Chapter 2
The Two Generals Problem

Distributed Systems

SS 2015

Fabian Kuhn

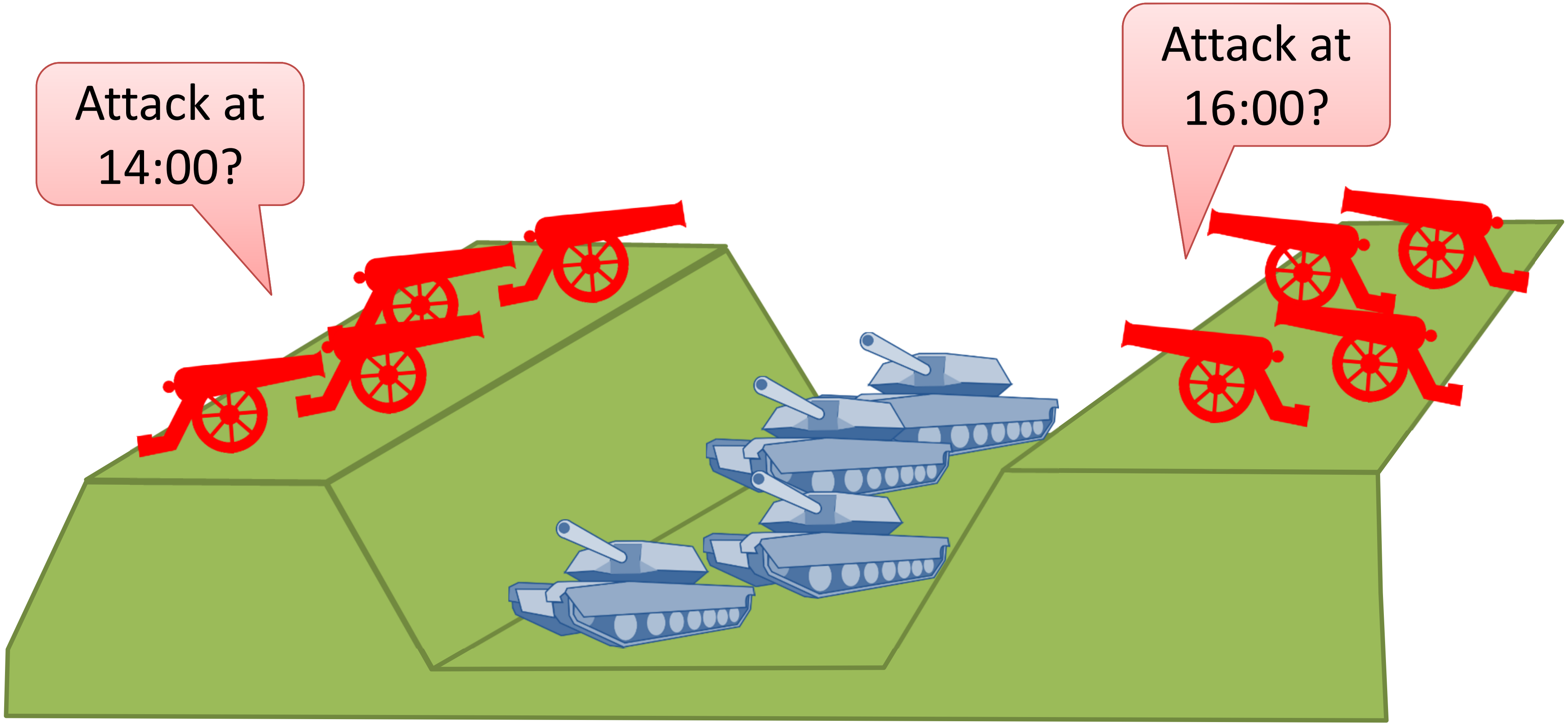
Agreement Problems

- In order to offer any non-trivial distributed service, the nodes / processes of a distributed system need to coordinate their actions.
- **Most basic coordination:** **agreeing** on some action / fact / ...

- We will study agreement problems in for various model assumptions.

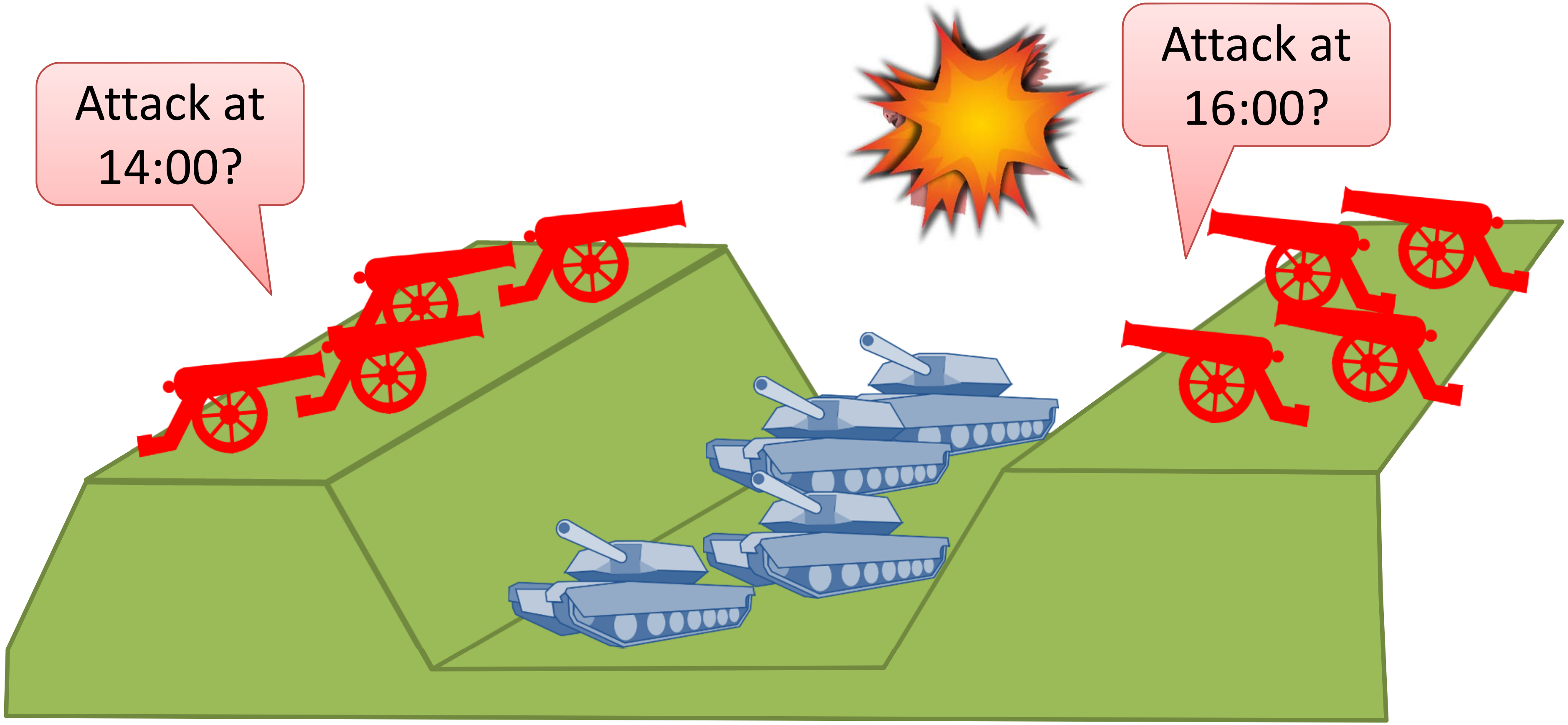
- To start, we consider a simple (but still interesting) one ...

The Two Generals Problem



- To win, the two red armies need attack together
- They need to agree on a time to attack the blue army

The Two Generals Problem



- Communication across the valley only by carrier pigeons
- Problem: pigeons might not make it

The Two Generals Problem

Problem is relevant in the real world...

- Alice and Bob plan to go out on Saturday evening
- They need to agree on:
  - when and where to meat
  - who makes the dinner reservation
  - ...
- They can only communicate by an unreliable messaging service

- Nodes in a network need to agree on
  - who's the leader for some computation
  - which of two / several conflicting data accesses to perform
  - whether to commit a distributed database transaction
  - ...

Two Generals More Formally

Model: two deterministic nodes, synchronous communication, unreliable messages (messages can be lost)

Input: node starts with one of two possible inputs 0 or 1

- say input encodes time to attack

Output: Each node needs to decide either 0 or 1

Agreement: Both nodes must output the same decision (0 or 1)

Validity: If both nodes have the same input $x \in \{0,1\}$ and no messages are lost, both nodes output x.

- If nodes start with different inputs or one or more messages are lost, nodes can output 0 or 1 as long as they agree.

Termination: Both nodes terminate in a bounded # of rounds.

Solving the Two Generals Problem?

msg. can be lost

when to terminate?

→ if all msg. are lost

→ if all msg. are delivered

Two Generals: Impossibility

$S|v = S'|v$

Indistinguishability Proof:

E has schedule S
E' " " S'

- Execution E is indistinguishable from execution E' for some node v if v sees the same things in both executions.
  - same inputs and messages (schedule)
- If E is indistinguishable from E' for v, then v does the same thing in both executions.
  - We abuse notation and denote this by $E|v = E'|v$

Similarity:

- Consider all possible executions $E_1, E_2, \ldots$
- Call E_i and E_j **similar** if $E_i|v = E_j|v$ for some node v

$$E_i \sim_v E_j \;\Leftrightarrow\; E_i|v = E_j|v$$

Two Generals: Impossibility

Consider a chain $E_0, E_1, E_2, \dots, E_k$ of executions such that for all $i \in \{1, \dots, k\}$, E_{i-1} and E_i are similar.

- $\forall i \in \{1, \dots, k\} : E_{i-1} \sim_v E_i$ for some node v

$$E_{i-1}|v = E_i|v$$

→ v does the same thing in E_{i-1} & E_i

→ v outputs the same decision

Agreement: all nodes output the same value in E_{i-1} & E_i

$$E_0 \sim E_1 \sim E_2 \dots\dots \sim E_k$$

Two Generals: Impossibility

v_1 —— v_2

Proof Idea:

- Assume there is a T-round protocol
  - Then, nodes can always decide after exactly T rounds
- Construct sequence of executions $E_0, E_1, \dots, E_k$ s.t.
  - For all $i \in \{1, \dots, k\}$ $E_{i-1} \sim_v E_i$ for some node $v \in \{v_1, v_2\}$
  - In E_0 output needs to be 0 and in E_k output needs to be 1

Execution $\boldsymbol{E_0}$: both inputs are 0, no messages are lost

Execution $\boldsymbol{E_k}$: both inputs are 1, no messages are lost

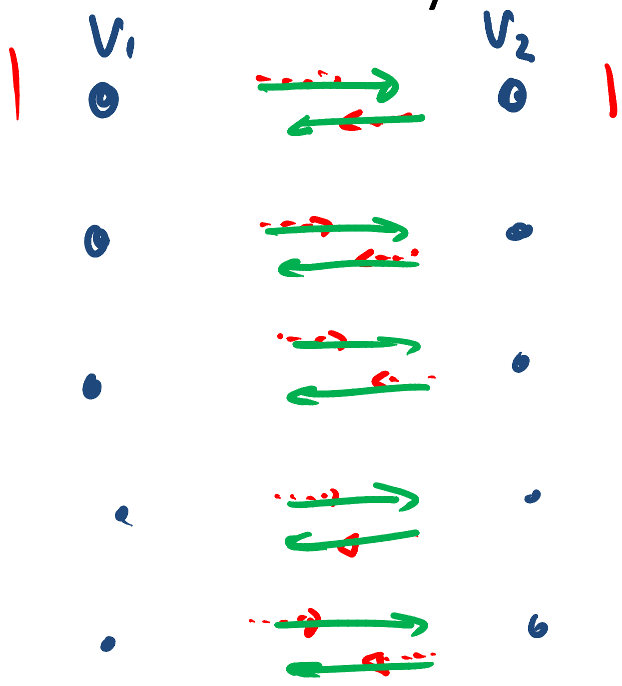
E_0: Validity → both output 0

E_k: " " " " 1

Two Generals: Impossibility

Nodes always decide after exactly T rounds

output: 0



→ contradiction

$E_0, \ldots, E_k$ → validity: both need to output 1

have to output 0

Two Generals: Impossibility

Nodes always decide after exactly T rounds

Execution $\boldsymbol{E_0}$: both inputs are 0, no messages are lost

Execution $\boldsymbol{E_1}$: one of the messages in round T is lost

Execution $\boldsymbol{E_i}$: last message M is delivered in round t

Execution $\boldsymbol{E_{i+1}}$: drop message M

Execution $\boldsymbol{E_{2T}}$: both inputs are 0, no messages are delivered

- All nodes output 0 (because of similarity chain)

Two Generals: Impossibility

Execution $\boldsymbol{E_{2T}}$: both inputs are 0, no messages are delivered

- All nodes output 0 (because of similarity chain)

Execution $\boldsymbol{E_{2T+1}}$: input of v_1 is 0, input of v_2 is 1, no msg. delivered

Execution $\boldsymbol{E_{2T+2}}$: input of both nodes are 1, no msg. delivered

Execution $\boldsymbol{E_{4T+2}}$: input of both nodes are 1 and no msg. are lost

- from E_{2T+2} to E_{4T+2} deliver messages one by one
- same chain as from E_0 to E_{2T}, but in opposite direction
- **In $\boldsymbol{E_{4T+2}}$, all nodes must output 1 $\Longrightarrow$ contradiction!**

Two Generals Impossibility: Summary

- We start with an execution in which both nodes have input 0 and no messages are lost $\Longrightarrow$ both nodes must decide 0.
- We prune messages one by one to get a sequence of executions s.t. consecutive executions are similar.
- From an execution with no messages delivered and both inputs 0, we can get to an execution with no messages delivered and both inputs 1 (in two steps).
- By adding back messages one-by-one, we get to an execution in which both nodes have input 1 and no messages are lost $\Longrightarrow$ both nodes must decide 1 $\Longrightarrow$ contradiction!

- Not hard to generalize to an arbitrary number $n \geq 2$ of nodes
- Upper bound on number of rounds not necessary
  - as long as nodes need to decide in finite time

Two Generals: Randomized Algorithm

- The two generals problem can be solved if
  - we allow (one of) the two generals to flip coins
  - we are satisfied if agreement is only achieved with probability $1 - \varepsilon$ (for ε small enough)

- But first, we look at a simple algorithm:

The Level Algorithm (Overview):

- Both nodes compute a level
- At the end, the two levels differ by at most one
- The levels essentially measure the number of successful back and forth transmissions

The Level Algorithm

$u \bullet \qquad \bullet\, v$

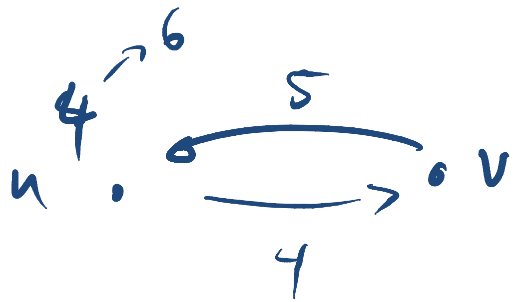
1. Both levels are initialized to 0

2. **In each round:**

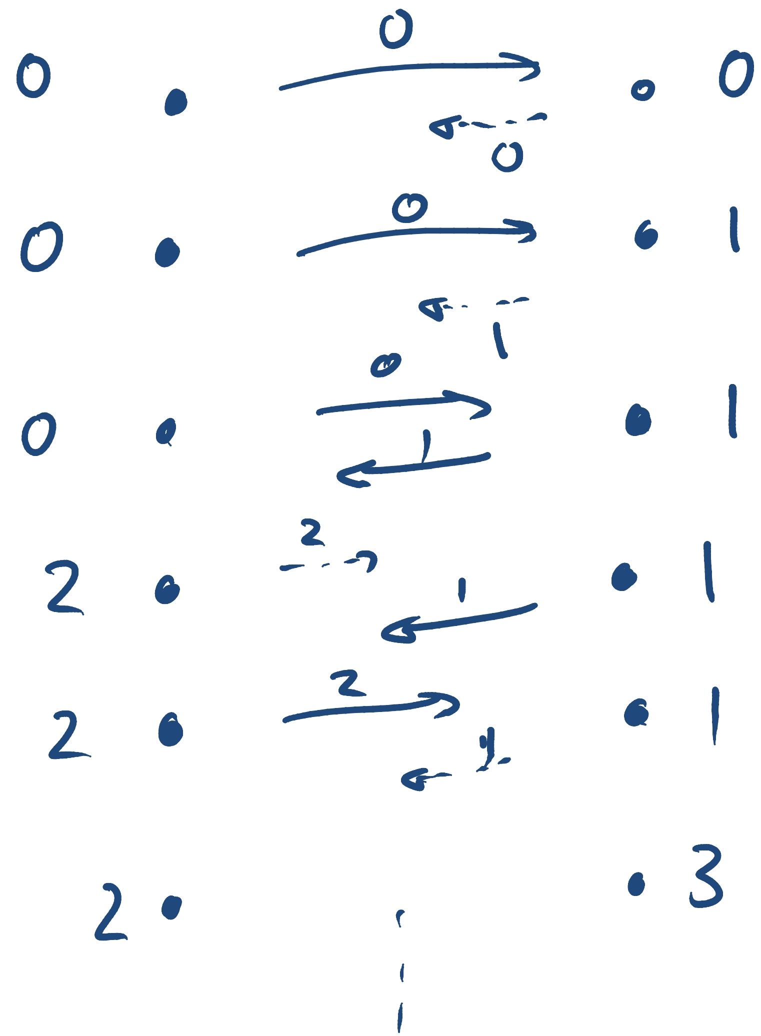
   Both nodes send their current level to each other

3. Assume node u with level ℓ_u receives level ℓ_v from v

   u updates its level to $\boldsymbol{\ell_u} := \mathbf{max}\{\underline{\boldsymbol{\ell_u}}, \underline{\underline{\boldsymbol{\ell_v} + \mathbf{1}}}\}$



The Level Algorithm: Example



The Level Algorithm: Properties

Lemma: At all times, the two levels differ by at most one.

$\underline{\ell_u \le \ell_v + 1}$

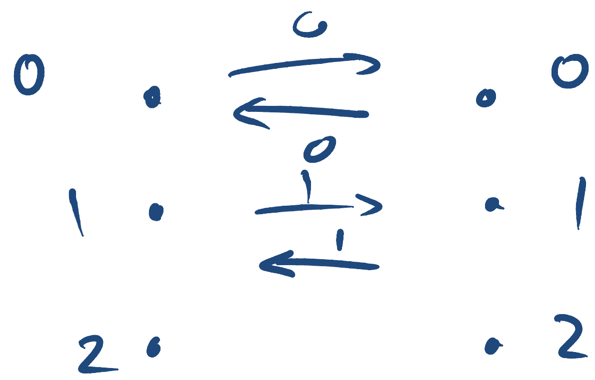
ℓ_u u ℓ_v v ℓ_v

ℓ_u' ℓ_v'

$$\underline{\ell_u'} \le \max\{\ell_u, \ell_v + 1\} = \ell_v + 1 \le \underline{\ell_v' + 1}$$

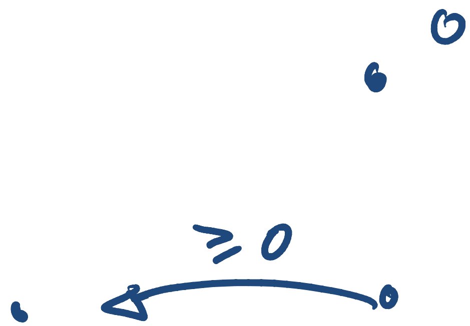
The Level Algorithm: Properties

Lemma: If all messages are delivered, the two levels are equal to the number of rounds.



The Level Algorithm: Properties

Lemma: The level ℓ_u of a node u is 0 if and only if all of the messages to u have been dropped.



The Level Algorithm: Summary

The Level Algorithm (between 2 nodes):

If the algorithm is run for r rounds:

1. At the end, the two levels differ by at most one
2. If all messages succeed, both levels are equal to r
3. The level ℓ_u of a node u is ≥ 1 if and only if u successfully received at least one message

The Randomized Two Generals Algorithm

- Assume that the two nodes are called u and v and that u is the leader node (e.g., the one with lower ID).
- Also, assume that the possible inputs are 0 and 1

param,

1. Node u picks are (uniform) random number $R \in \{1, \dots, r\}$
2. The nodes run the level algorithm for r rounds
   – In each message, both nodes also include their inputs and node u also includes the value of R
3. At the end, a node decides 1 if and only if:
   – Both inputs are equal to 1
   – The node knows R and it has seen both inputs
   – The level of the node is $\geq R$

   $\ell_u \geq R$
4. Otherwise, the node decides 0

The Randomized Two Generals Algorithm

Lemma: If at least one input is 0, both nodes output 0

to output 1, need to see both inputs, and both are 1

Lemma: If both inputs are 1, then both nodes

a) output 1 if no message is lost

b) output the same value unless $\{\ell_u, \ell_v\} = \{R - 1, R\}$

a) $\ell_u = \ell_v = r$

$\ell_u, \ell_v \geq R \in \{1, \ldots, r\}$

$\ell_u \geq R$ and $\ell_v \geq R$ $\quad R \geq 1$

both output 1

$\ell_u < R$ and $\ell_v < R$

both output 0

The Randomized Two Generals Algorithm

Theorem: If at least one of the inputs is 0, both nodes output 0. If both inputs are 1, if no message is lost, both nodes output 1, otherwise both nodes output the same value with probability at least $1 - {}^{1}/_{r}$.

Lemma: output same value unless $\{\ell_u, \ell_v\} = \{R-1, R\}$

- levels only depend on which msg. are lost

$\{0,1\}$, $\{1,2\}$, $\{2,3\}$, ..., $\{r-1, r\}$

$\uparrow$

$R=3$

prob. to pick the bad R is $\frac{1}{r}$

Lower Bound on Error Probability

Using similar techniques as for the impossibility of the deterministic two problem, we can prove a lower bound on the error probability.

Stronger version of the problem (stronger validity condition):

- If at least one input is 0, both nodes need to output 0
  - our randomized algorithm satisfies this

To prove the lower bound, we assume that if both inputs are 1,

- if no messages are lost, both outputs must be 1,
- otherwise, the nodes need to output the same value with probability at least $1 - \varepsilon$ (probabilistic agreement).

Lower Bound on Error Probability

$q_u \leq r \cdot \varepsilon$

$q_u = 1$

$\boxed{\varepsilon \geq \frac{1}{r}}$

u | v

1 4ε • | • 4ε 0

3ε • | • 3ε

$q_u \leq 2\varepsilon$ • | • $q_v \leq 2\varepsilon$

$q_u \leq \varepsilon$ • | • $q_v = 0$ $\quad q_v \leq \varepsilon$

r rounds

no msg. del.

$q_u, q_v \leq r \cdot \varepsilon$

For both nodes:

$Pr(\text{output} = 1) = 1$

$q := Pr(\text{output} = 0) = 0$

Lower Bound on Error Probability

Theorem: In the strong version of the two generals problem, if nodes need to decide within r rounds, the probability ε for not agreeing on the same value (if both inputs are 1) is at least

$$\varepsilon \geq \frac{1}{r}.$$

Remark: For the original version of the problem, a similar proof givers a lower bound of

$$\varepsilon \geq \frac{1}{2r+1}.$$