Carnegie Mellon
SCHOOL OF COMPUTER SCIENCE

FACULDADE
DE CIÊNCIAS
UNIVERSIDADE DE LISBOA

ROBUST & SPECULATIVE BYZANTINE RANDOMIZED CONSENSUS

WITH CONSTANT TIME COMPLEXITY IN NORMAL CONDITIONS

Bruno Vavala
University of Lisbon, Portugal
Carnegie Mellon University, U.S.

Nuno Neves
University of Lisbon, Portugal

CONSENSUS

- Fundamental problem in distributed computing
  - Examples: SM Replication, Leader Election, Coordination, Group Membership, etc.
- Impossible to attain deterministically with crash-faults (partial correctness)
- Termination achievable with:
  - weaker models (ev. synchrony assumption)
  - randomization (almost-surely)

RANDOMIZED CONSENSUS

- Properties
  - Validity: *if all correct processes propose v, then v is the only possible decision*
  - Agreement: *no two correct processes decide differently*
  - Probabilistic Termination: *all correct processes eventually decide with probability **1***
- Assumptions
  - Reliable channels
  - Source-authenticated channels

BRACHA'S ALGORITHM

(PODC 1984)

- Seminal algorithm
- Asynchronous
- Byzantine resistant
- Resilient-optimal (3f+1)
- Correct under the Strong Adversary model

BRACHA'S ALGORITHM

(PODC 1984)

- Seminal algorithm
- Asynchronous
- Byzantine resistant
- Resilient-optimal (3f+1)
- Correct under the Strong Adversary model

1) RBcast value
2) Set majority value

1st phase
(set majority)

BRACHA'S ALGORITHM

(PODC 1984)

- Seminal algorithm
- Asynchronous
- Byzantine resistant
- Resilient-optimal (3f+1)
- Correct under the Strong Adversary model

1) RBcast value
2) Set majority value

1st phase
(set majority)

4) RBcast value
5) Set quorum value
(if any, or default v)

2nd phase
(try-lock)

BRACHA'S ALGORITHM

(PODC 1984)

- Seminal algorithm
- Asynchronous
- Byzantine resistant
- Resilient-optimal (3f+1)
- Correct under the Strong Adversary model

1)RBcast value
2)Set majority value

1st phase
(set majority)

4)RBcast value
5)Set quorum value
(if any, or default v)

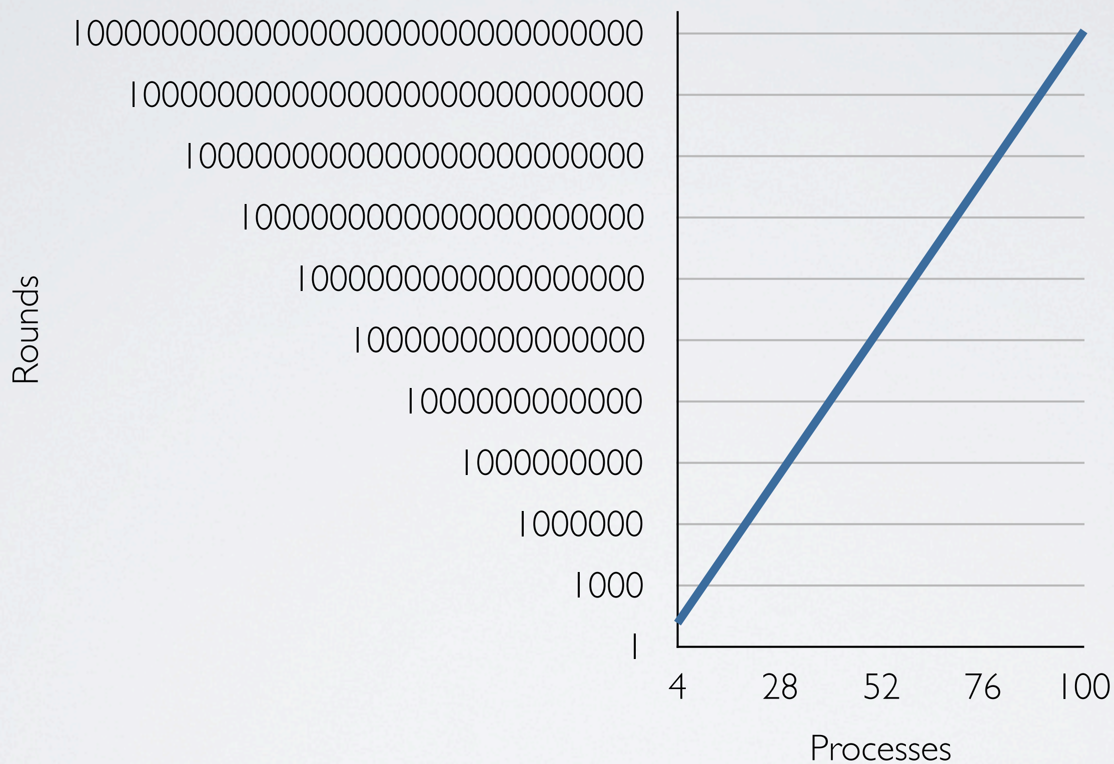
2nd phase
(try-lock)

7)RBcast value
8)Set decision value
(if any, or majority)
(if any, or *flip coin*)

3rd phase
(try-decide)

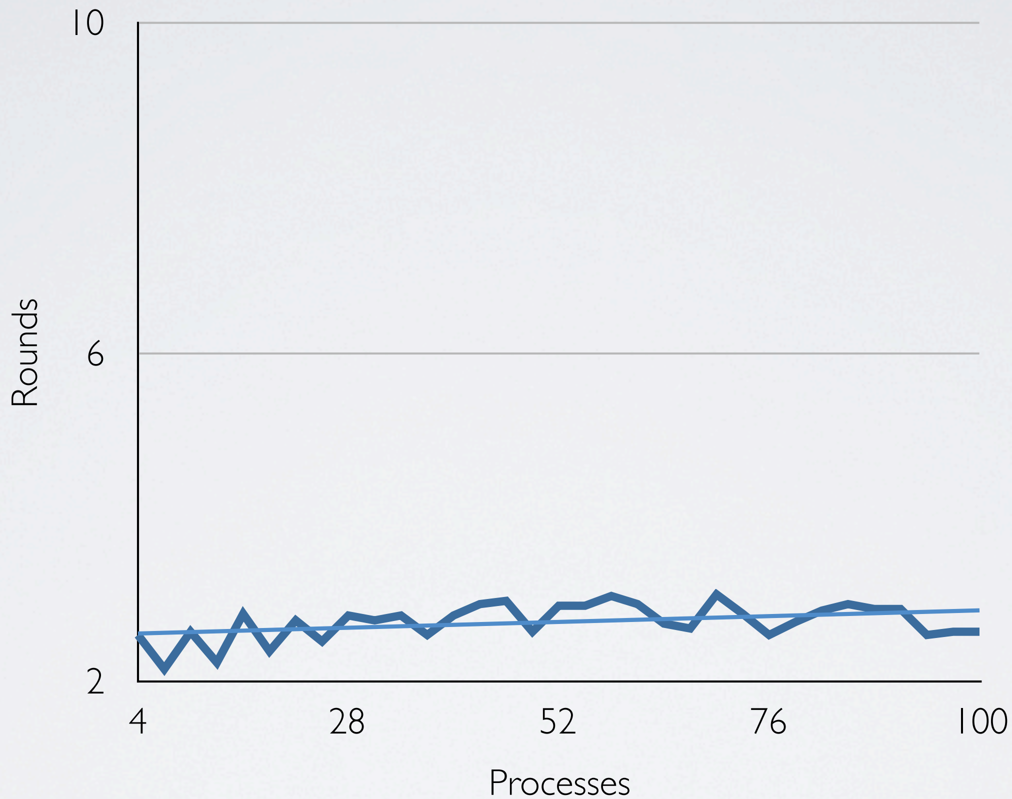
10)start new round

IN THEORY



Potential problem: expected exponential time execution under adverse conditions

IN PRACTICE



In reality: it terminates in a constant number of rounds under normal conditions

RELIABILITY VS. PERFORMANCE
WHAT'S THE MODEL?

WHAT IS NORMAL?

- Asynchrony? ✓
- Crash failures? ✓
- Byzantine failures? ✗
- Content-independent message scheduler? ✓
- Full information adversary? ✗
- Adversary message scheduler? ✗

AN EXPERIMENT

FIRST ROUND

first phase >

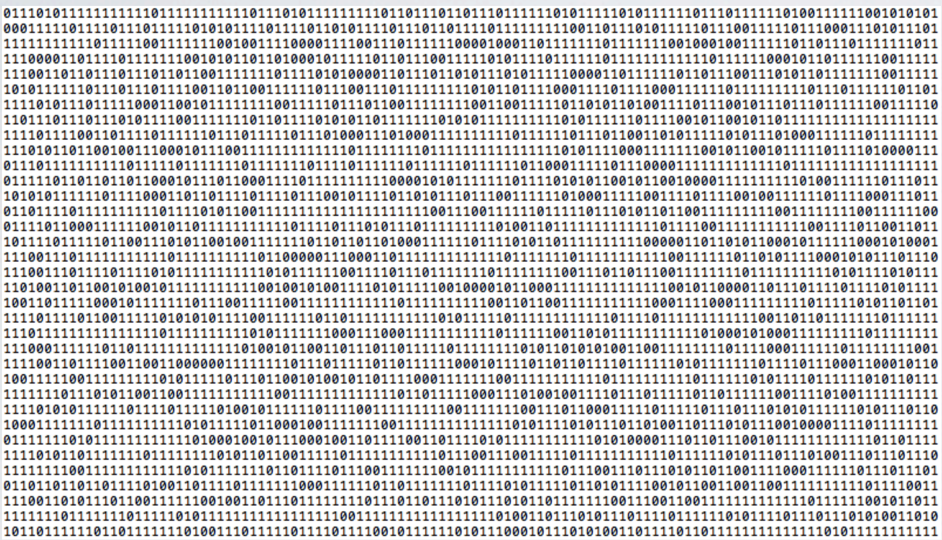
< second phase

third phase >

toss a coin

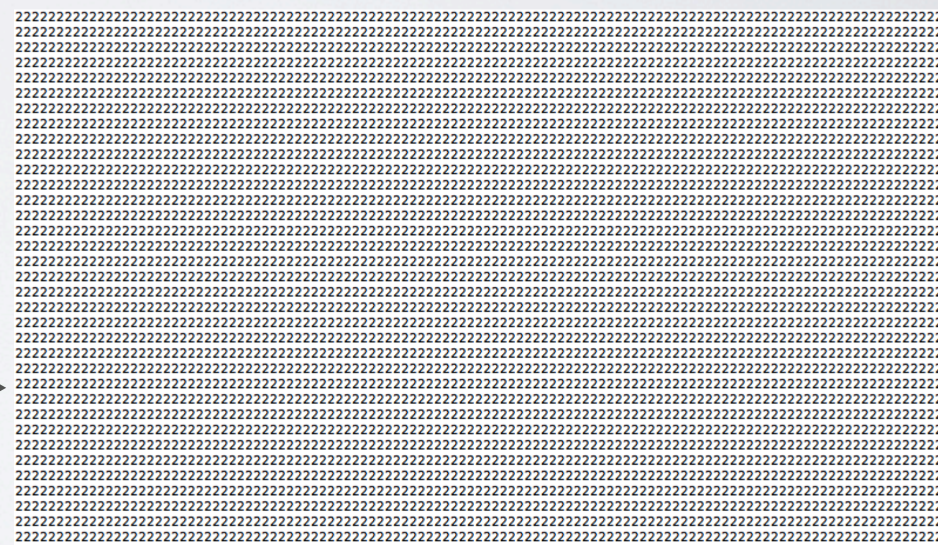
SECOND ROUND

first phase >



< second phase

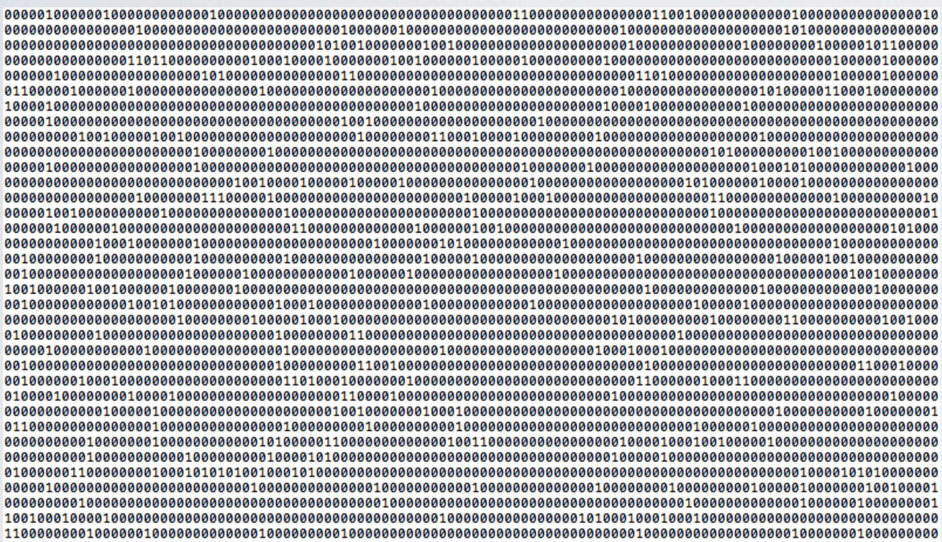
third phase >



toss a coin 

THIRD ROUND

first phase >



< second phase

third phase >

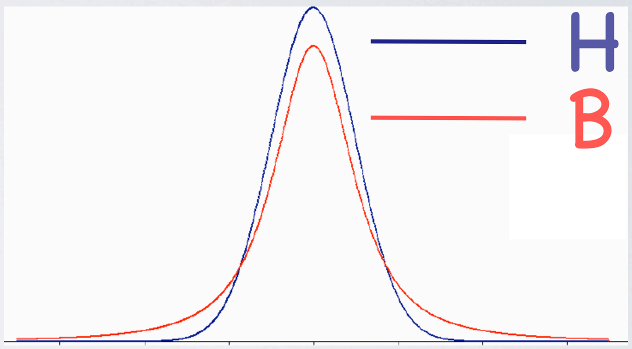
decision

PROBABILISTIC ANALYSIS

INGREDIENTS

- Hypergeometric distribution

$$\mathcal{H}(n,\ k,\ n{-}f)$$



- Binomial distribution

$$\mathcal{B}(n,p)$$

- Normal distribution

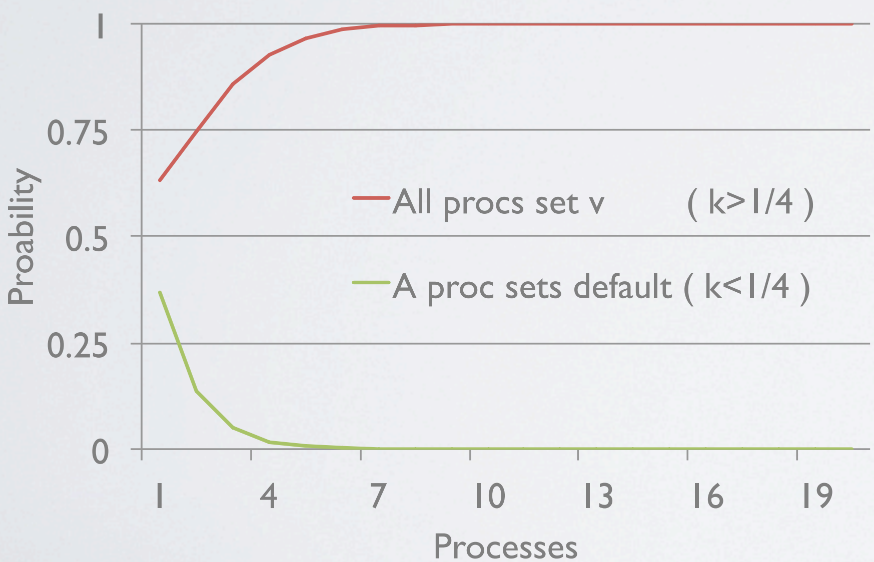
$$\mathcal{N}(np, np(1-p))$$

- Some approximations

$$P\left(\mathcal{B}(n,p) \leq i\right) \approx \Phi\left(\frac{i-np}{\sqrt{np(1-p)}}\right)$$

KEY

- Threshold of half plus `1/4` of procs proposing v at the end of 2nd phase



GOING BACKWARDS

- decision on v

message exchange
3rd Phase

- Linear bias of constant **`1/4`** of procs proposing v

message exchange
2nd Phase

- Procs have constant probability of setting v
- Linear bias of (just) **`positive`** constant beyond the average

message exchange
1st Phase

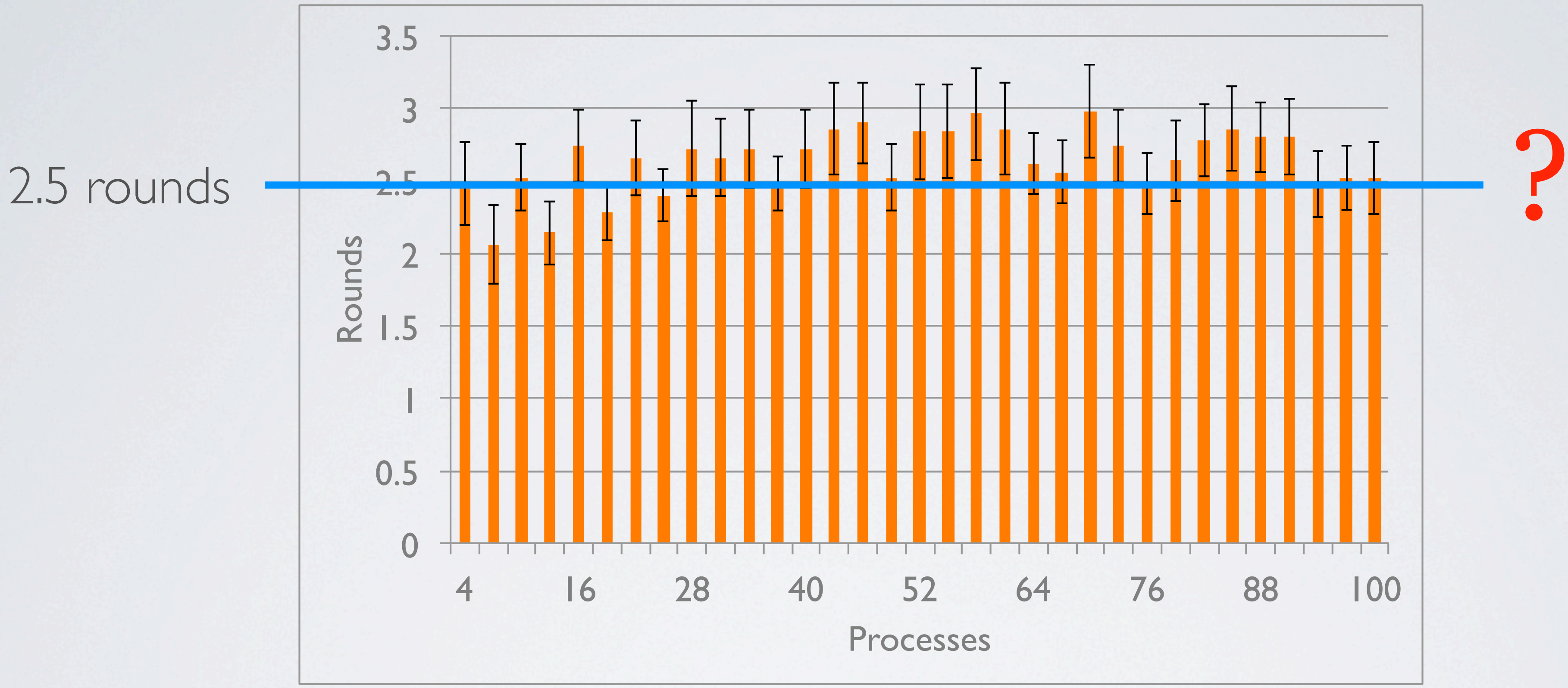
- Square root bias beyond the average of procs proposing v

- Back to coin tossing, this is a ...

Basic property of the Normal Distribution: **`p=2/5`** (or **`2.5 rounds`**)

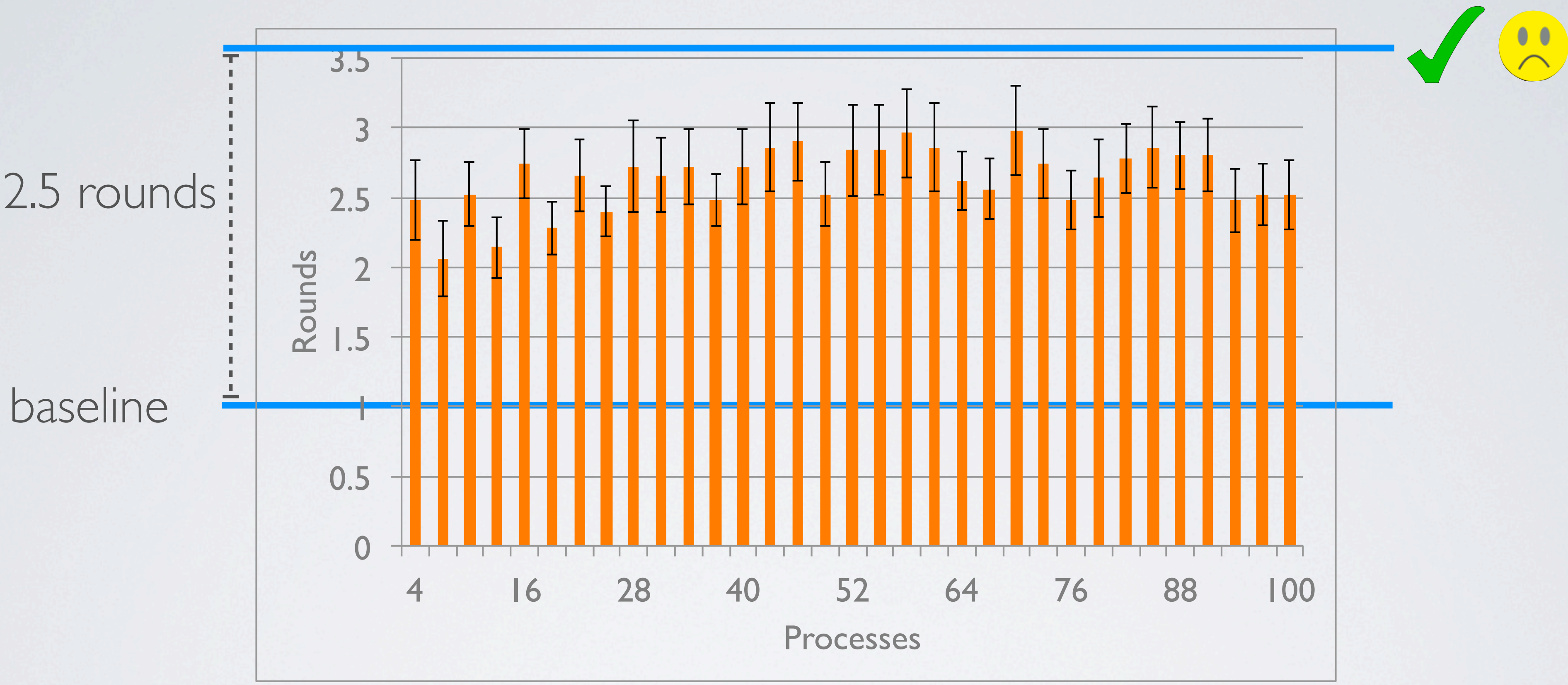
EVALUATION

PERFORMANCE



- Cluster of 6 nodes
- Up to 100 processes
- n = 3f + 1
- Divergent initial configuration

PERFORMANCE



- Analysis says 2.5 rounds after coin flipping
- Baseline at 1 round
- Theoretically satisfactory, but practically not precise, constant complexity

LOOK AT THE CONSTANTS

- Approximations are theoretically good
- Loss of precision when computing constant values

$$P(\mathcal{H}(n,\ k,\ n-f) \leq i) \overset{\text{ours}}{\leq} \Phi\left(\frac{i-\mu_{\mathcal{B}}}{\sigma_{\mathcal{B}}}\right)$$

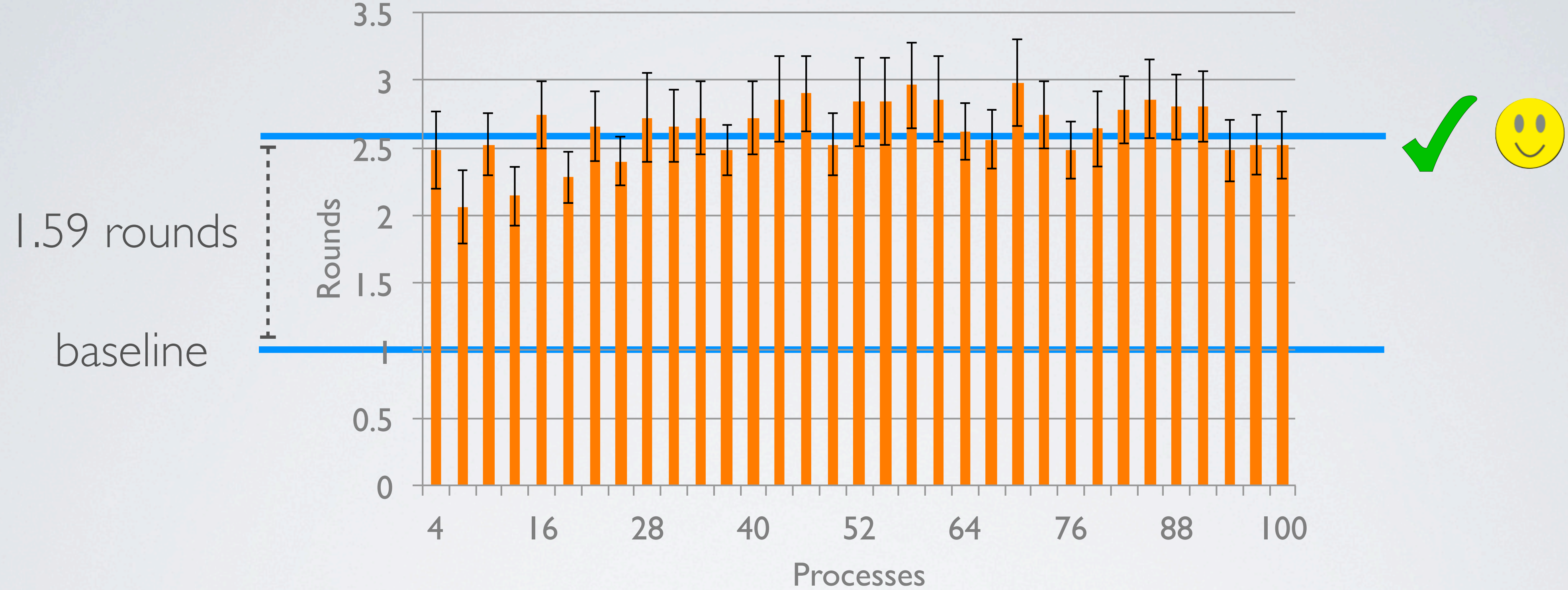
- A better approximation is available

$$P(\mathcal{H}(n,\ k,\ n-f) \leq i) \overset{\text{Feller}}{\approx} \Phi\left(\frac{i-\mu_{\mathcal{H}}}{\sigma_{\mathcal{H}}}\right)$$

- A multiplicative constant impacts noticeably just on constants

$$\Phi\left(\frac{i-\mu_{\mathcal{H}}}{\sigma_{\mathcal{H}}}\right) = \Phi\left(\frac{i-\mu_{\mathcal{B}}}{\sigma_{\mathcal{B}}}\sqrt{3}\right)$$

PERFORMANCE

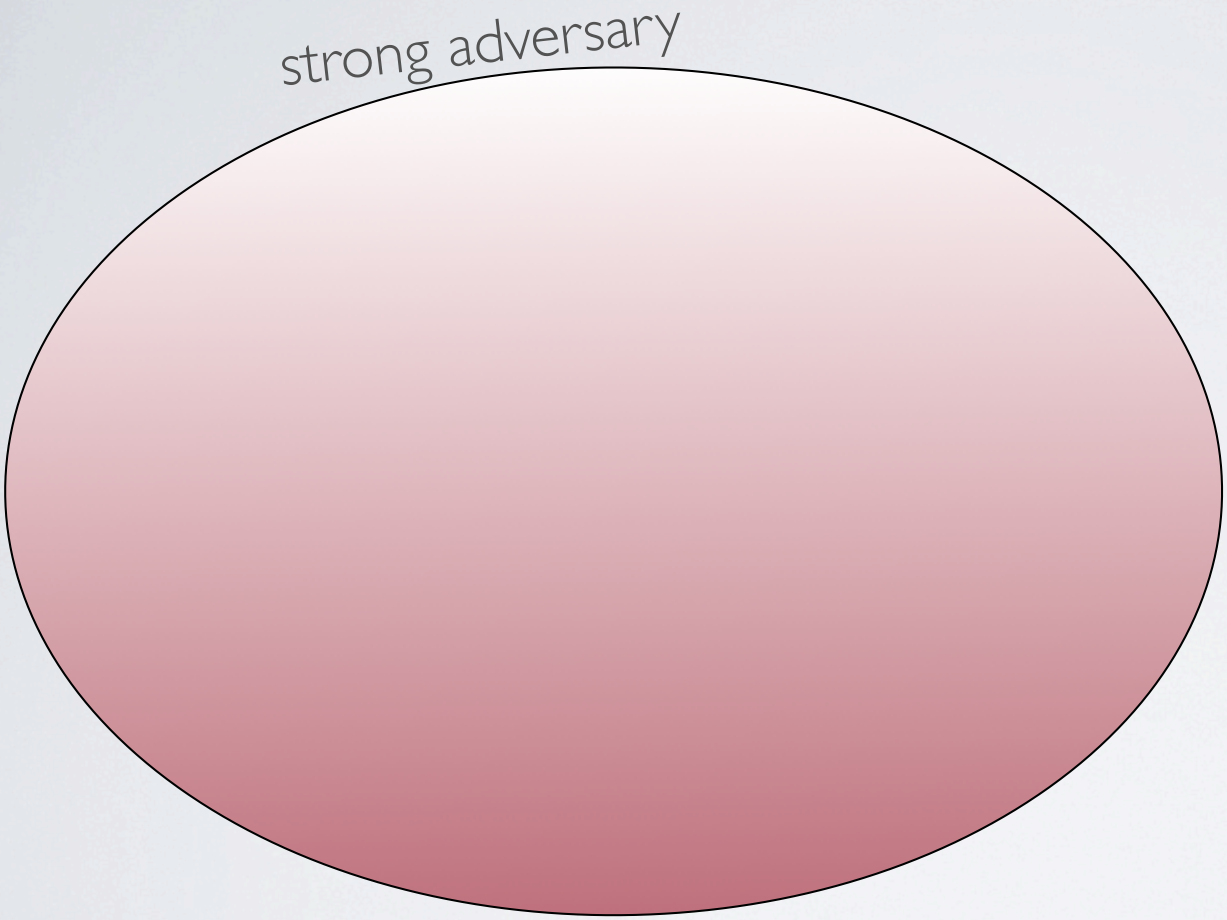


- Analysis says _1.59_ rounds after coin flipping
- Baseline at 1 round
- Theoretically satisfactory *and practically rather precise* constant complexity

HIGH LEVEL VIEW

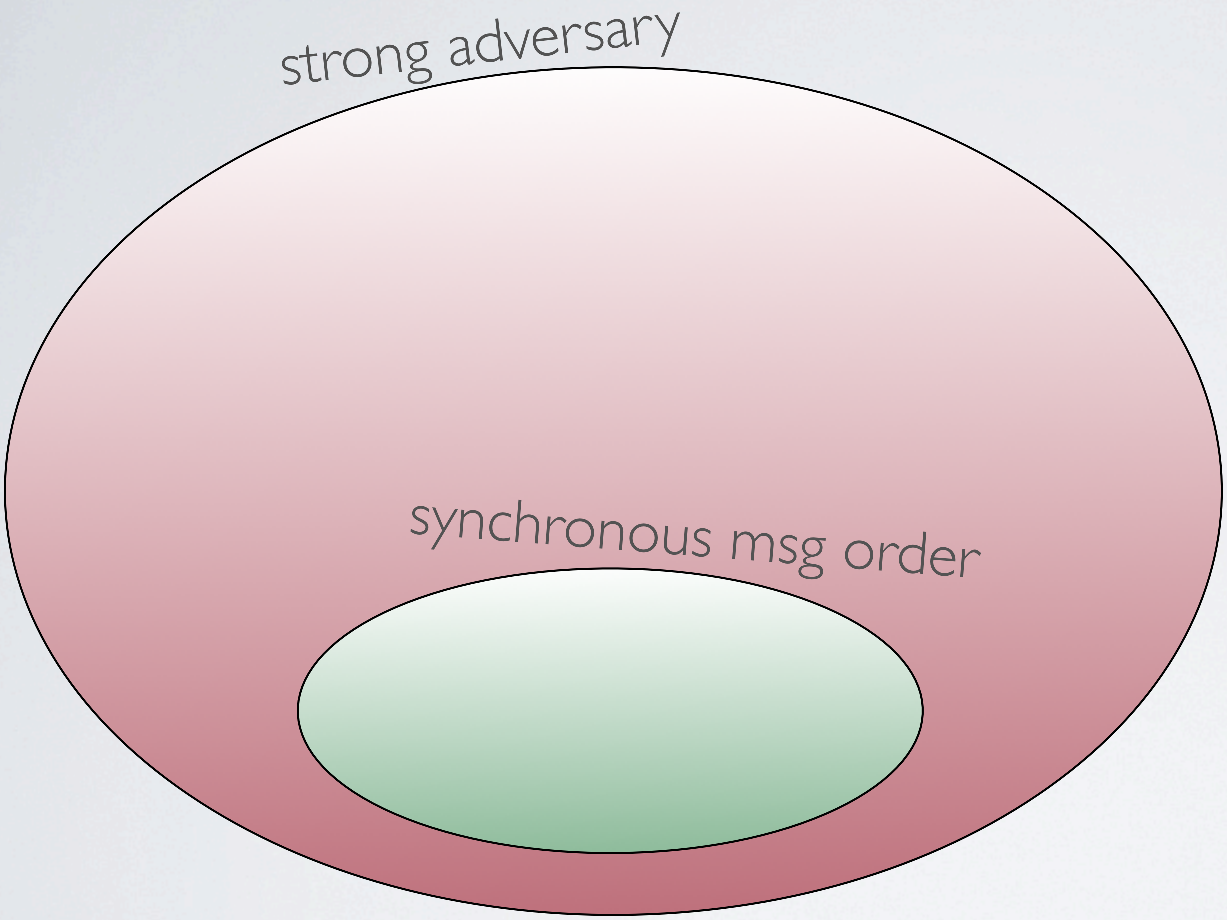
Model	Complexity
SA	$O(2^n)$
WA	O(1)
SMO	1 round

HIGH LEVEL VIEW



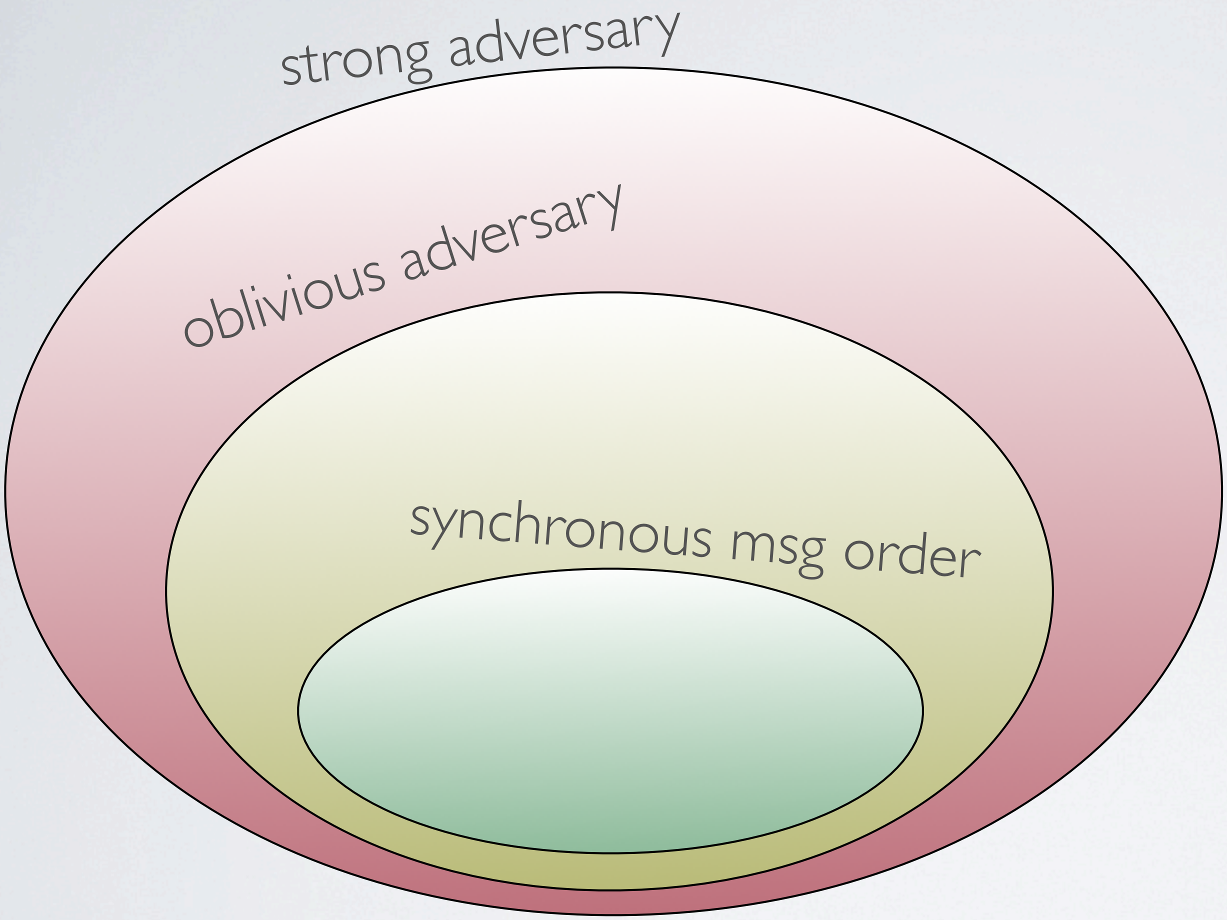
Model	Complexity
SA	$O(2^n)$
WA	$O(1)$
SMO	1 round

HIGH LEVEL VIEW



Model	Complexity
SA	`O(2^n)`
WA	`O(1)`
SMO	`1 round`

HIGH LEVEL VIEW



Model	Complexity
SA	$O(2^n)$
OA	$O(1)$
SMO	1 round

Complexity values are all relative to the Bracha's algorithm

LET'S GO BEYOND

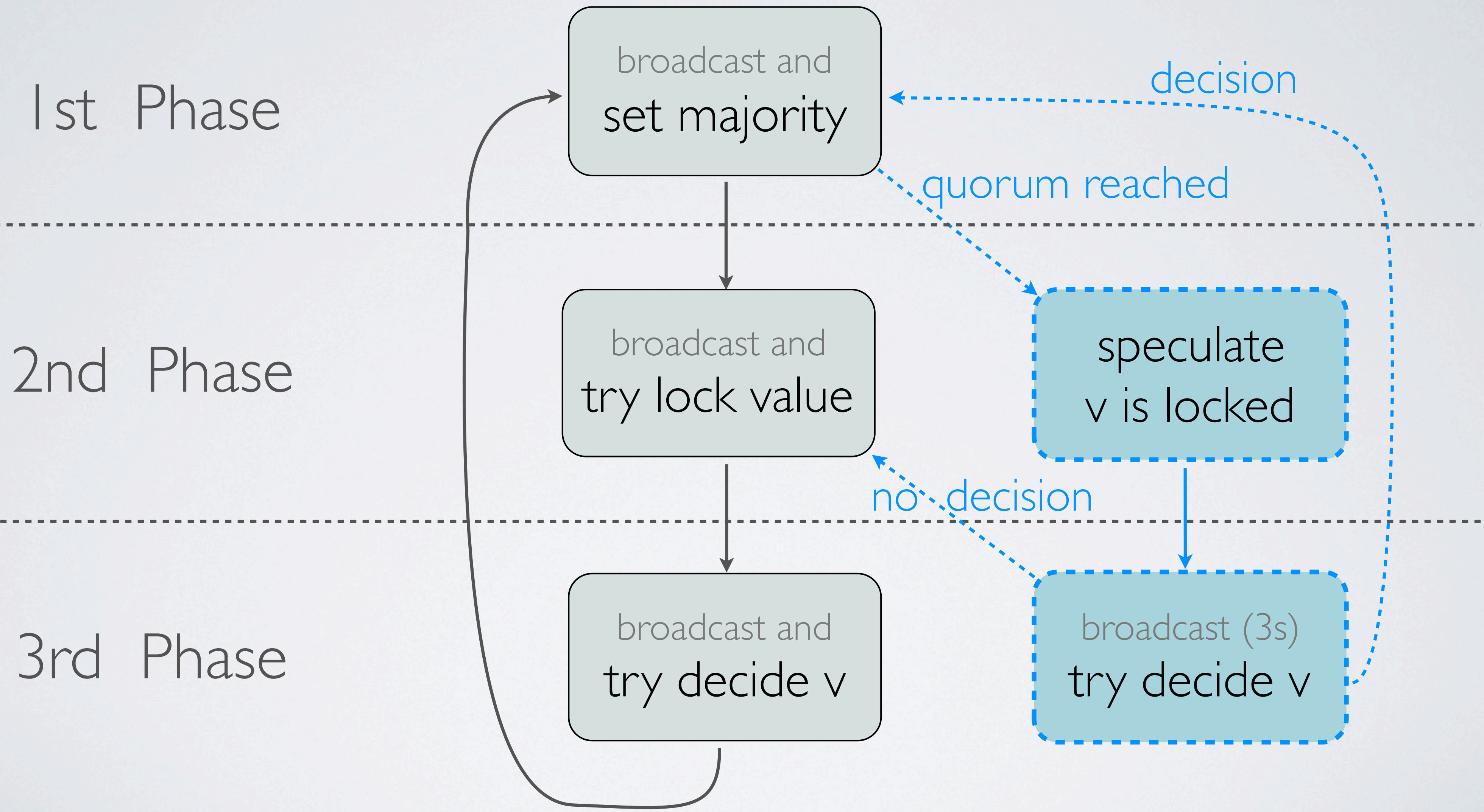
OVERVIEW

Termination in { 1-2 rounds ✓ (good)
3-6 phases ✗ (bad) }

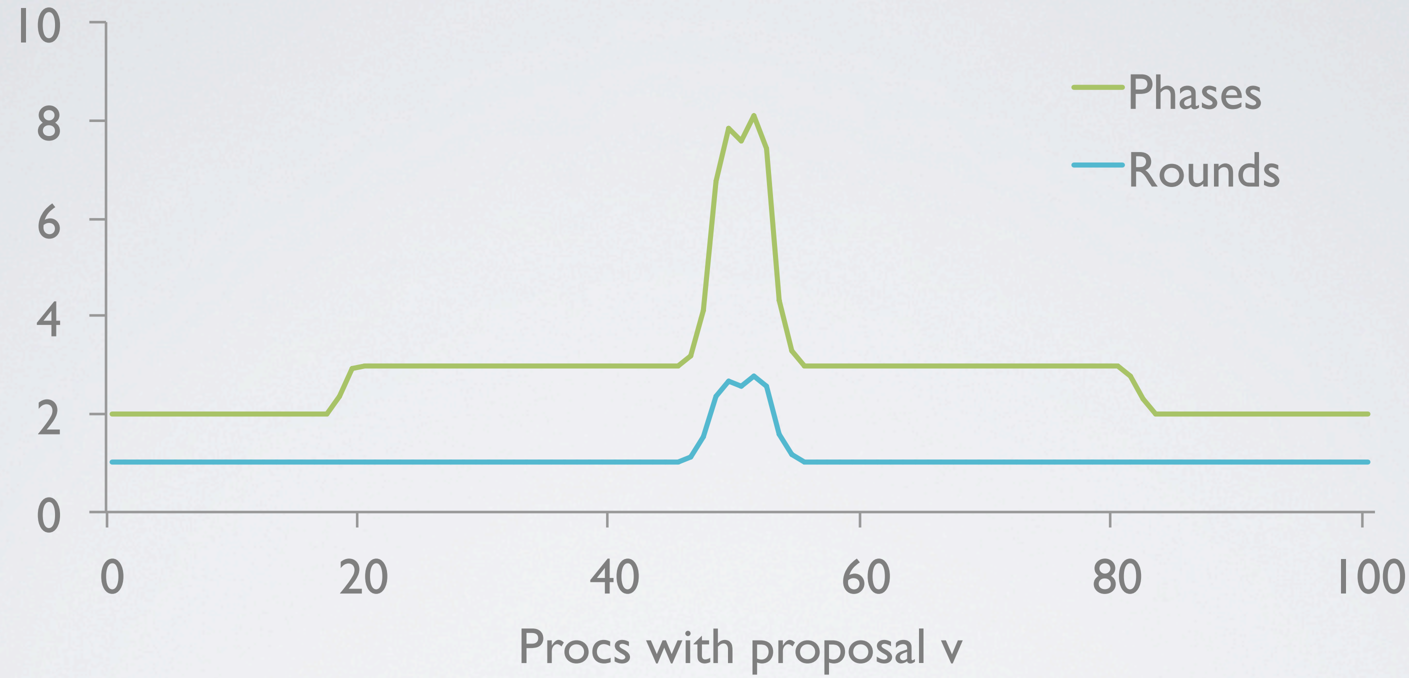
Objective: can we improve phase complexity in normal conditions while maintaining reliability?

- (oblivious) crash-failures may happen
  - Decision in 1 phase possible in a weaker model
- Focus on the set of (n-f) received messages

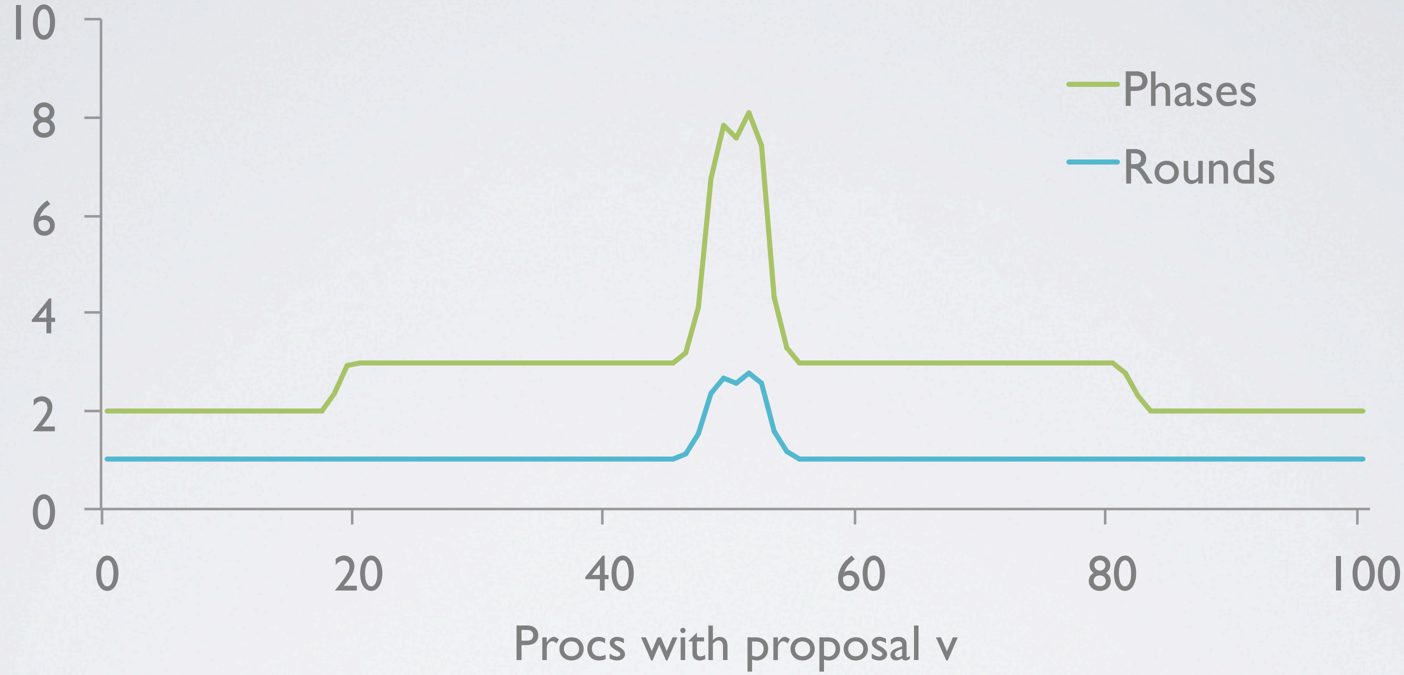
SPECULATION



PERFORMANCE



PERFORMANCE



- 2-phase termination more frequent with more msgs

BENEFITS AND DRAWBACKS

PROs	CONs
2 phases/round in the best case	Algorithm complexity increased due to speculation
3 phases/round if speculation fails	Fragile for near divergent proposals
Does not compromise original algorithm's properties	

SUMMARIZING

- Bracha's algorithm (PODC 1984) terminates in constant time (1.59 expected rounds) in normal conditions
    - First cross-model (non-trivial) analysis
    - Enhanced detection of anomalous/malicious behavior
- (Almost) matching upper-bound with respect to Attiya-Censor's lower bound (PODC 2008)
- Improved algorithm through inexpensive Speculation

THANK YOU!