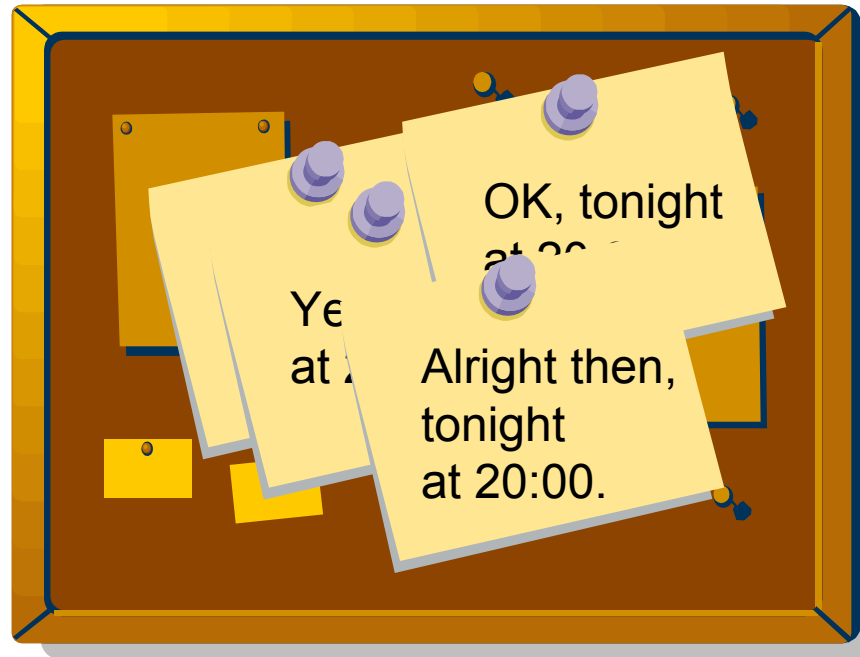

Tight Bounds for Asynchronous Randomized Consensus

Hagit Attiya & Keren Censor
Department of Computer Science
Technion

Consensus



Alice and Bob want to meet for dinner.

They write and read notes on the bulletin board.

Deterministic asynchronous consensus is impossible
[Fischer, Lynch & Paterson 1985]

Consensus

Each process p_i has an input value $x_i \in \{0, 1\}$
and should produce an output value $y_i \in \{0, 1\}$

- **Agreement:** all the outputs are equal
 - for every y_i, y_j that are assigned, $y_i = y_j$
- **Validity:** the output is the input of some process
 - for every y_i that is assigned, $y_i = x_j$ for some j
- **Termination:**
every nonfaulty process eventually decides
 - every nonfaulty process p_i eventually assigns y_i

Randomized Consensus

Each process p_i has an input value $x_i \in \{0, 1\}$ and should produce an output value $y_i \in \{0, 1\}$

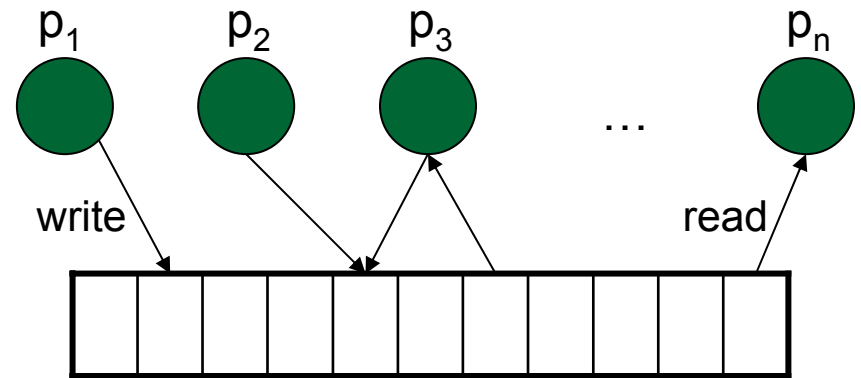
- **Agreement**: all the outputs are equal
 - for every y_i, y_j that are assigned, $y_i = y_j$
- **Validity**: the output is the input of some process
 - for every y_i that is assigned, $y_i = x_j$ for some j
- **Termination**: **with probability 1**, every nonfaulty process eventually decides
 - every nonfaulty process p_i eventually assigns y_i

The **total step complexity** is the **expected** total number of steps taken by all the processes

Shared Memory Multiprocessor

n asynchronous processes

Multi-writer multi-reader
shared registers



- At most **f crash failures** (at some point the process stops taking steps)
 - ⇒ Cannot tell whether a process has failed or is just slow
- The ordering of steps is determined by a **strong adversary**
 - Makes the scheduling decision after observing the local coin-flips

Bounds on the Total Step Complexity

- $O(\exp(n))$ [Abrahamson 1988]
- $O(n^4)$ [Aspnes & Herlihy 1990, Aspnes 1993]
- $O(n^3)$ [Saks, Shavit & Woll 1991]
- $O(n^2 \log n)$ [Bracha & Rachman 1991]
- **$O(n^2)$**
- **$\Theta(n^2)$**
- **$\Omega(n^2)$**
- $\Omega(n^2 / \log^2 n)$ [Aspnes 1998]

Upper Bound: Shared Coin

Shared coin with **agreement parameter δ** : for $v \in \{0, 1\}$, the probability that all processes return v is at least **δ**

⇒ Randomized consensus with step complexity **$\delta^{-1}T$** where **T** is the step complexity of the shared coin

[Aspnes & Herlihy 1990]

Naïve shared coin: every process flips its own coin, agreement parameter is $1/2^n$

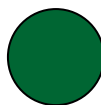
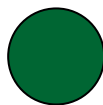
[Abrahamson 1988]

☞ We present a shared coin with constant agreement parameter and **$O(n^2)$** total step complexity

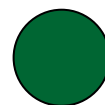
Shared Coin: Flipping Many Coins

Stop flipping coins when there are enough

A



...



p_1

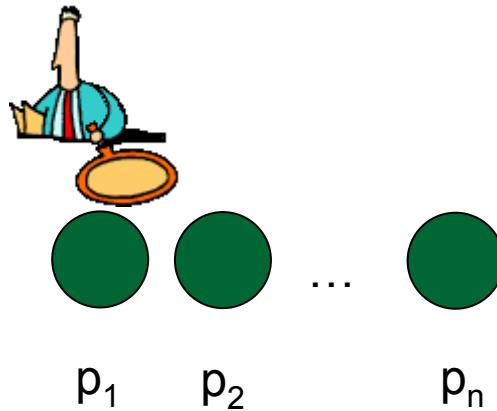
p_2

p_n

Shared Coin: How Many Coins?

Collect to check how many coins were flipped

A



Shared Coin: Balancing Act

Collect to check how many coins were flipped

[Bracha & Rachman 1991]

A



collect costs n steps

⇒ perform **fewer** collects



more coins written between collects

⇒ perform **more** collects



p_1

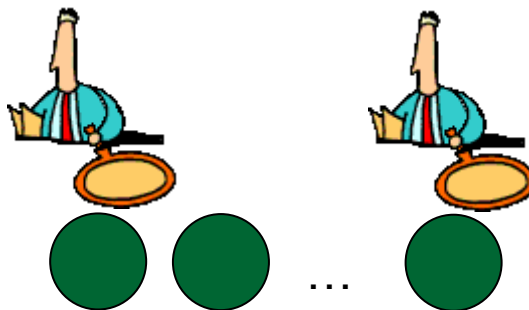
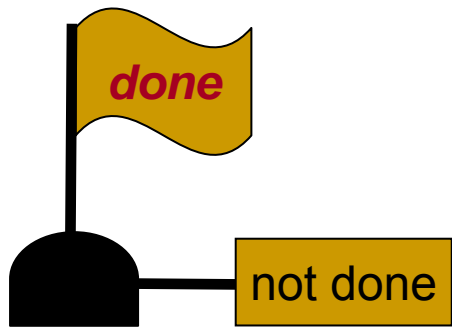
p_2

...

p_n

Shared Coin: Raising the Flag

A



checking *done* costs **1** step!

p_1

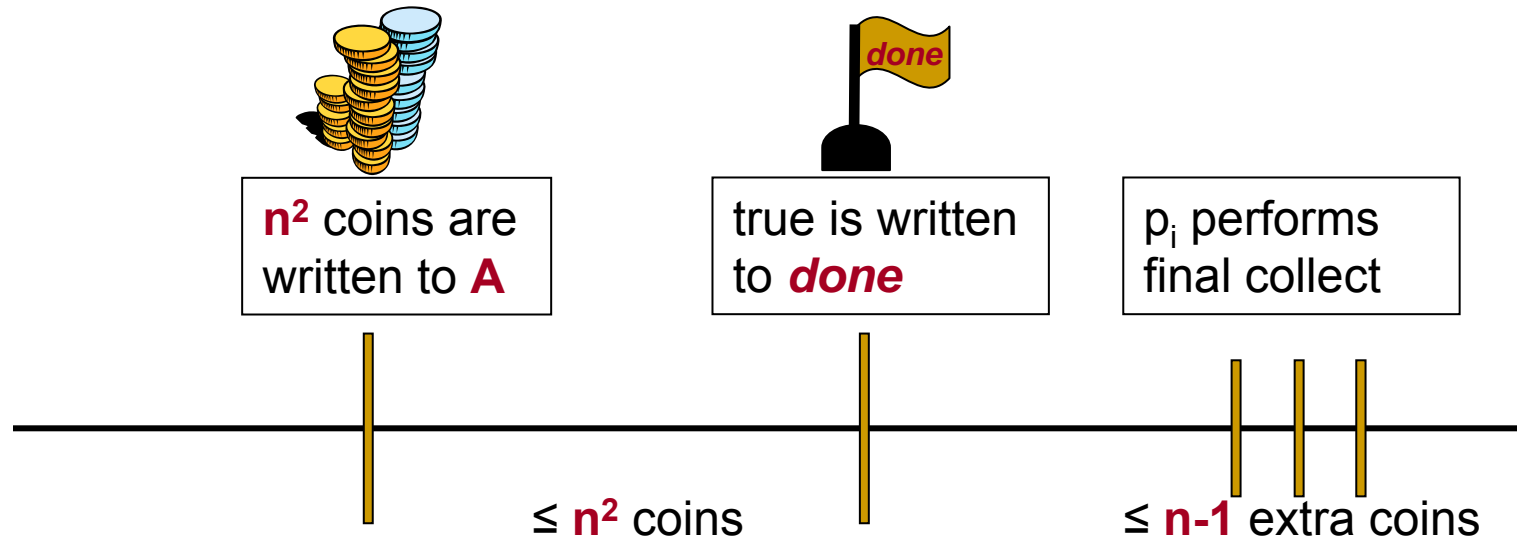
p_2

p_n

Shared Coin Algorithm

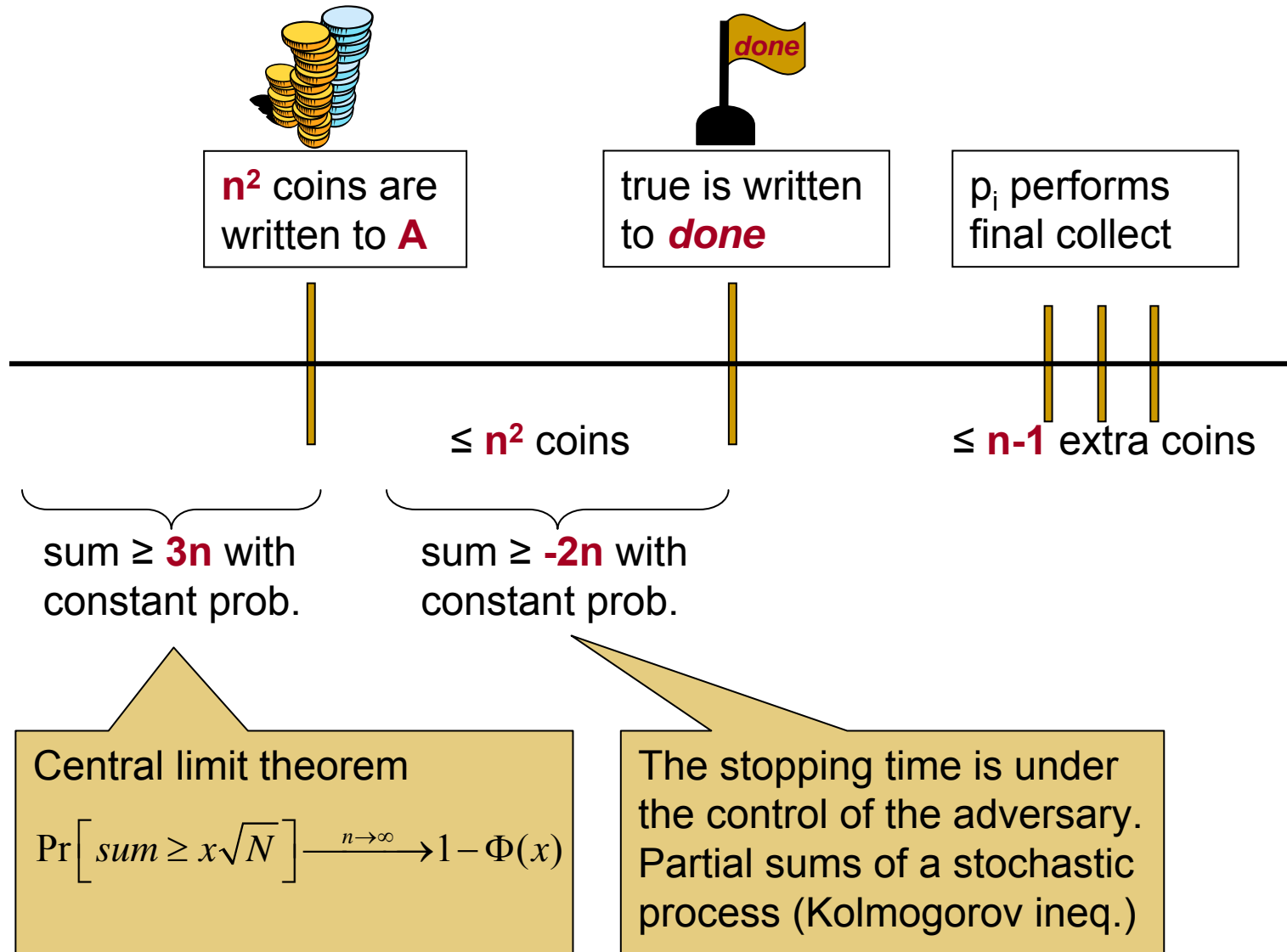
```
while not done do
  A[i] = {count++, sum+random(-1,+1)}
  every n coin flips
    collect A
    if > n2 coins were flipped
      then done = true
collect A
return the majority value of the coin flips
```

Total Step Complexity

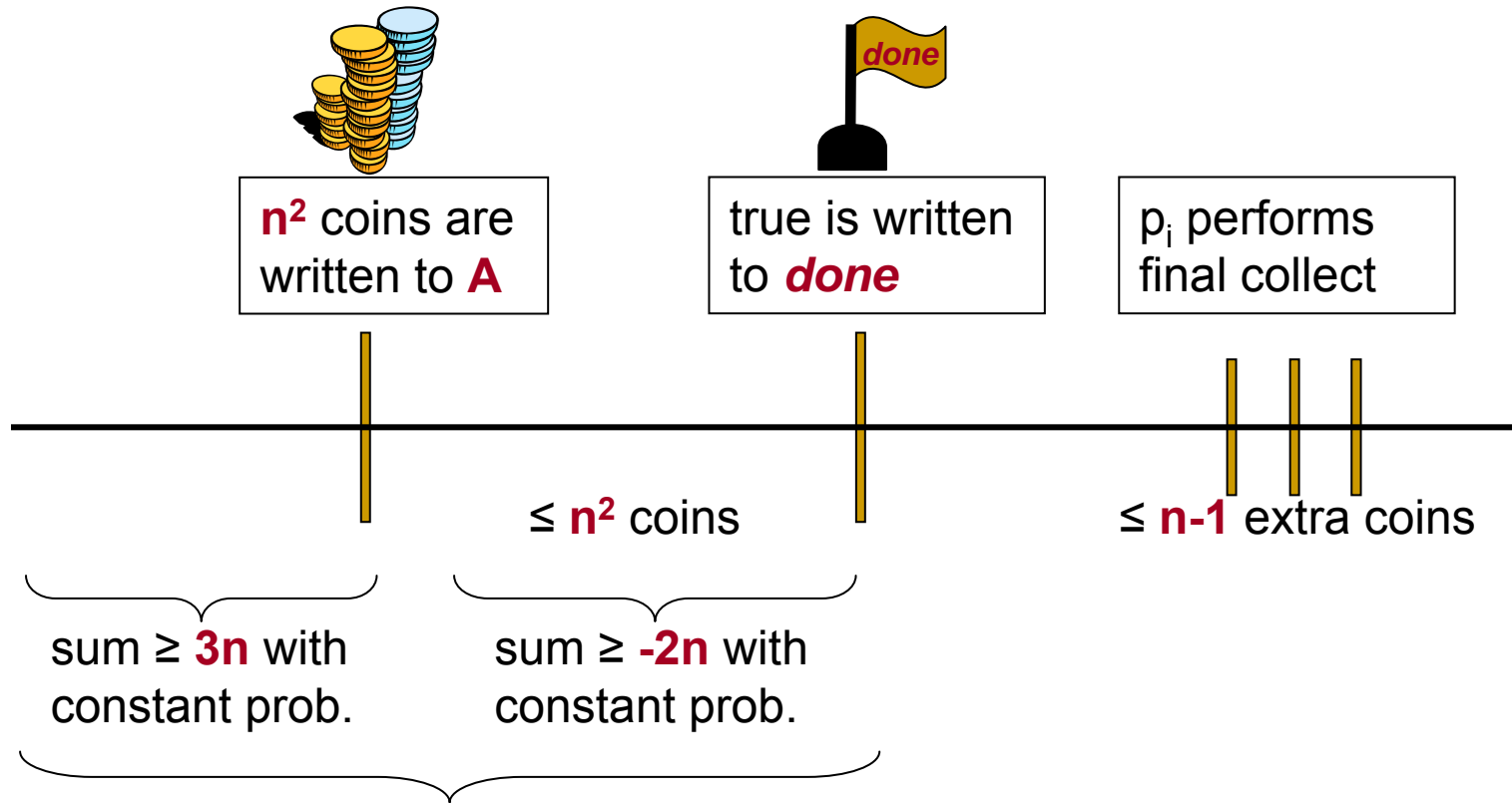


Writing $O(n^2)$ coins, and reading *done* $\Rightarrow O(n^2)$ operations,
Every n coins a process performs a collect (n read operations) $\Rightarrow O(n^2)$ operations

Agreement Parameter: Overview



Agreement Parameter: Summing Up



\Rightarrow sum $\geq n$ with constant prob.

\Rightarrow extra coins cannot make sum negative

\Rightarrow every output is **1** with constant prob. (same for **-1**)

Partial Sums of Random Variables

A sequence of independent random variables X_1, X_2, \dots, X_m
with $E[X_i]=0$ and $\text{Var}[X_i]=1$

Yields a sequence of partial sums $S_j = X_1 + \dots + X_j$

$$E[S_j] = E[X_1 + \dots + X_j] = E[X_1] + \dots + E[X_j] = 0$$

$$\text{Var}[S_j] = \text{Var}[X_1 + \dots + X_j] = \text{Var}[X_1] + \dots + \text{Var}[X_j] = j$$

$$\text{Kolmogorov's Inequality: } \Pr \left[\max_{1 \leq j \leq m} |S_j| \geq \lambda \right] \leq \frac{1}{\lambda^2} \text{Var}[S_m]$$

$$m = n^2, \lambda = 2n: \Pr \left[\max_{1 \leq j \leq n^2} |S_j| \geq 2n \right] \leq \frac{1}{4n^2} n^2 = \frac{1}{4}$$

$\Rightarrow S_j \geq -2n$ for every $1 \leq j \leq n^2$, with prob. $3/4$

So...

- The total step complexity of randomized consensus can be improved.

... but only so much

Lower Bounds: A Brief History

History repeats itself. ~~Historians~~ repeat each other
computer scientists

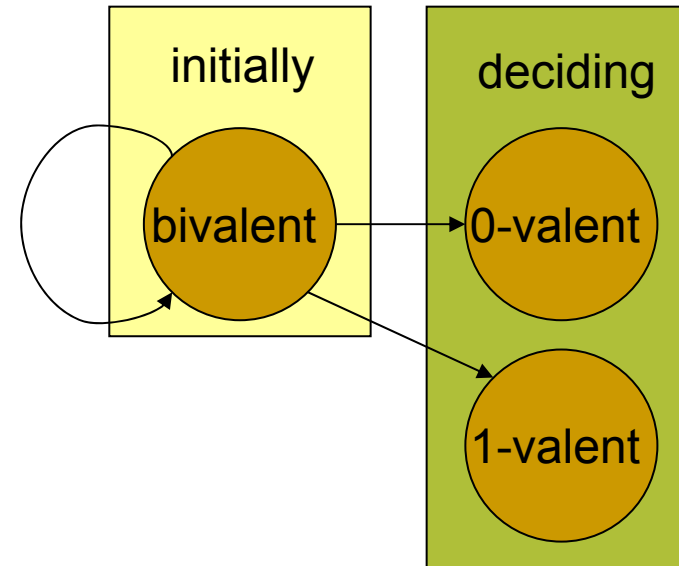
- $\Omega(n^2/\log^2 n)$ coin flips (asynchronous sm)
[Aspnes 1998]
- $\Omega(\sqrt{n/\log n})$ rounds (synchronous mp)
[Bar-Joseph & Ben-Or 1998]
- ⇒ $\Omega(n\sqrt{n/\log n})$ total step complexity
(asynchronous mp/sm)
⇒ Worse than previous bound

FLP: Valency

0-valent: only deciding 0

1-valent: only deciding 1

bivalent: deciding 0
and deciding 1



Valency w/ Randomization

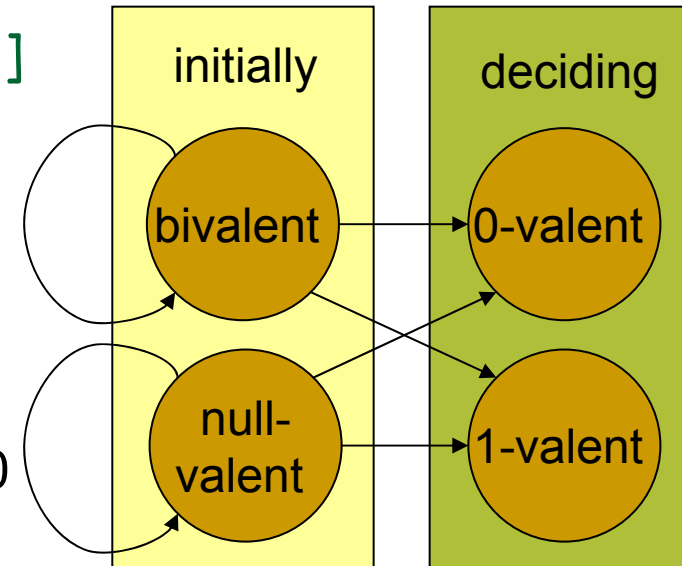
[Aspnes 1998]

0-valent: high probability for deciding 0

1-valent: high probability for deciding 1

bivalent: high probability for deciding 0
and high probability for deciding 1

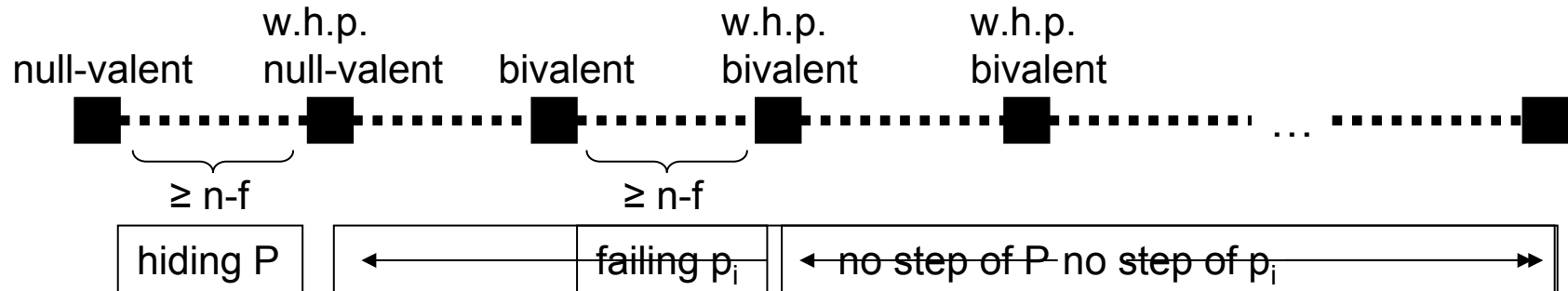
null-valent: **not** high probability for deciding 0
and **not** high probability for deciding 1



The Synchronous Lower Bound

[Bar-Joseph & Ben-Or 1998]

Executions proceed in rounds



May need to hide $O(\log n \sqrt{n})$ processes in a round
 \Rightarrow cannot go for more than $\Omega(\sqrt{n/\log n})$ rounds

☞ An asynchronous process can be delayed without failing

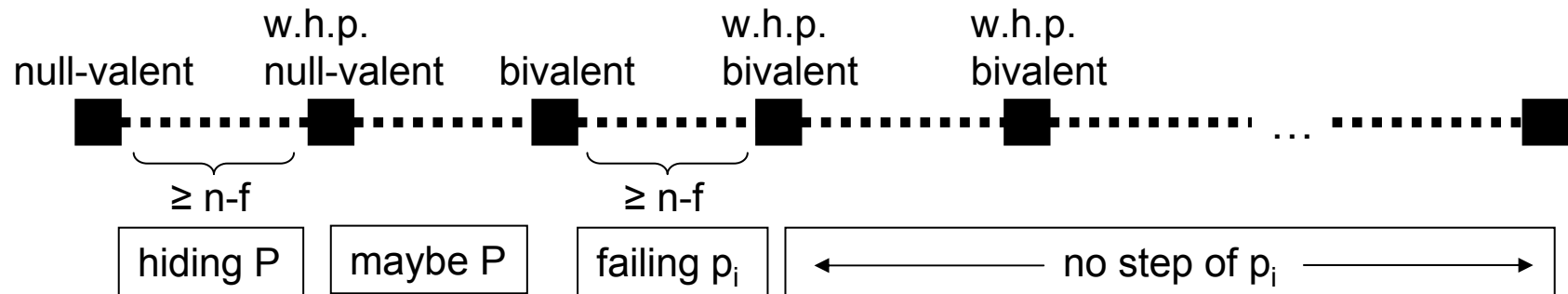
□ Synchronous model w/ **mobile** failures

\Leftrightarrow asynchronous model [Santoro, Widmayer 1989]

Our Asynchronous Lower Bound

Executions proceed in **layers** (sequence of $\geq n-f$ distinct processes)

[Moses & Rajsbaum 1998]

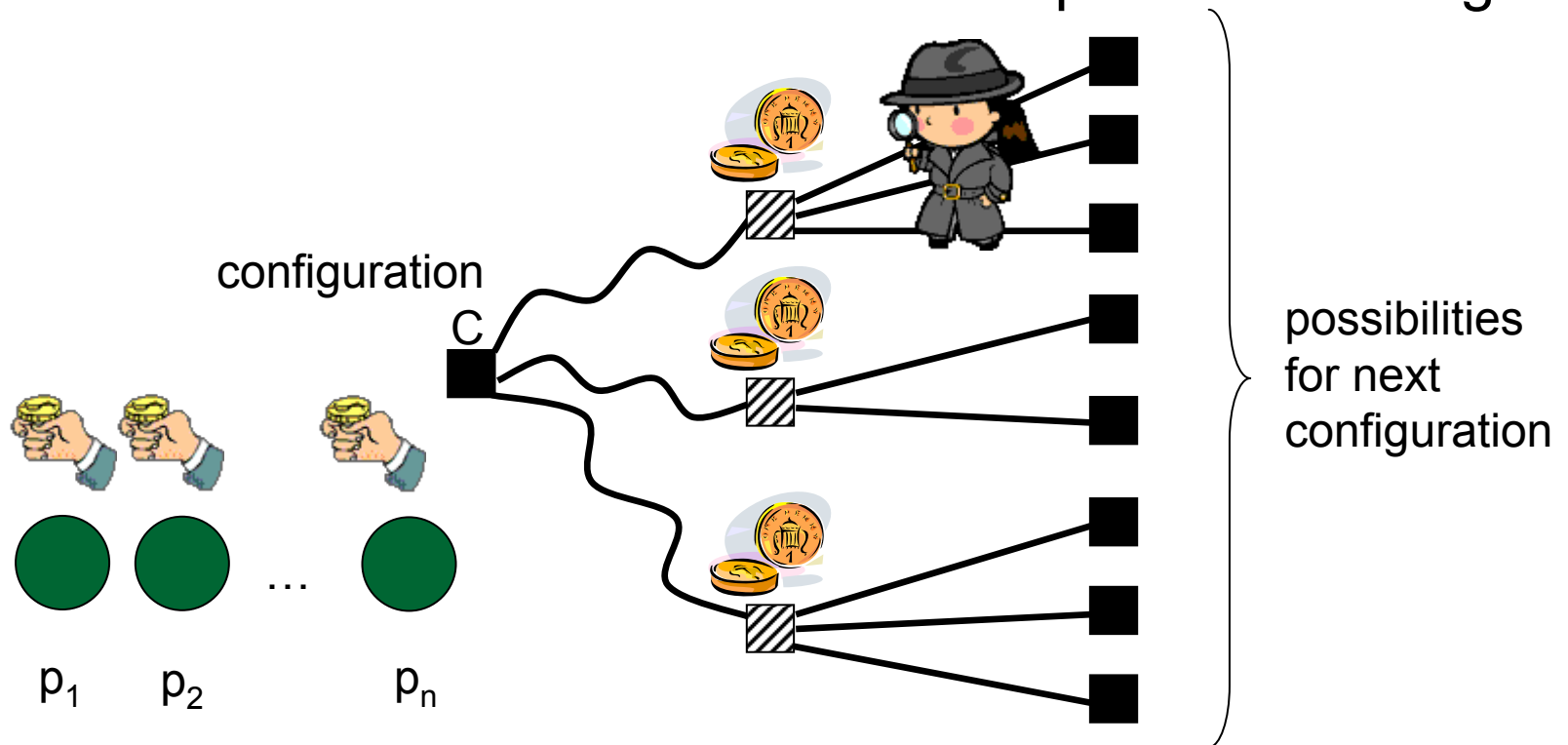


An asynchronous process can be hidden, by delaying, without failing

f layers, each with at least $n-f$ steps $\Rightarrow f(n-f)$ steps

Strong Adversary, In More Detail

Two sources of non-determinism: coin-flips & scheduling



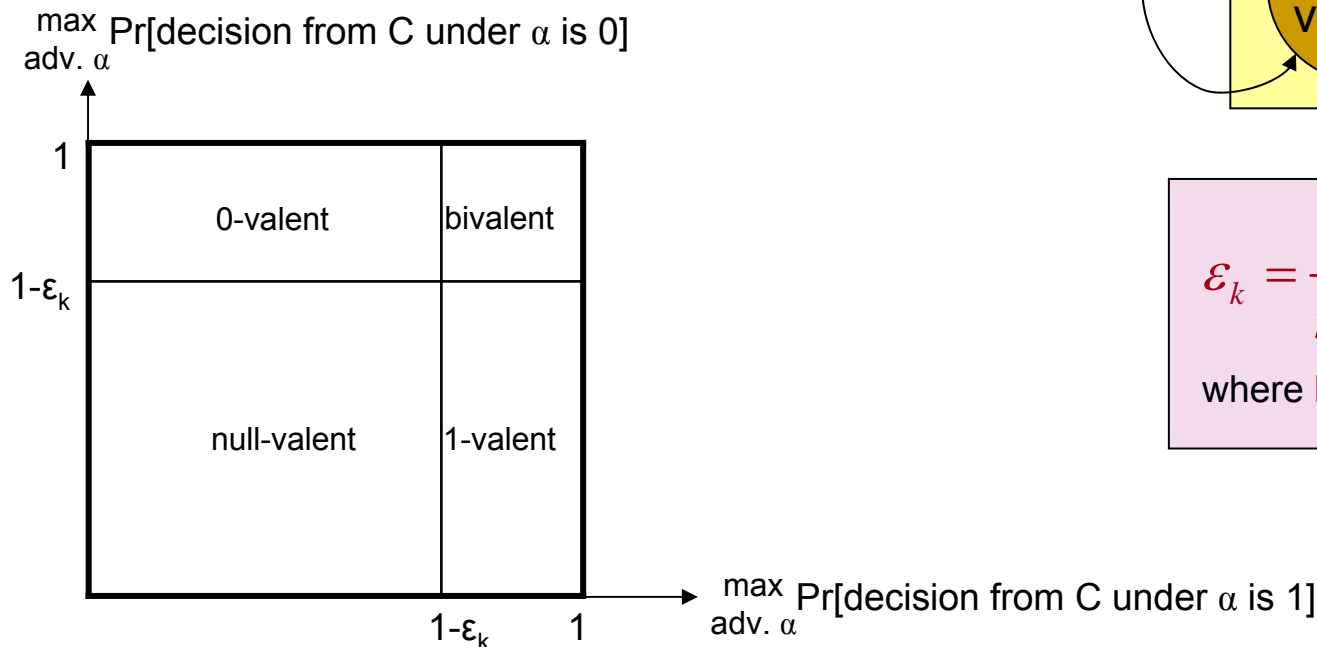
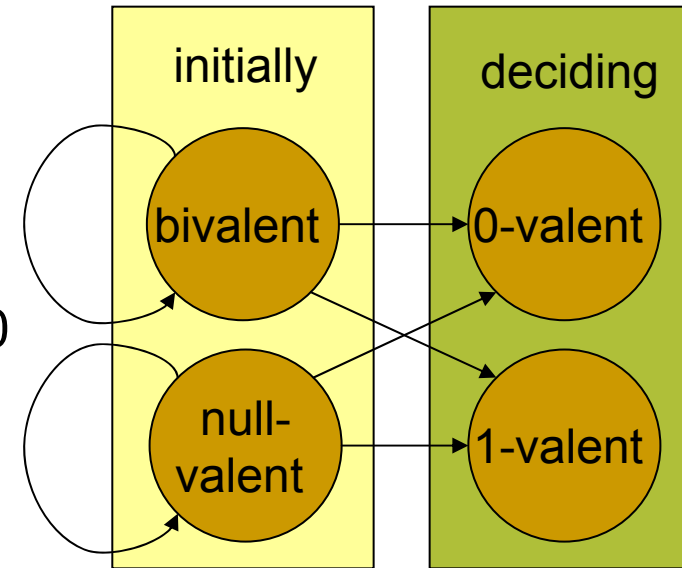
Valency, In More Detail

0-valent: high probability for deciding 0

1-valent: high probability for deciding 1

bivalent: high probability for deciding 0
and high probability for deciding 1

null-valent: **not** high probability for deciding 0
and **not** high probability for deciding 1



$$\epsilon_k = \frac{1}{n\sqrt{n}} - \frac{k}{(n-f)^3}$$

where k is the layer number

Remaining Null-Valent

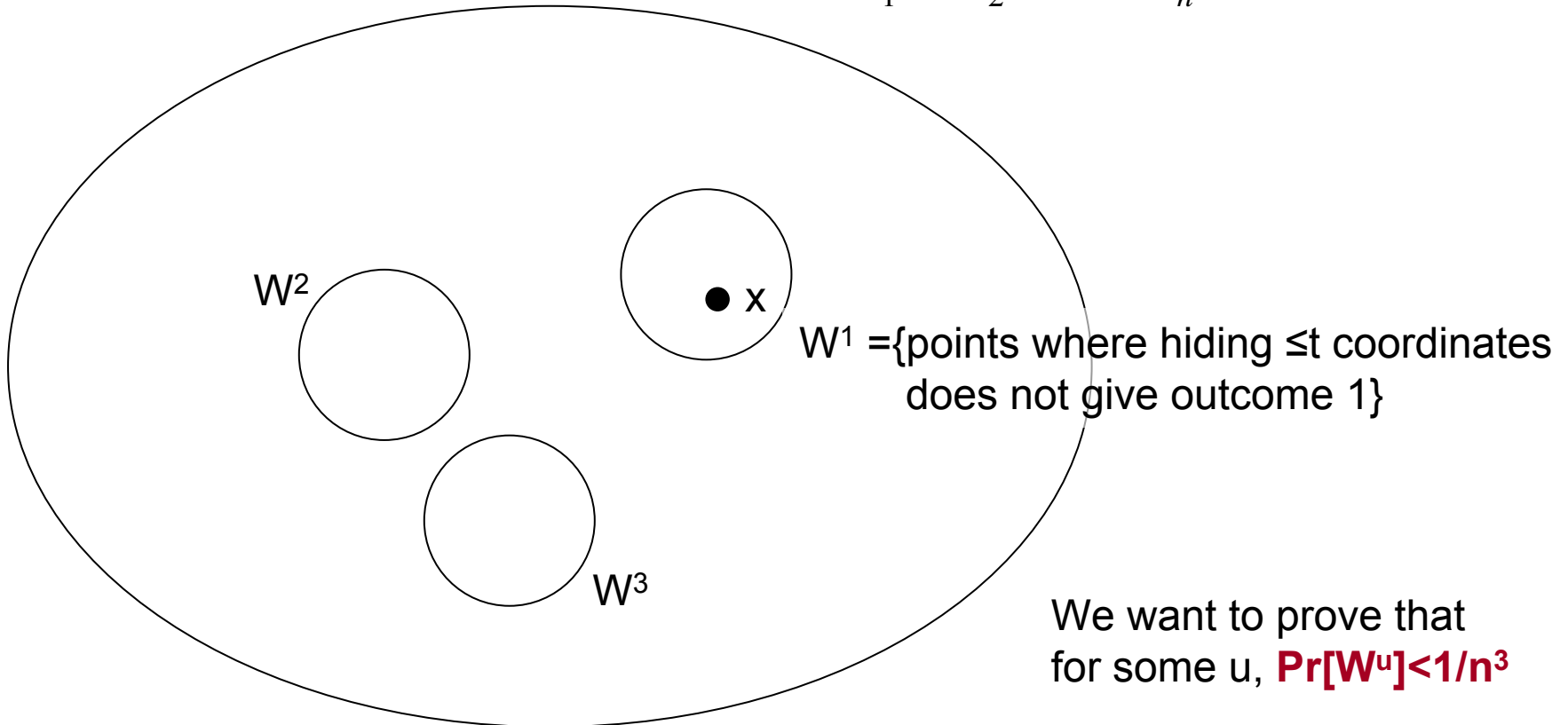
$$g: \{X_1 \cup \perp\} \times \{X_2 \cup \perp\} \times \dots \times \{X_n \cup \perp\} \rightarrow \{1, 2, 3\}$$

3-valued one-round coin-flipping game, X_i is a random variable

Theorem: one outcome has high probability

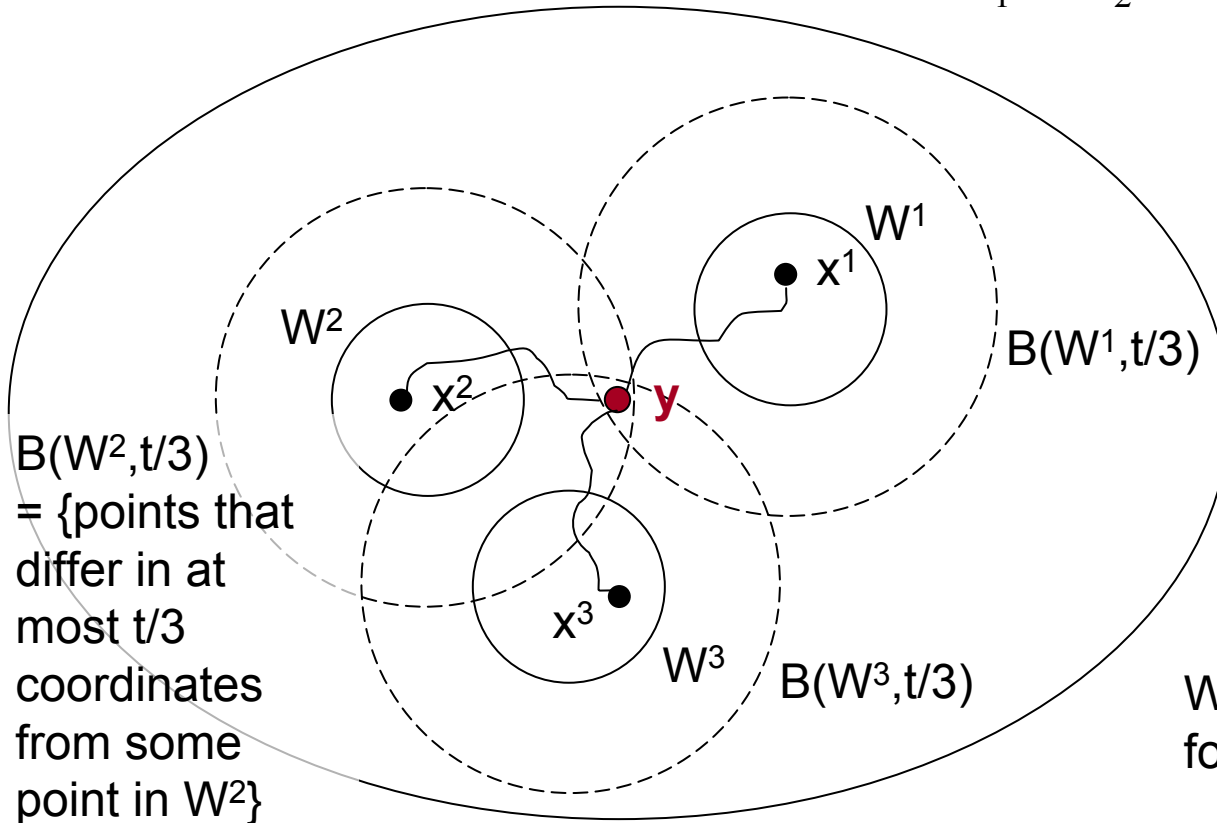
One Outcome has High Probability

The probability space $X = X_1 \times X_2 \times \dots \times X_n$



One Outcome has High Probability

The probability space $X = X_1 \times X_2 \times \dots \times X_n$



$$g(y_I) = g(x^1_I) = g(x^2_I) = g(x^3_I) = ?$$

$$y_I = \begin{array}{|c|c|c|c|c|} \hline y_1 & \perp & \dots & \perp & \perp & y_n \\ \hline \end{array}$$

$$x^1_I = \begin{array}{|c|c|c|c|c|} \hline & \perp & & \perp & \perp & \\ \hline \end{array}$$

$$x^2_I = \begin{array}{|c|c|c|c|c|} \hline & \perp & & \perp & \perp & \\ \hline \end{array}$$

$$x^3_I = \begin{array}{|c|c|c|c|c|} \hline & \perp & & \perp & \perp & \\ \hline \end{array}$$

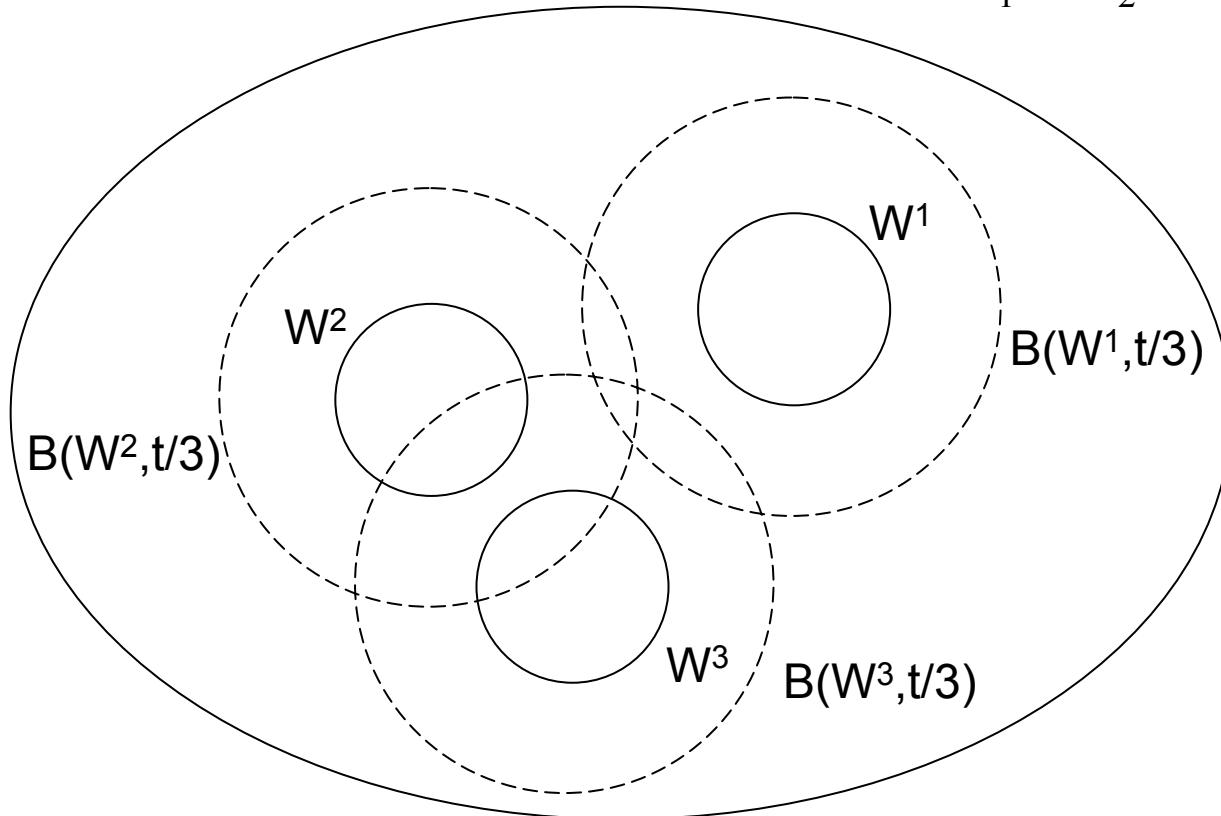
We want to prove that for some u , $\Pr[W^u] < 1/n^3$

W^u – adversary cannot reach outcome u . Assume $\Pr[W^u] \geq 1/n^3$

Isoperimetric inequality: if $\Pr[W^u] \geq 1/n^3$ then $\Pr[B(W^u, t/3)] \geq 1 - 1/n^3$

One Outcome has High Probability

The probability space $X = X_1 \times X_2 \times \dots \times X_n$



W^u – adversary cannot reach outcome u .

For some u , $\Pr[W^u] < 1/n^3$.

The adversary can reach the outcome u with probability $\geq 1 - 1/n^3$.

Must be the null-valent category.

Remaining Null-Valent

$$g: \{X_1 \cup \perp\} \times \{X_2 \cup \perp\} \times \dots \times \{X_n \cup \perp\} \rightarrow \{1, 2, 3\}$$

Product probability space = results of local coin-flips

\perp stands for a process not taking a step in the layer

- ❑ 0-valent or bivalent configuration \rightarrow the outcome of g is **1**
- ❑ 1-valent configuration \rightarrow the outcome of g is **2**
- ❑ Null-valent configuration \rightarrow the outcome of g is **3**

\Rightarrow some outcome can be forced (by hiding processes) w.h.p.

Since we started from a null-valent configuration

1. not high probability of deciding 0
 \Rightarrow cannot have high probability for reaching a 0-valent configuration
2. not high probability of deciding 1
 \Rightarrow cannot have high probability for reaching a 1-valent configuration

\Rightarrow the **null-valent** category must be the one with high probability

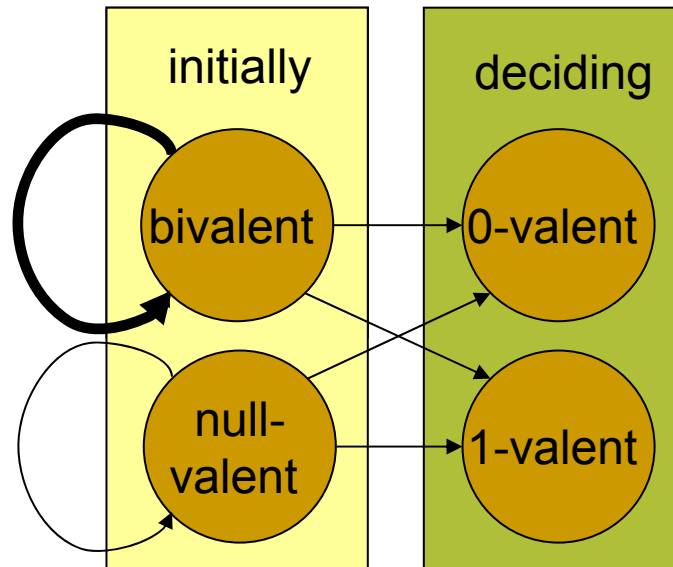
Must Remain Null-Valent

- Null-valent configuration has probability for deciding 0 at most $1-\varepsilon_k$
- Assume 0-valent or bivalent configuration can be reached with probability $1-1/m^3$
- New configuration has probability for deciding 0 at least $1-\varepsilon_{k+1}$
- Together, the probability for deciding 0 from the null-valent configuration is:

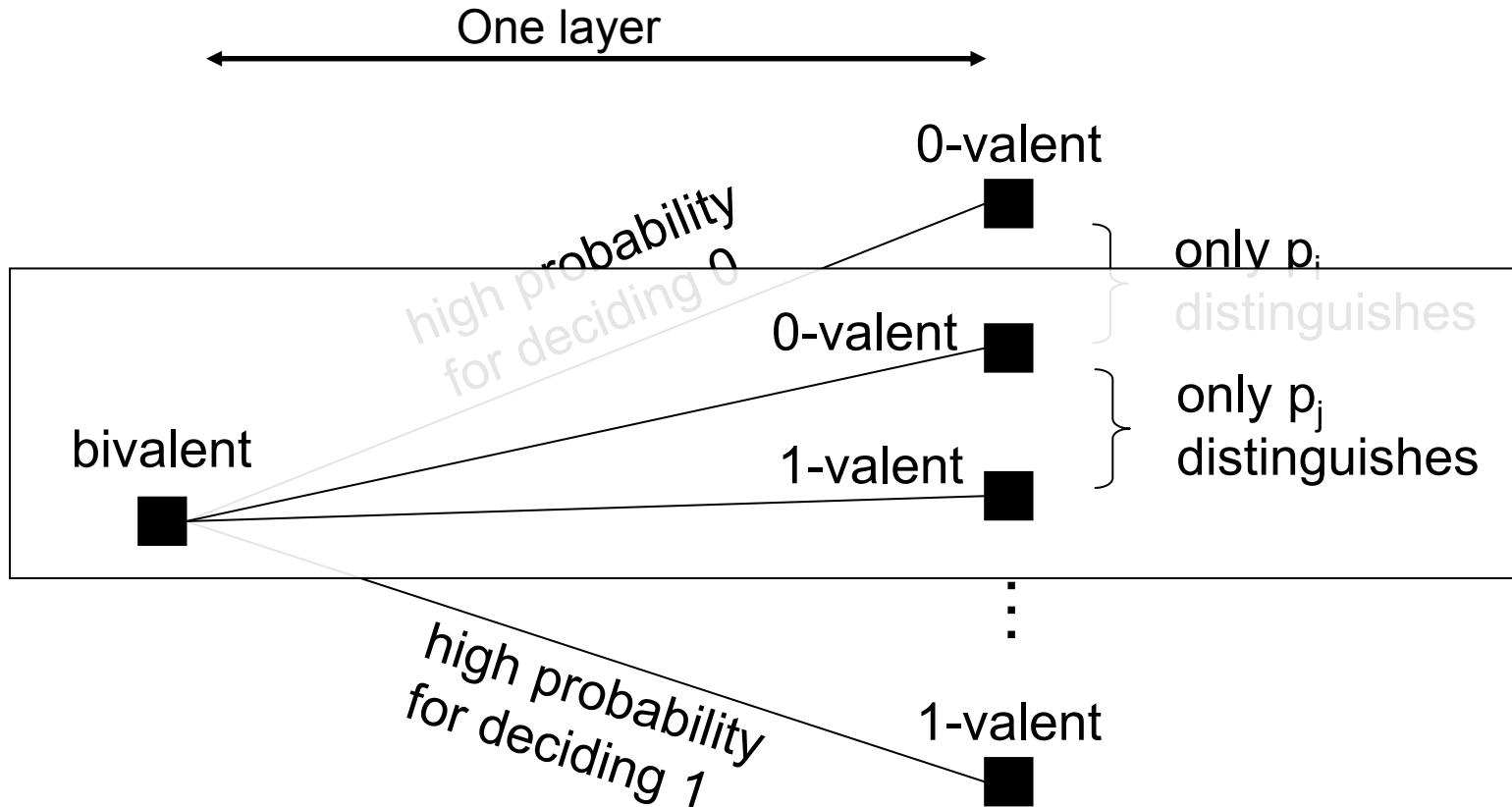
$$\begin{aligned} &\geq \left(1 - \frac{1}{(n-f)^3}\right) (1 - \varepsilon_{k+1}) = \left(1 - \frac{1}{(n-f)^3}\right) \left(1 - \frac{1}{n\sqrt{n}} + \frac{k+1}{(n-f)^3}\right) \\ &= 1 - \frac{1}{n\sqrt{n}} + \frac{k}{(n-f)^3} + \frac{1}{(n-f)^3} \frac{1}{n\sqrt{n}} - \frac{k+1}{(n-f)^6} \\ &> 1 - \frac{1}{n\sqrt{n}} + \frac{k}{(n-f)^3} = 1 - \varepsilon_k \end{aligned}$$

Cannot have high probability for reaching a 0-valent or bivalent configuration

Big Picture: Bivalent Configurations

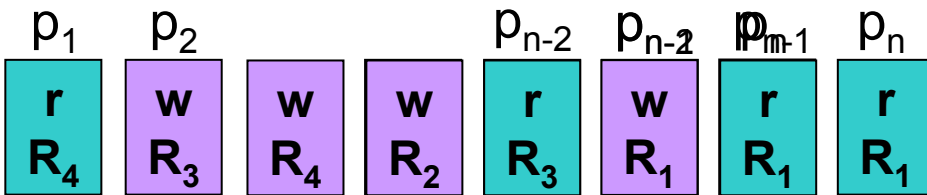
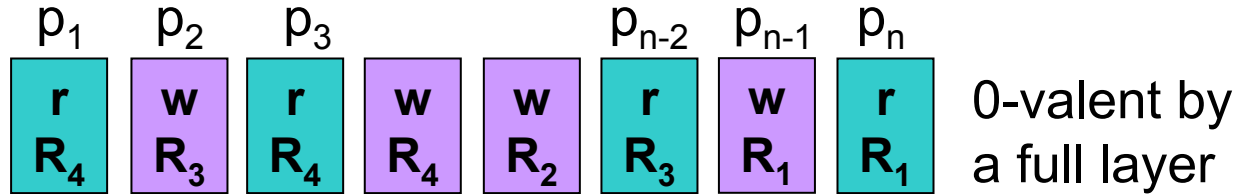


Bivalent Configurations: Connectivity

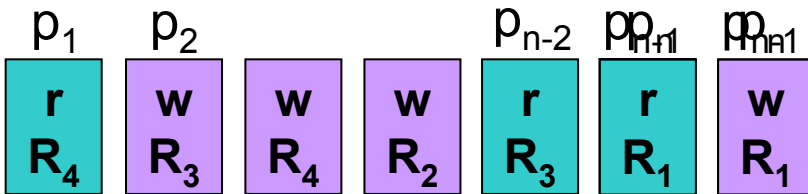


fail $p_i \Rightarrow$ must have same valence \Rightarrow null-valent

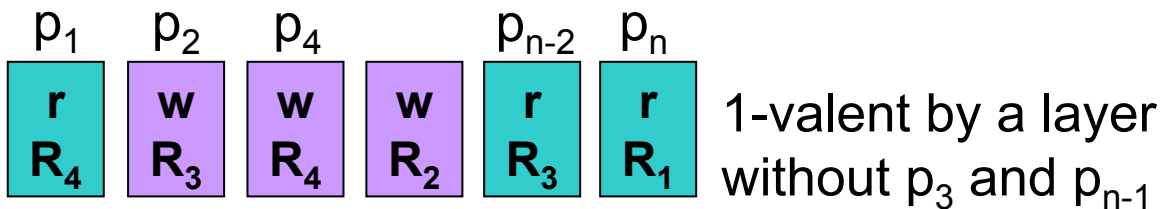
Connectivity in Shared Memory



only p_3 distinguishes



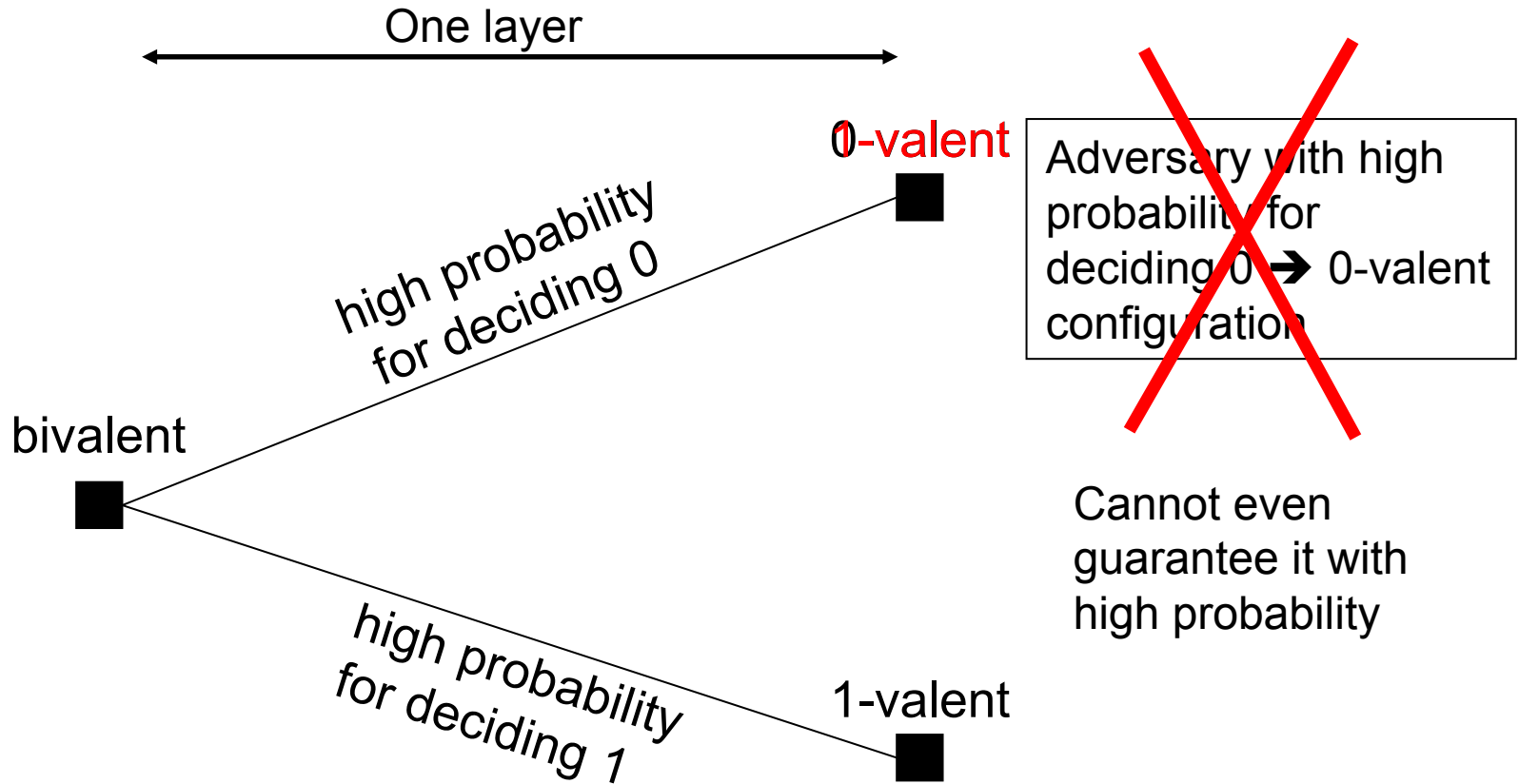
both p_{n-1} and p_n distinguish



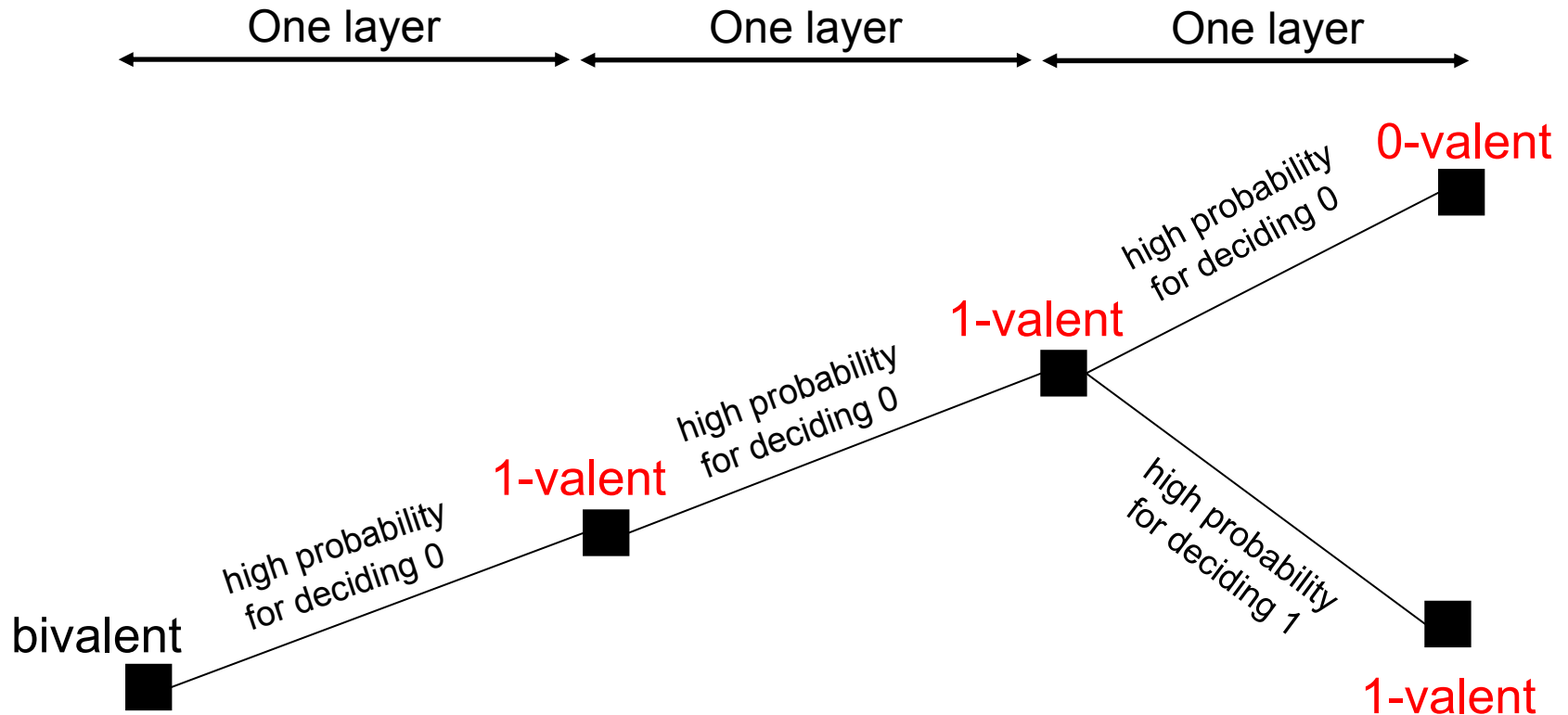
only p_{n-1} distinguishes

0-valent and 1-valent configurations which are distinguishable only to p_i

Bivalent Configurations Revisited



v-Switching Configurations



The above 0-valent configuration exists with high probability (at least $1-\epsilon_k$)

And now continue as from a bivalent configuration

Individual Step Complexity


The expected number of steps taken by a single process

- $O(n \log^2 n)$ using single-writer registers [Aspnes & Waarts 1996]
- In our algorithm, a process running solo must generate all n^2 coins alone

A multi-writer register allows randomized consensus with $O(n \log n)$ individual step complexity

- The coins of a process have increasing weights
- ☞ A process that runs alone flips only $O(n \log n)$ coins
- ☞ But now the coins are not independent – weights of flipped coins can be different

Wrap-Up: What's this Work About?

- At face value: Clever math...
 - Stochastic processes, martingales, Kolmogorov
- Really, confluence of models
 - Asynchronous \Leftrightarrow synchronous w/ mobile failures
 - Taking connectivity arguments from message passing to (multi-writer) shared memory
 - Multi-writer bit induces instantaneous views
- Still needs to do your math... 

What's Next?

Randomized consensus revisited...

- ✓ Weaker adversaries?
- Single-writer registers?
- Message-passing model?
- Byzantine failures?
 - Cryptographic requirements for BFT?

And other problems...

- k-set consensus, renaming
- Seems to defy existing techniques