Building a Performance Model of the *Tendermint* Consensus Algorithm

Jonas Vanden Branden - 2 feb 2018

Motivation

Tendermint is a (relatively) recent open-source blockchain consensus platform.

Experimental results, no (known) formal model or simulations yet.

Gather knowledge about building blocks: algorihtm

Save time, save resources.

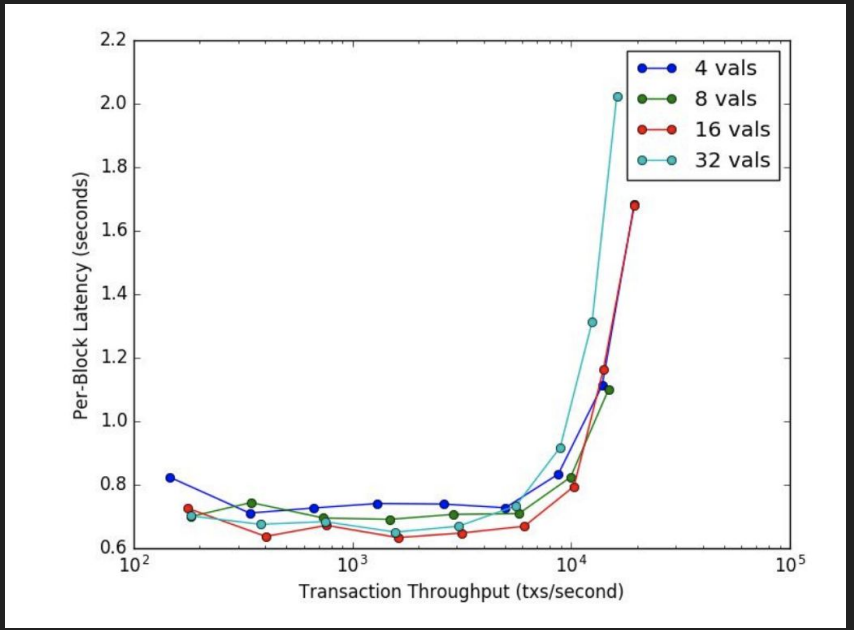

Fig. 1: Experimental results [1]

Redefinition of Paper topic

Previously: *“Modelling Read Cache Solutions for Blockchains “*

- Ambiguous
- Too general
- Very little related work

More focus on:

- Consensus algorithm
- Performance model (throughput & latency)

Which Formalism?

Stochastic Petri Nets?

+ Modeling activities
+ Control flow modeling
- Limited expressiveness [6]
- No hierarchy
- Complex
- No

-> Look for extensions

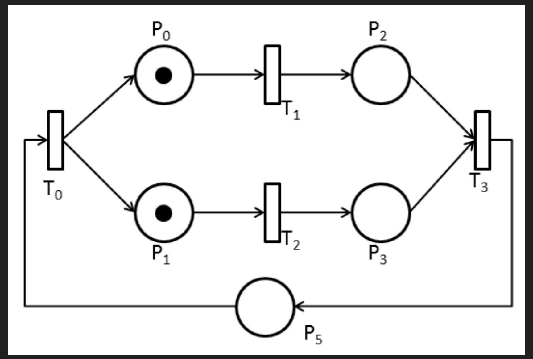

Fig. 2a: General SPN example

Which Formalism?

Possible candidate: Stochastic Reward Net

+ Related work (hyperledger)
+ More powerful than SPN

but...

- Limited expressiveness
- No hierarchy
- Complex

-> Other options?

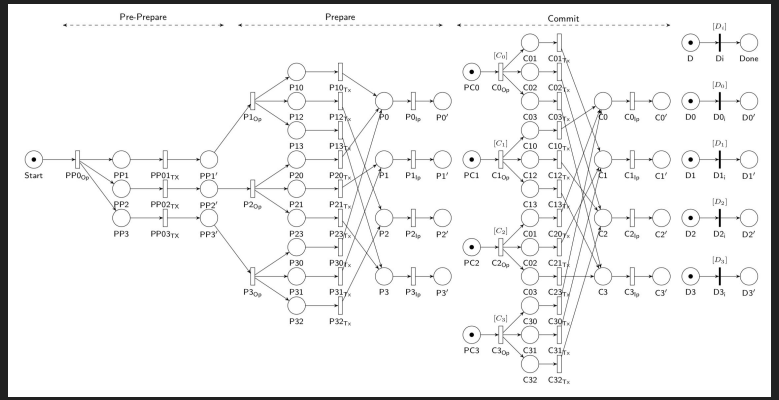

Fig. 2b: Hyperledger pBFT SRN Model [2]

Which Formalism?

Hierarchical Stochastic Activity Networks

+ Related work ([3])
+ More powerful than SRN
+ Hiërachical
+ Very expressive

Extension of SPN:

· Transition = activity
· Addition of (I/O)-gates

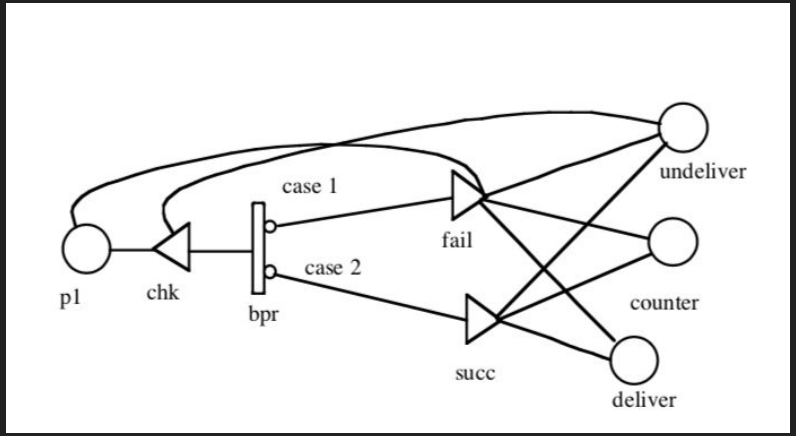

Fig. 3: Example SAN [4]

Tool used: Möbius

“Model-based environment for Validation of System Reliability, Availability, Security, and Performance”

- Developed by University of Illinois
- Free for educational use
- Supports SAN & Repl/Join formalism

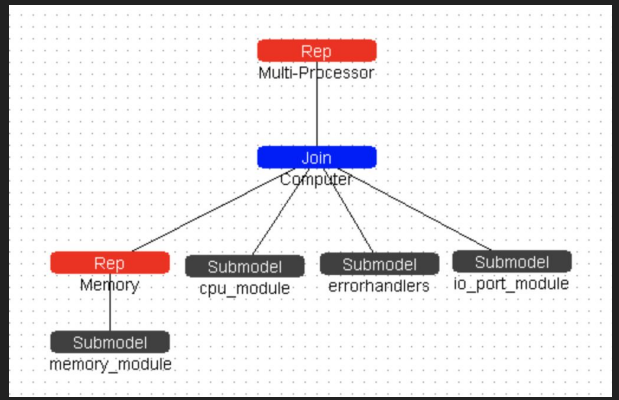

Fig. 4: Repl/Join example [5]

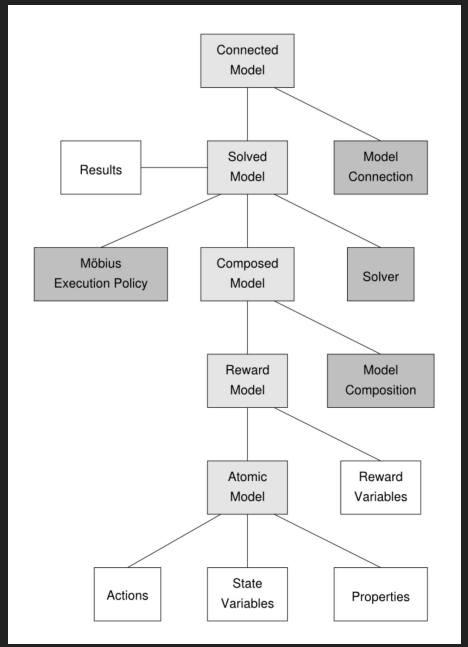

Fig. 5: Möbius Structure [5]

The Algorithm

General overview in state machine

Divided into steps:

- Propose
- Pre-vote
- Pre-commit
- Commit

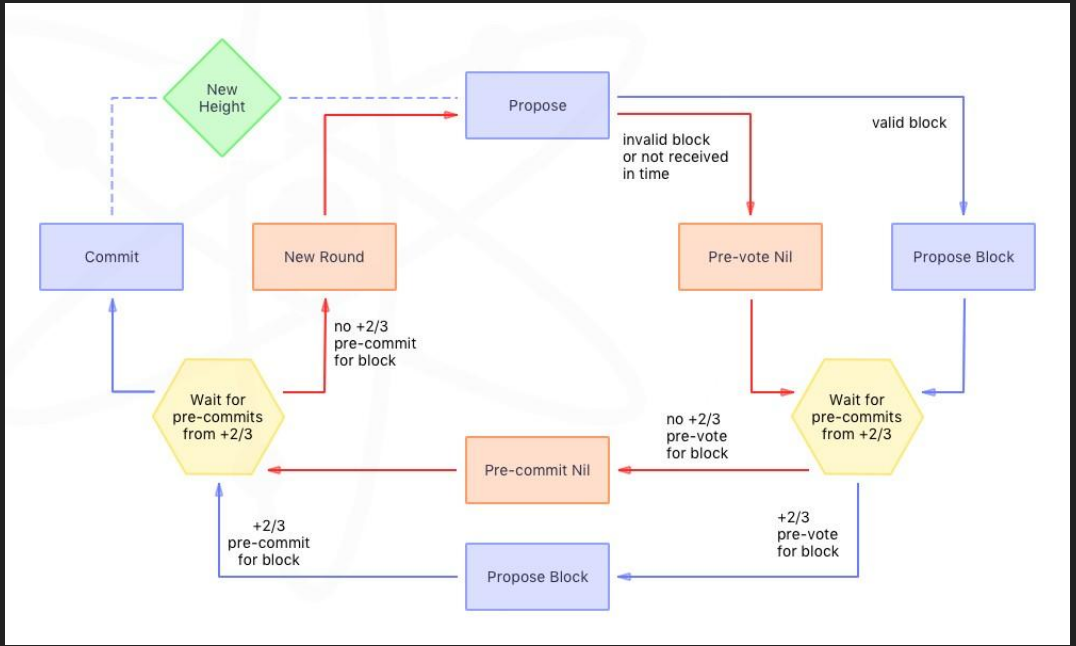

Fig. 6: Algorithm State Machine [6]

Variables: Round R, Height H

Deterministic round robin selection for designated proposer

Modeling Activity

Model each step in separate atomic SAN-model.

Flow logic:

- in the models' topology
- In IN/OUT-gates
- In custom code

Data

- Extended places
- Global variables

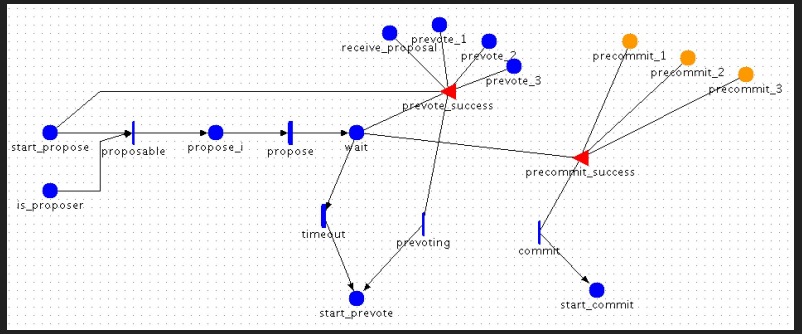

Fig. 7: Propose Step

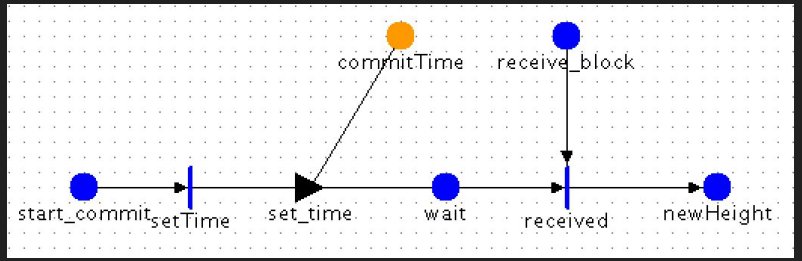

Fig. 8: Commit Step

Modeling Activity

Merge sub-models into composed models

Formalism: Repl / Join

Steps → Node
Nodes + Round Robin → ‘Network’

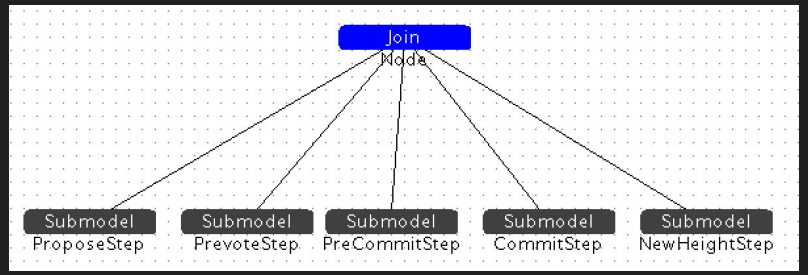

Fig. 9: Composed Node model

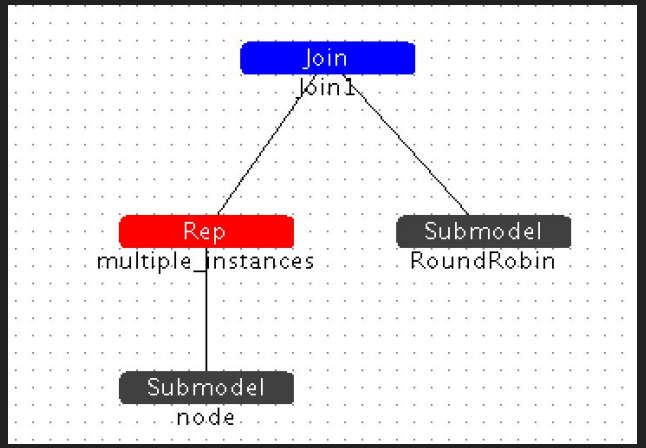

Fig. 10: Composed Network Model

Difficulties

- Algorithm description sometimes ambiguously described
- Network communication is implicit in description
- Technical limitations of the model & little knowledge

Future Work

- Work out the complete model for the algorithm
- Model more advanced network communication congestion, latency, ...
- Dynamic scaling of amount of nodes

References

[1] Ethan Buchman, ”On the Design and Accountability of Byzantine Fault Tolerant Protocols” - 2017

[2] Sukhwani, Harish, et al. ”Performance Modeling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger Fabric)” - 2017

[3] A. Schiper et al, ”Performance Analysis of a Consensus Algorithm Combining Stochastic Activity Networks and Measurements”, Universita di Firenze, 2002

[4] Andrea Mario Coccoli, ”On Integrating Modelling and Experiments in Dependability and Performability Evaluation of Distributed Applications.”, PhD thesis, University of Pisa, Italy, 2002

[5] Möbius wiki www.mobius.illinois.edu/wiki, University of Illinois, 2018

[6] *Nicole Sergent, “The performance of the consensus algorithm running on FDDI” - 1998*