ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
DIPARTIMENTO DI
INFORMATICA - SCIENZA E INGEGNERIA

A FORMAL ANALYSIS OF THE BITCOIN PROTOCOL

COSIMO LANEVE ADELE VESCHETTI

CONTENTS

THE BITCOIN PROTOCOL

* it implements a **replicated database** where blocks are only addded
* the replicas are stored on nodes of an **unreliable peer-to-peer system**
* if any node **tries to update the database** all other nodes **can detect and prevent it**

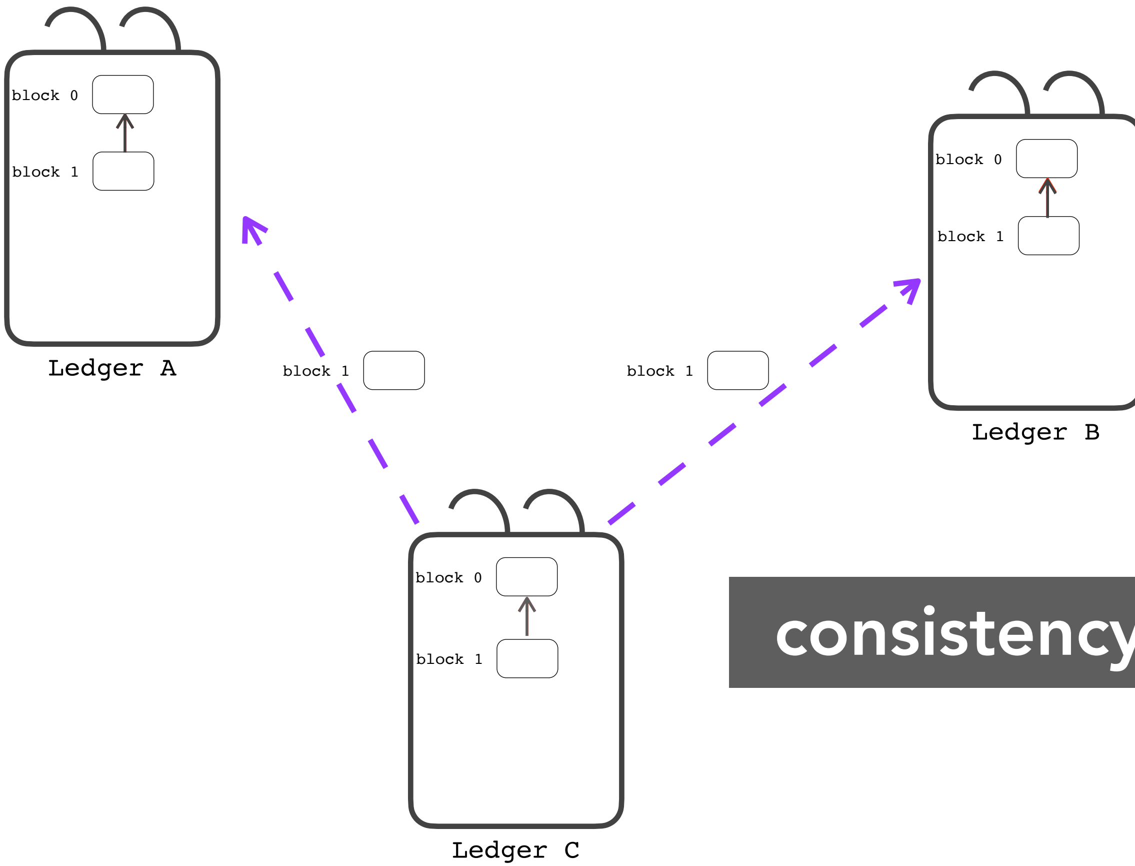
the protocol realizes a decentralized ledger

BITCOIN: THE CONSENSUS ALGORITHM

there is no algorithm reducing to 0 the probability that a distributed database is inconsistent [Fischer-Lynch-Paterson 1985]

BITCOIN: THE CONSENSUS ALGORITHM

the **blockchain** is **a** longest path in the **ledger** beginning at a leaf node



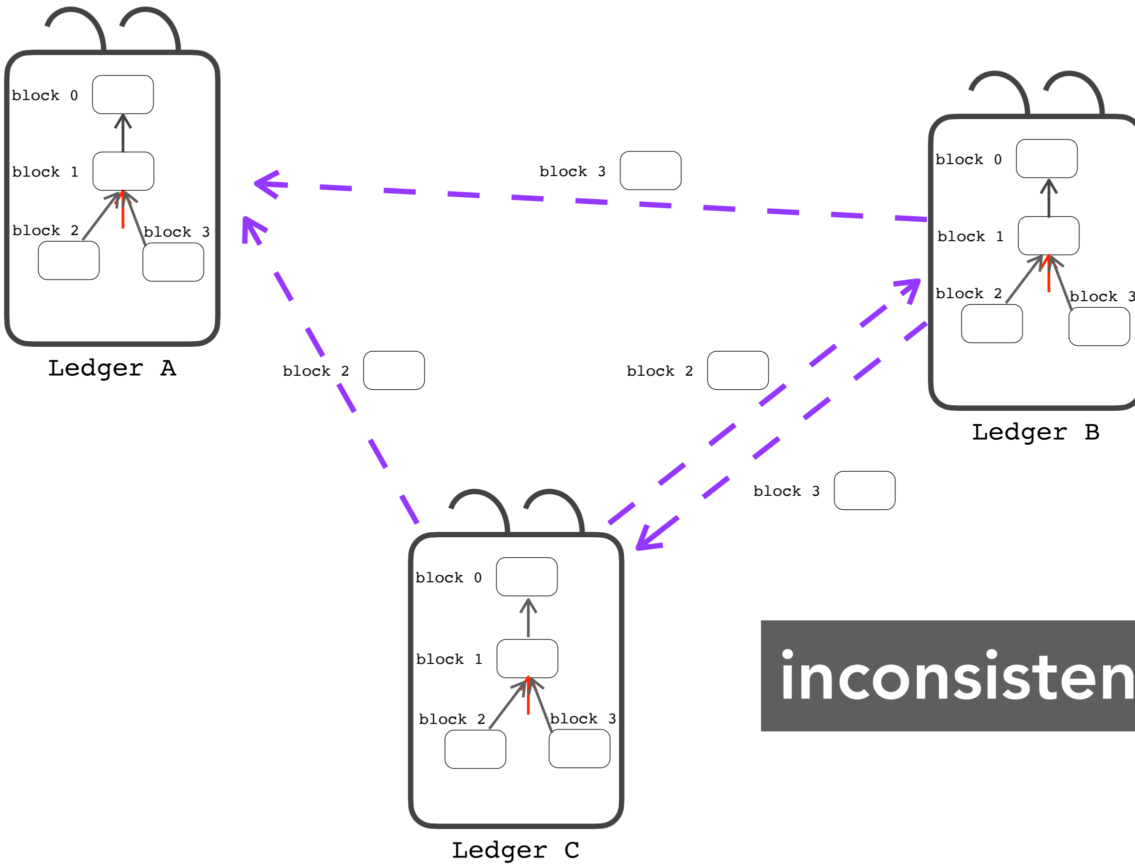
consistency again!

the consistency is reached by admitting inconsistent states

the situation is worse than this!

BITCOIN: THE CONSENSUS ALGORITHM

the **blockchain** is **a** longest path in the **ledger** beginning at a leaf node



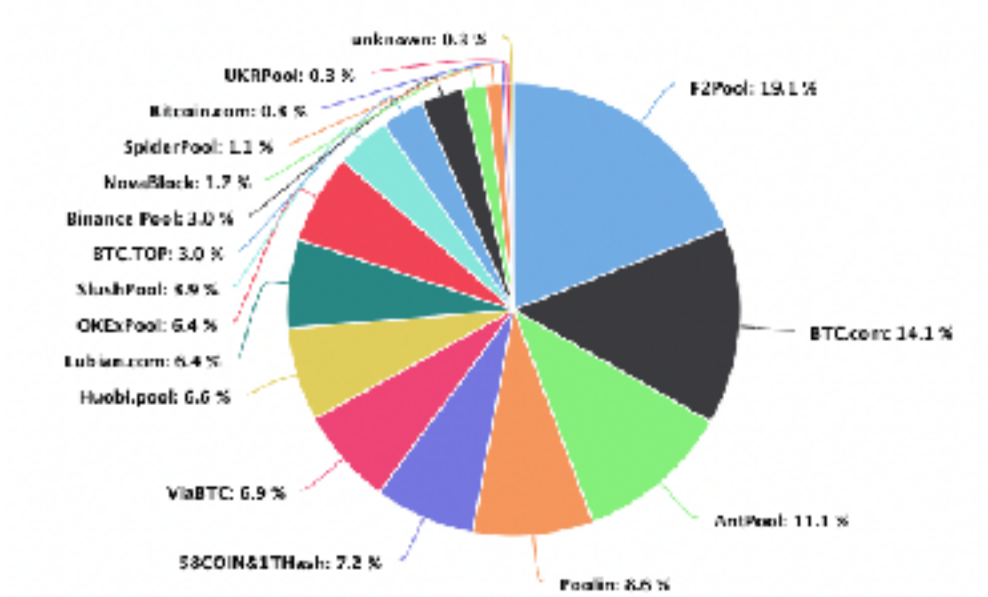
inconsistent state: a fork

the probability of this inconsistency is "low" in Bitcoin

BITCOIN CORRECTNESS

Bitcoin nodes cluster because, mining a new block, amounts to win a computationally expensive challenge — **proof of work**

BITCOIN MINING POOLS
MAY 2020



the system is **secure** as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes

OUR ANALYSIS

we undertake a formal analysis of the Bitcoin protocol

- by modelling the protocols with a **stochastic process calculus**
- we use an extension of `PRISM` with the **ledger datatype**: `PRISM+`
- in `PRISM+` **channels have a rate** (we can easily model broadcast delays and mining speed)
- because `PRISM+` has a formal model, we demonstrate the key properties of the protocol
- because `PRISM+` has a simulator, we may (also) verify our results in silico

PRISM+ DEFINITION OF BITCOIN

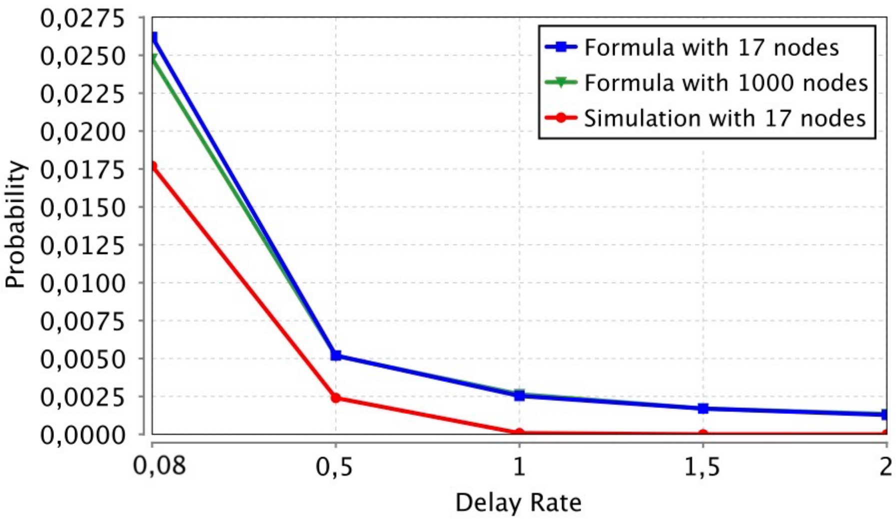
$$\mathtt{MINER}_1 \parallel \cdots \parallel \mathtt{MINER}_n \parallel \mathtt{NETWORK}$$

```
module Miner_i
  integer Miner_i_STATE = Init;
  block b_i = (gen^0,gen^0);
  ledger L_i = ⟨{(gen^0, gen^0)}; gen^0⟩;
  integer c_i = 0;
  queue QMiner_i = [];

  [] Miner_i_STATE=Init ->
              mR×hR_i : c_i'= c_i+1
                        & b_i' = NewB(Miner_i,c,handle(L_i))
                        & Miner_i_STATE'= Winner;

  [] Miner_i_STATE=Init&canAdd(L_i,top(QMiner_i)) ->
                     r : QMiner_i'= dequeue(QMiner_i)
                        & L_i'= addB(L_i,top(QMiner_i));

  [] Miner_i_STATE=Init&!canAdd(L_i,top(QMiner_i)) ->
                     r : QMiner_i'= deq_enq(QMiner_i);

  [addBlock_i] Miner_i_STATE=Init ->
                   r_i : QMiner_i'= enqueue(QMiner_i,top(Q_i))

  [addBlock_i] Miner_i_STATE=Winner ->
                   r_i : L_i'= AddB(L_i,b_i)
                        & Miner_i_STATE'= Init;
endmodule
```

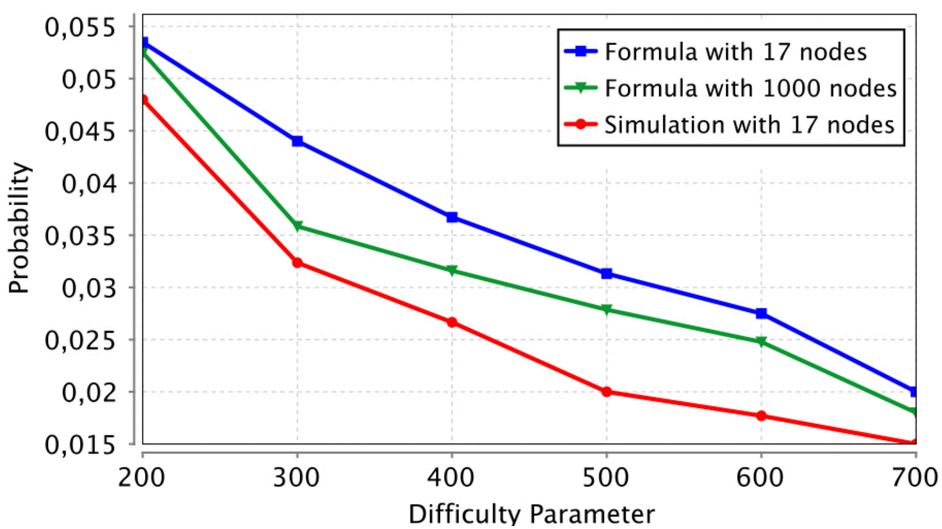
OUR RESULTS

we compute probabilities of forks that are functions of

* the number of nodes
* the broadcast delay
* the rates of mining
* the cryptopuzzle difficulty



Probability of a fork of length 1 by varying the broadcast delay



Probability of a fork of length 1 by varying the cryptopuzzle difficulty

the probability of a fork is 10^{-2} in Bitcoin

OTHER RESULTS

we also analyze

* the probability of creating forks of increasing length
* the attack of a hostile miner that tries to create an alternative chain

a companion paper extends `PRISM` to `PRISM+` and reports a bunch of simulations

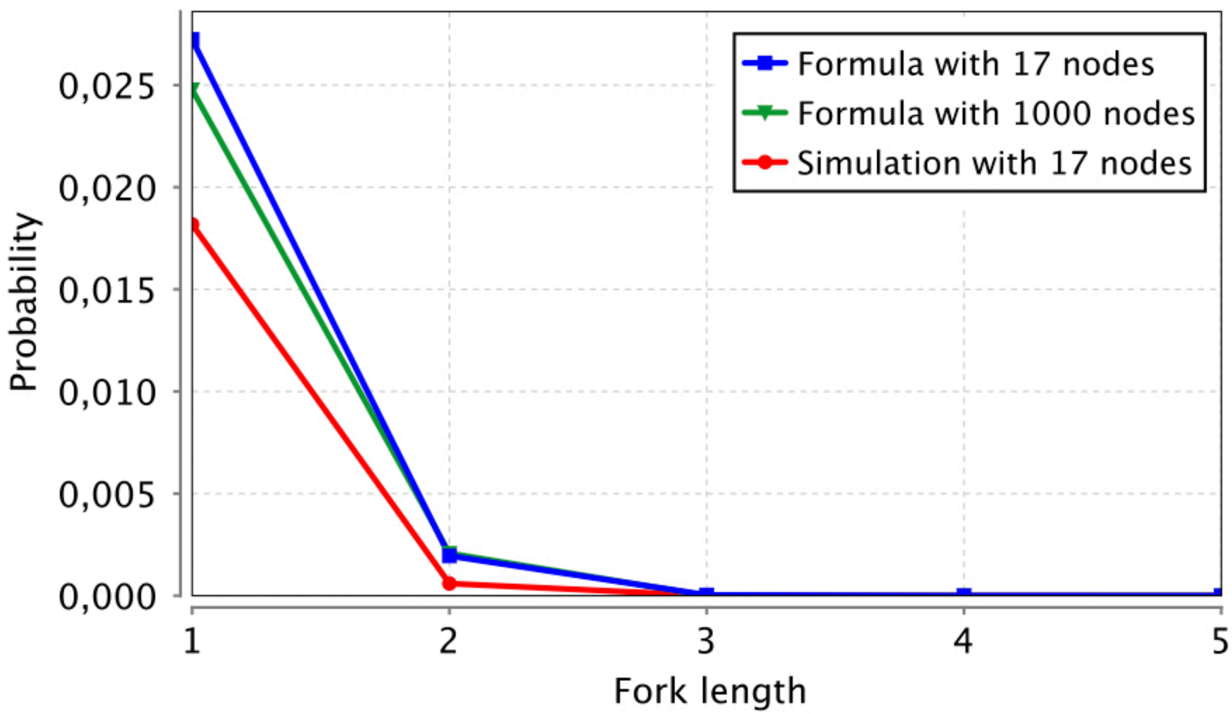
HAPPY 60 YEARS MAURIZIO!

QUESTIONS

FORK OF INCREASING LENGTH

we analyze

* the probability of creating forks of increasing length

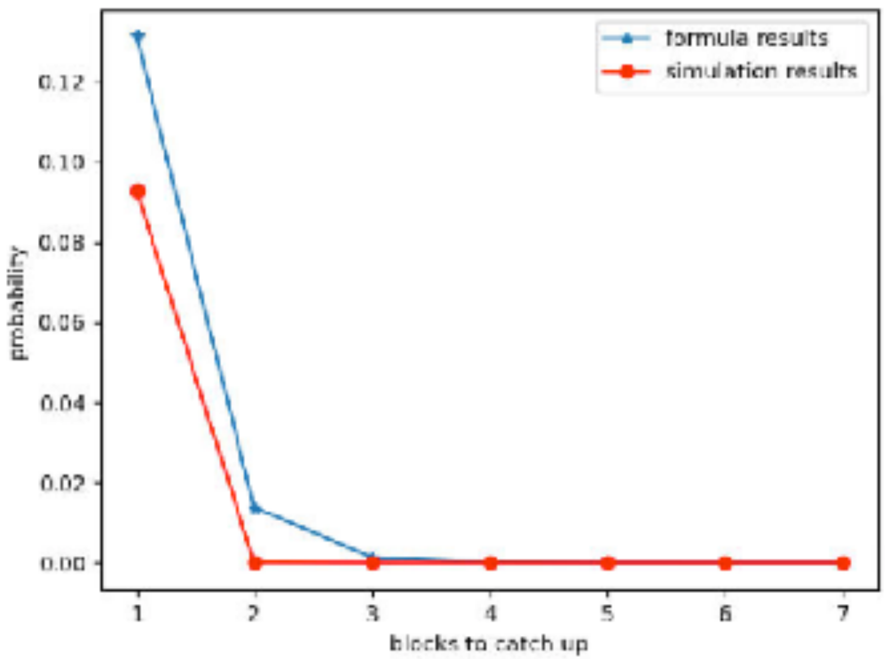


Probability of a fork of increasing length.

ANALYSIS OF AN ATTACK

we also analyze a double spending attack scenario

* the behaviour of the malicious miner differs for the fact that mines a block that is not the correct one



Probability of a successful attack for one of the main pools of Bitcoin