Future Generation Computer Systems 118 (2021) 392–416

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Performance evaluation of Hyperledger Fabric-enabled framework for pervasive peer-to-peer energy trading in smart Cyber–Physical Systems[☆]

Ankur Lohachab [*], Saurabh Garg, Byeong Ho Kang, Muhammad Bilal Amin

School of Technology, Environments and Design, University of Tasmania Hobart, Tasmania, Australia

A R T I C L E   I N F O

Article history:
Received 23 September 2020
Received in revised form 8 December 2020
Accepted 15 January 2021
Available online 26 January 2021

Keywords:
Blockchain
Hyperledger fabric
Cyber–physical systems
Smart grids
Energy trading
PHEV-to-PHEV
Performance benchmarking

A B S T R A C T

The in-depth collaboration of Cyber–Physical Systems (CPSs) and smart grids constitute the novel paradigm of distributed energy trading, in which computation and process control are managed in an adaptive Peer-to-Peer (P2P) manner. To further strengthen this collaboration, Hyperledger Fabric (HF) can be prominently considered as a mean to implement next-generation secure and intelligent communication. However, implementing real-world applications on this platform may concern performance issues. For the constructive exploration of these issues, initially, we design a novel P2P energy trading framework for improving resource utilization and consequently addressing the impending electricity crisis challenge. Thenceforward, we evaluate the results based on the different system operational parameters for establishing a proof-of-concept. For determining performance bottlenecks and best-configuration, these results are investigated independently by using the Nectar Research Cloud, thereby sustaining scalability. The proposed evaluation approach will largely contribute to determining the system operational-level parameters of enterprise applications that will utilize the HF platform as their communication tool-support. In addition, a benchmark is presented based on the Hyperledger Caliper tool to facilitate application designers and developers in the form of selecting an appropriate implementation model across the two latest stable HF model versions. The illustrative CPS-enabled energy trading scenario corroborates the feasibility of the proposed framework to foster the development of HF-assisted smart P2P energy trading mechanisms.

1. Introduction

Seamless integration of the notion of Cyber–Physical Systems (CPS) with the traditional segregated physical objects for expanding the smart infrastructure offers convenience to the end-users. CPS extends the possibility of explicit integration of computation, sensing, networking, and control functionalities into the traditional physical objects [1,2]. These CPS-enabled objects have the potential to affect various smart city applications, among which smart grids and smart transportation are the paramount ones that are witnessing unprecedented growth over the past few years. Smart grids enabled by CPS provide autonomous and resilient energy trading mechanisms. The concept of smart grid-equipped energy trading can allow independent stakeholders to build a symbiotic relationship through real-time communication among each other [3]. Based on the design considerations, a CPS-enabled energy paradigm can be envisaged because of the following two possible advancements —

- Energy harvesting technologies: The vision of energy harvesting is to ensure sustainable and environment-friendly energy that can be achieved through various techniques, including photovoltaic, piezoelectric, electromagnetic, and thermo-electricity energy harvesting [4,5]. This utilization of energy harvesting can be significantly enhanced with the help of flexible and efficient distributed trading. This contemporary vision of trading harvested energy in a distributed manner can introduce potential benefits to both consumers and producers for addressing the limitations of conventional trading.
- PHEVs-to-PHEVs: In the energy distribution market, efficient use of smart-grids and bi-directional charging technology can harness the potential of energy trading through (Plug-in

☆ This research was supported by Tasmania Graduate Research Scholarship (TGRS), Australia. Furthermore, we gratefully acknowledge the tentative support of the Nectar Research Cloud, a collaborative Australian research platform supported by the Australian Research Data Commons (ARDC), Australia and National Collaborative Research Infrastructure Strategy (NCRIS), Australia.

* Corresponding author.
E-mail addresses: ankur.lohachab@utas.edu.au (A. Lohachab), saurabh.garg@utas.edu.au (S. Garg), byeong.kang@utas.edu.au (B.H. Kang), bilal.amin@utas.edu.au (M.B. Amin).

https://doi.org/10.1016/j.future.2021.01.023
0167-739X/© 2021 Elsevier B.V. All rights reserved.

Hybrid Electric Vehicles) PHEVs [6]. This distributed energy trading system encourages more and more PHEVs to sell their surplus amount of energy or buy energy that meets their commensurate demands. The flexible adaption of microgrids in this approach can develop transformational procedures of energy trading among PHEVs.

State-of-the-art smart grid systems largely rely upon centralized platforms for storing, processing, and governing a vast amount of data, which is used to power organizational functions [7]. However, these centralized networking-assisted solutions encounter a series of vulnerabilities from external and internal actors. Nevertheless, these solutions still largely establish themselves as an underlying urban information infrastructure. On the other hand, next-generation decentralization networking and information techniques revolutionize the possibility of transforming the operational models of CPS [8] and the Industrial Internet of Things (IIoT). Fueled by the necessity in the development of decentralized led support infrastructure, blockchain will provide an opportunity to develop real-time synergized solutions in a smart energy management environment. This technology can be simply interpreted as a distributed and decentralized paradigm of managing information by utilizing the intertwined relationship of data replication and security.

Though the unprecedented hype in the concept of blockchain originated from the notable impact of cryptocurrencies on the global economy; however, the state-of-the-art blockchain platforms are not just limited to cryptocurrencies or financial applications [9]. Integration of smart services has transformed these platforms to encompass multi-sided marketplaces, where distinguished pieces of this technology will collaboratively and ubiquitously upgrade the ability to manage the assets. This will lead to re-architect the current way of how our legal, financial, and social infrastructure work. Thus, by collaborating CPS-enabled energy trading with blockchain, the boundaries of the centralized trading realm will become vague. Thus, the paradigm of distributed Peer-to-Peer (P2P) energy trading can play a constructive role in encouraging the redesign of the existing energy trading principles.

1.1. Motivation and scope

Recently, the emergence of distributed energy trading has attained unprecedented attention from academia and enterprises, which has been highlighted in a number of existing studies [10–19]. Besides, these studies have also proposed numerous solutions for the energy trading environment supported by P2P capabilities that can be grouped according to their research directions. For example, [11,15–18] addressed security concerns while implementing blockchain-enabled energy trading. Some works focus on negotiation, optimal scheduling [14], and crowdsourcing [18] during energy trading between sellers and buyers for enabling cost effectiveness [13] and flexible trading [10]. [12] points out the social welfare problem through incentivizing mechanisms by utilizing consortium blockchain. Compared with these works, the proposed framework can be considered as a unified solution to tackle security concerns, trading flexibility, price optimization, and more importantly addresses the real-world impending electricity crisis problem [20]. Moreover, by conducting a comprehensive analysis of the above-discussed literature work, it is observed that these studies have not evaluated the underlying blockchain performance as they assumed that the blockchain operational parameters do not incur any complexity. It simply means they considered the best-case scenario in which their underlying network performance remains the same, even if there is an increase in the number of nodes, block size, transaction arrival rate, and other system parameters.

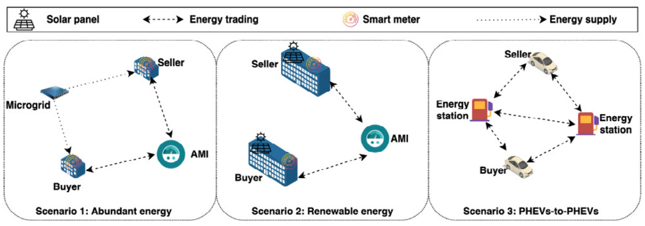


Fig. 1. Common P2P energy trading scenarios involving CPSs.

For addressing this issue, this paper extensively investigates the performance of the underlying blockchain platform to incorporate core capabilities like flexible reconfiguration, quality of service, and ability to analyze the impact of system parameters on performance metrics.

Thus firstly, for selecting an appropriate blockchain platform to implement the proposed framework, we studied various blockchain-based platforms that are used for addressing the continuously growing demand for decentralized applications. Among these platforms, it has been found that the majority are related to cryptocurrency applications and are not able to capture the flexible requirements of enterprise-based applications. Meanwhile, it is claimed that the Hyperledger Fabric (HF) platform is the most desirable platform to target the diverse requirements of the enterprises [21]. Moreover, it is believed that this is an enterprise-grade blockchain due to its intrinsic modular and pluggable capabilities [22]. Though HF may have various advantages over other platforms; however, to explore its capabilities, its performance should be assessed by its conjuncture with real-world applications. Since, its implicit capabilities (such as in contrast to other distributed networked systems, resiliency in HF platform in the context of the specific role of the nodes [23]) can introduce a broad range of implementation and performance complexities.

To analytically study these complexities, there exist few studies that conduct experiments on the HF v0.6 to v1.0 [24–29]. However, these experiments lack in comprehensively analyzing the effect of imperative system parameters on the performance of real-world enterprise applications. Besides, these studies do not explicitly identify the best-suitable configurations and bottlenecks while executing individual transactions according to the transaction workflow. Thus, the system architects face difficulties during the modeling of the applications. It motivates us to examine the feasibility of using the HF platform for establishing communication in the context of smart CPS-enabled P2P energy trading. It allows us to argue on the approaches of determining suitable configuration selection and also, developing the proof-of-concept by describing the relationship between various system parameters and performance attributes. To sum up, this paper broadly formulates two research questions, as summarized in Table 1.

1.2. Research contributions

This work aims to explore the feasibility of collaborating HF platform in an energy trading environment, and unlike focusing on vehicle-to-grid or grid-to-grid transmission, this work studies a novel energy trading scenario supported by CPS-enabled objects. Though there may exist various situations under which energy trading can be performed; however, the common potential scenarios are illustrated in Fig. 1. By considering these scenarios, this paper proposes a unified energy trading framework that primarily addresses the impending electricity crisis and enhances efficient resource utilization. It is done by presenting the near-optimal objective function, where the proximate price

Table 1
Research questions.

Research questions	Rationale
RQ 1: How Hyperledger Fabric can be utilized as an underlying information management platform to develop a real-world P2P energy trading scenario?	This question would seek for implementing a conceptual model using the Hyperledger Fabric platform as an information management system, whereby independent nodes can participate in a P2P energy trading network for secure and transparent interactions. This would foster perspective research in the HF platform-enabled P2P-based enterprise applications.
RQ 2: How does the Hyperledger Fabric key performance and processing attributes behave depending on the system parameters?	The purpose of this question is to seek for the efficient utilization of Hyperledger Fabric within the context of P2P energy trading. The overarching purpose is to analyze the performance issues and optimized configuration of the system to present insights to the application designers and developers.

of the energy can also be minimized based on the Karush–Kuhn–Tucker (KKT) conditions [30]. Besides, this framework will balance the monotonicity of the energy sellers, where they are not able to exploit the consumers based on the increasing demand. Thenceforward, this paper validates the proposed framework and describes the performance bottlenecks and benchmarks. Thus, this paper puts forward the Hyperledger Fabric-based energy trading case study to analyze the feasibility of other real-world enterprise applications. To summarize, the major contributions of this paper are as follows —

- It proposes a novel and secure P2P energy trading conceptual model that can be effectively implemented with private and permissioned blockchain-based platforms. This model allows sellers and buyers to trade the energy without complex procedures (i.e., crowd-sourcing, auction of assets, etc.). Moreover, it addresses the impending electricity crisis problem, thereby balancing the effect of electricity requirements on its pricing.
- It examines the potential of HF platform as an underlying information management system in implementing secure and efficient interactions during energy trading among participating nodes in a distributed and decentralized manner.
- The proposed performance evaluation approach simulates our model by decomposing the trading procedures into distinguishing transactions corresponding to the transaction flow. It assists in analyzing the effect on system performance due to the different HF system parameters during various phases of our model. In addition, this will identify the performance bottlenecks and also estimate the best-possible configuration for the maximum utilization of the system.
- It conducts an informal benchmarking study of the proposed model using the Hyperledger Caliper tool (v0.3.0) on the Nectar Research Cloud for two different stable versions (v1.4.0 and v1.4.1) of the HF platform.

1.3. Organization of the work

The paper organization is as follows. Section 2 provides a brief overview of the background concepts and also highlights the related work to make a comparative analysis with our proposed work. Section 3 introduces the theoretical model to formulate the research problem and its potential solution. In addition, this section introduces the core participating framework components as well as the configuration specification of the underlying information infrastructure. Section 4 presents the working mechanism to determine the operations performed by the participating components. Section 5 discusses the methodology for conducting the experiments. Section 6 discusses the findings by systematically analyzing the simulation results. Section 7 gives discussions by highlighting the answers to the research questions. Finally, Section 8 gives the conclusion, along with future work.

2. Background and related work

To blur the boundaries of the physical world and integrate it with the virtual world, autonomous systems are leveraging the self-aware, self-configuring, and self-managing behavior of the next generation embedded computational sensors and actuators. These incorporated intelligent behaviors build interactive and inter-connected smart pervasive systems (also referred to as smart CPSs), which can be exploited to constantly improve the quality of service in a range of application domains, such as energy, transportation, industrial, and medical sectors. Although correlations among the underlying concept of smart CPSs is akin to current engineering and scientific practices in the domain of Internet of Things (IoT) and embedded systems. However, to reinforce a scalable, robust, and decentralized way of intensive computing, the smart CPSs design approach usually incorporates a higher degree of autonomous components. The essence of these autonomous components enables high-fidelity applications by consolidating multi-dimensional capabilities, including real-time sensing and feedback, dynamic decision-making and control, etc. One such value-creation application is P2P energy trading, which offers innovative ways of automating the trading process without depending on the sophisticated traditional centralized trading environment. By virtue of CPSs, the P2P trading environment envisions effective solutions to maintain core functions at a satisfactory level of operational normalcy. Moreover, through autonomous producers and consumers of energy, next-generation energy delivery networks aim to achieve inseparable and sustainable energy models [36]. Moreover, the notion of Energy Internet [37–43] is closely coupled with the focus on the need for developing next-generation smart-grids to seamlessly facilitate clean and renewable energy. In addition, incorporating various other core concepts at the unit and system level, including smart micro-grids, energy harvesting, Advanced Metering Infrastructure (AMI), PHEV, and distributed energy sources contribute in enhancing the entire trading ecosystem. The self-enforcing features of these concepts as compared to traditional ones can flexibly automate the demand–supply process of collecting, producing, and diagnosing the energy data. To enrich the development of critical information infrastructure of CPSs will require innovative solutions to meet their unprecedented transparency and security demands. Thus, by means of correlating the blockchain-enabled platform and CPSs-enabled energy trading, the resultant integrated environment can lay a foundation for the promising next-generation framework.

In simpler terms, the notion of blockchain provides a secure and immutable way of managing the data within a distributed network that cannot be easily achievable through the traditional concept of distributed networks. Thus, being an innovative P2P decentralized technology, its current supported platforms have

Table 2
Layered description of Hyperledger Project Frameworks.

Framework	Layers			
	Data model layer	Execution layer	Consensus layer	Distributed ledger technology
Fabric [31]	Block consists of header, data, and metadata	Node roles: Client, Peer, and Ordering service	Kafka and SOLO-based algorithms	Comprises of ledger block store and peer transaction manager
Indy [32]	Supports transaction as well as batch of transactions	Node roles: Client, Validator, and Observer	Redundant Byzantine Fault Tolerant-based algorithms	Controlled and managed by Sovrin
Iroha [33]	Supports both pull and push interactions	Node roles: Client, Peer, and Ordering Service	Sumeragi Yet Another Protocol Voting-based algorithms	Introduced the Ametsuchi storage component
Sawtooth [34]	Supports smart contract abstraction. Chain and Blocks are managed by Journal	Node roles: Client and Validator	PoET, DevMode, and RAFT-based algorithms	Uses the on-chain key–value storage
Burrow [35]	Application state consists of name registry, accounts, and the validator set	Node roles: Client, and Validator	Implements Tendermint Consensus engine and supports "pluggable consensus"	Deploys the set of validators using Helm/Kubernetes

various built-in features that include automation, transparency, verifiability, traceability, auditability, immutability, and provenance [44]. Based on the underlying concepts, which primarily include participation authorization in governance services, cryptographic primitives, and consensus algorithms, blockchain can be broadly categorized into two categories — (1) permissionless blockchain, (2) permissioned blockchain [45]. From the security and performance perspectives, permissioned blockchain-based platforms are considered as a preferable option, as trust can be easily established during the consensus mechanism and the threat of potential participants intentionally behaving maliciously is very low [46]. Considering these promising benefits, generally, different organizations may develop their underlying information management infrastructure based on the permissioned blockchains.

There may exist various permissioned blockchain-based platforms that can be utilized for enterprise-based applications, among them HF platform's in-built features seem to provide unparalleled peculiarities for these applications. HF is a framework under the umbrella of Hyperledger (an open-source cross-industry collaborated initiative that focuses on providing the utmost support for the organizations that want to take advantage of the global cross-industry blockchain-based technologies [47]), and it is observed that HF acquires immense attention among researchers and industry stakeholders among the other five frameworks (Fabric, Indy, Iroha, Sawtooth, and Burrow) of the hyperledger family. Moreover, hyperledger also expedites development of these frameworks by provisioning various tools, such as Composer, Cello, Explorer, Caliper, and Quilt [48].

HF is a private and permissioned blockchain framework, which means that users need to register through Membership Service Provider (MSP) to become a part of the framework. Access rights can be restrained to the authorized members so that transactions are not transparent to everyone on the framework [49]. It has no native cryptocurrency; however, various cryptocurrencies can be added according to the requirements of the organization. Similar to the Ethereum Blockchain, smart contracts (in the form of chaincodes) can be incorporated to provide more enhanced functionalities to the applications running on HF. Due to its modular and pluggable enterprise-ready framework, its functionality is not confined to a definite set of industry applications [50]. Table 2 summarizes the features supported by the hyperledger frameworks according to their layered structure.

While designing the blockchain-enabled distributed energy trading framework, two fundamental issues should be considered: (1) how to devise the P2P energy trading conceptual framework? (2) once a framework is determined, how it can be efficiently implemented on the underlying blockchain? By considering these issues, this paper analyzes literature work based on the two categories, (i) methods of designing energy trading framework and (ii) effects of system parameters on blockchain performance metrics.

2.1. Design approaches for distributed energy trading

To highlight recent works on the energy trading realm, we choose a list of studies that implement energy trading by considering various scenarios. Table 3 summarizes the related works focused on P2P energy trading using various blockchain platforms.

In [10], a layered multi-agent coalition system is proposed, in which the first layer has been designed to establish the mechanism of negotiating energy between the consumer and producer network, and the second layer enables the trading settlement mechanism based on the blockchain. Su et al. [11] proposed a secure mechanism for allocating renewable energy among Electric Vehicle (EV) based on the permissioned blockchain. The authors also presented a reputation-based BFT consensus algorithm for their contract-based blockchain. In another similar work [12], the authors proposed a consortium blockchain-based local P2P energy trading model among PHEVs. It enables negotiations of electricity amount and pricing with the assistance of a double auction mechanism in order to maximize the social welfare function.

Paudel et al. [13] proposed a model for P2P energy trading among community-based on the concept of Stackelberg approach in game theory. The authors enabled trading using price competition at two distinctive strategies. Huang et al. [14] proposed a hybrid framework based on the consortium blockchain to enable charging among mobile charging vehicles. Compared to other trading electrical vehicle schemes, the authors considered an additional scenario in which vehicles are able to do charging from mobile vehicles. Moreover, they proposed an improved version of the genetic scheduling algorithm for achieving optimization at the user's cost and satisfaction level.

To address the security and privacy challenges, the authors in [15] proposed a scheme for anonymous energy trading using

Table 3
Summary of related works for P2P energy trading scenario.

Reference	Parameters					
	Environment	Market consideration	Security analysis	Optimization	Infrastructure management	Infrastructure performance analysis
[10]	Electric vehicle	⊙	⊗	✓	PuB	⊗
[11]	Electric vehicle	⊙	✓	✓	PeB	⊙
[12]	Electric vehicle	⊙	✓	✓	CoB	⊙
[13]	Microgrid	✓	⊗	⊗	GTM	⊙
[14]	Electric vehicle	⊙	⊙	✓	CoB	⊗
[15]	Smart grid	✓	✓	⊗	PuB	⊙
[16]	IIoT	⊙	✓	✓	CoB	⊙
[17]	Smart grid	⊗	⊗	⊗	CoB	⊙
[18]	V2G	⊙	✓	✓	CoB	⊙
[19]	Smart grid	✓	⊗	✓	HF	⊙
Our work	CPS	✓	✓	✓	HF	✓

✓— Considered, ⊙ — Partially considered, ⊗ — Not considered, PuB — Public Blockchain, PeB — Permissioned Blockchain, CoB — Consortium Blockchain, HF — Hyperledger Fabric, GTM — Game Theory Model.

blockchain and the concept of multi-signature. More specifically, they addressed the issue of transaction security by enabling encrypted streams for messaging. Another work similar to this is proposed by Li et al. [16], in which the authors utilized the features of consortium blockchain and implemented a credit-based pricing scheme for strengthening the process of trading. To address the challenge of optimization, they introduced an optimal scheme for pricing using the Stackelberg game approach. Gai et al. [17] proposed a scheme for preserving privacy while doing energy trading among neighbors. To restrict privacy leakage (i.e., particularly information about user's energy usage and physical location) the authors used the concept of consortium blockchain for proposing their noise-based approach. Moreover, their mechanism intended to achieve the impression of the differential privacy-preserving algorithm by introducing the counterfeit accounts without degrading the performance. In [18], the authors proposed a secure scheme by integrating the concept of edge computing, blockchain, and contract theory. Their scheme also includes the incentive mechanism and for optimization challenge, they proposed an iterative algorithm. Both Backward Induction and Stackelberg leader–follower game approaches are used for modeling the resource allocation challenge at two-stage. Wang et al. [19] proposed an architecture by enabling P2P energy trading for the management of various types of crowdsourcing and energy distribution scenarios. This approach also supports operations of energy trading from an island microgrids. Moreover, the authors prototyped their proposed architecture for seamless trading using Hyperledger Fabric (HF) platform.

Based on the aforementioned discussion, we investigated that various studies are working on the concept of blockchain-enabled energy trading. Typically, the previous works assume that the performance of the underlying blockchain platform is not degraded. Hence, the most of them did not measure and analyze the performance of the blockchain, as they are primarily focused on achieving optimization in the energy trading scenario. To unleash the enormous potential of the notion of integrating the permissioned blockchain with CPS for enabling P2P energy trading, we comprehensively analyze the performance of the proposed unified framework, which significantly distinguishes our work from the literature. There are various parameters that constitute the substantial groundwork for our work as compared to the previous works, as summarized in Tables 3 and 4.

2.2. Performance evaluation of blockchain platforms

This section highlights the research experiments performed in relation to evaluating and analyzing the performance of various blockchain platforms. In recent multifarious studies, various solutions have been proposed to improve the performance of HF framework. Apart from it, some studies focus on analyzing the underlying technologies of HF. The broader impact or accomplishments that can differentiate these works are summarized in Table 4.

Pongnumkul et al. [24] analyzed the performance of Ethereum and HF by conducting several experiments using various sets of workload, where the obtained results based on the latency and throughput show that HF performs better than Ethereum, as the performance of Ethereum degrades drastically after 10000 Transactions. These tests have been conducted on Ubuntu OS and Amazon Web Service (AWS) Elastic Compute Cloud (EC2) node by switching off the Consensus algorithm and selecting one node from each platform. Sukhwani et al. [25] explored the performance of the consensus process based on Practical Byzantine Fault Tolerance (PBFT). The authors used the Stochastic Reward Nets for modeling the consensus process and computed the meantime for up to 100 peers in the network to complete the consensus algorithm. In this experiment, they created a blockchain network that is focused on HF v0.6 with the help of the IBM Bluemix service.

Kocsis et al. [26] designed a methodology for modeling the performance characterization of the blockchain. The authors demonstrated it on HF v0.6 and found that structured performance analysis is not only decisive to blockchain deployment design, but it also contributes its assessment to software design. Thakkar et al. [27] observed that the endorsement policy verification, state validation and commit, and sequential validation of transactions are the three bottlenecks in the performance of HF. After classifying the bottlenecks in the overall performance of HF, the authors performed optimization techniques and found that implementing optimizations can alleviate these bottlenecks. Nasir et al. [28] conducted performance analysis among HF, v1.0, and v0.6 in terms of throughput, execution time, scalability, and latency, by diversifying the load. The authors found that HF v1.0 persistently outperforms HF v0.6. They also observed that under immense workload conditions, HF v1.0 did not attain a certain level of performance in a traditional database. Baliga et al. [29] describe scalability and performance of HF v1.0 by analyzing the latency and throughput. The authors conducted experiments on HF by tuning the number of channels, chaincodes, and peers.

Androulaki et al. [51] illustrated HF according to its architecture, pronounced aspects, and application programming model. Moreover, the authors analyzed the performance of HF by benchmarking a cryptocurrency. They observed that beyond a block size of 2 MB, the throughput does not significantly improve and in fact, latency increases. They found that in certain conditions, HF can accomplish throughput of more than 3500 Transactions Per Second (TPS). Sousa et al. [52] analyzed that HF 1.0 was launched without the BFT ordering service. Hence, the authors designed,

Table 4
Summary of prior works for Hyperledger Fabric framework.

Author et al. [Ref]	Year	Brief summary	Scope				
			Framework	Benchmarking	Performance analysis	Applications	Blockchain
Pongnumkul et al. [24]	2017	Performance evaluation for two different private blockchain platforms through variation in the number of transactions	⊗	⊗	✓	⊗	Ethereum and HF
Sukhwani et al. [25]	2017	Modeling the performance of consensus process based on PBFT, and focusing on HF v 0.6	⊗	⊗	⊙	⊙	HF
Kocsis et al. [26]	2017	Methodology for modeling the performance characterization of the blockchain	⊗	⊗	⊗	⊗	HF
Thakkar et al. [27]	2018	Identified the performance bottlenecks for HF and investigated the optimization techniques for alleviating these bottlenecks	⊗	⊗	✓	⊗	HF
Nasir et al. [28]	2018	Performance comparison of HF v1.0 and v0.6 with respect to throughput, execution time, scalability, and latency	⊗	✓	✓	⊗	HF
Baliga et al. [29]	2018	Studied the scalability and performance of HF v1.0 by analyzing the latency and throughput	⊗	✓	✓	⊗	HF
Androulaki et al. [51]	2018	Illustrates HF according to various parameters and analyzed the performance of HF by benchmarking a cryptocurrency	⊗	✓	✓	⊗	HF
Sousa et al. [52]	2018	Designed and evaluated an ordering service based on Byzantine fault-tolerant in order to speed-up the transaction processing	⊗	⊗	⊗	⊗	HF
Gupta et al. [53]	2018	Addressed the effective handling of temporal queries on HF framework and modeled for the improvement of the performance of temporal queries	⊗	⊗	⊗	⊗	HF
Shuo Wang [54]	2019	Analyzed the effect of malicious behavior on HF framework and designed the pattern for the malicious behavior	⊗	⊗	⊙	⊗	HF
Xu et al. [55]	2019	Addressed the problem of concurrency transaction conflicts and proposed a novel locking mechanism-based method for optimizing the performance of HF	⊗	⊗	⊙	⊗	HF
Javaid et al. [56]	2019	Analyzed and re-designed the validation phase of HF for performance optimization	⊗	⊗	⊙	⊗	HF
Andola et al. [57]	2019	Analyzed HF framework based on the security perspective, pointed out security vulnerabilities in HF, and proposed solutions to mitigate them	⊗	⊗	⊙	⊗	HF
Our work		Performance evaluation of HF to develop the proof-of-concept for enterprise-based applications	✓	✓	✓	✓	HF

✓— Considered, ⊙ — Partially considered, ⊗ — Not considered.

evaluated, and implemented a BFT service for the Fabric. By including optimization for wide-area deployment, an ordering service was built on top of the state machine consensus/replication library. They performed various experiments on their ordering service and found that their service can process 10000 TPS.

Gupta et al. [53] discussed the problem of effectively handling temporal queries on HF framework. Thereafter, the authors demonstrated two models in which the first model is used to create a copy of every event and then stores events together temporally. The other model flawlessly maintains the number of events and puts tags on metadata. Their results show that after implementing these two models on HF, the performance of handling temporal queries significantly increases. Wang et al. [54] considered malicious behavior to analyze the performance of HF. In this paper, the behavior of malicious patterns is also designed and performance of HF is evaluated based on these faulty patterns. Xu et al. [55] addressed the problem of concurrency transaction conflict and accordingly, proposed a novel method for optimizing the performance of HF. The authors designed a locking mechanism for exploring the transactions that are responsible for creating conflicts at the beginning of the flow. Based on this, they temporally stored the indexes of the conflicting transactions in the ledger in order to optimize the processing. They also demonstrated experiments on three data-sets based on their proposed methods.

Javaid et al. [56] performed a fine-grained analysis on the validation phase of HF. Based on the evaluation, the authors re-designed this phase in order to optimize the performance. The resulting phase simultaneously executes transaction reads and writes to the database, along with their validation. The authors implemented the proposed technique in HF v1.1 and claimed that it can be adopted in the upcoming versions of HF. Andola et al. [57] analyzed HF framework based on the security perspective. The authors pointed out two security vulnerabilities in HF and proposed solutions for them. They found that Denial of Service (DoS) attack can be done on the endorser node in order to degrade the network and wormhole attack can be conducted with the help of the channels. For addressing these attacks, they proposed two different mechanisms. In the first approach, they randomized the endorsing nodes by using the random verifiable function and after randomization, they provided pseudonyms to endorser nodes. For tackling the wormhole attacks, they used anonymous identities for both sender and receiver within a channel. Moreover, they analyzed the effect of DoS attack on HF platform and found that throughput is reduced and latency is increased.

The above-discussed works focus on the problems related to improving relative notions in HF. However, they do not evaluate and analyze HF performance with respect to a real-world scenario. Therefore, we examine a case study-based performance analysis, particularly in the domain of P2P-based energy trading.

3. Peer-to-peer energy trading using hyperledger framework

In this section, first, we discuss the theoretical basis for the proposed Hyperledger Fabric-enabled P2P framework for energy trading (HFPET). After that, we describe the core entities involved in HFPET. Lastly, we briefly determine HFPET underlying conceptual working mechanism.

3.1. Energy management theoretical framework

In this sub-section, we discuss the theoretical framework for energy trading among P2P Smart Energy Nodes (SENs) to illustrate the problem definition of resource utility management. This problem is driven by a set of factors augmented in the expeditiously impending energy crisis [20] around the world in the recent decade. The crucial aspect that deals with these factors is market design, and perhaps highlights the scarcity of the resources through the difference between energy supplied and energy demanded. Another approach that focuses on the electricity market perspective is presented in [12], which addresses the social-welfare problem using the KKT conditions [30]. However, this work did not focus on the impending electricity crisis problem that is primarily addressed in this paper. Moreover, the authors did not mention which consortium blockchain platforms they utilized for the experimentation. The detailed description of the resource utility management problem and the functions used to describe this problem are discussed as follows —

3.1.1. Energy fulfillment profiling

In an electricity trading scenario, SENs are able to establish a real-time connection with Microgrid (MG) that will facilitate as the mediator between the seller SENs and buyer SENs. A set of MGs (denoted as κ) is indexed by x, where x $\in \kappa = \{1, 2, 3, \dots, x\}$. Moreover, set of seller SENs are denoted by $\Gamma = \{SS_i \mid i \in \mathbb{N}\}$, and set of buyer SENs are denoted by $\gamma = \{BS_j \mid j \in \mathbb{N}\}$. $BS_j(\mathbb{ER}^{max})$ and $BS_j(\mathbb{ER}^{min})$ are the maximum and minimum electricity requirement of BS_j, respectively. Simultaneously, we denote the maximum and minimum requirement of electricity for SS_i as $SS_i(\mathbb{ER}^{max})$ and $SS_i(\mathbb{ER}^{min})$, respectively. The electricity demand of an individual BS_j is denoted as $\eth$ and the total electricity demand is denoted by $\mathbb{D} = \sum_{j=1}^{n} \eth_j$, from κ. We define before and after energy state of BS_j Energy Buffer (EB) as ST_j^B and ST_j^A, respectively. The energy state of SS_i(EB) before selling is denoted by ST_i^B, and after selling the energy as ST_i^A. BS_j^{Cap} and SS_i^{Cap} are capacity of BS_j(EB) and SS_i(EB), respectively. The amount of the electricity supply by SS_i is denoted as $\mathbb{E}^{SU}$ and the total energy supply is denoted as $T(\mathbb{E}^{SU}) = \sum_{i=1}^{n} \mathbb{E}_i^{SU}$. Thus, the impending electricity for an individual BS_j is

$$E_{Imp}^{BS_j} = (\eth - \mu(\mathbb{E}^{SU})), \forall i, t \in \mathbb{N}, \ and\ \mathbb{E}^{SU} \le \eth \tag{1}$$

where μ represents the average efficiency from SS_i, and the total amount of impending energy is

$$T(E_{Imp}) = (\mathbb{D} - \mu(T(\mathbb{E}^{SU}))) \tag{2}$$

Thus, based on the conditions mentioned in (1) and (2) the problem of resource utility management from the perspective of impending electricity can be expressed as follows —

$$RF(E_{Imp}^{BS_j}) : min \sum_{i=1}^{n} \sum_{j=1}^{n} (\eth_j - \mu(\mathbb{E}_i^{SU})) \tag{3}$$

subject to:

$$SS_i(\mathbb{ER}^{max}) \le SS_i^{Cap} \tag{4}$$

$$\eth \le BS_j(\mathbb{ER}^{max}) \tag{5}$$

$$ST_i^B - \mathbb{E}^{SU} \ge SS_i(\mathbb{ER}^{min}) \tag{6}$$

The problem of market equilibrium or economic equilibrium [58] is also indirectly pointing towards the resource utility management problem, as mentioned in (3). For the resource utility management problem, κ must obtain the relevant information regarding the energy demand and how much electricity seller SENs want to supply. The issue of SENs information security is addressed through sharing the needed account information of the participating SENs to the κ, as mentioned in the Algorithm 3. Moreover, the design mechanism to solve the problem mentioned in Eq. (1) and to achieve the balancing effect of market equilibrium have to consider the correlation between the requirement and price functions, as discussed in the following sub-section.

3.1.2. Energy requirement profiling

For maintaining the balance between price and requirement in a decentralized electricity supply-chain, the function should optimally balance these influences. From this perspective, we will derive a function (RF) that will estimate the price given the requirement. Now, the mathematical expression for the total energy supply $T(\mathbb{E}^{SU})$ can be derived from the relationship between the price and the demand of the electricity. A typical supply function can be expressed as follows —

$$P = h(D), where\ D = \mathbb{D} \tag{7}$$

In (7), price (P) is a function (h) of demand (D). In Fig. 2, the top left-most quadrant shows that if the price of the electricity becomes high, then the demand for electricity goes low, and vice versa. Moreover, we can easily determine the remaining demand in the chain through the requirement. Hence, the Demand-Left function can be expressed as:

$$D = g(R), where\ R = \mathbb{T}_R(BS_j) \tag{8}$$

In (8), D is a function (g) of requirement (R). The total electricity requirement can be represented $\mathbb{T}_R(BS_j)$. It can be calculated as $\mathbb{T}_R = \sum_{j=1}^{\mathbb{N}} BS_j\ (\mathbb{ER})$. This function will help in calculating the remaining demand on the basis of the requirements generated in the supply-chain. In Fig. 2, it can be easily noticed that when there is no requirement left in the system, then the demand also becomes zero. By comparing the condition given in (7) and (8), we can derive a new function (i.e, where price is a function of requirement) that will help to determine the price with respect to a particular electricity requirement. This function can be expressed as follows –

$$P = f(R) = h(g(R)) \tag{9}$$

This function is shown in Fig. 2 in the right-most quadrant, and it can be seen that we can easily find out the price to charge at any particular requirement, by assuming that we know the amount of demand that is yet to be fulfilled. The problem of resource utility management from the perspective of electricity requirement can be expressed as follows -

$$RF(\mathbb{T}_R(BS_j)) : min f(R) \tag{10}$$

Subject to:

$$g(R) \leq BS_j(\mathbb{ER}^{max}) \tag{11}$$

In other words, the primary objective given in (10) can be stated as follows – It is necessary to find the optimal requirement for which the price can be minimized. In (10), it can be observed that the objective function is strictly convex and smooth function. Therefore this problem of optimality can be solved using the KKT conditions [30]. Let us denote the objective point is $\bar{R}$, and it is known that we have to minimize the objective function. Hence, based on the KKT conditions, if $\bar{\mathbb{R}}$ is global minimizer, then there exists

$$\triangledown f(\bar{R}) + \sum_{i=1}^{m} \bar{\lambda}_i \triangledown g_i(\bar{R}) = 0 \tag{12}$$

$$\bar{\lambda}_i \triangledown g_i(\bar{R}) = 0 \ \forall i \tag{13}$$

$$g_i(\bar{R}) \leq 0 \ \forall i \tag{14}$$

$$\bar{\lambda}_i \geq 0 \ \forall i \tag{15}$$

For the experimentation purpose, in the reference framework, we assume the value of i = 1, so by putting this value in (12) to (15), the approximate optimal value of the requirement can easily be derived. The resultant value will be placed in the conditions mentioned in the Algorithm 4.

Remark. If the BS_j wants to find out the optimal requirement, they have to calculate the energy demand that is left and is yet to be fulfilled.

3.2. Overview of HFPET system components

In the P2P energy trading scenario, operations must be allowed ubiquitously to enhance sustainability in the energy demand–supply system. To establish a sustainable smart energy trading ecosystem, we propose a secure and unified model as shown in Fig. 3. Based on the design objectives, the main modules are discussed as follows –

- Microgrid: Microgrid with unified control has the responsibility of providing an array of services, including buffer energy storage, energy demand fulfillment, and energy buying/selling coordination. The properly designed mechanism help microgrids to play its different role during bi-directional trading. It can choose the role based on the request, such as it can act as an energy absorber in the case when SEN(s) wants to sell their abundant energy. In the energy demand scenario by satisfying the optimized amount of energy, it acts as an energy provider.

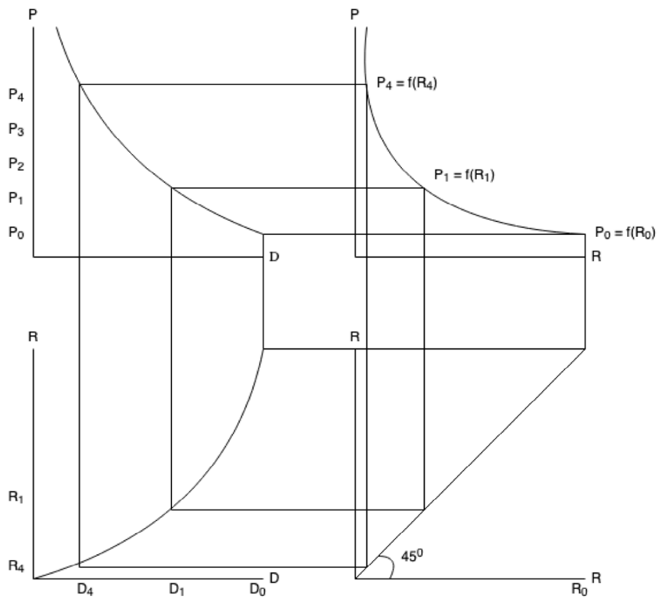


Fig. 2. Determining price for any particular requirement.

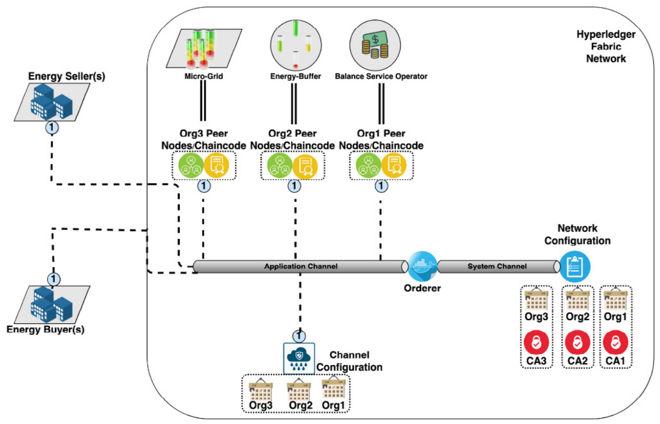


Fig. 3. Reference scenario for our proposed HFPET.

Algorithm 1: Ledger-Query interaction algorithm

```
   Result: $TP^i_{\Re}$
1  Initialization: $A_k$ generates the required $TP^j_{\Im}$;
2  while $A_k$ do not get the corresponding result of $TP^j_{\Im}$ do
3      Send $RQ^l_C$ & $TP^j_{\Im}$ to $GP_m$;
4      if $A_k(UID) ==$ Authenticated && $TP^j_{\Im}$ = ! Empty then
5          $GP_m$ distributes $TP^j_{\Im}$ to $EP^s_n$ according to $SC_p$;
6          $EP^s_n$ independently simulates $TP^j_{\Im}$ through invoking
           the corresponding $CC_r$;
7          $EP^s_n$ retrieve the values from $WS_q$ of the specific $L_q$;
8          $EP^s_n$ generates the signed results from the retrieved
           values for $TP^j_{\Im}$;
9          $GP_m$ returns the signed results to $A_k$;
10         $A_k$ retrieves $TP^i_{\Re}$ and reads the required value;
11         Request succeeded;
12     else
13         Request failed;
14     end
15 end
```

Algorithm 2: Ledger-Update interaction algorithm

```
Result: $B_f^C$
1  Initialization: $A_k$ generates the required $TP_{\Im}^j$;
2  while $A_k$ do not get the corresponding event of $T_x$ do
3      Send $RQ_C^l$ & $TP_{\Im}^j$ to $GP_m$;
4      if $A_k(UID)$ == Authenticated && $TP_{\Im}^j$ = ! Empty then
5          $GP_m$ distributed the $TP_{\Im}^j$ to $EP_n^s$ according to the $SC_p$;
6          $EP_n^s$ independently simulates $TP_{\Im}^j$ through invoking the corresponding $CC_r$;
7          $EP_n^s$ retrieve the values from $WS_q$ of the specific $L_q$;
8          $EP_n^s$ generates finite number of $TP_{\Re}^i$ from the retrieved values for $TP_{\Im}^j$;
9          $GP_m$ returns the $TP_{\Re}^i$ to the $A_k$;
10         while $A_k$ is able to receive $TP_{\Re}^i$ do
11             if $n(TP_{\Re}) == \tau(TP_{\Re})$ then
12                 $A_k$ sends $TP_{\Re}^i$ to $O_a$;
13                 $O_a$ puts $TP_{\Re}^i$ into $SO_y$ based on the implementation specified in $SC_p$;
14                 $O_a$ packages the batches of $TP_{\Re}^i$ into new $B_o$;
15                 if $BT_S(B_o) == D_S(B_o) \parallel t > \tau(ET)$ then
16                     $O_a$ distributes the copy of $B_o$ to the directly connected $P_g$;
17                     $P_g$ cascade $B_o$ to other member nodes of $P_g^s$ by utilizing the gossip protocol;
18                     $P_g$ will verify and validate every transaction in $B_o$ according to $SC_i$ and $SO_y$;
19                     if $\exists\ T_x \in O_y$ && $T_x == VT_x$ then
20                         $P_g$ will update $L_q$ ;
21                         $P_g$ will generate an appropriate $B_f^C$ to achieve target $UL_q^E$;
22                         $P_g$ will generate $B_h^{TE}$ and notify the same to $A_k$, if they are registered for the block event notification type;
23                     else
24                         $P_g$ will retain $FT_x$ for audit purpose;
25                         Execute Step 22;
26                     end
27                 else
28                     Return to Step 15;
29                 end
30             else
31                 Return to Step 11;
32             end
33         end
34         Request Succeeded;
35     else
36         Request Failed;
37     end
38 end
```

- Energy buffer: We assume that every SENs have built-in meters that together build AMI, and here energy buffer work as a record of the image of the energy of the SEN(s). Both seller and buyer SEN(s) can query their energy status and then accordingly, secure deals to trade the energy.
- Smart energy nodes: The scope of SENs is not just limited to Plug-in Hybrid Electric Vehicles (PHEVs), which means every individual or group of entities enabled by CPS is allowed to carry out P2P trading. The role of the SEN(s) is specific but not static, as based on the energy state they can choose the role. Moreover, there is no need to update the role separately, since their role is decided based on their request.
- Balance Service Operator (BSO): The proposed model allows incentivizing the nodes that are selling energy and simultaneously, also collects the amount from nodes that are buying the energy. To manage these transactions, BSO maps the amount to the accounts of the individual SEN(s) to complete the corresponding energy trading. This module also includes simplifying and abstracting the purchase/sell operations of the energy, through masking and protecting the data.

3.3. Configuration specification of HF platform

In this sub-section, we discuss the configuration specification of the modules and parameters that are needed for ensuring consistency and provide native support to the proposed model. Accordingly, we categorize them into five broader phases as follows —

- Security primitives initialization: HF Certificate Authority (CA) provides various configuration settings for CA client and server. In the configuration file of CA server, a section namely Certificate Signing Request (CSR) is responsible for managing customized configuration settings. HF also provides a tool, namely cryptogen that is used for generating the cryptographic preliminaries (i.e, signing keys and X.509 Certificates) for the network. In this paper, we use the Elliptic Curve Digital Signature Algorithm (ECDSA) cryptographic algorithm (i.e., specifically ECDSA-with-Secure Hash Algorithm (SHA-512) for generating certificates that adhere to the X.509 standard. The key size for this algorithm is selected as 521 bit with a secp521r1 curve [59]. Cryptogen takes a configuration file as an input and then generates a set of cryptographic parameters for the network topology.

- Certificates generation: Generating the certificates for components for ensuring authenticity is imperative for our model. Therefore, in this phase certificates are generated for various components of the model. In this regard, we utilize HF component, namely Fabric CA that consists of two entities: (i) CA Client, and (ii) CA Server. Fabric CA is responsible for providing various features that include identity registration, certification revocation and renewal, and grant enrollment certificates. Both CA client or Software Development Kit (SDK) can be used for establishing communication with the CA server or cluster of servers via REST APIs. Thus, for our model, we use SDK for establishing communication with the root CA. Since a CA server can consist of more than one CAs (it can be root CA or an intermediate CA); however, based on the demand of the organization, a cluster of servers can be used for handling the requests (as these servers share the same database). Moreover, our model provides scalability in terms of CAs, and HA Proxy endpoint is utilized on the CA server-side for tackling and managing a load of requests.
- Configuration block creation: A tool namely configtxgen is used to generate genesis block, configuration, and anchor peer transaction. Hence, accordingly, we first generate a genesis block for a solo ordering service. After writing the orderer genesis block, we generate a channel configuration transaction for our application channel, namely Energy Channel (EC). This file is broadcast to the orderer while creating the application channel.
- Participant's enrollment: In HFPET, only legitimate entities get the chance to participate and manage the services, and to become a legitimate participant, it should be get enrolled. This enrollment helps the participants to identify themselves to determine the specific permissions over network resources. Initially, we send the enroll call for the admin of every organization to issue an enrollment certificate for them. Then subsequently, after successful completion of this step, these admins can enroll and register other components of the network.
- Network initialization: This phase can be considered as the loading or joining phase, since all required artifacts are already generated in the previous phases. The initial step in this phase is that peer nodes will join the EC. Thereafter, to properly install pre-defined chaincodes on the EC, it is inherently required to install these on every endorsing peer. At last, for initialization or activation of the chaincodes we perform the instantiation of all chaincodes on EC with the help of a flag parameter in which we specify channelID.

4. HFPET working mechanism

In the previous section, we have already discussed the configuration specification for our proposed HFPET. Now, it is intuitive to understand the working mechanism of the proposed HFPET. It considers the scenarios mentioned in Fig. 1 and accordingly proposes two different chaincodes, as illustrated in Algorithms 4 and 5. While designing these algorithms, three scenarios are primarily taken into consideration that include abundant energy, renewable energy, and PHEVs-to-PHEVs. Although both receiver-initiated and sender-initiated energy trading perspectives are designed in a way such that they can effectively incorporate all the three scenarios; however, specifically to accommodate the abundant energy scenario, case 2 in the sender-initiated trading is determined where the sender has an energy level greater than its maximum requirement but within its capacity limit. For the scope of the proposed work, we make an assumption that in the current model, the knowledge for the energy units and corresponding price distribution is not proportional to the real-world scenario. The detailed discussion regarding the functioning of every participating entity corresponding to the proposed algorithms in HFPET is as follows —

- Client Application Initialization: There are three steps executed during this phase, (1) Client applications will be launched with their respective identities, (2) Client applications will be in a ready state for further communication, and (3) Network will acknowledge the Client Application's ready state by sending the received message to them.
- Account Opening: This is a critical phase for our model. In this phase, client applications will create their accounts in the energy buffer, microgrid, and BSO. The procedure of creating the accounts is well described in the Algorithm 3, where a client application will open its account to avail all the functionalities. We are assuming that the client application will send an amount that is not equal to zero and it has some value to send for its energy buffer. For the sake of simplicity in the Algorithm 3, we illustrate only the new account opening operations because in the update phase, the operations will remain almost the same. For opening an account in the microgrid, it is necessary that the client applications must have accounts in both BSO and energy buffer. The reason behind this is that microgrid will keep the information of both the accounts for further operations.
- Receiver-Initiated Trading: In this case, we assume that energy trading will be initiated by the receiver. Here, the receiver should be interpreted as a buyer of the energy and therefore, it is responsible for initiating the energy request in the network. There are following two potential cases that can be deemed under the service of requesting the energy by the buyer -
  - Case 1: In this case, if the energy level of the SEN's energy buffer is less than its minimum requirement, it will initiate the transaction request proposal. Thereafter, when microgrid will receive this request, it will process the request based on the specified procedure in the Algorithm 4. After receiving the energy and the corresponding bill, the buyer node will update its energy buffer and balance account.
  - Case 2: For accommodating the proactive energy requirements, this phase allows the buyers to purchase energy even if they have the energy level above the minimum requirement. Such a dynamic demand environment would foster more trading options and sellers would be more encouraged to sell their surplus energy. Before sending the energy demand transaction proposal, the buyer will calculate the energy demand by considering that the demand should not surpass the maximum requirement, as shown in Eq. (5). Henceforth, the microgrid will allocate energy based on the procedure as explained in Algorithm 4, so that the energy demand–supply chain can be maintained at a satisfactory level.
- Sender-Initiated Trading: In this scenario, SENs will work as energy seller in order to submit their proposal to sell the surplus energy. These nodes can also act as idle nodes, as in some scenarios they do not want to sell their surplus energy. Considering the case where SENs will choose their role as an energy seller, two possible scenarios are discussed as follows —

Table 5
Comparison between various Hyperledger Fabric framework versions.

Features	Fabric 0.6	Fabric 1.0	Fabric 1.4
Consensus algorithm	Solo	Solo, Kafka	Solo, Kafka, Raft
Ordering service	⊗	✓	✓
Channels	⊗	✓	✓
Database	levelDB	levelDB, CouchDB	levelDB, CouchDB
Endorsement policy	⊗	✓	✓

✓— Considered, ⊗ — Not considered

- Case 1: The operating principle for this energy trading distribution scenario depends on the surplus energy, which means that the seller SENs will submit their transaction request proposal, if their energy level is either equal or greater than their maximum energy requirement. Here, seller SENs surplus energy does not refer that their energy level exceed the capacity, which means that their energy level is maximum than their maximum requirement, but within the storing capacity. After completing the energy requirement, the seller SENs will send their transaction request proposal to the microgrid and then, they will get the incentives for the same after successful processing of their request proposal, as illustrated in the Algorithm 5.
- Case 2: To uphold the stability in the energy demand–supply chain, the SENs will submit their selling proposal even if they do not have the surplus energy level. To participate in energy trading, maybe seller SENs do not have the energy level above their maximum requirement, but their energy level should at least meet the minimum energy requirement. After satisfying this condition, it can submit the transaction request proposal and thereafter, the microgrid will read this proposal for examining the condition of minimum energy threshold requirement, as explained in Algorithm 5. This energy threshold requirement is introduced to determine the minimum energy requirement from microgrid in addition to the minimum requirement determined by the seller SENs. It will only accept the proposal for energy storage request if it meets the threshold condition. In the case of upholding the condition, it sends the corresponding incentive (preferably amount) to the same SENs, as explained in Algorithm 5.

5. Experimentation methodology

To check the applicability and feasibility of the proposed model, we simulate it on two different versions of HF. System Under Test (SUT) is simulated on the Nectar Cloud Platform in Ubuntu 18.04 LTS (Bionic) amd64 with 4 VCPU of 16 GB RAM and 30 GB disk space. We manually set up the different organizations for P2P interactions on HF, operated by independent containers. Both networking and computational requirements for HF have been configured as per the objective. To perform testing, the genesis block and the number of participant organizations are configured accordingly. By utilizing the pre-determined configurations, there are twelve rounds conducted for achieving reliable and effective performance test results.

5.1. Preliminaries

In the HF blockchain platform, each transaction can be represented as an update or query operation performed to its ledger world state [21]. Similarly, in our proposed HFPET, involved trading procedures can be determined using query-ledger interaction and update-ledger interaction. The notion of transfer transaction is more related to the channel and its corresponding chaincodes, as they provide a platform for executing the transfer transaction. Since from the perspective of the ledger, in the whole process of asset transfer, it requires only update operation resulting in an update to the ledger world state. Considering these operations to the ledger world state (query and update), our approach to exemplify the energy trading Algorithms 3, 4, and 5 passes through the Algorithms 1 and 2 for understanding the proposed energy trading procedures (account opening, buyer-initiated trading, and seller-initiated energy trading) in a well-defined way. Moreover, this approach, instead of the generic overview of the ledger-query interaction (Algorithm 1) and ledger-update interaction (Algorithm 2), considers their interlinking with the energy trading algorithms, as illustrated in Table 6. In Table 6, it is also to be noticed that the update transaction is the primary operation for all algorithms, and query transaction is relatively operated less frequently. This concept of frequent and infrequent transaction operations is also defined for the sensible design choice of the blockchain data structure [21] and hence, it can be stated that our proposed approach satisfies the default design approach.

Therefore, based on the sequence of the ledger interactions, the proposed HFPET energy trading procedures will be evaluated on two versions of HF (i.e., v1.4.0 and v1.4.1). The primary reason for considering the latest version is that there are various improvements introduced in the latest version of HF, which primarily include raft ordering service for providing a crash fault-tolerant feature to the orderer nodes. The architectures of v0.6, v1.0, and v1.4 possess some basic differences. For instance, in v1.0, the concepts of ordering services, endorsing peers, and committing peers are introduced compared to its predecessor. Further, in v1.4, many improvements have been made regarding logging, operational metrics, and health checks. The concepts of leader peers and anchor peers are introduced in the latest versions. Table 5 summarizes the important differences in v0.6, v1.0, and v1.4. The configuration for the experimental scenario is done in a controlled environment and the initial configuration of the channel is defined in the configuration block (in some cases it is also known as the genesis block). There are various other prerequisites libraries, languages, and tools (cURL tool, docker, docker compose, Go and Python programming language, Node.js, NPM), which we have been installed along with the HF platform. For illustrating the HFPET proof-of-concept and to conduct formal benchmarking, a formally accepted benchmarking tool is used namely, Hyperledger Caliper [60]. The primary reason for selecting this tool is that it provides a detailed performance report by relying on the functioning implementation of the target blockchain frameworks. This analysis assists system architects to design and develop the underlying information framework based on the needs specified by the customers or users.

5.2. Typical evaluation scenario for our proposed framework

To begin with, we consider a controlled HF-enabled information infrastructure to provide P2P energy trading in the CPS environment. The core elements of the typical reference blockchain environment for our proposed HFPET are shown in Fig. 3 and are discussed below -

- Organization: In this typical scenario, there are three organizations that jointly decide to use HF network to sell and buy the electrical energy. This energy trading network will only allow authorized participants for selling and buying suitable energy. Here, organizations are simply referring to the group of members that are generally managed by the

Algorithm 3: Application account opening pseudocode

```
   **Result**: Ensure account generation decision in $HF^N$
1  **Initialization:** $L_s^B = [T_i^I V]$; $L_t^E = [Eb_j^I V]$; $L_u^M = [MG_k^I V]$; $flag_A^0 \leftarrow A_k^\alpha$; $flag_{EB}^0 \leftarrow A_k^\alpha$; $flag_{MG}^0 \leftarrow A_k^\alpha$;
2  $A_k^\alpha$ wants to participate in $HF^N$;
3  **foreach** $A_k^\alpha \in S\{E(A_k^\alpha)\}$ **do**
4      Compute flag[$A_k^\alpha$];
5      **if** $flag_A^0 \leftarrow A_k^\alpha$ && $A_k^\alpha$ intents to open a $BA_k$ **then**
6          $tx_k \leftarrow <A_k^\alpha$(UID), ID($CC_r$), Payload($tx_k$), $t_k$, $A_k^\alpha Sig >$;
7          Payload($tx_k$) $\leftarrow$ <OP, $MD_k$>;
8          OP $\leftarrow$ <IBC>;
9          $MD_k \leftarrow <L_s^B>$;
10         $TP_{\mathfrak{I}}^j \leftarrow < P_{\mathfrak{I}}^j, tx_k, \{K_L \times V_L\}_i^s >$;
11         $TP_{\mathfrak{I}}^j(ID_A) \leftarrow H_i(tx_k)$;
12         $A_k^\alpha$ stores $TP_{\mathfrak{I}}(ID_A)$ and then, sends call Algorithm 2;
13         After receiving $TP_{\mathfrak{R}}^i$ corresponding to $TP_{\mathfrak{I}}(ID_A)$ from Algorithm 2, it will do $flag_A^1 \leftarrow A_k^\alpha$;
14         $TP_{\mathfrak{R}}(ID_A) \leftarrow H_i(TP_{\mathfrak{R}}^i)$;
15         $A_k^\alpha$ stores $TP_{\mathfrak{R}}(ID_A)$ for further communication;
16     **else**
17         Return to Step 3
18     **end**
19     **if** $flag_{EB}^0 \leftarrow A_k^\alpha$ && $A_k^\alpha$ intents to open a $EBA_k$ **then**
20         $tx_k \leftarrow <A_k^\alpha$(UID), ID($CC_r$), Payload($tx_k$), $t_k$, $A_k^\alpha Sig >$;
21         Payload($tx_k$) $\leftarrow$ <OP, $MD_k$>;
22         OP $\leftarrow$ <EBAC>;
23         $MD_k \leftarrow < L_j^E >$;
24         $TP_{\mathfrak{I}}^j \leftarrow < P_{\mathfrak{I}}^j, tx_k, \{K_L \times V_L\}_i^s >$;
25         $TP_{\mathfrak{I}}^j(ID_{EB}) \leftarrow H_i(tx_k)$;
26         $A_k^\alpha$ stores $TP_{\mathfrak{I}}(ID_{EB})$ and then send call Algorithm 2;
27         After receiving $TP_{\mathfrak{R}}^i$ corresponding to $TP_{\mathfrak{I}}(ID_{EB})$ from Algorithm 2, it will do $flag_{EB}^1 \leftarrow A_k^\alpha$;
28         $TP_{\mathfrak{R}}(ID_{EB}) \leftarrow H_i(TP_{\mathfrak{R}})$;
29         $A_k^\alpha$ stores $TP_{\mathfrak{R}}(ID_{EB})$ for further communication;
30     **else**
31         Return to Step 3
32     **end**
33     **if** $flag_A^1 \leftarrow A_k^\alpha$ && $flag_{EB}^1 \leftarrow A_k^\alpha$ &&$flag_{MG}^0 \leftarrow A_i^\alpha$ && $A_k^\alpha$ intents to open a $MGA_k$ **then**
34         $tx_k \leftarrow <A_k^\alpha$(UID), ID($CC_r$), Payload($tx_k$), $t_k$, $A_k^\alpha Sig >$;
35         Payload($tx_k$) $\leftarrow$ <OP, $MD_k$>;
36         OP $\leftarrow$ <MGAC>;
37         $MD_k \leftarrow < TP_{\mathfrak{R}}(ID_A), TP_{\mathfrak{R}}(ID_{EB}), L_u^M >$;
38         $TP_{\mathfrak{I}}^j \leftarrow < P_{\mathfrak{I}}^j, tx_k, \{K_L \times V_L\}_i^s >$;
39         $TP_{\mathfrak{I}}^j(ID_{MG}) \leftarrow H_i(tx_k)$;
40         $A_k^\alpha$ stores $TP_{\mathfrak{I}}(ID_{MG})$ and then send call Algorithm 2;
41         $TP_{\mathfrak{R}}(ID_{MG}) \leftarrow H_i(TP_{\mathfrak{R}})$;
42         After receiving $TP_{\mathfrak{R}}^i$ corresponding $TP_{\mathfrak{I}}(ID_{MG})$ to from Algorithm 2 $flag_{MG}^1 \leftarrow A_i^\alpha$;
43         $TP_{\mathfrak{R}}(ID_{MG}) \leftarrow H_i(TP_{\mathfrak{R}})$;
44         $A_k^\alpha$ stores $TP_{\mathfrak{R}}(ID_{MG})$ for further communication;
45     **else**
46         Return to Step 3
47     **end**
48 **end**
```

multinational corporation, and the size of the organization in terms of the number of members can range from 1 to n, where n ∈ ℕ. Org1 is serving as the management point for the microgrid, which subsequently utilizes two peer nodes for distributing valid blocks. Org2 manages the energy buffer ledger of the individual client application, and automates transactions based on the proposed model. Org3 actively builds and manages the valid account balance ledger for active client applications.

- Peer: In every organization, peers can adopt explicit roles from various perspectives, as described in the typical network topology. For our purpose, we have chosen one peer from every organization to install the chaincode. Hence, from the network topology perspective, three peers are referred to in the role of endorsing peer and six peers perform operations in the role of committing peer. Moreover, for establishing communication among peers of different organizations, we have elected one peer to be an anchor peer.

Algorithm 4: Receiver-initiated energy trading pseudocode

```
Result: $BS_j$ will get the energy
Require: Consider that $BS_j$ has access to $BA_k$, $EBA_k$, $MGA_k$, and $BS_j \in A_k^{\alpha}$;
procedure Local Energy Inquiry
while $BS_j$ do not get the current energy state from $Org_d^{EB}$ about $BS_j(EB)$ do
    $BS_j$ will generate $TP_{\Im}^{j}$ and call Algorithm 1;
    $BS_j$ interprets the $L_t^E$ from $ST_j^B$;
end
end procedure
procedure Energy Demand Request
Call procedure Local Energy Inquiry;
if $L_t^E \leq BS_j(\mathbb{ER}^{min})$ then
    $BS_j$ will send $TP_{\Im}^{j}$;
    to $Org_d^{MG}$ containing $\eth_t$, and $Pr_t$ related to the corresponding $\eth_t$;
    $Org_d^{MG}$ will solve the condition mentioned in (3) and (10);
    $Org_d^{MG}$ will $E_i^{SU}$ to the $BS_j$;
    $Org_d^{MG}$ send $BR_t$ based on $E_i^{SU}$ and then, sends $tx_k$ through locally calling Algorithm 2;
else
    Return to Step 9;
end
if $L_j^E \geq BS_j(\mathbb{ER}^{min})$ && $L_j^E \leq BS_j(\mathbb{ER}^{max})$ then
    $BS_j$ will send request to $Org_d^{MG}$ containing $\eth_t$, and $Pr_t$ related to the corresponding $\eth_t$;
    $Org_d^{MG}$ will query $L_u^M$ by calling Algorithm 1;
    if $MG_k^l V == \eth_t$ then
        $Org_d^{MG}$ will $E_i^{SU}$ to $BS_j$;
        $Org_d^{MG}$ send $BR_t$ based on $E_i^{SU}$ and then, sends $tx_k$ through locally calling Algorithm 2;
    else
        $Org_d^{MG}$ will solve the condition mentioned in (3) and (10);
        $Org_d^{MG}$ will $E_i^{SU}$ to the $BS_t$;
        $Org_d^{MG}$ send $BR_t$ based on $E_i^{SU}$ and then, sends $tx_k$ through locally calling Algorithm 2;
    end
else
    Return to Step 9;
end
end procedure
procedure Energy and Account Update
$BS_j$ gets $E_i^{SU}$ from the $Org_d^{MG}$;
$BS_j$ sends $tx_k$ to its $Org_d^{EB}$ consisting information about $E_i^{SU}$, and then call Algorithm 2;
$BS_j$ sends $tx_k$ to its $Org_d^{BSO}$ consisting information about $BR_t$, and then call Algorithm 2;
end procedure
```

- Channel: The concept of channel represents the mechanism under which components of the network can communicate and transact privately. The configuration of the communication mechanism is defined by the members of the network, and it has been stored in the configuration block. This configuration primarily has three properties that include versioned, permissioned, and hierarchical. In our model, two channels are responsible for handling the transactions — the application channel is configured for handling the transactions coming from application clients, and the system channel is responsible for dealing with the configuration transactions. Peers are responsible for hosting the physical copy of the ledger in a similar manner logical hosting of the ledger is done on the application channel. The network configuration consists of the beforehand agreed policies, although these can be changed at a later stage by configuration transaction, resulting in the generation of a new configuration block. In this way, the application channel provides complete isolation from the system channel.
- Orderer: The role of ordering service is primarily focused on the flow of transactions, and the nodes that are responsible for handling this service are known as orderer nodes. Practically, in the initial phase of the network, the ON will give administrative rights to Org1 based on NC. Org1 updates the HF network for adding Org2 and Org3 as administrator, so that each organization has equal rights. The ON is still primarily running under Org1's administration; however, now Org1, Org2, and Org3 will have equal rights over the network configuration. In contrast to the role played in the network, orderer's role is different in the channel, as it is responsible for distributing and gathering the transaction in the channel, while it acts as a management point for the network.
- Client Applications: The client applications typically represent users that often invoke interaction with our proposed model by sending their transaction proposals. Depending upon the assigned role, the client can be certainly represented as an observing or load-generating client in the network. In our proposed model, the clients are classified

Algorithm 5: Sender-initiated energy trading pseudocode

```
Result: $SS_i$ will be able to sold the energy
1  Require: Consider that $SS_i$ has access to $BA_k$, $EBA_k$, $MGA_k$, and $SS_i \in A_k^{\alpha}$;
2  procedure Local Energy Inquiry
3  while $SS_i$ do not get $ST_i^B$ from $Org_d^{EB}$ about $SS_i(EB)$ do
4      $SS_i$ will generate $TP_{\Im}^j$ and call Algorithm 1;
5      $SS_i$ interprets the $L_t^E$ from $ST_i^B$;
6  end
7  end procedure
8  procedure Energy Store Request
9  Call procedure Local Energy Inquiry;
10 if $ST_i^B \geq SS_i(\mathbb{ER}^{max})$ && $ST_i^B \leq SS_i{}^{Cap}$ then
11     $SS_i$ will send store $TP_{\Im}^j$ to $Org_d^{MG}$ consisting of $\mathbb{E}_i{}^{SU}$;
12     $Org_d^{MG}$ stores $\mathbb{E}_i{}^{SU}$, and send $Amt_t$ along with $BR_i$ for the corresponding $\mathbb{E}_i{}^{SU}$;
13     $Org_d^{MG}$ will sends $tx_k$ through locally calling Algorithm 2;
14 else
15     Return to step 9;
16 end
17 if $ST_i^B \leq SS_i(\mathbb{ER}^{max})$ && $ST_i^B > SS_i(\mathbb{ER}^{min})$ then
18     $SS_i$ will calculate value of $\mathbb{E}_i{}^{SU} \mid ST_i^A \geq (SS_i(\mathbb{ER}^{min}) + \tau(SS_i(\mathbb{ER})))$;
19     $SS_i$ will send store request to $Org_d^{MG}$ consisting of $\mathbb{E}_i{}^{SU}$;
20     $Org_d^{MG}$ stores the $\mathbb{E}_i{}^{SU}$, and send $Amt_t$ along with $BR_t$ for the corresponding $\mathbb{E}_i{}^{SU}$;
21     $Org_d^{MG}$ will sends $tx_k$ through locally calling Algorithm 2;
22 else
23     Return to step 9;
24 end
25 end procedure
26 procedure Energy and Account Update
27 $SS_i$ receives $Amt_t$ from the $Org_d^{MG}$;
28 $SS_i$ sends $tx_k$ to its $Org_d^{EB}$ consisting information about $E_i^{SU}$, and then call Algorithm 2;
29 $SS_i$ sends $tx_k$ to its $Org_d^{BSO}$ consisting information about $BR_t$, and then call Algorithm 2;
30 end procedure
```

Table 6
Description of notable input parameters and operations for the proposed algorithms.

Algorithms	Chaincode methods	Transactions	Frequency	Endorsement policy	Database
Application account opening	Init, createAsset	Update Query Transfer	Six Null Null	AND (‘O1.m’, ‘O2.m’, ‘O3.m’) or OutOf (2, ‘O1.m’, ‘O2.m’, ‘O3.m’)	LevelDB or CouchDB
Receiver-initiated energy trading	Init, getAsset, createAsset, getAssetsFromBatch	Update Query Transfer	Five Two One		
Sender-initiated energy trading	Init, getAsset, createAsset, getAssetsFromBatch	Update Query Transfer	Four One One		

Table 7
Description of configurations utilized in our experimentation.

Performance attributes		Batch timeout (in ms)	Maximum message count	Number of transactions	Transaction arrival rate	Number of clients
Configuration	A	300	100	200	120	10
	B	600	200	400	240	15
	C	900	300	600	360	20
	D	1200	400	800	480	25
	E	1500	500	1000	600	30
	F	1800	600	1200	720	35
	G	2100	700	1400	840	40

as energy buyers and sellers, and they work on behalf of them in the energy supply-chain. Their functionality is defined based on the associated script, and their number often varies based on our pre-defined configuration during experimentation.

- Certificate Authorities: It is responsible for issuing digital certificates that are complied with the X.509 standards. These certificates can be used to manage identities as well as for signing transactions to verify that an authorized organization can endorse transaction proposals. In our setting, we have a custom CA namely Fabric CA that is responsible for handling the entire chain of issuing certificates. MSP configuration sets the basis for deciding the permissions in context of CAs, and it also impersonates an important role in deciding the scope of the component within the organization.

- Consortium: The goal of the consortium is to allow a set of organizations that are having common goals, interact with each other in a private manner, and there can be any number of consortiums based on the requirement. When a new consortium is created, only one organization can work as a full administration and hence, it limits the other organization's authority in this case. Let us consider that in our case, there exists a consortium, namely X that consists of three member organizations. Then according to the NC definition, only Org1 will have the full rights as compared to Org2 and Org3.

6. Results and analysis

In this section, we present the performance observations and analysis obtained from experimentation on the distinctive components of our proposed framework. This evaluation covers various perspectives on the performance of HF in a controlled environment. The data is also compared across two different versions (i.e, v1.4.0 and v1.4.1), and these standard measurements assist in establishing a performance benchmark of HF. Analyzing the behavior of the chaincode cycle will help in understanding the potential impact of the relative performance of different phases of the cycle. These results can be taken into consideration by the stakeholders (more specifically system architect), while designing a framework based on HF.

6.1. System parameters configuration

The important task during the investigation of the performance of HF network is to find out the number of factors that can potentially affect the performance metrics. Therefore, in this subsection, we highlight the number of factors that can drive the performance of our proposed framework. Table 7 lists these factors with corresponding selected values for our framework, and we widely group these values into seven configurations for the experimentation purpose.

- Application Clients: In the context of our proposed framework, clients can be understood as the entities that are responsible for invoking the transactions or initializing the work in the system. The buyer and seller applications deployed in our proposed system that are responsible for interactions have been considered as the clients. The number of clients may relatively influence the choice of other parameters in the blockchain.
- Transaction Proposal and Arrival Rate: Transaction proposal impacts the network during the initial phase. The number of interactions will simply depend upon the number of transaction proposals, and therefore, how the network will perform will likely depend upon the number of the proposals that are sent. Similarly, transactions that are more complex (i.e, update proposal) will generally have a different impact on performance metrics rather than a simple transaction proposal (i.e, query proposal). The transaction arrival rate is very closely related to the number of transaction proposals, since from this rate, it is easy to determine the saturation point for the peer nodes. This rate is a measure of how many transactions are initially handled by the peer nodes within a definite time.
- Batch Timeout and Size: Like any other network where along with the structure, the process also affects the performance of the network, HF performance can also be affected by the interaction of the components of the network. Batch size and batch timeout are the unique parameters in the channel configuration parameters with respect to ordering service, which contribute to the process of ordering service performance. Batch size in itself contains three parameters through which values for the size and number of the transactions in the block can be configured. Maximum Message Count determines the number of transactions that can reside in the newly formed block. Absolute Maximum Bytes (AMB) is used for determining the maximum size of the block and Preferred Maximum Bytes (PMB) is used for the premature formation of the block, which means that it can enable block construction before reaching the maximum size. For our experimentation purpose, we configured the value for the AMB equal to 10 MB and PMB equal to 3 MB.

6.2. Performance analysis

For evaluating the feasibility of the framework, an independent study of its components should be performed based on the key observation points. The study details can reveal the performance characteristics of the proposed system for various environment parameters and workloads. Moreover, the test results might describe the bottlenecks in the system by calling out the relative complexity of the proposed framework. Thus, we observe the performance of the diverse set of interactions to assess the improvement or degradation made by different configurations.

6.2.1. Performance evaluation of chaincode

There are two possible ways through which we can reach the ledger: query and invoke. Chaincode functionalities that are responsible for generating proposal response from the ledger depend upon the type of transaction proposal. In case of retrieving the required information from the ledger, query chaincode function is utilized, and for updating or creating new information in the ledger, invoke chaincode function is applied. The fundamental operations that are related to the chaincode are shown in Fig. 4. Most of the tasks performed during these events are executed before the deployment of the framework. However, invoke and query chaincode interactions with the ledger allow analyzing the performance of the tasks performed by the chaincode through the perspective of the network. Thus, based on these two functionalities various performance parameters are discussed below —

- Throughput: Fig. 5 illustrates the throughput for implementing the query and invoke functions. We denote invoke function by f(n) and query function by g(n), where n is the number of transactions. Then, the throughput is depicted as $f(n) \leq cg(n)$, where $c > 0$, $n \geq n_0$ and $n_0 \geq 10$. Therefore, it can be derived that $f(n) = O(g(n))$, which simply means that the throughput for the query function is much higher than the invoke function regardless of the number of transactions. For our experimentation purpose, we have taken a large range for the number of transactions. Therefore, we have shown the number of transactions on a logarithmic scale. It is observed that when the number of transactions for both of the functions are in the low range (10–100), the throughput is slightly lower than when the number of transactions are in the middle or higher range (greater than 100). Moreover, an increase in the number of transactions causes the rate of difference between the query and invoke

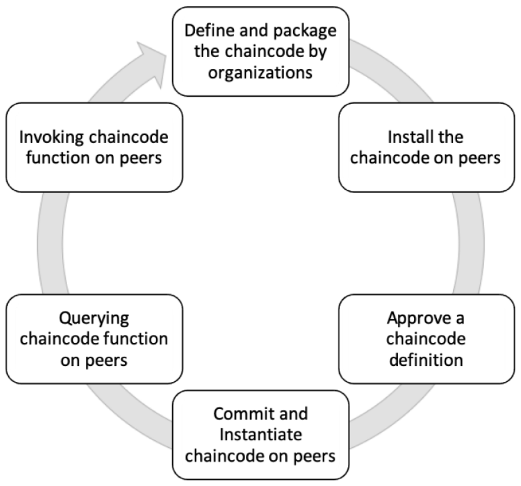

Define and package the chaincode by organizations
Install the chaincode on peers
Approve a chaincode definition
Commit and Instantiate chaincode on peers
Querying chaincode function on peers
Invoking chaincode function on peers

Fig. 4. Fundamental operations related to the Chaincode.

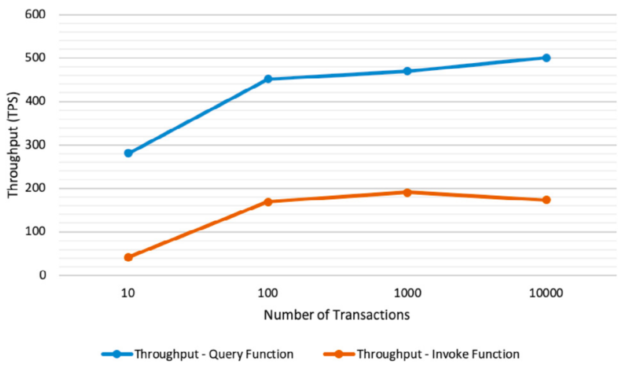


Fig. 5. Throughput for Query and Invoke functions.

function throughput to increase at an arbitrary rate. It is to be noticed that after a certain number of transactions, the throughput for both of the functions is approximately constant. This observation shows that in terms of reliability, HF is able to efficiently handle a large number of transaction proposals.

- Latency: Fig. 6 demonstrates the latency for the invoke and query functions. It can be observed that the latency for executing these functions is overlapping each other. While comparing the number of transactions, it is noticed that the latency for the lower range (10–100) and mid-range (100–1000) of transactions is better, rather than the latency for other ranges of transactions (i.e, greater than 1000). The rate of increase is linear for each individual range; however, if we observe the overall increase, it is approximately exponential. The primary reason behind this exponential behavior is an unexpected increase in the latency after the number of transactions increased from 1000. Hence, it can be concluded that the overall latency is not good in case of a large number of transactions or after a threshold value (i.e, in our case threshold value is 1000).
- Execution Time: Chaincode execution time with respect to query and invoke function through variation in the number of transactions is also analyzed, as shown in Fig. 7. In a way, it is clearly seen that the behavior of the execution time is somewhat similar to the behavior of the latency. Although, if we denote query function by f(n) and invoke function by g(n), where n is the number of transactions, then the execution time complexity is depicted as $f(n) \leq cg(n)$, where $c > 0$, $n \geq n_0$ and $n_0 \geq 1000$. Therefore, it can be derived as f(n) = O(g(n)), which simply means that the time complexity for the invoke function is more than the query function after a threshold value for the number of transactions. In addition after this threshold value, the difference between the rate of increase for the invokes and query functions increases rapidly. In fact, it can be evidently concluded that during large number of transactions, the execution time for the query functions is much better than the invoke function.

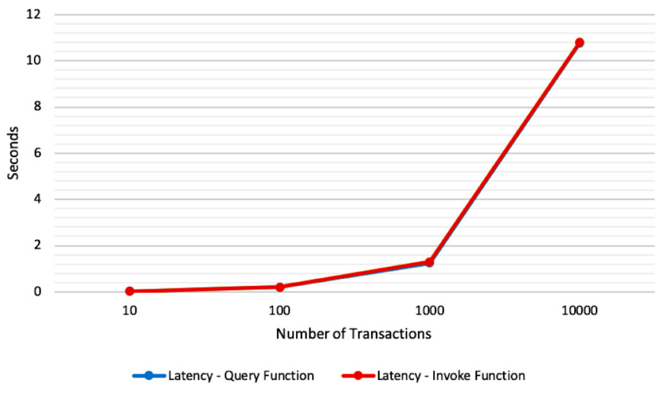


Fig. 6. Latency for Query and Invoke functions.

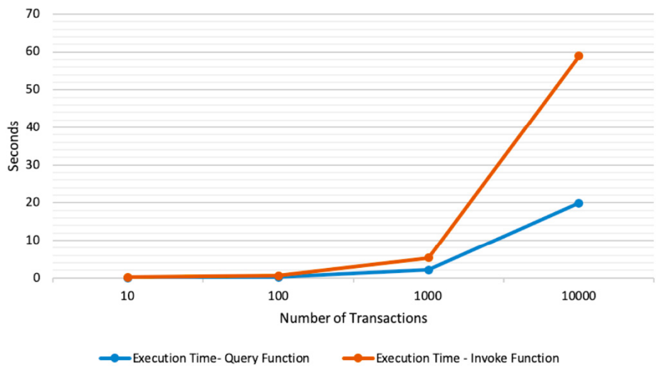


Fig. 7. Execution time for Query and Invoke functions.

After analyzing the above evaluations, it can be evidently seen that the throughput for the different transactions has only slightly changed. However, the variation in the latency and execution time increases at a much rapid rate after a certain threshold value. This analysis gives us a clear idea regarding the behavior of the interactions with the chaincode concerning the different values of number of transactions.

6.2.2. Hyperledger performance benchmarking

In case of HF, there are various informal ways according to which the process benchmarking can take place. However, this can pose challenge among stakeholders as which benchmarking model is suitable for their use case, since models can differ in terms of phases or parameters involved. To address this problem, it uses a standard framework for conducting a benchmark among two different versions of HF (i.e, v1.4.0 and v1.4.1). We utilize Hyperledger Caliper (v0.3.0) as a benchmarking software, as its framework is designed in a way that the testing workload is scalable and can be extended. In simplest terms, the caliper can be understood as a service under which we are able to test a workload on the predefined SUT to analyze its performance. For our case, the workload used for experimentation purposes is given in Table 7. The workload used for benchmarking is very

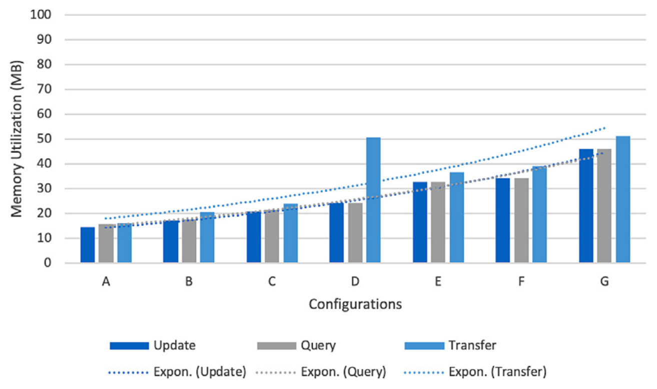


Fig. 8. Memory utilization by Orderer in Hyperledger Fabric v1.4.0 for various configurations (described in Table 7).

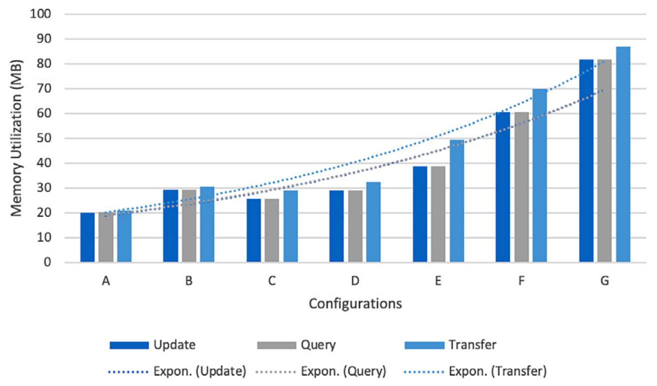


Fig. 9. Memory utilization by Orderer in Hyperledger Fabric v1.4.1 for various configurations (described in Table 7).

close to the actual small production usage, and it can assist in analyzing the approximate behavior of HF. We utilize observers for real-time reporting of data and monitors for collecting statistics regarding the utilization of resources. There are three monitors (i.e., Docker, Prometheus, and Process) that are supported by the caliper. However, as per our requirement, we primarily use a docker monitor. The substantial reason behind this choice is that our experimental setup is hosted on a Cloud platform and we are using containerization for configuring our organization. The docker monitor provides monitoring of docker containers that are hosted remotely. Thus, we define the list of all the containers as an array to the docker monitor that provides significant metrics. Now, we evaluate and compare the results achieved during the experimentation, as discussed below —

- Memory Usage: The memory consumption analysis is conducted to evaluate the allocated memory resources, and more importantly, this analysis helps in identifying the potential memory leakage. During the thorough analysis of the memory consumption of peer nodes belonging to different organizations, we found that they consume approximately an equal amount of resources. Further investigation shows that even during our predefined configurations, they consume an equal amount of energy. To summarize the memory consumption of the peer nodes, we take the cumulative average as per the conditions given in Eqs. (16) and (17).

$$M_{\mathrm{Avg}} = \sum_{i=0}^{n} \frac{(Mem(avg)_{P_i})}{n} \tag{16}$$

where,

$$(Mem(avg)\mathrm{P_i}) = \frac{Mem_{\mathrm{Max}_{P_i}} + Mem_{\mathrm{Min}_{P_i}}}{2} \tag{17}$$

Here,

$P_i \in O_j$

where,

$i, j \in W$

After putting the value of n = 6, the average memory consumption of the peer nodes is 193.3 MB. This clearly depicts that for the real-world application, this network can easily be deployed on the low-cost IoT devices like raspberry pi. For the chaincode and CA workload, consumption metrics show that the average memory consumed by the chaincode is around 6.0 MB, and the CA(s) belonging to all organizations consume 10.0 MB. In the case of the orderer (v1.4.0), the memory consumption metrics depict that with increasing number of transactions or the number of clients, the memory utilization increases relatively, as illustrated in Fig. 8.

Although it is to be noticed that during configuration D, the memory consumption of the transfer algorithm suddenly increases, and corresponding to this increase, there is a sudden decrease in the confirmation probability. The reason behind these behaviors that we found during our analysis is the endorsement policy, since even after consuming resources, it is still not able to produce higher confirmation probability. Comparatively, with the same configuration, we analyze the memory consumption metrics for v1.4.1 and found that in this version, the orderer consumes a higher amount of memory. In Fig. 9, metrics show that with the increase in the number of transactions or the number of clients as some of the prominent parameters among all (see Table 7), there is a gradual increase in memory consumption. After this analysis, it can be concluded that with respect to memory consumption, the performance of v1.4.0 is way better than the v1.4.1.

- CPU Usage: CPU utilization assists in identifying the response time and studying its related analysis. The usage patterns determine in which process the processor remains in the idle state or which process schedule the processor to be in running state. In most of the cases, the process of system diagnosis can be made simpler through the identification of sudden high/low in the CPU utilization or if there is any unusual behavior encountered in the CPU execution. The blockchain operations performed by the peer nodes demand computing power that can be analyzed through CPU utilization. During our investigation, we have found out that the maximum CPU utilization of endorsing peers across all invoking transactions is within the range of (39.9–63.5%), and for the committing peers, CPU usage is within the range of (5%–8%). The execution of the endorsement policy and generating the signed transaction responses is duly responsible for this behavior. Although update transactions also have higher CPU usage; however, transfer transactions still have certainly consumed more CPU usage than query or update transactions. The CPU utilization of both the CA(s) and chaincode(s) is zero, since both of them are not independently running during the processing of transactions. Consequently, the overall CPU utilization of the orderer for v1.4.0 and v1.4.1 does not achieve a significant difference, as shown in Figs. 10 and 11, respectively. In v1.4.0 there seems to be a saturation point for the update transactions after the values of specific parameters increased. On the contrary, in

v1.4.1, the CPU utilization has increased after the value of the specific parameters increased. Despite that, there exists a thin line between these two versions — the overall CPU utilization of v1.4.1 is more than that of v1.4.0 and therefore, we can conclude that in terms of this metric, v1.4.1 is better than its predecessor.
For the purpose of illustrating the empirical comparative analyses of the CPU utilization, a series of experimental simulations have been conducted on the appropriate system configuration. For this experimentation, we selected the configuration (specifically number of records, nature of queries, and three peer nodes for each organization) that is very similar to [61] in order to determine a conclusive and reasonable comparison. Another objective alongside comparative analysis is to evaluate relation between the performance of the CPU corresponding to the gradual increase in the number of records, as CPU performance can potentially affect other system performance metrics. The main point of difference between [61] and our proposed work is that former has utilized the PostgresSQL Database, while we have used the LevelDB database. The evaluation results based on the considered configuration under two different queries (read and write/update) depict that the overall CPU performance of our proposed framework outperforms the framework in [61]. Fig. 12(a) illustrates that the CPU performance of [61] is generally stable; however, the average CPU utilization for the read queries is reasonably higher than the update/write queries. Fig. 12(b) shows that in our proposed framework, peer nodes continuously maintain CPU utilization for both queries, and the results indicate that our proposed framework is more efficient as compared to the framework in [61]. Moreover, based on the comparative analysis, it is noticed that the difference in performance of update/write queries is not that notable; however, for the read queries it is quite considerable. This may be due to the use of the different database, Cloud platform, send rate, arrival rate, or combination of all of them. This ambiguity in determining the exact reason for creating the significant difference in update/write queries performance is because in [61], the authors did not explicitly mention the details of the system parameters.

- Network Usage: Network utilization analysis in context of blockchain can be interpreted as a tool for efficiently achieving the objectives. The amalgamation of this analysis with others can provide a highly intuitive and comprehensive performance feedback. This network analysis is specifically conducted based on the Traffic Input (Traffic In) and Output (Traffic Out) parameters. The statistical analysis shows that whenever we change the configuration, there are no anomalies or abrupt changes detected in the network performance. The effectiveness of the network shows that there is no unsolicited parameters that are able to affect the performance. However, there is a progressive response in the network traffic and this continuous increase is observed as normal behavior. After studying both the versions, we conclude that there is no unusual behavior detected and both the versions show similar behavior in terms of network performance.
- Disk Usage: The disk utilization analysis is the key enabler for independently evaluating the performance of the ledger. This selective analysis is able to derive the ledger size and in some cases, if any potential transactions are prohibitively large, they can be pruned to reduce the size of the ledger. Although this modified transaction may or may not be validated due to the selective pruning. During our experimentation, we noticed that all invoke transactions are generating zero number of bytes (i.e., disk read is zero bytes), as there is no need to perform read operations on the ledger. Another reason is that update and transfer transactions do not explicitly need to read data from the ledger, and query transaction reads data from world state database. Hence, in this case also, there are no reads or writes required from the blockchain. All peer nodes are writing the same number of bytes on the disk that ranges from 328 KB to 8.2 MB. Orderer is writing the number of bytes ranging from 380 KB to 6.5 MB. An important observation that we have made is that the number of bytes written during the update transactions is slightly greater than the transfer transactions. It can be concluded that according to the disk usage performance metrics, both versions (i.e., v1.4.0 and v1.4.1) show almost the same performance. Moreover, based on the analyzed disk reads and writes, it can be said that there is no need to prune the transactions.

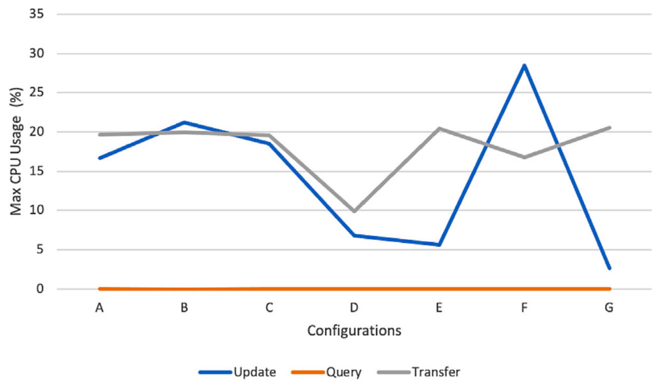


Fig. 10. Maximum CPU utilization of Orderer in Hyperledger Fabric v1.4.0.

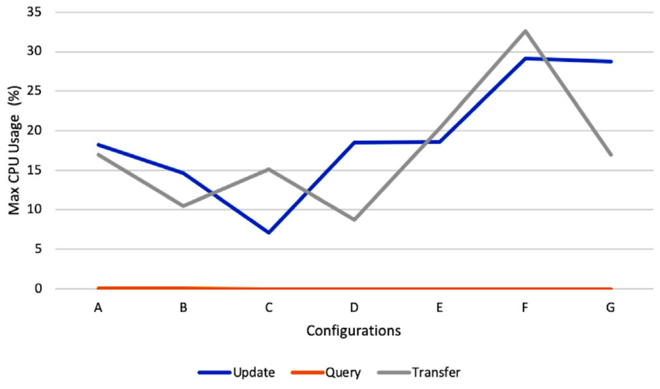


Fig. 11. Maximum CPU utilization of Orderer in Hyperledger Fabric v1.4.1.

6.2.3. *Performance evaluation from different perspectives*

There can be various possible scenarios under which experimentation can be conducted on HF platform. These scenarios are the subject for investigating the possible pathways to utilize the platform. In our case, we consider three realistic scenarios to provide an acceptable level of performance evaluation. In the first scenario, we observe in the context that how HF performs while managing the transactions (i.e., update, query, and transfer) used in our model, as discussed below —

- Ledger Query: We evaluate the performance of the query transaction pathway that tends to target the world state database without interacting with the orderer. The throughput, latency, and achieved send rate for this transaction are

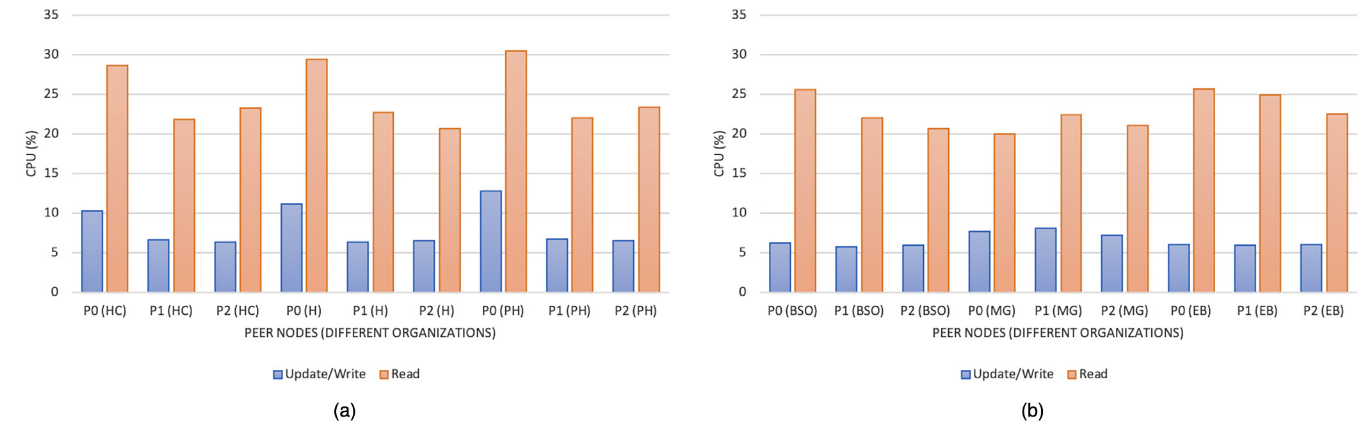


Fig. 12. Comparison of average CPU utilization by organization peer nodes between (a) [61] and (b) our proposed framework.

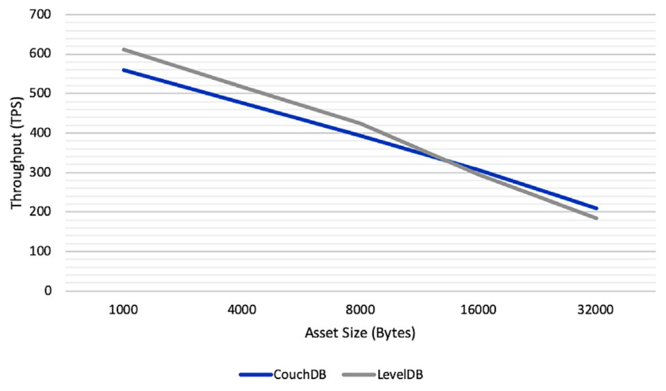


Fig. 13. Throughput of Query transaction within CouchDB and LevelDB-based networks.

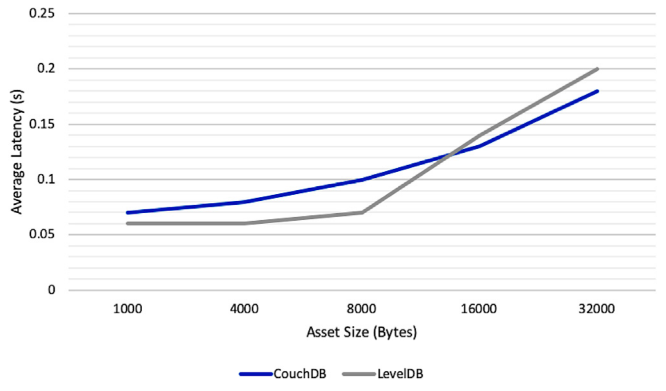


Fig. 14. Average latency of Query transaction within CouchDB and LevelDB-based networks.

highlighted in Figs. 15 and 16 in the context of various configurations. The observations for the network-wide view of this transaction significantly show that the performance of v1.4.0 is slightly better than the v1.4.1. This measurement gives observations, including — there is a linear increase in the average latency and a linear decrease in throughput when the values in the configurations are for a wider network. We noticed that there is a negligible impact of the endorsement policies on the consumed resources, while evaluating the query transaction. Since there is not a very specific difference in the performance of both versions, for further investigation, we have performed experiments on v1.4.0. The performance of the throughput and latency is compared within the context of LevelDB and CouchDB world state databases, as illustrated in Figs. 13 and 14, respectively. This statistical comparison shows that when there is an increase in the size of the asset, using CouchDB as the database rather than level DB database, is certainly increasing the throughput while decreasing the average latency simultaneously. Hence, in this case, the use of CouchDB clearly seems to be beneficial rather than LevelDB.

- Ledger-Update: In this paper, we define the ledger-update transaction as a transaction that is submitted by the application clients through the peer nodes for the world state database. This submission will result in appending the values to the ledger through the orderer. The performance of this transaction based on the investigated configurations is illustrated in Figs. 15 and 16. There is an increase in the size of the database during the successive round of writing the values. Hence, we have further investigated the performance by taking the LevelDB and CouchDB into consideration, as shown in Figs. 17 and 18. The observations show that LevelDB provides higher throughput and lower latency, even if the size of the asset gets increased. Although the benefits of latency will not remain the same when there is a successive increase in the size of the assets. CPU utilization is more in case of the CouchDB; however, it is beneficial in terms of disk read/write. Therefore, after a thorough investigation, we can conclude that LevelDB is slightly beneficial in case of writing or appending the ledger; however, the overall difference is not much appreciable.
- Ledger Transfer: This transaction is responsible for reading the asset value from one ledger and then updating the same value after transferring. In this way, it implicitly utilizes ledger-update and ledger-query transactions. Specifically, for our proposed model, ledger-transfer is not an independent transaction, and therefore, even the validation of this transaction depends upon ledger-query and ledger-update in a pipeline manner. These steps allow to measure the performance of this transaction based on the above analysis. Therefore, the performance analysis can split and then, measured in the phased manner. Since we have already done a detailed performance analysis of both implicit transactions, we can conclude that the speed of confirmed transactions in LevelDB is faster than the CouchDB database considering the security and resource utilization.

In the second scenario, we evaluate the transaction confirmation probability under different configurations as follows —

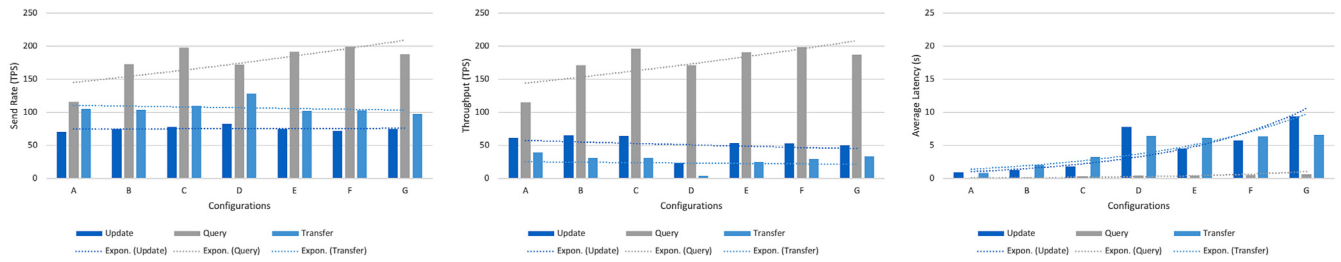


Fig. 15. Send rate, throughput, and average latency investigated for various configurations (described in Table 7) in HF v1.4.0.

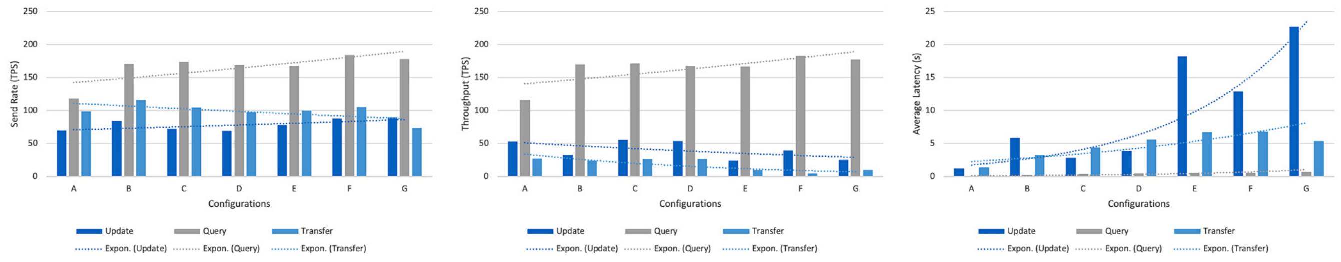


Fig. 16. Send rate, throughput, and average latency investigated for various configurations (described in Table 7) in HF v1.4.1.

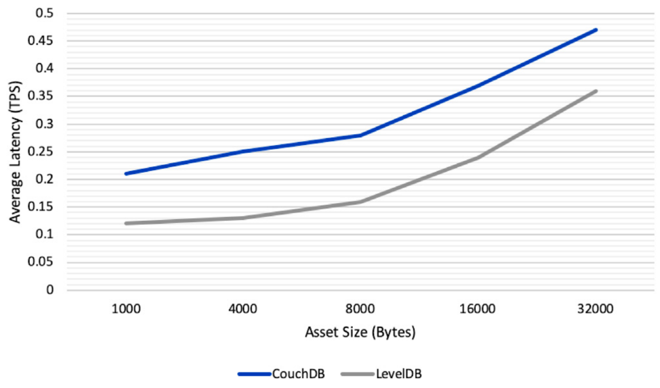


Fig. 17. Latency of Update transaction within CouchDB and LevelDB-based networks.

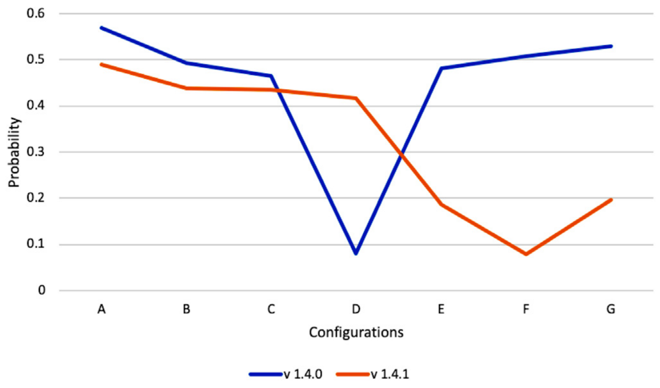


Fig. 19. Confirmed transfer transaction probability.

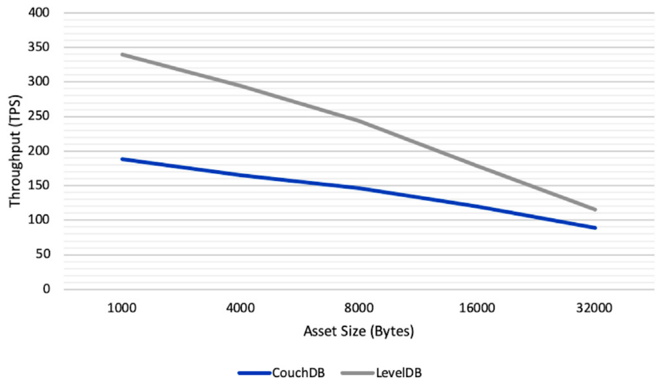


Fig. 18. Throughput of Update transaction within CouchDB and LevelDB-based networks.

- Probabilistic Finality: Probabilistic finality can be seen as a critical representative of transaction confirmation within the scope of lottery-based consensus algorithms. Since in these kinds of networks, the consistency of blocks is predominantly based on the subsequently validated transactions. Therefore, the chaining of the blocks is delayed until the transaction confirmation shows up. Although this metric of chaining blocks can vary according to the frequency or size of the blocks.
A performance test is done, as illustrated in Fig. 19, to measure the percentile of the successful transactions. This measurement shows that with the same configuration, the average transaction confirmation probability of v1.4.0 is higher than v1.4.1. Although there is an unexpected glitch during configuration D in v1.4.0; however, we have explained earlier the reason for the same. There can be several potential reasons behind transaction rejection that generally include syntax errors, consensus errors, and version errors. Among these reasons, it is a complex task to find out the exact reasons behind the transaction failure. However, after doing a thorough analysis, we point out the primary reason for this failure is Multi Version Concurrency Control (MVCC) conflict. HF possess this feature of MVCC to provide resistance against a double-spending attack. However, during our analysis, we find out that this feature can act as a bottleneck, especially in the case of level DB database. The relation between the transaction proposal and the time when this transaction gets stored in the blockchain is non-deterministic. Hence, in time-critical applications, there is a need to find out the solutions to remove this

barrier. One possible solution that we found is the use of the CouchDB database. However, there is a possibility of a double-spending attack in this case.

In the last scenario, to evaluate the overall performance of our model, we also evaluate the simulation results to calculate the blockchain work under considered configurations.

- Blockchain Work: This is a function that assists in the process of understanding the overall performance of the network rather than expressing the performance-focused on a particular parameter. This can be derived through transaction throughput (T_T) and the corresponding network size (N_S). This can be expressed as follows —

$$B_w = f(T_T, N_S) \tag{18}$$

Eq. (18) can be rewritten as -

$$B_w = T_T * \text{Log}(N_S) \tag{19}$$

In this paper, we utilize the function given in Eq. (19) as an insight to characterize the overall blockchain performance. Since in some cases, the throughput of the network is high. However, there is no assurance of the integrity and availability of the network. To analyze this function, we performed experimentation based on our pre-determined configurations and two versions of HF, as shown in Figs. 20 and 21. Moreover, this function is calculated based on the three transactions that include a query, update, and transfer. The detailed assessment demonstrated that overall, HF v1.4.0 outperforms its successor (i.e, v1.4.1). Achievable Blockchain work is investigated for each of the transaction, and in the successive rounds, the configurations are changed dynamically. The observations show that the query transaction has achieved the highest blockchain work than the update transaction and transfer transaction. This characterization also shows that as the number of transactions and client applications get increased, the blockchain work for both update and transfer transactions degrades at a much faster rate. After conducting the detailed analysis, we found that for both update and transfer transactions, validation is required through consensus, and this consensus is directly responsible for low throughput that results in the degradation in the blockchain work. It is to be noticed that in v1.4.0 (Fig. 20) for configuration D, blockchain work is at its lowest level, and after this configuration, again the blockchain work increases. The reason behind this is the change in the endorsement policy, as the endorsement policy for configurations A to D is AND (‘O1.m’, ‘O2.m’, ‘O3.m’), and afterwards, it is OutOf (2, ‘O1.m’, ‘O2.m’, ‘O3.m’). For v1.4.1 (Fig. 21), we did not change any endorsement policy resulting in continuous degradation in the blockchain work. This simply means that when the number of validating nodes decreases, then the throughput of the network increases, and the reliability and availability decrease.

6.3. Security analysis

In our proposed framework, the notion of security is itself injected through the blockchain features. However, we assume that during phases of the transaction propagation, an attacker has the capability to threaten the security of the proposed system. Considering the sensitivity of the data in our proposed framework, we analyze how the threats can be purged. In HF, it is possible to purge the sensitive data (i.e., with the use of transient data store) after the chaincode is successfully executed, and when there is no further need for that particular data. Hence, what is most important is the integration of HF security features with the default blockchain features to provide the expected security and privacy. The following discussion points will show that how our proposed framework is able to achieve the security features for keeping the data private among organizations.

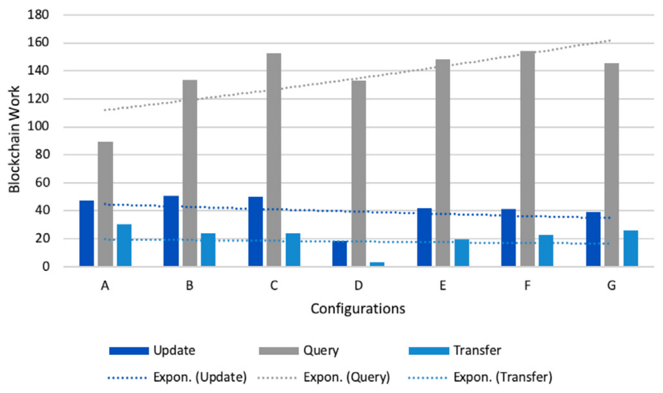


Fig. 20. Amount of work done by HF v1.4.0 for various transactions (depicted in Table 6).

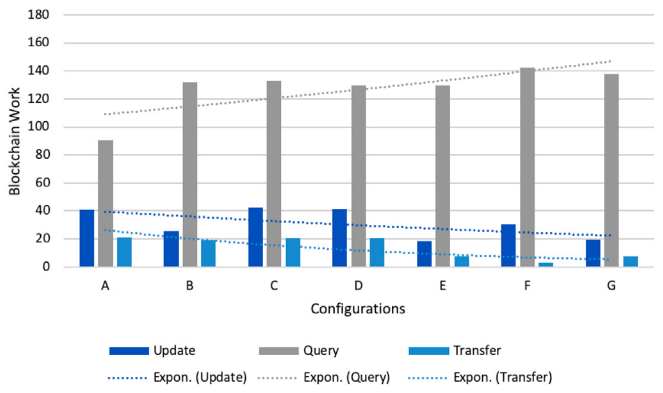


Fig. 21. Amount of work done by HF v1.4.1 for various transactions (depicted in Table 6).

- Transaction proposal: Consider a case where either buyer or seller wants to submit the transaction proposal to either query or update the ledger. Here, no unauthorized party is in the control of reading or writing the sent proposal, since the concept of a channel implicitly decides authorization with the help of defined policies. These channels will also provide privacy and security from an intruder by making transmitted data private. They provide service that enables clients to send their data in the transient field of the transaction proposal.
- Transaction partial response: After successfully endorsing the proposal, the representative peer node will send the response back to the corresponding client. This response is sent using the public data. Now, assume that during this process, an intruder has the capability to see the transaction response. If this intruder will be able to successfully read the data, it can forge the blockchain using the masquerade technique. However, here the data that is being returned is not the same private data that clearly means that only hash of that data key and value will be returned back to the clients. Thus, even if intruders are able to interpret the message, they cannot do any kind of forgery.
- Transaction distribution: The endorsed transactions when submitted back to the orderer for distribution, will still

remain in their private state (i.e., only hash of the private data is revealed). In the validation phase, authorized participant peer nodes of the channel are able to validate without knowing any information about the private data. Thus, privacy can be easily achieved during the internal distribution of the data. Moreover, to provide prevention from the insider attack during processing phases, data is stored in the transient data store of every peer node.

- Transaction committed: The final major phase in a transaction way is the block-committal time, since until a transaction contained in a block gets committed, it will not be considered as a successful transaction. Based on the authorization policy, peer nodes will access the private data that they collected during endorsement time, and if not, they will collect this data from another anchor peer using the gossip protocol. Thereafter, they will commit the data that is successfully verified against the hashes and then, will delete the same data from their respective transient data store. Thereafter, the committed data gets stored in the blockchain and it will be kept in secure storage through replication among nodes.

7. Discussions

CPS-assisted P2P energy trading encourages end-users for remote participation by eliminating the need for trusted third-party to promote the notion of sustainable and renewable energy. Taking this as a motivation, this paper presents the idea of utilizing the private and permissioned blockchain as underlying information infrastructure, in particular Hyperledger Fabric platform. The integration of this platform in the energy trading framework may lead to performance bottlenecks, as various operational parameters may behave differently for varying configurations. Therefore, firstly this paper introduces the energy management theoretical framework for P2P energy trading (Section 3), which addresses the problem of resource utility management considering the factors, including impending electricity, demand, requirement, and price. Subsequently, this paper discusses the role of the core entities (representing real-world CPS-enabled entities) in the proposed HFPET (Section 3.2), along with the corresponding configuration specification (Section 3.3). For connecting this proposed framework to the underlying HF platform, working mechanism (Section 4) presents various trading scenarios through different algorithms. To conclude this discussion, we recall the research questions presented in Section 1 and provide answers to them as follows —

(1) How Hyperledger Fabric can be utilized as an underlying information management platform to develop a real-world P2P energy trading scenario?
The first step towards creating an information management platform is to design a conceptual framework that allows the exchange of assets in a distributed and decentralized way. This conceptual framework defines the participating entities and simultaneously highlights their respective independent roles. Hence, considering this framework as an initial point of support, the operating and functional approaches set the critical requirements for representing the information infrastructure. The imperative theoretical basis and requirements simplify the complexity of determining the system configuration and transaction flow. The representation of the entities constituting the framework is modeled using the concept of organizations in the HF platform that assists in private interactions among them using channels. To scale the network parameters, such as peer nodes and to accommodate the network complexity, an efficient Cloud platform could be utilized. Depending on the end-user's requirement, the configuration of the operational system parameters may vary, thus efficiently improve the collective performance of the information management platform. Most importantly, the empirical proof-of-concept presented in the paper can be considered while developing a similar enterprise-based application using the HF platform.

(2) How does the Hyperledger Fabric key performance and processing attributes behave depending on the system parameters?
To analyze and evaluate the HF platform's overall performance, this work performs experimental simulations under a range of configurations. It is observed that individual or combined configuration parameters directly influence the underlying platform's performance. Moreover, the conducted extensive simulation results identify bottlenecks using various computing performance indicators (such as throughput, average latency, CPU and memory utilization, transaction confirmation probability, and hyperledger workdone). These performance indicators also assist in optimizing the system performance by determining the most suitable configuration parameters for particular applications. In [28], it is observed that the phases involved in ordering service consume more resources as compared to other phases, as this service creates a well-defined sequence of endorsed transaction proposals and packs them accordingly. However, according to our simulation results, it is attainable to minimize the performance bottleneck induced due to the orderer node by customizing its configuration. Moreover, based on our evaluation, it is ascertained that the transaction endorsement and commitment phase impacts performance parameters in various ways that are identified as primary indicators to cause performance bottlenecks. The influence of the request arrival rate of transactions is notable on the overall throughput and latency, as an increase in the arrival rate gradually starts saturating both latency and throughput. These identified performance bottlenecks assist in the development of solutions for removing the likelihood of the substandard consequences during the deployment of HF in P2P applications. In addition, in this work, the results are depicted in a controlled environment; however, the way of analysis can be easily applied to the more complex infrastructure (i.e., with more number of organizations, peer nodes, and channels).

8. Conclusions and future work

In this paper, we propounded the notion of a blockchain-assisted smart P2P energy trading system by proposing a framework that has integrated and incorporated the permissioned and private blockchain with CPS-enabled smart grids. This potential integration will assist in stabilizing and automating the process of P2P energy trading. Thus, we considered an underlying information infrastructure having an HF platform, as it has the potential to establish seamless communication among distributed and decentralized CPSs. Compared with literature studies (as described in Section 2), the proposed approach is more simplistic; however, it provides flexibility to end-users to independently participate in the energy trading process. This framework decouples the impending electricity crisis challenge by proposing a near-optimal solution through determining objective function (based on the KKT conditions) where sellers are not able to exploit the consumers according to the increasing electricity demand. Thereafter, this paper analyzed the effect of different configurations

Table 8
Notations used for various elements in our model.

Notation	Brief description
κ	Set of microgrids
Γ	Set of seller SENs
γ	Set of buyer SENs
SS_i	Seller SENs
BS_j	Buyer SENs
$SS_i(\mathbb{ER}^{max})$	Maximum energy requirement of SS_i
$SS_i(\mathbb{ER}^{min})$	Minimum energy requirement of SS_i
$BS_j(\mathbb{ER}^{max})$	Maximum energy requirement of BS_j
$BS_j(\mathbb{ER}^{min})$	Minimum energy requirement of BS_j
$\backepsilon$	Electricity demand of an individual BS_j
$\mathbb{D}$	Total energy demand of γ
ST_j^B, and ST_j^A	Before and after energy state of the BS_j(EB), respectively
ST_i^B, and ST_i^A	Before and after energy state of the SS_i(EB), respectively
BS_j^{Cap}	Capacity of BS_j(EB)
SS_i^{Cap}	Capacity of SS_i(EB)
$\mathbb{E}^{SU}$	Amount of electricity supplied by SS_i
$TP_{\mathfrak{R}}^i$	Transaction proposal response, ∀ i ∈ ℕ
$TP_{\mathfrak{I}}^j$	Transaction proposal, ∀ j ∈ ℕ
A_k	User application, ∀ k ∈ ℕ
RQ_C^l	Request connection, ∀ l ∈ ℕ
GP_m	Gateway peers, ∀ m ∈ ℕ
A_k(UID)	User application identity
EP_n^s	Endorsing peer, ∀ n ∈ ℕ
SC_p	Smart contract, ∀ p ∈ ℕ
CC_r	Channel chaincode, ∀ r ∈ ℕ
WS_q	World state, ∀ q ∈ ℕ
L_q	Ledger, ∀ q ∈ ℕ
O_a	Orderer node, ∀ a ∈ ℕ
B_o	Block, ∀ o ∈ ℕ
P_g	Peer node, ∀ g ∈ ℕ
P_g^s	Subset of peer nodes, ∀ g ∈ ℕ
SO_y	Strict ordering, ∀ y ∈ ℕ
τ(ET)	Threshold elapsed time
$\tau(TP_{\mathfrak{R}})$	Threshold transaction responses
$D_S(B_o)$	Desired block size, ∀ o ∈ ℕ
$BT_S(B_o)$	Generated block size, , ∀ o ∈ ℕ
VT_x	Validated transaction, ∀ x ∈ ℕ
B_h^{TE}	Block transaction event, ∀ h ∈ ℕ
B_f^C	Committed block, ∀ f ∈ ℕ
T_x	Transactions, ∀ x ∈ ℕ
FT_x	Failed transactions, ∀ x ∈ ℕ
UL_q^E	Updated ledger, ∀ q ∈ ℕ
HF^N	Hyperledger Fabric network
L_s^B	Account Balance ledger, ∀ s ∈ ℕ
L_t^E	Energy buffer ledger, ∀ t ∈ ℕ
L_u^M	Microgrid ledger, ∀ u ∈ ℕ
$T_i^I V$	Balance transaction initial value, ∀ i ∈ ℕ
$Eb_i^I V$;	Energy buffer transaction initial value, ∀ i ∈ ℕ
$MG_k^I V]$;	Microgrid transaction initial value, ∀ i ∈ ℕ
A_k^α	Seller or buyer user application, ∀ k ∈ ℕ
S{E(A_k^α)}	Set of enrolled applications, ∀ k ∈ ℕ
$P_{\mathfrak{I}}$	Propose message
tx_k	Propose message transaction, ∀ k ∈ ℕ
$\{K_L \times V_L\}^s$	Set of key and value
H_i	One-way hash function, ∀ i ∈ ℕ
Org_d^{EB}	Energy Buffer Organization, ∀ d ∈ ℕ
Org_d^{MG}	Microgrid Organization, ∀ d ∈ ℕ
Org_d^{BSO}	Balance Service Operator Organization, ∀ d ∈ ℕ
BR_t	Balance Receipt, ∀ t ∈ ℕ
Amt_t	Amount, ∀ t ∈ ℕ

Table 9
Acronyms used in this paper.

Acronym	Definition
AMI	Advanced Metering Infrastructure
AMB	Absolute Maximum Bytes
AWS	Amazon Web Service
BSO	Balance Service Operator
CA	Certificate Authority
CPSs	Cyber–Physical Systems
CSR	Certificate Signing Request
DoS	Denial of Service
EBAC	Energy Buffer Account
EB	Energy Buffer
EC2	Elastic Compute Cloud
ECDSA	Elliptic Curve Digital Signature Algorithm
EC	Energy Channel
EI	Energy Internet
EV	Electric Vehicle
HF	Hyperledger Fabric
HFPET	Hyperledger Fabric-enabled P2P framework for energy trading
IBC	Amount Bank Account
IoT	Internet of Things
IIoT	Industrial Internet of Things
KKT	Karush–Kuhn–Tucker
MD	Metadata
MGAC	Microgrid Account
MSP	Membership Service Provider
MG	Microgrid
MVCC	Multi Version Concurrency Control
OP	Operation
P2P	Peer-to-Peer
PBFT	Practical Byzantine Fault Tolerant
PHEVs	Plug-in Hybrid Electric Vehicles
PMB	Preferred Maximum Bytes
RBFT	Redundant Byzantine Fault Tolerant
SDK	Software Development Kit
SENs	Smart Energy Nodes
SHA	Secure Hash Algorithm
SUT	System Under Test
TPS	Transactions Per Second
YAC	Yet Another Protocol

of system parameters on imperative performance metrics of the underlying HF platform. Based on the extensive experimentation, the system bottlenecks are identified and consequently, we determine optimized performance metrics through making a selection among various configurations. In addition, a comprehensive benchmark of the HF platform is depicted through conducting a number of simulations using the Hyperledger Caliper tool on Nectar Research Cloud. The comprehensive analysis of the results established a proof-of-concept for deploying the proposed framework in practical scenarios. It can also be utilized as a typical reference point for developing innovative enterprise applications enabled by permissioned blockchain.

Currently, the proposed model has been simulated in a controlled environment on Nectar Research Cloud. Therefore, as of future work, we are planning to map our proposed model to a real-world operational model, where the framework will be emulated using the capabilities of CPS-enabled smart grids. Moreover, performance of other commercial Clouds would be examined within the context of deploying the HF platform. Furthermore, for practical considerations, real-time energy demand and pricing would be employed, before configuring chaincodes. Besides, the performance analysis of HF will be made with state-of-the-art blockchain platforms in order to comprehensively compare the performance metrics based on their corresponding system parameters.

CRediT authorship contribution statement

Ankur Lohachab: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing - original draft, Visualization. **Saurabh Garg:** Conceptualization, Methodology, Resources, Writing - review & editing, Visualization, Supervision, Project administration. **Byeong Ho Kang:** Conceptualization, Resources, Supervision. **Muhammad Bilal Amin:** Conceptualization, Resources, Writing - review & editing, Supervision, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

The notations and acronyms used in this paper are summarized in Tables 8 and 9, respectively.

References

[1] F.C. Delicato, A. Al-Anbuky, I. Kevin, K. Wang, Smart Cyber–Physical Systems: Toward Pervasive Intelligence systems, Elsevier, 2020.

[2] A.S. Bretas, N.G. Bretas, B.E. Carvalho, Further contributions to smart grids cyber-physical security as a malicious data attack: Proof and properties of the parameter error spreading out to the measurements and a relaxed correction model, Int. J. Electr. Power Energy Syst. 104 (2019) 43–51.

[3] R. Skowroński, The open blockchain-aided multi-agent symbiotic cyber–physical systems, Future Gener. Comput. Syst. 94 (2019) 430–443.

[4] H. Erdem, V.C. Gungor, On the lifetime analysis of energy harvesting sensor nodes in smart grid environments, Ad Hoc Netw. 75 (2018) 98–105.

[5] J. Xu, R. Zhang, Cooperative energy trading in CoMP systems powered by smart grids, IEEE Trans. Veh. Technol. 65 (4) (2015) 2142–2153.

[6] Z. Fan, A distributed demand response algorithm and its application to PHEV charging in smart grids, IEEE Trans. Smart Grid 3 (3) (2012) 1280–1290.

[7] C. Clastres, Smart grids: Another step towards competition, energy security and climate change objectives, Energy Policy 39 (9) (2011) 5399–5408.

[8] H. Li, A. Dimitrovski, J.B. Song, Z. Han, L. Qian, Communication infrastructure design in cyber physical systems with applications in smart grids: A hybrid system framework, IEEE Commun. Surv. Tutor. 16 (3) (2014) 1689–1708.

[9] A. Lohachab, A perspective on using blockchain for ensuring security in smart card systems, in: Research Anthology on Blockchain Technology in Business, Healthcare, Education, and Government, IGI Global, 2021, pp. 529–558.

[10] F. Luo, Z.Y. Dong, G. Liang, J. Murata, Z. Xu, A distributed electricity trading system in active distribution networks based on multi-agent coalition and blockchain, IEEE Trans. Power Syst. 34 (5) (2018) 4097–4108.

[11] Z. Su, Y. Wang, Q. Xu, M. Fei, Y.-C. Tian, N. Zhang, A secure charging scheme for electric vehicles with smart communities in energy blockchain, IEEE Internet Things J. 6 (3) (2018) 4601–4613.

[12] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, E. Hossain, Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains, IEEE Trans. Ind. Inf. 13 (6) (2017) 3154–3164.

[13] A. Paudel, K. Chaudhari, C. Long, H.B. Gooi, Peer-to-peer energy trading in a prosumer-based community microgrid: A game-theoretic model, IEEE Trans. Ind. Electron. 66 (8) (2018) 6087–6097.

[14] X. Huang, Y. Zhang, D. Li, L. Han, An optimal scheduling algorithm for hybrid EV charging scenario using consortium blockchains, Future Gener. Comput. Syst. 91 (2019) 555–562.

[15] N.Z. Aitzhan, D. Svetinovic, Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams, IEEE Trans. Dependable Secure Comput. 15 (5) (2016) 840–852.

[16] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, Y. Zhang, Consortium blockchain for secure energy trading in industrial Internet of Things, IEEE Trans. Ind. Inf. 14 (8) (2017) 3690–3700.

[17] K. Gai, Y. Wu, L. Zhu, M. Qiu, M. Shen, Privacy-preserving energy trading using consortium blockchain in smart grid, IEEE Trans. Ind. Inf. 15 (6) (2019) 3548–3558.

[18] Z. Zhou, B. Wang, M. Dong, K. Ota, Secure and efficient vehicle-to-grid energy trading in cyber physical systems: Integration of blockchain and edge computing, IEEE Trans. Syst. Man Cybern. A 50 (1) (2019) 43–57.

[19] S. Wang, A.F. Taha, J. Wang, K. Kvaternik, A. Hahn, Energy crowdsourcing and peer-to-peer energy trading in blockchain-enabled smart grids, IEEE Trans. Syst. Man Cybern. A 49 (8) (2019) 1612–1623.

[20] R.L. Hirsch, Impending United States energy crisis, Science 235 (4795) (1987) 1467–1473.

[21] Hyperledger Fabric release 2.2, https://hyperledger-fabric.readthedocs.io/en/release-2.2/ledger/ledger.html. (Accessed 20 November 2020).

[22] F. Benhamouda, S. Halevi, T. Halevi, Supporting private data on hyperledger fabric with secure multiparty computation, IBM J. Res. Dev. 63 (2/3) (2019) 3–1.

[23] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, F.-Y. Wang, Blockchain-enabled smart contracts: Architecture, applications, and future trends, IEEE Trans. Syst. Man Cybern. A 49 (11) (2019) 2266–2277.

[24] S. Pongnumkul, C. Siripanpornchana, S. Thajchayapong, Performance analysis of private blockchain platforms in varying workloads, in: 2017 26th International Conference on Computer Communication and Networks, ICCCN, IEEE, 2017, pp. 1–6.

[25] H. Sukhwani, J.M. Martínez, X. Chang, K.S. Trivedi, A. Rindos, Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric), in: 2017 IEEE 36th Symposium on Reliable Distributed Systems, SRDS, IEEE, 2017, pp. 253–255.

[26] I. Kocsis, A. Pataricza, M. Telek, A. Klenik, F. Deé, D. Cseh, Towards performance modeling of hyperledger fabric, in: International IBM Cloud Academy Conference, ICACON, 2017.

[27] P. Thakkar, S. Nathan, B. Viswanathan, Performance benchmarking and optimizing hyperledger fabric blockchain platform, in: 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, MASCOTS, IEEE, 2018, pp. 264–276.

[28] Q. Nasir, I.A. Qasse, M. Abu Talib, A.B. Nassif, Performance analysis of hyperledger fabric platforms, Secur. Commun. Netw. 2018 (2018).

[29] A. Baliga, N. Solanki, S. Verekar, A. Pednekar, P. Kamat, S. Chatterjee, Performance characterization of hyperledger fabric, in: 2018 Crypto Valley Conference on Blockchain Technology, CVCBT, IEEE, 2018, pp. 65–74.

[30] S. Boyd, S.P. Boyd, L. Vandenberghe, Convex Optimization, Cambridge university press, 2004.

[31] Hyperledger Fabric, https://hyperledger-fabric.readthedocs.io/en/release-2.2/blockchain.html. (Accessed 20 November 2020).

[32] Hyperledger Indy, https://indy.readthedocs.io/en/latest/. (Accessed 21 May 2020).

[33] Hyperledger Iroha, https://www.hyperledger.org/use/iroha. (Accessed 21 May 2020).

[34] Hyperledger Sawtooth, https://sawtooth.hyperledger.org/docs/core/releases/latest/. (Accessed 22 May 2020).

[35] Hyperledger Burrow, https://www.hyperledger.org/use/hyperledger-burrow. (Accessed 22 May 2020).

[36] W. Tushar, T.K. Saha, C. Yuen, T. Morstyn, M.D. McCulloch, H.V. Poor, K.L. Wood, A motivational game-theoretic approach for peer-to-peer energy trading in the smart grid, Appl. Energy 243 (2019) 10–20.

[37] K. Mahmud, B. Khan, J. Ravishankar, A. Ahmadi, P. Siano, An internet of energy framework with distributed energy resources, prosumers and small-scale virtual power plants: An overview, Renew. Sustain. Energy Rev. 127 (2020) 109840.

[38] F. Al-Turjman, M. Abujubbeh, IoT-enabled smart grid via SM: An overview, Future Gener. Comput. Syst. 96 (2019) 579–590.

[39] Y. Saleem, N. Crespi, M.H. Rehmani, R. Copeland, Internet of things-aided smart grid: Technologies, architectures, applications, prototypes, and future research directions, IEEE Access 7 (2019) 62962–63003.

[40] Y. Kabalci, E. Kabalci, S. Padmanaban, J.B. Holm-Nielsen, F. Blaabjerg, Internet of Things applications as energy internet in smart grids and smart environments, Electronics 8 (9) (2019) 972.

[41] K. Wang, J. Yu, Y. Yu, Y. Qian, D. Zeng, S. Guo, Y. Xiang, J. Wu, A survey on energy internet: Architecture, approach, and emerging technologies, IEEE Syst. J. 12 (3) (2017) 2403–2416.

[42] S. Chen, H. Wen, J. Wu, W. Lei, W. Hou, W. Liu, A. Xu, Y. Jiang, Internet of Things based smart grids supported by intelligent edge computing, IEEE Access 7 (2019) 74089–74102.

[43] S. Hussain, F. Nadeem, M.A. Aftab, I. Ali, T.S. Ustun, The emerging energy internet: Architecture, benefits, challenges, and future prospects, Electronics 8 (9) (2019) 1037.

[44] K. Leng, Y. Bi, L. Jing, H.-C. Fu, I. Van Nieuwenhuyse, Research on agricultural supply chain system with double chain architecture based on blockchain technology, Future Gener. Comput. Syst. 86 (2018) 641–649.

[45] H.M. Kim, H. Turesson, M. Laskowski, A.F. Bahreini, Permissionless and permissioned, technology-focused and business needs-driven: Understanding the hybrid opportunity in blockchain through a case study of insolar, IEEE Trans. Eng. Manage. (2020).

[46] K.M. Khan, J. Arshad, M.M. Khan, Investigating performance constraints for blockchain based secure e-voting system, Future Gener. Comput. Syst. 105 (2020) 13–26.

[47] N. Lu, Y. Zhang, W. Shi, S. Kumari, K.-K.R. Choo, A secure and scalable data integrity auditing scheme based on hyperledger fabric, Comput. Secur. 92 (2020) 101741.

[48] P. Moriggl, P.M. Asprion, B. Schneider, Blockchain technologies towards data privacy—Hyperledger sawtooth as unit of analysis, in: New Trends in Business Information Systems and Technology, Springer, pp. 299–313.

[49] N. Andola, M. Gogoi, S. Venkatesan, S. Verma, et al., Vulnerabilities on hyperledger fabric, Pervasive Mob. Comput. 59 (2019) 101050.

[50] N. Lu, Y. Zhang, W. Shi, S. Kumari, K.-K.R. Choo, A secure and scalable data integrity auditing scheme based on hyperledger fabric, Comput. Secur. 92 (2020) 101741.

[51] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, et al., Hyperledger fabric: A distributed operating system for permissioned blockchains, in: Proceedings of the Thirteenth EuroSys Conference, 2018, pp. 1–15.

[52] J. Sousa, A. Bessani, M. Vukolic, A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform, in: 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN, IEEE, 2018, pp. 51–58.

[53] H. Gupta, S. Hans, K. Aggarwal, S. Mehta, B. Chatterjee, P. Jayachandran, Efficiently processing temporal queries on hyperledger fabric, in: 2018 IEEE 34th International Conference on Data Engineering, ICDE, IEEE, 2018, pp. 1489–1494.

[54] S. Wang, Performance evaluation of hyperledger fabric with malicious behavior, in: International Conference on Blockchain, Springer, 2019, pp. 211–219.

[55] L. Xu, W. Chen, Z. Li, J. Xu, A. Liu, L. Zhao, Locking mechanism for concurrency conflicts on hyperledger fabric, in: International Conference on Web Information Systems Engineering, Springer, 2019, pp. 32–47.

[56] H. Javaid, C. Hu, G. Brebner, Optimizing validation phase of hyperledger fabric, in: 2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, MASCOTS, IEEE, 2019, pp. 269–275.

[57] N. Andola, M. Gogoi, S. Venkatesan, S. Verma, et al., Vulnerabilities on hyperledger fabric, Pervasive Mob. Comput. 59 (2019) 101050.

[58] Y. Chen, W. Wei, F. Liu, E.E. Sauma, S. Mei, Energy trading and market equilibrium in integrated heat-power distribution systems, IEEE Trans. Smart Grid 10 (4) (2018) 4080–4094.

[59] A. Lohachab, et al., Ecc based inter-device authentication and authorization scheme using mqtt for iot networks, J. Inf. Security Appl. 46 (2019) 1–12.

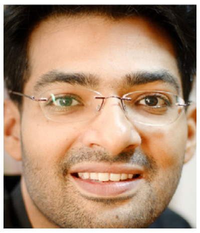
[60] Hyperledger Caliper, https://www.hyperledger.org/use/caliper. (Accessed 18 May 2020).

[61] C. Stamatellis, P. Papadopoulos, N. Pitropakis, S. Katsikas, W.J. Buchanan, A privacy-preserving healthcare framework using hyperledger fabric, Sensors 20 (22) (2020) 6587.

Ankur Lohachab is currently a Ph.D. candidate in the discipline of Information and Communication Technology, School of Technology, Environments and Design at the University of Tasmania. He has been awarded the Tasmania Graduate Research Scholarship for supporting his studies. He received his Master of Technology and Bachelor of Technology degree from the Kurukshetra University, India, in the year 2018, and 2015 respectively. His research interests include security in Blockchain, IoT, Cryptography, Quantum Key Distribution, cloud computing, Cyber–Physical Systems, Security, and privacy techniques. He has published his research work in International platforms of high repute, including Elsevier, IEEE, Springer, and IGI Global. He is actively serving as a reviewer of various reputed journals and conferences.

Saurabh Garg is currently working as a Senior lecturer at the University of Tasmania, Hobart, Tasmania. He had completed his Ph.D. from the University of Melbourne in 2010. He completed his Bachelor of Technology and Master of Technology from the Indian Institute of Technology (IIT) Delhi, India. He has published more than 100 papers in highly reputed journals and conferences including IEEE, Elsevier, ACM, Springer, Wiley, Taylor & Francis, Inder science, etc. He has gained about three years of experience in the Industrial Research while working at IBM Research Australia and India. His area of interests are Distributed Computing, Cloud Computing, HPC, IoT, BigData analytics, and education analytics.

Byeong Ho Kang, computer scientist, is a Professor in School of Technology, Environments and Design, University of Tasmania, Australia. He leads the Smart Services and Systems research group of postdoctoral scientists which has carried out fundamental and applied research in research areas, expert systems, Web Services, SNS analysis and smart industry areas. He has served as a chair and steering committee member in many international organizations and during conferences. He received his Ph.D. from the University of New South Wales, Sydney, in 1996, and has worked as a visiting researcher in the Advanced Research Lab HITACHI situated in Japan. He has also taken part in major research and development projects with industries and research organizations including the Smart Internet Collaborative Research Centre, the US Air Force, Hyundai Steel and many others. His research interests include basic knowledge acquisition methods and applied research in internet systems as well as medical expert systems.



Muhammad Bilal Amin is currently working as a lecturer at the University of Tasmania, Hobart, Tasmania. He had worked as a Postdoctoral fellow in Kyung Hee University, Seoul, South Korea. He has published papers in highly reputed journals and conferences including IEEE, Elsevier, ACM, Springer, Wiley, Taylor & Francis, Inder science, etc. His area of interests are Distributed Computing, Cloud Computing, IoT, Blockchain and semantic web technologies.