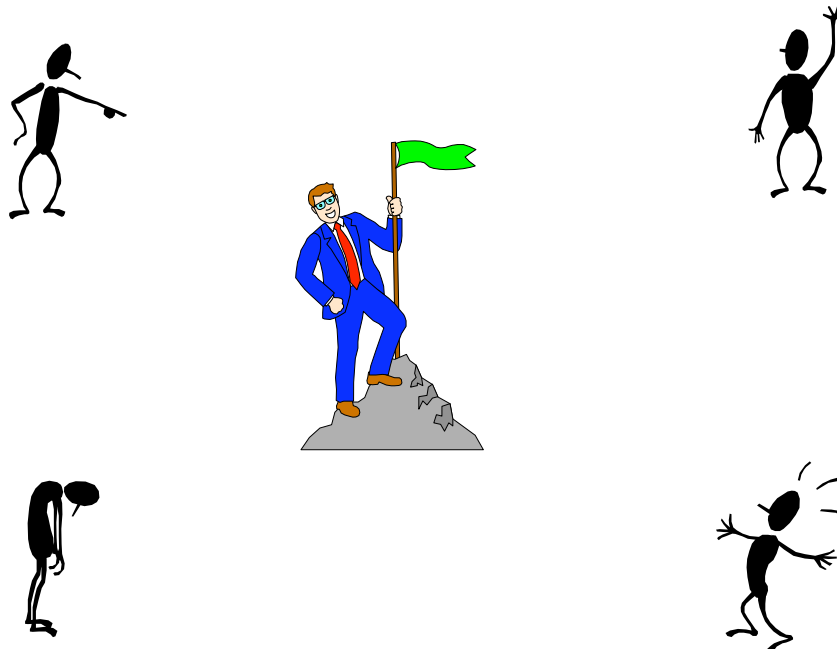


Introduction FT Agreement

- ◆ We will discuss fault tolerant agreement algorithms during this class.
- ◆ We want to start out the discussion with the Byzantine General Problem
 - L. Lamport, R. Shostak, and M Pease, "The Byzantine Generals Problem"
- ◆ Variations of the problem will follow us throughout the rest of the semester.
- ◆ What started it all?
 - Clock synchronization problems in SIFT

Byzantine General Problem



Byzantine General Problem

- ◆ Objective
 - A) All loyal generals must decide on the same plan of action
 - B) A “small” number of traitors cannot cause the loyal generals to adopt a “bad” plan.
- ◆ Types of agreement
 - exact agreement
 - approximate agreement
- ◆ Applications, e.g.
 - agreement in the presence of faults
 - event, clock synchronization

Byzantine General Problem

- ◆ Key to disagreement
 - 1) Initial disagreement among loyal generals
 - 2) Ability of traitor to send conflicting messages
 - » asymmetry
- ◆ Reduction of general problem to simplex problem with 1 General and n-1 Lieutenants
 - General gives order
 - Loyal Lieutenants must take single action

Byz. Gen. Prob. (Simplex)

◆ Want

IC1: All loyal Lieutenants obey the same order

IC2: If the commanding General is loyal, the every loyal Lieutenant obeys the order he sends

- IC1 & IC2 are called *Interactive Consistency Conditions*.
- If the General is loyal, then IC1 follows from IC2.
- However, the General need not be loyal.

◆ Any solution to the simplex problem will also work for multiple-source problems.

- the i^{th} General sends his value $v(i)$ by using a solution to the BGP to send the order “use $v(i)$ as my value”, with the other Generals acting as the lieutenants.

BGP: Oral Message Solution

◆ Oral Message

- message whose contents are under the control of the sender (possibly relays)

◆ Practical implication, sensor example

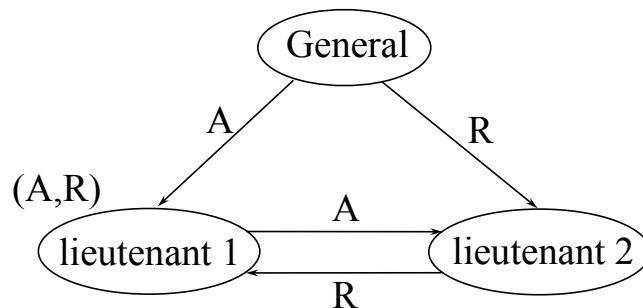
- General = sensor
- Lieutenants = processor redundantly reading sensor
- Initial disagreement
 - » time skew in reading, bad link to sensor
 - » analog - digital conversion error, any threshold function
- Asymmetry
 - » communication problem, noise, V-level, bit timing

BGP: Oral Message Solution

- ◆ The Byzantine Generals Problem seems deceptively simple, however
- ◆ no solution will work unless more than two-third of the generals are loyal.
- ◆ Thus, there exists no 3-General solutions to the single traitor problem using oral messages
- ◆ Assume the messages sent are
 - A = Attack
 - R = Retreat

BGP: Oral Message Solution

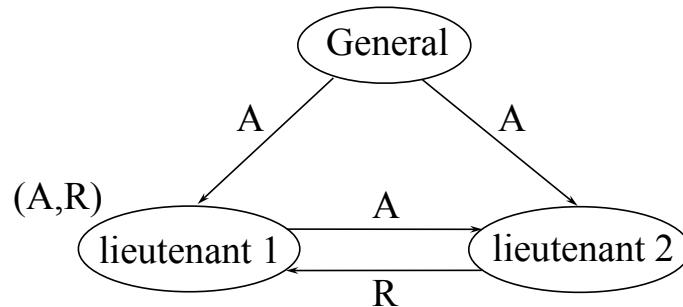
- ◆ Case 1: Commander is traitor:



- commander is lying
- who does lieutenant 1 believe
- could pick default

BGP: Oral Message Solution

- ◆ Case 2: Lieutenant 2 is traitor:



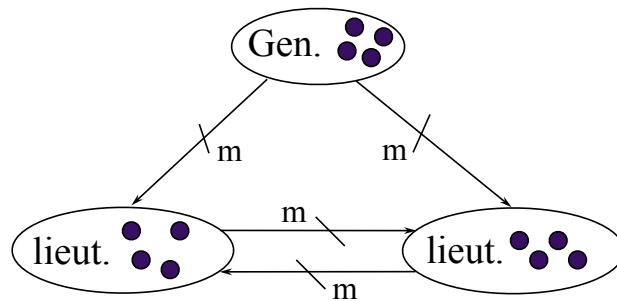
- lieutenant 2 is lying
- who does lieutenant 1 believe
- could pick default, but what if it is R
 - » then General has A and Lieutenant 1 has R !!!

BGP: Oral Message Solution

- ◆ Given case 1 and case 2, lieutenant 1 cannot differentiate between both scenarios, i.e. the set of values lieutenant 1 has is (A,R).
- ◆ In general: Given m traitors, there exists no solution with less than $3m+1$ generals for the oral message scenario.
- ◆ Assumptions about Oral Messages
 - every message that is sent is delivered correctly
 - the receiver of a message knows who send it
 - the absence of a message can be detected
 - how realistic are these assumptions?

BGP: Oral Message Solution

- ◆ General case:
 - regroup generals
 - » n Albanian generals
 - » n/3 act as unit => 3 general Byzantine General Problem



BGP: Oral Message Solution

Algorithm OM(0)

- 1) The commander sends his value to every lieutenant
- 2) Each lieutenant uses the value he receives from the commander, or uses the value RETREAT if he receives no value

Algorithm OM(m), $m > 0$

- 1) The commander sends his value to every lieutenant.
- 2) For each i , let v_i be the value lieutenant i receives from the commander, or else be RETREAT if he receives no value. Lieutenant i acts as the commander in Algorithm OM(m-1) to send the value v_i to each of the $n-2$ other lieutenants.
- 3) For each i , and each $j \neq i$, let v_j be the value lieutenant i received from lieutenant j in step 2) (using algorithm OM(m-1), or else RETREAT if he received no such value. Lieutenant i uses the value

$$\text{majority}(v_1, \dots, v_{n-1})$$

BGP: Oral Message Solution

OM(m) -- same thing, different wording

IF $m = 0$ THEN

- a) commander sends his value to all other $(n-1)$ lieutenants.
- b) lieutenant uses value received or default (i.e. RETREAT if no value was received).

ELSE

- a) each commander node sends value to all other $(n-1)$ lieutenants
- b) let v_i = value received by lieut. i (from commander OR default if there was no message)
Lieut. i invokes OM($m-1$) as commander, sending v_i to other $(n-2)$ lieutenants.
- c) let v_{ji} = value received from lieutenant j by lieutenant i .
Each lieutenant i gets $v_i = \text{maj}(\text{what everyone said } j \text{ said in prev. round, } \underline{\text{except } j \text{ himself}})$

trust myself more than
what others say I said

example $n=4 \Rightarrow$ one traitor

◆ procedure OM(1)

IF {not valid since $m=1$ }

ELSE

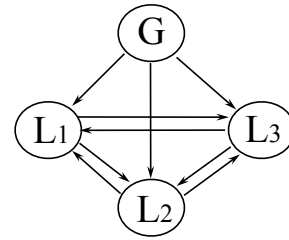
- 1) commander transmits to L1,L2,L3
- 2) values are received by L1,L2,L3
so lieuts call OM(0)

each lieut has
received 3 values
(use majority)

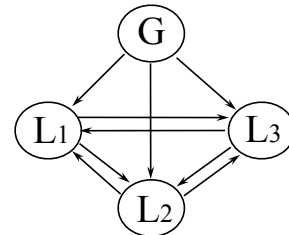
procedure OM(0)
IF { $m=0$ }
1) each lieut sends value to
other 2 lieuts
ELSE {not valid}

BGP example

- ◆ case 1: L3 is traitor
 $v_0 = 1$
 each loyal L has vector
 110 or $111 \Rightarrow \text{maj}(1 \ 1 \ 0/1) = 1$

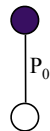


- ◆ case 2: G is traitor
 $v_0 \Rightarrow L1=1 \ L2=1 \ L3=0$
 L1 has 110
 L2 has $110 \quad \text{maj}() = 1$
 L3 has 011

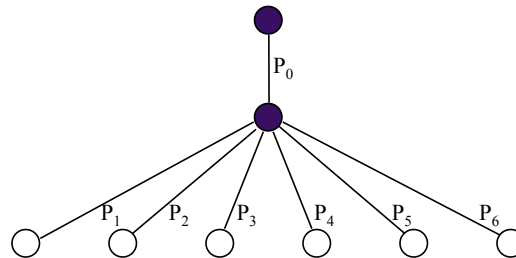


BGP with $N = 7$

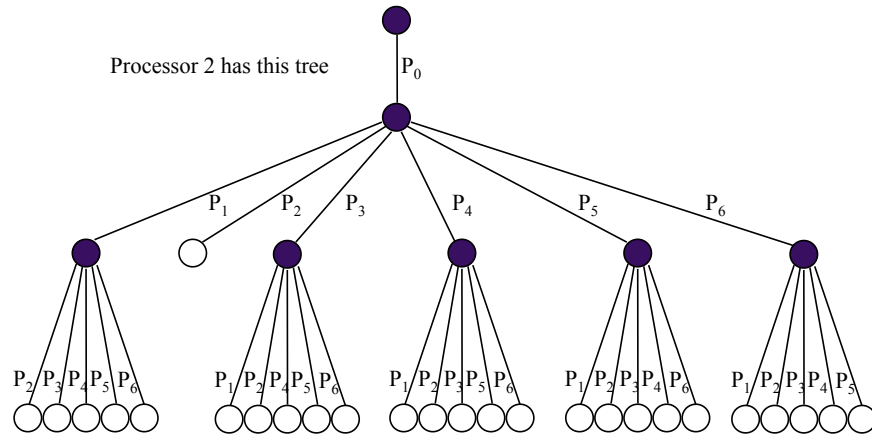
General sends message



After first rebroadcast



BGP with $N = 7$



BGP with $N = 3m + 1$

extra blank

BGP with $N = 7$

