

IN BYZANTIUM

Ashish Vulimiri and Shehla Rana





THE BYZANTINE GENERALS PROBLEM

Leslie Lamport, Robert Shostak, and
Marshall Pease

Presented by: Ashish Vulimiri

OUTLINE

- Introduction
- Impossibility results
- Algorithm: oral messages
- Algorithm: signed messages
- Discussion

- Extensions



INTRODUCTION



INTRODUCTION

- Generals coordinate via messengers
- Problem: traitors
- Attack will only succeed with enough troops
- Coordinate in the presence of traitors
- Note on system model:
 - Don't really need a commander
 - Distributed Consensus \leftrightarrow Byzantine Agreement
 - Just simplifies presentation



PROBLEM

- Commander, $N-1$ lieutenants. a of the generals arbitrarily capricious
- Commander sends out Boolean order. Ensure:
 - All loyal lieutenants obey same order
 - If commander is loyal, his order must be the one obeyed
- Assumptions
 - Synchronous, reliable communication
 - Fully connected network
 - Sender identity cannot be forged

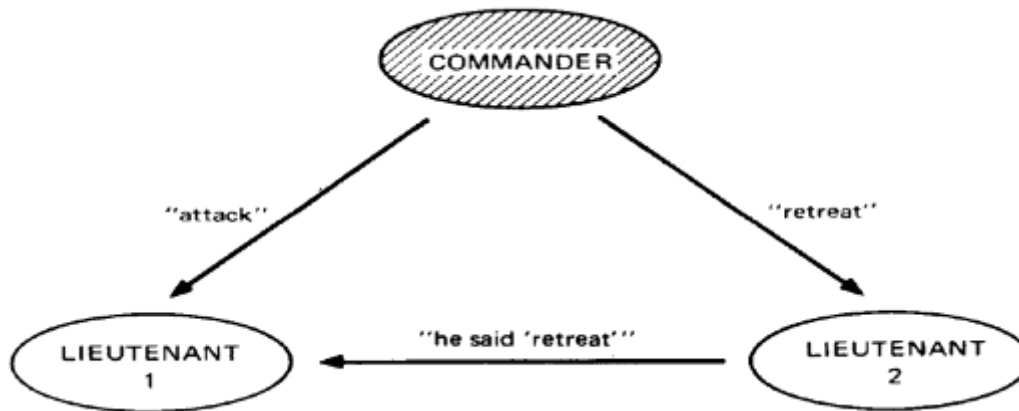
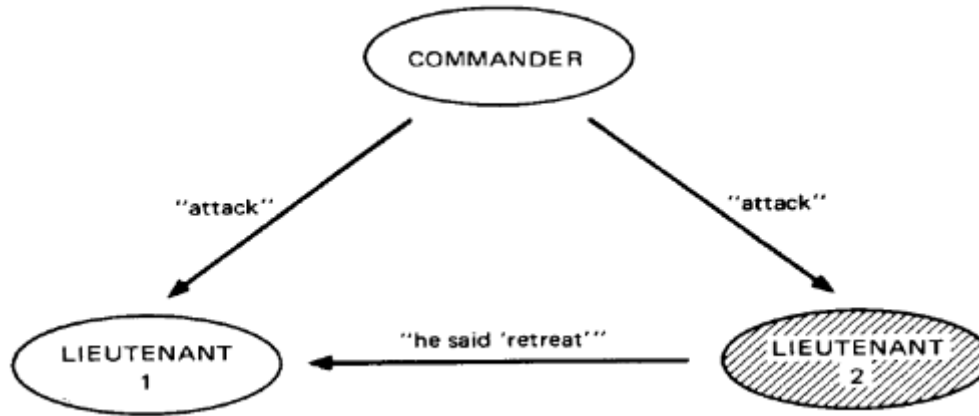


IMPOSSIBILITY RESULT

- Can't be done with $N \leq 3a$
- Specifically, for $N=3$, $a=1$:



IMPOSSIBILITY RESULT (N = 3, A = 1)



DOES THIS CONSTRUCTION GENERALIZE?

- Are the numbers $N=3$, $a=1$ special?
- No
- Suppose we have a solution for some (N, a) , with $N \leq 3a$
- Can simulate the three node case
 - Intuition: have each of the three nodes simulate roughly $N/3$
 - Warning: can be tricky to formalize



DOES THIS CONSTRUCTION GENERALIZE?

- Would using non-Boolean values help?
- No
- Suppose we had ints (e.g. timestamps), required only that the final values be within a certain range
- Reduction: can simulate Boolean case
 - E.g. final value $bValue = (10 \leq iValue \leq 15)$
- Note: reducibility isn't everything



ALGORITHM

- Suppose we have $N > 3a$. Can solve.
- Notation:
 - G = set of generals
 - N ($= |G|$) and a as earlier
 - $\text{BFT}(G, a)$ – the problem we want to solve
 - $\text{Broadcast}(G, a, t)$ – the algorithm
- Final result:
 - $\text{Broadcast}(G, a, a)$ solves $\text{BFT}(G, a)$



ALGORITHM

Broadcast($G, a, 0$):

$T = \text{now}$

Commander c sends value x_c to all lieutenants

Receive messages for $T = \text{now}$

$x_p = (\text{message received}) ? x_c : \text{default}$

Every lieutenant p agrees on $y_p = x_p$



ALGORITHM

Broadcast(G , a , t):

$T = \text{now}$

Commander c sends x_c to all lieutenants

Receive messages for $T = \text{now}$

Each lieutenant p does

$x_p = (\text{message received}) ? x_c : \text{default}$

Act as general in Broadcast($G \setminus \{p\}$, $a-1$, $t-1$)

$T = \text{now} + t$

Receive messages for $T = \text{now} + t$

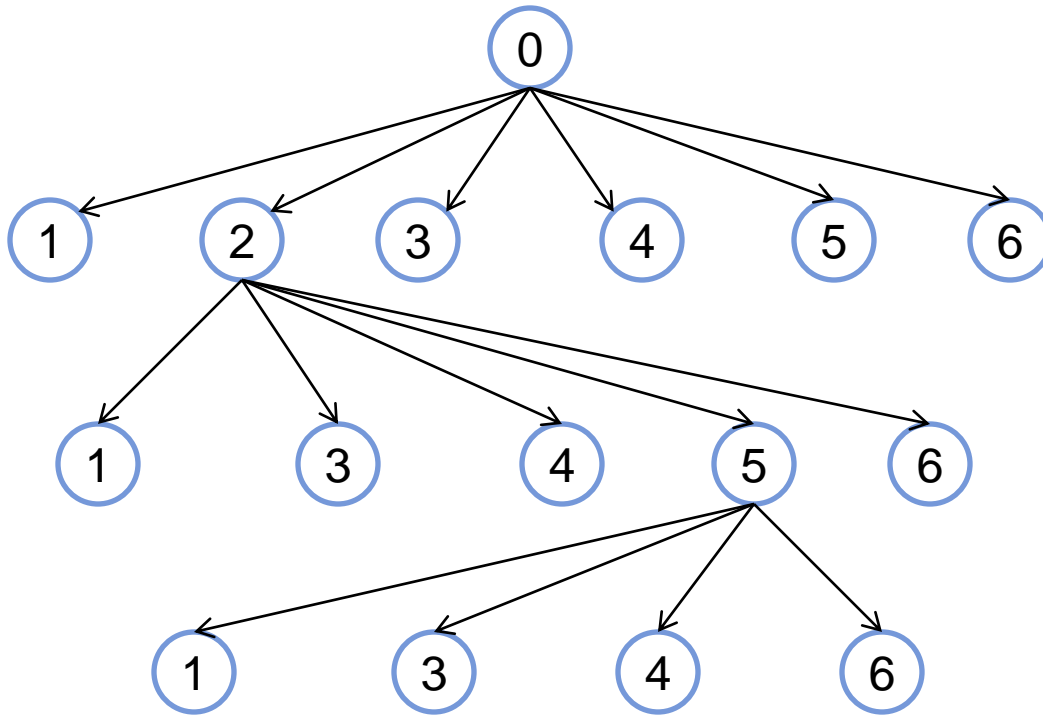
c decides on x_c

p decided on a value for each p' in $G \setminus \{p\}$

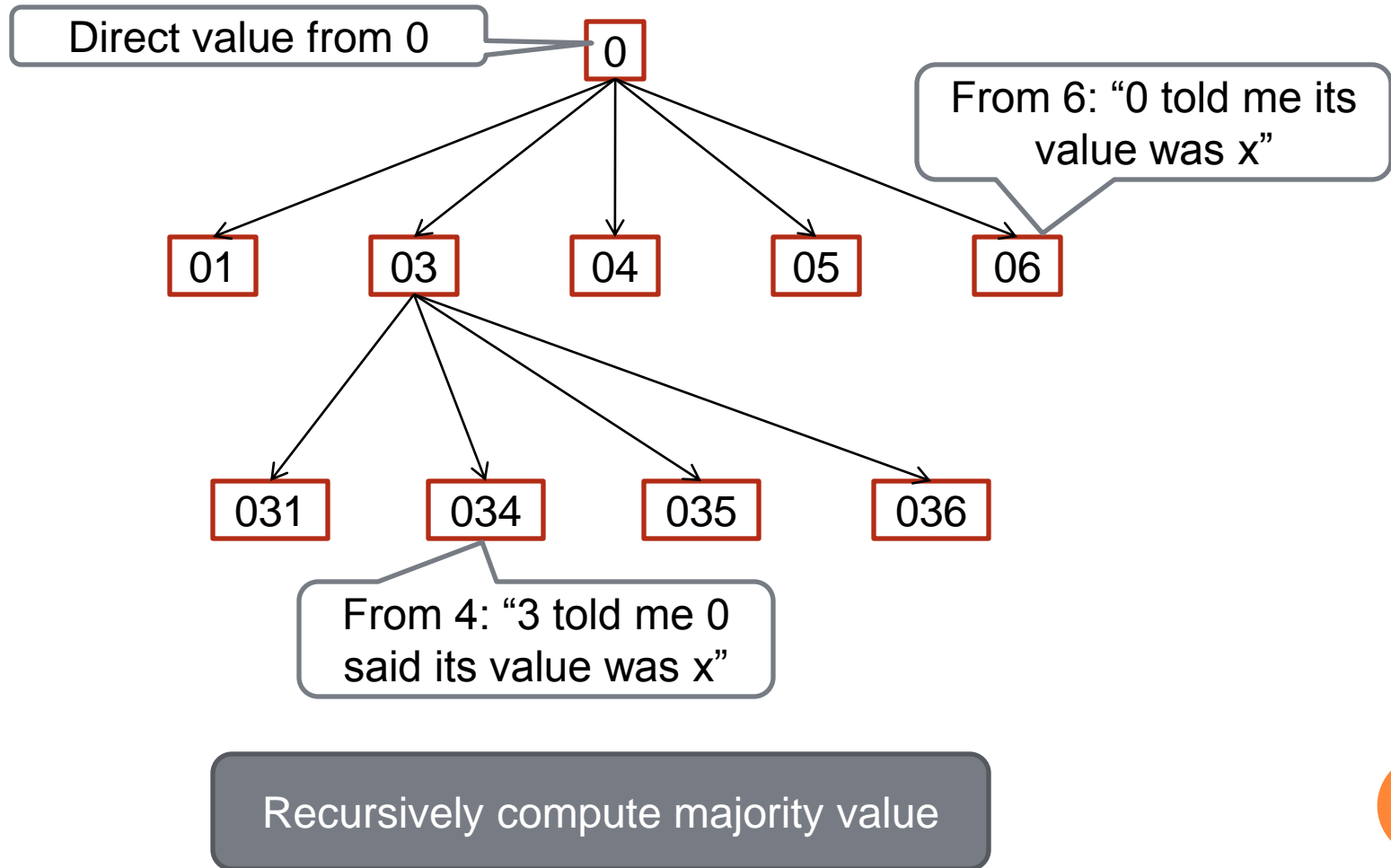
$y_p = \text{majority}(\text{value_set})$



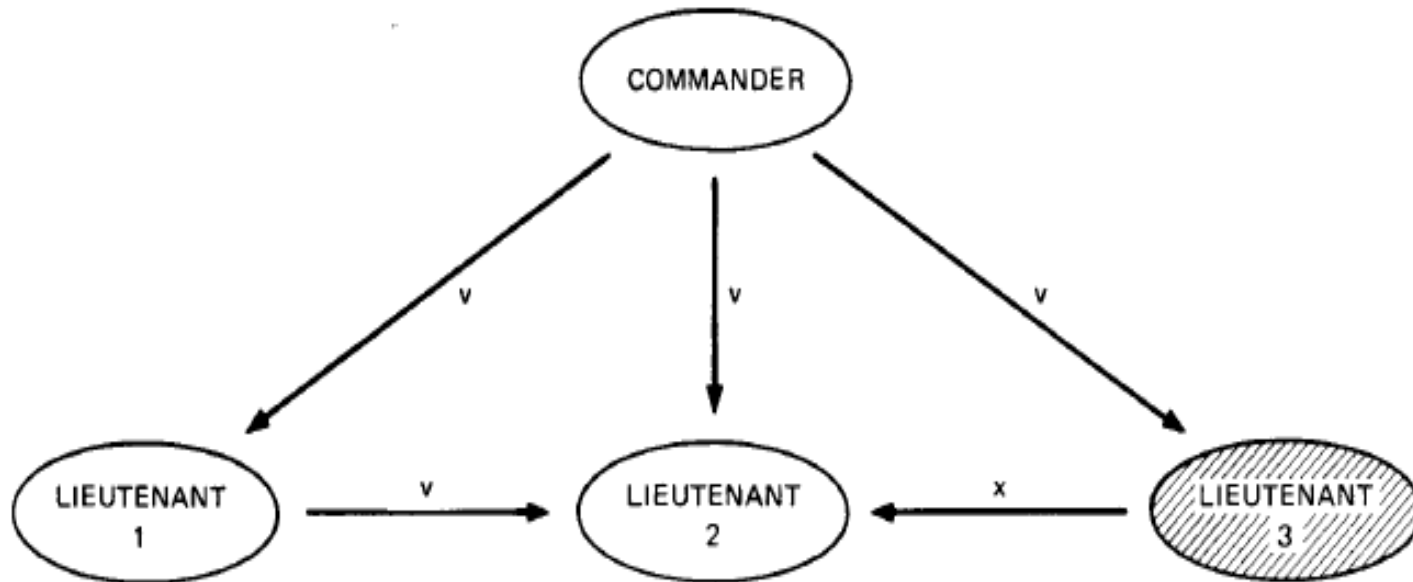
PHASE I: MESSAGE TREE (N=7, A=2)



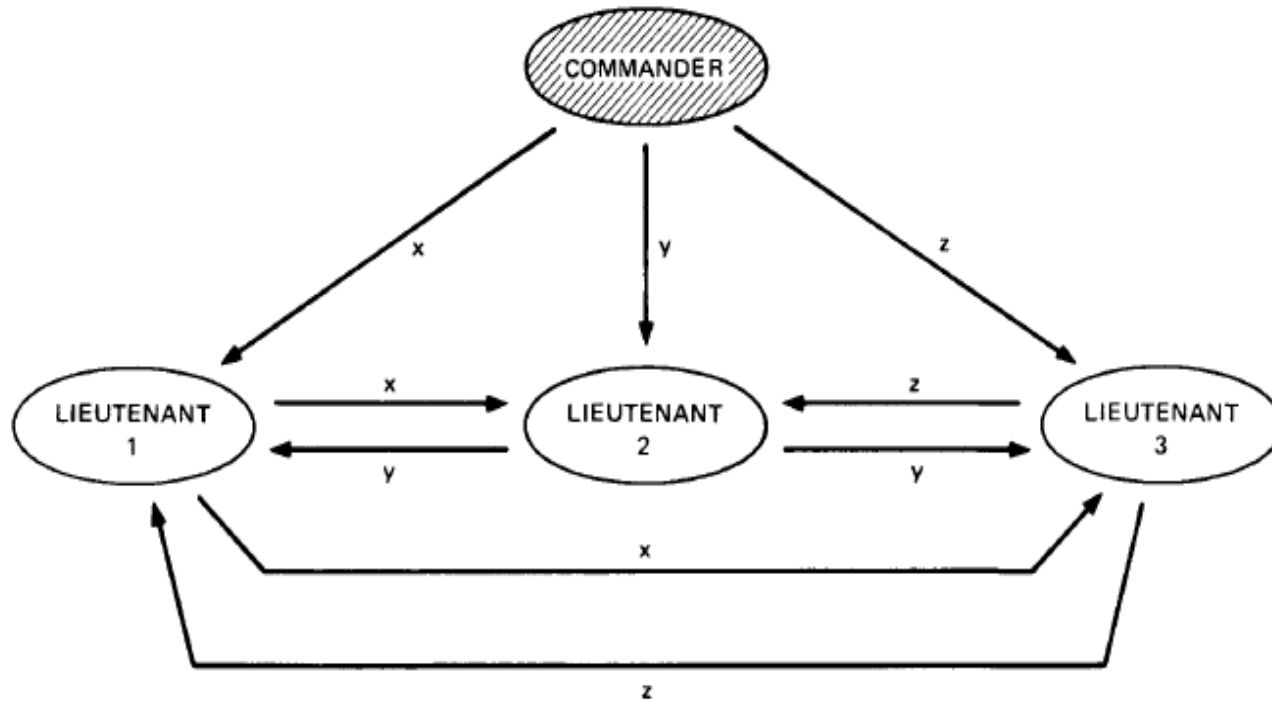
PHASE II: DECISION TREE (AT NODE 2)



EXAMPLE ($N = 4, A = 1$): I



EXAMPLE ($N = 4, A = 1$): II



WHY DOES THIS WORK?

- Primary result:
 - If $N > 2a + t$, Broadcast(G, a, t) solves BFT(G, a) if commander is loyal
- That is: enough to ensure *some* node in every path is loyal
- This is why we executed for $a+1$ rounds
- See paper for details



SIGNED MESSAGES

- What if we had signed messages?
- Remember: we needed multiple rounds to exchange messages like “A told me B told him C said his value was D”
- We only had one-hop unforgeability guarantees
- If we had end-to-end signatures: could solve for any N , any a
- Algorithm in paper is reactive (asynchronous)
- Details omitted: see paper



DISCUSSION

- Time complexity $O(a)$
- But message complexity $O(N^a)$
- Better algorithms exist (see any distsys text)
- Can only overcome faults – cannot identify source



EXTENSION: OTHER TOPOLOGIES

- Lamport et al. show that same algo works for 3-regular graphs
- Other special cases:
 - Rings
 - Random graphs
 - Hierarchical clusters
- General topology?



EXTENSION: NO SYNCHRONY

- BFT impossible in asynchronous networks (Fischer et al, 1985)
- However, good approximation algorithms exist
- Dolev et al's MSR – in each round, do:
 - Label value with round number. Broadcast to everyone else
 - Receive at least $N-a$ values, collect into multiset
 - Drop largest and smallest a values
 - Replace own value with mean
- Problem: *what* do you converge to?



EXTENSION: BETTER FAULT MODELS

- E.g. work by Azadmanesh, Kieckhafer
- a = arbitrarily capricious faults
- s = symmetric faults
- b = benign/detectable faults (e.g. crashes)
- Requirement is $N > 3a + 2s + b + 1$
- They also have a five-mode model
 - Incorporates two modes of network failure

