blockchains

Article

Performance of PBFT Consensus under Voting by Groups [†]

Vojislav B. Mišić [1,*], Jelena Mišić [1] and Xiaolin Chang [2]

[1] Department of Computer Science, Toronto Metropolitan University, Toronto, ON M5B 2K3, Canada; jmisic@torontomu.ca
[2] Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing 100091, China; xlchang@bjtu.edu.cn
[*] Correspondence: vmisic@torontomu.ca

[†] This paper is an expanded and revised version of the paper 'The impact of vote counting policy on the performance of PBFT' by Mišić, V.B.; Mišić, J.; Chang, X. Presented at 2021 IEEE Canadian Conference on Electrical and Computer Engineering CCECE-21, online, 12–17 September 2021.

Abstract: Practical Byzantine Fault Tolerance (PBFT) is the protocol of choice for many applications that require distributed consensus between a number of participant nodes. While PBFT assumes a single voting committee, many applications recognize different groups of participants that need to reach a consensus separately before accepting a proposal. To this end, we propose to count the votes by separate groups or committees of participating nodes, instead of all together as in the original PBFT. We then investigate the performance impact of this approach on the mean time to accept a data block and the number of nodes involved in making the final decision. Our results indicate that the proposed solutions impose a slight performance penalty which may be countermanded by reducing the quorum numbers needed in different subsets of the original committee.

Keywords: practical Byzantine fault tolerance (PBFT); bicameral voting; performance evaluation

1. Introduction

Blockchain technology offers high reliability and resilience in many applications, including cryptocurrency and financial systems, but also Internet-of-Things (IoT), healthcare, and identity management (IM) solutions [1–4]. One of the crucial components of such systems is the consensus protocol, a process through which a number of nodes or validators vote (explicitly or implicitly) to accept a block of data that may contain financial transactions, healthcare data, IoT readings, or identity transactions and insert it into the replicated, decentralized blockchain ledger [5]. Many such protocols have been proposed and utilized, and the Practical Byzantine Fault Tolerance or PBFT [6] is perhaps the best-known one, with many derivatives proposed over the years [5]. PBFT requires a permissioned voting committee, i.e., the one in which the identity of each voter is known to everyone else. It is fast and resilient to failures or malicious behavior of up to one-third of the voters. Furthermore, it offers finality as the acceptance decision, once made, cannot be changed anymore. However, PBFT also suffers from a number of drawbacks, most notably from poor scalability as the number of messages exchanged in a single consensus round increases as $\mathcal{O}(N^2)$ for the committee with N nodes. Also, the performance of PBFT consensus depends very much on the transmission delays between the committee nodes, which renders it less than desirable for systems that operate in geographically large areas [7–10].

Scalability may be improved through Proof-of-Work (PoW) consensus which is used in financial blockchain-based systems such as Bitcoin [11] and Ethereum in its initial version [1], which has been replaced by a Proof of Stake-based version in September 2022 [12]. However, their permissionless architecture brings some downsides as well. First, the consensus in such systems suffers from an inherent lack of finality [13]. Second, the use of node-to-node, gossip-like block propagation leads to low throughput which is sometimes artificially constrained to ensure consistency, as has been conducted in Bitcoin.

check for updates

Citation: Mišić, V.B.; Mišić, J.; Chang, X. Performance of PBFT Consensus under Voting by Groups. *Blockchains* **2024**, 2, 134–149. https://doi.org/10.3390/blockchains2020007

Academic Editors: Keke Gai and Liehuang Zhu

Received: 29 February 2024
Revised: 15 April 2024
Accepted: 23 April 2024
Published: 26 April 2024

Regardless of the differences between permissioned and permissionless consensus protocols, two common features can be identified. First, all nodes in a consensus committee have to vote together to reach a consensus decision, and second, the vote of each node carries equal weight. While this is acceptable in many cases, it is less than perfect in identity management systems, and in recent self-sovereign identity systems in particular [14]. Namely, proofs of identity are issued by many different kinds of entities, and they are also verified by those entities when the users present them in order to obtain service. However, the level of trust assigned to different issuers and verifiers are not the same across the board: government agencies such as the Department of Motor Vehicles or the Department of Healthcare are more trusted than commercial entities such as supermarket chains or online retailers [15].

What this means for consensus protocols is that the votes cast by different committee members should have different weights, which corresponds much better to different trust levels encountered in real applications. Trust level may be monitored by changing individual node weights according to the manner in which the nodes vote. This leads to protocols such as Proof of Stake [16] and Proof of Authority [17], among others.

Another approach is to allow the consensus committee to split into a number of disjoint committees that may vote either together, i.e., in a plenary-like mode, or separately, in which case a consensus decision requires that all committees individually accept the proposed data block. This approach is well suited to hierarchical (layered) Internet of Things (IoT) systems [8] as well as to sharded blockchain systems [18,19] in which the global blockchain is replicated in a number of overlapping sub-blockchains or shards.

In this work, however, we focus on the basic scenario where the node committee is split into two or three subsets which are disjoint except for the primary (leader) node. Other than the manner in which votes are counted, i.e., jointly or separately by subsets, there are no changes to the original PBFT protocol. We outline the basic differences between the original PBFT and the split-committee variant and use discrete-event simulation to analyze and compare their performance. Our results show that committee splitting leads to a deterioration of the mean time needed to reach the consensus. Performance could be restored by reducing the quorum requirements for the voting subsets, but this may or may not be acceptable in a given application. These findings provide the foundation for the subsequent design of the consensus layer for hierarchical and/or sharded blockchain applications.

The paper is organized as follows. Section 2 discusses earlier work in this area. Section 3 describes the vote counting approaches in more detail and discusses some design decisions that have to be made to make them work in the most efficient manner. Section 4 evaluates the impact of split voting on the time needed to reach a consensus. Section 5 presents and discusses the results of the performance evaluation of those approaches. Finally, Section 6 concludes the work and outlines some promising directions for further research. This paper is an expanded and revised version of the paper 'The impact of vote counting policy on the performance of PBFT' [20] by Mišić, V.B.; Mišić, J.; Chang, X., Presented at 2021 IEEE Canadian Conference on Electrical and Computer Engineering CCECE-21, online, 12–17 September 2021

2. Related Work

PBFT [6] and related protocols assume that the identities of all participants in the consensus are known; in addition, they generally suffer from scalability issues as the number of messages exchanged to reach consensus increases as $\mathcal{O}(N^2)$, where N is the number of participating nodes. On the other hand, initial implementations of blockchain-based systems such as Bitcoin [11] and Ethereum [1] relied on Proof-of-Work (PoW) consensus which is better suited to their permissionless setting and enjoys better scalability. However, PoW systems suffer from excessive power consumption and lack of finality [13], which was the motivation for a number of approaches to consensus that have been proposed since, including Proof of Stake (PoS) [21], Proof of Activity [16], and Proof of Authority [17], among others.

Many of those systems include simplifications in the original PBFT protocol in order to improve scalability; unfortunately, those improvements often come at a price, as they are susceptible to attacks that can slow down the protocol [22]. Moreover, a fundamental challenge of PBFT, namely the dependence on the truthful behavior of the committee leader is usually resolved by rotating the leader role, often in a pseudo-random schedule and (less often) by allowing competition between prospective leaders [23].

Likely the best-known PoS system nowadays is Ethereum 2.0 [12] which uses a sophisticated voting scheme that involves hundreds of thousands of participants in an attempt to decentralize decision-making. The scheme, known as Gasper [24], combines a protocol for choosing the best chain with another one that targets the finalization of a group of blocks in a chain. However, having this many validators requires relaxing the timing requirements, which makes this protocol vulnerable to a number of sophisticated attacks [25–27]. In addition, the weight of individual validator's votes, as well as the rewards and penalties incurred in the voting process, depend on their effective stake, which can vary in the range of 16 to 32 Ethers (ETH) so as to limit the possibility of centralization of voting power in a small number of validator.

In recent years, new application areas have brought additional challenges to the design of consensus protocols. Those areas include Internet of Things systems [2], identity management systems [28], and integration with blockchain technology [29]. Probably the most important among those challenges is the wide geographical coverage that such systems require and the need to include a large yet time-varying sets of validator nodes that participate in the consensus protocol. Consensus operation in wide area network setting, initially addressed by a comparatively small number of research works [7,10], is becoming more important with the modern Internet of Things (IoT) systems that operate in such environments. Identity management systems have been investigated in more detail [14,30], although the application of blockchain in this context is still far from maturity.

Sharded blockchain systems [18] are gaining traction in recent years, mainly to improve blockchain scalability as not all nodes need to process all data. This makes sharded systems particularly well suited for data sharing in IoT applications [8,31,32] and cloud computing [33]. Sharded blockchains are also used in cryptocurrency systems but many problems remain to be solved, not least because of the complexity of introducing cross-shard transactions which are a necessity in this setup [34]. We note that sharding is planned to be introduced in Ethereum 2.0 but this upgrade has yet to be implemented [12].

3. Approaches to Vote Counting

Let us now discuss the PBFT consensus mechanism and alternatives that may enhance it. Clients of the system submit blocks of data for approval to the designated leader of the consensus committee, as in the original PBFT [6], or to any member of the committee, as in some recent variants of PBFT [8,23]; in the discussions that follow, we will adopt the original approach with a single dedicated leader. The committee consists of a number of nodes referred to as orderers, one of which is designated as the leader or primary, while the others are referred to as replicas.

In the original PBFT [6], upon receiving a block from a client, the primary initiates a new consensus round by broadcasting a PREPREPARE message containing the block to all replicas which validates the block and, if successful, broadcasts a PREPARE message to every other node on the committee. When a sufficient number of those messages is received, all the nodes broadcast a COMMIT message. Nodes that receive a sufficient number of COMMIT messages insert the block into their copies of the blockchain and inform the client about it by sending an ACCEPTED message. However, if the primary finds that the block is invalid, it will not initiate a consensus round, and if the committee does not reach consensus, the block is discarded and the client is informed about the decision. This process is schematically shown in Figure 1.

In this manner, a PBFT committee manages to reach a consensus even if up to (but not including) one-third of the replicas are faulty or behave in a malicious manner. This number will be referred to as a quorum; if the committee consists of $N = 3f + 1$ replicas, the quorum is $2f + 1$ so that at most f faulty or malicious nodes can be tolerated. Together with the finality of the decision, this makes PBFT extremely attractive for many distributed applications.

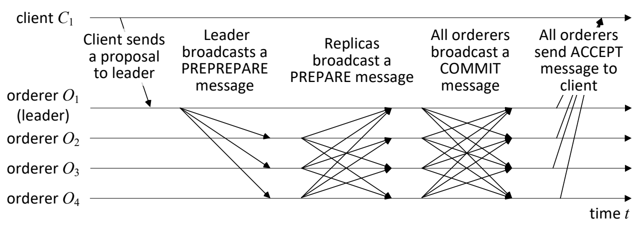

client C_1
Client sends a proposal to leader
Leader broadcasts a PREPREPARE message
Replicas broadcast a PREPARE message
All orderers broadcast a COMMIT message
All orderers send ACCEPT message to client
orderer O_1 (leader)
orderer O_2
orderer O_3
orderer O_4
time t

Figure 1. Operation of PBFT consensus (after [6]).

Different applications might require different types of orderer nodes. Classification may be conducted according to the type of nodes and their roles in the application. One criterion that is frequently encountered is the level of trust that the nodes enjoy: for example, the confirmation of an identity check by a government agency such as the Department of Motor Vehicles would carry more weight than a confirmation by a financial institution, which in turn, would carry more weight than a commercial entity such as a convenience store.

When multiple categories are identified, they may have different numbers of orderers: there may be a dozen or so government agencies, but a hundred or two hundred financial institutions; commercial entities are likely to be counted in thousands. If all orderers vote together and their votes carry equal weight, the quorum is calculated by simply counting the votes. As a result, a straightforward application of PBFT would lead to the possibility for lower-trust orderers to outvote the high-trust ones. For example, in a PBFT committee with 16 orderers, 4 of which are high-trust ones, the low-trust orderers can simply outvote the high-trust ones each time as there are 12 of them and the required quorum is 11 orderers.

In many real-life situations, different categories of participants vote separately, and the proposal needs to be accepted separately by each category to be accepted. A similar approach in a different context is bicameral voting utilized in parliaments of many countries in the world (comparative data on the structure of parliament at Inter-Parliamentary Union, https://data.ipu.org/ (accessed on 22 April 2024). Let us assume that we have N orderers grouped into c categories or groups, labeled with $0 \mathinner{.\,.} c-1$ for convenience, where 0 corresponds to the group with the highest trust and $c-1$ to the one with the lowest trust. Each group i consists of $n(i)$ orderers so that $N = \sum_{i=0}^{c-1} n(i)$ nodes. Each group has its own quorum $q(i)$ which may or may not be equal to the regular PBFT case, as we will see below.

We can then formulate at least three different vote-counting modes as follows:

- In MODE-1 (also referred to as plenary mode), all votes are counted together just as in the original PBFT. In other words, there is a single group with N orderers, at least $q = \lceil (2/3)(N-1) \rceil$ of which have to vote for a proposal to accept it.
- In MODE-2, we allow two 'supergroups' so that some of the actual groups are allocated to the high priority supergroup while others are allocated to the low-priority supergroup. The vote tally is kept separately for each supergroup and a proposed block is accepted only if consensus is reached separately by both supergroups.
- Finally, in MODE-3, each group counts its votes separately and consensus is reached if and only if all groups individually reach a consensus.

A number of design decisions have to be made here. First, do we utilize group voting just in the COMMIT stage, or in both PREPARE and COMMIT stages? (Please refer to Figure 1 for details). The answer to this question depends on the performance of those approaches, which we will investigate in more detail in Section 5.

Second, how do we define the quorums for each group? In other words, do we define a 'more than two-thirds correct orderers' criterion, as in the original PBFT, or adopt a more relaxed approach? However, the former case means that we are modifying the quorum requirement beyond that required by PBFT.

A viable alternative would be to use the original PBFT quorum value as the quorum total, but partition it into quorums for individual groups so that $Q = \lceil (2/3)(N-1) \rceil = \sum_{i=0}^{c-1} q(i)$, where c is the number of groups. In this case, only one of the groups could actually use the original quorum while the quorum in all others would be relaxed to just two-thirds of the available group member count. We will have to say more on this issue in Section 5.

Another open question is, how many categories or groups of orderers should there be? And how do we assign orderers to each category? Namely, having more groups allows for finer control over the consensus protocol but also complicates defining the requirements and certainly makes the protocol harder to understand. On the other hand, a small number of groups makes the system simpler. For obvious reasons, these questions cannot be properly answered without knowing the details of the actual application, which is why we will not consider it in this paper.

4. Split Voting: The Impact on Quorum Delay

We will now show the manner in which committee splitting occurs so that quorum has to be reached in both sub-committees separately affecting the delay to reach the quorum.

Let N be the number of nodes in the entire voting committee. One of the nodes is the primary which initiates a PBFT consensus round by sending a PREPREPARE message with the actual proposal to all other nodes and replicas. Let $t = 0$ denote the time this message is sent.

Assuming that nodes are numbered 0 to $N-1$ and that node 0 is the primary, replica $i = 1 \,..\, N-1$ will receive the PREPREPARE message after a delay. This one-way delay is a random variable that depends on the distance between the nodes, the bandwidth of their interconnection, and the number of routers along the way.

When the replica i receives the PREPREPARE message, it will check the proposed data item and, if validated, respond with a PREPARE message to all other nodes, primary and replicas. These messages will be received after appropriate one-way delays.

The probability of this event occurring at time τ_1 may be obtained as the probability that the total delay from the primary through a replica $j \neq i$ to the target node i is less than or equal to τ, which we denote with $F_{0,j,i}(\tau)$. This is, in fact, a cumulative distribution function (cdf) for the two-hop delay from primary (node 0) to target (node i) via replica $j, j > 0, j \neq i$.

Note, that the primary will not send a PREPARE to any other node, and the target replica i will not send a PREPARE to itself. In that case, the target replica will pronounce that the quorum has been reached at time τ if it has received exactly q PREPARE messages from as many nodes, while the remaining $N-2-q$ such messages have not yet arrived:

$$\begin{aligned} P^{(1)}(N;q;\tau) = \text{Prob}(&q \text{ messages received by } \tau \\ &\text{AND} \\ &N-q \text{ messages not received at } \tau) \end{aligned} \tag{1}$$

As a result, the time to reach quorum τ_1 will be the minimum value of τ for which the above probability equals 1. This means that exactly q PREPARE messages have been received by the target node i, as shown in Figure 2a.

But what would happen if the vote counting was split into two sub-committees, would the quorum be reached in both of them, just one, or none? Note, that we are referring to the same event but with a different manner of counting; this does not include any additional delay except when counting the votes, but this time is negligible compared to transmission delays between nodes.

Let us assume that the voting committee is split into two sub-committees with N_1 and N_2 nodes, respectively. Assuming that no node is allowed to vote in both sub-committees, with the exception of the primary, the number of nodes in the two sub-committees would have to satisfy the equation $N_1 + N_2 = N + 1$.

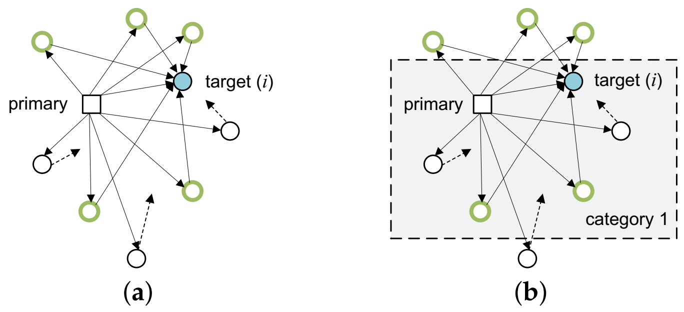


Figure 2. The impact of split voting on quorum delays. (**a**) The case of joint vote: node *i* receives the necessary votes to achieve quorum at time τ_i. (**b**) The case of split vote: node *i* receives votes at time τ_i.

Likewise, the quorum would be split into sub-quorums q_1 and q_2. For obvious reasons, $q_1 < N_1, q_2 < N_2$, and $q_1, q_2 > 0$. But in fact, the smallest meaningful quorum is 3, $3 \leq q_1, q_2$, for a sub-committee of four, as stipulated by PBFT. At the upper end, both quorums may not exceed the original quorum minus three: $q_1, q_2 \leq q - 3$, as a larger value, would mean that one or both of the sub-committees (which have disjoint membership) has more stringent quorum requirements than the original PBFT. In the worst case, $q_1 + q_2 = q$, but this would mean that one of the sub-committees has less stringent requirements than the original PBFT.

Let the target node *i* belong to sub-committee 1, as shown in Figure 2b. In this case, the quorum would be reached by the time τ_1 if q_1 PREPARE messages arrive at the node by that time. Since the total of q such messages have arrived, at least q_1 of the corresponding source nodes from the original quorum must remain in sub-committee 1. However, sub-committee 1 will include not just nodes that did participate in the original quorum, but also those that did not. This is schematically shown in Figure 2b.

Then, the probability that one of the q nodes from the original quorum is included in the sub-committee 1 is $p_1 = \frac{q}{N-2}$, as the primary and the target node do not enter the count; the probability that another one of the original quorum nodes included is $p_2 = \frac{q-1}{N-2-1}$ and so on.

In general, the probability that q_1-th of the nodes from the original quorum is included will be

$$p_{q1} = \frac{q-q_1+1}{N-2-(q_1-1)} \tag{2}$$

The joint probability that all q_1 nodes in the new quorum belong to the original one is

$$P_{q1} = \frac{\prod_{i=0}^{q_1-1}(q-i)}{\prod_{i=0}^{q_1-1}(N-2-i)} = \prod_{i=0}^{q_1-1} \frac{q-i}{N-2-i} \tag{3}$$

If the original quorum has been reached at time τ_1 so that $P^{(1)}(N;q;\tau_1) = 1$, probability for a random selection of nodes for category 1 to reach the required quorum of $q1$ at that same time will be P_{q1}.

Let us now calculate the value of P_{q1} for different values of N and q_1. The simplest committee that can be split into two has $N = 9$ orderers, in which case $N_1 = N_2 = 4$, and each sub-committee requires a quorum of $q_1 = 3$. As the data in the first row of Table 1 show, the probability that the required quorum will be achieved by the time τ is only 0.417;

in other words, only 41.7% of all possible sub-committee partitions will reach quorum by the same time the joint vote reaches the quorum.

If we add more orderers and increase the quorum requirements, we obtain the numbers in other rows of Table 1; for clarity, possible values of P_{q1} (with q_1 ranging from 3 to $q - 3$, both inclusive) are shown in boldface. Note, that the joint quorum q is actually a stepwise function of the number of orderers N, an increase in N may or may not lead to an increase in q.

As can be seen, all values of P_{q1} are lower than 0.5, which means that, more often than not, the time for a sub-committee to reach the quorum will be *longer* than in the case of a joint vote by all orderers. Note, that these results pertain just to the quorum needed in the PREPARE stage; the COMMIT decision requires an additional round of broadcasts and vote counting, as explained above, which will lead to further differences between the two voting modes.

Table 1. Values of probability P_{q1} for different values of q and corresponding possible values of q_1.

N	q	$q_1 = 3$	$q_1 = 4$	$q_1 = 5$	$q_1 = 6$
9	7	0.417	0.278		
10	7	0.292	0.167	0.083	
11	8	0.339	0.212	0.121	
12	9	0.382	0.255	0.159	0.091

Due to the randomness of transmission delays, not every vote will result in the same set of orderers forming a quorum. However, the argument outlined above still holds.

Values of P_{q1} do increase when both N and q increase, but the resulting probabilities nonetheless remain below 0.5.

The overall conclusion is that split voting will lead to an increase in consensus time compared to joint voting. This will be confirmed by the simulations presented in the next Section.

5. Performance Evaluation

To evaluate the performance of the proposed voting modes, we have built a simulator of the PBFT network using the multi-paradigm simulation tool Anylogic by Anylogic, Inc., Oakdale Terrace, IL, USA. The main performance indicators in our experiments were mean time to accept a block, the mean number of blocks not accepted for the duration of the experiment, and the mean number of nodes needed to reach a consensus needed to accept a block.

As explained above, orderers were divided into three different groups of nodes, labeled 0, 1, and 2, respectively. In accordance with the the discussion above, we have set up three voting modes:

- In MODE-1, all nodes vote together regardless of the group they belong to.
- In MODE-2, nodes from group 0 vote as a group while nodes from groups 1 and 2 vote together as a single group.
- Lastly, in MODE-3 each group votes independently of the others, and a consensus has to be reached by each group separately for a block to be accepted.

Group counts and resulting quorums N_i/q_i, $i = 0, 1, 2$, are shown in Table 2. In both MODE-2 and MODE-3, group 0 has four nodes and a quorum of three while the remaining nodes are equally split between the other two groups (which vote together in MODE-2, but separately in MODE-3). As outlined above, the primary is included ex officio in the node count and the COMMIT quorum for each group, which is why the corresponding sums are off by one in MODE-2, and by two in MODE-3. The inclusion of the primary in every quorum means that the primary's vote is worth more than that of any other node, but the actual quorum, i.e., the number of distinct nodes that need to vote for a proposal, is still equal to the original one.

Table 2. Number of nodes/quorum for different total number of orderers and different voting modes.

Total	MODE-2 Groups		MODE-3 Groups		
16/11	4/3	13/9	4/3	7/5	7/5
25/17	7/5	19/13	7/5	10/7	10/7
34/23	10/7	25/17	10/7	13/9	13/9
43/29	13/9	31/21	13/9	16/11	16/11
52/35	16/11	37/25	16/11	19/13	19/13
61/41	19/13	43/29	19/13	22/15	22/15
70/47	22/15	49/33	22/15	25/17	24/17
79/53	25/17	55/37	25/17	28/19	28/19

The voting modes were applied either in both PREPARE and COMMIT stages or only in the COMMIT stage (see Figure 1 for details). In the latter case, the decision in the PREPARE stage is reached by counting votes from all nodes together, equivalent to MODE-1.

Orderer nodes were randomly positioned in a square area that was smaller than the client area; the ratio of delays in the former to those in the latter will be referred to as the orderer area factor δ. Orderer nodes were interconnected in a full graph using TCP connections with 100 Mbps bandwidth per node.

In all experiments, a total of 50 client nodes generated data blocks to be linked and were randomly positioned in a square area corresponding to a large metropolitan area network with maximum delay of 5 ms along one edge of the square.

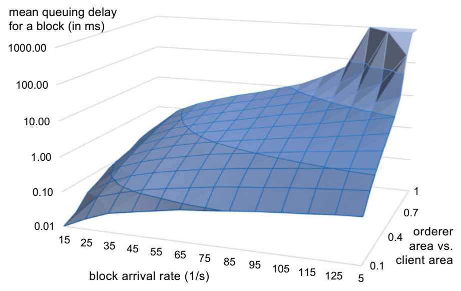
Data blocks with a mean size of 64 KB were generated following a Poisson process at a prescribed aggregate rate expressed in blocks per second (bps). Blocks were injected at a randomly chosen client node, which then sent it to the primary that queued them and subsequently initiated the PBFT consensus protocol.

5.1. Baseline: MODE-1 with Fixed Number of Orderers

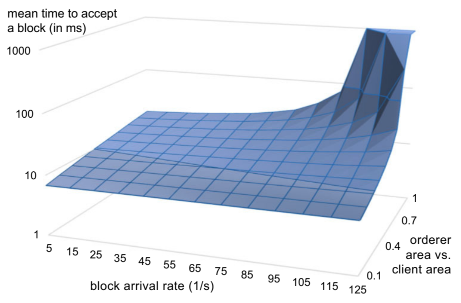
In our first experiment, we varied the block arrival rate in the range of 5 to 125 bps, in steps of 10 bps, and the orderer area factor in the range of 0.1 to 1; the latter range corresponds to different scenarios of orderer placement, from close vicinity to one another, $\delta = 0.1$, to being widely dispersed throughout the client area at $\delta = 1$. The total number of orderers was kept constant at 25, with group membership count chosen as shown in the second row of Table 2. In the first set of runs, we have used MODE-1 voting to establish baseline performance, with each run lasting for 1000 s of simulated time.

The delay performance of the network operating in MODE-1 is shown in Figure 3. Mean queuing delay, i.e., the time that a block spent in the primary orderer queue is shown in Figure 3a. As can be seen, when the orderer area factor is small and the block arrival rate is low, incoming blocks are taken into processing almost instantaneously. Queuing delay increases with both the orderer area factor and block arrival rate and the increase becomes very steep at values of δ above 0.7 or so, and at block arrival rates above 100 bps, which is to be expected as the offered load of the primary orderer queue becomes closer to saturation.

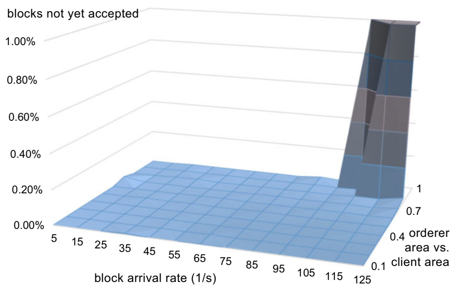
Figure 3b shows the total delay, i.e., mean time to accept a block, which includes the transmission time from the client to primary orderer, the time to reach a consensus decision, and the time until the client receives the necessary number of ACCEPT notifications from orderers. It is fairly constant throughout the observed range, but it increases at high block arrival rates and large orderer area factor due to the increase in queuing delay at the primary. Large queuing delays are also the cause of an abrupt increase in the number of blocks that were not accepted during the experiment run, as can be seen in Figure 3c.



(**a**)



(**b**)



(**c**)

Figure 3. Baseline performance: MODE-1 voting with 25 orderers. (**a**) Mean queueing delay for a block (i.e., time from receiving a block to sending a PREPREPARE broadcast). (**b**) Mean time to accept a block. (**c**) Number of blocks sent in but not yet accepted.

In all experiment runs in this set, the quorum required to accept a block was 17 orderers out of a total of 25 in both PREPARE and COMMIT stages of the protocol.

5.2. MODE-2

In our second set of experiments, we have applied the MODE-2 voting mode, again under a fixed number of orderers. Performance results obtained in this set of experiments are shown in Figure 4. The diagrams in the top row correspond to the case where separate voting is used at both PREPARE and COMMIT stages, while those in the bottom row show the performance for the case where separate voting is used in the COMMIT stage only.

Figure 4a,b show the ratio of mean delay to accept a block vs. the baseline value for a separate voting at both PREPARE and COMMIT stages and for separate voting at the COMMIT stage only, respectively. As can be seen, the delays are up to 20% higher than those in the baseline case in a wide range of values for block arrival rate and orderer area factor. Only in the top right corner, where the block arrival rate reaches 85 bps or higher, and orderer block area factor exceeds 0.6, is an increase in the delays noticeable.

Note, that the baseline value already includes the impact of increased queuing delays at high block arrival rates and large orderer area factors. Hence, the increase shown here is due to the difference in vote counting policy which requires the system to reach a consensus in two separate groups, rather than a single, joint one. In this case, random transmission delays may lead to a scenario in which consensus is reached in one of the groups but not the other one. In fact, one of the voting groups may reach an even higher number of votes before the other one reaches the necessary minimum. As a result, the time to reach consensus increases with voting by groups.

This observation may be confirmed by comparing the diagrams in Figure 4a,b: the latter one corresponds to the case where separate voting applies only to the COMMIT stage but not the PREPARE stage. Therefore, the delay increase with respect to the baseline, case is lower; the difference is small but noticeable.

Similar observations may be made from the diagrams of the number of blocks not accepted during the experiment run, as shown in Figure 4c,d for the two variants of consensus, respectively. The area with non-zero packet loss is similar in both cases, but it is slightly smaller in the latter case, where separate voting is applied only in the COMMIT stage.

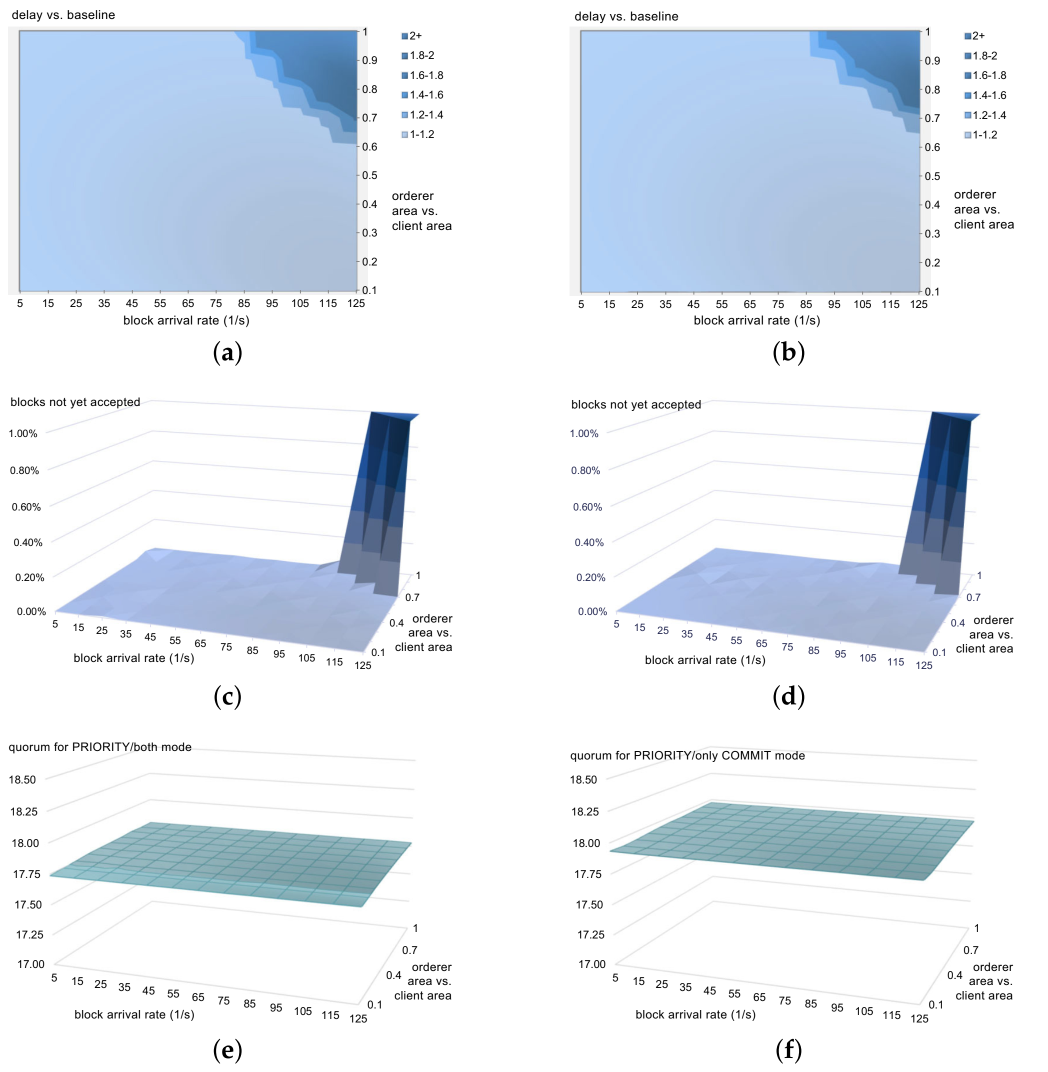


Figure 4. Performance of MODE-2 voting with 25 orderers in two voting committees. (**a**) Mean time to accept a block for MODE-2 with separate voting at both PREPARE and COMMIT stages vs. corresponding time for MODE-1. (**b**) Mean time to accept a block for MODE-2 with separate voting at COMMIT stage only vs. corresponding time for MODE-1. (**c**) Number of blocks not yet accepted under MODE-2 with separate voting at both PREPARE and COMMIT stages. (**d**) Number of blocks not yet accepted under MODE-2 with separate voting at COMMIT stage only. (**e**) Mean number of orderer nodes that confirmed a block at the time consensus is reached under MODE-2 with separate voting at both PREPARE and COMMIT stages. (**f**) Mean number of orderer nodes that confirmed a block at the time consensus is reached under MODE-2 with separate voting at COMMIT stage only.

The mean number of orderers that confirmed a block at the time consensus is reached, summed over all voting committees, is shown in Figure 4e,f for the two voting variants, respectively. As can be seen, more orderers will have the block confirmed than in the baseline case with MODE-1 voting mode, which is caused by the separation of voting committees, as explained above. The increase is not high—only about 0.75 to 0.9, or about 4.4 to 5.3%, above the baseline value of 17. Interestingly enough, the increase is lower in the variant where separate voting is applied both in the PREPARE and COMMIT stages. We hypothesize that this is due to the randomizing effect of transmission delays which is more pronounced in the variant with separate voting in the COMMIT stage only.

5.3. MODE-3

In our third set of experiments, we have applied MODE-3 voting in which consensus must be reached separately in each of the three groups of orderers, as per Table 2. Performance results obtained in this set of experiments are shown in Figure 5. The diagrams in the top row correspond to the case where separate voting is used at both PREPARE and COMMIT stages, while those in the bottom row show the performance for the case where separate voting is used in the COMMIT stage only.

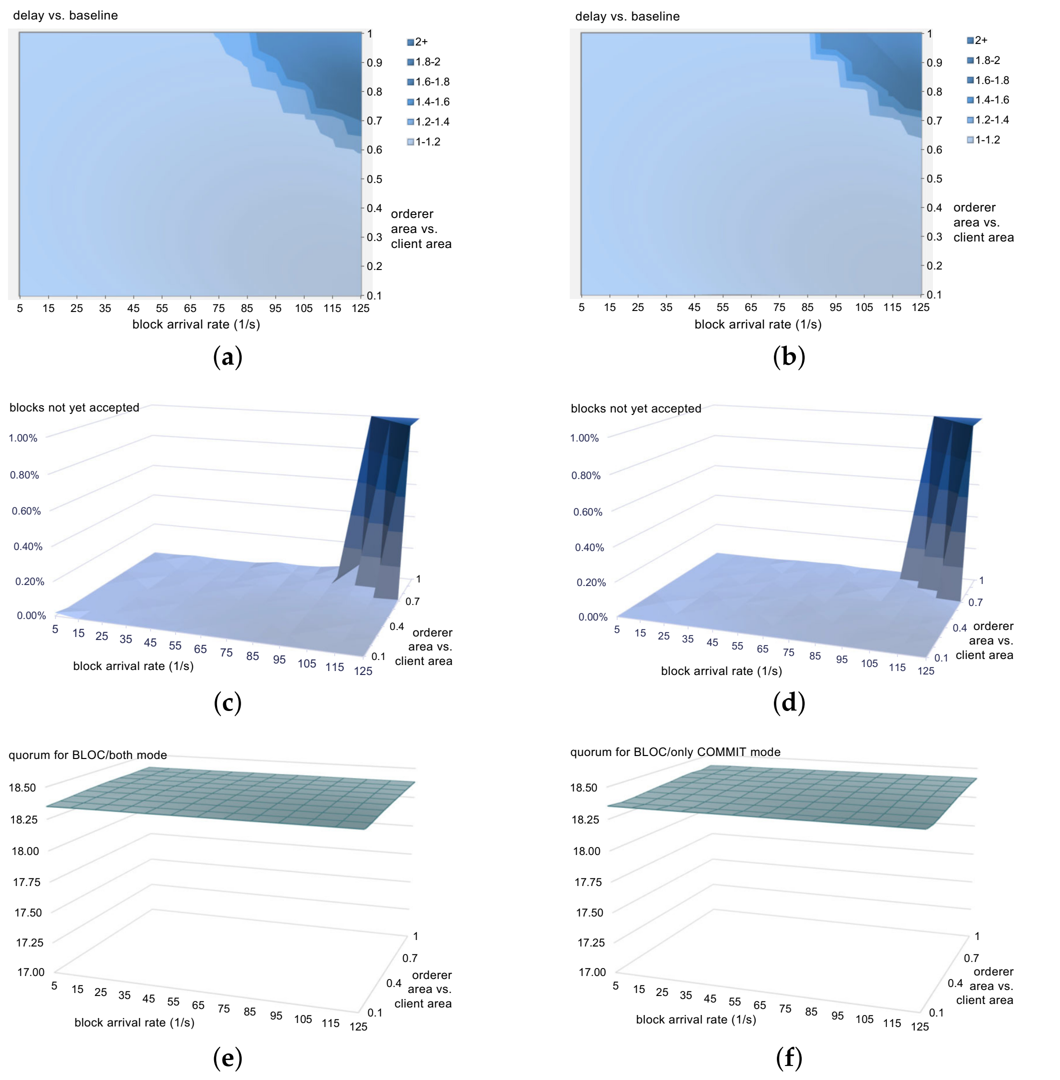


Figure 5. Performance of MODE-3 voting with 25 orderers in three voting committees. (**a**) Mean time to accept a block in MODE-3 voting at both PREPARE and COMMIT stages vs. corresponding time for MODE-1. (**b**) Mean time to accept a block in MODE-3 with voting by bloc at COMMIT stage only vs. corresponding time for MODE-1. (**c**) Number of blocks not yet accepted in MODE-3 voting at both PREPARE and COMMIT stages. (**d**) Number of blocks not yet accepted in MODE-3 with voting by bloc at COMMIT stage only. (**e**) Mean number of orderer nodes that confirmed a block at the time consensus is reached under MODE-3 with voting by bloc at both PREPARE and COMMIT stages vs. corresponding number for MODE-1. (**f**) Mean number of orderer nodes that confirmed a block at the time consensus is reached under MODE-3 with voting by bloc at COMMIT stage only vs. corresponding number for MODE-1.

The mean time needed to accept a block, relative to the baseline performance (MODE-1) is shown in Figure 5a,b. The former diagram corresponds to the variant with separate voting in both the PREPARE and COMMIT stages, while the latter depicts the variant with separate voting in the COMMIT stage only. As can be seen, delays are up to 20% higher than in the baseline case in a wide range of block arrival rates and orderer area factor values, and they increase when both of these parameters are near the high end of their respective ranges. However, the delay increase occurs slightly earlier than under MODE-2, which can be explained by the same mechanism as above. Namely, separate voting means that one or, in this case, even two groups may reach a consensus or even overshoot it, but still have to wait for the vote in the remaining group. As a result, reaching the final decision will take more time than in the baseline case and more time than in the MODE-2 with two voting committees.

At this point, we may ask what would happen if we were to use the usual PBFT rule (i.e., two-thirds plus one) separately in each group, instead of the relaxed values as per Table 2? Obviously, this would require more votes in each committee than is currently the case, and that would in turn lead to even longer delays.

The number of blocks not accepted during the experiment runs for the variants with separate voting in both PREPARE and COMMIT stages and in the COMMIT stage only, are shown in Figure 4c,d, respectively. Again, there is no packet loss in the large portion of the observed range of block arrival rates and orderer area factors, and a steep increase in the high end of the range. The zone with non-zero packet loss is slightly larger than in the corresponding diagrams under MODE-2, Figure 4c,d.

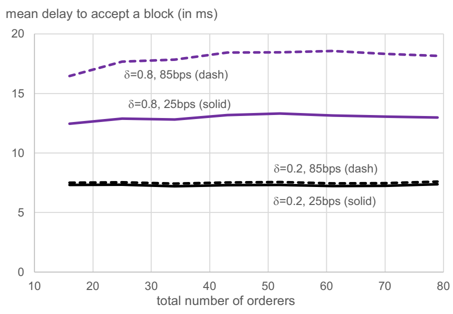
The mean number of nodes that confirmed a block at the moment of reaching consensus is shown in Figure 5e,f for the separate voting in PREPARE and COMMIT stages and in the COMMIT stage only, respectively. In both cases, there is a noticeable increase of about 1.35 (or 8%) over the baseline case. The slight increase with the orderer area factor visible in Figure 5f confirms that the main mechanism for this increase is the randomness of transmission delays, which is more pronounced when the orderers are located in a larger area.

5.4. The Impact of the Total Number of Orderers

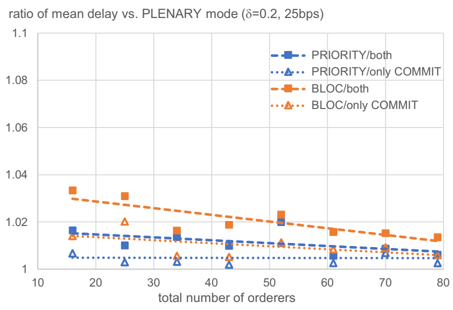
To learn more about the behavior of the proposed schemes, we have conducted another set of experiments in which the total number of orderers varied from 16 to 79, as per Table 2. The orderer area factor and block arrival rate took two discrete values each: 0.2 and 0.8 for the former, and 25 and 85 bps for the latter. These values were chosen to correspond to densely and sparsely packed orderers, under low and high block load, based on the results presented in Sections 5.1–5.3 above.

As before, we began with the baseline case, i.e., MODE-1; the resulting values for the mean delay needed to accept a block are shown in Figure 6a. As can be seen, when the orderer area factor is small ($\delta = 0.2$), there is virtually no difference between delays at low and high load values, which confirms the results shown in the corresponding diagrams above. However, when the orderer area is close to the client area ($\delta = 0.8$), the impact of load begins to show, as mean delays are 35 to 40% higher at a high load (i.e., block arrival rate of 85 bps). Interestingly enough, the delays show very little dependence on the number of orderers, esp. in the case of small orderer areas where the delays are virtually constant. This is to be expected since the complexity of the PBFT protocol stems from the number of messages, while transmission delays depend mostly on the distance between orderers.

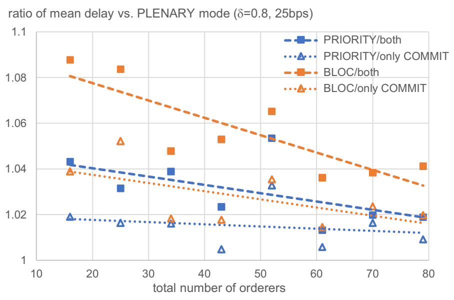
The next set of diagrams, Figure 6b–e, show the ratio of mean delay to the baseline case for MODE-2 and MODE-3 voting modes with two variants each. The actual data points are shown with symbols while the lines show the corresponding linear approximations. To facilitate comparison, diagrams in the same row—which correspond to the same load, as defined by the block arrival rate—are drawn using the same scale.



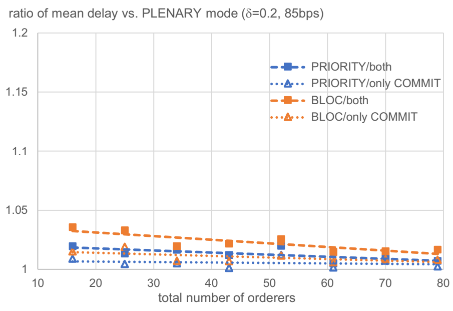
(**a**)



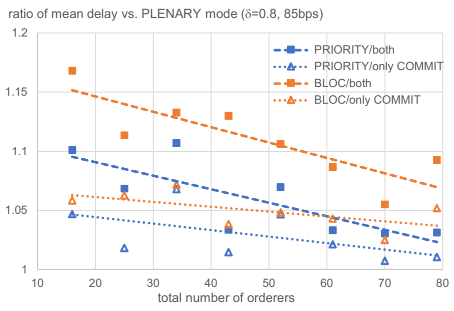
(**b**)



(**c**)



(**d**)



(**e**)

Figure 6. Performance of the system under variable number of orderers: delays and delay ratios. Symbols denote data points, lines are linear approximation of those points. (**a**) Delays in MODE-1. (**b**) Delay ratios vs. MODE-1, orderer area factor $\delta = 0.2$, block arrival rate 25 bps. (**c**) Delay ratios vs. MODE-1, orderer area factor $\delta = 0.8$, block arrival rate 25 bps. (**d**) Delay ratios vs. MODE-1, orderer area factor $\delta = 0.2$, block arrival rate 85 bps. (**e**) Delay ratios vs. MODE-1, orderer area factor $\delta = 0.8$, block arrival rate 85 bps.

As can be seen, when the orderer area is small ($\delta = 0.2$, two diagrams on the left), the delays are up to about 3% higher than in the baseline case. However, when the orderer area is large ($\delta = 0.8$, diagrams on the right), the delays are noticeably higher. The increase relative to the baseline case is up to 8% in the case of low load, and as high as 15% in the case of high load.

In all cases, MODE-3 results in higher delays than MODE-2, while the variants where separate voting is utilized in the COMMIT stage only gives rise to lower delays than when both PREPARE and COMMIT stages utilize separate voting. This is again the consequence of the need for all groups to reach consensus separately, which extends the wait for consensus and, consequently, mean delays.

The final set of diagrams shows the change in the mean number of votes at the time a consensus decision is reached, Figure 7. The numbers are quite close, which is why we have shown MODE-1 mode (baseline values corresponding to the first column on Table 2) with a solid line, while the values for MODE-2 and MODE-3 are shown in blue and orange, respectively. As can be seen, the increase in the mean number of votes is higher for MODE-3 than that for MODE-2, which is higher than that of MODE-1 mode. Relative increase, however, remains limited throughout the observed range of the total orderer count. Finally, as before, lower delays are obtained when separate voting is utilized in the COMMIT stage only.

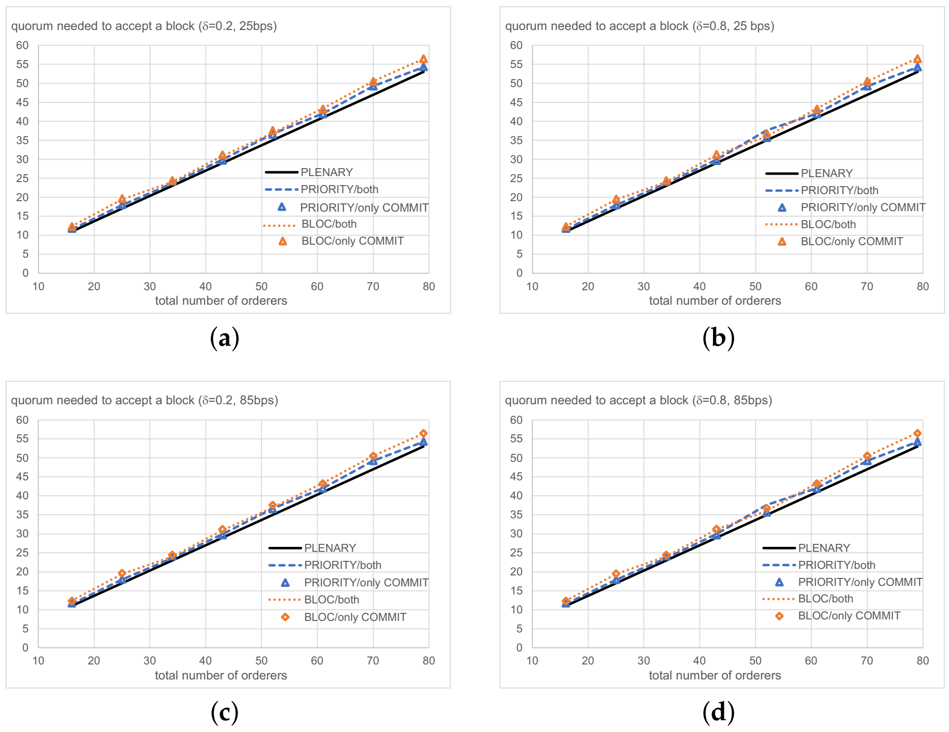


Figure 7. Performance of the system under variable number of orderers: mean number of orderer nodes that confirmed a block at the time consensus is reached. (**a**) Orderer area factor $\delta = 0.2$, block arrival rate 25 bps. (**b**) Orderer area factor $\delta = 0.8$, block arrival rate 25 bps. (**c**) Orderer area factor $\delta = 0.2$, block arrival rate 85 bps. (**d**) Orderer area factor $\delta = 0.8$, block arrival rate 85 bps.

6. Conclusions

To summarize, in this paper we have considered the performance of PBFT in which the voting committee is split into two or three separate groups, each of which has to reach a consensus independently of the others. Global consensus is reached when all of the groups reach their individual consensus decisions. The required PBFT quorum of two-thirds of the nodes plus one is kept for the entire committee, and partitioned among all groups; the permanent leader is considered to belong to all groups.

Our results show that, as long as the system load (which is mostly dictated by the block arrival rate) is low and the orderers are confined to a portion of the area in which the client nodes that generate blocks are located, the delays are generally low and virtually all of the blocks are accepted in time.

More importantly, we show that delays are lower when the split-vote approach is utilized only in the COMMIT stage of the protocol, rather than in both the PREPARE and COMMIT stages.

In addition, this solution is shown to lead to an increase in the mean delay needed to accept a block, and also in the increase in the number of votes that have to be received before reaching a consensus. The said increase would be even larger if each of the groups were to use the original PBFT quorum value (two-thirds plus one node), rather than the simplified version we have adopted. Note, that in this case, the total quorum requirement, calculated over all sub-committees, would exceed that of the original PBFT.

If these performance parameters are to be brought back to the level achieved with traditional PBFT in which all orderers vote as a single group (i.e., in MODE-1), we would need to reduce the quorum thresholds in some or all of the groups. Whether this is feasible or not, depends on the actual application that will use this approach to PBFT consensus.

Alternatively, we might create the voting groups by location proximity, which would render the scheme applicable in a wide range of scenarios including hierarchical systems, identity management, and sharding. In addition, the split-vote scheme can easily be adapted to use different weights for different groups of orderer nodes. These directions will be addressed in our future work.

Author Contributions: Conceptualization, V.B.M.; methodology, V.B.M., J.M., X.C.; formal analysis, V.B.M., J.M., X.C.; writing: V.B.M., J.M., X.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NSERC Discovery Grants—Individual (V.B.M., J.M.) and by National Natural Science Foundation of China under Grant 62272028 (X.C.).

Data Availability Statement: There is no data available for this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Buterin, V. Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. 2017. Available online: https://ethereum.org/en/whitepaper/ (accessed on 22 April 2024).
2. Conoscenti, M.; Vetro, A.; De Martin, J.C. Blockchain for the Internet of Things: A systematic literature review. In Proceedings of the 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Agadir, Morocco, 29 November–2 December 2016; pp. 1–6.
3. Mišić, V.B.; Mišić, J.; Chang, X. Towards a Blockchain-Based Healthcare Information System. In Proceedings of the 2019 IEEE/CIC International Conference on Communications in China (ICCC), Changchun, China, 11–13 August 2019.
4. van Bokkem, D.; Hageman, R.; Koning, G.; Nguyen, L.; Zarin, N. Self-sovereign identity solutions: The necessity of blockchain technology. *arXiv* **2019**. [CrossRef]
5. Bano, S.; Sonnino, A.; Al-Bassam, M.; Azouvi, S.; McCorry, P.; Meiklejohn, S.; Danezis, G. SoK: Consensus in the Age of Blockchains. In Proceedings of the 1st ACM Conference on Advances in Financial Technologies, Zürich, Switzerland, 21–23 October 2019; pp. 183–198.
6. Castro, M.; Liskov, B. Practical Byzantine Fault Tolerance. In Proceedings of the OSDI: Symposium on Operating Systems Design and Implementation, New Orleans, LA, USA, 22–25 February 1999.
7. Amir, Y.; Danilov, C.; Kirsch, J.; Lane, J.; Dolev, D.; Nita-Rotaru, C.; Olsen, J.; Zage, D. Scaling byzantine fault-tolerant replication to wide area networks. In Proceedings of the International Conference on Dependable Systems and Networks (DSN'06), Philadelphia, PA, USA, 25–28 June 2006; pp. 105–114.
8. Mišić, J.; Mišić, V.B.; Chang, X. Optimal multi-tier clustering of permissioned blockchain systems for IoT. *IEEE Trans. Veh. Technol.* **2022**, *71*, 2293–2304. [CrossRef]
9. Sousa, J.; Bessani, A. Separating the WHEAT from the Chaff: An Empirical Design for Geo-Replicated State Machines. In Proceedings of the 2015 IEEE 34th Symposium on Reliable Distributed Systems (SRDS), Montreal, QC, Canada, 28 September–1 October 2015; pp. 146–155.
10. Veronese, G.S.; Correia, M.; Bessani, A.N.; Lung, L.C. EBAWA: Efficient Byzantine Agreement for Wide-Area Networks. In Proceedings of the 2010 IEEE 12th International Symposium on High Assurance Systems Engineering, San Jose, CA, USA, 3–4 November 2010; pp. 10–19.
11. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System, 2008. Available online: https://bitcoin.org/en/bitcoin-paper (accessed on 22 April 2024)
12. Edgington, B. *Upgrading Ethereum: A Technical Handbook on Ethereum's Move to Proof of Stake and Beyond*; Online PDF Edition 0.3: Capella [WIP]; Ethereum Foundation: Zug, Switzerland, 2023. Available online: https://eth2book.info/capella/ (accessed on 22 April 2024)
13. Cachin, C.; Vukolić, M. Blockchain consensus protocols in the wild. *arXiv* **2017**. [CrossRef]
14. Stokkink, Q.; Pouwelse, J. Deployment of a Blockchain-Based Self-Sovereign Identity. *arXiv* **2018**. [CrossRef]
15. Windley, P.; Reed, D. *Sovrin: A Protocol and Token for Self-Sovereign Identity and Decentralized Trust*; White Paper; Sovrin Foundation: Salt Lake City, UT, USA, 2018. Available online: https://sovrin.org/wp-content/uploads/2018/03/Sovrin-Protocol-and-Token-White-Paper.pdf (accessed on 22 April 2024).
16. Bentov, I.; Lee, C.; Mizrahi, A.; Rosenfeld, M. Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract]. *ACM Sigmetr. Perform. Eval. Rev.* **2014**, *42*, 34–37. [CrossRef]

17. De Angelis, S.; Aniello, L.; Baldoni, R.; Lombardi, F.; Margheri, A.; Sassone, V. PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain. In Proceedings of the Italian Conference on Cyber Security, Milan, Italy, 6–9 February 2018.
18. Hong, Z.; Guo, S.; Li, P. Scaling Blockchain via Layered Sharding. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 3575–3588. [CrossRef]
19. Okegbile, S.D.; Cai, J.; Alfa, A.S. Practical Byzantine Fault Tolerance-Enhanced Blockchain-Enabled Data Sharing System: Latency and Age of Data Package Analysis. *IEEE Trans. Mob. Comput.* **2024**, *23*, 737–753. [CrossRef]
20. Mišić, V.B.; Mišić, J.; Chang, X. The impact of vote counting policy on the performance of PBFT. In Proceedings of the Annual IEEE Canadian Conference on Electrical and Computer Engineering CCECE-21, Virtually, 12–17 September 2021.
21. Mišić, J.; Mišić, V.B.; Chang, X. Proof of Stake Voting in Multiple Entry PBFT System. In Proceedings of the ICC 2022-IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2022; pp. 1–6.
22. Amir, Y.; Coan, B.; Kirsch, J.; Lane, J. Prime: Byzantine replication under attack. *IEEE Trans. Dependable Secur. Comput.* **2010**, *8*, 564–577. [CrossRef]
23. Mišić, V.B.; Mišić, J.; Chang, X. Arbitration Mechanisms for Multiple Entry Capability in PBFT for IoT Systems. In Proceedings of the ICC 2022-IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2022; pp. 1–6.
24. Buterin, V.; Hernandez, D.; Kamphefner, T.; Pham, K.; Qiao, Z.; Ryan, D.; Sin, J.; Wang, Y.; Zhang, Y.X. Combining GHOST and Casper. *arXiv* **2020**. [CrossRef]
25. Schwarz-Schilling, C.; Neu, J.; Monnot, B.; Asgaonkar, A.; Tas, E.N.; Tse, D. Three Attacks on Proof-of-Stake Ethereum. *arXiv* **2021**. [CrossRef]
26. Neuder, M.; Moroz, D.J.; Rao, R.; Parkes, D.C. Low-cost attacks on Ethereum 2.0 by sub-1/3 stakeholders. *arXiv* **2021**. [CrossRef]
27. Mišić, V.B.; Mighan, S.N.; Mišić, J.; Chang, X. Decentralization Is Good or Not? Defending Consensus in Ethereum 2.0. *Blockchains* **2024**, *2*, 1–19. [CrossRef]
28. Windley, P.J. Multisource Digital Identity. *IEEE Internet Comput.* **2019**, *23*, 8–17. [CrossRef]
29. Bano, S.; Sonnino, A.; Al-Bassam, M.; Azouvi, S.; McCorry, P.; Meiklejohn, S.; Danezis, G. SoK: Consensus in the age of blockchains. *arXiv* **2017**. [CrossRef]
30. Lundkvist, C.; Heck, R.; Torstensson, J.; Mitton, Z.; Sena, M. uPort: A Platform for Self-Sovereign Identity. White Paper, 2017. Available online: https://whitepaper.uport.me/uPort_whitepaper_DRAFT20170221.pdf (accessed on 22 April 2024).
31. Yun, J.; goh, Y.; Chung, J.M. DQN-Based Optimization Framework for Secure Sharded Blockchain Systems. *IEEE Internet Things J.* **2021**, *8*, 708–722. [CrossRef]
32. Okegbile, S.; Cai, J.; Chen, J.; Yi, C. Shard-based and Reputation-enhanced Byzantine Fault-tolerant Scheme for Secure Data Sharing in Latency-sensitive Computing Services. *TechRxiv* **2023**. [CrossRef]
33. Okegbile, S.D.; Cai, J.; Alfa, A.S. Performance Analysis of Blockchain-Enabled Data-Sharing Scheme in Cloud-Edge Computing-Based IoT Networks. *IEEE Internet Things J.* **2022**, *9*, 21520–21536. [CrossRef]
34. Kokoris-Kogias, E.; Jovanovic, P.; Gasser, L.; Gailly, N.; Syta, E.; Ford, B. Omniledger: A secure, scale-out, decentralized ledger via sharding. In Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–24 May 2018; pp. 583–598.