+

The Byzantine Generals Problem

Leslie Lamport, Robert Shostak and Marshall Pease

Presenter: Jose Calvo-Villagran

jcalvovi@uwaterloo.ca

+

Overview

- The Byzantine Generals Problem
- A solution: Oral Messages
- A solution: Signed Messages
- Missing Communication Paths
- BGP in Practice
- Final Thoughts
- Proofs*

*Available upon request

Introduction

- Reliability on distributed systems
- Failed components
- Conflicting information
- Abstractly defined as Byzantine Generals Problem

The Byzantine Generals Problem (1/2)

- Several divisions camping outside enemy city
- Each division has a general
- Communicate only by messenger
- Traitors!

The Byzantine Generals Problem (2/2)

- A commanding general must send an order to his n-1 lieutenant generals such that:
  - IC1. All loyal lieutenants obey the same order.
  - IC2. If the commanding general is loyal, then every loyal lieutenant obeys the order he sends.
- Solution: (3m + 1) generals for m traitors

A solution: Oral Messages (1/2)

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.

A solution: Oral Messages (2/2)

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

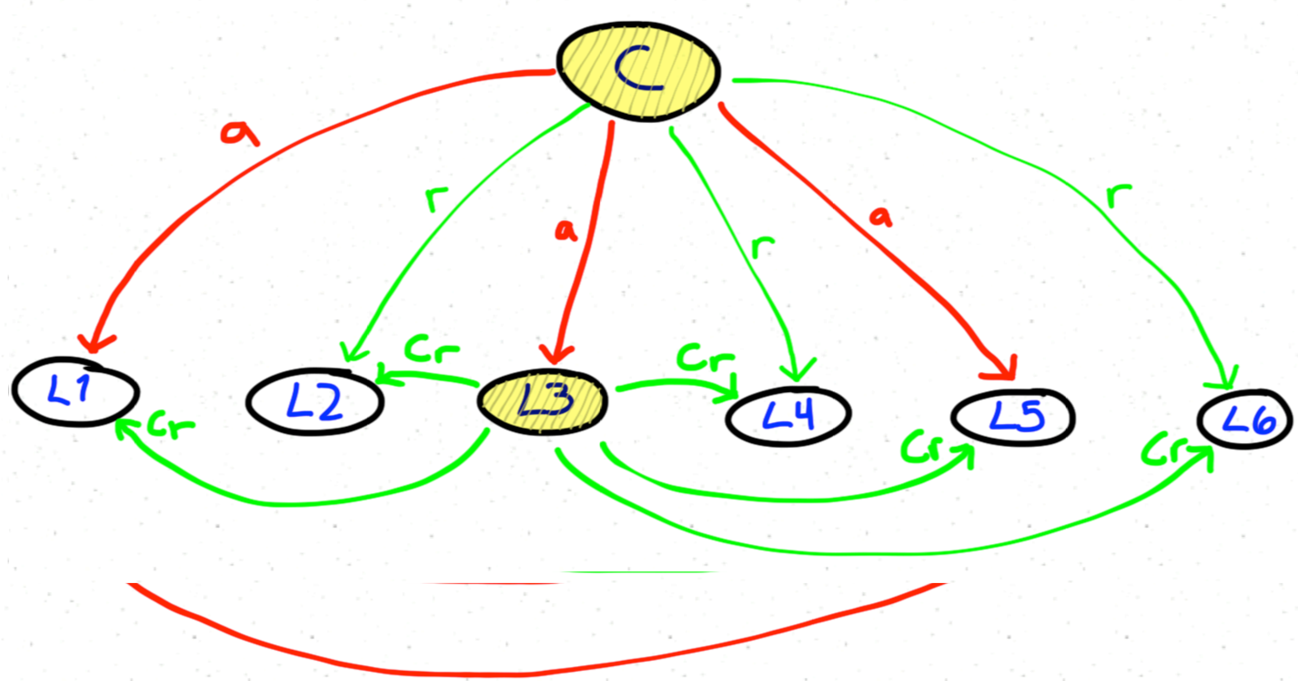
(1) The commander sends his value to every lieutenant.

(2) For each *i*, v_i be the value Lieutenant *i* receives from the commander, or else RETREATE 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* <> *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 *majority*(v_1, …, v_{n-1}).

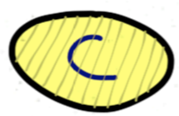
Algorithm OM: Example (1/4)

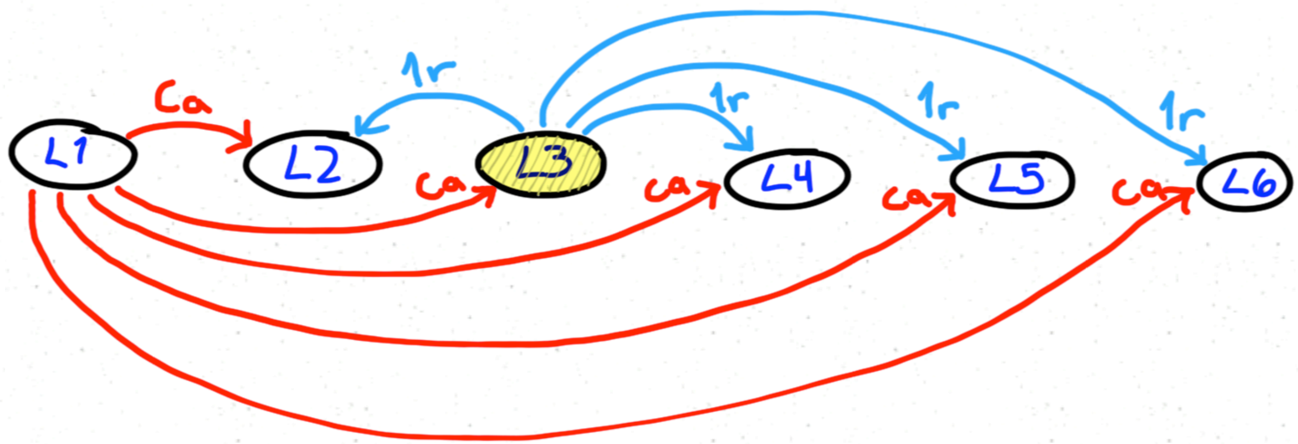
Generals	L1	L2	L3	L4	L5	L6	
C	A	R	A	R	A	R	**OM(2)**
L3	X	CR	CR	CR	CR	CR	**OM(1)**



Algorithm OM: Example (2/4)

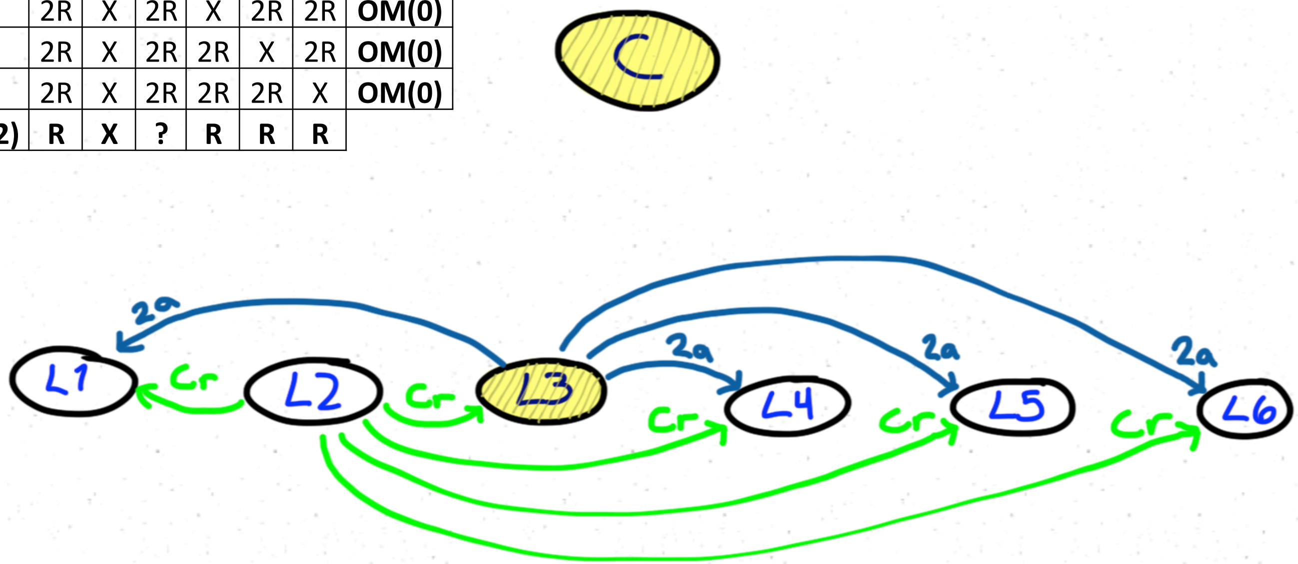
Generals	L1	L2	L3	L4	L5	L6	
C	A	R	A	R	A	R	**OM(2)**
L1	X	CA	CA	CA	CA	CA	**OM(1)**
L2	X	X	1A	1A	1A	1A	**OM(0)**
L3	X	1R	X	1R	1R	1R	**OM(0)**
L4	X	1A	1A	X	1A	1A	**OM(0)**
L5	X	1A	1A	1A	X	1A	**OM(0)**
L6	X	1A	1A	1A	1A	X	**OM(0)**
MAJORITY (L1)	**X**	**A**	**?**	**A**	**A**	**A**	





Algorithm OM: Example (3/4)

Generals	L1	L2	L3	L4	L5	L6	
C	A	R	A	R	A	R	**OM(2)**
L1	X	X	2R	2R	2R	2R	**OM(0)**
L2	CR	X	CR	CR	CR	CR	**OM(1)**
L3	2A	X	X	2A	2A	2A	**OM(0)**
L4	2R	X	2R	X	2R	2R	**OM(0)**
L5	2R	X	2R	2R	X	2R	**OM(0)**
L6	2R	X	2R	2R	2R	X	**OM(0)**
MAJORITY(L2)	**R**	**X**	**?**	**R**	**R**	**R**	



Algorithm OM: Example (4/4)

Decision time…

Generals	L1	L2	L3	L4	L5	L6
MAJORITY(L1)	A	A	?	A	A	A
MAJORITY(L2)	R	R	?	R	R	R
MAJORITY(L3)	R	R	?	R	R	R
MAJORITY(L4)	R	R	?	R	R	R
MAJORITY(L5)	A	A	?	A	A	A
MAJORITY(L6)	R	R	?	R	R	R
MAJORITY	**R**	**R**	**?**	**R**	**R**	**R**

A solution: Signed Messages (1/2)

- BGP is difficult because traitors can lie.

A4.

(a) A loyal general’s signature cannot be forged, and any alteration of the contents of his signed messages can be detected.

(b) Anyone can verify the authenticity of a general’s signature.

- It works with m traitors for any number of generals.

A solution: Signed Messages (2/2)

Algorithm SM(m).

Initially $V_i = \{\}$.

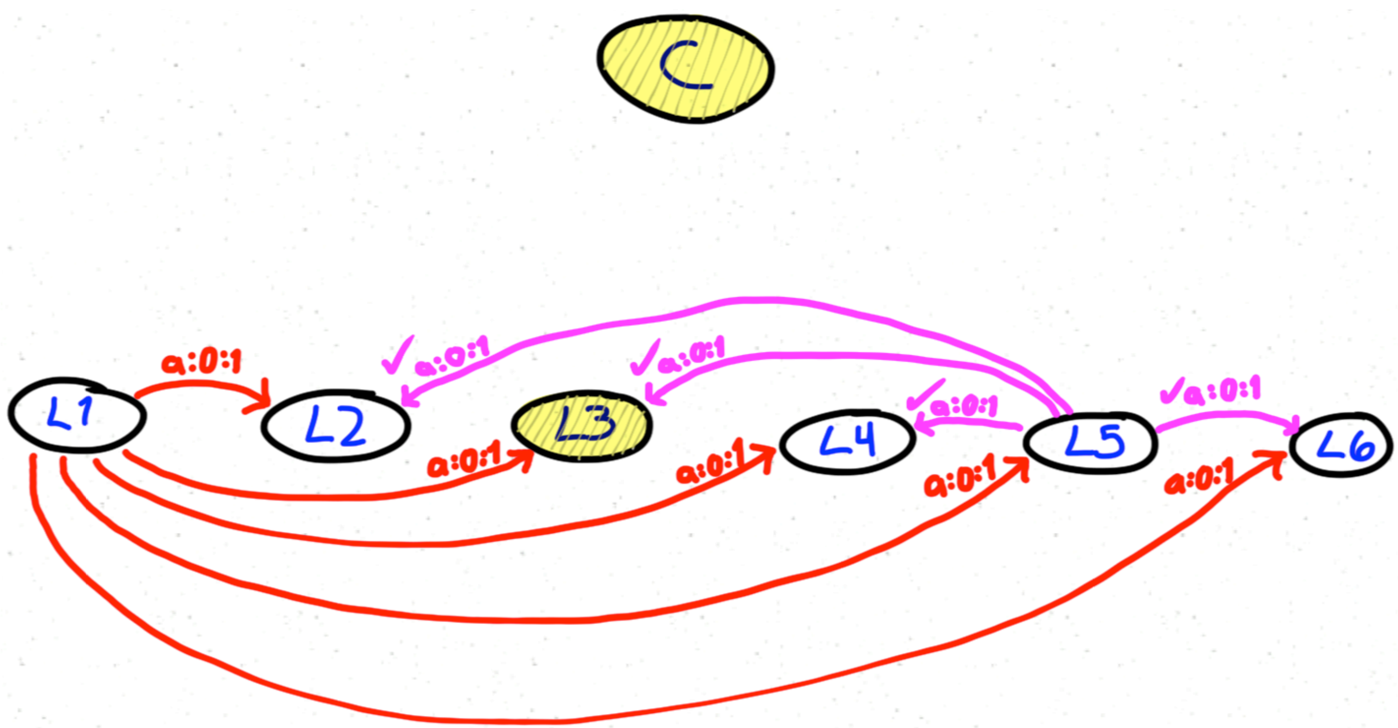
1. The commander signs and sends his value to every lieutenant.

2. For each i:
   A. If Lieutenant i receives a message of the form $v{:}0$ from the commander and he has not yet received any order, then
      i. He lets V_i equal $\{v\}$;
      ii. He sends the message $v{:}0{:}i$ to every other lieutenant.
   B. If lieutenant i receives a message of the form $v{:}0{:}j_1{:}\ldots{:}j_k$ and v is not in the set V_i, then
      i. He adds v to V_i;
      ii. If $k < m$, then he sends the message $v{:}0{:}j_1{:}\ldots{:}j_k{:}i$ to every lieutenant other than $j_1,\ldots j_k$.

3. For each i: When Lieutenant i will receive no more messages, he obeys the order $choice(V_i)$.

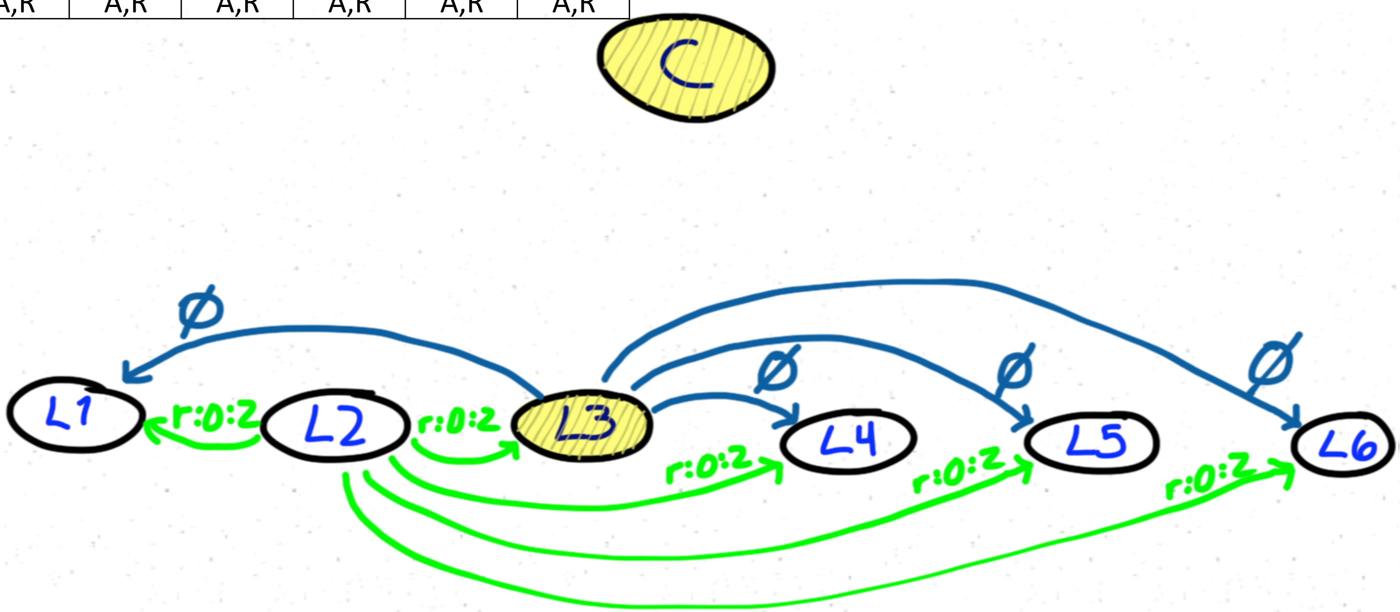
Algorithm SM: Example (1/4)

Generals	L1 L1	L2 L2	L3 L3	L4 L4	L5 L5	L6 L6
C	A:A:0	R:0 R:0	A:0 A:0	R:0 R:0	A:0 A:0	R:0 R:0
Mi / M1	A -	A:R:A:0:1	A:0:A:0:1	A:R:A:0:1	A:0:A:0:1	A:R:A:0:1
M2	A -	A,R -	A A:0:1:2	A,R:0:1:2	A A:0:1:2	A,R:0:1:2
M5 / Vi	A	A,R / ACK	A / ACK	A,R / ACK	A	A,R / ACK
Vi	A	A,R	A	A,R	A	A,R

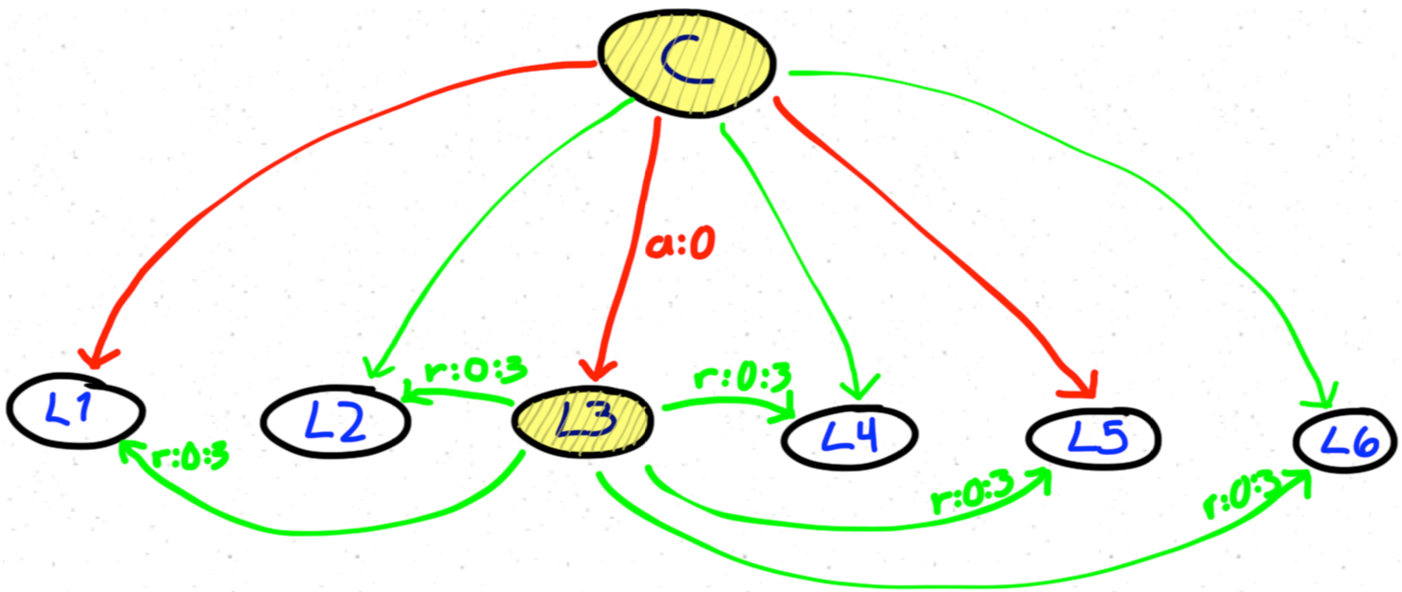


Algorithm SM: Example (2/4)

Generals	L1	L2	L3	L4	L5	L6
C	A:0	R:0	A:0	R:0	A:0	R:0
L1	-	A:0:1	A:0:1	A:0:1	A:0:1	A:0:1
L2	-	-	A:0:1:2	A:0:1:2	A:0:1:2	A:0:1:2
L5	-	ACK	ACK	ACK	-	ACK
L2	R:0:2	-	R:0:2	R:0:2	R:0:2	R:0:2
L1 / Vi	- / A,R	R:0:2:1 / A,R	R:0:2:1 / A,R	R:0:2:1 / A,R	R:0:2:1 / A,R	R:0:2:1 / A,R
L3 / Vi	R / A,R	R / A,R	R / A,R	R / A,R	R / A,R	R / A,R
Vi	A,R	A,R	A,R	A,R	A,R	A,R



Algorithm SM: Example (3/4)



+ Algorithm SM: Example (4/4)

Decision time…

Generals	**L1**	**L2**	**L3**	**L4**	**L5**	**L6**
CHOICE(V)	A, R	A, R	?	A, R	A, R	A, R

Missing Communication Paths

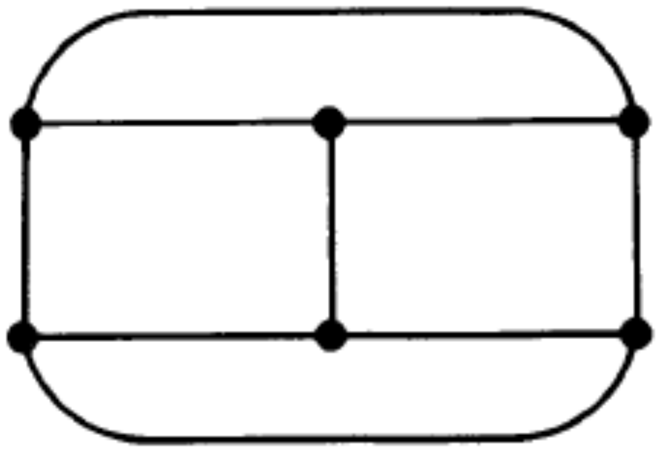


Fig. 6. A 3-regular graph.

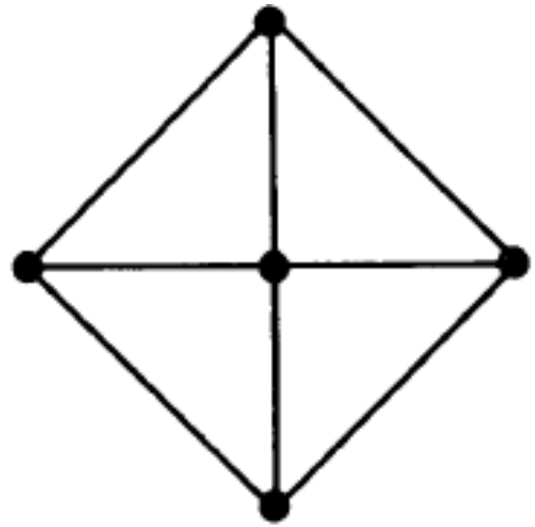


Fig. 7. A graph that is not 3-regular.

Algorithm OM(m,p) [Remarks]

- BGP is solved by OM(m, 3m) (3m + 1 generals minimum)
- If one lieutenant is unreachable, more than half of his/her paths connect with loyal lieutenants
- Recursively name one of your lieutenants as the new commander and send the order

- Applying OM(m, 3m) with 3m+1 generals is the same as OM(m)!

Algorithm SM(m) weakly connected [Remarks]

- Can have missing links
- Requires subgraph of loyals is connected
- Can be solved with SM(n-2)

BGP in practice

- Majority voting as a way to provide reliability
- What does it take to work?
  - Input synchronization (IC1)
  - If input is non-faulty, all non-faulty processes provide same output (IC2)
- A1 – communication line vs node failure
  - No problem: OM(m) or SM(m) can deal with it
- A2 – Fixed lines vs switching network
  - Not needed if A4 is assumed
- A3 - Timeouts
- A4 - Cryptography

Final Thoughts [Conclusion]

- Reliability involves coping with failure of components
- Two solutions: Oral and Signed Messages
- Expensive:
  - Time: time spent with signatures and message latencies
  - Messages: message paths $\geq$ m+1; O(n^(m-1)) messages;
- BGP used for input synchronization, handling m faults

Final Thoughts [Discussion]

- Can we afford 3m+1 nodes?
- How to determine m?
- How do you build a sub-graph of loyal lieutenants in a hostile environment?
- What alternatives can we implement:
  - On a secure environment (just dealing with failures)?
  - On a hostile environment (dealing with traitors)?

+

BGP: Example

From Amazon's S3 Service Health Dashboard (July 7, 2008):

"We've now determined that message corruption was the cause of the server-to-server communication problems. More specifically, we found that ***there were a handful of messages*** *on Sunday morning* ***that had a single bit corrupted such that the message was still intelligible****, but the system state information was incorrect. We use MD5 checksums throughout the system, for example, to prevent, detect, and* ***recover from corruption*** *that can occur during receipt, storage, and retrieval* ***of customers' objects****. However,* ***we didn't have the same protection in place to detect whether this particular internal state information had been corrupted****. As a result, when the corruption occurred, we didn't detect it and it spread throughout the system causing the symptoms described above. We hadn't encountered server-to-server communication issues of this scale before and, as a result, it took some time during the event to diagnose and recover from it."*

Reference: [2]

+

Proofs*

*Available upon request

3 Generals cannot handle 1 Traitor

- If Commander is loyal, IC2 is always satisfied and IC1 follows from IC2
- If Commander is traitor, then:
  - Lieutenant 1 will attack
  - Lieutenant 2 will retreat
- IC1 is violated!

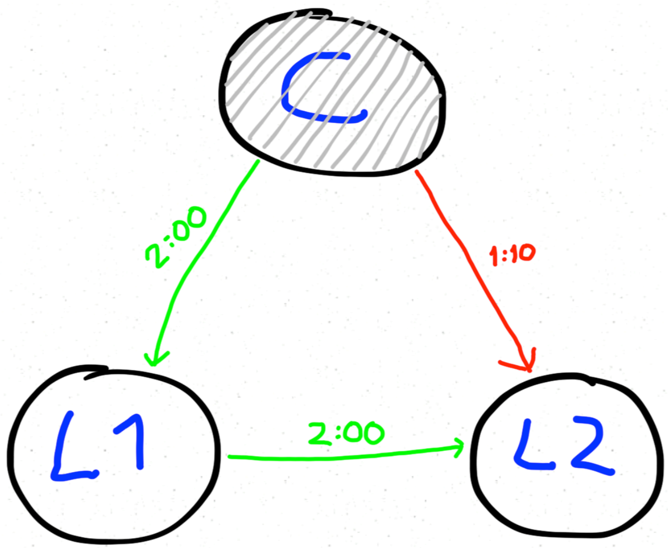
No solution with fewer than 3m+1 exist

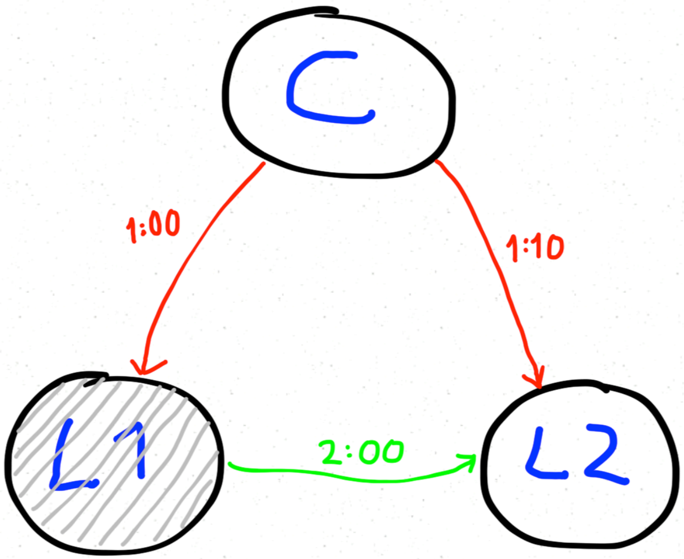
- Assume a solution with 3m or fewer exist
- Call the 3m generals and the m traitors the *Albanian Generals*
- Each Byzantine general "*simulates*" at most m Albanian generals
- The Byzantine commander simulates the Albanian commander and m-1 Albanian generals
- Since only one Byzantine general can be a traitor, and he simulates at most m Albanians, at most m Albanians are traitors.
- By previous proof, no solution exists.
- Contradiction!

Approximate agreement (1/2)

- IC1'. All loyal lieutenants attack within 10 minutes of one another.
- IC2'. If the commanding general is loyal, then every loyal lieutenant attacks within 10 minutes of the time given in the commander's order.
- After receiving the attack time from the commander, a lieutenant does one of the following:
  - If the time is 1:10 or earlier, then attack.
  - If the time is 1:50 or later, then retreat.
  - Otherwise, continue to step (2).
- Ask the other lieutenant what decision he reached in step (1).
  - If the other lieutenant reached a decision, then make the same decision he did.
  - Otherwise, retreat.

Approximate agreement (2/2)





Lemma 1 – OM(m)

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

- Only prove when commander is loyal. By induction:
  - If commander is loyal, OM(0) is trivial.
  - Assume is true for m-1, m > 0
    - Step (1), commander sends v to all $n-1$ lieutenants
    - Step (2), loyal lieutenants apply OM($m-1$) with $n-1$ generals
    - Since $n > 2k + m$, $(n-1) > [2k + (m-1)]$, by induction hypothesis you can conclude every loyal lieutenant gets $v_j = v$ for each loyal lieutenant j.
    - Since there are at most k traitors, and $(n-1) > [2k + (m-1)] \geq 2k$, a majority of the $n-1$ lieutenants are loyal. Hence, each loyal lieutenant has $v_i = v$ for a majority of the $n-1$ values i, so he obtains $majority(v_1,\ldots v_{n-1}) = v$ in step (3), proving IC2.

Theorem 1 – OM(m)

- For any m, Algorithm OM(m) satisfies conditions IC1 and IC2 if there are more than 3m generals and at most m traitors.
- Using induction on m:
  - If there are no traitors, OM(0) is trivial and satisfies IC1 and IC2.
  - Assume OM(m-1) is true and prove for OM(m):
    - If commander is loyal, make k=m in lemma 1, OM(m) satisfies IC2. IC1 follows from IC2 if commander is loyal.
    - Only prove IC1 when commander is traitor:
      - There are at most m-1 traitors (commander is one of them)
      - OM(m-1) satisfies IC1 and IC2.
      - For each j, any two loyal lieutenants get the same value for v_j in step (3) and the same value for $majority(v_1,\ldots v_{n-1})$.

How to determine no more messages will be sent?

- By induction on k,
  - A sequence of lieutenants j1,…,jk with $k \le m$
  - A lieutenant can receive at most one message of the form v:0:j1:…:jk in step (2)
  - By A3, the lieutenant will have to sign and send the message or send a message reporting that he will not send the message.

+ Theorem 2.

- Theorem 2. For any m, Algorithm SM(m) solves the Byzantine Generals Problem if there are at most m traitors.
  - First prove IC2:
    - Commander sends v:0 to every lieutenant in step (1).
    - Every loyal lieutenant will receive the order in step (2)(A).
    - Since no traitor can forge the order, then loyal lieutenants do not receive additional orders on step (2)(B).
    - The set Vi consists only of v.
  - IC1 only needs to be proved when the commander is a traitor:
    - Only need to prove that if i puts order v into Vi in step(2), then j must put the same order into Vj:
      - If i receives the order v in step (2)(A), then he sends it to j in step (2)(A)(ii); so j receives it (by A1).
      - If I adds the order to Vi in step (2)(B), then he must have received v:0:j1:…:jk at some point. If j is one of the jr, then by A4 he must already received the order v. If not:
        - k < m, i sends it to j
        - k=m, since the commander is a traitor, then at most m-1 of the lieutenants are traitors. One of the loyal lieutenants must have signed the order and sent it to j.

Lemma 2

- Lemma 2. For any $m>0$ and any $p \geq (2k + m)$, Algorithm OM(m,p) satisfies IC2 if there are at most k traitors.
  - For $m=1$
    - Lieutenant obtains majority($v_1, \ldots, v_p$) where each vi is sent along a path disjoint from other paths.
    - Since $p \geq (2k + m)$, more than half of the paths are composed by loyal lieutenants
    - The majority of the values will be the same as that of the commander
  - Assume for $m-1$, $m > 1$
    - If commander is loyal each of the p lieutenants in N gets the correct value.
    - Since $p>2k$, a majority of them are loyal and by hypothesis, each one of them sends the correct value.

Theorem 3

- Theorem 3. For any $m>0$ and any $p \geq 3m$, Algorithm OM(m,p) solves BGP if there are at most m traitors.
  - By lemma 2, $k=m$ solves IC2.
  - Prove IC1 when commander is traitor:
    - If $m=1$ all lieutenants get the same values in step (4) because paths don't go through the commander.
    - If $m>1$, apply induction
    - $m=0$ is trivial
    - Assume $m-1$ is true
      - Since commander is traitor, you have $p-1$ other lieutenants which $(p-1) \geq (3m-1) \geq 3(m-1)$
      - $(p-1)/3 \geq (m-1)$, by induction hypothesis holds true and all loyal lieutenants apply majority

Theorem 4

- Theorem 4. For any m and d, if there are at most m traitors and the subgraph of loyal generals has diameter d, then Algorithm SM(m + d -1) solves the Byzantine Generals Problem.
  - First prove IC2:
    - Commander sends v:0 to every loyal lieutenant in step (1) (guaranteed by hyp).
    - Every loyal lieutenant will receive the order in step (2)(A).
    - Since no traitor can forge the order, then loyal lieutenants do not receive additional orders on step (2)(B).
    - The set Vi consists only of v.
  - IC1 only needs to be proved when the commander is a traitor:
    - Only need to prove that if i puts order v into Vi in step(2), then j must put the same order into Vj:
      - If i receives the order v in step (2)(A), then he sends it to j in step (2)(A)(ii); so j receives it (by A1).
      - If I adds the order to Vi in step (2)(B), then he must have received v:0:j1:….:jk at some point. If j is one of the jr, then by A4 he must already received the order v. If not:
        - k < m, i sends it to j
        - k=m, since the commander is a traitor, then at most m-1 of the lieutenants are traitors. One of the loyal lieutenants must have signed the order and sent it to j.

Corollary

- If the diameter of the sub-graph of loyal lieutenants is d, then there must be d+1 lieutenants.
- Therefore, $m = n - d - 1$.
- By theorem 4, $SM(m+d-1) = SM(n-d-1+d-1) = SM(n-2)$

Reference

[1] Leslie Lamport, Robert Shostak and Marshall Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3): 382-401, July 1982.

[2] "AWS S3 Availability Event", http://status.aws.amazon.com/s3-20080720.html . Amazon Rec. 10/2013