Distributed systems:

The Byzantine Generals Problem

Matej Pavlovič
Distributed Programming Laboratory
System model so far

• \( n \) processes, message passing

• Process crashes
  • Algorithms become non-trivial
  • Additional assumptions required (P, correct majority..)

• What if processes could lie?
Retreat!

Retreat!
Retreat!

Retreat!
Retreat
Attack
Retreat
Retreat!
Requirements

• All *loyal* generals choose the same plan (Attack / Retreat)

• A few traitors cannot impose a bad plan on the loyal generals
Let’s formalize
$\nu_1 = \text{Retreat}$

$\nu_2 = \text{Attack}$

$\nu_3 = \text{Retreat}$
Let’s formalize

• $n$ generals
• $v_i = i$-th general’s opinion (value: Attack / Retreat)
• generals only exchange oral messages

... 2 conditions ...
Recall: Requirements

• All *loyal* generals choose the same plan (Attack / Retreat)

• A few traitors cannot impose a bad plan on the loyal generals
Let’s formalize

• $n$ generals
• $v_i = i$-th general’s opinion (value: Attack / Retreat)
• generals only exchange oral messages

1) Every *loyal* general makes his decision based on the same information $(d_1, ..., d_n)$
Traitor 1: Retreat
2: Attack
3: Retreat

don’t care

1

\(d_1: \text{Retreat}\)
\(d_2: \text{Attack}\)
\(d_3: \text{Retreat}\)

2

3
Recall: Requirements

• All *loyal* generals choose the same plan (Attack / Retreat)

• A few traitors cannot impose a bad plan on the loyal generals
Traitor

don’t care

\[ d_1 : \text{Attack} \]
\[ d_2 : \text{Attack} \]
\[ d_3 : \text{Attack} \]
Traitor

don’t care

\[ v_2 = \text{Retreat} \]

\[ d_1: \text{Attack} \]
\[ d_2: \text{Attack} \]
\[ d_3: \text{Attack} \]

\[ v_3 = \text{Retreat} \]

\[ d_1: \text{Attack} \]
\[ d_2: \text{Attack} \]
\[ d_3: \text{Attack} \]
Let’s formalize

- *n* generals
- \( v_i = i\)-th general’s opinion (value: Attack / Retreat)
- generals only exchange oral messages

1) Every *loyal* general makes his decision based on the same information \( (d_1, \ldots, d_n) \)

2) If *i*-th general is loyal, every *loyal* general must base his decision on \( d_i = v_i \)
Let’s formalize

- $n$ generals
- $v_i = i$-th general’s opinion (value: Attack / Retreat)
- generals only exchange oral messages

1) Every *loyal* general makes his decision based on the same information $(d_1, \ldots, d_n)$
   \[\iff\text{Every *loyal* general uses same value as } d_i\]

2) If $i$-th general is loyal, every *loyal* general must base his decision on $d_i = v_i$
Commander and Lieutenants

• Solve once for each general $i$:
  • 1 commander (general $i$)
  • $n - 1$ lieutenants (other generals)
  • commander $i$ sends value $v_i$ to lieutenants
Byzantine Generals Problem

Commander must send an order to \( n - 1 \) lieutenants, such that:

**BG1:** All loyal lieutenants obey the same order

**BG2:** If commander is loyal, then every loyal lieutenant obeys commander’s order

In our case, command is “Use ‘Attack’ / ‘Retreat’ as \( d_i \)”
3 generals, 1 of them traitor
To satisfy BG2, Lieutenant 1 must obey “Attack!”. 
“Commander said ‘Attack!’”

“Retreat!”

“Retreat!”

To satisfy BG2, Lieutenant 1 must obey “Retreat!”.
3 generals, 1 of them traitor

To satisfy BG2, a loyal lieutenant must obey the order directly received from the commander.
Commander said ‘Attack!’

“Attack!”

Traitor

“Retreat!”

BG1 violated!

Lieutenant 1

“Commander said ‘Attack!’”

Lieutenant 2

“Commander said ‘Retreat!’”
3 generals, 1 of them traitor

To satisfy BG2, a loyal lieutenant must obey the order directly received from the commander.

\[ \downarrow \]

If commander is a traitor, BG1 is violated.

\[ \downarrow \]

No algorithm can satisfy BG1 and BG2 for 3 generals and 1 possible traitor.
Impossibility result

• No algorithm can solve the “Byzantine Generals Problem” for 3 generals, if one of them can be a traitor.

• Generalization: There is no algorithm for $3f$ generals, if $f$ or more of them can be traitors. (proof by reduction from 3 generals, 1 traitor)
3f generals, f of them traitors

• Proof by contradiction:
  1. Assume a solution for BGP(3f, f) for some f
  2. Use it to solve BGP(3,1)

\[ \downarrow \]
Contradiction with “there is no solution to BGP(3,1)”
Albanian generals

Some algorithm for BGP(3f,f) (exists by assumption)
Albanian generals

Some algorithm for $\text{BGP}(3f,f)$ (exists by assumption)

$\text{traitors}$

$\text{simulates}$
Unsolvability for BGP(3f,f)

If algorithm for BGP(3f,f) existed
\[ \Downarrow \]
Could use it to solve BGP(3,1)
\[ \Downarrow \]
Contradiction to unsolvability of BGP(3,1)
\[ \Downarrow \]
Conclusion: No alg. for BGP(3f,f) exists.
Conclusion

• If faulty processes can lie (not only crash)
  • Correct **majority** is **not enough**!
  • Even **two thirds** are **not enough**!
  • True for any synchrony assumptions

• What can we do? (next lecture)
  • Stronger assumption: > 2/3 are correct
  • Use signed messages