Introduction to Robust Algorithms

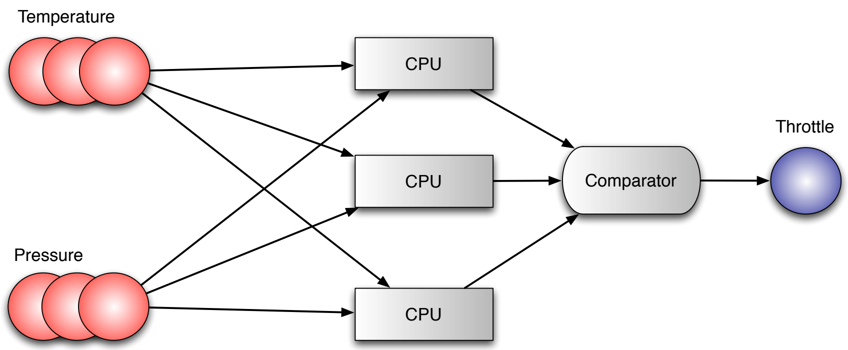
Sébastien Tixeuil
UPMC & IUF

Motivation

Approach

- *Faults* and *attacks* occur in the network
- The network's user must *not* notice something wrong happened
- A *small* number of faulty components
- **Masking** approach to fault/attack tolerance

Principle



Problems

- Replicated input sensors may not give the same data
- Faulty input sensor or processor may not fail gracefully
- The system might not be tolerant to software bugs

Telling Truth from Lies

The Island of Liars and Truth-tellers

- An island is populated by two tribes
  - Members of one tribe **consistently lie**
  - Members of the other tribe **always tell the truth**
  - Tribe members can recognize one another, but an external observer can't

Puzzle 1

- You meet a man and ask him if he is a truth-teller, but fail to hear the answer
- You inquire: "Did you say you are a truth-teller?"
- He responds: "No, I did not."
- To which tribe does the man belong ?

Puzzle II

- You meet a person on the island.
- What single question can you ask him/her to determine whether he/she is a liar or a truth-teller?

Puzzle III

- You meet two people *A* and B on the island
- *A* says: "Both of us are from the liar tribe."
- Which tribe is *A* from ?
- What about *B* ?

Puzzle IV

- You meet two people, *C* and *D* on the island.
- *C* says: "Exactly one of us is from the liars tribe."
- Which tribe is *D* from ?

Puzzle V

- You meet two people *E* and *F* on the island
- *E* says: "It is not the case that both of us are from truth-tellers tribe."
- Which tribe is *E* from?
- What about *F*?

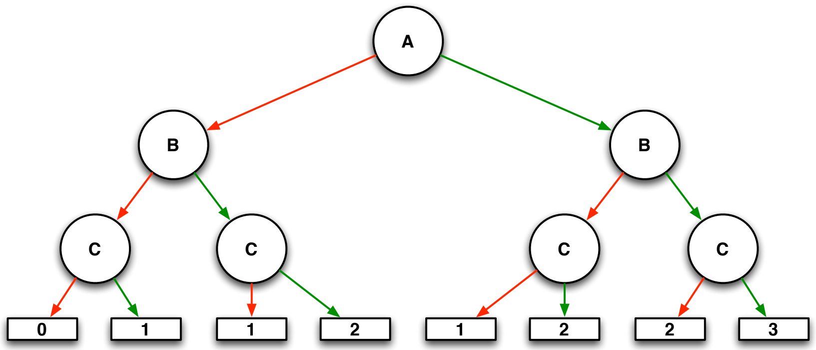
Puzzle VI

- You meet two people *G* and *H* on the island
- *G* says: "We are from different tribes."
- *H* says: "*G* is from the liars tribe."
- Which tribes are *G* and *H* from ?

Puzzle VII

- You meet three people *A*, *B*, and *C*
- You ask *A*: "how many among you are truth-tellers?", but don't hear the answer
- You ask *B*: "What did *A* say?", hear "one."
- *C* says: "*B* is a liar."
- Which tribes are *B* and *C* from?

Puzzle VII



The Island of Selective Liars

- Inhabitants lie consistently on Tuesdays, Thursdays, and Saturdays. However, they always say the truth on the remaining days.
- You ask: "What is today?" "Tomorrow?"
- Responses: "Saturday.", "Wednesday."
- What is the current day ?

The Island of Random Liars

- A new Island has three tribes
  - truth-tellers
  - consistent liars
  - randomly lie or tell the truth
- How to identify three representants of each tribe standing in a line with only three yes/no questions?

Byzantine Generals

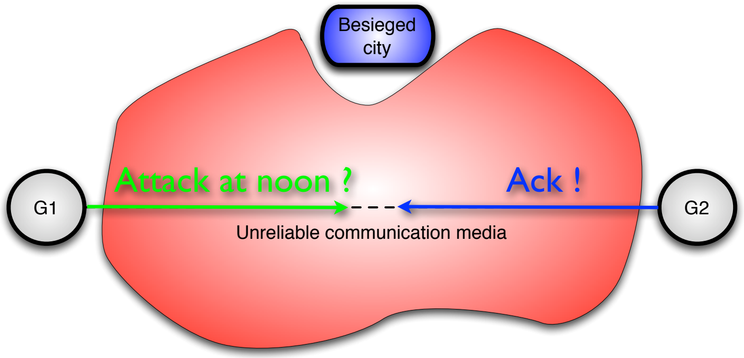
Settings

- Byzantine generals are camping outside an enemy city
- Generals can communicate by sending messengers
- Generals must decide upon common plan of action
- Some of the Generals can be traitors

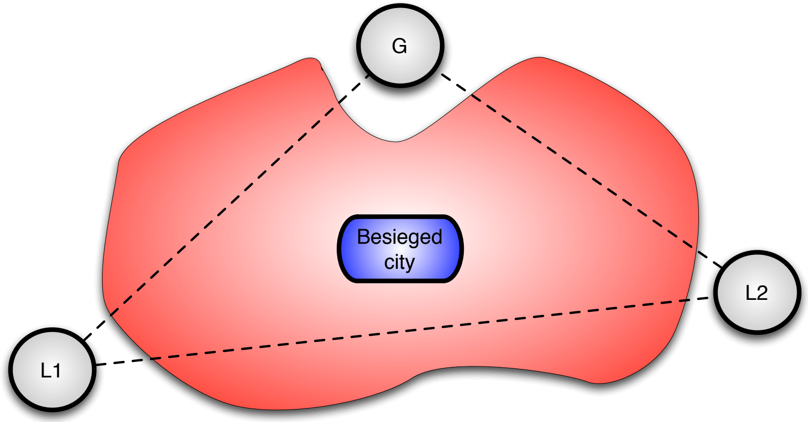
Goal

- All loyal generals decide upon the same plan of action
- A small number of traitors cannot cause the loyal generals to adopt a bad plan

Two Generals Paradox



The Byzantine Generals Problem



The (simple) Byzantine Generals Problem

- Generals lead *n* divisions of the Byzantine army
- The divisions communicate via reliable messengers
- The generals must **agree** on a plan ("attack" or "retreat") even if some of them are **killed** by enemy spies

Oral Model

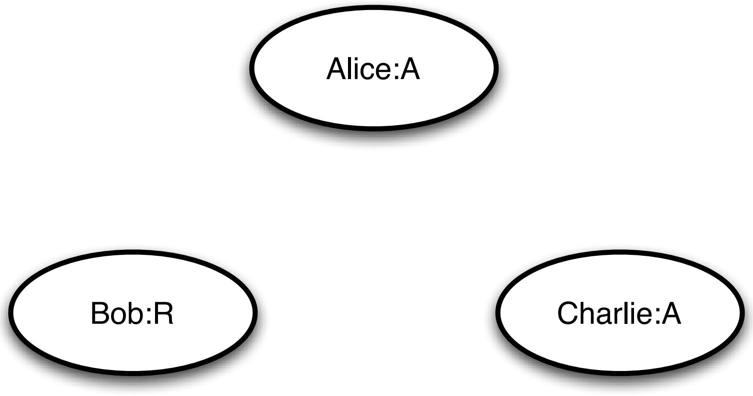
- **A1**: Every message that is sent is delivered correctly
- **A2**: The receiver of a message knows who sent it
- **A3**: The absence of a message can be detected

Solution?

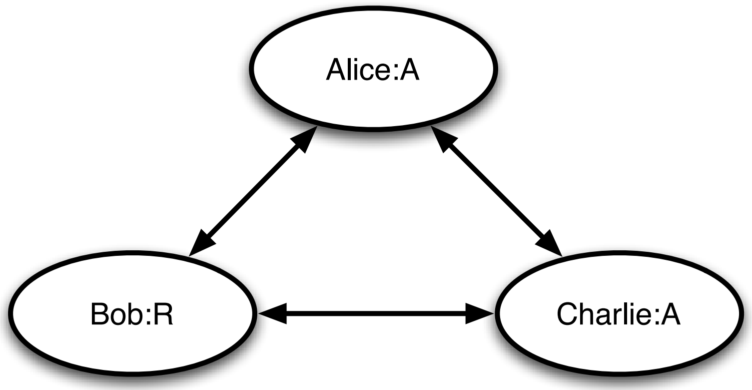
plan: **array of** {A,R}; finalPlan: {A,R}

```
plan[myID] := ChooseAorR()
for all other G send(G, myID, plan[myID])
for all other G receive(G, plan[G])
finalPlan := majority(plan)
```

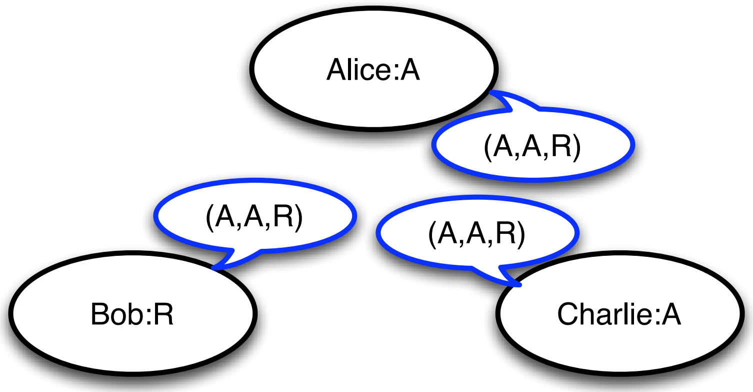
Reliable Networks



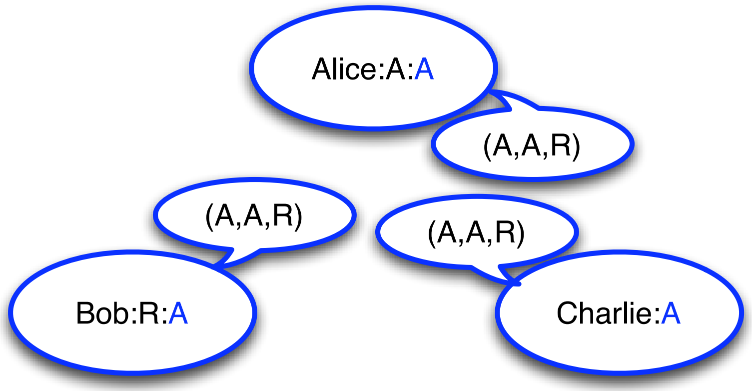
Reliable Networks



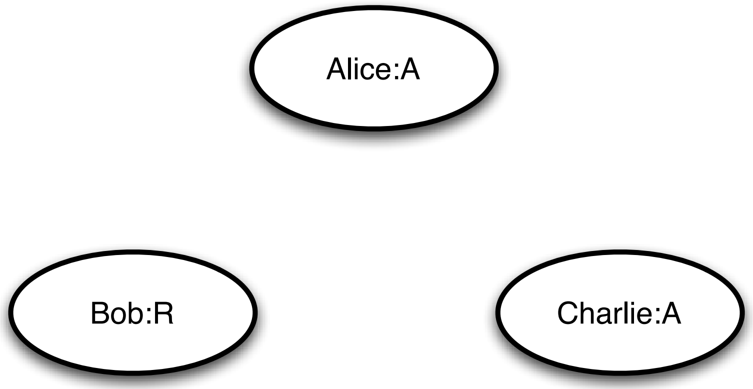
Reliable Networks



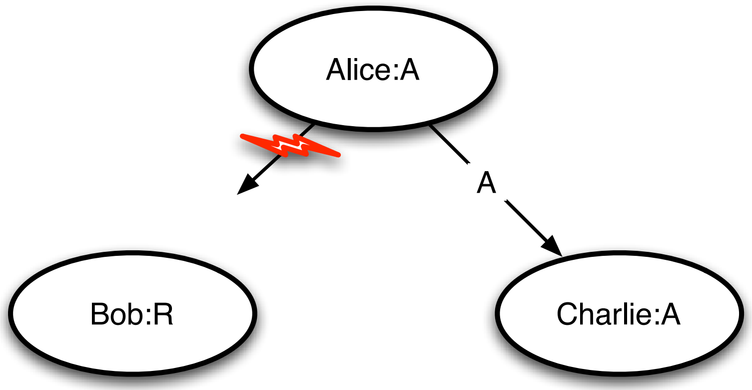
Reliable Networks



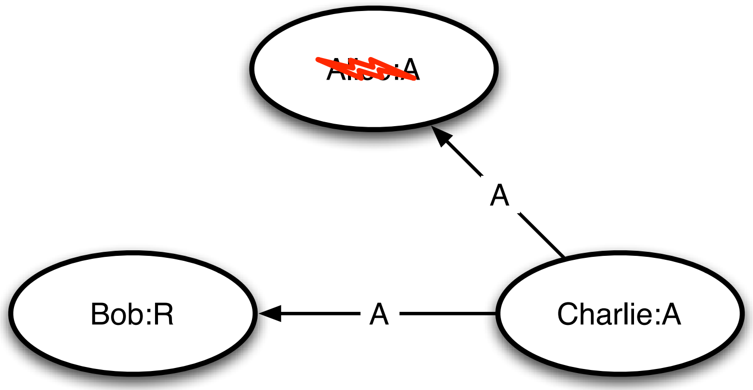
Crashing Networks



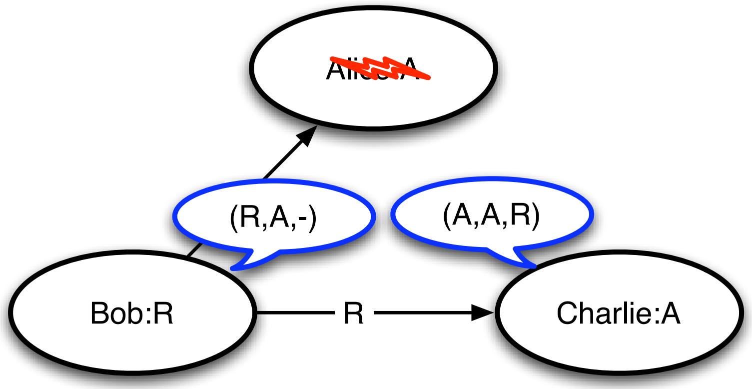
Crashing Networks



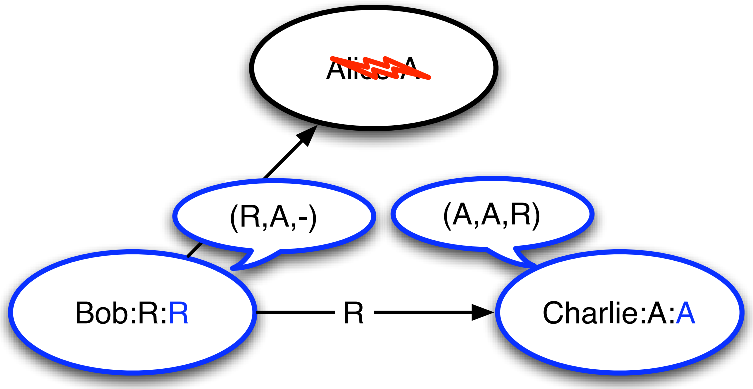
Crashing Networks



Crashing Networks



Crashing Networks



The Byzantine Generals Problem

- A general and *n-1* lieutenants lead n divisions of the Byzantine army
- The divisions communicate via messengers that can be captured or delayed
- The generals must **agree** on a plan ("attack" or "retreat") even if some of them are **traitors** that want to prevent agreement

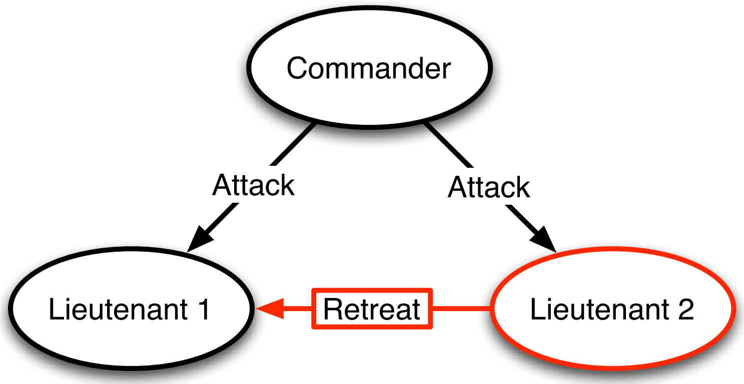
The Byzantine Generals Problem

- A commanding general must sent an order to his n-1 lieutenants 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

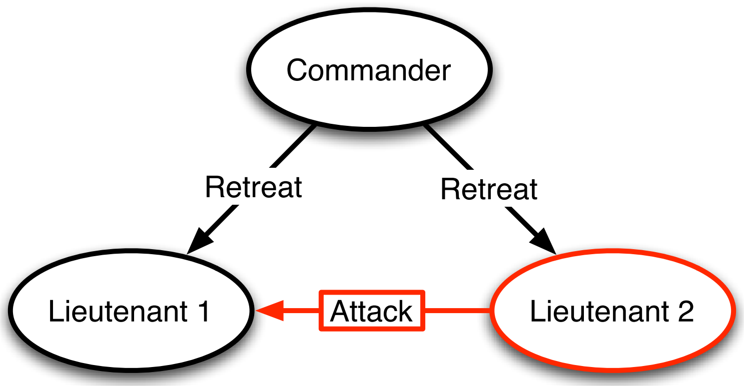
Oral Model

- **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

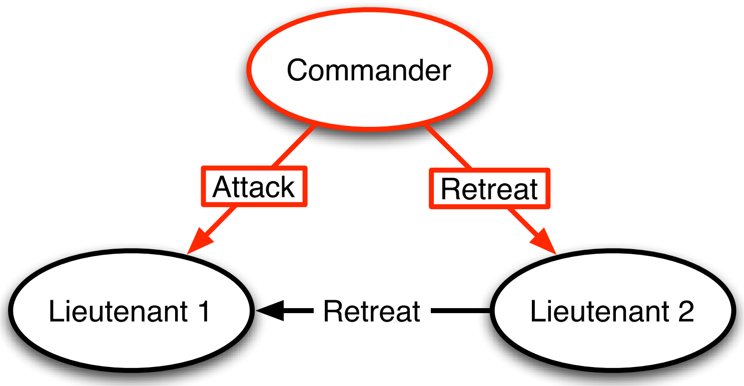
3k+1 nodes are necessary (oral model)



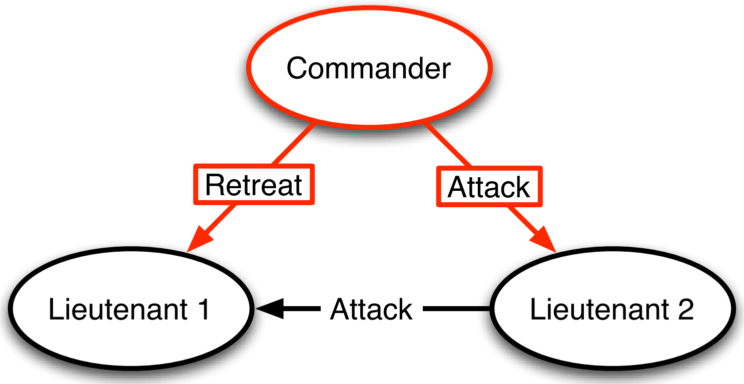
3k+1 nodes are necessary (oral model)



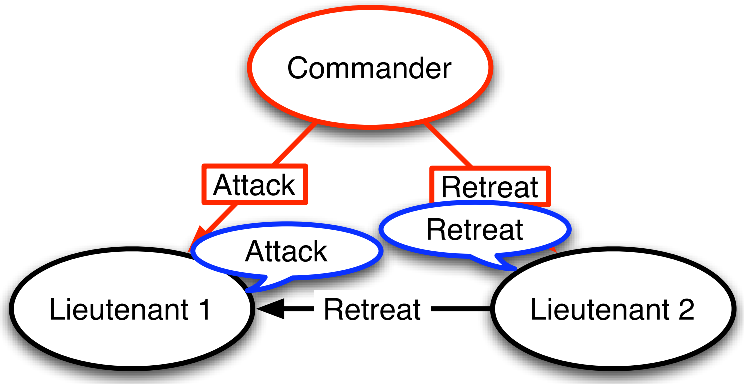
3k+1 nodes are necessary (oral model)



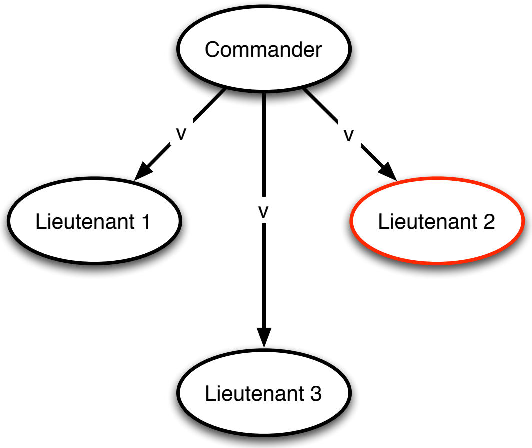
3k+1 nodes are necessary (oral model)



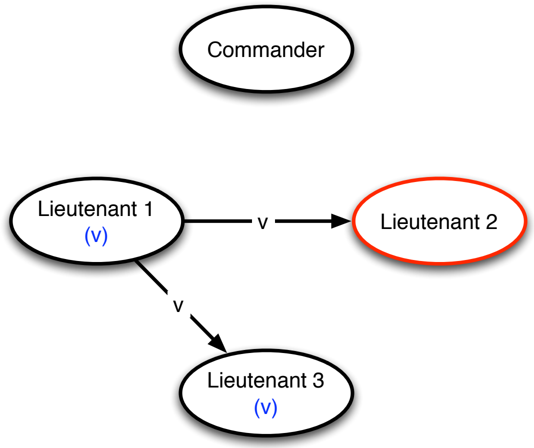
3k+1 nodes are necessary (oral model)



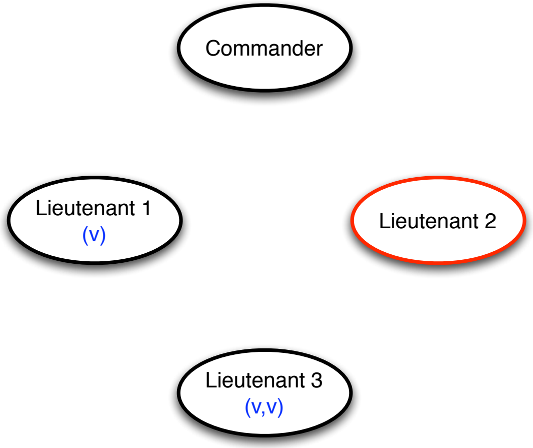
3k+1 nodes are sufficient (oral model)



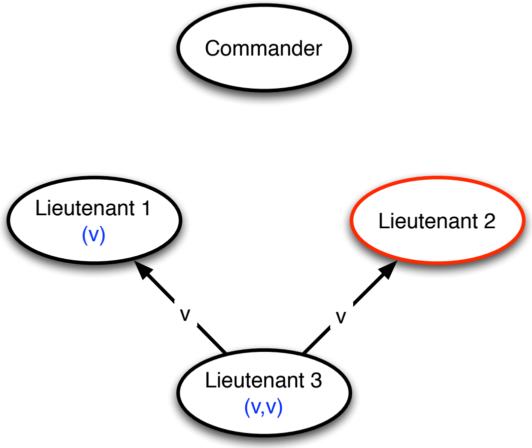
3k+1 nodes are sufficient (oral model)



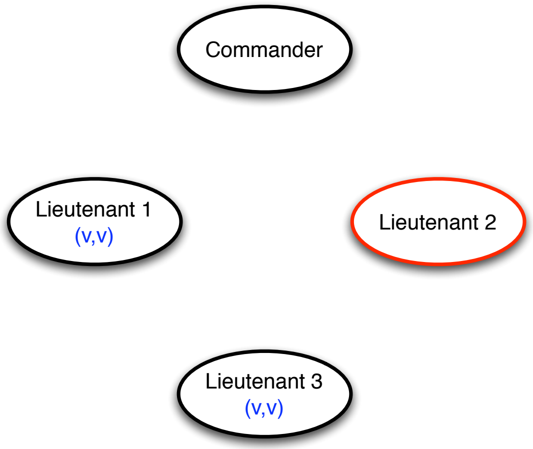
3k+1 nodes are sufficient (oral model)



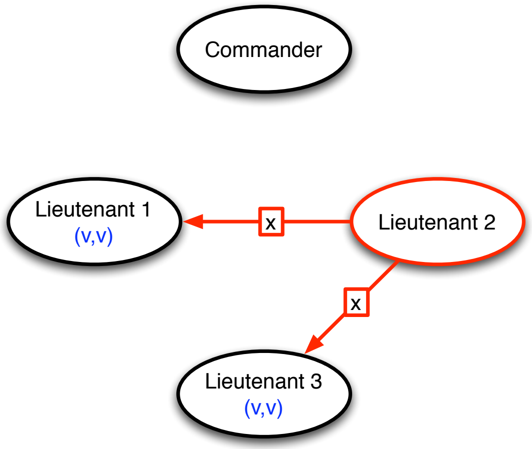
3k+1 nodes are sufficient (oral model)



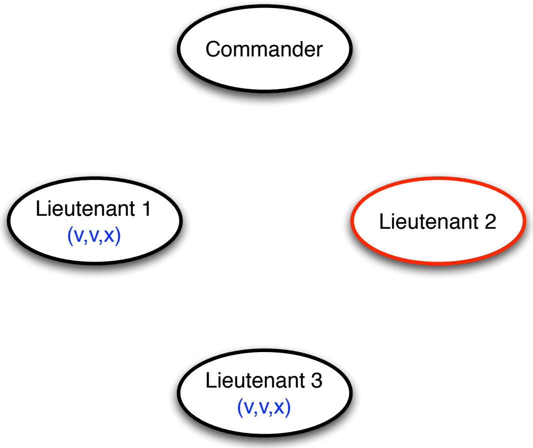
3k+1 nodes are sufficient (oral model)



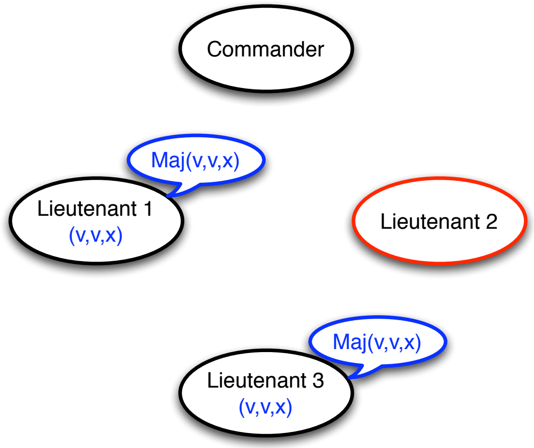
3k+1 nodes are sufficient (oral model)



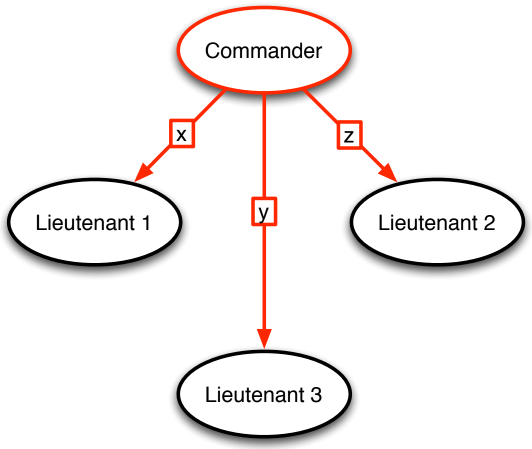
3k+1 nodes are sufficient (oral model)



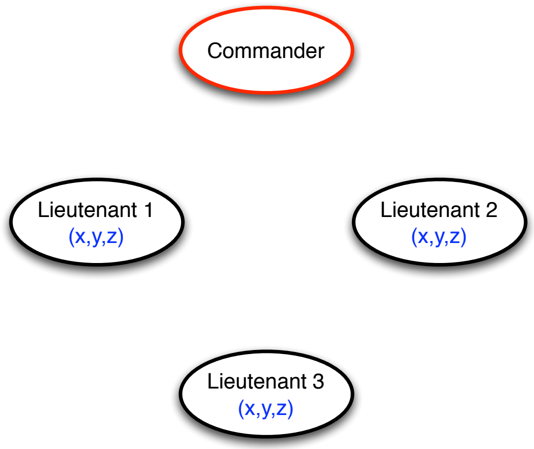
3k+1 nodes are sufficient (oral model)



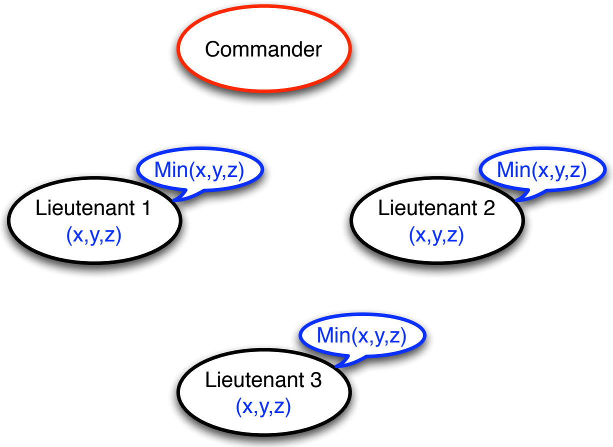
3k+1 nodes are sufficient (oral model)



3k+1 nodes are sufficient (oral model)



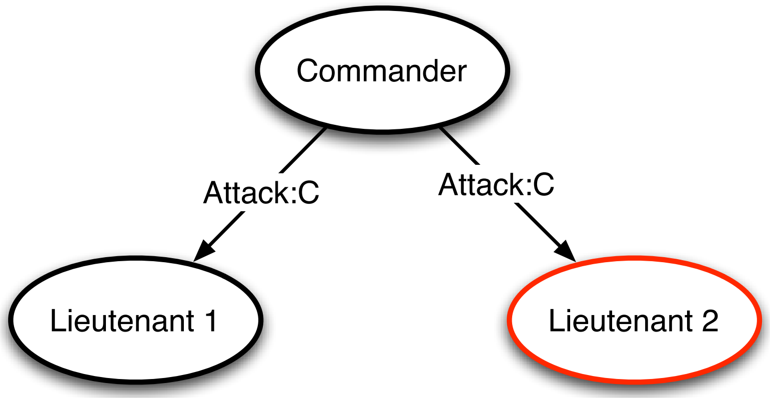
3k+1 nodes are sufficient (oral model)



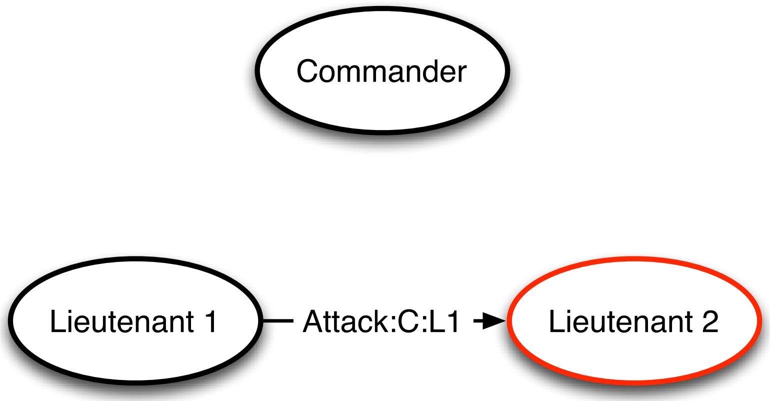
Written Model

- **A1-A3**: Same as before
- **A4**:
  - A loyal general's signature cannot be forged, and any alteration of the contents of his signed messages can be detected
  - Anyone can verify the authenticity of a general's signature

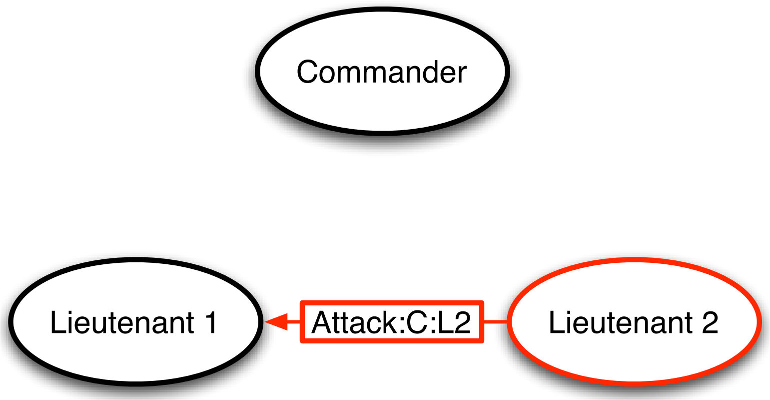
k+2 nodes are sufficient (written model)



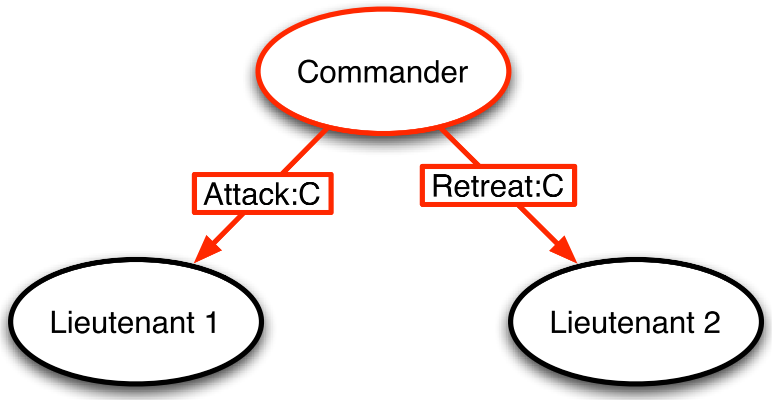
k+2 nodes are sufficient (written model)



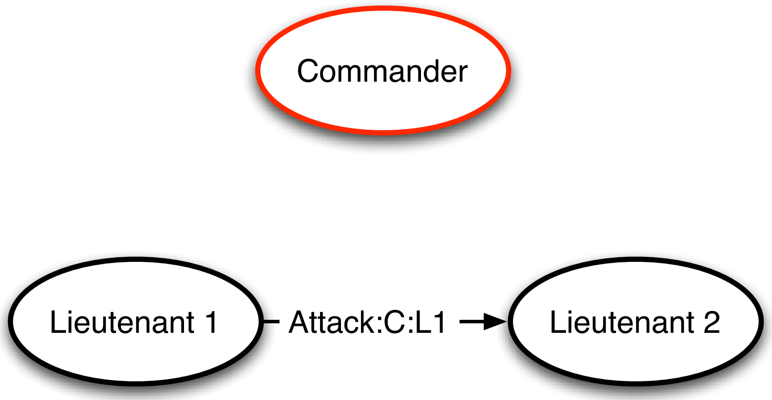
k+2 nodes are sufficient (written model)



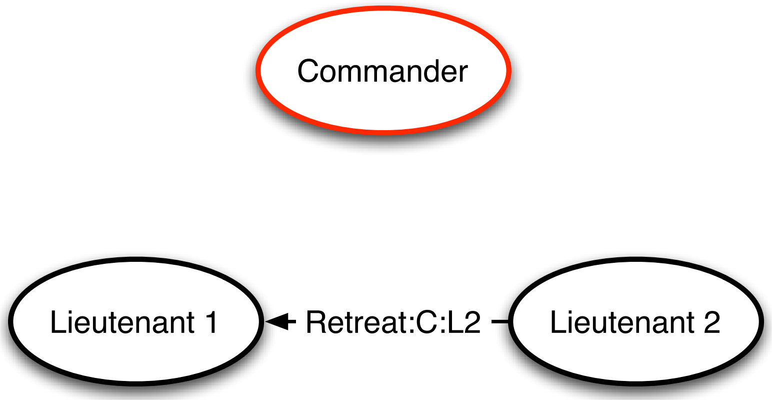
k+2 nodes are sufficient (written model)



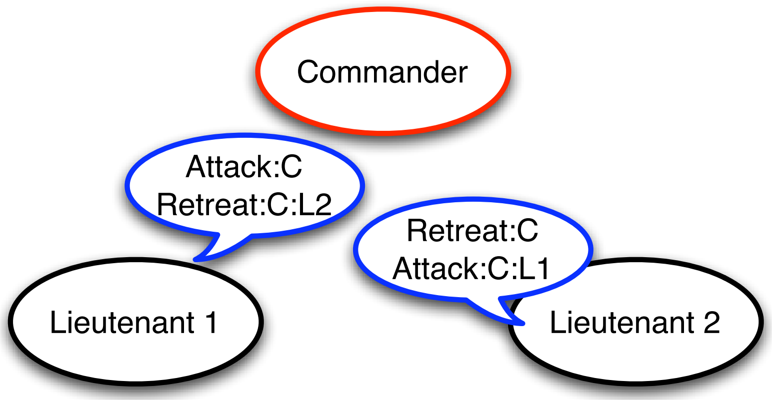
k+2 nodes are sufficient (written model)



k+2 nodes are sufficient (written model)



k+2 nodes are sufficient (written model)



Arbitrary Networks

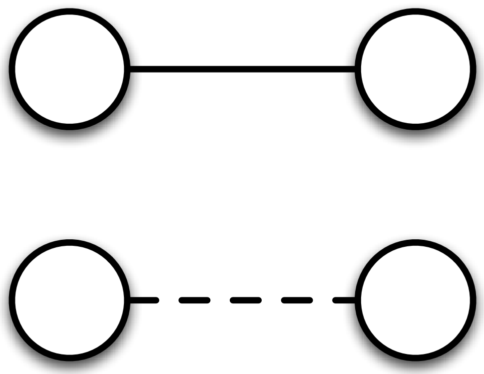
Topology Discovery

- **Given**
  - asynchronous network
  - up to *k* Byzantine nodes
  - each node knows its immediate neighbors identifiers
- **Goal**
  - each node must discover the complete network topology

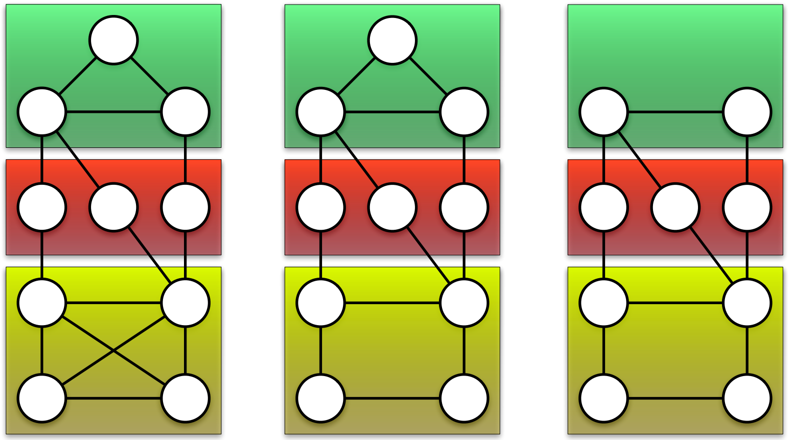
Weak Topology Discovery

- **Termination**
  - either all non-faulty processes determine the system topology or at least one detects fault
- **Safety**
  - for each non-faulty process, the determined topology is subset of actual
- **Validity**
  - fault detected only if it indeed exists

Weak Topology Discovery



Weak Topology Discovery



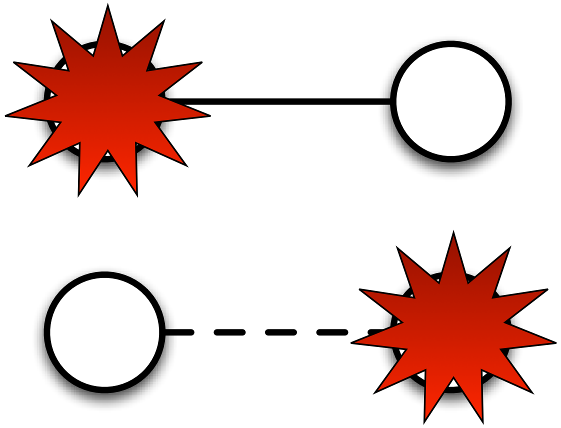
Weak Topology Discovery

- **Bounds**
  - cannot determine presence of edge if two adjacent nodes are faulty
  - cannot be (completely) solved if network is less than *k+1* connected

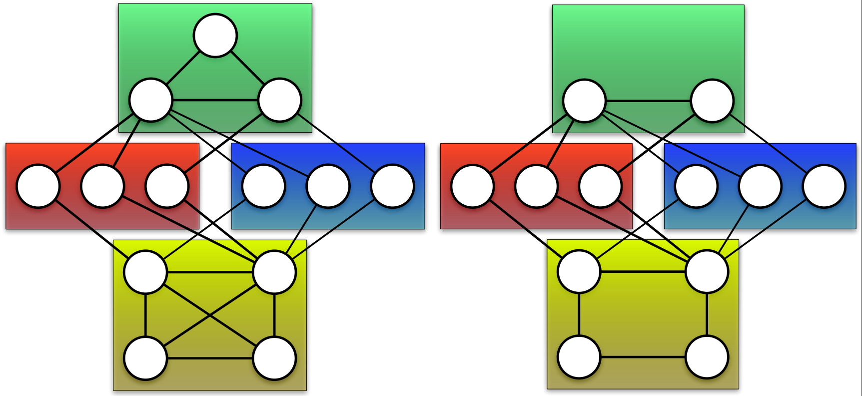
Strong Topology Discovery

- **Termination**
  - all non-faulty processes determine the system topology
- **Safety**
  - for each non-faulty process the determined topology is subset of actual

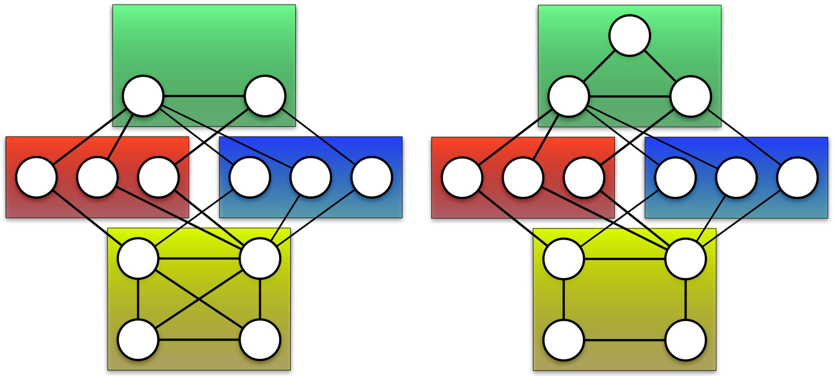
Strong Topology Discovery



Strong Topology Discovery



Strong Topology Discovery



Strong Topology Discovery

- **Bounds**
    - cannot determine presence of edge if one neighbor is faulty
    - cannot be solved if network is less than *2k+1* connected

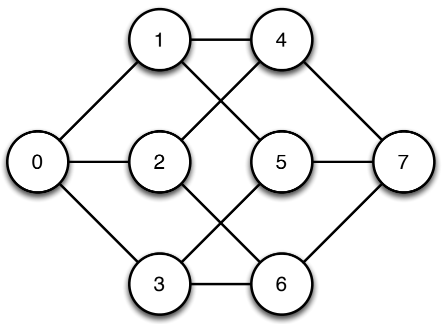
Solutions Preliminaries

- **Main idea**
  - *Menger's theorem*: if a graph is *k* connected then for any two vertices there exists k internally node-disjoint paths connecting them
  - a single (non-source) node cannot compromise info if it travels over two node-disjoint paths

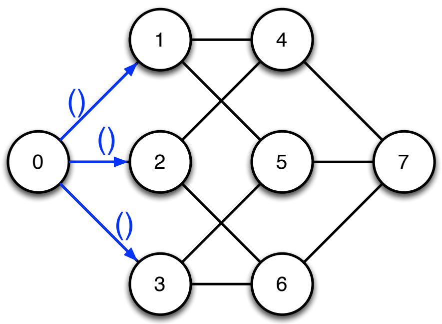
Dolev's Algorithm

- Store traveled path in message, forward message that contains simple path to all outgoing links
- Accept message if received through k+1 node-disjoint paths

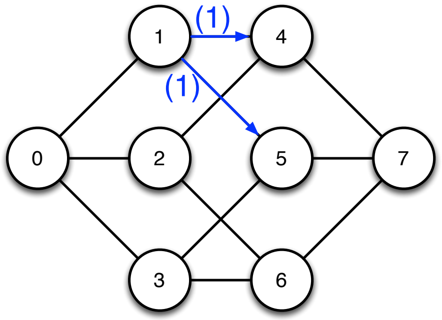
Dolev's Algorithm



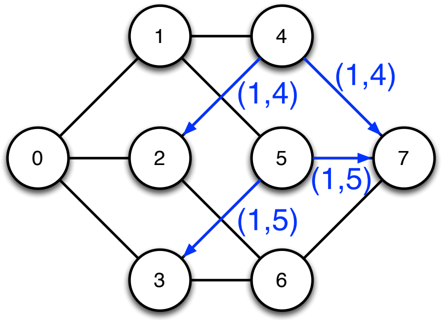
Dolev's Algorithm



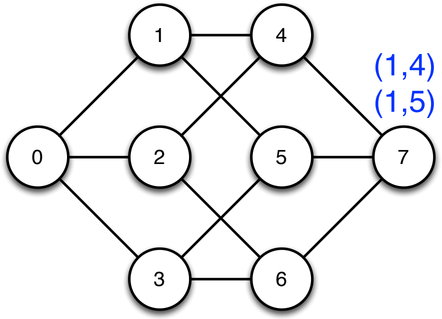
Dolev's Algorithm



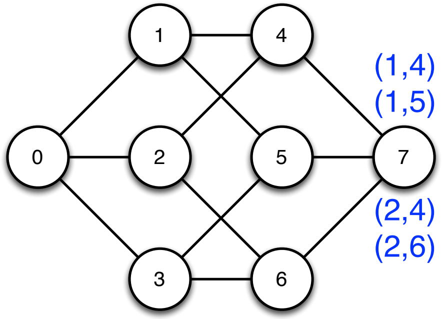
Dolev's Algorithm



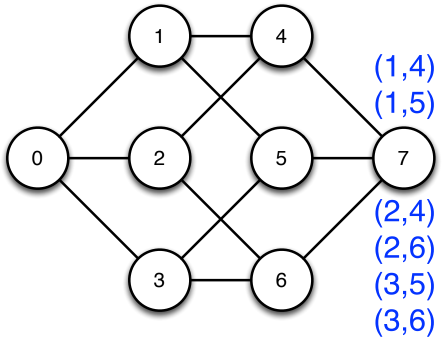
Dolev's Algorithm



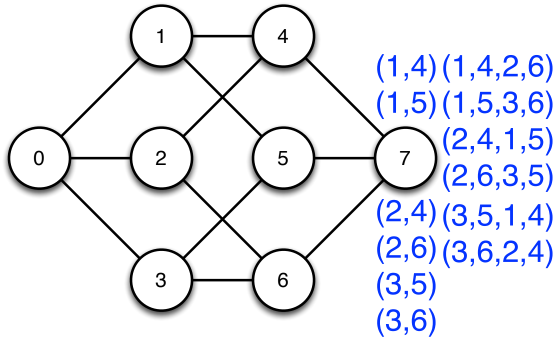
Dolev's Algorithm



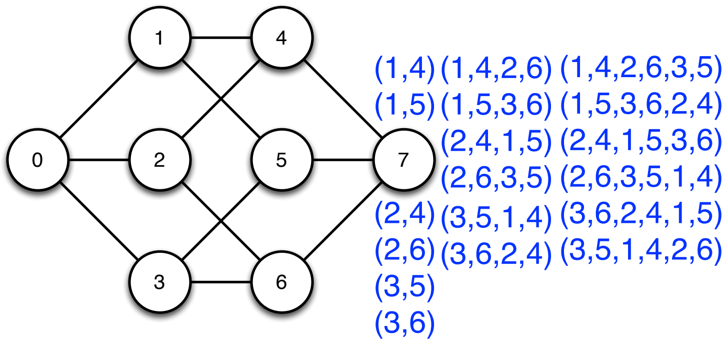
Dolev's Algorithm



Dolev's Algorithm



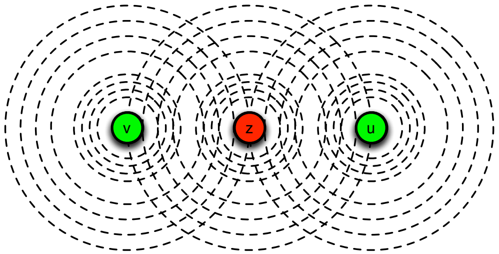
Dolev's Algorithm



Wireless Networks

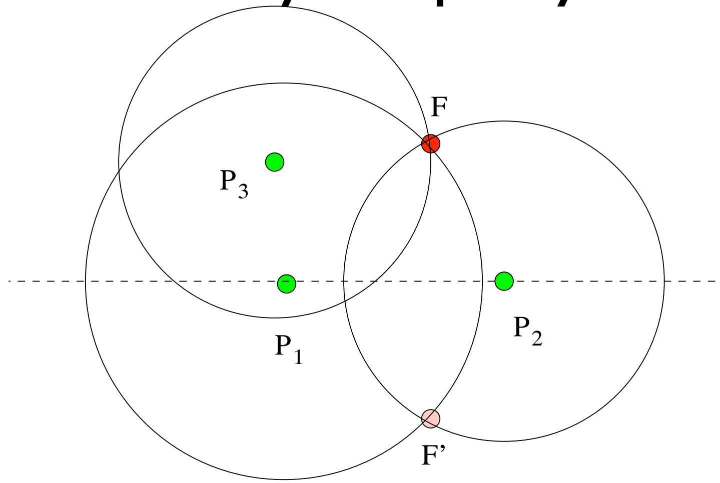
Traps and Pitfalls

- No way to assess sender

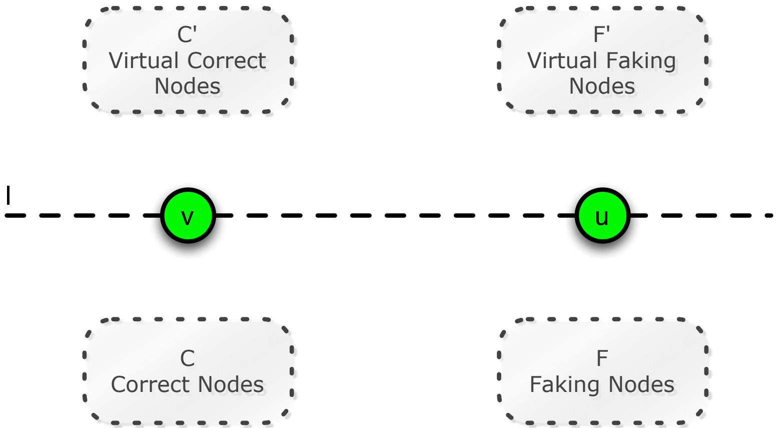


- Byzantine must lie consistently

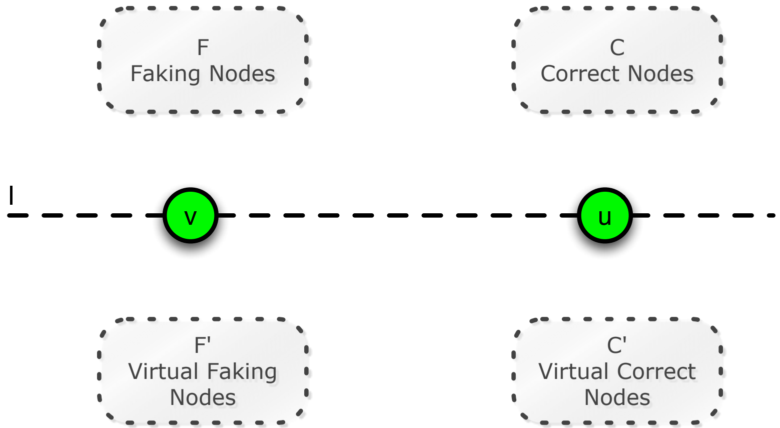
A Key Property



Lower bound



Lower bound



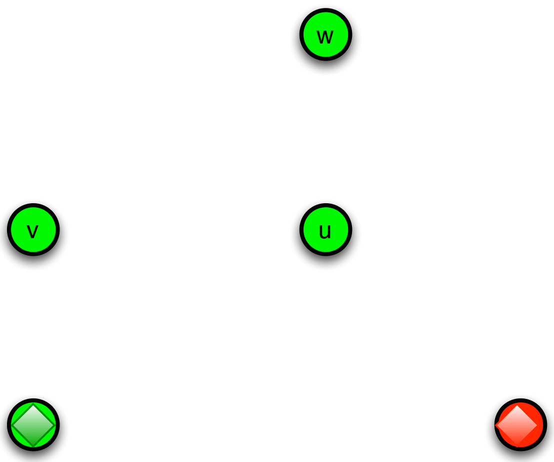
Assumptions

- No three nodes are colinear
- No more than *f* faking nodes, with *n-f-2 > f*
- Distance is impossible to fake
- Faking nodes send at most one message per round

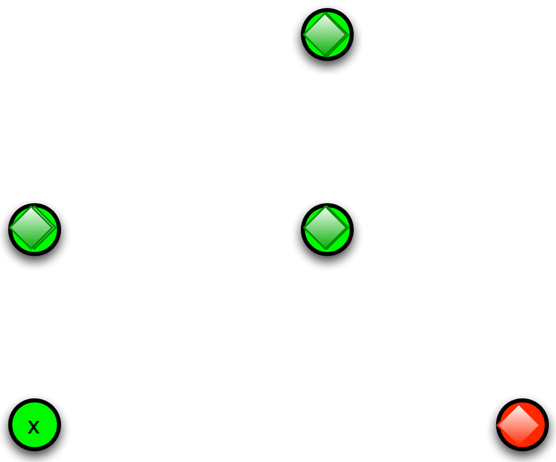
A Naive Protocol

- For every annoucement by a node v
  - *Report* **OK(v)** if perceived distance matches annouced distance, else *report* **KO(v)**
- *Count* **OK(v)**s and **KO(v)**s for every report
  - If #**KO(v)** > #**OK(v)** - 2, v is faulty

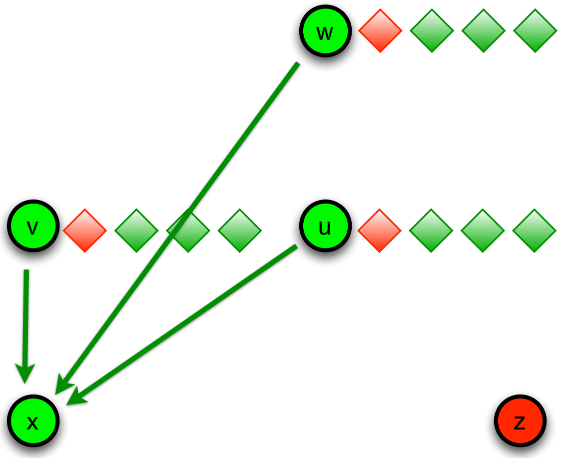
A Naive Protocol



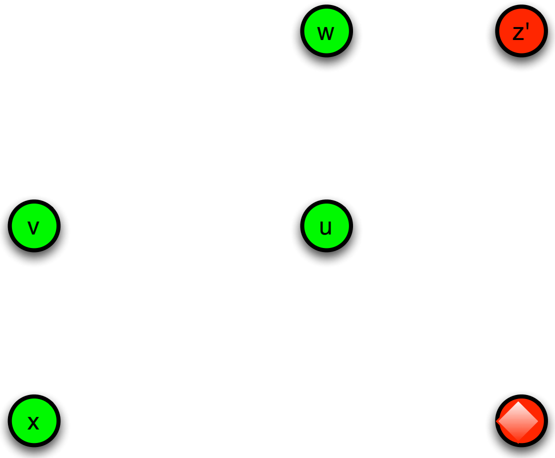
A Naive Protocol



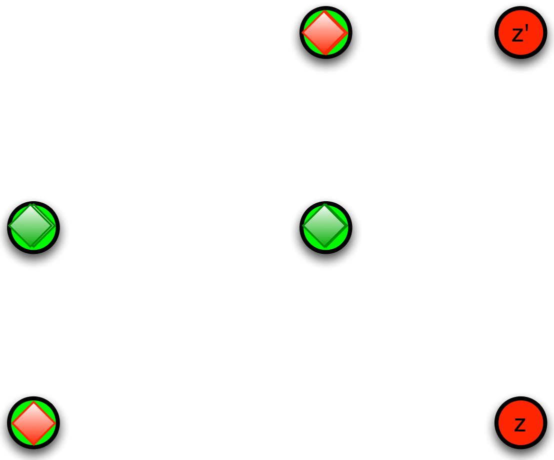
A Naive Protocol



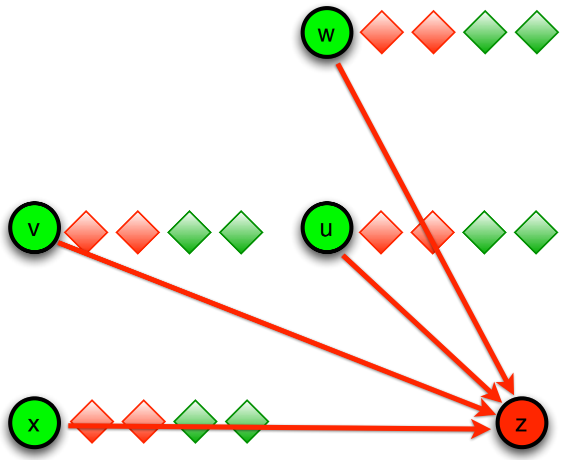
A Naive Protocol



A Naive Protocol



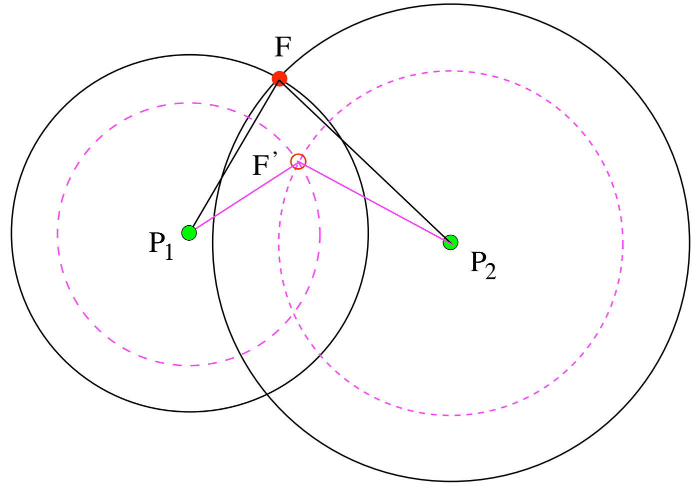
A Naive Protocol



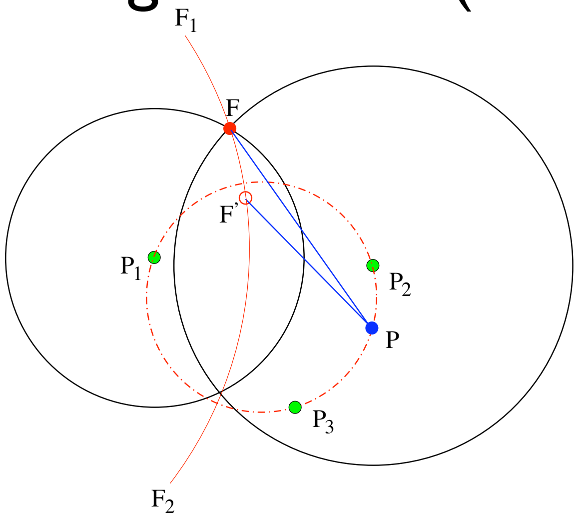
Faking the Distance

- **RSS** $S_r = S_s \left(\frac{\lambda}{4\pi d} \right)^2$
  - Change emiting signal strength
  - Must be consistent for *all* nodes
- **ToF & DAT**
  - Change processing speed or timestamps
  - Must be consistent for *all* nodes

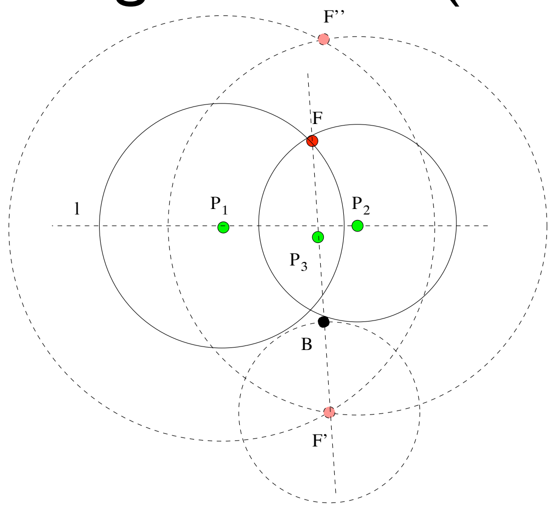
Faking Distance (RSS)



Faking Distance (RSS)



Faking Distance (ToF)



Faking Distance (ToF)