Delay analysis of a mempool-based blockchain protocol under asymptotic hypothesis

Khouloud Hwerbi, Ichrak Amdouni, Cédric Adjih, Philippe Jacquet, Leila Azouz Saidane, Anis Laouiti

Delay Analysis of a Mempool-based Blockchain Protocol Under Asymptotic Hypothesis

Khouloud Hwerbi * ‡, Ichrak Amdouni *, Cedric Adjih §, Philippe Jacquet §, Leila Azouz Saidane *, Anis Laouiti ‡
*ENSI, Tunisia, University of Manouba, name.surname@ensi-uma.tn
‡Telecom SudParis, France, khouloud_hwerbi@telecom-sudparis.eu, anis.laouiti@telecom-sudparis.eu
§Inria, France, name.surname@inria.fr

***Abstract*—Delays in blockchain networks are mainly related to consensus protocols. Among these protocols, we focus on a specific family of protocols, where the mempool's role in the consensus mechanism is explicitly examined. A mempool is a temporary storage area for transactions waiting to be included in a block. This study investigates the round duration of two mempool-based protocols: one requiring a single quorum of messages and another demanding two. We perform the delay analysis with two approaches. First, we elaborate on a Markov chain to determine the distribution of the round durations. Second, we establish an analytical model of message delays while assuming an exponential distribution of message propagation delays. Finally, asymptotic analysis is conducted to estimate the time of quorum formation. We end the paper by comparing the simulation results with the theoretical ones. Results show that both results are very close. This research offers valuable insights into the performance characteristics of mempool-based consensus protocols, aiding in the design and optimization of blockchain systems.**

***Index Terms*—Blockchain, Delay Analysis, Performance Evaluation, Markov Chain, Asymptotic hypothesis, Exponential Distribution.**

I. Introduction

Since Satoshi Nakamoto introduced Bitcoin in 2008 [1], the development of cryptocurrencies has accelerated. Blockchain, the core technology behind Bitcoin, has attracted considerable attention [2]. Operating on a decentralized, peer-to-peer network, blockchain is secured through cryptographic functions and governed by a consensus algorithm. It also supports self-executing code via smart contracts. As its name suggests, it consists of a sequence of linked transactions, or blocks, connected by hash functions forming a ledger. This ledger is shared among all peers on the network, making blockchain versatile and significant for various applications and research. Blockchain technology has been utilized in many fields like managing medical information, security services, smart agriculture, and smart home applications [3], [4], [5], [6], [7]. The adoption of blockchain technology in the industries mentioned requires a thorough understanding of its performance characteristics. Evaluating the performance of blockchain systems is crucial to identifying their strengths and limitations in real-world applications. Performance analysis involves assessing several key aspects, including efficiency, scalability, and security. By examining these factors, researchers and developers can optimize blockchain solutions to satisfy the requirements of different use cases better. To address performance issues, we need to provide mathematical modeling and analysis using tools such as Markov processes [8], Markov decision processes, queuing networks, Petri nets, and game theory models. On the other hand, it is also worth considering benchmarking and simulation approaches [9]. In this article, we tackle the challenge of analyzing blockchain delays, focusing on a specific blockchain family which is based on the mempool. A mempool, short for "memory pool", is an essential component in blockchain networks. It is a temporarily storage area for transactions that have been broadcast to the network but have not yet been included in a block. The mempool protocol is crucial for the efficiency of a blockchain network, ensuring that transactions are properly validated, prioritized (if needed), and included in blocks. By understanding and optimizing the mempool protocol, blockchain networks can improve their performance and reliability. As an example of mempool-based protocol, we can cite Narwhal [10].

Unfortunately, only a little work has tackled the performance analysis of blockchain protocols. Therefore, our objective is to evaluate the performance of a mempool-based protocol using different techniques and models: Markov processes, queueing, and analytical models. In this research work, we study a mempool protocol that works as follows. It exchanges blocks and messages, builds the mempool, proceeds in rounds and moves from one round to the next one upon receiving one or more quorum of messages. In particular, we consider two cases: (i) a protocol where nodes wait for a quorum of acknowledgments and (ii) a protocol where nodes wait for two quorums. In the first case, any node broadcasts a block of transactions and waits to receive a quorum of q acknowledgments or signatures from the other nodes before moving to the next round. In the second case, a node must wait to receive two quorums. This means an initial quorum is required to propose a block, followed by another quorum to confirm it. The second case is more consistent and reliable, as it ensures a higher level of agreement among the nodes, reducing the risk of conflicting blocks. Although this approach introduces additional communication overhead and potential delays, it significantly enhances the integrity and robustness of the system.

The contributions of this paper are as follows: (1) Our analysis is general: it applies to different blockchain protocols based on the mempool while considering two different

quorum configurations: a protocol with a single quorum of messages, and a protocol with two quorums of messages, (2) We use different modeling approaches : (i) Markov chains for the first and second families, (ii) analytical and asymptotic analysis assuming exponential distribution for the propagation delays for the second family, (3) We validate the proposed mathematical models using simulations.

II. State of the Art

In this section, we will discuss the blockchain performance evaluation methods and categorize the reviewed solutions into three categories.

A. Empirical Analysis of Blockchain Performance

From the perspective of empirical analysis, blockchain performance evaluation approaches consist of benchmarking, monitoring measurements, self-designed experiments, and simulations. In practice, these approaches are usually used to provide more evidence for evaluation of blockchain performance. Hyperledger Caliper [11] is a performance evaluation framework that mainly benchmarks Hyperledger blockchains (Fabric, Sawtooth, Iroha, Burrow, and Besu). The system architecture includes two main components: (1) the Caliper core which handles the main functionality and workflow of the performance evaluation, and (2) the Caliper adaptors which allow this framework to be flexible and extensible to other blockchain systems. DAGbench [12] is a performance evaluation framework designed specifically for benchmarking Directed Acyclic Graph (DAG) based blockchains such as IOTA, Nano, and Byteball. The framework evaluates various performance indicators including throughput, latency, scalability, resource consumption, transaction data size, and transaction fees. Authors in [13] proposed BlockSim, a framework that simulates blockchain systems based on the Proof of Work (PoW) consensus protocol. In this framework, block creation performance is analyzed. The simulator was validated by comparing its results with real-life systems.

B. Analytical Analysis of Blockchain Performance

Blockchain systems performance modeling was also investigated using analytical modeling such as Markov chains, queuing models, and Petri nets. Authors in [14] derive the average response time of transactions by regarding some key factors such as the transaction initiation, the mining process, the block generation, the blockchain-building, the block size, and the block verification. They model the system using queuing theory with batch and gated services. Authors in [8] develop a general framework to study blockchain systems by establishing a continuous-time Markov process of $GI/M/1$ type. In addition, they provided the average transaction–confirmation time in a general blockchain system.

Stochastic Petri net (SPN) models are also analytical tools used to model blockchain systems. Authors in [15] proposed a Generalised Stochastic Petri Nets (GSPN) model to analyze the performance of Hyperledger Fabric-based blockchains. Using this model, they investigated how different ordering strategies affect system performance and identified performance bottlenecks. The results show that the number of transactions in a block significantly impacts system performance.

C. Other Performance Analysis of Blockchain Systems

The authors of [16] have used game theory to extend Markov decision processes by considering competition between several rational decision-makers for the prediction of their actions in interactive situations. These game-theoretic models have been commonly applied to examine different aspects of the blockchain mining process. The game-theoretic analysis identified five types of Nash equilibrium. In addition, the results showed that excessively long waiting times can lead to negative marginal benefits on transaction fees for some users with low time costs, making them reluctant to offer transaction fees. Authors in [17] presented a framework in 2019 that combines machine learning and queuing theory to analyze Bitcoin transaction confirmation times. The integration of machine learning allows for identifying transactions that will be confirmed and queuing theory enables the characterization of their confirmation times.

III. Basics of the Studied Mempool-based Protocol

A. System Model

The mempool protocol studied operates within a network of n nodes, where $n = 3f + 1$ and f represents the number of Byzantine nodes. A round r is a designated period or phase in the protocol for each node i to propose one block of transactions $B_{i,r}$. When a node broadcasts such a block, it waits to receive a quorum of $q = 2f + 1$ block acknowledgments $A_b^{j,i}$ from other nodes j to form a quorum $Q_{i,r}^{(1)}$ before it can broadcast a new block of transactions in the next round $r + 1$. Each block is associated with a specific round number, ensuring that a node can propose only one block per round. The blocks broadcast by the nodes form a Directed Acyclic Graph (DAG), where each vertex represents a block. This structure ensures that the protocol can handle Byzantine faults while maintaining a consistent and verifiable order of transactions.

B. Notations and Assumptions

Table I summarizes the notations used in our analysis.

TABLE I
Notation Used in the Models Establishment

$B_{i,r}$	A block B broadcast by node i in the round r.
$A_b^{j,i}$	A block acknowledgment sent from node j to node i.
$A_q^{j,i}$	A quorum acknowledgment sent from node j to node i.
$Q_{i,r}^{(1)}$	A quorum of q block acknowledgment A_b received by node i.
$Q_{i,r}^{(2)}$	A quorum of q quorum acknowledgment A_q received by node i.
$X_{i,r}(t)$	The number of messages received by node i in round r.
$P_i(t)$	The probability of being in state i at time t.

We consider the following assumptions:

A1. A message transmission time is exponentially distributed with mean μ and rate parameter λ.

A2. All nodes in the network have equal capacity and computational power.
A3. Nodes operate over a fully connected peer-to-peer (P2P) network in which they are directly connected pairwise (i.e., each node is directly connected to $n-1$ nodes).
A4. There is no loss for both broadcast and unicast of messages transmission.

IV. Delay Analysis of a Mempool Protocol Based on One Quorum of Messages

We are interested in modeling the described protocol from the point of view of validator (acting as a miner) node V_i, denoted i as well. The protocol starts when i broadcasts a message $B_{i,r}$ in the round r to the network. Then, it waits for a quorum $Q_{i,r}^{(1)}$ of block acknowledgment $A_b^{j,i}$ from other nodes j. Fig.1 describes the defined protocol.

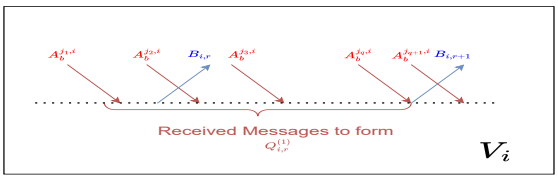

Fig. 1. Exchanged Messages from the point of view of validator V_i

We will model this mempool protocol with one quorum using a Continuous-Time Markov Chain (CTMC). The CTMC framework is particularly suitable for modeling the stochastic nature of the system, capturing the random arrival and processing of transactions. Its transition rate diagram, depicted in Fig. 2, illustrates the various states and the rates at which transitions occur between these states. Here, λ represents the average message arrival rate, which determines how frequently new messages are proposed by nodes j.

Let $X_{i,r}(t)$ be the number of messages (i.e. block acknowldgement $A_b^{j,i}$) received by i, according to assumption A1, $\{X_{i,r}(t)\}_{t\in\mathbb{R}_+}$ be a continuous-time Markov chain.

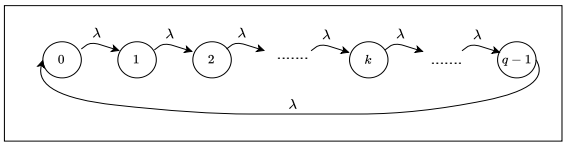

Fig. 2. Markov Chain for the Described Protocol

The balance equations (or equilibrium equations) are given by:

$$\begin{aligned}\lambda P_1 &= \lambda P_0\\ \lambda P_2 &= \lambda P_1\\ &\vdots\\ \lambda P_{q-1} &= \lambda P_{q-2}\end{aligned}$$

Thus,

$$P_0 = P_1 = \dots P_{q-2} = P_{q-1} = \frac{1}{q} \tag{1}$$

Let $R = n-1$. A node generates a new message in the next round, i.e. $B_{i,r+1}$ when it reaches the quorum, that is when it becomes at the state $q-1$ with probability $\frac{1}{q}$. The average message arrival rate for node i is then given by:

$$\lambda = \frac{nR}{q} \tag{2}$$

For a round to end, q messages (i.e. block acknowledgment) must be received, thus q transitions must end. Each transition happens at an average rate λ, following an exponential distribution. The distribution of the duration of a round r, denoted by $f_{1round}(t)$, is given by:

$$f_{1round}(t) = \bigotimes_{qtimes} \lambda e^{-\lambda t} \tag{3}$$

Or,

$$F^*_{1round}(s) = (\frac{\lambda}{\lambda+s})^q \tag{4}$$

Where $F^*_{1round}((s)$ is the Laplace transform of $f_{1round}(t)$.

V. Delays Analysis of a Mempool Protocol Based on Two Quorum of Messages

In this generalized two-quorum protocol, a node i initiates the process by broadcasting a block of transactions $B_{i,r}$ to the network. It then waits to receive a first quorum of acknowledgments $A_b^{j,i}$ to form the initial quorum $Q_{i,r}^{(1)}$. To enhance consistency and agreement among the network peers, node i subsequently broadcasts the formed quorum $Q_{i,r}^{(1)}$ to the network. Following this, it waits to receive a second quorum of messages $A_q^{j,i}$. The described protocol is illustrated in Fig. 3,

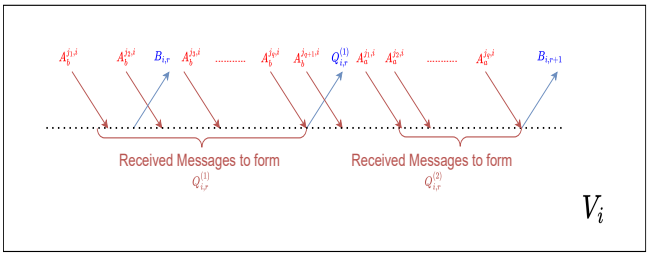

Fig. 3. Exchanged Messages from the point of view of validator V_i

where node i must receive two quorums of different message types, $A_b^{j,i}$ and $A_q^{j,i}$, before it can proceed to the next round and broadcast a new block of transactions.

A. Marcov Chain

Let $X_{i,r}^{(1)}(t)$ be the number of acknowledgments$A_b^{j,i}$ received by node i (in order to reach $Q_{i,r}^{(1)}$), and, let $X_{i,r}^{(2)}(t)$ be the number of acknowledgments $A_q^{j,i}$ received by node i (in order to reach $Q_{i,r}^{(2)}$). According to assumption A1, $\{X_{i,r}^{(1)}(t), X_{i,r}^{(2)}(t)\}_{t\in\mathbb{R}_+}$ is a continuous-time Markov chain described in Fig. 4. We define:

- λ_1 is the average message, $A_b^{j,i}$, arrival rate,
- λ_2 is the average message, $A_q^{j,i}$, arrival rate.

Note that λ_1 is defined by Assumption A1, while λ_2 will be determined later. To determine λ_2, we model the broadcasting of both the block $B_{i,r}$ and the quorum $Q_{i,r}^{(1)}$ using the queuing network shown in Fig. 5. This model allows us to analyze the dynamics and interactions affecting λ_2. In this model,

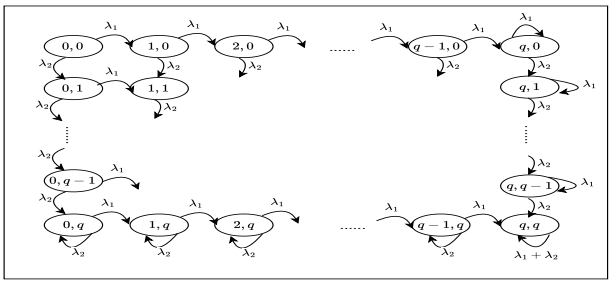

Fig. 4. Markov Chain for the two quorums Protocol

broadcasting a message (either $B_{i,r}$ or $Q_{i,r}^{(1)}$) to $n-1$ nodes (denoted as R) involves creating R duplicate copies of the message, assuming an infinite number of servers.

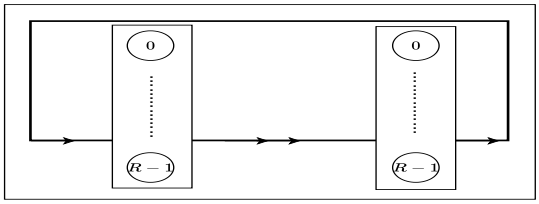

Fig. 5. Two stations queuing network for messages broadcast

As k acknowledgments are received from nodes confirming the message, there are $R-k$ nodes still need to receive it. Consequently, $R-k$ copies of the message remain being transmitted across the network.

The first station represents the block transmission, and the second represents the $Q_{i,r}^{(1)}$ transmission. Both stations have an infinite number of servers (without waiting times), and R is the total number of customers in the queuing network. The state of each station corresponds to the number of customers in each state. A Continuous-time Markov chain models this system. To determine the mean message generation rate, we focus on station 2. Fig.6 describes the transition rate diagram of this second station using a Markov chain.

Let P_k is the probability to have k $A_b^{j,i}$ being transmitted, P_k is given by:

$$P_k = P_0 \frac{R*(R-1)\dots(R-k+1)\mu}{k!\mu}$$

where, R is the total number of possible $A_b^{j,i}$, k is the specific number of $A_b^{j,i}$ we're interested in, P_0 is the probability of having 0 $A_b^{j,i}$ transmitted, and μ is the generation rate associated with each $A_b^{j,i}$. The expression $R\times(R-1)\times\dots\times(R-k+1)$

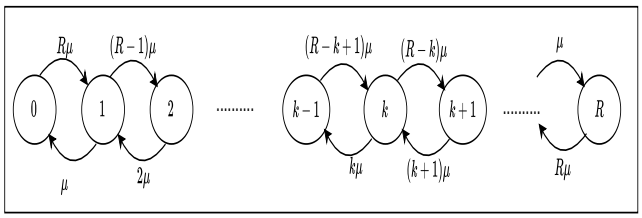

Fig. 6. Messages generation rate for station 2

can be written as $\frac{R!}{(R-k)!}$. Thus, the probability P_k simplifies to:

$$P_k = P_0 \times C_R^k \tag{5}$$

By normalization of probability, $\sum_{k=0}^{R} P_k = 1$. Using Eq. 5, this becomes: $\sum_{k=0}^{R} P_0 \times C_R^k = 1$ Knowing that $\sum_{k=0}^{R} C_R^k = 2^R$, P_0 is given by $P_0 = \frac{1}{2^R}$ Thus, P_k is then given by: $P_k = \frac{C_R^k}{2^R}$

The mean generation rate λ_2 of $Q_{i,r}^{(1)}$ can be derived from the expected number of $A_b^{j,i}$ transmitted, denoted by $\bar{N_2}$: $\bar{N_2} = \sum_{k=0}^{R} k \times P_k$

Substituting this into the expression for the mean generation rate λ_2, we obtain:

$$\lambda_2 = \mu \times \bar{N_2} = \mu \sum_{k=0}^{R} k \times P_k \tag{6}$$

Here, μ represents the rate at which each $A_b^{j,i}$ is generated. Given that P_k follows a binomial distribution, $P_k = \frac{C_R^k}{2^R}$, where C_R^k is the binomial coefficient, the mean generation rate λ_2 can be rewritten as:

$$\lambda_2 = \mu \sum_{k=0}^{R} k \times \frac{C_R^k}{2^R} \tag{7}$$

The duration of a round corresponds to q vertical transitions and q horizontal transitions to go from state $(0,0)$ to state (q,q) in the diagram of Fig. 4 and this whatever the path followed. Thus the distribution of the duration of a round is given by:

$$f_{2round}(t) = \bigotimes_{qtimes} \lambda_1 e^{-\lambda_1 t} \bigotimes_{qtimes} \lambda_2 e^{-\lambda_2 t} \tag{8}$$

Or,

$$F^*_{2round}(s) = (\frac{\lambda_1}{\lambda_1+s})^q (\frac{\lambda_2}{\lambda_2+s})^q \tag{9}$$

B. Analytical Delay Analysis Assuming Exponential Messages Propagation Delays

In this section, as in all the paper, we assume that all delays are exponentially distributed with rate parameter λ abd mean μ. If D is a random delay, then we have:

$$P(D > x) = e^{-x/\mu} \tag{10}$$

This is a common assumption in networked systems because exponential distributions are memoryless and provide a good

fit to describe the transmission time of a message in communication systems [18].

We start by determining the first quorum time $Q_{i,r}^{(1)}$ for node i which will be identically distributed among the nodes. The Laplace transform of the delay is given by:

$$D^*(s) = E[e^{-sD}] = \frac{1}{1+\mu s} \tag{11}$$

Consequently, the Laplace transform of the double delay (*i.e.* of $Q_{i,r}^{(1)}$) is $D^*(s)^2$, and the Laplace transform of the CDF is $F^*(s) = \frac{D^*(s)^2}{s}$. By standard inversion of the Laplace transform we have:

$$F(x) = (1 + x/\mu)e^{-x/\mu} \tag{12}$$

This provides the probability distribution of the total delay of $Q_{i,r}^{(1)}$. As the network size is equal to $3f+1$ and the quorum size is equal to $2f+1$, we will compute the quantity:

$$F^{-1}\left(\frac{2}{3}\right) = \mu W\left(\frac{e^{-1}}{3}\right) \approx 2.289281414\ldots \times \mu \tag{13}$$

where W is the Lambert function. This quantile indicates the delay value below which $\frac{2}{3}$ of the observations fall.

Regarding the second quorum time $Q_{i,r}^{(2)}$ for node i, assuming the same propagation delay distribution for all nodes, then the same quorum time distribution G. The objective is to compute the Laplace transform $G^*(s)$ of the cumulative distribution function $G(x)$.

The Laplace transform of the remaining time ρ_j distribution is $\frac{1-e^{-sT}}{Ys}$, where T is the high probability value of T_j. Thus the Laplace transform of the distribution of $Q_{i,r}^{(2)}$ is $\frac{1-e^{-sT}}{Ts}D^*(s)$ and finally :

$$G^*(s) = \frac{1-e^{-sT}}{Ts^2}D^*(s) = \frac{1-e^{-sT}}{Ts^2(1+\mu s)} \tag{14}$$

By reverse Laplace we have:

$$\begin{aligned} G(x) \quad = \quad & \tfrac{1}{T}\left(x + \left((1-\exp\left((T-x)/\mu\right)\right)\mu + T - x\right)Y(x-T) \\ & -\mu(1-e^{-x/\mu})\Big) \end{aligned} \tag{15}$$

where $Y(.)$ is the Heaviside function.

The last move is to find T such that $G(T) = \frac{2}{3}$, according to equation (15) we should have:

$$1 - \frac{\mu}{T}(1-e^{-T/\mu}) = \frac{2}{3} \tag{16}$$

The resolution of this equation gives:

$$\frac{T}{\mu} = W(-3e^{-3}) + 3 = 2.821439372\ldots \tag{17}$$

In passing it turns out that with a high probability the quorum $Q_{i,r}^{(1)}$ time is before the quorum $Q_{i,r}^{(2)}$ time, the latter closing the round with a high probability.

C. Asymptotic Analysis of a Mempool Protocol based on Round-Trip Quorum

In this section, we assume that the number n of nodes is large. We also believe that the quorum process is in a stationary state. The duration of a random round for any node i is M_i. If $F_{ij}(x)$ is the CDF of one propagation delay from the node i to node j, then assuming that the propagation delay in different directions is independent, we will consider the CDF of one round trip delay from i to any other node j. If those delays are identically distributed when j varies, we can denote $F_i(x)$ as CDF.

We know [19] that when the number n of samples of the same random variable with CDF $F(x)$, the occurrence of the kth smallest sample happens with high probability (*i.e.* with small variance) at delay x such that $F(x) = \frac{k}{n}$. Thus the quorum $Q_{i,r}^{(1)}$ time Y_i^1 is with high probability equal to $F_i^{-1}\left(\frac{2}{3}\right)$.

VI. Simulations and Results

In this section, we present the simulation results of the protocol for both one and two quorum configurations. We compare the distribution of the simulation outcomes with the theoretical distributions derived in equations 3 and 8. To ensure accuracy, we conducted simulations over 1000 rounds.

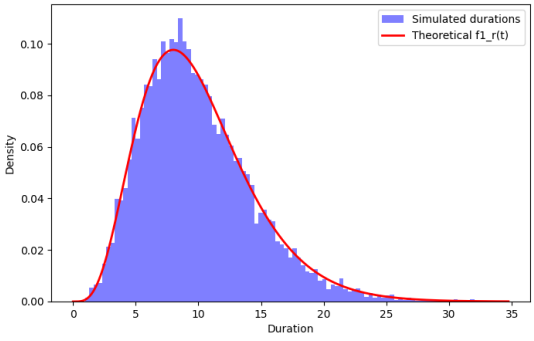

Fig. 7. Distribution of Round Duration for the protocol based on One-Quorum of messages

The plots in Fig. 7 and Fig. 8 visually confirm the close agreement between the simulated data (histograms) and the theoretical distributions: $f_{1round}(t)$ in Eq. 3 and $f_{2round}(t)$ in Eq. 8 (red lines). This strong alignment indicates that our mathematical model accurately captures the behavior of the mempool process, supporting the validity of our theoretical expectations.

In Fig. 8, we observe that the round durations are longer compared to those in Fig. 7. This increase is due to the additional communication overhead and potential delays associated with the requirement of waiting to receive two quorums of messages for each node to move to a new round and propose a new block. This approach significantly enhances the integrity and robustness of the system, reducing the risk of conflicting blocks.

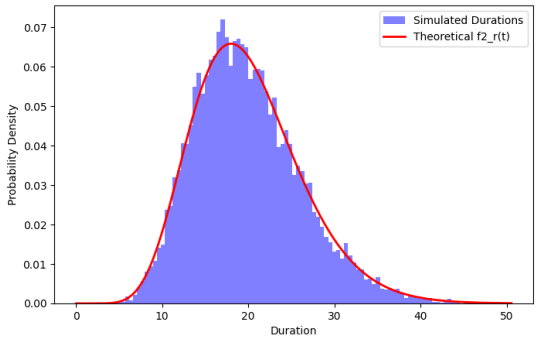

Fig. 8. Distribution of Round Duration for the protocol based on Two-Quorums of messages

The plot in Fig. 9 helps us understand how delays are distributed in the system and provide a critical value T which can be used to make informed decisions about system performance and reliability. The green vertical line at $x = T$ (where $T \approx 2.8214$) tells us that about 67% of all delays will be less than or equal to 2.8214μ. This means if we set a timeout or an expectation for delay at around 2.8214μ, we can be confident that approximately 67% of the delay instances will fall within this time frame.

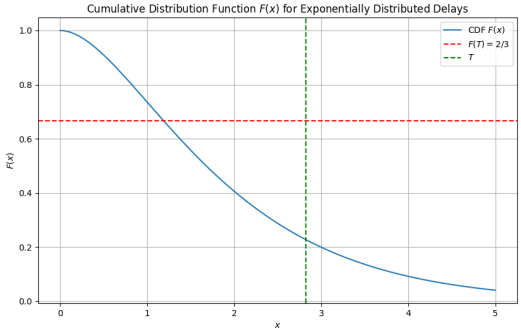

Fig. 9. Distribution of the CDF

VII. Conclusion

This paper examines the round duration distribution in a mempool-based consensus protocol, focusing on both one-quorum and two-quorum requirements. A Markov chain model is employed, also an analytic model that assumes exponentially distributed message propagation delays is established. The asymptotic distribution of quorum formation time is derived. Simulation results are compared with theoretical predictions to assess their alignment.

Acknowledgment

This research was conducted under the project PHC-Maghreb ANGEL 24MAG18.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] M. Onifade, J. A. Adebisi, and T. Zvarivadza, "Recent advances in blockchain technology: Prospects, applications and constraints in the minerals industry," International Journal of Mining, Reclamation and Environment, pp. 1–37, 2024.

[3] W. A. Al-Nbhany, A. T. Zahary, and A. A. Al-Shargabi, "Blockchain-iot healthcare applications and trends: a review," IEEE Access, 2024.

[4] O. Odeyemi, C. C. Okoye, O. C. Ofodile, O. B. Adeoye, W. A. Addy, and A. O. Ajayi-Nifise, "Integrating ai with blockchain for enhanced financial services security," Finance & Accounting Research Journal, vol. 6, no. 3, pp. 271–287, 2024.

[5] K. Hwerbi, I. Amdouni, A. Laouiti, C. Adjih, and L. Saidane, "Veterinary drone: Blockchain-based system for cattle health monitoring," in 2023 International Wireless Communications and Mobile Computing (IWCMC). IEEE, 2023, pp. 1613–1618.

[6] K. Hwerbi, N. Benalaya, I. Amdouni, A. Laouiti, C. Adjih, and L. Saidane, "A survey on the opportunities of blockchain and uavs in agriculture," in 2022 IEEE 11th IFIP international conference on performance evaluation and modeling in wireless and wired networks (PEMWN). IEEE, 2022, pp. 1–6.

[7] H. Yang, Y. Guo, and Y. Guo, "Blockchain-based cloud-fog collaborative smart home authentication scheme," Computer Networks, vol. 242, p. 110240, 2024.

[8] 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, pp. 1–28, 2019.

[9] C. Fan, S. Ghaemi, H. Khazaei, and P. Musilek, "Performance evaluation of blockchain systems: A systematic survey," IEEE Access, vol. 8, pp. 126 927–126 950, 2020.

[10] G. Danezis, L. Kokoris-Kogias, A. Sonnino, and A. Spiegelman, "Narwhal and tusk: a dag-based mempool and efficient bft consensus," in Proceedings of the Seventeenth European Conference on Computer Systems, 2022.

[11] R. K. Kaushal and N. Kumar, "Exploring hyperledger caliper benchmarking tool to measure the performance of blockchain based solutions," in 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO). IEEE, 2024, pp. 1–6.

[12] Z. Dong, E. Zheng, Y. Choon, and A. Y. Zomaya, "Dagbench: A performance evaluation framework for dag distributed ledgers," in 2019 IEEE 12th international conference on cloud computing (CLOUD). IEEE, 2019, pp. 264–271.

[13] M. Alharby and A. Van Moorsel, "Blocksim: a simulation framework for blockchain systems," ACM SIGMETRICS Performance Evaluation Review, vol. 46, no. 3, pp. 135–138, 2019.

[14] J. Qi, J. Yu, and S. Jin, "Nash equilibrium and social optimization of transactions in blockchain system based on discrete-time queue," IEEE access, vol. 8, pp. 73 614–73 622, 2020.

[15] P. Yuan, K. Zheng, X. Xiong, K. Zhang, and L. Lei, "Performance modeling and analysis of a hyperledger-based system using gspn," Computer Communications, vol. 153, pp. 117–124, 2020.

[16] J. Li, Y. Yuan, S. Wang, and F.-Y. Wang, "Transaction queuing game in bitcoin blockchain," in 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018, pp. 114–119.

[17] S. Ricci, E. Ferreira, D. S. Menasche, A. Ziviani, J. E. Souza, and A. B. Vieira, "Learning blockchain delays: A queueing theory approach," ACM SIGMETRICS Performance Evaluation Review, vol. 46, no. 3, pp. 122–125, 2019.

[18] M. J. Rendas, Traitement Statistique du Signal, ESINSA, 1999/2000, 4-ème année.

[19] B. C. Arnold, N. Balakrishnan, and H. N. Nagaraja, A first course in order statistics. SIAM, 2008.