2**: The receiver of a message knows who sent it.
- **A3**: The absence of a message can be detected.

Let us describe the algorithm $OM(m)$, for $m \geq 0$, to solve the problem with at most m traitors. We assume the existence of a function `majority` that returns the majority value from a list, or RETREAT if no majority exists.

Algorithm OM(0)

Algorithm 1 OM(0)

1: The commander sends his value v to every lieutenant.
2: **for** each lieutenant i **do**
3: **if** Lieutenant i receives no value **then**
4: Let v_i = RETREAT
5: **else**
6: Let v_i be the value received from the commander.
7: **end if**
8: **end for**
9: Each lieutenant uses the received value v_i, or RETREAT if no value is received.

Algorithm OM(m) for $m > 0$

Algorithm 2 OM(m)

1: The commander sends his value v to every lieutenant.
2: **for** each lieutenant i **do**
3: Let v_i be the value received from the commander, or RETREAT if no value is received.
4: Lieutenant i invokes Algorithm $OM(m-1)$ with the value v_i and sends it to every other lieutenant.
5: **end for**
6: **for** each lieutenant i, and each lieutenant $j \neq i$ **do**
7: Let v_j be the value received by Lieutenant i from Lieutenant j.
8: **if** Lieutenant i receives no message from Lieutenant j **then**
9: Let v_j = RETREAT
10: **end if**
11: Lieutenant i computes the majority value v_i = majority$(v_1, v_2, \ldots, v_{n-1})$
12: **end for**

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

Proof of Lemma 1

Proof.

Step 1: The commander sends a value v to all $n-1$ lieutenants.

Step 2: Each lieutenant applies Algorithm $OM(m-1)$ to all other generals.

Step 3: By the induction hypothesis, each lieutenant gets $v_j = v$ for each loyal lieutenant.

Step 4: Since there are at most k traitors, a majority of the values are loyal.

Step 5: Thus, each loyal lieutenant computes the majority value v = majority$(v_1, v_2, \ldots, v_{n-1})$.

□

Correctness of Algorithm OM(m)

Theorem 1. *Algorithm $OM(m)$ satisfies conditions IC1 and IC2 if there are more than $3m$ generals and at most m traitors.*

Proof of Theorem 1

Proof. We prove the theorem by induction on m.

Base Case: When $m = 0$, the commander sends the value v to all lieutenants. Since no traitors exist, all lieutenants get the same value v. Thus, IC1 and IC2 are satisfied.

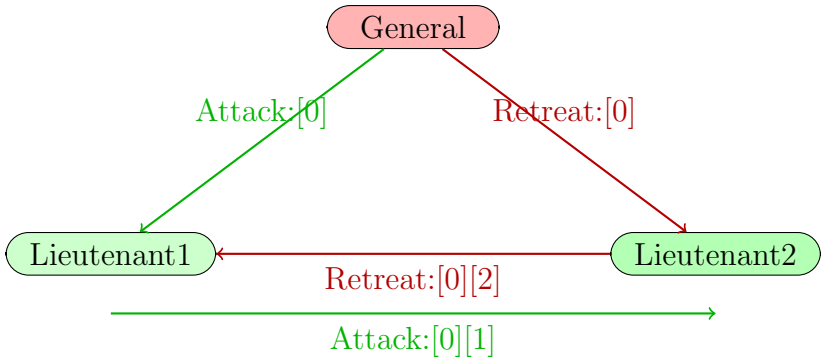
Inductive Step: Assume that the theorem holds for $m - 1$. Now, consider $m > 0$. If the commander is loyal, then by Lemma 1, OM(m) satisfies IC2. IC1 follows from IC2 for loyal commanders.

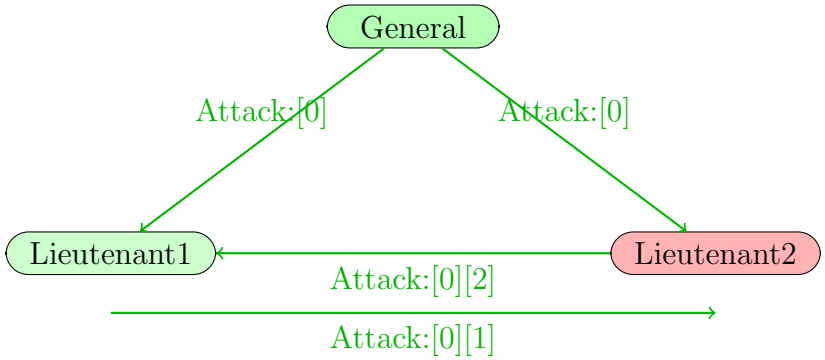
If the commander is a traitor, at most $m-1$ lieutenants are traitors. Since there are more than $3m$ generals, there are more than $3m-1$ lieutenants, and $3m - 1 > 3(m - 1)$. Thus, by the induction hypothesis, OM(m-1) satisfies IC1 and IC2 for each lieutenant. Hence, the loyal lieutenants compute the same majority value in step 3, satisfying IC1. □

Signed Messages

Signed messages introduce digital signatures to authenticate communication, ensuring that malicious actors cannot forge orders. This method significantly improves fault tolerance.

When a message is sent , it is sent with the author's signature attached to it . So , it is very easy to identify who is sending different messages to different entities.





In second case , although the lieutenant 2 is a traitor , he cannot infiltrate since , the message contains a signature and he cannot forge the signature of the general.

Applications

A.3 Blockchain and Cryptocurrencies

Byzantine Fault Tolerance (BFT) algorithms ensure consensus in decentralized systems, such as Bitcoin and Ethereum, even in the presence of malicious nodes.

A.4 Distributed Databases

Ensuring consistent data across nodes in distributed databases often relies on BFT principles to handle faults and maintain reliability.

Conclusion

The Byzantine Generals Problem is a cornerstone of reliability in distributed systems. It enables consensus, scalability, and resilience, making it essential for modern decentralized technologies like blockchain.

The Byzantine Generals Problem highlights the challenges of achieving consensus in distributed systems, particularly in the presence of malicious or faulty actors. It underscores the importance of designing robust protocols to ensure that honest participants can agree on a single course of action, even

when some participants are unreliable or adversarial. The problem's solutions, such as Byzantine Fault Tolerance (BFT) algorithms, have become foundational in modern distributed computing and blockchain systems, ensuring reliability, security, and consistency. By addressing the core issues of trust and coordination, the Byzantine Generals Problem has significantly influenced the evolution of fault-tolerant computing.

References

Lamport, Leslie (1982). "The Byzantine Generals Problem". In: *ACM Transactions on Programming Languages and Systems* 4.3, pp. 382–401.