A Formal Analysis of Blockchain Consensus

Cosimo Laneve and Adele Veschetti

4th February 2020

3rd Distributed Ledger Technology Workshop (DLT 2020)

Our Analysis

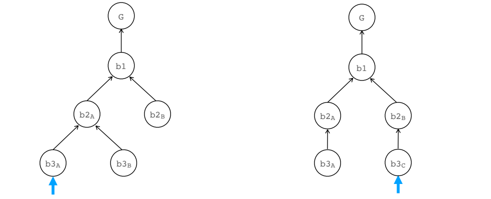
- We model the Blockchain protocol with a variant of stochastic pi calculus
  - Channels have rates and these rates drive the dynamic behaviour of processes
  - All enabled activities attempt to proceed, but fastest ones succeed with higher probabilities
- We define the behaviour of its key participants, the miners
- We derive the specification of the whole system as a parallel composition of miners
- The properties of the blockchain protocol are derived by studying the model of the stochastic pi calculus
- Later, we apply the same technique to analyze an attack to Blockchain

The Ledger Data Type

A *ledger*, noted L, L′, ⋯, is a pair (T, h) where

- T, noted by *tree*(L), is a nonempty *tree of blocks*
- *h* is an *handle*, noted by *handle*(L). It is always a pointer to a leaf block at maximal depth.

The root of *tree*(L) is called *genesis block*.



the ledger L1 – $L1 \uparrow = [b3_A, b2_A, b1, G]$ the ledger L2 – $L2 \uparrow = [b3_C, b2_B, b1, G]$

The *blockchain of* L, noted $L\uparrow$, is the sequence $[b_0, b_1, b_2, \cdots]$ such that $b_0 = handle(L)$ and, for every i, b_{i+1} is the parent of b_i.

The abstract model of Blockchain

The key participants of the protocols are the *miners* that create blocks of the ledger and broadcast them to the nodes of the network. In our model a blockchain system is

$$(\nu z_1 @ r_1, \cdots, z_n @ r_n)\Big(\prod_{z_i \in \{z_1, \cdots, z_n\}} \mathsf{Miner}_{\{z_1, \cdots, z_n\} \setminus z}(\mathtt{G}, \emptyset, z_i) \Big)$$

where $\mathtt{G}$ is the ledger with the genesis block only.

How we model it

- The system consists of n miners
- They communicate through channels $z_1, \cdots, z_n$ with rates $r_1, \cdots, r_n$,

Formally, the definition of a Miner_U is

$$
\begin{aligned}
\text{Miner}_U(L, X, z) \;=\; & (\nu\ w@r_w)(\\
& \quad (\ z?(b).\ \text{Miner}_U(L, X^\frown b, z) \\
& \quad\ + w\,!\,\mathit{newBlock}(L) \\
& \quad\ + \textbf{if}\ (X = \varepsilon)\ \textbf{then}\ \tau_{r'}.\ \text{Miner}_U(L, X, z) \\
& \qquad \textbf{else if}\ (\mathit{head}(X).\texttt{id} \in L.\texttt{blocks})\ \textbf{then} \\
& \qquad\qquad \tau_{r'}.\text{Miner}_U(\mathit{addBlock}(L, \mathit{head}(X)), \mathit{tail}(X), z) \\
& \qquad \textbf{else}\ \tau_{r'}.\ \text{Miner}_U(L, \mathit{tail}(X)^\frown \mathit{head}(X), z) \\
& \quad)\ \mid\ w?(b).(\text{Miner}_U(\mathit{addBlock}(L, b), X, z) \mid \textstyle\prod_{z' \in U} z'\,!(b)) \\
&)
\end{aligned}
$$

Properties

Definition
A state of a blockchain system is called *completed* when it is structurally equivalent

$$(\nu\ z_1@r_1, \cdots, z_n@r_n)\Big(\prod_{i \in 1..n} \mathsf{Miner}(B_i, \varepsilon, z_i) \Big) .$$

Namely, in a completed state, there is no block to deliver and the blocks in the local lists X_i have been already inserted in the corresponding ledgers.

Theorem

Let P be a completed state of a blockchain system consisting of n miners with ledgers $B_1, \ldots, B_n$, respectively.

Let B_1 and B_{k+1} have fork of length m. Then the probability $\text{Prob}(P_{\rightsquigarrow m+1})$ *to reach a completed state with fork of length $m+1$ is smaller than*

$$\sum_{i,\ell,j} \Theta(i,\ell,j), \textit{where} \begin{cases} 1 \le i \le n \\ H \subset \{1,\cdots,n\} \setminus i,\ \ell = |H| \\ i \le k \ \Rightarrow\ j \in \{k+1,\cdots,n\} \setminus H \\ i > k \ \Rightarrow\ j \in \{1,\cdots,k\} \setminus H \end{cases}$$

where

$$\Theta(i,\ell,j) = \frac{r_{w_i}\ r_{w_j}}{R\ (R + (n-1-\ell)r)} \prod_{1 \le h \le \ell} \frac{h\ r}{R + (n-h)r} \prod_{1 \le a \le 2n-2-\ell} \frac{a\ r}{R + a\ r}$$

1. r_{w_i} represents the time $i-$th node needs to solve the block problem:

$$r_{w_i} = \frac{h_i}{D}, \forall i \in \{1, \ldots, \mathtt{n}\}$$

2. In the blockchain protocol the messages incur in a propagation delay, represented by r_i

Analysis of Possible Attacks

- We model the scenario in which a hostile miner tries to create an alternate chain faster than the honest one
- The difference with Miner_U is that the dishonest miner, called ${\text{Miner}^{\mathsf{D}}}_U$, mines on a block d that is not the correct one
- We use the operation $\mathit{newBlock}^{\mathsf{D}}(\mathtt{L}, d)$ that takes a ledger `L` and a block $d \in \mathtt{L.blocks}$ and returns a new block whose pointer is d (therefore it will be connected to d).

Model of an attacker

The definition of $\mathsf{Miner^D}_U$ is

$$
\begin{aligned}
&\mathsf{Miner^D}_U(L, X, z, d) \; = \\
&\quad (\nu\, w@r)\Big(\quad \big(z?(b).\ \mathsf{Miner^D}_U(L, X^\frown b, z, d) \\
&\qquad\qquad + w!\,newBlock^{\mathsf{D}}(L, d) \\
&\qquad\qquad + \textbf{if } (X = \varepsilon) \textbf{ then } \tau_{r'}.\ \mathsf{Miner^D}_U(L, X, z, d) \\
&\qquad\qquad\quad \textbf{else if } (head(X).\mathtt{id} \in L.\mathtt{blocks}) \textbf{ then} \\
&\qquad\qquad\qquad\qquad \tau_{r'}.\mathsf{Miner^D}_U(addBlock(L, head(X)), tail(X), z, d) \\
&\qquad\qquad\quad \textbf{else } \tau_{r'}.\ \mathsf{Miner^D}_U(L, tail(X)^\frown head(X), z, d) \\
&\qquad\quad \big) \mid\ w?(b).(\mathsf{Miner^D}_U(addBlock(L, b), X, z, b) \mid \textstyle\prod_{z' \in U} z'!(b)) \\
&\quad \Big)
\end{aligned}
$$

Theorem

Let P be a completed state of a blockchain system of n miners with exactly one that is hostile and let h_d its hashing power. The probability $\mathrm{Prob}(P_z)$ *to reach a completed state where the hostile miner has created an alternate chain longer than the honest one from* $z, z \geq 1$, *blocks behind is smaller than*

$$\sum_{k}^{+\infty}\Big[\Big(h_d\prod_{i=1}^{n-1}\frac{i\ r}{R+(n-i)r}\Big)^{k}\Big(\sum_{j=1}^{n-1}h_j\prod_{h=1}^{n-1}\frac{h\ r}{R+(n-i)r}\Big)^{k-1}\Big]^{z}\leq\Big(\frac{h_d}{1-h_d}\Big)^{z}$$

Conclusions

- We model the blockchain consensus protocol with a stochastic pi calculus
- Properties are derived by studying the transition system
- We computed the probability of devolving into a larger inconsistency and of a successfull attack
- It is possible to conform our upper bounds for both Bitcoin and Ethereum protocols, with instantiating the formula with the rate-values of the two systems
- We are currently extending a stochastic analyzer with the ledger datatype for experimenting in silico the dynamics of our specifications