Stochastic Models for blockchain analysis

Pierre-O. Goffard

Institut de Science Financières et d'Assurances
pierre-olivier.goffard@univ-lyon1.fr

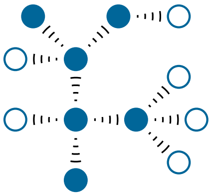
10 mai 2023

Blockchain

Introduction

A decentralized data ledger made of blocks maintained by achieving consensus in a P2P network.

- Decentralized
- Public/private
- Permissionned/permissionless
- Immutable
- Incentive compatible



Consensus protocol

Definition

Algorithm to allows the full nodes to agree on a common data history

It must rely on the scarce resources of the network

- bandwidth
- computational power
- storage (disk space)

Types of consensus protocols

1. Voting based

   L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, pp. 382–401, July 1982.

2. Leader based
   - Proof-of-Work (computational power)
   - Proof-of-Capacity and Proof-of-Spacetime (storage)
   - Proof-of-Interaction (bandwidth)
   - Proof-of-Stake (tokens)

Conflict resolution in blockchain

Fork

A fork arises when there is a disagreement between the nodes resulting in several branches in the blockchain.

LCR

The *Longest Chain Rule* states that if there exist several branches of the blockchain then the longest should be trusted.

In practice

- A branch can be considered legitimate if it is $k \in \mathbb{N}$ blocks ahead of its pursuers.
- Fork can be avoided when

$$\text{block appending time} > \text{ propagation delay}$$

Blockastics project

Stochastic models to assess

1. Efficiency (Queueing models)
   - Average number of transactions processed per time units
2. Decentralization (Stochastic process with reinforcement)
   - Distribution of the decision power accross the nodes
3. Security (Risk theory)
   - Resistance to attacks

. https://pierre-olivier.goffard.me/BLOCKASTICS/

What's inside a block ?

A block consists of

- a header
- a list of "transactions" that represents the information recorded through the blockchain.

The header usually includes

- the date and time of creation of the block,
- the block height which is the index inside the blockchain,
- the hash of the block
- the hash of the previous block.

Question

What is the hash of a block ?

Cryptographic Hash function

A function that maps data of arbitratry size (message) to a bit array of fixed size (hash value)

$$h : \{0,1\}^* \mapsto \{0,1\}^d.$$

A good hash function is

- deterministic
- quick to compute
- One way
  - ↪ For a given hash value $\overline{h}$ it is hard to find a message m such that

$$h(m) = \overline{h}$$

- Colision resistant
  - ↪ Impossible to find m_1 and m_2 such that

$$h(m_1) = h(m_2)$$

- Chaotic

$$m_1 \approx m_2 \Rightarrow h(m_1) \neq h(m_2)$$

SHA-256

The SHA-256 function which converts any message into a hash value of 256 bits.

Example

The hexadecimal digest of the message

`Blockastics is fantastic`

is

`60a147c28568dc925c347bce20c910ef90f3774e2501ac63344f3411b6a6bf79`

Mining a block


```
Block Hash: 1fc23a429aa5aaf04d17e9057e03371f59ac8823b1441798940837fa2e318aaa
Block Height: 0
Time:2022-02-25 12:42:04.560217
Nonce:0
Block data: [{'sender': 'Coinbase', 'recipient': 'Satoshi', 'amount': 100, 'fee': 0}, {'sen
der': 'Satoshi', 'recipient': 'Pierre-O', 'amount': 5, 'fee': 2}]
Previous block hash: 0
Mined: False
--------------
```

Figure – A block that has not been mined yet.

Mining a block

The maximum value for a 256 bits number is

$$T_{\max} = 2^{256} - 1 \approx 1.16e^{77}.$$

Mining consists in drawing at random a nonce

$$\text{Nonce} \sim \text{Unif}(\{0, \dots, 2^{32} - 1\}),$$

until

$$h(\text{Nonce}|\text{Block info}) < T,$$

where T is referred to as the target.

Difficulty of the cryptopuzzle

$$D = \frac{T_{\max}}{T}.$$

Mining a block

Examples of consensus protocol

If we set the difficulty to $D = 2^4$ then the hexadecimal digest must start with at least 1 leading 0

```
Block Hash: 0869032ad6b3e5b86a53f9dded5f7b09ab93b24cd5a79c1d8c81b0b3e748d226
Block Height: 0
Time:2022-02-25 13:41:48.039980
Nonce:2931734429
Block data: [{'sender': 'Coinbase', 'recipient': 'Satoshi', 'amount': 100, 'fee': 0}, {'sen
der': 'Satoshi', 'recipient': 'Pierre-O', 'amount': 5, 'fee': 2}]
Previous block hash: 0
Mined: True
--------------
```

Figure – A mined block with a hash value having on leading zero.

The number of trial is geometrically distributed

- Exponential inter-block times
- Lenght of the blockchain = Poisson process

Bitcoin protocol

- One block every 10 minutes on average
- Depends on the hashrate of the network
- Difficulty adjustment every 2,016 blocks ($\approx$ two weeks)
- Reward halving every 210,000 blocks

Check out https://www.bitcoinblockhalf.com/

Mining equipments

How it started

- CPU, GPU

How it is going

- Application Specific Integrated Chip (ASIC)
  - Increase of the network electricity consumption https://digiconomist.net/bitcoin-energy-consumption
  - E-Waste
  - Centralization issue https://www.bitmain.com/
    - Mining pool ranking at https://btc.com/
    - Mining equipment profitability at https://v3.antpool.com/minerIncomeRank

Proof of Stake

PoW is slow and ressource consuming. Let $\{1,\ldots,N\}$ be a set of miners and $\{\pi_1,\ldots,\pi_N\}$ be their share of cryptocoins.

PoS

1. Node $i \in \{1,\ldots,N\}$ is selected with probability π_i to append the next block

Nodes are chosen according to what they own.

- Nothing at stake problem
- Rich gets richer ?
- https://www.peercoin.net/

F. Saleh, "Blockchain without waste : Proof-of-stake," *The Review of Financial Studies*, vol. 34, pp. 1156–1190, jul 2020.

Using bandwidth

Proof-of-Interaction

- The node receives a list of node they must get in touch with
- The first one who is able to complete the task gets a reward and share it with the responding nodes

J.-P. Abegg, Q. Bramas, and T. Noël, "Blockchain using proof-of-interaction," in *Networked Systems*, pp. 129–143, Springer International Publishing, 2021.

For an up-to-date list of consensus protocol

https://tokens-economy.gitbook.io/consensus/

Double spending attack

1. Mary transfers 10 BTCs to John
2. The transaction is recorded in the public branch of the blockchain and John ships the good.
3. Mary transfers to herself the exact same BTCs
4. The malicious transaction is recorded into a private branch of the blockchain
   - Mary has friends among the miners to help her out
   - The two chains are copycat up to the one transaction

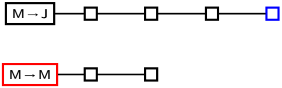
Fact (Bitcoin has only one rule)

The longest chain is to be trusted

Double spending in practice

Stochastic Models : Security of PoW blockchain

Vendor are advised to wait for $\alpha \in \mathbb{N}$ of confirmations so that the honest chain is ahead of the dishonest one.



In the example, vendor awaits $\alpha = 4$ confirmations, the honest chain is ahead of the dishonest one by $z = 2$ blocks.

Fact (PoW is resistant to double spending)

- Attacker does not own the majority of computing power
- Suitable α

Double spending is unlikely to succeed.

S. Nakamoto, "Bitcoin : A peer-to-peer electronic cash system." Available at https://bitcoin.org/bitcoin.pdf, 2008.

Mathematical set up

Assume that

- $R_0 = z \geq 1$ (the honest chain is z blocks ahead)
- at each time unit a block is created
  - $\hookrightarrow$ in the honest chain with probability p
  - $\hookrightarrow$ in the dishonest chain with probability $q = 1 - p$

The process $(R_n)_{n\geq 0}$ is a random walk on $\mathbb{Z}$ with

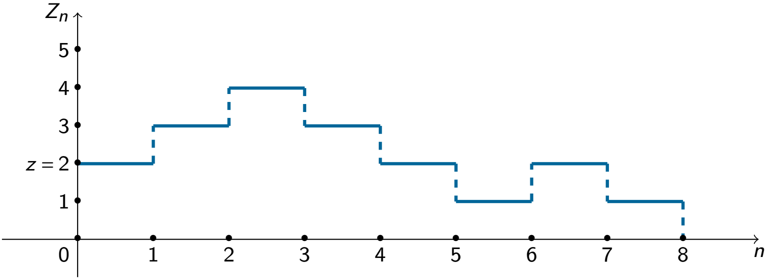
$$R_n = z + Y_1 + \ldots + Y_n,$$

where $Y_1, \ldots, Y_n$ are the **i.i.d.** steps of the random walk.

Double spending rate of success

Double spending occurs at time

$$\tau_z = \inf\{n \in \mathbb{N}\,;\; R_n = 0\}.$$



Double spending theorem

If $p > q$ then the double-spending probability is given by

$$\phi(z) = \mathbb{P}(\tau_z < \infty) = \left(\frac{q}{p}\right)^z.$$

Proof of the double spending theorem I

Analogy with the gambler's ruin problem. Using a first step analysis, we have

$$\phi(z) = p\phi(z+1) + (1-p)\phi(z-1), \; z \geq 1. \tag{1}$$

We also have the boundary conditions

$$\phi(0) = 1 \text{ and } \lim_{z\to+\infty} \phi(z) = 0 \tag{2}$$

Equation (1) is a linear difference equation of order 2 associated to the following characteristic equation

$$px^2 - x + 1 - p = 0$$

which has two roots on the real line with

$$r_1 = 1, \text{ and } r_2 = \frac{1-p}{p}.$$

The solution of (1) is given by

$$\phi(z) = A + B\left(\frac{1-p}{p}\right)^z,$$

Proof of the double spending theorem II

where A and B are constant. Using the boudary conditions (2), we deduce that

$$\phi(z) = \left(\frac{1-p}{p}\right)^{z}$$

as announced.

Refinements of the double spending problem

The number of blocks M found by the attacker until the honest miners find α blocks is a negative binomial random variable with **pmf**

$$\mathbb{P}(M=m)=\binom{\alpha+m-1}{m}p^{\alpha}q^{m},\ \ m\geq 0.$$

The number of block that the honest chain is ahead of the dishonest one is given by

$$Z=(\alpha-M)_{+}.$$

Applying the law of total probability yields the probability of successful double spending with

$$\mathbb{P}(\text{Double Spending})=\mathbb{P}(M\geq\alpha)+\sum_{m=0}^{\alpha-1}\binom{\alpha+m-1}{m}q^{\alpha}p^{m}.$$

M. Rosenfeld, "Analysis of hashrate-based double spending," *arXiv preprint arXiv :1402.2009*, 2014.

C. Grunspan and R. Perez-Marco, "Double spend race," *International Journal of Theoretical and Applied Finance*, vol. 21, p. 1850053, dec 2018.

Refinements of the double spending problem

Let the length of honest and dishonest chain be driven by counting processes

- Honest chain $\Rightarrow z + N_t$, $t \geq 0$, where $z \geq 1$.
- Malicious chain $\Rightarrow M_t$, $t \geq 0$
- Study the distribution of the first-*rendez-vous* time

$$\tau_z = \inf\{t \geq 0\ ,\ M_t = z + N_t\}.$$

If $N_t \sim \text{Pois}(\lambda t)$ and $M_t \sim \text{Pois}(\mu t)$ such that $\lambda > \mu$ then

$$\phi(z) = \left(\frac{\mu}{\lambda}\right)^z,\ z \geq 0.$$

P.-O. Goffard, "Fraud risk assessment within blockchain transactions," *Advances in Applied Probability*, vol. 51, pp. 443–467, jun 2019. https://hal.archives-ouvertes.fr/hal-01716687v2.

R. Bowden, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor, "Modeling and analysis of block arrival times in the bitcoin blockchain," *Stochastic Models*, vol. 36, pp. 602–637, jul 2020.

Perspectives

Stochastic Models : Security of PoW blockchain

- Include network delay

  A. Dembo, S. Kannan, E. N. Tas, D. Tse, P. Viswanath, X. Wang, and O. Zeitouni, "Everything is a race and nakamoto always wins," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ACM, oct 2020.

- Double spending in block-DAGS

  E. Anceaume, A. Guellier, R. Ludinard, and B. Sericola, "Sycomore : A permissionless distributed ledger that self-adapts to transactions demand," in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, IEEE, nov 2018.

Proof of Stake protocol

PoS is the most popular alternative to PoW.

- A block validator is selected according to the number of native coins she owns
- Update the blockchain and receive a reward or do nothing

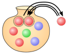
Two problems

- ⚠ Nothing at stake $\Rightarrow$ Consensus postponed
- ⚠ Rich gets richer $\Rightarrow$ Risk of centralization

Risk of centralization ?

Block appending process

- Draw a coin at random
- The owner of the coin append a block and collect the reward
- The block appender is more likely to get selected during the next round

Similar to Polya's urn 

- Consider an urn of N balls of color in $E = \{1, \dots, p\}$
- Draw a ball of color $x \in E$
- Replace the ball together with r balls of color x

p is the number of peers and r is the size of the block reward.

Theorem

The proportion of coins owned by each peer is stable on average over the long run

I. Roșu and F. Saleh, "Evolution of shares in a proof-of-stake cryptocurrency," *Management Science*, vol. 67, pp. 661–672, feb 2021.

Proof

Consider the balls of some color $x \in E$, and denote by

- N_x the number of balls of color x initially in the urn
- Y_n the number of balls of color x in the urn after n draws
- Z_n the corresponding proportion of balls of color x.

We show that $(Z_n)_{n\geq 0}$ is a $\mathscr{F}_n-$Martingale where $\mathscr{F}_n = \sigma(Y_1,\ldots,Y_n)$. We have

$$\mathbb{E}(Z_{n+1}|\mathscr{F}_n) = Z_n \frac{Y_n + r}{N + r(n+1)} + (1 - Z_n)\frac{Y_n}{N + r(n+1)} = Z_n$$

It follows that

$$\mathbb{E}(Z_n) = \mathbb{E}(Z_0) = \frac{N_x}{N}, \text{ for } n \geq 0.$$

hence the stability. Furthermore, because $|Z_n| < 1$, then $\lim_{n\to\infty} Z_n = Z_\infty$ exists and it holds that $\mathbb{E}(Z_\infty) = \mathbb{E}(Z_0)$.

What is the limiting distributions of the shares ?

Dirichlet distribution

A random vector $(Z_1,\dots,Z_p)$ has a Dirichlet distribution $\text{Dir}(\alpha_1,\dots,\alpha_p)$ with **pdf**

$$f(z_1,\dots,z_p;\alpha_1,\dots,\alpha_p) = \frac{1}{B(\alpha)}\prod_{i=1}^{p} z_i^{\alpha_i-1},$$

for $\alpha_1,\dots,\alpha_p > 0$, $0 < z_1,\dots,z_p < 1$ and $\sum_{i=1}^{p} z_i = 1$, where

$$B(\alpha) = \frac{\prod_{i=1}^{p}\Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{p}\alpha_i)}.$$

Theorem (Convergence toward a Dirichlet distribution)

Suppose that $r = 1$ and let X_n be the color of the ball drawn at the n^{th} round then

$$\{\mathbb{P}(X_\infty = x), x \in E\} \sim \text{Dir}(\{N_x,\ x \in E\}).$$

Extensions and perspectives

- How to include more peers along the way ?
- What if the peers are not simply buy and hold investors ?
- Find ways to monitor decentralization and take action if necessary

I. Roșu and F. Saleh, "Evolution of shares in a proof-of-stake cryptocurrency," *Management Science*, vol. 67, pp. 661–672, feb 2021.

Efficiency

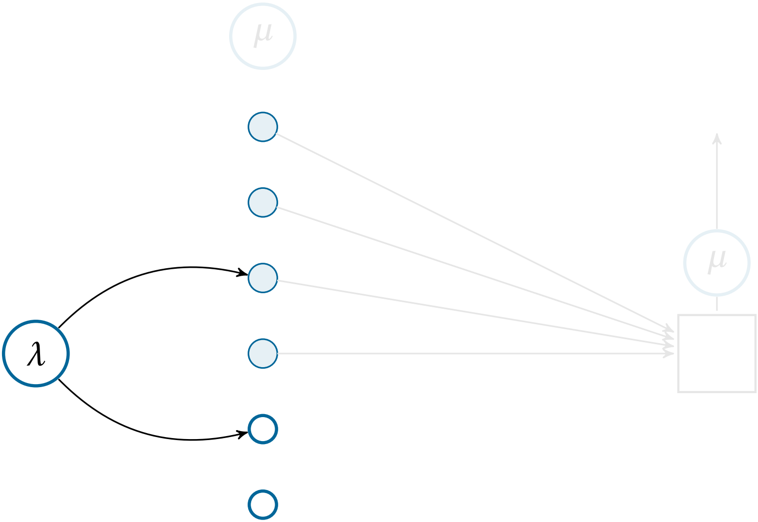
Efficiency is characterized by

- Throughputs : Number of transaction being processed per time unit
- Latency : Average transaction confirmation time

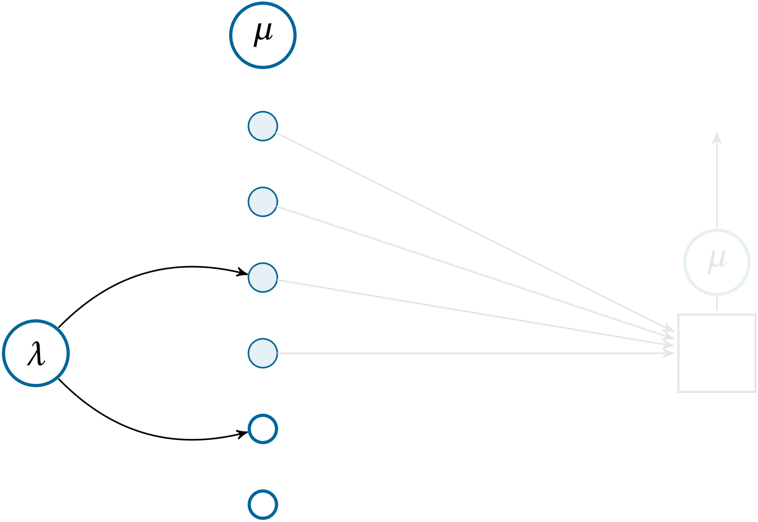
We focus on a PoW equipped blockchain and study the above quantities using a queueing model.

Queue settings

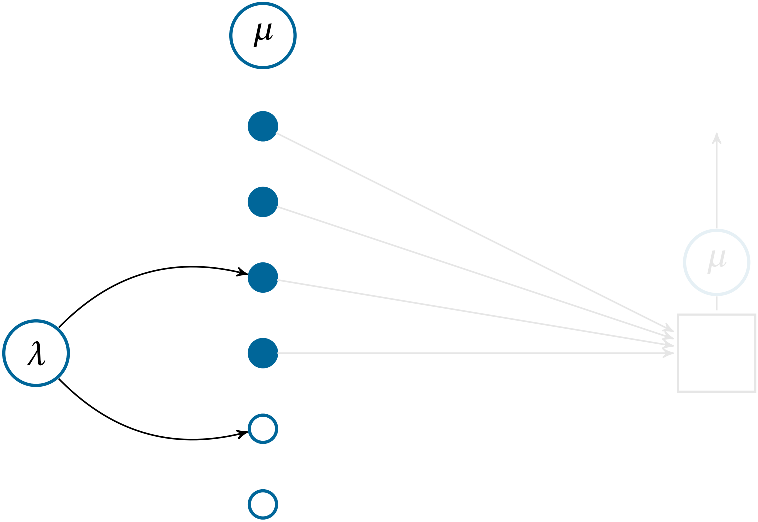
Stochastic Models : Blockchain efficiency



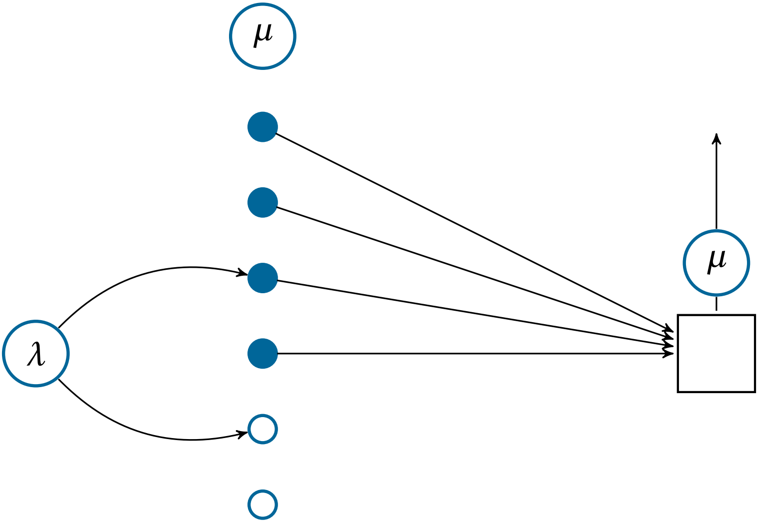
Queue settings



Queue settings



Queue settings



Queueing setting

- Poisson arrival with rate $\lambda > 0$ for the transactions
- Poisson arrival with rate $\mu > 0$ for the blocks
- Block size $b \in \mathbb{N}^*$ $\Rightarrow$Batch service

⚠ The server is always busy

This is somekind of $M/M^b/1$ queue.

Y. Kawase and S. Kasahara, "Transaction-confirmation time for bitcoin : A queueing analytical approach to blockchain mechanism," in *Queueing Theory and Network Applications*, pp. 75–88, Springer International Publishing, 2017.

N. T. J. Bailey, "On queueing processes with bulk service," *Journal of the Royal Statistical Society : Series B (Methodological)*, vol. 16, pp. 80–87, jan 1954.

D. R. Cox, "The analysis of non-markovian stochastic processes by the inclusion of supplementary variables," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, pp. 433–441, jul 1955.

Queue length distribution

The queueing process eventually reaches stationarity if

$$\mu \cdot b > \lambda. \tag{3}$$

We denote by N^q the length of the queue upon stationarity.

The blockchain efficiency theorem

Assume that (3) holds then N^q is geometrically distributed

$$\mathbb{P}(N^q = n) = (1-p)\cdot p^n,$$

where $p = 1/z^*$ and z^* is the only root of

$$-\frac{\lambda}{\mu}z^{b+1} + z^b\left(\frac{\lambda}{\mu}+1\right) - 1,$$

such that $|z^*| > 1$.

Proof of the efficiency theorem I

Let N_t^q be the number of transactions in the queue at time $t \geq 0$ and X_t the time elapsed since the last block was found. Further define

$$P_n(x,t)\mathrm{d}x = \mathbb{P}[N_t^q = n, X_t \in (x, x+\mathrm{d}x)]$$

If $\lambda < \mu \cdot b$ holds then the process admits a limiting distribution given by

$$\lim_{t\to\infty} P_n(x,t) = P_n(x).$$

We aim at finding the distribution of the queue length upon stationarity

$$\mathbb{P}(N^q = n) := \alpha_n = \int_0^\infty P_n(x)\mathrm{d}x. \tag{4}$$

Consider the possible transitions over a small time lapse h during which no block is being generated. Over this time interval, either

- no transactions arrives
- one transaction arrives

Proof of the efficiency theorem II

We have for $n \geq 1$

$$P_n(x+h) = e^{-\mu h}\left[e^{-\lambda h}P_n(x) + \lambda h e^{-\lambda h}P_{n-1}(x)\right]$$

Differentiating with respect to h and letting $h \to 0$ leads to

$$P_n'(x) = -(\lambda+\mu)P_n(x) + \lambda P_{n-1}(x),\ n \geq 1. \tag{5}$$

Similarly for $n = 0$, we have

$$P_0'(x) = -(\lambda+\mu)P_0(x). \tag{6}$$

We denote by

$$\xi(x)\mathrm{d}x = \mathbb{P}(x \leq X < x + \mathrm{d}x | X \geq x) = \mu \mathrm{d}x$$

the hazard function of the block arrival time (constant as it is exponentially distributed). The system of differential equations (5), (6) admits boundary conditions at $x = 0$ with

$$\begin{cases} P_n(0) = \int_0^{+\infty} P_{n+b}(x)\xi(x)\mathrm{d}x = \mu\alpha_{n+b}, & n \geq 1, \\ P_0(0) = \mu\sum_{n=0}^{b}\alpha_n, & n = 0,\dots,b \end{cases} \tag{7}$$

Proof of the efficiency theorem III

Define the probability generating function of N^q at some elapsed service time $x \geq 0$ as

$$G(z;x) = \sum_{n=0}^{\infty} P_n(x) z^n.$$

By differentiating with respect to x, we get (using (5) and (6))

$$\frac{\partial}{\partial x} G(z;x) = -\left[\lambda(1-z)+\mu\right] G(z;x)$$

and therefore

$$G(z;x) = G(z;0) \exp\left\{-\left[\lambda(1-z)+\mu\right] x\right\}$$

We get the probability generating function of N^q by integrating over x as

$$G(z) = \frac{G(z;0)}{\lambda(1-z)+\mu} \tag{8}$$

Proof of the efficiency theorem IV

Using the boundary conditions (7), we write

$$
\begin{aligned}
G(z;0) &= \sum_{n=0}^{\infty} P_n(0)z^n \\
&= P_0(0) + \sum_{n=1}^{+\infty} P_n(0)z^n \\
&= \mu\sum_{n=0}^{b}\alpha_n + \mu\sum_{n=1}^{+\infty}\alpha_{n+b}z^n \\
&= \mu\sum_{n=0}^{b}\alpha_n + \mu z^{-b}\left[G(z) - \sum_{n=0}^{b}\alpha_n z^n\right] \qquad (9)
\end{aligned}
$$

Replacing the left hand side of (9) by (8), multiplying on both side by z^b and rearranging yields

$$
\frac{G(z)}{M(z)}[z^b - M(z)] = \sum_{n=0}^{b-1}\alpha_n(z^b - z^n), \qquad (10)
$$

where $M(z) = \mu/(\lambda(1-z)+\mu)$. Using Rouche's theorem, we find that both side of the equation shares b zeros inside the circle $\mathscr{C} = \{z \in \mathbb{C}\,;\, |z| < 1+\epsilon\}$ for some epsilon.

Proof of the efficiency theorem V

Rouche's theorem

Let $\mathscr{C} \in \mathbb{C}$ and f and g two holomorphic functions on $\mathscr{C}$. Let $\partial\mathscr{C}$ be the contour of $\partial\mathscr{C}$. If

$$|f(z) - g(z)| < |g(z)|,\ \forall z \in \partial\mathscr{C}$$

then $Z_f - P_f = Z_g - P_g$, where Z_f, P_f, Z_g, and P_g are the number of zeros and poles of f and g respectively.

We have $\partial\mathscr{C} = \{z \in \mathbb{C}\,;\ |z| = 1+\epsilon\}$. The left hand side can be rewritten as

$$G(z)\left[-\frac{\lambda}{\mu}z^{b+1} + \left(1+\frac{\lambda}{\mu}\right)z^b - 1\right].$$

Define $f(z) = -\frac{\lambda}{\mu}z^{b+1} + \left(1+\frac{\lambda}{\mu}\right)z^b - 1$ and $g(z) = \left(1+\frac{\lambda}{\mu}\right)z^b$. We have

$$|f(z) - g(z)| = |-\frac{\lambda}{\mu}z^{b+1} - 1| < \frac{\lambda}{\mu}(1+\epsilon)^{b+1} + 1 \to \frac{\lambda}{\mu} + 1, \text{ as } \epsilon \to 0.$$

Proof of the efficiency theorem VI

and

$$|g(z)| = \left(1 + \frac{\lambda}{\mu}\right)(1+\epsilon)^b \to \frac{\lambda}{\mu} + 1, \text{ as } \epsilon \to 0.$$

Regarding the right hand side, define $f(z) = \sum_{n=0}^{b-1} \alpha_n (z^b - z^n)$ and $g(z) = \sum_{n=0}^{b-1} \alpha_n z^b$. We have

$$|f(z) - g(z)| < |\sum_{n=0}^{b-1} \alpha_n z^n| < \sum_{n=0}^{b-1} \alpha_n (1+\epsilon)^n \to \sum_{n=0}^{b-1} \alpha_n, \text{ as } \epsilon \to 0.$$

and

$$|g(z)| = (1+\epsilon)^b \sum_{n=0}^{b-1} \alpha_n \to \sum_{n=0}^{b-1} \alpha_n, \text{ as } \epsilon \to 0.$$

One of them is 1, and we denote by z_k, $k = 1, \ldots, b-1$ the remaining $b-1$ zeros. Given the polynomial form of the right hand side of (10), the fundamental theorem of algebra indicates that the number of zero is b. Given the left hand side

$$G(z)\left[-\frac{\lambda}{\mu} z^{b+1} + \left(1 + \frac{\lambda}{\mu}\right) z^b - 1\right].$$

Proof of the efficiency theorem VII

we deduce that there is one zeros outside $\mathscr{C}$, we can further show that it is a real number z^*. Multiplying both side of (10) by $(z-1)\prod_{k=1}^{b-1}(z-z_k)$, and using $G(1)=1$ yields

$$G(z) = \frac{1-z^*}{z-z^*}.$$

N^q is then a geometric random variable with parameter $p = \frac{1}{z^*}$.

Latency and throughputs

Little's law

Consider a stationary queueing system and denote by

- $1/\lambda$ the mean of the unit inter-arrival times
- L be the mean number of units in the system
- W be the mean time spent by units in the system

We have

$$L = \lambda \cdot W$$

J. D. C. Little, "A proof for the queuing formula :L= λW," *Operations Research*, vol. 9, pp. 383–387, jun 1961.

- Latency is the confirmation time of a transaction

$$\text{Latency} = \frac{p}{(1-p)\lambda} + \frac{1}{\mu}$$

- Throughput is the number of transaction confirmed per time unit

$$\text{Throughput} = \mu \mathbb{E}(N^q \mathbb{I}_{N^q \leq b} + b\mathbb{I}_{N^q > b}) = \mu \sum_{n=0}^{b} n(1-p)p^n + bp^{b+1}.$$

Perspective

1. Include some priority consideration to account for the transaction fees

Y. Kawase, , and S. Kasahara, "Priority queueing analysis of transaction-confirmation time for bitcoin," *Journal of Industrial & Management Optimization*, vol. 16, no. 3, pp. 1077–1098, 2020.

2. Go beyond the Poisson process framework

Q.-L. Li, J.-Y. Ma, and Y.-X. Chang, "Blockchain queue theory," in *Computational Data and Social Networks*, pp. 25–40, Springer International Publishing, 2018.

Q.-L. Li, J.-Y. Ma, Y.-X. Chang, F.-Q. Ma, and H.-B. Yu, "Markov processes in blockchain systems," *Computational Social Networks*, vol. 6, jul 2019.

A. Dembo, S. Kannan, E. N. Tas, D. Tse, P. Viswanath, X. Wang, and O. Zeitouni, "Everything is a race and nakamoto always wins," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ACM, oct 2020.

E. Anceaume, A. Guellier, R. Ludinard, and B. Sericola, "Sycomore : A permissionless distributed ledger that self-adapts to transactions demand," in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, IEEE, nov 2018.

Two generals problem

Two nodes who must agree are communicating through an unreliable link.

- Analogy with two generals besieging a city

The generals exchange messages through enemy territory

G1

`"I will attack tomorrow at dawn, if you confirm"`

G2

`"I will follow your lead, if you confirm"`

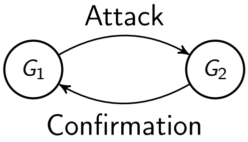


Figure – Message and confirmation loop

Byzantine General problem

n generals must agree on a common battle plan, to either

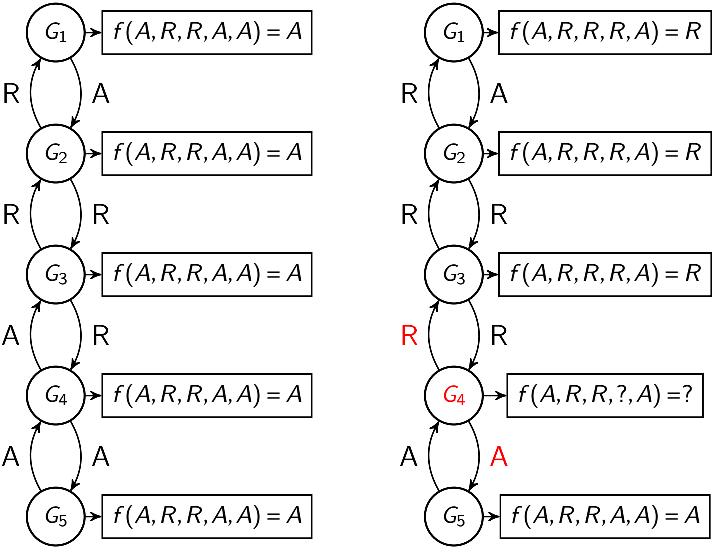
- Attack (A)
- Retreat (R)

Problem

There are $m < n$ traitors among the generals

1. message $m(i,j)$ is sent to general j by general i
2. Consensus is reached as general j applies

$$f(\{m(i,j);\ i = 1,\dots,n\}) = \begin{cases} A, & \text{if } \sum_{i=1}^{n} \mathbb{I}_{m(i,j)=A} > n/2, \\ R, & \text{else.} \end{cases}$$



(a) No traitor

(b) One traitor

Figure – Majority vote with or without a traitor

Commanders and Lieutenants

One general is the commander while the others are the lieutenants

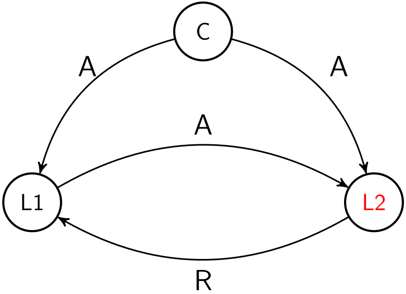
Objective

Design an algorithm so that the following conditions are met :

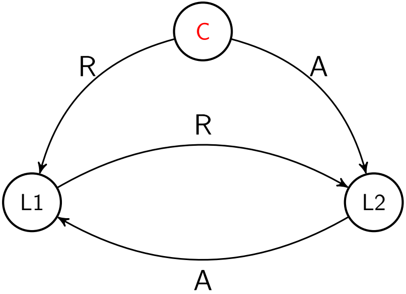
- C1 All the loyal lieutenants obey the same order
- C2 If the commanding general is loyal, then every loyal lieutenants obey the order he sends

Byzantine Fault Tolerance Theorem (Lamport et al.)

There are no solution to the Byzantine General problem for $n < 3m+1$ generals, where m is the number of traitors.



(a) Commander is loyal



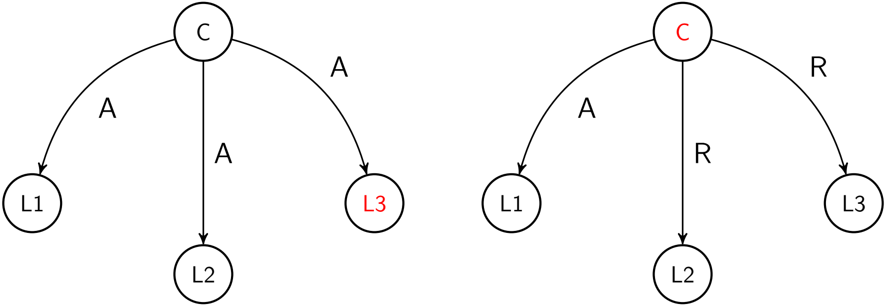
(b) Commander is a traitor

Figure – Majority vote with or without a traitor

Algorithm The Oral message algorithm OM(m)

```
if m = 0 then ;
    for i = 1 → n−1 do
        Commander sends v_i = v to lieutenant i
        Lieutenant i set their value to v
    end for
end if
if m > 0 then ;
    for i = 1 → n−1 do
        Commander sends v_i to lieutenant i
        Lieutenant i uses OM(m-1) to communicate v_i to the n−2 lieutenants
    end for
    for i = 1 → n−1 do
        Lieutenant i set their value to f(v_1,...,v_{n−1})
    end for
end if
```

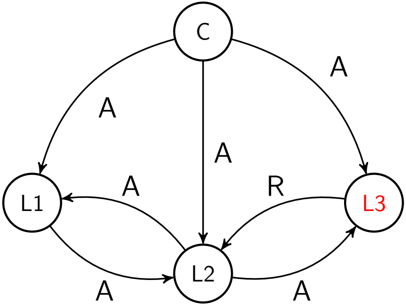
$n = 4$ and $m = 1$: Step 1



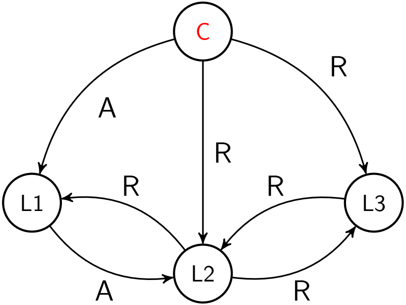
(a) Commander is loyal (b) Commander is a traitor

Figure – Illustration of the OM(m) algorithm in the case where $n = 4$ and $m = 1$.

$n = 4$ and $m = 1$: Step 2



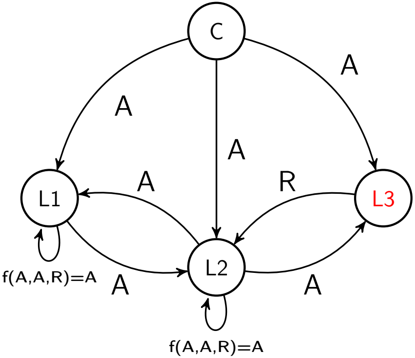
(a) Commander is loyal



(b) Commander is a traitor

Figure – Illustration of the OM(m) algorithm in the case where $n = 4$ and $m = 1$.

$n = 4$ and $m = 1$: Step 3



(a) Commander is loyal, C1 and C2

(b) Commander is a traitor, C1

Figure – Illustration of the OM(m) algorithm in the case where $n = 4$ and $m = 1$.

The problem with majority vote

The OM algorithm requires to send n^{m+1}

- ⚠ Communication overhead
- ⚠ Denial of service

Solution

Leader based protocols !

Proof-of-Work

Objective

Elect a leader based on computational effort to append the next block.