CSE 486/586 Distributed Systems
Byzantine Fault Tolerance --- 1

Steve Ko
Computer Sciences and Engineering
University at Buffalo

Recap

- Spanner
  - Geo-distributed database
  - Supports a relational data model with a SQL-like language
  - Supports distributed transactions with linearizability
- Transaction ordering for linearizability
  - Tight time synchronization
  - TrueTime-based timestamps
  - Principle: using a time value that is certain
- TrueTime
  - TT.now() returns an interval [earliest, latest].
  - TT.after(t) is true if t has definitely passed.
  - TT.before(t) is true if t has definitely not arrived.

Byzantine Fault Tolerance

- Fault categories
  - Benign: failures we've been talking about
  - Byzantine: arbitrary failures
- Benign
  - Fail-stop & crash: process halted
  - Omission: msg loss, send-omission, receive-omission
  - All entities still follow the protocol
- Byzantine
  - A broader category than benign failures
  - Process or channel exhibits arbitrary behavior.
  - May deviate from the protocol
  - Can be malicious (attacks, software bugs, etc.)

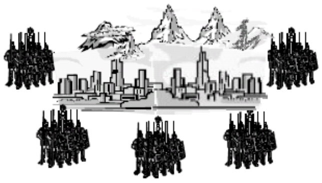
Byzantine Fault Tolerance

- Result: with *f faulty nodes*, we need *3f + 1* nodes to tolerate their Byzantine behavior.
  - Fundamental limitation
  - Today's goal is to understand this limitation.
  - Next lecture: a protocol that provides this guarantee.
- How about Paxos (that tolerates benign failures)?
  - With *f* faulty nodes, we need *2f + 1* to obtain the majority.

"Byzantine"

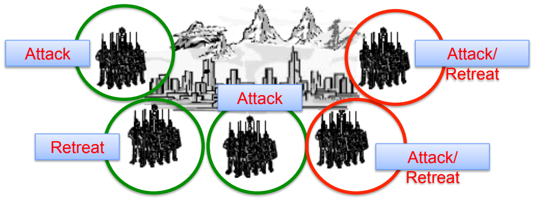
- Leslie Lamport (again!) defined the problem & presented the result.
- *"I have long felt that, because it was posed as a cute problem about philosophers seated around a table, Dijkstra's dining philosopher's problem received much more attention than it deserves."*
- *"At the time, Albania was a completely closed society, and I felt it unlikely that there would be any Albanians around to object, so the original title of this paper was The Albanian Generals Problem."*
- *"…The obviously more appropriate Byzantine generals then occurred to me."*

Introducing the Byzantine Generals



- Imagine several divisions of the Byzantine army camped outside of a city
- Each division has a general.
- The generals can only communicate by a messenger.

Introducing the Byzantine Generals



- They must decide on a common plan of action.
  - What is this problem?
- But, *some of the generals can be traitors*.

Requirements

- All loyal generals decide upon the same plan of action (e.g., attack or retreat).
- A small number of traitors cannot cause the loyal generals to adopt a bad plan.
- There has to be a way to communicate one's opinion to others correctly.

The Byzantine Generals Problem

- The problem boils down to how a single general sends the general's own value to the others.
  - Thus, we can simplify it in terms of a single commanding general sending an order to lieutenant generals.
- Byzantine Generals Problem: a commanding general must send an order to *n-1* lieutenant generals such that
  - All loyal lieutenants obey the same order.
  - If the commanding general is loyal, then every loyal lieutenant obeys the order the commanding general sends.
- We'll try a simple strategy and see if it works.
  - All-to-all communication: every general sends the opinion & repeatedly sends others' opinions for reliability.
  - Majority: the final decision is the decision of the majority
  - Similar to reliable multicast

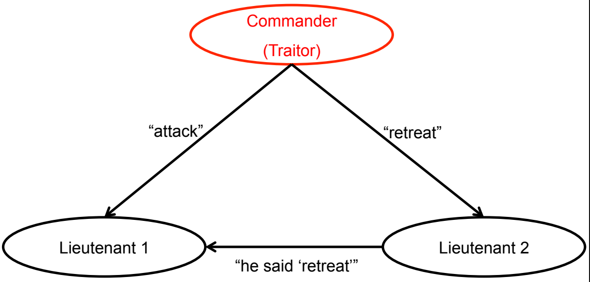
CSE 486/586 Administrivia

- PA4 due next Friday @ 1:59pm
- Final: 5/14, Wednesday, 3:30pm – 6:30pm
  - Norton 112
  - Everything
  - No restroom use (this quickly becomes chaotic)
  - Bring an erasure, if you'd like.
- Important things about the final week
  - PA4 scores will be released by Wednesday.
  - Thursday and Friday office hours are for PA4.
  - No office hours from Monday to Wednesday
  - Scoring will hopefully be done by the end of the week.

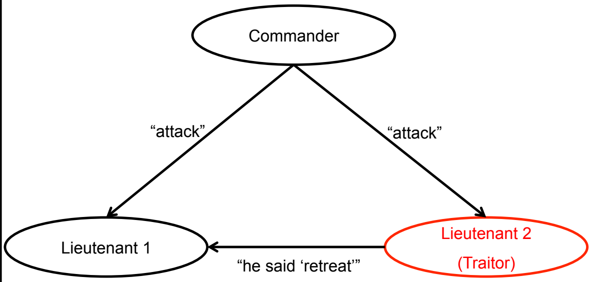
Question

- Can three generals agree on the plan of action?
  - One commander
  - Two lieutenants
  - One of them can be a traitor.
  - This means that we have 2*f* + *1* nodes.
- Protocol
  - Commander sends out an order ("attack"/"retreat").
  - Lieutenants relay the order to each other for reliability.
  - Lieutenants follow the order of the commander.
- Can you come up with some scenarios where this protocol doesn't work?

Understanding the Problem



Understanding the Problem



Understanding the Problem

- With three generals, it is impossible to solve this problem with one traitor.
- Why not Paxos?
  - Paxos works with $2f + 1$ nodes when f nodes are faulty.
  - In Paxos, f nodes can *fail (or disappear)* from the system, but *they don't lie and they are not malicious*.
- In the Byzantine generals problem, f nodes might be alive and malicious.
- In general, you need $3f + 1$ nodes to tolerate *f faulty nodes* in the Byzantine generals problem.
- Why?

Intuition for the Result

- Problem setting
  - General question: how do we reach consensus in the presence of faulty (malicious) nodes?
  - Let's say each honest node runs a deterministic algorithm that gives the same answer (yes/no).
  - We choose a quorum's answer, since there can be malicious nodes that give a wrong answer intentionally.
- Question: how many votes do I need?
  - In Paxos, I need $f + 1$ votes (agreeing on either yes or no) out of $2f + 1$ nodes, since that's the majority.
- Will this work with Byzantine failures?
  - I.e., just like Paxos, let's just collect $f + 1$ answers.
  - The principle is that the outcome should be determined by the answers of the honest nodes, not the malicious nodes.

Intuition for the Result

- Let's apply this to the Byzantine generals problem.
  - Principle: The outcome should be determined by the answers of the honest nodes, not the malicious nodes.
  - Let's say we obtain $f + 1$ votes.
  - Up to f nodes can be malicious → getting $f + 1$ votes means that the result can contain up to f wrong answers.
- Example
  - $2f + 1$ nodes, and outcome by $f + 1$ votes.
  - f faulty nodes say no.
  - f non-faulty nodes say yes
  - 1 non-faulty node says yes.
  - Ideal outcome?
  - Actual outcome?
- What do we need?

Intuition for the Result

- We need more votes from the honest nodes than the faulty nodes, so the faulty nodes can't influence the outcome.
- Unlike Paxos, we can't simply collect $f + 1$ votes, since malicious nodes might give wrong answers.
- We need to obtain $2f + 1$ answers. Then we have at least *$f + 1$ votes from honest nodes*, one more than the number of potential faulty nodes.
- Then we need to see if $f + 1$ votes say the same thing out of $2f + 1$.
- This way, we can make sure that honest nodes determine the outcome.

Intuition for the Result

- But, f nodes still might just simply fail, not reply at all.
- How do we get $2f + 1$ replies when there are f failed nodes?
- Thus, we need at least $3f + 1$ processes in total to tolerate f faulty processes.

Summary

- Byzantine generals problem
  - They must decide on a common plan of action.
  - But, some of the generals can be traitors.
- Requirements
  - All loyal generals decide upon the same plan of action (e.g., attack or retreat).
  - A small number of traitors cannot cause the loyal generals to adopt a bad plan.
- Impossibility results
  - With three generals, it's impossible to reach a consensus with one traitor
  - In general, with less than *3f + 1* nodes, we cannot tolerate *f* faulty nodes.

Acknowledgements

- These slides contain material developed and copyrighted by Indranil Gupta (UIUC).